



AKDENİZ ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ



Zuhal HAZAR

LEIBNİZ'DEN YAPAY ZEKÂYA: BİÇİMSEL DİLLER ÜZERİNE MANTIKSAL VE
EPİSTEMOLOJİK BİR İNCELEME

Felsefe Ana Bilim Dalı
Doktora Tezi

Antalya, 2022



AKDENİZ ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ



Zuhal HAZAR

LEIBNİZ'DEN YAPAY ZEKÂYA: BİÇİMSEL DİLLER ÜZERİNE MANTIKSAL VE
EPİSTEMOLOJİK BİR İNCELEME

Danışman

Doç. Dr. Mehmet Fatih DOĞRUCAN

Felsefe Ana Bilim Dalı

Doktora Tezi

Antalya, 2022

Akdeniz Üniversitesi
Sosyal Bilimler Enstitüsü Müdürlüğüne,

Zuhal HAZAR'ın bu çalışması jürimiz tarafından Felsefe Ana Bilim Dalı Doktora Programı tezi olarak kabul edilmiştir.

Başkan : Prof. Dr. Süleyman DÖNMEZ (İmza)

Üye (Danışmanı) : Doç. Dr. Mehmet Fatih DOĞRUCAN (İmza)

Üye : Doç. Dr. Şevki IŞIKLI (İmza)

Üye : Doç. Dr. Mustafa KAYA (İmza)

Üye : Dr. Öğretim Üyesi Ali Bilge ÖZTÜRK (İmza)

Tez Başlığı: Leibniz'den Yapay Zekâya: Biçimsel Diller Üzerine Mantıksal ve Epistemolojik Bir İnceleme

Onay : Yukarıdaki imzaların, adı geçen öğretim üyelerine ait olduğunu onaylarım.

Tez Savunma Tarihi : 21/06/2022

Mezuniyet Tarihi : 21/07/2022

(İmza)

Prof. Dr. Engin KARADAĞ

Müdür

AKADEMİK BEYAN

Doktora Tezi olarak sunduđum “Leibniz’den Yapay Zekâya: Biçimsel Diller Üzerine Mantıksal ve Epistemolojik Bir İnceleme” adlı bu çalışmanın, akademik kural ve etik değerlere uygun bir biçimde tarafımda yazıldığını, yararlandığım bütün eserlerin kaynakçada gösterildiğini ve çalışma içerisinde bu eserlere atıf yapıldığını belirtir; bunu şerefimle doğrularım.

Zuhal HAZAR



T.C.
AKDENİZ ÜNİVERSİTESİ
SOSYAL BİLİMLER
ENSTİTÜSÜ



06 /07 / 2022

TEZ ÇALIŞMASI ORJİNALLİK RAPORU BEYAN BELGESİ

Öğrenci Bilgileri	
Adı-Soyadı	Zuhal HAZAR
Öğrenci Numarası	20175232001
Anabilim Dalı	Felsefe
Programı	Doktora
Danışman Öğretim Üyesi Bilgileri	
Unvanı, Adı-Soyadı	Doç. Dr. Mehmet Fatih DOĞRUCAN
Doktora Tez Başlığı	Leibniz'den Yapay Zekâya: Biçimsel Diller Üzerine Mantıksal ve Epistemolojik Bir İnceleme
Turnitin Bilgileri	
Ödev Numarası	1867093426
Rapor Tarihi	06.07.2022
Benzerlik Oranı	Alıntılar hariç: %5 Alıntılar dahil: %12
SOSYAL BİLİMLER ENSTİTÜSÜ MÜDÜRLÜĞÜNE,	
<p>Yukarıda bilgileri bulunan öğrenciye ait tez çalışmasının a) Kapak sayfası, b) Giriş, c) Ana Bölümler ve d) Sonuç kısımlarından oluşan toplam 222 sayfalık kısmına ilişkin olarak Turnitin adlı intihal tespit programından Sosyal Bilimler Enstitüsü Tez Çalışması Orijinallik Raporu Alınması ve Kullanılması Uygulama Esaslarında belirlenen filtrelemeler uygulanarak yukarıdaki detayları verilen ve ekte sunulan rapor alınmıştır.</p> <p>Danışman tarafından uygun olan seçenek işaretlenmelidir: (X) Benzerlik oranları belirlenen limitleri aşmıyor ise: Yukarıda yer alan beyanın ve ekte sunulan Tez Çalışması Orijinallik Raporunun doğruluğunu onaylarım. () Benzerlik oranları belirlenen limitleri aşıyor, ancak tez/dönem projesi danışmanı intihal yapılmadığı kanısında ise: Yukarıda yer alan beyanın ve ekte sunulan Tez Çalışması Orijinallik Raporunun doğruluğunu onaylar ve Uygulama Esaslarında öngörülen yüzdeleri aşılmasına karşın, aşağıda belirtilen gerekçe ile intihal yapılmadığı kanısında olduğumu beyan ederim.</p>	
Gerekçe:	
<p>Benzerlik taraması yukarıda verilen ölçütlere uygun olarak tarafımda yapılmıştır. İlgili tezin orijinallik raporunun uygun olduğunu beyan ederim.</p> <p style="text-align: right;">Danışman Öğretim Üyesi Unvanı, Adı-Soyadı Doç. Dr. Mehmet Fatih DOĞRUCAN</p> <p style="text-align: right;">İmza</p>	

İÇİNDEKİLER

ŞEKİLLER LİSTESİ	iv
TABLolar LİSTESİ	v
RESİM LİSTESİ	vi
KISALTMALAR LİSTESİ	vii
ÖZET	viii
SUMMARY	x
ÖNSÖZ	xii
GİRİŞ.....	1

BİRİNCİ BÖLÜM

CHARACTERISTICA UNIVERSALIS VE CALCULUS RATIOCINATOR'A GİDEN YOLDA LEIBNİZ VE DÖNEMİ

1.1. Leibniz'i Yorumlamak.....	15
1.2. Leibniz'in İçine Doğduğu Yüzyıl ve Avrupa Coğrafyası.....	18
1.3. Leibniz'i Etkileyen On Yedinci Yüzyıl Düşünürleri ve Dilbilimcileri	22
1.3.1. Francis Bacon (1561-1626)	23
1.3.2. Thomas Hobbes (1588-1679)	24
1.3.3. René Descartes (1596-1650)	26
1.3.4. Port-Royal Mantığı (1660)	27
1.3.5. John Wilkins (1614 – 1672) ve George Dalgarno (1616 – 1687)	29
1.3.6. John Locke (1632-1704).....	31
1.4. Gottfried Wilhelm Leibniz (1646-1716).....	35
1.4.1. İlk Gençlik Yılları.....	37
1.4.2. Mainz Yılları	41
1.4.3. Paris Yılları.....	42
1.4.4. Hannover Yılları	45
1.4.5. Leibniz Felsefesinin Belirginlik Kazandığı Dönem	50
1.5. <i>Characteristica Universalis</i> ve <i>Calculus Ratiocinator</i>	65
1.6. Leibniz'in Evrensel Dil Hayali Üzerine Kısa Bir Değerlendirme	72

İKİNCİ BÖLÜM

LEIBNİZ'DEN TURING'E BİÇİMSEL DİLLER

2.1.	Leibniz ve İkili Sayı Sistemi.....	74
2.1.1.	Leibniz Sonrası On Sekizinci Yüzyıl	78
2.2.	On Dokuzuncu Yüzyılda Mantık.....	79
2.2.1.	August De Morgan (1806-1871)	80
2.2.2.	George Boole (1815-1864).....	83
2.2.2.1.	Boole Cebiri	88
2.2.2.2.	Boole Cebiri Bir <i>Calculus Ratiocinator</i> mıdır?	95
2.2.3.	Friedrich Ludwig Gottlob Frege (1848-1925).....	96
2.2.3.1.	Gottlob Frege'nin <i>Kavram-Yazısı Bir Characteristica Universalis</i> midir?	107
2.3.	Yirminci Yüzyılda Matematiğin Temelleri	111
2.3.1.	Alfred North Whitehead (1861-1947) ve Bertrant Russell (1872-1970)	111
2.3.2.	David Hilbert (1862-1943) ve Programı	116
2.3.2.1.	Hilbert Programının Çöküşü	124
2.4.	Leibniz'in Evrensel Dil Hayalinin İmkânı.....	135

ÜÇÜNCÜ BÖLÜM

ALGORİTMALAR, PROGRAMLAMA DİLLERİ, YAPAY ZEKÂ VE EVRENSELLİK

3.1.	Turing Makinesi ve Algoritma.....	138
3.2.	Bilgisayarların, Yapay Zekânın, Biçimsel Dillerin Doğuşu ve Gelişim Süreçleri ..	143
3.2.1.	Birinci Nesil Bilgisayarlar ve Birinci Nesil Programlama Dilleri.....	146
3.2.2.	İkinci Nesil Programlama Dilleri, Bilgisayarlar ve Yapay Zekâ.....	149
3.2.3.	Üçüncü Nesil Programlama Dilleri, İkinci ve Üçüncü Nesil Bilgisayarlar.....	155
3.2.3.1.	FORTTRAN (FORMula TRANslator), ALGOL (ALGORithmic Language) ve COBOL (COMmon Business Oriented Language)	156
3.2.3.2.	Simula, BASIC (Beginners All Purpose Symbolic Instruction Code), Pascal, C ve Java	161
3.2.4.	Dördüncü Nesil Programlama Dilleri, Dördüncü Nesil Bilgisayarlar ve İnternet .	167
3.2.5.	Beşinci Nesil Programlama Dilleri.....	170
3.3.	Programlama Dillerinde Evrensellik Sorgulaması.....	174
3.3.1.	Programlama Dilleri <i>A Priori Felsefi</i> Diller midir?	177

3.3.2. Programlama Dilleri Evrensel Diller midir?	179
3.3.2.1. Linguistik Bağlamda Evrensellik Ölçütü	179
3.3.2.2. Turing- <i>tamlık</i> ve Hesaplama Kuramı Bağlamında Evrensellik Ölçütü	187
3.4. Yapay Zekâ Bağlamında Evrensellik Sorgulaması.....	197
SONUÇ	209
KAYNAKÇA.....	222
ÖZGEÇMİŞ	238

ŞEKİLLER LİSTESİ

Şekil 1.1 Dalgarno'nun varlık düzleminin kısıtlı bir örneği.....	30
Şekil 2.1 <i>I-Ching</i> Heksagram modelinden bir kesit	75
Şekil 2.2 Leibniz eşleştirmesinden bir örnek.....	75
Şekil 2.3 Joachim Bouvet tarafından Leibniz'e gönderilen <i>I-Ching</i> ve Leibniz'in yaptığı eşleştirme	76
Şekil 2.4 Venn diyagramlarından bir örnek	89
Şekil 3.1 <i>Assembler</i> arabirimi ve çalıştırılma süreci	150
Şekil 3.2 <i>Compiler</i> (derleyici) arabirimi ve çalıştırılma süreci	151
Şekil 3.3 Bir bilgisayar sisteminin katmanları.....	152
Şekil 3.4 C programlama diliyle yazılmış bir programın Windows işletim sisteminde çalıştırılma süreci.....	166
Şekil 3.5 <i>Interpreter</i> (yorumlayıcı) arabirimi ve çalıştırılma süreci.....	170
Şekil 3.6 Programlama dillerinin birbirleriyle olan ilişkisi	175
Şekil 3.7 Sözdizimi, türetme ve ayrıştırma ağacı arasındaki ilişkisi.....	184
Şekil 3.8 Beş gizli nöron kullanılmış bir sinir ağı gösterimi	204
Şekil 3.9 Yapay bir sinir hücresinin yapısı	205

TABLULAR LİSTESİ

Tablo 2.1 Toplama için bir TM (program) örneği.	133
Tablo 3.1 Mantık kapıları örneği.	143

RESİM LİSTESİ

Resim 3.1 Cambridge Üniversitesinde bulunan EDSAC adlı bilgisayar	148
---	-----

KISALTMALAR LİSTESİ

Alm.	: Almanca
İng.	: İngilizce
Lat.	: Latince
Bkz.	: bakınız
vb.	: ve bunun gibi
vd.	: ve diğerleri

ÖZET

Kusursuz dil, ideal dil ya da evrensel dil arayışı yüzyıllar boyunca çeşitli şekillerde devam etmiştir. Bu arayış sonucunda, doğal dillerin bir sentezini içeren *a posteriori* diller ortaya çıktığı gibi tamamen keyfi seçimler sonucu *a priori* diller de ortaya çıkmıştır. *A posteriori* diller, konuşulabilir diller iken *a priori* diller genellikle biçimsel dillerdir. Tezimizin amacı, kavram sınıflandırmasını mantıksal temellere dayandıran *a priori felsefi* dillerin birer yansıması olan programlama dillerinin evrenselliğini tartışmaktır.

Tartışmamızın temel dayanağı *a priori felsefi* bir dil olan Leibniz'in kavram göstergeleri dili yani *characteristica universalis* ve bu dilde yargıların hesaplanabilmesi için önerdiği *calculus ratiocinator*dur. Leibniz'in evrensel dilinin, diğer tüm evrensel dil çalışmalarından ayrıldığı özellikleri, tezimizin doğru bir yolda ilerlemesi için belirlenmiş hareket noktalarıdır.

Leibniz, yalnızca bir evrensel dil hayal etmekle veya önermekle kalmaz, bunun için her kavramın ilksellerine birer karakteristik sayı vermeyi önerir. Önerdiği bu karakteristik sayılarla yapılan basit dört işlem sonucunda ise argümanlarımızın doğruluğunun veya yanlışlığının yargılanabileceğini de vurgular.

Leibniz'in evrensel dili için yaptığı çalışmalar ortaya bir evrensel dil çıkarmamıştır ancak ortaya oldukça zengin yan ürünler çıkarmıştır: “Dil olarak mantık” ve “hesap olarak mantık”. Mantığın ayrımlanmış bu iki alanı, bizi programlama dillerinin evrenselliği sorgulamasına götürecek nüveler barındırmaktadır.

Bu çalışmada, Leibniz'in evrensel dil hayalinden hareketle, günümüz bilişim teknolojilerinde önemli bir yere sahip programlama dillerinin evrenselliğine uzanacağız. Bunun için öncelikle, Leibniz'in evrensel dilinin mantıksal ve matematiksel yapısını serimlemeye çalışacağız. On dokuzuncu yüzyıl sonlarından itibaren Leibniz bağlamında tasarlanmış başka bir *a priori felsefi* dil örneği olan Frege'nin *Begriffsschrift* (Kavram-Yazısı) adlı çalışmasının bir sonucu olarak, matematiğin günümüz notasyonlarının ve biçimsel dilinin kesin sözdiziminin çeşitli mantıklar üzerinden gelişimini takip edeceğiz. Leibniz'in evrensel dilinin izini kaybetmemek adına, matematik ve mantık alanındaki bu gelişmelerin aktarımı olmadan açıktır ki programlama dillerinde bir evrensellik sorgulamasını gerçekleştiremiyoruz. Çünkü Leibniz'den programlama dillerine uzanabilmek için açıkça on dokuzuncu yüzyılda matematik ve mantık köprüsünü kullanmak gerekiyor.

Programlama dilleri, komut ve döngülerini İngilizce benzeri emir kiplerine sahip ifadelerle gerçekleştirirken, sözdizimleri tamamen mantıksal ve matematiksel notasyonlardan

oluşur. Bu halleriyle, gerçek birer *a priori felsefi* dil oldukları açıktır. Bu nedenle bu dil yapılarının evrensel olma özelliği taşıdıklarına dair iddialar ortaya çıkmıştır.

Ayrıca değinilmesi gereken bir diğer husus da *a priori felsefi* dil özelliğindeki programlama dillerinin, tamamen matematiksel işlemler ve hesaplamalar üzerinden geliştirilen programların yazılması için kullanıldığıdır. Bu diller sayesinde bir bilgisayara istediğimiz işleri yaptırabilmekteyiz. Bunun olması için gerçekleşen süreç hesaplama üzerinedir. Böylece karşımıza bir de hesaplama kuramı bağlamında evrensellik ölçütü çıkıyor.

Tezimiz bu anlamda, bir yandan programlama dilleri üzerinden diğer yandan hesaplama kuramı ve yapay zekâ bağlamında Leibniz'in evrensel dil hayalinin izini sürecektir. Sonuç olarak matematiksel mantığın temel kuramları aracılığıyla böyle bir evrensel dilin mümkün olmadığını ifade edebiliriz ancak bir evrensel dil gerekliliğini de canlı tutmaya devam edeceğiz.

Anahtar Kelimeler: Leibniz, *Characteristica Universalis*, *Calculus Ratiocinator*, Evrensel Dil, *A Priori Felsefi* Diller, Mantık, Matematik, Matematiksel Mantık, Episteme, Linguistik, Dil Felsefesi, Programlama Dilleri, Yapay Zekâ, Yapay Öğrenme, Derin Öğrenme, Hesaplama Kuramı.

SUMMARY

**FROM LEIBNIZ TO ARTIFICIAL INTELLIGENCE: A LOGICAL AND
EPISTEMOLOGICAL STUDY ON FORMAL LANGUAGES**

The search for the perfect language, ideal language or universal language has continued in various forms over the centuries. As a result of this search, *a posteriori* languages, which include a synthesis of natural languages have emerged, as well as *a priori* languages as a result of completely arbitrary choices. While *a posteriori* languages are talkable languages, *a priori* languages are usually formal languages. The aim of our thesis is to discuss the universality of programming languages, which are a reflection of *a priori philosophical* languages that base concept classification on logical foundations.

The basis of our discussion is Leibniz's characteristic language, *characteristica universalis*, which is an *a priori philosophical* language, and *calculus ratiocinator*, which he proposed for calculating judgments in this language. The features of Leibniz's universal language that distinguish it from all other universal language studies are the starting points set for our thesis to move on the right path.

Leibniz not only imagines or proposes a universal language, but proposes to give a characteristic number to the primary form of each concept. He also emphasizes that as a result of four simple operations with these characteristic numbers, the truth or falsity of our arguments can be judged.

Leibniz's work for the universal language did not end up with a universal language, but it did yield quite rich by-products: "Logic as language" and "logic as calculus". These two separate areas of logic contain very important points that will lead us to question the universality of programming languages.

In this study, we will handle the universality of programming languages, which have an important place in today's information technologies, starting from Leibniz's dream of a universal language. For this, first of all, we will try to reveal the logical and mathematical structure of Leibniz's universal language. As a result of Frege's *Begriffsschrift* (Concept-Script), another example of *a priori philosophical* language designed in the context of Leibniz since the end of the nineteenth century, we will follow the development of the precise syntax of today's notation and formal language of mathematics through various logics. In order not to lose the trace of Leibniz's universal language, it is clear that we can not carry out a question of universality in programming languages without the transfer of these developments in the fields of mathematics and logic. Because in order to reach programming languages from

Leibniz, it is clearly necessary to use the bridge of mathematics and logic in the nineteenth century.

While programming languages perform their commands and loops in expressions with English-like imperative moods, their syntax consists entirely of logical and mathematical notations. As such, it is clear that they are genuine *a priori philosophical* languages. For this reason, claims have emerged that these language structures have the feature of being universal.

In addition, another point that should be mentioned is that programming languages, which are *a priori philosophical* language, are used to write programs that are developed entirely on mathematical operations and calculations. Thanks to these languages, we can make a computer do the work what we want. The process that takes place for this to happen is computational. Thus, we face with the criterion of universality in the context of computational theory.

In this sense, our thesis will follow Leibniz's dream of a universal language through programming languages on the one hand, on the other hand through context of computational theory and artificial intelligence. As a result, we can state that such a universal language is not possible through the basic theories of mathematical logic, but we will continue to keep the necessity of a universal language alive.

Keywords: Leibniz, *Characteristica Universalis*, *Calculus Ratiocinator*, Universal Language, *A Priori Philosophical* Languages, Logic, Mathematics, Mathematical Logic, Episteme, Philosophy of Language, Programming Languages, Artificial Intelligence, Machine Learning, Deep Learning, Computation Theory.

ÖNSÖZ

Bu çalışmanın yürütüldüğü sıralarda, atlatmak için büyük mücadele içinde olduğu kanser hastalığına rağmen, hastane odalarında dahi olsa desteğini hiçbir zaman esirgemeyen, sesini kaybetmiş olmasına karşın yine de iletişimin yazılı yollarıyla çok değerli yorum ve eleştirileriyle, çalışmanın başından sonuna kadar biçimlenmesine ve benim vazgeçtiğim yerde elimden tutarak tamamlanmasına katkıda bulunan çok değerli danışmanım Doç. Dr. Mehmet Fatih DOĞRUCAN'a teşekkürlerimi sunarım. Eğer bu çalışmanın konuyla bağlantılı literatüre bir katkısı olacaksa bu katkı, bana gücü nispetinde cesaret ve çalışma azmi veren, kanser tedavisini gördüğü bu zor koşullarda desteğini sürdürmeye devam eden danışman hocamın sayesinde.

Çalışmanın hazırlanma sürecinde Leibniz üzerine literatürünü ve kıymetli bilgilerini paylaşarak, ayrıca şahsıma verdiği derslerin yanı sıra çalışmanın özellikle birinci bölümünün gelişmesi yönünde eleştirilerini sunarak katkıda bulunan hocam Doç. Dr. Mustafa KAYA'ya teşekkürü borç bilirim. Aynı şekilde, çalışmanın hazırlanma sürecinde matematiksel mantık alanında şahsıma verdiği derslerle, çalışmanın Leibniz'den yapay zekâyâ bağlanmasında köprü görevi gören ikinci bölümünün güçlendirilmesine katkı sunan ve bu bölümün gelişmesinde büyük rolü olan hocam Dr. Öğretim Üyesi Ali Bilge ÖZTÜRK'e teşekkür etmeliyim. Onların desteği olmasaydı bu çalışma eksik kalırdı.

Teorik bilgisayar bilimlerinin bazı kuramlarını içeriyor olmasından ötürü, çalışmanın üçüncü bölümü, Tez İzleme Komitesinin ortak kararı ile bu alanda çalışmaları olan Dr. Öğretim Üyesi Müge Erel ÖZÇEVİK hocamın kontrolünden geçmiştir. Kendisinin çalışmanın üçüncü bölümü ve tezin geneli için yaptığı yorum, doğrudan alıntı olarak aşağıda sunulmuştur.

Tezin üçüncü bölümünde, evrensel dil arayışının yeni nesil programlama dilleri ile özdeşleştirilmesi ve detaylı yorumu net bir şekilde ele alınmaktadır. Özellikle yapay zekâ içeren algoritmaların yürütüldüğü programlama dilleri evrenselleştirilebilir mi sorusunu cevaplayan bu bölüm, yapay zekâyı matematiksel olarak biçimlendirmeye çalışmaktadır. Bu yönüyle, bu tezin disiplinlerarası bir çalışma olarak literatüre önemli katkı sağlayacağı öngörülmektedir (Erel-Özçevik, 09 Mayıs 2022 tarihli görüntülü görüşme).

Yukarıda yorumunu paylaştığım değerli hocama teşekkürlerimi sunarım. Savunma jürisindeki kıymetli eleştirileri ve yorumlarının yanı sıra övgülerine mazhar olduğum Prof. Dr. Süleyman DÖNMEZ ve Doç. Dr. Şevki IŞIKLI hocalarıma teşekkürü borç bilirim.

Çalışmanın bir de görünmeyen yüzünde bulunan emektarlar var: Tezin ikinci bölümünü matematiksel yaklaşımla değerlendiren Dr. Öğretim Üyesi Abdulhamit KÜÇÜKASLAN ve tezin üçüncü bölümünde söz edilen programlama dillerini nesillendirme konusunda yol gösteren bilgisayar mühendisi Tufan BAĞDU ile sunduğu çok değerli bakış açısı sonucunda, yapay zekâ araştırmalarına eğilmemi salık veren eski Arş. Gör. Deniz AYDIN'a değerli katkılarından dolayı teşekkür ederim.

En az danışman hocam Doç. Dr. Mehmet Fatih DOĞRUCAN kadar emeği geçtiğini düşündüğüm biricik evladıma ve ayrıca hayat arkadaşıma, bu süreçte bana gösterdikleri anlayıştan ve sabırdan ötürü ne kadar teşekkür etsem az kalır. Onlara minnettarım.

Bu çalışmanın elde edeceği değer, yukarıda adı geçen isimlerin katkıları ve özverileri sayesinde. Tüm kusurlar ve hatalar, şahsıma aittir.

Zuhal HAZAR

Antalya, 2022

GİRİŞ

Gottfried Wilhelm Leibniz (1646-1716) bir hayal kurdu ve bu hayalini gerçekleştirmek üzere harekete geçti. Leibniz'in hayali, bir "evrensel dil" yaratma üzerineydi. Leibniz'in evrensel dilini bu şekilde yorumlamak kulağa ilginç gelse de inancı doğrultusunda, önceden belirlenmiş harmoniye inanan Leibniz'in ekümenist, monist, holistik ve bireyci felsefesinin gereği, gerçek bilgiye ve gerçek dine her bireyin ulaşabilmesi için doğal dillerin ötesinde, anlam kargaşasına izin vermeyecek rasyonel biçimsel bir dil gerekiyordu. Bu dil, gittiği her yere pratik ve teorik bilginin yanı sıra gerçek dini de götürecekti.

Leibniz'in evrensel dil hayali, on yedinci yüzyılda dilbilimciler, düşünürler ve hatta ticaretle uğraşan başkalarının da hayaliydi. On yedinci yüzyıl bu anlamda oldukça zengin yapay ve özellikle Leibniz'in *characteristica universalis*inin de aralarında bulunduğu biçimsel dil ürünlerinin yüzyılıdır.

On yedinci yüzyılda, örneğin mesajları şifrelemeye yarayan ve bilinmeyen bir dildeki mesajları çözmek için de kullanılabilen *steganografiler* üzerine olduğu kadar, herhangi bir ulusa mensup her bireyin anlayıp iletişim için kullanabileceği karakteristikleri içeren *poligrafiler* üzerine de ciddi çalışmalar vardı. Bu tür biçimsel dil yapılarından *steganografiler*, mesajlaşmanın veya şifrelemenin yapılacağı doğal dilde kullanılan harflerin olasılığına dayanırken, evrensel bir iletişim aracı olması hedeflenen (Strasser, 2007: 23) *poligrafiler*, kaynak dil ve amaç dil arasında mesajlaşmayı sağlayacak bir terimler sözlüğünü gerektiriyordu.

Evrensel olmaları yönünde çabaların sergilendiği yukarıda adı geçen bu tip biçimsel diller, genel olarak *pasigrafi* adı altında toplanır. Sadece yazılması amaçlanan yapay dillere özgü bir tanımlama olan "*pasigrafik* sistemin sembolü, herkes tarafından anlaşılır ancak (okuyucunun diline bağlı olarak) birçok farklı şekilde telaffuz edilir" (Libert, 2018: 7). Birer *pasigrafi* olarak kabul edilen diğer biçimsel dil yapıları ise "imgelerle evrensel bir kod geliştirmeye çalışıldığında" (Eco, 1995a: 173) kullanılan yani kelimenin harfleri yerine fikri temsil eden *ideografiler* ve *piktografilerdir*. Bunun yanı sıra yalnızca rakamlar kullanılarak üretilmiş biçimsel dil yapıları da mevcuttur.

Biçimsel diller, özel olarak tasarlanmış keyfi karakteristiklerle icat edildiğinden dolayı *a priori* diller grubunda değerlendirilir. Bunun dışında, doğal dillerin bir kombinasyonunu içeren ve ayrıca yalınlaştırılmış dilbilgisine dayanan türetilmiş diller ise *a posteriori* diller grubunda yer alır.

Yapay dilleri sınıflandırmanın geleneksel bir yolu, onların doğal dillerden ne ölçüde yararlandıklarını göstermekten geçer. Esas olarak bir veya daha fazla doğal dilden malzeme alarak oluşturulan yapay dillere *a posteriori* diller (Esperanto gibi iyi bilinen dilleri içerir), doğal dilleri kaynak olarak kullanmayanlara ise *a priori* diller denir (ancak birçok *a posteriori* dilde sınırlı miktarda *a priori* malzeme bulunur ve bazı *a priori* dillerde az sayıda *a posteriori* bileşen bulunur) (Libert, 2018: 1 italik vurgu bana ait).

Özellikle on yedinci yüzyılda doğal dillerin ötesinde aranır hale gelmiş ideal diller, doğal dillerin açıkça bir kusuru olarak da görülebilecek anlamda müphemliği ortadan kaldırmak adına tasarlanan biçimsel ve *felsefi* dillerdi ki bunlar birer yardımcı dil olarak da görülmekteydi. Ancak *felsefi* dillerin de dünyayı doğal dillerden daha net ve kesin bir şekilde temsil etme amacı vardı. Bu nedenle *felsefi* dillerin çoğu veya tamamı *a priori* türdendi (Libert, 2018: 4). Bunun tersi her zaman geçerli değildi yani her *a priori* dil, *felsefi* değildi.

On yedinci yüzyılda çeşitliliği göze çarpan rasyonel yapay dillerdeki ilk girişimlerin çoğu, Wilkins'te (1668) sunulan sistemde olduğu gibi *a priori* ve *felsefi* türdendi (Libert, 2018: 7-8) yani keyfi kelimeleriyle *a priori* görünümde iken kavramsal bir sınıflandırma içerdiklerinden dolayı, ayrıca *felsefi* dil görünümündeydi. George Dalgarno (1616-1687) ve John Wilkins'in (1614-1672) üretme çabasında oldukları yapay dilleri, töz ve ilinekler bağlamında mantıksal bir sınıflandırmayı gerektiriyordu. Bu nedenle Wilkins'e göre, eğer bir evrensel dil olacaksa bu dil felsefi ilkelere göre inşa edilmeliydi (Subbiondo, 1990: 357) ve Dalgarno'ya göre bu tür bir yapay dilin kurulumunda, "filozofun çalışması, dilcinin çalışmasını" (Eco, 1995a: 223) öncelikle olmalıydı.

Leibniz'in evrensel dili tıpkı Dalgarno ve Wilkins'te olduğu gibi *a priori felsefi* bir dil olarak *pasigrafik* (yazılabilir) görünümdeydi. Özellikle Dalgarno'nun çalışmalarına odaklanmış Leibniz'e göre, doğru uygulanmış kavram analizi ile elde edilecek bir kavramlar sözlüğüne ek olarak, *pasigrafik* dil yapılarında olduğu gibi her kavramın yerini alacak karakteristikler aracılığıyla evrensel bir dil yaratılabilirdi.

Yerel diller üzerine geniş bir literatüre sahip olan Leibniz, "dilleri insanlık durumunun incelenmesinde zengin bir referans olarak görmekteydi" (Remnant ve Bennett, 1996: xxiv). Bu nedenle evrensel diline uygun *felsefi* bir dilbilgisinin, anlayışın çeşitli biçimlerini açığa çıkarmak için gerekli olduğunu (Leibniz, 1996: III. VI. §3. 280) düşünmekteydi. Bu haliyle, Leibniz'in evrensel dilinin diğerlerinden farkı anlaşılabilir olsa da Leibniz'in, ideal veya evrensel dil arayışında olanlardan belirgin olarak ayrıldığı bir nokta vardı: Leibniz'den önce

hiçbir çalışmada, her kavrama kendi karakteristik sayısının verilebileceğine dair bir fikir (Leibniz, 1969ç: 222) ortaya atılmamıştı.

Leibniz'in, "karakteristik" olarak adlandırdığı düşünce alfabesi sayesinde, bilimsel gerçeklerin mantıksal bağlantısı ve hatta bilimlerin hiyerarşik düzeni belirlenebilecekti (Couturat, 2004: 18). Genel bilimler ansiklopedisi olarak da bilinen, bilimlerin hiyerarşik düzeninin belirleniminin temelinde ise Leibniz'in esinlendiği ünlü Katalan misyoner Ramon Llull'un (1232-1316) *Ars Magna* adlı çalışmasında görülen kombinasyon sanatının yargı ve keşif gücü vardı. "Sanat, yanlış kombinasyonları doğrudan (analitik kısımdan) ayırmak için kullanılacaktı" (Welch, 1990: 76). Böylece elde edilen ürün "dünya hakkındaki en genel gerçeklerin toplamı yani kesin felsefe" (Welch, 1990: 76) olacaktı.

Leibniz'in evrensel dili, yalnızca bir biçimsel iletişim aracı olarak tasvir edilemeyecek kadar geniş bağlama sahipti çünkü alanına ait her tür bilginin hatasız bir şekilde hazır bulunacağı, böylece insanlığın doğru epistemeye ulaşabileceği içeriğe sahip bir genel bilimler ansiklopedisinin de dili olacaktı.

Nispeten basit olan bu fikir, ideal bir dil inşa etmeye yönelik tüm rasyonel yaklaşımın özünü tanımlar. Bununla birlikte, bu fikri uygulamaya koymak için gereken, daha sonra sıralanabilecek sonlu bir dizi sabit ve farklı fikir üreten bir felsefenin geliştirilmesidir. Bu basit fikirler, sonrasında daha karmaşık fikirler üretmek için birleştirilebilir ta ki insan aklının alabileceği her düşüncüyü içerene kadar. Böyle bir girişimin doğal sonucu, bir tür düşünce matematiği veya a priori bir felsefi dil üretmek olacaktır (Laporte, 2018: 593).

Leibniz'in epistemolojik kaygıları sonucunda oldukça geniş bir bağlama oturttuğu görülen evrensel dili gerçekten de bir hayaldi. Öyle ki Leibniz'in bu bağlamda yürüttüğü hiçbir çalışması yayımlanmamıştı çünkü bu dili anlatmaya çalıştığı hiç kimsede kendi coşkusunun yansımalarını bulamıyordu. Bu nedenle Leibniz'in evrensel dili, ölümünden yüz elli sonrasına kadar eskizler ve mektuplar arasında uykuda kaldı ta ki Leibniz'in ikincil literatürü derleninceye değin. Bundan sonra ise evrensel dil üzerine yaptığı çalışmalarla birlikte, Leibniz'in hem ekollerden gelen hem de modern felsefeye dayanan çok bağlamli çalışmaları, parçalara ayrıldı ve her disiplinin kendi alanlarına hizmet etmek üzere bölünerek serpildi, genişledi.

Evrensel dili de Leibniz'in diğer çalışmalarıyla aynı kaderi paylaşarak, sembolik mantığın iki farklı türüne ayrıldı. Öncelikle Leibniz'in çalışmalarından haberdar olmayan George Boole'un (1815-1864) çabalarıyla yaratılan cebirsel mantık, Leibniz'in evrensel dilinden haberdar olan Ernst Schröder'in (1841-1902) ellerinde genişletildi ve Boole cebirine dönüştürüldü. Boole cebiri, cebirsel mantık olarak kazandığı görünümle, Leibniz'in

karakteristik sayılar arasında bir hesaplama prosedürü olarak önerdiği *calculus ratiocinatorunun* ancak kısıtlı bir evrenini temsil edebiliyordu ve “hesap olarak mantık” alanına giriyordu. Sonra da Leibniz’in evrensel dil çalışmalarından haberdar olan Friedrich Ludwig Gottlob Frege’nin (1848-1925) ellerinde yaratılan birinci seviye yüklem mantığı, sözdizimsel yapısıyla bir *lingua characteristica (characteristica universalis)* görünümündeydi ve “dil olarak mantık” alanını temsil ediyordu.

Frege ve Schröder arasında yaşanan “dil olarak mantık” ve “hesap olarak mantık” ayrımlamalarına kadar, hiç kimse Leibniz’in evrensel dil hayalini ciddiye almamıştı. Bundan sonra ise bir polimat olan Leibniz’in adı, onca başarılı çalışmalarıyla birlikte sembolik mantık alanında mantıksal uslamlamaya odaklanan ilk kişilerden biri (Nilsson, 2019: 28) olarak anılmaya başlandı. Leibniz’in evrensel dili açıkça bir kavram göstergeleri diliydi ve

simgesel mantığın çeşitli akımlarının kaynağını ondan aldığı kesindir. Kavram göstergeleri yazısının temelinde yatan asal sezgi şudur: Karakterler nedensiz olarak seçilse ve uslamlama adına ilksel oldukları varsayılan kavramların gerçekten de ilkseller olduğundan emin olunmasa bile şu olguyla hakikat güvence altına alınır: *Önermenin biçimi nesnel bir hakikati yansıtır* (Eco, 1995a: 281).

Leibniz’in hayali, kendini matematik ve mantık ile ortaya koymaktaydı çünkü Leibniz, matematiğin evrenselliğine inanıyordu ve onun dili, mantığa dayandığı ölçüde matematiksel bir dildi. Gerçekten de Leibniz’in evrensel dili, ilk değerini *lingua characteristicası* ile matematiğin mantıksal temeller üzerinde yükselmesi gerektiğine inanan Frege’nin ve *calculus ratiocinatoru* ile mantığın doğal dile değil, matematiğe yakın olması gerektiğine inanan Boole’ün çalışmalarında bulmuştu. “Hesap olarak mantık” alanında yapılan çalışmalarda, Boole’e göre “uzlaşım ve yorum yapmak için gerekli olan mantığın kesin matematiksel gelişimi, simgesel olmayan Aristotelesçi mantığın birikmiş geleneklerini bir kenara atmadan yapılamazdı” (Lewis, 1918: 51).

Bundan sonra ise Leibniz’in evrensel diline ait bu iki ayrı mantık alanının bir birleşimini içinde barındıran matematiğin biçimsel dilinin, en olgun seviyesini görmesini sağlayan Alfred North Whitehead (1861-1947) ve Bertrant Russell’in (1872-1970) ellerinde *Principia Mathematica*’da bir yansımasını görüyoruz.

Matematikte *biçimselci* yaklaşımın öncülerinden David Hilbert (1862-1943), matematiğin dilinin sarsılmaz olduğuna dair inancın, yine bu dil içinde verilecek ispatlarla pekinleştirilmesi gerektiğini düşünenlerdendi. Bu nedenle Hilbert, matematiğin, dolayısıyla matematiğin dilinin tam, tutarlı ve bir argümanın doğruluk değerine yine bu dilin içinden bir algoritmayla karar verilebiliyor olmasına dair kanıtlar istemişti. Matematiğin tutarlılığı, yeteri

kadar güçlü bir aksiyomatik sistemin aksiyomlarından çelişki çıkarılamamasına dayanıyordu ve tamlığı ise yeteri kadar güçlü bir sistemin biçimsel dilinde verilen her önerme veya deęilinin yine bu sistemin aksiyomlarından çıkarılabilmesi gereklilięine dayanıyordu. Bunun yanı sıra yeteri kadar güçlü bir sistem içindeki her önermenin doęruluk deęerine, bu sistem içerisinde algoritmik olarak karar verilebilmeliydi.

Hilbert'e göre matematik tamdı, tutarlıydı ve bu dilin içinde her argüman için bir karar algoritması vardı. Oysa öyle deęildi. Hilbert'in aradıęı kanıtlar bulunabilmişti ancak bu kanıtlar matematięin içindeki temel aritmetięi kapsayacak kadar güçlü aksiyomatik sistemlerin içinde karar verilemez önermelerin bulunduęunu, dolayısıyla matematięin biçimsel dizgesinin eksik olduęunu gösteriyordu.

Hilbert projesinin çökmesi, matematięin üzerinde yükseldięi aksiyomatik temellerin zayıf olduęunu gösterdięi gibi dolaylı bile deęil, doğrudan Leibniz'in evrensel dil hayalinin de gerçekteşemeyeceęini ortaya koymuştu fakat yine de bu hayal, yaşatılabileceęi ve hatta çok daha fazla tartıřılabılır durumda olacaęı bir noktaya da taşınmıştı.

Özellikle Hilbert'in "karar problemi" için Turing'in "Hesaplanabilir Sayılar" (1936) adlı makalesinde verdięi sonlu adımlı ispatta uyguladıęı algoritma o kadar belirli ve net adımlardan oluşuyordu ki bir insan matematikçi yerine bir makine de aynı işi yapacak olsa sonuç deęişmeyecekti. Bu makale, açıkça yeni bir çağı başlatmıştı: Makineler (bilgisayarlar) çağı.

Bundan sonra kavramsal ve kuramsal alt yapının, mühendislikten gelen donanım bilgisiyle harmanlanıp şekillendięi bilgisayarlara (makinelere) ne şekilde görev tanımlanabileceęine dair çalışmalar başlamıştı. Böylece, bir bilgisayara işleri yaptırabilmemizi sağlayacak komutlar listesi ile insan ve bilgisayar arasında bir iletişim görevi görmek üzere, *derleyiciler*, *yorumlayıcılar* ve sonra da programlama dilleri üzerine ciddi tasarım süreçlerine girişilmişti.

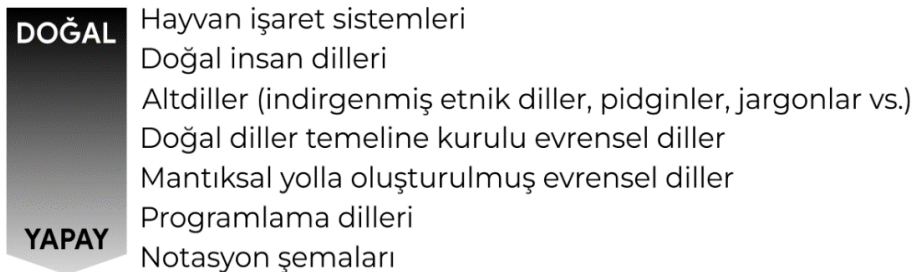
Bu makinelerin (bilgisayarların) anladığı dil, kodlanabilecek en kolay dil olarak 0 ve 1'lerden oluşuyordu. Her ne kadar Leibniz de "İkili Aritmetik Sistemin Açıklaması" (Explanation of Binary Arithmetic) adlı kısa risalesinde, ikili sistemin yaratabileceęi kolaylıkları konu edinip, her şeyin bu sayılara indirgenebileceęini ([1703] 2007b: 2) bulmuşsa da Leibniz'in buluşu metafizik kökenlere dayanıyordu yani Tanrı ve monadlar arasındaki ilişkide 1 ve 0, varlık ve yokluęun temsilcileriydiler.

Boole cebirinin önermelere doęruluk deęeri veren kısıtlı bir evreni vardı ve bu evren öncelikle Charles Sanders Peirce (1839-1914) tarafından (çalışmasını yayımlamadığı için), daha sonra ise yeniden keşfedilmek üzere Claude Elwood Shannon (1916-2001) tarafından

Röle ve Anahtarlama Devrelerinin Sembolik Analizleri (A Symbolic Analysis of Relay and Switching Circuits [1936]) adlı tezinde, bu mantıkta bulunan “ve”, “veya” ve “değil” gibi ilişkiler, anahtar devreleri üzerinde somuta indirgenebilmişti. Bu sistem kısaca “mantık kapıları” olarak bilinmektedir ve iletim varsa 1, yoksa 0 anlamına gelir.

Günümüze değin ulaşmış tüm bilgisayarların merkezi işlem birimleri (MİB [İng. CPU]) bilgisayara ne yaptırmak istediğimizi bu yolla anlamaktadır. Boole cebiri, Leibniz’in *calculus ratiocinatorunun* sınırlı bir yansımasıydı ve bilgisayarın beyni olarak bilinen en önemli parçasında, merkezi işlem biriminde yaşam bulmuştu. Elbette daha fazlası da gelecekti.

Programlama dillerinin İngilizce benzeri komut dizileri, koşullu ifadeleriyle (*if* statement), döngü ifadeleriyle (*while* statement) Frege’nin birinci seviye yüklem mantığının üzerinde kurulmuştu. Bu dillerin tüm notasyonları, matematiğin ve mantığın notasyonlarından doğrudan veya türetilme yoluyla oluşturulmuştu. Her programlama dili, çeşitli sözdizimlerine ve altlarında yatan mantıksal farklılıklara rağmen gerçek birer *lingua characteristic* idi. Böylece, Leibniz’in *characteristica universalisi* de programlama dillerinin özüne yerleşmişti. Aşağıdaki şekilde Leibniz’in evrensel dil çalışmasının programlama dilleri ile bağlantısı görülebilir.



Şekil 1. Sözlü olmayan hayvan iletişiminden bilgisayar dillerine ve notasyon şemalarına dair spektrum.

Kaynak: Laporte, 2018: 588

Leibniz’in rasyonel kavram göstergeleri dili, olgulara dayanmadığından dolayı (Eco, 1995a: 284) evrensel bir dil olma şansını yitirmişse de bir fikir ve bu fikir etrafında gerçekleşen çabanın sonuçları, günümüzde ayrı bir ontolojik zeminde birçok evrensel dil adayının ortaya çıkmasını sağladı. Artık bu hayal uykudan uyanmış ve bir daha hiç uyumamak üzere güçlenerek günümüze kadar gelmişti. Hangi nesil olursa olsun her programlama dili ve hangi mimariyle çalışıyor olursa olsun her işlemci birimi, kendi görevleri doğrultusunda birer *lingua characteristic* ve birer *calculus ratiocinator*ıdır.

Programlama dilleriyle yazılan programlar üzerinden yürütülen tüm işlemler birer hesaplama prosedürü içeriyordu. Bundan sonra ise bu programlama dillerinin, bu diller

üzerinden yürütülen programlarla yapılan hesaplamaların ve elbette yine programlama dillerinin biçimsel, mantıksal ve felsefi oluşlarının temele konduğu nedenlerden dolayı çeşitli evrensellik iddiaları ortaya atılmaya başlandı.

Bunlardan ilki, hangi programlama dilinin daha evrensel olduğuna dair bir soruyu içermektedir: “Leibniz’in zamanındaki sorun evrensel bir dilin olmamasıyken, bugün sorun belki de çok fazla evrensel dilin (ve hatta daha az evrensel olanın) olmasıdır” (Paleo, 2016: 4). Bir diğeri ise mantık temelli bir evrensellik sorgulamasıdır: Bu dillerin altlarında yatan mantık çeşitleri arasında hangi mantık, evrensel olarak kabul edilmelidir? Bunun dışında, farklı dillerin konuşucularınca anlaşılabilir olma potansiyeli taşıyan bu diller, sırf bundan ötürü evrensel birer sistem olarak tanımlanabilmektedirler. Öyle ki hantal ilerleyen felsefe ve diğer disiplinlere kıyasla, kısa zamanda nesillendirme evrimleri geçirmiş onca teknolojik ve dilsel boyutlarıyla teorik bilgisayar bilimleri, evrensellik iddialarına, “derin öğrenme” gibi tekniklerin kullanıldığı günümüzde “hesaplama kuramı” üzerinden yenilerini de ekleyerek devam ediyor.

Peki gerçekten öyle mi? Gerçekten bu programlama dilleri evrensel diller mi? Bunlar üzerinden yürütülen hesaplamalar evrensel mi yani her tür hesaplama yapılabilir mi ya da her tür hesaplamanın yapılabilir olduğuna karar verilebilir mi? Bu sorular, açıkça görüleceği üzere Leibniz’in evrensel dil hayalinin de sorgulaması için bize yeni bir zemin oluşturmuş oluyor.

İşte, tezimiz tam da bu sorgulamaların peşindedir. Bunu tek bir soru sorarak yapmaktayız: Leibniz’in evrensel dil hayali gerçekleşti mi? Sorduğumuz sorudan da anlaşılacağı üzere, Leibniz ve Leibniz’in evrensel dili üzerine bir tez yazma iddiamız bulunmamaktadır.

Tezimizin genelinde Leibniz’in evrensel dil hayalinin gerçekleşip gerçekleşmediğine dair yürütme çabasında olacağımız sorgulamamıza bir yanıt bulabilmek için öncelikle Leibniz ve felsefesine değinmek gerektiğini düşünüyoruz. Her şeyin her şeyle bağlantılı olduğu felsefesinin olgunlaştığı döneme değin takip edebilmek için Leibniz’i dönemlerine göre serimlemeye çalışacağız. Açıkçası, Leibniz’in tek bir ilkesi veya kavramını ele alıp, başka çözümler yapmadan ilerlemek mümkün değildir. Bu nedenle, Leibniz’i yalnızca evrensel dil bağlamında aktarmaya çalışmak da mümkün değildir çünkü bu dil epistemolojik kaygılardan, mantıksal ilkelere, matematikten, metafizikten ve teolojiden beslenerek ortaya konmuştu. Bu da bizim için ciddi bir yorum zorluğu getiriyor.

Leibniz, bir yandan en temel metafizik figür olarak gördüğü *sayı* ve keşfedilmiş şeylerin gücünü yansıtan aritmetiğin, evrenin bir tür değişmezleri olarak görev yaptığını (1969ç: 221) ifade edip; sayıları ilksel kavramların karakteristik belirleniminde kullanırken,

diğer yandan evrensel diline giden yolda, özne-yüklem ilişkisini “yeter neden ilkesi”ne dayandırdığı bir metafizik mantığını (Kahveci, 2011: 190) yürütmüştür. Aslında, Leibniz’in evrensel dile ulaşmak için sergilediği tüm bu çabaları, olgusal hakikatlere epistemolojik bir yaklaşım olarak da okunabilir.

Gözlemlediğimiz kadarıyla Leibniz, kendisinin ölümüne değin düşünsel canlılığını korumuş evrensel dili, sadece dinlerin veya mezheplerin birleşmesi ya da Tanrı’nın şanının yüceltilmesi için istememiştir, dolayısıyla konuya böyle yaklaşmak Leibniz’i tek bir boyutla ele almak demek olurdu.

Basit bir örneklendirme ile daha ayrıntılı açıklayacak olursak; Leibniz’in *characteristica universalis* bir yüklem mantığı yardımıyla uygulanabilecek analiz ve sentezi öngörmektedir. Burada kullanılan analiz yöntemi, yüklemi (bazen sıfat veya isim) olabilecek en yalın haline getirmeyi amaçlamaktadır. Leibniz, yüklem arasında süreklilik içeren bir analiz öngörerek kendisine ait bir kavram olan “yeter neden ilkesi”ni yürütürken bu ilkenin yürütme yöntemini, yine kendisinin matematik keşfi olan “sonsuz küçükler”den almıştır. Leibniz’in “sonsuz küçükler”i, aslında fizik dünyada şeyler arasında yitip giden farkları konu edinen tamamen metafizik kökene sahip bir keşiftir. Bu keşif, Leibniz’in *Monadoloji* isimli çalışmasında, aralarında içsel düzlemde gerçek bir aynılıktan söz edilemeyecek *monad*ların açıklanmasında da yardımcı bir öge olarak kullanılmıştır. Kendisine ait bir kavram olan “ayırt edilemezlerin özdeşliği ilkesi”ne dayandırarak açıklamasını verdiği *monad*ların birbirlerine göre farklılıklarını Leibniz, eğrisel bir geometri kullanmak suretiyle, süreklilik halinde bir eğride gerçekleşmiş tekilliklere dayandırmıştır. Farklılaşmayla belirginleşen tekillikler, Leibniz’in, bilginin algı ve tamalgı düzeyinde açıklık ve seçikliğine dair görüşlerini psikoloji ve matematik yardımıyla açıklamasını sağlamıştır. Böylece bir töz için algı düzeyinde kalan bulanık bilgi, başka bir töz için tamalgı olarak seçik bilgiyi verebilmektedir. Leibniz’in, bilginin açık seçikliğine dair bu çözümlemesinin temelinde ise yine her bir *monad*ın tek bir nokta-i nazarından gördüğü dünyayı açıklamak üzere devreye soktuğu “konum geometrisi” (analysis situs) göze çarpmaktadır. “Konum geometrisi” ayrıca Leibniz’in fiziğini açıklamak için de kullandığı bir aygıttır. Burada Leibniz, uzay ve zamanı mutlak olmaktan çıkarmakla kalmaz, ayrıca şeylerin birbirine göre durumları üzerinden hareketi açıklamaya çalışır. Özetlemek gerekirse, Leibniz’in herhangi bir görüşü veya keşfi üzerinde çalışılırken çok katmanlı ve geniş ağa sahip felsefi sistemine tek boyutlu bir yaklaşım sergilemek tamamen hatalıdır çünkü Leibniz’de her şey her şeyle bağlantılıdır ve önceden belirlenmiş bir harmoni içermektedir.

Bu nedenle, Leibniz'in evrensel dil hayalinin gerçekleşip gerçekleşmediğine dair sorumuzu sormadan önce tezimizin birinci bölümünde, Leibniz'i bir evrensel dil yaratma çabasına sürükleyen ve ne olursa olsun bu hayalin peşinden gitmesi gerektiğine dair inançlarını ortaya koymaya gayret edeceğiz. Bunun için öncelikle Leibniz'i evrensel dili bağlamında etkilemiş olan düşünürlere, filozoflara, dilbilimcilere ve onların çalışmalarına, Leibniz'in etkilendiği ölçüde yer vermeye çalışacağız.

Leibniz'in yaratma çabasında olduğu evrensel dilinin, bir filozoftan çok bir misyoner olan Llull'un kombinasyon sanatından etkilendiğini daha önce vurgulamıştık. İlk defa yargı sanatını dokuzlu çark sistemiyle tanıtan Llull'un sistemi gerçek bir kombinasyon sanatına dayanıyordu ve Leibniz, bu sistemde Llull'un göremediği başka bir yön keşfetmişti: Kombinasyon sanatına dayanan bu sistemle yalnızca yargıda bulunmak değil ayrıca keşif yapmak da mümkündü. Kaldı ki Llull'un kombinasyon sanatı, Dalgarno ve Wilkins gibi dilbilimcilerin çalışmalarından esinlenilerek elde edilecek rasyonel yapay bir dilin yargı ve keşif gücünü ortaya koyabilirdi. Bu dille argümanlar, Leibniz'in oldukça etkilendiği Thomas Hobbes'un (1588-1679) "düşünmek hesap yapmaktır" (1996: 24) fikrinin altında yatan *kör düşünce* yöntemiyle her kavrama verilmiş karakteristik sayılar arasında basit dört işlemlerle, nihayetinde hesaplanabilir bir duruma da gelebilirdi. Tezimizin birinci bölümünde Leibniz'in tüm bu düşüncelerden ve çalışmalardan hareket ederek ulaştığı dil yapısını, ayrıntılandırarak serimlemeye çalışacağız.

Tezimizin ikinci bölümünde ise sembolik mantığa yaptığı katkılarını serimlemeye çalışacağımız Leibniz'in, bu alanda herhangi bir yayımlanmış risalesi bulunmadığından dolayı nesilsizliğini (Kahveci, 2012: 19) hesaba katarak hareket edeceğiz. Elbette Boole cebiri Leibniz bağlamında bir *calculus ratiocinator* olsa da Boole, sistemini Leibniz bağlamında kurmamıştı çünkü Leibniz'in çalışmalarından haberdar değildi. Açıkçası bir *lingua characteristic* olarak Leibniz'in evrensel dilinin ilk ortaya çıktığı çalışma, Frege'nin *Kavram-Yazısı* adlı eseridir. Frege de çalışmasının asıl niyetinin "Leibniz bağlamında bir *lingua characteristic* yaratmak" (2019a: 144; 2020: 20) olduğunu, çeşitli konferanslarda yaptığı konuşmalarında dile getirmiştir.

Frege'nin kavram göstergeleri dili, yukarıdan aşağı ve soldan sağa doğru bir genişleme içeren *gotik* bir görünüm sergilemektedir. Böyle bir dilin, bilimin her alanında gerekliliğinin dışında en azından başlangıçta matematiksel ifadeler için oldukça gerekli olduğunu, bu dilin ekonomik kullanım ve açıklık avantajlarından dolayı kestirebilmek zor değildir. Oldukça değerli olan bu fikrin sahibi Frege'nin, dili için uygun gördüğü *gotik* gösteriminden daha kolay kullanıma sahip olan Giuseppe Peano'nun (1858-1932) *ideografik* gösterimleri,

Frege'nin *a priori felsefi* dil fikrinin güçlenerek günümüze ulaşmasında etken bir rol oynamıştır.

Doğal dilin müphemliğinden azade rasyonel bir bilim dili olmaya aday matematiksel dil, açıkça mantığın ilkelerine dayanan, *a priori* ve *felsefi* bir dil olarak on dokuzuncu yüzyıl sonlarına doğru Leibniz'in evrensel dil hayalinin de canlanmasına neden olmuştur. Leibniz bağlamında bir *lingua characteristican*ın peşinde olan Frege, notasyonları Frege'nin gösterimlerine baskın gelen Peano ve tüm bu çalışmaları *Principia Mathematica* adlı dev eserlerinin potasında eriterek sentezleyen Russell ve Whitehead'in çabaları sonucunda, günümüzde hemen her disiplinin biçimsel dilinde göze çarpan notasyonlarıyla matematiksel dil elde edilmiştir.

On dokuzuncu yüzyıl sonlarında, böyle bir rasyonel dilin ortaya çıkmasının nedenlerinin başında, mantıksal ve epistemolojik kaygılar olduğunu eklemekte yarar var. Biçimsel olan bu dilin çelişkilerden uzak tutulabilmesi de başkaca bir kaygı olarak eklenebilir. Bu nedenle, öncelikle yirminci yüzyıl başlarında Hilbert'in ortaya attığı gözle görülür problemlerin çözümü sonucunda matematik ve kullandığı dilin, tutarlı ve tam olduğunun ispatı aranmıştır. Her ne kadar verilen ispatlar tersi yönde bir sonuç doğurmuş olsa da tüm bu çabaların gösterilmesi ve bir rasyonel dilin peşinden gidilmesi, tezimizin üçüncü bölümünde serimlemeye çalışacağımız üzere pek çok biçimsel dilin yanı sıra beraberinde yapay zekâ projelerinin de doğmasını sağlamıştır.

Hilbert'in matematiğin sarsılmazlığına yönelik aradığı kanıtlara, "Löwenheim-Skolem teoremleri", Kurt Gödel'in (1906-1978) "eksiklik teoremleri" ve Alan Mathison Turing'in (1912-1954) "durma problemi" sonucunda geliştirdikleri kuramlar ışığında, günümüzde farklı bir ontolojik zeminde, biçimsel dillerin evrensellik sorgulaması yine bu kuramlar eşliğinde yapılabilmektedir.

İkinci bölümde mantık ve matematikte yaşanan gelişmeler neticesinde ortaya atılan biçimsel ve sarsılmaz bir rasyonel dil çalışmasının, Leibniz'e ve Leibniz'in yaşadığı yüzyıl öncesine götürülebilecek kadar eski kökenlere sahip olduğunu, önceki sayfalarda serimlemeye çalıştık. Bilimsel bir gerekçelendirme sonucunda, Leibniz'in evrensel dilinin en azından yukarıda belirttiğimiz üzere on üçüncü yüzyılda yaşamış Llull'un kombinasyon sanatına dayandığını biliyoruz.

Tezimizin ikinci bölümüne geniş yer ayıracak olmamızın nedeni, üçüncü bölümde yapmaya çalışacağımız evrensellik sorgulamalarını, matematiksel ve mantıksal temellere sahip kuramlar eşliğinde yürütebilmek ve Leibniz'den Turing'e uzanan biçimsel dil serüvenini, programlama dilleri üzerinden yapacağımız sorgulamayla bağlantılandırabilmektir:

“Yüzyıllara dayanan kusursuz dil arayışı ve özellikle *a priori felsefi* bir dil arayışı çabası olmasa, bu kuramları geliştirmek mümkün olur muydu?” (Eco, 1995b: 312 italik vurgu bana ait). Bu sorudan da anlaşılacağı üzere, Leibniz’e kadar gelmiş tüm rasyonel yapay dil çalışmaları Leibniz’in evrensel dilinde harmanlanmıştı, dolayısıyla bu evrensel dil ışığında gelişen mantık ve matematiğe dair ortaya çıkmış kuramlara değin geçirilen süreç aktarılmadan, günümüz problemi olarak gördüğümüz makine ontolojisi üzerinden bir evrensellik sorgulamasının yapılamayacağı kanaatindeyiz.

Tezimizin amacı, programlama dillerinin linguistik bağlamda evrenselliklerinin yanı sıra bu diller üzerinden yazılmış programların hesaplama evrenselliklerine dair iddiaların nedenlerine odaklanarak, özellikle üçüncü bölümde, bu iddiaların neden çürütülebilir olduklarını “özdevinim ve dil kuramı” ile daha genel olarak “hesaplama kuramı” bağlamında matematiksel mantık temelleri üzerinden yürüteceğimiz argümanlarla ortaya koyabilmektir.

Bir dilin sosyal zemindeki evrenselliği, herkesçe konuşulabilir veya konuşulabilir değilse de biçimsel yapısıyla anlaşılabilir olmasında yatar. Hatta biçimsel olduğu sürece bir evrensel dil, farklı dillerin konuşucularınca bozulma riskinden de korunaklıdır. Evrensellik sorgulamasını yapmak istediğimiz programlama dilleri, Leibniz’in evrensel dili için hayal ettiği biçimsellik ölçütlerinin ötesinde bile görülebilirler. Biz bu dilleri elbette doğal dillerle karşılaştırma niyetinde değiliz çünkü tıpkı Leibniz’in dilinin olmadığı gibi bu diller de sosyal zeminde ve olgulara dayanan yapıda değildirler. Evrensellik sorgulaması içinde olacağımız bu dillerin, makine gibi farklı bir ontolojik zeminden yükseliyor olması, evrensellik sorgulamamızı çeşitli ölçütler üzerinden yürütme zorunluluğunu doğurmaktadır.

Tezimizin üçüncü bölümünde, programlama dilleri üzerine yapma gayretinde olacağımız evrensellik sorgulamamız, mantıksal ve matematiksel olduğu kadar epistemolojiktir. “Bir bilgisayar dili düşüncesi artık sıradan bir olgudur” (Churchland, 2018: 39) ve evrenselliklerini sorgulamaya çalışırken, bu dillerin mahiyeti ya da özellikleriyle ilgili yapacağımız yorumlar, epistemolojinin alanına girmektedir.

Programlama dillerinin nasıl geliştirildiği ve tasarım amaçlarının yanı sıra bu dillerin paradigmaları ve niteliklerine dair sorgulamalarla başlayacağımız üçüncü bölümün devamında, epistemolojik kaygılarla soracağımız sorulara gelmeyi planlamaktayız: Programlama dilleri neden ve ne şekilde evrensel diller olarak kabul ediliyor ve gerçekten de öyle midir? Böyle bir soruya yanıt bulabilmek için çözümlemeci bir yöntemle bu tip biçimsel dillerin analizini ve sentezini yapmayı öngörüyoruz.

Yukarıda belirlediğimiz sorular ışığında, amacımızı gerçekleştirebilmek adına tezimiz tarihsel olduğu kadar, analitik yönetime de dayanmaktadır. Bu nedenle, öncelikle programlama

dillerinin teorik bilgisayar bilimlerinde kabul edildiği şekilde nesillenme süreçlerini ve bu nesillenmeye neden olan teknolojik gelişmelerin gerektirdiği paradigmaları serimlemeye çalışacağız. Günümüzde ulaştığı toplam beş nesil üzerinden, bu dillerin öncelikle ne tür dil grubuna dahil olduklarını belirleyeceğiz. Dahil oldukları dil grubu üzerinden bu dillerin sırasıyla linguistik, “hesaplama kuramı”, “hesaplama karmaşıklığı kuramı” ve Turing-*tamlık* ölçütlerinin yanı sıra ayrıca yapay zekâ bağlamında evrenselliklerini konu edineceğiz. Dayanağını “matematik, mühendislik ve linguistik temellerden alan” (Denning vd., 1978: ix) makine ontolojisi zemininde gücümüz yettiğince, dokunacağımız teorik bilgisayar bilimleri alanına ait kuramlar arasında bulunan “özdevinim ve dil” kuramı da evrensellik sorgulamasında destek alacağımız bir diğer kuramdır.

Son tahlilde, Leibniz’in evrensel dil hayalinin imkanı üzerinde tartıştığımız zemini genişleterek, günümüz teknolojilerinden yapay zekâ ve onun bir alt dalı olan “yapay öğrenme” alanlarında çok ses getiren başarılı sonuçlara ulaşmış “derin öğrenme” tekniği bağlamında da evrensellik sorgulamamızı yürütme gayretinde olacağız. Nihayetinde tüm sıraladığımız bu tür sistemler, birer biçimsel dil olan ve *a priori felsefi* diller sınıfının devamlılığını sağlayan *pasigrafik* yapıdaki programlama dilleri kullanılarak geliştirilmektedirler.

“Yapay öğrenme”nin bir tekniği olan “derin öğrenme” alanında göz dolduran gelişmelerin, beraberinde getirmesi söz konusu olan etik, pratik, ekonomik, psikolojik, sosyolojik veya hukuksal sorunlar tezimizin konusunun dışındadır. Biz yalnızca hesaplama evrenselliği bağlamında epistemolojik, mantıksal ve matematiksel bir değerlendirme yapmak üzere “derin öğrenme” tekniğine ve dolayısıyla yapay zekâyâ yer vereceğiz. Yukarıda bazıları sıraladığımız ve tezimiz bağlamında olmayan alanlarda yaşanan problemlerin, günümüz felsefe tartışmalarında çok önemli bir yeri olması gerektiğini vurgulamak isteriz.

Öyle ki yapay zekâ, insanlığı ahlâktan ekonomiye, bilimden sanata, felsefeden dine, hukuktan siyasete; kısaca hayatın her alanına sirayet edecek olan bir dizi mesele ile uğraşmak zorunda bırakacak görünmektedir. Bu durumda yapay zekânın insanlığa ne getirip ne götüreceği sorunu çok yönlü olarak tartışmaya açılmalıdır (Dönmez, 2020: 751).

Umalım ki felsefi tartışmalar, bu alanlarda kuramsal alt yapıların geliştirilmesinde önemli katkılarda bulunsun. Felsefi sağduyu olmadan gelişen teknolojilerin, yarardan çok zarar getirme olasılığı her zaman ihtimal dahilindedir.

Bizim ise sorgulamamızı evrensel dil temelinde yürütüyor olmamızın başat nedeni, bugün “derin öğrenme” alanında yaşanan gelişmelerin yanı sıra kuantum bilgisayarları ve büyük veri setleri için daha hızlı ve çok daha yoğun, insanın anlama ve yorumlama

kapasitesinin çok üzerinde karmaşık matematiksel işlemlerin yürütülüyor olmasıdır. Leibniz, Henry Oldenburg'a 1675 yılında yazdığı mektubunda şu ifadeleri kullanmıştır: "Haklı olarak bu kadar saygı duyduğumuz cebir, bu genel düzenin sadece bir parçasıdır. Oysa cebir o kadar başarılıdır ki istesek de hata yapamayız ve hakikat, bir makine yardımıyla kağıt üzerinde resmedilmiş gibi kavranabilir" (1969c: 166). Elbette, paradigması uygun programlama dilleri sayesinde, *yorumlayıcı* makinelerden programın yaptığı iş ve işlemlere dair çıktılar alabiliyoruz ancak bu çıktılarda yazan kodlamalar veya açılımlar öylesine karmaşık ki bu karmaşık yapı, anlam sınırlarımızı aşmaktadır.

Yukarıda tezimizi yazma nedenlerimizi sıralamış bulunmaktayız. Tezimizin bir diğer amacı da "derin öğrenme" tekniğinin gelişmesi sonucunda karşı karşıya kaldığımız sorunlara mercek tutmaya çalışmaktır: Bilme ve anlama yani episteme sorunu. Bu sorunun giderilebilmesi için de programlama dillerinin günümüz paradigmalarından farklı bir paradigma ile üretilmesi gerektiğini iddia ediyoruz. Bu da ancak yeni nesil (altıncı nesil) programlama dillerinin etkisiyle gerçekleştirilebilir.

Özellikle yapay zekâ ve "yapay öğrenme" ya da "derin öğrenme" alanlarında kullanılan beşinci nesil programlama dillerinde, üretilen sistemlerin karar algoritmalarının ne yaptığına dair açık ve net bilgiyi gözler önüne serebilen bir dil yapısı bulunmamaktadır. Dolayısıyla bizim için karmaşık hesaplama belirsizliklerini ortadan kaldıracı ve başka bir ontolojik zeminle aramızda kurmayı hedeflediğimiz anlamlı bilgi edinmenin ilk koşulu olan dilsel durumu geri kazanmak gerekmektedir.

Günümüzde sıklıkla kullanılan yapay zekâ sistemleri determinist ilerleyen, tündengimsel mantığa dayalı sistemlerin tersine olasılıksal, istatistiksel ve tümevarımsaldırlar. Programlama dillerinin çerçevesi belirgin yapılarının üzerinde yükselen yapay zekâ sistemlerinin arkalarında yatan mantığın belirginliğini gittikçe kaybediyor olmasından ötürü istesek de önümüzdeki yakın bir tarihte, bu tür sistemlerin iddiası içinde oldukları evrenselliklerini ne epistemolojik ne de mantıksal ve matematiksel olarak sorgulayabileceğiz. Nasıl yapıyorlar? Evrensel olduklarını hangi nedenlerden ötürü iddia edebiliyorlar? İşte bu sorular öyle görünüyor ki yakın gelecekte yanıtız kalabilirler.

Bu nedenle, teknoloji ve yeni geliştirilen tekniklerle üretilen algoritmaların karar mekanizmalarının determinist ölçütlerin dışında, tümevarımsal ve olasılıksal ilerliyor olması ve insan için zor problemler ya da makineler için zor işler olarak tanımlanabilecek alanlarda sergiledikleri başarılarından dolayı bu sistemlerin kabul görmeleri, yeni bir sorunu daha ortaya çıkarıyor: Epistemenin yerine inanç yerleşiyor. Bu alanın git gide belirsiz bir hal alıyor oluşunu izlemek ve ancak neler olduğunun doğruluğunu, çözümleme sınırlarımızı aşan

karmaşık matematiksel işlemlerin doğruluğuna bağlamak, bizi yalnızca inanç alanına itecektir. Bunun yerine epistemeyi öncelemeliyiz.

Hayatımızın olağan akışı içinde her şeyin birer algoritma sayesinde yolunda gittiği günümüzde, bu algoritmaların herhangi bir sebepten duracak olması, bildiğimiz anlamda hayatlarımızdaki konforu tehdit edebilecek düzeye erişmiştir. Makinelerin ve algoritmaların neden ve nasıl bu tip kararları verdiklerini bilemediğimiz gibi bu makinelerden bir *yorumlayıcı* aracılığıyla istediğimiz sonuçlar ise daha şimdiden makinenin insanın anlayamayacağı düzeylerde dillere ve işlem karmaşıklığına eriştiğini göstermektedir.

Leibniz'in farklı dil ve kültürde insanlar arasında gerçek bir iletişim dili olmasını hayal ettiği matematiksel kavram göstergeleri dili, günümüzde insanlar ve makineler arasında böyle bir evrensel dilin geliştirilmesi gereğiyle yeniden tartışılır düzeye gelmiş durumdadır. Umudumuz o yönde ki yeni nesil programlama dilleri, makinelerin ne yaptığını ve neden bunu yapmaya karar verdiğini anlamamızı sağlayacak şekilde bir paradigmaya sahip olarak tasarlanırlar.

Teknoloji geliyor ve “yapay öğrenme” tekniklerine uygun yapı sergileyen programlama dilleri önerildiği kadar, teknolojik gelişmeler nispetinde Türkiye’de yapılan çalışmalar da dahil olmak üzere, yapay zekâ tabanlı yeni mimariler de (Erel-Özçevik vd., 2020: 31) öneriliyor. Açıkçası önüne geçilemez bir hızla ilerleyen bilgisayar ve yapay zekâ teknolojilerinin gelecekte neler getirebileceği bilinmiyor.

Tüm bu teknolojik gelişmelere yeri geldiğinde değinmeye çalışarak, önceki bölümlerden elde edeceğimiz dayanaklar üzerinden sonuç bölümünde, Leibniz bağlamında bir evrensel dilin imkanı (veya imkansızlığı) nihayetiyle serimlenecektir.

Ayrıca biz de yakın gelecekte ne olacağının belirsizliği içinde, bu makinelerle aramızda anlam evreni oluşturabilecek paradigmayla tasarlanmış bir evrensel dil hayalini tıpkı Leibniz gibi canlı tutacağına benziyoruz.

BİRİNCİ BÖLÜM

CHARACTERISTICA UNIVERSALIS VE CALCULUS RATIOCINATOR'A GİDEN YOLDA LEIBNİZ VE DÖNEMİ

Yaşadığı zamanın çok ötesine uzanmış fikirleri ve verdiği ilham ile yirmi birinci yüzyılda, günümüze kalan fragmentlerinden görüldüğü kadarıyla, kendi döneminde takdire şayan gerçek bir zihinsel performans gösteren Leibniz, 1646 yılında doğmuş bir on yedinci yüzyıl filozofu, diplomatı, mantıkçısı ve daha birçok ünvan sahibidir. Biyografisini yazanlar ve üzerine çalışma yapan uzmanlarca “son evrensel deha” olarak anılmasının azımsanmayacak kadar çok nedeni sıralanabilir. Ölümünden sonra, ortaya çıkan birincil literatürden başkaca, ayrıca ölümüne kadar ömrünün ikinci yarısının büyük kısmını masa başında, Guelf Hanedanlığının Ortaçağ’a dayalı şeceresini (genealogy) hazırlamakla geçirdiği Hannover’de 1830’lar ve sonrasında ortaya çıkan mektuplar, kitapların kenarlarına yazılmış eskizler, kendisine ait çalışmalarının tutanakları ve diğer fragmentleri Leibniz’in dehasını ortaya çıkarmış ve onu daha doğru anlamamızı sağlayan ikincil literatürü günümüze taşımıştır. Hannover’de bir zamanlar dev kütüphanesi ile birlikte Leibniz’in yaşadığı ev şu anda Hannover Üniversitesine ait konferans salonu ve misafirhane olarak kullanılmaya devam etmekte olup, birkaç sokak ötede ise Leibniz’e ait kütüphane ve tüm arşive ulaşılabilir (Brown ve Fox, 2006: 20). Bu arşivlerin derlenip toparlanmasından sonra görülmüştür ki Leibniz’in sistematik olmadığı düşünülen felsefesi, sanıldığı aksine oldukça derin analizler sonucunda geliştirilmiş bağlamların ve disiplinler arası etkileşimin sağlam bir temel kazandığı, kendi içinde bütünlüklü ve çelişkisizdir.

1.1. Leibniz’i Yorumlamak

Leibniz’in geniş kapsamlı ve çok katmanlı örüntüye sahip felsefi sistemini bir bütün halinde yorumlamak ve bu sistemin oluşma nedenleriyle birlikte, gelişim sürecini derinlikli bir çalışmayla ortaya koymak başlı başına bir kitap ya da tez konusudur. Leibniz üzerine bir tez yazma iddiamız bulunmamaktadır. Bu nedenle, Leibniz’in evrensel dil hayalinin gerçekleşmiş olma ihtimalinin sorgulaması içinde olduğumuz tezimizde, Leibniz’in *characteristica universalis* (evrensel karakteristikler) ve *calculus ratiocinatoris* (mantıksal hesaplayıcı, cebirsel mantık) üzerinden bir hareket noktası belirleyerek ilerlemeye çalışacağız.

Akademik dünyada, Leibniz’in evrensel dilini teolojik bir yaklaşımla Ademi dil arayışı bağlamında değerlendirenler mevcuttur, oysa dinsel kaygıları dışında Leibniz, gerçek bir felsefi ve bilimsel dilin peşinde olmuştur. Bu dil, her şeye ve her insana hizmet edeceği için

kapsamı tüm bilimleri içine almakta; dilin kurulumu tamamen mantığa ve dilin işleyişi ise yüklemeler analizini yapmamızı sağlayacak olan aritmetik hesaplamaya dayanmaktadır. Bu dil, Leibniz'in felsefi sisteminde onun değindiği her alanla ilişki içindedir. Dolayısıyla tezimiz açısından önemli bir yorum zorluğu buradan başlamaktadır.

Leibniz'in felsefinin tek boyutlu ele alınışının ve yorum zorluğunun bir nedeni olarak ileri sürebileceğimiz ilk husus, filozofumuza ait bütünlüklü felsefesini yansıtabilecek eserlerinin bulunmaması; yazdığı iki kitaptan birinin yaşarken, diğerinin ise ölümünden ancak elli yıl sonra yayımlanmış olmasıdır. Leibniz, uzun ömrü boyunca iki büyük hacimli kitabından yalnızca *Theodicy*'nin yayımlanmasına izin vermiştir. Yayımlanmak üzere tamamladığı ve John Locke'ın (1632-1704) *İnsanın Anlama Yetisi Üzerine Bir Deneme*¹ (1690) adlı kitabına eleştiri olarak kaleme aldığı *İnsanın Anlama Yetisi Üzerine Yeni Denemeler*²'in (1704) bitimi ise Locke'un ölümünden sonrasına denk geldiği için Leibniz cevap hakkını kullanamayacak olan filozofun hatırına, kitabını yayımlamaktan vazgeçmiştir. Felsefesi, yazdığı kitaplardan net olarak çıkarsanamayan Leibniz'in sisteminin anahtar yapıları, ancak çok güvendiği mektup arkadaşlarıyla yaptığı yazışmaları da takip edilerek ve bağlantılandırılarak serimlenebilmektedir. Hacimli mektuplarının derlemeye başlanması ise Leibniz'in ölümünden çok sonra, yaklaşık yüz elli yıl sonrasını beklemiştir.

Böylece bir diğer husus, Leibniz'in mektuplarının ve diğer fragmentleri ile eskizlerinin derlemesinin on dokuzuncu yüzyıl sonlarında bile henüz tamamlanmamış olmasıdır. Binlerce sayfalık çalışmalarının ve makalelerinin dışında, zamanının büyük dehalari, saray erkanı ve asilzadelerle mektuplaşmalarının oluşturduğu ikincil literatür sayılan ve 1875'lerden sonra derlenen yazınsal mirası sayesinde, Leibniz'in sistemi görünür hale gelmeye başlamıştır.

Bilinmektedir ki Leibniz, mektuplarını belli bir sistematik içinde yazmış ve yanıtlamıştır. Sistemini parçalı yansıtırken, karşılık olarak sunulan eleştirilerden faydalanma konusunda oldukça pratik ve pragmatik bir tavır sergileyen Leibniz, mektuplaştığı kişilerin o konudaki maharetlerine, uzmanlık veya ilgi alanlarına ve ayrıca zekâlarına göre seçim yapmaya özen göstermiştir. Deleuze'e göre, bu anlamda Leibniz'i yorumlamak oldukça zorlaşmış ve ayrıca filozofun kendisiyle çeliştiğine dair izlenimler de uyanmıştır.

Leibniz kime mektup yazdığına göre ya da hitap ettiği kesime göre bütün sistemini şu ya da bu düzeyde sunar. (...) Bu düzeyler, bu katlar örtük bir şekilde bizzat Leibniz'in metinlerinin parçalarını oluşturdukları için ortaya büyük bir yorum problemi çıkar. Karmaşıktır, çünkü metnin sistemin hangi

¹ *An Essay Concerning Human Understanding*. Tezimizde "Deneme" olarak kısaltacağız.

² *New Essays On Human Understanding*. Tezimizde "Yeni Denemeler" olarak kısaltacağız.

katına, düzeyine tekabül ettiği düzeyi hissedilmeden Leibniz'in o metnine asla yaslanamayız. (...) Öyle ki Leibniz'in metninin yer aldığı düzeyi hesaba katmazsanız sürekli olarak kendisiyle çeliştiği hissine kapılabilirsiniz-oysa asla kendisiyle çelişkiye düşmemiştir (Deleuze, 2007: 21-22).

Son bir husus olarak, çok hacimli çalışmalarının incelenmesi ve derlenmesi neticesinde ortaya çıkan, hemen her alanda oldukça verimli ve hala geçerliliğini sürdüren fikirleri, Leibniz'in felefesinin anlaşılmasında çok katmanlılık ve disiplinlerarası bir bağın gerekliliğini doğurmaktadır ki bu da Leibniz'i yorumlanması zor bir filozof yapmaktadır. Bu sebeptendir ki Leibniz, ayrı alanlara hizmet eden akademisyenler için verimli birçok araştırma konusu sunduğu kadar, ilgili konunun bütünlüklü bir bakış açısıyla yorumlanması gerekliliği doğduğundan, şayet bütünlüklü bakışla bir çözümleme gerçekleşmez ise sonuç itibarıyla hem Leibniz felsefesi zarar görerek ikircikli bir görünüm sergilemekte hem de ilgili konunun aktarılması eksikliğe uğramaktadır.

Daraltılarak verilebilecek bir örnek olarak; matematik tarihi ve felsefe tarihinde büyük öneme sahip olan Leibniz, birbirlerini sürekli beslemiş olan matematik ve felsefe gibi iki alan için bile farklı noktalardan ele alınmakta ve yorumlanmaktadır. Felsefi kaygılarla Leibniz'i ele alan düşünürün, matematik bilgisinin Leibniz'i anlaması ve yorumlaması için yetersiz olması ya da matematikçinin, Leibniz'in matematik keşiflerinin altında yatan metafizik, fizik veya dil kaygılarından habersiz olması ortaya parçalanmış bir Leibniz çıkarmaktadır. Diğer çalışma alanları ve disiplinlere genişletilebilecek olan bu sorun, hangi Leibniz'i ele aldığımızı sorgulatmaktadır.

Biyografi yazarları arasında bugüne kadar literatür ve bağlantılı metinler üzerinden yapılmış araştırmalardan en geniş kapsamlısını sunan Maria Rosa Antognazza'nın (1964-...) Leibniz biyografisinden anladığımız kadarıyla Leibniz, kuramsal ve uygulamalı disiplinler açısından felsefeden matematiğe, doğa bilimlerinden filoloji ve şiire, tarih ve teolojiden maden mühendisliğine ve mucitliğe kadar uzanmış geniş bir disiplinler yelpazesinin hem kuramcısı hem de pratisyenidir.

Felsefeden matematiğe diye daraltılarak tahlil edilen yazıları bilimler ansiklopedisinin sınırlarını aşyordu: Astronomi, fizik, kimya, jeoloji, botanik, psikoloji, tıp, doğa tarihi, hukuk, etik, siyasi felsefe, tarih, antik eserler, Çince, Almanca ve diğer Avrupa dilleri, dilbilim, etimoloji, filoloji ve şiir; doğa teolojisi ve vahiy teolojisi; dini düşüncelere dalmanın yanı sıra bir dizi pratik olgularla uğraşmak: Hukuk reformundan kiliselerin birleştirilmesine, diplomasi ve uygulamalı politikadan kurumsal reforma, teknolojik gelişmeye ve bilimsel cemiyetlerin, kütüphanelerin ve kitap ticaretinin düzenlenmesine varıncaya değin (Antognazza, 2013: 2).

Leibniz üzerine kitap yazabilecek kadar araştırmalarda bulunmuş olan Russell ise Leibniz'in felsefesini çelişkiler üzerinden olmasa da iki farklı temsile sahip olması konusunda bir anlamda iki yüzlü bulmuştur. Leibniz felsefesinin iki yönlü sisteme sahip olduğunu iddia eden Russell'a göre "ilki, iyimser, ortodoks, fantastik ve sığdı; son dönem editörler tarafından el yazmalarından yavaş yavaş ortaya çıkarılan diğeri ise derin, tutarlı, büyük ölçüde Spinozist ve şaşırtıcı derecede mantıklıydı" (1945: 581). Russell'ın Leibniz üzerine çalışmaları yaptığı esnada, Leibniz'in henüz basılmamış yayınlarının varlığı hesaba katılmalıdır. Günümüzde daha geniş kapsamlı yorumlar ve editörlükler sayesinde Leibniz'in kendi yazılarına daha kolay erişim sağlanabildiğinin altı çizilmek suretiyle, filozofumuzun kendi görüşleri üzerine yazdıkları ve yazdıklarına dair yapılmış yorumlar birlikte dikkatle incelendiği takdirde rahatlıkla denilebilir ki filozofumuz, kendi çağının gereksinimleri doğrultusunda düşüncelerini geliştirmiş; fikirlerinin toplumsal yansımalarını hesaba katmış bir düşünür olarak sistemini üretmiştir.

Leibniz'in yaşadığı dönemde Russell'ın yaptığı türden yorumlarla karşılaşmış olabileceğini düşündürten, kendisiyle ilgili oldukça dikkat çekici yorumuna değinmekte yarar var. Leibniz'in Vincentius Placcius'a gönderdiği mektubunda (1696), bir düşünür olarak yaşadığı dönemde (günümüzde de yukarıda söz edilen nedenlerden ötürü) yeteri kadar anlayamadığına dair izler bulunmaktadır. Mektubunda, "Beni sadece yayımladığım eserlerden tanıyan kişi, aslında beni tanımıyordu" (Leibniz'den aktaran Antognazza, 2013: 288) diye yazan Leibniz, dönemi ve sonrası için bir dipnot bırakmış gibidir. Bu nedenle, tezimiz bağlamında Leibniz üzerinde yaptığımız çalışmalarda tek boyutlu yaklaşımdan kaçınarak, bizi yorum zorluğundan kurtarabileceğine inandığımızdan, evrensel dil hayalini gerekçelendirmek için Leibniz'i, diğer filozoflarla çatıştığı ve hemfikir olduğu noktalardan felsefi olarak ele almanın yanı sıra Leibniz'i yaşadığı dönemin gerektirdikleri çerçevesinde sosyolojik olarak ele alarak, filozofumuzun felsefesini, belirginleştiği haliyle aktarmaya çalışacağız. Böylece, bu bölümde Leibniz'in evrensel dil hayalini sosyolojik zeminden yürüterek, nihayetinde *characteristica universalis* ve *calculus ratiocinator* için kurulmuş epistemolojik dayanakları serimlemeye çalışacağız.

1.2. Leibniz'in İçine Doğduğu Yüzyıl ve Avrupa Coğrafyası

Skolastik dönemden modern döneme geçiş diye de karakterize edilebilecek Rönesans döneminde keşiflerin artması, ticaretin yön değiştirmesinin yanı sıra ulus bazında kullanılan yerel dillerin değer kazanması ile ters orantılı olarak skolastik düşünce zayıflamıştır. Rönesans etkisiyle (özellikle İtalya'da Dante tarafından yerel dilin Latince yerine

kullanılmaya başlamasının siyasi sonuçlarla paralelliği) yeni olan her şeyin dilsel manada güç kazanması, bilim ve din temsiline dili sayılan Latincenin etkisinin azalmasıyla birlikte kilisenin eski gücünü yani merkezi otoritesini kaybetmeye başlaması, bilimsel alanda yeniliklerin açık zihinlere hitap etmesi ve düşünce çevrelerince olumlu karşılık bulması, felsefede de ciddi yöntem sorgulamalarını beraberinde getirmiştir. Rönesans'ın etkisiyle bu yüzyıl, bir çılgılık yüzyılı olarak (Deleuze, 2007: 29) ulusal dillerin bilim ve edebiyatta kullanılmaya başlanmasıyla³ birlikte, kilise merkezli yani skolastik temelli bilim ve bilim dili olarak kabul edilen Latincenin yanı sıra, özgün dilde yazılmış yazıların çevirilerinin de artmasına neden olacak gelişmelere tanıklık etmiş, *dil* de bu yüzyıl bağlamında ileride değineceğimiz üzere başka bir form kazanmak üzere ele alınmaya başlanmıştır.

Avrupa'nın gördüğü son din savaşı olan Otuz Yıl Savaşları ardından toparlanmaya çalışan yaralı ülkeler arasında Almanya gibi çok zayıf düşmüş olanların bilimde, sanatta ve edebiyatta geç ilerleme katettiği bu dönemde; edebiyatta İtalyanca güçlü diller arasında sayılırken, savaşın güçlüleri arasındaki ülkelerin özellikle bilim ve diğer alanlarda daha hızlı ilerlemesi neticesinde kendi dillerinde yazılmış bilimsel yayınların yayılmasıyla birlikte Avrupa dilleri arasında Latincenin yanı sıra Fransızca ve sonrasında İngilizce gibi diller ise güçlü bilim dilleri arasına girmiş görünmektedir.

Savaştan en ağır yara alan Almanya ve diğer ülkelere nazaran, savaş sonrası en güçlü Fransa ile savaştan pek de etkilenmemiş İngiltere gibi ülkelerde yeni dünya görüşü sayesinde bilim daha özgür bir şekilde filizlenip dallanıp budaklanmış, felsefe ise bilimin yöntemleri ışığında, deneycilik ve tümevarımsal doğrulukların temelinde epistemolojiler gelişmiştir. Böyle bakıldıkta, bu yüzyıl Avrupası her şeyini yıkıp tekrar inşa etme ve geçmişten gelen değerlerine sahip çıkma arasındaki ikilemi askeri, dini, politik alanların yanı sıra özellikle bilim ve felsefe alanında yaşarken, öte yandan Avrupa'nın genelinde hala sürüyor olan mezhep çatışmaları mevcuttur ve bu durum ülkeler arasında bilimsel, iktisadi, felsefi açıdan ciddi eşitsizliklere yol açmıştır. Özellikle Almanya'da "Protestan kiliselerinin bölünmüşlüğü İmparatorluk düzeyinde müzakereleri zorlaştırırken" (Antognazza, 2013: 19) topraklar da hanedanlıklar arasındaki güç mücadelesinden dolayı parçalanmıştır. Öyle görünüyor ki bu dönem en zayıf düşmüş ülke olan Almanya'da mezhep çatışmaları, uzlaşımı zorlaştırmakla birlikte bilimsel gelişmelerin de önündeki en büyük engeldir.

³ Mehmet Fatih Doğrucan, *Medeniyet Dili Olarak Türkçe* isimli kitabında Latincenin önemini kaybetmesinde matbaanın rolünün büyüklüğüne (2020: 69) vurgu yapmaktadır. Ona göre, sınırlı sayıda seçkine has bir dil olarak kabul edilen Latincenin karşısında, seri üretim yapan matbaanın gerektirdiği bir pazar söz konusu olmuş, dolayısıyla bu pazar, müşterisi en fazla olan dillere odaklanarak kurulmuştur. Böylece filozoflar ve düşünürler de yazılarını ulusal dillerde kaleme almaya başlamıştır (2020: 69).

Mezheplerin uzlaşması bir ölçüde teolojik uzlaşya bağlıydı ve tam da felsefi bir devrimin yaşandığı dönemde ortaya çıkan bu teolojik tartışmalar iyice arapsaçına dönmüştü. Orta Avrupa'nın askeri ve siyasi bir çalkantıya sürüklendiği evrede Batı Avrupa aynı ölçüde sarsıcı bir isyan dönemine girmişti. Batlamyus [Ptolemaios], Kopernik ve Tycho Brahe'nin rakip dünya sistemlerini içeren dar meseleleri dallanıp budaklanıp doğa felsefesine temel yaklaşımlar arasında çatışmaya dönmüş ve Aristoteles'in saygın otoritesinin karşısına Galileo, Bacon, Gassendi ve Descartes'ı yerleştirmişti (Antognazza, 2013: 22).

Nicolaus Kopernik (1473-1543), Galileo Galilei (1564-1642), Johannes Kepler (1571-1630), Pierre Gassendi (1592-1655) gibi bilim insanlarının çığır açıcı keşifleri ile birlikte matematik temelli niceliksel formlar, düşünsel bazda milat sayılabilecek bir şekilde bilimde yeni bir anlatı ve aktarım sisteminin temel taşlarına dönüşmüştür. Bu durum, Aristoteles fiziğinin karşısına, modern düşüncenin İlkçağ düşünürlerinin ilk kaynaklarına ulaşarak edindiği bilgilerin tohumu olan mekanik doğa görüşünü çıkarmıştır. Aynı bilimsel devrimlerden etkilenen farklı coğrafyanın düşünürleri bu mekanik doğa görüşünde iki temel ayırım üzerinden yürüyen sistemler geliştirmişlerdir. Batı Avrupa'nın düşünsel devriminin temelinde tümevarım, deney ve gözlem yatarken, Orta Avrupa'nın düşünsel devriminin altında matematiksel düşünme biçimi, dolayısıyla tümdengelimsel yöntemin sarsılmazlığı yatmaktadır.

Kopernik, Tycho Brahe ve Kepler'in yazılarında astronomi ve kozmolojide bir devrim olarak başlayan şey, Galileo, Descartes ve Gassendi'nin eserlerinde fizik ve metafiziğe kadar uzanmıştı. Bu filozoflar, Kopernik kozmolojisiyle uyuşan yeni bir fizik arayışı içinde gökyüzünü tarif etmekte başarılı olduğu giderek daha fazla belli olan nicel ifadelerle dünyanın yapısını ve hareketlerini açıklama çabasına giriştiler. Fiziksel olguları, nihayetinde maddi olmayan "tözel formlarla" açıklayan Aristoteles'in niteliksel fiziğinden tümüyle kopan mekanikçi filozoflar, yersel ve göksel fiziği yalnız hareket halindeki cisimlerin geometrik özellikleriyle açıklama peşine düştüler (Antognazza, 2013: 53).

İngiltere'de Kraliyet Cemiyeti⁴ ve Fransa'da Paris Kraliyet Bilimler Akademisi⁵ adı altında bilim cemiyetleri kurulmuş ve üyelerinin saygın topluluklar oluşturduğu bu ve benzeri bilim cemiyetlerinin mensuplarınca, çoğunlukla pozitif bilimler ve teoloji üzerine çalışmalar yapılmıştır. Bu tip tanınmış Kraliyet Cemiyetlerinin dışında, dönemselsel olarak zaman zaman zengin kişilerin himayesi altında kaldıkları zenginlerin evlerini kullanan düşünürlerin, kendi çapında fikirlerini takdim ve takas etmek üzere mektuplaşma ya da buluşma ayarladıkları durumlar da mevcuttur.

⁴The Royal Society of London for Improving Natural Knowledge

⁵Académie Royale des Sciences

1623'te Paris'teki manastır odasından Mersenne, Gassendi ile yakın çalışarak Kepler, Galileo, Campanella, Descartes, Hobbes, Torricelli, Fermat ve diğer matematikçiler ve zamanın bilimsel araştırmacılarına bağlanan bir mektuplaşma çevresi oluşturdu. Mersenne örgütsel liderdi (...) Mersenne'nin çevresi 1648'e kadar sürdü; 1657'de hayatta kalanlar, zengin patronların, Montmor'un ve Kartezyen lider Rohault'nun evlerinde düzenli toplantılara başladı. Bu çevre de bilimsel ve matematiksel keşifleri duyuran bir mektup takas alanı olarak Descartes'ın yayımlanmamış el yazmalarını Malebranche ve Leibniz'e sunmada hizmet etti (Collins, 2002: 528).

Disiplinlerarası diyalogun bir şekilde zıt görünümlü birçok disiplini ayrılmak üzere bir araya getirdiğini ve fizik-metafizik ayrımından, simya-kimya ayrımına kadar ya da inanç-iman bağlamında tarikat veya mezhep ayrımlarının Avrupa kültüründe evrensel din olarak görülen katoliklik ışığında sonlandırılabilmesine dair yeni zeminsel arayışların olduğu da gözlemlenmektedir. Bu cemiyet mensuplarının ürettiği kavramlar, geliştirdikleri kuramlar ve icatları, dahil oldukları disiplinler ışığında bilimin yüceltilmesini sağlarken, öte yandan felsefi düşünme biçimine de yöntemleri bağlamında etki etmiş, felsefe açısından artık ontolojik kaygılardan çok epistemolojik kaygılar neticesinde zihinsel aktivitenin doğru uygulanışı olarak da görülebilecek, ihtiyaç duyulan yöntem arayışları gereği insan, beden-ruh ayrımına tabii tutulurken insan zihni ve bilinci merkeze oturmuştur.

Nasıl bildiğimiz, bildiğimizi nasıl ve nereden anladığımız, bildiğimiz şeyin doğruluğunu nasıl ispatlayacağımız ve bildiğimiz şeyi karşı tarafa kelimelerle açıklarken yöntemimizin hareket noktasının ne olduğu ya da ne olabileceğine dair sorgulamalar, özellikle modern çağın düşünsel bazda miladı sayılabilecek René Descartes (1596-1650) ve modern çağın filozoflarını oldukça etkilemiş Francis Bacon'un (1561-1626) izinde, artık temel sorulara dönüşmüştür. Bacon ve Descartes, sırasıyla bilimsel devrimin gözlemsel ve matematiksel taraflarındaki ağırları temsil eden (Collins, 2002: 562) hareket noktaları olmuşlardır. Başarıyı genel felsefede bulan Bacon için liderlik etmek istediği alanın temelinde daha çok bilimsel yöntem sorunu diyebileceğimiz bir sorgulama yatarken; Descartes'ın başarısının altında yatan sorgulama bilimsel, felsefi ve pratik maksatlı olmuştur (Collins, 2002: 563).

Bu yüzyıl, epistemeyi doğru zeminde kurma ve buna uygun kavramları yaratma arayışı içinde, gerçek bir Rönesans ruhuyla eskiden kopup yeniyi oldurmanın hikayesini barındırmaktadır. Sonsuz fikrinden kurtulamayan filozofların Tanrı'ya yaslanmaya devam eden ben merkezli epistemelerini, dilin doğru kullanımları nispetinde serimlemeye çalıştığı bir yüzyıldır. Bu yüzyıl, üzerine çokça yazılan bilimden felsefi paradigma değişimlerine, askeri dönemlerden iktisadi dönüşümlere kadar ciddi bir külliyata sahiptir. Bununla birlikte, yine bu yüzyılda, düşünme etkinliğinin matematik yapmak gibi bir şey olduğuna dair filozof

görüşleriyle birlikte, bu yolla eğer doğru yöntem belirlenebilir ise tıpkı akıl hakikatleri sayılan matematik önermeleri gibi zihnimize beliren düşüncelerde, açık ve seçik önermelerin hakim olabileceğine; bu yolla düşüncede karmaşıklığın giderilmesinin mümkün olacağına dair felsefi görüşler de oldukça değer kazanmıştır.

Ortaçağ'da Âdemi dil arayışı çerçevesinde mistisizme dayalı matematiksel bir inşa olarak da değerlendirilebilecek, “İbranicede ‘gelenek’ anlamına gelen ve Kabbala olarak da bilinen, dilden dile geçen dinsel kaynaklarla yaygınlaşan batını Yahudi mistisizmi” (Antognazza, 2013: 462) Kabalacılık⁶, kök dil arayışı olarak bir değere sahip iken her dönemde yeri farklı şekillerde belirlenen *dil* de bu yüzyılın değişen paradigmalara bağlı olarak tıpkı diğer her şey gibi matematiksel düşünme bağlamında gündemdeki yerini almıştır.

Bu yüzyıl, doğru ve yalın aktarımın önem kazandığı, Âdemi dil arayışlarından azade, evrensel nitelikte yeni bilimsel ve felsefi yapay dillerin gelişimi açısından değerlendirilebilecek kadar önemli ürünler çıkarmış olup, üçüncü bölümde değineceğimiz günümüz yapay zekâ ve “yapay öğrenme” uygulamalarında kullanılan beşinci nesil programlama dillerinin, diğer nesillerden olan üçüncü ve dördüncü nesil bilgisayar programlama dillerinin ve ayrıca pek çok çeşitli mantığa dayalı biçimsel dillerin kökenini bünyesinde barındırmaktadır.

1.3. Leibniz’i Etkileyen On Yedinci Yüzyıl Düşünürleri ve Dilbilimcileri

Eklektik sistemine dahil ettiği düşünceler, düşünürler, icatlar ve meslekleri sığdırmaya çalışırken, Leibniz’i tezimizde ele almamızın gerçek sebebinden, filozofumuzu yorumlama zorluğundan dolayı, uzaklaşabiliriz. Tezimizin ana temasının *evrensel dil* olması nedeniyle bu yüzyıl Avrupası’nın bazı filozof ve dilbilimcilerini Leibniz’in perspektifinden değerlendirmeye çalışacağız. Yaşadıkları topraklar ve yüzyıl bağlamında düşünme etkinliğinde gerekli doğru yöntem arayışları içinde olan Bacon, Descartes, Hobbes ve deneyci felsefenin öncülerinden Locke gibi filozofların yanı sıra Port-Royal Mantığı birer alt başlık altında kısaca dil bağlamında değerlendirilecek olup, aksiyomlara dayalı geometrik yöntem ile felsefesini ikame eden Baruch Spinoza (1632-1677) ise Leibniz başlığı altında yeri geldiğinde ayrıca ele alınacaktır.

⁶Umberto Eco, *Avrupa Kültüründe Kusursuz Dil Arayışı* isimli kitabında Kabalacılığın, Mesih’in gelmesinden sonra var olan her dilin, sonunda içinde eriyeceği kutsal dilin bir tür arayışı (1995a: 44) olarak da tasvir edildiğini ve bu yolla, diller arasında farklılıkların ortadan kalkacağına dair mistik bir inanç olduğunu aktarır. Aynı kitapta, Kabalacılığın ciddi bir kombinasyon sanatına dayandığı bilgisi de mevut olmakla birlikte, bu sanata dair ileri çalışmalar yapmak isteyenler için örneklendirmeler bulunmaktadır.

Diğer yandan evrensel dil arayışlarını dini temeller üzerinden değil, zamanın ruhuna yakışır şekilde zihnin bulanıklığını yok etmeye yönelik çabalar içinde yürütmeye çalışan dilbilimcilerden Dalgarno ve psikopos Wilkins, aynı alt başlık altında birlikte değerlendirilmeye çalışılacaktır.

Bütün dillerin tek bir dile indirgenmesine yönelik poligrafik (herkese açık uluslararası bir alfabe projesi) evrensel diller üzerinde çalışan Alman Cizvit Athanasius Kircher (1602-1680) ileride değineceğimiz üzere, aralarında gerçekleşen mektuplaşmalar neticesinde Leibniz'in entelektüel merakını Çin mistisizmi ve Çince üzerine çekmiş biri olarak, tezimizde yeri geldikçe konu edilmeye çalışılacaktır.

Sözü edilen isimler ve bu isimlerin yapıtları üzerinden entelektüel etkilenim yaşayan Leibniz'i aldığı eğitim ile birlikte değerlendirecek olursak, bu etkilenim Leibniz'in eklektik felsefesinin yanı sıra evrensel dil hayali için yönünü belirleyen, döneminin önemli etkenler bütünü olarak karşımıza çıkmaktadır.

1.3.1. Francis Bacon (1561-1626)

Filozof olmanın dışında hukukçu ve siyaset adamı Bacon'un yaşadığı dönem için en önemli özelliği, bilime büyük önem vererek felsefesinin temellerini deney ve gözlem üzerine kurmuş olmasıdır. Edebi bir esere dönüşmüş *Deneme* (Essay [1597]) isimli yapıtı ve bilimsel ütopya örneklerinin başlıcalarından sayılan *Yeni Atlantis* (The New Atlantis [1626]) isimli yapıtının yanı sıra bilime dayalı akıl yürütme yaparak gündelik kullanıma mahsus dilin yanlış yönlendirmelerinden sıyrılabileceğimiz yöntemlerin sıralandığı *Novum Organum* (1620) isimli yapıtı, döneminde büyük yankı uyandırmıştır. “Bacon, bilimsel devrimin altında yatan sosyal özelliği, hızlı keşif biliminin ortaya çıkışını en açık şekilde yakalamış” (Collins, 2002: 564) bir filozof olarak, çağdaşlarını açıkça etkilemiştir.

Özellikle *Novum Organum*'da, *idola*⁷ sebebiyle yani gündelik dilin yanılsamalarıyla insan zihninin bulanıklaştığı; bu sebeple epistemeye ulaşmak için doğru bir yardımcı mantık ya da yöntemin bulunması gerektiğine dair söylemler mevcuttur. Bu eserde, doğru bir yönlendirme yapabilmek adına, zihinsel bulanıklığa sebep olduğu aktarılan *idolalar* dört ayrı başlık altında toplanmıştır: Bacon, eserinin alt başlıklarında belirttiği birinci tür için “kabilenin idolleri”, ikinci tür için “mağaranın idolleri”, üçüncü tür için “çarşı idolleri” ve dördüncü tür için “sahne idolleri” (2003: I. 40) tanımlamalarını uygun bulmuştur. Bacon, yaptığı ayrımlamalar neticesinde, zihinde berraklaşmayı sağlayacak ilk aracın *dil* olduğuna

⁷ *Idola* latince kökenli olup, “illüzyon” ve “tabu” gibi anlamlarının yanı sıra ayrıca, “zihindeki yanlış fikirler” ve “idol” anlamına gelmekte olup tezimizde, bu dört anlam da yerine göre kullanılmaktadır.

vurgu yapmaktadır. Bacon'a göre, dile dayalı tüm dışavurumlar, halk tabakasının anlayışına göre düzenlenmiş olup, bu şekilde yanlış ve uygunsuz bir düzenleme zihni olağanüstü ölçüde engellemektedir (2003: I. 43). Bu eserde Bacon, dilin hafızamıza oturan gündelik kullanımının anlam içeriklerinin zamanla oturmuş çok anlamlı ve müphem yapısının, hafızaya alındığı şekliyle zihnin düşünsel süreçlerini etkilediği gerekçesiyle zihnin hakikate, epistemeye gitmek için çoklukla yanlış temelden bir başlangıç yaptığını dair görüşlerini sıralamaktadır. Böyle bakıldıkta, Bacon'un dil anlayışının temelinde, doğal dilleri kullanım biçimimizden kaynaklı olduğunu bildiğimiz kusurlardan biri sayılan, öylece kabul edip doğrulamaya gerek duymadığımız anlam içerikleri ve bir diğeri olan, anlam belirsizliklerinin giderilmesi gerektiği fikrinin yattığı görülmektedir.

Kusursuz dile dayalı ihtiyacın, geçmişte olduğundan farklı gerekçelerden kaynaklandığı bu yüzyılda, bir filozofun gözüyle serimlemeye çalıştığımız şekliyle dili arındırma ihtiyacının, bilimin doğru seyirde ilerlemesi için bir esas olduğu söylenebilir. “Bilimsel yöntemin yenileyicisi Francis Bacon, kusursuz dille yalnızca marjinal olarak ilgilenir, ancak bu marjinal gözlemlerinin kendisi açısından kayda değer felsefi bir önemi vardır” (Eco, 1995a: 213). Böylelikle denilebilir ki düşünce faaliyetimizin hafıza, tecrübe ve dile dayalı olmasının kabulü; zihnimizi gereksiz yere yoran doğal dillerin kusurlarından kurtaracak yeni yapay bir dilin, Yeniçağ'da bilim yapabilmek adına gerekli olduğunun bir göstergesidir.

1.3.2. Thomas Hobbes (1588-1679)

Materyalist felsefesinin bir sonucu olarak, dünyayı ve insanı mekanik hareketler bütünü gibi resmettiği düşüncesini geliştirirken, Galilei'nin doğa görüşünden ve Bacon'un emprizminden etkilenen Hobbes yine de deney ve gözleme dayalı yöntemden ziyade bir matematik yöntem hayranıdır. “1630'larda Gassendi ve Galileo ile tanıştıktan kısa bir süre sonra Hobbes, zamanın çalkantılı havasının üstesinden gelmek için tasarlanmış bir hükümet sistemi de dahil olmak üzere her şeyi bedenlerin hareketinden çıkarmak için büyük bir plan” (Collins, 2002: 595) geliştirmiştir. Kendisi de iyi bir matematikçi olan Monarşi yanlısı Hobbes, siyasi bir çalkantının ortasında İngiltere'den kaçarak 1640-1651 yılları arasında Fransa'da geçirdiği on bir yıllık zaman aralığının sonunda, geliştirdiği planının sonuçları olan ilk baskısı 1642 tarihli, Latince kaleme alınmış *De Cive* isimli yapıtının yanı sıra *Leviathan* (1651) isimli yapıtını da ortaya çıkarmıştır.

Hobbes'un Devonshire Earl'ü William'a ithafen güç ve yurttaşlık üzerine kaleme aldığı *De Cive* isimli yapıtında, “insan insanın Tanrı'sıdır ve insan insanın kurdudur” (2018:

1) sözü, günümüzde hala söylemden öte bir bakış açısını ifade etmek için kullanılmaktadır. Hobbes'a göre "insan insanın Tanrısı'dır" sözü, yurttaşların; "insan insanın kurdudur" sözü ise devletlerin arasındaki ilişkiler için doğrudur (2018: 2). Yurttaşlık üzerine kaleme aldığı bu eserinde Hobbes, gerçek bilgelige vurgu yaparken felsefeye başvurmakta ve adlarla yürütülen tümevarımsal bir yöntem önermektedir.

Gerçek bilgelik, kısaca, her konuda gerçeğin bilgisidir. Kesin ve sonlu isimler yoluyla ortaya konan şeylerin hatırasından çıkarıldığı için, konusu, içimizdeki anlık parlamalar değil ama doğru akıldır, eş deyişle buna felsefe denir. Zira felsefe, tekil şeylerin gözleminden evrensel ilkelere giden yolu açar (Hobbes, 2018: 2).

