EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

(DOKTORA TEZİ)

SİLLOJİSTİK AKIL YÜRÜTMENİN BİLGİSAYARDA MODELLENMESİ ÜZERİNE

Selçuk TOPAL

Tez Danışmanı: Doç. Dr. Tahsin ÖNER

Matematik Anabilim Dalı

Bilim Dalı Kodu: 403.05.01
Sunuş Tarihi: 16.01.2015

Bornova-İZMİR
2015

Selçuk TOPAL tarafından **Doktora Tezi** olarak sunulan "**Sillojistik Akıl Yürütmenin Bilgisayarda Modellenmesi Üzerine**" başlıklı bu çalışma E.Ü. Lisansüstü Eğitim ve Öğretim Yönetmeliği ile E.Ü. Fen Bilimleri Enstitüsü Eğitim ve Öğretim Yönergesi'nin ilgili hükümleri uyarınca tarafımızdan değerlendirilerek savunmaya değer bulunmuş ve 16.01.2015 tarihinde yapılan tez savunma sınavında aday oybirliği/oyçokluğu ile başarılı bulunmuştur.

Jüri Üyeleri:		**İmza**
Jüri Başkanı	: Doç. Dr. Tahsin ÖNER	.....
Raportör Üye	: Prof. Dr. Urfat NURİYEV	.....
Üye	: Prof. Dr. Mehmet TERZİLER	.....
Üye	: Prof. Dr. Şaban EREN	.....
Üye	: Doç. Dr. Vecdi AYTAÇ	.....

EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ
ETİK KURALLARA UYGUNLUK BEYANI

E.Ü. Lisansüstü Eğitim ve Öğretim Yönetmeliğinin ilgili hükümleri uyarınca Doktora Tezi olarak sunduğum "Sillojistik Akıl Yürütmenin Bilgisayarda Modellenmesi Üzerine" başlıklı bu tezin kendi çalışmam olduğunu, sunduğum tüm sonuç, doküman, bilgi ve belgeleri bizzat ve bu tez çalışması kapsamında elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara atıf yaptığımı ve bunları kaynaklar listesinde usulüne uygun olarak verdiğimi, tez çalışması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını, bu tezin herhangi bir bölümünü bu üniversite veya diğer bir üniversitede başka bir tez çalışması içinde sunmadığımı, bu tezin planlanmasından yazımına kadar bütün safhalarda bilimsel etik kurallarına uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul edeceğimi beyan ederim.

.... / .... / 20..

İmzası

Adı-Soyadı

ÖZET

SİLLOJİSTİK AKIL YÜRÜTMENİN BİLGİSAYARDA MODELLENMESİ ÜZERİNE

TOPAL, Selçuk

Doktora Tezi, Matematik Anabilim Dalı
Tez Danışmanı: Doç. Dr. Tahsin ÖNER
16 Ocak 2015, 60 sayfa

Bu tezin amacı, Doğal Lojik Programı kapsamında İngilizce doğal diline ait bazı temel sillojistik akıl yürütmelerin mantıksal türetimlerinin ve karşıt modellerinin, Python programlama dili kullanılarak İngilizce dilinin doğal yapısı çerçevesinde nasıl modüller oluşturulacağı üzerinde durmaktır.

Tez, altı bölümden oluşmaktadır.

Tez ile ilgili genel bilgilerin yer aldığı giriş bölümünden sonra, önbilgiler bölümünde temel kavramlara yer verildi.

Üçüncü bölümde, sillojizm kavramının doğuşu, gelişimi, klasik ve modern yorumlarına yer verildi. Ayrıca, klasik ve modern sillojizmler arasındaki farklılıklar ve benzerlikler verildi.

Dördüncü bölümde, temel sillojistik lojikler ve kardinalite ve kesişen sıfatlardan oluşan sillojistik lojikler hakkında lojiksel ve cebirsel bilgiler verildi.

Beşinci bölümde, önceki bölümde bahsedilen sillojistik lojiklerin türetim algoritmaları, doğal dildeki gramer yapıları, bilgisayarda karşılık gelen veri yapıları, sonsuz döngüden nasıl kurtarılacağı ve modüllerin girilen veri sayısına karşılık ne kadar zamanda sonlacanağı hakkında bilgiler verildi.

Altıncı bölümde, tez boyunca elde edilen bilgilerin ve sonuçların bir özeti verildi.

Anahtar Kelimeler: Uygulamalı Mantık, Doğal Mantık, Doğal Dillerin Mantığı, Bilgisayar Biliminde Mantık, Mantık Uygulamaları.

ABSTRACT

ON MODELLING OF SYLLOGISTIC REASONING ON COMPUTER

TOPAL, Selçuk

Ph.D. Thesis, Mathematics Department
Supervisor: Doç. Dr. Tahsin ÖNER
16 January 2015, 60 pages

The aim of this thesis is to give information about how to create modules for counter-models and logical inferences of syllogistic reasonings in Natural English Language with the scope of Natural Logic Program by using Python programming language.

This thesis consists of six sections.

After the introduction chapter in which the general information concerning to the thesis is given, in the Preliminaries, fundamental and basic notions are presented.

In Section 3, genesis, rise, classical and modern interpretations of the syllogism notion are mentioned. Also, differences and similarities between classical and modern syllogisms are given.

In Section 4, logical and algebraic properties that are related to basic syllogistic logics and syllogistic logics that consist of cardinality and intersecting adjectives are given.

In Section 5, we present tools which are used for implementing syllogistic logics and show inference algorithms of the logics, structures of grammatical in the natural language, data structures of the logics corresponding to computer, how to avoid infinite loop and also how much time are needed to stop programs of created modules that depend on numbers of input data.

In Section 6, we final to the thesis with summary of the thesis.

Keywords: Applied Logic, Logic of Natural Languages, Logic in Computer Science, Applications of Logic.

TEŞEKKÜR

Lisans dönemimden başlayarak, yüksek lisans ve doktora tez danışmanlığımı üstlenerek, gerek çalışmalarımı yürütmem konusunda ve gerekse özel hayatımdan tez sürecime yansıyan sorunları aşmamda yardımlarını asla esirgemeyen Sayın Doç. Dr. Tahsin ÖNER'e, alanında sahip olduğu engin tecrübelerinden yararlanmama izin veren Sayın Prof. Dr. Mehmet TERZİLER'e ve Sayın Prof. Dr. Urfat NURİYEV'e, desteği, sevgisi ve değerli görüşleri ile her zaman yanımda olan aile bireylerim, Babam Cengiz TOPAL, Annem Fatma TOPAL, Kardeşlerim Gökhan TOPAL ve Hasan TOPAL'a, beraber geçirdiğimiz günler boyunca bir doktora öğrencisinin nasıl olması gerektiğini öğreten ve dil bilimleri üzerine olan bilgilerini benimle paylaşan İndiana Üniversitesi'nden Sayın Dr. Muhammad ABDULMAGEED'e, gereken bilgi, yardım, tavsiye ve önerileriye eksiksiz şekilde teorik ve uygulamalı lojiğin nasıl çalışılması gerektiğini bana aktaran İndiana Üniversitesi'nden Prof. Dr. Lawrence S. MOSS'a sonsuz teşekkürlerimi sunarım.

İÇİNDEKİLER

Sayfa

İÇİNDEKİLER (devam)

Sayfa

İÇİNDEKİLER (devam)

Sayfa

ŞEKİLLER DİZİNİ

Şekil Sayfa

TABLOLAR DİZİNİ

SİMGELER DİZİNİ

Simge	Açıklama
S	Temel sillojistik lojik
$S^{\dagger}$	İsim düzey tümleme içeren temel sillojistik lojik
$\mathcal{CARD}$	Küme kardinalitesi karşılaştırması içeren sillojistik lojik
$\mathcal{CARD} + IA$	Kesişen sıfatlar içeren $\mathcal{CARD}$
χ	Ex falso quadlibet kuralı
$\vdash$	Mantıksal türetim
$\models$	Modelde doğruluk
$O()$	Üst sınır zaman sınıfı
$:\Leftrightarrow$	Eğer ve yalnız eğer

1. GİRİŞ

Aristo'dan (MÖ 384 – 7 Mart MÖ 322) bu yana özellikle son 50 yıl içerisinde yoğun olarak sillojizm üzerinde pek çok fikir ortaya atılmıştır ve halen devam etmektedir (Łukasiewicz, J. ,1957), (Rose, L. E., 1968), (Smiley, T. J., 1973), (Thom, P., 1981). Bu anlamda sillojizm, kökü tarihsel açıdan çok derinlere uzanan ve her dönemde kendisinden çok farklı disiplinlere ait alanlarda bahsettiren konusu, (Khemlani, S. and Johnson-Laird, P., 2012), (Li, Z., 2008) onu insanın anlamlandırmaya çalıştığı doğadaki varlıkların çokluğunu kıyaslayan önemli bir kavram yapar (Thom, P., 2007).

Temel anlamda Aristo ve Łukasiewicz sillojizme farklı açılardan bakar, en azından Łukasiewicz sillojizmi daha formel bir yapı içerisinde görmek ister. Bu doğrultuda, Łukasiewicz'in bakış açısı daha yapısalcıdır ve şunu ifade eder: "Mantık bir bitki veya insan bilimi değildir, o tamamen bu nesnelerden başka diğer neslere uygulanabilirdir". Łukasiewicz'e göre Aristo'nun sistemi bir çıkarsamadır ve bunun kanıtı ona göre Aristo'nun çıkarsama cümlesinin 'therefore' ile başlamasıdır (Łukasiewicz, J. ,1957).

Bu tez, klasik ve modern sillojizm üzerindeki düşünceleri İngilizce doğal dilinin gramer özellikleri dikkate alınarak bilgisayar ortamına aktarılması amacıyla oluşturulduğu için Aristo'nun klasik sillojizmi, sillojizme modern bakış açıları, İngilizce dilinin özellikleri, türetimlerin algoritmik özellikleri ve uygulamaların bilgisayar üstündeki teknik ölçümleri mercek altına alınacaktır. Bu çalışma doğal lojik, metinsel çıkarımlar, bilgi çıkarımı alanlarına katkı sağlayacaktır.

2. ÖN BİLGİLER

Bu bölümde, tezin okunabilirliğini kolaylaştırmak amacıyla bazı temel kavramlar verilmiştir.

Tanım 2.1 (Miller, G. A., 1995) İki öncülden, tümdengelimsel akıl yürütme yoluyla bir hüküm türetilmesine ***sillojizm*** denir.

Tanım 2.2 (Iyanaga, S. and Kawada, Y., 1980) Bir G ***grafı*** ya da bir ***basit graf***, bir V tepeler kümesi ve V nin 2-li alt kümelerinden oluşan E ayrıtlar kümesinden oluşan bir $G = (V, E)$ sıralı ikilisidir.

Tanım 2.3 (Bang-Jensen, J. and Gregory G., 2000) Bir G ***yönlü graf*** , V nin sıralı 2-li alt kümelerinden oluşan E ayrıtlar kümesini içeren bir $G = (V, E)$ sıralı ikilisidir.

Tanım 2.4 (Champin, P. A. and Solnon, C. , 2003) Bir ***etiketli yönlü graf*** aşağıdaki özellikleri sağlayan bir $G = \langle V, r_V, r_E \rangle$ üçlüsüdür:

- Tepelerin sonlu bir V kümesi,

- v_i tepesi ve l etiketine sahip, (v_i, l) ikililerinin kümesi olan bir $r_V \subseteq V \times L_V$ bağıntısı,

- Bir $r_E \subseteq V \times V \times L_E$ bağıntısı (v_i, v_j, l), (v_i, v_j) ikilileri ve l etiketine sahip üçlülerin bir kümesidir.

Tanım 2.5 Bir ***kısıtlanmış etiketli yönlü graf (KEYG)***, bir etiketli yönlü grafın V tepeler kümesi, L etiketler kümesi ve r_E etiketli yönleri dikkate alınan kısmıdır.

Örnek 2.6 $V = \{a, b, c\}$ tepeler kümesi, $L = \{git, ata\}$ etiketler kümesi ve $r_E = \{(a, b, git), (b, c, ata), (a, c, ata)\}$ yönlü etiketli tepeler kümesi.

Tanım 2.7 (Ulman, J. H. and. J., 2006) ***Bir şeritli deterministik Turing makinesi*** aşağıdaki özelliklere sahip bir $M = \langle Q, \Gamma, b, \sum, \delta, q_0, F \rangle$ sıralısıdır:

- Boş olmayan sonlu durumların bir Q kümesi,

- Boş olmayan sonlu şerit sembollerinin bir Γ kümesi,

- $b \in \Gamma$ boşluk sembolü,

- $\sum \in \Gamma \setminus \{b\}$ girdi sembollerinin kümesi,

- $q_0 \in Q$ başlangıç durumu,

- L sol ve R sağ yönü temsil etmek üzere $\delta : (Q \setminus F) \times \Gamma \longmapsto Q \times \Gamma \times \{L \times R\}$ geçiş fonksiyonu.

Tanım 2.8 (Sipser, M., 2006) ***Deterministik olmayan Turing makinesi*** aşağıdaki özelliklere sahip bir $M = \langle Q, \Gamma, b, \sum, \delta, q_0, F \rangle$ sıralısıdır:

- Boş olmayan sonlu durumların bir Q kümesi,

- Boş olmayan sonlu şerit sembollerinin bir Γ kümesi,

- $b \in \Gamma$ boşluk sembolü,

- $\sum \in \Gamma \setminus \{b\}$ girdi sembollerinin kümesi,

- $q_0 \in Q$ başlangıç durumu,

- L sol ve R sağ yönü temsil etmek üzere $\delta : (Q \setminus F) \times \Gamma \longmapsto \mathcal{P}(Q \times \Gamma \times \{L \times R\})$ geçiş fonksiyonudur.

Tanım 2.9 (Sipser, M., 2006) M tüm girdiler için duran bir deterministik Turing makinesi olsun. M nin ***çalışma zamanı*** veya M nin ***zaman karmaşıklığı*** bir $f : \mathbb{N} \longmapsto \mathbb{N}$ fonksiyonudur öyle ki $f(n)$, M nin kullandığı n uzunluktaki her girdi için en büyük adım sayısıdır. Eğer $f(n)$, M nin çalışma zamanı ise M, $f(n)$ zaman çalışır denir ve M ye bir $f(n)$ zaman Turing makinesi denir.

Tanım 2.10 (Sipser, M., 2006) f ve g fonksiyonları $f, g : \mathbb{N} \longmapsto \mathbb{R}^+$ olsun. Eğer her $n \geq n_0$ tamsayısı $f(n) \leq c.g(n)$ olacak şekilde pozitif n_0, c pozitif tamsayıları varsa *f(n)=O(g(n))* denir. Burada $g(n)$, $f(n)$ için ***bir üst sınırdır*** denir ve

daha kesin bir ifadeyle $g(n)$, $f(n)$ için ***asimtotik üst sınırdır*** denir.

Örnek 2.11 $h_1(n)$ = $6n^4$+$3n^2$+$8n$ olsun. $h_1(n)$ = $O(n^4)$ olur. c = 7 ve n_0 = 3 seçersek ve her $n \geq 3$ alırsak $5n^4$ + $3n^2$ + $8n \leq 7n^4$ sağlanır. Ayrıca n^5 fonksiyonu da h_1 için bir üst sınır olduğu için $h_1(n)$ = $O(n^5)$ dir.

Tanım 2.12 (Sipser, M., 2006) $t : \mathbb{N} \longmapsto \mathbb{R}^+$ bir fonksiyon olsun. $TIME(t(n))$ yani ***zaman karmaşıklığı sınıfı***, $O(n)$ zamanlı Turing makinesi tarafından karar verilebilen tüm dillerin sınıfıdır.

Tanım 2.13 (Sipser, M., 2006) ***P***, polinom zamanda çözülebilen problemlerin sınıfıdır. Diğer bir deyişle ***P*** = $\bigcup_{k\in\mathbb{N}} DTIME(n^k)$ sınıfıdır (Rudich, S. and Wigderson, A., 2004).

Tanım 2.14 (Sipser, M., 2006) ***NP***, polinom zamanda denetlebilen problemlerin sınıfıdır. Diğer bir deyişle ***NP*** = $\bigcup_{k\in\mathbb{N}} NTIME(n^k)$ sınıfıdır (Rudich, S. and Wigderson, A., 2004).

Tanım 2.15 V = $\{1, 2, 3, ..., n\}$ tepelerinden oluşan küme ile verilen bir yönlü graf G = (V, E) nın **yansıyan geçişken kapanışı** yönlü ayrıt kümesi E^*=$\{(i, j)$: G *içinde i den j ye bir yol vardır* $\}$ özelliği ile donatılmış G^* = (V^*, E^*) grafıdır.

3. ARİSTO SİLLOJİZMİ VE MODERN YAKLAŞIMLAR

Bu kısımda, klasik ve güncel *sillojizm* yaklaşımlarına yer verilecektir. Bölüme sillojizm kavramının sahibi olan Aristo'nun yorumu ile başlayıp, daha sonra modern yaklaşımlarla bazı karşılaştırmalarına değinilecektir.

