T.C.
BİTLİS EREN ÜNİVERSİTESİ VE MUŞ ALPARSLAN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MATEMATİK ANABİLİM DALI
YÜKSEK LİSANS TEZİ

İSİMLERİN KARDİNALİTE KARŞILAŞTIRMASINDAN VE KESİŞEN SIFATLARDAN OLUŞAN BİR SİLLOJİSTİK LOJİĞİN ALGORİTMİK ANALİZİ VE BİR BİLGİSAYAR UYGULAMASI

Yasin AKÜNSOY

AĞUSTOS 2017

MATEMATİK ANABİLİM DALI

YÜKSEK LİSANS TEZİ

İSİMLERİN KARDİNALİTE KARŞILAŞTIRMASINDAN VE KESİŞEN SIFATLARDAN OLUŞAN BİR SİLLOJİSTİK LOJİĞİN ALGORİTMİK ANALİZİ VE BİR BİLGİSAYAR UYGULAMASI

Hazırlayan

Yasin AKÜNSOY

Danışman

Yrd. Doç. Dr. Selçuk TOPAL

Jüri Üyeleri

Prof. Dr. Heybetkulu MUSTAFAYEV

Yrd. Doç. Dr. Ökkeş ÖZTÜRK

Yrd. Doç. Dr. Selçuk TOPAL

AĞUSTOS 2017

Yasin AKÜNSOY tarafından hazırlanan "**İsimlerin Kardinalite Karşılaştırılmasından ve Kesişen Sıfatlardan Oluşan Bir Sillojistik Lojiğin Algoritmik Analizi ve Bir Bilgisayar Uygulaması**" adlı tez çalışması 18/ 08/ 2017 tarihinde yapılan sınavla aşağıdaki jüri tarafından oy birliği ile Bitlis Eren Üniversitesi Fen Bilimleri Enstitüsü Matematik Anabilim Dalı'nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri	**İmza**
Prof. Dr. Heybetkulu Mustafayev (Başkan)	
Yrd. Doç. Dr. Selçuk TOPAL (Danışman)	
Yrd. Doç. Dr. Ökkeş ÖZTÜRK (Üye)	

Bu tezin kabulü, Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 16./12./ 2017..gün ve 41/96 Sayılı kararı ile onaylanmıştır.

Doç. Dr. Koray KÖKSAL
Enstitüsü Müdürü

ÖZET

İSİMLERİN KARDİNALİTE KARŞILAŞTIRMASINDAN VE KESİŞEN SIFATLARDAN OLUŞAN BİR SİLLOJİSTİK LOJİĞİN ALGORİTMİK ANALİZİ VE BİR BİLGİSAYAR UYGULAMASI

Yasin AKÜNSOY

Yüksek Lisans Tezi

Bitlis Eren Üniversitesi Fen Bilimleri Enstitüsü
Matematik Anabilim Dalı
Danışman: Yrd. Doç. Dr. Selçuk TOPAL
AĞUSTOS 2017, 26 sayfa

Hazırlanan çalışmanın giriş bölümünde, tezin oluşmasında rol alan tarihsel bilgiler ve tez hakkında genel bilgi verilmiştir. İkinci kısımda, Aristo sillojizminin klasik ve modern bakış açıları hakkında bilgiler verilmiştir. Ayrıca klasik ve modern sillojizm arasındaki farklılıklardan bahsedilmiştir. Üçüncü kısımda materyaller bölümünde ise tezin daha rahat anlaşılabilmesi için gerekli tanımlar verilmiştir. Dördüncü ve son bölümde $\mathcal{L}(More, IA)$ dilinin lojiğine ait türetim ve karşıt model algoritmaları verildi ayrıca bu algoritmalarınbir uygulaması Python programı kullanılarak oluşturuldu.

Anahtar kelimeler: Tasım, Bilgisayar Biliminde Mantık, Mantık Uygulamaları, Algoritma Analizi, Dil Bilimi

ABSTRACT

ALGORITHMIC ANALYSIS AND A COMPUTER IMPLEMENTATION OF A SYLLOGISTIC LOGIC COMPOSED OF CARDINALITY COMPARISONS OF NOUNS AND INTERSECTING ADJECTIVES

Yasin AKÜNSOY

Master Thesis

Bitlis Eren University Graduate School of Natural and Applied Sciences
Department of Mathematics
Supervisor: Asst. Prof. Dr. Selçuk TOPAL
August 2017, 26 pages

In the introductory part of the prepared work, historical information on the role of the thesis and general information about the thesis are given. In the second part, information about the classical and modern aspects of Aristotele's syllogismis given. Also, the differences between classical and modern syllogism is mentioned. In the third part, in the material section, definitions are given so that the thesis can be understood more easily. In the fourth and last chapter, the derivation and counter model algorithms of $\mathcal{L}(More, IA)$ language are given. In addition, an implementation of these algorithms is built using the Python program.

Keywords: Syllogism, Logic of Natural Language, Logic in Computer Science, Applications of, Algorithmic Analysis, Linguistics

TEŞEKKÜR

Bu çalışmanın planlanmasından yazımına kadar, her aşamasında bana yardımcı olan, beni yönlendiren kıymetli danışman hocam Sayın Yrd. Doç. Dr. Selçuk TOPAL'a sonsuz teşekkür eder, saygılarımı sunarım. Bu süreçte, bana destek veren, cesaretlendiren ve hep yanımda olan aile bireylerim, babam Ethem AKÜNSOY, annem Birgül AKÜNSOY, kardeşlerim Emrah AKÜNSOY ve Veysel AKÜNSOY'a tüm kalbimle teşekkür ederim.

İÇİNDEKİLER DİZİNİ

ŞEKİLLER DİZİNİ

SİMGELER DİZİNİ

IN	Doğal sayılar kümesi
χ	Ex falso quadlibet kuralı
$S^{\dagger}$	İsim düzey tümleme içeren temel sillojistik lojik
$\mathbb{C}$	Kompleks sayılar kümesi
$\vdash$	Mantıksal türetim
$\models$	Modelde doğruluk
IR	Reel sayılar kümesi
S	Temel sillojistik lojik

KISALTMALAR DİZİNİ

CARD+IA	Kesişen sıfatlar içeren *CARD*
CARD	Küme kardinalitesi karşılaştırması içeren sillojistik lojik

1.GİRİŞ

Aristo (M.Ö. 384-322) tarihleri arasında yaşamıştır. Aristo'nun zamanından bu zamana kadar özellikle son 60-70 yıl içerisinde baskın biçimde sillojizm üzerine çok fazla fikir ortaya atılmıştır ve halende bu fikirlerin üretkenliği devam etmektedir [1-2]. Bu yüzden sillojizm, kökeni tarihsel olarak çok eskilere dayanan ve neredeyse Aristo sonrası tarihin tüm periyotlarında kendisinden çok çeşitli branşlarda bahsettiren konudur [3].

Aristo ve Łukasiewicz sillojizme değişik yönlerden bakar, Łukasiewicz sillojizmi daha çok formel bir yapıda görmek ister. Bu doğrultuda Łukasiewicz daha yapısalcıdır.

Bu tez temel olarak, kardinalite temelli cümlelerden oluşan bir lojiğin, türetim olduğunda bu türetim aşamasını ve türetim gerçekleşmediğinde bir ters model veren bir algoritma ve nihayetinde bilgisayar programı oluşturmaya dayanmaktadır.

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, bilgi çıkarımı alanlarına katkı sağlayacaktır.

2. ÖNCEKİ ÇALIŞMALAR

Bu bölümde, klasik ve güncel sillojizm araştırmalarına yer vereceğiz. Bu kısımda sillojizmi kavramını ortaya atan Aristo'nun fikirleriyle yola çıkıp, daha sonra modern yaklaşımlarla devam edeceğiz.