Hobbes'un batı siyasetini oldukça etkilemiş olan bir diğer önemli eseri *Leviathan*, Hobbes'un ele aldığı "toplum sözleşmesi" kuramıyla günümüz dünyasında bile halen ciddiyetini ve başucu kaynağı olma pozisyonunu korumaktadır. Materyalist felsefesi ve mekanik dünya görüşünün yanına eklenen katı nominalist tutumu neticesinde hem dünyayı hem de insanı birer makineden ibaret sayan Hobbes'un özellikle *Leviathan*'da değindiği şekliyle, zihnimizden geçen düşüncelerimizle yaptığımız şey (bunu adlarla yapıyoruz), bir tür hesap yapmak gibidir.

Adların konusu, bir hesaba girebilen veya bir hesapta düşünülebilene ve birbirine eklenip toplanabilene veya birbirinden çıkarılıp bir bakiye bırakan her şeydir. (...) Yunanlılar hem akıl hem de söz için λόγος sözcüğüne sahiptiler. Onlar, akıl olmadan sözün olmayacağını düşünmediler fakat söz olmadan akıl yürütmenin olamayacağını düşünüyorlardı ve akıl yürütme eylemini, bir söylemden diğerine geçişin sonuçlarının tümüne işaret eden *sylogism* olarak adlandırdılar (Hobbes, 1996: 24-25)⁸.

İnsana dair sözel anlatı ile zihinsel bağlantı kurmanın yani düşünürken adlara tabi olmanın, tecrübe edilen şeylerin bir tür hafızaya alınması ve iletişimde kullanılması olduğunu dile getirirken, Hobbes'un adlara yüklediği görev fayda odaklıdır.

Konuşmanın genel yararı, zihinsel diskurumuzu sözel diskura veya düşünceler zincirimizi sözcükler zincirine çevirmektir; bunun da iki yararı vardır. Birisi (...) düşüncelerimizin kaydedilmesidir. Böylece, adların ilk yararı *işaretler* veya hatırlama *notları* olarak hizmet etmeleridir. Diğerisi ise çok sayıda insanın, (...) duygularını birbirlerine aktarmak için aralarındaki bağlantı ve düzenden dolayı aynı sözcükleri kullanmalarınıdır. İşte bu yarar nedeniyle, onlara *işaretler* denilir (Hobbes, 2007: 26-27).

⁸Hobbes'un *Leviathan* isimli eserinin Türkçe çevirisinin doğrudan alıntı yapmak istediğimiz bölümü tezimize uygun bir yapı sergilemediğinden ötürü, doğrudan alıntımızı çevirisini yapmak suretiyle İngilizce kaynaktan almayı tercih ettik.

Tezimiz bağlamında özellikle bu bölümde, Hobbes'un Leibniz'e evrensel dil arayışında esin kaynağı olmuş "kör düşünce" fikrine değinmek yerinde olacaktır. "Kör düşünceden, anlamı zorunlu olarak bilinmeyen ya da anlamları hakkında açık ve belirli bir fikir elde edilemeyen simgeler üzerinde hesaplamalar yürütüp, kesin sonuçlara ulaşma olanağı kastedilir" (Eco, 1995a: 277). Leibniz'in "kör düşünce" ile ilgili şahsi görüşü şu şekildedir:

Ben onlara *cogitationes caecae* diyorum. Demek istediğim, ele alınan geometrik şekillere yalnızca aralıklı dikkat göstererek cebirsel olarak hesap yapıldığında olduğu gibi algı ve duyarlılıktan yoksunlar ve sembollerin tamamen yardımsız kullanımından oluşuyorlar. Sözcükler, bu bakımdan aritmetik ve cebir simgeleriyle genellikle aynı şeyi yapar. Çoğu zaman nesnenin kendisi aklımızda neredeyse yokken, kelimelerle akıl yürütürüz (Leibniz, 1996: II. XXI. §35. 160).

Katı bir nominalist olan Hobbes, insanların düşüncelerini aktarırken aynı adları kullandıkları ve düşünürken bu adların anlam içeriklerine odaklanmadan adların aralarındaki bağlantılar üzerinden, yani işaretler üzerinden çıkarımlarını ifade ettiklerini savunurken, bu sav Leibniz'de *characteristica universalis* için bir yöntemsel fikir olarak *sembol manipülasyonuna* kadar uzanmıştır. İleride göreceğimiz üzere bu şekilde nitelikler, kavram ve nosyonlar bir sembol manipülasyonu vasıtasıyla niceliksel bir forma kavuşacaktır.

1.3.3. René Descartes (1596-1650)

Niyeti, kuşkudan arındırılmış epistemeye ulaşmak olan Descartes, matematiğin (o dönem *matematikler* olarak geçmektedir) yöntemini kullanmıştır. Yaşadığı dönemde tıpkı Bacon gibi Descartes da yöntem arayışı içinde olan bir filozoftur. Galilei'nin matematik ve fiziği birleştirerek serimlediği mekanik dünyadan oldukça etkilenen Descartes, filozof olmanın yanında ayrıca iyi bir matematikçi olduğundan, geometriyi cebir ile evlendirerek analitik geometrinin temellerini atmıştır. "Descartes seleflerine atıfta bulunmaktan dikkatle kaçındığı için Hârizmî'nin fikirlerinden etkilenip etkilenmediğini bilmiyoruz" (Kline, 1972a: 211) fakat yine de analitik geometrinin yöntemiyle hareket ettiği felsefesine yaptığı yöntemsel kuşkuculuk eklentisi sayesinde, modern felsefenin kurucusu sayılmaktadır. Descartes'ın kendisiyle özdeşleşmiş olan önemli kavramlarından birini, epistemelerini özneliliğin üzerinden yürütmeyi sağlayacak *cogitoyu*, ünlü *cogito ergo sum* (düşünüyorum, öyleyse varım) ifadesine ulaşacağı şekilde işleyişe soktuğunu bilmekteyiz. Descartes, felsefesini *benin* varoluşuna uyguladığı şüphecilik sayesinde geliştirdiği yöntem üzerinden açık ve seçik bilgiye ulaşacağı şekilde ikame etmiştir. Descartes'ın *Yöntem Üzerine Konuşma* adlı eserinin

II. bölümünde epistemeye ulaşmak için uyguladığı yöntemini aşamalarıyla anlattığı üzere bu yöntemde, kendisinden haberdar *özne* hareket noktası olarak kabul edilmektedir.

Birincisi, doğruluğunu apaçık bilmediğim bir şeyi doğru diye almamak, (...) İkincisi, inceleyeceğim güçlüklerden her birini olabildiğince parçalara ayırmak ve onları en iyi çözümlenebilecek duruma getirmek. (...) Üçüncüsü, düşüncelerimi en basit ve tanınması en kolay olan nesnelere başlayarak ve yavaş yavaş, derece derece ilerleyerek en karmaşık bilgilere kadar götürmek ve doğal olarak birbiri ardından gelmeyen şeyler arasında da bir düzen varsaymak. (...) Sonuncusu, her yerde bütünsel saymalar ve en genel gözden geçirmeler yaparak hiçbir şeyi dışta bırakmadığımdan güvenli olmak (Descartes, 1998: II. 46).

Açık seçik bilgiye ulaşmanın bir yöntem sorunu olduğu kadar bir kavram sorunu olduğu da yukarıdaki alıntıdan görüleceği üzere şeyler, nesnelere bağlamında kendini ele vermektedir. Yüzyılın bir karakteristiği haline gelmiş olan bu yöntem sorunu ve ayrıca kavramların ya da *adların neliği* problemi de bilginin doğruluğuna dair sorunsalın temelinde yatıyor görünmektedir.

Yöntem sorununu felsefesinin temeline oturtan Descartes, dönemin modern düşüncesi üzerinde ortak etkileri olduğu bir diğer filozof Bacon gibi doğal dilin müphemliğinden dem vurmaktadır. Descartes'a göre dil yetisi, ne olursa olsun insana bahşedilmiş bir yetenek olup, en aptalın bile bir şekilde meramını anlatabildiği bir işaretler bütünü olarak *dil*, iletişim aracıdır. İnsan için her tür işaretler bütünü olarak *dil* büyük bir yardım kaynağıdır ama yine de doğal dillerden kaynaklı kusurlar vardır. Descartes'a göre doğal dil konuşanlarca kullanılan kelimeler veya kavramlar, anlamlarından ziyade-ister doğru ister yanlış kullanılıyor olsun-hafızaya alındığı şekliyle kullanılmaktadır. Bu durum, iletişim için kullanılan dilin hafızaya alındığı şekliyle yanlış kullanımlarının, düşüncelerimizde muhakememizi olumsuz anlamda etkilediği sonucunu ortaya çıkarmaktadır.

Epistemolojik kaygılar üzerinden yöntem arayışında bulunan Descartes, dönemin düşünürlerinden farklı olarak doğal dillerin karışıklık içeren görünümünden uzaklaştırılabileceği herhangi bir yöntem önermemiştir. Kendisinin böyle bir girişimi olmamışsa da "Descartes gündelik dilin kifayetsizliği karşısında, bir 'kusursuz/ideal dil' icat etme fikrini yabana atmamıştır" (Altınörs, 2020: 44).

1.3.4. Port-Royal Mantığı (1660)

Descartes'ın etkisi altında kartezyen dil kuramını geliştiren Antoine Arnauld (1612-1694), Claude Lancelot (1615-1695) ve Pierre Nicole (1625-1695), Platon'un idealar kuramını takip eden ve Descartes'ın izinde yürüyen kartezyenciler olarak, düşüncenin dilden

önce (Buroker, 2003: xxvi) geldiğini dikkate alarak hareket etmektedirler. Port-Royal bilgi teorisinin tamamı Descartes'tan alınmadır (Buroker, 2003: xx). Port-Royalciler Descartes'ın yöntemine ve rasyonalizmine göre hareket edip, matematik ve özellikle geometrinin yöntemine uygun paradigmlar geliştirmişlerdir.

Mantık veya Düşünme Sanatı [Logic or the Art of Thinking (1662)] adlı yapıtlarında, Arnauld ve Nicole'e göre, insanlar zihinde canlandırma ile gerçek bir akıl yürütmeyi çoğunlukla birbiriyle karıştırarak (2003: 26) bu ayrımı gözden kaçırmaktadırlar. Öyle ise şeylerin ideası olmadıkça ne söylendiğinin bir anlamı olamaz çünkü kelimelere döktüğümüz şeylerin açık ve seçik bilgisini ancak idealardan elde ederiz (2003: 26). Böyle bakıldıkta, idealar şeylerin dolaysız temsili olurken, doğru bir akıl yürütme sayesinde şeylerin ideasının dildeki temsili yalnızca kelimelerdir. Bu durumda hakikat kelimelerde değil, ideaların kendi arasındaki bağıntılardadır (Altınörs, 2019: 125). Bu noktada, kelimelerin anlamları da bulanıklığı azaltma bağlamında değer kazanmaktadır.

Port-Royalci Arnauld ve Nicole, doğal dillerde karşılaşılan kelimelerin yaptığı zihinsel bulanıklıktan kaçınmak için yeni kelimeler ve dolayısıyla bu kelimelerden oluşan yeni bir dil (2003: 60) önermektedirler. Amaçları mantıkla kategorize edilmiş (Aristoteles'in mantığından ayrı) bir dil geliştirmektir. Bu yeni dilde, "örneğin, töz ve nitelik kategorilerine, sırasıyla, isim ve sıfat gramer kategorileri karşılık" (Altuğ, 2001: 173) gelmektedir. Teklif edilen bu yeni dil, evrensel dilbilgisinin mantıksal temelleri üzerinden yükselen yapıya sahiptir.

Arnauld ve Lancelot tarafından kaleme alınan *Port-Royal Mantığı* (General and Rational Grammar: The Port-Royal Grammar [1660]) isimli yapıtın günümüze uyarlanmış edisyonunun önsözünde şöyle yazmaktadır: "Böylece, Port-Royal Mantığı için dilin takriben iletişim aracı olduğu, paralel olan iç süreçlerin (onları hesaba katmaktan çok) dışa dönük ifadeleri olarak anlaşıldığı duygusu elde edilir" (Danto, 1975: 16). Buradan hareketle denilebilir ki içeride mantıksal çerçeveye sahip evrensel bir dilbilgisinin temelleri üzerinde yükselen dışsal bir kelime-anlam iletim inşası söz konusudur.

Port-Royalciler, Descartes'ten ve kartezyen görüşten etkilendikleri kadar, tüm dillerde yaygın kullanımı olan bazı genel ilkelerin varlığına (Arnauld ve Lancelot, 1975: 171) dair görüşleriyle, günümüzde evrensel dilbilgisinin derin yapılar üzerinden tüm insanlıkta ortak olduğunu iddia eden Noam Chomsky (1928-...) olmak üzere, Yeniçağ'da ideal dil arayışında olanları da oldukça etkilemişlerdir. Locke için böyle bir ideal dil hayali yoksa bile o da tıpkı Leibniz gibi Port-Royalcilerin etkisi altında kalmıştır.

1.3.5. John Wilkins (1614 – 1672) ve George Dalgarno (1616 – 1687)

Yeniçağ'da dilin, paradigma değişimlerinden etkilenecek artık kök dil (günümüzde konuşulabilir olan pek çok dilin atası) veya Âdemi dil (bütün dillerin kendisinden çıktığına inanılan tam kapsayıcı dil) arayışı gibi konularla anılmaktan ziyade, felsefi içeriklere sahip yapay diller üzerinden anıldığını daha önce belirtmiştik. Bu nedenle, bu yüzyılda yaşamış filozoflar ve dilbilimciler için bilime ve olgulara yanıt verebilecek felsefi diller, doğal dillerdeki belirsizliği ortadan kaldıracak sistemlermiş gibi görünmüştür.

Dalgarno ve Wilkins yaşadıkları on yedinci yüzyılda, doğal diller her geçen gün daha fazla yozlaştıklarından ve dolayısıyla başlangıç mükemmelliklerini koruyamadıklarından dolayı, hayran oldukları İbranice veya Latince üzerinden bile olsa bir doğal dil üzerinden Âdemi dile veya bir dilin orjinine/orjinaline ulaşamayacakları gerçeğiyle yüzleşen ve evrensel özellik barındıran *a priori felsefi* dil arayışlarında başka yollar deneyen dönemin iki önemli dilbilimcisidirler.

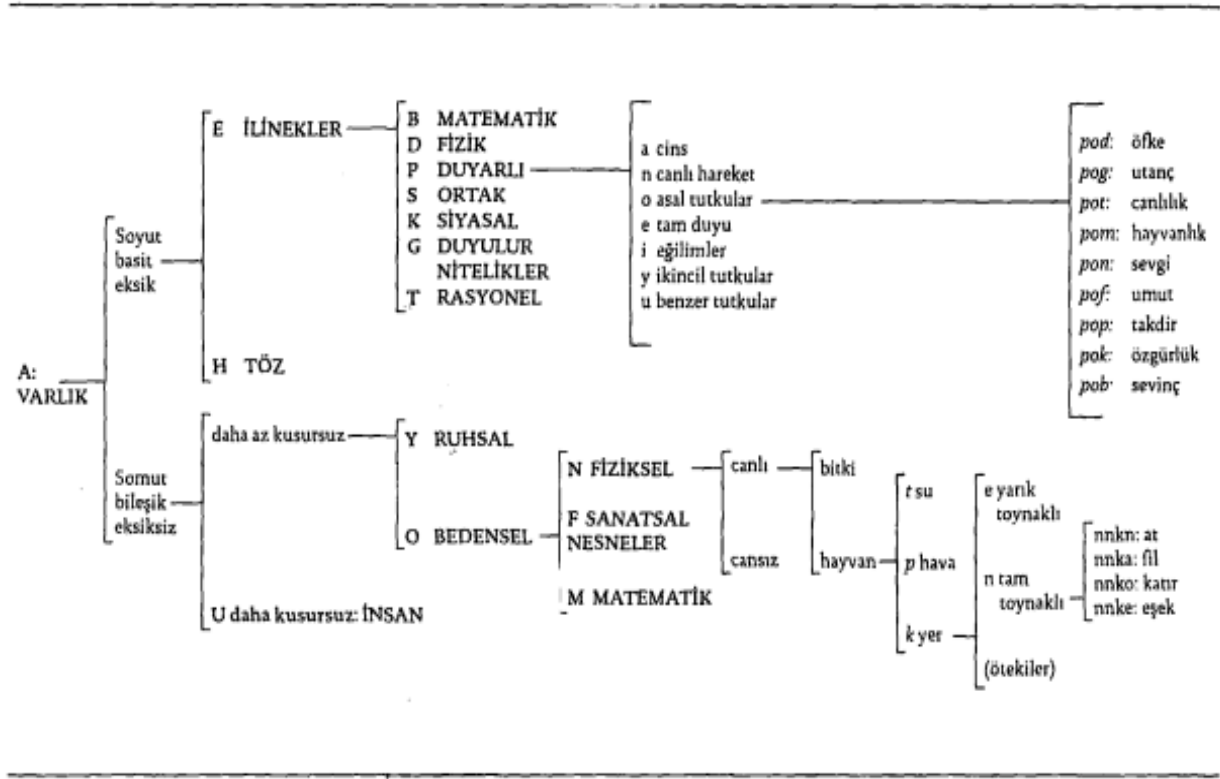
Dalgarno ve Wilkins, dilbilgisi ve mantığı, yalnızca göstergenin gösterilenden farkı nispetinde, evrensel dilbilgisinin tamamlayıcı yönleri olması bakımından bir bütün olarak görürler (Scaglione, 1990: 346) ve bu nedenle “önceliği mantığa verme konusunda hemfikir olmaları bakımından evrensel ya da felsefi dilbilgisinin etkisi altına girerler” (Scaglione, 1990: 346).

Dalgarno, Wilkins kadar iyi bir çevreye sahip olmayan bir öğretmendir fakat Wilkins, Royal Society ile yakın ilişkileri olan bir Oxfordludur. Bu yüzden Dalgarno'dan çok Wilkins'in *a priori felsefi* yapay dil çalışmaları bilinir. Dalgarno, Leibniz'in onun dil sistemine ve bu sistemin varlık düzlemine verdiği önem neticesinde adı geçen biridir.

Teorik olarak bu sistem her kavramı, her biri karşılık gelen kelimeler için bir başlangıç görevi gören bir harfle gösterilen on yedi temel sınıf veya kategoriden birine dağıtmaktan oluşuyordu. Her sınıf daha sonra aynı ilk harfle belirtilen alt sınıflara bölündü ve ikinci bir harfin eklenmesiyle birbirinden ayırt edildi. Son olarak, her alt sınıfta aynı harflerle karakterize edilen ve bir son harfteki varyasyonlarla ayırt edilen birkaç kelime toplandı. Kısacası bu, sistematik olarak rastgele bir sırayla oluşturulmuş ve geleneksel indeksler veya etiketler rolünü oynayan kelimelerle ifade edilen kavramların mantıksal bir sınıflandırmasıydı (Couturat, 2012: 5).

Dalgarno'nun *a priori felsefi* dilinde keyfi belirlenmiş kelimeler, “VARLIK”tan başlamak üzere on yedi temel cins ayrılır ve bu temel cinsler, etiketlendikleri büyük harflerin küçükleriyle ara cins ve türlere ayrılarak (Eco, 1995a: 232) bir nevi töz ve ilinek ilişkisini sunan bir varlık düzlemi oluşturur. Bunun yanı sıra keyfi olarak ortaya çıkan yeni gösterimlere sahip kelimeler arasında zıtlık ve benzerlik türünden ilişkiler de kurulmuş olur.

Aşağıdaki şekilde Dalgarno'ya ait varlık düzleminin yalnızca kısıtlı bir bölümü görülmektedir.



Şekil 1.1 Dalgarno'nun varlık düzleminin kısıtlı bir örneği

Kaynak: Eco, 1995a: 233

Dalgarno, her ne kadar evrensel bir dil arayışında olmuşsa da ulaşma gayretinde olduğu bu dil, *a priori felsefi* görünümde olduğundan dolayı bu dilin tek başına oluşturulamayacağını bilmekteydi. “Felsefi diller, genellikle evrendeki şeylerin, organizmaların ve diğer kategorilerin karmaşık bir sınıflandırmasını içeriyordu ve benzer anlamlara sahip kelimeler benzer biçimlere sahipti” (Libert, 2018: 4). Bu nedenle, kavramlar hiyerarşisi oluşturmak sanılanın aksine oldukça zor bir iştir. Dalgarno'ya göre *a priori felsefi* bir yapay dilin kurulumunda, filozofun çalışması, dilcinin çalışmasından önce gelmelidir (Eco, 1995b: 222)⁹ çünkü *a priori felsefi* bir dilde öncelikle kavramlarda ilksellere (varsayımsal olarak) varabilmiş olmak gerekir ki daha sonra içerik düzenlemesi icap eder.

⁹ Umberto Eco'nun *Avrupa Kültüründe Kusursuz Dil Arayışı* adıyla İtalyancadan Türkçeye çevrilen kitabında “önsel felsefi diller” ve “sonsal diller” olarak yapılan çeviriler, İngilizceye *The Search For The Perfect Language* adıyla çevrilen kitapta “a priori philosophical languages” ve “a posteriori languages” olarak vurgulanmıştır. Yaptığımız İngilizce diğer okumalarda da aynı ifadeler geçtiğinden dolayı biz de bu ifadeleri tarafımızca İngilizceden çevirisi yapılmak suretiyle “a priori felsefi diller” ve “a posteriori diller” olarak tezimize dahil ettik. Bu nedenle, yalnızca bu kavramların kullanılması durumunda Eco'nun İngilizce çevirisinden faydalanacağız.

Dalgarno evrensel bir dilin birbirinden net olarak ayrılmış iki yönü içermesi gerektiğini biliyordu: Filozofun eseri olan bir bilgi sınıflandırması (içerik düzlemi) ile karakterleri bu sınıflandırmanın belirlediği nesne ve kavramlara göndermede bulunacak şekilde düzenleyen bir dilbilgisi (anlatım düzlemi). Dalgarno bir dilci olduğundan, başkalarının tamamlamasını dileyerek, yalnızca sınıflandırmanın ilkelerine değinir (Eco, 1995a: 231).

Wilkins için de aynı düşünce geçerlidir. Wilkins'e göre, eğer bir evrensel dil olacaksa bu dil felsefi ilkelere göre inşa edilmiş yapay bir dil olmalıdır (Subbiondo, 1990: 357). İki dilbilimcinin dönem dönem bu yapay dilin nasıl olması gerektiğine dair fikir telakkisinde buldukları bilinmektedir ki çalışmaları da yakın tarihlerde yayımlanmıştır. İkisinin de sisteminde oldukça hacimli bir varlıklar düzlemi mevcuttur. Çoklukla töz ve ilineklere oluşan varlık düzlemleri, onlar için tözler ve ilineklere arasında hiyerarşi sorunu çıkarmış gibi görünmektedir. Bu yüzden her ne kadar özellikle Wilkins'in sistemi, on yedinci yüzyılın Locke'tan *Encyclopédie*'ye en güvenilir projesi olarak anılsa da (Eco, 1995a: 231) yine de hayata geçirilememiş bir proje olarak kalmıştır.

1.3.6. John Locke (1632-1704)

Doktor olmasının yanı sıra dönemin modern düşünürlerinin etkisinin göze çarptığı John Locke, deney ve gözleme dayandığı deneyci felsefesiyle bilinir. Locke, deneyci felsefeye yaptığı katkılardan dolayı öncü bir filozof sayıldığı yaşadığı dönem Avrupa'sının aydınlanmasının büyük düşünürüdür. Bacon'un açtığı bilimsel düşünme yolundan giden Locke, tıpkı selefi gibi dilin yanlış kullanımlarının önüne geçebilmek için sözcükler ve anlamları üzerinden, zihin-dış dünya-bilgi arasındaki bağı, *idelerle*; *ideler* arasındaki bağı bilgi aktarımında kullanılmasını sağlamak için de işaretler olarak gördüğü kelimelerle, ifade etmeye çalışmıştır. Tıpkı Hobbes gibi İngiltere'de yaşanan siyasi çalkantılardan ötürü Fransa'ya iltica etmiş, bir dönem Hollanda'da kalmıştır.

Kendisine ait birçok eserini başka isimlerle yayımlamış olsa da Fransa ve Hollanda yıllarında tamamladığı kendi ismi ve imzasını taşıyan baş yapıtı *İnsanın Anlama Yetisi Üzerine Bir Deneme* (1690) isimli eserinde, Locke, dört ayrı kitap halinde düşüncelerini sunmuştur. Bu eserde, birinci kitap "doğuştan ideler"; ikinci kitap "yalın ve karmaşık ideler"; üçüncü kitap "sözcükler" (kelimeler); dördüncü kitap ise "bilgi" üzerinedir. *Deneme*'nin özellikle birinci kitabında denk geldiğimiz *ide* kavramı, kullanımındaki sıklıktan dolayı özür dilemek zorunda kalan Locke için uyanıkken açık algılarımızla dışarıdan aldığımız tüm deneyimlerimiz hakkında zihnimizde oluşan her şeydir.

Fakat öncelikle, okuyucumdan, deneme boyunca göreceği üzere, “ide” sözcüğünü oldukça sık kullandığım için özür dilemeliyim. Bana göre bu terim, bir insan düşünürken anlama yetisinin nesnesi haline gelen şeyleri simgeliyor; bu yüzden de *düşünme ediminde zihnin kullanılabildiği imge, kavram, tür ya da başka herhangi bir şey ile demek istenenin* dile getirilmesinde “ide”yi kullandım” (Locke, 2000: 38).

Locke, *Deneme*’nin tüm idelerimizin kaynağını dış duyum nesnelere aldığımızı dair savlarını sıraladığı birinci kitabında, özetle doğuştan ideleri reddetmekte, aklın aktif kullanımının ancak ve ancak dış duyum nesnelere zihnimizdeki refleksiyonuna bağlı olduğunu; oysa doğuştan ide kabulünün neticesinde kanıt gereksinim duymadan, bilginin doğuştan geldiğine inanıldığı şekliyle olduğu gibi kabullenilmesinin ancak tembelliğe sevk edilmiş zihinler doğuracağını iddia etmektedir. *Deneme*’nin ikinci kitabında, özetle tüm karmaşık idelerimizin yalın idelerin bir çeşit kombinasyonu olduğu, bu yolla zihinde sonsuz sayıda karmaşık ideye sahip olunabileceği, karmaşık idelerin ilineksel ayrımları sonucunda ise soyutlama denen şeyin olduğu vurgulanmaktadır.

Zihnin yalın ideleri üzerinde kendi gücünü harekete geçirdiği edimlerinin başlıcaları şunlardır; (1) Birkaç yalın ideyi tek bir *bileşik idede* birleştirmesi ki sonuçta tüm bileşik ideler bu şekilde yaratılır. (2) Yalın ya da bileşik, iki ideyi bir araya getirmesi ve tek bir ideye dönüştürmeden ikisine aynı anda ulaşabilmek için başka bir ide yoluyla onları saklaması: Tüm bağıntı ideleri bu yolla elde edilir. (3) Gerçek varlıklarında onlara eşlik eden tüm idelerden ayırması: Bu soyutlama diye adlandırılan, genel idelerin oluşturulmasına yarayan bir işlemdir (Locke, 2000: II. XII. 1. 220-1).

Locke *Deneme*’in ikinci kitabında ünlü *tabula rasa* fikrinden hareketle, zihnin doğuştan boş bir levha (sayfa) olduğunu; bilgiyi edinme şeklimizin tamamen deneyimden kaynaklandığını ifade ederek (2000: II. I. 2. 133-4), tüm bilgimizi dış duyum aracılığıyla doğrudan edindiğimiz yalın ideler ve zihnin refleksiyonu neticesinde, onların birleşiminden dolaylı olarak ürettiğimiz karmaşık idelerimizden elde ettiğimizi savunmaktadır. *Deneme*’in üçüncü kitabında değindiği şekliyle bir sözcük atomcusu olduğu görülen Locke, özetle yalın idelerin dolaysız olmalarından sebep, herkesçe açık olduğunu ve fakat yalın idelerin birleşiminden elde edilen karmaşık idelerin içlerinde barındırdıkları başka ideler nispetinde, anlam bakımından bulanıklaştığını (1999: III. 9-10) vurgulamaktadır.

Deneme’in dördüncü kitabının referansı ile, tüm bilgi aktarımını kelimeler üzerinden yaptığımız (1999: IV. XXI. 4. 482) düşünülürse, “Locke’un yaklaşımı açısından bir kelimenin

yerini tuttuğu ide¹⁰ ne kadar karmaşık ise yani bir arada bağladığı ide sayısı ne kadar fazla ise kelimenin anlamı o ölçüde müphemleşir” (Altınörs, 2019: 130). Buradan hareketle, Locke’a göre dış duyum nesnelere elde edilen ideler arasında “bileşimi yapan zihin olsa da onları sınımsız yapan düğüm, addır” (1999: III. V. 10. 57). Buradan doğal olarak şu sonuca ulaşabiliriz: Yalın idelerden karmaşık ideler koleksiyonunu yapan zihin olduğuna göre ve zihin de doğuştan idelerle (Locke’un reddiyesidir) değil, tamamen deneyim ve tecrübeden edindiği ideler üzerinden karmaşık ideler ürettiğine göre her zihin, kendi öz deneyimlerinin mefhumlarına sahiptir çünkü deneyim bireyseldir.

Bilginin zihinde ideler aracılığıyla kaydedilmesinin nedenini iletişime; iletişimin çimentosu kelimelere (1999: III. IX. 1. 112) bağlayan Locke’a göre bugün edindiğimiz karmaşık ide bilgimiz, zaman içinde o ideye ait yeni bilgiler neticesinde değişebilir (1999: III. IX. 6. 115) olduğu kadar, ayrıca iletişim halinde olduğumuz başka bir zihinde oluşan karmaşık ide de bizim zihnimize oluşturduğumuz karmaşık ideden karma mod ve bağıntı idelerinden dolayı farklı olabilir (2000: II. XXXI. 5. 503).

Adlarını aynı telaffuz ettiğimiz kelimelerin işaret ettiği karmaşık idelerimizin bağıntı ideleri nispetinde zihinlerimiz ölçütünde ayrılmış yapısı, iletişimde kullandığımız kelimelerin müphem görünümüne sebep olduğu kadar, bu belirsizlik hali tüm doğal dillerin kusuru olarak görülmektedir. “Locke tüm yalın idelerimizin, şeylerin gerçekliğiyle uyumu bakımından müsavi olduğu kanaatine paralel olarak, yalın idelere verilmiş adların anlamının hiçbir surette keyfi olmadığını savunur” (Altınörs, 2016: 41). Bu durum, karmaşık idelerin anlam içeriklerinde zihinsel açıdan eksik-fazla bigi durumlarımıza rağmen, iletişim halindeyken üzerinde mutabık kaldığımız açık seçik idelerin işaretleri sayılan kelimelere bağımlı kalmamızın nedenini ortaya koymaktadır. Böyle bakıldıkta, Locke için yaşadığı dönemin çehresinde bir “mükemmel dil”, “evrensel dil” ya da “kök dil” arayışı anlamsız görünmektedir ki Locke, bir insanın dünya üzerindeki başka milletlerin dillerini mükemmel forma kavuşturma çabasını iddialı ve gülünç bulur.

Bir insanın kendi ülkesinde konuşulan dil düzeyinde olmayan dünya dillerini tamamen düzeltme iddiasında bulunup da kendini gülünç duruma düşürmeyeceğini sanacak kadar aptal değilim. İnsanların

¹⁰ *İdea* kavramı batı felsefe tarihinde Platon’dan bu yana filozofların farklı anlam içeriklerine tabii kıldığı bir kavram olarak yaygın kullanıma sahiptir. Ancak Locke, Platon’dan farklı olarak tercihen *ide* kavramını kullanmaktadır. Tezimizde doğrudan alıntılarını yaptığımız Atakan Altınörs’ün Locke bağlamında yazılarında ise *ide* kavramı yerine *idea* kavramını kullandığını görmekteyiz. Locke’un kendi yazılarına sadık kalmak adına biz de Atakan Altınörs’den yaptığımız doğrudan alıntılarda *idea* olarak yazılan kavram yerine *ide* kavramını kullanacağız.

kendi sözcüklerini, hep aynı anlamda ve yalnızca belirtik ve tek tip ideler için kullanmalarını beklemek, bütün insanların aynı kavramlara sahip olduklarını ve yalnızca açık ve seçik idelerini taşıdıkları şeylerden söz ettiklerini düşünmek demektir ki insanları çok konuşmaya ya da susmaya ikna edebileceğini sanacak kadar budala olmayan hiçbir insan böyle bir beklenti taşımaz (Locke, 1999: III. XI. 2. 157).

Locke nezdinde, dilin doğru kullanımı için öncelikle bireysel açıdan değerlendirmek gerekirse zihinlerin bilgiye ulaştığı ve kaydettiği şekliyle, ideler konusunda algılarımızı deneyimlerimize doğru kanalize etmemiz, açık seçik temellerden hareket etmemiz gerektiği söylenebilir.

Locke iletişim bozukluklarına neden olan kelimelerin anlam belirsizliklerine karşı, Yeniçağ'da filozoflar arasında yaygın olan bir "ideal dil" icat etmemin gerekliliği fikrine karşı mesafeli bir duruş sergiler. (...) Âdem ya da bir başka insanın kelimelerle, sadece – şeylerin işaretleri olarak – zihindeki ideleri, temsilleri veya kavramları adlandırdığı söylenebilir. Bu bakımdan, Locke için artık "eşyanın tabiatını yansıtan mükemmel dil" diye bir şey söz konusu olamaz (Altınörs, 2016: 49-51).

Locke'un *Deneme*'sinin, İngilizce yayımlanmasının ardından Fransızcaya çevrilerek, İngilizce okuma yaptıktan sonra Fransızca çeviriyi bekleyen Leibniz dahil, Avrupa topraklarında geniş bir okuyucu kitlesine ulaşması üzerine (Antognazza, 2013: 342), Leibniz'in Locke ile gerçek bir iletişime geçme talebi (Locke'un Leibniz'i dar görüşlü bulması, nedenlerden biri olsa da başka bir neden Leibniz'in Newton'dan dolayı İngiliz dünyasında sevilmemesi de olabilir) bu iletişim talebi mektuplaşma yoluyla gerçekleşecek bile olsa Locke tarafından her defasında reddedilmiştir¹¹. Bu sebeptendir ki Leibniz, rasyonalist çevreleri Locke'un *Deneme*'sine karşı korumak için mektuplaşma yoluyla cevap alamadığı fikirlerini, Platon'un yazma şeklinden esinlenerek karşılıklı Locke Leibniz diyaloguna çevirmek suretiyle, *Yeni Denemeler* (1704) adlı yapıtını kaleme almıştır.

¹¹ 1695'ten başlayarak, Locke'un 1704'te ölümüne kadar Leibniz, Locke ile doğrudan bir iletişim kanalı açamamış ya da mektuplaşma imkanı bulamamıştır. Leibniz, ancak ve ancak Thomas Burnett ve Leydi Domaris Masham aracılığıyla mektuplarını ve *Deneme* ile ilgili eleştirilerini iletebilmiş olsa da Locke, Leibniz'in eleştirileri hakkında fikrini iletmekten çekinerek sadece şahsına ait son yazdığı kitapçıkları göndermekle yetinmiştir (Antognazza, 2013: 338-9). Leibniz hakkında gerçek görüşlerini belirttiği 1697 yılında William Molyneux'a yazdığı mektubun, Locke'un ölümünün ardından 1708 yılında yayımlanması üzerine Leibniz, Locke'un kendisiyle ilgili gerçek görüşünden haberdar olmuştur. Bu mektupta, Locke, Leibniz'in yazdıklarına dair eleştirileri üzerine "(...) ve şu kaniya vardım ki büyük insanlar bile büyük düşünmüyorlarsa, hiçbir konuya vakıf olamazlar ve geniş zihinlerin de dar fikirleri olabilir" (Locke'tan aktaran Antognazza, 2013: 340) şeklinde kendini ifade etmektedir. Locke'un kendisi hakkındaki gerçek görüşlerine çok geç vakıf olan Leibniz 1714'te Nicolas Rémond'a yazdığı bir mektupta hakkında duyduğu ifadelerle ilgili şu karşılığı vermiştir: "Biz prensipte biraz farklı insanlardık ve benim önerdiğim şey ona çelişkili gelirdi. (...) Bay Locke ince yanları ve yeteneği olan biriydi. Ayrıca nasıl cazibeli kılacağını bildiği yüzeysel bir metafiziğe sahipti ama matematikçilerin yönteminden habersizdi" (1969k: 656; Leibniz'den aktaran Antognazza, 2013: 340) cümlelerine başvurmuştur.

Leibniz'in Locke'a eleştirisini içeren *Yeni Denemeler*'ini bazı kesitleriyle evrensel dil bağlamında tezimizin 1.4.5. alt başlığında aktarmaya çalışacağız.

1.4. Gottfried Wilhelm Leibniz (1646-1716)

Leibniz'in doğduğu 1646 yılından sadece iki yıl sonra mezhepler çatışmasından kaynaklanan Otuz Yıl Savaşları sonlanırken, arkasında günümüz Almanya'sının parçalanmış topraklarında çareyi kendi içlerine kapanmak ve inançlarının gerektirdiği ama savaşla daha da katılmış öğretilerine sığınmakta bulan insanlar bırakmıştır. Leibniz, bu savaştan en çok yara almış ülkelerden biri sayılan bugünün Almanya'sının Leipzig kentinde (yaşadığı dönemde Hannover Hanedanlığına bağlıydı) doğmuştur. Lutherci protestanlık hakimdir ve savaş travmasıyla içe göçmüş, cemiyet ve bilim dünyası da dahil olmak üzere eğitim kurumları katı otoriter ve dogmatik sayılabilecek bir yaşam tarzına ve eğitim sistemine sahiptir.

Eğitimini katı sayılabilecek yeni-Platonculuktan etkilenmiş skolastik bir sistem içerisinde Aristoteles'in töz kavramları üzerine alan Leibniz, henüz altı yaşında iken kaybettiği Moral profesörü (üniversitede ahlak dersleri veren bir Hukukçu) babasının kütüphanesinden sekiz yaşından itibaren özgürce faydalanmanın tadını çıkarmıştır. Leibniz'in yaşadığı dönemde bir kütüphaneden faydalanmak izne tabii olduğundan "Leibniz'in (...) eşsiz düşünsel yeteneğinden etkilenen bilgili ve çok gezip görmüş bir asilzade komşu, Leibniz'in babasının kütüphanesini kullanması için onun ailesini ikna etmeyi başarmıştır" (Antognazza, 2013: 34). Bu sayede kendi kendine öğrendiği Yunanca ve Latince dillerinde Platon ve Aristoteles okumuş (Timuçin, 1999: 18), zihni oldukça farklı alanlarda çok zor sayılabilecek bilgilere erken yaşlarda aşına olmuş; sorgulamayı, mantıksal sistemler kurmayı da yine erken yaşlarda bir düşünce normali olarak edinmiştir.

Yaptığı özgür okumaların yanında eğitimini aldığı katı sistem üzerine ciddi sorgulamalar yapan filozofumuz, özellikle Descartes'ın modern çağı başlatan felsefesinin etkisi altında kalmış, bu nedenle bilimsel yönden dogmatik sayılabilecek eğitim sistemiyle bağlarını esneterek, yaşadığı yüzyılın paradigma değişimlerine de zihinsel sorgulamalarını açık tutmuştur. Düşünsel dünyasına katkıda bulunacağını düşündüğü her filozof, dilbilimci, teolog ve onların ürettikleri sistemlerine odaklanan Leibniz'in, Descartes'a meyletmesinin nedeni, matematik mekanizmdir ki Leibniz, felsefi kariyerine hakikaten de Descartes'ın bir tilmizi/öğrencisi olmakla başlamıştır (Boutroux, 2017: 27).

Leibniz'de diğer filozoflardan farklı olarak disiplinler, ayrılmak üzere bir araya gelmiş olamazlar, aksine hepsi bir bütünün aynı kaynağından çıkan olmazsa olmaz yansımalarıdır. Leibniz, bir disiplinin bir olguyu açıklamada yetersiz kaldığı yerde diğer disiplinin

açıklamaya girişmesinde rasyonalist bir tavır takınır. Örneğin, fiziğin hareketi açıklamadaki başarısını mekanik olarak değerlendirirken Leibniz, Nicolas Rémond'a 1714 tarihli mektubunda “mekaniğin ve hatta hareket yasalarının nihai nedenlerini aradığımda, bunların matematikte bulunmadığını görünce metafiziğe dönmek zorunda kaldım” (1969k: 655) diye yazmıştır.

Böylece Leibniz, hareketin ilk nedeninin metafizik açıdan değerlendirilerek Aristoteles'in “ilk neden” ilkesinden yardım alınması gerektiğini iddia etmiş, geleneksel bilim ile modern çağın doğurduğu bilimi uzlaştırmaya çalışmıştır. Bu yüzden Leibniz, “Bacon'ı ve Descartes'ı dikkate aldığı kadar, Thomas Aquinas'ı da göz önünde bulundurur. Raimundus Lullius'un icat ettiği sanattan bile alınacak dersler olduğunu düşünür” (Boutroux, 2017: 40).

Leibniz'in eski olan ile yeni olanı bağdaştırma becerisini, birbirinden ayırık görünüm sergileyen matematik ve metafiziği birleştirmesinden okuyabilmekteyiz. Bir örnek vermek gerekirse; sonsuz küçükler mefhumunu ilk kez matematiksel işleme tabii tutan ve “ölçülemeyen oranlar ve her türlü büyüklükler için oranlar teorisi” (Kline, 1972a: 177) Eudoxus'a atfedilen (MÖ 408-355) “tüketme yöntemi”, bilindik geometrik şekillerden hareketle, alan veya hacimlere dair net ölçümün alınmadığı durumlarda, hata payları yüksek ölçüm sonuçları elde edilmesini sağlayan bir teoriydi. Metafiziği dışlamayan bir modern filozof olan Leibniz'in Newton'la eş zamanlı keşfini gerçekleştirdiği *kalkülüs* (sonsuz küçükler–diferansiyel hesap) sayesinde doğru ölçümler elde edilmeye başlanmıştı ama bilinmektedir ki Leibniz'in *kalkülüsü*, “yitip giden fark” görüşü sebebiyle metafizik temelliydi.

Tezimize özel değerlendirecek olursak, epistemoloji bağlamında Leibniz'in bakış açısı, dönemin dualist modernitesinde gelişmesine rağmen, bütünleştirici bir monizm etrafında gerçek şeklini kazanmıştır. Böylece ilk bakışta oldukça dağınık, sistematik olmaktan uzak görünüm sergileyen Leibniz felsefesinin gerçekte eklektik bir yapı üzerinde yükseldiğini söylemek mümkündür. Bu yüzden, “Leibniz, skolastik felsefe ile Yeniçağ düşüncesi arasında köprü kuran bir geç dönem Rönesans alimi portresi sergiler” (Boutroux, 2017: 9).

Ortaçağ ve Rönesans etkisi altındaki fikirler ile Modern çağ arasına sıkışıp kalmış Leibniz ve selefleri, felsefelerinde epistemolojik kaygılarla bütüncül bir yol bulmanın derdiyle sistemlerini kurmaya çalışmışlardır. Eski dünyanın karanlığından çıkmalarını sağlamış Rönesansın yansıttığı ışığı kullanarak yeni dünyanın topraklarında yürümektedirler. Reddedişlerle dolu zihinsel sadeleşmelerin ötesinde Leibniz'in belki de en büyük meziyeti, birçoklarınınca belirsiz hatları olduğu dile getirilen hiçbir bilgiyi gereksiz saymaması, ona sistemini kurduran bu tür bilgilerini bir harmoni içinde şekillendirebileceğine dair inancını,

öldüğü zamana kadar diri tutmuş olmasıdır. Öyle ki Leibniz 1714'te Nicolas Rémond'a yazdığı mektubunda, gençlik yıllarında içinde bulunduğu halet-i ruhiyesini, Rosental adıyla anılan bir korulukta tek başına dolaştığı sıralarda, kadimlerin ve skolastiklerin cevherî/tözsel formlar öğretisini kendi düşüncesine nasıl adapte edebileceğini düşünmekle geçirdiğini ifade ederek aktarır:

Aristoteles'i bir delikanlı olduğum dönemlerde keşfettim ve o dönem skolastikler bile beni geri çevirmedi; şimdi bile bunlardan pişman değilim. Ama sonra Platon ve Plotinus da beni biraz tatmin etti, daha sonra danışacağım diğer antik düşünürlerden bahsetmiyorum bile. Önemli okulları bitirdikten sonra, Modernlere rastladım ve on beş yaşında Leibzig'in eteklerinde Rosental denilen bir koruda yürüdüğümü ve tözsel formlar öğretisini koruyup korumamam gerekliliği üzerine tartıştığımı hatırlıyorum. Mekanik görüş sonunda galip geldi ve kendimi matematiğe adamama izin verdi (Leibniz, 1696k: 654-5).

Bu harmonik sistemde, fizikte hareketi boş uzay bağlamında değil, iki cismin birbirine karşı durumu olarak açıklarken; benzer bir yaklaşımı hukukta “adalet” kavramı üzerinden bireylerarası adalet olarak sergilemiş ve monist tutumunu “birey” bağlamında bir nominalist gibi sürdürmüştür. Elbette, Leibniz gerçek bir nominalist değildir. Leibniz, Ockhamlı William'ı takdir eden bir üslup ile nominalizmini şöyle ifade etmektedir: “*Isı* adını kullandığımda belirsiz bir konudan bahsetmem gerekmiyor, daha çok bir şeyin *sıcak* olduğunu söylemem gerekiyor ve bu ölçüde en azından geçici olarak nominalistim” (Leibniz'den aktaran Mates, 1986: 182). Böyle bakıldıkta, Leibniz'in katı bir felsefî sistemi olmadığı gibi komplike bir çokluklar kümesiyle oluşturduğu kendi sisteminden birliği, aşağıda değinmeye çalışacağımız bilgi durakları neticesinde elde ettiği iddia edilebilir.

1.4.1. İlk Gençlik Yılları

Felsefe ve hukukta aldığı eğitimleri yirmi gibi oldukça küçük sayılabilecek bir yaşta, hukuk üzerine doktora derecesini de almak suretiyle tamamlayan Leibniz, kendi çağının öncesinde Orta Avrupa'da yaşamış bütünleştirici ansiklopedistlerden veya diğer bir isimle yarı-Ramistlerden oldukça etkilenmiştir.

Felsefe bölümüne kabulü için hazırladığı, yarı-Ramist bir geleneğin ışığında Aristoteles mantığının yardımıyla, pedagojik bir ansiklopedist gibi yazdığı *Dissertation on the Art of Combinations* [Kombinasyon Sanatı Üzerine Tez (1666), bundan sonra kısaca *Tez* olarak adlandıracağız] isimli çalışmasında, Leibniz'in bütün bir hayatının büyük planı ana hatlarıyla şekil kazanmaya başlamıştır. Bu çalışmada Leibniz'in bilgi kuramının başlangıç düzeyini de yakalamak mümkündür. Yarı-Ramist geleneğin izinde bilgi kuramının ve düşünce

sisteminin çekirdeğini bulabildiğimiz *Tez*'inde Leibniz, sergilediği kuramsal yapıların pratik düzleme uygulanmasını konu edindiği şekliyle pedagojik bir tarzda hareket etmiştir.¹² Elbette Leibniz bağlamında yaklaştığımızda “Ramizmin asıl değeri teknik mantıksal içeriğinde değil, düzenlenmiş bir bilgiyi sistematik biçimde aracı olarak sağladığı pedagojik verimlilik ve yararlılıkta yatmaktadır” (Antognazza, 2013: 41).

Mantığın değeri üzerine Gabriel Wagner'e yazdığı mektubunda Leibniz, bir kişinin, imgesini görüp beyninde ulaşamadığı unutulmuş bir şeyi, kategoriler aracılığıyla tahmin etme ya da hatırlama yönteminin eğlenceli bir keşfini yaptığını dile getirirken, bu noktada Ramist ve yarı-Ramistlerin hakkını teslim etmek gerektiğini ifade etmekte ([1696] 1969i: 464) ve burada yarı-Ramist geleneğin kendi düşünce sistemine etkisinden söz etmektedir.

Leibniz'in *Tez*'inde çalışmalarını yönlendiren ve onu kendi zihninde “analiz-sentez”e yönelten, nihayetinde sistemine hem temelden hem de yöntemsel açıdan etki eden yarı-Ramist gelenekten gelen ansiklopedistler olan Johann Heinrich Alsted (1588-1638) ve Johann Heinrich Bisterfeld'ün (1605-1655) en büyük etkisi aslında kendi çalışmalarının da ötesindedir. Bu ansiklopedistler, Leibniz'i, *Tez*'inde açık etkileri görülen ünlü Katalan filozof ve misyoner Ramon Llull'un (1232-1316) öldürülmesine neden olan ünlü *Ars Magna*'sının (*Ars Combinatoria*) birleştirme sanatına kadar götürmüşlerdir.

Hobbes'un akıl yürütmenin basitçe hesaplama olarak görülebileceğine dair fikrini hareket noktası olarak alan Leibniz (Çevikbaş, 2006: 25), Llull'un kombinasyon sanatını incelemiş ve onda ilksel veya yalın kavramlardan birleştirme yoluyla pek çok bileşik kavramın matematiksel olarak çıkarılabileceğini görmüştü. Bundan daha fazlası vardı: Bu sanatla yalnız yargı değil, ayrıca keşif yapmak da mümkündü. Leibniz'in *Tez*'inde kullandığı yöntem, Llull'un kombinasyon sanatından bir esinlenme (Welch, 1990: 2) olmakla birlikte “Llull, birleşik bir hesabın (kalkülüs) biçimsel formülünden yoksundu ve *ars combinatoria*'yı geleneksel mantığa uyarlamaya ilgi duymamıştı” (Antognazza, 2013: 64) fakat Leibniz tam da bu noktadan hareket ederek, Llull'da görülen biçimsel formülasyon eksikliğini nasıl giderebileceğinin yollarını arayacaktı.

Leibniz, “insan düşüncesinin alfabesi” fikrine, Katalan ansiklopedist Ramond Llull'dan hareketle ulaştı. Llull'un Büyük Sanat'ı, düşünülebilen bütün önermelerin oluşturulmasına imkan veren genel bir metodu. Llull, altı serilik bir kategoriler tablosu oluşturdu; o dokuz mutlak sıfat, dokuz bağıntı, dokuz

¹² Leibniz'in çekirdek çalışmalarının çoğunda göze çarpan ortak özellik, bu çalışmaların ansiklopedik, pedagojik ve yarı-Ramist ya da post-Ramist olmasıdır. “Johann Heinrich Alsted (1588-1638), Alman post-Ramist geleneği dönemin en büyük ansiklopedisiyle taçlandırmakla kalmadı, aynı zamanda Ramist diyalektiğin donuk klişelerini Ramon Llull'dan esinlenerek bir birleşik mantıkla değiştirip daha güçlü bir pedagojik araç geliştirmenin peşine” (Antognazza, 2013: 41) düşmüştür. Leibniz de aynı yolu takip etmiştir.

soru, dokuz özne, dokuz erdem ve dokuz kötülüğü birbirinden ayırır. Llull, kategoriler tablosunu oluştururken matematiksel birleştirme yarasını bilmiyordu ama bu birleştirmeyi mekanik bir yolla yapmaya çalıştı (Çevikbaş, 2006: 113).

Llull ve Hobbes'un etkisiyle matematiksel ve mantıksal pekiştirmeler yaptığı *Tez*'inde Leibniz, kavramsal bölünmeleri ve ilksellere ulaşmayı ya da ilksellerden yeniden sentez yoluyla kavramlar oluşturmayı konu edinmiştir. Leibniz üzerine ciddi bir derleme yapmış olan Loemker, Leibniz'in *Tez*'i üzerine şu ifadeleri kullanmıştır:

Bu çalışma, hayatının geri kalanında düşüncelerini meşgul edecek bir evrensel karakteristik ve mantıksal hesaplamanın planının özünü içermektedir. Proje burada, basitten karmaşık kavramlara doğru aritmetik kombinasyon problemi olarak tasarlanmıştır, Leibniz permütasyon ve kombinasyon üzerine temel teoremleri türetmekte ve bunları mantık, hukuk, teoloji ve diğer düşünce alanlarındaki durumların sınıflandırılmasına uygulamaktadır (Loemker, 1969: 73).

Leibniz'in *Tez*'i, üç ana bölümden oluşmaktadır. Bunların ilki Tanrı kanıtlanmasına ayrılmışken; ikinci bölümde mantık, metafizik, fizik ve pratik alana ait kavramların genel durumları üzerine aritmetik tartışma için bir hazırlık göze çarpmakta ve son olarak üçüncü bölümde ise çalışmanın kendisini tanıtan tanımlar ile birlikte birinci ve ikinci bölümlerde adı geçen problemlere önerilen çözümlerle birlikte teorik olanın pratik alan üzerindeki uygulamaları görülmektedir. Bu bölümler dışında bir tanımlar ve bir de problemler bölümü mevcuttur. Leibniz'in iddiası odur ki şu ana kadar çalışılmamış olmasına karşın "adalet" doktrini bile bilimsel olarak çalışılabilir (1969a: 76). *Tez*'in "Tanrı İle!" başlığını taşıyan üçüncü bölümü daha çok metafizik temellerden yükselen bir çalışmayı temsil eder. Bu bölümde Leibniz, varlık ve varlığın durumlarını tartışmaya açar ve varlığın niteliğini, bağlı olduğu bütün ile ilişkisi içinde ele alır (1969a: 76). Bu bölümde Leibniz, çeşitli derecelerde sonsuzlar bulunduğunu iddia ederken, sayıları bütünü ve bütün ile parçaların ilişkisini soyutlamada kullanır: "Nicelik parçaların sayısıdır" (1969a: 76). Sayıyı en büyük evrenselliğe sahip bir şey olarak (1696a: 77) tanımlayan Leibniz'e göre, sayı açıkça metafizik alana aittir ve bu alanda varlığın durumları üzerine kavramsal çözümleme ve bütünleştirmede (1969a: 77) kullanılmalıdır. Leibniz, *Tez*'inde şöyle devam eder:

Bir bütün ve bu yüzden sayı ya da bütünlük, daha küçük bütünlükler olarak sayı ya da parçalara ayrılabilir. Bu, birbirinden farklı daha küçük bütünlüklerde ortak kısımların bulunduğu anlaşılması şartıyla, "karakter" in temelidir. Örneğin bütünü ABC olarak alalım, onun parçaları, AB, BC, AC olacaktır. Aynı zamanda en küçük parçaların ya da birbirleriyle ve bütünle ilişki içinde en küçük olarak alınanların (yani birliklerin) düzenini değiştirmek mümkündür; bu *situs* (konum) olarak adlandırılır (1969a: 77; Çevikbaş, 2006: 182 italik vurgu bana ait).

Burada bütün ve parça arasındaki ilişkide *görünüm* ve *konum* (complexion ve situs) üzerine çalışan Leibniz, görünüm ve konumun çeşitli varyasyonlarıyla elde edilebilecek durumların kombinasyonlarını ele alabilmek için *Tez*'inin "Tanımlar" başlığının 1'den 13'e numaralandırmasında çeşitli tanımlar vermektedir. Bu tanımlar "kombinasyon sanatının özünü oluşturan 'değişme', 'değişkenlik', 'konum', 'yakınlık', 'düzen' ve 'bir araya getirme' gibi" (Çevikbaş, 2006: 181) kavramlara aittir. "Değişkenlik" hem kombinasyon (birleştirme) hem de permütasyon (sıra değişikliği) yoluyla gerçekleştirilebilir. Örneğin, mutlak konumda öncelik ve sonralık önemlidir ve bu nedenle mutlak konum "düzen" olarak adlandırılırken, görelî konum "bir araya getirme" veya "derleme" olarak adlandırılır (Leibniz, 1969a: 77-8; Çevikbaş, 2006: 181). Böylece artık, değişmelerin matematiksel ifadesine ulaşılmış olur. Örneğin, dört şey yirmi dört biçimde değiştirilebilirken, bu dört şey on beş farklı yolla birbirine bağlanabilir (1969a: 78; Çevikbaş, 2006: 182).

Leibniz'in *Tez*'inden bir örnek olarak, bütünü ABCD olarak aldığımızda, daha küçük olan bütün iki parçadan oluşacaksa yani AB, AC, AD, BC, BD, CD olacaktır ki bu durumu ifade eden şey (exponent) iki (1969a: 78); şayet üç parçadan oluşacaksa yani ABC, ABD, ACD, BCD olacaksa bu durumu ifade eden şey üç (1969a: 78) olacaktır. Basitçe alınan tüm görünümler, tüm durum ifadeleri için hesaplanan görünümlerdir; örneğin 4 sayısının 15 durumunda 4 birim, 6 *com2natio* (*combinatio*), 4 *connatio* (*conternatio*) ve 1 *con4natio* (1969a: 78) vardır.

Tez'i incelendiğinde görülmektedir ki Leibniz, genel olarak Tanrı kanıtlanması, mantık, metafizik, fizik ve pratik alan üzerine ilksellere ulaşıldığı varsayılan terimler arasında kombinasyonlara bağlı önermeler türetmiştir. *Tez*'in "Tanımlar" (1969a: 78) başlığının yukarıda serimlemeye çalıştığımız açıklamalarından hareketle yapılabilecek doğru bir çıkarsama şu şekilde olabilir: Doğru bir analiz sonucunda, ulaşılabilecek ilk terimlerden (şeyler, tarzlar veya özellikle ilişkiler) türetilen terimler arasında kurulacak doğru bir birleştirme sonucunda, niyahet bir hiyerarşi oluşturulabilmektedir. Bu birleştirme bir senteze dayanmakta olup, böylece, sentez basitten karmaşığa doğru gelişen bir kavramlar ya da terimler hiyerarşisine ulaşılmasını sağlar.

Bundan sonra Leibniz, *Tez*'inde gözler önüne serdiği tüm çıkarımlarını ilerleyen sayfalarda aktarmaya çalışacağımız üzere, sonraki tüm çalışmalarında ve evrensel dilinde de kullanmak üzere geliştirmiştir.

1.4.2. Mainz Yılları

Tez'iyle büyük hayranlık uyandırmış olan Leibniz, üniversitede kürsü yerine tercih ettiği diplomatik görevler dışında, saray danışmanlığı gibi görevlerden dolayı 1667-1672 yılları arasında bir süre çalıştığı Mainz'de kaleme aldığı "A New Method for Learning and Teaching Jurisprudence" [Hukuk Bilgisini Öğrenmek ve Öğretmek İçin Yeni Bir Yöntem (1667)] isimli çalışmasının I.24. bölümünde, *Tez*'i üzerine geliştirdiği yöntemin temelinde yatan unsurlara şu şekilde yer vermiştir: "Konuların veya icat sanatının temeli *locid*dir, yani bütün, neden, madde, benzerlik ve bunlar gibi aşkın ilişkilerdir. *Dissertation on the Art of Combinations*'da gösterdiğimiz gibi önermeler, ilişkileri kombinasyon sanatı yoluyla birbirine bağlanan şeylerden yapılıdır" (1969b: 88).

Gündelik hayatında, verilmiş saray görevlerini yerine getirdiği Mainz yıllarında Leibniz, tüm yaşamının büyük planının bütünlüklü ilk resmi olarak ifade edebileceğimiz, zamanla ismi değişime uğramış ve asla bitirilememiş *Demonstrationes Catholicae*'yi [Evrensel Tanıtlamalar veya Katolik Tanıtlamalar], üniversite yıllarında yazdığı *Tez*'inin temelleri üzerine inşa etmeye odaklanmıştır. Bu çalışmanın belirgin bir resmi elimizde olmasa da Leibniz, *Evrensel Tanıtlamalar*'ı "şimdiye değin düşünülmüş tüm (olabildiğince) önermelerin ve faydalı şeylerin bir sistemi" (Antognazza, 2013: 85) olarak tanımlamıştır.

Tanrı'nın şanını yüceltmek ve insanlık durumunu iyileştirmek gibi bir hayale odaklanmış görünen ekümenist Leibniz'in *Tez*'inin temel fikir sağladığı *Evrensel Tanıtlamalar*'ı, zaman içerisinde *Scientia Generalis* ve sonrasında da *Elements de la Philosophie Generale et de la Theologie Naturelle* gibi isimler almış olsa da hayata geçirilememiş bir hayalden öteye gidememiştir. Ancak asıl önemli olan, Leibniz'den günümüze tüm bu çalışmaların anlaşılabilmesi için kullanılmasını öngördüğü bir evrensel dil, *characteristica universalis* adında bir kavramsal alfabe fikrinin miras kalmış olmasıdır. Leibniz, *Evrensel Tanıtlamalar*'ın epistemeye ve gerçek felsefeye götüren bilgisinin her bireye ulaşması için öngördüğü evrensel dilinde Llull'un *Ars Magna*'sındaki kombinasyon sanatından faydalanacaktı.

Esinlendiği Llull'un *Ars Magna*'sı hem kapalı bir sistem hem de matematiksel bir mantık temelinde yükselmesi nedeniyle evrensel bir sistem görünümüne sahip olup, dokuzlu çark sistemiyle Tanrı'nın sıfatlarına dair tüm önermeleri sistemin içinde kalmak suretiyle doğru sonuçlar vermekte ise de bu sistemde yeni bir şey üretmek mümkün değildir. *Ars Magna* bu haliyle bir yargı mantığının gereğini yerine getirebildiği bir sistemdir fakat daha önemlisi, Llull'un kombinasyon sanatından çok etkilenen Leibniz, bu sistemin bir *ars inveniendi*, henüz ayırdında olunmayan önermelerin yaratılabileceği bir keşif mantığının

öncüsü olduğunu ilk fark eden düşünürdür. Bu bağlamda, Leibniz, evrensel dil üzerine diğer tüm sistem üretme çabası gösteren dилcilerden ve filozoflardan ayrı bir yerdedir.

Ars Magna'da kullanılan kombinasyon sanatının esin verdiği ilk ismiyle andığımız projesi *Evrensel Tanıtlamalar*, şayet tamamına erseydi Leibniz'in nezdinde mantığın, matematiğin, metafiziğin, fiziğin, pratik alanda etik ve siyasetin ve diğer alanların ilk ilkelerinden itibaren aranan doğru bilginin bulunabileceği tek adres olacaktı.

Erişilebilir bir bilgi kaynağı olarak, yerel dilden bağımsızlaşmayı sağlayacak ortak bir evrensel dil sayesinde bu proje herhangi bir medeniyet ve din mensubunun da epistemeye kolayca ulaşabilmesi için bir alfabe ihtiyacı olduğunu göstergesiydi. “Böylece gerçek evrensel ya da daha doğrusu felsefi dil, aynı anda bu devasa, iki yönlü emeği varsayıyordu” (Couturat, 2004: 17): Kavramlarda ilksellere ulaşarak elde edilebilecek bir düşünce alfabesi ve bu alfabenin dayandırıldığı ansiklopedinin detaylandırılması. Leibniz, düşünce alfabesini “karakteristik” olarak adlandırmıştı ve “karakteristik, bilimsel gerçeklerin mantıksal bağlantısını ve hatta bilimlerin hiyerarşik düzenini belirlemeye hizmet ettiğinden, ansiklopedinin kurulması için vazgeçilmez” (Couturat, 2004: 18) bir konumdaydı.

Leibniz'in dehası, mantığı ilk ilkeler olarak disiplinlere uygulamak değil, kuramsal olanı pratiğe geçirirken zamanla tanımları değişebilen kavramların *neliğinin* değişmeyeceğinden hareketle, bir barbarda bile olduğundan emin olduğu çelişmezlik ilkesi dolayısıyla, basit kavramların sistematik tespitinde, kavramlar arasındaki *neliğe* bağlı bağlantıyı hesaba katmasında yatmaktadır. Böylece denilebilir ki Leibniz'in evrensel dil hayalinin temelinde yalnızca doğru kavramlara erişme hedefi bulunmadığı gibi, ayrıca *neliğinden* hareket edilen kavramların her birey tarafından doğruluğunun *a priori* olarak teyit edilmesi fikri de yatmaktadır.

1.4.3. Paris Yılları

Mainz yıllarında üstlendiği saray danışmanlığı görevi sonrası, Otuz Yıl Savaşlarının bitmesinden sonra bile Fransa'nın Hollanda, Viyana gibi Avrupa ülkelerine saldırma niyetine karşı duruş sergilemek ve hatta engel olabilmek için bazı diplomatik görevler verilmiş olarak, 1672 yılında Paris'e giden Leibniz burada, ülkesinde tanıdığı olduğu bilimsel çalışmaların tersine, dogmatik görünüm sergilemeyen bilim dünyasının çekiciliğine kapılmıştır. Daha önceki çalışmalarıyla hukuk, teoloji, mantık alanlarında başarılarla imza atmış düşünür olmasına rağmen, Paris yıllarında ona oldukça çekici gelen matematik, fizik ve metafizik gibi alanlarda kendisini eksik hissetmiştir. Paris'te karşılaştığı şekliyle denilebilir ki gerçek

anlamda özellikle matematik bilgisini, burada bilim cemiyetinden tanıştığı ve birlikte çok yönlü çalışmalar yaptığı insanlar ve önerdikleri yapıtlar sayesinde geliştirmiştir.

Leibniz, Port-Royal mantıkçılarının önde gelen isimlerinden Antoine Arnauld ile görüşmelerinin yanı sıra Paris'te bulunduğu 1672 sonbaharında, kendisiyle ilgili Heinrich Oldenburg'tan oldukça olumlu mektup almış olan Christiaan Huygens ile yaptığı görüşmelerden birinde, sonsuz serilerin toplamına dair bir yöntemi olduğunu dile getirerek dikkati üzerine çekmiştir (Aiton, 1985: 41-2). Diplomatik görevinin dışında bilim cemiyetinde yaptığı görüşmeler neticesinde, kendi icatlarından sadece biri olan, Blaise Pascal'ın (1623-1662) 1642 yılında icat ettiği hesap makinesinden çok daha fonksiyonel bir görünüm sergileyen hesap makinesinin tanıtımını 1673 yılında Londra'da yapma fırsatını bulup, bu sayede Kraliyet Cemiyeti üyeliğine seçilmiştir.

Pascal'ın icat ettiği hesap makinesi yalnızca toplama ve çıkarma gibi basit aritmetik işlemler yapabilmekteydi. Sonradan, makinenin içine yerleştirdiği bir aygıt sayesinde biraz daha geliştirilmiş haliyle, "Leibniz Çarkı" olarak da bilinen Leibniz'in icat ettiği hesap makinesi, bölme, çarpma, karekök alma gibi üst fonksiyonlara sahipti¹³.

Paris'te tanıştığı Huygens ile yaptığı yoğun matematik çalışmaları, Londra Kraliyet Cemiyeti sekreteri Oldenburg ile tanışıp, hesap makinesini sergilemesi ve Hollanda'da Spinoza ile Tanrı ve metafizik üzerine derinlikli sohbetlere dalarak geçirilen süreleri kapsayan yaklaşık beş yılın sonunda Leibniz, gerçekten felsefesi ve düşünsel dünyası açısından büyük bir bilgi edinimi ve sorgulama alanları sağlayacak tüm yeni donanımlarıyla birlikte, eski fikirlerini yeni yöntemlere bağlayacağı oldukça verimli bir çalışma alanına sahip olmuştur.

Pozitif bilimler ve bunlara dayalı icatlar, fizik-metafizik ayrımı gibi konular da dahil olmak üzere Leibniz'in Paris'te bulunduğu yaklaşık beş yıllık süre içerisinde fizik alanında yaptığı ses getiren çalışmaları olmuştur. Böylece, "çağın aşan düşünceleriyle uzayın ve zamanın mutlak değil, izafi olduğu fikrini geliştirmesiyle izafiyet teorisinin¹⁴ ve "hissedilir

¹³ Bu sistemi içeren cihazlar yirminci yüzyıla kadar üreilmeye devam etti (Davis, 2018: 5).

¹⁴ Leibniz Paris yıllarında fizik üzerine yaptığı çalışmalarında, Newton'un *Principia*'sında (1998: 72) uzay ve zamanı kendi başlarına varolan mutlak yapılar ve şeyleri bu iki yapının içinde tasvir ettiği fizik öğretilerine karşı, uzay ve zamanın, içinde bulunan şeylere göre göreceli olduğuna dair sonuçlara ulaşmıştır. Bu nedenle Boutroux'ya göre, Leibniz izafiyet teorisinin temellerini atmıştır (2017: 10) fakat Rescher aynı şekilde düşünmemektedir. Rescher'e göre, "teorisinin genel doğası gereği Leibniz, bazen Einstein'ın ve modern görelilik teorisinin habercisi olarak görülür. Ancak bu görüş yanlıştır veya her halükarda yanıltıcıdır" (1981: 84). Rescher, bu yanılmanın nedenini, Leibniz ve Einstein'ın izafiyet teorilerinin, temelde taşıdığı gerekçelerin özüne inilmeden aynı nedene bağlanmış olmasında bulmaktadır. Rescher'e göre Leibniz, ortak bir dünyada yer alan her şeyin karşılıklı uyumunu, tözlerin karşılıklı "algıları"nın uyumu dışında hiçbir varlığı olmayan uzay ve zamanın temeli (1981: 84) olarak almıştır. Buna mukabil, bilinmektedir ki Leibniz, şeylerin yanında uzay ve zamanın da göreceli olduğuna dair fizik öğretisi için yeni bir geometri üzerinde çalışmıştır. Leibniz'in Huygens'a yazdığı 1679 tarihli mektubunun (1969f: 249) ekinde dile getirdiğine göre, tözlerin birbirine karşı konumlarını belirleyebilmek adına *Analysis Situs* isimli risalesini de mektubuyla birlikte göndermiştir. Bu risalede serimlenen geometri, günümüzde

olmayan algılar” yaklaşımıyla da modern bilinçaltı psikolojisinin temellerini atmıştır” (Boutroux, 2017: 10). Leibniz, “On Copernicanism and the Relativity of Motion” [Kopernikçilik ve Hareketin Göreliliği Üzerine (1689)] isimli risalesinde, maddesiz uzayın hayali bir şey olduğu; bu minvalde matematiksel kesinlikte hareketin, cisimlerin birbirlerine göre konumlarındaki (situs) değişimden başka bir şey olmadığını (1989e: 91) iddia etmiştir.

Bütün bu buluşlarının ve bilime katkılarının yanında yine de en çok 1675 yılında buluşuna imza attığı, yaşadığı yüzyıl için devrim niteliğinde sayılan *sonsuz küçükler analizi* (infinitesimal calculus) ile ses getirmiştir. Bu buluş, bir devrim olduğu kadar ayrıca Newton ve İngiliz bilim dünyası ile Leibniz arasında, Leibniz’in ölümü sonrasında bile devam edecek olan intihal¹⁵ savaşlarının da nedeni olmuştur.

Her ne kadar Newton tarafından başta kibar bir dille sonra da ağır ithamlarla Leibniz’in kendi çalışmalarından intihal yaptığına dair söylemlerde bulunulmuşsa da geçmişe dayalı bilgilerden görüleceği üzere iki büyük düşünürümüz bu keşfi birbirlerinden bağımsız yapmışlardır. Newton ve Leibniz arasında gerçekleşen mektuplaşmalarda başlangıçta Newton, temkinli bir tavırla Leibniz’in çalışmasını duyurduğu 1675 ve yayımladığı 1676 yılından birkaç yıl önce aynı buluş üzerinde çalışma yaptığına dair kanıtlarını sunar. Bunun sonrasında iki dahi arasındaki mektuplaşmalarda Newton’un ilk buluş hakkını elinde tutmaya çalıştığı görülmektedir. Leibniz ne ile suçlandığını bilse de Newton’un kendisine ait olduğunu 1671 tarihli çalışmalarıyla ispatladığı ifadeler ve notasyonlar Leibniz için tanıdık değildir (Hall,

manifold yapılar ve topolojinin ilk nüvelerini taşıdığı düşünülen yeni bir geometri türüdür. Risi, *Geometry and Monadology* adlı kitabında, Leibniz’in matematik ve geometri dehası üzerine verdiği örnekte *topoloji* için Leibniz’in *analysis situs*’unu işaret etmektedir. Leibniz’in fikirlerinin ön ayak olduğu onlarca bilimsel alanın yanında, Risi kitabında, Einstein’ın klasik mekaniği Newton’un mutlak uzay teorisinden kurtarmış görelilik fikrini (bazılarına göre) kesinlikle Leibniz’e borçlu olduğunu (2007: 3) eklemiştir. Bilinmektedir ki Leibniz’in ölümünde sonra, ancak on dokuzuncu yüzyılda Riemann ve Grassmann bu yeni geometri üzerine çalışmalar yapmış (Loemker, 1969: 248) ve geliştirmişlerdir. İfade etmek gerekirse, “Riemann sistemini niteleyen özelliklerden biri doğruların sonlu ama sınırsız sayılması, diğeri ise uzay kavramının yeni bir içerik kazanmasıdır” (Hazar, 2017: 27). Nihayetinde Riemann’ın üç boyutlu Öklidyen geometriye zaman mefhumunu bir bilinmeyen gibi kabul edilmek suretiyle dördüncü bir boyut olarak katması üzerine Einstein, izafiyet teorisini Riemann geometrisi kullanarak (Kline, 1972c: 1131) serimlemiştir. Her ne kadar Rescher, metafizik temellerinden dolayı Leibniz ve Einstein arasında doğrudan bir bağ kurulmasını hatalı görüyor olsa da Leibniz’in zaman ve mekânın göreliliğinden hareketle yeni bir geometri üzerinde çalıştığı ve bu yeni geometrinin Riemann ve Grassman’da yer bularak taraflarınca geliştirildiği, dördüncü boyut olarak zamanın yerleştiği küresel bir geometrinin Einstein’da kuramsal açıdan izafiyet teorisinin açıklanmasında kullanıldığı gerçeğine dayanarak denilebilir ki doğrudan değilse de bilimsel süreklilik açısından izafiyet teorisinin ilk nüveleri Leibniz’de bulunmaktadır.

¹⁵ “Newton 1712’de Royal Society başkanı olma sıfatıyla kalkülüsün geliştirilmesinde öncelik sorunu üzerine bir soruşturma başlattı ve aslında kendi yazdığı bir bildiriye (...) onaylatarak Leibniz’in hırsızlıkla suçlanmasını ve kendisinin kalkülüsün yaratıcısı olarak doğrulanmasını sağladı” (Yardımlı, 1998: 43). Üzerinden çok zaman geçmiş olmasına karşın, bilindiği kadarıyla birbirlerinden habersiz yaptıkları buluşta, günümüz modern matematik kitaplarında Leibniz’in notasyonları mevcuttur. Leibniz ve Newton’un eş veya yakın (Newton’un kişiliğinden dolayı tarihler belirgin değil) zamanlı buluşu hakkında daha ileri okuma yapılmak istenmesi durumunda, Hall’un *Philosophers at War-The Quarrel Between Newton and Leibniz* (2002) isimli kitabı incelenebilir.

2002: 17). Öyle ki ikisinin buluşlarındaki hareket noktaları ve kaygıları farklıdır. Tümdengelimli ve sistematik felsefesi on yedinci yüzyıla yayılmış olan Descartes'ın özellikle hareketin önemi ile ilgili yazıları, Newton'u oldukça etkilemişti (Kline, 1972a: 327) dolayısıyla Newton için fizik yasaları temelde dururken, Leibniz için ancak Tanrı'nın müdrikesinde bitecek sonsuz mefhumuna dayalı metafizik kaygılar temelde yatmaktadır.

Newton fonksiyonların varyasyonlarını maddi cisimlerin hareketleriyle mukayese ederek, hız fikrini, yeni hesabının temeli haline getiriyordu. Leibniz ise bunun aksine, yeni analiz yönteminde, sonsuz küçük nicelikler mefhumunu [*notion*] konuya dahil ederek, fiziki dünyadan alınmış bir imaj olmayan metafizik bir ideayı kendine hareket noktası alıyordu (Boutroux, 2017: 21).

Bu buluş, bir yandan Leibniz'in başta Newton olmak üzere bilim çevrelerine karşı kendini aklama problemini getirdiği kadar, diğer yandan Leibniz'in düşünsel dünyasının bir sonucu olan monist sistemini en iyi açıklayabileceği organon haline gelmiştir. Düşüncenin hesaba indirgenebileceğine dair Hobbes'tan aldığı ve *kör düşünce* olarak adlandırdığı yaklaşımla birlikte, sonsuz küçükler sembolleştirmesinde yaptığı sembol manipülasyonu (günümüzde hala geçerliliğini koruyan notasyonlar olan, farklı derecelerin toplamları için f sembolü ve farkları için d sembolünün manipülasyonu) işe yaramış ve bu sayede, matematiğini aynı yolla yani kavramlara verdiği karakteristik sayılar ve bunlar arasındaki cebirsel işleme dayandırmıştır. Böylece, sonunda o ana kadar henüz bir hayal olan *evrensel dil* yaratma girişiminde de aynı şekilde sembol manipülasyonundan yararlanabileceğini keşfetmiştir.

1.4.4. Hannover Yılları

Onu destekleyen hamisinin ölümü üzerine işini kaybetmemek için Leibniz, 1672-1676 yılları arasında Paris'te özümsemeye çalıştığı tüm düşünsel etkinliklerini, yeni kazanımı olan modern bilimin diliyle Hannover'de yeniden yorumlamak için çalışmalarına başlamıştır. Onu neredeyse her gün zorunlu olarak masa başında tutan Guelf Hanedanlığının geneolojisini çıkarma görevi dışında, Harz madenlerindeki yorucu maden mühendisliği gereği, Harz madenlerinde kaldığı zaman dilimlerinde gerçek bir maden mühendisi olarak görev almıştır. Bir maden mühendisi, bir tarihçi, bir kütüphaneci olarak çalışan Leibniz, ayrıca etik, teoloji ve hukuk alanlarında da çalışmalarını sürdürmeye devam etmiştir.

Dehasının kapsamına yaraşır şekilde, eski fikirlerini yeniden yorumlamasına önyak olan Paris döneminin ardından, Hannover'de 1678-1679 yılları arasında, *characteristica universalis* ve *calculus ratiocinator* üzerine yaptığı çalışmalarla birlikte *logical calculus* (aritmetik kalkülüs ya da mantıksal hesaplama) adında yeni bir mantık alanının temelini

atmıştır. Leibniz'in, Aristoteles mantığından farklı biçimde olan yeni mantık sistemini açıkladığı 1690 tarihli "A Study In The Logical Calculus" [Mantıksal Hesap Üzerine Bir Çalışma, (1969: 371-382)] isimli risalesinde, sembol manipülasyonu yaparak, mantığın dilsel yapısını sembolleştirme çabası neticesinde, bugün bildiğimiz toplam sembolünü ve diğer sembolleri işleyiş şekli kayda değerdir. Sembol manipülasyonunu bizim için can alıcı bir nokta haline getiren kısım, Leibniz'in bu risalesinde (ikinci aksiyomunda) ¹⁶ günümüzde kullandığımız şekliyle (+) işaretini bir matematiksel ögenin verdiği anlamda "toplam" olarak değil, bir mantıksal öge olarak kullanmış olmasıdır.

Mantık ve karakteristikler üzerine yaptığı çalışmalar dışında, Leibniz kendisine ait *arctanjant* hesabının yanında, *sonsuz küçükler hesabının* da etkisiyle fizikte kuvvete alan açarak (Descartes'ın yorumundan farklı bir şekilde) termodinamiğin birinci yasasının temelini atmıştır.

Leibniz, Descartes'ın maddenin saf uzam olduğuna dair görüşünü yetersiz bulup, maddenin hem uzam hem de kuvvet olduğunu iddia etmiştir. Leibniz, kuvvet ile hem pasif momentumu hem de aktif kinetik enerjiyi kastetmiştir. Böylece Leibniz, uzamın madde için gerekli olmadığına karar verdiğinde, tek başına kuvvetin varlığını gerekli görmüştür. O halde uzam, kuvvetle donatılmış noktalar arasındaki ilişkilerin düzenidir (Agassi, 1969: 332).

Leibniz'in, Paris'te olduğu dönem 1675 yılında *infinitesimal calculuste* yapmayı başardığı sembol manipülasyonundan sonra, her şeyin her şeyle bağlantılı olduğu felsefesinde özellikle büyük hayali *Evrensel Tanıtlamalar*'ı anlaşılabilir kılacağı *characteristica universalis* keşfetmesi ¹⁷ oldukça heyecan vericidir. Leibniz'in *characteristica universalis* üzerine düşünce etkinliğinin ilk nüvelerini Paris'te bulunduğu 1675 yılında Henry Oldenburg'a yazdığı mektupta görebilmekteyiz. Bu mektubunda Leibniz, Dalgarno ve Wilkins tarzında bir evrensel dille ilgilenmediğini, "kendi karakterler fikrinin, gerek Çince imler modeline dayalı bir evrensel yazı kurmak isteyenlerin, gerek anlamsal belirsizliklerden arındırılmış bir felsefi dil kurmuş olanların fikirlerinden büyük ölçüde farklı olduğunu yineler" (Eco, 1995a: 268). Mektubunda Leibniz, gerektiği kadar vakit ve sağlığı olması durumunda, ilk duyacaklar için kulağa çok farklı gelecek olsa da sıklıkla "combinatorial

¹⁶ Leibniz'in 1690 tarihli "A Study In The Logical Calculus" [Mantıksal Hesap Üzerine Bir Çalışma, (1969: 371-382)] isimli risalesindeki sembol manipülasyonuna ve bu risalenin ikinci aksiyomuna, tezimizin 2.2.2 altbaşlığında Boole ile bağlantısını serimlemek üzere yeniden değineceğiz.

¹⁷ Leibniz'in Henry Oldenburg'a 1675 yılında yazdığı mektubun (1969c) tezimizde kullandığımız çevirisinde, Dalgarno ve Wilkins ya da Çince ile ilgili herhangi bir bilgi bulunmamakla birlikte, Umberto Eco'nun, ilk Leibniz derlemelerinden biri olan 1875 tarihli Gerdhardt alıntısında (1995a: 268) bu mektup daha kapsamlı görünmektedir. Bu nedenle tezimizde, iki kaynaktan da faydalanmış bulunmaktayız.

characteristic” (kombinsayonal karakteristik) olarak adlandırdığı kelimelerin gücünü, kurallar ve örneklerle açıklayabilmeyi (1969c: 166) ümit etmektedir.

Hobbes’un öğretilerini de izleyen biri olarak Leibniz, “insan düşüncesinin yazılı veya sözlü kelimeler, kağıda yazılan şekiller veya başka maddi veya zihinsel imgeler gibi algılanabilir ‘işaretler’le desteklenmeye her zaman ihtiyaç duyduğu” (Antognazza, 2013: 83) görüşündedir. Leibniz, *Evensel Tanıtlamalar* için yaptığı çalışmalarda özellikle çağdaşı Dalgarno’nun varlık listesi üzerinde yoğunlaşmıştır.

Kendi bileşim ölçütüne karar verirken Dalgarno, oldukça cesur bir görüşü savunur; buna göre, her töz, bir ilinekler bileşkesinden başka bir şey değildir. Bu Porphyrios sınıflamasının neredeyse kaçınılmaz bir sonucuydu, ancak Aristoteles geleneği umarsız bir çabayla bu sonucu göz ardı etmeye çalışmıştı. Dalgarno sorunla yüzleşir, ancak ilineklerin sonsuz olduğunu teslim eder. Öte yandan, aşağı türlerin de çok fazla olduğunu fark eder (sayılarının 4.000 ila 10.000 arasında olduğunu hesaplar) ve iki binden fazla türden oluşan bir sınıflamaya ulaşan Wilkins’in önerilerini kabul etmemesinin nedeni de olasılıkla budur. Dalgarno’nun korkusu şudur: İnsan böyle bir yöntemle, bir kadavrayı Petrus’u Jacobus’tan ayırt etmek olanaksız hale gelinceye kadar en ince parçalarına ayıran anatomistin durumuna düşebilir (Eco, 1995a: 232).

Çalışmaları neticesinde, Dalgarno’nun varlık listesinde göze çarpan töz-ilinek problemini gözlemleyen Leibniz, kendine has yöntemini Dalgarno ve Wilkins’in girdiği çıkmazdan kaçınmak için seferber etmiştir. Leibniz, matematiksel formlar ve içeriklerden faydalanarak kendi dil sistemini geliştirme çabasına girmiştir.

Orijinal işaretlerle akıl yürütmenin matematiği değiştirdiği gibi, uygun işaretlerle akıl yürütme, insan araştırmasının her alanını geliştirmeye yardımcı olabilir. George Dalgarno ve John Wilkins’in çalışmalarını genişleten Leibniz, bilgiyi iletmek, organize etmek ve yaymak için daha güçlü işaret sistemleri üretmeye çalıştı (Jones, 2006: 233).

Böylece denilebilir ki on yedinci yüzyılın düşünce karakteristiğine dönüşmüş matematiksel yöntem, düşünürler ve filozoflar nezdinde episteme için başlangıç noktasının doğru belirleniminde elzem sayılırken; Leibniz, matematiğin sadece yöntemini değil ayrıca ve özellikle dilini de almıştır. Bu farkındalık eşliğinde Leibniz, kendi çağında yirmili yaşlarının sonlarında Kraliyet Cemiyeti üyeliğine kabulünü sağlayan hesap makinesini icat etmiş olması neticesinde, nitel olanın nicel açıklamayla dile getirilebileceği bir evrensel dil ile insanların bir hesaplayıcı sayesinde argümanlarının doğruluğunu tartabileceğini iddia etmiştir.

(y)anılmıyorsam, akıl yürütmenin sayılar aracılığıyla kanıtlanabileceğini gösterecek şık bir araç düşündüm. O halde, bu çok harika karakteristik sayıların önceden verildiğini hayal ediyorum ve

karakteristik sayıların belirli genel özelliklerini gözlemledikten sonra, bu sayıların her ne olurlarsa olsunlar, o özelliğe sahip olduklarını hayal ediyorum. Bu sayıları kullanarak tüm mantıksal kuralları gösterebilirim, ayrıca sayılar aracılığıyla şaşırtıcı bir şekilde belirli argümanların uygun biçimde olup olmadığını nasıl bilebileceğimizi gösterebilirim. Şeylerin gerçek karakteristik sayılarına sahip olduğumuzda, nihayet, herhangi bir zihinsel çaba ya da hata tehlikesi olmaksızın, argümanların gerçekten maddi olarak sağlam olup olmadığını yargılayabilecek ve doğru sonuçları çıkarabileceğiz (Leibniz, 1989b: 9-10).

Bu icat ışığında, zihin üzerine yapılabilecekler konusunda bir öngörü kazandığı açıkça görülen Leibniz, düşüncenin hesaba indirgenmesinde, mantıktan yola çıktığı “Preface to Universal Characteristic” [Evrensel Karakteristikler İçin Önsöz (1989b)] ya da başka kaynaklarda kullanılan diğer ismiyle “On the General Characteristic” [Genel Karakteristikler Üzerine (1696ç)] ve daha sonra “Samples of the Numerical Characteristic” [Nümerik Karakteristiklere Örnekler] isimleri altında iki çok önemli çalışmasını kaleme almıştır. Bu çalışmaların kaleme alındığı 1679 yılı, Leibniz’in hayatının sonuna değin sürdürdüğü hayali evrensel dilinin yanı sıra mantık, matematik, geometri, fizik ve daha nice alanlarda en üretken olduğu dönem olarak bilinmektedir.

Bu dönem, büyük projesi *Evrensel Tanıtlamalar* ve evrensel ansiklopedi fikirleri için Leibniz’in önceden yaptığı tüm hazırlık ve çalışmalarının bir araya geldiği bir dönemdir. “Genel Karakteristikler Üzerine” isimli risalesinde özetle; en temel metafizik figür olarak gördüğü sayı ve keşfedilmiş şeylerin gücünü yansıtan aritmetiğin, evrenin bir tür değişmezleri olarak görev yaptığını ifade etmektedir (1969ç: 221). Bu risalede Leibniz’e göre, gizeminin nedeni çözülemeyen *sayı*, Pisagor’dan sonra da zaman içinde batıl inançlar eşliğinde Kabalacılıkta, harflere nümerik anlamlar vermek suretiyle yapılan sayı mistisizminin dışında büyü için de kullanılmıştır. Aynı şekilde, yakın dönemlerde mucizelerin sayı, karakterler, kimilerinin Âdemi dil dediği yeni bir dil aracılığıyla ifade edilebileceğine dair eğilimler ortaya çıkmıştır. Onca çaba gerektiren çalışmaya rağmen, Leibniz’e göre asıl önemli olan, hiç kimsenin her şeyin kendi karakteristik sayısıyla belirlenebileceğini fark edememiş olmasıdır. Leibniz, bu risalesinde şöyle devam etmektedir:

Kısa bir süre öncesine kadar, bazı seçkin kişiler, farklı ulusların düşüncelerini kendi dillerinde iletebilecekleri ve her birinin kendi dilinde, diğerinin yazdıklarını okuyabileceği, tüm kavramların ve şeylerin güzelce düzenlendiği bir dil veya evrensel karakteristik tasarladılar. Lakin hiç kimse işaretleri veya karakterleri, sayılar için aritmetik işaretlerle aynı görevi yerine getiren ve soyut olarak kabul edilen büyüklükler için cebirsel işaretlerin yaptığı şekilde hem keşif sanatını hem de yargılama sanatını aynı anda bünyesinde barındıran bir dil ya da özellik öne sürmemiştir (1969ç: 222).

Keşfinin değerinin farkında olan Leibniz, Foucher'e yazdığı mektubunda (1675) özetle, Descartes'ın aktardığı onca güzel ve orijinal fikirlerini düşünürken, Descartes'ın başaramadığı şeylerden ziyade başardığı şeylere bakarak hayranlık içinde kaldığını ifade eder (1989a: 2). Ona göre, bir insanın aynı anda tek başına her şeyi başarabilmesi mümkün değil iken Descartes bunu yapmıştır. Leibniz'in, "Evrensel Karakteristikler İçin Önsöz" isimli risalesinde ise değindiği üzere, *Organon* ve *Metafizik* gibi iki eseri yazarken düşüncenin nosyonlarına derinlikli nüfuz etmeyi başarmış olan Aristoteles'in, bilimleri sınıflandırmada belki de Descartes'dan daha başarılı olan Joachim Jungius'in ve matematiksel düşünce patikasını bulan ve daha yukarılara çıkmamızı sağlayan Descartes'ın (1989b: 7) evrensel dil yaratmak için çaba sergilememiş olmaları ve düşüncenin matematiksel forma indirgenme fikrine ulaşamamış olmaları konusunda şaşkınlığını gizleyemez.

Descartes'ın evrensel bir dil bulma çabası içinde olmadığı tarihsel olarak da açıktır. Ancak yine de bu durum, Descartes'ın ne dil problemlerinden ne de evrensel dil fikrinden habersiz olduğu anlamına gelir. 1629 yılında Rahip Marin Mersenne'e yazdığı mektupta Descartes, Des Vallées adındaki kişinin "kusursuz dil" fikri hakkında şu yorumda bulunmuştur:

Ben böyle bir dilin mümkün olduğuna ve bu dile imkan verecek bilimin keşfedileceğine inanıyorum; bu dil aracılığıyla köylüler, hakikate filozofların şu anda yaptığından daha iyi hükmedecektir. Ancak söz konusu dilin kullanıldığını görmeyi asla umut etmeyiniz; bu dil, eşyanın düzeninde büyük değişiklikleri gerektirir ve evvela bütün dünyanın bir yeryüzü cenneti olmasını icap ettirir (Descartes'dan aktaran Altınörs, 2020: 55).

Bir evrensel dil yaratma çabasında olan Leibniz'in yukarıda adı geçen aynı tarihli makalelerinin yanı sıra Leibniz, yine aynı yıl, 1679'da hamisi Hannover dükü Johann Friedrich'e büyük planının ve hayalinin ne işe yaracağını anlattığı bir mektup yazarak kendisinden projesi için ayrıca maddi destek de istemektedir.

Eğer Tanrı, Siz Yüce Efendimiz'e, bana tahsis etme lütfunu gösterdiğiniz 1200 gümüş sikkeyi sürekli bir gelire dönüştürme düşüncesini esinlerse, Ramon Llull gibi mutlu olacağım, belki de bunun karşılığında daha büyük bir hizmet sunarak... Çünkü benim buluşum aklın eksiksiz kullanımını içerir: Görüş ayrılıklarında bir yargıç, bir kavram yorumcusu, bir olasılıklar cetveli, deneyimler okyanusunda bize yol gösterecek bir pusula, bir nesnel envanteri, bir düşünceler çizelgesi, var olan şeyleri irdelemek için bir mikroskop, uzak olanları kestirmek için bir teleskop, genel bir hesap, masum bir büyü, hayal ürünü olmayan bir Kabala, herkesin kendi dilinde okuyacağı bir yazı, hatta birkaç haftada öğrenilebilecek ve tüm dünyada yaygınlık kazanacak bir dildir. Ve bu dil, gittiği her yere gerçek dini götürecektir (Leibniz, 1969g: 262; Eco, 1995a: 13-4).

Bu yıllarda Leibniz, bir evrensel dil yaratma konusunda aynı coşkuyu paylaştığı çağdaşları olan Dalgarno ve Wilkins'in etkisi altında sesletilebilir bir ideal dil projesinin peşinde giderken, gerçek karakteristiklere ulaşabileceğini ve bunun düşüncenin alfabesi olacağını rahatlıkla iddia ediyor olsa da sonraki dönemlerinde artık sesletilebilir değil, daha mütevazı bir dil teklif etmektedir. Bu dil gittikçe biçimselliğe kayan, sentaktik kurulumunu cebirsel kılan ve içerikten ziyade “kavramlar ve önermeler arasındaki mantıksal ilişkileri ifade eden” (Jolley, 2004: 10) bir dile dönüşür.

1.4.5. Leibniz Felsefesinin Belirginlik Kazandığı Dönem

Paris yılları sonrası Hollanda'da geçirdiği kısa süre zarfında, Spinoza'nın Tanrı tanımazlıkla suçlanmasına neden olan felsefi görüşlerinin sıralandığı, geometrinin aksiyomatik sistemiyle yazılmış *Ethica* adlı eserinin bir kopyasını okuma şansını yakalayan Leibniz ile Spinoza arasında geçen derinlikli sohbetlerin ardından, döndüğü Hannover'de belki de en çok risale kaleme aldığı yıllar olan 1679 ve sonraki yıllar, Leibniz'in felsefi sisteminin belirginleşmeye ve olgunlaşmaya başladığına dair ciddi göstergelere sahiptir.

Öncelikli olarak, bir tilmizi olduğu Descartes'ın uzamsallık ve doğrusallık barındıran, Öklidyen katı mekanik sistemi ve felsefesinden oldukça etkilendiği Spinoza'nın Öklid geometrisinin katı kurallarının yönettiği “her şeyin bir nedeni vardır” gibi *nedensellik ilkesi*, Leibniz'i felsefi sistemini geliştirme konusunda uyarmıştır. Spinoza ile geçirdiği zamanlarda yaptığı derinlikli sohbetler sonucunda Spinoza, Leibniz'e etkilendiği ve felsefesini Spinozist görünmekten korumak adına çözmek zorunda kalacağı problemler bırakmıştır.

Spinoza'ya göre her şey mutlak bir mantıksal zorunluluk tarafından yönetilir. Zihinsel alanda özgür irade veya fiziksel dünyada şans diye bir şey yoktur. Olan her şey Tanrı'nın esrarengiz doğasının bir tecellisidir ve olayların olduğundan farklı olması mantiken imkansızdır (Russell, 1945: 571).

Spinoza'nın *Ethika*'da beyan ettiği düşüncelerine göre, nedensellik ilkesi bağlamında değerlendirildiğinde Tanrı bile özgür olamaz, çünkü Tanrı da tıpkı insan gibi doğası gereği davranmaktadır ama insan, Tanrı'dan farklı olarak kendini özgür iradeyle hareket ediyor zanneder.

(i)nsanlar kendilerinin özgür olduğunu düşünürler, çünkü kendi seçimlerinin ve arzularının farkındadırlar ama kendilerini bunları seçmeye ve arzulamaya sevk eden nedenlerden bihaberdirler; hatta öyle bihaberdirler ki bu nedenleri rüyalarında dahi görse inanmazlar. Ayrıca yine bu öncüllerden şöyle bir sonuç çıkacak: İnsanlar daima bir amaç doğrultusunda hareket eder, özellikle elde etmek istedikleri çıkarları doğrultusunda. Bu iki sonuç dikkate alındığında, anlaşılır ki insanlar sadece

gerçekleşen olayların son nedenlerine bakar (...) başka bir neden arama ihtiyacı duymazlar (Spinoza, 2012: 73).

Leibniz, “Comment on Spinoza’s Philosophy” [Spinoza Felsefesi üzerine Yorum (1707)] isimli risalesinde, *Ethika’da* Spinoza’nın, Tanrı’nın zorunlu olarak var olduğunu, kendisinin ve diğer her şeyin nedeni olduğunu vurguladıktan sonra, buna karşın Tanrı’nın şeyleri özgürce ürettiğine dair düşüncelerini kabul edemeyeceğini (1989f: 275) dile getirmektedir. Gerçekten de zorunluluktan doğan bir özgür yaratma durumu oldukça çelişik görünmektedir. Spinoza’nın “nedensellik ilkesi”nden hareketle A’nın B’ye; B’nin C’ye ve bu şekilde devam eden nedensellikte “determinizm”inin özgür iradeye alan bırakmadığını gören Leibniz, kendi felsefesine alan açabilmek için bir ilke önermek zorunda kalacaktır.

Anlaşılan o ki Leibniz, “doğada hakiki olumsuzluğa giden yolu tıkayan bir zorunlulukçuluktan giderek daha fazla hoşnutsuz hale” (Rescher, 1981: 109) gelmiştir. Leibniz, felsefesinde Spinozist görünmekten kaçınabilmek için “yeter neden ilkesi”ni yürütmekten başka, Spinoza’nın evrensel zorunlulukla sonuçlanan “nedensellik” ilkesine çözüm olarak ürettiği bir de “mümkün dünyaların en iyisi” görüşünü pekinleştirme çabasına girecektir. Bu görüşlerin ilkinin temelinde olumsuzluk, ikincisinin temelinde ise olasılık mantığı ve teorisi vardır. Leibniz’in “yeter neden ilkesi”ni ilerleyen sayfalarda serimlemeye çalışacağımız üzere, Descartes’ın biraz da sezgisel görünen “açık seçik bilgi”nin imkanına dair argümanlarını çürütmek üzere kullandığı da rahatlıkla söylenebilir.

Descartes’ın felsefesine, fiziğine, metafiziğine, bilgi kuramına karşı getirdiği argümanlarıyla eleştirel yaklaşan Leibniz için Spinoza’nın *tözün* tek ve ancak Tanrı olduğuna dair düşüncesi ne kadar eksikli görünüyor ise Descartes’ın *tözü* uzamsal bir şey olarak kabul etmesi de bir o kadar eksik ve hatta yanlıştır.

Descartes için maddenin özü, uzam iken Spinoza’da uzam ve düşünce Tanrı’nın yüklemeleridir. Leibniz’e göre uzam töze yüklem olamaz, çokluk içerir ve tözler yığımına aittir. Uzam tözün bir yüklemi olmaktan çıkarıldığında doğal olarak gerçekte varolan her monadın bir ruh olduğu sonucuna ulaşılabilir (Russell, 1945: 583).

Bilinmektedir ki Leibniz için *töz* yani *monad*, aldığı her yüklemle kendini açığa çıkaran bir şeydir ve hatta denilebilir ki tözler yani monadlar ile yüklemeler arasındaki ilişki, geometrik ve matematiksel bir süreklilik durumudur.

Spinoza ve Descartes’a eleştirilerini sıralarken kendi kavramlarını üreten Leibniz, özne ve yüklem (predicament) arasında sonsuza kadar sürdürülebilecek bir ilişki öngörmüştür fakat bunu özdeşlik ilkesiyle değil, bir tür sonsuz kavram analizi ile yapmıştır. Bunun

nedenine gelince, “varoluş hakikatleri düzleminde Leibniz’i ilgilendiren şey yüklemle öznenin özdeşliği değildir; bir yüklemden ötekine geçilmesidir” (Deleuze, 2007: 64).

Leibniz’in herkeste bulunduğundan emin olduğu çelişmezlik ilkesi dolayısıyla, biçimsel formülasyonda “A, A’dır” önermesi türünde önermeler, analitik bir yargı oluşturan doğru önermelerdir. Bu tip yargılara “öz hakikatleri” diyen Leibniz’e göre matematik hakikatler ve Tanrı hakikatleri de yine burada bulunmaktadır. Biçimsel formülasyonunda tersinin de “A, A’dır” olması dolayısıyla yeni bir şey vermeyen bu ilkenin, Leibniz’in yaratıcı düşüncesinde felsefi açıdan değerlendirilmesi sonucunda; “her analitik önerme doğrudur” açıklamasıyla o halde “her doğru önerme zorunlu olarak analitiktir” açıklaması gerçekten aynı şey midir sorusu Leibniz’i açık-seçik bilgi kavrayışında yeni bir ilke yaratmaya sevk etmiştir: “Yeter neden ilkesi”. “Yeter neden ilkesi” şöyle ifade edilebilir: “Bir öznenin başına gelecek her şey, mekân ve zamansal belirlenimler, ister bağıntı isterse olay olsun, bir özneye olacak her şey, yani onun hakkında doğru olarak söylenebilecek her şey ihtiva edilmelidir” (Deleuze, 2007: 28).

Bilinmektedir ki özdeşlik ilkesine bağlı hakikatler, tersi mümkün olmayan ya da tersi çelişik olan öz hakikatleri veya zorunlu hakikatlerdir. Oysa yeter neden ilkesine bağlı hakikatler-nihayetinde Leibniz’i “mümkün dünyaların en iyisi” düşüncesine kadar götüren ve Leibniz’e olasılık felsefesinin temellerini attıran-tersi mümkün olabilen olumsal (contingent) yani varoluşa bağlı hakikatlerin bilgisine götürür. İki hakikat türüne ulaşmada yürütülen ilkelerin yöntemsel açıdan belirgin bir farkı da mevcuttur: İlkinde, öz hakikatlerine örnek olarak gösterdiği matematik hakikatlerden hareketle, çelişmezlik ilkesini yürütmek için bir sonlu analiz veya sonlu adımlarla yürütülebilecek ispat yöntemi önermektedir. İkincisinde ise sonsuz analizin varlığı söz konusudur.

Öyle ise öz hakikatleri sonlu analiz isteyen matematik hakikatleridir. Varoluş hakikatleri sonsuz analiz isteyen ve yalnızca Tanrı’nın kavrayışında gerçekleşen metafizik hakikatlerdir. Leibniz’in kaleme aldığı “On Freedom and Possibility” [Özgürlük ve Olasılık Üzerine (1680-82?)] adlı kısa metni, öz hakikatleri ile varoluş hakikatleri arasında ayırım yaparak Descartes ve Spinoza’nın birey özgürlüğüne alan açmayan katı felsefi sistemlerine bir yanıt niteliğindedir.

Çünkü her şeyden önce, mümkün olan ama yine de zorunlu olmayan ve gerçekten var olmayan bazı şeylerin olduğuna dair bir olasılık ve zorunluluk kavramına sahibim. Bu yüzden, özgür bir zihni her zaman bir şeyi diğerine tercih etmeye zorlayan bir akıl (bu akıl, Tanrı’da olduğu gibi bir şeyin mükemmelliğinden mi yoksa bizim kusurumuzdan mı kaynaklanıyor?) özgürlüğümüzü ortadan kaldırmaz (Leibniz, 1989ç: 20).

Leibniz, özgürlük ve olumsuzluğun nedenler zinciri ve öngörü ile nasıl uyumlu olduğuna dair sorunsalın bir çözümünü, “On Freedom” [Özgürlük Üzerine (1679)] başlıklı risalesinde varoluş hakikatlerini dayandırdığı “yeter neden ilkesi” ile serimlemeye çalışırken şu ifadeleri¹⁸ kullanmıştır.

İki temel ilke vardır: Birincisi, bir çelişkiyi ima eden her şeyin yanlış olduğu şeklindeki zorunlu şeyler ilkesi ve diğeri, daha mükemmel olan veya daha fazla sebebe sahip olan her şeyin doğru olduğu olumsuzluk ilkesi. Metafiziğin tüm hakikatleri veya mantık, aritmetik, geometri ve benzerleri gibi kesinlikle zorunlu olan tüm hakikatler, önceki ilkeye dayanır çünkü onları reddeden birine tersi her zaman bir çelişki anlamına gelir. Yalnızca Tanrı'nın veya başka bir varlığın iradesi hipotezine ihtiyaç duyan, doğası gereği olumsal olan tüm gerçekler, ikinci ilkeye dayanır (1969ğ: 264).

Daha önce de aktarmaya çalıştığımız şekliyle, sonsuz küçükler analizi hem fizik hem de metafiziğe uygulanabilecek nüveler taşımaktadır, dolayısıyla “Leibniz metafizikte belli bir sonsuz analiz fikrini keşfettiğinde, bunun yine bizzat kendinin keşfettiği belli bir hesap yöntemiyle, yani sonsuz küçüklükler hesabıyla bir yansıma ilişkisi içinde olmadığını düşünmek imkansızdır” (Deleuze, 2007: 52). Sonsuz küçükler analizini bulmasından sonraki dönem, büyük olasılıkla 1679 yılına denk gelen bir kavram olan “yeter neden ilkesi”nin, sonsuz küçükler analiziyle olduğu kadar ileri tarihlerde Leibniz’in *Monadoloji*'sinde aktaracağı şekliyle *monad*larla ve Leibniz'in perspektif felsefesiyle açık bağları mevcuttur.

Peki varoluş hakikatlerini sonsuz analize tabii tutsak bile ancak Tanrı'nın müdrikesince bilineceğine dair iddiasına rağmen Leibniz, sonsuz analiz fikrini elden bırakmadan, sonsuz analizin belirsizliğine sınır getirmeyi nasıl başarmıştır?

Spinoza'nın nedensellik ilkesi, belirsizliğe açılan bir sonsuz analiz demektir. Daha önce serimlemeye çalıştığımız üzere Leibniz, varoluş hakikatlerinde *bireyci* (individualist) yaklaşımının bir sonucu olarak tanımlayabileceğimiz “yeter neden ilkesi” neticesinde *bireyi/tözü/şeyi*, kendi mefhumuyla ilişkisi içinde tanımlamaktadır. Bu durumda, bu ilke gereği tözün başına gelmiş olan her şey bireysel mefhumunda içerilmiş (Deleuze, 2007: 117) demektir. Her kavram analizi töz bağlamında kaldığından, kendi içinde kesinti ya da sıçrama yaşamaz çünkü doğada süreklilik vardır. Bu süreklilikte, analiz sonsuza doğru yürütüldükçe bir diferansiyel hesabı neticesinde yitip giden farklar söz konusu olacaktır. Nasıl ki bir çokgenin kenar sayısı arttırıldıkça, bir daireyle sınırlandırılabilir ya da bir eğri, bir doğru ile

¹⁸ Leibniz'in farklı zamanlarda yazdığı risalelerin ve mektupların birer derlemesi olan ve tezimizin bu bölümünde sıklıkla başvurduğumuz Loemker (1969) ile Ariew ve Garber'in (1989) çeviri ve editörlüklerini yaptıkları ayrı kaynaklarda, risalelerin isimlerinden de anlaşılacağı üzere her ne kadar yakın başlık isimleri kullanılmışsa da yalnızca yukarıda görülen son alıntı, bir paragraf şeklinde ortaktır. Geri kalan kısımlar birbirinden ayrı ilerlemektedir.

sınırlandırılabilirse, bu belirlenim ve sınırlandırma Leibniz'e göre, varoluş hakikatlerinde de işletilebilirdi. Böylelikle, bireysel mefhum bağlamında geriye doğru işletilebilir bir kavram analizinin niceliksel formla yeni bir açıklama alanı edindiği söylenebilir:

dx, *x*'in diferansiyelidir, *dy* ise *y*'nin diferansiyeli (...) uyuşma dayalı olarak diyeceğiz ki, *dx* ya da *dy* *x*'e ya da *y*'ye eklenen ya da onlardan çıkartılan (varsayıyoruz bunu) sonsuz küçük niceliklerdir (...) yani ele alınan niceliğin en küçük değişmesi, varyasyonu. Bu uyuşmsal olarak herhangi bir değerle yüklenemez. Demek ki *x* için $dx=0$ 'dır: *x*'in değişebileceği en küçük nicelik... demek ki sifıra eşittir, *y*'ye göreyse $dy=0$. Kaybolmakta olan fark mefhumu ete kemiğe kavuşmaya başlar. Bu bir değişme ya da farktır *dx* ya da *dy*... verilmiş ya da verilebilir bütün niceliklerden daha küçüktür bunlar (...) İşte diferansiyel hesabın sembolizminde büyüleyici olan şey: *x*'e göre $dx=0$ -en küçük fark, *x* ya da *y* niceliklerinin yapabildikleri en küçük artış- sonsuzca küçük (...) dx/dy sifıra eşit değil... üstelik dx/dy tastamam ifade edilebilecek bir nicelik... Bunlar sadece görelî... *dx*, *x*'e göre bir hiç, *dy*, *y*' e göre bir hiç ama (...) dy/dx işte bir şey... bir nicelik (...) Eğer diferansiyel ilişkiyi ele alırsanız sıfır değildir- belirlenmiştir, belirlenebilir (Deleuze, 2007: 67).

