

Türkçe Metinler için Olay Sıralaması

Şadi Evren Şeker
Yıldız Teknik Üniversitesi
Bilgisayar Mühendisliği Böl.
academic@sadievrenseker.com

Banu Diri
Yıldız Teknik Üniversitesi
Bilgisayar Mühendisliği Böl.
banu@ce.yildiz.edu.tr

Özetçe

Bu çalışmanın amacı, doğal dilde hazırlanan metinlerin içerisinde geçen olayları sıralı göstermek için kullanılan sistemlerin, Türkçe için genişletilmesidir. Günümüzde doğal dildeki metinler üzerinde, doğal dil işleme yapılarak çıkarılan anlambilimsel gösterimlerin, olaylar arasındaki zamansal bağlantıyı göstermesi için veya işlenen doğal dildeki metinde bulunan olayları sıralaması için, ilave işlemler gerekmektedir. Bu işlemlerin sistematik olarak dilbilimsel bir temele oturtulmasını sağlayan başta İngilizce olmak üzere çeşitli dillerde çalışmalar bulunmaktadır.

Her dilde ortak olan zamansal mantıklara ilave olarak, her dilin kendisine özgü bazı farklı zamansal özellikleri de bulunmaktadır.

Bu çalışma kapsamında öncelikle literatürde bulunan zamansal mantıklar incelenmiştir. Bu mantıklardan, doğal dil için kullanımı öne çıkan bazıları, daha sonra hesaplanabilir dillere dönüştürülebilmektedir. Bu hesaplanabilir diller ve dayandıkları mantıklar incelenerek, Türkçe TimeML üzerinden iyileştirme yapılmıştır.

Abstract

Event Ordering in Turkish Texts

Aim of this study is advancing current event ordering methodologies to cover Turkish temporal logic. Currently, some additional operations are required to demonstrate the relation between events or ordering events in a natural language text, after outputting the semantical representation. There are some systematic studies based on English temporal logic and covering most of the Latin family. There are some differences between temporal logics in the

languages in addition to common temporal properties.

In this study, the temporal logics in the literature are researched. Some of these temporal logics are suitable for machine computation and some are suitable for natural language processing. An optimization is suggested on these computable and natural language processing suitable temporal logics to cover Turkish temporal logic.

Keywords: Event Ordering, Chronology Extraction, Temporal Logic, Computable Temporal Languages, Olay Sıralama, Kronoloji Çıkarımı, Zamansal Mantıklar, İşlenebilir Zamansal Diller, TimeML

1. Giriş

Olay sıralama çalışmaları, doğal dil işleme, soru cevaplama, ontoloji ve kronolojik gösterim gibi çeşitli alanlarda kullanılmaktadır. Olay sıralama çalışmalarının amacı, bir metindeki veya bir örneklem düzlemindeki olayları çıkararak bu olaylar arasındaki ilişkileri belirlemektir. Bu tanım itibarıyla olay sıralama çalışmalarında iki temel unsurdan bahsedilmesi gerekir. Birincisi, üzerinde ilişkilerin kurulu olduğu olaylar ve bu olayların sonucu etkileyen özellikleri, ikincisi ise bu olayların birbiri ile olan ilişki tipleridir. Bu çalışma kapsamında anlatılan farklı zamansal mantıkların hepsinde olaylar arasındaki ilişkiler için yeni bir özellik önerisi getirilmiş ve eklenmiştir. Olay sıralama çalışmalarını üç seviyede incelemek mümkündür.

- Mantık Seviyesi
- İşlenebilir Dil Seviyesi
- Uygulama Seviyesi

Her olay sıralaması çalışmasında mantık seviyesinde bir dayanak gerekmektedir. Bu yüzden makalenin ikinci bölümünde zamansal değerlere ve üçüncü bölümünde de zamansal mantık

çalışmalarına yer verilmiştir. Ne yazık ki bütün zamansal mantık modelleri, bilgisayarlar tarafından işlenebilir bir seviyeye çıkamamaktadır. Zamansal mantıkların ancak bir kısmı, bilgisayarlar tarafından kullanılabilen işlenebilir bir dile temel oluşturabilmektedir. Makalenin dördüncü bölümünde işlenebilir zamansal dillerden bahsedilmekte ve bu dillerden en öne çıkanı TimeML detaylandırılarak anlatılmaktadır. TimeML dili, yapı olarak bir XML özelliği gösterse de üçüncü bölümde anlatılan çoğu zamansal mantığı kendi bünyesinde barındırmaktadır.