2.1. Aristo Sillojizmi

All men are mortal
All Greeks are men

∴ All Greeks are mortal

Yukarıdaki örnekten yola çıkarak Aristo sillojizmi hakkında bilgiler verilecektir. Başlangıç olarak ∴ simgesinin önündeki cümle bir ***sonuç*** cümlesi olduğunu gösterir. Burada *All men are mortal* ve *All Greeks are men* birer ***öncül*** ve *All Greeks are mortal* cümlesine bir **sonuç** cümlesi denir. *Greeks*, *mortal* ve *men* kelimelerine birer ***terim*** denir. Sonuçta yer almayan fakat her iki öncülde de yer alan men kelimesine **orta terim,** hüküm cümlesinin 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 yada hüküm ***tümel olumlu*** formunda ise ***A tipi*****;** ***tümel olumsuz*** formunda ise ***I tipi*****;** ***tikel olumlu*** formunda ise ***E tipi*****;** ***tikel olumsuz*** formunda ise ***O tipi*** cümledir. Bir sillojizm iki öncül ve bir sonuçtan oluştuğu için sırasıyla büyük öncül, küçük öncül ve sonuç cümlesinin tipleri yan yana yazılarak bir ***mod*** meydana getirilir. Örnek verdiğimiz sillojizm **AAA** modundadır [4].

Yukarıdaki örnekte de gördüğümüz gibi sillojizm de iki önerme ve bir sonuç durumu vardır. Dört tane mod olduğundan bunlar üçer üçer alındığında 64 mümkün mod bulunur. Bu modları düzenlediğimizde 52 mod'un sonuç vermediğini göreceğiz. AEO ve IEO modlarının da sonuç vermediği sillojizm kurallarına bakılarak anlaşılabilir. Bu iki durumda sillojizmin kurallarının ihlal edilmesinden değil, kurallar sonucu kalıbın geçersizliğinden dolayı ortaya çıkmasıdır. O halde sonuç vermeyen modların sayısı toplam 54 tane olurken sonuç verenler toplam 10 tanedir. Aşağıda toplam 64 tane mod belirtilmiş ve geçerli modlar kalın harflerle yazılmıştır.

1. **AAA**	2. AAE	3. **AAI**	4. AAO
5. AEA	6. AEO	7. AEI	8. **AEE**
9. AIA	10.AIE	11.**AII**	12.AIO
13.AOA	14.AOE	15.AOI	16.**AOO**
17.EEE	18.EEA	19.EEO	20.EEI
21.**EAE**	22.**EAO**	23.EAI	24.EAA
25.EIA	26.EIE	27.EII	28.**EIO**
29.EOA	30.EOE	31.EOI	32.EOO
33.IIA	34.IIE	35.IIO	36.III
37.**IAI**	38.IAE	39.IAO	40.IAA
41.IEA	42.IEE	43.IEO	44.IEI
45.IOA	46.IOE	47.IOI	48.IOO
49.OOA	50.OOE	51.OOI	52.OOO
53.**OAO**	54.OAE	55.OAI	56.OAA
57.OEO	58.OEA	59.OEI	60.OEE
61.OIA	62.OIE	63.OII	64.OIO

Şekil 2.1 : Bütün sillojizm kurallarını içeren tablo

Bu modların tamamının geçerli olmadığını, sillojizm kurallarını ve şekillerini dikkate alarak bu modların geçerli sonuç verip vermediğine birkaç örnekle göz atacağız. Örneğin;

1-) AAO : Sonuç vermez. Çünkü iki olumlu öncülden olumsuz sonuç çıkmaz.

2-) OOE : Sonuç vermez. Çünkü iki olumsuzdan bir sonuç çıkmaz.

3-) AEO : Sonuç vermez. Çünkü olumlu öncüllerden olumsuz sonuç çıkmaz.

4-) EEI : Sonuç vermez. Çünkü iki tikel önermeden bir sonuç çıkmaz.

Sillojizmde orta terimin bulunduğu yere göre sillojizm şekillere ayrılır. Dört tane sillojizm şekli vardır. Bunlar birinci şekil, ikinci şekil, üçüncü şekil ve dördüncü şekil olarak adlandırılırlar.

Daha önceden 64 farklı modun bulunduğunu ve bunların yalnızca 10 tanesinin sonuç verdiğini söylemiştik. Bu 64 tane mod 4 tane şekil ile birleştiği zaman 64x4=256 çeşit mod elde

etmiş oluruz. Fakat bu 256 sillojizm modlarınında hepsi sonuç vermez. Zira belli bir şekil içinde sillojizmin sonuç vermesi için, o şekillerin bağlı bulunduğu özel kurallara uymak zorundadır. Aşağıda her şekil için uyulması gereken kuralları vereceğiz.

1. Şekil: Orta terim büyük önermede özne, küçük önermede yüklem olursa birinci şekilden sillojizm olur. Birinci şekil sillojizm diğer sillojizm şekillerine göre daha net sonuç verdiği için Aristotales tarafından mükemmel olarak adlandırılmıştır. O yüzden birinci şekil sillojizm diğerlerine nispeten daha çok kullanılır.

Birinci şeklin iki kuralı vardır. Bunlar:

1- Küçük önerme olumlu olmalıdır.

2- Büyük önerme tümel olumlu olmalıdır.

Bu kurallar sonucunda ortaya çıkabilecek geçerli toplam dört kalıp vardır. Bunlar:

AAA (B**ARBARA)**

EAE (C**ELARE**NT)

AII (D**ARII**)

EIO (F**ERIO**)

Bu modlar aşağıda örnekler ile gösterilmiştir.

1.

Y = Canlı , Z = Ölümlü , T = Bitki ;

Her canlı ölümlüdür.	(A)	Her Y, Z 'dir.	(A)
Bitkiler canlıdır.	(A)	Her T, Y'dir.	(A)
Bitkiler ölümlüdür.	(A)	Her T, Z'dir.	(A)

2.

Y= Çiçek, Z = Suyu sevmek, T= Bitki ;

Bütün çiçekler suyu sever.	(A)	Her Y, Z'dir.	(A)
Bazı bitkiler çiçektir.	(I)	Bazı T, Y'dir.	(I)
Bazı bitkiler suyu sever.	(I)	Bazı T, Z'dir.	(I)

3.

Y= İnsan, Z = Kuş , T = Güvercin;

Hiçbir insan kuş değildir.	(E)	Hiçbir Y, Z değildir.	(E)
Her güvercin kuştur.	(A)	Her T, Y'dir.	(A)
Hiçbir güvercin insan değildir.	(E)	Hiçbir T, Z değildir.	(E)

4.

Y= Siyasetçi, Z = Adil, T = Muhtarlar

Hiçbir siyasetçi adil değildir.	(E)	Hiçbir Y,Z değildir.	(E)
Bazı muhtarlar siyasetçidir.	(I)	Bazı T, Y'dir	(I)
Bazı muhtarlar adil değildir.	(O)	Bazı T, Z değildir.	(O)

2.Şekil: Orta terim her iki öncülde de yüklem ise buna ikinci şekil sillojizm denir.

İkinci şeklin iki kuralı vardır:

1- İki öncülden birinin olumsuz olması gerekir.

2- Büyük önerme tümel olmalıdır.

Bu kurallar sonucunda ikinci şekilde geçerli olarak 4 mod vardır. Bunlar:

EAE	(C**E**S**A**R**E**)
AEE	(C**A**M**E**STR**E**S)
EIO	(F**E**ST**I**N**O**)
AOO	(B**A**R**O**C**O**)

Bu modların örnekleri aşağıda verilmiştir.

1.

Y = Medeni İnsan, Y = Şiddet, T = Terörist

Hiçbir medeni insan şiddeti sevmez.	(E)	Hiçbir Z, Y değildir.	(E)
Her terörist şiddet yanlısıdır.	(A)	Her T, Y' dir.	(A)
Hiçbir terörist medeni değildir.	(E)	Hiçbir T, Z değildir.	(E)

2.

Z= Kaplumbağa, Y= Yavaş, Z = Tavşan

Her kaplumbağa yavaş ilerler.	(A)	Her Z, Y'dir.	(A)
Hiçbir tavşan yavaş ilerlemez.	(E)	Hiçbir T, Y değildir.	(E)
Hiçbir tavşan kaplumbağa değildir.	(E)	Hiçbir T, Z değildir.	(E)

3.