Leibniz, “yeter neden ilkesi”nin doğru bir yürütmesini yapabilmek için ayrıca “ayırt edilemezlerin özdeşliği ilkesi”ni kavramsal olarak icat etmiştir. “Bu ilkeye göre, evrende birbirinin aynı olan iki birey var olamaz” (Çevikbaş, 2010: 13). Leibniz böylece, yitip giden farklar argümanı neticesinde doğada birbirinin aynı hiçbir töz bulunmadığını, bulunamayacağını, yine sembol manipülasyonu yaptığı diferansiyel hesap ile göstermiştir. Bu şekilde, bir sınır komşuluğunda her töz kendine içkin, belirli ve tekil bir nokta gibidir. Matematiksel görü ile iyi çalışılmış argümanlar neticesinde, her tözün bireyselliğine ulaştığı bu nokta, yani “ayırt edilemezlerin özdeşliği ilkesi”nin icadı, Leibniz'in *monad* kavramının doğru bir teknikle anlatılabileceğinin en önemli işaretlerindedir.

Yukarıda serimlemeye çalıştığımız üzere, Leibniz'in diferansiyel hesap üzerine metafiziksel açıklamalar sunabilen bu keşfi, cebir için farklı kuvvetlere sahip bilinmeyenler üzerinde yaşadığı matematiğe dayalı işlemsel teknik sorunların da bir çözümünü sağlamıştır. Aslında Leibniz'e göre cebirsel işlemler yürütülürken diferansiyel hesap bir şekilde hep oradaydı.

Aşkınların analizinin veya bir sonsuzun dikkate alınmasını içeren geometrinin mükemmelliği, kuşkusuz, yaptığı her şeyde sonsuzu tanıtan doğa üzerinde yapılabilecek uygulamalar nedeniyle daha da önemli olacaktır ama belki de bunun tek nedeni, onun hakkında yeterli fikirlere sahip olamamız ve her zaman tam bir seçiklikle algılayamayacağımız unsurları ortaya koymasındır (Guénon, 2004: 36).

Dönemin bilim çevrelerince, sonsuz kavramının doğası gereği belirsiz duruşu, bu mefhumun yeteri kadar matematikselleştirilemeyeceğinin bir göstergesi olarak algılanmıştır.

Bu nedenle diferansiyel hesap teorisinin (ya da sonsuz küçükler hesabının), sonsuz için bir belirlenim ve sınır getirmesi, yeteri kadar matematiksel görülmemiştir. Oysa Leibniz'e göre, sonsuz küçük niceliklerin keşfi, oldukça sağlam temelli kurgulara sahiptir. Bu ifadenin “anlaşılması gereken anlama gelince; Leibniz, sonsuzların ve sonsuz küçüklerin, geometri alanında ve hatta doğada tamamen gerçekmiş gibi ele alınabilecek şekilde kurulduğunu deklare etmiştir” (Guénon, 2004: 36). Çünkü Leibniz'e göre doğada bulunan tüm şeyler, Tanrı'dan nüveler taşır ve bu nüvelere ulaşmamızı sağlayan, doğa bilimlerine uygulanacak matematik gibi bir bilim ve onun bir aygıtı olan *infinitesimal calculus* aracılığıyla bu nüveler nihayet seçik olarak bilinme imkanına kavuşur.

Yaptığı felsefi ve metafizik açıklamalarının ötesinde Leibniz, “Letter To Varignon, With A Note On The 'Justification Of The Infinitesimal Calculus By That Of Ordinary Algebra” [Olağan Cebir Aracılığıyla Sonsuz Küçükler Hesabının Doğrulanması (1702)] adlı mektubunda, buluşunun metafiziksel menşeli olmasının dışında ayrıca fizik dünyada kullanılabilirliğini, cebirsel işlemlere dayandırmak suretiyle bir ispat niteliğinde verme girişiminde (1969j: 542-46) bulunmuştur. Tüm bunların ötesinde diferansiyel hesabın katkılarıyla Leibniz'in zihninde, evrensel dil olarak sunmayı planladığı düşüncenin alfabeti karakteristiklerin kavramları için de bir çözüm belirginleşmeye başlamıştır.

Diferansiyel hesap üzerine yaptığı çalışmalardan sonra hayalini kurduğu ve üzerine çalıştığı evrensel dil için de bir teknik geliştirme çabasında olan Leibniz, yeniden kavram analizine başvurur fakat burada bir sonsuzluk fikri kabul edilemez olduğundan, bileşik kavramlarda, varsayımsal “ilkseller” olarak kabul edebileceğimiz en yalın hallerini buluncaya kadar bir analiz öngörür. Aksi halde, böyle bir yapay dil elde edilemez çünkü nerede duracağı bilinmeyen bir belirsizlik baş gösterir. Leibniz, Dalgarno ve Wilkins'le töz ve ilinekler bağlamında aynı soruna-Dalgarno'nun kadavra örneğinde belirttiği üzere-takılı kalmaktan korktuğu için bu belirsizlikten bilinçli şekilde kaçınmıştır.

Characteristica universalis üzerine yaptığı çalışmaların incelenmesi neticesinde Leibniz'in, yapay evrensel dili için kavram analizine götürmeyi başaracak düzeyde bir geçerliliği olduğu kadar; bir mefhumun açık seçik bilgisine ulaşmak için de kullandığı “yeter neden ilkesi”ni, varoluş hakikatleri üzerinde işleterek, böylece bir seçik bilgi edinme yolu geliştirdiği açıkça görülmektedir.

Descartes, soyut kavramlara mahsus geometrinin kartezyen bir sistemde cebirsel ifadelerle dönüşerek somuta indirgenmesi yoluyla elde ettiği analitik geometrinin bir yöntemsel benzerliğini felsefede açık seçik bilgiye ulaşmaya çalışırken de uygulamaya koymuştur. Daha önce tezimizin 1.3.3. alt başlığında vurgulamaya çalıştığımız üzere, açık

seçik bilgi için “Descartes açıklığı, hakikatin alameti olarak görüyordu ve seçikliği de açıklık ile aynı şey olarak mütalaa etmemekle birlikte, en azından ona doğal olarak eşlik eden bir şey sayıyordu” (Boutroux, 2017: 29). Oysa Leibniz’e göre, Descartes’ın yöntemi epistemeye ulaşmak için yeterli değildi. Leibniz, “Hukuk Bilgisini Öğrenmek ve Öğretmek İçin Yeni Bir Yöntem” (1667) isimli çalışmasının I.25. bölümünde, felsefesinde ciddi etkiye sahip Descartes’a gönderme yapmakla kalmaz ayrıca eleştirilerini de sunar:

Analitik veya yargı sanatı bana neredeyse tümüyle iki kurala indirgenebilir görünmektedir: (1) Hiçbir kelime, açıklaması yapılamadan kabul edilmemelidir ve (2) kanıtlanmadan, hiçbir önerme kabul edilmemelidir. Bunların *First Philosophy*’deki dört kartezyen kuraldan ilki olan, açık ve seçik bir şekilde algıladığım her şeyin doğru olduğu kuralından çok daha mutlak olduğuna inanıyorum. Bu kural bir sona vardırması açısından aldatıcıdır (1969b: 88).¹⁹

Descartes’ın bir tilmizi olarak başladığı felsefesinde Leibniz, kavramlar arasındaki *nelikte*, açık seçik bilgiye Descartes’dan farklı şekilde anlam yükleyerek, açıklık ve seçiklik olarak bilginin iki ayrı şekilde temsil edildiğini göstermeye çalışmıştır. Descartes’a göre (biraz da sezgisel olarak) açık ve seçik olarak bilinen şeyin ispata ihtiyacı yoktu.

Cebir, özellikle bazı içeriklerden kurtulma meselesidir. Bu nedenle, Descartes’ın keşfi sayesinde geometrik kanıt tamamen biçimsel olarak düşünülebilir. Leibniz, Descartes’ın kısa sürede durduğunu ve ispatların yürütülebileceği tamamen genel bir soyut evrensel özelliğe, daha yüksek bir bilime giden yolu göremediğini düşünmektedir (Hacking, 1973: 177).

Descartes’ın biraz da sezgisel olarak açık ve seçik bilginin imkanını ortaya koyma çabasının belki de en önemli nedeni, Descartes’ın ispatı bilmiyor olmasıydı fakat Leibniz ispatı biliyordu (Hacking, 1973: 175). Bu nedenle Leibniz, Descartes’ın bilgi için sunduğu yöntemde eksiklik olduğunu düşünmekteydi.

Felsefesinin ilk olgun örneği sayılan “Meditations on Knowledge, Truth, and Ideas” [Bilgi, Hakikat ve İdealar Üzerine Meditasyonlar (1684)] başlıklı risalesinde Descartes ve kartezyen mantığa ciddi eleştirilerde bulunan Leibniz’e göre, herhangi bir ispata gerek kalmadan açıkça bildiğimizden emin olduğumuz, örneğin renk bilgisinin *neliği* bulanık olabilir çünkü bir köre kırmızı anlatılamaz (1969h: 291; 1989d :24). Risalesinde özetle, bilgiyi temelde müphem veya açık, açık bilgiyi seçik veya bulanık; seçik bilgiyi eksik veya upuygun ve hatta sembolik veya görüsel olarak ayırmıştır. Sembolleştirmenin aktif uygulandığı *kör düşünce* gerektiren matematik ve aslında düşünmenin kendi faaliyet alanında

¹⁹ Leibniz burada *First Philosophy* derken, aslında Descartes’ın *Yöntem Üzerine Konuşma (Discourse On The Method)* adlı yapıtını kastetmiştir.

ve yargıda bulunduğumuz her anda, elimizdeki doğasını düşünerek yorum yapmak yerine, anlamlar müphem bir şekilde idealarımızın yerine geçmektedir.

Leibniz'e göre (1969h: 292) upuygun bilgiye sahip olduğumuzda, bir imkanın *a priori* bilgisine sahibizdir, sonlu analizini yapabildiğimiz ve çelişki doğmadığı sürece kavram açıkça mümkündür. Leibniz'e göre açıkça bildiğimiz bir bilgi türünden hareketle, böylece bilgilerimiz ispatlanmaya açık kılınabilir. Keza, açık bilgi olarak öz hakikatlerinden sayılan matematik hakikatleri-sonsuz ispat yalnız Tanrı müdrikesinde mümkün iken-sonlu bir işlemler sıralamasıyla ispatlanabilir ve varoluş hakikatleri de sonsuz analizin bir sınırlamasıyla belirginliğe kavuşabilir.

Açık [*claire*] olan, bir şeyi yekdireğinden ayırt etmeye imkan veren bilgidir; seçik [*distincte*] olan ise bir şeyin detaylarıyla bilinmesidir. O halde, bir mefhum seçik olmadan da pekala açık olabilir. (...) Cebirde kullanılan işaretler, açık mefhumlar olduğu halde, temsil ettikleri şeylerin görüsüne [*intuition*] sahip olmadığımız için seçik değildir. Öyleyse, Descartes'ın birbirinden ayırmadığı iki bilgi türü söz konusudur: Sahiden de seçik tek bilgi olan görüsel bilgi ve açık ama bulanık olan sembolik ya da kör bilgi. Leibniz (...) görü ile basbayağı kavram arasındaki farkı ortaya koyar; bu farklılıktan da çok ciddi sonuçlar çıkartır: Açıklık, mantıki imkanın delili olabilir fakat mevcudiyetin imkanının sadece açık olmakla kalmayıp aynı zamanda seçik bir mefhum da olan, yani birbiriyle bağdaşan öğeleri içeren bu imkanın delili değildir (Boutroux, 2017: 29-30).

Seçik bilginin imkanında “yeter neden ilkesi” yürütülerek sonsuz kavram analizinin “ayırt edilemezlerin özdeşliği ilkesi”ni doğurduğu ve bu ilkenin yürütülmesinden, tözlerin tekilliklerinin sonsuz küçükler hesabına dayandırılarak bulunması neticesinde Leibniz, sonradan *monad* olarak adlandırdığı tözlerin her birinin sonsuz analiz sonucunda bütün dünyayı barındırdığını iddia etmiştir.

Metafiziğinin en olgun seviyesinde kendi mefhumları ışığında her bireyin yani *tözün*, dünyayı kendi penceresinden gördüğü şekliyle yansıttığını, böylece her öznenin dünyayı kendine sunulan bakış açısı kadar tanımlayabileceğini serimlemek üzere Leibniz, yeniden açık seçik bilginin imkanını ele alır. Euclides'in *Elementler*'i üzerine yaptığı okumalar sonucunda Leibniz, genel karakteristiklerin geometriye uygulanmasına duyduğu ilgi neticesinde 1679'da Huygens'e yazdığı “Studies In A Geometry Of Situation With A Letter To Christian Huygens” [Christian Huygens'e Bir Mektup ile Birlikte Konum Geometrisi Üzerine Çalışmalar] isimli mektubunda, “Analysis Situs”un²⁰ (1679) yazarı olarak büyük ilerleme kazandırmış olduğu

²⁰Bu risale, Einstein'ın izafiyet teorisinin kökenlerini Leibniz'de bulma çabalarında sıklıkla başvurulan bir risaledir. “Leibniz'in böyle bir geometri bulma çabaları, on dokuzuncu yüzyılda Riemann ve Grassmann ilgili çalışmaları üstleninceye kadar bir yanıt bulamamıştır” (Loemker'in mektup için önsözü, 1969f: 248).

geometrinin bu alanında, geometrideki şekilsel büyüklüklerden ziyade konum analizine vurgu yaparak ve yansımalar geometrisiyle ilişki kurarak, bir yandan *monad*ların birbirine göre durumunu izah etmeye çalışırken öte yandan bir perspektif felsefesinin de temellerini atmıştır.

Leibniz'e göre hepimiz, bakış açımıza göre belirlenim kazanmış birer tözüz. Dolayısıyla, “öznenin bir tür özelliği, özlülüğü, nesneliği vardır ve bu nesnelik işte yine bir bakış açısıdır” (Deleuze, 2007: 38). Belirlenmiş bu perspektif sayesinde böylece, kendi mefhumumuzda (nosyon) bizim için açık ve seçik olan bir bilgi başka bir özne tarafından bulanık veya belirsiz olabilir.

Öyleyse bir bakış açısını diğerinden ayırt eden nedir? Öte yandan açık ve seçik bir şekilde ifade ettiğim küçük bir parçası var dünyanın ve her öznenin, her bireyin kendine ait minik bir açık seçik payı var dünyadan-acaba hangi anlamda? Şu anlamda ki benim açık ve belirgin bir şekilde ifade ettiğim dünya parçası başka bütün öznelere tarafından da karanlık ve belirsiz bir biçimde ifade edilmektedir. Benim bakış açımı tanımlayan şey bir nevi projektör olmalıdır-o projektör, karanlık ve belirsiz, karmaşık dünyanın uğultusu içinde kısıtlı, sınırlı bir ifade bölgesini aydınlık, açık ve belirgin tutmaktadır (Deleuze, 2007: 39).

Peki ne zaman bulanık bir algı, seçik bir bilgiye dönüşmektedir? Leibniz, bu durumu sonsuz küçükler hesabından hareket ederek, ileride psikoloji için kullanılacak bir kavram tercihiyle ifade eder: Tamalgı. Ona göre bulanık olan bir bilgi küçük algılar halinde zamansal akışa tabi bir süreklilik durumunda, belli bir farklılaşmaya kadar bulanıklığını korur ve fakat bir tekilliğe ulaştığı anda algımız için bulanık olmaktan çıkarak bir tamalgıya dönüşür. İşte bilgi, seçik olma halini burada kazanır. İlk defa, 1696'da Fardella'ya yazdığı mektubunda, *töz/ özne* yerine *monad*²¹ kavramını kullanan Leibniz'in 1714 yılında maddeler halinde kaleme aldığı *Monadoloji*'sinin üçüncü maddesinde Leibniz, *monad*ların, doğanın gerçek atomları olduğunu yani onların, şeylerin doğası olduğunu vurgular (2011: 98). *Monad*lar birbirlerine indirgenemeyen bireysel tözlerdir. Yitip giden farklar nedeniyle her bir *monad*, kendinde şeye dönüşür. Başka türlü bir ifadeyle, “ayırt edilemezlerin özdeşliği ilkesi” temelinde, *monad*ların tekilliğine yapılan gönderme ile *monad*ların farklılıkları ortaya konmuş olur. Leibniz, *Monadoloji*'de şöyle devam eder:

Her *monad*, diğer her birinden farklı olmalıdır; zira tabiatta, biri diğeriyle mükemmelen benzer olan ve aralarında içsel [interne] bir fark ya da meşrep [dénomination] temeline dayalı bir fark bulmak mümkün

²¹ Leibniz, *monad* terimine benzer bir terim olarak *monadik* terimini 1666 tarihinde *Tez*'inde kullanmış olup, “daha önce 1663'te öğretmeni Jacop Thomasiaus, Leibniz'in ilk felsefi eseri olan “Disputatio Metaphysica de Principio Individui” adlı çalışmasına yazdığı önsözde başlı başına bir türü içeren bir bireyi ifade etmek için 'monadik' terimini kullanmıştı” (Antognazza, 2013: 294).

olmayan iki varlık asla yoktur. (...) Dışsal bir neden *monadı* içinden etkilemeyeceğinden, söylediklerimizden çıkacak sonuç, *monadların* tabii değişimlerinin içsel bir ilkedен ileri geldiğidir. (...) “Birlik”te ya da yalın cevherde bir çokluk ihtiva veya temsil eden geçici hal, devamında ortaya koyacağımız gibi, “farkındalık” [aperception] yahut bilinçten ayırt edilmesi gereken, algı diye adlandırılardan başka bir şey değildir. (...) *Monadlar* konu/obje bakımından değil, konunun/objenin bilgisi bakımından sınırlandırılmıştır. Hepsi de bulanık şekilde sonsuza, bütüne uzanır; ama seçik algılarındaki dereceler itibariyle sınırlandırılmış ve birbirlerinden ayırt edilmiştir (Leibniz, 2011: 99, 100, 112 italik vurgular bana ait).

Bu durumda, dışarıdan tecrübe ettiğimiz bulanık algılarımız olmadan ve bulanık algıların sıralı tekillikleri gerçekleşmeden, bir seçik bilgiye ulaşmak mümkün değildir. “Böylece Descartes’ın aksine Leibniz, bulanık düşünceye sevk eden tecrübeye, seçik düşüncenin ve rasyonel bilginin zorunlu temelini görür” (Boutroux, 2017: 86). Öyle ise her *monad*, diğer *monadların* seçik algılarının bulanıklığını tecrübe ederek bir arada olmaklık icabını gerçekleştirmektedir. Yani varoluş için tözün tekilliği yetmediği gibi diğer tözlerle birlikteliği de gerekir. Bu durumda, bir arada mümkün olmanın ne demek olduğunu sorgulamak gerekir.

Leibniz, *Metafizik Üzerine Konuşma* (1686) adlı yapıtında bir örnek vermektedir: Bu örnekte Sezar’ın Rubicon nehrini geçen bir diktatör olmasının, mümkün olduğunu ve fakat bu durumun Rubicon ırmağını geçmemiş bir Sezar’ın da mümkün olmasını engellemediğini (2011: 53) ifade eder ancak, Sezar’ın Rubicon ırmağını geçmediği bir dünya bu dünya olamazdı. Varoluş hakikatlerinin olumsuzluğuna dayandırdığı bu savdan hareketle Leibniz, bugün içinde yaşadığımız dünya açısından, tüm mümkün dünyalar arasında en olumsuz olanın, Sezar’ın Rubicon ırmağını geçtiği dünya olduğunu (2011: 53) açığa kavuşturmuş olur.

Mümkünleri ortaya koyan da onlar arasında bir seçim yapıp gerçeklikleri oluşturan da Tanrı’nın kendisidir. Ancak ilkinde Tanrı’nın akli söz konusu iken, ikincisinde ise Tanrı’nın iradesi etkin olmaktadır. Bir arada mümkün olanların seçildiği bir dünyada, bazı şeylerin seçilip bazılarının reddedilmesi iradenin gerçekleştirmesi gereken bir eylemdir (Kaya, 2014a: 178).

Leibniz’in yaşadığı dönemde yayımlanmasına izin verdiği tek kitabı olan *Theodicy* (1710), Tanrı’nın irade gerektiren eylemi olarak da görülebilecek bir arada mümkün olmanın azami derecesini, ahlaki olarak değil, bir olasılık felsefesi gereği serimleme girişimi (Deleuze, 2007: 57) olarak görülebilir. Bu eserde Leibniz, bir arada mümkün olmaklığı ancak ve ancak Tanrı’nın mümkün dünyaların en iyisini olasılıkla hesaplamış olmasına bağlamaktadır.

Leibniz'in kötülük hakkında yaptığı, metafizik, fizik ve ahlakî kötülük ayrımı, her şekliyle kötülüğün sadece yoksunluktan kaynaklandığını gösterir. Anlaşıyor ki Leibniz için öyle ya da böyle bu dünyada kötülük olmak zorundadır. Ama biz yine de Leibniz'e şunu sorabiliriz: Madem ki bu dünya, mümkün dünyaların en iyisidir, o halde bazı kötülüklerin bulunduğu bu dünya niçin yaratıldı? Leibniz'in bu soruya yanıtı kısadır. Varoluş olmayıştan daha iyidir. Başka bir ifadeyle varlık yokluktan iyidir (Kaya, 2014a: 255).

Bu dünyada Âdem'in günah işlemesi, Sezar'ın Rubicon'u geçmesi gibi varoluşsal durumlar, bir arada mümkün olmak için ve hatta var olmayı deneyimlemek için gereklidir. Kötülük probleminde bir yanıt bulmaya çalıştığı *Theodicy*'de Leibniz, bir "mümkün dünyaların en iyisi" olması gerekliliğini mitolojik bir esinle Jüpiter'in kızı Athena'nın tapınağında uykuya daldığı sırada, Theodorus'un gördüğü bir rüyayla²² örneklendirerek serimlemektedir (2007a: 375). Her katta bulunan mümkün bir dünyanın varlığına tanıklık ederken, bu mümkün dünyalar arasında en güzelinin içinde bulunduğumuz dünya, varoluşa geçebilecek bir arada mümkün en fazla sayıda tözün bulunduğu dünya olarak görmüştür. Çünkü bu dünya, sürekliliklerin bozulmadığı yani bir yüklemden diğerine geçişin süreklilikle sürdürülebildiği haliyle mümkündür. Daha açık ifade etmek gerekirse, "öz hakikatlerini yöneten özdeşliktir, varoluş hakikatlerini düzenleyen ise süreklilik" (Deleuze, 2007: 64) yasasıdır.

Süreklilik yasası hem yerden yere, hem formdan forma hem de durumdan duruma geçişlerde görülür. Değişimler sürekli olmaktadır ama görünüşte sıçrama gibi algılanmaktadır. Sonsuz küçüklükteki değişim basamaklarını göremediğimiz için, sanki doğada süreksizlik varmış gibi düşünürüz. Cismin parçaları bir süreklilik oluşturur ve bütün dünya bir sürekliliktir (Kaya, 2014b: 59).

Yüklemler arası geçişte bir süreklilik olması gerektiği fikrinin hem *Metafizik Üzerine Konuşma* hem de *Monadoloji*'sinde, Leibniz'in, metafizik ögeler üzerinden yükselirken her şeyin her şeyle bağlantılı olduğuna dair görüşüyle doğrulandığını gözlemlemekteyiz. Bu iki çalışmada da yine Leibniz'in Descartes'ın epistemolojisine açık eleştirileri mevcuttur. Leibniz'e göre, açık seçik bilginin imkanında, Kartezyencilerin ikincil nitelikler olarak kabul ettikleri renk, koku, tat gibi algılarımıza bağlı olan temsillerin yanında, birincil nitelikler olarak belirlenen şekil, büyüklük, hareket gibi unsurlar da aslında algımıza bağlıdır (2011:

²² Bu rüyada, Tanrıçanın Theodorus'un yüzüne bir zeytin dalıyla dokunması üzerine, Theodorus farklı bir ülkenin sarayında gözlerini açar. Bu sarayda birçok salon vardır (Leibniz, 2007a: 375) ve Tanrıçanın sarayın bu salonlarından birine götürdüğü Theodorus, buranın artık bir salon değil bir dünya olduğunu görür. Bu dünyada Sextus'un tüm hayatı bir bakışta gözlerinin önüne serilir. Bu hayatta bağlantılı olan diğer dünya halleri gibi. Diğer bir salona geçtiklerinde başka bir dünya; başka bir Sextus görür. Burada da Sextus bilindiği halinden başka bir hayat sürmektedir (2007a: 376). Salonlar, bir piramid üzerinde yükselmektedir. Nihayet piramidin en yüksekinde şu ana kadar gördüğü dünyalar arasında en güzeline ulaşır. Bu dünya piramidin en tepesindedir ancak bu öyle bir piramittir ki tepesi görünürken, tabanı görünmemektedir. Sonsuza doğru bir uzanımdadır (2007a: 377).

50). Öyle ise tüm bu birincil ve ikincil nitelikler, seçik olarak değil, ancak bulanık olarak küçük algılar halinde bulunmaktadır.

Leibniz'e göre küçük algıları gözden kaçıran Descartes ve Kartezyenciler gibi "Locke'un yaklaşımı göz önünde bulundurulduğunda, "bilincinde olmadığımız algıları" tamamen dışarıda bıraktığı aşikardır" (Altınörs, 2016: 75). Bilindiği üzere, Locke'a göre algıladığımız (perception) haliyle bir dış duyum (sensation) nesnesinin ancak zihinsel bir aktivite olarak (refleksiyon) zihnin iç duyum nesnesine dönüştürülmesi, bilginin imkanını sağlar. Bilgimiz ancak bu yolla elde edilir.

Böylece denilebilir ki Locke'a göre bilgi, her şeyin (kendi terminolojisinde *idenin*) deneyimden elde edildiği ve aktarıldığı haliyle doğuştan boş olan zihnin bir yeteneğinin sonucunda elde edilebilir. Leibniz bu fikre katılmadığı gibi daha önce değindiğimiz üzere, Locke'un *Deneme*'sinde yazan her fikre özel olarak eğilmek suretiyle, *Yeni Denemeler*²³ adlı eserinde Locke ile diyalog tadında bir tartışma sürdürmektedir.

Yeni Denemeler'in önsözünden hareketle, Leibniz'in eserini kaleme alma niyetinin empirist felsefenin Locke'un etkisiyle yayılmasının önüne geçmek olduğu da söylenebilir. Bu eserin önsözünden anlaşıldığı üzere Locke'un, doğuştan üzerinde hiçbir şeyin yazılı olmadığı boş bir sayfa olan zihnimize, bilgiyi ancak ve ancak duyu ve deneyimden ileri gelen ideler aracılığıyla aldığımızı dair Aristotelian *tabula rasa* görüşüne karşı, kendini Platonist bir bakış açısıyla tanımlayan Leibniz'in, zihnin, doğası gereği nosyon ve ilkelerin kaynağına sahip olduğu, dolayısıyla bilginin kaynağının tek rolünün, nosyon ve ilkeleri durumuna uygun canlandırmak olan dış duyum nesnelere ileri gelmediğini (Leibniz, 1996: 49-50) ispatlama çabası mevcuttur.

Leibniz'e göre, "daha önce duyuların içinde olmayan hiçbir şey zihnin içine giremez, zihnin kendisi hariç" (1996: II. I. §2. 99; Antognazza, 2013: 87; Domingos, 2017: 99). Leibniz, *monadların penceresizliği* ve kendine içkinliği fikrinin esinlediği şekilde Locke'a, zihnin deneyime ihtiyacı olduğu konusunda katılır fakat yine de zihnin Locke'un iddia ettiği üzere bilgiyi dışarıdan alan pasif bir yapı olduğu fikrine katılmaz. Leibniz'e göre zihni pasif bir alıcı olarak görmek yanıltıcıdır (1996: II. I. §2. 99) çünkü zihin (akıllı ruh), oluş, töz, birlik, aynılık, neden, algı, muhakeme ve benzeri nosyonlara sahiptir ki bunlar deneyimden elde edilemez.

²³ Leibniz, Platon'un diyalogları tadında bir eser olarak *New Essays Concerning Human Understanding* (1704) adlı kitabında, Locke'un eserinin açık bir eleştirisi olarak, Locke'un kitap bölümleri ve başlıklarına riayet edilmek suretiyle Locke'un görüşlerini temsilen *Philalèthe*, kendi görüşlerini temsilen *Théophile* isimlerini kullanmıştır.

Leibniz'e göre, "Locke'un epistemolojisinin eksikleri, matematiği ve onun kanıtlarının doğasını yeterli ölçüde anlamamış olmasından" (Antognazza, 2013: 348) kaynaklanmaktadır. Leibniz'e göre, seçik bilgiye götüren matematiksel bir analiz sonucunda edindiğimiz bilgilerin çoğu, zihinde bulunan idealarımızın refleksiyon sürecine katılmasıyla mümkündür. Bu durumda, matematiksel ve mantıksal *a priori* ilkeler, "görüsel bilgi" içerikleridir. Leibniz, zihinde doğuştan bulunduğunu iddia ettiği görüsel bilgilerimizin dışında farazi bilgimizi, "bir tür ilksellere götüren kavram analizi yaptığımız sırada, "sevke tabi" [*discursive*] düşünceyle, yani muhakemeyle" (Altınörs, 2016: 82) edindiğimizi iddia eder. Leibniz'in "düşünmenin bir çeşit hesap yani aritmetik kalkülüs" (Mugnai, 2010: 297) olduğuna dair görüşüne daha önce tezimizde yer vermiştik. Bu durumda, Leibniz terminolojisine göre muhakeme, aritmetik kalkülüstür.

Leibniz'e göre, muhakeme ederken faydalandığımız *kör düşünce* sayesinde zihnimiz büyük bir yükten kurtularak kavramsal düşünceyle, bilgiyi edinme işleyişini kolaylaştırır. "Leibniz'in yaklaşımının göstermeye çalıştığı şey, kavramsal düşüncenin, öncelikle müdrikenin "içsel düzenini" sağlayacak halisane mantıksal sınırların çizilmesiyle mümkün olabileceğidir (...) bu düzen akıllı ruhlarda müşterektir" (Altınörs, 2016: 94).

Öyle ise seçik bilgiyi elde etmek için uyguladığımız kavram analizinde, zihnin *kör düşünceyi* yürüterek yaptığı muhakemede, kavramların *neliklerinden* hareket ettiğimiz söylenebilir. Çünkü kavramların anlamları, içerilme ve dışlanma yoluyla zaman içinde değişime uğrayabilir. Locke gibi Leibniz de zihnimizde yansımaları olan idelerin veya kavramların işaretleri olarak gördükleri kelimelere, özel olarak eserlerinin üçüncü kitabında değinmiştir. Elbette birbirlerinden farklı şekillerde açıklamalar yaparak. Locke'un gözünden bakıldığında kelimelerimiz, farklı zihinlerin birbirleriyle bilgi alışverişinde idelerimizin açıklanmasında birincil esas iken; Leibniz için kelimeler muhakememiz açısından hafızamızda saklanmaları gereğince bir önceliğe sahip olsa da muhatabımız için sadece işaretlerdir (1996: III. IX. §3. 284). Böyle bakıldıkta, Locke için bir kelimenin doğru kullanımı, anlamı bakımından elzendir. Leibniz'e göre ise bir kelimenin işaret ettiği kavramın doğru anlamını vermesi, kelimenin ve temsil ettiği kavramın anlamının, zaman içinde bazı tanımlarını kaybetmesi veya yeni tanımları bünyesine katması hasebiyle değişebileceğidir.

Gerçekten de Leibniz'in kendi yaşadıkları dönemde, altının en ağır metal olarak tanımlanmasına rağmen, ileride altının belki de en ağır metal olma tanımına uymayacağına (1996: III. VI. §17. 264) dair örneği, altın üzerine verdiği tahminin gerçekleşmesi bakımından ilginçtir. Dolayısıyla daha önce de değindiğimiz üzere, seçik bilginin edinilmesinde Leibniz

için asıl önemli olan ve kabul edilmesi gereken şey, kavramların *nelikleri* ve aralarındaki *bağlantısallık*tır.

Locke kendi eseri *Deneme*'de, özellikle üzerinde durduğu altın örneğini ele alarak, “altın” derken, her bireyin zihninde yer eden altın idesinin farklı olduğu (1999: III. II. 3. 17), bu idenin, dolayısıyla da kelimenin anlamının kişinin deneyimine paralel olarak değişiklik gösterdiğini, bu nedenle kelimelerin anlamlarının belirsizleştiği için doğal dillerin, doğru anlamlara sahip kelimelerce ikame ettirilmesi gerektiğini savunur.

Locke'a göre kelimelerin müphemliğinin bir diğer nedeni ise kelimenin, çoğunlukla anlamının üzerinde durulmadan sadece bir ses, bir eklemli yapı olarak konuşmaya dahil edilmesidir. Locke'a göre insanlar, konuşurken çoğu durumda tıpkı bir papağan gibi kelimeyi doğru ya da yanlış kullanıp kullanmadıklarına önem göstermeden (1999: III. II. 7. 19), hafızada bulundukları şekliyle konuşmaya dahil ederler. Böylece, hafızaya alındığı şekliyle kullanılan kelime, çoğunlukla yanlış bir anlamda konuşmaya dahil edilmiş olur.

Locke, bu yanlış kullanımlarla mücadele etmek için normlar belirler; bu normların felsefi diller izleğiyle hiçbir ilgisi yoktur çünkü Locke'u ilgilendiren yeni sözlüksel ve sözdizimsel yapılar belirlemekten çok, doğal dilin sürekli denetlenmesini sağlayan bir tür felsefi sağduyu önermektir. Locke, dil sisteminin reformunu değil, iletişim sürecinin sıkı bir denetimini düşünür (Eco, 1995a, 286-7).

Bir Âdemi dil fikrinin asla oluşmadığı Locke'un, kelimelerin doğru kullanımlarının etimolojik bir yaklaşımla sağlanabileceği fikrine karşın, Leibniz'e göre kelimedenden hareketle etimolojik bir inceleme, kavramın *neliğini* vermeyecektir; ancak ve ancak bir kavram analiziyle kavramlar için “ilkseller” olarak varsayabileceğimiz yalın hallerine ulaşılarak müphemliğin yok edilebilmesi mümkün görünmektedir. “Bir Âdemi dil olsaydı, bu dilin ilk bilgisi, yalnızca sözcüklerin değil, nesnelere de bilgisi olurdu” (Hacking, 1988: 140). Bu durumun imkansızlığından haberdar olarak, yerel dillerin zamanla değişim ve dönüşümü neticesinde bir kavramın kök yani Âdemi dildeki ilkseline asla ulaşamayacağını bilen Leibniz, *Yeni Denemeler*'inde doğal dillerin bir kusuru olarak gördüğü müphemliğin, ancak kavramların sadece standart tanımlarının bulunduğu bir evrensel kavramlar sözlüğünün (1996: III. XI. §25. 302) hayata geçirilmesiyle giderilebileceğini iddia etmektedir.

Leibniz, özellikle sayısı az olduğu için vurgu ve tona dayalı değişim sergileyerek çeşitli anlamlara bürünebilen ve bu yolla konuşmada çeşitlilik sağlayabilen kelimelerin

mensubu olduğu doğal dillerin en çok tercih edileni olan Çinceye²⁴ değinerek argümanını güçlendirmiştir.

Yine de Çinliler gibi sözcüklerini çeşitlendirmek için tonlar ve aksanlar kullanan ve kullanılan sözcük sayısının çok az sayıda olduğu halklar vardır. Diller üzerinde büyük bir otorite olan ünlü matematikçi Golius'un görüşü, bu halkın dilinin yapay olduğu yönündeydi. Yani, Çin dediğimiz büyük toprakları işgal eden birçok farklı halk arasında, sözlü iletişim sağlamak için usta bir adam tarafından bir kerede icat edilmişti. Ancak bu dil de uzun süreli kullanımla değiştirilebilir (1996: III. I. §1. 232).

Çincenin iletişimde vurgu ve tonlamaya dayalı sesletilebilir dil yapısının dışında, “yalnızca Çinlilerin değil, ayrıca Japonların, Korelilerin, Koçinçinlilerin ve Formozalıların da rahatlıkla anladığı” (Eco, 1995a: 162-3) özellikle yazınsal iletişimi kolaylaştıran simgesel yazı dili Leibniz'i oldukça etkilemiş ve esin kaynağı olmuştur. Halklarının sembol manipülasyonu sayesinde ideogramlarla dolu pasigrafik yapıya bürünmüş bu uluslararası evrensel dili, Leibniz'e tamamlamayı umduğu yapay dilinin, nasıl bir görünüm ile sağlam ve sarsılmaz olabileceğine dair kendi karakteristiklerini belirlemesi konusunda yardımcı olmuştur.

Leibniz (1996: III. II. §1. 235), Çinceyi, Dalgarno ve Wilkins'in yapay dillerini, bütünüyle seçilmiş ve tamamen keyfi görünen yapay dillere örnek olarak verir. Ancak bu diller, Leibniz'e göre zaten bilinen dillerden yapıldığını bildiğimiz diller olup, üzerine inşa edildikleri dillerin seçilmiş özellikleri ile doğal ve tesadüfi özelliklerinin bir karışımını içerir (1996: III. II. §1. 235). Bunun için de Leibniz'e göre, tüm doğal dillerin ötesinde, kavramların neliği ve bağlantısallığının ön planda olduğu yapay evrensel bir dil oluşturulmalıdır.

Leibniz'in (...) nezdinde fenomenal planda farklı yerel dillerdeki doğru hüküm bildiren beyanların ya da farklı yapay dillerdeki sembolik bağıntı ifadelerinin, aslında bir ve aynı hakikate ilişkin kısmi kavrayışlarımızın değişik ifadeleri olduğunu söyleyebiliriz. Şu veya bu işaretler sisteminin; bir şeyi adlandırmada şu veya bu seslerin keyfi olarak seçilmesi, onlar vasıtasıyla icra ettiğimiz muhakemelerle vardığımız hükümlerin keyfiliği sonucunu doğurmaz (Altınörs, 2016: 113).

Locke için eşyanın tabiatını yansıtabilecek mükemmel bir dil ya da evrensel bir dil yoktur. Leibniz'e göre ise bir sembol manipülasyonu ve *kör düşünce* ile hesaba indirgenebilir bir dil, eşyanın tabiatına en yakın ilkselleri verecek şekilde kavramları analize tutulmuş haliyle mükemmel değilse de bir evrensel dil olarak, hizmet edebilir.

²⁴ Leibniz, hayatının son yirmi yılında, masasında duran tüm diğer çalışmalarının yanında ayrıca bir filolog, etimolog ve dilbilimci olarak yerel ve ulusal diller üzerine çok ciddi çalışmalar yaparak günümüz dilbilimcilerine azımsanmayacak bir literatür bırakmıştır. Bu diller arasında en çok ilgi duyduğu dil, Çince olmuştur. Çin'de misyonerlik yapan Avrupalı Cizvitler sayesinde, Çin kültürü ve Çince ile ilgili kaynaklara ilk elden ulaşabilen Leibniz, 1697 tarihinde “Çin'deki Cizvit misyonerlerinin mektuplarını ve denemelerini bir araya getiren *Novissima Sinica*'yı” (Eco, 1995a: 282) yayımlamıştır.

Leibniz'in yaşamı boyunca geliştirmeyi sürdürdüğü projeyi betimlemek istersek, hiç kuşkusuz dört temel noktayı göz önünde bulunduran uçsuz bucaksız bir felsefî-dilsel binadan söz etmemiz gerekir: (i) bir düşünce alfabesi ya da bir genel ansiklopedi biçiminde düzenlenmiş bir ilkseller sisteminin belirlenmesi; (ii) olasılıkla Dalgarno'nun önerdiği dilbilgisel yalınlaştırmaların Leibniz'e esinlediği ideal bir dilbilgisinin geliştirilmesi (Leibniz'in yalınlaştırılmış Latince'si bunun bir örneğini oluşturur); (iii) daha sonra, karakterlerin sesletilebilir olması için bir dizi kural; (iv) gerçek karakterlerden oluşan bir sözlüğün oluşturulması; konuşucu bu karakterler üzerinde kendisini otomatik olarak doğru önermeler formüle etmeye götürebilecek bir hesaplama yapabilmelidir (Eco, 1995a: 268).

Leibniz'e göre bu dil, bir *kavram göstergeleri dili* olarak, doğru bir zeminde kurulacak iletişimde, sentaktik bir kurulumla hakikati serimlemede, yargı ve keşifte bulunmada kullanılabilecek yegane dildir. “Leibniz, elbette, felsefî dilin değerinin, onun terimlerinin değil, biçimsel yapısının bir sonucu olması gerektiğini ve kendisinin habitudo ya da önerme yapısı adını verdiği sözdiziminin anlamdan daha önemli olduğunu gören ilk kişidir” (Eco, 1995a: 281). Nihayetinde Leibniz'in hayal ettiği yapay dili, konuşulabilir olmayı hedeflese de yazılabilir olarak yaşamaya devam eden *pasigrafik* bir dildir.

Bir evrensel dil için gerekli her şart, Leibniz'in düşünsel ve pratik bağlamda ciddi çabaları sonucu belirginleşmesine rağmen, bazı şartların (kavramlar sözlüğünün oluşturulması ve bunlara atanacak karakteristiklerin belirlenmesi gibi şartlar) yerine getirilememiş olmasından dolayı, Leibniz için bu çalışma, hayatı boyunca bir hayal olarak kalmaya devam etmiştir. “Aslında, Leibniz'in gerçek katkısı, öyle görünüyor ki projesinin dördüncü maddesinde ve sonunda öteki üçünü gerçekleştirmeye yönelik her tür çabayı bir yana bırakmış olmasında yatar” (Eco, 1995a: 268).

Bu bağlamda, daha önce tezimizin 1.4.3. bölümünde değindiğimiz, Leibniz'in bütün bir sisteminin çekirdeği olan evrensel bilimler ansiklopedisi ve kavramlar sözlüğü ile bunların seçik bilgisine ulaştıracak bir yapay evrensel dilin gerekliliğiyle 1679 yılında kaleme aldığı risalelerini ve yöntemsel çalışmalarını, bir sonraki bölümde ele alarak serimlemeye çalışacağız.

1.5. *Characteristica Universalis ve Calculus Ratiocinator*

Leibniz, “On Universal Synthesis and Analysis, or the Art of Discovery and Judgement” [Evrensel Sentez ve Analiz veya Keşif ve Yargılama Sanatı Üzerine (1679)] adlı risalesinde, evrensel dil hayalinin bir özelliği olarak belirlediği keşif mantığı için uyguladığı kombinasyon sanatını açıklarken, özellikle *Tez*'inde (1666) verdiği örneklerin bir genellemesini yapmıştır. Leibniz, bu dilin bir düşünce alfabesine dayanarak bir *şeye/töze/özneye* ait her mefhum için kavram analizi ve bir yüklem sentezini gerektirdiğini

de (1969d: 229) eklemektedir. Bu türden radikal bir yaklaşıma, Leibniz'in yazdıklarından görüleceği üzere, çocuk yaşlarda sahip olduğu matematik, mantık bilgisi ve derin düşünme edimi neticesinde ulaştığı açıktır. Leibniz'in kendi ifadesiyle:

Mantığı henüz çocuk yaşta iken öğrendim ve düşündüğüm şeyin nedenlerini derinlemesine araştırma alışkanlığı edindim. Aşağıdaki soruyu öğretmenlerime sordum. Kavramların sıralandığı basit terimler için kategoriler olduğuna göre neden doğruların sıralanabileceği karmaşık terimler için de kategoriler olmasın? O zamanlar geometricilerin önermeleri birbirlerine bağımlılıklarına göre gösterip sıraladıklarında tam da bu şeyi yaptıklarının farkında değildim (1969d: 229).

Uzun yıllar sonra Leibniz, geometri alanında yeni geometrilere kapı aralayabilecek kadar iyi bir geometrici olması ve dolayısıyla algoritmalarla yürütme yapabildiği ispat yöntemlerini çok iyi kullanabiliyor olması sonucunda, aynı etkinliğin türler ve cinsler arasında da yürütülebileceğini, adı geçen risalede iddia etmiştir.

Ancak bana öyle geldi ki ilk önce basit terimler için doğru kategorilere sahip olsaydık ve eğer bunları elde etmek için, a, b, c, d, e, f gibi en yüksek cinslerin veya en yüksek olduğunu varsaydığımız türlerin kombinasyonundan aşağı kavramların oluşturulabileceği bir kataloğun veya bir düşünce alfabesinin doğasında yeni bir şey kursaydık, bu evrensel olarak başarılı olacak. Çünkü cinslerin birbirinden ayrılaşma işlevi görebileceğini, böylece her farkın bir cins ve her cinsin bir fark olarak kavranabileceğini belirtmeliyiz (Leibniz, 1969d: 229).

Yukarıdaki alıntıdan anladığımız kadarıyla Leibniz, “fark” derken aslında matematiksel bir kavramı işaret etmektedir. Burada “fark” örneğin, matematiksel bir ifadenin *türevi* alınmak suretiyle yeni bir ifadeye dönüşmesi ile elde edilen ve bir *süreklilik* halinde giderek tükenen bir şeydir. Basit terimlerin kategorik ayrımının yapılabildiğini görünce, aynı kategorik ayrımı bileşik terimler için de önerdiği tezinden sonra Leibniz, adı geçen risalesinde kavram analizini şöyle ifade etmektedir:

Öyleyse, eğer belirli türlerle ilgili kanıtlanabilir önermeler bir düzen içinde sıralanabiliyor veya tüm yüklemeleri listelenebiliyorsa ister daha geniş ister dönüştürülebilir olsun, bu durumda aralarından daha anlamlı olanı seçilebilir. Düşüncenin doğal bir alfabesi yoluyla, y gibi bir türün $abcd$ gibi bileşik bir kavramı olsun. Bu kavramın ikili kombinasyonları ab yerine l ; ac yerine m ; ad yerine n ; bc yerine p ; bd yerine q ; cd yerine r olsun. Üçlü kombinasyonları abc yerine s ; abd yerine v ; acd yerine w ; bcd yerine x olsun. Bunların tümü y 'nin yüklemeleri olacaktır fakat yalnızca y : $ax, bw, cv, ds, lr, mq, np$ terimleriyle dönüştürülebilecektir (1969d: 229).

Bu risalede Leibniz, ergenliğinden hemen sonra kaleme aldığı *Tez*'ini örnek göstererek, Kircher'in uzun zaman önce vaat ettiği aynı adlı yapıtı ortaya çıkmadan önce *Tez*'inde çok daha fazla şey söylemiş olduğunu (1969d: 229-30) iddia etmektedir:

Tezimde tartışılan benzer konuları bulmayı umuyordum ancak daha sonra ortaya çıktığında, yazarın sadece Lullian sanatını veya ona benzer bir şeyi canlandırdığını; tıpkı felsefede reform yapmaya çalışan diğerleri gibi insan düşüncelerinin gerçek analizini hayal bile etmediğini gördüm (Leibniz, 1969d: 230).

Böylece, yukarıda görülen alıntılardan anlaşılacağı üzere, bir türe ait tüm yüklemelerin bir sentezi veya kombinasyonunun önelediği bir kavram analizi sayesinde, bir kavramlar sözlüğü elde edilebilir. Aynı risalede Leibniz, şöyle devam etmektedir.

Kombinasyon veya sentez, pusuladaki uygulamasını düşünmek için manyetik iğne verildiğinde olduğu gibi bir şeyin kullanımını veya uygulamasını keşfetmenin aracıdır. Analiz ise aksine, keşfedilmiş bir şeyin veya sonucu verilmiş bir önermenin araçlarını keşfetmek için en uygun olanıdır” (1969d, 233).

Analiz- sentez yoluyla kavramlar sözlüğü oluşturmak, Leibniz gibi bir dehanın bile tek başına yapabileceği bir şey değildir. Bu sebeptir ki ancak seçilmiş dehaların toplamda beş yıllık çalışmaları sonucunda bir kavramlar sözlüğünün hazırlanabileceğini (1969ç: 224) ifade ederken Leibniz, “Genel Karakteristikler Üzerine” (1679) isimli risalesinin I. kısmında²⁵ bu kavramlar için birer karakteristik sayı verilmesini (1969ç: 224) önerir. Leibniz'e göre hem bir keşif hem de bir yargı sanatının bir arada mümkün olduğu bu dil, iletişim için mükemmel bir araçtır.

Ayrıca risalesinde Leibniz, argümanların sayı yoluyla yargılanabileceği (1969ç: 225) yani bu dilde muhakemelerin cebirsel işlemler yoluyla doğruluk sorgulamasına tabi tutulmasını sağlayacak bir *calculus ratiocinator* de önerir. Leibniz'e göre çoğu kavramın karakteristik sayıları belirlendikten sonra, insanlık yeni bir tür yardımcı araca sahip olacak, öyle bir araç ki zihnin gücünü, optik lenslerin, mikroskopların ve teleskopların görüşümüzü arttırmasından daha fazlasını yapacak (1969ç: 224) bir fayda sağlayacaktır.

Leibniz'in evrensel dil hayalini gerçekleştirmek adına bir ön hazırlık için karakteristikler üzerinde yaptığı tamamlanmamış bir diğer çalışması “Nümerik Karakteristiklere Örnekler” (1679) isimli risalesinde, Leibniz, *characteristica universalis* hakkında bazı örnekler ve genellemeler üzerinden detay vermektedir. Aristoteles mantığını çalışırken, “mantığın ve tasım teorisinin evrensel bir matematik olması, yani Leibniz'in onda

²⁵ Başka bir kaynakta, “On The General Characteristic” (1679) isimli risalesinin sadece I. kısmı “Preface to Universal Characteristic” başlığı altında verilmiştir (Bkz. 1989b: 5-10).

düşünüşün formalizmesine doğru ilk büyük adımı görmesi” (Hızır, 1945: 433) nedeniyle Leibniz’de, Aristoteles’e karşı büyük hayranlık uyanmıştır. Yine de bu hayranlık, Leibniz’in Aristoteles mantığının eksiklerini görmezden gelmesini engellemez. Böylece Leibniz, mantıksal akıl yürütme kurallarını karakteristiklere uygulayarak, biçimsellik kazandırmaya çalıştığı Aristoteles mantığının önüne geçecek bir yeni tür mantık; aritmetik kalkülüse dayalı sembolik mantığı icat etmiştir.

Leibniz’in “Nümerik Karakteristiklere Örnekler” (1679) adlı risalesi, kavramlar arasındaki tüm doğrudan bağıntıların, sayısal olarak belirlendiği dört işlemle sınırlı bir mantıksal operasyon denemesi gibi görünmektedir. Üç alt başlık altında toplanan bu risale hem bir analiz hem de bir sentez gerektiren, özne ve yüklem arasındaki gerçek bağıntıdan kopmamayı esas alan, sembolleştirmeye dayanan cebirsel işlemler ile bağıntının belirlediği *sylogism* üzerinde yargı bildirecek durumları açıklamak üzere kaleme alınmıştır.

Risalenin A. alt başlığında, argümanların biçim ve konuya göre ayırt edilmesi gerektiği, bazen bir argümanın belirli bir konuyla ilgili işe yarasa bile aynı formun her örneğe uygulanamayacağını (örnek 1989c: 10’dan alınmıştır) aşağıdaki gibi göstermiştir:

Her metal mineraldir.
Bazı metaller altın değildir.
Öyleyse altın mineral değildir.

Yukarıdaki alıntı incelendiğinde, görülecektir ki form açısından doğru olan bu *kıyas* (sylogism) konu bazında değerlendirilmek üzere örneklendirildiğinde yanlış sonuçlar vermektedir. Leibniz bu örnekten hareketle, yargı sanatının konu bazına bağlı kalarak hatalı sonuçlar doğurmasını engelleyecek bir biçimsel hesabın, sadece öncüllerde bulunan her terim veya kavram için belirli birer karakteristik verilerek, bu karakteristikler arasında yapılacak cebirsel işlemler neticesinde, form üzerinden gerçekleştirilebileceğini göstermeye çalışır. Eğer, önerme içeriklerinden azade bir form yakalanabilir ise bu durumda bu form içeriklerinin nümerik hesabına dayalı sonuçlarının yargısı bizi ancak basit bir cebirsel sağlamanın yapılması kadar yorabilir.

Leibniz’e göre doğru karakteristik sayılar, ancak çok uzun bir disiplinli çalışmanın ardından kavramlar sözlüğü elde edildikten sonra verilebilecektir. Bu sebeple, evrensel karakteristik üzerine verdiği her örnek için keyfi kullanımları tercih etmiştir.

Her ne kadar her terim veya mefhumu bir karakteristik verilebileceğini keşfetmiş olsam da (bu yolla hesap yapma ve uslamlama gelecekte aynı şey olacak) şeylerin inanılmaz derecede karmaşık

olmasından dolayı, şu an için doğru karakteristik sayıları, genel kategorilerin altına düşen şeyleri düzenlemeden veremiyorum. Bununla birlikte, çıkarımların biçiminin bir hesapta ele alınabileceğini ve şu an için gerçek karakteristik sayıların yerine kullanılacak kurgusal sayılarla bu hesabın gösterilebileceğini düşündüm (Leibniz, 1989c: 11).

Her kategorik önerme içeriği, “özne” “yüklem” “copula” (-dır, -dir) nitelikleri bakımından “olumlu” veya “olumsuz” olan “her” veya “bazı” niceleyicilerinden oluşmaktadır. Böylece, bir kategorik önerme genel olarak “evrensel olumlu” “tikel olumlu” “evrensel olumsuz” ve “tikel olumsuz” olarak ayrılırken bu önermelerin içeriği olan özne ve yüklem arasında, olumlu önermelerde duruma ve bağıntıya göre tam veya belli derecelerde içerilmeler mevcut iken olumsuz önermelerde ise özne ve yüklem arasında bir çelişki veya değilleme mevcuttur. Leibniz’e göre dikkate alınması gereken husus her bileşik kavramın, olumlu veya olumsuz olsun başka kavramlardan oluştuğudur.

Leibniz, bileşik kavramlara özellikle “cins” ve “tür” bağlamında dikkat etmektedir. “Two Studies In The Logical Calculus”²⁶ [Mantıksal Hesaplama İki Çalışma (1679)] isimli risalesinde Leibniz, “cins” kavramının parçayı, “tür” kavramının ise cins ve farklılıkların bütünü temsil ettiğini (1969e: 237) ifade etmektedir. Leibniz, görüşünü örneklendirdiği aynı adlı risalede şu şekilde yazmıştır:

Örneğin altın ve metal kavramları parça ve bütün olarak birbirinden ayrılır, şöyle ki altın kavramında metal kavramından daha fazla içerik vardır, örneğin; metallerin en ağırlı olmak gibi. Bu yüzden altın kavramı metal kavramından daha geniştir. Ekoller tersini söylüyor, kavramları değil, evrensel kavram altında toplanan durumları (örnekleri) düşünüyorlar. Bu yüzden altının içerdiğinden daha fazla tür içerdiği için metalin daha geniş olduğunu söylerler. (...) Ben, evrensel kavramları veya idealleri ve onların bileşimlerini, bireyin varlığına bağlamamayı tercih ediyorum. Yani diyorum ki altın metalden daha büyüktür çünkü altın kavramı için metalinkinden daha fazla bileşen gerekir ve bir altını üretmek bir metali üretmekten daha fazlasını gerektirir (1969e: 237-8).

Olumlu terimlerin arasında Leibniz’in ileri sürdüğü türden “tür-cins” ilişkisi, olumsuz terimler arasında da mevcut olup, içerilme, tersi şekilde işlev görmektedir. Bu risalede, karakteristik sayılar arasındaki dört işlemin kusursuzluğundan emin olan Leibniz, yapması gereken açıklamalar ve belirlenimlerin tamamı için, özne-yüklem bağlamında kavramlar arasındaki ilişkinin doğru bir içerilmesinin yapılabilmesinin gerek neden olduğunu açığa

²⁶ “Two Studies In The Logical Calculus” (1969e [1679]: 235-247) isimli risalesi, “Samples Of The Numerical Characteristic” (1989c [1679]: 10-18) isimli risalesiyle konu bakımından aynı olsa da farklı kaynaklara ait bu çalışmalar içerik bakımından farklılık göstermektedir. İlkinde argümanlar ve ispatlar öncelikli sırada iken ikincisinde, özellikle karakteristik sayılarla yapılan işlemler belirgin durumdadır. Tezimizin bu kısmında, bütünlük sağlaması açısından iki çalışmadan da faydalandık.

vurmaya çalışmaktadır. Leibniz'e göre (+) ve (-) işaretleriyle donatılmış iki zıt sayıdan müteşekkil gerçek karakteristikler için doğru bir kavram analizi önceliklidir.

Tezimizde, “Nümerik Karakteristiklere Örnekler” (1679) isimli risalesinin “A. A Calculus of Consequences” [Sonuçlar Hesabı] alt başlığında Leibniz'in kavramlar için henüz gerçek bir karakteristik sayı belirleyememiş olmasından kaynaklı, kendisinin keyfi rakamlar vermek suretiyle sonuca götürdüğü işlemleri sıralamaya çalışacağız. Bu risalede verdiği bir örnekten (1989c: 12) yola çıkarak, böylece, Leibniz'in öngördüğü işlemler bütünü ve özne ile yüklem arasındaki içerilme ilişkilerini gözler önüne sermeye çalışacağız.

hayvan		rasyonel	
+13	-5	+10	-7
insan			
+130 -35			
Her insan			
+130	-35	rasyonel	bir hayvandır.
		+10	-7
		+13	-5

Yukarıda bulunan örnekten daha iyi anlaşılacağı üzere, öncelikle her kavramsal içerim bağıntılıdır, her kavram zıt işaretli ve aralarında asal olmak üzere, iki ayrı sayı ile belirlenir. Daha sonra, belirli bir kavramın ilkselleri konumunda bulunan kavramların aynı işaretli sayıları çarpılarak, sonuç, işaretler değiştirilmeden (iki negatif sayının çarpımı pozitif olmasına karşın, Leibniz'in işaretleri sadece karakteristiklere verilen belirlenimlerdir) belirli kavramın karakteristiği olarak yazılır.

Kavramsal içerilmenin genel çıkarım kurallarını belirlediği adı geçen risalesinin “B. A Fragment on Rules for Drawing Consequences” [Sonuç Çıkarma Kuralları Üzerine bir Fragment] alt başlığında Leibniz, çıkarım kurallarını karakteristik sayılar arasındaki ilişki olarak sunmuştur ve tüm bu çıkarım kuralları aynı işaretli sayılardaki ilişkilere dayanmaktan ziyade zıt işaretli sayılar arasındaki ilişki ve belirlenim bağlamında yürütülmektedir. Böylece, bu alt başlıkta detaylandırılmış çıkarım kurallarının bir özeti şu şekilde verilebilir (özet, [1989c: 14-17]'den çıkarsanmıştır):

- (1) Aynı terim içinde ortak bölene sahip sayı verilemez çünkü aralarında asal olmak zorundalar. Ayrıca bir terimin karakteristiği, terimin tersi için de terslenebilir; karakteristik sayı olarak, dindar (+10 -3) ise dindar olmayan (+3 -10) belirlenimi gibi.

- (2) Bir öncülde, sadece bir terim bulunuyorsa, karakteristiği dolaysız olarak belirlenebilir, yani istediğimiz karakteristiği verebiliriz.
- (3) Öncülümüz evrensel olumsuz ise o halde farklı terimlerin karakteristiklerinin zıt işaretli olanları arasında bir ortak bölen olabilir.
- (4) Olumlu bir tikel öncülde farklı terimlerin zıt işaretlere sahip sayıları arasında ortak bölen olmamalı.
- (5) Evrensel olumlu bir öncülde, öznenin karakteristiği her sayının, yüklem aynı işaretli sayısına bölünebilir olması beklenir. Bu durum içerilme bakımından önemlidir.
- (6) Tikel olumsuz bir öncülde ise ya farklı terimlerin zıt işaretli sayıları arasında ortak bir bölen olmalı ya da öznenin sayıları, yüklem sayılarına bölünmemelidir.

Yukarıdaki çıkarım kurallarının incelenmesi sonucunda tür ve cins arasındaki içerilme üzerinden belli basitleştirme kuralları elde edilebilir: Özne ve yüklem arasındaki içerilme bakımından, (5) nolu çıkarımdan hareketle, evrensel olumlu bir önerme için bir tam bölünebilme şartı yani $\bar{O}/P=y$ olarak gösterildiği yerde y 'nin bir tamsayı olma şartı aranmaktadır; (4) nolu çıkarımdan hareketle, tikel bir olumlu önerme için $\bar{O}/P=y$ olarak gösterildiği yerde y 'nin bir tamsayı değil, ancak $y=a/b$ gibi bir oran belirtmesi şartı aranmaktadır. “Bu uygulama tek birey-maddelerle ilişkili olan metafizik sonuçların belirlenmesinde yardımcı olur. (...) Bu bağlamda özne içeriği mükemmel bir seviyede anlaşılırsa, yüklem de özneye ait olduğu kolayca bulunur” (Kahveci, 2011: 191).

Aynı risalenin B. alt başlığının incelenmesi neticesinde yaptığımız tespit sonucunda, (3) ve (6) nolu çıkarımlarda ise Leibniz, olumlu önermelerde önerdiği tarzda bir bölmeyi gerekçelendirmeyi olumsuz önermelerde başaramamış görünmektedir. “Üstelik Leibniz, meselâ olumsuz tümelin tahakkuku için, ana prensiplere dayanmayan tamamen keyfi bir yöntem ileri sürmektedir” (Hızır, 1945: 437). Bu yöntemde ise artık sayılar üzerinden aritmetik değil, harfler üzerinden cebir mevcuttur. Burada harfler üzerinden teorem ve ispatlar konu edinerek Leibniz, adı geçen risalenin son alt başlığı olan “C. On Characteristic Numbers” [Karakteristik Sayılar Üzerine]’de, bu hesaplama şekliyle *sylogism*in her modu ve figürünün yalnızca sayısal kurallar yoluyla türetilebileceğini iddia ederken, bu hesaba dair bazı ispatları da (1989c: 18) vermeye çalışmıştır.

Leibniz, “Mantıksal Hesaplama İki Çalışma” (1969e [1679]: 239) adlı risalesinde, tür-cins ayırımından yararlanarak, kavramlar arasındaki içerilme durumunu gösterebileceği daha basit bir örnek seçmiştir: İnsanın karakteristik sayısı 6 ve maymunun karakteristik sayısı 10 ise birbirlerine tam bölünemeyen bu iki kavram birbirleri tarafından içerilmediklerinden

dolayı (1969e: 239) Leibniz'e göre kavram analizi için daha fazlasının yapılması gerekecektir. Bu nedenle, varsayılan ilkselleri doğru belirlemiş olmak öncelikli sıradadır.

Görüleceği üzere, Aristoteles'ten sonra pek az değişikliğe uğramış kıyas mantığı, Leibniz'in yaptığı bu biçimsel dönüşüm, *kör düşünce* ve sembol manipülasyonu sayesinde basit bir hesap mantığına bürünmüş gibi görünmektedir. Böylece, Leibniz'in bir resmini sunduğu evrensel dili ve temellendirildiği aritmetik kalkülüse dayandırılmış sembolik mantığı ile birlikte elimizde, konuşulmasa da *pasigrafik* yapıya sahip bir iletişim dili olmayı kavram göstergeleri üzerinden; önermelerinin doğruluğunun sayısal zeminde sağlamlığa dayandırıldığı, dünya ve metafizik hakikatlere dokunma çabası içinde gerçeklikle bağ kurmaya çalışan *a priori felsefi* bir yapay dilin görüntüsü olduğu açıkça kabul edilebilir.

1.6. Leibniz'in Evrensel Dil Hayali Üzerine Kısa Bir Değerlendirme

Evrensel dil, Leibniz'in bilgi teorisi üzerinden yükselen ve tüm hakikatlerin içinde bulunabileceği bir Genel Bilimler Ansiklopedisi olarak tanımladığı bilgi temelini, insanlığa aktarılabilmesi için elzemdir. Leibniz'in evrensel dil hayali iki ayrılmaz parçadan oluşan bir programa dayanmaktadır: Öncelikli sırada olan ilk parça, düşüncenin alfabesi olarak tanımlanabilecek evrensel karakteristiklerden (*characteristica universalis*), ikinci parça ise bir mantıksal hesaplayıcıdan (*calculus ratiocinator*) oluşmaktadır. Bu ikisi arasındaki ayırım belirgin olmasına karşın, keskin hatlara sahip olmadığından ötürü sıklıkla göz ardı edilse de tezimizin ikinci bölümünde görüleceği üzere sembolik mantık üzerine yapılan çok önemli çalışmalarla birlikte günümüzde “hesap olarak mantık” (İng. *logic as calculus*) ve “dil olarak mantık” (İng. *logic as language*) sorgulamaları altında iki farklı mantık sistemine evrilmiş görünmektedir.

Bir önceki alt başlıkta aktarmaya çalıştığımız şekliyle, hareket noktasını evrensel karakteristikten alarak bir mantıksal hesaba ulaşmaya çalışan Leibniz'in belirgin bir ayrıma gitmediği evrensel dil hayalinin iki ayrılmaz parçasının özellikleri, üç belirlenim halinde aşağıdaki gibi özetlenebilir. Bunlardan ilki, *characteristica universalis*in işlevini tanımlamaktadır.

Characteristica universalis, basit düşüncelerin tam listesinin elimizin altında olduğunu varsayar. Bu basit düşünceler, açık bir şekilde işaretlerle veya anlaşılabilir olduğu sürece karakterlerle belirtilecektir. Bu tasarım (gösterim) programı, basit düşüncelerin listesi ne kadar küçük olursa, o kadar kolay gerçekleştirilebilir. Bir *characteristica universalis*in tam olarak gerçekleştirilmesi ütöpik olabilir, bu nedenle ilk aşamada buluşsal (düzenleyici) bir fikir olarak hizmet eder. Bununla birlikte, matematiğin sembolik sistemlerinde kısmi gerçekleştirmeler bulunur. Gerçek dünya problemlerinin en azından bir

kısmına uygulanabilirler. Bunlar prensipte kısıtlamalar değil, insanın sınırlı güçlerinden kaynaklanan pratik kısıtlamalardır (Peckhaus, 2004: 6-7).

Bu belirlenimlerden ikincisi, *calculus ratiocinator*un Leibniz'in evrensel dil hayali programındaki işlevinin bir tanımını vermektedir. Üçüncü belirlenim ise ütöpik olmasına rağmen gerçekleşmesi durumunda, Leibniz'in sisteminin son halininin alacağı görünümü haber vermektedir:

Calculus ratiocinator, basit düşünceler listesinden tüm olası doğruları mekanik olarak çıkarmaya hizmet eder. *Lingua rationalis*'in sözdizimsel kısmını oluşturur. Karakteristikler, semantik kısımdan sorumludur. Bu programın kısmi gerçekleştirmelerinin ilk döneminde, karakter sistemleri çeşitli yorumlara izin verir. Son (ütöpik) aşamada, olası tüm basit düşüncelerin tam tanımına ulaştıktan sonra, sistem kategorik olacaktır (Peckhaus, 2004: 7).

Leibniz'in hayali olan ve evrensel dil olarak adlandırılmış ütöpik iletişim dili, bu tanımın içini tam olarak dolduramamaktadır. Şöyle ki sosyal zeminde değerlendirilecek olursa bir evrensel dilin en belirgin özelliği, konuşulabiliyor olmasında değil; olgulara dayanmasında aranır. İleride göreceğimiz üzere, Leibniz'in biçimsellik kazanmak suretiyle karakteristik alan kavramlar üzerinden yorumlanabilir evrensel dili, tüm felsefesi görünümüne karşın olgulara dayanmaz.

Bu nedenle, tezimizde Leibniz'in evrensel dilini değil, evrensel dil hayalini sorgulama çabasında olduğumuzu yeniden hatırlatmakta yarar görmekteyiz. İkinci ve üçüncü bölümlerde göreceğimiz üzere, Leibniz'in hayalinin bile uzanamadığı düzeylere gelmiş biçimsel diller vardır. Biçimsel dillerin devamında ise en kaba tabirle programlama dilleri olarak adlandırabileceğimiz bir değil birçok evrensel dil görünümlü *a priori felsefi dil* ve algoritmik yapıyla karşı karşıyayız. Bundan dolayı, sorgulamamız Leibniz'in evrensel dil programının iki ayağı olan *characteristica universalis* ve *calculus ratiocinator* bağlamında, bu dillerin evrenselliği üzerinden yürütülme çabasında olacaktır.

İKİNCİ BÖLÜM

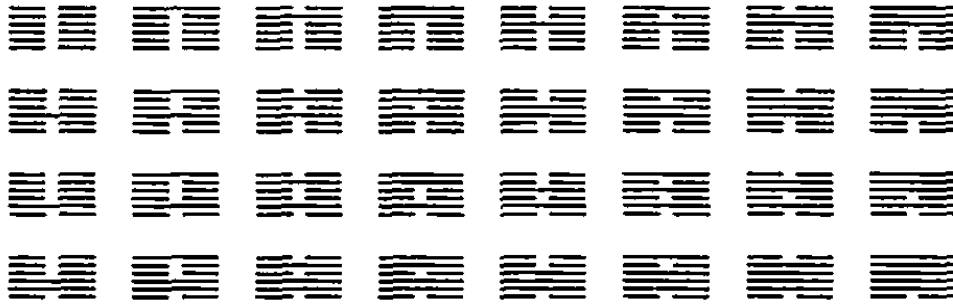
LEIBNİZ'DEN TURING'E BİÇİMSEL DİLLER

Sembolik mantık tarihçesinde rahatlıkla sağlam bir yer tutan Leibniz'in, konu bağlamında yazdığı (bizim de bazılarının, 1.5. alt başlığında içeriklerini vermeye çalıştığımız) risalelerini yayımlatmamış olması, mantık alanındaki yeniliklerin yüz elli yıldan fazla gecikmesine ve başka düşünürler tarafından yeniden keşfedilmesine neden olmuştur. On dokuzuncu yüzyıl ortalarından itibaren, bir önceki bölümde serimlemeye çalıştığımız Leibniz'in evrensel dil yaratma çabalarında yerini tutmuş sembolik mantık, “hesap olarak mantık” (İng. logic as calculus) ve “dil olarak mantık” (İng. logic as language) olmak üzere iki ayrı alanda ciddi çalışma konusu haline gelmiştir.

Tezimizin bu bölümünde, yirminci yüzyılda somut bir tanıma kavuşan algoritmaların üretildiği biçimsel dilleri, Leibniz'in etkisini de gözeterek serimlemeye çalışacağız. İkili sayı sisteminin güzel bir çalışmasını veren Leibniz'in, 0 ve 1 gibi doğruluk değerlerine ulaşamadığı sembolik mantığında, Boole'ün 0 ve 1 gibi doğruluk değeri ifadelerini mantığa dahil etmesiyle başlayan, Schröder'de keskin biçimsellik kazanan ve bir *calculus ratiocinator* olarak görülen Boole cebirine değineceğiz. Leibniz'in *lingua characteristic*a olarak da bilinen *characteristica universalis*inin *Begriffsschrift*'den başlayarak, doruk noktasına ulaştığı (bir hesaplama prosedürü kazandırılmış) *Principia Mathematica*'ya uzanan gelişimi ile birlikte, Hilbert programı, Gödel ve Turing'e kadar devam edecek olan Leibniz'in hayali *evrensel dil* sorgulamasında, Boole'ün mantığı cebirle evlendirme girişiminin, sonraki süreçlerde matematiğin dilinin pekinliğini sağlamak üzere ne tür başka gelişmelere yol açtığına değinmeye çalışacağız.

2.1. Leibniz ve İkili Sayı Sistemi

Leibniz, Çin'den gelen Cizvit misyonerlerinden yalnızca *characteristica universalis* için kör düşünceye dayalı sembol manipülasyonu içeren Çincenin pasigrafik dil yapısını esin kaynağı olarak almamıştır. Leibniz, Cizvit misyonerleri sayesinde, kendi metafiziğine uygun düşen Tanrı ve oluşa gelenler arasındaki bağlantıyı kolaylıkla serimleyebileceği bir örnek olarak, Çince değişimler kitabı “*I-Ching*'in 64 heksagramında temsil edildiğini gördüğü” (Eco, 1995a: 282) varlık ve hiçliğin Çin mistisizmine dayalı (1,0) gibi ikili sayı sistemine yönlendirilebilen, sonradan metafizik temeller üzerinden matematikselleştirme yapma olanağı bulacağı antik Çin felsefesiyle de tanışmıştır.



Şekil 2.1 *I-Ching* Heksagram modelinden bir kesit

Kaynak: Eco, 1995a: 283

Leibniz bu modelde, her biri altı parçadan oluşan kesikli veya düz çizgilerin bulunduğunu ve bu çizgilerin eskiklik ya da yokluk olarak ifade edilebilecek kesikli olanlarının 0 sayısı, tamlık ya da varlık olarak ifade edilebilecek düz olanlarının ise 1 sayısı ile temsil edilebileceğini görmüştür.

Leibniz, kısaltılmış başlığıyla “Explanation of Binary Arithmetic” [İkili Aritmetik Sistemin Açıklaması (1703)] isimli kısa risalesinde, bazı örnekler ve tablolar üzerinden ikili sistem ve kullanımı oldukça kolay olan onluk sistem arasında yaptığı kıyaslamalar üzerinden en az onluk sistem kadar kolay bir aritmetiğe sahip olan ikili sayı sisteminin kullanılabilirliğine dair tespitlerde bulunur. İkili sayı sisteminin basit dört işlem aracılığıyla kolaylıkla onluk sayı sisteminde bir gösterime kavuştuğu veya tam tersine, onluk sayı sistemindeki bir sayının kolaylıkla ikili sayı sisteminde bir gösterime kavuştuğu örnekleri ayrıca basit tablolar aracılığıyla gösterir. Leibniz’in aynı adlı risalesinde bulunan aşağıdaki tablo, *Fuxi figürleri* olarak da bilinen *I-Ching*’in, ikili ve onluk sayı sistemleri ile eşleştirilmesine bir örnek olarak verilmiştir.

000	0	0
001	1	1
010	10	2
011	11	3
100	100	4
101	101	5
110	110	6
111	111	7

Şekil 2.2 Leibniz eşleştirmesinden bir örnek

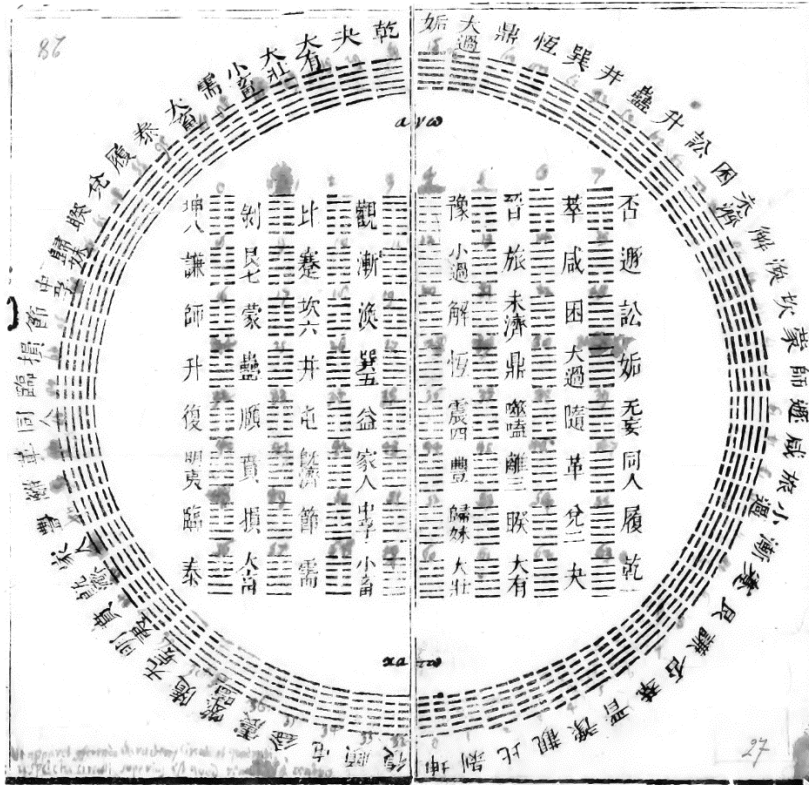
Kaynak: Leibniz, 2007b: 2

Leibniz, 1703 tarihli adı geçen risalesini kaleme almadan çok önce, hemen her konuda en verimli eserler ortaya koyduğu 1679 yılında belli ki ikili aritmetik ile Tanrı arasında teoloji

ve matematiğe dayalı bir izdüşüm elde etme çabasıdır. Leibniz'in 1679 yılına ait "De Organo Sive Arte Magna Cogitandi" [Büyük Sanatı Düşünme Yolu] adıyla kaleme aldığı risalesinin aşağıdaki alıntısından anlaşılacağı üzere, Leibniz, çok uzun zaman önce ikili aritmetik ile ilgilenmektedir.

Bizzat kendisi sayesinde kavranan tek bir varlık vardır, o da Tanrı'dır ve onun dışında hiçbir şey yoktur veya yokluk vardır. Bu hayranlık uyandıran bir gülümsemeyle izah edilebilir. Hesap yaparken genellikle onluk sistemi kullanırız, dolayısıyla ona geldiğimizde tekrar birden başlarız. Bunun doğru olup olmadığını şimdi tartışmayacağım; onun yerine ikili bir sistemi kullanmanın mümkün olduğunu göstereceğim. (...) Bu sistemin muazzam avantajlarına şimdi değinemem; muhteşem bir yolla bütün sayıların birlik ve yokluk ile ifade edilebildiğini belirtmekle yetineyim (Leibniz'den aktaran Antognazza, 2013: 301).

Leibniz'in ikili aritmetik konusunda yaptığı çalışmalar neticesinde, Çin'den gelen Cizvit misyonerlerinden Peder Joachim Bouvet ile yaptığı görüşmelerden birinde heyecanını aktarması üzerine, bu sistemde Çin heksagramlarının yapısının bir çeşit açıklaması olduğunu gören Bouvet, Leibniz'in eline ancak iki yıl sonra geçecek olan ve heksagramların dizilişini gösteren bir gravürün ekinde olduğu (Leibniz, 2007b: 3) 1701 tarihli mektubu gönderir.



Şekil 2.3 Joachim Bouvet tarafından Leibniz'e gönderilen *I-Ching* ve Leibniz'in yaptığı eşleştirme

Kaynak: Antognazza, 2013: 368

Heyecanında haklı çıkan Leibniz, yaptığı keşfin²⁷, bu mektup vesilesiyle Çin mistisizmi, onluk sayı sistemi, ikili sayı sistemi ve Tanrı ile monadlar arasında kurmaya çalıştığı bağlantısallığı verdiğini görür. Basit aritmetik işlemler aracılığıyla evrensel bir kanıt sunduğunu gösterdiği “İkili Aritmetik Sistemin Açıklaması” (1703) adlı risalesinde, keşfi hakkında şöyle yazar:

On ile sayma uygulaması daha kısadır ve sayılar o kadar uzun değildir. Ancak uzunluğunun telafisi olarak ikişer, yani 0 ve 1 ile hesaplamak, bilim için en temel hesap yöntemidir ve özellikle geometriciler için bile sayıların pratiği konusunda yararlı olduğu görülen yeni keşifler sunar. Bunun nedeni, sayılar 0 ve 1 gibi en basit ilkelere indirgendiği için baştan sona harika bir düzenin ortaya çıkmasıdır (2007b: 2).

İnsanlık için bir hizmet aracı olarak gördüğü *characteristica universalis* gibi ikili aritmetik keşfini de aynı noktada gören Leibniz’e göre ikili aritmetik, “analizcilerin her türlü kütleli az ağırlıklı tartmasını sağlayabilir ve çok az sözcük ile sözdiziminde birçok değer vermek için (2007b: 1) hizmet edebilirdi. Leibniz’in kendi içinde çığır açıcı keşfi, yaşadığı dönem ve genel itibarıyla on sekizinci yüzyılda değer göreceği bir noktaya ulaşmamış olsa da on dokuzuncu yüzyılın son çeyreğinden günümüze değerini arttırarak gelmiştir.

Bu 1’lerle bu 0’lar gerçekten de kör simgelerdir; onların sözdizimi işlevsel olup, ürettiği dizilere olası herhangi bir anlam verilmeksizin keşiflere olanak sağlarlar. Hiç kuşkusuz Leibniz, yüz elli yıl önceden, George Boole’un matematiksel mantığını önceler; ancak aynı zamanda bilgisayarların gerçek dilini de önceler (Eco, 1995a: 283-4).

Bir aritmetik kalkülüs tanımladığı ve içerikten azade olması gerektiğine inandığı yeni mantığı için en azından bir mantıksal hesaplayıcı bağlamında değerlendirecek olursak, Leibniz’in “istediği şey, kesinlikle matematiğin bir dalı olan ve Boole tarafından (...) geliştirilen modern sembolik mantık bilimine benziyordu” (Russell, 2005: 234). Boole’un

²⁷ Leibniz’in ikili sayma sistemine dair keşfin kendisine ait olmadığına; kendisinden uzun zaman önce yapılan başka çalışmalardan intihal yaptığına dair iddiaların mevcut olduğunu bildirmekte yarar olduğunu düşünmekteyiz. Hannover’de yayımlanmamış eserlerinin gün yüzüne çıkmasından hemen sonra, Leibniz’in bir tilmizi ve hayranı olan Couturat’ın, Leibniz’i ikili sayı sisteminin ve aritmetiğinin kurucusu olarak göstermesinden dolayı, Leibniz üzerine yazılan diğer tüm kitapların genel bir kabulde Couturat’ı referans noktası aldığından beisle, kronolojik bir tespit üzerinden intihali göstermeye çalışan bir makalede özetle şöyle yazmaktadır: “Leibniz ikili sistemi ve ikili aritmetiğini yayımlamadan otuz yıldan fazla bir süre önce, Caramuel, Harriot’tan bağımsız olarak, 1642 tarihli *Mathesis Audax* adlı çalışmasında ikili sistemini ortaya koydu (yani, zamansal öncelik)” (Ares vd., 2018: 15). Bu makaleye göre kronolojik olarak, ilk sırada Thomas Harriot (on yedinci yüzyıl başları fakat tarih belirsiz), daha sonrasında ise Francis Bacon (1605 ve 1623 yıllarında) ikili sistemi tanımlamış görünmektedir. Makaleye göre, 1670 yılında Spaniard Juan Caramuel de Lobkowitz, “*Mathesis Biceps* başlıklı 'büyük ansiklopedisinde' ikili aritmetiği vb. geliştirdi ve sistematik olarak açıkladı” (2018: 15). İleri okuma yapmak isteyenler için “Who Discovered the Binary System and Arithmetic? Did Leibniz Plagiarize Caramuel?” başlıklı makale, ilgili okuyucuların dikkatine sunulur.

matematiksel mantığını önceleyen tüm çalışmaları Leibniz'in yayımlanmayan, tamamlanmamış eserleri ve mektupları arasından gün yüzüne çıkmıştır. Aristoteles'e hayranlığını gizlemeyen ve hatta belki de sembolik mantıkta gösterdiği tüm başarıya rağmen Leibniz, Aristoteles'in kıyas mantığının ana hatlarını takip etmeye devam etmiş görünmektedir. Bu nedendir ki Leibniz, “önce genel mantığı kurup matematiği sistemin içine yerleştireceği yerde, aritmetiğe uyan bir mantık kurmakla uğraşmış, bu işi başaramadığını görünce de hem cebire hem yüklem mantığına uyan bir sistem kurmaya çalışmış” (Hızır, 1945: 440) fakat başarılı olamamıştır.

2.1.1. Leibniz Sonrası On Sekizinci Yüzyıl

Aristoteles'in karşıtlar karesinden yola çıkarak, Leibniz'in aritmetik kalkülüs yoluyla elde etmeye çalıştığı yeni mantık (calculus ratiocinator), evrensel dili için içerikten azade bir tür doğrulama yapacaktı. Leibniz'in 1716 yılında ölümünden sonra, Leibniz tarzında *a priori* yapay dil ve aritmetik kalkülüse dayalı mantık çalışmalarının izlerine rastlanılmamaktadır. On sekizinci yüzyılda hem evrensel dil arayışları hem de mantık iki ayrı yoldan devam etmiştir. *A posteriori* yapay dillerin atası sayılabilecek bir uluslararası yapay dil projesi, 1734 yılında “Carpophophilus” takma adıyla (Eco, 1995a: 317) yayımlanmıştır. Mantık ise salt düşünce alanına ait olmaktan çıkarak, etkinlik alanına (Aksoy, 2015:16) ait hale gelmiştir. Immanuel Kant'ın (1724-1804) “transcendental mantık” diye nitelendirdiği, saf düşüncenin, deney alanındaki nesne ile bir sentezine referans gösterilen mantık, Johann Gottlieb Fichte'nin felsefesinde (1762-1814) “vicdana dayalı mantık” dönüşümüyle düşünce alanından etkinlik alanına geçişi temsil etmiştir. Leibniz'den sonra mantık, nihayetinde, Georg Wilhelm Friedrich Hegel'in (1770-1831) diyalektik temelli, tez-antitez-sentez mantığına dönüşerek, felsefi bağlamda gelişimini sürdürmüş ve yeni paradigmalara yol açmıştır.

Tezimizin bu bölümünde bize yaklaşık yüz elli yıl zamansal bir epistemolojik sıçrama yaptıran nedene gelince; Leibniz'den Kant'a Christian Wolff (1679-1754) çizgisiyle gelen bilgide Leibniz'in evrensel dil ve yeni mantık arayışlarının henüz ortaya çıkmamış olması ve sonuç olarak felsefi mantık çizgisinde Leibniz'in dehasına dair bir ize rastlayamıyor olmamızdır.

Leibniz'in, kavram nicelemesi yapılabilmesini sağlayan hesaplamaya dayalı “Mantıksal Hesaplama İki Çalışma” (1679) ve sembol manipülasyonu yapılabilmesini sağlayan sembolik mantığa dayalı “Mantıksal Hesap Üzerine Bir Çalışma” (1690) adlı iki risalesinde geçen notasyonların aynı düşünsel etkinlikle Boole cebirinde görüldüğü açıktır. Bu nedendir ki henüz gün yüzüne çıkmadığı tarihçe doğrulanan, Leibniz'in Aristoteles

önergeler mantığını aritmetik kalkülüs aracılığıyla içerikten azade kılma fikri ile Leibniz'den habersiz Boole'ün, aynı niyetle olmasa da Aristoteles'in kıyas mantığını cebirleştirmeye dair çabasının, ortak bir izdüşüm vermiş olmasından ötürü çizgimizi Leibniz'den Boole'e ve on dokuzuncu yüzyıl mantıkçı, matematikçi ve düşünürlerine kaydırmış bulunmaktayız.

2.2. On Dokuzuncu Yüzyılda Mantık

Mantığın dil ile bağdaşık düşünüldüğü uzun yüzyıllardan sonra birbirinden ayrı ilerleme gösteren mantık ve matematiğin bir tür birleştirilme çabası, on altıncı yüzyılda Hobbes'un, düşünme etkinliğinin aritmetik hesap yapmakla aynı şey olduğuna dair fikir beyanından sonra Leibniz'in ellerinde büyük ilerleme kaydetmiş olsa da on dokuzuncu yüzyılda mantık ve matematik alanındaki gelişmelerde, iki filozofumuzun neredeyse hiçbir rolü yoktur²⁸. Bu yüzyılda mantık alanında, emprizme dayalı tümevarımsal bilimlerin izinde giden, filozof John Stuart Mill'in (1806-1873) "indüktif mantık" adıyla anılan mantığı ve buna dayalı felsefi çalışmaları yer bulmuştur.

Felsefe dünyasında mantığın bir tür kısıtlanışına şahit olduğumuz on dokuzuncu yüzyıl başlarında, özellikle "Cambridge Ağı" (İng. The Analytical Society) olarak isimlendirilmiş ve Büyük Britanya'da kurulmuş bir matematik topluluğu (Peckhaus, 1999: 438), ileride mantığın gelişimine büyük fayda sağlayacak konular üzerinde çalışmıştır. Tasarımı hayata geçirilememiş, "analitik motor" adıyla bilinen mekanik bilgisayar modelinin mucidi Charles Babbage²⁹ (1791-1871) ve George Peacock (1791-1858) gibi büyük matematikçilerin kurucuları arasında bulunduğu bu topluluk, özel olarak matematiksel notasyonlar üzerinde durmaktadır. Diferansiyel hesapta Newton'un akış teorisini terk ederek Leibniz notasyonlarını (simgeleştirmelerini) benimsemişlerdir. Peckhaus'a göre notasyondaki bir değişikliğin tetiklediği bu başarılı hareketin, sembollerle çalışma konusunda yeni veya en azından yeniden canlanan bir ilgi uyandırmış olabileceği varsayılabilir (1999: 438). Gerçekten de sonraki

²⁸ Özellikle Leibniz'in referans noktası olarak alındığı, "sembolik mantık", "mantık cebiri", "matematiksel mantık", "mantık hesabı", "algoritmik mantık" ve on dokuzuncu yüzyıl sonlarında "logistic" gibi isimlerle anılan modern mantığın Leibniz'in çalışmalarından bağımsız olarak kurulduğuna dair, kronolojik bir tespit bulunmakta yarar görmekteyiz. İlk olarak Louis Dutens 1768 yılında Leibniz'in yayımlanmamış yazılarını gün yüzüne çıkarır fakat burada henüz mantık ile ilgili bir risale bulunmamaktadır. Leibniz'in, Gabriel Wagner'e mantığın yararlarından söz ettiği mektubunun da (1696) içinde bulunduğu çalışmalar, ilk kez Johan Eduard Erdmann tarafından 1839 ve 1840 yıllarında iki cilt halinde yayımlanmıştır. Louis Couturat 1901 ve 1903 yıllarında Leibniz'in mantık üzerine çalışmalarını incelediği eserlerini sunmuştur. Ayrıca Leibniz'in büyük bir hayranı olan Henry Scholz, Leibniz'in İngiliz dünyasında gerçekleşen mantık çalışmalarında bir etkisi olmadığını (Peckhaus, 2018: 2) ifade etmiştir. Öyle görünüyor ki "Leibniz sembolik ve matematiksel mantığın mucidi olarak adlandırılabilmesine rağmen, O, nesilsiz olduğu için modern mantığın babası değildir" (Kahveci, 2012: 19).

²⁹ Yaşadığı dönemin teknolojisinin, tasarımını hayata geçirmesine olanak tanımadığı Babbage'ın "analitik motor" tasarımının, bugün teknolojisi ile değerlendirildiğinde, "evrensel bir bilgisayar olduğunu, yani istenen hesabı yapmaya programlanabileceğini görüyoruz" (Say, 2019: 106).

faaliyetler göstermektedir ki aritmetiğe dayalı cebir ile sembolik cebir arasında bir ayırım yapmak üzere girişimlerde bulunulmuştur.

Doğru bir sembolleştirme ile sayıların ve biçimsellikten uzak terimlerin hesaplamada getirebileceği yükten kurtulmanın mümkün olduğunu, Newton yerine Leibniz notasyonlarını kullanarak deneyimlemiş olan Babbage'a göre, "böylece, başlangıçta anlamları salt sayılarla sınırlı olan harfler, yavaş yavaş başka ikincil anlamlar kazandılar ve çeşitli durumlarda zaman, mekan, yön ve diğer koşulları ifade ettiler" (Babbage'den aktaran Phillips, 2013: 6). Cebirin sembolleştirilme girişimleri yalnızca matematikte sağlanacak bir fayda olarak görülmemekteydi; sağlanacak olan kolaylık, diğer disiplinler için de geçerliydi.

Notasyon kullanımı konusunda Babbage ile aynı noktada duran Peacock'un notasyonlara olan merakı neticesinde aritmetiğe dayalı cebir ile sembole dayalı cebir arasında yapmaya çalıştığı ayrımları genelleştiren Duncan F. Gregory (1813-1844), sembollere dayalı matematiksel işlemlerle özel olarak ilgilenmiştir. Aynı tarihlerde, Peacock'un sembollere dayalı cebirinin, aritmetiğe dayalı cebirden bağımsız olarak diğer disiplinlerde geçerli olamayacağına dair eleştirilerin en barizi, büyük matematikçi Sir William Rowan Hamilton (1805-1865) tarafından yapılmıştır.

Hamilton'un Peacock'a yönelttiği eleştiri, özünde Peacock'un cebiri düşünmek için kullandığı yöntemin, cebiri bir bilim olarak geliştirmekten ziyade, dilin nasıl geliştirileceğiyle fazlasıyla ilgili olduğu yönündeydi. Bu, modern bir perspektiften, çoğunlukla Peacock'un fikrinin dayandığı eşdeğer biçimlerin kalıcılığı ilkesini tam olarak doğrulamadığı için geçerli bir eleştiridir (Phillips, 2013: 17).

Boole ile yakın arkadaşlığı olan Gregory gibi August De Morgan da (1806-1871) kendisiyle ortak çalışmalara imza attığı Hamilton'un yaklaşımının tersi yönünde hareket ederek, Peacock'un, $\sqrt{-1}$ gibi karmaşık (complex) sayıların kavramsallaştırılmasına dair çalışmaları üzerinden devam etmiş, bu sistemi geliştirmeye çalışmıştır.

2.2.1. August De Morgan (1806-1871)

Aristoteles'in syllogistic mantığının eksiklerini "olmak ya da olmamak" veya başka bir ifadeyle, "varlık" ve "yokluk" gibi kısıtlayıcı ifadeler altında belirlenmiş ilişkiselliğe bağlayan De Morgan'a göre, "bu mantık, eğer bir at, bir hayvan ise bu durumda bir atın kuyruğunun, bir hayvanın kuyruğu olduğunu kanıtlayamazdı ve x y 'yi *sever* gibi bir ilişkiyi kesinlikle kaldıramazdı" (Kline, 1972c: 1189). Ancak De Morgan, kıyas mantığının ötesine geçemez.

De Morgan, mantık üzerine yaptığı ilk çalışmalarını *Biçimsel Mantık* (Formal Logic [1847]) adlı eserinde somutlaştırmıştır. Bu eserde özetle; alfabenin büyük harfleri kullanılarak

olumlu, aynı harflerin küçükleri kullanılarak olumsuz ifadelere kavuşması neticesinde sembolleştirilen Aristoteles'in kıyas mantığının karmaşık önermelerine, içerikten azade bir form kazandırılmaya çalışılmıştır.

Ortak olanı içeren basit bir gösterim, her kıyasın her biri yukarıda veya aşağıda vurgulanan üç harfle temsil edilmesinin araçlarını verir. Bunlardan birinin incelenmesiyle, 1. hangi kıyasın temsil edildiğini, 2. geçerli mi yoksa geçersiz mi olduğunu, 3. hemen nasıl yazılacağını, 4. zihin bu çıkarımı yaptığında, çıkarımın hangi aksiyomu içerdiğini veya edimin ne olduğunu hemen görür (De Morgan, 1847: 11).

De Morgan çalışmasında, saf geometrinin tüm önermelerinin, olabildiğince az sayıda basit kavramdan çıkarılabilecek şekilde içerildiğini ifade ederek (1847: 79), özellikle bu öncüllerden çıkarılabilecek şekilde içerildiğini ifade ederek (1847: 79), özellikle bu öncüllerden çıkarılabilecek şekilde içerildiğini ifade ederek (1847: 79), özellikle bu öncüllerden çıkarılabilecek şekilde içerildiğini ifade ederek (1847: 79) olduğunu vurgular. De Morgan'a göre mantık, özünde *-dir*, *-değildir*'in önermeler içinde kullanımının sonuçlarıyla ilgilidir (1847: 78). Bu nedenle, çalışmasında sembollerle yapılan *değillemelere* odaklanır. Böyle bir kavram ayrımını yapan ilk mantıkçı olan De Morgan, karmaşık terimlerini X, Y, Z, P, Q gibi büyük harflerle tanıtırken bu terimlerin değillenmiş hallerini x, y, z, p, q gibi küçük harflerle tanıtır. "P ve Q" bağlacı "PQ" ile gösterilir ve ayırıcı "P veya Q" ("veya", özel olmayan anlamda alınır) "P, Q" ile gösterilir (1874: 118). Böylece,

PQ'nun tersi p, q'dur; P, Q'nunki ise pq'dur.

Bu ifadelerin gelişmiş halini sunacağı sonraki çalışmalarında De Morgan, bugün *De Morgan Kanunları* adı altında bilinen tikel ve tümel değillemelerinin biçimsel halini tanıtır. De Morgan Kanunları özetle, bir bileşiğin karşıtının, bileşenlerin karşıtlarının toplamı olduğunu; bileşenlerin toplamının karşıtının da toplamın karşıtlarının bileşimi olduğunu, ifade eder. De Morgan'ın kendi ifadesiyle, "mükemmel bir dil hayal edebilseydik, bu dilin, her ismin karşıtını gösteren bir kip içereceğini varsaymamız gerekirdi" (1847: 73). Mantıksal gösterimde bu kanunlar aşağıda (Kline, 1972c: 1189) gösterildiği şekildedir:

$$1 - (x + y) = (1 - x)(1 - y).$$

$$1 - xy = (1 - x) + (1 - y).$$

De Morgan Kanunlarının önermesel mantıktaki gösterimi (Gensler, 2017: 462) ise aşağıdaki gibidir:

Hem A hem de B değil = A veya B değil

A veya B değil = Hem A hem de B değil

Yakın tarihlerde De Morgan gibi ünlü metafizikçi ve mantıkçı Sir William Hamilton da³⁰ (1788-1856) yüklem nicelemesi yoluyla geleneksel mantık üzerinde bazı modifikasyonlar önerdiği çalışmalar yayımlamıştı. Çalışması De Morgan'ın eserinden önce yayımlanan Hamilton, De Morgan'ın mantık üzerine 1846 yılında yaptığı daha önceki çalışmalardan, bu eserin yayımlanmasından çok önce haberdardır ve De Morgan'ın, “mantıktaki büyük keşfi olarak düşündüğü şeyi, 'yüklemin niceliği' dediği şeyi intihal ettiğini” (Davis, 2018: 20) duyurur³¹. Aslında, Hamilton'ın teklif ettiği yeni mantığının “oldukça hantal gösterimi, işlemlerin temeli yapılmaz, esasen yalnızca dilin bir kısaltmasıdır” (Lewis, 1918: 42). Oysa De Morgan'da durum çok daha farklıdır, onun eseri, Leibniz'in de öngördüğü fakat yürütmekte başarısız olduğu *ilişkiselliğe* dayalı gerçek bir sembolleştirme ile ilgilidir.

De Morgan, “Aristoteles mantığında özne-yüklem ilişkisine dayalı önerme formu yanında başka önerme formlarının ve dolayısıyla başka mantıksal ilişkilerin de varlığına dikkat çekmiştir” (Özlem, 2004: 224). De Morgan'ın yaptığı yeniliklerden verilebilecek bir örnek olarak; “*niceliğin* ya tam (örneğin, A'ların m'si) ya da (örneğin, A'ların çoğu) yaklaşık bir sayısal karaktere sahip olacak şekilde genelleştirildiği kıyas kavramının genişletilmesi” (Hailperin, 2004: 357) denilebilir. Burada, De Morgan'ın ne demek istediğini, Gensler'in verdiği bir örnekle açıklamak gerekirse: “Bütün köpekler hayvandır, bu nedenle tüm köpek kafaları hayvan kafalarıdır” (2017: 480) gibi bir çıkarımı geleneksel mantık, ilişkisel argümanlar bağlamında açıklamaktan acizdi. De Morgan böyle bir ilişkiselliği verebilecek mekanikleştirmenin peşindeydi.

Yeni mantık konusunda çeşitli öneriler ve tekliflerin sunulduğu çalışmaların, belki de en verimli yapıldığı yıl olan 1847'de Boole, yakın dostluk içinde olduğu De Morgan'ın

³⁰ Bazı kaynaklarda, aynı dönemde yaşamış olan metafizikçi Sir William Hamilton'un (1788-1856), büyük matematikçi Sir William Rowan Hamilton (1805-1865) ile karıştırıldığı gözlemlenmiştir. De Morgan aynı dönemde hem Sir W. Rowan Hamilton ile matematik ve mantık üzerine ortak çalışmalar yapmış hem de metafizikçi Sir W. Hamilton, mantık alanında kendi özel çalışmalarına sahiptir. Örnek vermek gerekirse, Yavuz Aksoy'un *Boole Cebiri ve Lojik Devre Sentezi* adlı kitabında (2015: 18) ve VII. Mantık Çalıştayında sunulmuş olan “George Boole'ün Cebirsel Mantığı ve Mantık Tarihindeki Yeri” adlı makalede (Öztürk, 2017: 574), De Morgan ile aralarında geçen intihal tartışmalarında, dönem dönem birlikte ortak çalışmalara imza atmış Sir W. Rowan Hamilton'un adı geçmektedir. Bu hatanın bir isim benzerliğinden kaynaklandığı açık olmakla birlikte, Boole'ün *The Mathematical Analysis of Logic (1847)* adlı eseri ve De Morgan'ın *Formal Logic: Or, The Calculus of Inference, Necessary and Probable (1847)* adlı eserinin incelenmesi sonucunda, görülecektir ki intihal tartışmalarının yaşandığı kişi, metafizikçi ve mantıkçı Sir W. Hamilton'dur.

³¹ De Morgan ve Hamilton arasında geçen intihal tartışmaları, uzun süre canlı kalır çünkü “Hamilton, bu konudaki önceliğini belirlemek için De Morgan ile uzun uzadıya” (Lewis, 1918: 42) tartışır. Öyle ki bu tartışmaların içinde geçtiği mektuplar bilim çevrelerince makale gibi yayımlanır. Böylece bilim ve düşün dünyası da tüm bu tartışmaları yakından takip eder. De Morgan, teklif ettiği biçimsel mantığın, mekanikleştirme esasına dayalı olmasından dolayı elbette akademik centilmenlik suretiyle tüm bu karşılıklı mektuplaşmaları, intihal konusu edilen çalışmalarının bulunduğu eserinin eki olarak kitabının en arka sayfalarında sunar. İncelenmek istenmesi durumunda, mektuplar, De Morgan'ın *Formal Logic: Or, The Calculus of Inference, Necessary and Probable (1847)* adlı eserinde görülebilir.

mantık üzerine çalışmalarının Hamilton tarafından doğru anlaşılmadığına ve ortada bir haksızlık olduğuna dair fikre sahiptir. Boole de tıpkı De Morgan gibi, “mantıksal olarak geçerli tüm çıkarımları temsil etmek ve manipüle etmek için genel bir yöntem geliştirerek geleneksel kıyas mantığının sınırlarını genişletme” (Barnett, 2011: 1) çabasıdadır. Aynı yıl, Boole’ün De Morgan’dan aldığı mektupta, De Morgan’ın “çalışmalarını temsil eden bir notasyonla geçiş yapmanın mekanik modlarını” (Smith, 1982: 25) geliştirmeye yönelik çabası olduğu görülmektedir.

Yakın dostu Gregory’nin sembollere dayalı cebir konusundaki fikirlerini inceleyen Boole, De Morgan’ı Hamilton’a karşı haklı çıkarma çabasında, De Morgan’ın aksine “cebirsal yöntemleri açıkça benimseyerek” (Barnett, 2011:1) bugün onun adıyla andığımız ve üniversitelerde okutulan Boole cebiri konusunda önemli bir adım atmıştır.

Peacock, Gregory ve De Morgan, gerçek³² ve karmaşık sayıların özelliklerinden bağımsız olarak, cebirden bir bilim yapmaya çalıştılar ve bu nedenle cebiri yorumlanmamış sembollerin ve bunların kombinasyon yasalarının bilimi olarak gördüler (Kline, 1972b: 775).

Elbette, Boole de aynı yoldan devam ediyordu. De Morgan’ın *Biçimsel Mantık* adlı eseriyle aynı gün yayımlanan (Lewis, 1918: 57), *Mantığın Matematiksel Analizi* (*The Mathematical Analysis of Logic* [1847]) isimli yapıtında Boole, mantıksal bir sembolün, "birleşim yasaları verildiğinde cebirsal olarak tanımlandığını ve aralarındaki kombinasyon yasalarının aynı olduğunda, bir sembolün belirli bir işlemi temsil ettiğini" (Peckhaus, 1999: 439) yazan Gregory’nin cümelerini bire bir alıntılıyarak, De Morgan’ın bir tür sembolleştirme içeren biçimsel mantığının ötesinde, cebirsal bir mantık üzerinde durarak çalışmalarına yoğunlaştı.

2.2.2. George Boole (1815-1864)

Bir matematikçi ve filozof olan Boole, mantığın filozofların elinde değil, matematikçilerin elinde geliştirilmesi gerektiğine inanmaktaydı. Mantık, ancak bu yolla sarsılmaz bir biçime kavuşturulabilirdi. Bu girişimin ilk örneklerinden biri olan *Mantığın Matematiksel Analizi* isimli eserinde Boole, önermelerin cebirsal yolla ifade edilebileceği bir cebirsal mantık inşa etmeye çalışmıştır. Önceliği kelimelerin anlamlarına veren Boole, bu

³² Türkçe bir matematiksel kavram olan “gerçek sayı”, genel kullanımda “reel sayı” olarak bilinmektedir. Karmaşık sayılar haricinde, e , π , $\sqrt{2}$ gibi irrasyonel sayılarla birlikte tüm diğer sayıların içinde bulunduğu ve oranlanabildiği kümenin ismi matematik dünyasında “gerçek sayılar kümesi” olarak yaygın kullanıma sahiptir. Tarafımızca İngilizceden çevirisi yapılan yukarıdaki alıntıda ise sayılar hakkında “reel” değil, “real and complex” olarak geçen ifade “gerçek ve karmaşık” anlamında kullanılmıştır.

yolla kavramların ya da isimlerin ait olduğu sınıflar (küme veya topluluk) arasında cebirsel bir form üzerinden basit aritmetik işlemler tesis etmiştir. Boole, eserinin ilk ilkelerini tanımladığı bölümünde, evreni tanımlamak için 1 sembolünü veya birliği kullanmayı teklif eder; bu sembolün gerçekten var olsun ya da olmasın akla gelebilecek tüm nesnelere sınıfını kapsadığını varsayar (1847: 15). X , Y ve Z gibi harfleri sınıfların bireysel üyelerini temsil etmek için kullanan Boole'e göre, X bir sınıfın her üyesine o sınıfın üyesi olarak uygulanır ve Y başka bir sınıfın her üyesine bu sınıfın üyesi olarak uygulanır (1847: 15) ki böylece, mantık üzerine yazılmış risalelerin diline göre bir devamlılık sağlanabilir. Ayrıca, bir x , y ve z sembolleri sınıfı tasarlayan Boole, bireyleri veya sınıfları içeren herhangi bir özne üzerinde çalışan x simgesinin, tüm X 'leri içerdiği o öznenen seçeceği; benzer şekilde, herhangi bir özne üzerinde işleyen y simgesinin, içinde yer alan Y sınıfının tüm bireylerini ondan seçeceği (1847: 15) varsayımı ile hareket ederek, öncüllerini belirler. Bu öncüllerden çıkan sonuç şöyledir: Bir Y sınıfının ve Y sınıfı ile aynı özelliklere sahip olan X sınıfının bu tür bireylerinin oluşturduğu xy çarpımının üyeleri hem X 'ler hem de Y 'ler olan sınıftır (1847: 16). Benzer şekilde, Z sınıfının bazı üyeleri ile aynı özellikleri barındıran Y sınıfının bu tür bireylerinin ondan seçimi ve X sınıfının aynı özelliklere sahip tüm bireylerinden bir seçim olduğu için xyz çarpımı, bir bileşik işlemi temsil edecektir (1847: 16). Buradan çıkan nihai sonuç X , Y ve Z için ortak sınıftır ki bu yolla, ortak bir nitelik taşıyan sınıfların gösterimi belirlenmiş olur.