3.1 Aristo Sillojizmi

All men are mortal
All Greeks are men
———————————-
∴ All Greeks are mortal

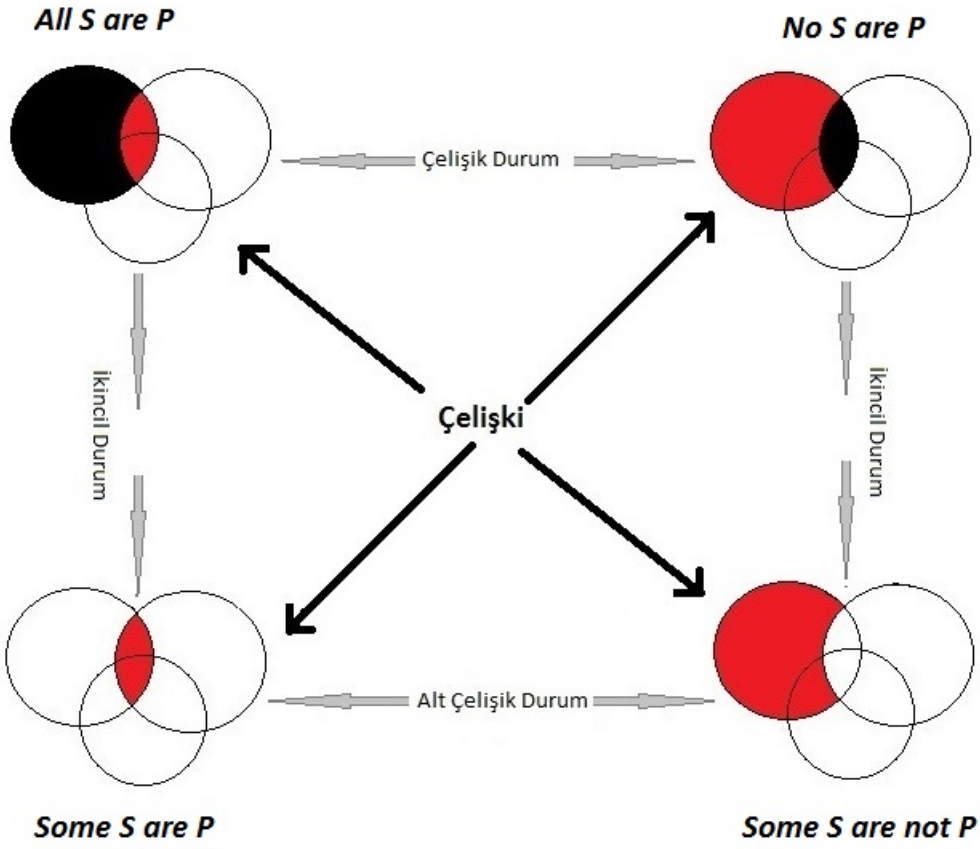
Klasik sillojizm kavramının daha kolay anlaşılması için yukarıdaki örnekten hareketle Aristo sillojizminin terminolojisi hakkında bazı bilgiler verelecektir. Öncelikle ∴ notasyonu önündeki cümlenin bir ***hüküm*** cümlesi olduğunu gösterir. Burada *All men are mortal* ve *All Greeks are men* birer ***öncül*** ve *All Greeks are mortal* cümlesine bir ***hüküm*** denir. *Greeks, mortal* ve *men* kelimelerine birer ***terim*** denir. Hükümde yer almayan fakat her iki öncülde de yer alan *men* kelimesine ***orta terim***, hüküm cümelesinin öznesi olan Greeks kelimesine ***küçük terim***, hüküm cümlesinin yüklemi olan *mortal* kelimesine ***büyük terim*** denir. Büyük terimin yer aldığı öncül cümlesine ***büyük öncül*** ve küçük terimin yer aldığı öncül cümlesine ***küçük öncül*** denir. Bir öncül ya da hüküm ***All subject(özne) are predicate(yüklem)*** formunda ise ***A tipi***; ***Some subject(özne) are predicate(yüklem)*** formunda ise ***I tipi***; ***No subject(özne) are predicate(yüklem)*** formunda ise ***E tipi***; ***Some subject(özne) are not predicate(yüklem)*** formunda ise ***O tipi*** cümledir. Bir sillojizm iki öncül ve bir hükümden oluştuğu için sırasıyla büyük öncül, küçük öncül ve hüküm cümlesinin tipleri yanyana yazılarak bir ***durum*** meydana getirilir. Örnek verdiğimiz sillojizm ***AAA*** durumundadır (Smith, R., 1989).

	Anımsatıcılar	Durumlar	Örnek
1	Barbara	AAA	*All M are P, All S are M ∴ All S are P*
2	Celarent	EAE	*No M are P, All S are M ∴ No S are P*
3	Darii	AII	*All M are P, Some S are M ∴ Some S are P*
4	Ferio	EIO	*No M are P, Some S are M ∴ Some S are not P*
5	Cesare	EAE	*No P are M, All S are M ∴ No S are P*
6	Camestres	AEE	*All P are M, No S are M ∴ No S are P*
7	Festino	EIO	*No P are M, Some S are M ∴ Some S are not P*
8	Baroko	AOO	*All P are M, Some S are not M ∴ Some S are not P*
9	Darapti	AAI	*All M are P, All M are S ∴ Some S are P*
10	Disamis	IAI	*Some M are P, All M are S ∴ Some S are P*
11	Datisi	AII	*All M are P, Some M are S ∴ Some S are P*
12	Felapton	EAO	*No M are P, All M are S ∴ Some S are not P*
13	Bocardo	OAO	*Some M are not P, All M are S ∴ Some S are not P*
14	Ferison	EIO	*No M are P, Some M are S ∴ Some S are not P*
15	Bramantip (Bamalip)	AAI	*All P are M, All M are S ∴ Some S are P*
16	Camenes (Calemes)	AEE	*All P are M, No M are S ∴ No S are P*
17	Dimaris (Dimatis)	IAI	*Some P are M, All M are S ∴ Some S are P*
18	Fesapo	EAO	*No P are M, All M are S ∴ Some S are not P*
19	Fresison	EIO	*No P are M, Some M are S ∴ Some S are not P*

Tablo 3.1: Sillojizm şekilleri

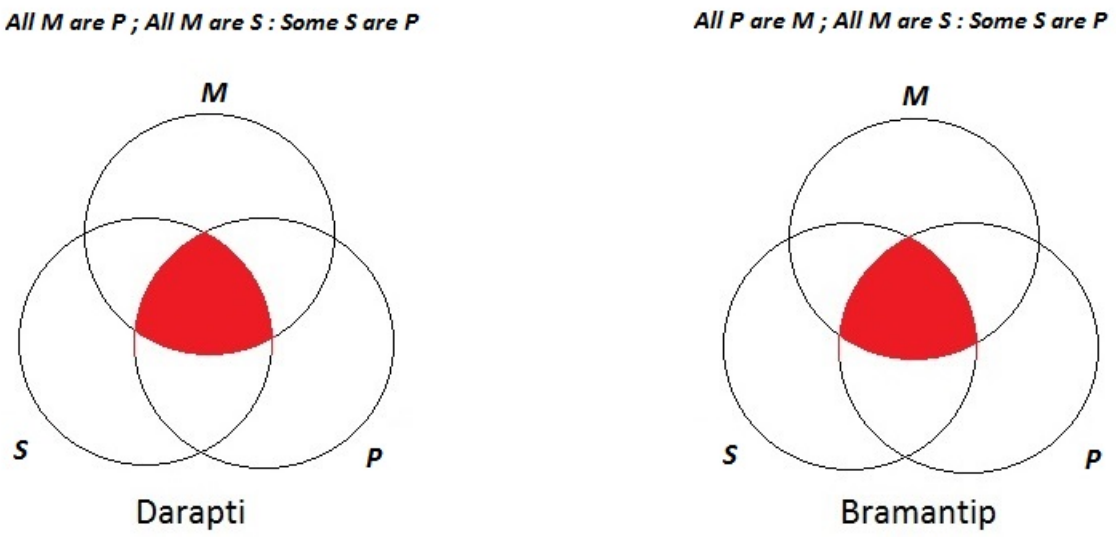
Tablo 3.1 de yer alan sıralanmış 19 sillojizm formundan ilk 14 tanesi Aristo tarafından sunulmuş, diğerleri Theophrastus (M.Ö 371–M.Ö 287) tarafından eklenmiş ve Petrus Hispanus (1215–1277) tarafından açıklanmıştır (Carruccio, E., 1964). *M*

orta terim, *P* yüklem ve *S* özne anlamına gelmektedir. Durumların herbirine karşılık gelen anımsatıcıların sesli harfleri durumları ifade etmektedir. Örneğin, AII durumu Darii hatırlatıcısının sesli harfleriyle ifade edilir.



Şekil 3.1: Aristo karşıtlık karesi (Apuleios, M.S II.yy)

Şekil 3.1 de siyah bölgeler kesin boş ve gri bölgeler kesin dolu alanlardır. Şekildeki kare üzerinde *All S are P* ifadesi ile *No S are P* ve *Some S are not P* ifadeleri arasında çelişkili bir durum olduğu gösterilmiştir. Aynı şekilde *Some S are P* ile *No S are P* arasında yine doğduran bir çelişki olduğu ve *Some S are P* ile *Some S are not P* arasında dolaylı bir çelişki durumu olduğu söylenmiştir. *All S are P* ifadesinin *Some S are P* ifadesinden çok daha kesin bir ifadeye sahip olduğu ve *No S are P* ifadesinin *Some S are not P* ifadesinden çok daha kesin bir yargıya sahip olduğu gösterilmiştir. Böyle bir karenin ortaya çıkmasındaki asıl neden ise Aristo'nun terimlerin semantiği için boş küme kavramını görmezden gelmesidir çünkü terimler gerçek yaşamdan esinlenerek oluşturulmuş ve bir Socrates veya bir Yunanlının bir boş küme olması bu anlamda düşünülemez.



Şekil 3.2: Darii ve Bramantip için Venn şemaları

Şekil 3.2 yi incelediğimizde Darapti ve Bramantip durumlarının hükümleri öncüller tarafından takip edilir çünkü *M, P* ve *S* terimleri semantik anlamda birer boş küme olmadığı için Ockham yöntemiyle Darii ve Bramantip birer geçerli sillojizmlerdir (Normore, C. G., 1999).

3.2 Sillojizme Modern Yaklaşımlar

Aristo'nun bakış açısına göre sillojizmin ne anlama geldiğine bir önceki kesimde değinildi. Şimdi ise Łukasiewicz, Lorenzen, Corcoran-Smiley, Westerståhl"in bakış açılarını ve son olarak bu tezin en büyük ilgi alanı içindeki Moss'un sillojizme bakış açısını yansıtıp, bir sonraki kısımda ayrıntılar verilecektir.

3.2.1 Łukasiewicz'e Göre Sillojizm

Tanım 3.1 (Łukasiewicz, J., 1957)

Sillojizmler, önbileşen olarak *öncüllerin* ve bir hükmün gerektirmesidir. Daha yalın bir ifade ile ψ ve ϕ iki *öncül*, φ *hüküm* olmak üzere, tüm Aristo sillojizmleri *If* ψ *and* ϕ*, then* φ şeklindedir. "ψ and ϕ" önbileşen ve φ hükmü ise bir koşullunun ikinci anabileşenidir.

Łukasiewicz'e göre Aristo'nun çalışmasında önermesel lojik tamamen göz ardı edilmiştir. Önermesel lojik ile sillojistik arasındaki farkı gözlemlemek için *Aab* ile $p \longrightarrow q$ yi incelemek yeterlidir. Bu temel fark argümanların yapısından kaynaklanmaktadır çünkü burada *a* ve *b* değişkenleri birer terimken *p* ve *q* birer önermedir (Fillion, N., 2007) ve (Łukasiewicz, J., 1957).

Tanım 3.2 (Łukasiewicz, J., 1957) ve (Fillion, N., 2007)

Aristo'nun lojiği için Łukasiewicz'in sillojistik lojik sistemi $\mathcal{LS}$ aşağıda tanımlanmıştır:

1. $\mathcal{LS}$ dilinin elemanları:
   (a) Önermesel bağlaçlar: $\neg, \Rightarrow, \wedge$,
   (b) Önermesel değişenler: $p, q, r, s, ...$,
   (c) Terim değişkenleri: $a, b, c, ...$,
   (d) Sillojistik operatörler: A, I.
2. $\mathcal{LS}$ nin atomik formülleri:
   (a) Herhagi a, b terim değişkenleri için Aab ve Iab atomik formüller,
   (b) (a) şıkkı dışında hiçbir atomik formül formu yoktur.
3. $\mathcal{LS}$ nin iyi biçimlendirilmiş formülleri (ibf):
   (a) Tüm atomik önermeler ibf dir,
   (b) Tüm önermesel değişkenler ibf dir,
   (c) Her $p,\ q$ önermesel değişkeni için $\neg p,\ p \wedge q,\ p \Rightarrow q$ birer ibf dir,
   (d) (a), (b) ve (c) şıkkı dışında hiçbir ibf formu yoktur.
4. $\mathcal{LS}$ nin aksiyomları:
   (a) Aaa (A için Özdeşlik Kuralı),
   (b) Iaa (I için Özdeşlik Kuralı),

(c) $(Abc \wedge Aab) \Rightarrow Aac$ (Barbara),

(d) $(Abc \wedge Iba) \Rightarrow Iac$ (Datisi),

(e) $p \Rightarrow (q \Rightarrow p)$ (Yalınlaştırma kuralı),

(f) $(p \Rightarrow q) \Rightarrow ((q \Rightarrow r) \Rightarrow (p \Rightarrow r))$ (Hipotetik Sillojizm Kuralı),

(g) $((p \Rightarrow (q \Rightarrow r)) \Rightarrow (q \Rightarrow (p \Rightarrow r))$ (Değişme Kuralı),

(h) $p \Rightarrow (\neg p \Rightarrow q)$ (Duns Scotus Kuralı),

(i) $(\neg p \Rightarrow p) \Rightarrow p$ (Clavius Kuralı),

(j) $(p \Rightarrow q) \Rightarrow (\neg q \Rightarrow \neg p)$ (Yer Değiştirme kuralı),

(k) $((p \wedge q \Rightarrow r)) \Rightarrow (p \Rightarrow (q \Rightarrow r))$ (Çıkma Kuralı),

(l) $p \Rightarrow (((p \wedge q) \Rightarrow r) \Rightarrow (q \Rightarrow r))$,

(m) $(s \Rightarrow p) \Rightarrow (((p \wedge q) \Rightarrow r) \Rightarrow ((s \wedge q) \Rightarrow r))$,

(n) $((p \wedge q) \Rightarrow r) \Rightarrow ((s \Rightarrow q) \Rightarrow ((p \wedge s) \Rightarrow r))$,

(o) $(r \Rightarrow s) \Rightarrow (((p \wedge q) \Rightarrow r) \Rightarrow ((q \wedge p) \Rightarrow s))$,

(p) $((p \wedge q) \Rightarrow r) \Rightarrow ((p \wedge \neg r) \Rightarrow \neg q))$,

(q) $((p \wedge q) \Rightarrow r) \Rightarrow ((\neg r \wedge q) \Rightarrow \neg p))$,

(r) $((p \wedge \neg q) \Rightarrow \neg r) \Rightarrow ((p \wedge r) \Rightarrow q))$.

5. $\mathcal{LS}$ nin aksiyomatik olarak reddedilmiş önermeleri:

   (a) $(Acb \wedge Aab) \Rightarrow Iac$,

   (b) $(Ecb \wedge Eab) \Rightarrow Iac$.

6. $\mathcal{LS}$ nin türetim kuralları:

   (a) (***İkame***) Eğer p, $\mathcal{LS}$ nin bir iddia ifadesi ise, geçerli bir ikame ile p den üretilen her ifade de bir iddia ifadesidir. ***Geçerli ikame*** sadece terim değişkenlerini terim değişkenlerinin yerine ikame eder.

   (b) (***Ayırma***) Eğer $p \Rightarrow q$ ve p, $\mathcal{LS}$ nin bir iddia ifadesi ise, o zaman q da bir iddia ifadesidir.

   (c) (***Ayırmayla Reddetme***) Eğer $p \Rightarrow q$ bir iddia fakat q sonucu reddedilmiş ise, p önbileşeni de reddedilmek zorundadır.

   (d) (***İkameyle Reddetme***) Eğer $p \Rightarrow q$ bir iddia fakat q sonucu reddedilmiş ise, p önbileşeni de reddedilmek zorundadır.

   (e) (***İkameyle Reddetme***) Eğer q, p nin bir ikamesi ve q reddedilmiş ise p de reddedilmek zorundadır.

Tanım 3.3 (Chierchia, G. and McConnell-Ginet, S., 2000) Doğal dil işleme alanında yer alan ***metinsel gerektirme***, metin parçaları arasındaki bir yönlü bağıntıdır. Bu bağıntı, bir metin parçasının doğruluğunun bir diğerinin doğruluğunu takip etmesini korur.

Örnek 3.4 Metin ve metinden çıkarılacak yargı olan hipotez için bir örnek:
Metin **:** Bilgisayar titanyumdan yapılmış malzeme ile kaplanmış.
Hipotez **:** Bilgisayar kırılmaya karşı dayanıklıdır.

Łukasiewicz'in oluşturduğu Aristo sistemi, Aristo'nun terimler ve nesneler üzerinden hareket ettiği noktayı lojik sistemine taşımış ve bu anlamda sillojizmin matematik lojik alanında bir taban bulmasını sağlamıştır. Łukasiewicz'in bu bakış açısı Aristo'nun sillojistik teorisini metinsel gerektirmeye çok daha yakın kılmıştır.

Tanım 3.5 (Chierchia, G. and McConnell-Ginet, S., 2000) ***Materyal içerme*** kavramı yeni cümleler yaratmak için kullanılabilecek bir ikili bağlaçtır. Diğer bir ifade ile nesne düzeyinde bir semboldür ve bir sabit model içindeki iki cümlenin doğruluk değerlerinin bir fonksiyonudur. Öte yandan, ***Mantıksal içerme*** ise "bir cümleyi doğru yapan her model diğer bir cümleyi de doğru yapar" özelliğine sahip, iki cümle arasındaki bir bağıntıdır. Yani nesneye bağlı olmayan meta-düzey bir kavramdır.

Łukasiewicz'in sillojizmi yeniden inşası sırasında mantıksal doğruluğu mantıksal çıkarıma tercih etmesi, materyal içerme yerine mantıksal içermeyi tercih ettiğini gösterir.

3.2.2 Parsons'a göre sillojizm

Parsons'un sillojizm yorumu geleneksel bir şekilde olmuştur. Her ne kadar yorumu yakın tarih (1999) olsa da Aristo'nun klasik bakış açısını aynen yansıtmıştır. Parsons, bu yorumlama yaklaşımı ile Aristo sillojizmi için bir geleneksel karşıtlık karesi örneklemiştir.

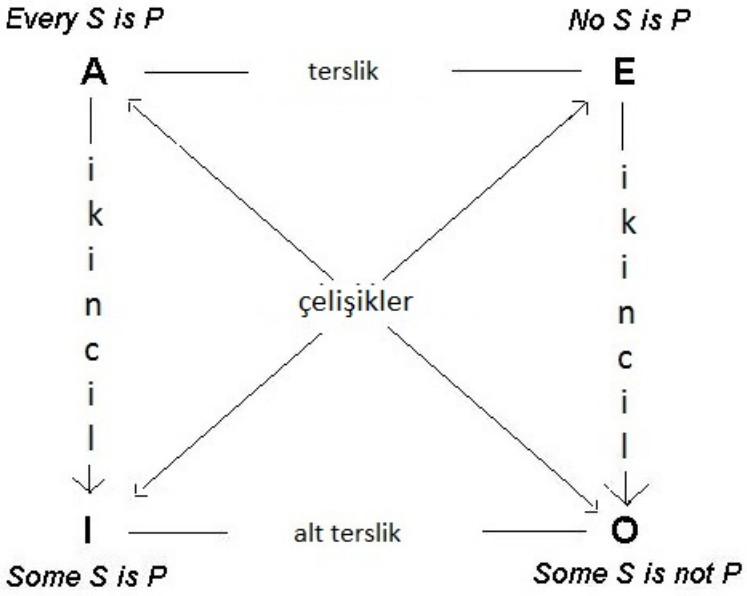
Tanım 3.6 (Parsons, T., 1999)

1. İki önerme ***çelişiktir*** *eğer ve yalnız eğer* iki önerme de doğru ve iki önerme de yanlış olamaz.

2. İki önerme ***tersdir*** *eğer ve yalnız eğer* iki önerme doğru olamaz fakat yanlış olabilir.

3. İki önerme ***alt tersdir*** *eğer ve yalnız eğer* iki önerme yanlış olamaz fakat doğru olabilir.