Z= Baba, Y = Çocuk, T= İnsanlar

Hiçbir baba oğlunu dövmez.	(E)	Hiçbir Z, Y değildir.	(E)
Bazı insanlar oğlunu döver.	(I)	Bazı T, Y' dir.	(I)
Bazı insanlar baba değildir.	(O)	Bazı T, Z değildir.	(O)

4.

Z= Matematik öğrencisi, Y=analiz, T = Tarihçiler

Her matematik öğrencisi analiz görür.	(A)	Her Z, Y'dir.	(A)
Bazı tarihçiler analiz görmez.	(O)	Bazı T, Y değildir.	(O)
Bazı tarihçiler matematik öğrencisi değildir.	(O)	Bazı T, Z değildir.	(O)

3.Şekil: Orta terim her iki öncülde de özne olursa sillojizmin üçüncü şekli olur. Üçüncü şeklinde iki kuralı vardır.

1-Küçük önerme olumlu olmalıdır.

2-Sonuç daima tikeldir.

Bu kurallar sonucunda ise üçüncü şekilde toplam 6 tane geçerli kural vardır. Bunlar

IAI (D**I**S**A**M**I**S)

AII (D**A**T**I**S**I)**

OAO (B**O**C**A**RD**O)**

EIO (F**E**R**I**S**O**N)

AII (D**A**R**A**PT**I**)

EAO (F**E**L**A**PT**O**N)

Bu modların örnekleri aşağıda verilmiştir.

1.

Y= Ders, Z= Zor, T=Faydalı

Bazı dersler zordur.	(I)	Bazı Y, Z'dir.	(I)
Her ders faydalıdır.	(A)	Her Y, T'dir.	(A)
Bazı faydalılar zordur.	(I)	Bazı T, Z'dir.	(I)

2.

Y= Hayvan, Z= Canlı, T = Yürümek

Her hayvan canlıdır.	(A)	Her Y, Z' dir.	(A)
Bazı hayvanlar yürüyemez	(I)	Bazı Y,T' dir.	(I)
Bazı yürüyemeyenler canlıdır.	(I)	Bazı T, Z' dir.	(I)

3.

Y= Eşşek, Z=Güçsüz, T = Binek

Bazı eşşekler güçsüzdür.	(O)	Bazı Y, Z değildir.	(O)
Her eşşek binektir.	(A)	Her Y,T' dir.	(A)
Bazı binekler güçsüzdür.	(O)	Bazı T, Z değildir.	(O)

4.

Y= Telefon, Z= Canlı, T = Güzel

Hiçbir telefon canlı değildir.	(E)	Hiçbir Y, Z değildir.	(E)
Bazı telefonlar güzeldir.	(I)	Bazı Y,T dir.	(I)
Bazı güzeller canlı değildir.	(O)	Bazı T, Z değildir.	(O)

5.

Y= İnsan, Z= Duygusal, T= Sevimli

Her insan duygusaldır	(A)	Her Y, Z dir.	(A)
Her insan sevimlidir.	(A)	Her Y, T'dir.	(A)
Bazı sevimliler duygusaldır.	(I)	Bazı T, Z' dir.	(I)

6.

Y= Kalem, Z= Araba, T= Yazmak

Hiçbir kalem araba değildir.	(E)	Hiçbir Y, Z değildir.	(E)
Her kalem yazı yazar.	(A)	Her Y, T'dir.	(A)
Bazı yazı yazanlar araba değildir.	(O)	Bazı T, Z değildir.	(O)

4.Şekil: Orta terim büyük önermede yüklem, küçük önermede özne olursa dördüncü şekil sillojizm olur.

Dördüncü şekilde 3 tane kural vardır.

1- Büyük önerme olumlu olursa, küçük önermede tümel olur.

2- Küçük önerme olumlu olursa sonuç daima tikel olur.

3- Olumsuz modlarda büyük önerme tümel olmalıdır.

Bu kurallar sonucunda dördüncü şekil sillojizmde toplam 5 tane geçerli kural vardır. Bunlar;

AEE (C**A**M**E**N**E**S)
IAI (D**I**M**A**R**I**S)
EIO (FR**E**S**I**S**O**N)
AAI (BR**A**M**A**NT**I**P)
EAO (F**E**S**A**P**O**)

Bu modların örnekleri aşağıda verilmiştir.

1.

Z= Bayat Balık, Y= Zararlı, T = Sağlıklı

Her bayat balık zararlıdır.	(A)	Her Z, Y' dir.	(A)
Hiçbir zararlı sağlıklı değildir.	(E)	Hiçbir T, Y değildir.	(E)
Hiçbir sağlıklı bayat değildir.	(E)	Hiçbir T, Z değildir.	(E)

2.

Z= Öğretmen, Y= İyi maaş, T= Rahat yaşamak

Bazı öğretmenler iyi maaş alır.	(I)	Bazı Z, Y dir.	(I)
Her iyi maaş alan rahat yaşar.	(A)	Her Y, T dir.	(A)
Bazı rahat yaşayanlar öğretmendir.	(I)	Bazı T, Z dir.	(I)

3.

Z= Öğretici film, Y = Zararlı, T = Yayınlanmaz

Hiçbir öğretici film zararlı değildir.	(E)	Hiçbir Z, Y değildir.	(E)
Bazı zararlılar yayınlanmaz.	(I)	Bazı Y, T dir.	(I)
Bazı yayınlanmayanlar öğretici film değildir.	(O)	Bazı T, Z değildir.	(O)

4.

Z = Bebek, Y = Sevimli, T= Sevilmek

Her bebek sevimlidir.	(A)	Her Z, Y' dir.	(A)
Her sevimli sevilir.	(A)	Her Y, T ' dir.	(A)
Bazı sevilenler bebektirler.	(I)	Bazı T, Z'dir.	(I)

5.

Z = Ders, Y = Zararlı, T = Men edilen

Hiçbir ders zararlı değildir.	(E)	Hiçbir Z, Y değildir.	(E)
Her zararlı men edilir.	(A)	Her Y, T' dir.	(A)
Bazı men edilen şeyler ders değildir.	(O)	Bazı T, Z değildir.	(O)

Kısaca, toplam 10 tane geçerli mod şekillere tatbik edilince 40 tane şekil oluşur. Buradan da sadece 19 tanesi şekillerin kurallarına uyarlar. O halde 40 tane sillojizm çeşidinden 1. şekilde 4, ikinci şekilde 4, üçüncü şekilde 6 ve dördüncü şekilde 5 tane olmak üzere toplam 19 sonuç elde ederiz.

Yukarıda yer alan 19 sillojizm moddan 14 tanesi Aristo tarafından sunulmuş, diğerleri Theophrastus (M.Ö 371-M.Ö 287) tarafından eklenmiş ve Petrus Hispanus (1215-1277) tarafından açıklanmıştır [5].

2.2 Modern Mantık ve Sillojizme Modern Yaklaşımlar

Bu kısımda sillojizme matematiksel olarak en yakın çalışan Lukasiewicz ve Moss'un sillojizme bakış açısını verdikten sonra diğer kısımda detaylar verilecektir.

2.3 Lukasiewicz'e Göre Sillojizm

Tanım 2.3.1. Sillojizmler, öncüllerin ve bir sonucun gerektirmesidir. Daha basit bir şekilde söyleyecek olursak α ve β iki öncül, δ sonuç olmak üzere, bütün Aristo sillojizmleri Eğer α ve β ise o halde δ şeklindedir [5].

Lukasiewicz'e göre Aristo tekdüze çalışmakta olup önermesel lojik bütünüyle göz ardı edilmiştir. Önermesel lojiğin sillojistik lojikten farklı olduğunu görebilmek için Aab ve $p \rightarrow q$ yi incelemek gerekmektedir. Bu önemli fark iddiaların farklı olmasından kaynaklanmaktadır. Burada a ve b değişkenleri birer terim iken p ve q birer önermedir [6].

2.4 Moss'a göre sillojizm

Yaptığı makaleyle [7] doğal lojik programının ve uygulamalı lojik alanlarının öncülerinden biri sayılabilir. Moss ve Pratt-Hartman tarafından yayınlanan [8] İngilizce doğal dilinin parçalarının kategorik gramer, lambda kalkulus ve birinci dereceden lojik kullanarak zaman karmaşıklıkları üzerine yaptığı çalışmaların tamlık ispatları ve algoritmaları üstünde durmuştur.