Seçilmiş simgelerle temsil ettiğimiz bu zihinsel işlemin analizine burada girmemize gerek kalmayacak. Bu soyutlama, *terimin* genel kabulüne göre bir soyutlama eylemi değildir çünkü bu yolla somutu asla gözden kaçırmayız fakat yine de karşılaştırma ve dikkat yetilerinin bir alıştırmaya atıfta bulunulabilir. Şu anki ilgimiz daha çok yaptığımız şeyin sonuçlarının yönetildiği kombinasyon ve ardıllık yasalarıyla ilgilidir (Boole, 1847: 16, italik vurgu bana aittir).

Boole'ün bu eserinde, sembolik cebire dayandırılmış üç kural ortaya çıkmıştır:

1. Bir seçim eyleminin sonucu, öznenin gruplandırılmasından veya sınıflandırılmasından bağımsızdır (1847: 16).

Bu açıklamanın matematiksel gösterimi aşağıdaki gibi olup, burada $u + v$ bölünmemiş özneyi temsil ediyor ve u ve v onun bileşen parçalarını temsil ediyor (1847: 17). Burada, “+” sembolü ilk defa kullanılıyor ve birbirini dışlayan sınıflar olarak görülmesi gereken u ve v (1847: 17) ile oluşturulan denklemi anlamlandırmak için ortaya çıkıyor. Ayrıca, Hailperin (2004: 350) Boole'ün 1. kuralının, küçük harflerin operatörlere atıfta bulunduğu kuralıyla çeliştiğine de dikkatimizi çekiyor.

$$x(u + v) = xu + xv$$

2. Ardışık iki seçim eyleminin hangi sırayla gerçekleştirildiği önemsizdir (1847: 17).

$$xy = yx$$

3. İki kez veya art arda herhangi bir sayıda gerçekleştirilen belirli bir seçim eyleminin sonucu, bir kez gerçekleştirilen aynı eylemin sonucudur (1847: 17).

$$xx = xx$$

veya

$$x^2 = xx$$

şayet, bu işlem n-defa gerçekleştirilecek olursa sonuç aşağıda görüleceği üzere yine değişmeyecektir.

$$x^n = x.$$

Boole'ün, yukarıda verdiğimiz toplam üç kuralına dair açıklamaları şöyledir:

Bu kuralların birincisinden, seçmeli sembollerin dağıtıcı olduğu görülmektedir. İkincisinden, değişmeli oldukları görülmektedir: Nicelik simgeleriyle sahip oldukları ortak özelliklerden dolayı, ortak cebirin tüm süreçleri mevcut sisteme uygulanabilir. Üçüncü kuralı, “katsayı kuralı” olarak adlandıracağız. Seçilebilir sembollere özgüdür ve sonuçlarımızı yorum için buluşan formlara indirgememizi sağlamada büyük önem taşıyacaktır (Boole, 1847: 18).

Eserinin “İfade ve Yorumlama” (Of Expression and Interpretation) bölümünde Boole, mantıkta kullanılan basit bir önermenin özelliklerini anlatırken, bir önermenin, özne ve yüklem arasındaki ilişkiden çıkarımının olumlu veya olumsuz olduğunu gösteren içsel yapısına değinir. Elbette “her” ve “bazı” gibi niceleyicilerle birlikte. Boole, “isim-fiil” ya da diğer bir deyişle, varlık belirten yüklem mantıkta tanınan tek fiil olduğunu ekleyerek, tüm diğer yüklemelerin, “olmak” fiili ve bir katılımcı veya sıfat aracılığıyla çözülebilir (1847: 20) olduğunu ifade eder. Bu şekilde ulaştığı dört tür saf kategorik önermenin açılımını verdikten sonra Boole, aynı özelliği barındırmayan sınıfların gösterimini 1 olarak kabul ettiği evrensel sınıf üzerinden cebirsel işlemler yürütmek suretiyle gösterir.

X sınıfı ve *X olmayan* sınıf birlikte evreni oluşturur. Ancak Evren *I*'dir ve *X* sınıfı *x* sembolü ile belirlenir, bu nedenle *X değil* sınıfı *1 - x* sembolü ile belirlenir. Dolayısıyla, belirli bir özneye eklenen *I - x* simgesinin işlevi, içerdiği tüm *X-olmayanları* ondan seçmek olacaktır (1847: 20-1, italik gösterimler bana aittir).

Boole, eserinin devamında başka bir yenilik daha yaparak, tümel evetleme görünümündeki “Tüm *X*'ler, *Y*'dir” tarzında bir önerme için şu ifadeyi kullanır: “Var olan

tüm X 'ler Y sınıfında bulunduğundan, evrenden tüm Y 'leri ve bunlardan tüm X 'leri seçmenin, evrenden tüm X 'leri bir kerede seçmeyle aynı olduğu açıktır" (1847: 21). Bu nedenle,

$$xy = x$$

veya

$$x(1 - y) = 0$$

Gördüğümüz kadarıyla, sözel anlatımının cebirsel ifadesinde ilk defa Boole, 0 sayısını kullanmıştır. Boole attığı bu cesur adımla, Leibniz'in bir adım önüne geçmiştir. "Mantıksal Hesap Üzerine Bir Çalışma" (1690) adlı risalesinin incelenmesi sonucunda görülecektir ki mantıksal hesaplama üzerine kurallar geliştirmeye çalışırken Leibniz, tamamen cebirsel görünümünden faydalanmaktadır fakat burada ne 0 ne de 1 sayılarının notasyon olarak kullanımı mevcuttur. Bununla beraber, Leibniz, kısaltılmış başlığıyla "Explanation of Binary Arithmetic" (İkili Aritmetik Sistemin Açıklaması) (1703) isimli kısa risalesinde ise 0 ve 1 notasyonlarını kesinlikle bir mantık hesabı üzerinden vurgulamamıştır. Bu notasyonlar her ne kadar matematiksel bir yenilik olarak görünse de aslında tamamen Tanrı ve oluşa gelenler hakkında bir açılımın verilebildiği mistik nüveler üzerinde yükselmektedirler. Dolayısıyla, "Leibniz, birçok farklı mantıksal olasılığı sınıflandırmak için önemli bir çalışma yaptı ancak sonuçlarını ifade etmek için uygun bir gösterimden yoksundu" (MacHale, 2014: 110).

Boole, tezimizde ismini *Düşünmenin Yasaları* şeklinde kısaltarak kullanacağımız *Mantık ve Olasılıkların Matematiksel Kuramlarının Dayandığı Düşünce Yasaları Hakkında Bir Araştırma* (An Investigation of the Laws of Thought on Which are Founded the Mathematical Theories of Logic and Probabilites [1854]) adlı eserinde, kendi geliştirdiği cebirsel formdaki mantık sistemini tanıtır. Boole'ün geliştirdiği sistemde, De Morgan'ın *Trigonometri ve İkili Cebir* (Trigonometry and Double Algebra [1849]) adlı eserinde " M ve N büyüklük olabilir ve "+" ikincinin birinciye eklenmesinin işareti; M ve N sayılar olabilir ve "+" birinciyi ikinciyle çarpmanın işareti; M ve N uluslar olabilir ve "+" N 'nin M ile bir savaşa olduğunun işareti" olabilir (1849: 94) tarzındaki biçimsel ifadeleri, esin kaynağı olmuştur. Bu sistem, sembolik mantık ve ileride göreceğimiz üzere hem matematik hem de bilgisayar bilimlerinde çok önemli bir yere sahiptir.

Boole'ün *Düşünmenin Yasaları* adlı eserinde x , y , z gibi hafler, daha önceki eserinde geçtiği tarzda seçilmiş sınıflar için birer işlem tanımlayıcısı değil, tam olarak sınıfların kendisidir. Boole'ün artık tamamen gerçek bir cebirin içinde geçebilecek soruları sorduğu açıktır. Şöyle ki düşünsel etkinlik göstergeleri olarak tanıttığı "+" ve "-" işaretlerini, tümleyici ve ayırıcı olarak kullanan Boole, dilbilimsel bir gösterim olarak "+" işaretini, "ve" ve "veya"

yerine; “-“ işaretini ise “fark” veya “hariç” yerine kullanır. Boole, mantıksal tümdengelim amaçlarına uygun bir şekilde, önermeleri oluşturmak için gerekli olan tek fiilin yani *copulanın* (-dır/dir veya *dırlar/dirler*) sembolik olarak '=' ile ifade edildiğini (Hailperin, 2004: 354) iletir. Böylece, mantıksal çıkarımlar, bu tür önermeler üzerinde cebirsel işlemlerle gerçekleştirilecektir. Boole’ün bundan sonra sorduğu soruya verdiği cebirsel ve basit aritmetiksel işlem cevabı, bizi hem Leibniz’in yüz elli yıl önce bulduğu ikili aritmetik işlemlerin tesisine hem de bu tesisin bilgisayarlar için temelde yatan en elzem ilksel olduğu gerçeğine ulaştırmaktadır: “Gerçek sayılar arasında $x^2 = x$ eşitliğini sağlayan x sayıları hangileridir?” (Boole, 1854: 34). Bu soru, cevabının ancak ve ancak 0 ve 1 olacağı bir sorudur. Boole, şöyle devam eder:

Bu nedenle, mantık simgelerinin genel olarak sayı simgeleriyle biçimsel uyumunun ölçüsünü belirlemek yerine, bunları yalnızca 0 ve 1 değerlerini kabul eden nicelik simgeleriyle karşılaştırmamız bize daha dolaysız önerilir. O halde, x, y, z vb. simgelerinin 0 ve 1 değerlerini ve yalnızca bu değerleri kayıtsızca kabul ettiği bir cebir düşünelim. Böyle bir cebirin yasaları, aksiyomları ve süreçleri, bir mantık cebirinin yasaları, aksiyomları ve süreçleriyle bütün boyutlarıyla aynı olacaktır. Onları sadece yorum farkı ayıracaktır (Boole, 1854: 34-5).

Övgüyü hak eden bir düşünsel etkinliğin sonucu olan çalışmasında Boole’ün, Aristoteles’in kıyas mantığını, kelimelerin anlamlarını düşünmekten vazgeçmeden, doğal dil ile yapılan kıyasları ve geliştirilmiş kıyas formlarını terk ederek, tamamen matematiğe ait bir cebirsel forma indirmediği görülmektedir. Açıklayıcı olması bakımından, Boole’ün ilk çalışmalarında yaptığı, aynı özelliğe ait sınıfların ortak özelliklerini temsil eden cebirsel formundan hareket ederek aşağıda görüleceği üzere, bir çıkarım elde edeceğiz.

$$x=xx \quad (1)$$

$$x-xx=0 \quad (2)$$

böylece,

$$x(1-x)=0 \quad (3)$$

bu durumda,

$$x=0 \text{ veya } x=1. \quad (4)$$

Yukarıda gösterdiğimiz (3) nolu çıkarım dikkatle incelendiğinde, Aristoteles’in “çelişmezlik ilkesi”nin tamamen cebirsel forma kavuşmuş halini görebiliriz. Burada, x sınıfında olan özelliklerin $(1-x)$ sınıfında olamayacağı açıktır. Dolayısıyla bu iki kümenin

çarpımı veya ortak bir özelliği bize ancak 0 sayısını yani imkansızı ya da daha genel bir ifadeyle, olmayan şeylerin kümesini verebilir. Aşağıda göstereceğimiz diğer bir eşitlik olan,

$$x+(1-x)=1 \quad (5)$$

eşitliği ise x sınıfında olan şeyler (veya özellikler) ile x sınıfında olmayan şeyler (veya özellikler) arasındaki toplam ya da başka bir ifadeyle bileşim yani evrenselin genel bir cebirsel formudur.

Boole'ün elde ettiği çıkarımların ilginç sonuçları vardır. İlk olarak, Leibniz'in "Mantıksal Hesap Üzerine Bir Çalışma" (1690) adlı risalesinde, "yeni bir şey eklenmiyorsa, yeni bir sonuç çıkmaz yani aynı şeylerin birbirine eklenmesi bir değişim yaratmaz" (1969: 371-382) şeklindeki mantıksal bir ifadeyi yakalamış olmasıdır. Leibniz'in aynı şeylerin sınıflarının birbirine eklenmesinin yine aynı şeyi vereceğine dair ikinci aksiyomununun, biçimsel olarak " $A \oplus A \infty A$ " (1969: 371-382) gösteriminde, sonradan Swoyer'in günümüz formuna kavuşturduğu " $A \oplus A = A$ " (1994: 4) şekli ile Boole'ün, üçüncü kural olarak tanıttığı, aynı özellikleri taşıyan sınıfların çarpımının yine aynı şeyi vereceğine dair $xx=x$ gösterimi, açıkça aynı şeyi söyleyen bir düşünsel etkinliğin sonucudur. Bu çıkarımlardan doğan bir diğer ilginç sonuç ise Boole, ilk ve son defa tüm mantıksal dedüksiyonun, matematiğin bir branşı olarak kurulabileceğini (Davis, 2018: 30) göstermiştir.

2.2.2.1. Boole Cebiri

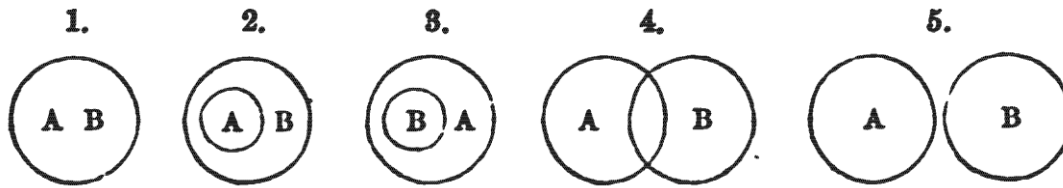
Eklemekte yarar gördüğümüz önemli bir husus şudur ki bugün bildiğimiz haliyle, Boole cebiri, Boole tarafından değil; Boole'den sonra sırasıyla William Stanley Jevons (1835-1882), Peirce ve özellikle Schröder tarafından geliştirilmiş, son halini ise Boole-Schröder adı altında anmamızdan anlaşılacağı üzere Schröder tarafından almıştır. Bu çalışmalara bir ek olarak, John Venn'in (1834-1923) Boole cebiri üzerinde yaptığı çalışmalarının sonucunda elde ettiği diyagramatik gösterimlerinden söz etmek gerekir.

Boole cebirinin bugünkü haliyle bire bir örtüşen diyagramları çizen Venn, (1, 0) ikili gösterimi ile uyumlu olacak şekilde bir sistem geliştirmiştir. Bu sistemde, kümeleri ifade eden A , B , C gibi büyük harf gösterimi, kaç adet kümenin işleme sokulacağını söyler. Bu kümeler üzerinden kaç tür belirlenim gerçekleşebileceğini ise ikili sistem üzerinden 2^n (n , birbirleriyle işleme sokulacak kümelerin toplamı veya tek bir kümenin işleme girecek eleman sayısı olarak ifade edilebilir) temsil eder. Bu sistem, örneğin, birbirinden farklı iki küme olan A ve B kümeleri için 2^2 farklı durum gerçekleşebileceğini rahatlıkla ifade etme gücüne sahip bir kombinasyon sanatıdır. Şayet, iki kümenin birbirinden farklı oldukları bilgisi verilmemiş olsaydı, bu durumda olasılıklarını değerlendirmek zorunda kalarak, bu iki kümenin aynı

olması durumunu da göz önünde bulunduracak ve $2^2 + 1$ farklı durumdan söz ediyor olacaktık.

Şimdi, bu iki sınıfın karşılıklı olarak dahil edilmesi ve dışlanması bakımından birbiriyle kaç tane olası ilişki vardır? Açıkça sadece beş. Buradaki soru için, anladığım kadarıyla: Bir sınıf bilinen durumda ve kapsamı itibarıyla belirlenmiş olarak verildiğinde, başka bir sınıf, birinci sınıfla, bilinen ve benzer şekilde belirlenmiş kaç farklı ilişki içinde olabilir? Yalnızca şu şekilde olabilir: Bir öncekiyle çakışabilir, onu içerebilir, onun tarafından dahil edilebilir, kısmen dahil edilebilir ve kısmen hariç tutulabilir veya tamamen hariç tutulabilir. Terimin bilinen her anlamıyla bunlar farklı ilişkilerdir ve bunlar, muhtemelen bu türden var olabilecek tek farklı ilişkiler gibi görünmektedir (Venn, 1881: 5-6).

Aşağıda, Venn diyagramlarıyla gösterimi verilmiş olan beş durum sıralanmıştır. Bunlar sırasıyla, A ve B sınıflarının çakışık olma, A sınıfının B sınıfı tarafından içerilmesi, B sınıfının A sınıfı tarafından içerilmesi, A ve B sınıfının ortak özellikler barındırması, A sınıfı ile B sınıfının hiçbir ortak özellik barındırmaması durumlarının temsilleridir.



Şekil 2.4 Venn diyagramlarından bir örnek

Kaynak: Venn, 1881: 6

Boole cebirinin, özünde bir küme teorisi üzerinde yükseldiği bilinmektedir. Dikkatle incelendiğinde aşağıda günümüz modern sembolik mantık gösterimi ile birlikte iki küme için verilmiş basit eşleştirmelerden anlaşılacağı üzere, birbirini mükemmel biçimde karşılayan, üç durum görülmektedir (aşağıdaki gösterim [Aksoy, 2015: 171]'den alınmıştır).

Kümelerde	Mantıkta	Boole Cebirinde
$A \cap B$	$P \wedge Q$	$a \cdot b$ ya da $x \cdot y$
$A \cup B$	$P \vee Q$	$a + b$ ya da $x + y$

Filozof ve mantıkçı Peirce, pragmatizmin kurucu babası olarak tanınsa da mantık alanında Boole cebirinin geliştirilmesi yönünde yaptığı katkılarla bilinmektedir. Peirce'e göre, Boole cebiri yalnızca iki değeri olan akla gelebilecek en basit nicelik sistemiydi (2010: 21) ve bu nicelik sisteminin bağıntılar üzerinden formel olarak geliştirilmeye ihtiyacı vardı. Mill'in induktif mantığının tesiri altında kalan Jevons'un, Boole cebirine yapmaya çalıştığı katkı daha ziyade mantıksal toplamın kapsayıcılığı üzerinedir. Boole cebirini tamamen mantıksal bir

gösterime kavuşturmaya çalışan Jevons, *değillemenin* basit bir sembolik temsilini bu sisteme dahil etmiştir fakat Jevons, Boole'un sistemini basitleştirirken onun matematiksel biçimini (Lewis, 1918: 118) bozmuştur. Peirce'ün, Boole cebirini basitleştirme ve geliştirmede sergilediği tutum, Jevon'un tersi yöndedir. Şöyle ki Peirce, bu sistemin matematiksel formunu korurken (Lewis, 1918: 118) örneğin, farklı nitelikleri taşıyan bireyin nitelikleri arasındaki ilişkiselliğe odaklanmış ve Boole'ün matematiksel temellerde yükselen cebirine bazı eklemelerde bulunmuştur. Peirce, Boole'ün sistemine bağıntının “yorumlayıcı”, “içerme” ve “ima etme” gibi dönüşümler geçirdiği tarzda eklemeler yapmıştır.

[Peirce], mantıksal sınıfların (Boole cebirindeki çarpma gibi) daha karakteristik olan ilişkilerini, aritmetik işlemlerle daha yakından bağlantılı olan ilişkilerden (Boole'daki çıkarma ve bölme gibi) ayırt ederek Boole cebirini geliştirdi. Ortaya çıkan cebir, Jevons sistemine göre belirli avantajlara sahiptir çünkü matematiksel yöntemler, geliştirme, dönüştürme, eleme ve çözüm içerir ve mantıksal sınıflar için her zaman yorumlanabilir olan işlemleri ve ilişkileri ayırt ettiği için Boole cebirine göre belirli üstünlükleri vardır (Lewis, 1918: 79).

Boole cebirine ait matematiksel görünüm kazanmış formel dil, doğal dillere göre varlığın bağıntı, kategorizasyon ve anlam durumlarının belirleniminde eksikliydi. Boole cebirinde görülen eksiklik, ancak doğal dillerde görülen bu özelliklerin matematiksel bir form kazanması ve sisteme eklenmesiyle giderilebilirdi. Tüm düşüncenin işaretler aracılığıyla gerçekleştirildiğini ifade eden Peirce (1994: 73), üç tür işarete dikkat çeker:

İlki, söylemin öznesi ile benzerlik ya da analogi sergileyen şematik işaret ya da simgedir; ikincisi, bir işaret zamiri veya bağlantı cümlecisi gibi, amaçlanan belirli nesneye (onu tarif etmeden) dikkati zorlayan göstergedir; üçüncü (veya sembol), işaretlerin isim ile gösterilen karakter arasındaki bir fikir birliği veya alışılmış bağlantı yoluyla nesnesini gösteren genel ad veya tanımdır (Peirce, 1994: 146).

İşaretlerin birbirleriyle bağıntıları sentaks, işaretin temsil ettiği nesnenin anlam üzerinden irdelenmesi semantik ve işaretlerin, kullanan ile yorumlayan üzerindeki etkilerini araştırmak pragmatik alanına girmektedir. Peirce, özellikle Boole cebirini geliştirmek için doğal dillerin en büyük gücü sayılabilecek pragmatik alanına giren unsurları kullanmıştır. Örneğin, “Başarısız olan tüm öğrenciler ya aptaldır ya da tembeldir” (Davis, 2018: 29) gibi bir cümle Boole cebirinde matematiksel forma dönüştürülemez çünkü “Tüm A’lar B’dir” gibi görünen bu örnekte, B’nin aptala ya da tembele referans gösterildiği, A’nın başarısız olan tüm öğrencilere referans gösterildiği yerde, A’nın aptallıktan mı yoksa tembellikten mi başarısız olduğu noktası belirsizdir. Peirce, bu tip bir ayrımı yapmakla ilgilenmiştir ki “bu analize göre

olması gerektiği gibi anlaşılabilir kopulaya, *dahil etme kopulası* denir (Peirce, 1992: 144). Daha anlaşılır olması açısından Peirce’ün kendi örneğiyle devam edilebilir (1992: 165-7):

Sen iyi bir kızsın

Peirce burada bir iddia ile başlar. Aşağıda görüldüğü gibi yazılırsa eğer,

Sen iyi bir kızsın

Sen annene itaat et

bu durumda, ikisi de belirtilmiş olur. Bu, bir ikili önermedir ve önermenin kendisi öne sürülmeden bir şey ileri sürülmek istendiğinde, Peirce “bunu bir ayırıcı şekilde” (1992: 165) yapmayı önerir. Böylece,

Sen iyi bir kızsın

(arzulanan şey bu)

bu defa,

Sen iyi bir kızsın

(fakat bu yanlış)

Peirce’e göre bir “önermenin yanlış olduğuna dair bu son iddia, onun hakkında mantıklı bir ifadedir. Aynı zamanda önermeler hakkında, onları ileri sürmeden söyleme fırsatımız olan en yaygın şeydir” (1992: 165-6). Bu durumda,

Sen iyi bir kızsın

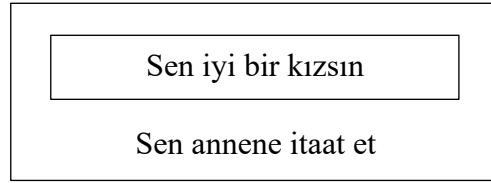
Sen annene itaat et

ikili önermesi “iyi bir kız değilsin ama annene itaat ediyorsun” (1992: 166) şeklindedir ve “bağlayıcı bir önermenin reddi, ayırıcı bir önermedir” (1992: 166). Bu yüzden,

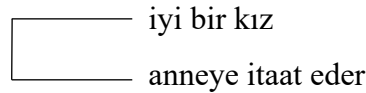
Sen iyi bir kızsın

Sen annene itaat et

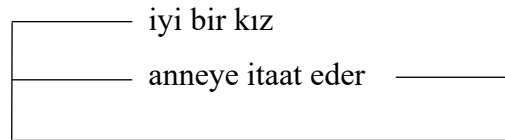
şeklindeki bu ayırıcı önerme, “hem iyi bir kız olduğunu hem de anneye itaat ettiğini reddeden; ya iyi bir kız olmadığını *ya da* (İng. “or”) anneye itaat etmediğini iddia eden ikili önermedir (1992: 166). Öyleyse aşağıdaki durum,



ya iyi bir kız olduğunu ya da anneye itaat etmediğini gösterir, yani "Annene itaat edersen iyi bir kızsındır" (1992: 166). Peirce' e göre bu önerme çifti aslında "koşullu bir önermedir ve ayırıcı önermenin bir türüdür. Şimdi, sistemin açıklamasını tamamlamak için yalnızca küçük bir özellik kalıyor ama bu küçük özellik her şeydir" (1992: 167). Peirce burada yeni bir şey deniyor ve özneleri yani kimlikleri belirgin kılmak için çizgilere başvuruyor. Bu durumda,



yukarıdaki gösterim, "iyi bir kız olan ve anneye itaat eden bir şeyle özdeş olan bir şeyin var olduğu anlamına gelir. Bu, mevcut iyi bir kızın annesine itaat etmesidir" (1992: 167). Peirce daha sonra, varolan bir özneye referans gösterilebilecek yeni bir çizgi daha ekliyor. Böylece,



yukarıdaki gösterim tam olarak şu ifadeye kavuşur: "Mevcut iyi bir kız, annesine itaat eder" (1992: 167).

Peirce, bu şekilde kullandığı iki ya da daha fazla çizginin olduğu durumlarda, sıradan mantığın bilmediği yeni çıkarım biçimlerinin mümkün hale geldiğini yukarıda temsille göstermiştir.

[Peirce] köşelere varlığın kendisini, iki varlığın birbirlerine göre durumlarını ve üçlü bir yapı içinde varlığı bir amaç ile değerlendirmeyi atar. Bu çerçevede yeşillige bakarak ona çayır ya da bitki/ot dememiz bir yorum içerir, dolayısıyla gösterim ve modellememiz seçimlerimize bağlı olarak değişir. Formel yapılarda bunun önemi, bir bağlamda bir nesne birincilik kategorisinde olabilirken, başka bir bağlamda ikincilik kategorisinde olabilir (Yargan, 2017: 700).

Böylece, Peirce'ün Boole cebirinde yaptığı eklemelerin bir ilişkiler mantığı üzerinde gerekçelendiği rahatlıkla ifade edilebilir. Peirce, bağlantılı olma veya olmama durumlarını

çizgisel ifadelerle genişleterek, kategorik yorumlamayı mümkün kılmıştır ancak Peirce'ün Boole cebirine katkıları yalnızca şematik düzen üzerinden değildir. Peirce'ün asıl katkısı, olasılığı dahil ettiği ilişkiler mantığını aktarmak üzere yürüttüğü sembolleştirmelerin uygulandığı alfabenin harfleri tarafından temsil edilen nesnelere veya olayların sınıfının gösteriminde yatmaktadır. “Peirce'ün, sınıflar üzerindeki işlemlere atıfta bulunurken Boole'un kullandığı standart matematiksel semboller ($=$, $+$) yerine, değiştirilmiş sembolleri ($\bar{\quad}$, $+$) kullanarak” (Barnett, 2011: 26) birer mantıksal ilişki sembolleştirmesi yapma girişiminde bulunduğu açıktır. Burada Peirce'ün, mantıksal toplam için “+,” sembolünü, mantıksal çarpım için “,” sembolünü kullandığı görülmektedir. Altında virgül bulunan eşitlik işareti özdeşliği ifade ettiğinde “ $a = b$ ” özdeşliği, a ve b 'nin aynı sınıfı veya aynı bireyler topluluğunu ifade ettiği anlamına gelir (Barnett, 2011: 24).

Sembolik mantıkta bir tür soyutlama çabası sergileyen Peirce'ün sembolik mantığa bir diğer katkısı, sınıflar arasındaki içerilme ilişkisi ve buna karşılık gelen önermeler mantığındaki maddi ima ilişkisini sistematik hale getirmiş olmasında yatmaktadır. Peirce, günümüz gösteriminde “içerilme” olarak tanımlanabilecek \subseteq sembolü ve “eğer...ise” anlamında \rightarrow sembolleri yerine “ $-<$ ” şeklinde tek bir sembol kullanmıştır. Lewis'e göre “ $-<$ ” sembolü aslında “ \leq ” anlamında (1918: 89) kullanılmıştır.

Olasılığa bağlı durumları bir doğrulama yaparak göstermeye çalışan Peirce, ortak öğeleri seçme sınıf işlemi için “çarpma” yerine kullandığı “,” sembolünü “mantıksal hesaplamalar” ve “olasılık hesaplamaları” analogisinde de kullanmıştır. Örnek vermek gerekirse “ a,b ”, a ve b sınıflarının aynı anda içerdiği bireyleri göstermek koşuluyla, a ve b ortak türlere sahiptir, şayet a ve b bağımsız olaylar olsaydı, bu durumda “ a, b ” olasılığı her birinin olasılıklarının ürünü olan bir olayı (Barnett, 2011: 26) göstermiş olurdu. Tıpkı Boole ve Venn gibi Peirce de 0 ve 1 gösterimlerini özdeşliklerinde kullanmıştır: “ $0 + 0 = 0$ ” mümkün olmayan olaylar ve “ $1 + 1 = 1$ ” kesinliğe sahip mümkün olayları temsil etmektedir.

Peirce, ayrıca “İlişkiler Mantığı” (The Logic of Relatives [1870]) adlı çalışmasının da bulunduğu editörü olduğu 1883 basımlı kitapta, 0 ve 1 arasında değişen bazı eşitsizlik formlarını temsil edebilecek gösterimler için “evrensel niceleyeci” yerine Π ve “bazı” ya da “varlıksal bildirim”i temsilen Σ sembollerini Boole cebirine yerleştirme girişiminde bulunmuştur. Peirce'e göre bu gösterimler eşliğinde herhangi bir önerme, “bu tür sayısal katsayıların bileşimlerinin ve çarpımlarının sıfırdan büyük olduğunu söylemeye eşdeğerdir” (1983: 271). Böylece bu tür önermelerin *varlığına* işaret edilmiş olur. Peirce'ün nicelemelerine örnek vermeden önce, L 'nin seven kişiyi, I 'nin, Isaac ve J 'nin Jessica'yı temsil ettiği yerde, Lij gibi bir gösterim, Isaac'ın Jessica'yı sevdiğine (Kenny, 2007: 37) dair bir

ilişki sentaksı verir. Peirce'ün kendi örneğinden aktardığımız aşağıdaki gösterimlerde, l seven kişiyi veya seven şeyi, b velinimeti (veya yardımsever kişiyi) ya da şeyi temsil etmek üzere (aşağıdaki gösterimler [Peirce, 1983: 272-3]'den alınmıştır):

$\Sigma_i \Sigma_j l_{ij} > 0$ eşitsizliği, bir şeyin bir şeyi sevdiğini gösterirken,

$\Pi_i \Sigma_j l_{ij} > 0$ eşitsizliği, her şeyin bir şeyi sevdiğini gösterir.

$\Pi_i \Sigma_j (l_{ij}(b)_{ji}) > 0$ eşitsizliği, her şeyin kendi velinimetini sevdiğini gösterirken,

$\Sigma_i \Sigma_k \Pi_j (l_{ij} + b_{jk})$ gösterimi, kendi velinimetleri dışında her şeyi sevmek ilişkisinde bir şeye karşılık gelen bir şey olduğu anlamına gelir.

Peirce, aynı çalışmada bireysel ikili ilişkileri “A : A” ve “A : B” olmak üzere ikiye ayırır. Bunlardan ilki kendi kendisiyle bir nesne çiftini, ikincisi de bir yabancı ile ilişkilendirmede belirlenim kazanan bir nesne çiftini temsilen gösterilir. Buna göre, her nesneyi kendisiyle ve diğerleriyle eşleştiren yalnızca bir ilişki vardır ve tüm çiftlerin toplamı olarak, doğal dile “birlikte var olmak” anlamında çevrilebilecek ∞ ile gösterilir (Peirce, 1983: 262). Peirce'ün çalışmasında “(A : A) + (B : B) + (C : C) + ...” gibi bir gösterim, doğal dilde “kendisiyle özdeş” anlamında 1 ile temsil edilir.

Yukarıda görülen ilişkiler mantığına dair bazı örnekler ve Peirce'ün orijinal şemalarının bir benzerini çizmeye çalıştığımız şema örnekleri, Boole cebirini geliştirme çabaları olarak görülmelidir. Peirce, tamamen kendisine ait bir sistem geliştirmemiştir. Bir mantıkçı olmanın dışında bir bilim insanı da olan Peirce, yine de kendi adıyla anılmasını hak ettiği, tezimizin üçüncü bölümünde değineceğimiz NOR (VEYADEĞİL) mantık kapılarının mucididir. “Buradaki temel fikir, "ve" ve "veya" gibi mantıksal işlevlerin mantık kapıları tarafından elektriksel olarak simüle” (Gensler, 2017: 466) edilebilmesi üzerinedir. Peirce, mantıksal işlemlerin doğrulukla yürütülebildiği mantık kapılarının, elektrik anahtar devrelerinde uygulanabileceğine dair düşüncelerini yayımlamadığından dolayı, bu icat 1938 [1936] yılında Claude Shannon tarafından (Gensler, 2017: 466) yeniden keşfedilmiştir.

İlişkiler mantığı, De Morgan'dan sonra Peirce tarafından ortaya atılmış ve “Schröder tarafından, Peirce'ün yaptığıyla uyumlu olarak kodlanmıştır” (Heijenoort, 1967: 240). Schröder, Peirce'ün yöntemlerini biçimsel olarak kesinleştirerek Boole cebirinin sentaktik kurulumunu belirgin hale getirmiştir. Bir örnekle açıklamak gerekirse, 1^1 gibi bir ifadenin, bireylerin evreni veya birinci seviyeden bir söylem evreninin temsili olduğu; birbirlerinden ve sıfırdan farklı A, B, C gibi büyük harflerin, elementleri veya bireyleri temsil ettiği yerde (Lewis, 1918: 117), Peirce'ün

“(A : A) + (B : B) + (C : C) + ...” gösteriminin 1 ile temsil edildiği çalışması, Schröder'de

$1^1 = A+B+C+\dots$ ile gösterilir (Lewis, 1918: 117).

Schröder, biçimsel türetmelerini bu yolla sürdürerek Peirce’ün “varlıksal niceleyici” Σ ve “evrensel niceleyici” Π sembolleri ile birlikte, A, B, C gibi elementlerin birer temsili olan i, j, k, l gibi harflerle, tekil söylem evreni 1^1 ve ikili ilişkilerin söylem evreni 1^2 (söylem evrenleri üçlü dördü vb. çoğaltılabilir) ile temsil edildiği aşağıdaki biçimleri elde etmiştir (aşağıdaki alıntı [Lewis, 1918: 117-8]’den alınmıştır):

$$1^1 = \sum_i i$$

$$1^2 = \sum_j \sum_i (i : j) = \sum_i \sum_j (i : j) = \sum_{ij} (i : j)$$

Lewis’e göre, Peirce’ün cebirsel mantık çalışmalarından (veya hesap ya da calculus) hareket eden Schröder’in Boole cebirine en önemli katkısı, iki boyutlu bireysel ikili ilişkilerin bir dizisini matrisler formunda göstermiş olmasıdır (1918: 120). Schröder’in matris formu aşağıda gösterildiği gibidir (aşağıdaki alıntı [Lewis, 1918: 120]’den alınmıştır).

$$\begin{aligned} 1 = \sum_{ij} i : j = & A : A + A : B + A : C + \dots \\ & + B : A + B : B + B : C + \dots \\ & + C : A + C : B + C : C + \dots \\ & + \dots \dots \dots \end{aligned}$$

Görüleceği üzere Schröder, yalnızca mantık cebiri üzerinde değil, ayrıca ilişkiler cebiri üzerinde de ciddi biçimselleştirmeler yapmıştır. Jevons ve Peirce’ün Boole cebirine yaptıkları eklemeleri kendi sistemiyle bütünleştiren Schröder, nihayetinde “Boole’ün sisteminden daha basit ama daha güçlü bir cebir” (Lewis, 1918: 118) geliştirmiştir. Böylece De Morgan ile başlayan, Peirce ile belirginlik kazanan ilişkiler cebiri ve Boole ile başlayan mantık cebiri, Schröder’de, mantıksal olarak mükemmel bir cebirsel forma evrilmiştir ve Venn, sunduğu diyagramları sayesinde Boole cebirinin kümeler teorisi üzerinden yükselişine dair sağlam bir temellendirme sunmuştur.

2.2.2.2. Boole Cebiri Bir *Calculus Ratiocinator* mıdır?

Daha önce de değindiğimiz üzere, Leibniz, Aristoteles mantığıyla ilgilenirken, tüm çıkarımların bir aritmetik kalkülüs aracılığıyla yapılabileceği yeni bir tür mantıkla da ilgilenir duruma gelmiş ve bu yeni mantığı, öznenin mefhumunda içerilen yüklem(ler) üzerinden yürütülebilecek şekilde organize etmeye çalışmıştır. Leibniz’in önerdiği bu yeni mantıkta, bir *calculus ratiocinator* aracılığıyla tüm çıkarımlar, bir aritmetik kalkülüs ile dedüksiyona dayalı

olarak yapılabilecekti. Bu anlamda Leibniz'in hayaline yaklaşan ilk mantık sistemi Boole'e aittir. "Boole, yalnızca bilinmeyenlerin ortadan kaldırılması ve hesaplanması için mantıksal problemlerin sistematik bir şekilde çözülmesine izin veren, cebire benzer bir teknik geliştirmeye" (Peckhaus, 2004: 10) çalışmıştır. De Morgan, Boole, Peirce ve Schröder ile sağlam bir mantık sistemine dönüşen Boole cebiri, bir *calculus ratiocinator* görünümü sergilemektedir. "Bununla birlikte, Schröder hiçbir zaman tam bir *calculus ratiocinator* geliştirme" (Millán, 2021: 440) çabasında olmamıştır.

Günümüzde genellikle "sembolik mantık" adıyla anlaşılan ve -adı olmasa da- esas olarak Boole'a dayandırılan şeyin, Leibniz'in "calculus ratiocinator" dediği şey olduğu ve "evrensel karakteristik" in yalnızca bir parçası olduğu unutulmamalıdır. Sembolik mantıkta Leibniz, şimdi mantıksal çarpma, toplama, olumsuzlama, özdeşlik, sınıf dahil etme ve boş-sınıf dediğimiz şeyin temel özelliklerini dile getirdi fakat Leibniz'in araştırmalarının amacı, söylediği gibi, aklın tüm doğrularının bir hesaba indirgenmesi gereken bir tür genel notasyon sistemi yaratmaktı (Couturat, 2004: ii).

Boole'ün mantık sisteminde, önermelerin salt doğruluk değerine indirgenebildiği bir hesaplayıcı (Peckhaus, 2004: 4) mevcuttur ve bu sistemde doğruluk değerleri ancak ve ancak 0 ya da 1 değerini alabilir. Bu haliyle, Leibniz'in *calculus ratiocinator* hayali, Boole cebirinde görünürlük kazanmış olsa da yalnızca kısıtlı bir doğruluk evreninde kendini göstermiş olur.

2.2.3. Friedrich Ludwig Gottlob Frege (1848-1925)

Bir mantıkçı, matematikçi ve filozof olan Frege, Boole'ün mantık sistemini eksikli bulmuş fakat bu sistemi geliştirmek yerine, tamamen yeni sayılabilecek bir mantık sistemi üzerinde yoğunlaşmıştır. Boole, mantığın matematiğin bir branşı olarak görülmesi ve geliştirilmesi yönünde düşüncelerine paralel düşen bir mantık sistemi geliştirmişken, Frege mantığın doğal diller ile bağlantısını kabul etmiş, bu manada matematiği, dilsel bağlantıların kullanılabileceği doğru temellerde yükselen iyi belirlenmiş bir mantığa dayandırma girişiminde bulunmuştur. Boole'ün sıradan cebirden yola çıktığı ve mantıksal ilişkileri yine cebirsel semboller aracılığıyla belirlediği mantığında, başka önermeler arasındaki ilişkiyi bildiren önermeler, "ikincil önermeler" olarak tanımlıdır. "Frege, önermeleri birbirine bağlayan aynı ilişkilerin, bireysel önermelerin yapısını çözümlmek için de kullanılabileceğini görmüş ve bu ilişkileri mantığının temeli" (Davis, 2018: 37) yapmıştır.

Peirce'ün Boole cebirinde yapmaya çalıştığı ekleme de ilişkiler üzerinedir fakat başlangıç noktasını, ilişkileri mantığının temeline almak olarak belirleyen Frege sayesinde, matematikte ve mantıkta bir daha geriye dönülmeyecek derecede bir biçimsellik adımı atılmıştır. Peirce, kaleme aldığı önemli çalışmalarını yayımlamamak veya uzun süre

bekletmekle ünlenmiş biridir. Amerika’da kaleme aldığı “General Algebra of Logic” [Mantığın Genel Cebiri (1884)] adlı çalışmasında Peirce’ün, bireysel değişkenlerin kapsayıcılık ilişkisini tanıtmış ve niceleyiciler olarak genelleştirilmiş toplam ve çarpımı yorumlamasından sonra (Millán, 2021: 440) ilişkiler ve bağıntılar ekleyerek geliştirdiği Boole cebirinden habersiz Frege³³, 1879 yılında, “Boole mantığı ile karşılaştırılabilir ifade gücüne sahip” (Millán, 2021: 440), doğal dillerin içinde barındırdığı mantıksal ilişkileri açıkça ortaya koyabileceği bir sembolizmi tasarladığı (Kenny, 2007: 38) *Kavram-Yazısı* (Begriffsschrift) adlı eserini yayımlar. Bu eser, Boole cebirinden farklı olarak, doğal dilden hareketle doğal dilde kurulması mümkün herhangi bir cümlenin belirsizliğini ortadan kaldırmaya dönük bir sembolleştirme içermektedir.

[Ö]ncelikle sıralama kavramını belli ölçüde mantıksal dizinlenime dayandırmaya çalışmam gerçekleşmiştir. Bu esnada fark edilmeden araya herhangi bir somut şeyin girmemesi için, her şeyin, çıkarım zincirinin bütünlüğüne bağlı olması ve herhangi bir boşluk barındırmaması önemliydi. Bu şartı en katı şekilde yerine getirmeye çalışırken, ifadede oluşan tüm ağır anlaşılabilirliği, aslında amacımın gerektirdiği şekilde, bağıntılar ne kadar karmaşık olursa bir o kadar az isabetli ortaya koyan dilin yetersizliği engeliyle karşılaştım. Bu çalışmanın konusu olan *Kavram-Yazısı* (Begriffsschrift) düşüncesi de bu gereksinimden ortaya çıkmıştır (Frege, 2019a: 72, italik bana ait).

Frege’nin bu çalışması, döneminde "belki de mantıkla yazılmış en önemli tek eser" (Davis, 2018: 37) övgülerine layık görülmüş olup, ileride tekrar değineceğimiz üzere, bu eserde tüm ilkel ve modern bilgisayar dillerinin temelinde yatan bir sözdizimsel yapı mevcuttur. Peki Frege’nin mantıkta ve matematikte üstesinden gelmeyi başardığı şey nedir? Bir sözdizimi üretmek mi? Öyle ise bu soruların cevabını arayacağımız *Kavram-Yazısı*’nı irdeleyerek sorgulamamızı yürütmek gerekir.

Frege’nin *Kavram-Yazısı* adlı eserinde, Boole’ün aksine, matematik alanına ait tüm tümdengelimli çıkarımların belirlenim kazanacağı, rahatlıkla ifade edilebileceği bir mantık sistemi öngörülür. Bu sistem, dilin belirsizliklerinden azade olabilmeli ve her matematiksel ifade (bir kavram veya nesne de olabilir) tamamen biçimsel olarak verilebilmelidir. Burada önemli bir husus şudur ki Frege, mantığı matematiğin üzerinde bir etkinlik olarak görmektedir dolayısıyla matematik, mantığın işlevsel bir kolu olarak mantık temelleri üzerinden

³³ Frege’nin, *Kavram-Yazısı* ile Peirce’ün ilişkiler ve bağıntılar eklemek suretiyle geliştirdiği Boole cebirine eşit bir sistem ortaya koyduğunu düşünen Schröder, Frege’yi Boole cebirine ekleme yapmak ve başkaca bir yenilik barındırmamakla itham eder. Bu ithamla birlikte, Frege ve Schröder arasındaki polemik konferanslarda yaptıkları konuşmalara kadar yansımıştır. Günümüz araştırmaları gösteriyor ki bu polemik arka planında, *lingua characteristica* ve *calculus ratiocinator* kavramlarına *Kavram-Yazısı*’nın mı yoksa Boole cebirinin mi daha yakınsak olduğu tartışması yatmaktadır. Bu polemik altında yatan Leibniz’in ideal diline yakınsama meselesine 2.2.3.1. altbaşlığında kısaca değinilecektir.

yürütülebilmelidir. Doğal dillerin anlam belirsizliklerine maruz bırakılmadan oluşturulması gereken bu yeni sistemde Frege, *özne* ve *yüklem* kavramlarının, *argüman* ve *fonksiyon* ile değiştirilmesini (2019a: 25) önerir. Öncelikli amacı *Kavram-Yazısı*'sını aritmetiğe uygulamak (2019a: 26) olan Frege, şöyle devam eder:

Bir içeriğin kavranmasının, bir argümanın fonksiyonu olarak kavramsallaştırılışının, ne ve nasıl bir etkisi olduğu kolayca görülmektedir. Bunun da ötesinde, örneğin “şayet/eğer” (wenn), “ve” (und), “değil” (nicht), “vardır” (es gibt), “bazı/birkaç” (einige), “hepsi/tamamı/tümü” (alle) vs. gibi kelimelerin gönderimleri arasındaki bağıntıyı kanıtlamanın, dikkatleri çekmesi umudunu taşıyorum (Frege, 2019a: 25).

Frege'ye göre mantık, kendisinin teklif ettiği matematiksel dönüşüme kadar, dil ve dilbilgisine yakın durmuştur. Frege'nin “Fonksiyon ve Kavram” adlı çalışmasından bir örnek olan “Sezar Galya'yı fethetti” önermesi bir fonksiyon olarak değerlendirilmek suretiyle, öncelikli olarak “Sezar” ve “Galya'yı fethetti” olarak ayrımlanır (2019b: 34). “... Galya'yı fethetti” cümlecigi Frege'nin kendi deyimiyle bir “doymamış bölüm”dür ve gönderim yaptığı bir argüman vardır; bu argüman *Sezar*'dır. Frege'nin verdiği başka bir örnekte ise “x'in başkenti”ni bir fonksiyon ifadesi şeklinde (2019b: 35) almaktadır. Bu örnekte seçilen argüman “Alman imparatorluğu” olursa, bu durumda *Berlin*, fonksiyonun değeri olarak karşımıza çıkacaktır. Görüleceği üzere, doğal dil ifadelerinin doğruluk değeri, argüman ile belirlenim kazanan fonksiyonlarla daha rahat bulunabilecek elastik bir sisteme dayandırılmaktadır. Bir önermenin doğruluk değeri, “ve”, “ise” ve “veya” gibi bağlaçların bağlantılı olduğu cümlecikler veya bu tip bağlaçlar yoksa bile özne ve yüklem, bu tip bağlaçların eklenmesi yoluyla ayrımlanması neticesinde ortaya çıkmaktadır.

Açıkça görülmektedir ki Frege'nin biçimselleştirmeye çalıştığı mantık, önermeler mantığıdır. Mantığın bir dalı olan önermeler hesabı, bir bütün olarak tümcelere uygulandığında olumsuzlama, birleşme, ayrılma vb. etkisine bağlı çıkarımlarla ilgilenir (Kenny, 2007: 38). Önermeler mantığı, Aristoteles'ten sonra Stoacılar tarafından çalışılmış ve geliştirilmiştir fakat ilk sistematik formülasyonu Frege yapmıştır (Kenny, 2007: 38). Tamamen biçimsel bir dizgenin peşinde olan Frege'nin yeni mantık sisteminin, *argüman* ve *fonksiyon* dışında bir de dilsel bağlaçların yerine geçecek sabit imlere (simgelere) ihtiyacı vardır. Bu simgeler, Frege'nin Aristoteles'te gördüğü, “her”, “bazı”, “değil” ve Stoacılardan aldığı Stoacı mantıkta kullanılan “ve”, “veya”, “ise” (Gensler, 2017: 481) gibi bağlaçların birer kombinasyonunu yapmamıza olanak verecek sembol manipülasyonudurlar. Frege'nin *gotik* çizgilerden oluşan biçimsel dizgesinde “bazı” ve “ve” için özel veya basitleştirilmiş

notasyonlar bulunmamaktadır. Aşağıda, Frege'nin kendi notasyonlarından bazı örnekler, açıklamalarıyla verilmiştir.

┌—

imi, tüm yargılar için geçerli olan ortak yüklemdir (Frege, 2019a: 32). Burada önemli olan ayrıca içeriktir, “örneğin bir 'ev' fikrinin olamayacağı gibi. Bu nedenle, yargılanabilir ve yargılanamaz içerikleri birbirinden” (Frege, 2019a: 31) ayırtırmak gerekir.

— A

imi, bir yargıyı ifade etmez (Frege, 2019a: 30).

└— A

imi, “değil A” için kullanılmaktadır.

┌— A
└— B

Yukarıdaki gösterimde, “iki yatay çizgiyi birbirine bağlayan dikey çizgi, koşulluk-çizgisi/imi olarak adlandırılabilir” (Frege, 2019a: 36). Böyle bir imleme bize “B ise A” koşullu yargısını ifade etmektedir. Bu notasyon aşağı doğru genişletilebilir özelliktedir.

—X—

imi, her x için geçerli bir yargı verme durumunun (Frege, 2019a: 53) gerçekleştirilebileceğini ifade etmek için kullanılmaktadır.

┌—X—┐
└— Bx
└— Ax

Yukarıdaki gösterim, “her A, B değildir” ifadesini sembolize etmektedir (bu örnek [Gensler, 2017: 482]’den alınmıştır). Frege’nin dizgesindeki bu gösterim, aynı zamanda “bazı A’lar B’dir” anlamını da taşımaktadır.

İcat ettiği notasyonlar sayesinde Frege, doğal dildeki çıkarımları rahatlıkla sembolize edebileceği bir dizge ortaya koymuş olsa da Frege’nin notasyonları kullanım açısından zorluk içeren bir dizgeye sahiptir ve bu nedenle günümüzde hiç kullanılmamaktadır. Bu notasyonların kullanım zorluğuna rağmen, *Kavram-Yazısı*, biçimsel mantığın “Hiçkimse her şeyi bilemez” türünde cümlelerle başa çıkmasını (Kenny, 2007: 39) sağlamıştır.

Frege’nin önermeler mantığı üzerinde çalışırken, mantığa yaptığı büyük katkı önermeler hesabının ötesinde, yüklemeler hesabı üzerine çalışmalarından kaynaklanmaktadır (Kenny, 2007: 39). Yüklemeler mantığı, bir bütün olarak ele alınan önermelerden ziyade önermelerin iç yapısıyla ilgilenen mantığın dalıdır. Mantık üzerine yapılan tüm bu çalışmalar sonucunda elde etme çabasında olunan biçimsel dizgelerin, matematikten ayrı düşünülemeyeceği aşikardır. Frege’nin bu konuda çok daha ilerisini gördüğünü eklemek gerekir.

Kavram-Yazısı, mantık tarihinde klasik bir metin olmasına rağmen Frege’nin yazma amacı mantıktan çok matematik ile ilgiliydi. Biçimsel bir mantık sistemi kadar biçimsel bir aritmetik dizgesi sunmak istedi ve daha önemlisi, iki dizgenin yakından ilişkili olduğunu göstermek istedi. Aritmetiğin tüm doğrularının, herhangi bir fazladan desteğe ihtiyaç duymadan mantığın doğrularından çıktığının gösterilebileceğini iddia etti (Kenny, 2007: 39).

Matematiğin temellerinin sorgulanmaya başladığı on dokuzuncu yüzyılda, bu alanda gerçekleştirilmiş tüm başarılarla rağmen, sayılar ve kümeler teorisi üzerinde yükselen matematikte sayılar, hala gerçek bir kavramsallığa ulaşamamıştı. Matematik alanında çalışmaların ivme kazandığı, Öklid dışı geometrilerin ve bir reel, bir de imajiner kısımdan oluşan $z=a+ib$ gibi karmaşık sayıların kavramsallaştırılma sorunlarının ortaya çıktığı on dokuzuncu yüzyıl ortalarından itibaren, matematik nesnelerin varlıkları bariz bir araştırma konusu haline gelmişti. On yedinci yüzyılın mekanik dünya görüşünün temellendirildiği ve açıklanabildiği doğanın dili matematik, on dokuzuncu yüzyıl ortalarından itibaren ardı ardına yapılan çalışmalarda göze çarpan fizik dünya ile bağdaşımsızlık ve kendi içinde anlaşmazlıklar içerir duruma gelmişti. Bu sorunlardan bir diğeri, Georg Cantor’a (1845-1918) atfedilen “sezgisel küme kuramı” (İng. naive set theory) olarak bilinen ve sayılabilir (veya sayılamaz) sonsuz kümelerin elemanları arasında yapılabilecek bir eşleştirmeyi sağlayan kuramdı. Bu kurama göre, her şey bir kümeye dönüştürülebilirdi. Öyle ki sayılamaz sonsuz

bir küme olan $(0, 1)$ açık aralığındaki elemanlar rahatlıkla Reel sayı kümesi \mathbb{R} ile bire bir eşleştirilebilir ve bu iki kümenin sonsuzluklarının aynı olduğu söylenebilirdi.

Cantor'dan önce “sonsuzluk” kavramı matematikte sadece “sonlu”nun karşıtı olarak bilinirdi, oysa “sonlu”nun bile tam matematiksel bir tanımı yoktu. Cantor sonsuzluk kavramına gerçek boyutunu kazandırmıştır: Sonsuzlukları derecelendirmiş, onları bir nevi sayı olarak görmemizi sağlamıştır. Doğal sayıların (\mathbb{N}) sonsuzluğunun kesirli sayıların (\mathbb{Q}) sonsuzluğuna eşit olduğunu, gerçel sayıların (\mathbb{R}) sonsuzluğunun doğal sayıların sonsuzluğundan büyük olduğunu ve \mathbb{R}^n kümesinin sonsuzluğunun \mathbb{R} kümesinin sonsuzluğuna eşit olduğunu kanıtlamıştır (Nesin, 2010: 203).

Richard Dedekind (1831-1916) ile birlikte sayılar teorisi üzerine çalışarak, sayıların doğası üzerinde analizlerde bulunan Cantor, kümeler kuramına yönelmişti. Elbette, *sonsuzluk* matematik için bile oldukça zorlayıcı bir kavramdı, kaldı ki Cantor farklı sonsuzluk dereceleri de belirleyerek sonsuzluklar arasında büyüklükler sıralamıştı. Örneğin, sayılabilir veya sayılamaz sonsuz kümelerden herhangi birinin tüm alt kümelerinden oluşan yeni bir küme oluşturmak istenildiğinde, elde edilecek olan yeni sonsuzluğun orijinal kümenin sonsuzluğundan daha büyük bir sonsuzluk olacağını göstermişti. Bu sonsuzluklar arasındaki büyüklük sıralamasını *kardinal sayılar*³⁴ (sonluötesi sayılar) temsil ediyordu. Sonsuz ya da sonsuzluklar üzerine çalışmalarından dolayı Cantor, oldukça fazla karşıt görüşle boğuşmak zorunda kalmıştı. Kolay kabul edilebilir bir kuram değildi ve Cantor bir kısmını ispatlamış olmasına rağmen, bu kuram paradokslara açık görünüyordu.

Böylece matematik üzerine bir epistemolojik tutum sergilenerek, matematiğin nesnelere belki de ilk defa birçok açıdan tartışmaya açılmıştı. Bu tartışma matematiğe, “matematikte hızlanan *saflaşma eğilimi*” (Öztürk, 2017: 3) olarak yansımıştı. Öklid geometrisinin aksiyomatik yapısına benzer bir aksiyomatik yapıyı, matematik için de belirlemek matematiği daha sağlam bir zemine çekmek demektir. Tıpkı Frege gibi Peano ve Dedekind tam da bu aksiyomatik sistemin peşindeydiler.

Dedekind ile birlikte yürüttükleri negatif ve irrasyonel sayıların temellendirilmesi yönünde önemli çalışmaların dışında, doğal sayılar kümesinin aksiyomatikleştirilmesi çalışması, bugün sıklıkla Peano'ya atfedilen Peano Aksiyomlarının ortaya çıkmasını sağladı. Elbette negatif ve irrasyonel sayıların aksiyomatik bir görünüme kavuşması, öncelikle doğal

³⁴ “Bu kümelerin öge sayısı (teknik adıyla kardinal sayısı veya sayal sayısı) \aleph_0 (alef sıfır) ile gösterilir. \aleph_0 en küçük sonsuz kardinal sayısı olarak kabul edilir. Kesirli sayılar, doğal sayılar ve tam sayıların öge sayısı budur. Bu konu, kümelerin öğelerinin sayılabilirliği konusuyla da ilişkilidir. Matematikte bir küme eğer sonlu sayıda öğeden oluşuyorsa veya sonsuz sayıda öğeden oluşup bu öğelerin sayısı doğal sayılar kümesinin öge sayısı olan \aleph_0 sayısına eşitse, bu kümenin öğeleri sayılabilir; başka bir deyişle, numaralandırılabilir olarak kabul edilir” (Öztürk, 2011: 48).

sayılar gibi, nispeten daha kolay anlaşılır bir kümenin aksiyomatikleştirilmesini gerektiriyordu. Kullanışlı notasyonlar içeren ideografik dil yapısına sahip *Aritmetik İlkeler* (Arithmetices Principia: Nova Methodo Exposita [1889]) adlı eserinde, Peano bunu başarmıştı. “Muhtemelen Dedekind’in 1888 tarihli çalışmasından etkilenmişti fakat bu fikirlere kendi eğilimini katmıştı” (Kline, 1972c: 988).

Gerçekten de Dedekind’in başlangıçta dört aksiyomla belirlediği doğal sayılar kümesi, belirlenim eksikliği taşımaktadır yani aynı özellikler başka bir kümeye de uygulanabilir. Bu nedenle doğru bir gönderme için eksikli kalır. Dolayısıyla Dedekind’in “bu sorunu aşmak amacıyla bulduğu çözüm, matematiksel tümevarım ilkesi olarak bilinen (...) ilkeyi, bu defa doğal sayılar kümesini tanımlamak amacıyla” yürütmesidir (Öztürk, 2011: 45). Dedekind-Peano aksiyomları olarak da bilinen bu aksiyomlar 1889 yılında Dedekind’den sonra, öncelikli olarak sekiz aksiyom ve bir tümevarım kuralı şeklinde Peano tarafından aşağıda görüldüğü şekilde verilmiştir³⁵ (gösterim [Peano, (1889)1973a: 123]’ten alınmıştır).

1. $1 \in \mathbb{N}$.
2. $a \in \mathbb{N} \cdot \mathcal{D} \cdot a = a$.
3. $a, b \in \mathbb{N} \cdot \mathcal{D} : a = b \cdot = \cdot b = a$.
4. $a, b, c \in \mathbb{N} \cdot \mathcal{D} \cdot \therefore a = b \cdot b = c : \mathcal{D} \cdot a = c$.
5. $a = b \cdot b \in \mathbb{N} : \mathcal{D} \cdot a \in \mathbb{N}$.
6. $a \in \mathbb{N} \cdot \mathcal{D} \cdot a + 1 \in \mathbb{N}$.
7. $a, b \in \mathbb{N} \cdot \mathcal{D} : a = b \cdot = \cdot a + 1 = b + 1$.
8. $a \in \mathbb{N} \cdot \mathcal{D} \cdot a + 1 - = 1$.
9. $k \in \mathbb{K} \cdot \therefore 1 \in k \cdot \therefore x \in \mathbb{N} \cdot x \in k : \mathcal{D}_x \cdot x + 1 \in k : : \mathcal{D} \cdot \mathbb{N} \supset k$.

Boole ve Schröder’in çalışmalarını kendi çalışmaları içinde eriten (Heijenoort, 1967: 83) Peano’nun notasyonu, kullanım kolaylığı açısından oldukça kayda değerdir. Mantıksal gösterimde bir kümenin elemanı olmayı ε simgesi ile gösteren Peano, “bazı” anlamında kullanılan \exists ile günümüz notasyonlarında halen canlılığını korumaktadır. “Bugün Peano’nun kitapçığını okumaktaki kolaylık, onun notasyonunun ne kadarının doğrudan ya da çok az değiştirilmiş bir biçimde çağdaş mantığa girdiğini gösteriyor” (Heijenoort 1967: 84). Yukarıda sıralanmış gösterimlerin, yalın ve doğal dilde anlaşılır ifadeleri, günümüzde

³⁵ Peano’nun orijinal gösterimi, dikkatle incelenecek olursa, yukarıdaki gösterimde 0 sayısı yerine 1 sayısından başlatılan doğal sayılar kümesi, üniversitelerde matematik bölümlerinde de aynı şekilde kullanılmaktadır. Doğal sayılar kümesi, 1 sayısı ile başlatılarak \mathbb{N}^+ ile gösterilirken bu kümenin 0 ile başladığını ifade etmek için genellikle $\mathbb{N}^+ + \{0\} = \mathbb{N}$ gösteriminden yararlanılmaktadır.

bilindiği şekliyle beş aksiyomun Peano tarafından geliştirilmiş hali, Russell tarafından aşağıda verildiği şekilde yeniden düzenlenmiştir (Russell, 1920: 10-1):³⁶

- (1) 0 bir sayıdır.
- (2) Her sayının ardılı, bir sayıdır.
- (3) Aynı ardıla sahip iki sayı yoktur.
- (4) 0 hiçbir doğal sayının ardılı değildir.
- (5) 0 sayısına ait bir özellik, her sayının ardılına da ait bir özelliktir; böylece tüm sayılara aittir.

Bertrand Russell'a (1872-1970) göre bu dizgede, "0", "sayı", ve "ardıl" gibi tanımlanmamış kavramlar kullanılmıştır. Dedekind için kullanılan sembollerin veya kavramların tanımlarından veya anlamlarından ziyade onların nasıl kullanıldığı önem teşkil etmektedir. Bu anlamda Dedekind, Frege ile başlamış olan mantıksalci tutumda bir mantık sistemi değil, sadece sembolik mantığa dayalı bir sistem geliştirme yönünde çalışmıştır. Peano, mantıksalci tutuma daha yakın dizgeler üretme çabasında olsa da mantığın matematik için sadece bir araç olduğu yönündeki görüşüyle, Frege'den ayrılmaktadır.

Peano'nun çalışması, sembolik mantığın daha da gelişmesini ve daha sonra Frege ve Russell'in mantık üzerine matematik inşa etme hareketini etkilemiş olsa da çalışmaları Frege ve Russell'in çalışmalarından ayırt edilmelidir. Peano, matematiği mantık üzerine inşa etmek istemiyordu. Ona göre mantık matematiğin hizmetkârıydı (Kline, 1972c: 988).

Peano ve arkadaşları tarafından, mantıksal çıkarımlar aracılığıyla matematiğin tüm dallarının aksiyomatikleştirilmesi ve bu şekilde matematiğin temellerinin kurulması yönünde çalışmaların seğılendiği ilk cildi 1892 yılında olmak üzere, beş cilt halinde yayımlanmış *Formüller*'i (Formulaire de Mathematiques [1892-1895]) bir "evrensel logistic dilinde" (Lewis, 1918: 5) ifade edilmiştir ve "bu çalışmada sembolik mantık ve logistic bir kez daha bir araya getirilmiştir" (Lewis, 1918: 5). Basit bir ifadeyle tanımlamak gerekirse, *logistic* kavramı, yaygın olarak sembolik mantığı ve bu mantığın yöntemlerinin diğer sembolik prosedürlere uygulanmasını birlikte belirtmek için (Lewis, 1918: 9) kullanılmaktadır. Peano

³⁶ Ancak bu gösterimde bulunan 5. aksiyom bir tümevarım kuralı olduğundan dolayı, matematiksel olarak ispatlar 0 sayısına ait özelliklerden yola çıkılarak değil, 1 sayısına ait özelliklerden hareket edilerek başlar. Russell'in bu gösterimi, en küçük sayıdan başlamak üzere tanımlanmamış kavramları belirlemek için kullandığını düşünmekteyiz.

ve arkadaşlarının kullandığı ideografik dil, Frege'nin kullandığı gotik dizgeden farklı gösterimlere sahip olup ayrıca kullanım açısından verdiği rahatlıktan dolayı Frege'nin diline baskın gelecektir.

Peano ve arkadaşlarının kurduğu bir topluluğun ortak çalışması olan *Formüller* adlı bu dev yapıtta, sadece matematik değil ve ayrıca “matematiksel bilimlerdeki çeşitli konularda, bilinen tüm önermelerin bir koleksiyonu olacak olan tümü mantık sembolleriyle yazılmış, ispatları ve tarihsel notları olan” (Peano, 1973b: 199) bir eser görünümü mevcuttur. Peano topluluğunun simgeler ve çizimler yardımıyla kavramsal ifadelerin kullanıldığı yapıtında üretilen *ideografik* (evrensel logistic) dil yapısı, matematik dünyası için oldukça önemli bir gelişmedir. Bu dil yapısının gelişmesini sağlayan etkenlerden biri, Peano'nun *a posteriori dillerle uğraşan ve hatta kendisinin de Latin Sinüs Fleksiyonu* (Latino Sine Flexione³⁷ [1903]) adında evrensel *a posteriori* dil üretmiş bir dilbilimci olması ve diğeri ise yine bir dilbilimci olması nedeniyle, matematik için bir sentaksa ulaşma isteğidir diyebiliriz. Peano'nun ideografik dil yapısına öncelikle, yukarıda Peano aksiyomlarının gösteriminden anlaşılacağı üzere, *Aritmetik İlkeler*'de rastlamaktayız. Bu küçük kitapçıkta, \sim , ε , $=$, \supset , \cap , \cup , \wedge gibi az sayıda sembole mantıksal ifadeler ve fikirler türetilmiştir. Bu tip bir sembolleştirmenin getirdiği avantajlardan diğeri yapıtlarında da faydalanan Peano der ki;

Bu analizin bir sonucu, mantığın tüm fikirlerini temsil edebilen bir grafik sembolizm veya ideografinin inşasıydı, böylece diğeri bilimlerin fikirlerini temsil etmek için semboller geliştirerek her teoriyi sembolik olarak ifade edebiliriz! İlk kez o kitapçıkta (*Aritmetik İlkeler*) bir teori, sembollerle eksiksiz olarak ifade edildi ve aritmetikte, tanımlanamayanlardan (ve kanıtlanamayanlardan) ayırt edilebilir olanı, ayırt etmek için tam olarak bunları kullandım (Peano, 1973b: 200 parantez içindeki italik bilgi bana aittir).

Peano'ya göre, bilimin fikirlerine bir düzen verilmiş olsa da hepsi tanımlanamaz. İlk fikir, öncülü olmadığı için tanımlanamaz; her tanımda yer alan “=” işareti tanımlanamaz (1973c: 253). Frege'ye göre, *ideografik* dil yapısıyla üretilen formüllerin netliği önemli bir gelişme olmakla beraber, matematiğin temel ifadeleri ve kavramlarının tanımları hala eksikli

³⁷ Peano'nun *Latin Sinüs Fleksiyonu* adlı *a posteriori* yapay dili, 1924 yılında kurulmuş Uluslararası Yapay Dil Derneği'nin (IALA) 1951 yılında tanıtılmış *a posteriori* dili olan *Interlingua*'nın atasıdır. Peano ürettiği bu dilde, Avrupa topluluğu dillerinin içinde Latincenin hala yaşayan kelimelerini ve ölmüş dilbilgisi yapısını hedef almıştır. Çekimsiz Latince olarak da görülebilecek bu dil, Latincenin ağır ve anlaşılması güç dilbilgisi yapısını neredeyse tamamen elimine edebilmiştir ki bu dil “gramersiz Latince” (Kennedy, 1973: 7) olarak da bilinmektedir. Peano'nun kendi sözleriyle “asgari dilbilgisi, dilbilgisi değildir” (Peano'dan aktaran Bodmer, 1946: 486). Bir dilbilimci olan Peano'nun matematik için *Formulaire*'sinde yaratmaya çalıştığı *a priori* yapay dilinin bir özelliği olan ekonomik dil yapısı aynı şekilde bir matematikçi olan Peano'nun doğal bir dil üstüne inşa ettiği *a posteriori* yapay dilinde de örtüşük olarak görülmektedir. Bodmer' göre, “dil planlamasının hiçbir öncüsü sayı, cinsiyet, zaman ve ruh halinin ilgisizliklerine karşı bu kadar gelenek kırıcı olmamıştı” (1946: 486).

kalmıştı. Matematiğin nesnelere kullanımında görülen tanım eksikliği (veya birçok büyük matematikçi tarafından gerekli bile görülüyor olması) matematiğin temellerinde bir problem olarak durmaktaydı ve bunun için öncelikli olarak sayı kavramının çözümlenmesi gerekiyordu. Frege sisteminde ise bu tip kavramlar, bir sembol manipülasyonu ile biçimsellik kazanacaktı.

Frege, “Cantor’un sonsuzluk üzerine kurulu kümeler teorisinden etkilenecek formel bir sistem inşa etmek istemiştir. Frege’nin niyeti matematiği mantık dışında büyütmekten engellemek ve aritmetiği de mantığın bir branşı haline getirmektir” (Hazar, 2017: 28). Frege’nin başlangıç noktası, Cantor’un kardinal sayılarıydı. Frege, *Kavram-Yazısı* adlı eserinde ürettiği yeni mantık sisteminin temelde olduğu, *Aritmetiğin Temelleri* (Die Grundlagen der Arithmetik) adını taşıyan eserini 1893 ve 1903 yıllarında iki cilt halinde yayımladı. Bu eserde Frege, aritmetiğin temellerini mantığa dayandırdığı sayı kavramı üzerine mantıksal-matematiksel incelemelerde bulunmuştur. Frege’ye göre “sayı” şeylerin birbirine eklenmesiyle ortaya çıkamaz. Kaldı ki "aynı cinsten çokluk" olarak tanımladığı "küme" ve "farklı cinsten çokluk" gibi terimlerin, “kendi bulanıklıkları nedeniyle sayının tanımlanmasında uygun terimler” (Frege, 2020: 141) olmadığını ifade etmiştir.

Bir sayı tümcesi bir kavram hakkında bir bildirim içerir. Belki bu en fazla 0 sayısında açıkça bellidir. Eğer "Venüs'ün 0 uydusu var" dersem, demek ki hakkında bir şey öne sürülecek herhangi bir uydusu ya da uydular grubu varolmamaktadır ama burada "Venüs'ün uydusu" kavramına bir özellik, yani o kavramın altına hiçbir şey düşmeme özelliği atfedilmiş [*beilegen*] olmaktadır. Eğer "Kralın arabası dört at tarafından çekiliyor" dersem, "Kralın arabasını çeken at" kavramına dört sayısını atfetmiş olurum (Frege, 2020: 142).

Frege’ye göre yaptığı bu ayırım da elbette karşıt görüşler doğuracaktır. Açıkça temkinli davranarak, 1 sayısı ile birim arasında da bir ayrıma gitmiştir. Dolayısıyla, “sayıları birlerin yan yana konulmasıyla elde etmenin bir anlamı yoktur” (Frege, 2020: 142). Frege, “ $1 + 1 = 2$ ” matematiksel ifadesinde “+” simgesinin böyle bir biraraya getirme göndermesi yapmadığını ifade ederek, aynı cinsten çokluk olarak kavramsallaştırmaya çalıştığı “sayı”yı küme teorisi üzerinden yürütmüştür.

[Frege], doğal sayıları tamamen mantıksal terimlerle tanımlayabilmek ve ardından özelliklerini türetmek için mantığını kullanmak istedi. Örneğin 3 sayısı mantığın bir parçası olarak açıklanacaktı. Bu nasıl mümkün olabilir? Bir doğal sayı, bir kümenin bir özelliğidir, yani elemanlarının sayısıdır. 3 sayısı (...) hepsinde ortak olan bir şeydir: Kutsal Üçlü, bir troyka çeken atlar seti, (normal) bir yonca yaprağı üzerindeki yapraklar seti, {a, b, c} harfleri seti. 3 sayısı hakkında bir şey söylemeden, bu kümelerden

herhangi ikisinin aynı sayıda elemana sahip olduğu görülebilir. (...) Frege'nin fikri, tüm bu kümelerin koleksiyonuyla 3 sayısını tanımlamaktı. Yani 3 sayısı sadece tüm üçlülerin kümesidir (Davis, 2018: 42).

Frege böylece, kümeler kuramı üzerinden sayı teorisini geliştirmiş ve bunu sarsılmaz olan mantıksal temellerden, salt mantıksal nosyonlardan hareket ederek gerçekleştirmiştir. Aritmetik nosyonlardan tamamen kurtulamamış Boole cebirinin aksine Frege'nin matematiksel mantığı, mantıksal "sınıf" nosyonu ile belirgin kılınır. "Böylece "doğal sayılar dizisi, özdeşlik, sınıf, sınıf-üyeliği ve sınıf-eşdeğerliği gibi salt mantıksal nosyonlardan oluşturulmuş olur" (Kenny, 2006: 368). Kardinal sayıları eşdeğer sınıfların, yani aynı sayıda üyeye sahip sınıfların sınıfları olarak ele alan Frege'ye göre, sıfır, hiçbir elemanı olmayan sınıf; bir sayısı, tüm birlik sınıfı, iki sayısı tüm çiftler sınıfı ve üç sayısı tüm üçlüler sınıfı (Kenny, 2007: 39) olmak üzere, tüm diğer sayılar bu yolla elde edilerek sayılar kümesi elde edilmiş olur.

Tüm bu açıklamalar ve kardinal sayıların tanımından hareketle, Frege'nin mantık nosyonuna göre bir sayı, eşdeğer sınıfların bir sınıfı olacaktır (Kenny, 2006: 368) ancak bu sistem de paradokslardan kaçamamıştır. Günümüzde Russell paradoksu olarak bilinen bu paradoks, Frege'nin güvendiği sisteminde çelişki olduğunu ve bu sistemin tamlik sergilemediği yönünde basit bir ispatın yine Frege'nin mantık sistemi içinde verilebileceğine dair açıklamaların bulunduğu, Russell tarafından yazılan mektupta (1902) iletilmiştir. Russell, mektubunda Frege'nin beşinci aksiyomu olarak bilinen aksiyomun çelişki doğurduğunu ifade etmiştir.

Frege, görünüşte zararsız bir aksiyom kullandı: x üzerindeki her koşul, yalnızca bu koşulu sağlayan öğeleri içeren bir küme seçer; yani " x bir kedir" koşulu kedi kümesini seçer. Ancak bazı kümelerin kendilerinin elemanı olduğunu (soyut nesnelere kümesi soyut bir nesnedir), diğer kümelerin ise olmadığını (kedi kümesi bir kedi değildir) düşünün. Frege'nin aksiyomuna göre, " x kendisinin bir elemanı değildir", yalnızca kendisinin elemanı olmayan şeyleri içeren kümeyi seçer. Buna " R kümesi" adını verin. Yani herhangi bir x , R 'nin bir elemanıdır, ancak ve ancak x , x 'in bir elemanı değilse (burada \in , "elemanıdır" ve \notin , "elemanı değildir" anlamına gelir): Tüm x için, $x \in R$ ancak ve ancak $x \notin x$ ise. Russell, 1902'de Frege'ye yazdığı bir mektupta şunu sordu: Peki ya R 'nin kendisi? Yukarıdaki ilkeye göre, R , R 'nin bir elemanıdır, ancak ve ancak R , R 'nin bir elemanı değilse: $R \in R$, ancak ve ancak $R \notin R$ ise. Yani R kendisinin bir elemanı mıdır? Eğer öyleyse, o zaman değildir-ve eğer değilse, o zaman öyledir; her iki şekilde de bir çelişki elde ederiz. Russell paradoksu olarak adlandırılan bu çelişki Frege'nin sisteminde kanıtlanabilir olduğundan, bu sistem kusurluydu (Gensler, 2017: 464).

Matematiğin, mantığın bir dalı olarak görülmesi gerektiğini düşünen Frege, bu yüzyılda kendinden önceki girişimlerde olduğu gibi mantık ve matematik arasındaki ilişkiyi sağlamlaştırma çabasında, matematiğin mantığa dayalı biçimsel bir dizgeden türetebileceğine

ve türetilmiş matematiksel ifadelerin ispatlar yoluyla korunurluk kazanacağına dair inancını, Russell'ın paradoksundan sonra kaybetmiş görünmektedir. Bununla birlikte, daha büyük bir ölçekte genelleme yapacak olursak, Boole ve Frege, on yedinci yüzyılın ikinci yarısında Leibniz ile başlayan bir süreci tamamlamış (Mugnai, 2010: 311) gibi görünmektedir.

2.2.3.1. Gottlob Frege'nin *Kavram-Yazısı Bir Characteristica Universalis* midir?

Frege'nin, *Kavram-Yazısı* adlı eserinden anlaşılacağı üzere, dilin iki düzeyi veya daha açıkçası iki farklı dil ile uğraşı söz konusudur. Bunlardan ilki, doğal dille meydana getirilen cümlelerin mantıksal görünümü; ikinci ise bu mantıksal görünümün tamamen biçimsel olarak sergilenişi üzerinedir.

Frege'nin *a priori* özellikte bir *lingua characteristica* üretme peşinde olduğu 1879 yılında, Avrupa'da dil üzerine başka türlü çalışmalar sergilenir olmuştur. Bu diller, daha ziyade birer yardımcı dil olma unsuru taşıyan ve doğal diller üzerinden üretilen yapay dillerdir. Bu bağlamda *a posteriori* özellikte dillerdir. Avrupa toplumunda bu dillerin ilki sayılabilecek, büyük üne kavuşmuş *a posteriori* bir dil olan *Volapük* (Dünya Dili) meydana çıkmıştır. Ortak ticaret dili olma amacıyla, Alman din adamı Johann Martin Schleyer (1831-1912) tarafından üretilen *Volapükte* karmaşık dilbilgisi kurallarından kaçınılmış olup, özü itibarıyla İngilizce ve Romence'ye dayanan sözcük kökleri mevcuttur.

Volapük mantıksal düşünceyi teşvik etmektedir; belirgin ve açık ifadeler kullanır ve bütüncül bir dilbilgisel kültürü hedefler. Öğretilmesi kolaydır çünkü az kuralı vardır ve istisnası yoktur. Yapay cinsiyeti yoktur ve tek bir bağlacı vardır. Ayrıca düzensiz fiili de bulunmamaktadır. Kelime kökleri bütün Avrupa dillerinden alınmıştır (Doğan, 2020: 173).

Sağlam bir dil projesi olarak görülen *Volapük*, hızla tüm dünyaya yayılmış ve yüz binlerce kişi tarafından konuşulur seviyeye gelmiş olsa da bu dilin önünü, yine onun gibi *a posteriori* bir dil olan ve Ludwig Lejzer Zamenhof (1859-1917) tarafından 1887 yılında üretilen *Esperanto* (Ümit Dili) kesmiştir. Bu dilde kitaplar yazılmış, konferanslar düzenlenmiştir. *Esperanto*, “özellikle Avrupa dilleri arasında optimum çevirideki başarısının yanı sıra bu dil üzerine biraz çalışanın hemen çözebileceği (oldukça tanıdık gelen) bir yapıya sahip olduğundan çok kolay yayılmıştır” (Doğrucan ve Hazar, 2019: 174). Günümüzde bilindiği kadarıyla yaklaşık on beş milyon insan tarafından bu dilde konuşulmaktadır, öyle ki anadili *Esperanto* olan insanlar bulunmaktadır. Dünya çapında üniversite ve okullarda okutulmakta olan *Esperanto*'nun hedeflediği uluslararası dil olma arzusu yine de eksikli kalmıştır. İlk nedeni Birleşmiş Milletler tarafından tanınmaması olsa da felsefi sağduyu açısından değerlendirmek gerekirse kolaylıkla anlaşılabilirliği üzere, bu tür diller etkileşime

açık olup, bizatihi dili yaşatmaya çalışanlarca bozulmaktadır. Böylece, genel kullanıma yayılmış bir uluslararası yardımcı dil olmasına rağmen, kusursuzluğunu kaybetmeye mahkumdur.

Önceki bölümde, Peano'ya ait *Latin Sinüs Fleksiyonu* (1903) adlı a posteriori dilden söz etmiştik. Bu dil, bu yüzyıla ait olmasa da burada yeniden değinmemizin asıl nedeni Leibniz'dir. Leibniz, on yedinci yüzyılda Latincenin sadeleşmiş hali üzerinde, Peano'nunkine benzer bir dil çalışmasına imza atmıştır. Böylece tarihsel olarak görüleceği üzere, Leibniz, bu yüzyıldan çok öncesinde, on yedinci yüzyılda, Avrupa'nın ilk uluslararası yardımcı dili üzerine çalışma yapmış kişidir³⁸. Leibniz'in *a priori* dil çalışması olan *characteristica universalis* gibi sadeleştirilmiş Latince olarak anılabilecek *a posteriori* görünümlü bu dil çalışması da eksik kalmıştır.

Öte yandan Leibniz, yirminci yüzyılın başında Giuseppe Peano'nun icat ettiği *latino sine flexione*'ye az çok benzeyen bir konuşulabilir dil biçimi oluşturmak için de büyük bir çaba göstermişti. Bu dil, tek bir ad çekimi ve tek bir fiil çekimi, cinslerin ve sayının ortadan kaldırılması, sıfatla edatın özdeşleştirilmesi, fiillerin kopula + sıfata indergenmesi yoluyla, dilbilgisinin önemli ölçüde kurallı ve yalın hale getirilmesini öngörüyordu (Eco, 1995a: 267-8).

Bu yüzyılda, Frege'nin *Kavram-Yazısı*, *a priori* ve *felsefi* yapay bir dil örneği olarak karşımıza çıkmaktadır. Bu dilde sembol manipülasyonu olmakla beraber temelini özdeşlik, çelişmezlik ve üçüncü halin imkansızlığı ilkelerinden almaktadır. Konuşulabilir olmayan, pasigrafik görünüme sahip bir kavram göstergeleri temelinde yükselen biçimsel dildir. Frege'ye göre, yarattığı bir kavram-yazısı olan aritmetik denklem dili, bir konuşmacıya ihtiyacı olmadan nesneyi doğrudan ifade etmektedir (2019a: 160) ve bu şekliyle, "basit bir yargının içeriğinin tek bir satıra yerleştirilmesine izin veren kısalığa ulaşır" (2019a: 160).

Frege'nin dizgesi yukarıda adı geçen nedenlerden ötürü, tam da Leibniz'in üretmeye çalıştığı *characteristica universalis*ini ve bu temelde yükselen yapay dil hayalini referans göstermektedir. Frege, bu anlamda, Leibniz'in *calculus ratiocinator*una referans gösterilen

³⁸ Leibniz'in on yedinci yüzyılda çalışmalarını yaptığı ilk uluslararası yapay dil projesi, yine de dünyanın ilk yapay dil projesi ünvanını alamaz çünkü bir biçimde ortadan kaybolduktan sonra, ilginç tesadüfler sonucunda yeniden gün yüzüne çıkan *Bâleybelen* adında *a posteriori* yapay dil, şu ana kadar bilinen ilk yapay dildir. Bu dilin Anadolu topraklarında çıktığı bilinmektedir. *Bâleybelen*, on altıncı yüzyılda Anadolu topraklarında, Edirne'de doğan Muhyî-i Gülşenî tarafından yaratılmıştır. Bu dilin geçirdiği tüm serüveni kitaplaştırmış olan Mustafa Koç'a göre Muhyî, çalışmasında iddialıdır: "Öyle müstakil bir dil icat ettim ki böylesini Ademoğlu yaratmadı. Türkçe ve Farsçayı bu dile aktardım, Arap dizilişiyle bu binayı sağlamlaştırdım. Adını Bâleybelen koyduğum bu dilde on konu tertip edildi. Her ilimde fazilet sahiplerinin aşamadıkları yüz meseleyi yetkince yazdım. Hak ehli birçok eser meydana getirdiler, ama hiçbirisi böylesini meydana koymuş değil" (Muhyî'den aktaran Koç, 2014: 273). Konu hakkında daha fazla bilgi için Mustafa Koç'un *Bâleybelen Muhyî-i Gülşenî İlk Yapma Dil* (2005) adlı kitabı incelenebilir.

Boole'den ayrılmaktadır fakat Schröder'e göre, Frege yeni bir mantık sistemi değil, Boole cebirini genişlettiği bir çalışma yapmaktaydı ve bu nedenle Boole cebiri, Frege'nin sistemine tercih edilmeliydi.

Aslında hem “Frege hem de Schröder, kendi biçimsel sistemlerinin Leibniz'in ideal dilinin daha iyi bir gerçekleşmesi olduğunu iddia ettiler ve rakip sistemi yalnızca bir *calculus ratiocinator* olarak” (Millán, 2021:412) görüyorlardı. Schröder'e göre, *Kavram-Yazısında* temel kavramların kelime dağarcığına rastlanamıyordu ve bu da kategoriler için sembollerden yoksun olan bu dilin, karmaşık kavramları tanımlayacak güçte olmadığı (Millán, 2021: 422) anlamına geliyordu. Bu nedenle Schröder, Frege'nin *Kavram-Yazısını* bir “yargılar hesabı” (Millán, 2021: 422) olarak tanımlamış ve bu durumda bu sistemin ancak ve ancak bir *calculus ratiocinator* olabileceğini iddia etmiştir. Kaldı ki Leibniz'in ideal dilinin öncülü *characteristica universalis* tarafıdır ve Heijenoort'a göre mantığın evrenselliğini oluşturan da *lingua characterica* yönüdür (Peckhaus, 2004: 4). Frege, 1883 yılında “Kavram-Yazısının Amacı Üzerine” (Über den Zweck der Begriffsschrift [1882]) adlı çalışmasıyla konuşmacı olarak katıldığı bir konferansta, Schröder'in eleştirilerine karşı aşağıdaki yanıtı vermiştir:

Amacım, formüllere dökülmüş soyut bir mantık ortaya koymak değil ama sözcüklerle yapılandırılmış çok daha kesin ve açık bir şekilde *içeriği* simgeler aracılığıyla dile getirmek. Aslında yapmak istediğim bir *Calculus Ratiocinator* değil, Leibniz'in düşüncesine uygun bir *Lingua Characteristica* yaratmaktır (Frege, 2019a: 144; 2020: 20).

Leibniz'in bir tilmizi olan Couturat'a göre, Frege'nin sisteminin bir *lingua characteristica* olarak görülmesi, bir *calculus ratiocinator* olarak görülmesinden çok daha doğru bir yaklaşımdır (2006: 6) çünkü Frege'nin dizgesi, “en azından prensipte matematikçiler tarafından kullanılan sıradan muhakemeyi kapsayan bir matematiksel mantık sistemidir” (Davis, 2018: 41). Davis'e göre, bu sistemin de muhakemeyi kapsamak adına, içine düştüğü boşluklar vardır: Frege'nin sistemindeki bazı öncüllerden hareket edilerek istenen sonuçlara ulaşmak için girişimlerde bulunulabilir fakat girişim başarısız olursa, bunun nedeni her zaman Frege'nin sisteminden açıkça çıkarsanamayabilir (2018: 41). Bu yaklaşımdan yola çıkarak, Schröder'in hem Frege'nin sistemini hem de Leibniz'in evrensel dil hayalini tam okuyamadığını iddia edebiliriz. Bir adım daha ileri giderek, “Frege'nin mantığının, Leibniz'in "hesaplayalım" sözleriyle mantık kurallarını bilenlerin bir sonucun gelip gelmediğini hatasız bir şekilde belirleyebilecekleri rüyasını yerine getiremediğini” (Davis, 2018: 41) de iddia edebiliriz. Bu durumda Frege'nin sistemi bir *calculus ratiocinator* olamaz. Frege aynı konferansta, Schröder'in bir *lingua characteristica* olduğunu ileri sürdüğü rakip sistem hakkında konuşmasına şöyle devam etmiştir:

Buna göre aritmetik denklem dili, mantıksal bağıntılar açısından ifadelerden yoksundur ve dolayısıyla tam anlamıyla bir kavram-yazısı ismini hak etmemektedir. Leibniz'e dayanan, yakın zaman önce Boole, R. Grassman, St. Jevons, E. Schröder ve diğerleri tarafından yenilenen mantıksal bağıntıların tanımlamalarında ise tam tersi bir durum söz konusudur. Burada, tam olmasa da mantıksal biçimler mevcuttur ancak içerik eksiktir. Burada basit harfler yerine, örneğin analitik denklemleri koyma girişiminin her biri, bütünün görünür olmaması ve anlaşılır olmaması, belirsizliği, evet hatta ortaya çıkan formüllerin çok anlamlılığı nedeniyle bu tanımlamanın, gerçek bir kavram-yazısı oluşturmak için ne kadar uygun olmadığını göstermiş olurdu (2019a: 160-1).

Frege bu konuda oldukça haklıdır. Elbette hem içeriksel bazda hem de anlatım düzlemi bağamında değerlendirilecek olursa Frege'nin sistemi, Boole'ün sistemine göre söylem evreni açısından ekonomik olduğu gibi ayrıca biçimsel dilde bağlantısallık (ilişkisel) açısından oldukça gelişmiş bir sistemdir. Bu nedenle Frege'nin sistemi, iddia ettiği üzere bir *lingua characteristic*a olarak tanımlanmaya daha yatkındır.

Frege'nin *Kavram-Yazısı* ile yarattığı kavram göstergeleri dili, biçimsel bir dil olsa da Frege'nin doğal dildeki unsurlarla ve dil felsefesinin sorunlarıyla ilgilenmesini gerektirmiştir. Mantık ile aritmetik arasında kurduğu köprüde doğal dilden faydalanan Frege'nin, 1891 ile 1892 yılları arasında dil felsefesine büyük katkı sunduğu üç makalesi (veya denemesi) yayımlanmıştır. Sırasıyla “Fonksiyon ve Kavram” (Alm. Funktion und Begriff), “Kavram ve Nesne Üzerine” (Alm. Über Begriff und Gegenstand) ve “Duyum ve Gönderim Üzerine” (Alm. Über Sinn und Bedeutung) adlarını taşıyan makaleleri, Frege'nin bileşenlerin semantik içeriğine vurgu yaptığını göstermektedir.

Düşüncelerinin günümüz bilgisayar bilimleri için de büyük önem taşıdığı “Duyum ve Gönderim Üzerine” adlı makalesinde Frege³⁹, “bir önermenin gönderiminin, önermenin doğruluk değeri olduğu varsayımımız doğruysa, o zaman bu değer, başka bir önerme ögesiyle aynı gönderime, ancak farklı bir duyuma sahip bir başka ifadeyle değiştirildiğinde aynı kalmalıdır” (2019b: 61) demektedir. “Buna göre, cümleler kendilerini oluşturan kavram-terimlerinin doğruluk fonksiyonu olarak yorumlanmaktadır” (Altınörs, 2003: 238).

³⁹ Frege yalnızca dil felsefesinde ve bilgisayar bilimlerinde etkili olmuş bir mantıkçı-matematikçi değildir. Bugün analitik felsefenin de üzerinde büyük etkisi olduğu bilinen Frege'nin asıl etkisini Viyana çevresinde görmek mümkündür. “Viyana Çevresi üyelerinin dil analizine dayanan çalışmalarının, metafizik aleyhtarlığının ve deneysel bilimleri model alan felsefi tartışmalarının merkezinde 'doğrulama' ilkesi bulunur. Bu ilkenin, sonradan mantıkçı pozitivistlerin uyarlayacağı biçimiyle ilk formülasyonu Frege tarafından yapılmıştır. Frege bir cümlenin 'anlamını bilme'nin, o cümlenin 'hangi koşullarda doğru olduğunu bilmek' ile aynı şey olduğunu savunmaktadır. Mantıkçı pozitivistler için 'sözde-önermeler' olan metafizik cümleler, doğrulanabilir olmadıkları için anlamdan yoksundur” (Altınörs, 2019: 191). Bu çevrenin en ünlüleri arasında Frege, Russell ve Ludwig Josef Johann Wittgenstein (1889-1951) sayılabilir. Wittgenstein'in, erken dönem çalışması olan *Tractatus Logico-Philosophicus* adlı eseri, olgulara dayanan ve metafiziksel öğelere alan açmayan, mantık ile yazılmış bir yapıttır. Bu eserde, her göstergenin tek bir anlam ifade etmesi gerektiğine vurgu yapan Wittgenstein, “mantıksal dilbilgisine-mantıksal sözdizimine-boyun eğen bir im dili” (2013: 3.325. 39) dilemektedir.

Düşüncesinin temelini Leibniz’de bulan Frege’ye göre öğelerin “gönderimlerinin önemli olduğu ve belirtilen biçimde yerine geçmesi durumunda değişmeden kalan, genel olarak cümleye ait olanda, doğruluk değerinden başka” (2019b: 61) bir şey bulunmadığı sonucu ortaya çıkar.

2.3. Yirminci Yüzyılda Matematiğin Temelleri

Russell’ın göstdediği antinomiden sonra, Frege için bir ümitsizlik söz konusu olmuşsa da bu yüzyılda sayılar ve kümeler kuramı üzerine çalışmalar hızla devam etmiştir. Frege’nin, mantık ve matematik arasında kurduğu bağdaşımdan ve mantık temelleri üzerinde yükselmesi gerektiğine inanılan matematiğin, mantığın bir dalı olduğuna dair inancının bir sonucu olarak, bu iki alanın bir birleşimi gibi görülebilecek mantıksalcılık, bu yüzyılda Frege’nin yeni mantık sistemi için kullanılan genel bir isim olmuştur. Mantıksalcı yaklaşım Russell ve Whitehead’de doruk noktasına ulaşır.

Mantıksalcı yaklaşımla aynı yerde durmayan ve günümüz matematik felsefesi için aktif bir araştırma konusu olan matematiğin temellerinin farklı görüşler üzerinden sorgulaması, halen devam etmektedir. Matematiğin temellerinin sorgulandığı çeşitli yaklaşımlar arasında belirginlik kazanan özellikle üç tür yaklaşım mevcuttur: *Mantıksalcılık* (İng. logicism), *sezgicilik* (İng. intuitionism) ve *biçimselcilik* (İng. formalism). Tezimiz bağlamında sezgiselci yaklaşım üzerinde durmayacağız. On dokuzuncu yüzyılda erken dönem biçimselci yaklaşımın en etkilileri arasında, “matematikçiler Thomae ve Heine ile birlikte, bir mantıkçı olan Schröder’in” (Brown, 2008: 68-9) adı geçmektedir. Erken dönem biçimselcilerden Schröder ile birlikte, tüm (erken veya Hilbert öncesi) biçimselci ekol Frege tarafından 1884 yılında hezimete uğratılmıştır (Brown, 2008: 69).

Matematiğin temellerinin sarsılmaz bir tutarlılıkla sağlamaştırılmasının gereği üzerinde duran biçimselci görüşü benimseyen Hilbert okulunun çalışmaları yirminci yüzyılda çok önemli bir yer edinmiştir. Bu okul da tıpkı Hilbert gibi biçimselci (formalist) yaklaşımı benimsemiştir. Bizim için, Leibniz’in evrensel dil hayali bağlamında değerlendirilmesi gereken önemli bir ayrıntıyı barındıran Hilbert okulunun çalışmalarına ve bu çalışmaların uğradığı yıkıma bir alt başlıkta değinilecektir.

2.3.1. Alfred North Whitehead (1861-1947) ve Bertrant Russell (1872-1970)

Russell ve Whitehead, tıpkı Frege gibi mantıksalcı bir tutum sergiler. Onlar da Frege gibi tanımsız bırakılan her kavramın bir tanıma ve belirlenime ihtiyacı olduğunu düşünürler. Öncelikle, matematiğin temellerinin hiçbir boşluk kalmaksızın herhangi bir antinomiye maruz bırakılmadan sağlamaştırılması, sonra da aynı tutumla devam edilerek geometrinin

temellerinin sağlamlaştırılması hedeflenir. Bunun için Russell ve Whitehead, çok meşakkatli bir işe girişerek öncelikle *Principia Mathematica* adlı üç ciltlik anıtsal eserlerini sırasıyla 1910, 1912 ve 1913 yıllarında yayımlamışlardır fakat yaşadıkları zorluklardan dolayı, geometrinin temellendirilmesi girişiminden vazgeçerler. Russell paradoksunun gösterdiği üzere, Cantor'un kümeler teorisinden (İng. naive set theory) etkilenen Frege'nin sistemi, antinomilere açtı. Bu nedenle, kendi sistemlerinde çelişkiye düşmemek için *Principia Mathematica*'da Russell'ın 1908 yılında önerdiği *tipler teorisi* (İng. theory of types) kullanılmıştır.

Russell, tipler teorisini geliştirirken, evreni seviyelere veya tiplere bölerek başlamıştır. Bu teoriye göre, belirli bir koşulu sağlayan şeylerden ancak ve ancak hepsi aynı tipten ise söz edebiliriz. O halde bir sınıfın üyelerinin hepsi tek tip olmalıdır (Heijenoort, 1967: 151). Russell, bir kısır döngüye sürükleyen paradokslardan kurtulmanın çözümünü “hiçbir bütünlüğün kendi içinde tanımlanmış üyeleri içermemesi” gerektiğine dair belirlediği ilkesinde bulmuştur. Buna göre, “görünür bir değişken içeren her şey', bu değişkenin üzerinde bulunduğu şeylerden daha yüksek bir tipte olmalıdır” (Heijenoort, 1967: 151). Bu teoriye göre, en altta nesnelere bulunduğu kümeler arasında hiyerarşik bir derecelendirme sağlanmış olacaktır.

Hiyerarşinin en altında kümeler değil, nesnelere bulunmaktadır. Onun da üzerinde ise birinci dereceden kümeler bulunur ve bu şekilde devam eder. Her nesne veya küme, belli bir tiptedir ve bir küme, yalnızca daha alt tipteki kümeleri veya nesnelere içerebilir. Ayrıca hiçbir küme kendi kendisini içerebilir. Bir küme içerecek tek şey daha üst tipteki bir kümedir. Cantor'un kümeler teorisinde çelişki çıkaran Russell kümesi gibi kümeler de bu sistemde düşünülemez. Çünkü bu küme sistemde belirtilen hiçbir tipe ait değildir. Böylece Russell, kümeler ve nesnelere arasında yapay bir hiyerarşi kurarak, Russell paradoksu gibi çelişkilerin kümeler teorisinin dışına atılmasını sağlamıştır (Öztürk, 2011: 53).

Russell ve Whitehead tarafından yazılan *Principia Mathematica*'nın antinomilerden korunaklı, tutarlı bir yapıda tüm tanımları kesinleşmiş şekilde sunulabilmesi için her türlü çaba sarfedilmiştir. Öyle ki bir teoremin ispatının verilebilmesi ve hatta çok basit görünümlü bir eşitliğin sunulabilmesi için oldukça uzun ve sıkıcı bir işlemler ve tanımlar silsilesi kullanılır. Bir örnek vermek gerekirse, “çetrefilli bir notasyonla yazılan kitabın 1912'de basılan ikinci cildinin 86. sayfasında nihayet $1+1$ 'in 2 'ye eşit olduğunun kanıtı tamamlanıyor” (Say, 2019: 24). Bunun nedenini, sonradan Russell'ın felsefesinde de görüleceği üzere, her kavramın atomik olarak incelenmesinde arayabiliriz çünkü ancak bu yolla, sistemde bütünlük sağlanabileceği düşünülmüştür. Russell ve Whitehead'ın kendi ifadeleriyle aktarmak gerekirse:

Bununla birlikte hem tartışmadan hem de genel felsefeden kaçındık ve ifadelerimizi biçimsel olarak dogmatik hale getirdik. Bunun asıl gerekçesi, matematiğin ilkeleri üzerine herhangi bir teorinin lehine gerçekleşen başlıca akıl yürütmenin her zaman tümevarımsal olması gerektiğidir; yani söz konusu teorinin, yalın matematiği ortaya çıkarmamızı sağlaması gerekliliğidir. Matematikte, kendini kanıtlamanın en büyük derecesi genellikle başlangıçta değil, daha sonraki bir noktada bulunur. Bu nedenle, bu noktaya ulaşılan kadar ilk çıkarımlar, gerçek sonuçlar onlardan çıktığı için öncüllere inanmamıza dair; daha sonra öncüllerden çıktıkları için sonuçlara inanmamıza dair nedenler verir. (Russell ve Whitehead, [1910] 1963: 5)

Bu eser, tüm zor görünümü ve ağır işleyişine rağmen, içeriğinde kendisinden önce yapılmış tüm mantık ve matematik alanındaki çalışmaların gerçek bir resmini barındırmaktadır.

Principia Mathematica'da, asıl amacın matematiğin temelleri olduğunu; üzerine çalışmalar yapılan alanın aslında tam olarak mantık değil, matematik olduğunu vurgulamakta yarar görmekteyiz. Matematik, özü itibariyle kendine has bir mantıkla yürütülmektedir: İlişkiler mantığı.

Klasik mantık ve mantık cebiri matematiğe hemen hemen hiç yarar sağlamazken, matematik, ilişkiler mantığında kavramlarını ve temel ilkelerini bulur; matematiğin gerçek mantığı, ilişkilerin mantığıdır. Mantık cebirinin kendisi, belirli bir matematiksel teori olarak kabul edilen saf mantıktan doğar çünkü örtük olarak öne sürülen ve tüm sembolleştirmenin yanı sıra tüm mantık cebirinin temeli oldukları için cebirsel veya sembolik ifadeye duyarlı olmayan ilkelere dayanır. Buna göre mantık cebirinin şekli ve yöntemi ile matematiksel bir mantık olduğunu söyleyebiliriz, ancak matematiğin kendi mantığı ile karıştırılmamalıdır (Couturat, 2004: 89).

Matematiğe sağlam bir temellendirme yapılma niyetiyle ortaya konan *Principia Mathematica*'da, Frege'nin birinci seviye yüklem mantığının kullanıldığını fakat biçimsel dizgesinin zorluğu nedeniyle, Peano'nun ideografik dil yapısından faydalanmayı gerektirdiğini daha önce vurgulamıştık. Bu eserin bize kazandırdıklarını, birer örnek eşliğinde incelemenin faydalı olacağını düşünmekteyiz. Aşağıda, “ise” bağlacının kullanımını gösteren bir örnek görülmektedir (aşağıdaki örnek [Davis, 2018: 37]’den alınmıştır):

1. “Tüm atlar, memelidir”: x bir at ise x bir memelidir. (eğer...ise...)
2. “Bazı atlar safkandır”: x bir at ve x safkan. (... ve ...)

Aşağıda gösterildiği şekilde $\forall x$ sembolü “her” veya “tüm” gibi bir anlamda kullanılır ve $\exists x$ sembolü ise varoluşu temsilen “bazı” ya da “öyle bir özellik vardır ki...” gibi bir anlama gelir. Bu iki cümle, günümüz gösteriminde geçerli niceleyiciler kullanılarak şu şekilde yazılabilir:

1. $(\forall x)$ (eğer x bir at ise x bir memelidir)

2. $(\exists x) (x \text{ bir at ve } x \text{ safkan})$

Dolayısıyla, ilk cümlemiz tüm atlar için geçerli olan genel bir özelliğe vurgu yaparken, ikinci cümlemizde belirtilen özellik sadece bazı atlar için mevcuttur. O halde ikinci cümle aslında şöyle ifade edilebilir:

Öyle bir x vardır ki x bir at ve x safkandır.

Günümüz gösteriminde, “... veya ...” \vee sembolüyle gösterilirken, “eğer ...ise ...” \supset simgesiyle sembolize edilir. “... ve ...” \wedge ile sembolize edilir. Bu sembolleri kullanarak cümleler şu hale gelir (Davis, 2018: 37):

1. $(\forall x) (x \text{ bir attır } \supset x \text{ bir memelidir})$
2. $(\exists x) (x \text{ bir attır } \wedge x \text{ safkandır})$

Biraz daha kısaltılmış halleri aşağıdaki gibidir:

1. $(\forall x) (\text{at}(x) \supset \text{memeli}(x))$
2. $(\exists x) (\text{at}(x) \wedge \text{safkan}(x))$

Biçimsel bir dilde, aynı ifadeler şu şekilde görünür (Davis, 2018: 38):

1. $(\forall x)(a(x) \supset m(x))$
2. $(\exists x)(a(x) \wedge s(x))$

Şimdi, daha önce verdiğimiz örnek (Davis, 2018: 29) ile Frege’nin birinci seviye yüklemeler mantığı ile yürütülen *Principia Mathematica*’da, günümüz notasyonları ile Boole’dan farklı olarak neyi başardığını serimlemeye çalışacağız. Boole cebirinde “başarısız olan tüm öğrencilerin ya aptal ya da tembel olduğu” ifadesinin çözümlenemeyeceğine dair verdiğimiz bu örnek, Frege’in mantığında, günümüz notasyonları ile rahatlıkla ifade edilebilir:

$B(x)$, x başarısız öğrenci,

$A(x)$, x aptal,

$T(x)$, x tembel,

olarak belirlendikten sonra,

$$(\forall x)(B(x) \supset A(x) \vee T(x))$$

şeklindeki biçimsel ifadeye bürünür.

Yukarıda verilmiş örneklerden görüleceği üzere, *Principia Mathematica* ile Frege’nin mantığı, kesin bir sözdizimine sahip, önermeler arasındaki ilişkilerin sembol manipülasyonu

yapılmış doğal dil ifadeleriyle verilebildiği, biçimsel olmasının yanı sıra henüz anlamdan tam soyutlanmamış *a priori felsefi* yapay bir dil görünümünde sunulabilmektedir. Böylece, kümülatif bir çabanın belki de geldiği son nokta olarak görülebilecek *Principia Mathematica*, günümüz notasyonlarının kullanım kolaylığından da anlaşılacağı üzere, Russell ve Whitehead'ın ellerinde doğru bir sentezden geçirilmiş olduğunun kanıtıdır.

Formulaire anıtsal bir yapıttır ancak, matematiksel faydaları *logistic* olduğu kadar ansiklopediktir ve yöntemin tüm olasılıkları kullanılmaz veya netleştirilmez. Matematikte sembolik mantık ile *logistic* yöntemin mükemmel birliğini sergilemek, *Principia Mathematica*'da Whitehead ve Russell'a kaldı. Bu çalışmanın yayımlanması kuşkusuz konunun tarihinde bir çığır açmaktadır. Boole'den Schröder'e mantık cebirinin gelişiminde, De Morgan'dan Schröder'e ilişkiler cebirinin gelişiminde ve Cantor, Dedekind ve Frege'nin sayı teorisinin temellerinde gelişen eğilimler burada bir araya getirilmiştir. Daha sonraki araştırmalar büyük olasılıkla *Principia Mathematica*'nın formülasyonlarına dayanacaktır (Lewis, 1918: 5 italik bana ait).

Peano'nun *ideografik* dili ya da başka bir deyişle kolay anlaşılır sembolik görünümlü dili, Frege'nin anlaşılması ve uygulaması zor biçimsel dizgesine çok iyi bir alternatif olarak, *Principia Mathematica*'da kullanılmış ve Frege'nin "... eğer ... ise ..." türündeki mantıksal çıkarım yönteminin rahat bir kullanım alanı vermesini sağlamıştır. Tekrar değinilmesi gereken bir husus olarak, Frege'nin biçimsel dizgesinin temelinde kavram göstergeleri bulunmaktaydı. Bu nedenle, rahatlıkla denilebilir ki *Principia Mathematica* bize *a priori felsefi* yapay bir dille sunulmuştur. Yalnızca bir sözdizim içeren *Principia Mathematica*'nın dili, Russell'a göre, bir sözlüğün eklenmesiyle mantıksal açıdan kusursuz olacaktı. Bununla birlikte, bir kavramlar sözlüğü eklenmesi sonucu, "Russell, kurulabilmesi durumunda böyle bir dilin katlanması olanaksız ölçüde uzun ve sıkıcı olacağını kabul ediyordu" (Eco, 1995a: 308). Öyle bile olsa, *Principia Mathematica*'da gösterilen şekliyle, kavram göstergeleri ile birleşen *ideografik* dilin, Leibniz'in hayaline oldukça yaklaşmış bir dil yapısı olarak sergilendiği iddia edilebilir.