Gerek bu makalede anlatılan zamansal mantıklar ve gerekse TimeML dili için eksik olan nokta, Türkçe dili için kullanılan ve geçerliliği olan bazı zamansal özellikler içerilerinde barındırmamalarıdır. Bunun sebebi, zamansal mantıklar üzerinde çalışma yapanların büyük bir çoğunluğu Latin kökenli dilleri kullanmakta ve aynı zamanda modellemek için karşılaştıkları problemler Latin dil kökenli olmaktadır. Olay sıralama çalışmalarının resmi olarak kullanıldığı problem gruplarından soru cevaplama, arama motorları veya metin özetleme gibi çalışmalar için günümüzde yayınlanmış derlemlerin tamamı Latin kökenli dillerden oluşmaktadır. Örneğin, işlenebilir bir dil olan TimeML dili ve bu dilin testi için oluşturulan TimeBank İngilizce için hazırlanmış olup, bu konuda oldukça başarılı sonuçlar sunmaktadır. Ancak, benzer bir çalışma henüz Türkçe için yapılmamıştır.

Makalenin dördüncü bölümünde, TimeML dilinin detayları verilip, Türkçe için yetersiz kaldığı noktalar incelenip çözüm önerileri sunulmuştur. Son bölümde ise, TimeML dili üzerinde, Türkçe için yapılan bazı değişiklikler ve bu değişiklikler kullanılarak yapılan uygulamalara örnek verilmiştir.

2. Doğal Dil Açısından Zamansal Değerler

Türkçe metinlerde bulunan olayların sıralanması için öncelikle doğal dillerde bulunan zamansal öğelerin çıkarılması gerekmektedir. Temel olarak bir doğal dilde bulunan her fiil zaman içerir ve zaman içermeyen bir olay olamaz. Ayrıca olayların dışında zamansal ifadeler de yer alabilir. Örneğin, “yarın”, “saat on” gibi ifadeler olay belirtmeyen zamansal ifadelerdir. Aslında her zamansal ifade bir atıftan (referans) ibarettir ve bu atıflar içinde

bulduğumuz mekana ve önceki olay geçmişine bağlıdır.

2.1. Uzak Zaman İlişkisi

Doğal dillerdeki ve insanlığın kullandığı güncel bütün zaman bilgilerinin birer atıftan ibaret olduğunu söyleyebiliriz. Bir kabul olarak, sonsuz mesafelere gidebilen ve içindeki yolcuysa sonsuza kadar uyutarak saklayan bir uzak taşıtı hayal edin. Sonsuzluğa giden bu yolculukta aniden uyanan bir yolcuysa zamanın sorulması halinde cevap verilemez. Örneğin, “saat kaç?”, “günlerden ne?” veya “ne kadar zamandır uyuyorsun?” gibi sorulara cevap verebileceği hiçbir atıf bulunamaz. Bu atıfların tamamı, içinde yaşadığımız güneş sistemine veya tarihte yaşanmış olaylara aittir. Bu anlamda doğal dillerde bulunan ve insanlığın kullandığı diğer bütün zamansal ifadeler birer atıftan ibarettir diyebiliriz.

2.2. Dilbilimde Zamanlar

Dilbilimsel olarak her dilde üç temel zaman bulunur. Geçmiş, şimdiki ve gelecek zaman şeklinde Türkçede de bulunan bu zamanlar, bazı dillerde bulunmazlar. Örneğin, İngilizcede gelecek zaman yoktur, bunun yerine İngilizceye özgü olan bakış (aspect) veya modal kullanılması gerekir. Ayrıca, Türkçede bu zamanlara ek olarak geçmiş zaman bilgisi rivayet ve hikaye şeklinde ikiye ayrılmıştır.

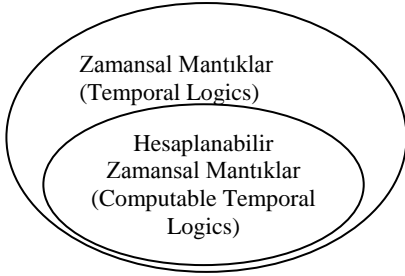
Genelde fiiller, bir olay ifade ederler ve zamansal değer taşırlar. “Geliyorum”, “Gideceğim” gibi fiillerin doğal dilde karşılığı olan zamanları bulunmaktadır. Bunun dışında zamansal değerler, gerçekleşen olayın zamanı hakkında bilgi de verebilirler. Örneğin, “yarın”, “saat onda” gibi ifadeler bu anlamda değerlendirilebilir.

Doğal dildeki bir metinde ise en az üç farklı zamandan bahsetmek gerekir. Metnin geçtiği zaman, olay zamanı ve olayın atıfta bulunduğu zaman şeklinde sıralanabilecek bu zamanlar, doğal dilin kullanıldığı her metinde analiz gerektirmektedir.

3. Zamansal Mantıklar

Zamansal mantıklar temel olarak, zaman akışı üzerinde çeşitli önerme kısıtları ve işlemler tanımlayan matematiksel sistemler olarak tanımlanabilirler. Buna göre bir zamansal mantığın en önemli işlem yaptığı değer, olaylar, durumlar ve zamansal gösterimlerdir. Zamansal mantıkların

temel amacı, bu değerlerin doğru olarak modellenmesi ve bu değerler arasındaki ilişkilerin doğru olarak ifade edilmesidir. Zamansal mantıkların tamamı hesaplanabilir (computable) değildir. Diğer bir deyişle zamansal mantıkları hesaplanabilir ve hesaplanabilir olmayan (a-computable) olarak iki grupta incelemek gerekir.