2.5 S, S^+, $CARD$, ve MORE LOJİKLERİ

Bu bölümde S, S^+ ve $CARD$ lojiklerinin sintaks ve semantikleri incelenecektir.

2.5.1 $\mathcal{L}(All, Some, No)$ Dili ve S Lojiği

$\mathbb{P}$ 1-li atomlar içeren isimlerin bir koleksiyonu ve $p, q \in \mathbb{P}$ olsun [7]. S lojiğinin cümlelerinin kümesi olan $\mathcal{L}(All, Some, No)$ dili $All\ p\ are\ q$, $Some\ p\ are\ q$ ve $No\ p\ are\ q$ sintaksları 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 aşağıdaki özelliklere sahiptir.

$\mathcal{M} \vDash All\ p\ are\ q$ ancak ve ancak $[[p]] \subseteq [[q]]$,
$\mathcal{M} \vDash Some\ p\ are\ q$ ancak ve ancak $[[p]] \cap [[q]] \neq \emptyset$,
$\mathcal{M} \vDash No\ p\ are\ q$ ancak ve ancak $[[p]] \cap [[q]] = \emptyset$,

Tanım 2.5.1.1 Γ cümlelerin bir kümesi olmak üzere $\mathcal{M} \vDash \Gamma$ ancak ve ancak her $C \in \Gamma$ için $\mathcal{M} \vDash C$ dir.

Tanım 2.5.1.2 $\Gamma \vDash C$ ancak ve ancak Γ içindeki her cümleyi doğru yapan her model C yi de doğru yapar.

Tanım 2.5.1.3 Bir Γ üzerindeki **ispat ağacı** düğümleri dildeki cümleler ile birleşik ve bir düğümü ya Γ nın bir elemanı ya da kuralların uygulanması ile elde edilmiş sonlu ağaçtır.

Tanım 2.5.1.4 Eğer her C için $\Gamma \vdash C$ ise Γ **tutarsızdır** denir.

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

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

$$\frac{No\ p\ are\ q}{No\ q\ are\ p} \qquad \frac{All\ p\ are\ q\ \ No\ r\ are\ q}{No\ r\ are\ p}$$

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

$$\frac{Some\ p\ are\ q \quad No\ p\ are\ q}{\mathrm{C}}\chi$$

Şekil 2.2 $\mathcal{L}(All, Some, No)$ için kurallar tablosu

Şekil 2.2 de χ kuralı tutarsızlık olarak adlandırılan kuraldır. Kural ispat ağacına göre *Some x are y* ve *No x are y* türetebiliyor olduğuna göre bu bize modelimizin tutarsız olduğunu ve böylece modelden bu $\mathcal{L}(All, Some, No)$ dilindeki bütün cümlelerin türetilebileceğini gösterir.

Fakat bu lojik tam olduğu için bu $\mathcal{L}(All, Some, No)$ dili içerisindeki bütün cümlelerin modelde doğru olduğunu gösterir.

Teorem 2.1.5.5 *S* lojiği Şekil 2.2 deki kurallar altında tam ve sağlamdır [7].

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

$\mathbb{P}$ 1-li atomlar içeren isimlerin bir koleksiyonu ve $p, q \in \mathbb{P}$ olsun. $S^{\dagger}$ lojiğinin cümlelerinin kümesi olan $\mathcal{L}(All, Some, ')$ dili, *All p are q, All p are non-q ve All non-p are 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 [17]. Semantikler model üzerinde inşa edilecektir. $\mathcal{L}(All, Some, ')$ dili için bir $\mathcal{M}$ modeli, M evren ve her p ∈ $\mathbb{P}$ için $[[p]] \subseteq M$ve $[[p']] \neq M \backslash [[p]]$ olacak şekilde aşağıdaki gibi inşa edilir.

Bir $\mathcal{M} = \left(\mathrm{M}, [\![\ \ ]\!]\right)$ modeli aşağıdaki özelliklere sahiptir.

$\mathcal{M} \vDash$ *All p are q* ancak ve ancak $[\![p]\!] \subseteq [\![q]\!]$,

$\mathcal{M} \vDash$ *Some p are q* ancak ve ancak $[\![p]\!] \cap [\![q]\!] \neq \emptyset$,

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

$$\frac{All\ p\ are\ q'}{All\ q\ are\ p} \qquad \frac{All\ x\ are\ y\ All\ y\ are\ z}{All\ x\ are\ z}$$

$$\frac{All\ p\ are\ p'}{All\ p\ are\ q} \qquad \frac{All\ p\ are\ q\ All\ q\ are\ r}{All\ p\ are\ r}$$

$$\frac{All\ p'are\ p}{All\ q\ are\ p} \qquad \frac{All\ q\ are\ n\ \ Some\ p\ are\ q}{Some\ p\ are\ n}$$

$$\frac{Some\ p\ are\ q\ \ All\ p\ are\ q'}{\mathrm{C}}\chi$$

Şekil 2.3 $\mathcal{L}(All, Some, ')$ için kurallar tablosu

Şekil 2.3 de *No p are q* cümlelerini kullanmamamızın sebebi *All p are non-q* ile ifade edilebildiği için dil içerisinde *No p are q* cümlerine ihtiyaç yoktur.

Teorem 2.5.1.6 $S^{\dagger}$ lojiği Şekil 2.3 deki kurallar altında tam ve sağlamdır [9].

3. MATERYAL VE YÖNTEM

3.1. Temel Tanımlar

Bu bölümde, tezin anlaşılabilirliğini kolaylaştırmak için bazı temel tanımlar verilmiştir.

Tanım 3.1.1. İki öncülden, tümdengelimsel akıl yürütme yoluyla bir hüküm türetilmesine ***sillojizm*** denir [10].

Tanım 3.1.2. $\beta \subset A\ x\ A\ ve\ x, y, z \in A$ olsun. Bir A kümesi üzerinde tanımlanan bağıntı, Akümesinin tüm elemanları için yazılabilecek (x, x) ikililerini içeriyorsa **yansıyandır** denir. Formel olarak $[\forall x(x \in A \Longrightarrow (x, x) \in \beta)]$ gösterilir [11].

Tanım 3.1.3. $\beta \subset A\ x\ A\ ve\ x, y, z \in A$ olsun. Bir β bağıntısı(x, y) ikilisini içerirken aynı anda (y, x) ikilisini de içeriyorsa **simetriktir** denir. Formel olarak $[(x, y) \in \beta \Longrightarrow (y, x) \in \beta]$ gösterilir [11].

Tanım 3.1.4. $\beta \subset A\ x\ A\ ve\ x, y, z \in A$olsun. Bir β bağıntısı(x, y) ikilisini içerirken aynı anda (y, x) ikilisini de içermiyorsa **ters simetriktir** denir. Formel olarak $[(x, y) \in \beta \Longrightarrow (y, x) \notin \beta]$ gösterilir [11].

Tanım 3.1.5. $\beta \subset A\ x\ A\ ve\ x, y, z \in A$ olsun. Bir βbağıntısı(x, y) ve (y, z)ikilisini içerirken aynı anda (x, z) ikilisinide içeriyorsa **geçişkendir** denir. Formel olarak$[(x, y) \in \beta \wedge (y, z) \in \beta \Longrightarrow (x, z) \in \beta]$ gösterilir [11].

Tanım 3.1.6. Eğer bir kavram, ifade ettiği grubun tamamını kapsıyorsa (bütün insanlar, tüm gezegenler, tüm çiçekler vb.) **tümel** kavram denir [12].

Tanım 3.1.7. Eğer bir kavram ifade ettiği grubun tümü kadar büyük, bir tek üyesi kadar küçük olmayan bir kısmını kapsıyorsa (bazı insanlar, bir kısım gezegenler birkaç çiçek vb.) **tikel** kavram denir [12].