Fantastik ifadeye ve biraz abartmaya rağmen, burada *logistic* programını tanıyoruz. Tüm bilimin yeniden inşası çok iddialı bir proje olsa bile yine de genel olarak kesin bilimin böyle bir yeniden inşasının ideal olasılığını ve arzu edilirliliğini korumalıyız. Nihayetinde, *ideografik dil*, Peano'nun *Formulaire*'inde, *Principia Mathematica*'da ve *logistic* yöntemin tüm başarılı uygulamalarında gerçekliğini bulmuştur. Leibniz, bilimin ve genel olarak insan düşüncesinin daha hızlı ve düzenli ilerlemesi için böyle bir *dilin* önemini vurgular (Lewis, 1918: 7 italik bana ait).

Leibniz'e göre, cebirin gerçek bir gizemi olarak gördüğü sembol manipülasyonu, cebirde sembolik ifadelerin doğru şekilde kullanımında hayat bulur. "Sembollerin doğru kullanımına gösterilen bu özen, bilim insanına kendi karakteristiğini yaratmada rehberlik

edecek olan *Ariadne ipliği*⁴⁰” (Lewis, 1918: 6; Davis, 2018: 11) olacaktır. Leibniz’in tüm insan muhakemesinin hesaba indirgenebileceğine, bunun açıklayıcısı olacak evrensel dilin yaratılabileceğine dair hayali ve inancı tüm bu çalışmalar doğrultusunda yeniden hayata dönmüş gibidir.

Matematikte biçimselliğin geldiği noktadan tekrar bakmak gerekirse, Peano’nun doğal sayılar kümesinde gerçekleştirdiği aksiyomatikleştirme ve sembolleştirme büyük bir başarıydı ancak, yeterli miydi? Frege’nin kavram göstergeleri dili yaratması ve doğal dillerde bulunan bağlaçlar üzerinde yaptığı sembol manipülasyonu alkışlanacak bir başarıydı ancak bu dilin, öncüllerden istenen sonucun çıkarılamadığı durumlarda, başarısızlığın nedenini verebilecek bir prosedürü var mıydı? *Principia Mathematica*’da Russell ve Whitehead, tüm matematiğin genel olarak sembolik mantık ve ilişkiler mantığı aracılığıyla tamamen biçimselleştirilebildiği, çıkarım yapmanın mümkün olduğu, *tipler kuramı* ile iyi belirlenmiş kümeler üzerinden teorem ve ispatların yapılabileceğinin en gelişmiş kanıtıydı ancak, bu sistemler tutarlı ve tam mıydı? Bu sorularla, bu sistemlerin yaratıcıları ya yeteri kadar ilgilenmemişti ya da önemsemişti (özellikle Dedekind ve Peano) fakat döneminin en ünlü matematikçisi Hilbert bu soruların yanıtlarının peşinde olacaktı.

2.3.2. David Hilbert (1862-1943) ve Programı

Matematiğin tamamen biçimsel bir dille ifade zemini kazanmaya başladığı yirminci yüzyılda biçimsel dizge, matematiğin tüm alanlarında bir genelleme ve soyutlama durumunu da beraberinde getirmişti. Tüm bu soyutlamalar ve aksiyomatikleştirmelerden kaynaklanan sorunlar, çözümsüz kalmaması gerektiği nispetinde, özellikle Hilbert gibi dönemin büyük matematikçisi ve yaşadıkları dönemde çalışmalarıyla ses getirmiş, bugün hala büyük matematikçiler ve mantıkçılar olarak anılan bir grubu, aksiyomatikleştirmeler üzerine çözüm(ler) bulmaya itmişti. Öklid geometrisinin aksiyomatik sistemindeki herhangi bir tutarsızlığın, gerçek sayıların aritmetiğinde bir tutarsızlıkla sonuçlanacağını göstererek, Öklid geometrisinin tutarlılığını aritmetiğe indirgemeyi (Davis, 2018: 74) başarmış olan Hilbert için, matematiğin temellerinin sarsılmazlığı gittikçe daha önemli bir hale gelmişti. Açıkçası

⁴⁰ Leibniz *Ariadne ipliği* derken, özellikle sembol manipülasyonu yapılacak kavram ile karakteristik arasındaki bağı işaret eder. Doğru bir sembol manipülasyonu bize kavramın anlamını değil, neliğini verecektir. Çünkü bir kavram zaman içinde yeni anlamlar kazanabileceği gibi kendisiyle bilinen anlamlarını kaybedebilir. Oysa kavramın neliği ve kavramlar arasındaki bağlantısallık her zaman kalır. *Ariadne*, mitolojik bir figürdür ve genellikle hikayelerde ya bu iplik sayesinde yolunu bulup kurtulur ya da ipliği başkalarının yolunu bulması için verir: “Theseus Giri’te Minotaurus’la çarpışmaya geldiğinde Ariadne yigidi görmüş ve görür görmez de ona tutulmuştu. Minotaurus’un bulunduğu bin bir dehlizli Labyrinthos mağarasında kaybolmaması için eline bir yumak iplik vermişti. Theseus da karışık ve karanlık dehlizlerden ilerledikçe yumağı açıp ipliği yere bırakıyormuş. Canavarı öldürdükten sonra çıkış yolunu ona bu iplik göstermiş” (Erhat, 1996: 50).

aritmetiğin tutarlılığı kanıtlanabilirse Hilbert'in indirgemeyi başardığı Öklid geometrisinin de tutarlılığı kanıtlanabilmiş olacaktır. Hilbert, bu nedenle aritmetiğin tutarlılığını sorgulama girişiminde bulunmuştu. Hilbert programının özünde, tamamen biçimsellik kazandırılmaya çalışılan matematiğin aşağıdaki özellikleri taşıması gerekliliği yatmaktadır (aşağıdaki öncüller [Çevik, 2021:83]'den alınmıştır):

- (i) Sistem tutarlı olmalı. Yani aksiyomlardan çelişki çıkarılmamalı.
- (ii) Sistem tam olmalı. Yani sistemin biçimsel dilinde verilen her önerme (ya da değili) aksiyomlardan çıkarılmalı.
- (iii) Her önermenin sistem içindeki doğruluk değerine (yani aksiyomlardan çıkarılabilir olup olmadığına) algoritmik olarak karar verilebilmeli.

1900 yılında Paris'te düzenlenen Uluslararası Matematikçiler Kongresinde Hilbert, konuşmasına "Bir matematik teorisi, onu sokakta karşılaştığımız ilk adama açıklayabilecek kadar netleştirmedikçe tamamlanmış sayılmaz" (1900: 438) sözleriyle başlar. Meramını açıkça anlattığı sözlerinden sonra, matematiğin temellerinin sağlamlaştırılmasına dair belirlediği toplamda yirmi üç adet sorun teşkil eden durumun açıklığa kavuşturulması konusunda talepte bulunur. Bu soruların ilki, Cantor'un *süreklilik hipotezi* üzerinedir. *Süreklilik hipotezi* özetle, 1877 yılında Cantor'un \mathbb{R} 'nin \mathbb{N} doğal sayılardan daha büyük ama gerçel sayılardan daha küçük bir altkümesinin olup olmayacağına (Nesin, 2020: 301) dair sorusuyla⁴¹ ortaya çıkar. Günümüz teknik gösterimiyle (aşağıdaki gösterim [Nesin, 2020: 301]'den esinlenilerek geliştirilmiştir) aşağıda görüldüğü üzere, bu hipotezde bir kardinalite sorgulaması mevcuttur.

$$|\mathbb{N}| = \omega = \aleph_0 < \omega_1 \leq 2^\omega = 2^{\aleph_0} = |\mathbb{R}|$$

Asıl amaç, ω_1 gibi bir kardinal sayısının bulunmadığını göstermektir. *Süreklilik hipotezinin iyi sıralı* olmakla bağlantısı, yukarıda verdiğimiz gösterimden anlaşılacağı üzere Hilbert'in, *süreklilik hipotezinin* kanıtlanmasından önce başka bir kanıtın gerekliliğini vurgulamasına neden olmuştur. Öncelikli olarak her kümenin iyi sıralanabileceğinin kanıtlanması (Nesin, 2020: 212) gerekiyordu. Hilbert'in etkisiyle analiz üzerine yaptığı çalışmalardan sonra kümeler teorisine yönelmiş olan Ernst F. F. Zermelo (1871-1953), genç yaşta ünlenmesine neden olan *aksiyomatik küme kuramı* ve yaşadığı dönemde kabullenilmesi

⁴¹ Bu soru daha açıkça başka şekillerde de sorulabilir: "Yani \mathbb{R} 'nin sayılamaz sonsuzlukta olan ama kardinalitesi 2^ω olmayan bir altkümesi var mıdır? Ya da \mathbb{R} 'nin her sonsuz altkümesi ya \mathbb{N} ya da \mathbb{R} ile eşlenik olmak zorunda mıdır? Daha modern bir deyişle, $\omega_1 = 2^\omega$ eşitliği doğru mudur? Cantor yanıtın olumsuz olacağını tahmin etti ama bir türlü kanıtlayamadı. Cantor'un bu tahminine Süreklilik Hipotezi denir" (Nesin, 2020: 301).

zor, *seçim aksiyomu* olarak bilinen aksiyomunu dahil ettiği sistemin ortak kuramıyla, matematiğin temellerinin güçlendirilmesi çabasıydı. Zermelo'nun kümeler kuramının aksiyomatikleştirilmesi üzerine çalışmalarını sunduğu 1904 yılında, her kümenin iyi sıralanabileceğine dair sunduğu kanıtlarında *seçim aksiyomu* kullanılmıştı ve bu aksiyom, matematik dünyası tarafından hoş karşılanmamıştı.

Russell'ın *Principia Mathematica*'da kullanmak üzere “tipler teorisi”ni sunduğu 1908 yılında Zermelo, daha iyi kanıtlarla tekrar ortaya çıkmıştı. Sonlu sayıda iyi sıralı kümeler ile işlem yapılıyor olması durumunda ihtiyaç duyulmayan *seçim aksiyomu*, boş olmayan iyi sıralı kümelerin (sonsuz sayıda olabilir) her birinden seçilebilecek bir elemanla oluşturulacak kümenin boş olmayacağına dair belirlenim veren bir aksiyomdur. Daha teknik ifade etmek gerekirse: “İkili ayrık boş olmayan kümelerin her A kümesi için, A'daki her kümeden tam olarak bir eleman içeren bir küme vardır” (Bagaria, 2021: 3). *Seçim aksiyomunun* yanı sıra büyük çoğunluğu Zermelo'ya ait aksiyomlara 1922 yılında Abraham Adolf Fraenkel'in (1891-1965) ekleme yaptığı “yerleştirme aksiyomu” ve “temellendirme aksiyomu” (bu aksiyom aynı dönemde Thoralf Skolem [1887-1963] tarafından bağımsız bir şekilde bulunmuştu) ile birlikte, günümüzde ZF kümeler kuramı (İng. Zermelo-Fraenkel Set Theory) veya *seçim aksiyomunun* eklenmesiyle ZFC kümeler kuramı (İng. Zermelo-Fraenkel Set Theory and Axiom of Choice) kısaltmalarıyla bilinmektedirler. *Seçim aksiyomu* ile birlikte bu aksiyomların sayısı “sonlu (10 tane) gibi gözükabilir ama bu yanıltıcı⁴²: 3'üncü ve 9'uncu aksiyomlar aslında sonsuz sayıda aksiyomdan oluşuyorlar, her biri her ϕ özelliği için ayrı birer aksiyomu simgeler” (Nesin, 2020: 162). Aşağıda ZFC Aksiyom Sistemi günümüz gösterimleriyle verilmiştir (aksiyomlar ve gösterimleri [Nesin, 2020: 163]'den alınmıştır):

1. Boş Küme Aksiyomu. Hiç elemanı olmayan bir küme vardır: \emptyset .
2. Eşitlik Aksiyomu. Aynı elemanlara sahip iki küme birbirine eşittir.

⁴² Günümüzde kümeler kuramı denilince akla ilk gelen, matematikçilerin çoğunlukla tercih ettiği ZF veya *seçim aksiyomu* dahil edildiğinde ZFC küme kuramı olsa da bunun dışında ZFC ile “aşağı yukarı eşdeğer birkaç kümeler kuramı vardır” (Nesin, 2020: 162). ZF veya ZFC kümeler kuramı dışında “Bertrand Russell'in tipler kuramı ve von Neumann, Bernays ve Gödel'in kümeler kuramı (NBG) vardır” (Nesin, 2020: 162). Bu kuramlar arasında bir ayrıma gitmek gerekirse, “ZFC kümeler kuramında sonsuz sayıda aksiyom vardır. Montague 1961'de bu sistemin sonlu sayıda aksiyoma indirgenemeyeceğini kanıtlamıştır. Gödel, Bernays ve von Neumann'ın bulduğu NBG aksiyom sistemi sonludur. Her iki sistemde de kümelerle ilgili aynı sonuçların kanıtlanacağı biliniyor. Ancak NBG sisteminde küme olmayan sınıflardan da söz edildiğinden, sonlu olmasına karşın, bir anlamda NBG sistemi ZFC'den daha zengindir diyebiliriz. Bir başka deyişle, ZFC sisteminin dili \forall , \exists , $=$ gibi standart matematik simgeleri dışında sadece \in simgesini kullanırken, NBG sistemi \in simgesi dışında, küme olmayan topluluklardan söz edebilmek için fazladan bir simge daha kullanır” (Nesin, 2020: 164).

3. Tanımlanabilir Altküme Aksiyomu. Eğer φ bir özellikse ve x bir kümeysse, x 'in φ özelliğini sağlayan elemanlarını, eleman olarak içeren ve bunlardan başka eleman içermeyen bir küme vardır:

$$\{y \in x : \varphi(y)\}.$$

4. Bileşim Aksiyomu. Eğer x bir kümeysse, eleman olarak sadece ve sadece x 'in elemanlarının elemanlarını içeren bir küme vardır:

$$\cup x = \cup_{y \in x} y = \{z : \text{bir } y \in x \text{ için } z \in y\}.$$

5. İki Elemanlı Küme Aksiyomu. Eğer x ve y birer kümeysse, eleman olarak sadece ve sadece x ve y 'yi içeren bir küme vardır:

$$\{x, y\}.$$

6. Altkümeler Kümesi Aksiyomu. Eğer x bir kümeysse, eleman olarak sadece ve sadece x 'in altkümelerini içeren bir küme vardır:

$$\wp(x) = \{y : y \subseteq x\} = \{y : y\text{'nin her elemanı } x\text{'in de elemanıdır}\}.$$

7. Tümevarımsal Küme Aksiyomu. Boş kümeyi içeren ve içerdiği her x kümesi için

$$x \cup \{x\}$$

kümesini de içeren (en küçük) bir küme vardır.

8. Temellendirme Aksiyomu. [Fraenkel] Eğer x boş olmayan bir kümeysse, o zaman

$$x\text{'in } x \cap y = \emptyset$$

eşitliğini sağlayan bir y elemanı vardır (bu aksiyom sadece kümeler kuramında kullanılır).

9. Yerleştirme Aksiyomu. [Fraenkel ve Skolem] a bir küme ve $\varphi(x, y)$ bir özellik olsun.

$$\text{Her } x \in a \text{ için, } \varphi(x, y)$$

özelliğini sağlayan bir ve bir tane y kümesi varsa o zaman bir $x \in a$ için $\varphi(x, y)$ özelliğini sağlayan y 'ler bir küme oluştururlar. Yani

$$\{y : \exists x(x \in a \wedge \varphi(x, y))\}$$

topluluğu bir kümedir.

10. Seçim Aksiyomu (C). Elemanları boş olmayan kümeler olan her kümenin bir seçim fonksiyonu vardır.

ZFC kümeler kuramının (Zermelo-Fraenkel küme kuramının kısaltması, tezimizde bu kısaltmayı kullanarak ilerleyeceğiz) yedinci aksiyomu olan “tümevarımsal küme aksiyomu”nun incelenmesi sonucunda, bu aksiyomun tanımından başlamak üzere (ve diğer aksiyomlar yardımıyla) iyi sıralı bir doğal sayılar kümesinin kurulabileceğini görebiliriz. $x \cup \{x\}$ ile verilmek istenen tanım şu şekilde sunulabilir: Eğer x bir kümeysse $Sx = x \cup \{x\}$. Bu

tanımın açılımında, hiçbir elemanı olmayan küme yani boş küme başlangıç noktası olarak alınmak suretiyle aşağıda görüleceği üzere,

$$0 = \emptyset$$

$$1 = \{0\} = 0 \cup \{0\}$$

$$2 = \{0, 1\} = 1 \cup \{1\}$$

$$3 = \{0, 1, 2\} = \{0, 1\} \cup \{2\} = 2 \cup \{2\}$$

⋮

şeklindeki inşa sonsuza kadar sürdürülebilir. Burada, Zermelo'nun tanımında $2 \in 3$ (yani $\{\{\emptyset\}\} \in \{\{\{\emptyset\}\}\}$) denilebilirken, $2 \notin 4$ olmaktadır. John von Neumann'un (1903-1957) tanımı bu anlamda daha esnektir. Örneğin, $2 \in 4$ bu sistemde $(\{\emptyset, \{\emptyset\}\} \in \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}, \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\})$ gösterimiyle kolaylıkla ifade edilebilir (Brown, 2006: 80). Frege'nin yapmak istediği doğal sayı inşasının, von Neumann tarafından elde edildiğine dair günümüz teknik gösterimi aşağıdaki gibidir (aşağıdaki gösterim [Çevik, 2021: 36]'dan alınmıştır):

$$0 = \emptyset,$$

$$1 = \{0\} = \{\emptyset\},$$

$$2 = \{0, 1\} = \{\emptyset, \{\emptyset\}\},$$

$$3 = \{0, 1, 2\} = \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\},$$

⋮

Hilbert okulunun çalışmaları, sordukları soruların yanıtları ışığında (ki bu yanıtların matematiği sarsılmaz hale getireceğinden şüpheleri yoktu) matematiğin temellerinin sağlam bir mantıksal zeminde kurulmasını sağlayacak bir projeye dayanıyordu. İleride serimlemeye çalışacağımız üzere, Hilbert projesi burada değindiğimiz üç soruya (bu sorulardan biri, “çelişkisizlik” ve “tamlik” bağlamında aslında iki soru olarak değerlendirildiğinden dolayı toplamda dört soru olduğunu iddia edebiliriz) verilen yanıtlar doğrultusunda çökmüş bulunmaktadır. Hilbert'in 1900 yılında Paris'te düzenlenen Uluslararası Matematikçiler Kongresinde sorduğu diğer bir soru olan ikinci sorusu, aritmetiğin çelişkisizliği üzerinedir.

Daha yakından incelendiğinde şu soru ortaya çıkıyor: Birbirinden tamamen bağımsız olacak bir aksiyomlar sistemine ulaşmak isteniyorsa, bunlar izole edilmelidir. Öyleyse, tekil aksiyomların belirli ifadeleri herhangi bir şekilde birbirine bağlı mıdır ya da aksiyomlar ortak belirli parçalar içerir mi? Aslında, her şeyden önce belirtmek isterim ki aksiyomlarla ilgili sorulabilecek sayısız sorudan en önemlisi şudur: Aksiyomların çelişkili olmadıklarını, yani onlara dayalı sonlu sayıda mantıksal adımın asla çelişkili sonuçlara yol açamayacağını kanıtlamak (Hilbert, 1900: 447).

Aritmetiğin çelişkisizliğine ve kümeler teorisinin tutarlılığına dair yalnızca Hilbert değil, ayrıca Hilbert okulunun da bir çözüm bulma arayışında olduğu bilinmektedir. Bunlar arasında asistanı Paul Isaac Bernays (1888-1977), öğrencisi Wilhelm Ackermann (1896-1962) ve bir sonraki bölümde yeniden önemine değineceğimiz büyük matematikçi ve mantıkçı John von Neumann vardı fakat von Neumann, Hilbert'in öğrencisi değildi, sadece Hilbert'in problemlerinin çözümüyle yakından ilgiliydi. Hilbert ve beraberindekilere göre, bu problemlerin çözümü neticesinde sezgisel yaklaşımın içine dahil edilemeyeceği biçimsel pür matematik elde edilebilecekti. "Hilbert ve okulu basit biçimsel sistemlerin tutarlılığını gösterdiler ve bu nedenle aritmetiğin ve kümeler teorisinin tutarlılığını kanıtlama amacını gerçekleştirmek üzere olduklarına inanmaktaydılar" (Kline, 1972c: 1206). Onlar, birçok başarılar imza atmışlardı ve dahasının geleceğine dair bir şüpheleri yoktu.

Hilbert'in kendi çalışmalarından da aşına olduğu, Peano aksiyomlarında ve ZFC kümeler kuramında görüleceği üzere, doğru bir aksiyomatikleştirmenin yapıldığı iyi sıralı doğal sayılar kümesinden sırasıyla bu kümeye negatif sayıların eklenmesiyle tamsayılar kümesi; oluşan yeni kümeye rasyonel sayıların eklenmesiyle rasyonel sayılar kümesi ve oluşan bu yeni kümeye de irrasyonel sayıların eklenmesiyle reel sayılar kümesi oluşturulabilir. Bu tip iyi sıralı kümeleri temel kabul etmek suretiyle birçok matematiksel alanın kurulumu sorunsuz gerçekleştirilebilir. Daha ileri gidilerek, yüksek matematik gerektiren Analiz veya İleri Analiz ve daha fazlası kurulabilir. Hilbert, bu yaklaşıma "genetik yöntem" (Kline, 1972c: 1009) adını vermişti. Gerçekten de aksiyomatikleştirilmesi tamamlanmış gibi görünen doğal sayılar kümesine yeni aksiyomlar dahil edilerek ve bu aksiyomlar arasında oluşabilecek çıkarımsal aksiyomlar belirlenerek daha geniş küme yapılarına ulaşılabilir. Bu tip aksiyomlar arasında her zaman çıkarım yakalamak mümkündür.

Bu yaklaşımın mantıksal temeli, örneğin Peano'nun aksiyomları gibi, yalnızca doğal sayılarla ilgili bazı iddialar dizisidir. Buradan diğer tüm sayılar oluşturulur. Hilbert, yukarıdaki yaklaşımı genetik yöntem olarak adlandırdı. (...) Genetik yöntemin pedagojik veya sezgisel değere sahip olabileceğini kabul ediyor, ancak tüm gerçek sayı sistemi için aksiyomatik yöntemi uygulamanın mantıksal olarak daha güvenli olduğunu söylüyordu (Kline, 1972c: 1009).

Hilbert'in aritmetiğin tutarlılığı sorgulaması, yukarıda serimlemeye çalıştığımız üzere kümelerin ek aksiyomlarla genişletilebileceği fikrinin tersi yönde işletilerek daha sınırlı aksiyomları olması nedeniyle doğal sayıların aritmetiğine (sayı teorisi) indirgenebilir olması üzerinden yürütülmüştür. Leibniz'de ilk örneklerine rastladığımız indirgemeci yöntem olan Hilbert'in metodu açıkça şunu ister: Örneğin, geometride kullanılan aksiyomların, $1=0$ gibi bir çelişkiyi doğurmayacak şekilde bir tamlığa sahip olmasını gerektirir (Cryan vd., 2013: 52).

Burada önemli olan unsur şudur: Matematiğin her alanı, kendi doğru olduğu bilinen (varsayılan) aksiyomları üzerinden ispatlanabilmelidir yani kendi içinde kalmalıdır (Cryan vd., 2013: 50). Hilbert'in bir tanımını verdiği görülen bu yöntem, aslında Hilbert'in adıyla bilinen ispat teorisi (proof theory) ya da başka bir ifadeyle *metamatematiktir*. Her şeyin matematik içinde kalarak, üzerinde çalışılan branşın aksiyomları dışında başka hiçbir sistemden veya kümeden aksiyom (belit) alınmadan sonlu bir aksiyom sayısıyla sonlu bir işlem gerektiren bir ispat yöntemidir.

İstenen tutarlılık kanıtı metamatematik içinde gerçekleştirilecekti. Biçimsel sistem içinde her türden matematiksel yöntemin en eksiksiz ve sınırsız kullanımına izin verilirken, metamatematiksel yöntemler, tartışmasız bir şekilde Hilbert'in "sonlu" olarak adlandırdığı yöntemlerle sınırlandırılacaktı (Davis, 2018: 82).

Bu ispat yöntemi, matematik dünyasında *biçimselci* (formalist) görüş olarak bilinen ve Hilbert'e atfedilen bir yaklaşımın sonucunda ortaya çıkmıştır. Bu yaklaşım gereği, çelişkiye neden olması muhtemel anlam içeriklerinin terk edilmesi esastır, dolayısıyla anlamdan soyutlanmış mekanik bir işlemler zinciri söz konusudur. Buradan görüleceği üzere Hilbert, matematiğin mantıkla yürütülmesiyle değil, "ispat teorisi"nin matematiksel versiyonuyla ilgilenir (Cryan vd., 2013: 84), yani matematiksel bir ifade ancak ve ancak matematiğin kendi diliyle ispatlanabilmelidir (Cryan vd., 2013: 84). Ancak bu yolla, matematiksel dil kendine yeter düzeyde olabilir. "Buradaki fikir, matematiğin bahsettiği şeylerin sembollerden başka bir şey olmadığıdır. Bu semboller, kendi başlarına anlamsızdır. Onları nasıl manipüle edeceğinizi bildiğinizde, onlar hakkında her şeyi bilirsiniz" (Cryan vd., 2013: 85). Hilbert, ayrıca bu işlemler zinciri içindeki "olası etkileşimleri açıklamak için özyinelemeli kurallar dizisini" (Cryan vd., 2013: 85) tanımlamıştır. Böylece, sonlu adımlı bir ispatta gerçekten sistemin dışına çıkmadan, gerektiğinde yine sistemin kendisinde bulunan başka bir aksiyom ya da formülden faydalanılması sağlanmış olur.

Hilbert'in biçimselci yaklaşımında böylece, kelimeler ve sentakstan müteşekkil bir dil ortaya çıkar hem de diğer tüm dil yapılarından soyutlanmış olarak. Bu dilin cümleleri yani aksiyomları, formülleri veya teoremlerinin tamamı doğru cümlelerden oluşmalıydı. Denilebilir ki Leibniz'in hayalini kurduğu rasyonel dil de pasigrafik olarak tam da böyle bir dil yapısına sahip olmalıydı.

Metamatematiğe dayalı biçimsel dilde, o dönemde kısmi bir çözümüne ulaşılan aritmetiğin tutarlılığı sorunu hala tam bir çözüme ulaşamıştır. Henüz bu sorunun verilmiş yanıtları tatmin edici değilken, Hilbert başka sorular soracağı sistemler üzerinde çalışmaya başlar. Günümüzde birinci seviye yüklem mantığı olarak yerini bulmuş Frege'nin *Kavram-*

Yazısının ve Peano'nun doğal sayılar üzerindeki aksiyomatizasyonunun temel teşkil ettiği Russell ve Whitehead'ın ideografik dilde yazılmış Principia Mathematica'sı Hilbert'in araştırma konusuna dönüşür. Hilbert'in çalışmalarından sonra,

Frege, iki dil düzeyiyle yani inşa ettiği yeni bir biçimsel dille ve bu yeni dilin tartışılabileceği sıradan dille uğraştığını açıkça anlamış olsa da Whitehead-Russell'in başyapıtı, bu konuda belirsizdi ve iki düzeyi birbirine karıştırıyordu. Bu, Hilbert için çok önemli olan tüm yapının tutarlılığı sorununun Russell bağlamında çözülemeyeceği anlamına geliyordu (Davis, 2018: 77-8).

Mantığın ve dilin düzeyleri arasındaki bulanıklığa işaret eden Hilbert ve öğrencisi Ackermann tarafından, 1928 yılında Hilbert'in 1917 yılından itibaren verdiği mantık derslerine dayanan bir kitap olan *Matematiksel Mantığın İlkeleri* (İng. Principles of Mathematical Logic) yayımlandı. Bu kitapta yalnızca mantık dersleri bulunmuyordu, ayrıca mantığın günümüzde aşına olduğumuz şekline kavuşmasını sağlayacak çalışmalar da mevcuttu. Buna göre, elde bulunan yüklem mantığına evrensel ve varoluşsal niceleyiciler eklenerek ikinci seviye mantık elde edilebilirdi. Hilbert'in genetik yönteminden de bildiğimiz üzere, bu tür bir mantık seviyelendirmesi, birinci seviye yüklem mantığından başlamak üzere genişleyerek sürdürülebilirdi.

Böylece, *evrensel ve varoluşsal niceleyicileri cümle ve yüklem değişkenlerine de uygulayarak ve bu türden serbest ve bağlı değişkenler arasında ayırım yaparak*, birinci seviyeden yüklem hesabının doğal bir uzantısına yönlendiriliriz. Mantıksal formül kavramı (...), daha sonra karşılık gelen bir genişlemeye uğrar. Bu şekilde, *ikinci seviyeden bir yüklem hesabına* ulaşırız (Hilbert ve Ackerman, [1928]1950: 125 italik vurgular Hilbert ve Ackerman'a aittir).

Hilbert ve öğrencisi Ackermann bu kitapta, matematiğin matematiksel mantık veya sembolik mantık olarak adlandırılabilen bağlantısallığı neticesinde, içerimsel olarak cümlesel hesap (İng. sentential calculus) yapabilen bir disipline dönüştüğüne ([1928]1950: 3) vurgu yaparak, doğruluk ve yanlışlığın bu cümlesel hesap formunda anlamlı bir şekilde aktarılabilirdiğini ifade ediyorlardı. Bu kitapta, cümlesel hesap formunda bir kanıtlarının talep edildiği iki çok önemli soru bulunmaktaydı. Bu sorulardan ilki, aritmetiğin temelleri üzerineydi. Aslında Hilbert'in aritmetiğin temelleri üzerine sorduğu sadece bir tane gibi görünen fakat gerçekte iki adet olan soru vardı: Bunlardan ilki, 1900 yılında konferansta yaptığı konuşmasında geçen ikinci soru yani aritmetiğin çelişkisiz olduğuna dair kanıt; ikinci soru ise 1928 yılında kaleme alınmış Hilbert ve Ackermann'nın *Matematiksel Mantığın*

İlkeleri adlı kitaplarının konusu, *Principia Mathematica*'nın tamlığı üzerineydi. Hilbert ve Ackermann,

[m]antık üzerine 1928 tarihli ders kitabında *Principia Mathematica*'nın kurallarında boşluklar olup olmadığı sorusunu sormuştur, yani doğru olması gereken, ancak kuralların öncüllerden sonuç çıkarmak için yeterli olmadığı tümdengelimli çıkarımlar. Böyle bir boşluk olmadığına inanıyordu, ancak durumun böyle olduğuna, kuralların eksiksiz olduğuna dair bir kanıt istedi (Davis, 2018: 90-91 italik kısım benim yaptığım eklemelerdir).

Ackermann ile birlikte yazdıkları kitaplarında ([1928]1950: 153), yaptıkları çalışmaların sonucunda belli bir noktaya kadar gelmiş olsalar da Hilbert, metamatematiksel olarak tatmin edici bir kanıtın peşindeydi. Aynı kitapta, tezimiz açısından oldukça önemli bir yer tutan diğer bir soru ise Hilbert'in ünlü "karar problemi" (Entscheidungsproblem) olarak biliniyordu. Bu sorulara aranan yanıtlar metamatematiksel olarak verilmeliydi. Hilbert'e göre, tüm bu sorular için bulunacak yanıtlar, matematiğin lehine olacaktı fakat Hilbert programı, 1920'den başlamak üzere 1931 ve 1936 yıllarında sırasıyla alacağı üç darbe sonucunda çökmüştür.

2.3.2.1. Hilbert Programının Çöküşü

Hilbert'in umudu, yukarıda sözünü ettiğimiz soruların olumlu şekilde bir yanıt bulacağı yönünde olsa da gerçekler tam aksi yönde gelişmiş, mutlak bir mantıkla sarılması hedeflenen matematiğin temelleri çok daha fazla sorgulanır hale gelmiştir. Bu sorgulamaya neden olan ilk darbe, LST olarak kısaltacağımız (1915 yılında Leopold Löwenheim [1878-1957] ve 1922 yılında Thoralf Skolem'in [1887-1963] katkılarıyla) Löwenheim-Skolem teoreminin sonuçlarından doğmuştur. Bu sonuçlar, Hilbert programına bir darbe olduğu yerde, ayrıca matematiğin temellerinin de LST'nin gölgesinde kalmasına neden olmaktadır. LST, birinci seviye mantık içinde kalan matematiğin temelleri dahil olmak üzere, sınırlı sayı ile belirlenmiş herhangi bir aksiyomlar kümesinden elde edilen sistemin model-kuramsal yapısına vurgu yapmaktadır. Skolem'in 1922'de ulaştığı (kimilerince bir paradoks olduğu düşünülen) sonuçlarla birlikte LST, aşağı ve yukarı olmak üzere genellikle iki ayrımının olduğu bir teoremdir. LST'ye göre, "her σ simgesi için, her sonsuz σ -yapıda M ve her sonsuz kardinal sayısı $\kappa \geq |\sigma|$ için, öyle bir σ -yapıda N vardır ki $|N| = \kappa$ olmak üzere" (doğrudan alıntı ve aşağıdaki iki gösterim [Nourani, 2014: 160]'dan alınmıştır);

1. Eğer $\kappa < |M|$ ise N , M 'nin temel bir alt yapısıdır (Aşağı LST).
2. Eğer $\kappa > |M|$ ise N , M 'nin temel bir uzantısıdır (Yukarı LST).

Biz burada bu teoremin “aşağı” ve “yukarı” ayrımına girmeden, tezimiz açısından bu bölümde ZFC kümeler kuramı ve dolayısıyla Hilbert programı üzerindeki etkisini serimlemeye çalışacağız. LST, aksiyomatik sistemlerde kullanılan ve sonlu-sınırlı bir söylem evreni içinde kalan aksiyomlardan herhangi birinin veya bazılarının, herhangi bir özellik veya formül için sayılamaz sonsuzlukta bir küme olarak sonsuz döngüye girmesi durumunda, aksiyomatik sistemin referans olduğu yapıda “model kuramsal tutarlılık” sağlanamayacağı, başka bir deyişle “çelişmezliğin” sağlanamayacağını vurgulayan iki yönlü (aşağı ve yukarı olmak üzere) versiyonun bulunduğu bir teoremdir. Bu teoremin teknik açılımına girmeden ifade etmek gerekirse, LST’ye göre bir kuramdan istenen şey, mümkün olduğu ölçüde kuramın tek bir modelinin olmasıdır. Şayet kuram başka modellerle de betimlenebiliyorsa, bu durumda bu modeller eşyapılı yani izomorfik olmalıdır. “Bir aksiyomatik kuramın aksiyomlarını gerçekleştiren bütün yapılar izomorfikse bu tür kuramlara kategorik kuramlar denir” (Öztürk, 2015: 1723). Dolayısıyla, kuramın kategorik olmaması, “model kuramsal” çelişkiye neden olacaktır. “Çünkü bu kuramın aksiyomlarını gerçekleştiren yapıların yapısal olarak farklılıklar barındırması sebebiyle bir yapıda doğru olan en az bir önerme başka bir yapıda yanlış” (Öztürk, 2015: 1723) olacaktır.

Matematiğin mantığı olarak bilinen “ilişkiler mantığı”nın (Couturat, 2004: 89) yanı sıra matematiğin temelleri için tercihen birinci seviye mantıkla çalışan sistemler kullanılmaktadır. Bunlara verilecek örnekler arasında genel bir kullanıma sahip Peano aksiyomları ve ZFC küme kuramı gibi aksiyomatikleştirmiş sistemler bulunmaktadır. Doğal sayılar kümesinin tanımını vermek için kullanılan Peano aksiyomları, birinci seviye mantıkla çalışmasına rağmen, özelliklere vurgu yapan “1 sayısına ait bir özellik, 1 ve diğer her sayının ardılına da ait bir özelliktir; böylece tüm sayılara aittir” şeklindeki son aksiyomu bir aksiyom değil, aslında bir “tümevarım kuralı” olarak görülmelidir.

Dikkat edilebileceği gibi aksiyom belirli bir özellik (veya matematiksel formül) hakkında değil, doğal sayıların sahip olabileceği bütün özelliklere işaret ederek bir nicelemede bulunmaktadır. Bu nedenle bu yargı ancak ikinci seviye bir mantık dili içerisinde tam olarak biçimselleştirilebilir (Öztürk, 2015: 1716).

Sonsuz bir döngü yaratabilecek olan bu aksiyomlar topluluğunun dışında diğer aksiyomların, kendilerini ve birbirlerini iyi betimliyor olmaları gerekir. Tümevarım kuralının, geri kalan diğer Peano aksiyomlarına eklenmemesi durumunda, bu aksiyomların başka bir kurama referans olmayacağına bir garantisi yoktur. Bu nedenle, LST ile anlaşılması gereken husus, “sayılabilir olmayan bir modeli olan hiçbir biçimsel sistemin kategorik olmadığı”

(George, 1985: 76) yönündedir. Bu durumda, sonlu sayıda Peano aksiyomuna eklenmiş olan “tümevarım kuralı” ile birlikte değerlendirildiğinde, bu kuram ne şekilde *kategorik* sayılabilmektedir? Eklenen “tümevarım kuralı”, içinde çalışılan kümenin özelliklerinin, kümenin tüm elemanları için geçerli olduğunu söyleme gücünde olduğundan dolayı elzemdir lakin burada başka bir sorun ortaya çıkmaktadır. Peano aksiyomlarının “tümevarım kuralı” birinci seviye mantıkla değil, *özelliklere* işaret ettiğinden dolayı ikinci seviye mantıkla çalışmaktadır. O halde karşımıza LST’nin gölgesinde kalan matematik temellerine dair bir sorunsal çıkmaktadır. Bu sorunsalın meali şudur:

- 1) LST’nin hedefinde olan birinci seviye mantık içinde kalınarak, içeriden ve dışarıdan kuramı işaret eden sayılabilir aksiyomatik sistemler dışında “model-kuramsal tutarlılık” ispatlanamaz, başka deyişle çelişkisizlik ispatlanamaz;
- 2) LST’nin hedefine girmeyen ve *özelliklerin* daha esnek ifade edilebildiği ikinci seviye mantık içinde kalınarak, sonsuz döngüye girme sorunu ortadan kalkacak olsa da ikinci seviye mantık tam değildir⁴³ ve bu nedenle matematikçiler tarafından birinci seviye mantık tercih edilmektedir⁴⁴.

LST, matematiğin temelleri için kabul görmüş birinci seviyeden mantıkla çalışan sistemlere dokunduğundan dolayı yukarıda serimlemeye çalıştığımız üzere, Peano’nun aksiyomatik sistemi kendisini bu sorundan kurtaramamaktadır ve aynı durum, LST’den kaçamayan birinci seviye mantıkla çalışan diğer sistemlerde olduğu gibi ZFC kümeler kuramı için de geçerlidir. Örnek vermek gerekirse, ZFC kümeler kuramının “tanımlanabilir altküme

⁴³ İkinci seviye mantığın, birinci seviye mantıkla kıyaslandığında ne şekilde bir eksiklik barındırdığına dair tartışmanın sonucuna göre bu iki ayrı seviyeden mantık sistemi arasında temelde duran belli türden ayrımlara gidilmiştir: “Bu ayrım temelinde mantık sistemleri incelendiğinde, önermeler mantığı tamdır ve onun tamlığı 1918’de Paul Bernays tarafından kanıtlanmıştır. Yani önermeler mantığı, kendi içerdiği semantik sonuç ilişkisini bütünüyle ele geçirebilecek sonlu bir biçimsel çıkarım kuralları kümesine de sahiptir. Birinci seviye mantık da tamdır ve onun tamlığı 1929 yılında mantıkçı-matematikçi ve filozof Kurt Gödel tarafından kanıtlanmıştır. Dolayısıyla bu iki mantık sistemi, en azından kuramsal olarak, güçlü birer ispat kuramı sağlayabilmektedir. Ancak diğer taraftan ikinci seviye mantık tam değildir: İkinci seviye mantık, 1931 yılında ortaya çıktığı üzere, kendisine ilişkin standart semantiğin belirlediği semantik sonuç ilişkisini tam olarak ele geçirebilecek bir biçimsel kurallar kümesine sahip değildir” (Öztürk, 2015: 1722).

⁴⁴ Günümüzde henüz tartışmaları devam eden LST, matematiğin güvenli fakat kısıtlı bir evren sayılan birinci dereeden mantığın söylem evreni içinde kalması mı yoksa ilişkiler ve özellikleri biçimselleştirme konusunda daha kullanışlı olsa da henüz eksik olan ikinci seviyeden mantığın söylem evreniyle mi yürütülmesi gerektiği konusunda matematiksel, mantıksal ve felsefi soruları beraberinde getirmiştir. “Bugün biliyoruz ki, daha yüksek ifade gücüne sahip olmasına rağmen ikinci seviye mantığın gerek pedagojik yapıtlardaki yeri gerek matematiksel pratikteki uygulama alanı ve gerek doğal dil argümanlarının sınanması açısından uygulama alanı birinci seviye mantığa göre karşılaştırılamayacak kadar düşüktür. Buna yol açan nedenlerin başında matematikçilerin ve felsefecilerin kendi alanlarına ilişkin çalışmalarda birinci seviye mantığa ikinci seviye mantığa göre daha büyük bir araçsal rol vermesi gelmektedir. Dahası bugün çoğu felsefeci ve matematikçi için mantık kavramının kaplamı, ikinci seviye mantığı dışta bırakacak şekilde, yalnızca birinci seviye mantık olarak belirlenir. Bu gruba karşıt olarak pek çok mantıkçı ve felsefeci ise ikinci seviye mantığın da mantıksal bir sistem olduğunu savunmaktadır” (Öztürk, 2015: 1717).

aksiyomu” olarak da bilinen üçüncü aksiyomu, φ özelliğini taşıyan bir x kümesinin φ özelliğini sağlayan elemanlarından başka eleman içermeyen bir kümenin varlığına işaret eden yalnızca bir değil, aksine sonsuz sayıda aksiyom içermektedir. Aynı şekilde ZFC kümeler kuramının “yerleştirme aksiyomu” olarak bilinen dokuzuncu aksiyomu da bir sonsuz döngü yaratmaktadır. O halde birinci seviye mantık sınırları içinde kalan ZFC küme kuramının (çelişkili olduğu ispatlanamamış olsa da) çelişkisizliği ispatlanamaz.

İlk darbeyi, mantıkçılar tarafından kabul görmesi zaman almış, LST kısaltmasıyla tezimizde yer verdiğimiz Löwenheim-Skolem teoreminden alan Hilbert programı, ikinci ve daha ağır darbeyi Gödel’in eksiklik teoremlerinden almıştır. LST’den darbe almış “Zermelo, sisteminin çelişkisiz olduğunu kanıtlamaya çalışmışsa da başaramamıştır. Başaramamasının nedeni vardı: Sistemin çelişkisiz olduğunun kanıtlanamayacağını bugün Gödel sayesinde biliyoruz” (Nesin, 2020: 163).

Henüz yirmi beş yaşındaki Gödel’in 1931 yılında “Principia Mathematica ve İlişkili Dizgelerin Biçimsel Olarak Kararlaştırılamayan Önergeleri Üzerine – I” (On Formally Undecidable Propositions of Principia Mathematica and Related Systems-I) adıyla yazdığı makalesi, Hilbert’in bütün programının özü itibarıyla çökmesine; Gödel’in ise “yirminci yüzyılın en büyük mantıkçısı” ünvanıyla anılmasına neden olmuştur. Peki öyleyse Gödel, kendinden önce gelen mantıksal-matematiksel koca bir birikimin bütünleştiricisi ve sağlamalaştırıcısı olma rolünün verildiği Hilbert programını nasıl yıkmış olabilirdi?

Hilbert’in programı, adım adım ilerleyen ve her adımda kendini açıkça gösteren (ifade edebilen) bir kanıtın peşindeydi. Bu kanıt elbette ne Frege ne Peano ne de Russell ve Whitehead bağlamında sunulmalıydı. Çünkü bu sistemler, Hilbert’e göre kullandıkları dil dolayısıyla verilen kanıtın anlaşılabilirliğini bozabilirdi.

Gödel, Hilbert’in düşüncelerine ve çalışmalarına yabancı değildir. Hilbert’i Bologna’da bir seminerde dinleyen Gödel, 1930 yılında yazdığı doktora tezinde, birinci seviyeden kalkülüs önermelerinin eksiksizliğini kanıtladığı konuyu bu yolla seçer. Gödel 1931 yılında ise Hilbert’in 1900 yılında sorduğu ikinci soruyu seçerek makalesinde, Hilbert’in metamatematiksel yöntemini kullanır, böylece matematiğin kendisi üzerine konuşarak yorum yapabilmesini sağlar ve sonuç itibarıyla Gödel’in bu makalede sunduğu sonlu sıralı kanıtlar, Hilbert’in kabul etmek zorunda kalacağı kendi biçimsel diliyle yazılmış olur. Aslında Gödel’in bu işlemleri yansımali şekilde gösterebilmek için kendi dilini yarattığı da söylenebilir. Gödel,

[b]u çalışmada, Russell ve Whitehead tarafından kaleme alınan üç ciltlik *Principia Mathematica*’nın kendi kendisinin konusu olduğunu, başka bir deyişle *Principia Mathematica* sisteminin formüllerinin

birbirleriyle ilgili konuşan ya da kendileri hakkında konuşan formüller gibi olduğunu göstermeyi başarmıştır (Hazar, 2017: 32 italik vurgular bana ait).

Gödel, üzerinde çalıştığı *Principia Mathematica*'nın kendisi hakkında konuşmasını sağlayabilmek için Richard paradoksu⁴⁵ olarak bilinen paradokstan faydalanmış ve belli başlı dönüşüm kurallarını sayılaştırma üzerinden belirlemiştir. Richard paradoksu, Gödel kanıtlamasında kilit role sahiptir çünkü Gödel, *Principia Mathematica*'da “ben kanıtlanamam” diyen doğru bir önermenin veya bir tamdeyimin peşindedir. Şayet “ben kanıtlanamam” tamdeyimi kanıtlanırsa sistem tutarsız olacaktır, kanıtlanamazsa sistem tutarlı kalacaktır.

Gödel, Richard paradoksunda olduğu gibi sayılaştırma yoluyla, tüm bilinmeyenler ve sabitler ile 1 sayısından başlamak üzere, doğal sayılar arasında bire bir eşleme yapabilmiş ve *Principia Mathematica*'da çelişkiye varabileceği şekilde ürettiği açık ve net ifadeleri takip edebilmiştir. Gödel'in kullandığı üst dili, her satırı bir işlem tanımına çevirmeyi başaran işlevsel bir dildir. Böylece öncelikli olarak, *Principia Mathematica*'nın biçimsel dizgesinde sentaksın oluşumunu sağlayan *sabit imleri* ile sayı eşleştirmesi yapmıştır: “~”: 1, “V”: 2, “→”: 3, “∃”: 4, “=”: 5, “0”: 6, “s”: 7, “(“: 8, “)” :9, “,”: 10, “+”: 11, “x”: 12. Sayılaştırmanın 1'den başladığına dikkat etmekte yarar var. Bu tip eşleştirmelerde şayet çarpım ile verilecek bir aritmetik hesap var ise 0 sayısı çözümsüzlük getirir.

⁴⁵ Richard paradoksu (1905), sayılabilir bir kümede aritmetiksel özelliklerin verilmiş tanımlarının numaralandırılması fikriyle başlar: “Bu tanımlardan her birinin ancak sonlu sayıda sözcük içereceklerini, dolayısıyla sonlu sayıda abece harfi içereceklerini rahatlıkla görebiliriz. Buna göre tanımları dizisel bir sıraya göre yerleştirebiliriz. Eğer bir tanımdaki var olan harflerin sayısı, bir diğer tanımın harflerinin sayısından küçükse, ilk tanım diğerinden önce yazılacaktır ve eğer iki tanım da aynı sayıda harfe sahip ise bunlardan hangisinin önce geleceği, her birinin sahip olduğu harflerin abecedeki sırasına bağlıdır. Bu sıra temel alınarak her bir tanıma bir tek tamsayı karşılık gelecektir ve tanımın dizideki yerini temsil edecektir. Örneğin, en az sayıda harfe sahip tanım 1 sayısına karşılık gelecek, dizide bundan sonra gelen tanım 2 sayısına karşılık gelecek ve bu böyle sürecektir” (Nagel ve Newman, 2010: 78-79). Daha sonra görülecektir ki gerçekten her tanıma bir tamsayı denk gelmektedir. Öyle ise “bazı durumlarda tanımın belirttiği özelliklerle ona karşılık gelen tamsayının aynı özelliklere sahip olduğu durumlar olabilir. Örneğin, “1'den ve kendinden başka hiçbir tamsayıyla bölünemez” tanımındaki deyimın sıra sayısının 17 olduğunu varsayalım; 17'nin kendisinin tanımdaki deyimle aynı özelliğe sahip olduğu açıktır. Öte yandan, “bir tamsayının kendisiyle çarpılmasından elde edilen” tanımındaki deyimın sıra sayısının 15 olduğunu varsayalım; 15'in deyimde belirtilen özelliğe sahip olmadığı da açıktır. İkinci örnekteki durumu, 15 sayısının Richardcı özelliğe sahip olması biçiminde betimleyelim; bu durumda ilk örnekteki 17 sayısı da Richardcı olma özelliğine sahip olmayacaktır. Genel olarak, “x'in Richardcı olması”nı “x'in tanımlar kümesinde karşılık geldiği sayının tanımda belirtilen özelliğe sahip olmaması” olarak kısaca ifade edebiliriz” (Nagel ve Newman, 2010: 79). Bundan sonrasında bize Russell paradoksunu hatırlatacak bir sorunun geleceğini kestirebiliriz: “Richardcı olma özelliğini tanımlayan deyim açıkça tamsayıların sayısal özelliklerini betimlemektedir. Dolayısıyla tanımın kendisi de yukarıda sözü edilen tanımlar dizisine aittir. Yani bu tanımın kendisine de karşılık gelen ve onun yerini belirleyen bir tamsayı vardır. Bu sayının n olduğunu varsayalım. Şimdi Russell paradoksunu anımsatan bir biçimde şu soruyu soralım: n Richardcı mıdır? (...) Çünkü n, yalnız ve yalnızca karşılık geldiği tanımlayıcı deyimın belirttiği özelliğe sahip değilse Richardcıdır (yani n Richardcı olma özelliğine sahip değildir). Kısaca, n Richardcıdır yalnız ve yalnızca n Richardcı değilse; yani n Richardcıdır önermesi hem doğrudur hem de yanlış (Nagel ve Newman, 2010: 79-80).

Gödel daha sonra, *Principia Mathematica*'dan seçtiği sayısal değişkenlere dönüşebilecek cebirsel bilinmeyenleri, sabit imlerin sayılaştırmasının bittiği en son sayının en yakını olacak ilk asal sayıyla eşleştirmiş ve bu eşleştirmeyi asal sayılar üzerinden sürdürmüştür: Sayısal değişkenler olan “x”, “y” ve “z” cebirsel ifadeleri burada sırasıyla 13, 17 ve 19 asal sayılarıyla eşleştirilir. Dikkat edilecek olursa, aynı sayı farklı eşleştirmelerde kullanılmamaktadır. Bu yolla işlem sırası kolaylıkla takip edilebilen bir kodlama gerçekleştirilmiş olur.

Burada, Nagel ve Newman'ın *Gödel Kanıtlanması* adlı kitabından aldığımız örnekle devam ederek, *Principia Mathematica*'nın bir tamdeyimi olan, “ $(\exists x) (x=sy)$ ” (2010: 89) cümlesini ele alalım. Bu tamdeyim tam olarak şunu ifade etmektedir: Öyle bir x vardır ki bu x, y'nin ardılıdır (2010: 89). Burada rahatlıkla görülecektir ki “y”nin yerine hangi sayı gelirse gelsin, bir sayı ardılı olacaktır. Bu durumda, Gödel sayılarıyla yapılacak eşleştirme aşağıdaki gibi görünecektir:

$$\begin{array}{cccccccc} (\exists x) (x = sy) \\ \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow \\ 8 \ 4 \ 13 \ 9 \ 8 \ 13 \ 5 \ 7 \ 17 \ 9 \end{array}$$

Sonrasında ise Gödel, *Principia Mathematica*'daki herhangi bir eşitlik ya da cümleyi bu sayısal eşleştirme üzerinden göstererek, kendi üst dilini, *Principia Mathematica*'nın içindeki formüllerin kendi aralarında konuşabilecekleri bir dile çevirmiştir. Aslında bu yöntemde çok daha fazlası vardır: İçeriden bakıldığında tündengelimli işlemler silsilesi olan ve sadece matematik denilen şey varken, dışarıdan bakıldığında “anlam dikkate alınmadan ele alınabilecek çok sayıda sembol manipülasyonu” (Davis, 2018: 82) vardır. Gödel, özetle dışarıdan bakıldığında, içeride ne olduğunun ortaya çıktığı bir dil sistemi kurmuştur. Gödel burada, şöyle devam edecektir. En küçük asal sayıdan başlamak üzere belirlenmiş olan Gödel sayıları bu asal sayıların üssü olarak yazılacak ve sonuçta tümünün çarpımı elde edilecektir.

$$2^8 \times 3^4 \times 5^{13} \times 7^9 \times 11^8 \times 13^{13} \times 17^5 \times 19^7 \times 23^{17} \times 29^9 = m$$

Gödel, bu yolla *Principia Mathematica*'da bulunan her im, her tamdeyim, her sabit veya her cebirsel değişkeni, “m” gibi tek bir Gödel sayısı ile gösterebileceğini kanıtlamıştır. Bu formülasyonda çarpımın asal sayılar üzerinden, başka bir deyişle asal çarpanlar üzerinden tercih edilmiş olması ve ayrıca yukarıda görülen eşleştirmede oluşan tüm Gödel sayılarının,

sırasıyla en küçük asal sayıdan başlamak üzere, bir üs olarak kullanılması bugün şifreleme alanında kullanılan bir yöntem olarak göze çarpmaktadır.

Gödel, aşağıda görüleceği üzere kurduğu metamatematiksel dilin bir sağlamasını vererek, bu yolla tam bir mekanikleştirmeye ulaştığını göstermiştir. Nagel ve Newman'ın kaleme aldığı *Gödel Kanıtlanması*'ndan bir örnekle gösterecek olursak (2010: 93);

A	243.000.000
B	$64 \times 243 \times 15.625$
C	$2^6 \times 3^5 \times 5^6$
D	656 ↓↓↓ 0=0
E	0=0

Böylece, Gödel bir G tamdeyimine karşılık gelecek bir g Gödel sayısı ile oldukça ağır ve anlaşılması güç bir uslamlama yaparak, Hilbert'in aritmetikte tutarlılık ve *Principia Mathematica*'da tamlık sorularına metamatematiksel bir yanıt vermiştir. Richard paradoksundan faydalanan Gödel uslamlamasının gelmek istediği yer, açıkça çelişki elde edebileceği bir tamdeyim elde edebilmektir.

Gödel uslamlamasının devamını burada ele almadan, özetle çıkarımını gerçekleştirdiği sonuçlara gelecek olursak, *Principia Mathematica* tutarlı formel bir yapıya sahip ise ne G ne de $\sim G$ formülleri ispatlanabilir. Bu durumda, çelişik olmayan bir sistemde her zaman doğru olan ve karar verilemeyen bir G tamdeyimi mevcuttur. Sistemin dışından alınacak başka bir tamdeyim ile şayet tutarlılık sağlanıyor ise bu durumda sistem tam değildir çünkü Hilbert'in ispat teorisine göre, sistem kendi içindeki aksiyomlarla tutarlılığını sağlamalıdır. Böylece Gödel'in eksiklik teoremleri aşağıdaki gibi özetlenebilir:

- Toplama ve çarpmaya dair temel aritmetiği kapsayacak kadar güçlü biçimsel bir dizgenin içinde kararsız önermeler (veya hipotezler) vardır. Bu önermelerin (veya hipotezlerin) doğrulukları ispatlanamadığı gibi yanlışlıkları da ispatlanamaz (Gödel'in Birinci Eksiklik Teoremi);
- Yeterince güçlü bir biçimsel dizgenin tamlığı dizgenin kendi içinde ispatlanamaz (Gödel'in İkinci Eksiklik Teoremi).

Böylece, Hilbert'in büyük umutlar bağladığı programının, ilk darbeyi aldığı Löwenheim ve Skolem tarafından, ikinci darbeyi aldığı Gödel tarafından özü itibarıyla çöktüğünün bir resmi verilmiş olur. Hilbert programının aldığı son darbeye geçmeden önce,

Gödel'in eksiklik teoremlerinden sonra mutlak bir mantıkla tamamen biçimsellik kazandırılmaya çalışılan matematikte, ispat ve doğruluk kavramlarının nasıl karşılandığına bakmakta yarar var.

Bu sonuçlar, her tümdengelimli dizge için (özellikle tüm aritmetiğin ifade edilebileceği dizgeler için) Hilbert'in önerisinin sonlu gereklerini sağlayan bir mutlak tutarlılık kanıtlanmasının, mantıksal olarak olanaksız olmasa da pek olası olmadığını ortaya koydular. Ayrıca bu sonuçlar, herhangi bir aksiyom kümesinden belirli çıkarım kurallarıyla biçimsel olarak çıkarılamayacak sonsuz sayıda doğru aritmetik önermesi olduğunu da gösterdiler. Bu sonuçlardan sonra, örneğin sayılar kuramına aksiyomatik bir yaklaşımın, aritmetiksel doğruluk alanını tüketemeyeceği anlaşılıyor. Ayrıca matematiksel kanıtlama sürecinden anladığımız şeyin, biçimselleştirilmiş aksiyomatik bir yöntemin kullanılmasıyla tam çakışmadığı da anlaşılıyor. (...) Sonuç olarak, geçerli matematiksel kanıtlamaların kesin mantıksal biçimi hakkında son söz söylenemez (Nagel ve Newman, 2010: 115).

Gödel'in eksiklik teoremleri, Hilbert'in çok önemli bir sorusunu kapsamamaktaydı. Bu soru, Hilbert ve Ackermann'nın *Matematiksel Mantığın İlkeleri* adlı kitaplarının konusu ikinci soru olan "karar problemi" (Entscheidungsproblem) olarak biliniyordu. Bu soru Frege'nin birinci seviye yüklem mantığı üzerinden yükselmekteydi ve özetle, "kalkülüsün belirli bir formülünün evrensel olarak geçerli olup olmadığına karar verme sorunu" (Hilbert ve Ackermann, 1950: 132) veya daha anlaşılır bir ifadeyle, birinci seviye yüklem mantığıyla betimlenmiş "herhangi bir mantıksal önermenin verilen belitler kullanılarak kanıtlanabilir olup olmadığını saptayabilen bir yöntemin" (Say, 2019: 31) ya da günümüz anlayışına göre bir "algoritma"nın varlığına dair bir sorgulamaydı. Böyle bir algoritmanın olmadığına dair yanıt, 1936 yılında Turing tarafından "Hesaplanabilir Sayılar: Karar Verme Probleminin bir Uygulaması" (On Computable Numbers, With an Application to the Entscheidungsproblem) adlı makalede tamamen mekanik bir yöntemle verilmiştir. Turing'in makalesinde verdiği yanıt, bizi matematiğin biçimsel dil ekseninden *makinelere* (Turing'in yaklaşımıyla *programların*) biçimsel dil eksenine kaydırmıştır.

Turing'in makalesinde "karar problemi" için bulduğu çözüm, Hilbert'in talebi doğrultusunda olmalıydı yani işlem, sonlu bir kurallar dizisi ile sonlu bir zamanda gerçekleşmeli ve biçimsel dizge içindeki herhangi bir önermenin doğru veya kanıtı olup olmadığı hakkında kesin bilgi vermeliydi. Özü itibarıyla insanlar hesaplamayı sonlu bir zaman tüketerek belirli bir kurallar dizisi ile yapmaktadırlar. Bu nedendir ki Turing "soruna, hesaplama yaparken insanların gerçekten ne yaptığını dikkate alan sağduyuya uygun bir bakış açısıyla yaklaştı" (Casti ve DePauli, 2004: 109) ve makalesinde hesaplama yapmakta olan bir

insanın hesaplama süreci ile sınırlı sayıda koşulu gerçekleştiren bir “makine” arasında benzeşim (Turing, 1937: 231) kurdu.

Turing’in “makine” terimini sonlu zamanda, sonlu işlemler dizisi yürüten bir *program* anlamında kullandığını eklemekte yarar görmekteyiz. Turing’in makalesinde geçen “makine” terimi, sonradan matematikçi Alonzo Church (1903-1995) tarafından “Turing makinesi” olarak adlandırılmıştır. Biz de tezimizde, günümüzde Turing’e atfen “makine” yerine kullanılan “Turing makinesi” için TM kısaltmasını kullanacağız. Eklemekte yarar gördüğümüz bir diğer husus ise TM’lerin (biri program olduklarını hatırlayacak olursak, bir değil her iş için ayrı birer Turing makinesi varsayılabilir) kesinlikle “maddi aygıtlar” (Casti ve DePauli, 2004: 113) olmadığıdır. Tezimizin ilerleyen sayfalarında sıkça rastlayacağımız TM’ler, günümüz bilgisayar bilminde “makine” terimine karşılık geldiği haliyle donanıma ilişkin herhangi bir unsur veya kavram olarak düşünülmemelidir.

Turing’in makalesinde modellemesini yaptığı bir TM üzerinden verdiği benzeşime göre bir insan matematikçi veya muhasebeci, bir *hesaplayıcı* (Turing’in kullandığı şekliyle, *computer*⁴⁶) olarak görev yapmaktadır. Turing’in makalesinde, hesaplamanın somut resmini veren bir TM⁴⁷ yani program için verdiği bazı tanımlar ve komutlar mevcuttur: Program tarafından yürütülecek hesaplama, her karesine yalnızca bir sembol gelecek şekilde tasarlanmış sonlu simgeler içeren tek boyutlu sonsuz bir bant üzerinde gerçekleşir, hesaplamayı yapan matematikçi (veya muhasebeci) her adımda yalnızca sembolün bulunduğu kareye odaklıdır, hesaplayıcının üzerinde bulunduğu karedeki sembol bir sonraki adımı belirler, hesaplayıcıya yön tayininde bulunan “L” sol, “R” sağ komutları, hesaplayıcının (veya teyp kafasının) mevcut durumunu sürdürmesi veya durmasını sağlayacak komutlar dizisi bulunmaktadır.

Bir Turing makinesi iki bileşenden oluşur: (1) sonlu bir simgeler kümesinden birini içeren karelere ayrılmış sonsuz bir bant ve (2) hesaplama işleminin her bir adımında sonlu sayıda durumdan veya düzenleştirmeden birinde olabilen bir tarama kafası. (...) Turing makinesinin davranışı bir algoritma ya da şimdiki adlandırmamızla, bir program tarafından kontrol edilir. Program sonlu sayıda komuttan oluşur;

⁴⁶ Alan Turing, “computer” terimini günümüzde aşına olduğumuz anlamda bir dijital (sayısal) bilgisayara gönderme yaparak kullanmamaktır. Turing’in makalesini yazdığı dönemde henüz bir dijital bilgisayarın somut hali ortada yoktur. Turing’in makalesinde geliştirdiği modelde kullandığı “computer” kelimesi aslında kanıt yapan bir matematikçidir (Çevik, 2021: 101).

⁴⁷ Hesaplamanın ne demek olduğunun somut bir modelini ilk defa Turing vermiştir. “Kuramsal açıdan Turing makinesi konusunda önemli olan onun biçimsel bir matematiksel nesneyi temsil etmesidir. Bu nedenle, Turing makinesinin yaratımıyla ilk kez, bir şeyi hesaplamanın ne anlama geldiğinin iyi tanımlanmış bir kavramına sahip” (Casti ve DePauli, 2004: 114-5) olduğumuz söylenebilir. Turing makinesi bir matematiksel soyutlamadır ve Turing’in tanımına göre hesaplama sürecinin her aşamasında, bir makinenin hareketi tamamen konfigürasyon (düzenleşim) tarafından belirleniyorsa, makineye “otomatik makine” (*a-machine* [*a*, “automatic” anlamında kullanılmaktadır]) denir (Turing, 1937: 232).

bu komutlar şu olasılıklar kümesinden seçilir: Kafanın içinde bulunduğu durumu sürdür veya değiştir; yeni bir simge yaz veya mevcut karedeki eski simgeyi muhafaza et; bir kare sağa veya bir kare sola hareket et; dur (Casti ve DePauli, 2004: 110).

Bir TM'nin (yani programın) $2+5=7$ gibi basit bir toplama işlemini yapma tarzını, Casti ve DePauli'den aldığımız bir örnek üzerinden göstererek ilerlemeye çalışacağız. Elimizdeki örnekte A, B, C gibi 3-durumlu bir TM var ve bant üzerine yazılan simgeler ise 0 ve 1 sayılarından oluşmaktadır. Aşağıdaki tablo, bir TM'nin yani programın komutlar dizisini vermektedir.

Tablo 2.1 Toplama için bir TM (program) örneği.

Durum	Okunan Simge	
	1	0
A	1, R, A	1, R, B
B	1, R, B	0, L, C
C	0, DUR	DUR

Kaynak: Casti ve DePauli, 2004: 111

Tablodaki komutlar dizisine göre, ilk giriş ögesi kafanın yazacağı simge olacaktır, ikinci öge kafanın hareket edeceği yönlerden herhangi biri olan R (sağ) veya L (sol) olacak ve üçüncü öge yani son öge ise kafanın içine gireceği durum olacaktır. Yukarıdaki tablodan görüleceği üzere, makine C durumuna girdiğinde “dur” konumuna gelmektedir. Makinenin okuyacağı sonsuz uzunluktaki girdi bandının iki 1 ve beş 1'i aralarına sıfır gelecek şekilde iki farklı sayı olarak gösterdiğini varsayalım. Bandın geri kalan tüm kısımları sadece 0'lardan oluşacaktır. Sonsuz bant aşağıdaki gibi görünecektir:

... 0 1 1 0 1 1 1 1 1 0 0 ...

Yukarıdaki tabloya göre, programın gerektirdiği algoritmayı yürütmek üzere, kafanın A durumunda başlarken soldaki ilk sıfır olmayan simgeyi okuduğunu varsayalım. Görüleceği üzere bu simge 1 sayısıdır. Şimdi, program makineye üzerinde bulunduğu kareye 1 yazmasını ve A içsel durumunu koruyarak bir kare sağa gitmesini söylemektedir. A içsel durumundan devam eden makine yeniden 1 simgesini yazar A içsel durumunda yeniden bir kare sağa gider. Makinenin iki 1 yazdıktan sonra girdi bandında gördüğü (okuduğu) simge 0'dır. Dolayısıyla, program makineye, 1 yazmasını ve sağa doğru bir kare ilerledikten sonra B durumuna geçmesini söyler. B durumuna geçen makinenin okuduğu simge 1'dir. Şimdi program makineye üzerinde bulunduğu kareye 1 yazmasını, bir kare sağa gitmesini ve B içsel

durumunu korumasını söyler. B durumunun dikkatle incelenmesinden görülecektir ki aynı işlem üç kez daha tekrar edecektir ve toplam beş 1'den oluşan dizi tamamlandığında, kafa hala B durumundayken 0 simgesini okuyacaktır. Dolayısıyla, program makineye 0 yazmasını ve bir kare sola gittikten sonra C durumuna geçmesini söyler. Tablodan görüleceği üzere, C durumuna giren makine otomatik olarak duracaktır. Böylece bant, yatay bir sırada yedi 1'e sahip bir gösterime kavuşacaktır. Buradan anlaşılacağı üzere $2+5=7$ toplamı gerçekleşmiştir.

Bir komutlar dizisi tarafından yönlendirilen bu tarz bir hesaplama prosedüründe, insan yerine bir makinenin getirilmesi halinde ne hesaplama süreci ne de çıktıdan alınması beklenen sonuç değişecektir. Öyle ise matematikçiler ve mantıkçılar tarafından bir çözüme kavuşturulması beklenen “karar problemi” için yürütülmesi gereken prosedür, bir insan yerine bir TM tarafından da gerçekleştirilebilir. Önceliği algoritmik hesaplamaların nasıl yapıldığını ortaya koymak olan Turing, bir TM aracılığıyla hesaplama yapma işleminde, hesaplama prosedürünü belirleyen komutlar dizisini içeren bir geçiş fonksiyonu tanımlamıştır (aşağıdaki formül [Çevik, 2021:101]'den alınmıştır):

$$\delta : Q \times \Sigma \rightarrow Q \times \Sigma \times \{\text{sol}, \text{sağ}\}$$

Yukarıdaki formülde Σ , sembollerden (alfabe) oluşan sonlu bir küme, Q , sonlu durum kümesi ve δ , TM için verilen tanımlar ve komutların gerçekleşmesini sağlayacak geçiş fonksiyonu olarak adlandırılabilir. Algoritmanın kendisi olan geçiş fonksiyonunun prosedürlerinin sağlanması durumunda, verilmiş olan problemin sonucunun sonlu bir zamanda hesaplanabiliyor olması yani teyp kafasının “dur” komutunu uygulayabiliyor olması gerekir. Öyle ise bir TM tarafından gerçekleştirilen prosedür sonucu, bir önermenin kanıtlanabilir olup olmadığına karar verebilmek için öncelikle makine herhangi bir sonlu adımda *durma durumuna* geçebilmelidir. Bu nedenle Turing, “karar problemi” için indirgemeci bir davranışla öncelikle “durma problemi” için yani bir TM'nin verilmiş olan bir girdide sonlanıp sonlanmayacağına dair karar verme problemini çözmek için hesaplama prosedürünü uygulamaya geçmiştir.

Yukarıda verdiğimiz örnekte, sonlu sayılabilir sayılar vardı ve makine daha büyük sayıların verilmesi durumunda daha uzun bir zaman harcayacak olsa da muhtemelen “dur” durumuna geçebilecektir. Aynı şeyin reel sayılar için olması pek muhtemel değildir. Birçok reel sayı sayılamaz sonsuzlukta olabilmektedir ki bu da onların hiçbir hesap makinesiyle hesaplanamaz durumda olduğunu göstermektedir. Böyle bir durumda, makine gerçekten “dur” durumuna geçebilecek midir? Casti ve DePauli, eğlenceli bir isim verdikleri (Çalışkan Kunduz) program üzerinden bu sorunun yanıtının olumsuz olduğunu göstermektedirler:

Bir n-durumlu Çalışkan Kunduz programı tarafından yazılan 1'lerin sayısını $CK(n)$ olarak tanımlayalım. Böylece Çalışkan Kunduz fonksiyonu $CK(n)$, herhangi bir durdurma programının bir n-durumlu Turing makinesinin bantı üstüne yazabileceği 1'lerin en büyük sayısıdır. (...) Ama tıpkı bir kitap hakkında kapağına (ya da başlığına) bakarak yargıda bulunamayacağımız gibi, bir fonksiyonun davranışı hakkında da bağımsız değişkenin birkaç değerinden yola çıkarak yargıda bulunamazsınız. Gerçekten de ayrıntılı bir inceleme sonucunda $(12) \geq 6 \times 4096^{4096^{4096 \dots 4096^4}}$ olduğu ortaya çıkar, burada 4096 sayısı noktalı bölgede 166 kez görünür! Böylece bir 12-durumlu Turing makinesinin Çalışkan Kunduz fonksiyonunun değerini hesaplamaya çalışır çalışmaz, etki olarak sonsuz olan çok büyük bir sayıya ulaşırız. Yeterince büyük n değerleri için $CK(n)$ değerinin aynı n sayı için değeri hesaplanan herhangi bir hesaplanabilir fonksiyonun değerini aştığı ortaya çıkar. Başka bir deyişle, Çalışkan Kunduz fonksiyonu $CK(n)$ hesaplanamazdır (Casti ve DePauli, 2004: 118-120).

Turing'in "karar problemi" için indirgeme yaparak bir çözümünün olup olmadığını sorguladığı "durma problemi" de karar verilemezdir. Aranan algoritma her TM için geçerli bir genel yöntem olmalıdır, dolayısıyla benzetim yoluyla elde edilmiş olmamalıdır. Ne yazık ki böyle bir algoritmanın bulunamayacağı, biçimsel bir mantıksal dizge ile özdeş olarak görülebilecek TM'lerle ulaşılabilecek sonuçlar üzerinden Turing'in 1936 yılında kaleme aldığı "Hesaplanabilir Sayılar" adlı makalesinde sunduğu kesin ölçütler ve kanıtlarla ortaya konmuştur. Bu nedenle Hilbert'in ünlü "karar problemi" için de kesin bir algoritma yoktur. Böylece Hilbert programı, en ağır darbeyi aldığı Gödel'in eksiklik teoremlerinin somut görüntüsünü veren Turing'in "durma problemi" arasındaki örtüşmeden dolayı tamamen çökmüş, bu nedenle de "bir kurallar kümesi izleyerek şeylerin özüne ulaşma çabamız" (Casti ve DePauli, 2004: 124) sonuçsuz kalmıştır.

2.4. Leibniz'in Evrensel Dil Hayalinin İmkani

Hilbert'in aradığı benzeri görülmemiş bu algoritma kabaca, tüm insan çıkarımsal muhakemesinin bir hesaba indirgenebilirliğini öngörebilecek güce sahipti (Davis, 2018: 127) ve bulunması durumunda, Leibniz için büyük bir başarı olduğu söylenebilirdi. Çünkü bu algoritma büyük ölçüde *calculus ratiocinator* bağlamında olacaktı.

Leibniz tam da böyle bir dilin hayalini kurmuştur yani anlamlarından soyutlanmış olan kavramların neliği ve bağıntıları üzerinden verilmiş karakteristiklerle belirlenimin yapılabildiği ve tutarlı basit aritmetikleştirilmenin imkanı ile tüm düşüncenin hesaplanabilir olduğu bir dil. Ancak öyle görünüyor ki Leibniz'in hayali, Hilbert'in programının önce Löwenheim-Skolem teoremi, sonra Gödel'in eksiklik teoremleri ve son olarak Turing'in "durma problemi" ile verdiği olumsuz sonuçlardan dolayı çökmüş olması nedeniyle, gerçekleşmeye en yakın olduğu yerde dururken yıkılmıştır.

Leibniz'in rasyonel yapay dil önerisinden sonra Frege, *Kavram-Yazısı*'nda "matematikçilerin olağan mantıksal akıl yürütmesinin gerçekten nasıl yakalanabileceğini göstermişti" (Davis, 2018: 98). Whitehead ve Russell, Frege-Peano sistemi üzerinden "gerçek matematiği yapay bir mantık dilinde geliştirmeyi" (Davis, 2018: 98) başararak bugünkü notasyonları belirlemişlerdi. Hilbert, bu tür biçimsel dillerin metamatematiksel çalışmasını yani matematiksel bir üst dil önermişti. "Ancak Gödel'den önce hiç kimse bu metamatematiksel kavramların dillerin içine nasıl yerleştirilebileceğini göstermemişti" (Davis, 2018: 98). Gödel, yaptığı sayılaştırma ve eşleştirme neticesinde, dilin semantik ve sentaktik örtüşmesini yakalamış, kendinden öncekilerin ve bir biçimde Leibniz'in rasyonel dilinin bir resmini ortaya koymuştur fakat ironik olan şudur ki böyle bir evrensel dil var ise şeylerin özüne dair her hakikati matematiğin biçimsel diliyle mekanikleştiremez (bu durum, matematiğin veya zihnin sınırlarından kaynaklanıyor olabilir). Bu nedenle denilebilir ki Leibniz bağlamında bir *evrensel dil* kendini görünür kılmasına rağmen, bu dil eksiktir yani tam değildir. Bu dilin eksikli olması da Leibniz'in hayalindeki evrensel dilin ulaşılamazlığının en açık göstergesidir.

ÜÇÜNCÜ BÖLÜM

ALGORİTMALAR, PROGRAMLAMA DİLLERİ, YAPAY ZEKÂ VE EVRENSELLİK

Tezimizin ikinci bölümünde Leibniz’in evrensel dil hayalinin gerçekleşme imkanını, bu yolda önemli gelişmelerin yaşandığı süreci konu alarak serimlemeye çalıştık. Leibniz’in evrensel dilinin ana hatlarıyla incelendiği birinci bölümde, bu dilin evrensellik ölçütünü *characteristica universalis* karşılıyordu ve hesaplama prosedürünü veren bir *calculus ratorator* ile tüm düşüncenin hesaba aktarılarak, argümanların doğruluklarının ölçülebilir olması hedefleniyordu. İkinci bölümde ise Gödel’in eksiklik teoremlerinden hareketle serimlemeye çalıştığımız üzere her doğru argüman veya önerme mekanik olarak verilemez, dolayısıyla ispatlanamaz. Bu tespitin en yalın ve somuta indirgenmiş halini Turing’in 1936 yılında yazdığı “Hesaplanabilir Sayılar” adlı ünlü makalesinde “karar roblemi” için verdiği olumsuz yanıtı dair çözümlerle anlamış bulunmaktayız⁴⁸. Bu makalede verilen çözüm için uygulanan prosedürde, bir “teorem kanıtlayıcı insan matematikçi” ile bir makine arasında kurulan benzeşim, Leibniz’in evrensel dil hayalinin ontolojik bir zemin kaymasına uğramasına neden olan makineler (bilgisayarlar) çağını başlatmıştır. Evrensellik ölçütü, biçimsel bir dili olan matematik zeminden, özellikle günümüzde biçimsel birçok dili olan ve üzerinde biçimsel dil eklememelerinin yapılabildiği makinelerin zemininde aranır ve tartışılır olmuştur.

Tezimizin ikinci bölümünde Gödel’den sonra Leibniz’in hayali bağlamında bir evrensel dilin mümkün görünmediğini serimlemeye çalışmıştık. İddia odur ki şimdilik kabaca programlama dilleri diyeceğimiz birçok biçimsel dil, Leibniz’in evrensel dil hayalini gerçekleştirmiştir. Bu bölümün ilerleyen sayfalarında, bu iddia üzerinden sorgulamamızı sürdürmeye çalışacağız. Hareket noktası olarak belirlediğimiz Turing’in 1936 yılında yazdığı makalesi ile hayatımıza giren “algoritma”, “computer”, “Turing makinesi”, herhangi bir hesaplanabilir diziyi hesaplamak için kullanılacak tek bir makine (Turing, 1937: 241) yani “evrensel Turing makinesi” gibi birçok yeni terimden başlayarak, günümüzde birer biçimsel dil olarak görülen makine dili ve programlama dilleri üzerinden Leibniz’in evrensel dil hayalinin sorgulaması içinde olacağız. Sorgulamamızda bilgisayarları, “yapay zekâ”yı ve

⁴⁸ Turing’in “karar problemi” çözümünü vermek üzere indirgeme yaptığı “durma problemi” için uyguladığı prosedür, aslında Gödel’in soyut olarak ispatladığı “eksiklik teoremlerinin”nin somut bir resmini vermektedir. Gödel ve Turing’in iki ayrı problemmiş gibi görünen sorulara verdikleri yanıtlar ve ortaya çıkan sonuçlar, aynı problem için belirlenmiş soyut ve somut çözümler kadar birbirleriyle örtüşüktür.

“yapay öğrenme”yi de kapsayacak şekilde bir tarihsel süreci gözetererek ilerlemeye gayret göstereceğiz.

3.1. Turing Makinesi ve Algoritma

Turing’in “Hesaplanabilir Sayılar” (1936) adlı makalesinde “karar problemi” için bir tanımını verdiği TM’nin yanı sıra belirlediği somut sonlu adımlar listesi, bugüne kadar problem çözümünde insan zihninin en basit düzeyde bir toplamadan en karmaşık düzeyde hesaplama yordamına kadar, her türüyle meşgul olduğu problemlerin çözümünde uyguladığı soyut düşünce etkinliğinin, başka bir deyişle, çözüm üretme prosedürlerinin açık bir tanımını bize sunmuştur. Bu tanımın ışığında, yapılan işin prosedürü *algoritma* olarak adlandırılmıştır. Turing’in verdiği somut tanım ile beraber, özellikle yirminci yüzyılda ve günümüzde çok önemli bir yere sahip olduğu gerçeğiyle birlikte *algoritma* terimi oldukça eski tarihlere dayanmaktadır. İlk algoritma, Özbek asıllı Ebu Abdullah Muhammed İbn Musa el Hârizmî (780-850) tarafından Arap rakamları kullanılarak *Hisab el-Cebir ve el-Mukabala* adlı kitabında (Mutlu, 2020: 4) sunulmuştur⁴⁹.

Hangi tür olduğuna bakılmaksızın, yemek yapmaktan matematik sorusu çözmeye değin, belirli bir problemi çözmek için tasarlanan sonlu adımlar içeren yol, bir algoritmadır. Verilen algoritmada, sürdürülecek işlemler listesinin adım adım tasarlanmış olması yani her adımının açık ve uygulanabilir olması beklenir. Günümüze programlar ve programlama dilleri açısından büyük bir önem teşkil ederek taşınmış olan *algoritma* kavramı, Hilbert’in daha önceki bölümde aktardığımız problemlerine birer çözüm arayışı içinde, 1900’lü yılların ilk dönemlerinden itibaren çeşitli matematiksel tanımlamalarla belirginlik kazanmıştır. Bu kavram üzerine yapılan tanımlamalar ise paradigma değişimlerini beraberinde getirmiştir. Bu tanımlamaları vermiş isimlerden, tezimizdeki yerleri bağlamında, ilki diyebileceğimiz Gödel ile birlikte anılabilecek diğer iki büyük isim Church ve Turing’dir.

Birbiriyle örtüşük olan Gödel’in “eksiklik teoremi” ve Turing’in “durma problemi” bir yanda dururken, öte yandan Church’ün (*lambda*) λ -kalkülüs fonksiyonu ve “Turing makinesi” arasında da “hesaplama kuramı” (İng. computability theory) açısından bir eşdeğerlik görülmektedir. “Hesaplama kuramı”, bir hesaplama modeli üzerinde kullanılan algoritma ile problemlerin çözümlenip çözülemeyeceği; çözülebilir ise bu çözümlerin verimliliğini konu edinen teorik bilgisayar biliminin bir çalışma alanıdır (Kabanda, 2019: 2) ve matematiksel

⁴⁹ Cebir üzerine yazılmış ilk kitap ünvanına sahip *Hisab el-Cebir ve el-Mukabala*, el Hârizmî’nin hesaplama prosedürlerini sistematik şekilde verdiği içeriğe sahiptir. Avrupa’ya ulaşmış ünüyle birlikte, el Hârizmî’nin isminin telaffuzunun zorluğu nedeniyle, kitabın Latince çevirisinde *Algoritmi de Numero Indorum* olarak kullanılan başlıktan sonra “algoritma” sözcüğü hayatımıza girmiştir.

mantık kökenine sahiptir. “Hesaplama kuramı”, “özdevinim kuramı (İng. automata theory) ve dil”, “hesaplama karmaşıklığı kuramı” (İng. computational complexity theory) ile birlikte olmak üzere üç temel başlık altında toplanmaktadır (Kabanda, 2019: 2). Dilbilimle yakından ilişkili olan ve bizim de tezimizin 3.3.2.1 alt başlığında kısaca konu edineceğimiz “özdevinim kuramı ve dil”, soyut matematiksel makineler ve bu makineler kullanılarak çözülebilecek hesaplama problemlerinin incelenmesi üzerine (Kabanda, 2019: 2) çalışmaların yapıldığı teorik bilgisayar biliminde, “hesaplama kuramı”nın bir dalıdır. “Hesaplama kuramı”nın diğer bir alt dalı olan ve bizim de tezimizin 3.3.2.2 alt başlığında konu edineceğimiz “karmaşıklık kuramı” ise bilgisayar bilimcilerinin, bir bilgisayarın yapabileceklerinin (veya yapamayacaklarının) sınırlarını belirlemelerine yardımcı (Kabanda, 2019: 6) olmaktadır.

Karmaşıklık kuramı, hesaplamanın merkezinde yer alır. Karmaşıklık sınıflarında bilgisayar bilimcilerinin sorunları ilişkilendirmelerine ve gruplandırmalarına yardımcı olur. Bazen bir problem çözüldüğünde, diğer problemleri çözme olasılıklarını ortaya koyabilir. Bu durum, genellikle bir sorunu çözmek için gereken zaman ve alan miktarıyla ölçülen bir sorunun karmaşıklığını belirlemeye yardımcı olur (Kabanda, 2019: 6).

“Hesaplama kuramı”na geri dönecek olursak, Church’ün λ -kalkülüs fonksiyonuna dayalı soyutlamaları, Turing’in somut sıralı adımlarıyla kıyaslandığında, çok farklı temellerden yükseliyor olmasına karşın aynı sonuçları veriyordu. Öyle olmasına karşın, Gödel, Turing tarafından verilmiş algoritmanın “karar problemi” için gerçek bir çözüm olduğunu düşünürken, Church’un soyutlamalarla dolu algoritmasını tatmin edici bulmamıştı (Say, 2019: 32). Zaman içinde “Turing’in, Church’den bağımsız biçimde makine odaklı yaklaşımının genel yinelemeliğe ve λ -tanımlanabilirliğe eşdeğer olduğu ortaya çıktığında, Church, tezinin doğruluğu için daha fazla kanıt elde etti” (Davis, 2018: 107) ve Gödel de Church’ün algoritmasının, Turing’in algoritması kadar tatmin edici olduğu konusunda fikir beyan etti (Kleene, 1986: 345). Hesaplanabilirlik kavramı için verilmiş bu iki *algoritma*, özellikle mantıkçılar tarafından eşdeğer görülse de henüz bir kanıt ulaşılamadığından ve karşıt bir örnek de sunulmadığından (Nilsson, 2019: 60) dolaylı günümüzde *Church-Turing Tezi*⁵⁰ olarak kabul görmeye devam etmektedir.

⁵⁰ Turing’in “Hesaplanabilir Sayılar: Karar Verme Probleminin bir Uygulaması” (1936) adlı makalesinin yayımlanmasından sadece birkaç ay önce, matematik ve mantık arasındaki ilişkiyi sağlamlaştırma çabasında olan ünlü mantıkçı Alonzo Church, “An Unsolvability Problem of Elementary Number Theory” [Temel Sayı Teorisinin Çözülemez Bir Problemi (1936)] makalesini yayımlamıştır. Church, makalesinde Hilbert’in “karar problemi” için verdiği çözümde, hesaplanabilen bir fonksiyonlar sınıfı belirleyerek bu fonksiyonlar sınıfını “özçağrılı fonksiyonlar” (İng. recursive functions) (1936: 354) olarak adlandırmıştır. Church, aynı makalede λ -tanımlanabilirlik fonksiyonunu (günümüzde [lambda] λ -kalkülüs olarak da bilinmektedir) tanımlayarak,

Aslında *algoritmik* kelimesiyle kastedilen şey o dönem için Hilbert'in sonlu aritmetik felsefesine uyan yöntemdir. (...) Tabii ki sonraki yıllarda bir düşünsel veya hesaplama işleminin algoritmik olma koşulunun tanımı değişmiştir. Örneğin, Gödel *genel hesaplanabilir fonksiyonları* bir hesaplama modeli olarak görmüştür. Alonzo Church (*lambda*) *λ-kalkülüs* dediğimiz, bugünkü fonksiyonel programlama dillerine benzeyen bir sistemi model olarak görmüştür. Alan Turing ise bugün en yaygın olarak kabul edilen *Turing makinesi* adını verdiğimiz bir modeli algoritma kavramını betimleyen bir matematiksel nesne olarak bize sunmuştur (Çevik, 2021: 87).

Yukarıda sözü edilen üç algoritmadan özellikle Turing makinesi modeli, ortaya atıldıktan kısa süre sonra yalnızca “hesaplama kuramı” açısından değil, ayrıca bir otomatik işlem makinesi veya döneminde anıldığı şekilde bir mantık makinesi için belirli yol haritalarının çizilmesinde etkili olmuştur. Turing makinesi, bir donanıma referans gösterilebildiği kadar, makineye işlerin yaptırılabilmesini sağlayacak programlara da referans olmuş, bunun sonrasında ise bu soyut matematiksel makineler için üretilen algoritmalar, programlama dillerinde elzem duruma gelmiştir. Bununla birlikte, Church'ün *λ-kalkülüs* ile belirlediği algoritması, günümüz fonksiyonel programlama dillerinin temeline yerleştirilmiştir. Öte yandan, Gödel'in kırk beş adımlık genel hesaplanabilir fonksiyonları da aynı şekilde günümüzde hala geçerliliği süren fonksiyonel programlama dillerine oldukça benzer bir yapı sergilemektedir.

Davis'e göre programlama dillerini bilen biri, Gödel'in makalesinde yazdığı kırk beş adımlı bir formül dizisi bulacak ve bu dizinin bir bilgisayar programına çok benzediğini söyleyecekti (2018: 97). Gerçekten de bir önceki bölümde serimlemeye çalıştığımız üzere, Gödel'in adım adım ilerleyebildiği algoritması, her adımda işlevsel davranabilen semboller dizisine sahiptir.

Ağırlıklı olarak yazılım endüstrisinde kullanılan programlama dilleri (C ve FORTRAN gibi) genellikle emir kipli olarak tanımlanır. Bunun nedeni, bu dillerde yazılan programların ardışık satırlarının bilgisayar tarafından yürütülecek komutlar olarak düşünülmesidir. C++ gibi nesne yönelimli diller de emir kiplidir. İşlevsel olarak adlandırılan programlama dillerinde (LISP gibi), bir programın satırları, işlemlerin tanımlarıdır. Bilgisayara ne yapacağını söylemek yerine, bilgisayarın ne sağlayacağını tanımlarlar. Gödel'in özel dili, işlevsel bir programlama diline çok benzer (Davis, 2018: 198-9).

Daha önceki bölümde aktarmaya çalıştığımız üzere Turing, algoritma tarafından verilen komutları işleyen bir TM (yani program) üzerinden verdiği örnek ile mekanik işlemler

Turing'den farklı olarak makinelere değil, oldukça soyut kavramlar ve içerimlere sahip ifadelere yer vermiştir. Neredeyse “sezgisel hesaplama kuramı” olarak tanımlanabilecek fonksiyonlar, LISP programlama dilinden başlamak üzere, günümüz işlevsel programlama dillerinde önemli bir yere sahiptir.

yaptırılabilir programlanabilir makinelerin mümkün olduğunu kağıt üzerinde göstermişti. Turing, çığır açıcı makalesinde “prensipte, herhangi bir olası hesaplamayı tek başına gerçekleştirebilecek tek bir 'evrensel' makinenin mümkün olduğunu” (1937: 241) ifade ederek, günümüz bilgisayarlarının ortaya çıkmasında kuramsal açıdan (ve sonrasında fiziki olarak da katkılarına sunarak) önemli bir rol oynamıştır.

Her ne kadar zorunlu olarak sonlu, fiziksel bir cihaz, sonsuz bantı ve keyfi olarak büyük verilerle başa çıkma yeteneği ile gerçek bir evrensel Turing makinesini taklit edemese de Turing'in evrensel "makinesinin" matematiksel bir soyutlaması olarak bile varlığı, evrenselliğe yararlı bir şekilde yaklaşabilecek bir makine inşa etme hedefini odak noktası (Davis, 199) haline getirmiştir.

Turing makinelerine ilişkin örnekler, programalama sanatının halihazırda örnekleridir, özellikle “evrensel” makine, *yorumlayıcı* bir programın ilk örneği (Davis, 2018: 143 italik vurgu bana ait) olarak görülmektedir. Yalnızca kuramsal açıdan değil ve ayrıca Church tarafından verilen adıyla Turing makinesi (Turing'in makale içeriğinden dolayı her ne kadar bir program olarak tanımlanması daha uygun olsa da), kuramsal olmaktan öte otomatik işlem makinelerinin elle tutulur bir donanımının mümkün olabileceğini de göstermiştir.

Aynı tarihlerde başka bir kavramsal ve kuramsal atılım Shannon tarafından gerçekleştirilmiştir. Peirce'ün Boole cebiri kullanılarak mantık kapılarının elektrik anahtar devrelerinde uygulanabileceğine dair yayımlanmayan keşfi, Shannon'un (yirminci yüzyılın gelmiş geçmiş en etkili yüksek lisans tezi olarak kabul edilen) 1936 yılında yazdığı tezinde, yeniden keşfedilmiştir⁵¹. *Röle ve Anahtarlama Devrelerinin Sembolik Analizleri* (A Symbolic Analysis of Relay and Switching Circuits) adlı tezinin 1938 yılında bir makale şeklinde yayımlanmış halinde Shannon, önermeler mantığı, Boole cebirine ait postulatlar ve De Morgan kuralları ile karmaşık anahtar devreleri hesabı arasında bir analogi kurarak, bu analoginin matematiksel teorisini oluşturmuştur.

⁵¹ Shannon ile aynı zamanlarda, Moskova Devlet Üniversitesi'nden Victor Shestakov, Boole cebirine dayanan bir elektrik anahtarları teorisi önermiştir (O'Regan, 2016: 49) fakat iki çalışma arasında ciddi bir zaman farkı bulunmaktadır. Shannon'un çalışması 1937 ve 1938 (makale) tarihlerinde yayımlanmışken, Shestakov'un çalışması Rusça dilinde 1941 yılında yayımlanabilmiştir. Görülmektedir ki zamansal öncelik, Shannon'un lehine gerçekleşmiştir. Başka kaynaklarda ise yalnızca zamansal öncelik değil ayrıca dilsel bir önceliğin olduğuna dair vurgular da mevcuttur. Şöyle ki Shannon'un çalışmasından yirmi yıl kadar önce aynı tür çalışma, Rusça ve Japonca dilinde zaten literatüre geçmiş bulunmaktadır: “Bu fikir, 1910'da Paul Ehrenfest tarafından Rus literatüründe önerilmiş ve 1934'te V. I. S. Sestakov tarafından takip edilmiştir. Aynı zamanda Akira Nakasima ve Masao Hanzawa tarafından 1936 tarihli Japon yayınında da yer almıştı. Bununla birlikte, bunlardan hiçbiri Shannon'un makalesinin çektiği kadar dikkat çekmedi çünkü makalesi İngilizce olarak yayımlandı ve izomorfizmin ayrıntılı bir hesabını, devre tasarımı teorisi üzerindeki değerini vurgulayacak şekilde sundu” (Aspray, 1990: 117).




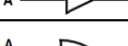


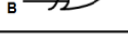
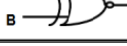
[t]artışmamızı önermeler hesabıyla sınırladığımızda, anahtarlama devreleri hesabı ile sembolik mantığın bu dalı arasında mükemmel bir analoginin var olduğu açıktır. (...) Bu benzetme nedeniyle, önermeler hesabının herhangi bir teoremi, röle devreleri açısından yorumlandığında, aynı zamanda gerçek bir teoremdir (Shannon, 1938: 474).

Shannon'un “ve”, “veya”, “değil” gibi Boole cebirine ait ifadeleri birer mantık devresi olarak karmaşık elektrik devreleri üzerinde fiziksel bir temele oturtması, mantık problemlerinin 1 (kapalı devre) ve 0 (açık devre) simgelerinin basitçe akım var ve akım yok olarak tanımlandığı fizik temel üzerinden yürütülmesini sağlamıştır. Shannon'un bu kavramsal girişimi sonucunda, ileride göreceğimiz üzere, günümüze değin dijital bilgisayarların temel mimarisinde bulunan tüm mikroişlemcilerde kullanılan mantık biriminin (devrelerinin) tasarımı gerçekleşmiştir. Dolayısıyla, “(m)antik işlemek için elektrik anahtarlarının özelliklerinin kullanılması, tüm modern elektronik dijital bilgisayarların altında yatan temel kavramdır” (O'Regan, 2016: 49). Günümüz dijital bilgisayarlarında, 1 ve 0 ikilisi kullanılmaktadır ve Boole cebirine dayalı mantıksal işlemler, birer fonksiyon olarak görülebilecek “ve”, “veya” ve “değil” (“AND”, “OR” ve “NOT”) elektronik kapıları tarafından gerçekleştirilebilmektedir (aşağıdaki gösterim [Aspray, 1990: 116]'dan alınmıştır).

MANTIK	DEVRE
doğru	kapalı
yanlış	açık
ve	seri
veya (kapsayıcı)	paralel

Bu fonksiyonlarla, basit devrelerden daha karmaşık devrelere doğru (aritmetik gibi) çeşitli tasarımlar geliştirilebilmektedir. Aşağıdaki tabloda Boole cebiri, önermeler mantığı ve devreler arasında kurulmuş analogide, birer fonksiyon olarak görülebilecek “ve”, “veya” “değil” ile bu fonksiyonların değillemeleri ve diğer genişletilmiş halleri daha açık görülebilmektedir. Bu fonksiyonlarla işletilen notasyonların, mantıksal açıdan neyi ifade ettikleri ve ne tür bir işleme konu oldukları sembolik mantık dilinde sunulmuş olup, önermeler mantığında olduğu gibi birer doğruluk tablosu üzerinde 1 ve 0 simgeleri ile temsilleri verilmiştir. Birer anahtar devresi veya herhangi bir devre üzerinde temsil ettikleri fiziksel görünüm de ayrıca sunulmuştur.

Tablo 3.1 Mantık kapıları örneği.

	SEMBOL	İŞLEM	Doğruluk tablosu															
AND VE		$X = A \cdot B$ or $X = AB$	<table border="1"> <thead> <tr><th>A</th><th>B</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	X	0	0	0	0	1	0	1	0	0	1	1	1
A	B	X																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OR VEYA		$X = A + B$	<table border="1"> <thead> <tr><th>A</th><th>B</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	X	0	0	0	0	1	1	1	0	1	1	1	1
A	B	X																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
NOT		$X = A'$	<table border="1"> <thead> <tr><th>A</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </tbody> </table>	A	X	0	1	1	0									
A	X																	
0	1																	
1	0																	
Buffer		$X = A$	<table border="1"> <thead> <tr><th>A</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td></tr> </tbody> </table>	A	X	0	0	1	1									
A	X																	
0	0																	
1	1																	
NAND		$X = (AB)'$	<table border="1"> <thead> <tr><th>A</th><th>B</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	A	B	X	0	0	1	0	1	1	1	0	1	1	1	0
A	B	X																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
NOR		$X = (A + B)'$	<table border="1"> <thead> <tr><th>A</th><th>B</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	A	B	X	0	0	1	0	1	0	1	0	0	1	1	0
A	B	X																
0	0	1																
0	1	0																
1	0	0																
1	1	0																
XOR Exclusive OR		$X = A \oplus B$ or $X = A'B + AB'$	<table border="1"> <thead> <tr><th>A</th><th>B</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	A	B	X	0	0	0	0	1	1	1	0	1	1	1	0
A	B	X																
0	0	0																
0	1	1																
1	0	1																
1	1	0																
XNOR Exclusive NOR or Equivalence		$X = (A \oplus B)'$ or $X = A'B + AB$	<table border="1"> <thead> <tr><th>A</th><th>B</th><th>X</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	X	0	0	1	0	1	0	1	0	0	1	1	1
A	B	X																
0	0	1																
0	1	0																
1	0	0																
1	1	1																

Kaynak: Yanmaz, 2018: 1

Turing ve Shannon'un, bir otomatik işlem makinesine referans olan kavramsal girişimleri ve bu dönemde yaşanan teknolojik gelişmeler, işlemci makinelerin (günümüz deyiimiyle bir bilgisayarın), donanımı ve Turing makinesinin gösterdiği gibi bir problem için iyi belirlenmiş algoritmayı işleyen bir programı yani yazılımı olabileceğine dair inancı güçlendirmiştir. İkinci Dünya Savaşının yakın dönemlere denk gelmiş olması, “hesaplama kuramı” üzerine olan çalışmaların ivmesini, bilgisayar bilimleri yönünde hızlandırarak değiştirmiştir. Bu durum, farklı ülkelerin bilim insanlarınca ciddi çalışmaların başlatılmasına neden olmuştur. Farklı kıtaların farklı laboratuvarlarında, donanıma ait çeşitli mantık mimarileri ortaya çıkmış, böylece 1940'lı yıllardan başlayarak günümüze değin neredeyse hiç hız kesmeyen bir teknolojik atılım serüveni başlamıştır.

3.2. Bilgisayarların, Yapay Zekânın, Biçimsel Dillerin Doğuşu ve Gelişim Süreçleri

Turing makinesi fikrinin oluşmasının temelinde yatan nedenleri, yaşanan süreci de gözeterek tezimizin önceki bölümlerinde konu edinmiştik. Turing'in “Hesaplanabilir Sayılar” (1936) adlı makalesinin, motivasyonu ve özü tamamen matematiksel olsa da bilgisayar biliminin doğuşunu temsil ettiğini (Eberbach vd., 2004: 162) vurgulamıştık. Bu bölümde ise Turing makinesi ve Shannon'un mantık kapıları fikrinin verdiği kavramsal gelişmenin yarattığı etki ile birlikte, bilgisayarlar için teknolojik alt yapının da gelişim serüvenine kronolojik olarak değinmekte yarar görmekteyiz.

Bir mantık makinesi ya da bir işlemci makinenin hayata geçirilmesi fikri, genellikle tarihsel açıdan oldukça eski dönemlere, M.Ö. 500'lü yıllarda kullanılmaya başlanmış basit bir

hesaplama aygıtı olan abaküse dayandırılmaktadır. Modern çağlara gelindiğinde ise “ilk makineler, aritmetik işlemleri yapabiliyor ama programlanamıyordu” (Nilsson, 2019: 57) ve bunlardan ilki sayılabilecek makine, Wilhelm Schickard (1592-1635) tarafından 1623 yılında gökbilim cetvellerindeki hesaplamalarda kullanılmak üzere (Aspray, 1990: 35) icat edilmiştir. Pascal’ın 1642 yılında vergi tahsildarı babasının hesaplamalarını kolaylaştırmak üzere icat ettiği toplama ve çıkarma üzerine basit aritmetik işlemler yapabilen hesap makinesi, “kendisine ait yaklaşık elli hesap makinesinin ilkidir” (Nilsson, 2019: 57). Leibniz’in 1673 yılında Kraliyet Cemiyeti üyeliğine seçilmesini sağlayan hesap makinesi toplama, çıkarma, çarpma ve bölme gibi basit aritmetik işlemlerin yanı sıra karekök alma gibi fonksiyonları da yerine getirebilmekteydi ve Leibniz’in sonradan makinenin içine yerleştirdiği bir aygıttan dolayı “Leibniz Çarkı” olarak da bilinen bu hesap makinesi, yirminci yüzyıl başlarına kadar kullanılmaya devam etmiştir (Davis, 2018: 5).

Mekanik hesaplayıcılar dışında tarihin ilk mantık kanıtlayıcı makinesi, Charles Stanhope (1753-1816) tarafından 1805 yılında icat edilmiştir. Döneminde kullanılan adıyla “Stanhope kutusu” olasılık ve basit tasım mantığına dayalı problemleri çözmek üzere geliştirilmiş bir tür “analog bilgisayar” (Nilsson, 2019: 30) olarak kabul görmüştür. Bu mantık makinesi, bir *kanıtlayıcı* olarak özel bir yere sahipti çünkü başkalarına, “mantık problemlerinin mekanik yollarla çözülebileceğini” (Aspray, 1990: 108) göstermiştir. “Stanhope kutusu” geleneksel tasıma dair “hiçbir M, A değildir; tüm B’ler M’dir, dolayısıyla hiçbir B, A değildir” (Nilsson, 2019: 31) mantıksal çıkarımını doğru sonuçlandırabildiği gibi ayrıca olasılığa dair “on adet A’dan sekizi B; on A’dan 4’ü C’dir; dolayısıyla en az iki B, C’dir” (Nilsson, 2019: 30) tarzındaki sorulara da yanıt verebiliyordu. “Lull’un şemalarının ve Leibniz’in umutlarının aksine, Stanhope gerçekten çalışan ilk mantık makinesini inşa etmişti. (...) Belki de onun eseri, mantıksal usamlamanın gerçekten mekanikleştirilebileceği inancını güçlendirmişti” (Nilsson, 2019: 31).

Mühendislikten gelen teknolojik geçmişten seçtiğimiz Joseph-Marie Jacquard’ın (1752-1834), 1804 yılında dokuma tezgahları için hayata geçirdiği “delikli kart” kullanma fikri ise Jacquard’ın bilgisayar bilimlerinde adının sıklıkla anılmasını sağlamıştır. “Tarihin ilk otomatik makinelerinden biri” (Nilsson, 2019: 49) olarak bilinen Jacquard dokuma tezgahlarında kullanılan delikli kartlar aracılığıyla, mekiklerin hareketleri denetlenerek kumaşlar için belirlenmiş tasarımların doğru okunması ve otomatik bir biçimde kumaş üretim evresinin gerçekleşmesi sağlanıyordu. Bu teknik, delikli kartlarla programlama yapmanın somut bir örneği olarak kabul edilmektedir.

Aynı teknik⁵², farklı bir uslamlayla Babbage tarafından kullanılmıştır. İkinci bölümde kısaca değindiğimiz İngiliz matematikçi Babbage'ın 1834-1837 yılları arasında tasarımı üzerine çalıştığı “analitik motor” adıyla da bilinen “çözümlemeci makine” fikrinin, günün teknolojisi uygun olmadığından dolayı hayata geçirilemediğini vurgulamıştık. Asıl önemli olan ise Babbage'ın tasarımının günümüz teknolojisi ile değerlendirildiğinde, istenen hesabı yapmaya programlanabilen bir “evrensel bilgisayar” tasarımı olduğudur.

Babbage, “analitik motor” tasarımı ile programlanabilir bir evrensel bilgisayar fikrini çağrıştırmakla kalmaz, ayrıca tarihin ilk programcısı ünvanını alan De Morgan'ın matematik öğrencisi Lovelace Kontesi Augusta Ada King'in (1815-1852) ismi, genellikle Babbage ile birlikte anılır. Ada Lovelace'ın, Babbage'ın tasarımı üzerine uzun notlar düştüğü bilinmektedir, olabilir ki bu nedenle tarihin ilk programcısı⁵³ olduğuna dair genel bir kanı gelişmiş olsa da bu yönde açık bir ispat yoktur. Lovelace'ın aldığı notlardan biri “Jacquard dokuma tezgahının, çiçekleri ve yaprakları örmesi gibi çözümlemeci makinenin de cebirsel desenler ördüğünü söyleyebiliriz” (Lovelace'dan akt. Goldstine, 1993: 22) şeklindedir. Kendi hatıratında kaleme aldığı “çözümlemeci makinenin, herhangi bir şey icat ediyormuş iddiası yok. *Nasıl yapılacağını bildiğimiz her şeyi, buyruk verdiğimizde yapabilir*” (Lovelace'dan akt. Nilsson, 2019: 68 italik vurgu Lovelace'a ait) sözlerinden de anlaşıldığı üzere bu makinenin programlanabilir ve hayata geçirilebilir bir makine olduğunu öngördüğü açıktır.

Bir bilgisayarın, yalnızca herhangi bir özel nedene hizmet etmek için icat edildiğini, yukarıda oldukça kısa bir özetini vermeye çalıştığımız teknolojik, ticari (Jacquard dokuma tezgahı kısa sürede on bin adetten fazla satmıştır) ve daha önceki bölümlerde geniş yer ayırma gayretinde olduğumuz mantıksal, matematiksel ve felsefi (tezimizin başlangıç noktasının Leibniz olduğunu hesaba katarak) nedenlerden ötürü düşünmek imkansızdır. Doğru bir betimlemesini vermek gerekirse, bilgisayar, kendisinden önce gelen ve kendisiyle birlikte gelişen her şeyin kümülatif bir göstergesidir. Tezimiz bağlamında bu bölümde, bunlar arasında sayılabilecek iki unsur, yaşamdan formların belirli bir taklidi görünümünü veren

⁵² Babbage'dan sonra delikli kart kullanma fikri, “1890 Amerikan nüfus sayımı verilerini sıralamak amacıyla Herman Hollerith tarafından benimsenecek, ayrıca otomatik piyanolarda” (Nilsson, 2019: 49) kullanılacaktır. Piyanolarda, delikli kart yerine delikli rulolar vardı. “Kavra-ve-yerleştir tarzıyla çalışan ilk fabrika robotları, bu fikrin ancak çok az geliştirilmiş bir halinden” (Nilsson, 2019: 49) faydalanmıştır.