4. Bir önerme bir diğerinin ***ikincilidir*** *eğer ve yalnız eğer* ikincil doğru ise kendisi de doğru ve ikinicil yanlış ise kendisi de yanlış olmak zorundadır.



Şekil 3.3: Parsons yorumuyla geleneksel karşıtlık karesi

Tanım 3.6 ya göre önermelerin durumu incelenirse aşağıdaki durumlar ortaya çıkar:

i. *Every S is P* ve *Some S is not P* çelişiktir.

ii. *No S is P* ve *Some S is P* çelişiktir.

iii. *Every S is P* ve *No S is P* tersidir.

iv. *Some S is P* ve *Some S is not P* alt tersidir.

v. *Some S is P*, *Every S is P* önermesinin ikincilidir.

vi. *Some S is not P*, *No S is P* önermesinin ikincilidir.

Açıklama 3.7 Şekil 1.1 deki kümesel semantikler ve Şekil 3.3 deki kare incelendiğinde *Every S is P* önermesine karşılık gelen *All S are P* ifadesinin doğru olması için $S \setminus P$ boş küme ve $S \cup P$ boştan farklı olmalıdır. $S \cup P$ olması ne S ne de P kümesinin boş olmadığını ve en az bir elemana sahip olduklarını gösterir. Fakat *Some S is not P* ifadesinin doğru olması için $S \setminus ((M \cup P) \setminus S)$ boştan farklı olmalıdır. Bu nedenle çelişik durum ortaya çıkmaktadır. Öte yandan, herhangi bir terim için boş kümeye izin verilseydi bu çelişki ortaya çıkmayacaktı.

Açıklama 3.8 *Every S is P* ve *No S is P* önermeleri ters olsun. *Every S is P* ve *No S is P* önermelerinin yanlış ve ayrıca $S = \varnothing$ ve $P \neq \varnothing$ varsayalım. O zaman $S \subseteq P$ ve $S \cap P = \varnothing$ dir. Böylece *Every S is P* ve *No S is P* önermeleri doğrudur. Bu *Every S is P* ve *No S is P* önermelerinin ters olmadığı anlamına gelir. Fakat ters kabul etmiştik. Bu durumda ortaya bir çelişkidir. Çelişkinin ortaya çıkış nedeni S kümesinin bir boş küme seçilmesidir. Burada yine Aristo'nun boş küme kavramını düşünce sisteminde barındırmadığını görmekteyiz.

Parsons'un da Aristo gibi, boş küme kavramını kabul etmediği gözlemlenmiş ve bunu geleneksel karşıtlık karesine yansıtmıştır.

3.2.3 Corcoran-Smiley'e göre sillojizm

Bu yaklaşım daha çok argümanlar ve argümanların geçerliliği üzerine şekillenmektedir. Bu anlamda bir argümanın geçerli olması hükmün, doğru öncüllerin bir mantıksal sonucu olmasıyla değerlendirildiği için bir mantıksal sisteme oturtulmalıdır çünkü Aristo'nun ortaya koyduğu her argüman geçerli olmadığı için bahsedilecek olan mantıksal sisteme ihtiyaç duyulacaktır. Bir önceki kısımda bahsedildiği üzere Łukasiewicz'in sistemi sillojizmin içerisinden değil, bir doğal tümden-

gelimden ziyade önermesel lojik üzerine kurulmuştur. Fakat Corcoran-Smiley sistemi doğrudan ve dolaylı doğal tümdengelimsel bir sistem üzerine kurulmuştur. Şimdi bir kaç önemli tanım verdikten sonra Corcoran-Smiley'ın doğal tümdengelimsel sistem incelenecektir.

Tanım 3.9 (Fillion, N., 2007) Bir ***argüman***, bir P öncüller olarak adlandırılan cümlelerin kümesinden ve bir d hüküm cümlesinden oluşan (P, d) sıralı ikilisidir.

Tanım 3.10 (Gamut, L., T., F., 2014)

i. Bir argüman ***geçerlidir*** $:\Leftrightarrow$ hüküm, mantıksal olarak öncülleri takip eder.

ii. Bir formül ***geçerlidir*** $:\Leftrightarrow$ bu formül her yorum altında doğrudur.

iii. Bir argüman formu ***geçerlidir*** $:\Leftrightarrow$ bu mantıksal formdaki her argüman geçerlidir.

Tanım 3.11 (Corcoran, J., 1972), (Fillion, N., 2007) ve (Smiley, T. J., 1973)

Aristo'nun lojiği için Corcoran-Smiley'in formal ***doğal tümdengelimsel sistem*** $\mathcal{CS}$ aşağıda tanımlanmıştır:

1. $\mathcal{CS}$ dilinin elemanları:
   (a) $a, b, c, ...$ terim sabitlerini içeren bir V kümesi,
   (b) Sillojistik operatörler A, E, I, O.
2. $\mathcal{CS}$ için iyi biçimlendirilmiş formüller:
   (a) Herhangi farklı a ve b terim sabitleri için Aab, Eab, Iab, Oab iyi biçimlendirilmiş formüllerdir.
   (b) (a) da yer alan formlar dışında hiçbirşey iyi biçimlendirilmiş formül değildir.
3. Herhangi $x, y, z \in V$ için $\mathcal{CS}$ için türetim kuralları:
   (a) $Exy \vdash Eyx$ (Dönüşme 1),
   (b) $Axy \vdash Iyx$ (Dönüşme 2),
   (c) $Ixy \vdash Iyx$ (Dönüşme 3),
   (d) $Azy, Axz \vdash Axy$ (Barbara),
   (e) $Ezy, Axz \vdash Exy$ (Celarent),

(f) $Azy, Ixz \vdash Ixy$ (Darii),

(g) $Ezy, Ixz \vdash Oxy$ (Ferio),

(h) $P, C(d) \vdash e$ ve $P, C(d) \vdash C(e)$ ise $P \vdash d$ (Reductio).

Tanım 3.12 (***CS de doğrudan tümdengelim***) P den d ye bir doğrudan tümdengelim, P içindeki All ya da $Some$ cümleleriyle başlayan ve sonra ya bir önceki satırın tekrarı ya bir önceki satırın CS dönüşümü ya da iki önceki satırın bir CS çıkarımından elde edilen d ile biten cümlelerin sonlu bir listesidir (Andrade, E. J., and Becerra, E., 2007).

Tanım 3.13 (***CS de dolaylı tümdengelim***) P den d ye bir dolaylı tümdengelim, d nin çelişik durumunu takip eden P içindeki All ya da $Some$ cümleleriyle başlayan ve sonra ya bir önceki satırın tekrarı ya bir önceki satırın CS dönüşümü ya da iki önceki satırın bir CS çıkarımından elde edilen e ve $C(e)$ çelişmelerinin bir çifti içinde biten cümlelerin sonlu bir listesidir (Andrade, E. J., and Becerra, E., 2007).

Tanım 3.14 (Andrade, E. J. and Becerra, E., 2007) CS için ***semantik sistem*** aşığıdaki gibi tanımlar:

$R := \{a, e, i, o\}$ ve V boş olmayan bir küme olsun. P_V cümleler kümesi, $S, P \in V$ ve $S \neq P$ olmak üzere SaP, SeP, SiP, SoP formlarını içersin. $M = \{U_i\}_{i \in I}$ boş olmayan kümelerin bir ailesi ve $g : V \to M$ olsun. ***Yorum fonksiyonu*** $[[\cdot]]_{M,g} : P_V \to \{0, 1\}$ aşağıdaki gibi tanımlanmıştır:

1. $[[SaP]]_{M,g} = 1$ *eğer ve yalnız eğer* $g(S) \subseteq g(P)$,
2. $[[SeP]]_{M,g} = 1$ *eğer ve yalnız eğer* $g(S) \cap g(P) = \varnothing$,
3. $[[SiP]]_{M,g} = 1$ *eğer ve yalnız eğer* $g(S) \cap g(P) \neq \varnothing$,
4. $[[SoP]]_{M,g} = 1$ *eğer ve yalnız eğer* $g(S) \nsubseteq g(P)$.

Tanım 3.14 deki SaP, SeP, SiP, SoP formları sırasıyla *All S are P*, *No S ar P*, *Some S are P*, *Some S are not P* daha modern bir şekilde ifade edilebilir. Tanım 3.11 deki dönüşme 2 kuralı $\frac{\textit{All x are y}}{\textit{Some x are y}}$ şeklinde ifade edilebilir. Bu kural " $g(x) \subseteq g(y)$ doğru ise $g(S) \cap g(P) \neq \varnothing$ de doğrudur " anlamındadır. Buradan hiç bir değişkenin semantiğinin bir boş küme olmayacağı görülür. Bu yorum

için karşıtlık karesi sistem şekil 3.3 deki Parsons yorumuyla ifade edilmiş karşıtlık karesidir. Corcoran sillojistik lojiği için modern bir kurallar tablosu aşağıdaki gibi ifade edilir.

$$\frac{All\ p\ are\ q}{Some\ p\ are\ q}$$

$$\frac{Some\ p\ are\ q}{Some\ q\ are\ p} \qquad \frac{No\ p\ are\ q}{No\ q\ are\ p}$$

$$\frac{All\ q\ are\ n \quad Some\ p\ are\ q}{Some\ p\ are\ n} \qquad \frac{All\ p\ are\ n \quad No\ n\ are\ q}{No\ p\ are\ q}$$

$$\frac{No\ m\ are\ p \quad Some\ s\ are\ p}{Some\ m\ are\ not\ s} \qquad \frac{All\ p\ are\ n \quad All\ n\ are\ q}{All\ p\ are\ q}$$

Tablo 3.2: Corcoran sillojistik lojiği kuralları

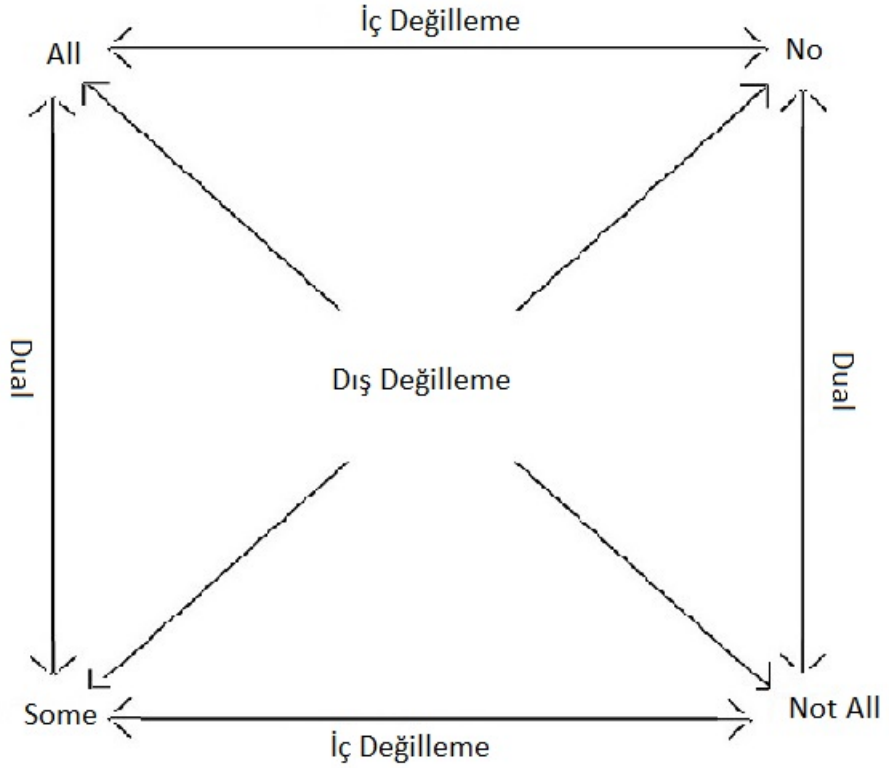
3.2.4 Westerståhl'e göre sillojizm

Westerståhl yaklaşımı bu tezde daha önce gördüğümüz yaklaşımlardan doğal dile en yakın olanıdır. Bunun sebebi *varlıksal içe aktarım* dır. Eğer sisteminizin içerisine varlıksal içe aktarımı dahil ederseniz, sisteminizdeki her varlık *boş olmayan* bir kümeye denk gelmektedir ve böylece CS sistemi de dahil olmak üzere daha önce bahsettiğimiz tüm sillojistik sistemler klasik karşıtlık karesini meydana getirirler. Fakat Westerståhl yaklaşımı bunu kabul etmez. Westerståhl sisteminin bunu neden kabul etmediğini bir örnekle açıklayalım.

Örnek 3.15 "*Evdeki tüm öğrenciler tembeldir*" önermesi CS sistemi içerisinde *Dönüşme 2* kuralından dolayı otomatik olarak "*Evdeki bazı öğrenciler tembeldir*" önermesini doğru kılar.

Fakat evde öğrenci olup olmadığı hakkında bir bilgiye sahip değiliz. Bu nedenle evdeki öğrenciler kümesinin boş olup olmadığı da saptanamaz. Bu durumda "*Evdeki bazı öğrenciler tembeldir*" çıkarımı hakkında birşey söylenemez ve doğrudur denemez.

Varlıksal içe aktarımın sillojistik akıl yürütmeye etkisi klasik karenin değişimine yol açmıştır çünkü, Şekil 3.3 ve Şekil 3.4 de görülebileceği gibi, terslik, birincil ve ikincil gibi kavramlar, değilleme ve dual kavramlarına dönüşmüştür.



Şekil 3.4: Modern karşıtlık karesi (Westerståhl, D., 2005)

3.2.5 Moss'un sillojistik lojiği

Moss tarafından yayınlanan ilk makale (Moss, L. S., 2008) doğal lojik programının ve uygulamalı lojik alanlarının ilk adımlarından biri sayılabilir. Moss, Pratt-Hartmann tarafından yayınlanan (Pratt-Hartmann, I., 2004) ve İngilizce doğal dilinin parçalarının katerorik gramer, lambda kalkulus ve birinci mertebe lojik kullanarak zaman karmaşıkllıkları üzerine yaptığı çalışmaların tamlık ispatları ve algoritmaları üzerinde durmuştur. 4. Bölümde Moss'un çalışmaları üzerinde ayrıntılı olarak incelemeler yapılacaktır.

4. S, $S^{\dagger}$, $\mathcal{CARD}$ ve $\mathcal{CARD}$ + IA LOJİKLERİ

Bu bölümde, S, $S^{\dagger}$, $\mathcal{CARD}$ ve $\mathcal{CARD}$ + IA lojikleri sintaks ve semantikleri incelenecektir.

4.1 $\mathcal{L}(All, Some, No)$ Dili ve S Lojiği

(Moss, L. S., 2008) $\mathbb{P}$ 1-li atomlar olan isimlerin bir koleksiyonu ve $p, q \in \mathbb{P}$ olsun. S lojiğinin cümlelerinin kümesi olan $\mathcal{L}(All, Some, No)$ dili, $All\ p\ are\ q$, $Some\ p\ are\ q$ ve $No\ p\ are\ q$ sintakslarını içerir. Semantikler *model* üzerinde inşa edilecektir. $\mathcal{L}(All, Some, No)$ dili için bir $\mathcal{M}$ modeli, M evren ve her $p \in \mathbb{P}$ için $[[p]] \subseteq M$ olacak şekilde aşağıdaki gibi inşa edilir. Bir $\mathcal{M} = (M, [[\ ]])$ modeli aşıığıdaki semantik özelliklere sahiptir:

$\mathcal{M} \vDash All\ p\ are\ q$ eğer ve yalnız eğer $[[p]] \subseteq [[q]]$,

$\mathcal{M} \vDash Some\ p\ are\ q$ eğer ve yalnız eğer $[[p]] \cap [[q]] \neq \varnothing$,

ve

$\mathcal{M} \vDash No\ p\ are\ q$ eğer ve yalnız eğer $[[p]] \cap [[q]] = \varnothing$.

Tanım 4.1 Γ cümlelerin bir kümesi olsun. $\mathcal{M} \vDash \Gamma$ eğer ve yalnız eğer her $C \in \Gamma$ için $\mathcal{M} \vDash C$ dir.

Tanım 4.2 $\Gamma \vDash C$ eğer ve yalnız eğer Γ içindeki her cümleyi doğru yapan her model C yi de doğru yapar.

Tanım 4.3 Bir Γ üzerindeki ***ispat ağacı*** düğümleri dildeki cümleler ile etiketli ve her bir düğümü ya Γ nın bir elemanı ya da kuralların uygulanması ile elde edilmiş sonlu ağaçtır.

Tanım 4.4 Eğer her $C \in \Gamma$ için $\Gamma \vDash C$ ise Γ cümleler kümesi ***tutarsızdır*** denir.

$$\frac{}{All\ x\ are\ x}$$

$$\frac{No\ x\ are\ x}{All\ x\ are\ y} \qquad \frac{All\ x\ are\ y \quad All\ y\ are\ z}{All\ x\ are\ z}$$

$$\frac{No\ x\ are\ y}{No\ y\ are\ x} \qquad \frac{All\ x\ are\ y \quad No\ y\ are\ z}{No\ x\ are\ z}$$

$$\frac{Some\ x\ are\ y}{Some\ y\ are\ x} \qquad \frac{All\ x\ are\ y \quad Some\ x\ are\ z}{Some\ y\ are\ z}$$

$$\frac{Some\ x\ are\ y \quad No\ x\ are\ y}{C}\ \mathcal{X}$$

Tablo 4.1: $\mathcal{L}(All, Some, No)$ için kurallar tablosu

Tablo 4.1 de $\mathcal{X}$ Kuralı *ex falso quadlibet* olarak adlandırılan kuraldır. Kural, ispat ağacından $Some\ x\ are\ y$ ve $No\ x\ are\ y$ türetilebiliyor ise modelin tutarsız olduğunu ve böylece modelden bu dil içerisindeki her cümlenin türetilebileceğini söyler. Ayrıca bu lojik tam olduğu için bu dil içindeki her cümle bu modelde doğru olur.

Teorem 4.5 S lojiği Tablo 4.1 deki kurallar altında sağlam ve tamdır (Moss, L. S., 2008).

4.2 $\mathcal{L}(All, Some, ')$ Dili ve $S^{\dagger}$ Lojiği

(Moss, L. S., 2011) $\mathbb{P}$ 1-li atomlar olan isimlerin bir koleksiyonu olsun. $p, q \in \mathbb{P}$ olsun. $S^{\dagger}$ lojiğinin cümlelerinin kümesi olan $\mathcal{L}(All, Some, ')$ dili, *All p are q*, *All p are non-q*, *All p are non-q*, *All non-p are non-q*, *Some p are q*, *Some non-p are q* ve *Some non-p are non-q* sintakslarını içerir. Semantikler *model* üzerinde temellenecektir. $\mathcal{L}(All, Some, ')$ dili için bir $\mathcal{M}$ modeli, M evren ve her $p \in \mathbb{P}$ için $[[p]] \subseteq M$ ve $[[p']] = [[M]] \setminus [[p]]$ olacak şekilde bir $\mathcal{M} = (M, [[\ ]])$ modeli aşağıdaki semantik özelliğe sahiptir:

$\mathcal{M} \vDash All\ p\ are\ q$ eğer ve yalnız eğer $[[p]] \subseteq [[q]]$,

$\mathcal{M} \vDash Some\, p\, are\, q$ eğer ve yalnız eğer $[[p]] \cap [[q]] \neq \varnothing$.