Tanım 3.1.8. Bir G **çizge** ya da bir **basit çizge**, bir V tepeler kümesi ve V nin 2-li alt kümelerinden oluşan E ayrıtlar kümesinden oluşan bir $G = (V, E)$ sıralı ikilisidir [13].

Tanım 3.1.9. Bir G **yönlü çizge**, V nin sıralı 2-li alt kümelerinden oluşan E ayrıtlar kümesini içeren bir $G = (V, E)$ sıralı ikilisidir [14].

Tanım 3.1.10. Bir ***etiketli yönlü çizge*** aşağıdaki özellikleri sağlayan bir $G = \langle V, r_V, r_E \rangle$ üçlüsüdür [15].

- Tepelerin sonlu bir V kümesi,

- v_i tepesi 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 (u_i, v_j, l), (v_i, v_j) ikilileri ve l etiketine sahip üçlülerin bir kümesidir.

4. BULGULAR

Bu bölüm, $\mathcal{L}(More, IA)$ dilinin lojiğini inceleyeceğiz. Lojik "more" (daha fazla) olarak adlandırılan kardinalite karşılaştırması yapan ve birinci mertebe lojiğin dilinde ifade edilemeyen niceleyiciyi içermektedir. Cümle formları temel olarak *x* ve *y* çoğul cins isimler olmak üzere "*x den daha fazla y vardır*." şeklindedir. Kesişen sıfatlar ile kombine edilen cins isimlerin cümle formları a ve *b* kesişen sıfatlar olmak üzere "a *x den daha fazla b y vardır*." şeklindedir. Bu tipte niceleyiciye sahip cümlelerin türetim algoritmaları ve türetim sağlanmadığında karşıt-model inşa algoritmaları üzerinde duracağız.

4.1 $\mathcal{L}(More, IA)$ Dilinin Lojiği

$$\frac{\exists^{>}(n,p)\exists^{>}(p,q)}{\exists^{>}(p,q)} \qquad \frac{\exists^{>}(p,p)}{\varphi}$$

Şekil 4.1 $\mathcal{L}(More)$ lojiği için kurallar tablosu

P kümesi x, y, z, ... çoğul cins isimlerini içeren değişkenler olsun. $More$ un cümleleri "*There are more x than y.*" formundadır ve Türkçe olarak "*y den daha fazla x* vardır." şeklinde okunur. $\exists^{>}(x, y)$ sintaksı *"There are more x than y*"ın kısa yazılışı olarak kullanılacaktır. Semantikler cümlelerin sonlu kümesi Γ ve sonlu evren *M* üzerinde inşa edilecektir. *P* den *M* nin alt kümelerine olan $[[\ \ ]]$ yorum fonksiyonu, her $x \in P$ için, $[[x]] \subseteq M$ şeklinde tanımlanır. $[[x]]$ in kardinalitesi, $|[[x]]|$ ile gösterilir. Bu dil $\mathcal{L}(More)$ olarak adlandırılacaktır. Şekil 4.1 $\mathcal{L}(More)$ dilinin lojiğini göstermektedir. Bu dile kesişen sıfatlar ekleyerek daha geniş bir dil olan $\mathcal{L}(More, IA)$ yı tanıtacağız. Burada kesişen sıfat dememizin sebebi her sıfat her ismin önüne gelmekte mantıksız olabilir. Örnek vericek olursak uzun bir öğrenci denilebilir ama uzun bir basketbolcu demek tam doğru olmaz. Bu sebepten dolayı uzun kelimesi her yerde tam olarak doğrudur diyemeyiz.Burada ingilizce dilindeki bazı sıfatlar $More$ diline eklenecektir. Kesişen sıfatlar (IA) olarak bilinen sıfatlara blue (mavi), green (yeşil), black (siyah), ... ve çoğul cins isimler ise cats (kediler), cars (arabalar), machines (makineler) gibi örnekler verilebilir. Bu dilde ise sintakslar basit isim *x, y, z*, ... ve kesişen sıfatlar ise a_1, a_2, a_3 ... ile verilecektir. Bir kompleks isim (ax) bir isimdir öyle ki *x* basit bir isim ve *a* bir kesişen sıfattır. *x, y, z* değişkenleri basit isimler ve *p, q, r* ise isimler için kullanılacaktır. $\exists^{>}(p, q)$ cümlelerinin kombinasyonlarını içeren bu dile $\mathcal{L}(More, IA)$ diyeceğiz. Semantikler sonlu cümle kümeleri ve sonlu *M* evreni üzerinde

inşa edilecektir. Her basit isim x için $\left[\left[x\right]\right] \subseteq M$. $a\ x$ isminin semantiği $\left[\left[ax\right]\right] = \left[\left[a\right]\right] \cap \left[\left[x\right]\right]$ ile tanımlanır. Bir $\mathcal{M} = \left(M, \left[\left[\ \ \right]\right]\right)$ modeli için"$\mathcal{M} \vDash \exists^{>}(p,q)$ ancak ve ancak $\left|\left[\left[p\right]\right]\right| > \left|\left[\left[q\right]\right]\right|$, $\mathcal{M}$ nin içindedir." doğruluk özelliğine sahip olacaktır.

$$\frac{\exists^{>}(n,p)\exists^{>}(p,q)}{\exists^{>}(p,q)}(\text{tr}) \qquad \frac{\exists^{>}(ax,q)}{\exists^{>}(x,q)}(\text{ia1}) \qquad \frac{\exists^{>}(p,x)}{\exists^{>}(p,ax)}(\text{ia2})$$

$$\frac{\exists^{>}(p,p)}{\varphi}(\chi 1) \qquad \frac{\exists^{>}(ax,x)}{\varphi}(\chi 2)$$

Şekil 4.2 $\mathcal{L}(More, IA)$ lojiği için kurallar tablosu

Şekil 4.2 de (χ1) ve (χ2) kuralları tutarsızlık olarak adlandırılırlar. Kural ispat ağacına göre *more p than p* veya *more ax than x* türetebiliyor ise bu bize modelimizin tutarsız olduğunu ve böylece modelden $\mathcal{L}(More, IA)$ dilindeki bütün cümlelerin türetilebileceğini söyler. Şekil 4.2 de (ia1) ve (ia2) kuralı (intersecting adjectives) kesişen sıfatları içeren kurallardır. (ia1) kuralı ispat ağacına göre *more ax than q* türetilebiliyor ise buradan *more x than q* türetilebildiğini söylemektedir. (ia2) kuralı ise ispat ağacına göre *more p than x* türetilebiliyor ise *more p than ax* türetilebildiğini söylemektedir.Son olarak (tr) kuralı transitive (geçişli) olarak adlandırılan kuraldır. (tr) kuralı *moren than p* ve *more p than q* türetilebiliyor iken bize *moren than q* türetilebildiğini söylemektedir.

4.2. Türetimler ve Karşıt-Modeller

Bu kısımda $\mathcal{L}(More, IA)$ lojiğindeki cümlelerin türetimleri ve şayet türetim meydana gelmiyor ise nasıl karşıt model oluşturulacağı üzerinde duracağız.

4.2.1. Türetimler

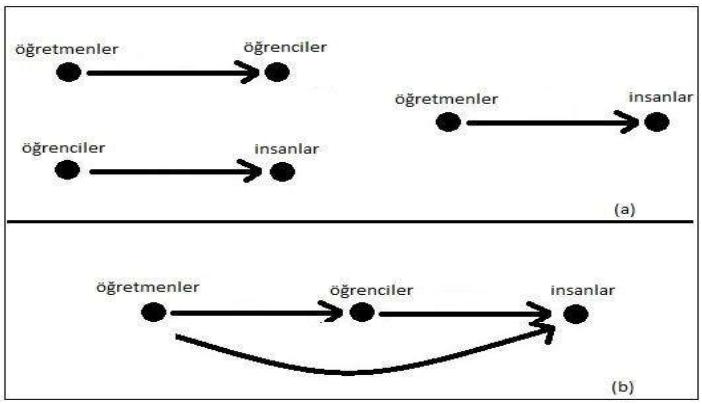
Öğretmenlerden daha fazla insan vardır.
Öğrencilerden daha fazla öğretmen vardır.
——————————————— (1)
Bu nedenle, öğrencilerden daha fazla insan vardır.