⁵³ Tarihin ilk programcısı ünvanı ile anılan ve bilgisayar bilimlerinde bu anlamda özel bir yer verilen Ada Lovelace'ın değil, aslında Babbage'ın bu ünvanla anılması gerektiğine dair şu alıntıyı paylaşmakta yarar var: “Romantik cazibesi olan bu imge temelsizdir. Lovelace'ın notlarında (Babbage'ın İtalya'da, Torino şehrinde vermiş olduğu seminerin açıklaması için yaptığı çeviride) atıfta bulunulan programların biri hariç hepsi, üç ila yedi sene önce Babbage tarafından hazırlanmıştı. O tek istisnaysa, Babbage tarafından Lovelace için hazırlanmış, gerçi Lovelace bu program içinde 'hata' tespit etmiştir. Ada Lovelace'ın, Çözümlemeci Makine için program hazırladığını gösteren hiçbir bulgu olmadığı gibi, Babbage'la mektuplaşmaları bunu yapacak bilgisinin olmadığını gösterir” (Bromley, 1990: 88-9).

yapay zekâ sistemleri ve bir donanıma ne yaptıracağımızı aktarabilmek için yarattığımız biçimsel dillerdir. Açık bir ayrıştırmasını yapamadığımız bilgisayar, yapay zekâ ve ihtiyaca binaen geliştirilmiş (yapay) biçimsel diller, bu bölümde ayrı birer başlık altında alınmaktan ziyade nesiller bazında gelişim süreçlerinin takibinin daha kolay yapılacağı düşüncesiyle, alt başlıklarda birlikte aktarılacaktır.

3.2.1. Birinci Nesil Bilgisayarlar ve Birinci Nesil Programlama Dilleri

Turing'in bir bilgisayar, algoritma ve program için kuramsal alt yapıyı sağladığı makalesinin yayımlandığı 1936 yılı ve sonrasında, farklı ülkelerde farklı mantık mimarisine sahip makineler üzerinde çalışmalar başlamış görünmektedir. “Bir bilgisayar programını bilgisayarın belleğinde verilerle birlikte saklama fikrini ilk kimin düşündüğünü belirlemek zor” (Nilsson, 2019: 62). Bu nedenle Turing, makalesinde depolama işini bant üzerinde yapmayı önermiş (Nilsson, 2019: 63) olmasına karşın, genellikle lüteratürde Turing dışında başka birçok isim karşımıza çıkmaktadır. Turing ile oldukça yakın tarihlerde Berlin’de Alman mühendis ve mucit Konrad Zuse (1910-1995), New York’da Bell laboratuvarlarında George Stibitz (1904-1995), Harvard Üniversitesinde Howard H. Aiken (1900-1973), Presper Eckert (1919-1995), John W. Mauchly (1907-1980) ve adlarını sayamadığımız niceleri, rölelere dayanan hesaplama sistemleri ile programlanabilir bilgisayarlar üzerinde çalışmışlardır.

Zuse, 1936 yılında ikili sayı sistemi kullanılarak programlanabilir bilgisayar olan Z1’in ardından Z2 ve son olarak 1943 yılında tanıtımını yaptığı, “dünyanın ilk serbestçe programlanabilen tam işlevsel bilgisayarı olan Z3’ü” (Nilsson, 2019: 63) icat etmiştir. İkinci Dünya Savaşı sırasında Z3, hava saldırısında imha edilmiştir. Zuse’nin, 1946 yılında *plankalkül*⁵⁴ adında ilk programlama dilini tasarladığı da (O’Regan, 2016: 192; Nilson, 2019: 63) ayrıca not edilmesi gereken bir bilgidir.

Savaşların, bilim ve teknoloji üzerine yapılan çalışmalarda ivme kazandırıcı olduğu tarihsel olarak açıktır. İkinci Dünya Savaşının da bilgisayar bilimleri üzerinde hızlandırıcı etkisi vardı. Almanya tarafından gizli mesajların şifrelenmesi ve yeniden çözümlenmesi için kullanılan *Enigma* adlı makinenin çalışma prensibinin çözülmesi için Turing ve arkadaşları,

⁵⁴ Konrad Zuse, aslında yalnızca ilk programlama dilini değil, en eski yüksek seviye programlama dilini yaramıştır. “En eski yüksek seviye programlama dili (...) Plankalkül, 'Plan' ve 'Kalkül' başka deyişle bir 'program hesabı' anlamına gelir. 1946'da geliştirilen bu dil, nispeten modern bir dildir. O zamanlar dil için bir derleyici (compiler) yoktu ve ancak elli yıl sonra bu dil için bir derleyici geliştirildi” (O’Regan, 2016: 192). Bundan sonra anlaşılmalıdır ki *Plankalkül* bir satranç oyunu programı için yazılmış, erken dönem ilk yüksek seviyeli programlama dilidir (Nilsson, 2019: 127). Yüksek seviye bir programlama dili olmasının dışında, *plankalkül* şu özellikleri taşımaktaydı: “Dilin ana yapıları, değişken atama, aritmetik ve mantıksal işlemler, korumalı komutlar ve *while* döngüleridir. Bazı liste ve küme işleme işlevleri de vardır” (O’Regan, 2016: 192). *While döngüleri* (while loops), bilgisayar bilimlerinde, bir talimatın veya talimat grubunun yalnızca belirli bir koşul karşılanmaya devam ederken yürütüldüğü bir bilgisayar kodu bölümü, anlamında kullanılır.

büyük çaba sarfetmişlerdi. Bir tür kod çözücü olarak geliştirdikleri makine ile savaşın akıbeti, daha az sivilin ölmesi ve savaşın süresinin kısalması olarak değerlendirilebilecek şekilde olumlu yönde değişmişti. Bu savaş sırasında, makineler üzerinde yapılan çalışmalar, İngiltere’de ikili sistemle çalışan Colossus I gibi yeni dijital bilgisayarların daha gelişkin hallerinin ortaya çıkmasını sağlamıştı.

Savaşın devam ettiği 1944 yılında Amerikan ordusu mensubu Aiken, Babbage’ın “analitik motor” veya diğer adıyla “çözümleyici makine” tasarımına tamamen bağlı kalarak, Babbage’ın hayalini gerçekleştirmişti fakat kullandığı onlu sayı tabanı, hesaplamaları güçleştirmekteydi. Harvard Mark 1 adındaki bu makine, Zuse’nin Z3’ünün gerisinde kalmış ve genel hesaplayıcı olarak tanımlanamamıştı.

Elektronik hesaplama yapabilen ilk genel amaçlı bilgisayarlar, askeri amaçlar doğrultusunda geliştirilen makineler olarak karşımıza çıkar. İlk elektronik bilgisayar olarak bilinen ENIAC⁵⁵, Amerika’da Mauchly ve Eckert tarafından 1943-1946 yılları arasında geliştirilmişti. Tonlarca ağırlıkta, geniş odalara sığmayan ebatlarda olan ENIAC, “farklı sorunları çözmek için fiziksel olarak yeniden bağlanması gereken sabit programlı bir bilgisayardı” (O’Regan, 2016: 56) ve bir mantıksal karmaşa içermekteydi. EDVAC, ENIAC’ın karmaşık mantıksal problemlerinin giderilebilmesi için 1945 yılında von Neumann tarafından taslağı, başka bir deyişle mantık mimarisi oluşturularak, depolanmış program kavramının gündeme gelmesini sağlayan ilk makine oldu.

Günümüze kadar ulaşılmış bu bilgisayar türlerinin donanımı için “von Neumann mimarisi” adı kullanılmaktadır. “İdeal von Neumann mimarisi, (göreve özgü) saklanmış programı, herhangi bir programın komutlarını (sıralı olarak) icra eden (çok amaçlı) donanım devrelerinden ayırır⁵⁶” (Nilsson, 2019: 63). Hilbert projesiyle yakından ilgili olan mantıkçı-matematikçi von Neumann, Gödel’in eksiklik teoremlerinden sonra mantık ile olan bağı, Turing aracılığıyla somut mimari içeren makineler üzerinden sürdürmüştür. Bu değişimin

⁵⁵ Bilgisayarların ilk dönemlerinden önce, Turing’in “Hesaplanabilir Sayılar” (1936) adlı makalesinden hatırlayacağımız üzere “computer” sözcüğü insan hesaplayıcılar için, “calculator” sözcüğü ise bir nevi hesap makinesi anlamında kullanılmaktaydı. Bu gelenek, bilgisayarların başlangıç döneminde de korunmuştur. “İnsan hesaplayıcılar genelde mekanik hesap makineleri kullanan ve hemen hemen hepsi kadın olan kullanıcıları. Örneğin, diferansiyel denklemlerin nümerik çözümleri için gereken ardışık işlemler insan hesaplayıcılara uygun biçimde dağıtılıyor ve bir tür seri üretim mantığı uygulanarak binlerce aritmetik işlemden oluşan çözümler aynı anda çalışan onlarca hesaplayıcı tarafından elde ediliyordu. Bu gelenek bilgisayar döneminde de devam etmişti ve ENIAC’ın ilk altı programcısının tümü de kadındı. Bilgisayarların yaygınlaşmasıyla insan hesaplayıcılara gerek kalmamış, ‘computer’ sözcüğü de zamanla bilgisayarlar için kullanılmaya başlanmıştır” (Mutlu, 2020: 10).

⁵⁶ Günümüzde donanım da bu tip bir ayrımı yapmak oldukça zorlaşmıştır. Şöyle ki programın yazılım olarak kabul edildiği ve bu yolla bilgisayar donanımından ayrıldığı mimarilerden daha esnek dolayısıyla daha bulanık mimariler ortaya çıkmıştır. Buna bir örnek vermek gerekirse, günümüzde “çoğu bilgisayarda programlardan bazıları doğrudan devrelerin içine (Nilsson, 2019: 63) yerleştirilmektedir.

olumlu yanı, mühendislik harikası olan bilgisayarların altında yatan gerçek başarının, mantık temelli olduğunun hatırlatıcı etkisidir.

Bir mühendislik güç gösterisi olmasına rağmen, ENIAC mantık karmaşasıydı. von Neumann'ın bir mantıkçı olarak uzmanlığı ve İngiliz mantıkçı Alan Turing'den öğrendikleri, onun bir hesaplama makinesinin bir mantık makinesi olduğu temel gerçeğini anlamasını sağladı. EDVAC'ın devrelerinde, yüzyıllar boyunca geliştirilen olağanüstü bir mantıkçılar koleksiyonunun damıtılmış içgörülere somutlaştırılmıştır. Bilgisayar teknolojisinin nefes kesici bir hızla ilerlediği günümüzde, mühendislerin gerçekten dikkate değer başarılarına hayran kaldığımızda, fikirleriyle tüm bunları mümkün kılan mantıkçıları gözden kaçırmak çok kolaydır (Davis, 2018: xv).

Turing aynı yıllarda, 1945'lerin sonunda ACE olarak adlandırdığı mantık makinesini geliştirdi. Aynı probleme farklı yaklaşımlar sunan iki matematikçi olarak von Neumann'ın EDVAC'ı ve Turing'in ACE'i, yapmaları öngörülen işler bağlamında ayrışık görünüyorlardı. EDVAC'ın mantıksal mimarisi, sayısal hesaplamalar üzerine işlemlerin hızlandırılmasına odaklıyken (Davis, 2018: 162), ACE minimalist bir yaklaşımla, Turing'in "Hesaplanabilir Sayılar" (1936) adlı makalesinde yer eden "Turing makinelerine daha yakın bir şekilde" (Davis, 2018: 162) tasarlanıp organize edilmişti. EDSAC, 1949 yılında von Neumann mimarisi ile donatılmış haliyle, kendisinden öncekilerin gelişmiş versiyonu olarak ortaya çıkmıştı. Bu yıllarda ortaya çıkan bilgisayarlar, ancak birkaç kod sığdırılabilecek kadar küçük belleklere sahip olsa da matematik işlemleri, normalinden çok daha hızlı yapma potansiyeline sahipti. Ciddi eksikleri de yok değildi, örneğin bir girdinin sunulabileceği herhangi bir klavye benzeri donanım henüz geliştirilmemişti. Dönemin gelişmiş bilgisayarlarından biri olan EDSAC'ın görüntüsü aşağıda verildiği şekildedir.



Resim 3.1 Cambridge Üniversitesinde bulunan EDSAC adlı bilgisayar

Kaynak: Nilsson, 2019: 60

Bilgisayar bilimleri ve günümüz bilgisayar teknolojisi tarihi açısından önemli yerleri olan bazı örnekleri yukarıda sıralamaya çalıştık. Birinci nesil bilgisayar örneklerinin 1946’lardan itibaren ortaya çıktığı bu dönemde, bilgisayarların temelinde yer eden ikili sayı sistemi, bilgisayarın ya da makinenin dili olarak kabul edilmiştir. Bilgisayar bilimlerinde “makine dili” dendiğinde, bilgisayarın komutları anladığı ve uyguladığı kendi dili anlaşılmaktadır. Bu dil, makinenin altında yatan mimariye göre genellikle sekiz adet sıralı 0 ve 1’lerden oluşan komutları anlamaktadır. Dolayısıyla “makine dili” bilgisayarın donanımına bağımlı durumdadır.

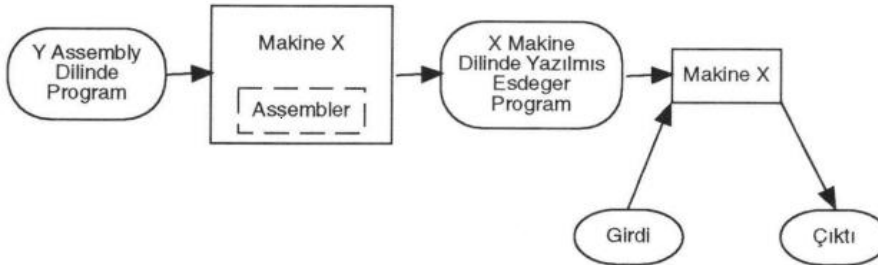
Bilgisayarlarda olduğu gibi programlama dillerinde de nesillendirme yapılmıştır. “Birinci nesil programlama dili (veya 1NPd), 0 ve 1’lerden oluşan makine düzeyinde bir programlama dilidir” (O’Regan, 2016: 190). Birinci nesil bilgisayarlar için üretilmiş kodları veren bir makine dili, bu bilgisayarların mimarilerindeki farklılıklardan dolayı genellikle başka bir bilgisayarın makine dilinden önemli ölçüde ayrışıklık sergileyebilmekteydi (O’Regan, 2016: 190), bu nedenle makine kodları taşınabilir durumda değildi. Dolayısıyla program, hemen her farklı bilgisayar için yeniden yazılıyordu (O’Regan, 2016: 190).

Bir makinenin doğal dili, 0 ve 1’lerden oluşmuş komut dizileridir. Bilgisayar ne kadar gelişmiş olursa olsun, anladığı dil her zaman makine dili olacaktır. Kaldı ki bilgisayar donanımı, kendisi için yazılmış komutlar (yazılım) olmadan ne yapacağını bilemez. Makine dili, doğrudan bilgisayarın donanımına yapacağı işi buyurduğundan dolayı, işlemler oldukça hızlıdır fakat makinenin anladığı dilde, 0 ve 1’lerden oluşan komutlar dizisi programlamak insan için o kadar kolay bir iş değildir. Bu sorunu gidermek adına *Assembly* (çeviri) dili gelişmiştir. Düşük seviye programlama dillerinden olan *Assembly*, ikinci nesil bir programlama dilidir (2NPd.) ve makine dilleri ile insan kullanıcılar arasında bir köprü görevi görmüştür.

3.2.2. İkinci Nesil Programlama Dilleri, Bilgisayarlar ve Yapay Zekâ

Bilgisayarların çalışma mantığının daha anlaşılır hale geldiği 1950’li yıllar, sadece bilgisayar donanımı açısından değil, ayrıca yazılım alanında da insanların daha kolay yazabildikleri kodların, bilgisayarlara iş yaptırmak için kullanıldığı yıllardı. Düşük seviyeli ikinci nesil programlama dillerinden *Assembly*, makine dili talimatlarını temsil eden kodlar yazılmasını sağlıyor ve sonra bir *assembler* (çevirici) tarafından bu kodlar, makine diline (O’Regan, 2016: 190) yani ikili sisteme çevriliyordu. Bu dille yazılmış komut içeren kodlar, genellikle basit düzeydeydiler ve bu kodlardan oluşmuş program, *mnemonic* (sembol) serisinden oluşmaktaydı. Böylece, makine dilinin nümerik sistemine 1947’den itibaren

Londra laboratuvarlarında çalışmalarına başlanmış bu yeni dilin (Assembly) sağladığı alfabetik bir sembolleştirme ile ulaşılabilirdi. *Assembly* diliyle yazılmış bir kod, *assembler* (*assembly*-derleyicisi) tarafından çevrildiğinde belleğe yüklenebiliyordu. Aşağıdaki şekilde, bir *assembler* ara biriminin çalışma prensibi görülmektedir.



Şekil 3.1 *Assembler* arabirimi ve çalıştırılma süreci

Kaynak: Eldeniz, 1994: 141

Aşağıda verilmiş örnekte, ekrana “Hello World” (Merhaba Dünya) yazdıran bir makine dili gösterimi ile 0-1’ler yerine aynı kodun 16’lık tabanda (hexadecimal şekilde) gösteriminin, *Assembly* dili ile bir karşılaştırması mevcuttur (aşağıdaki örnek, [Şamlı, 2020: 54]’den alınmıştır):

```

1011101100010001 0000000110111001 0000110100000000 1011010000001110
1000101000000111 0100001111001101 0001000011100010 1111100111001101
0010000001001000 0110010101101100 0110110001101111 0010110000100000
0101011101101111 0111001001101100 0110010000100001
  
```

```

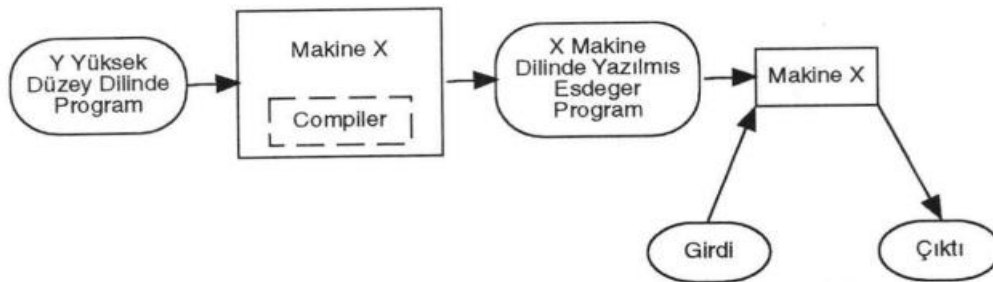
BB 11 01 B9 0D 00 B4 0E 8A 07 43 CD 10 E2 F9 CD 20 48 65 6C 6C 6F 2C 20 57 6F 72 6C 64 21
  
```

Assembly dillerinin, birer biçimsel dil olarak makine dillerine göre kodlamada esneklik ve verimlilik açısından büyük avantajları olmasına karşın, bu laboratuvar dilleri de işlemciye özgü dillerdi yani donanım-bağımsız değildi. *Assembly* dili, işlemcinin ana dilinde olduğu için önemli hız avantajlarına sahipti fakat *Assembly* dilinde yazılmış bir programın farklı bir platform için yeniden yazılması zorunluluğu (O'Regan, 2016: 190), programlama yapma sürecinde büyük çabalar gerektiriyordu ve bu anlamda ciddi bir zaman kaybı söz konusuydu. İnsan ve makine arasında, insanın anlama düzeyine daha yakın bir dilsel yapıya ihtiyaç duyulmaktaydı.

Bilgisayar programcılığı tarihinde önemli bir yere sahip Grace Murray Hopper (1906-1992), ilk büyük ölçekli ticari bilgisayar ünvanını elinde tutan UNIVAC’ın yapımında 1949 yılından itibaren yer alırken, burada tarihin ilk *derleyici* programını (İng. compiler) yazdı.

Bundan iki yıl sonra, 1951’de Hopper, “ilk *derleyici* olan A-0’ı ve onun ana devamı olan A-2’yi ve orijinalinde AT-3 olarak adlandırılan erken matematik dillerinden birini yaratan bir bölümü denetledi ve sonunda MATH-MATIC olarak yeniden adlandırdı (Sammet, 1981b: 6). Bu derleyicinin gelişmiş versiyonları, daha sonra sırasıyla ARITH-MATIC ve 1955 yılında FLOW-MATIC adlarını aldı. FLOW-MATIC, ileride yeniden değineceğimiz ilk modern programlama dillerinden biri olan geç dönem üçüncü nesil yüksek seviye programlama dili COBOL’un da atasıdır.

Bir *derleyici* (compiler), özü itibariyle bir tercüman gibi davranır yani iki program arasında kaynak kod dönüştürmesi yapar. Bir *derleyicinin* yaptığı iş, iki farklı şekilde olabilir: Aynı seviye iki programlama dili arasında dönüştürme işlemi veya seviye itibariyle daha yüksek bir programlama dilini daha düşük seviye programlama diline dönüştürme işlemi. Kaynak kod dönüştürmesinin yapıldığı düşük seviye bir dil, orta seviye veya *Assembly* olacağı gibi *makine dili* de olabilir. Bu nedenle *derleyiciler* “özel bir ara program olarak” (Kayabaş, 2020: 203) da görülebilir. Aşağıdaki şekilde, bir *derleyici* (İng. compiler) ara biriminin çalışma prensibi görülmektedir.

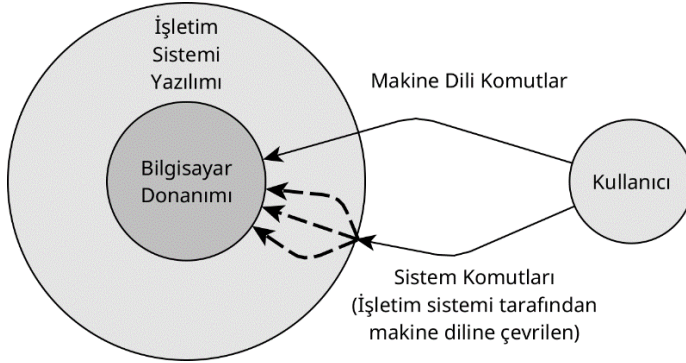


Şekil 3.2 *Compiler* (derleyici) arabirimi ve çalıştırılma süreci

Kaynak: Eldeniz, 1994: 142

İnsan kullanımına daha yakın programlama dilleri üzerine çeşitli laboratuvarlarda çalışmalar yapılırken, bilgisayar teknolojisi de “bellek” ve “çevre” gibi saklama alanına ve donanım güçlendirici yardımcı öğelere odaklanmıştı. Önemli bir gelişme donanım alanında yaşanmıştı. “Transistör, daha önceleri kolayca arızalanabilen ve çok ısı yayan vakum tüpleri ile gerçekleştirilen elektronik işlemlerin, bir nohut tanesi büyüklüğündeki katı bir malzeme ile gerçekleştirilmesini” (Mutlu, 2020: 11) sağlamıştı. Böylece yarı iletken teknolojisi ile birlikte, donanımda kullanılan parçaların hacimleri küçülmeye, etkileri büyümeye başlamıştı. İlk defa 1951 yılında MIT’de (Massachusetts Institute of Technology) bir bilgisayar için çevre birimi olan klavye ve bunun yanı sıra katotlu tüp kullanılmış (Mutlu, 2020: 12), klavye sayesinde ilk

kez insan-makine etkileşimi gerçekleştirilmişti. Böylece bilgisayar, daha kullanıcı dostu bir görünüme kavuşmuştu. Aşağıdaki şekilde, bir bilgisayar sisteminin katmanları ve bir kullanıcı ile olan ilişkisi görülmektedir.



Şekil 3.3 Bir bilgisayar sisteminin katmanları

Kaynak: Eldeniz, 1994: 141

Bilgisayar bilimlerinde özellikle 1951 yılı, yukarıda sıralamaya çalıştığımız nedenlerden ötürü önemli bir yere sahiptir. İlk *derleyici* program da Hopper tarafından yine 1951’de yaratılmıştı. Makineden bağımsız programlama dilleri teorisini geliştiren ilk kişi olan Hopper, programlama dillerinde doğal dil (asıl istediği dil İngilizceydi) kullanılması gerektiğini savunuyordu. Hopper’ın, konuşmasında dile getirdiği üzere, çalışmalarının temelinde yatan fikir de buydu:

Ancak bu derleyiciyi oluştururken, özellikle uzun süre hayatta kalmasını sağlamak için oldukça ilginç bulduğum bir şey yaptık. Kelimeyi çabucak almak için -o sırada algoritmaları ayrıştırma hakkında hiçbir şey bilmiyorduk- yapılan şey şuydu; fiili aldınız ve sonra bu tür cümleyi ayrıştıran bir alt programa atladınız. Bunu hızlı bir şekilde yapmak ve aynı zamanda bu sıçramayı yapmayı kolaylaştırmak için FLOW-MATIC'teki tüm fiillerin birinci ve üçüncü harfleri tekti (benzersizdi). (...) Dolayısıyla, iki şey FLOW-MATIC'i matematiğin ve programlama dillerinin ötesinde etkiledi. Birincisi, programları herhangi birinin dilinde yazmak istedik, bu da tüm ifadelerin buyruklu (emir kipli) ifadeler olması gerektiği anlamına geliyordu çünkü Almanca'da fiille başlayan tek cümle türü buydu. İkincisi ise fiilleri seçmemiz gerekti, böylece birinci ve üçüncü karakterler tek (benzersiz) olacaktı (Hopper, 1981: 18 parantez içinde verilen açıklamalar bana aittir).

Bir yandan donanım ve yazılım alanında birçok gelişme yaşanırken, öte yandan bilişsel bilimin temelini atıldığı “Davranışta Beyinsel Mekanizmalar” konulu Hixon

sempozyumunda⁵⁷ (1948), önde gelen psikologlar ve bilgisayar bilimcilerinin yanı sıra von Neumann'ın da katılımcısı olduğu bir toplantı (Nilsson, 2019: 75) gerçekleşmişti. Bu toplantının sonucunda, disiplinlerarası bilişsel bilimin temelleri atılmıştı. Aynı yıl Shannon, “Matematiksel Bir İletişim Kuramı” (A Mathematical Theory of Communication [1948]) adlı makalesinde, bilişsel bilimde oldukça önemli bir yer tutacak *enformasyon kuramını* matematik temeller üzerinde kavramsallaştırmayı başarmıştı. Shannon'un *enformasyon* kuramı, teorik olarak doğrulaması yapılmış bir kuram olup ayrıca günümüz bilgi ve iletişim teknolojilerinde, temel unsurlardan biri olarak görülmektedir. *Enformasyon*⁵⁸ özetle, bir gönderici tarafından bir alıcıya iletilen bilginin (mesajın) olabildiğince az sayıda *bit* (1 ve 0'lardan oluşan “ikili rakam” sözünün bilgisayar bilimlerindeki teknik adı) birimiyle ölçümünün sonucudur yani “bit birimiyle ölçülen bu 'bilgisizlikteki azalış'a 'enformasyon' diyordu Shannon (Say, 2019: 59).

Teletayp ve telgrafı, *enformasyon* iletmek için ayrı bir kanalın iki basit örneği olarak gösteren Shannon, makalesinde genel iletişim sisteminde bu tip kanallar üzerinden iletilen mesajın, kanallarda bulunan parazitten dolayı ne kadar kayıpla ve nasıl bir olasılıkta iletileceğine (1948: 380) dair matematiksel bir teori geliştirmişti. Shannon makalesinde, birer olasılık belirlediği harflerden oluşan kelime benzeri kodlar içerecek İngilizce, Almanca veya Çince gibi doğal dil temelli yapay dil sistemleri önermişti. Shannon'a göre bu yapay diller, çeşitli olasılıkları göstermek için faydalıydı ve mesajlaşma sürecinde kullanımları (1948: 385) uygundu.

Bu yapay diller, çeşitli olasılıkları göstermek için basit problemler ve örnekler oluşturmada faydalıdır. Bir dizi basit yapay dil aracılığıyla da doğal bir dile yaklaşım sergileyebiliriz. Sıfır dereceli yaklaşım, tüm harfleri aynı olasılıkla ve bağımsız olarak seçerek elde edilir. Birinci mertebeden yaklaşıklık, ardışık harflerin bağımsız olarak seçilmesiyle elde edilir ancak her harf, doğal dilde sahip olduğu olasılığa sahip olur (Shannon, 1948: 385).

Bilişsel bilim alanında, Hixon toplantısından daha büyük öneme sahip olduğu düşünülen, en az Shannon'un *enformasyon* kuramı kadar değerli olan Turing'in “Hesaplama

⁵⁷ Bu toplantıya makale sunanlar arasında, von Neumann'ın dışında bir de dönemin ünlü psikologlarından Karl Lashley (1890-1958) vardı ki onun yaptığı konuşma, bilişsel bilimin başlangıç noktası olarak alınmaktadır. “Lashley, beyin işlevi hakkında durağan bir görüş benimsediği için davranışçılığa yüklenmiş, insanın tasarlama ve dil yeteneklerini açıklama yolunda psikologların dinamik ve hiyerarşik yapıları gözden geçirmeye başlaması gerektiğini ileri sürmüştü. Lashley'in konuşması, ileride bilişsel bilim adımı alacak dalın temellerini döşemişti” (Nilsson, 2019: 75).

⁵⁸ *Enformasyon*, açıkça “bilgi” (İng. knowledge) sözcüğünün tanımının içini tam dolduramamaktadır. Bunun yerine *enformasyon*, İngilizce dilindeki “information” sözcüğüne biraz daha yakın bir anlama sahipmiş gibi görünmektedir.

Makineleri ve Zekâ” (Computing Machinery and Intelligence [1950]) adlı makalesi yayımlanmıştı. Bu makale ile birlikte düşünen makine fikri zihinlere yerleşmeye başlamıştı. Bu makale, bilgisayar gibi mekanik bir modelin geliştirilmiş olması sonucunda, insan düşünce etkinliğine benzer türe sahip yapay bir zekânın da yaratılmasının mümkün olabileceği fikrini gündeme getirmişti. Öyle ki sonraki yıllarda, 1956’dan itibaren “yapay zekâ” kavramı ilk defa ciddi anlamda tartışmaya açılmış, bu kavram üzerine bilgisayar bilimlerinde emek harcayanlar dışında dilbilimciler, felsefeciler, psikologlar, mantıkçılar ve matematikçiler tarafından bilimsel çalışmalar yapılmıştı.

Turing’in “Hesaplama Makineleri ve Zekâ” (1950) adlı makalesi, “Turing Testi” olarak bilinen “Taklit Oyunu” (İng. The Imitation Game) başlığı ile giriş yapıyor ve başlık ise 1950’lerin insanlarına kışkırtıcı bir soruyla açılış yapıyordu: Makineler düşünebilir mi? Sorunun, sorulduğu yıldan günümüze değin özellikle zihin felsefesi alanında hararetli tartışmaların konusu olduğu malumumuz. Turing sorduğu soruya, bir taklit oyununu tarif ederek yanıt bulmaya çalışıyor gibi yapsa da asıl niyeti bir makinenin düşünebiliyor olduğuna dair ikna edici kriterler belirlemektir. Bir aptalın ya da zihinsel özürünün dahi düşünebiliyor olduğuna dair hiçbir şüphe yaşanmıyordu, öyle ise bir makine hangi koşullarda “düşünüyor” olarak kabul edilebilirdi? Turing’in yüksek düzeyde kriterler belirlediği “taklit oyunu” için soruları soran bir insan sorgucu (veya hakem), bir bilgisayar (veya makine) oyuncu ve bir insan oyuncu gerekiyordu. Oyuncularla, onlardan farklı bir yerde bulunan sorgucu arasındaki iletişim sistemi yazılı mesajlaşma üzerine kuruluydu. Bu oyunun sonunda sorgucunun, oyunculardan hangisinin kadın, hangisinin erkek olduğuna dair bilgide, bilgisayar tarafından ikna edilmiş (veya yanıltılmış) olması bekleniyordu.

Sorunun yeni biçimi, 'taklit oyunu' dediğimiz bir oyunla tanımlanabilir. Bir erkek (A), bir kadın (B) ve herhangi bir cinsiyetten olabilecek bir sorgucu (C) olmak üzere üç kişi ile oynanır. Sorgucu, diğer ikisinden ayrı bir odada kalıyor. Sorgucu için oyunun amacı, diğer ikisinden hangisinin erkek, hangisinin kadın olduğunu belirlemektir. Onları X ve Y etiketleriyle tanıyor ve oyunun sonunda ya "X, A'dır ve Y, B'dir" ya da "X, B'dir ve Y, A'dır" diyor. (...) Ses tonlarının sorgucuya yardımcı olmaması için cevaplar yazılmalı ya da daha iyisi daktiloyle yazılmalıdır. İdeal düzenleme, iki oda arasında iletişim kuran bir teleyazıcıya sahip olmaktır. (...) Üçüncü oyuncu (B) için oyunun amacı sorgucuya yardım etmektir (Turing, 1950: 433).

Turing’in makalesinde belirlediği yüksek düzeyde kriterler, günümüzde geçerliliği devam eden kriterler olup, henüz tam olarak aşıldığı söylenemez. Bir oyun ve kriter belirlemekten daha fazlası olan Turing’in makalesi, yapay zekânın işaret fişeği olarak kabul edilmektedir ki gerçekten de bu makale büyük bir öngörüyle yazılmıştır. Öyle ki dönemin

insanlarını ve paradigmalarını, yakın gelecekte geleceği muhtemel dijital dünyaya hazırlıyor gibidir. Bu makalede, cinsel bir tahmin oyunu aracılığıyla "düşünme", "zekâ" veya "bilinç" in işlevsel bir tanımı fikrini (Hodges, 2012: 553) ortaya atan Turing, “yapay zekâ” kavramını ilk kullanan kişi olmamasına karşın yine de bu kavramın gündeme gelmesini, uygulanabilir bir alana dönüşmesini sağlamış gibi görünmektedir.

3.2.3. Üçüncü Nesil Programlama Dilleri, İkinci ve Üçüncü Nesil Bilgisayarlar

Turing’in “yapay zekâ”nın işaret fişegi olarak kabul görmüş makalesinin ardından 1955 yılında, “yapay zekâ” kavramının isim babası olduğu kabul edilen bilgisayar bilimcisi John McCarthy (1927-2011), *enformasyon kuramının* yaratıcısı Shannon ve bilişsel bilimin büyük ismi Marvin Minsky (1927-2016) gibi alanında üne sahip birçok isim, Dartmouth Kolejinde 1956 yazında “Yapay Zekâ Yaz Araştırma Projesi” adı altında bir etkinlik yapmak üzere, etkinlik maliyetlerini üstlenmesi beklenen Rockefeller Vakfı’na teklifte bulunmuştu. *Yapay zekâ*⁵⁹ kavramı, ilk defa bu teklifin yazılı metninde kullanılmıştı. Bu proje dahilinde etkinliğe katılanlar, kendi dönemlerinin en parlak bilim insanlarıydı.

New Hampshire, Hanover’deki Dartmouth Koleji’nde 1956 yazı boyunca gerçekleştirilecek, 2 aylık ve 10 kişilik bir *yapay zekâ* araştırması öneriyoruz. Bu araştırma, öğrenmenin ya da zekânın herhangi başka bir yönünün ilkesel olarak bir makine tarafından taklit edilecek şekilde tam olarak belirlenebileceği varsayımı temelinde ilerleyecek. Makinelere doğal dil kullandırmanın, soyutlamalar ve kavramlar oluşturtmanın, şimdilik insanlara özgü sorun türlerini çözdürtmenin, kendilerini geliştirtmenin yollarını bulmaya çalışacağız. Eğer özenle seçilecek bir bilim ekibi bütün yaz boyunca birlikte çalışırsa bu sorunlardan en az biri ya da daha çoğunda önemli ilerlemeler kaydedileceğini düşünüyoruz. (...) Mevcut amaç uyarınca, *yapay zekâ* sorunu, insan öyle davrandığında zeki olarak nitelendirilecek davranışları makinelere yaptırtmak olarak tanımlanacak (McCarthy vd., [1955] 2006: 12; Nilsson, 2019: 81 italik vurgu bana ait).

⁵⁹ Günümüzde ayrı bir disiplin olarak kabul edilen ve birçok özel araştıma konusunun çatısı altında toplandığı *yapay zekâ* kavramı, ortaya çıktığı 1956 yılından bu yana tartışma konusu olmayı sürdürmektedir. Şöyle ki ilk çıktığı dönemlerde *yapay zekâ* kavramı üzerinde zaten bir anlaşmazlık vardır. Bu kavramın, topluluğun gerçek niyetini ortaya koymada yetersiz olduğuna ve farklı çağrışımlar uyandırdığına dair endişeler mevcuttur. *Yapay zekâ* kavramı henüz çok yeniyken, “makine öğrenimi” kavramını ortaya atan erken yapay öğrenme çağının öncülerinden Arthur L. Samuel (1901-1990) şöyle demiştir: “*Yapay* sözcüğü, bu işte bir düzmece olduğu hissini veriyor ya da sanki bu iş tamamen yapaymış, hiçbir gerçekliği yokmuş düşüncesini uyandırıyor. (...) Bu ifadeyi ne Newell ne Simon beğenmişti; dolayısıyla sonraki yıllarda kendi çalışmalarını, *karmaşık bilgi işleme* olarak adlandırdılar” (McCorduck, 2004: 115 italik vurgu bana ait). Newell, bir zaman sonra bu kavram üzerinde uzlaşma varmış gibidir: “O halde *yapay zekâ* adını bağrımıza basalım. Güzel bir isim bu. Tüm bilimsel alanların ismi gibi büyüyecek ve bu alanın tam da kastettiği anlama kavuşacaktır” (Nilsson, 2019: 82). Günümüz anlayışında bir *yapay zekâ* tanımı da Türk akademisyen Cem Say’dan gelmiştir: “Doğal sistemlerin yapabildiği (zekice olsun veya olmasın) her bilişsel etkinliği (gerekirse bedenleri olan) yapay sistemlere, daha da yüksek başarımlar düzeylerinde nasıl yapabileceğimizi inceleyen bilim dalıdır” (2019: 83).

Günümüzde pek çok ayrı alanda yürütülen yapay zekâ çalışmalarının kavramsal temelleri, yukarıda saydığımız isimlerle birlikte diğer katılımcılar tarafından atılmıştı. Bu isimler, yenilikçi fikirleriyle yapay zekâ kavramının etrafında toplanan sorunlar ışığında farklı çözüm yöntemleri önermişlerdi. Bu yöntemler, yapay zekânın disiplinlerarası çalışılmasının nedenlerini ortaya koyacak şekilde birbiriyle bağdaşık olmayan disiplinlerin deneyimlerinden faydalanmayı gerektiriyordu. Örneğin otomatlar, hesap makineleri, Turing makineleri ve dijital bilgisayarlar, mühendislikten gelen çalışmaların bir sonucuydu. Somut adımlar bu temelden yükselecekti. Davranış geliştirme, örüntü tanıma, yazı karakteri tanıma gibi hedefler ise evrim, psikoloji, bilişsel bilim ve sinir biliminden gelecekti. Bunlar, yaşamdan gelen unsurlardı. Tüm bu yapay zekâ çalışmalarının somutlaştırılması için gereken donanım ve diğer temel gereçler hazır olduğuna göre, yazılım kısmına odaklanılacaktı. Çalışmalar, yapay zekâyâ özel programlama dillerinin geliştirilmesi yönüne kaymıştı. Burada ise dilbilimle birlikte ağırlıklı olarak matematik, mantık ve felsefe devreye giriyordu.

Yapay zekâyâ özel, Frege'nin birinci seviye yüklem mantığı temelinde yükselen λ -*kalkülüs* ile yazılmış ilk işlevsel programlama dili olan LISP, bu amaçla ortaya çıkmıştı. Ancak, öncelikle nesillendirmeye sadık kalabilmek adına, beşinci nesil çok yüksek seviye bir programlama dili olan LISP'ten başka amaçlarla yaratılmış diğer programlama dillerine değinmekte yarar var.

3.2.3.1. FORTRAN (FORMula TRANslator), ALGOL (ALGORithmic Language) ve COBOL (COMmon Business Oriented Language)

Programlama dillerinin 1950'li yıllarda çeşitlenerek artmasının bir nedeni, bilgisayarların başarılı şekilde donanım kazandırılmış yapıları üzerinde problem çözmeye ve sorun ayıklamaya daha uygun ve ayrıca insan anlama seviyesinde dillere duyulan ihtiyaçtır. Diğer bir neden ise 1956'lı yıllardan sonra "yapay zekâ" kavramının, bilişim alanında yerleşik bir kavrama dönüşmesiyle birlikte, yapay bir zekâyâ iş yaptırabilmek için farklı ölçütler üzerinden programlama dilleri geliştirilmesi ihtiyacı olarak sıralanabilir. Bundan dolayıdır ki programlama dilleri, amaçlar doğrultusunda öncelikle iki farklı alanda dallanarak geliştiler. İlerleyen sayfalarda serimlemeye çalışacağımız üzere, zaman içinde teknoloji, enformasyon, bilgisayar bilimleri, yapay zekâ ve bilişsel bilimin gelişim ihtiyaçlarına binaen birçok farklı nedene bağlı olarak dallanmalar yapmış programlama dilleri yaratılmıştır. Bu bölümde ise üçüncü nesil dillerin erken döneminde ortaya çıkmış ata programlama dillerini serimlemeye çalışacağız.

Üçüncü nesil programlama dillerinden önce, 1954'lü yıllara kadar makine dilleri, *Assembly* dilleri ve *derleyiciler* mevcutken, bu yıldan itibaren gerçek bir programlama dilinin tanımını verebilecek tarzda programlama dilleri geliştirilmişti. Erken dönem-geç dönem ayırımına gidilmeden önce, denilebilir ki üçüncü nesil programlama dillerinin tamamı aynı kaygıyla ortaya çıkmıştı: “Nasıl programlanabilir?”

Tanıtımı 1957 yılında yapılan ilk yüksek seviye programlama dili FORTRAN'ın tasarımında, IBM 704'ün tasarımcıları rol almıştı. Günümüzde mühendislik alanında çeşitli sürümleri (bilinen son sürümü 2008 yılına ait olup, evrimi hala sürüyor) kullanılan FORTRAN, ilk bilimsel programlama diliydi. Veri işleme, kağıda döküm verme, heaplanabilir ifade, GOTO işleç kolaylığı (belirlenmiş bir satıra yönlendirme işleci), koşullu ifade olan *if... then*'li yapı (eğer...ise), yineleme yapmayı sağlayıcı iterasyon, DO döngüleri (YAP döngüleri) gibi özellikleriyle (O'Regan, 2016: 193) birlikte FORTRAN, basit dilsel yapısıyla, ilk genel amaçlı taşınabilir programlama diliydi. İlk defa makineden bağımsız kod üretme özgürlüğü gelmişti. Genel amaçlı bir dil olan FORTRAN, hem ticari hem de bilimsel amaçlara hizmet ediyordu. Kullanıcı dostu bir otomatik programlama diliydi fakat makineden bağımsızlık, kendi diline özel *derleyiciler* oluşturulmasını gerektirmişti. Bu, FORTRAN'ı daha özel bir dil haline getiriyordu. FORTRAN'in ilk versiyonundan sonra, FORTRAN I, II, IV, 77, 90, 95, 2003, 2008 versiyonlarıyla soyu devam etmektedir. Örneğin, FORTRAN77 programlama dilinde yazılmış “Hello World” ifadesi, aşağıdaki gibidir (aşağıdaki ifade [O'Regan, 2016: 194]'den alınmıştır):

```

PROGRAM HELLOWORLD
C   FORTRAN 77 SOURCE CODE COMMENTS FOR HELLOWORLD
      PRINT '(A)', 'HELLO WORLD'
      STOP
END

```

FORTRAN Boole cebiri, tamsayı, reel sayı içeren ilk versiyonundan sonra, karmaşık sayıların eklendiği yeni versiyonları ile birlikte özellikle matematik, bilim ve mühendislik hesaplama alanlarında yardımcı olmaktadır. Günümüzde FORTRAN, “iklim modellemesi, güneş sistemi simülasyonları, yapay uyduların yörüngelerinin modellenmesi ve otomobil çarpışma dinamiklerinin simülasyonu gibi uygulamalar için popüler bir bilimsel programlama dili” (O'Regan, 2016: 194) olarak kullanılmaya devam etmektedir.

Programlama dilleri evriminde önemli bir yere sahip ata dillerden FORTRAN'ın donanım bağımsız olması ve yukarıda saydığımız diğer özelliklere sahip olması, onu geniş

çevrelerce kullanılabilir duruma getirmişti fakat FORTRAN'ın başaramadığı işler de vardı. Bunlardan ilki, bilgisayar donanımının ve çevre birimlerinin hızla gelişmeye devam ediyor olmasıydı. Dijital bilgisayarlar üzerinde anlaşılması belli bir standart donanım ve mimari olmadığından, programların bu donanımlar için yeniden yazılması gerekiyordu, bu durum diğer programlama dilleri gibi FORTRAN'ı da geride bırakıyordu. Nihayetinde FORTRAN, sistem programlama işinde kullanılmak için üretilmemişti. Bu nedenle, sistem programlama işi *Assembly* tarafından yapılmaya devam ediyordu. İkinci bir neden ise Amerikan laboratuvarları kökenli olan FORTRAN, evrensel bir dil olarak kabul edilmiyordu. Bu dönemde yeni birçok programlama dili üzerinde çalışılıyordu ve Amerika başı çekiyordu. Avrupalı bilgisayar bilimcileri bu işin gerisinde kalmak istemiyorlardı. Böylece FORTRAN'ın eksiklerini gidermek üzere Amerika-Avrupa ortak çalışması olan ALGOL ortaya çıktı.

Kökenini atası FORTRAN'dan alan ilk ALGOL sürümü olan ALGOL58, atasına nazaran birçok yenilik içeren *algoritmik* bir yapı sergiliyordu. Bu dilin sergilediği *algoritmik* yapı ise Avrupalı atası *Plankalkül* adlı programlama dilinden geliyordu. Şöyle ki tezimizin 3.2.1 bölümünde yer verdiğimiz dünyanın ilk serbestçe programlanabilir bilgisayarı olan Z3'ün tasarımcısı Zuse'nin yarattığı Almanca kökenli bu dil, algoritmasının yazıldığı 1946 yılında İkinci Dünya Savaşı dolayısıyla hayata geçirilememişti. Kendisine özel bir *derleyici* ancak elli yıl sonra yazılabiliyordu ve ancak o zaman erken dönem bir yüksek seviyeli programlama dili olduğu anlaşılıyordu. Algoritmik yapısını *Plankalkül*den alan ALGOL, akademik çevrelerce oluşturulmuş güçlü bir dildi. Bu dil, FORTRAN'a nazaran sentaktik kurulum açısından özenle hazırlanmıştı (Bkz. 3.3.2.1.). FORTRAN gibi ALGOL de emir kipli (prosedürel) programlama dilleri ailesinin atalarından biriydi. Bir sonraki sürüm olan ALGOL60, 1960 yılında geliştirilmişti ve her açıdan halefleri ALGOLW ve ALGOL68'den daha mükemmeldi. İlk defa ALGOL dilinin ortaya çıkmasıyla birlikte, semantik açıdan değilse de sentaktik olarak ulusal dillerin ötesinde algoritmik bir *üst dil* inşası gerçekleşmişti.

Avrupa ALGOL'den yararlandı. Algoritmalar yalnızca ulusal dil engellerini aşarak iletilebilir hale gelmekle kalmadı, aynı zamanda mevcut herhangi bir dilden daha iyi bir dil tanımlanmıştı. (...) Özellikle ALGOL60, Avrupalı bilgisayar uzmanlarının dil işleme tekniklerinde üstün bir yeterlilik kazanacakları işlemci yapıları için yeterince yeni derleme tekniklerini gerektirdi. (...) Güzelce düzenlenmiş, baştan çıkarıcı derecede eksik, biraz belirsiz, okunması zor, formatta tutarlı ve kısa, aynı özelliklere sahip bir dil için mükemmel bir tuvaldi. İncil gibi, sadece okunması değil, yorumlanması da gerekiyordu. ALGOL60'ın sözdiziminin tanımlandığı kesinliğin, semantiği için elde edilmemiş olması şaşırtıcı değildi. (...) ALGOL60 tasarımcıları için ne onun ne de başka bir yeterli biçimsel yapı için anlamsal tanımlama tekniği mevcuttu. Artık anlambilimin sözdiziminden çok daha zor tedavi edildiğini biliyoruz. Sonuç olarak anlambilim İngilizce olarak verilmişti (Perlis, 1981a: 86, 89).

Özyinelemeli yapıları destekleyen (bir fonksiyon veya prosedürün kendisini aramasına izin veren), koşullu ifadeler *if... then* (eğer...ise) kullanılmasına olanak veren yapısal kurulumu sahip ALGOL ailesinin temel özellikleri şunlardır: Alt simge aralığının değişkenler tarafından belirtildiği “dinamik diziler” kullanılmasını sağlar. Programcı tarafından tanımlayıcı olarak kullanılmasına izin verilmeyen anahtar kelimeleri “ayrılmış kelimeler” olarak mevcuttur (O’Regan, 2016: 196). Kullanıcının kendi veri türlerini tasarlamasına izin veren “kullanıcı tanımlı veri türleri” (O’Regan, 2016: 196) vardır. “ALGOL parantez içinde ifade blokları kullanır ve blokları sınırlandırmak için başlangıç-bitiş çiftlerini” (O’Regan, 2016: 196) kullanır. ALGOL, başla-bitir (BEGIN-END) tarzında “ifade blokları” kullanan ilk dildir.

ALGOL68’den sonra ALGOL ailesine ait kalıtımsal özellikler, yüksek seviyeli geç dönem üçüncü nesil programlama dilleri olan C, BASIC ve PASCAL ailelerinde yaşamaya devam etmiştir. Bunun yanı sıra ALGOL, PL/1, SIMULA 67, ADA ve Java gibi başka programlama dillerinin de atası sayılmaktadır. ALGOL60’ın haleflerine ve başka dil ailelerine aktardığı kendine has özelliklerinden olan “değere göre çağırma “ve “ada göre çağırma” parametreleri, alt programlara geçmenin iki yolu olarak sunulmaktaydı (O’Regan, 2016: 195). Günümüzde, ALGOL’ün özünü taşıyan C/C++ ailesinde bu parametreler kullanılmaktadır. Değere göre çağrı parametresi, “fonksiyon veya prosedür girilmeden önce bir fonksiyon veya prosedürün argümanlarının değerlendirilmesini” (O’Regan, 2016: 195) sağlamaktadır. Ada göre çağrı parametresi ise “formel parametre her okunduğunda gerçek parametre ifadesinin yeniden değerlendirilmesini” (O’Regan, 2016: 195) sağlamaktadır.

ALGOL’ün yukarıda sayılan özellikleri dışında, bu dil üzerine yapılan toplantılarda, bir sentaks ve semantik ayrımı konuşulmaya başlanmıştır çünkü ALGOL, farklı kıtalarda kullanım alanına erişmiş *evrensel* bir dil olma potansiyeli taşıyordu. “Bu dil hem tanıtılan kavramlar hem de sözdizimi için biçimsel gösterimin ilk tanınan kullanımını açısından programlama dillerinin teknik gelişiminde bariz bir dönüm noktasıydı” (Sammet, 1981a: xviii), bu nedenle “1960’tan bu yana çoğu teorik ve pratik dil ve *derleyici* çalışması ALGOL60’a dayanmaktadır (Sammet, 1981a: xviii) ancak ALGOL dil ailesi ALGOL68’den sonra, iç nesillenmesini sürdürememiştir. IBM’in ticari baskısı⁶⁰, FORTRAN’ın kendi

⁶⁰ IBM firması, 1954’lü yıllardan itibaren bilgisayar dünyasında geniş bir pazar payı elde etmiştir. Örneğin, “Karayolları Genel Müdürlüğüne 1960 yılında kurulan Türkiye’nin ilk bilgisayarı da bir IBM 650 idi” (Mutlu, 2020: 12). IBM, kendi donanımına uygun geliştirilmiş FORTRAN’ın karşısına çıkan Avrupa laboratuvarları kökenli ALGOL’ü desteklememişti. Ticari olarak daha güçlü olan IBM’in yüksek pazar payına sahip bilgisayarlarının kullanım oranı yükseldikçe, FORTRAN da hem nesillendi hem de daha çok tercih edilir oldu. ALGOL, böylece akademik bir başarı olarak belirli bir kesim tarafından kullanılır ve takdir edilir duruma geldi.

nesillenmesi yoluyla sürdürülmesini sağladığı kadar ALGOL'ün önünü tıkayıcı bir rol oynamıştır (O'Regan, 2016: 195). Akademik bir çalışmanın ürünü olan ALGOL'ün hem makine bağımsız hem de daha önemlisi ticari ve politik bağımsız oluşu, ailenin iç nesilsizliğindeki diğer nedenler olarak sıralanabilir.

Ticari amaçlarda kullanılmak üzere tasarlanmamış olan FORTRAN ve ALGOL'ün aksine COBOL, tamamen ticari bilgi işlem maksadıyla ortaya çıkmıştı. Bilgisayarlar için yazılmış erken dönem matematik dillerinden olan FLOW-MATIC, COBOL'un atasıdır. COBOL ise FORTRAN ve ALGOL ile birlikte, PL/1 programlama dilinin atasıdır. PL/1, günümüzde kullanımı devam eden bir programlama dili olup, COBOL'dan üstündür. Hopper ve diğer bilgisayar bilimcileri tarafından 1959 yılında yaratılan COBOL, ALGOL'ün çok gerisinde olmasına karşın yine de çeşitli versiyonlarıyla günümüze kadar ulaşmayı başarmıştır⁶¹. Sıralanamayacak kadar çok versiyon yenilenmesi geçiren COBOL, ticari hayatın yeniliklerine ve taleplerine uygun şekilde kendini güncellemeye zorlanmış görünmektedir.

Hem bireyler hem de şirketler olarak, dili yeniden yazmanın kısmen onların işi olduğunu düşünüyorlar. (...) Söзде geliştirilmiş ve sabit bir dil olmasına rağmen, insanlar "parlak fikirlere" sahip oldukları için değişiklikler yapılıyor. (...) IBM, Minneapolis-Honeywell ve General Electric'in hepsinin farklı dilleri var. Doğal olarak sadece COBOL'u kendi dillerine göre bir kalıba sokmakla ilgileniyorlar (Sammet, 1981c: 215).

COBOL, genel amaçlı kullanımı hedefleyen FORTRAN ile cebirsel ve algoritmik yapısıyla evrensel görünüm kazanmış ALGOL'ün aksine, erken dönem yüksek seviyeli bir dil olarak, oldukça ilkel bir sözdizimi içermekteydi. Önceki sayfalarda değindiğimiz üzere Hopper, doğal dillere olabildiğince yakın görünüm sergileyecek, özellikle İngilizce dilinde bir programlama dili yaratmak istiyordu. Hopper'ın isteğinin gerçekleştiğini gösteren bu programlama dilinin sözdizimi, İngilizceye oldukça yakın bir görünüm sergilediği için öğrenmesi ve kullanması oldukça kolaydı. Bu dildeki veri türleri, yalnızca sayılar ve metin dizileriydi ve bu veri türleri, diziler ve kayıtlar halinde gruplandırılabilirdi (O'Regan, 2016: 195). Bu dil, ayrıntılandırılmış yapısına karşın, basit emir kipli yapılar sergileyen gramersiz İngilizce görünümü sergilemekteydi. Bu özelliğinden dolayı COBOL ile yazmak

IBM firmasının FORTRAN programlama dili için yaptığı yanlı tercihin, bugünkü programlama dilleri tarihini olumsuz yönde etkilediğine dair düşüncelerin olduğunu eklemekte yarar var.

⁶¹ COBOL, "kullanımı ABD Savunma Bakanlığı tarafından zorunlu kılınan ilk bilgisayar diliydi" (O'Regan, 2016: 195). COBOL, işyeri yönetiminde kullanılmaya devam ediyor. Bununla birlikte, günümüzde bankalarda ve ATM'lerde kullanılmaktadır.

hem kolay hem de zevkli oluyordu. Buna uygun bir örnek, (aşağıdaki örnek [O'Regan, 2016: 195]) aşağıdaki gibidir.

DIVIDE A BY B GIVING C REMAINDER
A'YI B'YE BÖLÜN C KALAN ŞEKLİNDE

Erken dönem üçüncü nesil yüksek seviye programlama dillerinin bir izdüşümünü vermeye çalıştığımız bu bölümde, bu dillerin ilk örneğinin *Plankalkül* olduğunu yeniden hatırlamakta yarar var. Özellikle, “Avrupa'daki algoritmik dillerin ilk fikir kaynağı, 1948'de Zuse tarafından yayımlanan *Plankalkül* adlı (Perlis, 1981b: 147) eserdir. Tezimizde özelliklerini kısaca özetlemeye çalıştığımız *Plankalkül*, yüksek seviye dillerden beklenen şartları yerine getirebilen ve koşullu yapı (*if...then/if...else*) sergileyen bir dil olup, oyun oynayan bir program için yazılmıştır. Bu dil, hayata geçirilip uygulama alanına alınmadığından dolayı başka hangi amaçlara hizmet edebilir durumda olduğuna dair bir fikrimiz bulunmamaktadır.

3.2.3.2. Simula, BASIC (Beginners All Purpose Symbolic Instruction Code), Pascal, C ve Java

Programlama dillerinin paradigma değişimleriyle geliştirilme ve yeniden yazılma sürecine hızlandırıcı etkide bulunan bilgisayarlar, kendi içinde de nesillenmeler yaşıyordu. Hemen her gün teknolojik yenilikler gerçekleşiyordu. Örneğin, transistörlü bilgisayarlarda vakum tüplerinin yerini ayrı ayrı kullanılan transistörler ve baskılı devrelerin alması neticesinde daha küçük, daha güçlü ve daha güvenilir bilgisayarların tasarlanabildiği (Mutlu, 2020: 13) ikinci nesil bilgisayarların dönemi (1959-1964) kısa sürede sonlanmıştı.

Üçüncü nesil bilgisayar döneminin başladığı (1964-1972)⁶² yıllarda, bilgisayar çevre birimine bilgisayar *faresi* (İng. mouse), yeni bir üye olarak katılmıştı. Bu dönemde ortaya çıkan yeni teknolojiler, “transistörlerin tek tek devre kartlarına lehimlenerek kullanılması zorunluluğunu kaldırarak, karmaşık devrelere ait transistörlerin hepsinin bir tümleşik devreye basılabilesine olanak” (Mutlu, 2020: 14) sağlamıştı. Bu dönem ortaya çıkan bilgisayarlar, mini bilgisayarlar olup, bağımsız firmalar tarafından üretiliyordu. Programlama dillerinin de üçüncü nesil bilgisayar teknolojisi paralelinde geliştirilmesi, amaca yönelik özelleşmesi

⁶² Bu dönem teknolojileri, günümüzü oldukça etkileyen enformasyon ve dijital çağın gelişiminin habercileridir. “1965 de ilk süper bilgisayar olan CDC6600 üretildi. IBM 1967’de floppy disk icat etti. (...) Douglas Engelbart 1968’de fare (İng. mouse), ekran pencereleri, hipermetin ve tam ekran kelime işlem olanaklarını içeren çalışan bir interaktif bilgi işlem demosu gerçekleştirdi. Aynı yıl Robert Noyce ve arkadaşları tarafından tümleşik devreler geliştirmek amacıyla Intel firması kuruldu. İlk programlanabilir hesap makinası olan Hewlett-Packard 9100A, bu yıl piyasaya sürüldü” (Mutlu, 2020: 15).

bekleniyordu. Kaldı ki bu dönemde, günümüz internet protokollerine dair ilk atılımlar da gerçekleşmişti. Bir ARPA projesi olarak 1969’da paket anahtarlama ağ geliştirilmiş ve bu yolla bilgisayarlar arasında güvenilir bir ağ oluşturulabilmişti (Mutlu, 2020: 15). İlk ARPANET bağlantısı, Stanford ve UCLA arasında kurulmuştu ki bu bağlantı gelecekteki internetin ilk iki düğümünü (Mutlu, 2020: 15) oluşturmuştu.

Şimdi, üçüncü nesil erken dönem programlama dillerinin soylarının devamında farklı laboratuvarlarda tasarlanmış olan bu neslin geç dönem örneklerinin kısa bir özetini, programcılıkta başarılı olmuş ilkleri sunabilmek adına vermeye çalışacağız. Bu dillerden ilki olarak ele alacağımız Norveç kökenli *Simula*, atası ALGOL60 soyunun sadık bir devamıydı. Ole-Johan Dahl (1931-2002) ve Kristen Nygaard (1926-2002) tarafından 1962 yılında tasarlanan bu dilin 1967 yılında tasarlanan *Simula67* versiyonu, programlamacılıkta bir paradigma değişimini haber veren ilk *nesne tabanlı* (İng. object-oriented) programlama dili olarak karşımıza çıkmaktadır. Nesne tabanlı bir programlama dilinde *nesne*, “belirli bir tipten yaratılan değişken gibi belirli bir sınıftan yaratılan bir kopyadır” (Kurt, 2002: 49). Bu şekilde program, doğrudan bu yolla belirlenmiş nesnelere üzerinde işlem yapar ki bu nesne tanımlamalarının her biri bir fonksiyon tanımlaması olarak görülebilir.

Programlamacılıkta, 1960’lı yılların sonundan 1975’li yıllara kadar olan dönem, teknolojik atılımlar karşısında programlama dillerinin rüştünü ispatlama dönemi gibidir. Yapısal programlama, kullanıcının büyük kod parçalarını daha kısa alt yordamlar halinde yazabilmesini sağlar. Yapısal programlama, yordamsal olarak da bilinen buyruklu (emir kipli) programlamalardan türemiş bir alt daldır. Yordamsal diller, her satırda bilgisayara ne yapması gerektiğini bildirir. “Yapısal programlama, koşul (*if / then / else*), döngü (*while / for*), blok, alt yordam (*procedure*) ve işlev (*function*) gibi yapısal ifadelerle; programın okunabilirliğini, kalitesini ve geliştirme süresini iyileştirmeyi hedef alan bir yaklaşımdır” (Kayabaş, 2020: 219). Bu yolla GOTO işlecinden (belirlenmiş bir satıra yönlendirme işleci) bağımsızlaşma hedeflenir. GOTO işlecinden kurtulma talebi, yapısal programcılığın önünü açmaya çalışan ve bu alanda ciddi çalışmalar yapmış olan Hollandalı Edsger W. Dijkstra’dan (1930–2002) gelir. Dijkstra,

bilgisayar biliminde yaptığı çok sayıda icatla ünlüdür ki grafiklerdeki en kısa (ya da en masrafsız) yolu bulan algoritma da bunlara dahildir. Aynı zamanda, “yapısal programlama” [*structured programming*] denen şeyi de savunuyordu; bu yöntem, program yazmanın (ve anlamının) verimini çok artırmıştır (Nilsson, 2019: 405).

Ataları FORTRAN ve ALGOL'ün soyundan gelen BASIC, 1963 yılındaki ilk tasarımıyla, Dijkstra'nın gelişim sürecine katkıda bulunduğu birçok yeni programlama diline nazaran eksiklikti. Amerikan laboratuvarları kökenli BASIC, John G. Kemeny (1926-1992) ve Thomas E. Kurtz (1928-...) tarafından 1963 yılında tasarlanmıştı. Adının açılımından anlaşılacağı üzere, BASIC diğer programlama dillerinden farklı olarak, programlama dili öğrenme merakı içinde olanlar ve programlama veya yazılım alanında yetişen öğrenciler için tasarlanmıştı. Bu dil, ataları FORTRAN ve ALGOL gibi GOTO ve koşullu yapı (*if...then*) tasarımıyla, özellikle 1980'li yıllara kadar yaygın bir kullanım alanına sahipti. Eğitim amaçlı tasarımı, günümüzde halen kullanılan tüm ata dillerin en basiti olmasına neden olmuştu ve birçok öğrenci için "birincil dil" (Sammet, 1981: xix) olarak kabul ediliyordu. BASIC, basitliği nispetinde kusurluydu çünkü bu dili öğrenerek programcılığa başlamış birinin diğer herhangi bir programlama dilini öğrenme çabasında gerekli disiplini göstermesi zordu. Aşağıda, 1'den başlayarak klavyede yazılacak bir N sayısına kadar yapılacak tüm sayı sıralamasının, BASIC programlama dili ile yazılmasına dair bir örnek görülmektedir (aşağıdaki örnek, [Aslan, 2006: 48]'den alınmıştır):

```

10 PRINT          "LÜTFEN BİR SAYI GİRİNİZ"
20 INPUT N
30 FOR   K=1 TO N
40 PRINT  K
50 NEXT  K
60 END

```

ALGOL soyunun devamı olan *Pascal* programlama dili, genel amaçlı bir dil olsa da BASIC gibi kolay öğrenilebilir yapısı sayesinde, programlamaya yeni başlayanlara yapısal programlamayı öğretme amacıyla Niklaus Wirth (1934-...) tarafından 1970 yılında tasarlanmıştı. *Pascal*, dönemin paradigmasına uygun şekilde yapısal bir programlama diliydi ve sonraki sürümleri nesne yönelimli olacak şekilde geliştirilmişti. 1970'li yılların başında özellikle mikro bilgisayar (ev bilgisayarı) pazarında çok başarılı olmuştu. Üniversite düzeyinde programlama dili eğitimlerinde 1990'lı yılların sonuna kadar yaygın bir kullanıma sahipti (Sebesta, 2016: 98). Tıpkı atası ALGOL60 gibi akademik çevrelerde yaşamaya devam eden *Pascal*, mükemmel bir programlama dili olmaya aday gösterilse de ticari yönelim bu dilin önünü kesmiş görünmektedir.

Çeşitli kaynaklarda orta seviyeli bir dil olarak tanımlanmasına rağmen, taşınabilirlikte⁶³ gösterdiği başarıdan dolayı çoğunlukla yüksek seviyeli bir dil olduğuna dair vurguların yapıldığı, yapısal ve nesne tabanlı C programlama dili, *Pascal* gibi ALGOL soyundan gelir. Ken Thompson (1943-...) ve Dennis Ritchie (1941-2011) tarafından Bell laboratuvarlarında doğmuş, Amerikan kökenli bir programlama dilidir. “C yordamsal bir programlama dilidir ve 'if ifadesi', 'switch ifadesi' gibi koşullu ifadeleri, 'while' ifadesi veya 'do' ifadesi gibi yinelemeli ifadeleri ve atama ifadesini içerir” (O’Regan, 2016: 198).

Kullanım kolaylığı ve verimlilik konusunda genel amaçlı evrensel bir programlama dili görünümü sergileyen “C dilinin geliştirilme amaçlarından bir tanesi Unix⁶⁴ işletim sistemini geliştirmektir” (Şamlı, 2020: 157). Sistem yazılımı konusunda iyi bir zemine sahip olan C ve kendi soyundan gelen C++ programlama dillerinin popülerliği devam etmektedir. *Assembly*’nin yapabildiklerini yapmak amacıyla tasarlandığı rahatlıkla söylenebilecek olan “C, gelişimini tamamlamış ve standardı oluşmuş bir dildir: C programlama dili, (...) *derleyicisi* değişse bile karakterler vb. için hangi tipi kullanacağını düşünmek zorunda kalmaz” (Şamlı, 2020: 158 italik vurgu bana ait). C dilinin zengin bir kütüphanesi vardır ve Pascal programlama diline göre daha kısa bir kod yazımına sahiptir. Şimdi sırasıyla bu iki programlama diline örnek vereceğiz. Aşağıda, Pascal programlama dili ile ekrana “Merhaba Dünya” yazdıran kodlar görülmektedir (örnek gösterim [Kayabaş, 2020: 209]’dan alınmıştır).

```
program OrnekProgram(output) ;
begin
    Write('Merhaba dünya!')
end.
```

⁶³ C ve soyundan gelen C++ günümüzde oldukça yaygın bir kullanıma sahiptirler. Taşınabilirlik özelliğini oldukça geliştirmiş C ailesi, neredeyse her sistemde kullanılabilir bir dil haline gelmiştir. “Örneğin C programlama dilini kullanarak yazılan bir bilgisayar programı Microsoft Windows işletim sistemini kullanan bir sistemde çalıştığı gibi, Linux, Unix ya da başka bir işletim sistemi kullanan bir makinede kolaylıkla çalıştırılabilir” (Şamlı, 2020: 157).

⁶⁴ C programlama dili, *Assembly*’e benzer şekilde makine ve programcı arasında dengeli bir ilişkiye sahiptir. “Orta seviye bir dil olduğundan hem hız ve performans gerektiren donanım bazlı işlemlerde hem de gelişmiş grafiksel kullanıcı arabirimleri oluşturma görevlerinde rahatlıkla kullanılacak dengeli bir yapıya sahiptir. Diğer dillere oranla öğrenmesi daha zor” (Kayabaş, 2020: 210) olsa da günümüz dünyasında *Assembly*’ye nazaran daha tanıdık ifadeler aracılığıyla kodlar yazılmasına izin verirken hızını düşürmeyen bu dengeli dil, *Assembly*’e ciddi bir rakip olarak görülmektedir. “Önceleri çoğunla UNIX tabanlı sistemlerde yoğun olarak kullanılan C, kişisel bilgisayarların yaygınlaşmaya başladığı 1980’ler ile birlikte hemen hemen her tür bilgisayar sistemine yayılmıştır. 1983 yılında ANSI tarafından standardize edildikten sonra yüksek oranda taşınabilir bir sistem programlama dili olmuştur. C++ ise C’den türetilmiş nesne yönelimli bir programlama dilidir. Windows, MacOS ve Android işletim sistemlerinin yanı sıra Google Chrome, Adobe Photoshop, AutoDesk Maya, MySQL gibi her biri kendi alanının en iyisi olan pek çok uygulama yazılımı C / C++ ile geliştirilmiştir. Sözdizimi ve çalışma mantığı pek çok farklı dili etkilemiştir” (Kayabaş, 2020: 210).

Aşağıda, C programlama dili ile ekrana “Merhaba Dünya” yazdıran kodlar görülmektedir (örnek gösterim [Kayabaş, 2020: 210]’dan alınmıştır).

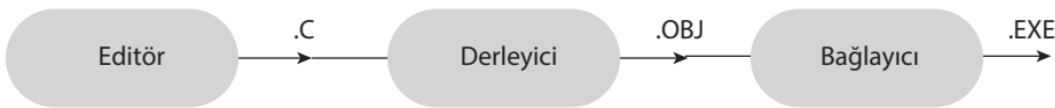
```
#include <stdio.h>
int main(void)
{
    printf(“Merhaba dünya!”);
}
```

Genellikle *Pascal* ve C programlama dilleri birbirleriyle kıyaslanır, bu kıyaslamadan ise C dilinin lehine sonuçlar doğar: “Pascal’ı orijinal hedefinin çok ötesinde bir şey için kullanmanın bir hata olduğunu hissediyorum. Saf haliyle Pascal, öğretmek için uygun ancak gerçek programlama için uygun olmayan bir oyuncak dildir” (Kernighan, 1981: 1). Kernighan’a göre *Pascal*, örneğin sistem programlama görevleri ve hatta işletim sistemleri için tavsiye edilmesine rağmen, bu anlamda C programlama diliyle yarışabilecek düzeyde bir dil yapısına sahip değildir. C’nin ve Unix işletim sistemi gibi sistemlerin gelişim süreçlerini birbirlerine yardım ederek geçirdikleri de söylenebilir. Zengin kütüphanesi ve genel amaçlı kullanımıyla yaygınlık kazanmış olan C programlama dili ailesi, yeri geldiğinde değineceğimiz “makine öğrenimi” ve “derin öğrenme” kitaplıklarının yazılımında da kullanılmıştır. Günümüzde oldukça yaygın bir kullanıma sahip olan C ailesi dilinde kodlama öğrenmenin ve yazmanın kolay olduğunu söylemek ise imkansızdır.

Birbirinden farklı amaçlarla ortaya çıkmış *Plankalkül*, FORTRAN, ALGOL ve COBOL’un amaçları dışında kalan alanlarda eksik kaldıkları yönlerini vurgulamaya çalıştığımız bu bölümde, gösterdikleri farklılıklara karşın, bu dillerin kullanım alanlarına göre sağladıkları avantajların ortak yönlerini de vurgulamakta yarar var. Bu diller, atası oldukları diğer geç dönem üçüncü nesil orta ve yüksek seviye programlama dilleri olan BASIC, Pascal, C, C++ gibi diller ile birlikte değerlendirildiğinde, barındırdıkları ortak özellikleri aşağıdaki verilen şekilde özetlenebilir (verilen özellikler [O’Regan, 2016: 191]’den alınmıştır):

- Kolay okunabilir olma
- Açıkça tanımlanmış sözdizimi (anlambilim açısından hala çalışılıyor)
- İş ve bilimsel uygulamalara uygun yapı
- Donanım bağımsız
- Diğer platformlara taşınabilirlik
- Hata ayıklamada sağlanan kolaylık
- İşlem ve program yürütme hızı

Yukarıda ata dil olma ve geniş kullanıcı kitlesine sahip olma gibi özelliklerinden dolayı yer vermeye çalıştığımız üçüncü nesil programlama dillerinin birer *derleyici* dil olduklarını da ekleyelim. Bu dillerde, kaynak kodlar arayüzler üzerinden yazılmaktadır ve bu kodlar kendilerine ait *derleyiciler* aracılığıyla amaç koda dönüştürülmekte, daha sonra da bir *bağlayıcı* aracılığıyla çalıştırabilir veya yürütülebilir (İng. executable) hale gelerek, bilgisayarın anladığı dil olan makine diline dönüştürülmektedir (Kayabaş, 2020: 208). *Bağlayıcılar*, derlenmiş bir amaç programı diğer yazılım kütüphaneleri ve alt programlarla bağlayarak işletim sistemlerinin çalıştırabileceği programlar haline dönüştürürler (Kayabaş, 2020: 208). Günümüzde yaygın kullanıma sahip Windows işletim sisteminde, C programlama diliyle geliştirilen bir programın nasıl çalıştırıldığını gösteren bir temsil aşağıdaki gibidir.



Şekil 3.4 C programlama diliyle yazılmış bir programın Windows işletim sisteminde çalıştırılma süreci

Kaynak: Kayabaş, 2020: 208

Yukarıdaki temsilden görüleceği üzere, Windows işletim sisteminde C programlama diliyle yazılmış kaynak kodlar için arayüz kullanılarak “çalıştır” komutu verildikten sonra sırasıyla “.c uzantılı kaynak dosyalar öncelikle *derleyici* tarafından yeniden yüklenebilen amaç koda (.obj), ardından da *bağlayıcı* tarafından çalıştırılabilir programa (.exe)” (Kayabaş, 2020: 208) dönüşerek süreç tamamlanır.

Dördüncü nesil programlama dilleriyle aynı dönem tasarlanmış ve günümüzde yapay zekâ uygulamalarında da favori dillerden biri haline gelmiş olan üçüncü nesil geç dönem programlama dili olan Java, üçüncü nesil programlama dillerininin yetersiz kalacağı düşünülen durumlar için oldukça geniş bir yelpazede kullanım alanına sahip olmuştur. Tezimizin *akronik* ilerlediğimiz bu alt başlığında, 1995 yılında ortaya çıkmış Java’yı özetlemeye çalışacağız. Özellikle güçlü bir web programlama dili de olan Java, (günümüzde nesillendirmeye konu olan fakat tezimizin konusu dışında kaldığından dolayı yalnızca örnek vermek amacıyla sözünü edeceğimiz) android uygulamalarla çalışan tüm telefon ve tablet gibi dijital aletlerin yazılımında da kullanılmaktadır.

Java 1995 yılında James Gosling tarafından Sun Microsystems bünyesinde geliştirilmiş nesne yönelimli güçlü bir dildir. Java’yı diğer dillerden ayıran en önemli özellik, aynı kaynak kodun ek işlem gerektirmeden herhangi bir işletim sisteminde çalıştırılabilmesidir. Masaüstü, mobil ve web uygulamaları geliştirmek için kullanılabilen çok yönlü ve popüler bir dildir. Açık kaynak kodlu olduğu

için geniş bir geliştirici topluluğuna sahiptir. Java uygulamaları direkt olarak işletim sistemi üzerinde değil, işletim sistemine ayrıca kurulan Java Virtual Machine (JVM) üzerinde çalışır. Bu sayede donanımdan bağımsız olarak JVM kurulu her bilgisayarda kullanılabilir. Java'nın kullanım alanları farklı çalışma çerçeveleri (framework) bağlamında genişletilebilir. Örneğin; JSP (Java Server Pages) ve Spring çalışma çerçeveleri Java'yı güçlü bir web programlama diline dönüştürür. Ayrıca tüm Android uygulamaları Java ile yazılabilmektedir. Netbeans, Eclipse ve IntelliJ gibi gelişmiş tümleşik geliştirme ortamları (IDE) tarafından desteklenmektedir (Kayabaş, 2020: 211).

Bir JVM (Java Virtual Machine) aracılığıyla donanımdan bağımsız hale gelen Java, bu yolla yorumlanabilir bir dil olma özelliği de kazanmış olur. Java programlama dilinin, diğer derlenen programlama dillerine göre oldukça kısa satırlı programlama yapabilme becerisi vardır. Bu özelliğinden ve yukarıda bahsi geçen diğer özelliklerinden dolayı Java'nın da tıpkı C programlama dili ailesi gibi yaşamaya devam edeceği rahatlıkla iddia edilebilir.

3.2.4. Dördüncü Nesil Programlama Dilleri, Dördüncü Nesil Bilgisayarlar ve İnternet

Aralarında evrensellik iddiası içinde genel amaçlı kullanıma uygun olan programlama dillerinin de bulunduğu üçüncü nesil programlama dilleri, kendi içinde kırımlar yaşayarak nesne yönelimli veya alt programlara parçalanabilen yeni nesilleriyle hayatta kalmayı başarmış olsa da bu dillerin de gelişen bilgisayar teknolojileri karşısında büyük eksikleri olduğu söylenebilir. Dördüncü nesil bilgisayar çağı olarak da bilinen 1972'li yıllar mikro bilgisayarların evlere girmeye başladığı dönemdir ve ilk dönem bilgisayarların oda büyüklüğünde alana sığdırılmayan “merkezi işlem birimleri”nin⁶⁵ (MİB [İng. Central Process Unit ya da kısaca CPU]) avuç içine sığacak boyutlara küçülmesiyle oldukça bağlantılıdır. Makine diliyle yazılan komutlarla çalışan merkezi işlem birimi, verilerin, aritmetik ve mantıksal olarak işlendiği birim olarak da tanımlanabilir.

1972'den sonra inşa edilen bilgisayarlarda genellikle bir yonga üzerindeki 500 veya daha fazla bileşen içeren tümleşik devreler kullanılmaya başlanmıştır ve yüksek ölçekli tümleştirme (LSI) kullanılan bu cihazlara “dördüncü nesil” bilgisayarlar denir. Beş yıl sonra tipik olarak 10.000 bileşen bulunan çok

⁶⁵ Dördüncü nesil bilgisayar dönemi için başlangıç nedeni sayılan mikroişlemciler, 1970'lerde Intel firmasında, tümleşik devrelere bir bilgisayarın merkezi işlem birimine ait tüm devrelerin yerleştirilebilmesi (Mutlu, 2020: 26) sonucunda ortaya çıkmıştır. Günümüzde mikroişlemci ve MİB aynı anlamda kullanılacak ölçütte yakınlaşmıştır. Bir merkezi işlem birimi için doğaya uygun bir benzetme yapmak gerekirse: “Anakart, insan vücudunun iskeleti gibi donanımları üzerinde taşıırken merkezi işlem birimi (central processing unit/CPU) ise insan vücudundaki beyine benzetilebilir” (Altınpulluk, 2020: 41). Teknik bir ifadeyle dile getirmek gerekirse: “Merkezi işlem birimi; aritmetik, mantıksal işlemleri ve karar mekanizmalarıyla ilgili süreçleri 'aritmetik mantık birimi' (İng. arithmetic logic unit) aracılığıyla gerçekleştirmektedir. Aritmetik mantık birimi, işlemcinin içindeki yazmaçlardan veri yükleyerek aritmetik ve mantıksal işlemleri gerçekleştiren dijital bir devredir” (Altınpulluk, 2020: 42).

büyük ölçekli entegrasyon (VLSI) ile üretilen entegre devreler ortaya çıkmıştır. (...) Günümüzde bir mikroişlemcide milyarlarca bileşen bulunmaktadır (Mutlu, 2020: 16).

Bu yıllarda (1970'ler) bilişim teknolojisinde yaşanan gelişmeler, günümüz bilgi teknolojilerini de kapsayacak şekildeydi. İlk defa bir ARPA projesi olarak, paket anahtarlamalı ağların geliştirilerek bilgisayarlar arasında güvenilir bir ağ oluşturulabildiği 1969 yılından sonra, günümüz internet temelini de oluşturan ağ protokolleri 1972'li yıllarda geliştirilmişti. Bir diğer büyük gelişme ise 1971'li yıllar itibariyle, günümüzde milyarlarca insan tarafından kullanılan e-posta uygulamasının hayata geçirilmiş olmasıydı. Bilgisayar programcısı olan Raymond Samuel Tomlinson (1941-2016), hizmete sunduğu e-posta adreslemelerinde “@” işaretini teknoloji dünyasına kazandıran kişiydi (Mutlu, 2020: 16). İnternet ağ protokolleri, e-posta gibi hizmetlerin yanı sıra 1980'ler ve sonrasında, bilgisayar çevre birimlerinin de disket, CD (İng. compact disk) gibi yeni eklemelerle genişletilmiş olması, bilgisayarların iş alanından konfor alanı olan evlere geçişini hızlandırmıştı. Bu makineler üzerinde ses ve görüntü tamamen dijitalleştirilebiliyor ve dahi oyun bile oynanabiliyordu.

1980'lerde başlayan dijital yakınsama (İng. digital convergence) sürecinde, metin, ses ve görüntü bilgisayarlar tarafından dijital ortamda işlenmeye başlamıştır. Buna paralel olarak telekomünikasyon sistemlerinin de dijitalleşmesi ile birlikte dijital verilerin ve enformasyonun iletimi olanaklı hale gelmiş; bilgi teknolojileri ile iletişim teknolojileri iç içe geçmiştir (Mutlu, 2020: 17).

Bununla birlikte, hacimsel küçülme oranı ile ters orantılı şekilde girdiği mekan (ev, okul, büro, iş, askeri, endüstriyel vb.) sayısında gözlemlenen artış, bu makinelerin interaktif şekilde kullanılabilmesi için küçük ölçekte kalmış e-posta veya internet protokolü gibi büyük başarıların dışarı taşmasını gerektirmişti. Artık kişisel bilgisayarların konu edildiği ciddi bir üretim ve tüketim pazarı söz konusuydu. “O dönemde henüz sadece ordu, üniversiteler ve araştırma merkezlerini birbirine bağlayan bir ağ olan internette kullanılan eski ağ mimarisi, 1983'de terk edilerek TCP/IP protokollerinin kullanımına (Mutlu, 2020: 17) geçilmişti. Tüm bu teknolojiler ciddi paradigma değişimlerini oldukça hızlı bir şekilde gerçekleştiriyordu.

Dijital yakınsama, dijital dünyalar getirmişti. Bilgi, hiç olmadığı kadar kolay erişilebilecek dijital bir dökümandan başka bir şey değildi. Böylece, dijital çağ ve bilgi çağı birlikte enformasyon çağının hızlandırıcı kısa dönem faktörleri gibi olmuştu. Teknoloji, devrimlerini kısa süreler içinde bir yenisini doğuracak şekilde gerçekleştiriyordu. Hız ve zamandan tasarruf, esneklik ve bilgiye kolay erişilebilirlik gibi talepler, hızlanmış teknolojinin gerektirdiği unsurlardı. Tüm bu gereksinimler sonucunda, coğrafya-bağımsız bir şekilde farklı

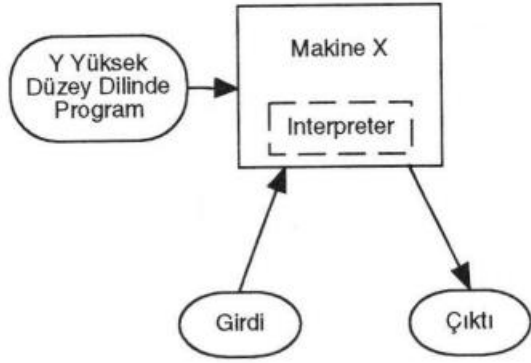
bilgisayar ve sistem türlerindeki belgeleri ve bilgileri sorunsuz bir şekilde erişilebilir kılmak için (Mutlu, 2020: 17) Cern laboratuvarlarında görevli olan Tim Berners-Lee (1955-...) 1989 yılında HTML (“hiper metin işaretleme dili” anlamında Hypertext Markup Language) *işaretleme dilini* geliştirerek, bilgi paylaşım sistemi olarak tanımlanabilecek *dünya çapında ağı* (İng. World Wide Web, kısaca www) icat etmişti.

Dördüncü nesil bilgisayarların gelişim süreci ve internet tabanlı bilgi paylaşım sistemlerinin icat edilmesi sonucunda programlama dillerinde yeniden bir paradigma değişiminin ortaya çıktığını vurgulamakta yarar var. Dördüncü nesil programlama dilleri, bu bağlamda değerlendirildiğinde çoğunlukla *web* tabanlı ve veri tabanına (İng. database) özel diller olarak bilinmektedirler. Bir örnek vermek gerekirse SQL (Structured Query Language [yapılandırılmış sorgu dili]), “1980’lerde veri tabanı belirtimi, rapor oluşturma ve bilgi alımı için bildirimsel bir programlama aracı olarak ortaya çıkmıştır” (Tucker ve Noonan, 2007: 10). İlişkisel cebirdeki kısmen güçlü matematiksel temelleri nedeniyle yaygın olarak kullanılan SQL, ilişkisel veri tabanlarından bilgi belirtmek ve almak için kullanılan baskın dildir (Tucker ve Noonan, 2007: 10).

Üçüncü nesil programlama dillerinden (derlenmiş diller) farklı olarak, dördüncü nesil programlama dilleri bilgisayara “nasıl yapılması gerektiğini” anlatan prosedürel bir yapı sergilemezler. Bilakis, bu dillerin paradigması nesne yönelimlidir, prosedürel olmayan ve bilgisayara “ne yapacağını” bildiren kodlar oluşturulmasını sağlar. Örneğin, JavaScript ve Smalltalk gibi programlama dilleri dördüncü nesil programlama dillerinden bazılarıdır.

1995 yılında Brendan Eich tarafından geliştirilen JavaScript, ilk olarak Netscape Navigator web tarayıcısı kapsamında kullanıma sunulmuştur. Genellikle JS olarak kısaltılan JavaScript, HTML ve CSS ile birlikte web uygulamaları geliştirmek için tasarlanmış nesne tabanlı bir programlama dilidir. Prototip tabanlı ve çoklu paradigma içeren bir dil olması nedeniyle nesne yönelimi ve işlevsel gibi farklı programlama yaklaşımlarını destekler. İsim benzerliği dışında Java programlama dili ile bir bağlantısı yoktur (Kayabaş, 2020: 212).

Dördüncü nesil programlama dillerinin bir diğer önemli özellikleri ise İngilizce benzeri doğal dil yapısı içeriyor olmalarıdır ve insanın doğal dilinde programlamayı desteklemeleridir. Dolayısıyla, dördüncü nesil programlama dilleri, “daha az programlama becerisine sahip kullanıcılar için programlamayı daha kolay, daha verimli ve daha etkili hale getirmeyi (Alzahrani, 2020: 178) amaçlamaktadır. İnsan kullanımına yakın doğal dil görünümünde olan dördüncü nesil diller, eş zamanlı olarak bir yorumlayıcı (İng. interpreter) tarafından yorumlanır. Aşağıdaki şekilde, bir yorumlayıcı ara biriminin çalışma prensibi görülmektedir.



Şekil 3.5 *Interpreter* (yorumlayıcı) arabirimi ve çalıştırılma süreci

Kaynak: Eldeniz, 1994: 142

Bir tarayıcı üzerinden çevrimiçi etkileşim olanağı sunabilen dördüncü nesil programlama dilleri ancak ve ancak tarayıcının sınırları bağlamında evrensellik iddiasında bulunabilirler. Kaldı ki bu dillerin üzerinde çalıştıkları tarayıcının ölçütlerine göre, bu dillerle yazılmış programlar hataya açık olabilmektedir. Bir diğer husus olarak bu diller, şu anlamda kusurlu sayılabilir: Bir *yorumlayıcı* tarafından yorumlanan bu diller, makine diline dönüşme sürecinde oldukça yavaşlamaktadır.

Dördüncü nesil diller, nasıl yapılması gerektiğinden çok ne yapılması gerektiğini belirtir. Programlama çabasını azaltmak için tasarlanmıştır ve rapor oluşturucuları ve form oluşturucuları içerir. Rapor oluşturucular, veri formatının ve oluşturulacak raporun bir tanımını alır ve ardından raporu üretmek için otomatik olarak bir program oluşturur. Form oluşturucular, uygulama sistemi kullanıcıları ile çevrimiçi etkileşimleri yönetecek programlar oluşturmak için kullanılır. Ancak, dördüncü seviye diller derlenmiş dillerle karşılaştırıldığında yavaştır (O'Regan, 2016: 191).

Dördüncü nesil programlama dilleri, amaç bir iş için tasarlanmış diller olup, “bu dillerin başarısının temel nedeni, sınırlı bir dizi sorunu çözmek için kullanılmaları gerçeğinde yatmaktadır” (Casimir, 1989: 83). Dolayısıyla, bu programlama dillerinin genel amaçlı kullanım için uygun oldukları söylenemediği gibi evrensellikleri de iddia edilemez.

3.2.5. Beşinci Nesil Programlama Dilleri

Beşinci nesil programlama dilleri, yukarıda serimlemeye çalıştığımız diğer tüm programlama dillerinden farklı bir yerde durmaktadır. Yordamsal olan bu programlama dillerinin paradigması mantıklarına ve işlevselliklerine dayanır ancak başka önemli özellikleri de sıralanmaya ve açıklamaya değerlidir. Beşinci nesil programlama dilleri, “nasıl yapmalı” veya “ne yapmalı” tarzında sorularla programcının ilgilendiği ve bunun için algoritma ürettiği

bir şekilde işlemez, bilakis, burada işi yapan programcı değil bilgisayarın ya da makinenin kendisidir.

Beşinci nesil bir programlama dili veya 5NPd, programcı tarafından yazılmış bir algoritma kullanmak yerine, programa uygulanan kısıtlamaları kullanarak problemleri çözmeye dayanan bir programlama dilidir. Beşinci nesil diller, (programcının değil) bilgisayarın sorunu çözmelerini sağlamak için tasarlanmıştır. Programcı, problemi ve yerine getirilmesi gereken kısıtlamaları belirler ve algoritma veya uygulama detayları ile ilgilenmez. Bu diller ağırlıklı olarak özellikle yapay zekâ alanında araştırma amaçlı kullanılmaktadır (O'Regan, 2016: 191).

Oldukça yakın bir geçmişe dayanan sürümleriyle birlikte değerlendirildiğinde, bu programlama dilleri, sayılar yerine semboller kullanır çünkü özünde yapay zekâ çalışmalarında kullanılmak üzere tasarlanmışlardır. Bunlar arasında sayabileceğimiz Miranda, soyutlama becerisi yüksek Haskell, ML, Scheme gibi yapay zekâyâ yönelik programlama dilleri, kökenlerini ataları LISP'ten (**L**ist **P**rocessor) alan işlevsel dillerdir. LISP'in tasarlandığı ilk günden itibaren farklı bir paradigmayla birçok yeni lehçesi ve kendi neslinden devam eden sürümleri yapay zekâ uygulamalarında kullanılmıştır.

Neredeyse tüm diğer programlama dillerinin atası sayılabilecek FORTRAN ile aynı dönemlerde yaratılmış olan Amerikan laboratuvarları kökenli LISP, 1958 yılında, ALGOL58'in de yaratıcılarından biri olan McCarthy⁶⁶ tarafından, bilgisayar dostu olduğu düşünülen mantık yani Frege'nin birinci seviye yüklem mantığı kullanılarak ve Church'un λ -kalkülüsünden esinlenilerek tasarlanmıştır. "Lambda hesabı, dönüşüm kurallarını kullanır ve bu kurallardan biri de değişken ikamesidir" (O'Regan, 2016: 203). Church-Turing tezinden de hatırlayacağımız üzere her ne kadar iki keyfi lambda hesabı ifadesinin eşdeğer olup olmadığını belirlemek için genel bir algoritma yoksa da herhangi bir hesaplanabilir fonksiyon, lambda hesabı kullanılarak ifade edilebilir ve değerlendirilebilir (O'Regan, 2016: 203).

Beşinci nesil programlama dillerinde geniş bir uygulama alanına sahip *lambda* ifadeleri, parametrelerin ifadeden sonra yerleştirilmesiyle uygulanır. Basit bir örnek ile

⁶⁶ John McCarthy, 1958 yılında Amerikan laboratuvarlarında yapay zekâ için yazılmış ilk çok yüksek seviyeli programlama dili olan LISP'i tasarlarırken, diğer yandan evrensellik iddiasıyla ortaya çıkmış ilk geniş algoritmik yapıya sahip yüksek seviyeli programlama dili olan ALGOL58'in de tasarlanma aşamasında etkin bir rol oynamıştır. Kariyeri başarılarla dolu olan McCarthy, önem sırasınca ilkleri hak eden bu iki yüksek seviye programlama dili üzerine teoriler geliştirmiştir. "McCarthy, matematiksel hesaplama teorisine ve programlama dillerinin biçimsel semantiğine de öncü katkılarda bulundu. 1960'da LISP'in hem işlemsel anlambilimini hem de özyineleme teorik analizini sundu ve evrenselliğini, daha doğrusu uygulama işlevinin evrenselliğini kanıtladı. 1966'da benzer teknikleri bir ALGOL parçasının analizine uyguladı. Bu çalışma, özyinelemeli programların özelliklerini kanıtlamak için bir dizi önemli fikir, özellikle de program denkleğini bir özyineleme tümevarım kuralı yoluyla kanıtlamak için sabit nokta özelliklerinin kullanımını tanıttı" (Israel, 1991: 3).

açıklamak gerekirse, $(\lambda(x) x * x * x)(2)$ ifadesi 8 sayısı olarak değerlendirilir (Sebesta, 2016: 661).

Birer fonksiyonel programlama dili olan beşinci nesil dillerin tasarımının amacı, aslında matematiksel fonksiyonları mümkün olan en geniş ölçüde taklit etmektir (Uzun, 2022b: 9). Oysa ki fonksiyonel programlama dilleriyle ortak özellikleri olan LISP, bir anlamda yalnızca fonksiyonel olarak değerlendirilemeyecek kadar zengin bir dildir. Öyle ki LISP işlevsel (fonksiyonel), yorumlanabilir ve prosedürel yapısıyla çoklu paradigmlar içeren bir dil olarak tanımlanmaya daha uygundur. LISP programlama dili, “öncelikle sembolik veri işleme içindir. Diferansiyel ve integral hesabı, elektrik devresi tasarımı, matematiksel mantık, oyun oynama ve diğer yapay zekâ alanlarında sembolik hesaplamalar” (Tucker ve Noonan, 2007: 362) yapmak için kullanılmaktadır. Günümüze dair genel bir örnek vermek gerekirse, LISP, kredi kartı işlemlerini kontrol etmek için kullanılmaktadır.

Amerikan kökenli LISP’in kendinden önce ortaya çıkmış IPL’den (1954) daha kolay olması ve kendinden sonra ortaya çıkmış FLP veya Pop-2 gibi diğer dillerden güçlü olması, onu yapay zekâ araştırmaları ve uygulamalarında uzun yıllar boyunca birincil dil haline getirmişti. PROLOG (**P**rogrammation en **L**ogique) ise Avrupa kökenli bir dil olarak, LISP’in yaygınlığını azaltabilecek güçte tasarlanarak Amerika dışında en sık kullanılan yapay zekâ uygulama diliydi. Robert E. Kowalski (1941-...) ile birlikte Alain Colmurer (1941-2017) ve doktora öğrencisi Philippe Roussel tarafından 1972 yılında geliştirilen PROLOG’da “programlar, düzenli bir mantıksal yargı dizisinden meydana geliyordu. Başka yapıların yanı sıra bu yargıların yazılma sırası, programın çalışmasında kilit noktaydı” (Nilsson, 2019: 206). Tıpkı LISP gibi PROLOG da bir *derleyici* ile değil, bir *yorumlayıcı* ile çalışır. LISP’ten farklı olarak PROLOG, mantık programlamaya, doğal dil işlemedeki uygulamaları incelemeye, problem çözmeye (Tucker ve Noonan, 2007: 414) ve sorun gidermeye odaklıdır. En iyi üne sahip mantık programlama dili olan PROLOG, birinci seviye yüklem mantığına dayalı olarak çalışır.

Mantıksal programlamada hesaplama, esasen mantıksal tündengelemdir ve mantık programlama dilleri birinci seviye yüklem hesabını kullanır. Başlangıç aksiyomlarından, arzu edilen bir gerçeği türetmeyi kanıtlayan teoremi kullanır. Bu ispatlar, teorik bir varoluş teoremine güvenmek yerine, kısıtlamaları karşılayan gerçek bir nesnenin üretilmesi anlamında yapıcıdır. Mantıksal programlama, nesnelere, bunlar arasındaki ilişkileri ve problem için karşılanması gereken kısıtlamaları belirtir: (i) Hesaplamaya dahil olan nesnelere kümesi. (ii) Nesnelere arasındaki ilişkiler. (iii) Belirli bir problemin kısıtlamaları (O’Regan, 2016: 207).

PROLOG programlama dili, asıl hedefi “hedef :- alt hedef₁, ..., alt hedef_n” (O’Regan, 2016: 207) şeklinde aralarında mantıksal çıkarımlar yapılacak bir formda alt hedeflere bölerek sorun çözmeye çalışır. Belirli bir hedef problemin kanıtlanması için alt hedef₁’in sırasıyla sürdürülen mantıksal çıkarımlar yoluyla alt hedef_n ile kanıtlanmaya başlaması gerekir. Bu dil, neyin nasıl olduğunu bildirmez, neyin var olduğunu bildirir. Böylece, PROLOG programlama diliyle hedef odaklı çözümlerde “her satır bir kuraldan veya bir olgudan oluşur” (O’Regan, 2016: 207) ve bu olgular veya kurallar, terimsel olarak “gerçekler” e dayanır. PROLOG dilinde yazılmış bir program parçası örneğini verdiğimiz aşağıdaki gösterimde, program parçasının bir kuralı ve iki gerçeği vardır (aşağıdaki örnek [O’Regan, 2016: 207]’den alınmıştır):

büyükanne (G, S) :- ebeveyn (P, S), anne(G, P).
anne (sarah, isaac).
ebeveyn (isaac, jacob).

Yukarıdaki örnekte, mantıkta kullanılan Horn⁶⁷ yan tümcelerinin biçimsel fikrinin gerçekleşmeleri görülmektedir. PROLOG programlama dilinde olgu, arkasından nokta (.) gelen bir terimdir ve sağ tarafı olmayan bir Horn cümlesine benzer. Kural ise arkasından “:-“ ve virgülle ayrılmış ve nokta (.) ile biten bir terim dizisi (Tucker ve Noonan, 2007: 417) olarak tanımlanır. Yukarıda bir örnek olarak verilen bu program parçasındaki “ilk satır, S'nin bir ebeveyni varsa G'nin S'nin büyükannesi olduğunu ve G'nin P'nin annesi olduğunu belirten bir kuraldır. Sonraki iki ifade, Isaac'ın Jacob'un ebeveyni ve Sarah'ın, Isaac'ın annesi olduğunu belirten gerçeklerdir” (O’Regan, 2016: 208).

Pür mantığa dayalı işlevsel programlama dillerinden olan beşinci nesil programlama dillerinin, diğer paradigmaları yürüten programlama dillerine görece saflığı nedeniyle, yazılım için iyi bir aday olduğunu düşünenler var ise de bu diller de kendi dar bağlamlarındaki başarıyla kısıtlı durumdadır.

[i]şlevsel programlama, bazıları tarafından yazılım tasarımı için temel (veya zorunlu) programlamadan daha güvenilir bir paradigma olarak görülüyor. Bununla birlikte, işlevsel programlamanın kullanıldığı (yapay zekâ) uygulamaların büyük çoğunluğu, temel (veya zorunlu) paradigmadaki çözümlere kolayca

⁶⁷ PROLOG’un, Horn tümceleriyle sağlaması kolaylıkla yapılabilecek mantık zinciri oluşturan tasarımı sayesinde, başarılı bir mantık programlama dili olarak geldiği noktadan bakan yapay zekâ uzmanları, bu dilin birçok yapay zekâ problemini çözmeye konusunda galip geleceğini iddia ediyor olsa da McCarthy aynı fikirde değildir. Ona göre, “Horn tümcelerinde ifade edilebilirlik, başarısızlık olarak olumsuzlama ile desteklenmiş olsun ya da olmasın, bir dizi olgunun önemli bir özelliğidir ve mantık programlama birçok uygulama için başarıyla kullanılmıştır. Bununla birlikte, bazı savunucularının umduğu gibi, yapay zekâ programlamaya hakim olması pek olası görünmüyor” (McCarthy, 1990: 9). Açıkça ifade etmek gerekirse, McCarthy’ye göre Horn tümceleriyle evrensel bir genelleme yapılabilmesi olanaklı değildir.

uygun olmadığından (ve bunun tersi de geçerlidir) dolayı bu görüşü belgelemek zordur (Tucker ve Noonan, 2007: 362).

Günümüzde, yapay zekâ uygulamalarında en hızlı büyüyen programlama dili, Python⁶⁸ tezimizin ilerleyen sayfalarında özel olarak örneklendirileceğinden dolayı özellikle nesillendirme yaparken değinmemiş olmamıza karşın yine de günümüzün en başarılı ve en çok tercih edilen işlevsel programlama dili olduğunu eklemekte yarar görmekteyiz.

Tezimizin ana sorgulaması olan evrensel dil bağlamında bir genelleme doğrultusunda olmasına gayret ettiğimiz bu bölümde, bir soybilimi yapmaya çalışarak şekil 3.6'da bir genetik şemasının verildiği programlama dilleri arasındaki bağlantıları, paradigmalarını ve kullanım alanlarını serimlemeye çalıştık. Bundan sonraki bölümlerde ise bu dillerin evrensellikleri, iddialar ve beklentiler çerçevesinde sorgulanmaya çalışılacaktır.

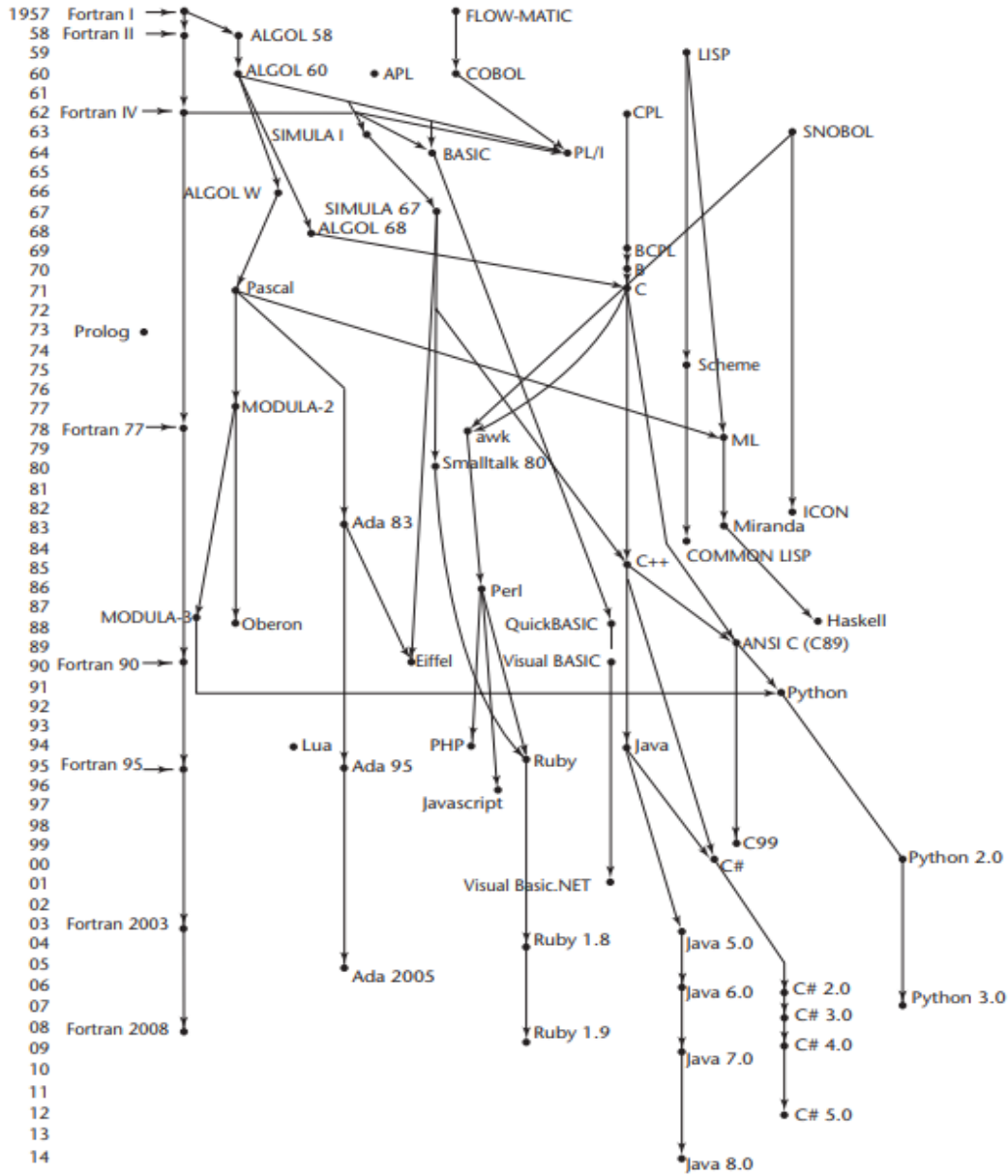
3.3. Programlama Dillerinde Evrensellik Sorgulaması

Yukarıda yer verme çabasında olduğumuz üçüncü, dördüncü ve beşinci nesil tüm programlama dillerini kökensel olarak geriye götürdüğümüzde karşımıza öncelikle FORTRAN, COBOL, ALGOL, LISP ve PROLOG programlama dilleri, ata diller olarak çıkmaktadır. Bu diller genel bir çerçevede değerlendirildiğinde ise çeşitli paradigmlar eşliğinde özetlenebilecekleri rahatlıkla görülebilir. Örneğin, üçüncü nesil programlama dilleri arasında bulunan FORTRAN, COBOL ve ALGOL programlama dillerini, paradigma olarak bilgisayara “nasıl yapılacağını” aktaran ve bir *derleyici* ile çalışan prosedürel temel (zorunlu) diller olarak sınıflandırabiliriz. Dördüncü nesil dilleri, paradigma olarak bilgisayara “ne yapacağını” bildiren ve bir *yorumlayıcı* ile çalışan nesne yönelimli ve prosedürel-olmayan diller olarak sınıflandırabiliriz. Beşinci nesil dilleri ise kendini yaşatan paradigması prosedürel-olmayan mantık dili PROLOG ve kendini lehçelendirerek nesillendiren çok paradigmali işlevsel, nesneye yönelik, prosedürel LISP programlama dilleri olarak özetleyebiliriz. Beşinci nesil dillerin ortak özelliği ise bilgisayar ve programcı arasındaki etkileşimin ontolojik bağlamıdır. Bu nesilde, programcı algoritma üretmek veya program yazmak zorunda değildir çünkü bunu bilgisayar, programcının yerine yapmaktadır. Günümüzde beşinci nesil bilgisayarlar⁶⁹ üzerine 1981 yılından bu yana yapılan çalışmaların,

⁶⁸ Ataları ALGOL ve FORTRAN olan Python programlama dili, ayrıca CPL, C ailesi ve BASIC ailesi ile de bağlantılıdır. Yapay zekâ uygulamalarında en çok kullanılan dil olan Python, düşük ve yüksek seviyeli programlama dengesini iyi kurmuştur. Oldukça uzun olabilen kodlama satırlarının, otuz ya da kırk satırlara indirgenebildiği Python, “derin öğrenme” alanlarını da kapsayacak şekilde Youtube, Reddit, Dropbox gibi web sitelerinin geliştirilmesinde kullanılan programlama dilidir.