Burada dikkat edilirse *No p are q* cümleleri *All p are non-q* ile ifade edilebildiği için dil içerisinde *No p are q* cümlelerine ihtiyaç yoktur.

$$\frac{}{All\, p\, are\, p}$$

$$\frac{All\, p\, are\, q'}{All\, q\, are\, p'}\ \textit{(Antitone)} \qquad \frac{Some\, p\, are\, q}{Some\, q\, are\, p}$$

$$\frac{All\, p\, are\, p'}{All\, p\, are\, q}\ \textit{(Zero)} \qquad \frac{All\, p\, are\, q \quad All\, q\, are\, r}{All\, p\, are\, r}$$

$$\frac{All\, p'\, are\, p}{All\, q\, are\, p}\ \textit{(One)} \qquad \frac{All\, p\, are\, q \quad Some\, p\, are\, r}{Some\, q\, are\, r}$$

$$\frac{Some\, p\, are\, q \quad All\, p\, are\, q'}{C}\ \mathcal{X}$$

Tablo 4.2: $\mathcal{L}(All, Some, ')$ için kurallar tablosu

Teorem 4.6 $S^{\dagger}$ lojiği Tablo 4.2 deki kurallar altında sağlam ve tamdır (Moss, L. S., 2011).

4.3 $\mathcal{CARD}$ ve $\mathcal{CARD}$ + IA Lojikleri

Bu kısımda kullanacağımız tanımlar ve teoremler "*On completeness and algorithms of logic of* $\mathcal{CARD}\ with\ IA$" (Öner, T. and Topal, S., 2014) isimli makaleden, (Moss, L. S., 2008) ve (Moss, L. S., 2010) makalelerden alıntılanacaktır.

Açıklama 4.7 $x,\ y,\ z, ...$ çoğul isimlerini içeren $\mathcal{P}$ değişkenler kümesi verilsin. $\mathcal{CARD}$ ın cümleleri *There are at least as many x as y* formundadır ve Türkçe olarak "*En az y kadar x vardır*" anlamındadır. $\exists^{\geq}(x, y)$ sintaksı *There are at least as many x as y* ın kısaltılmış hali olarak kullanılacaktır. Semantikler, cümlelerin sonlu kümesi olan Γ ve sonlu evren M üzerinde inşa edilecektir. $\mathcal{P}$ den M nin alt küme-

lerine olan $[[\,]]$ $yorum\ fonksiyonu$, her $x \in \mathcal{P}$ için, $[[x]] \subseteq M$ şeklinde tanımlanır. $[[x]]$ in kardinalitesi , $|[[x]]|$ ile gösterilir. Bir $\mathcal{M} = (M, [[\,]])$ modeli

"$\mathcal{M} \vDash \exists^{\geq}(x, y)$ eğer ve yalnız eğer $|[[x]]| \geq |[[y]]|$, $\mathcal{M}$ nin içindedir"

doğruluk özelliğine sahip olacaktır.

$$\frac{}{\exists^{\geq}(x,x)} \qquad \frac{\exists^{\geq}(x,y) \quad \exists^{\geq}(y,z)}{\exists^{\geq}(x,z)}$$

Tablo 4.3: $\mathcal{CARD}$ için kurallar tablosu

Açıklama 4.8 Γ, $\mathcal{CARD}$ deki cümlelerin bir alt kümesi olsun. $x \geq_c y$ notasyonu, $\Gamma \vdash \exists^{\geq}(x, y)$ ifade etmek için kullanılacaktır.

Teorem 4.9 $\mathcal{CARD}$, Tablo 4.3 deki kurallar altında sağlamdır (Öner, T. and Topal, S., 2014).

Kanıt $\mathcal{CARD}$ lojiğinin sağlamlığını ispatlamak için Γ üzerindeki ispat ağacının düğümlerinin sayısı üzerinden tümevarım yapacağız. Cümlelerin bir kümesi Γ ve φ nin bir cümle olduğunu kabul edelim. Γ üzerindeki tüm ispat ağaçları n sayısından küçük olsun. Kök ve ataları aşağıdaki gibi etkiletlensin:

$$\frac{\exists^{\geq}(x,y) \quad \exists^{\geq}(y,z)}{\exists^{\geq}(x,z)}$$

$\exists^{\geq}(y, z)$ ve $\exists^{\geq}(x, y)$ de sonlanan iki alt ağaç sırasıyla Υ_1 ve Υ_2 olsun. Bazı y için Υ_1 nın kökü $\exists^{\geq}(y, z)$ ve Υ_2 nın kökü $\exists^{\geq}(x, y)$ dır. Ayrıca, Υ_1 ve Υ_2 nin uzunlukları n sayısından küçüktür. Tümevarım hipotezinden $\Gamma \vDash \exists^{\geq}(y, z)$ ve $\Gamma \vDash \exists^{\geq}(x, y)$ dir. Γ ya ait cümlelerin hepsini birden doğru yapan keyfi bir $\mathcal{M}$ modeli alalım. Böylece, $\mathcal{M} \vDash \exists^{\geq}(x, y)$ ve $\mathcal{M} \vDash \exists^{\geq}(y, z)$ dir. O zaman $|[[x]]| \geq |[[y]]|$ ve $|[[y]]| \geq |[[z]]|$. Büyük veya eşit bağıntısının geçişme özelliği kullanılarak $|[[x]]| \geq |[[z]]|$ dir. $\mathcal{M}$ keyfi olduğu için $\Gamma \vDash \exists^{\geq}(x, z)$ dir. Diğer taraftan, ispat ağacı bir düğümlü ise, φ ya Γ nın elemanı ya da $\exists^{\geq}(x, x)$ formundadır. Eğer φ, Γ da ise Γ içindeki doğrulanan her cümle, φ yi de doğrular. Eğer φ , $\exists^{\geq}(x, x)$ formunda ise her model φ yi doğrular. □

Önerme 4.10 $\geq_c$, Γ değişkenlerinin kümesi üzerinde bir ön sıralama bağıntısıdır.

Teorem 4.11 $\mathcal{CARD}$ lojiği Tablo 4.3 deki kurallar altında tamdır.

Kanıt Tek elemanlı $M = \{*\}$ evrenini kullanarak aşağıdaki gibi bir $\mathcal{M}$ modeli inşa edilecektir.

$$[[z]] = \begin{cases} \{*\}, & \text{eğer } \Gamma \vdash \exists^{\geq}(z, y) \\ \varnothing\ , & \text{aksi taktirde} \end{cases}$$

$\Gamma \models \exists^{\geq}(x, y)$ olsun. $\exists^{\geq}(w, u)$ ifadesi Γ da ise $|[[w]]| \geq |[[u]]|$ dir. $|[[u]]| \neq \varnothing$ olsun. O zaman $|[[u]]| = \{*\}$ dir. Böylece $\Gamma \vdash \exists^{\geq}(u, y)$ dir.

$$\frac{\exists^{\geq}(w, u) \quad \exists^{\geq}(\overset{\vdots}{u}, y)}{\exists^{\geq}(w, u)}$$

Yukarıda görülebileceği gibi ispat ağacından $\exists^{\geq}(w, u)$ elde edilir. O zaman $|[[w]]| = \{*\}$ dir ve buradan $|[[w]]| \geq |[[u]]|$ dir. Eğer $\mathcal{M}$, Γ içindeki her cümleyi doğru yapıyorsa sonucu da doğru yapmak zorundadır. Bu yüzden $|[[x]]| \geq |[[y]]|$ dir. Her durumda $\Gamma \vdash \exists^{\geq}(y, y)$ olduğundan $|[[y]]| = \{*\}$ dir. Böylece $|[[x]]| = \{*\}$ ve $\Gamma \vdash \exists^{\geq}(x, y)$ dir.

□

Açıklama 4.12 İngilizce dilinki bazı sıfatlar bir önceki tanımla verilen $\mathcal{CARD}$ diline eklenecektir. $Kesişen\ sıfatlar$ (IA) isimli sıfatlara red (kırmızı), green (yeşil), white (beyaz), ... ve çoğul isimlere isimlere cats (kediler), cars (arabalar), students (öğrenciler), ... örnekleri verebilebilir. Bu dilde, sintakslar $basic\ nouns$(basit isimler) $x,\ y,\ z,\ ...$ ve kesişen sıfatlar (*intersecting adjectives IA*) (Keenan, E. L. and Faltz, L. M., 1985) ise $a_1,\ a_2,\ a_3,\ ...$ ile verilecektir. Bir kompleks isim ($complex\ noun\ a\ x$) bir isimdir öyle ki x bir basit isim ($basic\,noun$) ve a bir kesişen sıfattır. $x,\ y,\ z$ değişkenleri basit isimler, $p,\ q,\ r$ ise isimler için kullanılacaktır. $\exists^{\geq}(p, q)$ cümlelerinin kombinasyonlarını içeren bu dile $\mathcal{CARD}$ + **IA** denir. Semantikler, sonlu cümle kümeleri ve sonlu M evreni üzerine inşa edilecek-

tir. Her basit isim x için, $[[x]] \subseteq M$. $a\ x$ isminin semantiği $[[a\ x]] = [[a]] \cap [[x]]$ ile tanımlanır. Bir $\mathcal{M} = (M, [[\ ]])$ modeli

"$\mathcal{M} \vDash \exists^{\geq}(p,q)$ eğer ve yalnız eğer $|[[p]]| \geq |[[q]]|$, $\mathcal{M}$ nin içindedir"

doğruluk özelliğine sahip olacaktır.

$$\frac{}{\exists^{\geq}(p,p)}\ (ax1) \qquad \frac{}{\exists^{\geq}(x,a\,x)}\ (ax2)$$

$$\frac{\exists^{\geq}(p,q) \quad \exists^{\geq}(q,r)}{\exists^{\geq}(p,r)}\ (T)$$

Tablo 4.4: $\mathcal{CARD}$ + IA için kurallar tablosu

Teorem 4.13 $\mathcal{CARD}$ + IA lojiği Tablo 4.4 deki kurallar altında sağlam ve tamdır (Öner, T. and Topal, S., 2014).

Kanıt Sağlamlık ispatı daha elementer olduğu için atıf yapılan makalede ve $\mathcal{CARD}$ sağlamlık ispatına benzer olduğu için Teorem 4.8 de bulunabilir. Burada sadece tamlık ispatı verilecektir. $\mathcal{CARD}$ için inşa edilen model burada da kullanılacaktır. x, y ve z basit isimler ve p, q, m, n ve r basit ya da kompleks isimler için kullanılacaktır. Gösterilecektir ki eğer $\Gamma \vDash \varphi$ ise $\Gamma \vdash \varphi$ dir. $\Gamma \vDash \exists^{\geq}(p,n)$ olduğunu kabul edelim. Aşağıda verilen semantik özel olarak tek elemanlı M evrene sahip modeli üzerinde inşa edilecektir.

$$[[q]] = \begin{cases} \{*\}, & \text{eğer } \Gamma \vdash \exists^{\geq}(q,n) \\ \varnothing\ , & \text{aksi taktirde} \end{cases}$$

İddia edelim ki $\mathcal{M} \vDash \Gamma$. O zaman Γ daki φ için $\mathcal{M} \vDash \varphi$ dir. Keyfi bir $\exists^{\geq}(\alpha_1, \alpha_2)$ cümlesi alınsın. α_1, x ve α_2, y olsun ve $|[[x]]| \geq |[[y]]|$ olduğunu gösterelim. $[[y]] \neq \varnothing$ olduğunu kabul edelim. O zaman $[[y]] = M$. Böylece $\Gamma \vdash \exists^{\geq}(y,n)$ dir. Aşağıdaki ispat ağacı vardır.

$$\frac{\exists^{\geq}(x,y) \quad \exists^{\geq}(\overset{\vdots}{y},n)}{\exists^{\geq}(x,n)}$$

O zaman $\Gamma \vdash \exists^{\geq}(x, n)$ dir. Modelin tanımından $[[x]] = \{*\}$ dir. O zaman $|[[x]]| \geq |[[y]]|$ dir. α_1, $red\, x$ ve α_2, y olsun ve $|[[red\, x]]| \geq |[[y]]|$ olduğunu gösterilmeliyiz. $[[y]] = \{*\}$ alalım. Böylece $\Gamma \vdash \exists^{\geq}(y, n)$ dir. Aşağıdaki ispat ağacı vardır.

$$\frac{\exists^{\geq}(red\, x, y) \quad \exists^{\geq}(\vdots\, y, n)}{\exists^{\geq}(red\, x, n)}$$

O zaman $\Gamma \vdash \exists^{\geq}(red\ x, n)$. Modelin tanımından $[[red\ x]] = \{*\}$. O zaman $|[[red\ x]]| \geq |[[y]]|$. α_1 , x ve α_2 , $red\ y$ olsun ve gösterilmelidir ki $|[[x]]| \geq |[[red y]]|$. $[[red y]] = \{*\}$ alılım. Böylece $\Gamma \vdash \exists^{\geq}(red y, n)$ dir. Aşağıdaki ispat ağacı vardır.

$$\frac{\exists^{\geq}(x, red\, y) \quad \exists^{\geq}(\vdots\, red\, y, n)}{\exists^{\geq}(x, n)}$$

O zaman $\Gamma \vdash \exists^{\geq}(x, n)$ dir. Modelin tanımından $[[x]] = \{*\}$ dir. O zaman $|[[x]]| \geq |[[red\ y]]|$ dir. α_1, $red\ x$ ve α_2, $red\ y$ ve gösterilmelidir ki $|[[red\ x]]| \geq |[[red\ y]]|$. Kabul edelim ki $[[red\ y]] \neq \varnothing$ olsun. O zaman $[[red\, y]] = M$. Böylece $\Gamma \vdash \exists^{\geq}(red\, y, n)$ dir. Aşağıdaki ispat ağacı vardır.

$$\frac{\exists^{\geq}(red\, x, red\, y) \quad \exists^{\geq}(\vdots\, red\, y, n)}{\exists^{\geq}(red\, x, n)}$$

O zaman $\Gamma \vdash \exists^{\geq}(red\, x, n)$ dir. Modelin tanımından $[[red\, x]] = \{*\}$ dir. O zaman $|[[red\, x]]| \geq |[[red\, y]]|$ dir.

$\mathcal{M} \vDash \exists^{\geq}(p, n)$ var. Kabul edelim ki n ve p basit isimler olsun. $(ax1)$ den $[[n]] = \{*\}$ dir. O zaman $[[p]] = \{*\}$ dir. Böylece $\Gamma \vdash \exists^{\geq}(p, n)$ dir. n basit isim ve p kompleks isim olsun. $(ax1)$ den $[[n]] = \{*\}$ dir ve o zaman p, $red\, z$ formundadır. $(ax2)$ kullanılarak $[[z]] = \{*\}$ dir . $[[n]] = \{*\}$ dan $[[red]] = \{*\}$ dir. Böylece $\Gamma \vdash \exists^{\geq}(red\, z, n)$ $\clubsuit$. n, $red\, x$ formunda olsun. $(ax2)$ kullanılarak $[[x]] = \{*\}$ dir. Böylece $[[n]] = \{*\}$ dir. p basit isim ise $[[p]] = \{*\}$ dir ve bir kompleks isim ise, $green\ y$ formundadır. $(ax1)$ den $[[n]] = \{*\}$ dir ve böylece $[[x]] = \{*\}$ ve $[[red]] = \{*\}$ dir. $\clubsuit$ kullanılarak $\Gamma \vdash \exists^{\geq}(green\ y, x)$ dir. T kuralı kullanılarak $\Gamma \vdash \exists^{\geq}(green\, y, red\, x)$ dir. $\square$

5. DOĞAL LOJİK, ALGORİTMALAR ve VERİ YAPILARI

Doğal lojik akıl yürütme süreçlerini inceleyen ve bunların kurallarını anlamayı amaçlayan bir lojik alanıdır. Bu akıl yürütme çalışmaları yapılırken özellikle doğal dil içerisindeki kuralların ortaya çıkması amaçlanmıştır. Bu kurallar lojik temelli incelendiği için dildeki kurallar lojiksel formlardan oluşur. Bu nedenle sembolik lojik çalışmaları bu alan için bir ön adım oluşturur. Doğal lojik dil, bilgisayar, bilişsel alanların ortak alanında yer aldığı için çok disiplinli bir yapıya sahiptir.

Lojik ile dilin ortak çalışmalarının ilk örneği George Lakoff tarafından başlatılmış ve kitabında lojik, grametik yapılar ve dil içerisindeki türetimler üzerinde durmuştur (Lakoff G., 1972). Daha sonra özellikle düzenli olarak Van Benthem'in bir çok kitap ve makaleleri bu alana hizmet etmiştir (Van Benthem, J. F. A. K., 1986), (Van Benthem, J. F. A. K., 1987), (Van Benthem, J. F. A. K., 1991) (Van Benthem, J. F. A. K. and Ter Meulen, A., 1996). Ayrıca Bill MacCartney'nin de bu alana önemli katkıları olmuştur (MacCartney, B., Manning, C. D., 2007), (MacCartney, B. and Manning, C. D., 2008) ve (MacCartney, B. and Manning, C. D., 2014). Son 10 yıllık dönem içersinde özellikle doğal lojiğin sillojistik cümlelerin zaman karmaşıklığı, karar verilebilirlik, lojik ve dil bilimleri alanlarına Ian Pratt-Hartmann ve Lawrence S. Moss (Moss, L. S., 2008), (Moss, L. S., 2010), (Pratt-Hartmann, I., 2004), (Pratt-Hartmann, I. and Moss, L. S., 2009), (Pratt-Hartmann, I. and Third, A., 2006) katkılar sağlamışlardır. Bunlara ek olarak metinsel türetimler ve doğal lojik arasında kurulan ilişkiler, bilgisayar ortamları için yaratılmış pek çok gelişmeler de olmuştur (Third, A., 2006), (Van Eijck, J., 2007), (Schubert, L. K., Van Durme, B. and Bazrafshan, M., 2010), (Djalali, Alex J., 2013), (Winter, Y., 2013).

Bu çalışmada temel olarak doğal dil içerisindeki sillojistik yapıların bilgisayar ortamına nasıl aktarılacağı üzerinde duralacaktır. Bu nedenle bilgisayar bilimlerinin, dil bilimlerinin, matematiğin, lojiğin ve internet uygulamalarının nasıl bir araya getirileceğinden bahsedilecektir.