(1) numaralı türetimde (tr) kuralı kullanılmıştır. (tr) kuralı geçişken bağıntı özelliğine sahiptir. Bu özellik türetimlere bu şekilde yansıtılır.

Kırmızı arabalardan daha fazla yeşil bisiklet vardır.
Yeşil bisikletlerden daha fazla mavi traktör vardır.
———————————————(2)
Bu nedenle, kırmızı arabalardan daha fazla traktör vardır.

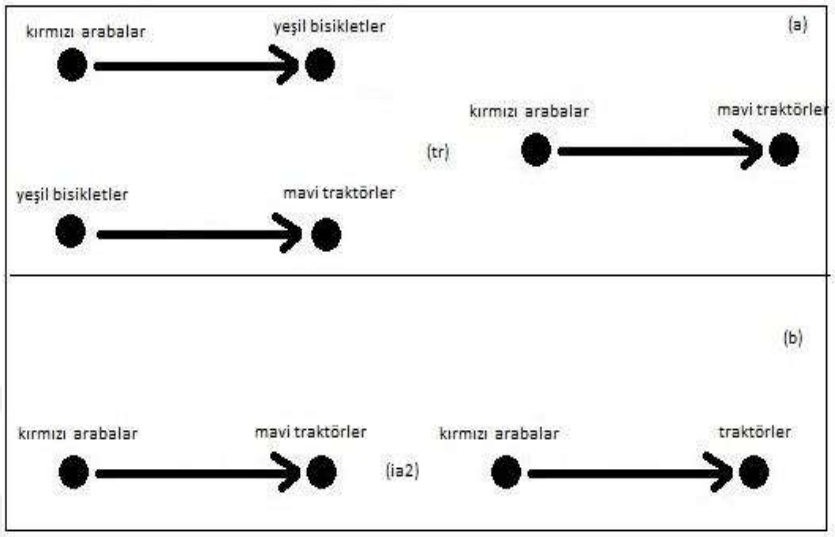
(2) numaralı türetimde önce (tr) kuralı ile kırmızı arabalardan daha fazla mavi traktör olduğu ve (ia2) kuralı ile traktörlerin kırmızı arabalardan daha fazla olduğu türetilir. Türetimlerin daha kolay anlaşılması ve bilgisayara veri tipi olarak aktarılması amacı ile $\mathcal{L}(More, IA)$ dilinin lojiği etiketli çizge teorik yapı ile temsil edilebilir [16]. Sadece "More" cümleleri üzerinde çalışıldığı için etiketleme işlemenine ihtiyaç yoktur.

Şekil 4.3 te türetim (1) için çizge teorik temsil verilmiştir. (a) ile gösterilen kısım iki öncül ve bir sonuç ile temsil edilmiştir. (b) ile gösteren kısım ise geçişken özelliğin direk uygulamasıdır.



Şekil 4.3 Türetim 1 için yönlü çizge teorik temsil

Şekil 4.4 te türetim *(2)* için çizge teorik temsil verilmiştir. (a) ile (tr) kuralının ve daha sonra (b) ile (ia2) kuralının uygulanışı verilmiştir.



Şekil 4.4 Türetim 2 için yönlü çizge teorik temsil

Kuralların uygulanış sırası bazen önemli olmasa da burada önemli olduğuna dikkat çekiyoruz.

4.2.2. Türetim Algoritmaları

Bu kısımdan lojiğin dili içinde verilen bir cümleler kümesinden bir bir cümlenin türetilip türetilemeyeceğinin sorgusunun nasıl yapılacağını anlatacağız. Esas amacımız verilen bir cümleler kümesinden türetilebilen tüm cümleleri elde etmektir.Buradaki ilk konu algoritmanın verilen cümleler kümesinin tutarlı olup olmadığını saptamaktır. Şekil 4.2 de görüleceği gibi ($\chi 1$) ve ($\chi 2$) olmak üzere iki tip tutarsızlık söz konusudur. ($\chi 1$) kuralı hiçbir ismin temsil ettiği kümenin kardinlitesinin kendisinden büyük olamayacağını söyler. Öncelikle lojiğin cümlelerinin her birine bir yönlü çizge karşılık getirdiğimizi hatırlatalım ve döngü tanımını verelim.

Tanım 4.3.1.1 G= (V,E) bir yönlü çizge ve $v \in G$ olsun. Şayet $(v,v) \in E$ veya v den v ye ulaşan en az bir yol var ise G döngü içerir denir [17].

Tanım 4.3.1.1 e göre verilen bir cümleler kümesi Γ şayet "*more v than v*"içeriyor veya "*more v than v*" bu Γ dan türetilebiliyor ise Γ tutarsız olacaktır. Γ daki cümlelerin çizge temsili ile $v \rightarrow v$

yolu varsa bir döngü bulunmuş ve tutarsızlığa ulaşılmış olur. Elbette bu döngü sadece *x, y* ve *z* tipinde isimleri içeren cümeler için değil *a x*, *b y* ve *c z* tipinde kesişen sıfatlar içeren cümleler için de geçerlidir. Bu tipte cümleleri elde etmek içeri ileri arama yapmak gerekir. Bunun için Γ kümesine (tr), (ia1) ve (ia2) kurallarını uygulayarak türetimleri elde etmek gerekir.

Örnek 1.

Verilen bir $\Gamma = \{more\ v\ than\ w, more\ w\ than\ h,\ more\ h\ than\ v\}$ cümleler kümesi için $v \to w \to h \to v$ yani $v \to v$ elde edildiği için Γ nın tutarsız olduğu görülür.

($\chi 2$) kuralı ile bulunacak tutarsızlık verilen bir Γ kümesi "*more ax than x*" gibi bir cümle içermelidir veya bu cümle Γ dan türetilmelidir. Γ dan türetilen cümleleri (tr), (ia1) ve (ia2) kurallarını uygulayarak elde ettikten sonra $ax \to x$ tipinde bir yol olup omadığını kontrol etmek tutarsızlığa ulaşmak için yeterli olacaktır.

Örnek 2.

Verilen bir $\Gamma = \{more\ ax\ than\ w, more\ w\ than\ h,\ more\ h\ than\ x\}$ cümleler kümesi için $ax \to w \to h \to x$ yani $ax \to x$ elde edildiği için Γ nın tutarsız olduğu görülür.

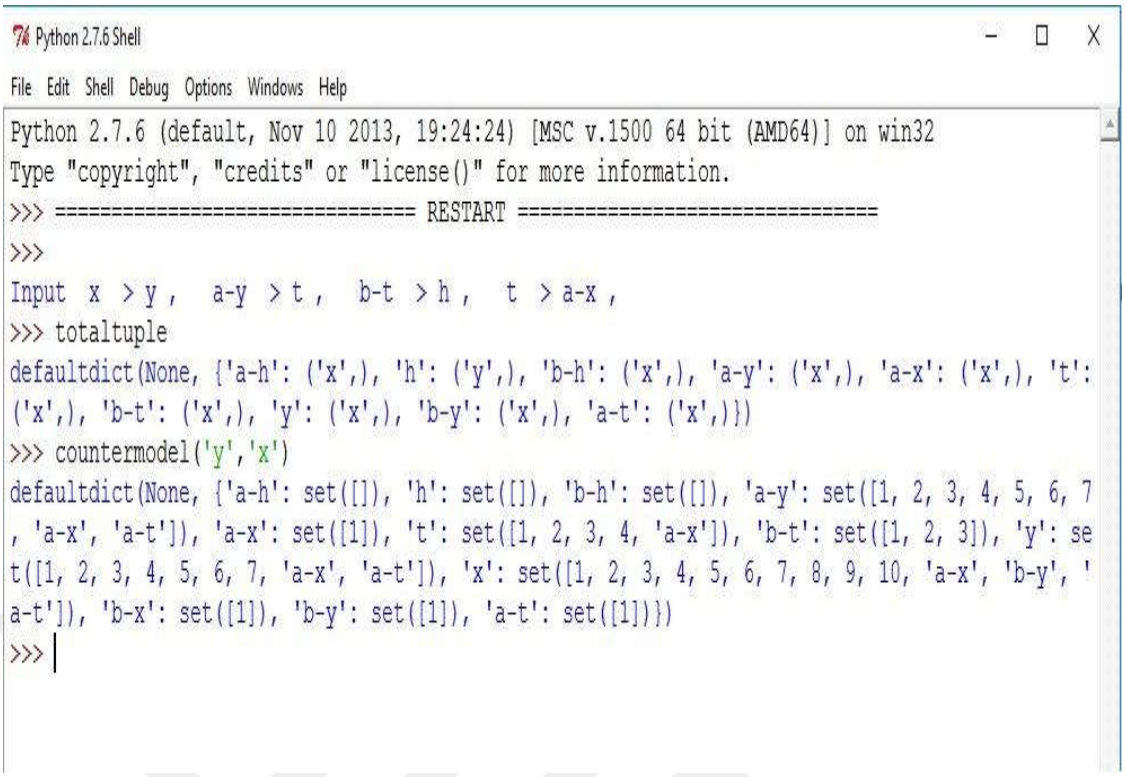
Temel olarak verilen bir Γ cümleler kümesinden elde edilen tüm türetimleri Şekil 4.2 deki kuralların lojiğin dilinin yönlü çizge yapısına dönüştürülerek uygulanışından elde ederiz. Bu uygulanış sırasında elde edilen her yol bir yol yani birbirine ulaşan her isim (her tepe) çizgeye eklenerek oluşturulacak türetim kümesi meydana getirilir. Dolayısıyla verilen bir cümleler kümesiden bir cümlenin türetilip türetilemeyeceğini saptamak için tüm türetimlerin kümesine ait olup olmadığını kontrol etmek yeterli olacaktır. Kümenin tuttarsız olması durumda herhangi bir ek işleme gerek kalmadan işlem sonlandırılabilirdir.