⁶⁹ Beşinci nesil bilgisayar çağının en olumlu etkisi, yapay zekâ uygulama alanları üzerine olmuştur. Özellikle tezimizin ilerleyen bölümlerinde yer vereceğimiz *yapay öğrenme* alanında ciddi atılımların gerçekleşmesini sağlamıştır. “Japonya Uluslararası Ticaret ve Sanayi Bakanlığı’nın 1981’de insanlarla konuşabilecek, resimleri yorumlayabilecek ve çeviri yapabilecek programlar üretilmesini hedefleyen Beşinci Nesil Bilgisayar Projesi’ne

özellikle beşinci nesil programlama dilleri bağlamında olduğunu yani programlama ve algoritma prosedürlerinin bilgisayara teslim edildiği bir sistem üzerinde yoğunlaştığını eklemekte yarar var. Aşağıdaki şekilde, programlama dilleri arasında yaptığımız nesillendirme ve köken ilişkisi daha ayrıntılı olarak görülmektedir.



Şekil 3.6 Programlama dillerinin birbirleriyle olan ilişkisi

Kaynak: Sebesta, 2016: 59

Programlama dillerinin ortaya çıktıkları (ilk laboratuvarların İngilizce kökenli olması bir sebep olarak görülebilir) ilk andan itibaren sözdizimlerindeki tüm anahtar kelimelerin

850 Milyon ABD doları kaynak ayırdığını açıklamasıyla” (Say, 2019: 89) “yapay zekâ kışı” olarak da bilinen süreç tersi yönde gelişmelerle yeniden hareketlenerek sonlanmıştır.

İngilizce kökenli olduklarını belirtmekte yarar var. Bu kelimelerden bazıları, koşul (*if / then / else*), döngü (*while / for*), blok, altyordam (*procedure*) ve işlev (*function*) gibi yapısal ifadelerle, programcayı yönlendirmektedir. Her ne kadar herhangi bir dilin konuşucusu tarafından anadilinde kelimelerle istenilen şey belirtilebiliyor olsa da programlama dilinin özelinde bulunan matematiksel işaretler, belirteçler ve döngü ya da iterasyon bildiren durumların ortak görünümünü veren sözdizimindeki anahtar kelimeler, bir önceki bölümde örneklerini vermeye çalıştığımız üzere İngilizcedir. Böyle bir durumda, ata programlama dilleri üzerinden yapılacak bir genelleme doğrultusunda denilebilir ki bu diller Hint-Avrupa dil ailesinden köken alan laboratuvar dilleridir.

Birer laboratuvar dili olan tüm programlama dillerinin ataları ve dolayısıyla devamları olan nesilleri, öncelikle bilgisayar dostu olarak kabul edilen Frege'nin birinci seviye yüklem mantığından izler taşırlar yani mutlaka iterasyon belirteci olarak *if...then* yapıları mevcuttur. Bu diller, 1900'lü yıllarda matematiğin ve sembolik mantığın birer dil olarak geldiği en biçimsel halinin tüm izlerini, kullandıkları alan(lar)a göre taşırlar. Bu diller ister bir *derleyici* ister bir *yorumlayıcı* aracılığıyla makine diline dönüştürülüyor olsun, bu makine dilinin Boole değerleri alan 0 ve 1'den başka bir şey anlamayan Boole cebirine ait basit aritmetik kurulumlara ve mantığa bağımlı olduğunu hatırlatmakta yarar var. Öyle ise bir programlama dili ne kadar kullanıcı dostu olarak görünse de temeli makine olan bir zeminde anlaşılabilir olmak için makinenin diline dönüştürülmek zorundadır. Dolayısıyla, yüksek seviyeli programlama dilleri, insan ile etkileşimde biçimsellik kazandığı matematiksel notasyonlar, özel işaretlemeler veya grafikler kullanırken, makine ile etkileşimde tamamen semboller kullanarak sonunda sadece 0 ve 1'ler olarak algılanmaktadır. Bu nedenle yüksek seviyeli dillerle yazılmış, altlarında yatan çeşitli mantıklarla çalışan (örneğin, birçok uygulama dışında ayrıca yapay zekâ uygulamalarında göze çarpan *bulanık mantık*⁷⁰ veya SQL dilinde kullanılan *üç değerli mantık*⁷¹ gibi) programların makine diline dönüşen komutlarıyla aralarında semantik sorunlar oluşması her zaman ihtimal dahilindedir.

⁷⁰ *Bulanık mantık*, önmeler mantığı veya Boole cebirinde olduğu gibi sadece 0 ve 1 değerlerini almaz. Bu tarz bir mantık, özellikle kesin olmayan veya bozuk bilgi yığınları üzerinde çalışılırken kullanılmaktadır. "Bulanık mantık, sonsuz sayıda doğruluk değeri önerir; bunlar 0.00 (tamamen yanlış) ile 1.00 (tamamen doğru) arasındaki gerçel sayılarla temsil edilebilir" (Gensler, 2017: 487).

⁷¹ *Üç değerli mantık*, iki değerli mantıktan farklı olarak "0", "1" ve "boş" (NULL) değerlerini alan ve ikili sayı tabanı yerine üç sayı tabanı ile doğruluk değeri belirlenimi yapılmasını sağlayan mantık çeşididir. Örneğin, SQL (yapılandırılmış sorgu dili) bu mantıkla çalışan bir dildir. Burada NULL, veriye ait değer inceleme kousu olan veri tabanında bulunmadığına dair bir belirteç anlamında kullanılmaktadır. *Bulanık mantıkta* olduğu gibi *üç değerli mantıkta* da hem programcı hem de makine bağlamında semantik sorunlar baş gösterebilir.

Yapılandırılmış sorgu dilinde (SQL), bilgisayar bilimlerinin alışık olduğu iki değerli mantığa dayanan düşünme tarzından farklı bir düşünme tarzı gerektirmektedir. İlişkisel veri tabanı yönetim sistemleri (RDMS) üç değerli mantığın en yaygın uygulama alanı konumundadır. Veri tabanı tasarımcıları, yazılım geliştiriciler ve kullanıcılar üç değerli mantığa dayalı bu düşünme tarzını öğrenmelidir. Aksi durumda hem mantıksal hem de semantik hatalardan kaçınmak zorlaşmış olacaktır (Kamer, 2015: 46).

Günümüz yapay zekâ çalışmalarından aşına olunduğu üzere, programlamacılık, istenen sonucun verildiği ve bilgisayarın bu sonuca ulaşabilecek algoritmalar üretebildiği seviyelere ulaşmış durumdadır. Büyük kolaylık ve rahatlığın yanı sıra doğal dile oldukça yakın seviyelere taşınmış bir programlama türüyle karşı karşıyayız. Bu durum insan için konfor belirteci olsa da tüm üst seviyeden mantıklarla ve yüksek seviye dillerle yazılmış bu programların, makine diline dönüşmesi durumu bir zorunluluk olarak devam etmektedir. Böylece, mantığa dayalı semantik sorunları tamamen ortadan kaldıramış dördüncü veya beşinci nesil bir programlama dili olduğunu söylemek zordur.

Günümüze değin hangi sayılara ulaşmış olduklarına dair dayanak noktası olabilecek bir kaynağa ulaşamadığımızdan dolayı, programlama dillerinin yaşamaya devam eden nesilleri göz önünde bulundurulduğunda, bugün bu dillerin sayılarının iki yüz elli veya daha fazla olduğu varsayımı akla yatkın görünmektedir. Çoğunun neslini sürdüremediği, kalanların ise birçok lehçeyle zamanın teknolojik ihtiyaçlarına ayak uydurma çabasında olduğunu da ekleyebiliriz.

Bu dillerin bazılarının, örneğin ALGOL neslinden C ailesinin, atadan gelen özellikleri doğrultusunda ve genel kullanıma uygunluğu ölçütünde evrensel dil olmaya aday görüldüğü söylenebilir. Bunun yanı sıra özellikle ALGOL60 gibi bir programlama dilinin tasarımının, akademik ve rasyonel bağlamda özellikle sentaktik kurulumunun, günümüz modern programlama dillerinde yaşamaya devam ettiği bilinmektedir.

Bu bölümde özele indirgeme yapma gereksiniminin ötesinde, Leibniz bağlamında bir evrensel dil sorgulaması içinde kalarak, makine temelinde tüm programlama dillerini, bir *lingua characteristic* veya *characteristica universalis* çerçevesinde ele almaya çalışacağız. Bu bölümde soracağımız soru şudur: Programlama dilleri evrensel diller midir? Sorduğumuz sorunun cevabını alabilmek için biçimsellik kazandırılmış bu yapay dillerin öncelikle hangi dil yapısı grubunda kategorize edilebileceğine karar vermemiz gerekiyor.

3.3.1. Programlama Dilleri *A Priori Felsefi* Diller midir?

Nesillerine ve paradigmalarına göre ayırdığımız tüm programlama dilleri, içinden *Assembly* ve makine dillerini (makine dilleri, *calculus ratiocinator* bağlamında sembolik

mantığa dayanır) ayırmak suretiyle, üçüncü, dördüncü ve beşinci nesil diller için bir genelleme yapmamız gerekirse, rahatlıkla denilebilir ki tamamı kesin bir sözdizimine sahiptir. Dolayısıyla oldukça biçimseldirler. Bu dillerin tümü, birinci nesil (makine dilleri) ve ikinci nesil (*Assembly* diller) programlama dilleri de dahil olmak üzere, altlarında yatan mantık mimarisi veya mantık çerçevesinin belirlediği sınırlar içinde kalmak zorundadırlar. Tıpkı matematik dilinin verdiği esneklikte olduğu gibi bu diller de coğrafya-tanımadırlar yani hangi dilin konuşucusu olarak bu diller üzerinden program yazdığının bir önemi yoktur. Bu durumda, bu dillerde yazmayı öğrenen kişiler arasında kullanıcıların (programcıların) doğal dillerinden bağımsız olarak bir *anlam zemini* söz konusudur.

Programlama dilleri, doğal dillerin bir kombinasyonunu içermekten ziyade, keyfi notasyonlarla belirlenmiştir. Bu nedenle bu dillerin, *a posteriori* değil *a priori* dil grubunda oldukları açıktır. Laboratuvar kökenli olan bu dillerin *Bâleybelen*, *Volapük* ya da *Esperanto* gibi *a posteriori* yani doğal bir dilin üzerinden işletilen bir dil yapısına sahip olduklarını söylemek mümkün değildir. Ancak bu diller konuşulabilir değil, yazılabilirler (pasigrafiktirler) ve matematiksel sembol manipülasyonuna dayandıklarından dolayı *ideografik* görünüm sergilemektedirler.

Programlama dilleri, kurallı bir sözdizimi içeren mantıksal yapının sınırları içerisinde kalacak şekilde tasarlanmışlardır ve bir programlama dilinin nasıl olması gerektiğine dair belirgin sınırları vardır. İnsan ve makine arasında kurulan ontolojik bağlamın açıklanma noktası olarak da görülebilecek programlama dilleri, insan ve makine bağlamında anlaşılabilme ölçütünü azami derecede vermek üzere tasarlanmış yapıdadır. Tıpkı matematik dilinin ulaştığı son noktadan bile bir *rasyonel dil* olma sınırını aşmadığı ölçüde, matematiğin temelleri üzerinden yükselen programlama dilleri de birer *rasyonel dil* olarak görülebilir. Bu nedenle, mantık çerçevesiyle belirlenmiş sözdizimi sınırlarından dolayı kavramsal bir ayırıştırma yapmaya olanak verdikleri ölçüde *rasyonel* ve dolayısıyla da *felsefidirler*. Bu hususta fikrini beyan eden dilbilimci Umberto Eco'nun iddiası şu yöndedir:

Gerçekten de bilgisayarlarla konuştuğumuz diller, *a priori* dillerdir; *Basic*'in ya da *Pascal*'ın sözdizimini düşünün. Bunlar, basit ancak kesin bir sözdizimi sundukları için dil konumuna ulaşmayan sistemler olup, boş simgelerine ya da bağlantısız değişkenlerine atfedilen anlamlar açısından öteki dillerin asalakları olarak kalırlar ve büyük ölçüde *if...then* gibi mantıksal bağlayıcılardan oluşurlar. Gene de değişik dillerin konuşucularınca eşit derecede anlaşılabilir olan *evrensel sistemlerdir* ve hatalara ya da çift anlamlılığa izin vermemeleri anlamında *kusursuzdurlar*. Doğal dillerin yüzeysel yapısına değil, olsa bütün dillerde ortak olduğu varsayılan derin bir dilbilgisini dile getiren kurallara dayandıkları anlamında *a prioridirler*. *Felsefidirler*; çünkü mantığın kurallarına uygun olarak kurulan bu derin dilbilgisinin gerek insanlarda gerek makinelerde ortak olan bir düşüncenin dilbilgisi olduğunu

varsayarlar. *A priori felsefi* dillerin iki temel sınırı bu diller için de geçerlidir: (i) kurallarını, birçoklarına göre, temellerini Hint-Avrupa dillerinin yapısı üzerine oturtan Batı uygarlığının geliştirdiği mantık üzerine kurarlar; (ii) dile getirebilecekleri şeyler sınırlıdır ve doğal bir dilin dile getirebileceği her şeyi dile getirmeye olanak vermezler (1995b: 311 italik vurgular bana ait)⁷².

Öyle ise programlama dillerinin evrenselliklerine dair sorgulamaya başlamadan önce sorduğumuz ilk soruya verilebilecek yanıt şu şekilde özetlenebilir: Programlama dilleri, matematiksel ve mantıksal temeller üzerinden yükselen, kesin bir sözdizimine sahip, anlamda müphemliği ortadan kaldırmaya yönelik kısıtlar barındıran pasigrafik yapılarıyla, *a priori felsefi* dillerdir.

3.3.2. Programlama Dilleri Evrensel Diller midir?

Bu bölümde, evrensellik sorgulamasını yürüteceğimiz ontolojik zeminde, programlama dillerini felsefi sağduyu ile tartışmaya çalışacağız. Bu tartışmanın temelinde, sosyal zemin değil teorik bilgisayar bilimlerinde önemli bir yere sahip olan “hesaplama kuramı” bulunmaktadır. “Hesaplama Kuramı” kendi içinde üç ayrı alt başlıkta çözümlene gerektirmektedir. Bundan dolayı, ilk altbaşlığımızda öncelikle linguistik bağlamda ele alacağımız evrensellik sorgulamasını “otomata (özdevinim) teorisi ve dil” temelinden yürüteceğiz. Daha sonra da ikinci altbaşlığımızda “hesaplama kuramı” ve “karmaşıklık kuramı”nın yanı sıra bir de bu dillerin Turing-*tam* olma ölçütlerini ele alarak evrensellik sorgulamamızı sürdüreceğiz.

Programlama dilleri üzerinden geliştirilen yapay zekâ sistemleri de “hesaplama kuramı” ışığında, evrensellik sorgulaması yapılması gereken bir alan olarak karşımıza çıkmaktadır. Bu sorgulama bağımsız olarak 3.4. altbaşlığında ele alınacaktır.

3.3.2.1. Linguistik Bağlamda Evrensellik Ölçütü

Üçüncü nesilden başlamak üzere günümüzde ulaşılan son seviyede insan doğal diline oldukça yakın görünen İngilizce benzeri komut dizilerine sahip tüm programlama dilleri, özünde içerdikleri sözdizimi bakımından, farklı dillerin kullanıcıları tarafından eşit ölçüde anlaşılabilir durumdadırlar. Eco’ya göre programlama dilleri “değişik dillerin konuşucularınca eşit derecede anlaşılabilir olan evrensel sistemlerdir” (1995a: 307). Bir dilin evrensel sayılabilmesi için *kusursuzluk* ölçütü gerek şart olmasa da Eco’ya göre programlama dilleri “hatalara ya da çift anlamlılığa izin vermemeleri anlamında kusursuzdurlar” (1995a: 307).

⁷² Türkçe çeviride “önsel” olarak geçen kavram yerine İngilizce kaynaktan olduğu şekliyle “a priori” kavramını kullandık. Bkz. 9. dipnot.

Günümüzde sayıları yüzlerle ifade edilen programlama dillerinin sözdizimleri, Frege'nin mantık ve matematik arasındaki ilişkiyi temellendirirken esinlendiği Leibniz'in, evrensel dili için keşfettiği *characteristica universalis* yönünü andırmaktadır. Dilbilimci yaklaşımda *a priori felsefi* bir dile evrenselliğini veren özellik, dilin *lingua characteristica* yönüdür. Bu durumda, insanların üzerinde anlaştıkları bir evrensel dilde karakteristikler, semantik kısımdan sorumlu (Peckhaus, 2004: 7) olacaklardır ve Leibniz bağlamında bir mantıksal oranlayıcı da (Lat. *calculus ratiocinator*) “basit düşünceler listesinden tüm olası doğruları mekanik olarak çıkarmaya (Peckhaus, 2004: 7) hizmet edecektir. Öyle ise *calculus ratiocinator*, *lingua rationalis*'in anlamsal değil, sözdizimsel kısmını (Peckhaus, 2004: 7) oluşturacaktır.

Makine ontolojisinde durumun biraz daha farklı bir boyutta ilerlediğini söylemek mümkün. Buna göre, temelleri tamamen mantık mimarisıyla donatılmış bir sistemin iletişim dili olan programlama dillerinde sözdizimi, semantik ile birlikte yürüyen biçimsel bir yapı sergiler. Özellikle ALGOL programlama dilinde yakalanan başarılı algoritmik yapıdan gelen evrensel görünüm sonucunda, bu dilin sözdiziminde, sentaktik kurulumun pekinliğini sağlamak üzere çalışmalar da başlamıştır.

ALGOL'ün tasarlandığı dönemlerde, yapay zekâ çalışmaları henüz başlamıştı ki programlama dilleri ve yapay zekâ çalışmalarından bağımsız bir şekilde dilbilimci Noam Chomsky (1928-...) “Enformasyon Kuramı Sempozyumu”nda sunduğu “Dilin Tanımına Dair Üç Model” (Three Models for the Description of Language [1956]) başlıklı makalesinde, öbek-yapısı dilbilgisi (İng. *phrase-structure grammar*) olarak adlandırdığı bazı sözdizimi kurallarını tanıtmıştı. Bu kurallar gereği, bir doğal dilin sözdizimi sadeleştirilse de özellikle bir bildirim cümlesinde anlam kayması yaşanması engellenebilirdi.

Bir yanda yalnızca zorunlu dönüşümlerin uygulanmasıyla, tümce yapısı dilbilgisinin uç dizilerinden türetilen bir temel tümce çekirdeğine sahibiz. Daha sonra, çekirdek cümlelerin altında yatan dizgelere, isteğe bağlı dönüşümler uygulanarak oluşturulan bir dizi türetilmiş cümleye sahibiz. İngilizce yapısının ayrıntılı bir incelemesini gerçekten yaptığımızda, çekirdeği çok küçük basit, aktif bir bildirim cümlesi kümesiyle (aslında, muhtemelen sonlu bir küme) sınırlandırırsak, “adam yemeği yedi” cümlesinde olduğu gibi dilbilgisinin büyük ölçüde basitleştirilebileceğini görüyoruz (Chomsky, 1956: 123).

Chomsky'nin dilin dönüşüm geçirerek sadeleştirilmesini sağlamak için doğal bir dilde kullanılan cümlelerin, söz öbeklerine ayrılarak matematiksel forma yakın bir biçimde dilbilgisel basitleştirmesini önerdiği bir hiyerarşinin bulunduğu *Syntactic Structures* (Sözdizimsel Yapılar [1957]) adlı kitabı, “Dilin Tanımına Dair Üç Model” adlı makalesinin hemen ardından yayımlanmıştı. Bu kitapta Chomsky, söz öbeklerini sırasıyla en basit yapıdan

en karmaşık yapıya doğru yaptığı hiyerarşiye göre “düzenli gramer” (İng. regular grammar), “bağlamdan bağımsız gramer” (İng. context-free grammar), “bağlama duyarlı gramer” (İng. context-sensitive grammar) ve “sınırsız gramer” (İng. unrestricted grammar) bağlamında ayırmıştı.

Buna göre, Chomsky'nin dört alt kümeden oluşuyor görünümü veren gramer hiyerarşisi, özünde sözdizimleri üzerine konuşan bir üst-dildi ve bu kümeleri belirleyen tüm tipler, matematiksel bir ayrıştırmayla basitleştirilebiliyordu. “Düzenli gramer”, “bağlamdan bağımsız gramer”in altkümesi; bu ikisi, “bağlama duyarlı gramer”in altkümesi ve son olarak bu üç küme, “sınırsız gramer”in altkümeleriydi. Elbette mantık ve matematik ile yakından ilgili olan Chomsky, bu çalışmasını doğal dil İngilizce üzerine yapmıştı yani programlama dillerine özel olarak çalışmamıştı fakat dilbilim alanındaki çalışmalar, teorik bilgisayar bilimleri için bu yıllarda büyük öneme sahipti çünkü dönüştürülebilir dilbilgisi ile sadeleştirme ve buna dayalı hesaplama, programlama dilleri için aranan şeydi.

Dönüşümsel analiz düzgün bir şekilde formüle edildiğinde, esasen cümle yapısı açısından tanımlamadan daha güçlü olduğunu görüyoruz, tıpkı ikincisinin soldan sağa cümleler üreten sonlu durum Markov süreçleri açısından tanımlamadan esasen daha güçlü olması gibi. Özellikle bu tür diller, bağlamdan bağımsız kurallarla, tümce yapısı açıklama sınırlarının ötesinde dönüşümlü olarak türetilir. Dönüşümsel bir düzey eklediğimizde, dilbilgisinin etkili bir biçimde basitleştiğini gözlemlemek önemlidir. Artık yalnızca çekirdek cümleleri için doğrudan tümce yapısı sağlamak gerektiğinden- $[\Sigma, F]$ dilbilgisinin uç dizileri, yalnızca çekirdek tümcelerinin altında yatanlardır (Chomsky, [1957] 2002: 47-8).

Chomsky'nin “biçimsel dil kuramında” basitten karmaşığa doğru, tersinden bir numaralandırmasını yaptığı tiplerin, teorik bilgisayar bilimlerinde programlama dillerinin sentaksı açısından nasıl bir yerde olduğunu özetlemek gerekirse, Tip-3, düzenli gramerdir ve “sonlu özdevinim” (İng. finite automata) için kabul gören bir dönüşümdür; Tip-2, bağlamdan bağımsız gramerdir (Davis vd., 1994: 327) ve “aşağı sürüklemeli özdevinim” (İng. push-down automata) için kabul gören bir dönüşümdür. Tip-1, bağlama duyarlı gramerdir ve “doğrusal sınırlı özdevinim” (İng. linear bounded automata) için kabul gören bir dönüşümdür ve son olarak Tip-0 ise sınırsız gramerdir ve “Turing makinesi” (TM) için kabul gören bir dönüşümdür (Davis vd., 1994: 327). Ancak teorik bilgisayar bilimlerinde programlama dilleri için yalnızca Tip-2 ve Tip-3 üzerinde durulmaktadır⁷³. “Bir dil, bazı açılardan sonlu bir

⁷³ Chomsky'nin tipler ayrımını daha geniş ölçekte değerlendirmekte yarar var: Tip-3 olarak bilinen “düzenli gramer, bağlamdan bağımsız veya BNF sözdizimlerinden önemli ölçüde daha zayıftır. Örneğin iyi tanımlanmış

otomata benzer, bu nedenle bir yığının kurallarına göre çalışan bir yardımcı belleğe sahipse ve aşağı sürüklemeli özdevinim tarafından kabul edilebilirse, bağlamdan bağımsız bir dilbilgisi tarafından tam olarak oluşturulabilir” (Martin, 2011: 164). Bu tanımın, Chomsky hiyerarşisinde bizi götürdüğü tip, Tip2’dir.

Örneklendirme yapmak gerekirse, programlama dillerindeki *token* (sembol) formları Tip-3 yani düzenli (regular) gramer sınıfına girer ve düzenli gramer, karakter akışlarını belirteçlere (veya temel sembollere) dönüştürmek için (Tucker ve Noonan, 2007: 59) kullanılır. “Sembol (token), bir sözcük birim sınıfını oluşturan sözdizimsel bir kategoridir. Bunlar programlama dili için 'isimler', 'filler' ve diğer konuşma parçalarıdır” (Uzun, 2022c: 10). Bir genelleme çerçevesinde değerlendirilecek olursa programlama dillerinin sözdizimi, Tip-2’dir (Uzun, 2022c: 13) yani bağlamdan bağımsız gramer sınıfına girer.

ALGOL’ün tasarımında görev almış John Backus (1924-2007) ve Peter Naur (1928-2016), ALGOL58 için Chomsky’nin hiyerarşisini kendilerine uyarlayarak, algoritmik görünümlü bu dilin formel yapısı üzerine, Chomsky’nin “bağlamdan bağımsız gramer” ile bir eşdeğerlik kurdukları BNF (Backus-Naur Form) formunu geliştirmişlerdir. Böylece, ALGOL’ün ilk tasarım sürecinden başlamak üzere günümüze değin ALGOL neslinden olsun ya da olmasın, tüm programlama dilleri, özellikle bir metadil olarak kabul ettikleri BNF ölçütlerinde belirlenir duruma geldi. Toerik bilgisayar bilimlerinde, Chomsky’nin Tip-2 sınıfı ile BNF eşdeğer kabul edilir (Tucker ve Noonan, 2007: 26; Barak, 2021: 375) ve BNF, programlama dilleri için bir standarttır. “Bir metadil olan BNF’de sözdizimsel yapıların sınıflarını göstermek için soyutlamalar (İng. abstractions) kullanılır. Bunlar sözdizimsel değişkenler gibi davranırlar” (Uzun, 2022c: 15) ki bu tip sözdizimsel değişkenlerin, birer

bir teorem $\{a^n b^n \mid n \geq 1\}$ görünümüyle bir dildir fakat 'düzenli' bir dil değildir yani bu dil, düzenli gramer ile oluşturulamaz. (...) Düzenli gramerler, tüm sol parantezlerin tüm sağ parantezlerden önce geldiği en basit parantez dengeleme durumunu işleyemez. Bu nedenle düzenli gramer, bir programlama dilinin sözdizimini düzgün bir şekilde tanımlamak için yetersizdir. Bağlamdan bağımsız gramerlerin bir avantajı, geniş bir açık dilbilgisi sınıfı için tabloya dayalı ayrıştırıcıların olmasıdır” (Tucker ve Noonan, 2007: 59). Bunun yanı sıra bağlama duyarlı gramer ile “ $\alpha, \beta \in (N \cup T)$ olmak üzere, $\alpha \rightarrow \beta$ iken $|\alpha| \leq |\beta|$ tarzında diller üretilebilmektedir. Yani, bağlamdan bağımsız gramerden farklı olarak, sol taraftaki simge, tek bir uçbirim dışı olmakla sınırlı değildir, bunun yerine hem uçbirimli hem de uçbirimli olmayan sembollerini içeren bir dizgeden oluşabilir. (...) Bağlama duyarlı gramer, program tanımlayıcılarının bildirilmesini ve bağlama uygun türe sahip olmasını sağlayabilir” (Tucker ve Noonan, 2007: 59). Ancak, bağlama duyarlı gramerin programlama dillerinde kullanılması, çeşitli semantik sorunlara yol açabilir yani iyi bir belirlenimden uzak, muğlak sözdizimlerinin üretilmesine neden olabilir. Belirsizlik, programlama dilleri için istenmeyen bir durumdur. Bu nedenle, bir TM’yi yürütebileceği düşünülen en üst küme konumundaki sınırsız gramerin çok daha fazla sorun çıkaracağı ortadadır.

fonksiyon olarak görüldükleri açıktır. Bir BNF belirtiminde, programlama dillerinin içinde olması beklenen aşağıdaki gibi bir dizi kural mevcuttur:

$\langle \text{sembol} \rangle ::= \langle \text{sembollerle ifade} \rangle$ (O'Regan, 2016: 209).

Burada $\langle \text{sembol} \rangle$ bir uç birim-*olmayan*dır. Biçimsel bir gramerde, uç birim-*olmayan*, gramerin tümcelerinde görünmeyen ancak sonunda bir dizi uç simgeye dönüştürülebilen bir simge anlamında kullanılmaktadır. İfade ise “bir seçimi belirten dikey çubuk '|' ile ayrılmış sembol dizisinden ve/veya dizilerinden oluşur” (O'Regan, 2016: 209). Bir ifadenin sol tarafında asla görünmeyen sembollere “uç birim” denir (O'Regan, 2016: 209). Aşağıda, bir programlama dilinde kullanılan çeşitli ifadeler ve onların kısmi tanımları görülmektedir (aşağıdaki gösterimler [O'Regan, 2016: 209]'dan alınmıştır):

$\langle \text{döngü ifadesi} \rangle ::= \langle \text{while döngüsü} \rangle | \langle \text{for döngüsü} \rangle$

$\langle \text{while döngüsü} \rangle ::= \text{while } () \langle \text{durum} \rangle$

$\langle \text{for döngüsü} \rangle ::= \text{for } () \langle \text{durum} \rangle$

$::= \langle \text{atama ifadesi} \rangle | \langle \text{döngü ifadesi} \rangle$

$\langle \text{atama ifadesi} \rangle ::= \langle \text{değişken} \rangle := \langle \text{ifade} \rangle$

Yukarıda görülen örnekte $\langle \text{koşul} \rangle$ ve $\langle \text{ifade} \rangle$ için üretim kuralları dahil edilmemiştir, verilen ($\langle \text{döngü ifadesi} \rangle$, $\langle \text{while döngüsü} \rangle$, $\langle \text{for döngüsü} \rangle$, $\langle \text{koşul} \rangle$, $\langle \text{ifade} \rangle$, $\langle \text{durum} \rangle$, $\langle \text{atama ifadesi} \rangle$ ve $\langle \text{değişken} \rangle$) gibi ifadeler, çeşitli uç birim olmayanları içerir (O'Regan, 2016: 209). Uç birimler arasında ise 'while', 'for', ':=', '(' ve ')' bulunur (O'Regan, 2016: 209).

Tezimizin 3.2. ve devamındaki tüm diğer alt başlıklarında serimlemeye çalıştığımız nesillendirmelerde, anlaşılır olabilmek adına verdiğimiz örneklerde görülen tüm sözdizimleri, birer BNF sözdizimi tanımlamasıdır. Her programlama dili, kendi içinde kullandığı atama değişkenlerinde, *while* döngüsünde veya koşullu durumlarda (İng. if statement) farklılık gösterse de her sözdizimsel belirlenim BNF standardı içinde kalmaktadır. Programlama dillerinde kullanılmış bir sözdiziminden üretilen bir ifadeden yapılan türetme, bu ifadenin her ihtimalini değerlendirebilmek için yapılır. Daha sonra, bir ayrıştırma ağacı ile bu türetmeler, hiyerarşik sözdizimi yapılarına bölünür ve burada türetilmiş ifadelerin bir soyutlaması oluşturulur. “Her dilbilgisi türünün, belirli bir programın dilbilgisine göre geçerli olup olmadığını belirlemek için kendi ayrıştırma algoritması vardır” (O'Regan, 2016: 209). Bir örnek vermek gerekirse,

Gramer $\langle \text{program} \rangle \rightarrow \langle \text{stmts} \rangle$

$\langle \text{stmts} \rangle \rightarrow \langle \text{stmt} \rangle \mid \langle \text{stmt} \rangle ; \langle \text{stmts} \rangle$

$\langle \text{stmt} \rangle \rightarrow \langle \text{var} \rangle = \langle \text{expr} \rangle$

$\langle \text{var} \rangle \rightarrow a \mid b \mid c \mid d$

$\langle \text{expr} \rangle \rightarrow \langle \text{term} \rangle + \langle \text{term} \rangle \mid \langle \text{term} \rangle - \langle \text{term} \rangle$

$\langle \text{term} \rangle \rightarrow \langle \text{var} \rangle \mid \text{const}$

Derivasyon

$\langle \text{program} \rangle \Rightarrow \langle \text{stmts} \rangle \Rightarrow \langle \text{stmt} \rangle$

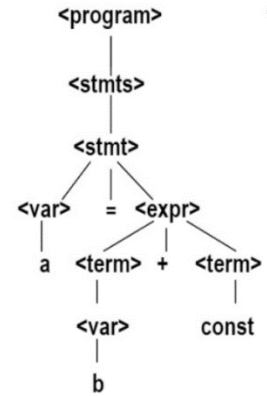
$\Rightarrow \langle \text{var} \rangle = \langle \text{expr} \rangle \Rightarrow a = \langle \text{expr} \rangle$

$\Rightarrow a = \langle \text{term} \rangle + \langle \text{term} \rangle$

$\Rightarrow a = \langle \text{var} \rangle + \langle \text{term} \rangle$

$\Rightarrow a = b + \langle \text{term} \rangle$

$\Rightarrow a = b + \text{const}$



Şekil 3.7 Sözdizimi, türetme ve ayrıştırma ağacı arasındaki ilişkisi

Kaynak: Uzun, 2022c: 29

Bir programlama dilinde program yazarken, sözdizimine verilen önem derecesinde olan iyi bir ifade belirlenimi, bu dilin semantik bağlamına dikkati çeker. Şöyle ki bir ayrıştırma ağacının, türetmelerle doğru dallanma yapabilmesinin önemli bir koşulu, sözdiziminin doğru türetilmesini sağlayacak anlamın yakalanmış olmasına dayanır. Bir program yazıldığı esnada, sözdiziminde kullanılan ifade bloklarının ne anlama geleceğini aktaran semantik durum, programın doğru yürütülmesini sağlar. Bir örnek vermek gerekirse, bir durum bloğu olan *while(boolean_exp) statement-block*, sözdiziminin vereceği “anlam bilgisi (semantik), Boole ifadesinin mevcut değeri doğru olduğunda, bloğun çalıştırılmasıdır. Daha sonra kontrol, işlemi tekrarlamak için dolaylı olarak *Boole ifadesine*⁷⁴ döner. Boole ifadesi yanlışsa, kontrol *while* yapısını izleyen duruma (statement’a) aktarılır” (Uzun, 2022c: 6). Bu nedenle, bir programlama dilinin semantik kısmının doğru belirlenmiş olması, programın başarıyla yürütülmesini sağlar.

Programlama dillerinde bir standart haline gelmiş BNF’nin yalnızca sözdizimine olumlu etkisi olduğu düşünülmemelidir. Bilakis, ifade belirsizliklerinden kurtulmayı sağlayacak semantik kısmın da belirli ölçütlerde korunmasını sağlar. BNF standartlarının bir programlama diline çeşitli kısıtlamalar getirdiği doğru olsa da BNF’nin eşdeğer olduğu düşünülen “bağlamdan bağımsız gramerlerin sınırlamaları bazı avantajlar sağlar. Örneğin,

⁷⁴ Bir programda, durum bloğu veya ifade bloğunun çalıştırılma süreci, aslında bir karar problemidir. Bu tip bloklar birer Boole fonksiyonu olarak görülür ve kullanılan programlama dilinin kendi içinde bir doğrulamasını sağlar: “Hesaplamalı görevlerin önemli bir özel durumu, çıktısı tek bit {0, 1} olan Boole fonksiyonlarının hesaplanmasına karşılık gelir. Bu tür işlevleri hesaplamak, bir EVET/HAYIR sorusunu yanıtlamaya karşılık gelir ve bu nedenle bu görev, bir karar problemi olarak da bilinir. Herhangi bir $F: \{0, 1\}^* \rightarrow \{0, 1\}$ ve $x \in \{0, 1\}^*$ işlevi verildiğinde, $F(x)$ hesaplama görevi, $x \in L$ iken $L = \{x : F(x) = 1\}$ olup olmadığına karar verme görevine karşılık gelir. Bu tip bir karar verme durumu, F işlevine karşılık gelen dil olarak bilinir” (Barak, 2021: 112).

bağlamdan bağımsız gramerlerin boşluğuna karar verilebilir” (Barak, 2021: 369). Aynı koruma örneğin, Tip-1 ve Tip-0 için geçerli olmadığı gibi Tip-3 için de geçerli değildir.

Öncelikle, istisnasız her programlama dili matematiksel bir dildir ve bu dillerin sözdizimlerinin doğru yürütülmesi için belirlenmiş olan tüm kurallar birer fonksiyondur ki bu fonksiyonlar tanımlandıkları durumlar için belirli teoremleri yürütebiliyor olmalıdır. Bu nedenle, bir programlama dilinin sentaksı dışında semantiği de hesaplanabilir olma kuralından müstesna değildir.

Bir programlama dilinin biçimsel anlamı, programların anlamı ile ilgilidir. Bir program, dilin kurallarına göre yazılır ve derleyici daha sonra sözdizimsel olarak doğru olup olmadığını kontrol eder ve eğer öyleyse eşdeğer makine kodunu üretir. Derleyici, dilin anlamını korumalıdır ve dilin sözdizimi bir programın anlamı hakkında hiçbir bilgi vermez (O'Regan, 2016: 209).

Programcının, niyeti doğrultusunda bir program yazması, programın hem sözdizimsel hem de semantik olarak doğruluğuna bağlıdır. Şöyle ki programın uygulanması beklenen durumlardan oldukça farklı bir şekilde davranan sözdizimsel olarak doğru programlar yazmak (O'Regan, 2016: 209), programlama dilleri için belirlenmiş tüm ölçütlere rağmen mümkündür. Chomsky'nin “Renksiz yeşil fikirler öfkeyle uyur” (2002: 15) cümlesi, sözdizimsel olarak doğru fakat semantik olarak anlamsız cümlelere örnek olarak gösterilebilir.

Programlama dillerinin teorik bilgisayar bilimlerinde aldığı linguistik değere gelince, muğlaklığın gelişeceği durumlar sakıncalı olduğundan ve bu alanda geliştirilmiş bir semantik kuram sözkonusu olmadığından, bir programın doğru işletilmesini sağlayacak kurallar dizgesi, henüz sözdizimsel seviyededir. Bu seviyenin de belirli açmazları elbette mevcuttur. BNF standardının dayandığı Tip-2'nin dışında, parantez gerektirmeyen basit durumlarda, Tip-3 kullanılmaktadır ancak Tip-3, parantez dengeleme durumlarında kullanılamaz çünkü yetersizdir. Tip-1 olarak da bilinen “bağlama duyarlı gramer” ise Tip-2'nin çok daha geniş bir sınıfını temsil edebilir yani “hem uçbirimli hem de uçbirimli olmayan sembollerini içeren bir dizgeyi” (Tucker ve Noonan, 2007: 59) oluşturabilir ancak bu gramerin sağladığı tüm sözdizimsel ifade gücü, semantik açıdan “karar verilemez” durumların gelişmesine neden olur. Bunun anlamı şudur: Semantik açıdan “karar verilemez” bir durum (İng. statement), bir makine tarafından “hesaplanabilir fonksiyon” olmaktan çıkar. Özetle, bir ifade durumu (İng. statement) için “karar verilemez” terimini kullanmak, “tüm girdileri için durması garantili bir bilgisayar programı yazamayacağınız ve soruya karar veremeyeceğiniz anlamına gelir. Bu gibi nedenlerle, programlama dili sözdiziminin geliştirilmesinde 'bağlama duyarlı gramer'

kullanılmaz” (Tucker ve Noonan, 2007: 60). Tip-0 olarak da bilinen “sınırsız gramer”e gelince, “karar verilemez” durumlar burada da geçerlidir.

Sınırsız gramer, uzunluk kısıtlamasını bir sağ tarafın boyutuna düşürür. Sınırsız gramer, Turing makinelerine veya tamlığa erişmiş C/C++'a eşdeğerdir. Sınırsız gramer, herhangi bir hesaplanabilir işlevi hesaplayabilir. Bir programlama dilinin sözdizimini tanımlama açısından, sınırsız gramer de bağlama duyarlı gramer ile aynı istenmeyen durumlardan muzdariptir (Tucker ve Noonan, 2007: 60).

Yukarıda sıralamaya çalıştığımız nedenleri özetlemek gerekirse, doğal bir dilde yakalanabilen birçok anlamsal durumun, çeşitlendirilmiş gramer türlerinin her bir sınıfıyla bir programlama dilinde yakalanamadığı, ancak iyi bir belirlenim ve standart getirilmiş “bağlamdan bağımsız gramer” ve ayrıştırma ağaçları algoritmaları ile belli bir dereceye kadar anlamsal koruma sağlanabildiği görülmektedir. Dolayısıyla, sözdizimi tek başına anlambilim için yeterli değildir ve Searle’e göre dijital bilgisayarlar, bilgisayar oldukları sürece, tanımları gereği bir sözdizimine sahiptirler yalnızca (2003: 32). Kaldı ki bir programlama dilinin, linguistik açıdan bir TM’yi kusursuzca yürütebilmesi ise mümkün değildir.

Böylece, Leibniz’in evrensel dilinin birinci ayağı olan *lingua characteristică*’sı, üzerlerine kurulan kütüphanelerle zenginleşmiş programlama dillerinde kendini gösterirken, *calculus ratiocinator*’si ise makinenin mantık mimarisi içine gömülü olan Boole cebirinde ve ifade durumlarında (İng. *while* statement), koşul ifadelerinde (İng. *if* statement) ya da BNF’lerin özel atamalarında görünürlük kazanır. Bu nedenle, programlama dilleri bağamında değerlendirmesini yaptığımız *characteristica universalis* ve *calculus ratiocinator*, Leibniz’in evrensel dil hayalini gerçekleştirmekten uzak görünüyor.

Dilbilimci bakışla değerlendirildiğinde, evrenselliğin gerçek anlamının biçimsel mantığın ötesinde olduğu açıktır yine de evrensellik tartışmasını sürdürme çabası içinde olduğumuz programlama dillerini, herhangi bir doğal dille yarıştırmadan, makine ontolojisinde değerlendirdiğimizi hatırlatmakta yarar var.

Bir bilgisayar dili düşüncesi artık sıradan bir olgudur: BASIC, PASCAL, FORTRAN, APL, LISP gibi dilleri düşünün. Bu yapay 'dillerin' yapı ve içerik bakımından insanın doğal dilinden çok daha basit olduğunu kabul edelim ancak bu farklar birer tür farkı değil de yalnızca derece farkı olabilir. Ayrıca, Noam Chomsky’nin kuramsal çalışması ve dilbilimdeki üretilebilir dilbilgisi yaklaşımı, insanın dili kullanma yeteneğini bilgisayar simülasyonlarının desteğini gerektirecek şekillerde açıklamak için epey yol katetmiştir (Churchland, 2018: 39-40).

Leibniz bağlamında bir evrensel dili hangi zeminde tartıştığımız göz önünde bulundurulmak suretiyle, programlama dillerinin evrenselliklerine yine de bir ölçüt getirebiliriz. Tasarlandıkları ilk dönemlerden bu yana gelişmeye devam eden tüm programlama dilleri ve aileleri, dilbilimci Umberto Eco'nun linguistik bakış açısıyla verdiği tanımlamalarına göre birer *evrensel dil* olmaktan ziyade *evrensel sistem*dirler.

Umberto Eco'ya göre birer *evrensel sistem* olan programlama dilleri, bilgisayar bilimleri bağlamında bilinmektedir ki henüz tasarım sürecindeyken genel amaçlı kullanım ve algoritmik yapıları temeline dayandırılarak, yaygınlaşma potansiyeli taşıdığı anda evrensel bir dil olmaya aday gösterilmektedirler. Daha önce de aktarmaya çalıştığımız üzere “ALGOL 60, bilimsel uygulamalar için evrensel bir programlama dili tasarlama çabalarının sonucuydu” (Sebesta, 2016: 75). Ondan sonra da her modern programlama dili, ALGOL60'ın algoritmik yapısı ve BNF metadil standartları çerçevesinde geliştirilmiştir.

Buna karşın, yukarıda serimlemeye çalıştığımız sentaktik başarılarla rağmen, semantik yetersizliklerin “karar verilemez” hesaplama durumlarını ortaya koyduğunu gördük. “Sözdizimi tanımlamak, anlambilimi tanımlamaktan daha kolaydır, çünkü kısmen, sözdizimi açıklaması için özlü ve evrensel olarak kabul edilmiş bir gösterim mevcuttur, ancak hiçbir anlambilim için henüz geliştirilmemiştir” (Sebesta, 2016: 135). Dar bağlamda, BNF ile sözdizimsel kriterler edinen bu dillerin, semantik kuram açısından değerlendirildiğinde, altlarında yatan mantıksal ve hesaplama dair sınırlandırmalara tabii olduklarını ve henüz bu yönde bir gelişme sağlanamadığını yinelemekte yarar var.

Bu durumda, iyimser bir kavrayışla denilebilir ki programlama dilleri, problemler için çözümler üretebildikleri uygulama alanlarındaki kullanıcılarca eşit ölçüde anlaşılabilir bir sözdizimine sahip olduklarından dolayı ancak ve ancak dar bağlamda bir *evrensellik* kriterini karşılayabilmektedirler. Teorik bilgisayar bilimlerinin önündeki açmazları hesaba kattığımızda ise linguistik açıdan programlama dillerinin, yalnızca çerçevesi iyi belirlenmiş kısıtlı koşullarda kalmak suretiyle yürütülebilir programlar üretebiliyor olması, onların birer *evrensel dil* olduğu anlamına gelmez.

3.3.2.2. Turing-tamlık ve Hesaplama Kuramı Bağlamında Evrensellik Ölçütü

ALGOL, LISP, PROLOG veya C ailesi gibi belirli programlama dillerinin, tasarlandıkları ilk dönemlerden itibaren, sözdizimleri ve algoritmik yapılarından dolayı belli bir evrensellik ölçütünü kendileriyle birlikte bir iddia olarak ortaya attıklarını serimlemeye çalışmıştık. Bu programlama dillerinin tasarımcıları tarafından genel amaçlı kullanım, işletim

sistemi dostu programlar üretme veya platform bağımsız olmak üzere çeşitli nedenlerle evrensel olarak tanımlanabildiklerini de görmüştük.

Günümüz teknolojisine adaptasyon sağlamış birçok ata programlama dilinin yanı sıra farklı paradigmlar doğrultusunda ortaya çıkmış programlama dilleri arasında “hesaplama kuramı” açısından bir fark olmadığını söylemek oldukça ilginç görünse de gerçekte birbirlerinden farklı sayılmazlar. Şöyle ki hangi paradigma sonucu tasarlandığının hiçbir önemi olmadan, altlarında yatan mantık farklılıklarına rağmen tüm programlama dilleriyle üretilen programların, bilgisayar bilimleri açısından öncelikle TM’yi (Turing makinesini) simüle edebiliyor olması gerek şarttır. Bu beceriyle tasarlanmamış FORTRAN, Basic veya COBOL gibi programlama dilleri de altlarında yatan mantığın genişletilmesiyle sonradan geliştirilen versiyonları sayesinde Turing-tamlık (İng. Turing-completeness) seviyesini yakalamışlardır. Böylece modern programlama dillerinin tümünün, “hesaplama kuramı” açısından, prensipte Turing-tam olduğu rahatlıkla ifade edilebilir.

Hatırlanacağı üzere tezimizin 2.3.2.1 alt başlığında TM’lerin fiziki birer nesne temsili olmadığı, bilakis TM’lerin birer program olarak anlaşılması gerektiğini vurgulamıştık. Somut anlamda bir algoritma tanımı vermek üzere Turing’in 1936 yılında kaleme aldığı “Hesaplanabilir Sayılar” adlı makalesinde Turing tarafından tasarlanmış TM’lerin, özünde “hesaplama kuramı”na vurgu yaptığını ve bu kuramın Gödel’in eksiklik teoremlerinin makine ontolojisinde bir izdüşümü olduğunu, ayrıca *Church-Turing Tezine* göre Church’un oldukça soyut (*lambda*) *λ-kalkülüs* hesabıyla eşdeğer görüldüğünü aktarmaya çalışmıştık. Bu makalede, bir insan tarafından yapılabilecek her türlü sonlu hesaplamanın birer programlama sanatı olarak görülen TM’ler (programlar) aracılığıyla da yapılabileceği, Turing tarafından açıkça ortaya konmuştu.

Bilgisayar bilimlerinde önemli bir paradigma olarak kabul edilen programlama dillerinin Turing-tam olması şu anlama gelmektedir: “Bir programlama dili, programları herhangi bir hesaplanabilir fonksiyonu hesaplayabiliyorsa Turing-tamdır (Tucker ve Noonan, 2007: 164). Elbette bu durum programlama dillerinin altında yatan çeşitli mantıkların, Böhm-Jacopini teoremine⁷⁵ göre atama ifadelerinin yanı sıra kontrol yapıları olarak kabul edilen dizi

⁷⁵ Yapılandırılmış program teoremi olarak da bilinen *Böhm-Jacopini teoremi* (1966) herhangi bir programlama diliyle bir program yazarken uyulması gereken mantıksal ve yapısal koşulların yanı sıra algoritmaların akış şemalarıyla ifade edilebilmesi gerekliliğinin bir sonucu olarak ortaya atılmıştır. “*Böhm-Jacopini teoremi*, program şematolojisindeki çoğu erken çalışmada olduğu gibi, genellikle birinci seviye yorumlanmış veya birinci seviye yorumlanmamış (şematik) düzeyde formüle edilir. Birinci seviyeden formülasyon, yeniden yapılandırma sırasında bilgiyi korumak için bireysel yardımcı veya Boole değişkenlerinin eklenmesine izin verir” (Kozen ve Tseng, 2008: 2). *Böhm-Jacopini teoremi* yapısal programlama dillerinin tasarım aşamasında koşul (İng. if) veya döngü (İng. while) yapıları, GOTO gibi belirli kısıtlamalar getirdiğinden dolayı, GOTO işlecinden kurtulma talebi, yapısal programcılığın önünü açmaya çalışan Edsger W. Dijkstra’dan gelmiştir. Bu yolla, *Böhm-Jacopini*

ifadeleri, koşul ifadeleri ve döngü ifadeleri (Tucker ve Noonan, 2007: 164) gibi belirli bazı ölçütleri karşılama durumunu gerektirmektedir. Daha önceki sayfalarda serimlemeye çalıştığımız üzere, tüm programlama dilleri tasarım paradigmalarına özel atanmış farklılıkların temelini oluşturan çeşitli mantıkları dışında, gerçekten de koşul, döngü ve dizi ifadelerini barındırma kriterlerinde ortak yapıdadırlar. Bunun nedeni ise 3.3.2.1 alt başlığında serimlemeye çalıştığımız üzere tüm programlama dillerinin, sözdizimsel yapı tanımlamalarında bir metadil olarak kabul edilen BNF standartlarına uygun olarak ve ayrıca “hesaplama kuramı”nın bir alt dalı olan ve biçimsel dil teorisiyle yakın bağlantılı “özdevinim kuramı”na göre geliştirilmesidir.

Turing-*tamlık* önemlidir çünkü minimum bir dil özellikleri koleksiyonunun, tasarlanabilecek tüm algoritmaları ifade edebileceği bir ölçü sağlar. Turing-*tamlık* zorunlu (temel) dillere özgü değildir; işlevsel, mantık ve nesne yönelimli diller de bu dillerden herhangi birinin, herhangi bir algoritmayı eşit derecede ifade edebilmesi anlamında Turing-*tamdır* (Tucker ve Noonan, 2007: 278 italik vurgu bana ait).

Öyle ise karşımızda programlama dillerinin kendi ölçütlerine göre biçimsellik karakteristiğini belirleyen BNF kriterleri ve bağlantılı “özdevinim kuramı”nın yanı sıra mantık kriterinin Turing-*tamlık* ile verildiği bir bilgisayar bilimleri paradigması durmaktadır. Elimizde üst-gereklilik gibi görünen Turing-*tamlık* ölçütü, programlama dillerinin zorunlu, işlevsel, mantık veya nesne yönelimli paradigmalarından dolayı bu dillerle bir fonksiyonun, ifade ve hesaplama şekli değişiklik gösterse de temelde “bir Turing makinesi için eşdeğer bir işlem dizisine sahip” (Michaelson, 2020: 3) olması olarak görülebilir. Özetlemek gerekirse tüm yürütülebilir kodlar, ister dolaylı olarak *derleme* (İng. compiler) yoluyla ister doğrudan bir *yorumlayıcı* (İng. interpreter) aracılığıyla olsun, fiziksel bir bilgisayarda makine koduna (Michaelson, 2020: 7) dönüştürülebilmelidir. Bu durumda, ifade gücü açısından eşdeğer olduklarından dolayı “belirli bir alana özel belirli bir dili seçmek için deneyim, dil desteği, kod tabanı veya müşteri gereksinimi tarafından yönlendirilen pragmatik nedenler olsa da ilke olarak Turing-*tam* programlama dilleri, birbirinin yerine geçebilir” (Michaelson, 2020: 8 italik vurgu bana ait). Bu nedenle, yüzeysel bir değerlendirme sonucunda kullanım alanı ve

teoreminin gerektirdiği kısıtlamaların önüne geçilmeye çalışılmıştır: “Yalnızca seçim ve ön test mantıksal döngüleriyle yazılan programlar, genellikle yapı olarak daha az doğaldır, daha karmaşıktır ve bu nedenle yazılması ve okunması daha zordur. Örneğin, C# çoklu seçim yapısı, belirgin bir olumsuzluk olmaksızın C# yazılabilirliği için büyük bir destektir. Diğer bir örnek, özellikle ifade basit olduğunda birçok dilin sayma döngüsü yapısıdır” (Sebesta, 2016: 383). Elbette tüm diğer modern programlama dilleri gibi bu programlama dilleri de Turing-*tamdır*.

kolaylığı ya da esnekliği bağlamında evrensellikleri tartışılabilir olan programlama dillerinin, sözdizimlerinin ötesine geçen farklı türden mantıklarının da evrensellikleri önem kazanır.

Hesaplamalı bir bakış açısından, en azından prensipte sorunu bu mantıklar içinde kanıtlayan bir teoreme, bunlara dayalı teorem kanıtlama sistemlerinin Turing-*tam* olması ve herhangi bir Turing makinesinin simüle edilebilmesi anlamında temel anlaşmazlıklarına rağmen, bu tür aday mantıkların çoğu birbirine eşdeğer kabul edilebilir. Bununla birlikte, sosyal bir bakış açısıyla, tek bir evrensel mantık üzerinde anlaşmamamız, belki de mevcut dillerimizden hiçbirinin henüz yeterince evrensel olmadığına bir işaretidir. Henüz her mantıkçı tarafından evrensel olarak kullanılan biçimsel bir dil yoktur (Paleo, 2016: 4 italik vurgu bana ait).

Gerçekten de pek çok programlama dili, döngülerden (*while/for* döngüleri) dolayı Boole değerleriyle (evet/doğru için 1 ve hayır/yanlış için 0 [Python gibi programlama dillerinde büyük harfle yazılmaları gerekir]) çalışırken, SQL (yapılandırılmış sorgu dili) üç değerli mantık (“1”, “0” ve “boş”) ile çalışır. Farklı türden mantıkları birleştirme çabasında olan yapay zekâ uygulamalarında ise kullanılan pek çok programlama dili, bulanık mantıkla da (0 ve 1 değerleri arasında herhangi bir değer alabilen) çalışmaktadır. PROLOG ve LISP, birinci seviye yüklemeler mantığını kullanırken, LISP’te yapılan son dönem eklemelerle birlikte başka mantıklarla uyumlama çabaları söz konusu olmuştur. Günümüzde programlamanın büyük oranda beşinci nesil programlama dillerinin özel alanı olan yapay zekâ uygulamalarına kaymasıyla birlikte bu alanda ayrıca önemli bir yere sahip, yüksek seviye mantıklar da vardır: “Bugün, evrensel biçimsel diller olarak kabul edilmeye en iyi adaylar Coq⁷⁶ ve Isabelle⁷⁷ gibi ispat yardımcıları, tip teorilerine dayanan yüksek seviye mantıklar⁷⁸ temelinde yükselir” (Paleo, 2016: 4).

⁷⁶ Coq bir ispat yardımcısıdır ve “pür işlevsel programlar da dahil olmak üzere matematiksel nesnelere biçimselleştiren ve onlar hakkında ispat yapmaya yardımcı olan bir araçtır” (Blanqui, 2013: 1). Gelişimi 1985’te başlamış olan Coq, gelişimini günümüzde de sürdürmeye devam etmektedir. Fransa’da Thierry Coquand (1961-...) ve Gérard Huet (1947-...) tarafından geliştirilmiştir.

⁷⁷ “Isabelle, çeşitli biçimsel teorilerde etkileşimli akıl yürütme için tasarlanmış genel bir teorem ispatlayıcısıdır. Şu anda yapısal tip teorisi, çeşitli birinci seviyeden mantıklar, Zermelo-Fraenkel küme teorisi ve daha yüksek seviyeli mantık için faydalı ispat prosedürleri” (Paulson, 1998: 3) sağlamaktadır. Isabelle teorem ispatlayıcı 1985 yılında, Lawrence C. Paulson (1955-...) tarafından geliştirilmiştir.

⁷⁸ Bilgisayar bilimlerinde kullanılan yüksek seviye mantıklar, tezimizin 2.3.1 alt başlığında serimlemeye çalıştığımız Russell’in *tipler teorisinin* bir devamı ve uygulama alanı olarak görülebilir. “Bu mantıkların hem teorik hem de pratik bakış açılarından evrenselliği, diğer mantıkları gömme/kodlama (ve hatta Turing makinelerini simüle etme) yetenekleri ve matematik, yazılım, donanım doğrulaması ve hatta metafizik dahil olmak üzere birçok farklı alanda uygulamaları ile kanıtlanır” (Paleo, 2016: 4). Bu mantıkların özellikle programlama dilleri başta olmak üzere biçimsel bir dile kazandırdığı ifade yeteneğine karşın, matematiksel mantık alanında henüz tamlıkları ispatlanabilmiş değildir ancak günümüz bilgisayar ve yapay zekâ uygulamalarında, işlevsellikleri dolayısıyla yaygın bir kullanım alanına sahiptirler.

Matematik dünyasında ispat yardımcıları olarak kullanılan Coq ve Isabelle gibi evrensel dil görünümlü programlama dillerinin yanı sıra Turing-*tamlık* sergileyen tüm Turing ve Church programlama dilleri, altlarında yatan çeşitli mantıkların sınırları çerçevesinde işlev kabiliyeti ve Turing-*tamlık* ölçütünde sınırları belirlenmiş olan dillerdir. Linguistik açıdan değerlendirildiğinde, dar anlamda biçimsel birer *evrensel sistem* olan programlama dillerinin evrensellikleri, bilgisayar bilimleri bağlamında değerlendirildiğinde, mantıksal ve yapısal temellerde yükselen henüz çözülmemiş sorunlar barındırmaktadır.

Bu sorunlardan ilki olan, öncelikle yapısal bir problem olarak görülebilecek Turing-*tamlık* özelliğinin verdiği sınırı ele alarak başlayalım. Özünde, Turing-*tam* bir programlama diliyle yazılmış herhangi bir program, esnek davranamaz yani bir insan kullanıcı veya programcının taleplerini tam olarak yerine getiremeyebilir. Bu nedenle, özellikle pratik uygulamalar için yazılmış programlama dillerinin, Turing-*tamlık* ölçütlerinin zorunluluğundan kurtulması gerektiğine dair görüşler, bu soruna ancak Turing-*eksik* olmak koşuluyla bir çözüm olarak sunulmaktadır.

[u]ygulama programcılarının karşılaştığı herhangi bir algoritmik sorun, prensipte bir Turing-*eksik* programlama dili kullanılarak çözülebilir. (...) Modern pratik programlama dillerinin tasarımcıları, Turing-*tamlıktan* kaçınmayı ve daha iyi doğrulama, güvenlik, otomatik test ve dağıtılmış bilgi işlem sağlamayı kesinlikle düşünmelidir (Pickard, 2020: 1).

Buna göre, çeşitli alanlarda örneğin pratik, web tabanlı veya kullanıcı etkileşimli teorem ispatlayıcılarının aktif rol aldığı uygulamalar için geliştirilmiş programlama dillerinin, Turing-*eksik* kalarak, esnekliklerini ve sorun giderme yeteneklerini koruyabilecekleri düşünülmektedir. Henüz Turing-*tamlık* ölçütünü karşılamadan önce *C* derleyicisiyle yazılmış olan Unix işletim sistemi, GOTO temelinde kalınmış olsaydı, yazılamayacak olan *görüntü işleme programı* (İng. photoshop) ve *çöp toplama* işlevi olmadan büyüemeyecek JavaScript ekosistemi (Pickard, 2020: 1) gibi örnekler, iyi yapılandırılmış Turing-*eksik* olmanın pratik amaçlar doğrultusunda yararına değinse de Turing-*eksiklik* ölçütü yine de kuramsal açıdan Hilbert'in "karar problemi" doğrultusunda Turing'in indirgeme yaptığı "durma problemi" için bir çözümü⁷⁹ gerekli kılmaktadır. Böyle bir çözümün bulunamayacağı ise matamatiksel-mantık temelinde ispatlanmıştır.

⁷⁹ Pickard'ın "Programming Languages Shouldn't and Needn't be Turing Complete" [Programlama Dilleri Turing-*tam* Olmamalı ve Olmaları Gerekmemektedir (2020)] adlı makalesinde yer verdiği, Turing-*eksik* programlama dillerinin program yazmada sağlayacakları verimlilikler şöyle sıralanmıştır: "(1) Bellekte veya zaman tüketiminde çalışma zamanı karmaşıklığı; (2) Kötü amaçlı yazılımları veya verilere yetkisiz erişimi dışlamak için güvenlik koşulları; (3) Yalnızca Tip güvenliğinin ötesinde karmaşık program özelliklerinin

Hatırlanacağı üzere işlevsel programlama dillerinin (*lambda*) λ -kalkülüs hesabını temel aldığı, ilk işlevsel dil olan LISP üzerinden aktarmaya çalışmıştık. Biraz daha geniş bir yelpazede sunmak gerekirse, FORTRAN, COBOL, ALGOL, SIMULA, C, C++, Smalltalk, Java, Python gibi programlama dilleri, Turing dilleri; LISP, ISWIM, PROLOG, ML ve Haskell gibi programlama dilleri ise Church dilleri (Uzun, 2022a: 7) olarak örneklendirilebilir. Turing-tamlık, TM ile ve TM de Church'ün (*lambda*) λ -kalkülüs hesabıyla eşdeğer kabul edildiğinden dolayı, soyutlamaya dayanan ve işlevsel programlama dilleri de Turing-tamlık kriterini sağlamaktadır.

Programlama dillerinin Turing-tam olması, onların ortak özellikleri olarak kendi tasarım paradigmalarının üstünde bir paradigmanın varlığını gösterdiği gibi ayrıca bu dillerin matematiksel mantıkta nasıl bir yere sahip olduklarını da açığa vurmaktadır. Şöyle ki Turing-tamlık, bir biçimsel dil örneği olan *a priori felsefi* özellikli herhangi bir programlama dilinde yazılmış bir F programı için "F'nin fiziksel dünyada hesaplanabilecek herhangi bir durumu temsil edecek kadar anlamlı olduğunu gösterir. Öte yandan, F programlarındaki bir dizi özelliğin, genel olarak kararsız olduğunu belirtir" (Saffidine, 2014: 2). Öyleyse fiziksel dünyada *karar verilemez* herhangi bir problem için herhangi bir Turing-tam programlama diliyle yazılmış bir program tarafından da makine düzleminde bir hesaplama prosedürü yani algoritma geliştirilemez veya geliştirilse dahi bu algoritma ile zaman ve/veya bellek kullanımının bir sonlu zamanda durup durmayacağını bilemeyiz. Dolayısıyla modern programlama dillerinin tümünde halen geçerli bir gereklilik olarak karşımıza çıkan Turing-tamlık şartı, beraberinde Turing makinelerinin bir izdüşümünü verdiği Gödel'in eksiklik teoremlerinin sonuçlarının makine ontolojisi için de geçerli olduğu gerçeğini getirir.

Bilgisayar bilimlerinde, Turing'in "hesaplama kuramı" bağlamında "karar problemi" veya indirgemesini yaptığı "durma problemi" gibi problemler, aslında "karmaşıklık kuramı" çerçevesinde değerlendirilmektedir. "Karmaşıklık kuramı", bilgisayar bilimlerinde alan ve zaman bağlamı olmak üzere iki ayrı alanda gelişme ihtimali olan sorunlar üzerinedir. Buna göre, alan karmaşıklığı bir algoritmanın bilgisayar belleğinde depolaması gereken bilgi bitlerinin sayısıdır ve algoritma, bilgisayarın sağlayabileceğinden daha fazla belleğe ihtiyaç

doğruluğu; (4) Bu spesifikasyondan otomatik olarak oluşturulan kod" (2020: 1-2). Pickard, Turing-eksik bir programlama dilinin kazanağı bu yetenekler doğrultusunda, uygulama alanında yaşanacak kazanımları ise şöyle sıralamıştır: "(1) Eklentilerin ve uygulama mağazası gönderilerinin daha iyi iş birliğine dayalı korumalı alanı ve kontrolü; (2) Öngörülebilir ve güvenli "sunucusuz" bulut işlevleri; (3) Doğrulanmış Bizans hata toleransı da (İng. Byzantine Fault Tolerance [BFT], bir bilgisayar sisteminin, bileşenlerinin bazılarının başarısız olması veya kötü niyetli davranılması durumunda dahi çalışmaya devam etme yeteneği) dahil olmak üzere, daha güvenli ve doğru dağıtılmış bilgi işlem; (4) Daha güvenli blok zinciri akıllı sözleşmeleri ve statik gaz faturalandırması; (5) Zengin çıkarılmış geri bildirimle sahip etkileşimli kodlama ortamları; (6) Daha basit izleme yapısı sayesinde, daha iyi izleme ve çalışma zamanı analizi" (2020: 2).

duyuyorsa faydasızdır, dolayısıyla ekarte edilmelidir (Domingos, 2017: 31). Zaman karmaşıklığı ise algoritmanın çalışmasının ne kadar zaman alacağı veya istenen sonuçları üretmeden önce transistörleri kaç kere daha tekrar tekrar kullanması gerektiğidir ve eğer bu durum makul bir süreden daha uzunsa algoritma faydasız (Domingos, 2017: 32) demektir. Bilgisayar bilimlerinde ister pratik maksatlı olsun ister kuramsal, çözülmesi beklenen birçok problem “karmaşıklık kuramı”nın alanına girmektedir.

“Karmaşıklık kuramı”, teorik bilgisayar bilimlerinde problemlerin çeşitli sınıflarda toplanmasını gerekli kılmıştır. Buna göre, belirli bir zamanda hatasız çözümleri elde edilebilen problemler, P (İng. polynomial) sınıfındadır. Bu tip problemlerin yani P sınıfında bulunan problemlerin çözümleri verimli bir şekilde kontrol edilebiliyorsa, bu problemler NP⁸⁰ (İng. nondeterministic polynomial time [belirsiz polinom zamanlı]) sınıfındadır. Bu nedenle P sınıfına ait tüm problemler, ayrıca NP sınıfındadır. Bunun yanı sıra “karmaşıklık kuramı”nda hem NP hem de NP-zor (polinom zamanda bir çözümü olduğu ispatlanamayan problemler) sınıflarının içerdiği tüm problemlerin bulunduğu sınıf, NP-*tam* sınıfı olarak bilinmektedir.

Gödel’in, von Neumann’a 1956 yılında yazdığı mektubunda⁸¹ ortaya attığı çıkarım da örneğin NP-*tam* sınıfına aittir: “*Entscheidungsproblem*’in karar verilemezliğine rağmen, bir

⁸⁰ Bilgisayar bilimlerinde oldukça önemli bir yere sahip olan P=NP sorusu henüz çözülebilmiş değildir. “Ünlü P=NP sorusu, verimli bir biçimde kontrol edilebilen her problemin aynı zamanda verimli bir biçimde çözümlenebileceğine ilişkindir. NP-*tamlık* nedeniyle, bu soruyu yanıtlamak için bir tek NP-*tamlık* probleminin verimli bir biçimde çözümlenebileceğini kanıtlamak yeterlidir” (Domingos, 2017: 65 italik vurgu bana ait). Bugün ödüllü sorular arasında olan P=NP, NP-*tam* sınıfındadır ve NP-*tam* sınıfında bulunan herhangi bir sorunun, belirli bir algoritmayla çözülebilmesi durumunda, bu sınıfa ait bir soruyu çözen kişiye, Clay Matematik Enstitüsü tarafından bir milyon dolar ödül verileceği duyurulmuştur. NP sınıfında bulunan bir soruyu örnek olarak vermek gerekirse, “Turing’in çözülemezliğini kanıtladığını gördüğümüz Karar Probleminin Gödel’ce 1956’da ortaya koyulan şu bütçeli sürümüdür: Verilen bir matematiksel önermenin, bu önermenin yanında verilen bir miktar kareli kağıda sığabilecek sayıda harf içeren bir kanıtı var mıdır? Verilen sınırlı boyda bir kanıtın hatalı olup olmadığını kontrol etmek kolay bir iş olduğundan, bu problem NP’nin içindedir” (Say, 2019: 73). Bir de zor mu kolay mı olduğu anlaşılamayan, dolayısıyla P’nin içinde mi değil mi karar verilemeyen problemler vardır. Buna bir örnek olarak, matematiksel olmaktan ziyade, gerçek yaşamdan bir soruyu verebiliriz: “Diyelim ki size bir ülkedeki tüm şehirlerarası otobüs hatlarının fiyat bilgileri ve belli bir miktar para verildi. Otobüslerin uğradığı tüm şehirlerden geçen bir tur yapmaya bütçeniz yeter mi yetmez mi? Bu ünlü ‘Seyyar Satıcı Problemi’dir. Ne yazık ki onca yıldır üzerinde çalışan nice parlak beyin, Seyyar Satıcı Probleminin P’nin içinde olup olmadığını bir türlü anlayamamıştır. Ne bu problem için polinom sınırlı zamanlı bir çözüm yöntemi bulunabilmiş ne de böyle hızlı bir yöntemin var olmadığı gösterilebilmiştir” (Say, 2019: 72).

⁸¹ Gödel, mektubunda günümüzde bilgisayar bilimlerinde önemli bir yer tutan P=NP sorusunu matematiksel formülasyon ile sormayı başarmış ilk kişi ünvanını ile de anılabilir, şayet 1956 tarihli bu kayıp mektup ancak 1988 yıllarında ortaya çıkmış olmasaydı. Gödel, mektubunda şu açıklamalarda bulunur: “Açıkçası bir Turing makinesi kolayca inşa edilebilir; bu, birinci seviyeden yüklem mantığındaki her formül F için ve her doğal sayı n için kişinin, n uzunluğunun F (uzunluk = sembol sayısı) bir kanıtı olup olmadığına karar vermesine izin verir. Bunun için makinenin ihtiyaç duyduğu adım sayısı $\psi(F, n)$ olsun ve $\varphi(n) = \max F \psi(F, n)$ olsun. Soru, optimal bir makine için $\varphi(n)$ ’nin ne kadar hızlı büyüdüğüdür. $\varphi \geq k \cdot n$ [bazı sabitler için $k > 0$] gösterilebilir. Gerçekten $\varphi(n) \sim k \cdot n$ (hatta $\sim k \cdot n^2$) olan bir makine olsaydı, bunun çok önemli sonuçları olurdu. Yani, *Entscheidungsproblem*’in karar verilemezliğine rağmen, bir matematikçinin Evet-veya-Hayır sorularına ilişkin zihinsel çalışmasının tamamen bir makine ile değiştirilebileceği açıktır. Sonuçta, doğal sayı n ’i olabildiğince

matematikçinin evet veya hayır sorularına ilişkin zihinsel çalışmasının tamamen bir makine ile değiştirilebileceği açıktır” (1956: 1). Gödel’in çıkarımı, akıllarda “bir matematikçinin (teoremleri ispatlamada) zihinsel çalışmasının tamamen bir makine ile değiştirilip değiştirilemeyeceği” (Barak, 2021: 482) şeklinde bir sorunun oluşmasına neden olmuştur.

İnsanlar, sezgisel mantıkla NP sınıfına ait problemleri yaklaşık olarak çözebiliyorken, NP-tam sınıfına ait problemlerin çözümü günümüzde yapay zekânın alanına girmektedir. “Yapay zekâ, bir tanıma göre, NP-tam problemlere sezgisel (höristik) çözümler bulmaktan oluşur” (Domingos, 2017: 65).

Genel amaçlı kullanıma uygun olsa da Turing-tam bir programlama dilinin her probleme bir çözüm geliştirme kabiliyetinin olmadığı; Turing-eksik bir programlama dilinin ise önünde “hesaplanabilirlik kuramı”na bağlı engeller olduğu, yukarıda serimlemeye çalıştığımız nedenlerden ötürü, açıktır. Bundan dolayıdır ki günümüzde birçok paradigmayı birlikte yürütebilme becerisi kazandırılmış programlama dillerinin gelişimi devam etmektedir ve özel amaçlı kullanım alanlarındaki başarıları, genel amaçlı kullanım alanlarındaki başarılarının üzerindedir.

Bunun bir nedeni, daha önce yer vermeye çalıştığımız üzere programlama dillerinin altında yatan mantıklara dayanmaktadır yani dar bağlamda değerlendirildiğinde, bu dillere çeşitli paradigmalardan eklenebilmesi için gerekli olan diğer mantıkların uyumlandırılması zorunluluğunun yanı sıra geniş bağlamda değerlendirildiğinde, programlama dillerinin temelinde yatan mantıklar üzerinde hemfikir olunamamasıdır.

Leibniz’in zamanındaki sorun evrensel bir dilin olmamasıyken, bugün sorun belki de çok fazla evrensel dilin (ve hatta daha az evrensel olanın) olmasıdır. Yüzeyle, günümüzün üst seviye mantıklarının çoğu birbirine benzer görünse de temellerinde büyük ölçüde farklılık gösterme eğilimindedirler. Örneğin Coq, formüllerin, bağımlı olarak türetilen *lambda* hesabı türleri olduğu ve genişleme gibi ilkelerin geçerli olmadığı sezgisel bir mantığa dayanırken, Isabelle/HOL, formüllerin polimorfik tipli lambda hesabının terimleri olduğu ve genişleme ilkelerinin geçerli olduğu klasik mantığa dayanmaktadır (Paleo, 2016: 4 italik vurgu bana ait).

Makine ontolojisinde evrensellik sorgulamamızda, sözdizime dayalı kriterlerin BNF ve “özdevinim kuramı”na göre belli bir çerçevede oluşturulduğu fakat semantik problemlerden dolayı dilbilimsel açmazların devam ettiğini hatırlatarak, Turing-tamlık dışında biçimsel bir dilin evrenselliğini sorgulatan bu sorunların bir diğeri, mantıksal bir problem

büyük seçmek yeterlidir ki makine bir sonuç vermediğinde, problem hakkında daha fazla düşünmenin bir anlamı yoktur” (1956: 1).

olarak görülebilecek programlama dillerinin altında yatan çeşitli mantıkların uzlaştırılmasının yanı sıra bu mantıkların, merkezi işlem birimiyle de (CPU) uyumluluğunun sağlanması olarak tanımlanabilir. Daha önce yer verdiğimiz üzere, yüzeyde görünür olan yüksek seviye mantıklarla uyumlu çalışan programlama dillerinin temelinde bulunan farklı mantıklar, temelden gelen bir evrensellik sorunu olarak kabul edilmektedir. Örneğin, bu tarzda geliştirilmiş programlama dilleriyle oluşturulmuş teorem ispatlayıcılar, yüksek seviye mantıklarının bir sonucu olarak sundukları esneklikten ötürü kullanıcı dostudurlar. Bu tip yardımcı programların üretildiği programlama dilleri, *lingua characteristic* olarak evrensel görünüme kavuşmuş sistemlerdir ancak altlarında yatan mantıkların, *calculus ratiocinator* olarak verimlilikleri düşmektedir. Bu tip programlama dillerinin yüzeyde belirgin olan ifade gücü ile temelde yatan hesaplama verimliliği arasında bir ters orantı olduğu açıktır.

Uzun bir süre oldukça anlamlı ve evrensel bir biçimsel dili hedefleyen sistemler, özdevinimden vazgeçip kullanıcı tarafından yönlendirilen etkileşimli akıl yürütmeye odaklanmak zorunda kalırken, verimli özdevinimi hedefleyen sistem, ifade ve evrensellikten vazgeçmek zorunda kaldı. Ancak daha yakın zamanlarda, Leibniz'in vizyonu için umut veren yeni bir eğilim ortaya çıktı. Buradaki fikir, her iki tür sistemi de entegre etmektir: Kullanıcı tarafından yönlendirilen etkileşimli teorem ispatlayıcı, sorunun kolay ve sıkıcı kısımlarını otomatik araçlara devredebilir ve çözümlerini kendi güvenilir mantıkları içinde otomatik olarak yeniden yapılandırabilir (Paleo, 2016: 6).

Kullanıcı dostu teorem ispatlayıcılar, istenen teoremin ispatı için bir yandan insan matematikçilerin zamansal bağlama dayalı başaramayacakları kadar uzun vakit isteyen durumlarda, zamandan tasarruf sağlarken, öte yandan boyutları yüzlerle ifade edilen terabaytlarda (bilgisayar bilimlerinde bir terabayt, bir trilyon bayttır) oldukça büyük alanlar kullanmak zorunda kalabilmektedir. Kullanılan alanın büyüklüğü şu anlama gelmektedir: Herhangi bir teorem ispatlayıcıyla yapılmış makul derecelerden daha uzun bir ispat, pratikte bir insan matematikçi (veya matematikçi topluluğu) tarafından ne zamansal ne de zihinsel güç bağlamında kontrol edilebilir. “Açıkçası, bir ifadenin en kısa kanıtı bir terabayt gerektiriyorsa, o zaman insan matematikçiler de bu kanıtı asla bulamazlar” (Barak, 2021: 508).

Bu soruna bir çözüm getirmek maksadıyla, insan matematikçi ve makine teorem ispatlayıcı arasında bir aracı ispatlayıcı birim yani otomatik işlem makinesi gibi davranan bir *yorumlayıcı* devreye sokulmaktadır. Bu durumda, ispatın büyük bir kısmı yine makine tarafından kontrol edilmektedir. Şimdi akıllara şu soru gelebilir: Bir teorem ispatlayıcı tarafından ispatı verilmiş bir teoremin kontrolünü yine bir makine gerçekleştiriyor ise bu ispatın güvenilir bir mantık çerçevesinde kalınarak gerçekleştirildiği ya da daha açık bir ifadeyle bu teoremin ispatının neden geçerli olduğu nasıl anlaşılabilir?

Yukarıda sorduğumuz soruya bir açıklık getirebilmek maksadıyla, bir örnek üzerinden yürümek daha doğru görünüyor. Marijn Heule, Oliver Kullmann ve Victor W. Marek, 2016 yılında “Pisagor Üçlüleri Problemi” olarak bilinen “ $N = \{1, 2, \dots\}$ doğal sayıların hiçbir kısmı $a^2 + b^2 = c^2$ olan bir (a, b, c) üçlüsünü içermeyecek şekilde iki parçaya bölünebilir mi?” (Heule vd., 2016: 1) sorusunu çözebilmek için, kontrol edebileceklerini düşündükleri makinelerden faydalanmışlardı. Bu problemin çözümünde, sekiz yüz işlemci çekirdeği paralel olarak toplam iki gün çalışmış ve ispat, iki yüz terabaytlık bir alana sığmıştı (Heule vd., 2016: 1). İki yüz terabaytlık alan, daha sonra altmış sekiz gigabaytlık bir alana sıkıştırılabildi. Bu ispatın yardımcısı ise bir programlama dili ile yazılmış olan bir teorem ispatlayıcıydı.

Makine tarafından oluşturulan kanıtları tam olarak anlayabilme umudunu tehlikeye atan bir başka sorun da boyutlarıdır. Birçok durumda, özellikle kapsamlı bir kombinasyon araştırmasının gerekli olduğu durumlarda, bilgisayarların teoremi kanıtlama kapasitesi, insan kapasitesini çoktan aşmıştır. Sonuç olarak, insanların kontrol edemeyeceği kadar büyük kanıtlar üretirler. Şu anda Pisagor Üçlüsü problemini çözen, otomatik olarak oluşturulduğu bilinen en büyük kanıt, 200 terabayt alan kaplar. Bir insanın bu kadar büyük ispatları kontrol etmesi pratik olarak imkansız olduğundan, mevcut eğilim ispatları kontrol eden bilgisayar programlarının yanı sıra ispatları sıkıştıran programları uygulamaktır. Bununla birlikte, kontrol edilen ispatların doğruluğuna olan güveni artırmada kesinlikle yardımcı olmalarına rağmen, (henüz) teoremin neden geçerli olduğuna dair kapsamlı bir açıklama çıkaramıyorlar (Paleo, 2016: 8).

Yukarıda sorduğumuz sorumuza, verdiğimiz örnek üzerinden bir yanıt olarak, rahatlıkla görülebilir ki hem zaman hem de alan açısından makul ölçülerin çok üzerinde olan böyle bir ispatın güvenilirliği, kanıtın yardımcı bir program tarafından kontrol edilmiş olmasından ötürü tartışmalıdır. Kaldı ki makul ölçülerde bir ispatın makine çıktısının insan matematikçi tarafından kontrolü durumunda, teorem ispatlayıcı programın sıkıcı ispat kısmını ayrıca yapan otomatik işlem makinesinin bir *yorumlayıcı* veya bir *derleyici* tarafından makine diline çevrildiğini ve bu ispatın kağıda dökümünün, bir insan kullanıcı tarafından anlaşılır olmadığını da eklemekte yarar var. Bu nedenle, makul ölçüler dışında kalan teorem ispatlarının geçerlilik nedeni anlaşılmadığı gibi yapılan bu ispatların makul ölçülerde olanları dahi makine ve insan arasında büyük bir anlam uçurumunun varlığını sergilemektedir.

İçinde bulunduğumuz bölüm için bir özet olarak, programlama dilleri, biçimsel evrensel dil sistemi görünümüleriyle, günümüzde Leibniz’in hayalinin ötesinde sentaktik kurulum ve anlaşılabilirlik kazanmışken, bu dillerin temelinde yatan mantıklar henüz kabul edilmiş bir evrensellik edinmemişlerdir. “Hesaplama kuramı” ve “karmaşıklık kuramı” açısından değerlendirildiğinde, rahatlıkla denilebilir ki bir insan zihni için çözülemez veya

karar verilemez görünen problemler bir makine için de çözülemez veya karar verilemez olabilir. Gödel'in eksiklik teoremleri hem fiziksel hem de zihinsel dünyalarımızda olduğu kadar, bu teoremlerin bir izdüşümü olan Turing'in karar probleminin gösterdiği üzere, makine ontolojisinde de geçerliliğini korumaktadır. Barak'a göre girdileri program olan işlevlerden uzak durduğumuz sürece, "hesaplanamazlık lanetinden" kaçınabileceğimizi umsak da ne yazık ki böyle bir şansımız yok (2021: 377). Dolayısıyla makine zemininde de her doğruluğun bir ispatının verilemeyeceği açıktır yani ispat hakikatı karşılayamaz. Buradan çıkan sonuç, en üstün yeteneklerle donatılmış en gelişmiş programlama dilleriyle dahi henüz Leibniz'in evrensel dil bağlamının gerektirdiği ölçütlerde her hakikat, biçimselleştirilemez.

3.4. Yapay Zekâ Bağlamında Evrensellik Sorgulaması

Beşinci nesil bilgisayarlar dönemi, 1990'lı yıllardan itibaren yapay zekâ sistemleri, ses tanıma sistemleri ve kuantum bilgisayarlarını destekleyecek şekilde donanıma sahip altıncı nesil bilgisayarlara yerini bırakarak kapanmıştır. Günümüzde özellikle saydığımız alanlar üzerinde yoğun çalışmalar sürmeye devam etmektedir.

Tezimizde, Leibniz'in evrensel dil hayalinin izini, günümüzde yaşanan gelişmeler üzerinden sürme çabamız olduğundan dolayı bu bölümde özellikle, yapay zekânın “yapay öğrenme” alanına odaklanacağız. Bunun çok önemli bir nedeni şu şekilde özetlenebilir: “Yapay öğrenme” alanı, günümüzde yazı karakteri tanıma, örüntü, görüntü ve ses tanıma, otonom araç sistemlerinde, NP-*tam* sınıfına ait problemlere çözüm üretme çalışmalarında, doğal dil işleme ve anlık çevirilerde, cep telefonlarındaki Siri gibi asistanlarda, borsa işlemlerinde ve daha pek çoklarını sayabileceğimiz, olağan hayatın içine sirayet etmiş hemen her sistemin arkasında bulunan algoritmaların üretilme yöntemlerini bünyesinde barındırmaktadır.

Genelleme yapacak olursak, “yapay öğrenme” yani kendi kendine öğrenen bilgisayar programları, aslında birer “makine öğrenimi”⁸² algoritmalarıdır. “En temel düzeyinde, makine öğrenimi, bir insan tarafından açıkça programlanmak zorunda kalmadan, kendi kendine 'öğrenebilen' her tür bilgisayar programını ifade eder” (Wehle, 2017: 2). Yapay zekânın bir alt alanı olarak, 1960'lı yıllarda ortaya çıkan “yapay öğrenme” 2000'li yıllarla birlikte yapay zekâ kavramının altında gelişimini sürdüreceği yerde, bu kavramın önüne geçmiş durumdadır.

⁸² Kendi kendine öğrenen makine fikri, kuramsal açıdan ilk defa Turing'in “Hesaplama Makineleri ve Zekâ” (1950) adlı makalesinde yer bulmuştur. Günümüzde bildiğimiz anlamda tüm hayatımızın olağan akışında ilerlemesini sağlayan doğal ortamımızın yanı sıra bir de örneğin hayatımızı oldukça kolaylaştıran “spam filtreleri, ürün tavsiyeleri ve dolandırıcılık dedektörleri dahil olmak üzere çoğu tahmine dayalı programa güç veren 'beyinler', makine öğrenimi algoritmalarıdır” (Wehle, 2017: 2). Aslında makine öğrenimi algoritmaları, bildiğimiz anlamda hayatın genel akışını sağlayan yardımcılarıdır.

Yapay zekânın doğduğu andan itibaren belli belirsiz görünürlük kazanan “yapay öğrenme” beş ana akım halinde birbirine rakipmiş gibi görünen alt dallara ayrılan yaklaşımlar içermektedir: “Sembolistler”, “bağlantıcılar”, “evrimciler”, “Bayesciler” ve “benzetimciler”. Bu akımların her biri, farklı bakış açılarından doğan farklı yöntemlere dayanmaktadır. Ayrıntılarına girmeden özetlemek gerekirse, “sembolistler, öğrenmeyi tümdengelim tersi olarak görür ve felsefe, psikoloji ve mantıktan beslenir. Bağlantıcılar, beyni tersten planlayıp düzenler, sinirbilim ve fizikten ilham alırlar” (Domingos, 2017: 20). Bizim de bu alt başlık altında özel olarak ele alacağımız “derin öğrenme”, bağlantıcı görüşün bir tekniğidir. Diğer akımların yaklaşımlarına dönecek olursak, “evrimciler, evrimi bilgisayarda simüle eder, genetik ve evrimsel biyolojiden faydalanırlar. Bayesciler, öğrenmenin olasılıksal çıkarsamanın bir biçimi olduğuna inanırlar ve kökleri istatistiğe dayanır” (Domingos, 2017: 20). Son olarak, “benzetimciler, benzerlik muhakemelerinden dışdeğerleme yaparak öğrenir, psikolojiden ve matematiksel optimizasyondan” (Domingos, 2017: 20) etkilenmektedirler.

Sembolistlerin formel dili mantıktır; kurallar ve karar ağaçları bunun özel durumlarıdır. Bağlantıcıları sinir ağlarıdır. Evrimcilerinki sınıflandırıcı sistemleri içeren genetik programlardır. Bayescilerin formel dili, Bayesci ağlar ve Markov ağaçları için kullanılan şemsiye bir terim olan grafik modellerdir. Benzetimcilerinki ise destek vektör makinesinde olduğu gibi büyük olasılıkla ağırlıklara sahip spesifik örneklerdir (Domingos, 2017: 305).

Tüm farklı yaklaşımlarıyla bir bütün olarak değerlendirmek gerekirse, “yapay öğrenme” yapay zekâ uygulamaları için geliştirilmiş beşinci nesil dillerin üzerinde yükselmektedir fakat bir farkla; “yapay öğrenme” veri yığınları arasında istatistiksel çıkarımlar yapan algoritmaları kendisi yazmaktadır yani bilgisayara ne yapacağını, bizim yerimize söyleyen algoritmaları kendisi üretir. Dolayısıyla “yapay öğrenme”, kendi kendini geliştiren bir teknolojidir (Domingos, 2017: 17) ve bu da hem yapay zekâ alanında hem de bilgisayar teknolojileri alanında ciddi bir paradigma değişimi olarak görülebilir.

Her alt alanı için ayrı birer tezin yazılabileceği “yapay öğrenme”, tezimize özel vermek istediğimiz örneklerden dolayı bu bölümde, “derin öğrenme” bağlamında yürütülecektir. “Derin öğrenme”, “makine öğrenimi”nin bir alt dalı olarak bilinen “bağlantıcılık” görüşünden doğmuş, çok katmanlı sinir ağları tekniğiyle hiyerarşik bir öğrenme süreci olarak (Wehle, 2017: 3) tanımlanabilir. Günümüzden çok öncesine dayanan çalışmaların bulunduğu 1960 yılında, “sinir ağı araştırmacıları, ağın ayarlanabilir ağırlıklarını değiştirip bütün ağın, 'öğrenme' girdilerine uygun çıktılar vermesini sağlayacak çeşitli yöntemler (Nilsson, 2019: 516) kullanmışlardı. Bundan sonra ise “yapay sinir ağları”,

günümüzün en hızlı ve başarılı ilerleyen teknolojilerinin birleştiği “derin öğrenme” alanında bir yapı taşına dönüştüler.

Derin öğrenmedeki "derin" teriminin anlamı, genellikle sinir ağları olan derin öğrenme modellerinde yerleşik olan birçok katmandan gelir. Bir evrimsel sinir ağı (İng. Convolutional Neural Network [kısaca CNN]), birçok model katmanından oluşabilir. Burada her katman, bir önceki katmandan girdi alır, onu işler ve papatya zinciri tarzında bir sonraki katmana verir (Wehle, 2017: 3 parantez içinde verilen bilgi bana ait).

Yapay zekâ ve “yapay öğrenme” uygulamalarında, biyoloji ve evrimden esinlenilerek tasarlanmış, günümüzün önde gelen popüler çalışma alanlarından biri olan “derin öğrenme”, sembolleştirmeye ve tündengeline değil, bilakis algıya, sezgiye ve tümevarıma dayanmaktadır. Bu temeller üzerinde, tek katmanla başlayıp günümüzde yüz yirmi katmana kadar ulaşmış eğitilebilir çok katmanlı sinir ağları aracılığıyla, “derin öğrenme”, makine öğreniminin eksik kaldığı doğal dil işleme, otonom araç, yüz veya ses tanıma, ilaç keşfi, internet sayfaları, hava basınç tahminleri, tüm mali işlemler gibi listenin uzatılabileceği sayısız pek çok alanda kullanılmaktadır. Örneğin, doğal dil işleme (DDİ [İng. Natural Language Processing kısaltması NLP]) alanında, 2016 yılında kritik eşiğin aşıldığı inancını uyandıran gelişme olarak görülen Google Çeviri motorunun (İng. Google Translate)⁸³ çevirilerine değinmekte yarar var. Çeviri motoru, aldığı geri bildirimler ve devasa veri setlerinin içinde, kelimeler arasında kurduğu ilişki ve örüntüler aracılığıyla, yüksek oranda doğrulukla, anlık çeviriyi hem yazılı hem de sesli olarak yapabilmektedir.

Devasa veri setleri üzerinde çalışırken, YZ (yapay zekâ) gerçek dünyada bu kelimelerin neye tekabül ettiğini bilmeden mevcut kelime ilişkilerini ve örüntülerini keşfeder. Mesela, her kelimeyi bazı belli başlı kelimelere ne kadar benzer olduğunu gösteren binlerce numaradan oluşan bir listeye temsil edebilir. Sonra da buradan “kral” ve “kraliçe” arasındaki farkın “koca” ve “karı” arasındaki farka benzediğini çıkarsayabilir (Tegmark, 2019: 124 parantez içindeki vurgu bana ait).

⁸³ Çevirilerde kullanılan tekniklerde yaşanan büyük iyileşmeye rağmen, henüz mükemmel seviyenin yakalandığı söylenemez ki felsefi sağduyu ile denilebilir ki her zaman ideal bir çeviri hayal olarak kalacaktır. İdeal bir çeviri, beraberinde bir kültürel hafızayı da ister fakat yine de “derin öğrenme” ile yapılan çeviriler yeterince iyidir. “Önceki sistemlerde GOFAI programıyla, her dil için yetenekli dilbilimcilerin en baştan geliştirmesi gereken elle kodlanmış gramerler kullanılıyordu. Yeni sistemler ise gözlemlenmiş kullanım örüntülerinden istatistiksel modelleri otomatik olarak yaratan istatistiksel makine öğrenimi teknikleri kullanıyor” (Bostrom, 2020: 32). İstatistiksel yöntemlere baskın gelen 1980'lere kadar olduğu düşünülen önceki dönem yapay zekâ çalışmaları, özellikle uzman sistemler, mantıkçı paradigmaya dayanıyordu. Türkçe’de “Güzel Eski Moda Yapay Zekâ”, İngilizce GOFAI kısaltmasıyla “Good Old-Fashioned Artificial Intelligence” olarak bilinmektedir.

Öyle ki “derin öğrenme” kullanılarak yapılan bu tarz bir çeviri yaklaşımında “dilbilimcilere⁸⁴ ihtiyaç kalmamıştır: Bu sistemleri yaratan programcıların çalıştıkları dili bilmelerine dahi gerek yoktur” (Bostrom, 2020: 32). “Derin öğrenme” alanının bu denli hızlı geliyor olması elbette internet, cep telefonları ve bilgisayarlar arasında oluşan insan kaynaklı geniş ağ bağlantısı neticesinde oluşan “büyük veri”ye dayanmaktadır. “Yapay öğrenme”nin “sembolist” yaklaşımı dışında kalan diğer alt alanları gibi “bağlantıcı” alt dalının yaklaşım tekniği olan “derin öğrenme” de tümevarımsal yaklaşımla ilerleyerek, büyük veri yığınlarından hipotezler oluşturur.

Fakat büyük hacimli ham veri, yararlı sınıflandırma, nicellenme ve bilgi çıkarımı için etkin “veri madenciliği” teknikleri gerektirir. Yapay öğrenme yöntemleri, veri çözümlemesinde gitgide önem kazanan bir rol oynuyor çünkü muazzam miktarda veriyle başa çıkabilirler. Aslında veri ne denli fazlaysa, o kadar iyidir. Çoğu yapay öğrenme yöntemi, verilerden hipotezler oluşturur. Öyleyse (klasik bir örnek kullanmak gerekirse), eğer büyük bir veri kümesi kuğuların beyaz olduğu birkaç durum içeriyorsa ve kuğuların başka renkte olduğu durum içermiyorsa, o halde yapay öğrenme algoritması, “tüm kuğular beyazdır” çıkarımı yapabilir. Bu tür bir çıkarsama, “tümdengelsel” değil “tümevarımsal”dır (Nilsson, 2019: 501).

Büyük veri yığınları arasında oldukça hacimli matematiksel işlemlerde, “derin öğrenme”den önce makine öğrenimi için yapılan çalışmalar, karmaşık matematik hesaplar karşısında oldukça yavaş kalan CPU’nun (merkezi işlem birimi) seri bağlantılı işlem hacmine ve CPU’nun da bağımlı olduğu makine mimarisine bağlıydı. İşlemlerin aylar veya yıllar sürmesi muhtemel görünen durumların temelinde, bildiğimiz anlamda bir genel işlemci olarak kabul edilen CPU ve dayalı olduğu makinenin mantık mimarisi bulunmaktaydı. Makine öğrenimi için, temelde duran mantığın bu denli karmaşık vektörel veya istatistiksel hesaplamalarla başa çıkmaya çalışırken yavaşlaması ve dolayısıyla hesaplamaları yürütemiyor olması, özellikle hız ve hesaplanabilirlik bağlamında yeni bir tekniğin geliştirilmesini gerektirmişti. Buna göre, yalnızca seri bağlantılı değil, paralel bağlantılı işlemcilerle başvurulacaktı ve bu işlemcinin, mantık mimarisinden bağımsız olması gerekiyordu çünkü

⁸⁴ Bu dilbilimcilerden biri de Chomsky’dir. Pedro Domingos’un Chomsky’nin dilbilimci yaklaşımla “yapay öğrenme” karşısında takındığı tutumuna karşı eleştirisi şu yöndedir: “Yapay öğrenmeye şüpheyle yaklaşan bir diğer önde gelen isim Noam Chomsky’dir. (...) Daha genel olarak, Chomsky tüm istatistiksel öğrenmeyi eleştirir. İstatistiksel öğrencilerin yapamayacağı şeyleri sıralar ama bu liste elli yıl önce miadını doldurmuştur. (...) Nihayetinde, yaşayarak görülmüştür: İstatistiksel dil öğrenciler işe yararken, elle düzenlenen dil sistemleri işe yaramaz. Çığır açıcı ilk örnek, 1970’lerde Pentagon’un araştırma kolu DARPA ilk büyük ölçekli konuşma tanıma projesini organize ettiği zaman ortaya çıkmıştır. Chomsky’nin kolayca alay ettiği türden basit bir ardışık öğrenci, bilgi dağarcığına dayalı gelişmiş bir sistemi, herkesi şaşkırtan bir biçimde yenilgiye uğratmıştır. Bunun gibi öğrenciler günümüzde *Siri* de dahil olmak üzere neredeyse her konuşma tanıtıcıda kullanılmaktadır” (2017: 68-9 italik vurgu bana ait).

öteki türlü örneğin, bir otonom araç için gereken grafik tanıma veya yüz tanıma uygulamalarında, karmaşık algoritmalar sonucu yapılması gereken hesaplamalar gerçekleştirilemiyordu.

CPU, içinde bulunan aritmetik ve mantık birimleriyle bilginin işlenmesini sağlamak dışında, özünde bilgisayarın bütün işlemlerini yapan birimdir. Örneğin, yapay zekâ uygulamalarında kullanılmak üzere CPU'da devreye giren mantık ve aritmetik birimi dışında kalan birimler, bir çeşit uyku moduna geçmektedir. Bu şu anlama geliyor: Herhangi bir yapay zekâ uygulamasında istenen matematiksel işlem performansı, CPU'nun uyku modunda kalan diğer birimlerinin bu iş için kullanılmıyor olmasından dolayı düşmektedir. Dolayısıyla, üzerinde bulunduğu mimariden bağımsız, yalnızca aritmetik ve mantık işlem birimleriyle donatılmış olan başka bir işlemcinin devreye sokulması hem CPU'nun yükünü azaltır hem de işlem hızını ve gücünü artırır. Grafik işlem birimi GPU⁸⁵ (İng. Graphics Processing Units), içinde basit matematiksel işlemler yapabilen ünitesi ve aynı işlemi birlikte yürütebilen paralel bağlantı özelliği sayesinde, aylarca sürececek hesaplamaların günler veya saatlere düşmesini sağlamaktadır. Bu sayede, GPU ve “derin öğrenme” arasında karmaşık hesaplamalarda tatmin edici sonuçlar elde edilmiştir, böylece otonom araç veya yüz tanıma gibi uygulamalarda da büyük gelişme kaydedilmiştir. Açıkça ifade etmek gerekirse, GPU ve “derin öğrenme”, günümüzden geleceğimize, programcılığın ve programların yazılmasını sağlayacak olan muhtemel gelecek nesil programlama dillerinin nereye gideceğinin bir işareti olarak görülebilir.

GPU'lar, derin öğrenme mimarileriyle hızlı eğitim ve test için günümüzde yaygın olarak kullanılmaktadır. Derin öğrenme için hesaplanan matematiksel işlemler, paralelleştirilebilir işlemlerdir. GPU'lardaki çok sayıda çekirdek sayesinde, işlemleri hızlandırmak için hesaplama işlemleri birçok çekirdeğe dağıtılabilir. (...) Birbirinden bağımsız olarak çalışabilen işlem birimlerinin (çekirdek) sayısının çokluğu, bir işlemin paralelleşme miktarında çok önemlidir. Ortalama donanım özelliklerine sahip bilgisayarlarda dört veya sekiz çekirdek üzerinde paralelleştirme işlemleri ile binlerce çekirdeğe sahip GPU'larda paralelleştirme işlemleri aynı seviyede olmayacaktır. Çok sayıda çekirdek, paralelleştirme miktarını artırma etkisine sahip olacaktır. GPU'lar çok sayıda çekirdeğe sahip

⁸⁵ Günümüzde CPU'yu geride bırakan hızı ve “derin öğrenme”ye dayalı alanlarda gösterdiği yüksek performansı sayesinde GPU, birçok zorlu işlem sürecini oldukça hafifletmiştir. “Modern GPU, yalnızca güçlü bir grafik motoru değil, aynı zamanda CPU eşdeğerini önemli ölçüde geride bırakan en yüksek aritmetik ve bellek bant genişliğine sahip, yüksek seviye paralel programlanabilir bir işlemcidir. GPU'nun hem programlanabilirlik hem de kapasitesindeki hızlı artışı, çok çeşitli hesaplama gerektiren, karmaşık sorunları GPU'ya başarıyla eşleyen bir araştırma topluluğu ortaya çıkardı. GPU-programlama olarak da bilinen GPU üzerinde genel amaçlı hesaplamadaki bu çaba, GPU'yu geleceğin yüksek performanslı bilgisayar sistemlerinde geleneksel mikroişlemcilere karşı, zorlayıcı bir alternatif olarak konumlandırdı” (Owens vd., 2008: 879).

olduğunda, CPU çekirdekleri daha yüksek işlemci frekansına sahip olacaklardır (Buber ve Diri, 2018: 1).

Altında yatan hesaplama prosedürlerinin GPU ile güçlendirildiği “derin öğrenme”, üzerinde çalışacağı yapay zekâ uygulamalarına has herhangi bir beşinci nesil programlama diliyle, diğer programlama paradigmaları ve dillerine nazaran, oldukça kısa satırlar süren algoritmayla (hatta bazen aynı algoritmayla) insana özgü birçok görevi yerine getirmeye çalışır. “Derin öğrenme”nin bu görevleri yerine getirmesindeki etken neden ise açıkça “büyük veri” denilen unsurdur. Bilginin artması, insanların dar alanlara yönelerek uzmanlaşmasını gerektirmiştir yine de her bilgiye bir sentez yapacak kadar ulaşabilmek bir uzman insan için mümkün değildir. Özellikle günümüzde ciddi çalışmaların yapıldığı ve duyulan ilginin her geçen gün daha da arttığı “derin öğrenme” aracılığıyla, “yapay öğrenme”, hemen her alanda uzmanlık gerektiren konularda asistanlık yapabilmektedir.

Yapay öğrenme, ilgili bilgi için literatürü tarayıp, bir alanın jargonunu bir başka alaninkine çevirerek ve hatta bilim insanlarının farkında olmadığı bağlantılar kurarak imdada yetişiyor. Yapay öğrenme, bir alanda geliştirilen modelleme tekniklerinin diğer alanlara girmesini sağlayan devasa bir aktarma merkezi vazifesini giderek daha fazla üstleniyor (Domingos, 2017: 44).

“Derin öğrenme”de altta yatan tüm istatistiksel ve vektörel hesaplamalar, bir algoritmanın üzerinde çalıştığı veri yığından, matematiksel fonksiyonlar aracılığıyla, neredeyse hiçbir doğal dile bağımlı olmadan, sonuç elde etmeyi sağlamaktadır. Burada çok önemli bir unsur, “derin öğrenme”nin istediğimiz bilgiyi kendi anlayacağımız dilde bize sunarken, özünde hiçbir dille bağlantılı olmamasıdır. Her şey, görünürde sezgisel mantık, olasılık, türevlenebilir çıktı fonksiyonları, yaklaşıklık hesaplanabildiği hata fonksiyonlarıyla matematik temellidir. “Derin öğrenme”nin, oldukça başarılı olduğu ve hatta bir algoritmanın çıkarılmasının zor olduğu sezgisel dama oyunu Go’da dahi *AlphaGo*⁸⁶ ile insandan daha iyi sayılabilecek performanslara ulaştığı da konumuzla bağlantılı olarak eklenebilir. İnsan

⁸⁶ AlphaGo sezgilere dayalı bir oyun türü olduğundan dolayı, yapay zekâ uzmanlarınca bu oyun için bir formül geliştirilemiyordu. Bunun içindir ki bu oyun, “derin öğrenme” yöntemiyle aşılına çalışıldı. “AlphaGo’nun sinir ağına önce binlerce insan oyunundan pozisyonlar gösterilerek hangi hamlelerin tercih edildiği öğretildi. Elbette ki insanları geçmek için yeterli olmayan bu altyapı, sonraki 'kendi kendine öğrenme' aşamasında oynayacak programların başlangıçta kullanacağı değerlendirme formülünü oluşturmakta kullanıldı. (...) Oyunun sonundaki bu 'ödül' sinyali pekiştirmeli öğrenme yoluyla önceki aşamalarda pozisyonlara yansıtılarak onların da 'makbul' olduğu bilgisi sinir ağına kodlandı. Böylece her aşamada daha da iyileşen değerlendirme formülü, bir sonraki aşamada kendi kendine oynanacak oyunların daha yüksek kalitede olmasına yol açıyordu. DeepMind mühendisleri bu döngüyü durdurdıklarında program insanüstü seviyeye ulaşmıştı. AlphaGo Zero’nun AlphaGo’dan farkı ise insan bilgisine başta bile ihtiyaç duymamasındaydı. Sadece oyunun hamle ve kazanma kuralları bilgisiyle donatılan AlphaGo Zero doğrudan kendi kendine oynamaya geçti” (Say, 2019: 119). AlphaGo Zero, atası AlphaGo’yu kırk gün sonra yendi.

seviyesine ulaşamasa da bir diğer “derin öğrenme” çalışması olan, doğal dil işlemede ve sinir ağları kullanılarak yapılan çevirilerde büyük gelişme sağlandığı da önceki sayfadan malumumuzdur.

Doğal bir dilin terimlerinin bütün anlamlarının tanımlanabileceği ve insan ile makine arasında “anamlı” konuşmalar ya da makinelerin doğal dillere özgü çıkarsamalar geliştirmelerine elverecek kusursuz bir dil düşü, çağdaş Yapay Zekâ araştırmalarında geri dönmektedir. Örneğin, makineye, bir öykünün tutarlı olup olmadığı “yargısını verebileceği” ya da belirli bir kişinin hasta olması olgusundan onun tedaviye gereksinimi olduğu sonucunu çıkarabileceği, vb. çıkarsama kuralları verilmeye çalışılır. Bu konuyla ilgili kaynakça çok geniştir ve önerilen sistemler de -hala temel ya da ilksel bileşenlere dayalı bir anlambilimin olanaklı olduğunu varsayan sistemlerden, makineye eylem, hatta durum şemalan (frames, scripts, goals) veren sistemlere kadar- çok çeşitlidir (Eco, 1995a: 307).

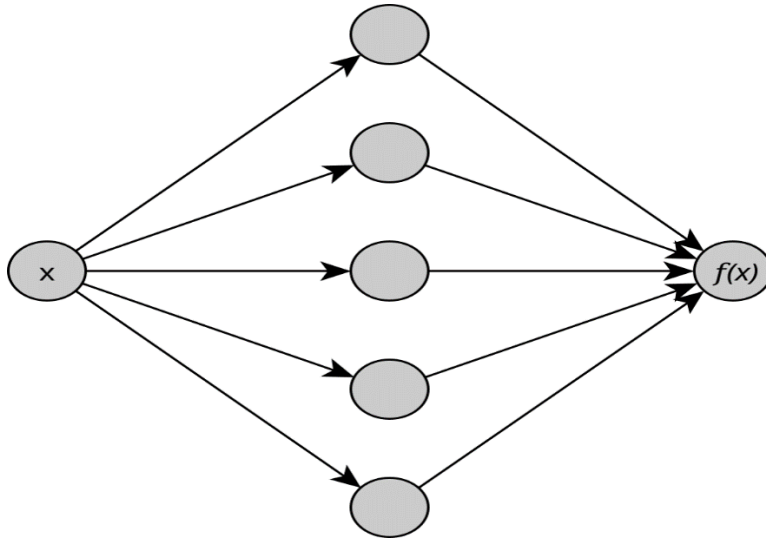
“Derin öğrenme”, genel bağlamda bir yere oturtmak gerekirse, insana ait bilişsel özelliklerin üstesinden gelmekte kullanıldığı kadar, yapay zekâ alanının, çözümü için görevlendirildiği NP-*tamlık* sınıfında bulunan problemlerin çözümünde de kullanılmaktadır. Günümüzde hayatın her alanına yayılmış durumda olan “derin öğrenme”, özellikle de aynı programlama diliyle yazılmış aynı algoritma ile istenen sonucun makine tarafından verilebilir olması bağlamında kayda değerdir.