Bu kısımda mantıksal türetimlerin ve türetim olmadığı taktirde bir karşıt model bulunması için gerekli olan algoritmalardan bahsedilecek ve Türkçe doğal dilinde anlaşılabilir kodlar verip daha sonra uygulamalar için kullanılan Python dili için kullanılan modüller ve veri yapıları üzerinde durulacaktır. Ayrıca programlara ait komut dosyalarının tüm kodlarını vermenin bir anlamı olmadığı için önemli olan ve konsepti kapsayıcı noktalar vurgulanacaktır.

5.1 Türetim ve Karşıt Model Algoritmaları

Türetim algoritmaları lojiğin bilgisayar uygulamaları için temel adımdır. Bu anlamda bir türetim meydana gelmesi için graf algoritmasının seçimi en önemli adımdır. Bilgisayar uygulamaları için bir tamlık teoremi içindeki sintaks ve semantikler arasındaki bağlantılar bu algoritmaların inşası için en önemli ip uçlarını verir.

Uyarı 5.1 Bu kısım ve sonraki kısımlar için $\leftarrow$ sembolü atama anlamında kullanılacaktır. Yani $p \leftarrow q$, q her ne değere sahip ise bunun p ye kopyalanması anlamına gelir.

5.1.1 S ve $S^{\dagger}$ için türetim algoritmaları

Algoritma 5.2 Verilen bir sonlu $\Gamma \subseteq S^{\dagger}$ cümleler ve D değişkenler kümesi için $\Gamma \vdash c$ olup olmadığına dair basit algoritma diline yakın *sözde kod*:

```
#If,Bool,Elseif,End, Print, Goto, Stop ve Endif dışında
# büyük harfle yazılan ifadeler kullanıcı tarafından
# oluşturulan fonksiyonlardır.
# All(), Some(), No() ve Celiski()
Begin
(1) p ve q değişkenlerini D kümesine ekle
(2)If {c = All p are q} Then
......For  x,y in  D:
......Bool(Celiski(x,y)) celiskidegeri isimli değişkene ata.
......If{celiskidegeri = True} Then Print "Tutarsızlık" ve
.............................. Goto(16) Endif. EndFor.
(3).... Elseif bool(All(p,q))=True Then Print All(p,q)
(4).... Else Bool(No(p,p))=g
(5).........if g=True Then Print No(p,p) Endif.
(6).........Else Print KarşıtModelAll((p,q)) and Goto(16)
(7)....Endif.
(8)Endif.
(9)Elseif {c = Some p are q} Then Print Some(p,q)
(10)Else Print KarşıtModel(Some(p,q)) ve Goto(16)
(11)Endif.
(12)Elseif {c = No p are p} Then
```

```
(13).....If bool(No(p,q))=True Then Print No(p,q)
(14).....Endif.
(15).....Else Print KarşıtModel(No(p,q)) ve Goto(16)
(16)Endif.
    Stop.
```

Uyarı 5.3 Bu kısımda kısaltmalara gitmek ve anlaşılmayı kolaylaştırmak amacıyla türetimleri Tanım 2.5 de verilen **KEYG** ile şayet türetim yok ise **KEYG** ifadelerinin önüne *değilleme* sembolü olan ¬ kullanılarak temsil edilecektir:

(i) $\Gamma \vdash All\ p\ are\ q$ için $p \overset{All}{\longmapsto} q$ kullanılacaktır. Ayrıca $p \overset{All}{\longmapsto} q$, p den q ya bir yol var anlamındadır,

(ii) $\Gamma \vdash$ *Some p are q* için $p \overset{Some}{\longmapsto} q$ kullanılacaktır,

(iii) $\Gamma \vdash$ *No p are q* için $p \overset{No}{\longmapsto} q$ kullanılacaktır,

(iv) $p \overset{All}{\longmapsto} q$ türetimi yok ise $\neg(p \overset{All}{\longmapsto} q)$,

(v) $p \overset{Some}{\longmapsto} q$ türetimi yok ise $\neg(p \overset{Some}{\longmapsto} q)$,

(vi) $p \overset{No}{\longmapsto} q$ türetimi yok ise $\neg(p \overset{No}{\longmapsto} q)$.

Uyarı 5.4 Aşığıdaki verilecek tüm gösterilimler ve notasyonlar için (Moss, L. S., 2008) başvurulabilir.

Algoritma 5.5 S lojiği içinde bir Γ cümleler kümesinden, bir *All p are q* ifadesinin türetilip türetilemeyeceğini bulmak için aşağıdaki durumlar kontrol edilir:

(i) Tablo 4.1 deki kurallar altında All lojiği yansıyan geçişken kapanış özelliğine sahiptir (Moss, L. S., 2008),

(ii) $\frac{No\ p\ are\ p}{All\ p\ are\ q}$ kuralı takip edilebilir ve eğer burada da bir sonuç alınmazsa *Ex Falso Quadlibet* kuralı kontrol edilebilir,

(iii) *All p are q* cümlesinin kümeye ait olması durumumda zaten $p \overset{All}{\longmapsto} q$ elde edilmektedir.

Algoritma 5.6 S lojiği içinde bir Γ cümleler kümesinden, bir *Some p are q* ifadesinin türetilip türetilemeyeceğini bulmak için aşağıdaki durumlar kontrol edilir (Pratt-Hartmann, I. and Moss, L. S., 2009):

i) *Some q are p* veya *Some p are q* nin kümeye ait olması,

ii) $\exists\, a,\, b$ için $a \overset{All}{\longmapsto} p$ ve $b \overset{All}{\longmapsto} q$ ve *Some a are b* $\in \Gamma$,

iii) $\exists\, a,\, b$ için $b \overset{All}{\longmapsto} p$ ve $a \overset{All}{\longmapsto} q$ ve *Some a are b* $\in \Gamma$,

iv) *Ex Falso Quadlibet* kuralı.

Algoritma 5.7 S lojiği içinde bir Γ cümleler kümesinden, bir *No p are q* ifadesinin türetilip türetilemeyeceğini bulmak için aşağıdaki durumlar kontrol edilir (Pratt-Hartmann, I. and Moss, L. S., 2009):

i) *No q are p* veya *No p are q* nin kümeye ait olması,

ii) $\exists\, a,\, b$ için $p \overset{All}{\longmapsto} a$ ve $q \overset{All}{\longmapsto} b$ ve *No a are b* $\in \Gamma$,

iii) $\exists\, a,\, b$ için $p \overset{All}{\longmapsto} b$ ve $q \overset{All}{\longmapsto} a$ ve *No a are b* $\in \Gamma$,

iv) *Ex Falso Quadlibet* kuralı.

Açıklama 5.8 $S^{\dagger}$ içinde tutarsızlık çıkabilecek durumların bir listesi aşağıda verilmiştir:

(i) $p \overset{Some}{\longmapsto} q$, $p \overset{All}{\longmapsto} q'$ (*Ex Falso Quadlibet* kuralı),

(ii) $p \overset{All}{\longmapsto} p'$, $p' \overset{All}{\longmapsto} p$, $m \overset{Some}{\longmapsto} n$,

(iii) $p \overset{Some}{\longmapsto} p'$.

Durum (ii) (Pratt-Hartmann, I. and Moss, L. S., 2009) çalışmasından sonuç olarak elde edilmiştir.

Önerme 5.9 $\Gamma \subseteq S^{\dagger}$ cümlelerin bir kümesi olsun. $\mathcal{M}_{\Gamma}$, Γ kümesinin oluşturulan model olsun. Eğer $\Gamma \vdash p \overset{All}{\longmapsto} p'$, $\Gamma \vdash p' \overset{All}{\longmapsto} p$ ve $\Gamma \vdash m \overset{Some}{\longmapsto} n$ ise $\mathcal{M}_{\Gamma}$ tutarsızdır.

Kanıt Γ cümleler kümesi ve $p \overset{All}{\longmapsto} p'$, $p' \overset{All}{\longmapsto} p$ ve $m \overset{Some}{\longmapsto} n$ olsun. O zaman evren M, en az bir eleman içermelidir öyke ki $M = \{*\}$, $m = \{*\}$ ve $n = \{*\}$ olsun. $p \overset{All}{\longmapsto} p'$ ise $[[p]] = \{\}$ ve $p' \overset{All}{\longmapsto} p$ ise $[[p]] = M$. Bu iki durumu aynı anda sağlayacak tek durum $M = \{\}$ dir. Fakat $m \overset{Some}{\longmapsto} n$ dolayı M en az bir eleman içermeliydi. Bu ise çelişkidir. Yani model tutarsızdır. □

Önerme 5.10 $\Gamma \subseteq S^{\dagger}$ cümlelerin bir kümesi olsun. $\mathcal{M}_{\Gamma}$, Γ kümesinden oluşturulan model olsun. Eğer $p \overset{Some}{\longmapsto} p'$ ise $\mathcal{M}_{\Gamma}$ tutarsızdır (Pratt-Hartmann, I. and Moss, L. S., 2009).

Kanıt $p \overset{Some}{\longmapsto} p'$ olsun. $S^{\dagger}$ tam olduğu için öyle bir $\mathcal{M}_{\Gamma}$ vardır ki $\mathcal{M}_{\Gamma} \vDash$ *Some p are* p' ve M en az bir elemana sahiptir. Bu ise $[[p]] \cap [[p']] \neq \varnothing$. $[[p']] = M \setminus [[p]$ demektir. Fakat böyle bir küme inşası imkansız olduğu için model tutarsızdır. □

Algoritma 5.11 $S^{\dagger}$ lojiği içinde bir Γ cümleler kümesinden, bir *All p are q* ifadesinin türetilip türetilemeyeceğini bulmak için aşağıdaki durumlar kontrol edilir (Pratt-Hartmann, I. and Moss, L. S., 2009):

(1) p den q ya bir yol,

(2) $p \overset{All}{\longmapsto} p'$,

(3) $q' \overset{All}{\longmapsto} q$,

(4) (Açıklama 5.8).

Burada *All p are q* cümlesinin kümeye ait olması durumumda zaten *All p are q* elde edilir.

Algoritma 5.12 $S^{\dagger}$ lojiği içinde bir Γ cümleler kümesinden, bir *Some p are q* ifadesinin türetilip türetilemeyeceğini bulmak için aşağıdaki durumlar kontrol edilir (Pratt-Hartmann, I. and Moss, L. S., 2009):

(1) *Some q are p* kümeye ait olması,

(2) $\exists\, a,\, b$ için $a \overset{All}{\longmapsto} p$ ve $b \overset{All}{\longmapsto} q$ ve *Some a are b* $\in \Gamma$,

(3) $\exists\, a,\, b$ için $b \overset{All}{\longmapsto} p$ ve $a \overset{All}{\longmapsto} q$ ve *Some a are b* $\in \Gamma$,

(4) (Açıklama 5.8).

Burada *Some p are q* cümlesinin kümeye ait olması durumumda zaten *Some p are q* elde edilmektedir (Pratt-Hartmann, I. and Moss, L. S., 2009). Ayrıca sorguda şayet girdi *Some p are* p' tipinde ise sorgu yapmanın bir anlamı yoktur.

Algoritma 5.13 En kısa yol algoritması S, $S\dagger$ için bir All cümlesi ve $\mathcal{CARD}$+IA için $ax1$ ve T kurallarının uygulandığı ispatlara cevap verir. Eğer bir yol var ise en kısa yol yani ispatın en sade ve en az adımda uygulamış hali görüntülenir fakat eğer bir yol yok ise $None$ döner.

```
#En Kısa Yol Algortiması

def ENK(graph, start, end, path=[]):
.... path = path +[start]
.... if start == end:
........ return path
.... if not graph.has_key(start):
........ return None
.... shortest = None
.... for node in graph[start]:
........ if node not in path:
............ newpath = ENK(graph, node, end, path)
............ if newpath:
.............. if not shortest or len(newpath)<len(shortest):
................... shortest = newpath
.... return shortest
```

Algoritma 5.14 S dili içersinde tutarsızlık kontrolü için verilen bir öncüller kümesinden $\Gamma \vdash$ *Some p are q* ve $\Gamma \vdash$ *No p are q* türetimleri elde edilmesi gerektiğini daha önceki bölümlerde ifade edilmişti. Bunun Python da bir uygulamasını verelim. Aşağıdaki tutarsızlık fonksiyonunun algoritması *False* döndüğünde bu bir tutarsızlık olmadığını ve *True* döndüğünde tutarsızlık olduğu anlamına gelir.

```
def Tutarsızlık():
....h=False
....for i in graph:
........if h==True:
............break
........for j in graph:
............if h==True:
................break
............if (bool(Some(i,j)) and (bool(No(i,j))
................h=True
................break
....return h
```

Açıklama 5.15 $S\dagger$ içerisindeki tutarsızlık algoritmalarının bir taslağını Açıklama 5.8 de bulunabilir.

5.1.2 S ve $S^{\dagger}$ için karşıt model algoritmaları

Algoritma 5.16 S lojiği içinde girdilere atanan semantik değerler:

1. *All p are q* için $[[p]] \leftarrow \varnothing$ ve $[[q]] \leftarrow \varnothing$ atanır,
2. *Some p are q* için $[[p]] \leftarrow \{p_1, p_2, \exists(p,q)\}$ ve $[[q]] \leftarrow \{q_1, q_2, \exists(p,q)\}$ atanır,
3. *No p are q* için, eğer $p = q$ ise $[[p]] = [[q]] \leftarrow \varnothing$, değilse $[[p]] \leftarrow \{p_1\}$ ve $[[q]] \leftarrow \{q_1\}$ atanır.

Algoritma 5.17 $S^{\dagger}$ lojiği içinde girdilere atanan semantik değerler:
Her p ve q girdi atomları $[[p]] \leftarrow \varnothing$ ve $[[q]] \leftarrow \varnothing$ ataması yapılır.

Algoritma 5.18 S lojiği içinde sorguların türetilmemesi sonucu karşıt modellerin oluşturulma aşamaları:

1. $\neg(p \overset{All}{\longmapsto} q)$ ise $[[q]] \leftarrow [[q]] \cup \{q_1\}$ ataması yap ve modeli güncelle,
2. $\neg(p \overset{No}{\longmapsto} q)$ ise $[[q]] \leftarrow [[q]] \cup \{q_1, q_2, \exists(p,q)\}$ ve $[[p]] \leftarrow [[p]] \cup \{p_1, p_2, \exists(p,q)\}$ atamalarını yap ve modeli güncelle,
3. $\neg(p \overset{Some}{\longmapsto} q)$ ise $[[q]] \leftarrow [[q]] \cup \varnothing$ ve $[[p]] \leftarrow [[p]] \cup \varnothing$ atamalarını yap ve modeli güncelle.

Algoritma 5.19 $S^{\dagger}$ lojiği içinde sorguların türetilmemesi sonucu karşıt modellerin oluşturulma aşamaları:

Değişkenlere küme atamaları:

1. $\neg(non-p \overset{All}{\longmapsto} q)$ ise $[[non-p]] \leftarrow \{1\}$ ve $[[non-q]] \leftarrow \{1\}$ ataması yap ve modeli güncelle,
2. $\neg(p \overset{All}{\longmapsto} non-q)$ ise $[[p]] \leftarrow \{1\}$ ve $[[q]] \leftarrow \{1\}$ ataması yap ve modeli güncelle,
3. $\neg(non-p \overset{All}{\longmapsto} non-q)$ ise $[[non-p]] \leftarrow \{1\}$ ve $[[q]] \leftarrow \{1\}$ ataması yap ve modeli güncelle,
4. $\neg(p \overset{All}{\longmapsto} q)$ ise $[[p]] \leftarrow \{1\}$ ve $[[non-q]] \leftarrow \{1\}$ ataması yap ve modeli güncelle,

5. $\neg(non-p \overset{Some}{\longmapsto} non-q)$ ise $[[q]] \leftarrow [[non-p]]$ ataması yap ve modeli güncelle,
6. $\neg(p \overset{Some}{\longmapsto} q)$ ise $[[non-q]] \leftarrow [[p]]$ ataması yap ve modeli güncelle,
7. $\neg(non-p \overset{Some}{\longmapsto} q)$ ise $[[p]] \leftarrow [[q]]$ ataması yap ve modeli güncelle,
8. $\neg(p \overset{Some}{\longmapsto} non-q)$ ise $[[q]] \leftarrow [[p]]$ ataması yap ve modeli güncelle.

Model inşası:

1. $sayac = 0$,
2. $Model = \{\}$ oluştur,
3. $Some = \{\}$ ve $All = \{\}$ oluştur,
4. Tüm $some$ girdilerini $Some$ ve tüm all girdilerini All kütüphanesine aktar,
5. Her p, q için eğer $p \overset{Some}{\longmapsto} q$ ise:
   - $sayac = sayac + 1$,
   - $[[p]] \longleftarrow \{sayac\}$ ve $[[q]] \longleftarrow \{sayac\}$
   - $Model[p] = set([sayac])$, $Model[q] = set([sayac])$
6. $Evren = set([])$
7. Her $[[p]] \in Model$ için $Evren = Evren.union(Model([[p]]))$,

Evren ve model güncelleme:

1. Her p ve q için eğer $p \overset{All}{\longmapsto} q$ ise $[[q]] \longleftarrow [[p]]$
2. Her p için p ***non***-sız ise $[[non-p]] \longleftarrow Evren \setminus [[p]]$
3. Her p için p ***non***- içeriyor ise p nin ***non***-sız olan kısmı x için $[[x]] \longleftarrow Evren \setminus [[p]]$.

5.1.3 $\mathcal{CARD}$ ve $\mathcal{CARD}$ + IA için türetim algoritmaları

Bu kısımda $\mathcal{CARD}$ + IA dili $\mathcal{CARD}$ dilinden daha zengin olduğu ve $\mathcal{CARD}$ dilindeki her cümle $\mathcal{CARD}$ + IA dili tarafından üretildiği ve ayrıca $\mathcal{CARD}$ için özel bir durum gerekmediği için sadece $\mathcal{CARD} + IA$ için türetimler ve karşıt model alogitmaları üzerinde durmak yeterli olacaktır.

Uyarı 5.20 Bu kısımda, kısaltmalara gitmek ve anlaşılmayı kolaylaştırmak amacıyla türetimleri Tanım 2.5 de verilen **KEYG** ile eğer türetim yok ise **KEYG** ifadelerinin önünde değilleme sembolü olan ¬ kullanarak temsil edilecektir:

1. $\Gamma \vdash \exists^{\geq}(p, q)$ için $p \overset{\mathcal{CARD}^{\mathcal{IA}}}{\longmapsto} q$ kullanılacaktır,
2. $p \overset{\mathcal{CARD}^{\mathcal{IA}}}{\longmapsto} q$ türetimi yok ise $\neg(p \overset{\mathcal{CARD}^{\mathcal{IA}}}{\longmapsto} q)$ kullanılacaktır.

Uyarı 5.21 $\mathcal{CARD}$ ve $\mathcal{CARD}$ + IA için türetim algoritmalarını (ÖNER, T. and TOPAL, S, 2014) isimli makalede verilmiştir. Bu kısımda sadece karşıt modeller üzerinde durulacaktır.