4.2.3. Karşıt Modeller

Bu kısımda karşıt modeller üzerine yoğunlaşacağız. Öncelikle karşıt modelin ne olduğunu açıklayalım. Bir Γ cümleler kümesi ve bir β cümlesinin Γ dan türetilemediği verilsin. Bu durumda Γ daki tüm cümleleri doğru fakat β cümlesini yanlış yapan bir model inşa edeceğiz. Bir basit bir örnek ile kavramı daha anlaşılır kılalım.

Örnek 3.

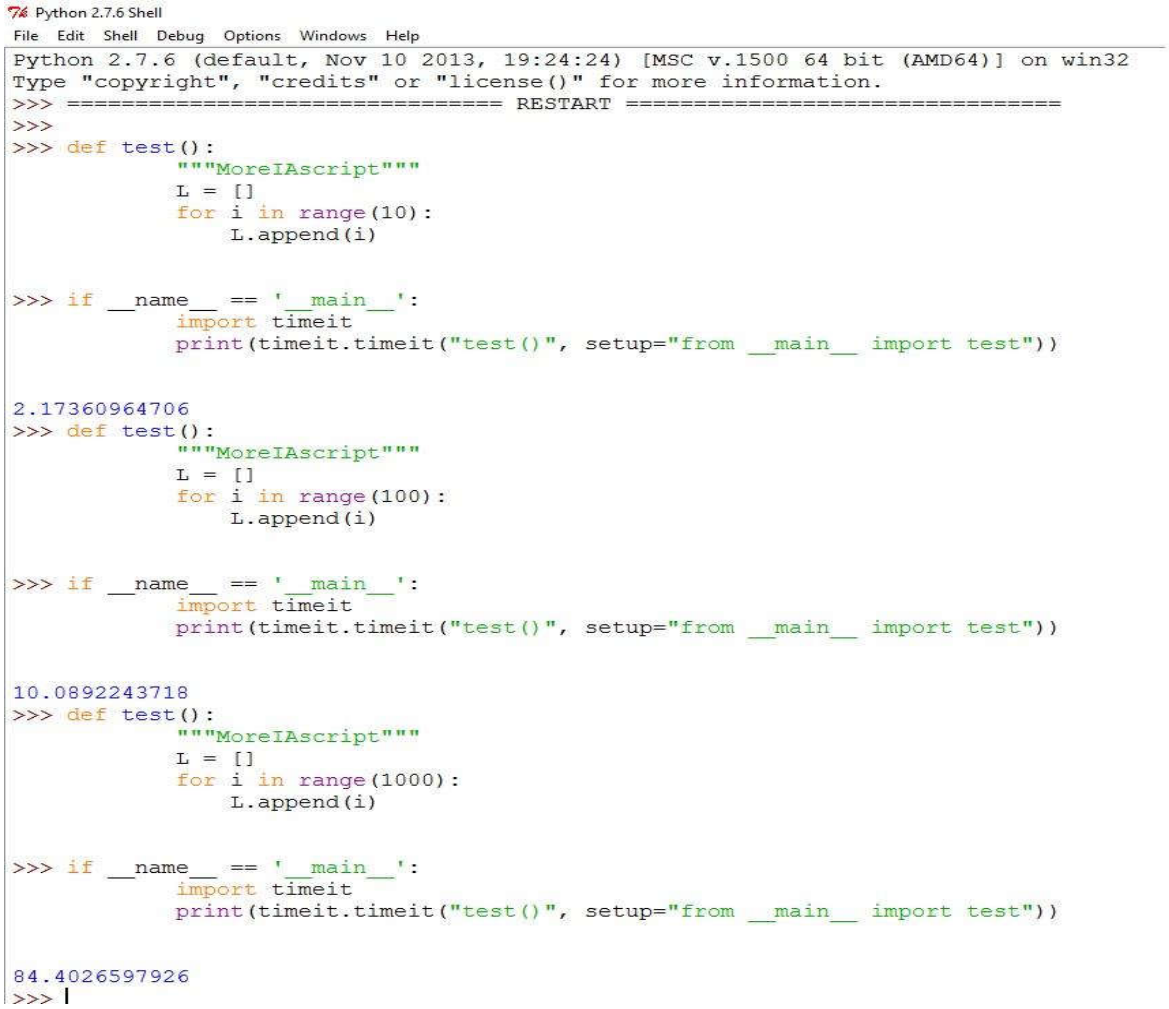
Verilen bir cümeleler kümesi $\Gamma = \{more\ v\ than\ w,\ more\ w\ than\ h\ ,\ more\ h\ than\ k\}$ için "*more k than v*" cümlesinin Γ dan türetilemeyeciği rahatlıkla görülebilir. Şimdi Γ nın cümlelerini doğru yapan fakat "*more k than v*" cümlesini yanlışlayan bir model kuralım. Modelin evreni $M = \{0,1,2,3\}$ ve $[[k]] = \{3\}$, $[[h]] = \{0,2\}$, $[[w]] = \{0,1,2\}$ ve $[[v]] = \{0,1,2,3\}$ olsun. Dikkat edilirse $[[k]]$ tek elemanlı, $[[h]]$ iki elemanlı, $[[w]]$ üç elemanlı ve $[[v]]$ dört elemanlı kümedir. Bu isimlerin kardinaliteleri bakımından bakıldığında Γ daki tüm cümlelerin bu model ile doğru olduğu görülür. Öte yandan "*more k than v*" cümlesi bu model ile yanlış olur çünkü $[[k]]$ tek elemanlı iken $[[v]]$ dört elemanlı bir kümedir. Böylece kümeden türetilemeyen bir cümle için bir karşıt model inşa ettik. Örnek 3 de verilen cümleler kümesi basittir ve model inşası kolaydır. Daha fazla sayıda cümle içerseydi ve bu cümlelerin isimlerinde kesişen sıfatlar olsaydı karşıt model inşası çok daha karmaşık ve kafa karıştırıcı olacaktı. Özellikle kesişen sıfatları içeren cümlelerde kümelerin kesişimleri söz konusu olduğu için model inşasına daha da dikkat etmek gerekmektedir. Çünkü bir $[[ax]]$ kümesi bir elemanı içeriyor ise doğal olarak $[[x]]$ kümesi de bu elemanı içermek zorundadır. Aksi halde Γ ya ait olan bir cümle yanlış olabilir. Anlaşılırlık açısından karşıt model algoritmasını Python [18] platformu üzerinde hazırlanan programın bir örnek çıktısı ile anlatacağız. Programa [19] adresinden ulaşılabilir.