Bilgisayar bilimlerinde evrensellik ölçütünün, bir programlama dili için tasarım aşamasında genel amaçlı kullanıma dayalı olduğunu, fakat 3.3.2.1 alt başlığında linguistik bağlamda serimlemeye çalıştığımız üzere, sözdiziminde kazanım sağlayan BNF kriterlerine rağmen, bu dillerin semantik problemlerinin henüz gerçek anlamda bir çözüme kavuşmadığını belirtmiştik. Öyle olmasına karşın, günümüzde “yapay sinir ağları”nın (İng. artificial neural networks) evrenselliği lehine çeşitli fikirler öne sürülmektedir. Bu durum aslında şu anlama da gelebilir: Herhangi bir beşinci nesil programlama diliyle yazılmış olan belirli bir algoritma ile her şey için (bir matematik teoremi ispatı, bir kuple şarkıdan şarkının adının ve sanatçısının çıkarımı, dil aileleri birbirinden oldukça uzak olan diller arası başarılı çeviriler gibi) bir hesaplama prosedürü geliştirilebilmesi, dolayısıyla bu algoritmanın uygulanabilirliği bağlamında evrensel oluşu yani bir sinir ağının herşeyi hesaplayabilir güçte oluşu demektir.

Yapay öğrenme alanında bir teknik olarak geliştirilmesi gerektiği fikri 1943'lere dayanan “yapay sinir ağı”, “Amerikalı sinir fizyoloğu Warren McCulloch (1899-1969) ve mantıkçı Walter Pitts'in (1923-1969) sinir hücresinin özünde bir 'mantık birimi' olduğunu” (Nilsson, 2019: 35) iddia etmesinden sonra, 1960'lı yıllarda yeniden keşfedilmiştir. Şimdi, buna dair bir yaklaşım üzerinden devam edeceğiz.

Evrensellik teoremleri, bilgisayar bilimlerinde o kadar yaygın ki bazen ne kadar şaşırtıcı olduklarını unutupyoruz. Ancak kendimize hatırlatmaya değer: Keyfi bir işlevi hesaplama yeteneği gerçekten dikkat çekicidir. Hayal edebileceğiniz hemen hemen her işlem, fonksiyon hesaplaması olarak düşünülebilir. Parçanın kısa bir örneğine dayalı olarak bir müzik parçasına isim verme problemini düşünün: Bu, bir fonksiyonu hesaplamak olarak düşünülebilir. Çince bir metni İngilizceye çevirme sorununu düşünün: Bu bir fonksiyonu hesaplamak olarak düşünülebilir. Bir mp4 film dosyası alma ve filmin konusunun bir tanımını üretme problemini ve oyunculuk kalitesiyle ilgili bir tartışmayı düşünün: Bu da bir tür fonksiyon hesaplaması olarak düşünülebilir. Evrensellik, prensipte sinir ağlarının tüm bunları ve daha fazlasını yapabileceği anlamına gelir (Nielsen, 2015: iv).

“Derin öğrenme” alanında iddia edilen evrensellekle ilgili en önemli olan husus, birer “mantık birimi” olarak görülebilecek nöronlardan oluşan sinir ağının katman sayısıdır ki bu katmanlar çoğaldıkça, aralarındaki matematiksel hesaplamalar da çoğalıp daha karmaşık bir hal almaktadır. “Derin öğrenme” alanında bahsi geçen hesaplamaya dair evrensellik ölçütü, istatistiksel ve olasılıksal yaklaşım ölçütüne dayanmaktadır yani bir sinir ağı katmanındaki gizili nöronların sayısı arttıkça, yaklaşımın, makineden istenen sonucu vermesi olasılığı artar.

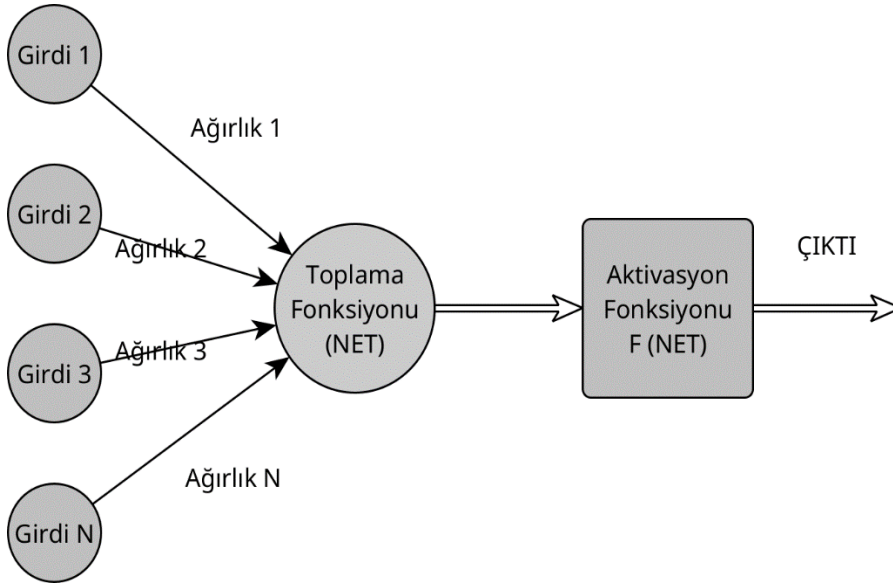


Şekil 3.8 Beş gizli nöron kullanılmış bir sinir ağı gösterimi

Kaynak: Nielsen, 2015: iv

Bir sinir ağının, değer fonksiyonları ve hata fonksiyonlarının belirlenmiş olmasının yanı sıra eğer bir insan uzman tarafından eğitilmesi gerekiyorsa, bu durumda bu sinir ağının, kullanılmadan önce eğitilmesi gerekmektedir. Bunun için sinir ağına, giriş ve çıkış bilgilerinin sunulduğu bir eğitim kümesi uygulanır (Ataseven, 2013: 102). Burada kullanılan fonksiyonlardan çıktı fonksiyonunun, türevlenebilir ve üst dereceden (fonksiyonun fonksiyonu gibi) olması beklenir. Genel bir görünümde ise mutlaka, “yapay sinir ağları”nın hücreleri arasında bir geri bildirim olduğunu ekleyebiliriz. Her geri bildirim süreci, olasılık

tabanlı bir hesaplama dayandır, kendi içinde birçok “evet” veya “hayır”ların olduğu bir iletişimle ağırlıkların hangi ağırlıkları “evet” veya “hayır” olarak algılayıp sürdüreceğini belirler.



Şekil 3.9 Yapay bir sinir hücresinin yapısı

Kaynak: Ataseven, 2013: 102

“Derin öğrenme”de elzem olan, büyük veriden “yapay sinir ağı” tekniği ile istenen sonuçların çıkarılabilmesi durumu, bu iş için belirlenmiş algoritmanın nasıl karar verdiği sorusunu akla getirmektedir çünkü bu işlemde karar verici birim, algoritmadır.

Hesaplama dair evrensellik ölçütünün şu an için “yapay sinir ağları”na bağlanıyor olması bir yana, oldukça iyimser görünen başka bir evrensellik ölçütü de altlarında yatan tüm istatistiksel görünüme karşın, genel olarak “yapay öğrenme” alanında gelişimini sürdürmeye devam eden tüm alt dalların yaklaşımlarının bir uyuşması sonucunda, evrenselliğin bir “master algoritma” ile sağlanabileceği yönündedir. Veri bilimindeki en prestijli ödüle (SIGKDD Innovasyon) layık görülen ve bu iddianın sahibi olan Pedro Domingos’a göre,

belki de gelecekte Amazon’un bulutunun bir yerlerinde gerçekleştirilen siber-mahkeme celsesinde, sürücüsüz otomobilinizle sahile giderken RoboCop’un size kestiği aşırı hız cezası bir robot hakim tarafından hükme bağlanacak ve Leibniz’in bütün argümanları hesaplama indirgeme rüyası sonunda gerçekleşecek (2017: 256).

Domingos’un Leibniz bağlamında bir *calculus ratiocinator* olarak gördüğü “master algoritma”sı açıkça yapay öğrenmenin, mantık temelli ters tümdengelim kullanan sembolistler, sinir ağları kullanan bağlantıcılar, genetik programlar kullanan evrimciler, grafik modeller ve sonsal olasılık kullanan Bayesciler ve destek vektörleriyle kısıtlı optimizasyon

kullanan benzeşimciler olarak sıralanabilecek tüm alt alanlarından gelen ekollerinin, bir tek algoritma üzerine hemfikir olmasına dayanmaktadır. Belli başlı iki ekol arasında kullanılan temel yaklaşımların örtüşük olması dolayısıyla mümkün olabilecek bir birliktelikten doğabilecek bir algoritmanın, diğer tüm temelden farklı yaklaşımları da kapsayacak seviyede örtüşüklük yakalaması mümkün görünmemektedir. Örneğin, mantık temelli çalışan sembolistler ile “yapay sinir ağı” temelli çalışan bağlantıcılar arasında nasıl ve ne tür bir temel uyulaşım sağlanabileceği merak konusudur.

Michael A. Nielsen’in, bu bölümün önceki sayfalarında yer verdiğimiz üzere, hesaplamının evrenselliğini “master algoritma”dan ziyade, güçlü algoritmaların karar mekanizmasını kontrol ettiği “sinir ağları” katmanlarına yerleştirilen gizli nöronların artırılmasına bağladığı açıktır. Gizli nöronlar sayesinde, yaklaşık değerdeki iyileşme ve bu ağlara verilecek her öğrenme işinin bir fonksiyon olarak belirlenebiliyor olması fikri, her fonksiyonun da makinede vereceği çıktının, yapılacak işin öğrenilmesi olduğu anlamına gelir. Bu durumda bir öğrenciye, her şey öğretilir veya yaptırılabilir. Aslında Nielsen’in fikrinin, Domingos’un fikriyle temelde aynı şeye vurgu yaptığı söylenebilir: Leibniz’in hayali gerçekleşiyor.

Gerçekten de *characteristica universalis*, Leibniz bağlamında bir *lingua characteristica* olarak Frege’nin *Kavram-Yazısı* ile gündeme gelmiş ve bundan sonra da gelişerek ve çoğalarak, günümüzde makine ontolojisinde çeşitli alanlarda kullanılmak üzere geliştirilmiş olan, birbirinden ayrık sözdizimlerinin yanında BNF standardı gereği temel ortak özelliklere dayanan, yüzlerce farklı programlama dili olarak, birer biçimsel dil formunda Leibniz’in hayalinin belki de uzanamayacağı seviyelerde gerçekleşme zemini bulmuştur.

Leibniz’in evrensel dil hayalinin bir diğer ayağı olan *calculus ratorator* bağlamında yapılan en önemli temel çalışma, Boole cebiri üzerinde yükselmektedir ve bu hayalin, canlılığını, Turing’den sonra günümüzde özellikle makine ontolojisinde, her argümanın hesaplanabilirliği bağlamında, “derin öğrenme” alanında gerçekleşen atılımlarla birlikte, koruduğu açıktır. Aslında Turing’in, makine ontolojisinde her argümanın hesaplanabilir olmadığına dair ispatının, bir şekilde günümüzdeki yapay zekâ alanında elde edilen başarılı sonuçların ışığında, her argümanın hesaplanabileceği ümidine dönüşmüş olması manidardır. Şöyle ki makine ontolojisinde aranan evrensellik, şu anda tamamen yapay öğrenme aracılığıyla yapılan hesaplamaların evrenselliği üzerinedir.

Elbette yapay öğrenme ile NP sınıfında bulunan bazı problemler çözülmüştür ve diğerleri üzerinde çalışmalar, gelişen teknolojiyle birlikte devam etmektedir fakat bizim felsefi sağduyuyu elden bırakmadan sorgulamamızı sürdürmemiz icap eder. Bunun bir nedeni,

“(p)aradoksal olarak, öğrenen algortimalar doğaya ve insan davranışına yeni pencereler açsa da gizemini koruyor” (Domingos, 2017: 18) olduğu içindir. Öncelikle, yukarıda örneklerini vermeye çalıştığımız “derin öğrenme”de kullanılan teknikleri bir yana bırakacak olursak, bizim değil, bu teknikleri kullanarak sinir ağlarını eğiten yapay öğrenme uzmanlarının, konu ile ilgili ifadelerine bakılacak olursa, bir “yapay sinir ağı”nın, istenen işi nasıl öğrendiği veya karar verici algoritmanın buna nasıl karar verdiği bilinmiyor.

Ancak her durumda, öğrenen algoritmalar bir kara kutu olmayı sürdürüyor. Büyük veri hakkında yazılmış kitaplar bile, bilgisayarların bütün bu terabaytları yalayıp yutup sihirli bir biçimde yeni kavrayışlara ulaştığında gerçekte ne olduğundan bahsetmiyor. En iyi ihtimalle, öğrenen algoritmaların olay çiftleri arasındaki (örneğin Google’da “grip ilacı” araması yapmak ile grip olmak) korelasyonları bulduğu izlenimini ediniriz. Ne var ki korelasyonları bulmak, yapay öğrenme açısından evlerin inşa edilmesinde kullanılan tuğlalardan farksızdır ve insanlar tuğlalarda yaşamaz (Domingos, 2017: 18-9).

Domingos’un yapay öğrenme alanında bir “master algoritma” ile hayata ve bilime dair her şeyin basitçe bir hesaplama indigenebilir olacağına dair sürdürdüğü ümidini, felsefi bir sağduyuyla matematiksel, mantıksal nedenlerden ötürü paylaşamayız. Şöyle ki öncelikle determinist yaklaşımla ortaya çıkmış makine teknolojisi ve buna dayalı programlama dilleri varken hem makinenin mantık mimarisi hem de programlama dillerinin dayandığı mantıklar belirliydi. Özellikle beşinci nesil programlama dillerinden bazıları sezgisel bir mantık temelinde yükselmişse de bunlara dayalı mantık da belirliydi. Oysa günümüzde, tüm bu veri yığınları arasında, yapay öğrenme aracılığıyla üretilen sonuçların arkalarında bulunan mantığın kestirilemiyor olması, felsefi ve epistemolojik problemleri beraberinde getirmektedir.

Bilgisayarların kendileri, yaklaşık, olasılıklı ve büyük veriye dayalı algoritmaların kullanımı yoluyla bizim için giderek anlaşılmaz hale geldiğinden, bilgisayarların kendi anlaşılmaz tartışmalarımızı ve mantıksızlıklarımızı katı bir mantıkla evcilleştirme hayalinin sadece bir kaba dönüşmesi ironiktir. Bu kaba mantık temelli çözümler arayışı, Leibniz’in rüyasının modern bir reenkarnasyonu olabilir (Paleo, 2016: 15).

Bütün modern programlama dillerinin, “hesaplama kuramı” bağlamında “karmaşıklık kuramı” gereği hem zamansal hem de alansal kısıtları mevcuttur. Her ne kadar “yapay öğrenme” alansal bir kısıtlamanın ötesinde teknikler kullanıyor olsa da belirli ispatların dökümünde, insan ile arasında ciddi bir anlam kısıtı olduğu açıktır.

Her modern programlama dili, temelde “hesaplama kuramı” bağlamında, dilin barındırması gereken özelliklerinden dolayı Turing-*tam* olmak zorundadır. Turing-*tamlık*

ölçütüne göre, her modern programlama dili, prensipte Turing makinesini simüle edebilmektedir. Dolayısıyla, bir Turing makinesi ile hesaplanamayan şeyler de bu programlama dilleriyle hesaplanabilir olamaz. İster bir “master algoritma” ümidi, ister “derin öğrenme”de kullanılan “yapay sinir ağları”nın hesaplamada ulaştığı iddia edilen evrenselliği konumuz açısından örneklendiriliyor olsun, her iki evrensellik iddiasında da bu tip algoritmaların, günümüz modern beşinci nesil programlama dillerinden biriyle yazıldığı açıktır. Bu durumda, bu programlama dillerinin Turing-*tamlık* ve “hesaplama kuramı” bağlamında, temelde taşıdıkları kısıtlar üzerinden fonksiyon belirledikleri ortadadır.

“Derin öğrenme” için üretilen algoritmaların, verilen görevleri nasıl yerine getirdiklerine dair gizem henüz çözülemediği gibi daha basit bir ifadeyle bu algoritmaların, temelde duran programlama dillerinin tündengelimsel yapısının dışında, istatistiksel bir tümevarım ile çalıştığı bilinmektedir. Bundan çok daha önemlisi, “derin öğrenme” süreci o kadar olasılıksal ilerlemektedir ki bu sürecin altında yatan mantık, neredeyse çözülemez durumdadır. Bunun en bariz örneği AlphaGo olsa da bugün, altındaki mantığı anlayamadığımız yapay öğrenme uygulamaları, hayatımızın olağan akışı içinde her yere sinmiş durumda ve biz, algoritmalara dayalı bu akışkanlık ve sirayet etmişliğin, pratik ve etik sonuçlarını da henüz bilmiyoruz.

AlphaGo'nun bazı hamleleri, seçimlerinin ardındaki nedenleri kendi oyun bilgisi açısından hemen anlayamayan uzman insan gözlemcileri şaşırttı. Bu tür karar verme teknolojisi şu anda yalnızca zararsız oyun oynama programlarında değil, örneğin otomatik ticaret algoritmalarında ve kendi kendini süren arabalarda da kullanılıyor. Bu tür kritik kullanım bağlamları, potansiyel olarak anlaşılabilir yapay zekâ tarafından alınan kararların güvenliği ve etiği hakkında artan endişelere yol açmıştır (Paleo, 2016: 14).

Henüz her mantıkçı tarafından kabul görmüş bir evrensel mantığın olmadığı yerde, makineler zemininde, “derin öğrenme”nin nasıl bir mantık kullandığının belirsizliği de açıkça sürmektedir. “Yapay öğrenme” alanında elde edilen başarılar ise bizim yapma gayretinde olduğumuz bu tip bir evrensellik sorgulamasını, gölgede bırakacak türdendir. Bu durumda, Nielsen’in iddia ettiği gibi “derin öğrenme”de bir hesaplama evrenselliği varsa, bu evrensellik, algoritmanın gizemi olarak kalmaya devam edecektir çünkü insan zihninin anlayışı için açıklanabilir bir süreç söz konusu değildir.

Bu nedenle, yapay zekâ bağlamında, günümüze kadar gelmiş tüm süreçler arasında elde edilmiş en belirgin başarılı uygulamaları seçmeye çalışarak verdiğimiz bazı örnekler ve iddialar üzerinden bir evrensellik sorgulaması yaptığımız bu bölüm için de Leibniz bağlamında bir evrensel dilin, tüm görünür başarılarla karşın gerçekleşemeyeceğini iddia edebiliriz.

SONUÇ

Bir evrensel dil hayali, Babil kulesi efsanesine dayandırılan Âdemi dil arayışı, kutsiyet atfedilmiş bir kök dil arayışı, ticari amaçlara hizmet edecek doğal diller üzerinden tasarlanmış bir *a posteriori* yapay dil arayışı veya aklın özünden gelen mantıksal temellere dayanan *a priori felsefi* bir dil arayışıyla, Ortaçağ ve Yeniçağda diller üzerine çok çeşitli çalışmaların yürütülmesine neden olmuştur. Bu dil çalışmalarının her biri, gerçek hayata uyarlanamayacak türden hezimetlerle sonuçlanmış, birer *verimli hata* (Lat. *felix culpa*) olarak nihayet Leibniz'in zengin zihinsel dünyasında, matematiğin diliyle harmanlanmışlardır.

Yerel ve doğal diller üzerine ciddi bir külliyata sahip olan, doğal dil Latince üzerinden *a posteriori* bir dil çalışması bulunan Leibniz, doğal dillerin oluş ve bozuluşa tabi olmalarından dolayı varsa bile bir Âdemi dilin ulaşılabilir olmadığını, *a posteriori* bir dilin kullanıma sunulduğu andan itibaren dilin kullanıcısı olan her topluma göre dilin değişim geçirmesi nedeniyle evrensellik özelliğini kaybedeceğinin idraki içindedir. *Yeni Denemeler*'inde Leibniz, doğal dillerin bir kusuru olarak gördüğü müphemliğin, ancak kavramların sadece standart tanımlarının bulunduğu bir evrensel kavramlar sözlüğünün (1996: III. XI. §25. 302) hayata geçirilmesiyle giderilebileceğini iddia etmektedir.

Evrensel olma özelliğini kaybetmeden sürdürülebilir bir yapay dil, Leibniz'e göre ancak ve ancak bir barbarda bile olduğundan şüphe duyulmayacak mantıksal temeller üzerinden yükselmeliydi ve bu dilin kavramları, anlamlara değil kavramlar arası *neliğe* bağlı olarak yani bağlantılar üzerinden yorumlanabilmeliydi. Matematiksel dilin temelinde yatan sembol manipülasyonlarının yapılabileceği pasigrafik bir dil sistemi oluşturmak, konuşulabilir olmasa da yazılabilir, tamamen biçimsel bir dil yaratmak mümkündü.

Hobbes'un "düşüncenin bir çeşit hesap yapmak olduğu" (1996: 24) fikrinden esinlenerek, zihnimizde kavramlarla düşünürken bu kavramların anlamlarına odaklanmadığımızı, bir tür *kör düşünce* gerçekleştirdiğimizi yani düşüncelerimizi bu kavramlar arasında işaretler ve sembollerle yürüttüğümüzü anlayan Leibniz, varsayımsal olarak her ilksel kavrama bir karakteristik sayı atanabileceğini buldu. Böylece Leibniz, kendisinden önce gelen tüm yapay dil çalışmalarının ötesinde bir keşif gerçekleştirdi. Düşüncenin alfabesi olarak tanımladığı *characteristica universalis* böyle ortaya çıkmıştı. Keşfinin değerinin farkında olan Leibniz, bu kavramlara verilen karakteristikler sayesinde basit aritmetik işlemlerle her argümanın doğruluk değerinin ölçülebileceği sonucunu çıkardı. Bu da Leibniz'in *calculus ratiocinator* dediği yöntemsel keşifti.

Kavramlarda ilksellere ulaşma fikri, aslında Leibniz'in evrensel dilini en başından itibaren ütöpik bir dile dönüştürüyordu (Peckhaus, 2004: 7) yine de Leibniz, bir kavram analiziyle herkesçe mutabık olunacak varsayımsal ilkseller önerecekti. Bir kavramlar analizi ve kavramlara karakteristik belirleme işi Leibniz'in tek başına yapamayacağını bildiği ciddi bir çalışma gerektiriyordu (Leibniz, 1989c: 11). Leibniz'in Nicolas Remond'a ölümünden iki yıl önce, 1714 yılında yazdığı mektuba bakılırsa, bir evrensel dil hayalini ömrünün sonuna değin sürdürmüş olduğu ve anlaşılamiyor olmaktan, fikirlerinin bir hayal gibi görünüyor olmasından dolayı üzüntü duyduğu bile söylenebilir.

Marquis de l'Hospital ve diğerleriyle evrensel karakteristiklerim hakkında konuştum ama onlara gördüğüm bir rüyayı anlatmış olmamdan daha fazla ilgi göstermediler. Ben de bunu bariz bir uygulamayla desteklemeliyim fakat bunu başarmak için karakteristiklerimin en azından bir kısmını çalışmam gerekir ki bu kolay olmayan bir görev. Özellikle şu anki durumumda, beni bu tür işlerde teşvik edebilecek ve bana yardım edebilecek insanlarla tartışmanın avantajı olmadan. (Leibniz, 1969k: 656)

Leibniz'in ölümüne değin böyle bir evrensel dilin yaratılabilir olduğuna dair inancı hiç kaybolmamışsa da bu konuda kendine taraftar bulamadığı ortadadır. Leibniz, bulunması durumunda hem keşfe hem de yargıya hizmet edecek bu rasyonel dilin hem bilimin hem de metafiziğin hakikatlerini ortaya koyacak güçte olacağını vurguluyordu. Öyle ki bu dil, mantıksal temellerden yükselen matematiğin diliyle yazılmış kolay öğrenilebilir bir dil olacağından dolayı, gittiği her yere Leibniz'e göre gerçek din Hıristiyanlığı da götürecekti. Leibniz, 1679'da hamisi Hannover dükü Johann Friedrich'e büyük planının ve hayalinin ne işe yaracağını anlattığı bir mektup yazarak projesi için destek istemişti (1969g: 259-262) fakat bu da işe yaramıştı.

Leibniz'in gerçekleştirebilmek için yardımcı ve destekçi bulamadığı evrensel dili, onun ölümünden ancak yüz elli yıl kadar sonra, çalışmaları ve eskizlerinin bir derlemesi yapıldıktan sonra yeniden gün yüzüne çıkmıştır. On dozuncu yüzyılın sonlarına kadar, Leibniz'in evrensel dili, Frege'nin mantıksalcı tutumla ele alacağı matematik aracılığıyla gelişip serpileceği günü beklemiştir. Frege'nin, *Kavram-Yazısı*'nı Leibniz'in *characteristica universalisi* yani *lingua characteristicası* bağlamında bir matematiksel-mantıksal dil olarak yazdığını açıklaması üzerine (2019a: 144; 2020: 20), bundan sonra Frege ve Schröder arasında Leibniz'in evrensel dilinin *lingua characteristicası* kısmının Boole cebirinde mi yoksa Frege'nin *Kavram-Yazısı*'nda mı olduğuna dair yaşanan tartışmalar baş göstermiştir. İki mantıkçı-matematikçi, birbirini karşı sistemin yalnızca bir *calculus ratiocinator* olması bağlamında (Millán, 2021:412) eleştiriyordu.

Böylece Leibniz'in evrensel dilinin ayrılmaz parçalarının, iki ayrı bağlamda ilerleyen mantık sahalarına dönüştüğüne şahitlik etmiş bulunuyoruz. Bu tartışmalardan sonra yapılan çalışmalar, Leibniz'in *characteristica universalis*ini bir *lingua characteristica* olması bağlamında, “dil olarak mantık” sahasına çekerken, *calculus ratiocinator*unu ise “hesap olarak mantık” sahasına itmiştir. Bu iki saha üzerine yapılan çalışmalardan biliyoruz ki Frege'nin matematiği mantıksal temeller üzerinden yürütme çabası, gerçekten bir dil yaratmıştır. Özünde bugün kullandığımız matematiksel notasyonlar, Frege'nin değilse de bu mantık Frege'nindir ve Peano, Whitehead ve Russell tarafından bugün kullanılan notasyonların ve biçimsel dilin temelleri atılmıştır. Boole'ün Leibniz'den bağımsız geliştirdiği sembolik mantığın üzerine, ölümünden sonra yapılan çalışmalar neticesinde oluşturulan Boole cebiri ise Schröder'in iddia ettiğinin tersine, bir *lingua characteristica* değil, sınırlı bir evrenin 0 ve 1'lere indirgenerek doğruluk yargılarının yapılabildiği bir *calculus ratiocinator*dur.

Frege'nin birinci seviye yüklem mantığı olarak da bilinen Leibniz bağlamında *lingua characteristica*sı, doğal dilde ifade edilebilecek her önermenin biçimselleştirilebilmesine olanak sağlıyordu fakat Russell'ın gösterdiği üzere bu dil, kırılıyordu yani paradokslara açıktı. Leibniz'in her muhakemenin hesaba indirgenmesini sağlamak için önerdiği *calculus ratiocinator*unun ancak sınırlı bir kısmını verebilen Boole cebiri ise doğal dilde üretilebilecek her önermeyi biçimselleştirme yetkinliğine sahip değildi.

Bundan dolayı Hilbert, matematiğin, her önermenin biçimselleştirilebileceği, antinomilerden korunaklı, yapılacak her hesaplamanın doğal dilden azade, yalnızca biçimsel dilin sınırları içerisinde kalması gereken sonlu ispat yöntemleriyle ortaya konmasını gerekli gördü. Bu yolda yapılan çalışmaların ana eksenini, matematiğin tamlığı, tutarlılığı ve bir önermenin sistem içerisindeki doğruluk değerine karar verebilen bir algoritmanın olup olmadığına dayanıyordu. Bu çalışmalardan doğan sonuçlar olumsuz olmasaydı, yalnızca matematiğin değil ayrıca Leibniz'in biçimsel dilinin evrensellik ölçütü büyük oranda sağlanmış olacaktı ki Leibniz'in evrensel dili, açıkça böyle bir dil olabilirdi.

Leibniz'in kendi fikrinin de bu yönde olduğunu teslim etmek gerekir: “Kendim için daha önce sahip olduğumdan tamamen farklı bir mantık fikri oluşturmaya başlıyorum. Bunu bir bilim adamının oyalanması olarak görmüştüm ama şimdi anlıyorum ki sizin anladığınız şekilde, bu tıpkı evrensel bir matematik gibi” (Leibniz, 1996: IV. XVII. §8. 415).

Matematiğin biçimsel dilinin en olgun seviyesi olarak bilinen *Principia Mathematica* (kısaca PM) Frege'nin Leibniz bağlamında oluşturduğu birinci seviye yüklem mantığına dayanıyordu ve Peano'nun ideografik dil yapısındaydı. Bu eserin temelinde, Frege'nin

sayıları doğru zeminde tanımlamak için kullandığı mantıksalci yaklaşımı, Peano'nun özellikle doğal sayıları aksiyometikleştirme biçimi ve notasyonları ile birlikte, ayrıca “ilişkiler teorisine dayalı kullanışlı kavramların olduğu” (Heijenoort, 1967: 162) Peirce ve Schröder'in ilişkiler cebiri yatmaktaydı. Her önerme öylesine biçimseldi ki Russell' a göre PM'ye bir kavramlar sözlüğü eklenecek olsa, bu dil mantıksal olarak kusursuz olacaktı.

Bu tür bir dil tamamen analitik olacak ve iddia edilen veya reddedilen olguların mantıksal yapısını bir bakışta gösterecektir. PM'de ortaya konan dilin bu türden bir dil olması amaçlanmıştır. Sadece sözdizimi olan ve hiçbir kelime hazinesi olmayan bir dildir. Bir kelime dağarcığının atlanması dışında, onun oldukça güzel bir dil olduğunu savunuyorum. Öyle bir dil ki bir sözlük eklerseniz, mantıksal olarak mükemmel bir dil olur (Russell, 2010: 25).

Hilbert'in matematiği korunaklı bir alanda sürdürme ve yaşatma çabası olarak da görülebilecek Hilbert projesi, Frege'nin birinci seviye yüklem mantığına dayanan PM üzerinden bir çalışmayı başlatmıştı. Hilbert, projesi için sorduğu her soruya verilecek yanıtların matematiğin lehine olacağını umut etse de bu proje üç adımda hezimete uğratılmıştı.

İlk olarak Löwenheim-Skolem teoremlerinin (kısaca LST) ışığında ortaya çıkan sonuçlara göre, birinci seviye yüklem mantığı kullanılarak oluşturulmuş, matematiğin temelinde bulunan Peano aksiyomları ve Zermelo-Fraenkel küme kuramı da dahil olmak üzere hiçbir aksiyomatik sistemin çelişkisizliği ispatlanamazdı. Bu sorundan kurtulmak için Peano'nun son aksiyomu ve Zermelo-Fraenkel küme kuramının *seçim aksiyomu*, bir tümevarım kuralı ile kendilerini sağlama alma çabasında olsalar da bu aksiyomatik sistemlerin tümevarım kuralları ikinci seviye mantık ile çalışıyordu fakat ikinci seviye mantık da tam değildi. Böylece en başından kabul etmek gerekir ki matematik, temelden itibaren Löwenheim-Skolem teoremlerinin gölgesindedir.

Hilbert'in projesine, LST'den belki de daha vurucu olan darbe Gödel tarafından “eksiklik teoremleri” ile vurulmuştu. Bu teoremler, Hilbert'in istediği koşullarda sonlu bir ispat yöntemi takip edilmek suretiyle matematiğin dili içinde kalınarak kanıtlanmıştı. Buna göre, tutarlı olma iddiasındaki yeteri kadar güçlü herhangi bir aksiyomatik sistem içinde her zaman doğru mu yanlış mı olduğuna karar verilemeyen önermeler olduğu gibi ayrıca doğruluğu bu güçlü sistemden çıkarılamayacak önermeler de vardı. Dolayısıyla, yeterince güçlü bir biçimsel dizgenin tamlığı dizgenin kendi içinde ispatlanamazdı. Bu durumda Gödel, yalnızca matematiğe değil ve ayrıca Leibniz'in evrensel dil hayaline de büyük bir darbe indirmiş oluyordu. Aslında, Gödel'in “eksiklik teoremleri” açıkça doğruluk ve ispat

arasındaki uçurumu göstererek (Casti ve DePauli, 2004: 124), Leibniz'in her hakikatin mutlak bir mantıkla biçimselleştirilebileceğine dair hayalini olanaksızlaştırıyordu.

Şayet Hilbert'in programı başarıyla sonuçlanmış olsaydı, bu durumda Leibniz'in tek başına gerçekleştiremeyeceği kadar büyük bir hayal olan evrensel dil projesi, on dokuzuncu ve yirminci yüzyılın matematik-mantık çalışmalarının kümülatif bir ürünü olarak sunulmuş olacaktı. Denilebilir ki Frege, Peano, Russell ve Whitehead'in katkılarından sonra evrensellik özelliğini barındıran bir *lingua characteristica* olarak bu biçimsel dil, tüm sembol manipülasyonlarıyla, Boole, Peirce, Schröder ve Hilbert'in katkılarından sonra bir *calculus ratiocinator* olarak aritmetik kalkülüs aracılığıyla, düşüncenin, mutlak mantıkla yürütülen mekanik bir hesap yoluyla aktarımının yapılabildiği kesinliğe kavuşmuş olacaktı.

Leibniz'in, evrensel dil hayalini gerçekleştirmek için yalnızca yöntemini değil, dilini de aldığı matematiğin geldiği bu nokta oldukça manidardır. Çünkü bu defa matematik, Leibniz'in evrensel dilinin iki ayrı mantık sistemi üzerinden kendisini olabildiğince biçimselleştirmişti ve Leibniz'in evrensel dilinin biçimsel bir görüntüsünü veren matematik dilinin uğradığı hezimetler, bir anlamda Leibniz'in de hezimetlerine dönüşüyordu.

Leibniz'in evrensel dili için farklı bir alan açan fakat Hilbert projesinin tamamen çökmesine neden olan üçüncü darbe, Turing'den gelecekti. Hilbert'in, bir önermenin sistem içerisindeki doğruluk değerine karar verebilen bir algoritmanın olup olmadığına dair sorusu (Hilbert ve Ackermann, [1928]1950: 132), böyle bir karar algoritmasının bulunamayacağına dair sonlu adımlı ilerleyen bir algoritma ile ortaya konmuştu ancak ilginç olan şeydi ki Turing, "karar problemi" için yazdığı "Hesaplanabilir Sayılar" (1936) adlı makalesinde, hesabı yapan insan matematikçi ile bir makinenin yer değiştirmesi durumunda da hiçbir hesaplama hatasının olmayacağını da kanıtlamıştı.

Turing'in kanıtı, Leibniz'in evrensel dil hayalinin böylece başka bir zeminde yaşatılmasına ve belki de en çok da günümüzde tartışılmasına neden olmuştur. Turing'in makalesinden hareketle, günümüzde Turing makinesi (kısaca TM) dediğimiz soyut matematiksel nesnelere hem programlama dili ve program yazmak için hem de günümüz bilgisayarlarını geliştirmek için gereken işaret fişeği gibiydi. Bilgisayarların mantık mimarisini ve yazılımını yöneten işlemci, *calculus ratiocinator* görünümüne Boole cebirine dayanan bir makine diliyle çalışarak, anladığı tek dil olan 0 ve 1'leri işlerken, programlama dilleri de matematiksel-mantıksal temellerde yükselen birer *lingua characteristica* görünümündeydi.

Bundan sonra önüne geçilemez bir hızda, nesillere ayrılacak düzeylere gelmiş yazılım ve donanım alanına ait teknolojik gelişmelerle toplamda beş olan tüm nesilleriyle günümüze ulaştığı bilinen yüzlerce programlama dili, altıncı neslini sürdüren bilgisayarlar,

cep telefonları ve bunun yanı sıra internetin ortaya çıkmasıyla birlikte, iyice olağan hayatın içine sirayet etmiş yapay zekâ ve “yapay öğrenme” sistemleri geliştirildi.

Programlama dilleri, birer laboratuvar tasarımı olarak baştan aşağı matematiksel mantıksal sözdizimleri ve İngilizce benzeri komut dizileriyle, gerçek birer biçimsel dil olup, Eco’ya göre bu diller, doğal bir dilin asalakları olarak *a priori* görünümündedir ve hataya yer vermedikleri ölçüde *kusursuzdurlar*; ayrıca makine ve insan arasında ortak olduğu düşünülen mantıksal temeller üzerinden yükseldikleri için de *felsefidirler*; bunun yanı sıra bu dilleri bilen veye öğrenen farklı dil konuşucularınca eşit düzeyde anlaşılır olmaları bağlamında birer *evrensel sistemdirler* (1995b: 311).

İlk tasarlanan FORTRAN hariç, sonraki FORTRAN nesli ve ALGOL programlama dilleri de dahil olmak üzere, sonradan nesillenerek geliştirilmiş bütün programlama dilleri için Chomsky’nin dönüşümsel dilbilgisi hiyerarşisinde Tip-2 olarak bilinen “bağlamdan bağımsız gramer” (aşağı sürüklemeli özdevinim [İng.push-down automata]) ile eşdeğerlik kurulan Backus-Naur Formu (kısaca BNF) bir metadil olarak bütün programlama dillerinin sözdizimsel yapıları için uyulması gereken bir kriterdir (Tucker ve Noonan, 2007: 26; Barak, 2021: 375). BNF ve Tip-2 arasında kurulan bağdaşım ise tamamen matematiksel ve mantıksal olup, küme teorisi üzerinden yükselmektedir.

Soyut makine çalışması için, bir makinenin farklı var oluş koşulları, işleyişini yöneten yönergeler veya davranış kuralları ve belirli davranışları ortaya çıkaran girdi sınıfları gibi varlıkları biçimsel olarak temsil etmemiz gerekir. Dilleri incelemek için alfabe, cümle ve dilbilgisi kavramlarını biçimselleştirmemiz gerekir. Bunların her biri, küme teorisinin kavramları ve gösterimi kullanılarak uygun şekilde temsil edilen nesnelere bir koleksiyonudur (Denning vd., 1978: 12).

Bununla beraber, daha önceki sürümler, altlarında yatan mantığı geliştirmek suretiyle, bütün modern programlama dilleri, bir TM ile yapılabilecek her türlü hesaplamayı yapabilmesini sağlayacak mantıksal forma kavuştuğunu simgeleyen Turing-*tamlık* ölçütleri gibi belirlenimlerle donatılmışlardır.

Programlama dillerinin geldikleri noktadan bakıldığında, Leibniz’in evrensel dilinin farklı birer formu oldukları söylenebilir. Öyle ki günümüzde bir evrensel dil bulma hayalinden ziyade, birçok evrensel dil adayı arasından hangisinin gerçek evrensel dil olabileceğine dair sorgulamalar bulunmaktadır. Makine ontolojisinde bir evrensel dilin varlığını sorgulamak, beraberinde evrenselliği birkaç bağlamda değerlendirme zorunluluğu doğurmaktadır. Çünkü gerçek anlamda bir evrensellik ölçütü olduğuna dair elimizde pek az kanıt olduğu gibi bu alanda farklı türden evrensellik iddiaları bulunmaktadır.

Bu nedenle üçüncü bölümde programlama dillerinin evrenselliğini, öncelikle linguistik bağlamda, daha sonra da “hesaplama kuramı” bağlamında değerlendirmemiz icap etti. Öyle ya Leibniz’in evrensel dili, matematiğin “dil olarak mantık” ve “hesap olarak mantık” gibi iki farklı mantık çeşitlenmesi üzerinden biçimselliğini yürütmüştü. Dolayısıyla, matematiğin biçimsel dili üzerinden yükselen programlama dilleri de bu bağlamlarda ayrı ayrı evrensellik sorgulamasına tabii tutulabilir.

Linguistik bağlamda programlama dilleri, BNF kriterlerinden dolayı sözdizimsel olarak korundukları gibi ayrıca programın doğru yürütülmesini sağlayacak olan ayrıştırma ağaçları sayesinde semantik açıdan belli bir ölçüye kadar korunabilmektedirler. Ancak bu koruma bir programlama dilinin semantik olarak anlamsız fakat sözdizimsel olarak mükemmel cümleler üretmesine engel değildir. “Renksiz yeşil fikirler öfkeyle uyur” (Chomsky, 2002: 15) örneğinde görüldüğü üzere, bu cümle sözdizimsel hiçbir sorun içermemesine karşın, tamamen saçma bir cümledir çünkü aynı cümlede hem “renksiz” hem de “yeşil” bulunmaktadır. Bir evrensel dilden beklenen en önemli özellik, anlamda muğlaklık yaratmayacak ölçütlere sahip olmasıdır fakat görünen o ki günümüz modern programlama dillerinin hiçbiri için böyle bir kurtarıcı semantik kuram mevcut değildir. “Artık anlambilimin sözdiziminden çok daha zor tedavi edildiğini biliyoruz” (Perlis, 1981a: 89). Öyle ise birer *lingua characteristic* olarak görülebilecek programlama dillerinde, linguistik bağlamda bir evrensellikten söz edilemez.

Programlama dillerinin “hesaplama kuramı” açısından evrenselliklerine gelecek olursak, öncelikle bütün modern programlama dilleri Turing makinesini simüle etme anlamında yani bir TM ile yapılabilecek her türlü hesaplamayı yapabiliyor olmaları bağlamında altlarında yatan mantık veya paradigma farklılıklarına rağmen eşdeğer kabul edilir (Tucker ve Noonan, 2007: 164). Bu nedenle, Turing-*tam* programlama dilleri, ilke olarak birbirinin yerine geçebilir (Michaelson, 2020: 8 italik vurgu bana ait). Dolayısıyla, bir programlama dili için önemli bir kriter olan Turing-*tamlık*, genelleme yapılmak suretiyle tüm programlama dillerinin sınırlarını da belirler. Şöyle ki Turing-*tam* programlama diliyle yazılmış herhangi bir program, programcının taleplerini tam olarak yerine getiremeyebilir çünkü bu anlamda mantıksal kısıtlamalara sahiptir ve bu yapısal bir problemdir.

Ayrıca, “hesaplama kuramı” bağlamında değerlendirildiğinde zor mu kolay mı olduğuna karar verilemeyen ve ayrıca “karmaşıklık kuramı” bağlamında çözüm bekleyen NP-*tam* sınıfına ait birçok problem mevcuttur. Dolayısıyla bu programlama dilleriyle üretilen programlar tarafından çözülemeyen birçok doğal problem vardır. Bu durumda, programlama dilleri “hesaplama kuramı” bağlamında da evrensel diller olarak görülemezler.

Bunun yanı sıra matematiksel mantık temelinden vurgulanması gereken çok önemli bir husus şudur ki programlama dillerinin temeli üzerinde yükseldiği mantık, birinci seviye yüklem mantığıdır ve bu mantık LST'ye göre kategoriden yoksundur. Sonradan yapılan eklemelerle çok daha kullanışlı ve yararlı hale gelen programlama dillerinin kullandığı yüksek seviyeden mantıklar ise tam-değildir. Dolayısıyla hem matematiğin hem de matematiğin dilinin yansıması olan biçimsel birer dil örneği formundaki programlama dilleri, LST'nin gölgesinde kalmaktadır.

Belki de biraz daha ağır olan diğer bir gölge ise Gödel'in "eksiklik teoremleri"nden kaynaklanmaktadır. Gödel'in "eksiklik teoremleri"nin sonuçlarının, makine ontolojisinde eşdeğeri olan TM'lerin "durma problemi"nin, makine ontolojisinde yarattığı hesaplama dair sınır, programlama dillerinin temel aldığı matematik dilinin, zihinlerimizce belirlenen aksiyomatik sınırlarından ileri gelmektedir.

Gödel'in Teoremi: Aritmetiğin bütün önermelerine karar verme-yani doğruluğunu ispatlama veya yanlışlığını ispatlama-iddiasındaki herhangi bir tutarlı F biçimsel dizgesi için bu dizge içinde ne doğruluğu ne yanlışlığı ispatlanabilen bir aritmetik önerme vardır. Dolayısıyla, F biçimsel dizgesi eksiklidir. Durma Teoremi: Bütün Turing makinesi programlarının durmasına veya durmamasına karar verme iddiasındaki herhangi bir H Turing makinesi programı için bir P programı ve I girdi verisi vardır, öyle ki H programı, P'nin I verisini işlediğinde durup durmayacağını belirleyemez (Casti ve DePauli, 2004: 124).

Buradan çıkan sonuca göre, programlama dilleri de Gödel'in gölgesinde kalmaktadır. Buna göre, bazı doğru argümanlar karar verilemezdir yani her doğru argüman ister matematik zeminde ister makine ontolojisinde olsun, ispatlanamaz. Daha açık bir ifadeyle, insan zihninin her çıkarımı biçimsel bir dille ifade edilemez.

Şimdi özetlemeye çalışacağımız üzere, henüz üzerinde karar kılınabilmiş, genel amaçlı bir evrensel dil yapısının herhangi bir programlama dili tarafından karşılandığını söylemek mümkün değildir.

Yukarıda serimlemeye çalıştığımız nedenlerin dışında, daha basit ifade edecek olursak, Leibniz'in evrensel dili, her insanın kolaylıkla öğrenebileceği ve kullanabileceği anlaşılabilirlikte basit bir sözdizimine sahip olmalıydı; "herkesin kendi dilinde okuyacağı bir yazı, hatta birkaç haftada öğrenilebilecek ve tüm dünyada yaygınlık kazanacak bir dil" (Leibniz, 1969g: 262; Eco, 1995a: 13-4) olmalıydı fakat günümüzde programlama dillerini öğrenmek ciddi bir süreç gerektirmektedir.

Leibniz bağlamında bulunacak evrensel dilin karakteristiklerinin yardımıyla, metafizik dahil her hakikat, ulaşılabilir ve keşfedilebilir olmalıydı fakat yukarıda değindiğimiz

nedenlerden ötürü hiçbir programlama dili aracılığıyla edinilmiş başarının, bu seviyeyi gördüğü veya göreceği söylenemez.

Programlama dilleri, sınırlı alana çözümler getirir durumda olmaktan henüz kurtulabilmiş değildir. Matematik, mantık ve doğa bilimleri temelli alanlarda bu diller, ciddi bir yere sahip olsalar da henüz beşeri bilimlerde gerçek anlamda yer edinmemişlerdir.

Her ne kadar Hilbert projesinin çökmesiyle birlikte, belirlediğimiz aksiyomlar üzerinden yükselen matematik ve matematiksel temelden yükselen programlama dilleri zemininde, Leibniz'in evrensel dilinin ütöpik olduğunu görmüşsek de yapay zekâ uygulamalarında özellikle “derin öğrenme” alanlarında yaşanan son gelişmelerden sonra bu ütöpik dilin oldukça yakınsamış olduğuna dair iddialar mevcuttur: “Leibniz'in bütün argümanları hesaplamaya indirgeme rüyası sonunda gerçekleşecek” (Domingos, 2017: 256) iddiasının arkasında duran nedene gelince, bu iddia tamamen “yapay öğrenme”nin hesaplamada geldiği noktadan temelini almaktadır.

“Yapay öğrenme” alanında çeşitli ilerlemeler kaydedilmesi için teklif edilen programlama dillerinin bazılarında göze çarpan asıl amaç, üzerine çalışılan sistemin zamandan ve mali yüklerden tasarrufunu sağlamak olarak okunabilir. Örneğin, dünya üzerinde CUDA dili dahil olmak üzere, Türkiye’de de çalışıldığını bildiğimiz teklif edilen dillerden bir diğeri YANG (Erel-Özçevik ve Canberk, 2021: 362) olarak bilinmektedir ve özel mimari teknolojiyle uyumlu bu tip birçok dil türemeye devam etmektedir. Bunlar, günümüz paradigmasına uygun tekliflerdir. Elbette bu tür bir tasarruf da hesaplamaya dayalı argümanlardan ortaya çıkmaktadır.

Leibniz'in evrensel dil hayalinin altında yatan asıl neden unutulmuş gibidir. Şöyle ki günümüzde makineler çağında, bu hayalin yerini yalnızca hesaplamaya dayalı argümanların çok daha güçlü yürütülebileceği bir dil imgesi almıştır. O halde bir dil, yalnızca hesaplamaya uygunluğu ve hesaplamaları hızlandırıcı hale getirme ölçütünde mi evrensel sayılabilecek?

Bir fikir olarak ilk ortaya atıldığı 1956'dan bu yana çeşitli duraklama dönemlerinden sonra felsefeye, matematiğe ve diğeri disiplinlere nazaran inanılmaz bir hızla gelişen yapay zekâ teknolojisi, bugün karmaşık matematiksel işlemler ve zor problemlerin ispatlarında insan matematikçilerin çok ötesine geçmiştir. Karmaşık matematiksel işlemlerin arka planda yürütüldüğü “yapay öğrenme” alanı, insana özgü diğeri doğal süreçlerde yani örüntü tanıma, ses tanıma, doğal dil işleme alanlarında anlık çeviri, otonom araç kullanma veya oyun oynama gibi işlerde ise büyük ilerleme kaydetmiştir.

Özellikle “derin öğrenme” tekniğinin mantık birimi olarak da kabul edilen nöronların kullanıldığı “yapay sinir ağları” ile büyük veri yığınlarından tümevarıma ve olasılığa dayalı

karmaşık matematiksel işlemler aracılığıyla hipotez üreterek bilgi sentezleyebilen algoritmalar, her şeyi hesaplanması mümkün bir fonksiyon olarak ele almaktadır. Bu algoritmalarla, “karmaşıklık kuramı”nın alanına giren ister pratik maksatlı olsun ister kuramsal, çözülmesi beklenen birçok problemin çözümü hedeflenmektedir. “Derin öğrenme” alanında iddia edilen hesaplamaya dair evrensellik ölçütü, istatistiksel ve olasılıksal yaklaşımdan kaynaklanmaktadır. Buna göre, bir sinir ağı katmanındaki gizli nöronların sayısı arttıkça, yaklaşımın, makineden istenen sonucu vermesi olasılığı artacaktır. Öyleyse evrensellik iddiası, sinir ağı katmanlarındaki gizli nöronlar arttığı sürece güçlenerek devam edecektir. Kaldı ki hesaplamada evrensellik iddiası, her fonksiyonun hesaplanabilir olması bağlamından yürütülecekse de bu tip olasılık ve istatistiğe dayalı sistemlerin birer kara kutu olması (Domingos, 2017: 18) bağlamında, biz insanlar bunu bilemeyecek ve anlayamayacak durumdayız.

Bir kuramcı olarak bence ilginç olan, son yıllardaki büyük atılımların ardındaki “derin öğrenme” gibi yapay öğrenme tekniklerinin nasıl çalıştığını tam olarak bilmiyor olmamız. Kimi durumlarda harika sonuç veren yöntemlerin özünü anlamak için henüz Turing’in hesaplanabilirliğin genel sınırları için daha ortada bilgisayarlarımız yokken kurduğuna benzer bir kuramsal temelden yoksunuz (Say, 2019: 91).

Turing’in amacı, bir hesaplayıcı makinenin yapabilecekleri ile yapamayacakları arasındaki sınırı tam olarak belirleyebilmektir ve vardığı sonuçlar, yalnızca soyut TM’ler için değil, günümüzün gerçek makineleri için de geçerlidir (Hopcroft vd., 2001: 1). Dolayısıyla, determinist ilerleyen bilgisayar teknolojisi ve buna uygun üretilen programlama dillerinin altında yatan mantıkların belirlenimi “hesaplama kuramı” çerçevesinde mümkündür fakat yapay zekânın bir alt dalı olmasına rağmen, onun önüne geçmiş “yapay öğrenme” ve özellikle günümüzde büyük yer edinmiş “derin öğrenme” istatistiğe dayalı olmakla beraber, bu sistemlerin altlarında yatan mantık çözülemez duruma gelmiştir. Bu alanın, bir kurama ihtiyacı olsa da henüz bu yönde bir gelişme kaydedilememiştir. Öyle ki istenen sonucu vermek üzere öğrenen veya karar veren algoritmaların, neye dayanarak karar verdikleri bilinmemektedir. Bundan ötürü, determinist ilerleyen alanlar için belirlenebilen epistemoloji, “yapay öğrenme” veya “derin öğrenme” alanlarında yerini inanca bırakmış durumdadır. Bu nedenle, insan ve makine arasındaki anlamsal uçurum giderek derinleşmektedir.

“Yapay öğrenme” gibi arkalarında yatan mantığın neredeyse belirsizleştiği tekniklerin kullanıldığı günümüzde, elde edilen büyük ve değerli başarılarla rağmen, “yapay öğrenme” algoritmaları, yapay zekâ için tasarlanmış beşinci nesil programlama dilleri üzerinden

yükselmektedir. Hilbert projesinin çökmesine neden olan sınırların matematik, mantık ve dil bağlamında olduğu kadar, bu bağlamlar üzerinde yükselen programlama dilleri için de geçerli olduğu aşikardır. Kaldı ki felsefi sağduyuyla Leibniz'in evrensel dilini hem *characteristica universalis* hem de *calculus ratiocanor* bağlamında değerlendirecek olursak “doğru” ve ispat” arasında sanıldığından çok daha derin bir uçurum bulunmaktadır. Bu nedenle, Hilbert programının çökmüş olması, matematiği zayıflatmamıştır aksine, matematiğin içine sirayet ettiği hiçbir disiplinin mutlak mantıkla sınırlandırılmayacağını ortaya koymuştur.

Yukarıda, programlama dillerinin ve bu diller üzerinde yükselen yapay zekâ sistemlerinin eksikliğini “Löwenheim Skolem teoremleri”, Gödel'in “eksiklik teoremleri”, Turing'in “durma problemi” ve diğer kuramlar ışığında sunmaya çalıştık. Şimdi konuyu bir de “yapay öğrenme” teknikleri bağlamında daraltarak, günümüzde bu sistemlerin asla çözemeyecekleri sorunları ele alarak değerlendirmeye çalışacağız. Tezimizin 2.3.2. alt başlığında yer verdiğimiz *süreklilik hipotezi* ve onunla bağlantısını ortaya koyduğumuz *seçim aksiyomunun*, yapay zekâ sistemleri tarafından ve “yapay öğrenme” teknikleri aracılığıyla da tıpkı matematik dünyasında olduğu gibi çözümsüz kaldığını ortaya koyan çalışmalar yürütülmüştür. Bu çalışmalardan birinin sunulduğu “Öğrenilebilirlik Kararsız Olabilir” (İng. Learnability Can Be Undecidable) adlı makalede özetle, çeşitli durumlarda “maksimumu tahmin etme” problemine dair bir çözümün, *süreklilik hipotezine* eşdeğer olduğu gösterilerek (Ben-David vd., 2019: 44), en başarılı öğrenme algoritmalarının bile matematiğin sınırlarına takılıp kaldığı ispatlanmıştır.

Öğrenilebilir olanı belirlemek, makine öğreniminin temel bir hedefidir. Bu amaca ulaşmak için önce öğrenilebilirliğin biçimsel bir şekilde ele alınmasına izin veren matematiksel bir çerçeve seçilmelidir. (...) Daha sonra, bu çerçevede öğrenilebilirliği karakterize etmek için somut yollar bulunmalıdır. Bu paradigma, birçok makine öğrenimi bağlamında başarıyla uygulanmıştır. Ancak bu çalışmada, bu paradigmanın iyi çalışılmış bir öğrenme modelinde başarısız olduğunu gösteriyoruz. Öğrenilebilirliğin, standart matematiğin aksiyomları (yani seçim aksiyomu ile Zermelo-Fraenkel küme kuramı veya ZFC küme kuramı) kullanılarak, karar verilemediği basit bir problem sergiliyoruz. Buradan, genel olarak öğrenilebilirliği karakterize eden boyut benzeri bir nicelik olmadığı sonucuna varıyoruz (Ben-David vd., 2019: 44).

Anlaşılabacağı üzere *seçim aksiyomu* ve *süreklilik hipotezi* yalnızca matematiğin değil, teorik bilgisayar bilimlerinin ve güçlü yapay zekâ sistemlerinin de üstesinden gelemediği birer aksiyom ve hipotezdirlere. Bu durum bizi oldukça felsefi bir açıklamaya itmektir: İnsan zihninin ürettiği düşünceler, yapay zekâ projeleriyle bütünlüklü olarak aktarılamaz yani insan zihni kopyalanamaz.

Fiziki olarak insan eli veya zihni tarafından gerçekleştirilemeyen matematikteki modern problemleri çözmek için yapay zekâ ve bilgisayar programlarının geliştirilmesiyle, seçim aksiyomunun uygulanması, birçok bilgisayar bilimcisini benzersiz bir modern ikileme karşı karşıya bıraktı: Programlar insan zihnini kopyalayamaz. Seçim aksiyomu, olağandışı mantığı ve sunumu, IMT Atlantique Bretagne-Pays De La Loire'den Dr. Nicolas Tabareau gibi bilgisayar bilimcilerinin, kopyalamada sorun olduğunu bildirdikleri aksiyomdur. Bu durum, aksiyomun kendisiyle doğrudan bir çelişki olmasa da aksiyomun katıksız insani yönü, teoremlerin ve kuralların normalliklerinden bir sapma, bu alandaki birçok kişiyi uzaklaştıran bir şeydir (Hulkewicz, 2021: 4).

Yapay zekâ sistemleri aracılığıyla insan zihnine erişilemezliğin yanı sıra programlama dillerinin tüm biçimselliğine karşın, Leibniz bağlamında bir evrensel dilin imkansızlığını matematiksel, mantıksal ve felsefi gerekçelere dayanarak sunmaya çalıştığımız bu bölümde, görüleceği üzere ne zihinsel ne de dilsel bağlamda insani olan, ele geçirilebilir değildir. “Matematikçi bilgisayarlaştırılmaz, mekanikleştirilemez. Sadece matematik değil, içinde tanım yapma fiili olan hiçbir disiplin mekanikleştirilemez” (Çevik, 2021: 85).

Tüm disiplinlerin ortak çatısı olan dil bağlamında yaklaşacak olursak, bugün geldiğimiz noktadan bakıldığında, hiçbir istisna yapmadan, bütün “yapay zekâ projeleri, bir biçimde *a priori felsefi* diller sorunsalının mirasçıları olup, *a priori felsefi* dillerle ilgili sorunlardan bazılarını ancak *ad hoc* çözümlerle ve doğal bir dilin bütün eylem alanının çok sınırlı bölgelerinde çözmeyi başarmaktadırlar” (Eco, 1995b: 312), dolayısıyla hem matematiğin hem de doğal dilin sınırlarının ne olursa olsun belirsizliğini koruduğu tüm bu zeminlerde, belli bir dereceye kadar, insana has dilsel ve zihinsel sonsuzluk ele geçirilebilir değildir.

Tezimizde yürüttüğümüz Leibniz bağlamında bir evrensel dil hayalinin gerçekleşmesi sorgulamamızda, yukarıda açıklamaları verilmeye çalışılan nedenlerin bir sonucu olarak, Leibniz’in evrensel dil hayalinin canlılığını özellikle günümüzde, geçtiğimiz yüzyıllara nazaran çok daha tartışılabilir bağlamlar üzerinden koruduğu görülmektedir.

Günümüz paradigması gereği, özellikle beşinci nesil programlama dilleri üzerinden geliştirilen programlar ve algoritmalar, büyük ölçüde makinelerce tasarlanmaktadır. Makinelerin programlama süreci ve sonuçları hakkında verdikleri çıktılar ise insanlar tarafından çoğunlukla anlaşılammamaktadır. Bu makineleri ya da programları anlayacak, daha sonra da bizim anlayış seviyemize indirgeyecek *yorumlayıcı* makineler veya programlar geliştiriyoruz fakat bu tip makine birimleri veya programlar birbirlerini anlıyorsa da biz onları anlayamıyoruz. Öyle ise basit bir çıkarsama yapacak olursak, insan ve makine gibi iki farklı ontolojik zemin üzerinden bir uzlaşma ihtiyacı olduğu görülmektedir.

Gödel'in, "eksiklik teoremleri"ni kanıtladıktan sonra bile matematik temelli bir evrensel dilin bulunabileceğine dair koruduğu umutları, belki de günümüz paradigmasıyla değerlendirme yapmaya daha uygundur. Gödel'in düşüncesine ek olarak kim bilir, belki de altıncı nesil programlama dilleri, bu paradigma üzerinden gelişir.

Şimdiye kadar yaptığımız analizler, aksiyomları oluşturmaya bile yetmiyor ise sadece meydana gelen kavramların analiziyle matematiksel problemlerin sistematik olarak çözülmesi nasıl beklenebilir? Fakat ümidimizi kaybetmeye gerek yok. Leibniz, *characteristica universalis* hakkındaki yazılarında büyük ölçüde ütopyacı bir akıl yürütmeden bahsetmiyordu (Gödel, [1944]1990: 140).

Tezimizin son bölümde ulaştığımız sonuçlara dayanarak, Leibniz'in evrensel dil hayalinin, hangi nedenlerden dolayı mümkün olmadığını aktarmaya çalıştığımız bu bölümde yine de Leibniz bağlamında bir evrensel dil hayalinin kapsamını, mercek tutmaya çalıştığımız epistemolojik zemine genişletmiş bulunmaktayız. Sonuç bölümünde belirtmeyi uygun bulduğumuz amacımız, gelecek nesil programlama dillerinin paradigmalarının, insan-makine gibi iki farklı ontolojik zemin arasında oluşan anlam uçurumunu ortadan kaldırması gerektiğine dair epistemolojik kaygıları sunmaya yöneliktir.

Teknolojinin özellikle "nesnelerin interneti", "derin öğrenme", "otonom araçlar", "akıllı asistanlar" gibi daha pek çokları sıralanabilecek kavramları kolaylıkla dillerimize ve dolayısıyla zihinlerimize yerleştirdiği günümüzde, büyük veri yığınlarından hipotez çıkarma gücüne sahip "derin öğrenme" tekniğiyle yapılan işler, arka planda işleyen algoritmalar sayesinde hayatımızın içine bu denli kolay yerleşirken, bu alanda henüz bir kuram geliştirilememiş olduğunu ve sistem tasarımcılarının, bu karar algoritmalarının karar verme sürecini açıklayamıyor olması, gerçekten de bir evrensel dile hiç olmadığı kadar ihtiyaç olduğunun başka bir göstergesidir. "Artık bu dijital veriye temellenen hesaplamalı düşünme, dijital kültür, dijital bilgi içeren büyük veri paradigmasının egemenliğine kendimizi hazırlamamız gerekir" (Işıklı, 2014: 120).

Öyle ise altıncı nesil programlama dillerinin insan ve makinenin anlam evrenini eşleştirme ya da örtüştürme üzerine olacağını düşünebilir miyiz?

KAYNAKÇA

- Agassi, J. (1969). "Leibniz's Place in the History of Physics". *J. Hist. Ideas*, 30. s.331-44.
- Aiton, E. J. (1985). *Leibniz – A Biography*. Adam Hilger Ltd., Bristol and Boston.
- Aksoy, Y. (2015). *Boole Cebiri ve Lojik Devre Sentezi. (2. Baskı)*. Yıldız Teknik Üniversitesi Basın Yayın Merkezi, İstanbul.
- Altınörs, A. (2003). *Dil Felsefesine Giriş*. İnkılap Kitapevi Yayınları, İstanbul.
- Altınörs, A. (2016). *İdealar ve Dil Bağlamında Locke ile Leibniz*. Bilge Kültür Sanat Yayınları, İstanbul.
- Altınörs, A. (2019). *50 Soruda Dil Felsefesi (4. Baskı)*. Bilim ve Gelecek Kitaplığı, İstanbul.
- Altınörs, A. (2020). *Dil Felsefesi Tartışmaları-Platon'dan Chomsky'ye (3. Baskı)*. Bilge Kültür Sanat Yayınları, İstanbul.
- Altınpulluk, H. (2020). "Biyisayar Donanımı". *Bilişim Teknolojileri*. Anadolu Üniversitesi Yayınları, Eskişehir, s.33-61.
- Altuğ, T. (2001). *Dile Gelen Felsefe*. Yapı Kredi Yayınları, İstanbul.
- Alzahrani, A. A. H. (2020). "4GL Code Generation: A Systematic Review". *International Journal of Advanced Computer Science and Applications*, 11(6), 178-185.
- Antognazza, M. R. (2013). *Leibniz*. (Çev. O. Düz). Türkiye İş Bankası Kültür Yayınları, İstanbul.
- Ares, J., Lara, J., Lizcano, D. ve Martinez, M. A. (2018). "Who Discovered the Binary System and Arithmetic? Did Leibniz Plagiarize Caramuel?". *Science and Engineering Ethics*, 1-19.
- Arnauld, A. ve Lancelot, C. (1975). *General and Rational Grammar: The Port-Royal Grammar*. (İng. Çev. J. Rieux ve B. E. Roll). J. Rieux ve B. E. Roll (Ed.). Mouton & Co. B.V. Publishers, The Hague.
- Arnauld, A. ve Nicole, P. (2003). *Logic or the Art of Thinking* (İng. Çev. J. V. Buroker). J. V. Buroker (Ed.). Cambridge University Press, Cambridge.
- Aspray, W. (1990). "Logic Machines". *Computing Before Computers*. W. Aspray (Ed.). Iowa State University Press, Iowa.
- Ataseven, B. (2013). "Yapay Sinir Ağları ile Öngörü Modellemesi". *Öneri*, 10(39), 101-15.
- Bacon, F. (2003). *The New Organon*. L. Jardine ve M. Silverthorne (Ed.). Cambridge University Press, Cambridge.
- Bodmer, F. (1946). *The Loom of Language. (4. Baskı)*. L. Hogben (Ed.). George Allen & Unwin Ltd., London.

- Boole, G. (1847). *The Mathematical Analysis of Logic*. Philosophical Library Inc., New York.
- Bostrom, N. (2020). *Süper Zekâ-Yapay Zekâ Uygulamaları, Tehlikeler ve Stratejiler*. (2. Baskı). (Çev. F. B. Aydar). Koç Üniversitesi Yayınları, İstanbul.
- Boutroux, E. (2017). *Leibniz Hayatı ve Felsefesi*. (Çev. A. Altınörs). Bilge Kültür Sanat Yayınları, İstanbul.
- Bromley, A. G. (1990). "Difference and Analytical Engines". *Computing Before Computers*. W. Aspray (Ed.). Iowa State University Press, Iowa, s.59-98.
- Brown, S. ve Fox, N. J. (2006). *Historical Dictionary of Leibniz's Philosophy*. The Scarecrow Press, Inc., Lanham, Maryland.
- Brown, J. R. (2008). *Philosophy of Mathematics, A Contemporary Introduction to the World of Proofs and Pictures*. (2. Baskı). Routledge, Taylor&Francis Group, New York.
- Buroker, J. V. (2003). *Logic or the Art of Thinking*. (İng. Çev. J. V. Buroker). J. V. Buroker (Ed.). Cambridge University Press, Cambridge. s. ix-xxvi.
- Casimir, R. J. (1989). "Fourth Generation Problems". *CM SIGPLAN Notices* 24(5), 83-86.
- Casti, J. L. ve DePauli, W. (2004). *Gödel-Mantığa Adanmış Bir Yaşam*. (Çev. E. Akça). Kabalcı Yayınevi, İstanbul.
- Chomsky, N. (2002). *Syntactic Structures*. Mouton de Gruyter, Berlin.
- Church, A. (1936). "An Unsolvability Problem of Elementary Number Theory". *American Journal of Mathematics*. 58(2), 345-363.
- Churchland, P. M. (2018). *Madde ve Bilinç*. (2. Baskı). (Çev. B. Ersöz). Alfa Basım Yayıncılık, İstanbul.
- Collins, R. (2002). *The Sociology of Philosophies* (4. Baskı). Harvard University Press, Cambridge, Massachusetts.
- Cryan, D. Shatil, S. ve Mayblin, B. (2013). *Introducing Logic*. Icon Books Ltd., London.
- Çevik, A. (2021). *Matematik Felsefesi ve Matematiksel Mantık*. (2. Baskı). Nesin Yayıncılık A.Ş., İstanbul.
- Çevikbaş, S. (2006). *Leibniz ve Felsefesi-Mantık, Fizik ve Metafizik*. Çizgi Kitabevi, Konya.
- Çevikbaş, S. (2010). "Bir Şeyin, Başka Her Şeyden Ayırt Edilebilirliğini Sağlayan İlke: Bireyleşim (Aristoteles, A. Thomas, D. Scotus, Leibniz)". *Atatürk Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 7(1), 1-20.
- Danto, A. C. (1975). *General and Rational Grammar: The Port-Royal Grammar*. (İng. Çev. J. Rieux ve B. E. Roll). J. Rieux ve B. E. Roll (Ed.). Mouton & Co. B.V. Publishers, The Hague, s. 11-17.

- Davis, D. M. Sigal, R. Weyuker, E. J. (1994). *Computability, Complexity, and Languages - Fundamentals of Theoretical Computer Science. (2. Baskı)*. Academic Press, New York.
- Davis, M. (2004). "The Myth of Hypercomputation". *Alan Turing: Life and Legacy of a Great Thinker*. C. Teuscher (Ed.). Springer- Verlag Berlin Heidelberg GmbH, s.195-211.
- Davis, M. (2018). *The Universal Computer-The Road from Leibniz to Turing. (3. Baskı)*. CRC Press, Taylor&Francis Group, New York.
- Deleuze, G. (2007). *Leibniz*. (Çev. U. Baker). Kabalcı Yayınevi, İstanbul.
- De Morgan, A. (1847). *Formal Logic: Or, The Calculus of Inference, Necessary and Probable*. Taylor and Walton, London.
- De Morgan, A. (1849). *Trigonometry and Double Algebra*. Taylor, Walton & Maberly, London.
- Denning, P. Dennis, J. B. ve Qualitz, J. E. (1978). *Machines, Languages, and Computation*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
- Descartes, R. (1998). *Yöntem Üzerine Konuşma*. (Çev. A. Timuçin). Cumhuriyet Yayınları, İstanbul.
- Doğan, E. (2020). *Avrupa'da İlk Yapma Dil Volapük. (2. Baskı)*. Kesit Yayınları, İstanbul.
- Doğrucan, M. F. ve Hazar, Z. (2019). "Yapay Zekâ Çalışmalarında Dilsel Arka Plan ve Felsefe" *Pamukkale Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, Denizli, (34), 159-167.
- Doğrucan, M. F. (2020). *Medeniyet Dili Olarak Türkçe (2. Baskı)*. Altınordu Yayınları, Ankara.
- Domingos, P. (2017). *Master Algoritma (2. Baskı)*. (Çev. T. Göbekçin). Paloma Yayınevi, Ankara.
- Dönmez, S. (2020). "Felsefi Bağlamda Yapay Zekâ ve 2025 Sendromu". *Çukurova Üniversitesi İlahiyat Fakültesi Dergisi*, Adana, 20/2, s.748-760.
- Eberbach, E. Goldin, D. Wegner, P. (2004). "Turing's Ideas and Models of Computation". *Alan Turing: Life and Legacy of a Great Thinker*. C. Teuscher (Ed.). Springer- Verlag Berlin Heidelberg GmbH, 159-194.
- Eco, U. (1995a). *Avrupa Kültüründe Kusursuz Dil Arayışı*. (Çev. K. Atakay). Afa Yayıncılık, İstanbul.
- Eco, U. (1995b). *The Search For The Perfect Language*. (İng. Çev. J. Fentress). Blackwell Publishers Inc., Cambridge, Massachusetts.
- Eldeniz, L. (1994). "Programlama Dilleri". *Marmara İletişim Dergisi*, (7), 139-144.

- Erel-Özçevik, M. Tekçe, F. ve Kayapınar, F. (2020). "Analytical Delay Model of virtual CPE for 5G eMBB services in SD-CDN". *2020 16th International Conference on Network and Service Management (CNSM)*.
- Erel-Özçevik, M. ve Canberk, B. (2021). "OFaaS: OpenFlow Switch as a Service for Multi Tenant Slicing in SD-CDN". *IEEE Transactions on Network and Service Management*, 18(1), 362-373.
- Erhat, A. (1996). *Mitoloji Sözlüğü. (6. Basım)*. Remzi Kitabevi A. Ş., İstanbul.
- Frege, G. (2019a). *Kavram-Yazısı*. (Çev. M. Özdemir). Külliyat Yayınları, İstanbul.
- Frege, G. (2019b). *Fonksiyon ve Kavram*. (Çev. M. Özdemir). Külliyat Yayınları, İstanbul.
- Frege, G. (2020). *Aritmetiğin Temelleri: Sayı Kavramı Üzerine Mantıksal-Matematiksel Bir İnceleme. (5. Baskı)*. (Çev. H. B. Gözkan). Yapı Kredi Yayınları, İstanbul.
- Gensler, H. J. (2017). *Introduction to Logic. (3. Baskı)*. Routledge, Taylor and Francis Group, New York and London.
- George, A. (1985). "Skolem and the Lowenheim-Skolem Theorem: A Case Study of the Philosophical Significance of Mathematical Results". *History and Philosophy of Logic*, 6(1985), 75-89.
- Goldstine, H. H. (1993). *The Computer from Pascal to von Neumann*. Princeton University Press, New Jersey.
- Gödel, K. (1990). "Russell's Mathematical Logic". S. Feferman (Ed.). *Kurt Gödel – Collected Works*. Oxford University Press, New York, s.119-141.
- Guénon, R. (2004). *The Metaphysical Principles Of The Infinitesimal Calculus*. (İng. Çev. M. Allen ve H. D. Fohr). S. D. Fohr (Ed.). Sophia Perennis, P.O. Hillsdale NY.
- Hacking, I. (1973). "Leibniz and Descartes: Proof and Eternal Truths". *Dawes Hicks Lecture on Philosophy*, 175-188.
- Hacking, I. (1988). "Locke, Leibniz, Language and Hans Aarsleff". *Thought and Language in the Philosophy of the Enlightenment*, 75(2), 135-153.
- Hailperin, T. (2004). "Algebraical Logic 1685-1900". *Handbook of The History of Logic, The Raise of Modern Logic: From Leibniz To Frege*. Gabbay, D. M. ve Woods J. (Ed.). Elsevier B.V. Holland, 3, 334-399.
- Hall, A. R. (2002). *Philosophers at War – The Quarrel Between Newton and Leibniz*. Cambridge University Press, Cambridge.
- Hazar, Z. (2017). *Dilbilim ve Matematik İlişkisinde Saussure, Gödel, Popper*. Yayımlanmamış Yüksek Lisans Tezi. Akdeniz Üniversitesi Sosyal Bilimler Enstitüsü, Antalya.
- Hazar, Z. (9 Mayıs 2022). Müge Erel-Özçevik ile görüntülü görüşme, İstanbul-Manisa.

- Hızır, N. (1945). “Yeni Mantığın Öncüsü Leibniz”. *Ankara Üniversitesi Dil Tarih Coğrafya Fakültesi Dergisi*, 3(4), 433-440.
- Hilbert, D. (1900). “Mathematical Problems”. (İng. Çev. M. W. Newson). *The International Congress of Mathematicians*. Paris, s.437-479.
- Hilbert, D. ve Ackermann, W. (1950). *Principles of Mathematical Logic*. R. E. Luce (Ed.). (İng. Çev. L. M. Hammond, G. G. Leckie ve F. Steinhardt). Chelsea Publishing Company, New York.
- Hobbes, T. (1996). *Leviathan*. J. C. A. Gaskin (Ed.). Oxford University Press Inc., New York.
- Hobbes, T. (2007). *Leviathan (6. Baskı)*. (Çev. S. Lim). Yapı Kredi Yayınları, İstanbul.
- Hobbes, T. (2018). *De Cive-Yurttaşlığın Felsefi Temelleri (3. Baskı)*. (Çev. C. D. Zarakoğlu). Belge Yayınları, İstanbul.
- Hodges, A. (2012). *Alan Turing: The Enigma*. Princeton University Press, New Jersey.
- Hopcroft, J. E. Motwani, J. ve Ullman, J. D. (2001). *Introduction to Automata Theory, Languages, and Computation. (2. Baskı)*. Addison-Wesley, USA.
- Hopper, G. M. (1981). “The Opening Session”. *History of Programming Language*. R. L. Wexelblat (Ed.). Academic Press, New York, s.7-20.
- Israel, D. (1991). “A Short Sketch of the Life and Career of John McCarthy”. *Artificial Intelligence and Mathematical Theory of Computation Papers in Honor of John McCarthy*. Lifschitz, V. (Ed.). Academic Press, Inc., San Diego.
- Jolley, N. (2004). *The Cambridge Companion to Leibniz*. Cambridge University Press, Cambridge.
- Jones, M. L. (2006). *The Good Life in the Scientific Revolution-Descartes, Pascal, Leibniz and the Cultivation of Virtue*. The University of Chicago Press Chicago & London.
- Kabanda, G. (2019). “An Introductory Essay”. *On the Theoretical Foundations of Computer Science*. GRIN Verlag, Munich. s.1-18.
- Kahveci, K. (2011). “Onyedinci Yüzyıl’da Bir Metafizik Mantığı: Özne-Yüklem İlişkisi”. *Sosyal Bilimler Dergisi*, 11(47), 189-203.
- Kahveci, K. (2012). *Gottfried Wilhelm Leibniz Felsefesinde Bilgi Teorisi ve Mantık*. Berikan Yayınevi, Ankara.
- Kamer, V. (2015). “Yapılandırılmış Sorgu Dili (SQL) ve Üç Değerli Mantık”. *Felsefe Arkivi*, 42(2015/I), 41-48.
- Kaya, M. (2014a). *Leibniz’de Kötülük Problemi ve Teodise*. Türkiye Alim Kitapları, İzmir.
- Kaya, M. (2014b). “Leibniz’de Kötülük Problemi ve Mümkün Dünyaların En İyisi Teodisesi”. *Doğu-Batı Düşünce Dergisi*, 17(70), 53-78.

- Kayabaş, İ. (2020). “Bilgisayar Programlama”. *Bilişim Teknolojileri*. Anadolu Üniversitesi Yayınları, Eskişehir, s.199-226.
- Kenny, A. (2006). *An Illustrated Brief History of Western Philosophy*. Blackwell Publishing Ltd., Oxford.
- Kenny, A. (2007). *Philosophy in the Modern World. (Vol. 4)*. Clarendon Press, Oxford.
- Kleene, S. C. (1986). “Introductory Note to 1934”. *Kurt Gödel – Collected Works I*. S, Feferman (Ed.). Oxford University Press, New York, s. 338-345.
- Kline, M. (1972a). *Mathematical Thought From Ancient to Modern Times – Vol.1*. Oxford University Press, Oxford ve New York.
- Kline, M. (1972b). *Mathematical Thought From Ancient to Modern Times – Vol.2*. Oxford University Press, Oxford ve New York.
- Kline, M. (1972c). *Mathematical Thought From Ancient to Modern Times – Vol.3*. Oxford University Press, Oxford ve New York.
- Leibniz, G. W. (1969a). “Dissertation on the Art of Combinations”. *Philosophical Papers and Letters. (2. Baskı)*. (İng. Çev. L. E. Loemker). Loemker, L. E. (Ed.). D. Reidel Publishing Co., Dordrecht, s.73-84.
- Leibniz, G. W. (1969b). “A New Method for Learning and Teaching Jurisprudence”. *Philosophical Papers and Letters. (2. Baskı)*. (İng. Çev. L. E. Loemker). Loemker, L. E. (Ed.). D. Reidel Publishing Co., Dordrecht, s.85-92.
- Leibniz, G. W. (1969c). “Letter to Henry Oldenburg”. *Philosophical Papers and Letters. (2. Baskı)*. (İng. Çev. L. E. Loemker). Loemker, L. E. (Ed.). D. Reidel Publishing Co., Dordrecht, s.165-166.
- Leibniz, G. W. (1969ç). “On The General Characteristic”. *Philosophical Papers and Letters. (2. Baskı)*. (İng. Çev. L. E. Loemker). Loemker, L. E. (Ed.). D. Reidel Publishing Co., Dordrecht, s.221-228.
- Leibniz, G. W. (1969d). “On Universal Synthesis and Analysis, or the Art of Discovery and Judgement”. *Philosophical Papers and Letters. (2. Baskı)*. (İng. Çev. L. E. Loemker). Loemker, L. E. (Ed.). D. Reidel Publishing Co., Dordrecht, s.229-234.
- Leibniz, G. W. (1969e). “Two Studies In The Logical Calculus”. *Philosophical Papers and Letters. (2. Baskı)*. (İng. Çev. L. E. Loemker). Loemker, L. E. (Ed.). D. Reidel Publishing Co., Dordrecht, s.235-247.
- Leibniz, G. W. (1969f). “Studies In A Geometry Of Situation With A Letter To Christian Huygens”. *Philosophical Papers and Letters. (2. Baskı)*. (İng. Çev. L. E. Loemker). Loemker, L. E. (Ed.). D. Reidel Publishing Co., Dordrecht, s.248-258.

- Leibniz, G. W. (1969g). "Letter To John Frederick, Duke Of Brunswick-Hanover". *Philosophical Papers and Letters. (2. Baskı)*. (İng. Çev. L. E. Loemker). Loemker, L. E. (Ed.). D. Reidel Publishing Co., Dordrecht, s.259-262.
- Leibniz, G. W. (1969ğ). "On Freedom". *Philosophical Papers and Letters. (2. Baskı)*. (İng. Çev. L. E. Loemker). Loemker, L. E. (Ed.). D. Reidel Publishing Co., Dordrecht, s.263-266.
- Leibniz, G. W. (1969h). "Meditations on Knowledge, Truth, and Ideas". *Philosophical Papers and Letters. (2. Baskı)*. (İng. Çev. L. E. Loemker). Loemker, L. E. (Ed.). D. Reidel Publishing Co., Dordrecht, s.291-295.
- Leibniz, G. W. (1969ı). "A Study In The Logical Calculus". *Philosophical Papers and Letters. (2. Baskı)*. (İng. Çev. L. E. Loemker). Loemker, L. E. (Ed.). D. Reidel Publishing Co., Dordrecht, s.371-382.
- Leibniz, G. W. (1969i). "Letter To Gabriel Wagner On The Value Of Logic". *Philosophical Papers and Letters. (2. Baskı)*. (İng. Çev. L. E. Loemker). Loemker, L. E. (Ed.). D. Reidel Publishing Co., Dordrecht, s.462-471.
- Leibniz, G. W. (1969j). "Letter To Varignon, With A Note On The 'Justification Of The Infinitesimal Calculus By That Of Ordinary Algebra". *Philosophical Papers and Letters. (2. Baskı)*. (İng. Çev. L. E. Loemker). Loemker, L. E. (Ed.). D. Reidel Publishing Co., Dordrecht, s.542-546.
- Leibniz, G. W. (1969k). "Letters To Nicola Remond". *Philosophical Papers and Letters. (2. Baskı)*. (İng. Çev. L. E. Loemker). Loemker, L. E. (Ed.). D. Reidel Publishing Co., Dordrecht, s.654-660.
- Leibniz, G. W. (1989a). "Letter to Foucher". *Philoshical Essays*. (İng. Çev. R. Ariew ve D Garber). R. Ariew ve D Garber (Ed.). Hackett Publishing Company, Indianapolis & Cambridge, s.1-5.
- Leibniz, G. W. (1989b). "Preface to Universal Characteristic". *Philoshical Essays*. (İng. Çev. R. Ariew ve D Garber). R. Ariew ve D Garber (Ed.). Hackett Publishing Company, Indianapolis & Cambridge, s.5-10.
- Leibniz, G. W. (1989c). "Samples Of The Numerical Characteristic". *Philoshical Essays*. (İng. Çev. R. Ariew ve D Garber). R. Ariew ve D Garber (Ed.). Hackett Publishing Company, Indianapolis & Cambridge, s.10-18.
- Leibniz, G. W. (1989ç). "On Freedom and Possibility". *Philoshical Essays*. (İng. Çev. R. Ariew ve D Garber). R. Ariew ve D Garber (Ed.). Hackett Publishing Company, Indianapolis & Cambridge, s.19-23.

- Leibniz, G. W. (1989d). "Meditations on Knowledge, Truth, and Ideas". *Philosophical Essays*. (İng. Çev. R. Ariew ve D Garber). R. Ariew ve D Garber (Ed.). Hackett Publishing Company, Indianapolis & Cambridge, s.23-27.
- Leibniz, G. W. (1989e). "On Copernicanism and the Relativity of Motion". *Philosophical Essays*. (İng. Çev. R. Ariew ve D Garber). R. Ariew ve D Garber (Ed.). Hackett Publishing Company, Indianapolis & Cambridge, s.90-94.
- Leibniz, G. W. (1989f). "Comment on Spinoza's Philosophy". *Philosophical Essays*. (İng. Çev. R. Ariew ve D Garber). R. Ariew ve D Garber (Ed.). Hackett Publishing Company, Indianapolis & Cambridge, s.272-281.
- Leibniz, G. W. (1996). *New Essays on Human Understanding*. (İng. Çev. P. Remnant ve J. Bennett). P. Remnant ve J. Bennett (Ed.). Cambridge University Press, Glasgow.
- Leibniz, G. W. (1999). *Metafizik Üzerine Konuşma*. (Çev. A. Timuçin). Cumhuriyet Yayınları, İstanbul.
- Leibniz, G. W. (2007a). *Theodicy-Essays on the Goodness of God, the Freedom of Man and the Origin of Evil*. (İng. Çev. A. Farrer). BiblioBazaar, Oxford.
- Leibniz, G. W. (2011). *Monadoloji-Metafizik Üzerine Konuşma*. (Çev. A. Altınörs). Doğu Batı Yayınları, Ankara.
- Lewis, C. I. (1918). *A Survey of Symbolic Logic*. University of California Press, Berkeley.
- Locke, J. (1999). *İnsanın Anlama Yetisi Üzerine Bir Deneme III- IV*. (Çev. M. Delikara Topçu). Öteki Yayınevi, Ankara.
- Locke, J. (2000). *İnsanın Anlama Yetisi Üzerine Bir Deneme I - II. (2. Baskı)*. (Çev. M. Delikara Topçu). Öteki Yayınevi, Ankara.
- Loemker, L. E. (1969). *Philosophical Papers and Letters. (2. Baskı)*. (İng. Çev. L. E. Loemker). Loemker, L. E. (Ed.). D. Reidel Publishing Co., Dordrecht. s.1-70, 73, 248.
- Martin, J. C. (2011). *Introduction to Languages and The Theory of Computation*. The McGraw-Hill Companies, Inc., New York.
- McCarthy, J. Minsky, M. L. Rochester, N. ve Shannon, C. E. (2006). "A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence". *AI Magazine Volume 27,4*, s.12-14.
- McCorduck, P. (2004). *Machines Who Think: A Personal Inquiry into the History and Prospects of Artificial Intelligence*. A K Peters, Ltd., Canada.
- MacHale, D. (2014). *The Life and Work of George Boole- A Prelude to the Digital Age*. Cork University Press, Cork, Ireland.

- Mates, B. (1986). *The Philosophy of Leibniz-Metaphysics and Language*. Oxford University Press Inc., New York.
- Michaelson, G. (2020). "Programming Paradigms, Turing Completeness and Computational Thinking". *The Art, Science, and Engineering of Programming*. s.1-21.
- Millán, J. B. S. (2021). "Lingua Characterica and Calculus Ratiocinator: The Leibnizian Background of the Frege-Schröder Polemic". *The Review of Symbolic Logic*, s.411-446.
- Mugnai, M. (2010). "Logic and Mathematics in the Seventeenth Century". *History and Philosophy of Logic*, 31(4), 297-314.
- Mutlu, M. E. (2020). "Bilişim Teknolojilerinin Gelişimi". *Bilişim Teknolojileri*. Anadolu Üniversitesi Yayınları, Eskişehir, s.3-30.
- Nagel, E. ve Newman, J. R. (2010). *Gödel Kanıtlanması*. (3. Baskı). (Çev. B. Gözkan). Boğaziçi Üniversitesi Yayınevi, İstanbul.
- Nesin, A. (2010). *Sezgisel Kümeler Kuramı*. (3. Baskı). Nesin Yayıncılık Ltd. Şti., İstanbul.
- Nesin, A. (2020). *Aksiyomatik Kümeler Kuramı II. Ordinaler, Kardinaler ve Seçim Aksiyomu*. Nesin Yayıncılık Ltd. Şti., İstanbul.
- Newton, I. (1998). *Doğal Felsefenin Matematiksel İlkeleri (Seçmeler)-Optik (Seçmeler)*. (Çev. A. Yardımlı). İdea Gençlik Arşivi, İstanbul.
- Nielsen, M. A. (2015). *Neural Networks and Deep Learning*. Determination Press.
- Nilsson, N. (2019). *Yapay Zekâ Geçmişi ve Geleceği*. (2. Baskı). (Çev. M. Doğan). Boğaziçi Üniversitesi Yayınevi, İstanbul.
- Nourani, C. F. (2014). *A Functorial Model Theory: Newer Applications to Algebraic Topology, Descriptive Sets, and Computing Categories Topos*. Apple Academic Press ve Taylor&Francis Group, Toronto.
- O'Regan, G. (2016). *Introduction to the History of Computing-A Computing History Primer*. Springer International Publishing, Switzerland.
- Owens, J. D. Houston, M. Luebke, D. Green, S. Stone, J. E. ve Phillips, J. C. (2008). "GPU Computing". *Proceedings of the IEEE*. 96(5), 879-899.
- Özlem, D. (2004). *Mantık, Klasik/Sembolik Mantık, Mantık Felsefesi*. (7. Baskı). İnkılap Kitabevi, İstanbul.
- Öztürk, A. B. (2011). *Kurt Gödel'in Eksiklik Teoremleri ve Platonculuğu Üzerine Felsefi Bir İnceleme*. Yayımlanmamış Yüksek Lisans Tezi. Pamukkale Üniversitesi Sosyal Bilimler Enstitüsü, Denizli.

- Öztürk, A. B. (2015). “İkinci Seviye Mantığın Mantıksallığı: İlgili Tartışmalar Üzerine Eleştirel bir Değerlendirme”. *International Journal of Human Sciences*, 12(2), 1712-1733.
- Öztürk, A. B. (2017). *Çağdaş Mantık, Matematik ve Bilgi Felsefelerinde A Priori Gereçlendirme Sorunu*. Yayınlanmamış Doktora Tezi. Akdeniz Üniversitesi Sosyal Bilimler Enstitüsü, Antalya.
- Öztürk, Z. (2017). “George Boole’ün Cebirsel Mantığı ve Mantık Tarihindeki Yeri”. Kamer, V. ve Ural, Ş. (Yayına haz.). *VII. Mantık Çalıştayı Kitabı, Mantık Derneği Yayınları, İstanbul*, s.571-584.
- Peano, G. (1973a). “The Principles of Arithmetic, Presented by a New Method”. *Selected Works of Giuseppe Peano*. (İng. Çev. H. C. Kennedy). H. C. Kennedy (Ed.). George Allen & Unwin Ltd., London.
- Peano, G. (1973b). “Studies in Mathematical Logic”. *Selected Works of Giuseppe Peano*. (İng. Çev. H. C. Kennedy). H. C. Kennedy (Ed.). George Allen & Unwin Ltd., London.
- Peano, G. (1973c). “Definitions in Mathematics”. *Selected Works of Giuseppe Peano*. (İng. Çev. H. C. Kennedy). H. C. Kennedy (Ed.). George Allen & Unwin Ltd., London.
- Peckhaus, V. (1999). “19th Century between Philosophy and Mathematics”. *Bulletin of Symbolic Logic*, 5(4), 433-450.
- Peckhaus, V. (2004). “Calculus Ratiocinator vs. Characteristica Universalis? The Two Traditions in Logic, Revisited”. *History and Philosophy of Logic*, s.1-16.
- Peirce, C. S. (1983). *Studies in Logic by Members of the Johns Hopkins University (1883)*. C. S. Peirce (Ed.). John Benjamins Publishing Company Amsterdam/Philadelphia.
- Peirce, C. S. (1992). *Reasoning and the Logic of Things-The Cambridge Conferences Lectures of 1898*. K. L. Ketner (Ed.). Harvard University Press, Cambridge.
- Peirce, C. S. (1994). *The Collected Papers of Charles Sanders Peirce*. C. Hartshorne ve P. Weiss (Ed.). Harvard University Press, Cambridge.
- Peirce, C. S. (2010). *Philosophy of Mathematics-Selected Writings*. M. E. Moore (Ed.). Indiana University Press, Bloomington.
- Perlis, A. J. (1981a). “The American Side of the Development of ALGOL”. *History of Programming Language*. R. L. Wexelblat (Ed.). Academic Press, New York, s.75-91.
- Perlis, A. J. (1981b). “Transcripts of Presentations”. *History of Programming Language*. R. L. Wexelblat (Ed.). Academic Press, New York, s. 139-147.

- Remnant, P. ve Bennett J. (1996). *New Essays on Human Understanding*. (İng. Çev. P. Remnant ve J. Bennett). P. Remnant ve J. Bennett (Ed.). Cambridge University Press, Glasgow. s.vii-xxx.
- Rescher, N. (1981a). "Leibniz and the Plurality of Space - Time Frameworks". *Leibniz's Metaphysics of Nature, The University of Western Ontario Series in Philosophy of Science; Vol 18*. D. Reidel Publishing Company, Dordrecht, Hollanda, s. 84-100.
- Rescher, N. (1981b). "The Contributions of the Paris Period (1672-76) To Leibniz's Metaphysics". *Leibniz's Metaphysics of Nature, The University of Western Ontario Series in Philosophy of Science; Vol 18*. D. Reidel Publishing Company, Dordrecht, Holland, s. 101-116.
- Risi V. D. (2007). *Geometry and Monadology – Leibniz's Analysis Situs and Philosophy of Space*. Historical Studies, Science Networks Vol. 33, Basel, Switzerland.
- Russell, B. (1920). *Intoduction to Mathematical Philosophy*. (2. Baskı). George Allen & Unwin Ltd., London.
- Russell, B. ve Whitehead, A. N. (1963). *Principia Mathematica*. (Vol. I). (2. Baskı). Cambridge University Press, Cambridge.
- Russell, B. (2005). *A Critical Exposition of the Philosophy of Leibniz*. Taylor & Francis Group, London, New York.
- Russell, B. (2009). *The Philosophy of Logical Atomism*. Taylor & Francis Group, London, New York.
- Sammet, J. E. (1981a). "Organization of the Conference". *History of Programming Language*. R. L. Wexelblat (Ed.). Academic Press, New York, s. xvii-xx.
- Sammet, J. E. (1981b). "Introduction of Captain Grace Murray Hopper". *History of Programming Language*. R. L. Wexelblat (Ed.). Academic Press, New York, s.5-7.
- Sammet, J. E. (1981c). "The Early History of COBOL". *History of Programming Language*. R. L. Wexelblat (Ed.). Academic Press, New York, s.199-243.
- Say, C. (2019). *50 Soruda Yapay Zekâ*. (13. baskı). 7 Renk Basın Yayım ve Filmcilik Ltd., İstanbul.
- Scaglione, A. (1990). "The Origins of Syntax: Descartes or the 'Modistae'?". *History and Historiography of Linguistics Vol I*. Niederehe, H. J. ve Koerner, E. F. K. (Ed.). John Benjamin Publishing Company, Amsterdam, s. 339-346.
- Searle, J. (2003). *Minds, Brains and Science*. (13. Baskı). Harvard University Press Cambridge, Massachusetts.

- Sebesta, R. W. (2016). *Concepts of Programming Languages. (11. Baskı)*. Pearson Education Limited, Essex.
- Shannon, C. E. (1938). "A Symbolic Analysis of Relay and Switching Circuits". *Transactions American Institute of Electrical Engineers*, 57, 471-495.
- Shannon, C. E. (1948). "A Mathematical Theory of Communication". *The Bell System Technical Journal*, 27, 379-423, 623-656.
- Smith, G. C. (1982). *The Boole-De Morgan Correspondence (1842-1864)*. Oxford University Press, New York.
- Spinoza, B. (2012). *Ethica-Geometrik Yöntemle Kanıtlanmış ve Beş Bölüme Ayrılmış Ahlak*. (Çev. Ç. Dürüşken). Kabalcı Yayınevi, İstanbul.
- Strasser, G. F. (2007). "The Rise of Cryptology in the European Renaissance". *The History of Information Security: A Comprehensive Handbook*. de Leeuw, K. ve Bergstra, J. (Ed.). Elsevier, B.V.
- Subbiondo, J. L. (1990). "John Wilkins' Theory of the Origin and Development of Language - Historical linguistics in 17th-century Britain". *History and Historiography of Linguistics Vol I*. Niederehe, H. J. ve Koerner, E. F. K. (Ed.). John Benjamin Publishing Company, Amsterdam, s. 357-365.
- Swoyer, C. (1984). "Leibniz's Calculus of Real Addition". *Studia Leibnitiana*, XXVI(1994), 1-30.
- Tegmark, M. (2019). *YAŞAM 3.0 – Yapay Zekâ Çağında İnsan Olmak*. (Çev. E. C. Göksoy). Pegasus Yayınları, İstanbul.
- Timuçin, A. (1999). *Metafizik Üzerine Konuşma*. (Çev. A. Timuçin). Cumhuriyet Yayınları, İstanbul, s.11-68.
- Tucker, A. B. ve Noonan, E. R. (2007). *Programming Languages: Principles and Paradigms (2. Baskı)*. McGraw-Hill Education, Singapore.
- Turing, A. M. (1936). "On Computable Numbers, With an Application to the Entscheidungsproblem". *Proceedings of the London Mathematical Society, Ser. 2*, s.230-265.
- Turing, A. M. (1950). "Computing Machinery and Intelligence". *Mind* 49, s.433-460.
- van Heijenoort, J. (1967). *From Frege to Gödel A Search Book in Mathematical Logic, 1879-1931*. Harvard University Press, Cambridge, Massachusetts.
- Venn, J. (1881). *Symbolic Logic*. Macmillan and Co., London.
- Wehle, H. D. (2017). "Machine Learning, Deep Learning, and AI: What's the Difference?". *Data Scientist Innovation Day*. s.1-6.

- Welch, J. R. (1990). "Lull, Leibniz, and the Logic of Discovery" *Catalan Review* IV, Nos. 1–2 (July–December 1990), 75–83
- Wittgenstein, L. (2013). *Tractatus Logico-Philosophicus*. (7. Basım). (Çev. O. Aruoba). Metis Yayınları, İstanbul.
- Yargan, D. (2017). "Formel Ontolojilerin Dinamikliği ve Esnekliği Üzerine". Kamer, V. ve Ural, Ş. (Yayına haz.). VII. Mantık Çalıştay Kitabı, Mantık Derneği Yayınları, İstanbul, s.697-703.

İnternet Kaynakları

- Aslan, K. (2006). *A'dan Z'ye C Klavuzu*.
<https://www.scribd.com/document/52654900/A-dan-Z-ye-C-Klavuzu-Kaan-Aslan>. (erişim tarihi: 24.01.2022).
- Bagaria, J. (2021). "Set Theory". *The Stanford Encyclopedia of Philosophy*. E. N. Zalta (Ed.),
<https://plato.stanford.edu/archives/win2021/entries/set-theory/>. (erişim tarihi: 17.01.2022).
- Barak, B. (2021). *Introduction to Theoretical Computer Science*.
<https://github.com/boazbk/tcs>. (erişim tarihi: 11.03.2022).
- Barnett, J. H. (2011). "Origins of Boolean Algebra in the Logic of Classes: George Boole, John Venn and C. S. Peirce". s. 1-35.
<https://www.cs.nmsu.edu/historical-projects/Projects/25520111217Boole-Venn-Peirce%20Intro%20to%20Boolean%20Algebra%20Project.pdf>. (erişim tarihi: 04.03.2021).
- Ben-David, S. Hrubeš, P. Moran, S. Shpilka, A. ve Yehudayoff, A. (2019). "Learnability Can Be Undecidable". *Nature Machine Intelligence*, 1, s.44–48.
<http://gtts.ehu.es/German/Docencia/1819/AC/extras/s42256-018-0002-3.pdf>. (erişim tarihi: 03.07.2022).
- Blanqui, F. (2013). "Introduction to the Coq Proof Assistant".
<https://who.rocq.inria.fr/Frederic.Blanqui/papers/iami13coq.pdf>. (erişim tarihi: 07.03.2022).
- Boole, G. (1854). *An Investigation of the Laws of Thought on Which are Founded the Mathematical Theories of Logic and Probabilities*.
<https://www.gutenberg.org/files/15114/15114-pdf>. (erişim tarihi: 04.11.2019).
- Buber, E. Diri, B. (2018). "Performance Analysis and CPU vs GPU Comparison for Deep Learning". *6th International Conference on Control Engineering & Information Technology*.

- https://www.researchgate.net/publication/334168063_Performance_Analysis_and_CPU_vs_GPU_Comparison_for_Deep_Learning. (erişim tarihi: 22.03.2022).
- Chomsky, N. (1956). “Three Models for the Description of Language”. <https://chomsky.info/wp-content/uploads/195609-.pdf>. (erişim tarihi: 19.02.2022).
- Couturat, L. (2004). *The Algebra of Logic*. (İng. Çev. L. G. Robinson). <https://www.gutenberg.org/files/10836/10836-pdf>. (erişim tarihi: 18.09.2020).
- Couturat, L. (2012). “The Universal Language”. *The Logic of Leibniz*. (İng. çev. D. Rutherford ve R. T. Monroe) s.1-18. <http://philosophyfaculty.ucsd.edu/faculty/rutherford/Leibniz/Couturatchapters/Chap3.pdf>. (erişim tarihi: 12.03.2022).
- Gödel, K. (1956). “Letter to von Neumann”. <https://www.anilada.com/notes/godel-letter.pdf>. (erişim tarihi: 21.03.2021).
- Heule, M. Kullmann, O. ve Marek, V. W. (2016). “Solving and Verifying the Boolean Pythagorean Triples Problem via Cube-and-Conquer”. <https://www.cs.utexas.edu/~marijn/publications/ptn.pdf>. (erişim tarihi: 07.03.2022).
- Hulkewicz, J. (2021). “The Axiom of Choice: The Last Great Controversy in Mathematics”. <https://www.preprints.org/manuscript/202103.0133/v1>. (erişim tarihi: 03.07.2022).
- Işıklı, Ş. (2014). “Büyük Veri, Epistemoloji ve Etik Tartışmalar”. *AJIT-e: Online Academic Journal of Information Technology*, 5:17, s.89-122. <https://dergipark.org.tr/tr/download/article-file/1114231>. (erişim tarihi: 04.07.2022).
- Kernighan, B. W. (1981). “Why Pascal is Not My Favorite Programming Language”. <https://www.cs.virginia.edu/~evans/cs655/readings/bwk-on-pascal.html#:~:text=Pascal%20was%20originally%20intended%20primarily,not%20suitable%20for%20serious%20programming>. (erişim tarihi: 24.02.2022).
- Koç, M. (2014). “Bilim Tarihinin İlk Yapma Dili Bâleybelen”. <https://www.ayk.gov.tr/wp-content/uploads/2015/01/KO%C3%87-Mustafa-B%C4%B0L%C4%B0M-TAR%C4%B0H%C4%B0N%C4%B0N-%C4%B0LK-YAPMA-D%C4%B0L%C4%B0-B%C3%82LEYBELEN.pdf>. (erişim tarihi: 12.12.2021). s.269-278.
- Kozen, D. ve Tseng, W. L. (2008). “The Böhm–Jacopini Theorem Is False, Propositionally”. *Proceedings of the 9th international conference on Mathematics of Program Construction*. https://www.researchgate.net/publication/225114059_The_Bohm-Jacopini_Theorem_Is_False_Propositionally. (erişim tarihi: 07.03.2022).

- Kurt, B. (2002). *C++ ve Nesneye Dayalı Programlama*.
https://web.itu.edu.tr/~bkurt/Courses/oop/lecture_notes.pdf. (erişim tarihi: 01.03.2022).
- Laporte, S. 2018. "Ideal Language." *Knowledge Organization* 45(7), 586-608.
https://web.itu.edu.tr/~bkurt/Courses/oop/lecture_notes.pdf. (erişim tarihi: 23.01.2022).
- Leibniz, G. W. (2007b). "An Explanation of Binary Arithmetic Using only the Characters 0 and 1, with Remarks about its Utility and the Meaning it Gives to the Ancient Chinese Figures of Fuxi". (İng. Çev. L. Strickland). <http://www.leibniz-translations.com/binary.htm>. (erişim tarihi: 16.11.2021).
- Libert, A. R. (2018). "Artificial Languages". *Oxford Research Encyclopedia of Linguistics*.
<https://oxfordre.com/linguistics/view/10.1093/acrefore/9780199384655.001.0001/acrefore-9780199384655-e-11>. (erişim tarihi: 14.04. 2022).
- McCarthy, J. (1990) "Artificial Intelligence, Logic and Formalizing Common Sense".
<http://jmc.stanford.edu/articles/ailogic.html>. (erişim tarihi: 18.01.2022).
- Paleo, B. W. (2016). "Leibniz's Characteristica Universalis and Calculus Ratiocinator Today". *300 Years After Leibniz*. C. Tandy (Ed.).
https://www.researchgate.net/publication/311456139_Leibniz's_Characteristica_Universalis_and_Calculus_Ratiocinator_Today. (erişim tarihi: 07.03.2019).
- Paulson, L. C. (1998). "Isabelle: The Next 700 Theorem Provers".
<https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-143.pdf>. (erişim tarihi: 08.03.2022).
- Peckhaus, V. (2018). "Leibniz's Influence on 19th Century Logic".
<https://plato.stanford.edu/entries/leibniz-logic-influence/>. (erişim tarihi: 18.09.2019).
- Phillips, D. (2013). "George Peacock and the Development of British Algebra 1800-1840".
<https://www.maths.cam.ac.uk/opportunities/careers-for-mathematicians/summer-research-mathematics/files/Phillips.pdf>. (erişim tarihi: 28.11.2021), s.1-24.
- Pickard, G. (2020). "Programming Languages Shouldn't and Needn't be Turing Complete".
https://www.gabrielpickard.com/turing-incomplete_edited.pdf. (erişim tarihi: 24.01.2022), s.1-6.
- Russell, B. (1945). *The History of Western Philosophy*.
<http://www.ntslibrary.com/PDF%20Books/History%20of%20Western%20Philosophy.pdf>. (erişim tarihi: 13.01.2020).

- Saffidine, A. (2014). “The Game Description Language is Turing Complete”.
<https://cgi.cse.unsw.edu.au/~abdallahs/Papers/2014/The%20Game%20Description%20Language%20is%20Turing%20Complete.pdf>. (erişim tarihi: 06.03.2022).
- Şamlı, R. (2020). *Programlama I-Endüstri Mühendisliği Lisans Tamamlama Programı*.
http://auzefkitap.istanbul.edu.tr/kitap/endustrimuhlt_ue/proglamlama1.pdf. (erişim tarihi: 20.02.2022).
- Uzun, E. (2022a). “Başlıca Programlama Dillerinin Gelişimi”.
https://adys.nku.edu.tr/OpenCourse/Course/Programlama_Dilleri_Prensipeleri/27702.
(erişim tarihi: 28.01.2022).
- Uzun, E. (2022b). “Fonksiyonel Programlama Dilleri (Functional Programming Languages)”.
https://adys.nku.edu.tr/OpenCourse/Course/Programlama_Dilleri_Prensipeleri/27702.
(erişim tarihi: 18.03.2022).
- Uzun, E. (2022c). “Sözdizimi (Syntax) ve Anlambilim (Semantics) Tanımlama”.
https://adys.nku.edu.tr/OpenCourse/Course/Programlama_Dilleri_Prensipeleri/27702.
(erişim tarihi: 18.03.2022).
- Yanmaz, H. (2018). “Lojik Kapılar”.
http://web.hitit.edu.tr/dersnotlari/hilmiyanmaz_20.03.2018_8I5U.pdf. (erişim tarihi: 16.02.2022)

ÖZGEÇMİŞ

Adı ve SOYADI	Zuhal HAZAR
EĞİTİM DURUMU	
Mezun Olduğu Lise	Yunus Emre Lisesi
Lisans Diploması	Dicle Üniversitesi, Fen-Edebiyat Fakültesi, Matematik Bölümü, Diyarbakır, 2005
Yüksek Lisans Diploması	Akdeniz Üniversitesi, Sosyal Bilimler Enstitüsü, Felsefe Ana Bilim Dalı, Antalya, 2017
Tez Konusu	Dilbilim ve Matematik İlişkisinde Saussure, Gödel, Popper
Yabancı Dil / Diller	İngilizce
İŞ DENEYİMİ	
Çalıştığı Kurumlar	Ticaret Bakanlığı 2009-...