Uyarı 5.22 $\mathcal{CARD}$ + IA için türetim algoritmaları girdileri için $p,\ q$ sembollerini kompleks (red cats, blue dogs vs.) isimler için ve x, y sembollerini basit (cats, dogs vs..) isimler için ve ayrıca $red\ x$ ve $blue\ y$ nin kısaltmaları için $r\ x$ ve $b\ y$ ve kompleks ya da basit isimlerin önemli olmadığı durumlarda m, n, k, h, v, w gibi notasyonlar kullanılacaktır.

Tanım 5.23 Bir **ilkel çaprazlama kümesi** (PPS_Γ), bir Γ cümleler kümesine ait tüm değişkenlerin A sıfat kısımlarının kümesi ile N basit isimlerin ve K kompleks isimlerin basit kısımlarının bir araya getirildiği ve tüm basit kısımları da içeren tüm olası isimlerin oluşturduğu kümedir. PPS^* ise PPS kümesi ile sorgu cümlesinin isimleri ile birleşiminden elde edilen kümenin ilkel çaprazlama kümesidir.

Örnek 5.24 $\Gamma = \{\exists^{\geq}(red\ x, blue\ y), \exists^{\geq}(green\ y, h)\}$ ise Γ için ilkel çaprazlama kümesi olan PPS_Γ:

$PPS_\Gamma = \{redx, bluey, greeny, redh, blueh, greenh, bluex, redy, greenx, x, y, h\}$.

5.1.4 $\mathcal{CARD}$ + IA için karşıt model algoritması

Algoritma 5.25 $\mathcal{CARD}$ + IA lojiği içinde sorguların türetilmemesi sonucu karşıt modellerin oluşturulma aşamaları:

Eğer $\neg(v \overset{\mathcal{CARD}^{\mathcal{IA}}}{\longmapsto} w)$ ise:

1. Her $h \in PPS^*$ isimleri için $[[h]] \leftarrow \varnothing$ ve model evreni $M \leftarrow \varnothing$ için ataması yapılır,
2. $[[w]] \leftarrow \{0, 1\}$ ve $[[v]] \leftarrow \{1\}$ atamaları yapılır,

3. Tüm sıfatlar ile tüm kompleks isimlerin basit kısmı ve tüm basit isimlerin, Γ içindeki her m ismi için eğer $x \overset{\mathcal{CARD}^{\mathcal{IA}}}{\longmapsto} m$ ise $[[m]] \leftarrow \{1\}$ ataması yapılır,

4. Γ içindeki her n ismi için eğer $n \overset{\mathcal{CARD}^{\mathcal{IA}}}{\longmapsto} y$ ise $[[n]] \leftarrow \{0, 1\}$ ataması yapılır,

5. Her $h, k \in PPS^*$ isimleri için:

   (i) Eğer $h \overset{\mathcal{CARD}^{\mathcal{IA}}}{\longmapsto} k$ ise $[[k]] \leftarrow [[h]]$ ata ve güncelle,

   (ii) Eğer $h \overset{\mathcal{CARD}^{\mathcal{IA}}}{\longmapsto} k$ ise $[[h]] \leftarrow [[h]] \cup [[k]]$ ata ve güncelle.

5.2 Lojiklerin Doğal Dildeki Gramer Yapıları

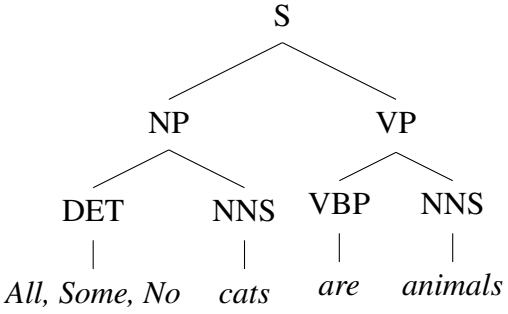
Bu kesimde sonraki kısımda algoritmik olarak incelenecek olan S, $S^{\dagger}$, $\mathcal{CARD}$ ve $\mathcal{CARD}$ + IA dillerine ait gramer bilgilerini verilecektir. Tanımlar (Manning, C., D., Schütze, H., 1999) den elde edilebilir.

Tanım 5.26 Bir **sözcük türü** bir cümle içerisinde sorulan belli sorulara karşılık kelimelerin davranış biçimiyle ortaya çıkan bir dilsel kategoridir.

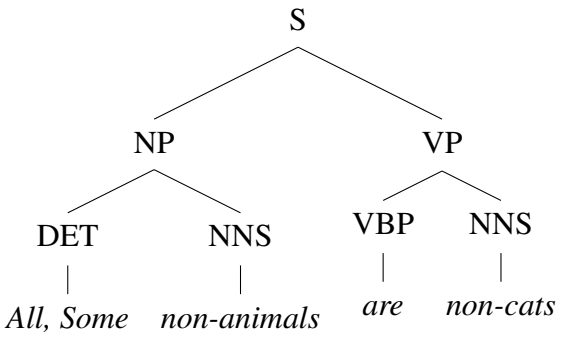
Tanım 5.27 Bir **bütünceme dilbilimi** gerçek dünya metinlerinin (doğal dil) örnekler içerisinde ifade edilerek çalışılan dil çalışmasıdır.

Tanım 5.28 Bir **sözcük türü etiketleme** bir bütünce içerisindeki kelimeleri sözcük türüne göre işaretlemektir.

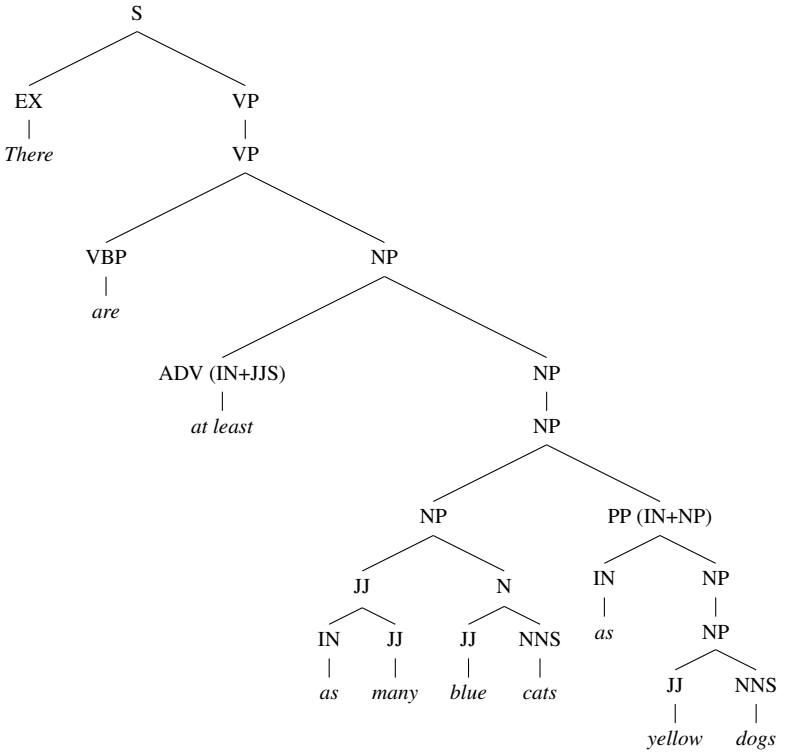
Tanım 5.29 Bir **seçim bölgeli çözümleme ağacı** bir bütünce içerisindeki bir kök, dallar ve düğümler ile sözcük türü etiketlenmesidir.



Şekil 5.1: S dili için bir seçim bölgeli çözümleme ağacı



Şekil 5.2: $S^{\dagger}$ dili için bir seçim bölgeli çözümleme ağacı



Şekil 5.3: $\mathcal{CARD}$ + IA dili için bir seçim bölgeli çözümleme ağacı

Açıklama 5.30 Şekil 5.1, Şekil 5.2 ve Şekil 5.3 de diller için verilen seçim bölgeli çözümleme ağaçlarının son düğümleri sözcükleri ve bir üst düğümleri sahip oldukları sözcük türlerini göstermektedir. Diller içerisinde sözcüklerin sahip olduğu etiketleri ağaçlar yardımıyla elde edip Tablo 5.1 elde edilmiştir.

Sintaktik Sembol	**Anlam**
NNS	Çoğul isim
JJ	Sıfat
JJS	En üstünlük sıfatı
DT	Belirleyici
IN	Edat
RBS	En üstünlük zarfı
Ex	Varlıksal *there*
VBP	Çoğul geniş zaman fiili

Tablo 5.1: Sözcük öbekleri tablosu

Örnekler 5.31 Tablo 5.1 içinde yer alan sözcük öbeklerini dillere ait cümleler üzerinde etiketli olarak gösteren bazı örnekler aşağıda verilmiştir:

(i) { (‘All’, ‘DT’), (‘cats’, ‘NNS’), (‘are’, ‘VBP’), (‘animals’, ‘NNS’)}

(ii) { (‘All’, ‘DT’), (‘cats’, ‘NNS’), (‘are’, ‘VBP’), (‘non-animals’, ‘NNS’) }

(iii) { (‘All’, ‘DT’), (‘non-cats’, ‘NNS’), (‘are’, ‘VBP’), (‘non-animals’, ‘NNS’) }

(iv) { (‘there’, ‘EX’), (‘are’, ‘VBP’), (‘at’, ‘IN’), (‘least’, ‘JJS’), (‘as’, ‘IN’), (‘many’, ‘JJ’), (‘cats’, ‘NNS’), (‘as’, ‘IN’), (‘dogs’, ‘NNS’) }

(v) { (‘there’, ‘EX’), (‘are’, ‘VBP’), (‘at’, ‘IN’), (‘least’, ‘JJS’), (‘as’, ‘IN’), (‘many’, ‘JJ’), (‘disable’, ‘JJ’), (‘cats’, ‘NNS’), (‘as’, ‘IN’), (‘dogs’, ‘NNS’) }

(vi) { (‘there’, ‘EX’), (‘are’, ‘VBP’), (‘at’, ‘IN’), (‘least’, ‘JJS’), (‘as’, ‘IN’), (‘many’, ‘JJ’), (‘yellow’, ‘JJ’), (‘cats’, ‘NNS’), (‘as’, ‘IN’), (‘blue’, ‘JJ’), (‘dogs’, ‘NNS’) }

(vii) { (‘at’, ‘IN’), (‘least’, ‘JJS’), (‘as’, ‘IN’), (‘many’, ‘JJ’), (‘cats’, ‘NNS’), (‘as’, ‘IN’), (‘dogs’, ‘NNS’), (‘exist’, ‘VBP’) }

(viii) { (‘at’, ‘IN’), (‘least’, ‘JJS’), (‘as’, ‘IN’), (‘many’, ‘JJ’), (‘blue’, ‘JJ’), (‘cats’, ‘NNS’), (‘as’, ‘IN’), (‘black’, ‘JJ’), (‘dogs’, ‘NNS’), (‘exist’, ‘VBP’) }

(ix) { (‘there’, ‘EX’), (‘at’, ‘IN’), (‘least’, ‘JJS’), (‘as’, ‘IN’), (‘many’, ‘JJ’), (‘cats’, ‘NNS’), (‘as’, ‘IN’), (‘there’, ‘EX’), (‘are’, ‘VBP’), (‘dogs’, ‘NNS’) }

5.3 Veri Yapıları

Bu kısımda, daha önce ayrıntıları verilen lojiklerin Python programlama dilinde nasıl inşa edileceği üzerinde durulacaktır.

5.3.1 Programlar (komut dosyaları) için Python veri tipleri

(i) **None** veri tipi olmayan, boş veya bilinmeyen şeklinde değerlendirilebilir.

(ii) **Boole değerler** *True* ve *False* olmak üzere iki tiptir. Herhangi bir değişken ya da veri *True* ve *False* değeri alır. Bu bool değerler ayrıca eşitlik ve değer karşılaştırması için kullanılabilir.

(iii) **Sayılar** tam sayılar, float sayılar (reel sayılar) ve kompleks sayılardır.

(iv) **Karakter dizileri** metinsel verilerdir ve ya ′ ′ ya da ¨ ¨ tırnak işaretleri arası formdadır.

(v) **Demetler** (23,¨demet ¨, ¨32¨) gibi içerisinde sayı ya da karakter dizi tipi barındırır ve üzerinde hiçbir değişim yapılamaz.

(vi) **Listeler** [23,¨demet¨, ¨32¨] gibi demetlerin üzerinde değişim ve işlemler yapılabilir formudur.

(vii) **Kümeler**, üzerinde birleşim, kesişim, fark gibi klasik küme işlemlerinin de yapilabildiği veri tipleridir. set([¨a¨, ¨b¨, ¨3 ¨, ¨2 ¨, ¨1 ¨]) şeklinde bir sintaksa sahiplerdir. Boş küme set([]) şeklinde ifade edilir.

(viii) **Sözlükler**, anahtar-değer sıralı ikililerinden oluşan veri tipleridir. {¨araba¨:5000} formunda olurlar ve anahtar ve değer verileri birer liste, demet veya küme de olabilirler.

5.3.2 Girdi, sorgu ve model inşası için veri tipleri ve kontrolleri

Girdi ve sorgu ifadeleri doğal dildeki ifadelerdir. Bu nedenle İngilizcenin konuşulan dilindeki ifadelerini aynen yansıtarak ve bu şekilde program yaparak doğal lojik programına daha iyi bir katkı sağlanabilir. Bu anlamda *All cats are dogs, All cars are non-horses*, *There are at least as many students as teachers* gibi ifadeleri karakter dizileri, listeler ve sözlüklerle Python için anlamlı hale getireceğim.

Açıklama 5.32 Her bir lojik için girdi ve sorgu veri tiplerine bakacağız. Burada a, b, c, d gibi harfler birer değişken olarak ifade edilecektir ve herbirinin tipi atamalar sonucunda karakter dizisine dönüşecektir. = sembolü atamaları temsil edecektir. p ve q içinde bulundukları dilin isim türlerini gösterecektir. Bu kısımda ve sonrasında Python sintaksına uygun olarak = atama ve == eşitlik kontrolü anlamında kullanılacaktır. Sözcüklerin türleri bir Python modülü olan "NLTK" ve kelimelerin İngilizce dilindeki doğal ve kullanılan kelimeler olup olmadığının kontrolü ise NLTK'in bir alt modülü olan "Wordnet Interface" yapılacaktır (Bird, S. 2006).

$S^{\dagger}$ ve $\mathcal{CARD}$ + IA için veri tipleri:

1. Öncül ve sorgu girdileri aşaması:

   (a) *All p are q* gibi bir girdi için $a =' $ *All p are q*$'$ karakter dizisi elde edilir.

   (b) *Some p are q* gibi bir girdi için $b =' $ *Some p are q*$'$ karakter dizisi elde edilir.

   (c) *No p are q* gibi bir girdi için $c =' $ *No p are q*$'$ karakter dizisi elde edilir.

   (d) *There are at least as many p as q* gibi bir girdi için $d =' $ *There are at least as many p as q*$'$ karakter dizisi elde edilir.

2. Parçalama ve sözcük türü denetleme:

   (a) a girdisi boşluklara göre parçalanır ve sonra $'All' ==' DT'$, $'are' ==' VBP'$, $'p' ==' NNS'$ ve $'q' ==' NNS'$ eşitliklerinin kontrolü sağlanır.

   (b) b girdisi boşluklara göre parçalanır ve sonra $Some == DT$, $'are' ==' VBP'$, $'p' ==' NNS'$ ve $'q' ==' NNS'$ eşitliklerinin kontrolü sağlanır.

   (c) c girdisi boşluklara göre parçalanır ve sonra $'No' ==' DT'$, $'are' ==' VBP'$, $'p' ==' NNS'$ ve $'q' ==' NNS'$ eşitliklerinin kontrolü sağlanır.

   (d) d girdisi boşluklara göre parçalanır ve sonra $'There' ==' DT'$, $are == VBP$, $'at' ==' IN'$, $'least' ==' JJS'$, $'as' ==' IN'$, $'many' ==' JJ'$,

$'p' == 'NNS'$, $'as' == 'IN'$ ve $'q' == 'NNS'$ eşitliklerinin kontrolü sağlanır. (p ve q sıfat içeriyorsa JJ kontrolü de yapılır).

3. Cümlelerin türlerine göre ayıklanması:

   (a) p ve q isimleri ile $All = \{p : q\}$ sözlüğü elde edilir.

   (b) p ve q isimleri ile $Some = \{p : q\}$ sözlüğü elde edilir.

   (c) p ve q isimleri ile $No = \{p : q\}$ sözlüğü elde edilir.

   (d) p ve q isimleri ile $There = \{p : q\}$ sözlüğü elde edilir.

4. İsimlerin semantiklere dönüştürülmesi:

   (a) Bir p ismi için boş küme ataması $Model['p'] = set([])$ şeklinde yapılır.

   (b) Bir p ismi için eleman ataması $Model['p'] = set(['p'_1,'p'_2,'\exists(p,q)'])$ şeklinde yapılır.

5.4 Halting ve Sonsuz Döngüden Kurtulmak

Tanım 5.33 Bir **karar problemi** bir koşulun sağlanıp sağlanamadığını evet-hayır şeklinde ikili olarak sorgulamaktır (Church, A., 1936).

Örnek 5.34 Verilen bir $G = (V, E)$ grafı, $V = \{a, b, c\}$ tepe kümesi ve $E = \{(a, b), (b, c)\}$için a ve c tepeleri arasında bir yol var mıdır?

Evet vardır ve yol a, b, c yoludur.

Tanım 5.35 David Hilbert'in 23 sorusundan biri olan "birinci mertebe lojiğin tüm totolojilerinin bir kümesi var mı?" sorusundan doğan **halting problemi**, verilen bir program ve bir girdi için programın çalışmasının durup durmayacağına karar verme problemidir (Turing, A., 1937).

Örnek 5.36 24 girdisi için program asla durmaz. Fakat 25 ve daha büyük sayı girdisi için ilk adımda durur.

```
While x<25:
      x=:x-1
```

Açıklama 5.37 $S\dagger$ ve $\mathcal{CARD}$ + IA lojiklerinin sonlu bir öncüller kümesi Γ dan bir φ cümlesinin türetilip türetilemeyeceği problemini durmayan problemi olmaktan kurtaran ve programın "evet türetilebilir" ya da "hayır türetilemez" cevabını vermesini sağlayan özellikleri şunlardır:

- φ bir $There$ cümlesi ise, sonlu bir bağıntı kümesinin yansıyan geçişken kapanışının da yine bir sonlu küme olması (Öner, T. and Topal, S. 2014),
- φ bir All cümlesi ise, sonlu bir bağıntı kümesinin yansıyan geçişken kapanışının da yine bir sonlu küme olması,
- φ bir $Some$ cümlesi ise, Algoritma 5.17 dan bir *Some p are q* cümlesinin türetilip türetilemeyeceğini saptamak için kontrol edilmesi gereken *Some a are b*$\in \Gamma$ aranması özelliği,
- φ bir No cümlesi ise, Algoritma 5.17 dan bir *No p are q* cümlesinin türetilip türetilemeyeceğini saptamak için kontrol edilmesi gereken *No a are b* $\in \Gamma$ aranması özelliği.