Şekil 4.5 Karşıt model için bir Python örneği

Şekil 4.5 te input (girdi) ile verilen cümleler kümesi kısaltma adına '$x > y,\ a - y > t,\ b - t > h,\ t > a - x$' şeklinde ifade edilmiştir. $x > y$ ifadesi '*'more x than y*'' anlamına gelmektedir. $a - y$ ifadesi a nın bir kesişen sıfat ve y nin bir çoğul cins isim olduğunu göstermek için aralarında **–** işareti içermektedir. Bu girdi kümesinden elde edilecek tüm türetimler *totaltuple* ile gösterilmektedir. '*a-h*:(*x*)' ifadesi "*x more than a-h*" anlamına gelmektedir. Karşıt modeli meydana getiren *countermodel('y','x')* fonksiyonu "*y more than x*" ifadesinin önce türetilip türetilmediğini kontrol eder. Eğer böyle bir cümle türetiliyor ise türetilebilir 'derivable' olduğunu yazdırır. Örnekte olduğu gibi bu cümle türetilemediği için bize bir karşıt model hesaplamaktadır. Türetim kümesinde herhangi bir isimden büyük olmayan '*h*' a boş küme (set([])) ataması yapılmıştır. '*h*' boş küme olduğu için doğal olarak '*b-h'* boş olduğu görülmektedir. Öte yandan '*x*' kümesi türetim sonucunda 11 tane isimden daha büyük kardinaliteye sahip olduğu için 0,1,2,3,4,5,6,7,8,9,10 elemanlarını barındırır. '*a-x'*, '*b-y'*, '*a-t*' kümeleri boştan farklı kümeler ve 'x' den daha küçük kardinaliteye sahip oldukları için 'x' kümesi tarafından eleman olarak barındırılırlar. Aksi halde olası bir durumda *'a-x*' ile 'x' in aynı sayıda elemana sahip olup türetim kümesindeki *('a-x': (x))* ifadesi ile çelişirdi. Bu çelişkiyi aşmak için '*x*' kümesine kardinalitesi kendinden küçük olan 'a-x' ifadesi eleman olarak ekledik. Böylece elde edilen model "$y > x$" ifadesini yanlışlarken girdiye ait tüm ifadeleri (türetimler dahil) doğrulamaktadır. Burada kullanılan algoritma tipi topolojik sıralama tabanlıdır. Topolojik

sıralama ile kardinalite karşılaştırmaları yapılırken bir ismin kendinden önce ve sonra gelen (daha büyük veya daha küçük kardinaliteli) isimlerin de kendi aralarındakarşılaştırılabilirliği sağlanmaktadır.



Şekil 4.6 Girdi-sonuç zaman ölçümleri

2.40 GHz, 8.00 GB RAM, 64 bit işlemci sistemine sahip bir bilgisayarda girdi sayısına karşılık sonuç elde edilmesine dair ölçümler Şekil 4.6 da verilmiştir. Ölçümlere dayanarak algoritmaların etkin zamana sahip olduğu görülür. Yaklaşık olarak O(n)$\cong \frac{n}{10}$ büyük o değerine sahiptir.

5. SONUÇ VE ÖNERİLER

Bu tezde temel olarak bir doğal lojik kapsamında sillojistik akıl yürütmenin bilgisayar ortamına nasıl aktarılacağı gösterilmeye çalışılmıştır. Bu aktarım sürecinde karşılaşılan problemler, akıl yürütmelerin tarihsel boyutları, türetim ve karşıt model algoritmaları üzerinde durulmuştur.

Bu tezde ilk defa tanıtılan ve tanımlanan çoğul cins ve kesişen sıfatların içerildiği ifadelerin kardinalite karşılaştırmalarından oluşan cümlelerin kıyaslanması amacı ile $\mathcal{L}(More, IA)$ dilinin lojiğine ait türetim ve karşıt model algoritmaları verilmiştir. Etkin zaman karmaşıklığına sahip bu algoritmalarınbir uygulaması Python programı kullanılarak oluşturulmuştur.

Lojik isimlerin yorumlarının kümesel tümleyenleri (non-cats, kedi olmayanlar gibi) eklenerek genişletilebilir. Lojiğin modelleri sonlu kümeler üzerinde oluşturulmuştur. Sonsuz sayılabilir kümeleler üzerinde oluşturulabilecek modeller için π_0^1sınıfları üzerinde çalışabilir [20-22].

KAYNAKLAR

[1] Rose LE, 1968. Aristotle's Syllogistic. Springfield III Press. İllinois.

[2] Thom P, 1981. The syllogism. Munich : Philosophia, 197-198.

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

[4] Smith R, 1989. Aristotle: Prior Analytics, translated with introduction notes and commentary. Indianapolis. Hackett.

[5] Łukasiewicz J, 1957. Aristotle's Syllogistic From the Standpoint of Modern Formal Logic. Oxford University Press. Oxford.

[6] Fillion N, 2007. Two Accounts of Aristotle's Logic. The University of Western Ontario Press. London.

[7] Moss LS, 2008. Completeness Theorems for Syllogistic Fragments. F. Hamm and S. Kepser (eds.) Logics for Linguistic Structures, Mouton de Gruyter, 29: 143-173.

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

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

[10] MillerGA, 1995. WordNet: a lexical database for English, Communications of the ACM, 38(11): 39-41.

[11] Church A, 1996. Introduction to mathematical logic. Princeton University Press. New Jersey.

[12] Özlem D, 2004. Klasik/sembolik mantık, mantık felsefesi: bütün eserlerine doğru-5. İnkilap yayınevi. Yeditepe Üniversitesi. İstanbul.

[13] Iyanaga S, Kawada Y, 2012. Encyclopedic Dictionary of Mathematics. MIT Press. Cambridge.

[14] Bang-Jensen J, Gregory G, 2000. Digraphs. Theory: Algorithms and Applications. Springer Verlag Press. Berlin.

[15] Champin PA, 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.

[16] Topal S, 2017. Bazı Sillojistik ve Kardinalite Karşılaştırmalı Lojiklerin Türetimlerinin Cebirsel ve Etiketli Çizge Teorik Özellikleri Üzerine. Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi, DOI-10. 19113/sdufbed.50072

[17] West DB, 2001. Introduction to graph theory. Upper Saddle River Press. Prentice hall.

[18] Van Rossum G, 2007, Python Programming Language. In USENIX Annual Technical Conference. https://www.usenix.org/ (Erişim Tarihi : 06.20.2007).

[19] Topal S, 2017. $\mathcal{L}(More, IA)$ Lojiğinin Python Uygulaması. http://pbs.beu.edu.tr/s.topal/docs (Erişim Tarihi: 15.05.2017).

[20] Çevik A, 2012. Hesaplanabilirlik kuramı ve Turing derecelerine giris. Gaziosmanpaşa Bilimsel Araştırma Dergisi, Tokat, Turkey. 1: 1-20.

[21] Çevik A, 2013. Antibasis theorems for classes and the jump hierarchy. Archive for Mathematical Logic, 4: 137-142.

[22] Çevik A, 2016. choice classes. Mathematical Logic Quarterly, 62(6): 563-574.

ÖZGEÇMİŞ

1991 yılında İzmir'de doğdu. İlkokulu ve Ortaokulu Vali Rahmi Bey İlköğretim Ortaokulu'nda ve liseyi de Karataş Lisesi'nde tamamladı. 2010 yılında Bitlis Eren Üniversitesi Fen Edebiyat Fakültesi Matematik Bölümüne kayıt yaparak lisans öğrenimine başladı. 2014 yılında mezun oldu ve aynı yıl Bitlis Eren Üniversitesi Fen Bilimleri Enstitüsünün Matematik Anabilim dalında açmış olduğu tezli yüksek lisans programına başladı. 2016-2017 yılları arasında Bitlis Halk Eğitim Müdürlüğünde ücretli öğretmen olarak görev yaptı.

Yasin AKÜNSOY