5.5 Teknik Bilgiler

Bu kısımda ilk olarak tez kapsamında Corcoran sillojistik lojiği (CSL) ve S lojiğinin (MSL) cümlelerinin türetim algoritmlarının zaman karmaşıklıklarının bir karşılaştırmasını ve lojiklere ait cümlelerin birbiri içinde ifade edilebilirliğini içeren (Topal, S. and Öner, T., 2014) makelede elde edilen sonuçlar verilecektir. Zaman karmaşıklıları hakkında açıklayıcı olabilecek Tablo 5.2 bazı zaman sınıflarının girdilere bağlı olarak davranışlarını verir. Daha sonra lojiklerin türetimleri ve karşıt modelleri için kullanılan ana fonksiyonların ve modüllerin 8 GB RAM, 64-bit işletim sistemi, 2.40Ghz işlemciye sahip bir bilgisayarda girdi-zaman karşılaştırması yapılarak performansları değerlendirilecektir. Her bir fonksiyon için girilen veri sayısı ve zaman ikilisi şeklinde üç nokta seçip Lagrange interpolasyonu (Whittaker, E. T. and Robinson, G, 1967) yardımıyla fonksiyonlar ve bu fonksiyonlar ait grafikler verilecektir.

O notasyonu	10 girdi için süre	100 girdi için süre
$O(1)$	1	1
$O(log\, n)$	3	7
$O(n)$	10	100
$O(n\, log\, n)$	30	700
$O(n^2)$	100	10000
$O(2^n)$	1024	2^{100}
$O(n!)$	3628800	100!

Tablo 5.2: 10 ve 100 adet girdi için asimptotik üst sınır değerleri

Açıklama 5.38 Bu tez kapsamında Corcoran sillojistik lojiği (CSL) ve S lojiğinin (MSL) cümlelerinin türetim algoritmlarının ve zaman karmaşıklıklarının bir karşılaştırması yapılan (Topal, S. and Öner, T., 2014) makeleden bir karşılaştırma tablosu aşağıdaki gibidir.

φ cümlesi	φ MSL de mi?	φ CSL de mi?	$\Gamma_{MSL} \vdash \varphi$ (?)	$\Gamma_{CSL} \vdash \varphi$ (?)
All p are q	✓	✓	$O(7n^3) + O(8n)$	$O(22n^3) + O(27n)$
Some p are q	✓	✓	$O(6n^3) + O(8n)$	$O(27n^3) + O(30n)$
No p are q	✓	✓	$O(6n^3) + O(8n)$	$O(25n^3) + O(30n)$
Some p are are not q	×	✓	×	$O(25n^3) + O(39n)$

Tablo 5.3: Zaman karmaşıklığı karşılaştırmaları ($n = |\mathcal{P}|$)

Γ, girdi cümlelerinin bir kümesini; $|\mathcal{P}|$, girdi cümlelerinin kümesinin atomlarını; Γ_{MSL}, MSL diline ait cümleler kümesini ve Γ_{CSL} , CSL diline ait cümleler kümesini gösterir. Hesaplamalar, notasyonlar ve daha fazlası için aynı makaleye başvurulabilir. Yukarıdaki tablodan *Some p are not q* cümlesi S diline ait olmasa da yani Corcoran lojiğinin dili daha zengin olsa da S lojiğinin Corcoran lojiğine oranla türetimleri çok daha hızlı verdiği görülür. Öte yandan *No p are p* cümlesi Corcoran lojiği içinde modelin tutarsızlığına sebep olurken S lojiğinde *All p are q* türetimi için bir kural olarak yer alır.

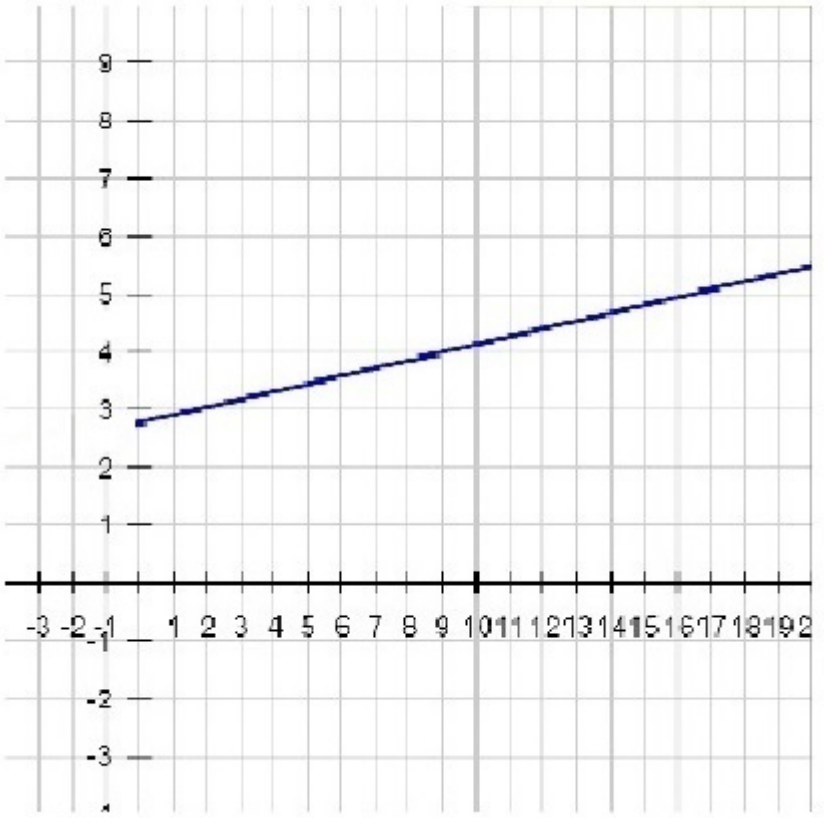
5.5.1 Fonksiyon ve komut dizini testleri

Teorem 5.39 $S^{\dagger}$ lojiğinin dili içerisindeki herhangi bir cümleler kümesinin doğrulanabilirliğinin saptanması problemi P zaman sınıfı, hatta $O(n^2)$ içindedir (Pratt-Hartmann, I and Third A., 2006).

Teorem 5.40 (Cormen, T. H., Leiserson, C. E., Rivest, R. L. and Stein, C., 2001) Verilen bir n tepeli yönlü G grafının yansıyan geçişken kapanışı olan G^{*} grafının hesaplanması $O(n^3)$ içindedir.

Açıklama 5.41 $S^{\dagger}$ dili içerisinde $Zero$, One, $Antitone$ ve karşıt model için oluşturulan fonksiyonların 10, 100 ve 1000 uzunluklu sözlükler için çalışma süreleri aşağıdaki gibidir. Bu değerlere karşılık gelen koordinat değerleri yaklaşık olarak (10, 4.312), (100, 16.375) , (1000, 128.759) şeklindedir. Bu üç adet ikilinin oluşturacağı fonksiyon ve grafik aşağıdaki gibidir.

$f(x) \approx -0.001x^2 + 0.136x + 2.376$



Şekil 5.6: $S^{\dagger}$ da $Zero$, One, $Antitone$ kuralları ve karşıt modeller için grafik

All, $Some$ ve çelişki arama fonksiyonları da çok yaklaşık değerlere sahip olduğu için kısaltma için yer verilmeyecektir.

```
>>> def test():
Zero
L=[]
for i in range(10):
L.append(i)
>>> if __name__==’__main__’:
from timeit import Timer
t=Timer("test()", "from __main__ import test")
print t.timeit()

4.4789950249

>>> def test():
```

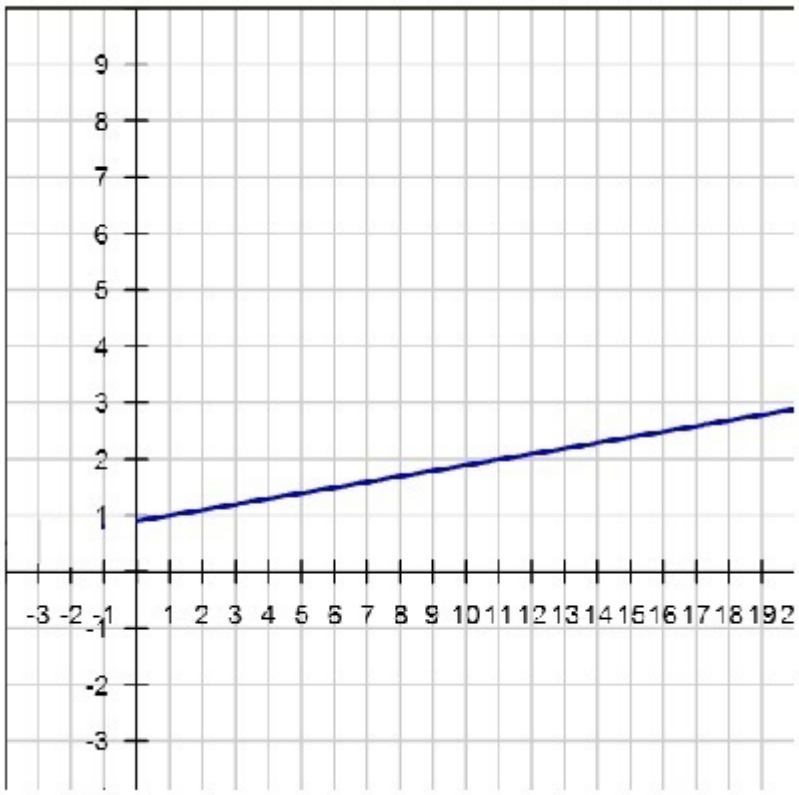
```
Zero
L=[]
for i in range(100):
L.append(i)
>>> if __name__==’__main__’:
from timeit import Timer
t=Timer("test()", "from __main__ import test")
print t.timeit()

13.\begin{9847086484D
>>> def test():
Zero
L=[]
for i in range(1000):
L.append(i)

>>> if __name__==’__main__’:
from timeit import Timer
t=Timer("test()", "from __main__ import test")
print t.timeit()

123.050429644
------------------------------------------------
>>> def test():
One
L=[]
for i in range(10):
L.append(i)

>>> if __name__==’__main__’:
from timeit import Timer
t=Timer("test()", "from __main__ import test")
print t.timeit()

4.31252891878

>>> def test():
One
L=[]
```

```
for i in range(100):
L.append(i)

>>> if __name__==’__main__’:
from timeit import Timer
t=Timer("test()", "from __main__ import test")
print t.timeit()

16.3758222341

>>> def test():
One
L=[]
for i in range(1000):
L.append(i)

>>> if __name__==’__main__’:
from timeit import Timer
t=Timer("test()", "from __main__ import test")
print t.timeit()

128.759086162
-----------------------------------------------
def test():
Antitone
L=[]
for i in range(10):
L.append(i)

>>> if __name__==’__main__’:
from timeit import Timer
t=Timer("test()", "from __main__ import test")
print t.timeit()

4.49082681405

>>> def test():
Antitone
L=[]
```

```
for i in range(100):
L.append(i)

>>> if __name__==’__main__’:
from timeit import Timer
t=Timer("test()", "from __main__ import test")
print t.timeit()

20.210036922

>>> def test():
Antitone
L=[]
for i in range(1000):
L.append(i)

>>> if __name__==’__main__’:
from timeit import Timer
t=Timer("test()", "from __main__ import test")
print t.timeit()

206.868146624
```

Açıklama 5.42 S dili içerisindeki tüm türetimler ve karşıt model için oluşturulan fonksiyonların 10, 100 ve 1000 uzunluklu sözlükler için çalışma süreleri aşağıdaki gibidir. Bu değerlere karşılık gelen koordinat değerleri yaklaşık olarak (10, 1.893), (100, 10.842), (1000, 98.629) oluşturacağı fonksiyon ve grafiği:

$$f(x) \approx -0.001x^2 + 0.99x + 0.898$$



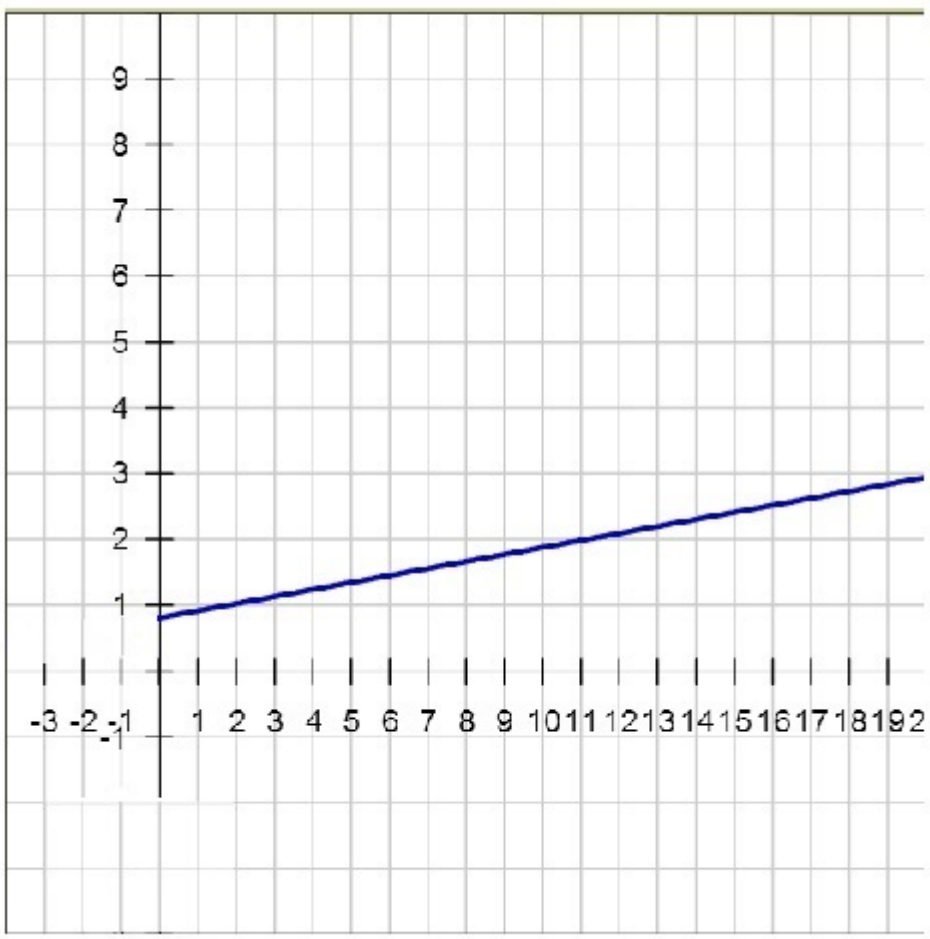
Şekil 5.7: S lojiği ve karşıt model modülü için grafik

```
>>> def test():
Slojik
L = []
for i in range(10):
L.append(i)

>>> if __name__==’__main__’:
from timeit import Timer
t = Timer("test()", "from __main__ import test")
print t.timeit()

1.89306915789

>>> def test():
Slojik
L = []
for i in range(100):
L.append(i)
```

```
>>> if __name__==’__main__’:
from timeit import Timer
t = Timer("test()", "from __main__ import test")
print t.timeit()

10.8422503663

>>> def test():
Slojik
L = []
for i in range(1000):
L.append(i)

>>> if __name__==’__main__’:
from timeit import Timer
t = Timer("test()", "from __main__ import test")
print t.timeit()

98.6294432285
```

Açıklama 5.43 S dili içerisindeki tüm türetimler ve karşıt model ve NLTK modülü için oluşturulan fonksiyonların 10, 100 ve 1000 uzunluklu sözlükler için çalışma süreleri aşağıdaki gibidir. Bu değerlere karşılık gelen koordinat değerleri yaklaşık olarak (10, 1.55298509697), (100, 9.31559195193), (1000, 82.3807380957) şeklindedir. Bu üç adet ikilinin oluşturacağı fonksiyon ve grafik aşağıdaki gibidir.

$$f(x) \approx -0.001x^2 + 0.086x + 0.682$$



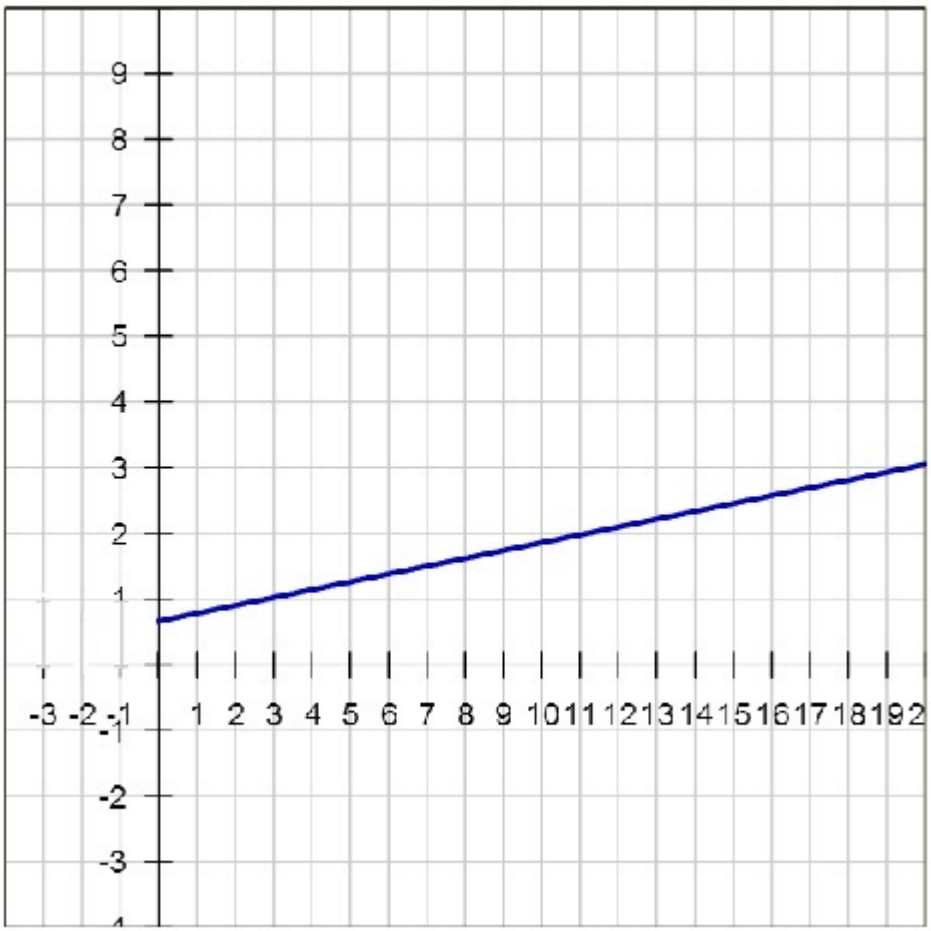
Şekil 5.8: S lojiği, karşıt model ve NLTK modülü için grafik

```
>>> def test():
SyllojisticsNLTK
L = []
for i in range(10):
L.append(i)

>>> if __name__==’__main__’:
from timeit import Timer
t = Timer("test()", "from __main__ import test")
print t.timeit()

1.55298509697
```

```
>>> def test():
SyllojisticsNLTK
L = []
for i in range(100):
L.append(i)

>>> if __name__==’__main__’:
from timeit import Timer
t = Timer("test()", "from __main__ import test")
print t.timeit()

9.31559195193

>>> def test():
SyllojisticsNLTK
L = []
for i in range(1000):
L.append(i)

>>> if __name__==’__main__’:
from timeit import Timer
t = Timer("test()", "from __main__ import test")
print t.timeit()

82.3807380957
```

Açıklama 5.44 $S\dagger$ dili içerisindeki tüm türetimler ve karşıt model için oluşturulan fonksiyonların NLTK modülü ile gramer ve cümle yapısı kontrollerini içeren modülün 10, 100 ve 1000 uzunluklu sözlükler için çalışma süreleri aşağıdaki gibidir. Bu değerlere karşılık gelen koordinat değerleri yaklaşık olarak (10, 1.866), (100, 12.473), (1000, 96.298) şeklindedir. Bu üç adet ikilinin oluşturacağı fonksiyon ve grafik aşağıdaki gibidir.

$$f(x) \approx -0.001x^2 + 0.117x + 0.664$$



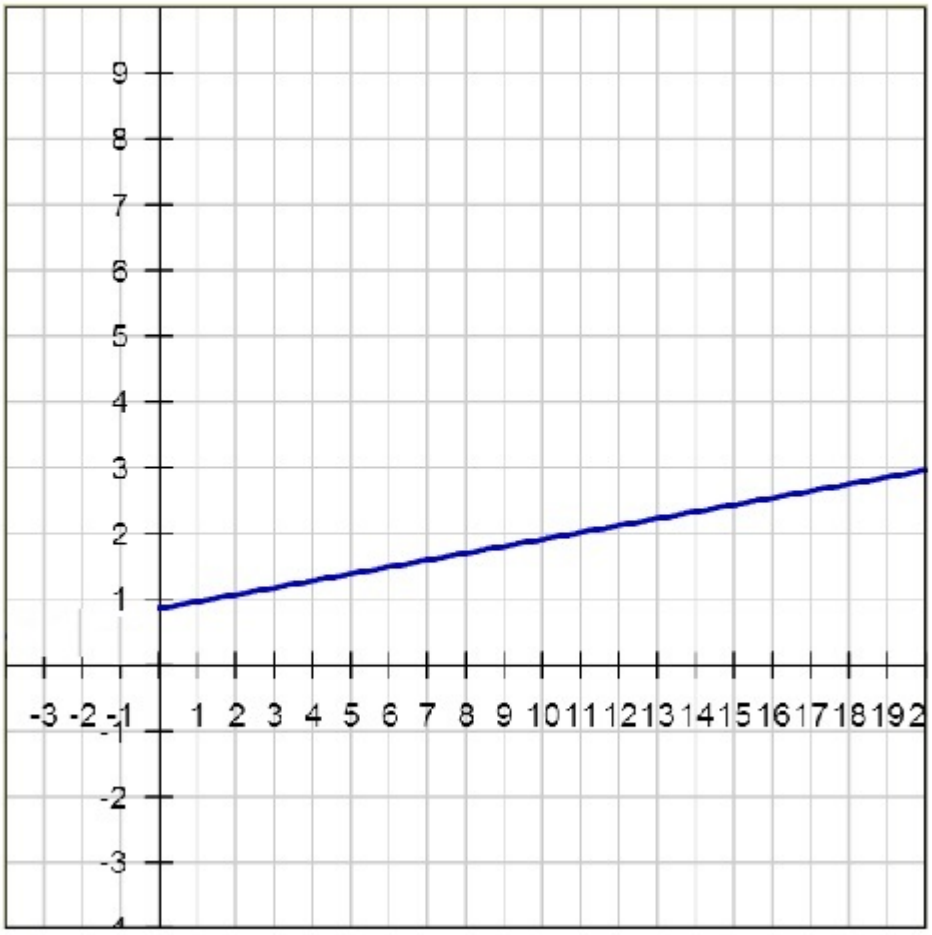
Şekil 5.9: $S\dagger$ lojiği, karşıt model ve NLTK modülü için grafik

```
>>>def test():
SdaggerCMefficentNLTK()
L=[]
for i in range(10):
L.append(i)
```

```
>>> if __name__ == '__main__':
import timeit
print(timeit.timeit("test()", setup="from __main__ import test"))

1.86640558216

>>> def test():
SdaggerCMefficentNLTK()
L=[]
for i in range(100):
L.append(i)

>>> if __name__ == '__main__':
import timeit
print(timeit.timeit("test()", setup="from __main__ import test"))

12.4738401617

>>> def test():
SdaggerCMefficentNLTK()
L=[]
for i in range(1000):
L.append(i)

>>> if __name__ == '__main__':
import timeit
print(timeit.timeit("test()", setup="from __main__ import test"))

96.2980921986
```

Açıklama 5.45 $\mathcal{CARD}+IA$ dili içerisindeki tüm türetimler ve karşıt model için oluşturulan fonksiyonların NLTK modülü ile gramer ve cümle yapısı kontrollerini içeren modülün 10, 100 ve 1000 uzunluklu sözlükler için çalışma süreleri aşağıdaki gibidir. Bu değerlere karşılık gelen koordinat değerleri yaklaşık olarak (10, 1.911), (100, 11.355), (1000, 97.231) şeklindedir. Bu üç adet ikilinin oluşturacağı fonksiyon ve grafik aşağıdaki gibidir.

$$f(x) \approx -0.001x^2 + 0.105x + 0.857$$



Şekil 5.10: $\mathcal{CARD}+IA$ lojiği, karşıt model ve NLTK modülü için grafik

```
>>> def test():
CArdwithIA
L = []
for i in range(10):
```

```
L.append(i)

>>> if __name__==’__main__’:
from timeit import Timer
t = Timer("test()", "from __main__ import test")
print t.timeit()

1.91104795439

>>> def test():
CArdwithIA
L = []
for i in range(100):
L.append(i)

>>> if __name__==’__main__’:
from timeit import Timer
t = Timer("test()", "from __main__ import test")
print t.timeit()

11.3559493015

>>> def test():
CArdwithIA
L = []
for i in range(1000):
L.append(i)

>>> if __name__==’__main__’:
from timeit import Timer
t = Timer("test()", "from __main__ import test")
print t.timeit()

97.2311521601
```

6. SONUÇ

Bu tezde temel olarak doğal dil ve doğal lojik kapsamında sillojistik akıl yürütmelerin bilgisayar ortamına nasıl aktarılacağı üzerinde durulmuştur. Bu aktarım sürecinde karşılaşılan problemler, akıl yürütmelerin tarihsel boyutları, bu akıl yürütmeler için oluşturulan lojiklerin tamlık teorileri, türetim ve karşıt model algoritmaları, doğal dil ile entegrasyonu ve modüllerin teknik analizleri üzerinde durulmuştur.

Giriş bölümünde, tezin kapsamı, tezin oluşmasında anlam teşkil eden tarihsel bilgiler ve tezin hizmet etmesi istenilen alanlardan bahsedilmiştir.

İkinci kısım olan ön bilgiler bölümünde, tezde kullanılacak olan kavramların tanımları verilmiştir.

Üçüncü kısımda, Aristo sillojizmi, Aristo sillojizmine felsefi ve hesaplamalı lojik pencerelerinden bakış açıları aktarmak amacıyla, Corcoran sillojizminin modern bir tablosu verildi. Son kısımda tezde kullanılacak olan Moss'un sillojistik lojikleri üzerinde durulmuştur.

Dördüncü kısımda, Moss'un sillojistik lojikleri tanıtılmış ve bu tez kapsamında (ÖNER, T. and TOPAL, S., 2014) de tamlık ispatı yapılan ve türetim algoritması verilen birinci mertebe dili aşan, $\mathcal{CARD}$ + IA lojiği tanıtılmıştır. Beşinci kısımda, doğal lojiğin amacı ve doğal lojik kapsamında taranan literatür bilgisi verilmiş ve bu bölümün türetim algoritmları kısmında tanıtılan lojiklerin türetim ve karşıt model algoritmaları üzerinde durulmuştur. Daha sonra bu lojiklerin doğal dil içindeki gramer bilgileri için açıklamalar yapılmıştır. Veri yapıları kısmında, kullanılacak olan girdi, sorgu ve model inşası için kullanılacak veri tipleri tanıtıldı. Sonsuz döngüden nasıl kurtulunacağından söz edilmiştir. Bu tez kapsamında Corcoran sillojistik lojiği ve Moss'un S lojiğinin cümlelerinin türetim algoritmlarının ve zaman karmaşıklıklarının bir karşılaştırması yapılan (Topal, S. and Öner, T., 2014) makelede yer alan sonuçlara değinilmiştir. Uygulaması yapılan lojiklerin cümlelerinin İngilizce'deki doğal kullanımlarının kullanıcı tarafından doğru kullanımı test edilerek modüllere yansıtılmıştır. Modüllerin ve modülleri oluşturan ana fonksiyonların girdi sayılarına göre ne kadar sürede sonuçlar vereceğinin ölçümü yapılıp, referans üç nokta seçilerek lagrange interpolasyonu yardımı ile grafiksel yaklaşık fonksiyonel hareketlerinin doğrusal olduğu gözlemlenmiştir.

KAYNAKLAR DİZİNİ

Andrade, E. J. and Becerra, E., 2007, Corcoran's Aristotelian syllogistic as a subsystem of first-order logic, Revista Colombiana de Matemàticas, 41(1): 67-80 pp.

Bang-Jensen, J. and Gregory G., 2000, Digraphs, Theory: Algorithms and Applications, Springer Verlag, 798p.

Bird, S., 2006, NLTK: the natural language toolkit, Proceedings of the COLING/ACL on Interactive presentation sessions, Association for Computational Linguistics.

Carruccio, E., 1964, Mathematics and logic in history and in contemporary thought, Aldine Transaction, 398p.

Champin, P. A. and Solnon C., 2003, Measuring the Similarity of Labeled Graphs, Case-Based Reasoning Research and Development, K. Ashley and D. Bridge, Springer Berlin Heidelberg, 2689: 80-95 pp.

Chierchia, G. and McConnell-Ginet, S., 2000, Meaning and Grammar, An Introduction to Semantics, The MIT Press, 2nd ed., 591p.

Cormen, T. H., Leiserson, C. E., Rivest, R. L. and Stein, C., 2001, Transitive closure of a directed graph, Introduction to algorithms, Cambridge: MIT press, Vol. 2, 632-634 pp.

Djalali, A. J., 2013, Synthetic logic, Linguistic Issues in Language Technology 9.

Fillion, N., 2007, Two Accounts of Aristotle's Logic, The University of Western Ontario, 39p.

Gamut, L. T. F., 1991, Logic, Language and Information, Volume 1: Introduction to Logic, University of Chicago Press, xiv+282p.

Iyanaga, S. and Kawada, Y., 1980, Encyclopedic Dictionary of Mathematics, MIT Press, Cambridge, MA, 1005p.

Keenan, E.L. and Faltz, L.M., 1985, Boolean Semantics for Natural Language, Synthese Language Library, Vol. 23., D. Reidel Publishing Co., Dordrecht.

Khemlani, S. and Johnson-Laird P., 2012, Theories of the syllogism: A meta-analysis, Psychological bulletin, 138(3): 427-457 pp.

Lakoff, G., 1972, Linguistics and natural logic, Semantics of natural language, Springer, 545-665p.

Li, Z., 2008, Classification Syllogism in the Farm Vehicle Design Based on the Kansei Engineering, Advanced Materials Research, 44: 703-710 pp.

Lukasiewicz, J., 1957, Aristotle's Syllogistic From the Standpoint of Modern Formal Logic, Oxford University Press, 222p.

KAYNAKLAR DİZİNİ (devam)

MacCartney, B. and Manning, C. D., 2014, Natural Logic and Natural Language Inference, Computing Meaning, Springer, 129-147 pp.

MacCartney, B. and Manning, C. D., 2007, Natural logic for textual inference, Proceedings of the ACL-PASCAL Workshop on T.E.P., ACL.

MacCartney, B. and Manning, C. D., 2008, Modeling semantic containment and exclusion in natural language inference, Proceedings of the 22nd International Conference on Computational Linguistics-Volume: 1, ACL.

Manning, C. D. and Schütze, H., 1999, Foundations of Statistical Natural Language Processing, Cambridge, Massachusetts: The MIT Press, xxxvii + 680p.

Miller, G. A., 1995, WordNet: a lexical database for English, Communications of the ACM, 38(11): 39-41 pp.

Moss, L. S., 2008, Completeness Theorems for Syllogistic Fragments, *in F. Hamm and S. Kepser (eds.) Logics for Linguistic Structures, Mouton de Gruyter*, 143-173 pp.

Moss, L. S., 2010, Intersecting adjectives in syllogistic logic, The Mathematics of Language, Lecture Notes in Computer Science, Volume 6149, Springer, 223-237 pp.

Moss, L. S., 2011, Syllogistic logics with complements, in J. van Benthem, A. Gupta and E. Pacuit (eds.), Games, Norms and Reasons: Logic at the Crossroads, Springer Synthese Library Series, 185-203 pp.

Moss, L. S., 2010, Syllogistic logics with verbs, *Journal of Logic and Computation*, 20(4): 761-793 pp.

Normore, C. G., 1999, Two Some Aspects of Ockham's Logic, *The Cambridge Companion to Ockham: 31*, 440p.

Öner, T. and Topal, S., 2014, On Completeness and Algorithms of Logic of $\mathcal{CARD}$ with IA, *Annals of Pure and Applied Logic*, (submitted).

Parsons, T., 1999, The Traditional Square of Opposition, *The Standford Encyclopedia of Philosophy*, http://plato.stanford.edu/.

Pratt-Hartmann, I. and Moss, L. S., 2009, Logics for the relational syllogistic, *The Review of Symbolic Logic*, 2(04): 647-683 pp.

Pratt-Hartmann, I. and Third, A., 2006, More fragments of language, *Notre Dame Journal of Formal Logic*, 47(2): 151-177 pp.

KAYNAKLAR DİZİNİ (devam)

Pratt-Hartmann, I., 2004, Fragments of language, *Journal of Logic, Language and Information*, 13(2): 207-223p.

Rose, L. E., 1968, Aristotle's Syllogistic, *Springfield, III.*

Schubert, L. K., Van Durme, B. and Bazrafshan, M., 2010, Entailment Inference in a Natural Logic-like General Reasoner, *AAAI Fall Symposium*.

Sipser, M., 2006, Introduction to the Theory of Computation, *Thomson Course Technology*, Tampa, 456p.

Smiley, T. J., 1973, What is a Syllogism?, *Journal of Philosophical Logic*, 2(1): 136-154 pp.

Smith, R., 1989, Aristotle: Prior Analytics, *translated with introduction notes and commentary, Indianapolis, Hackett*, 265p.

Third, A., 2006, Logical Analysis of Fragments of Natural Language, *University of Manchester, for the degree of Doctor of Philosophy*, 160p.

Thom, P., 1981, The syllogism, *Munich: Philosophia*, 197-198 pp.

Thom, P., 2007, Logic and Ontology in the Syllogistic of Robert Kilwardby, *Studien Und Texte Zur Geistesgeschichte Des Mittelalters*, 316p.

Topal, S. and Öner, T., 2014, A Note on Computational and Expressiveness Comparisons of Two Aristotelian Logics, *Ars Combinatoria*, (submitted).

Turing, A. M., 1937, On Computable Numbers, with an Application to the Entscheidungsproblem, *Proc. London Math. Soc. Ser.*, 2(42), 230-265 pp.

Ulman, J. D. and Hopcrioft J. E., 2006, Introduction to Automata Theory, Languages and Computation, *3rd ed., Addison-Wesley*, 750p.

Van Benthem, J. F. A. K. and Ter Meulen, A., 1996, Handbook of Logic and Language, *Amsterdam: Elsevier Science Publishers*, 1168 pp.

Van Benthem, J. F. A. K., 1986, *Essays in logical semantics, No: 29, Dordrecht: Reidel*.

Van Benthem, J. F. A. K., 1987, *Categorial grammar and lambda calculus, Springer*.

Van Benthem, J. F. A. K., 1987, Meaning: interpretation and inference, *Synthese*, 73(3): 451-470p.

Van Benthem, J. F. A. K., 1991, Language in action, *Journal of Philosophical Logic*, 20(3): 225-263 pp.

Van Eijck, J., 2007, Natural logic for natural language, *Logic, Language and Computation, Lecture Notes in Computer Science, Springer*, 216-230 pp.

Westerståhl, D., 2005, On the Aristotelian square of opposition, *Kapten Mnemos Kolumbarium, en festskrift med anledning av Helge Malmgrens 60-årsdag*.

KAYNAKLAR DİZİNİ (devam)

Whittaker, E. T. and Robinson, G, 1967, Lagrange's Formula of Interpolation, *The Calculus of Observations: A Treatise on Numerical Mathematics, 4th ed. New York: Dover*, 28-30 pp.

ÖZGEÇMİŞ

04.02.1983 tarihinde Osmaniye'de doğdu. İlkokul, orta ve lise öğrenimlerini İskenderun'da tamamladı. Haziran 2007 döneminde Ege Üniversitesi Fen Fakültesi Matematik Bölümü'nden mezun oldu. Aynı yıl 6 ay süreyle Hava Teknik Okullar Komutanlığı Astsubay Meslek Yüksek Okulu'unda Matematik Öğretmeni ve ardından 2008 – 2009 yılları arasında Bornova Belediyesi Genç Kaşifler Bilim Ev'inde Matematik Laboratuar kurucusu ve yöneticisi olarak görev yaptı. 2009 yılında Ege Üniversitesi Fen Bilimleri Enstitüsü Matematiğin Temelleri ve Matematik Lojik Anabilim Dalı'nda yüksek lisansı tamamladı. Ağustos 2010 döneminde Bitlis Eren Üniversitesi Fen Fakültesi Matematik Bölümü Matematiğin Temeleri ve Matematik Lojik Anabilim Dalı ÖYP araştırma görevlisi görevini kazandı. 07.02.2011 tarihinde Ege Üniversitesi Fen Bilimleri Enstitüsü Matematiğin Temelleri ve Matematik Lojik Anabilim Dalı'nda doktora eğitimine ve araştırma görevlisi görevine başladı. 15.02.2013 – 15.11.2013 tarihleri arasında Amerika Birleşik Devletleri İndiana Üniversitesi Matematik Bölümü'nde ve Bilişim ve Hesaplama Bölümü'nde YÖK Doktara Araştırma Görevlisi bursu ile Prof. Dr. Lawrence S. MOSS ve Dr. Muhammad ABDUL-MAGEED ile çalışmalarda bulundu.