Şekil-1: Zamansal mantıklar ve hesaplanabilir ilişkisi
Şekil 1’de görüldüğü üzere hesaplanabilir zamansal mantıklar, zamansal mantıkların bir alt kümesi olarak düşünülebilir ve bu iki kümenin farkını hesaplanabilir olmayan zamansal mantıklar oluşturur. Buradaki fark ise, yapısal olarak zamansal mantığın üzerine kurulu olan klasik mantıktır. Şayet, zamansal mantığın üzerine inşa edilen mantık, matematiksel bir mantık yapısına sahip veya hesaplanabilir mantık ise, bu mantığın bilgisayarlar tarafından işlenebilir olduğundan da bahsedebiliriz. Hesaplanabilir olmayan zamansal mantıklar bu çalışmanın kapsamı dışında olup, bu mantıklardan bahsedilmeyecektir. En sade zamansal mantık olan LTL (Linear Temporal Logic, Doğrusal Zaman Mantığı) ile başlanacak ve bu mantığa yapılan ilaveler ile diğer mantıklar anlatılmaya çalışılacaktır.

3.1. LTL (Linear Temporal Logic, Doğrusal Zaman Mantığı)

Doğrusal Zaman Mantığı, zamanı bir çizgi olarak düşünür ve geçmişten geleceğe uzanan bu çizgi üzerinde şimdiki bir nokta olarak kabul eder. Şayet, birden fazla olayın aynı anda gerçekleşmesi söz konusu ise, bu durumda alternatif olaylar arasında geçen doğrusal bir yolu LTL ile modellemek mümkün olabilmektedir. Üzerine kurulu olduğu önermeler mantığı ve doğrusal olarak modellediği önermeler itibarıyla literatürde Önerme Zamansal

Mantığı (Propositional Temporal Logic, PTL) adıyla da geçmektedir.

Hemen hemen bütün zamansal mantıklara temel teşkil eden bu yaklaşımda tanımlı olan bazı operatörler aşağıda sıralanmıştır [1]:

Gp Globally, p önermesinin her zaman için doğru olduğu anlamındadır.

Fp Future, p önermesinin gelecekte herhangi bir anda doğru olduğu anlamındadır.

pUq Until, p önermesinin q önermesinin doğruluğuna kadar doğru olduğu anlamındadır. Diğer bir deyişle, p olayı q olayından öncedir.

Xp Next, p önermesinin bir sonraki önerme olduğu anlamındadır.

Bahsedilen bu operatörleri örneklemek için aşağıdaki doğal dildeki cümleyi ele alalım:

“Ali uyuyorken Ayşe oyun oynuyordu. Ali uyanıp okula gitti.”

p : “Ali uyur”

q : “Ayşe oyun oynar”

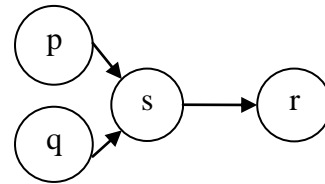
s : “Ali uyanır”

r : “Ali okula gider”

Bu önermeler arasındaki ilişkilerin LTL ile gösterilimi Eşitlik 1’deki gibidir.

$$p \ U \ s \ \wedge \ q \ U \ s \ \wedge \ X \ r \quad (1)$$

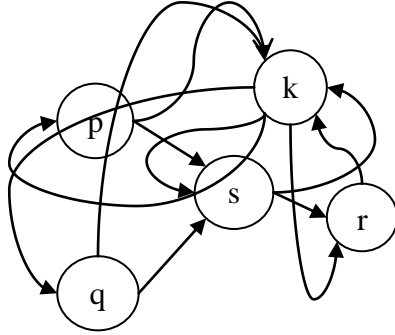
Yukarıdaki örnekte görüldüğü üzere zaman çizgisi üzerinde bir şimdi referansı alınmış ve bu referansa göre olaylar sıralanmıştır. Ayrıca, aynı anda olan farklı olayların bir olay üzerinden birleşmesi durumunda bu olaylara giden yollar çizilerek sıralama Şekil-2’deki gibi yapılabilmektedir.



Şekil-2: Örnek olay – ilişki şeması

Örneğimizde birinci yol $p \rightarrow s \rightarrow r$ şeklinde gidilirken, ikinci yol $q \rightarrow s \rightarrow r$ şeklinde gidilmekte ve bu iki yol aynı LTL modelinde temsil edilmektedir. LTL ile modelleme yapılırken bağlantısız bir bilgi

gösterilemez. Örneğin, yukarıdaki önermelere k ile isimlendirdiğimiz “Ahmet bisiklete biniyor” gibi ilişkisiz bir önerme eklenseydi bu bilgi LTL ile modellenemezdi (Şekil-3).



Şekil-3: İlerletilmiş olay – ilişki şeması

Bunun sebebi yeni gelen önermenin herhangi bir yol ile mevcut önermelere bağlanamamasıdır. Her olaydan sonra veya önce bu yeni olayın gelmesi mümkündür.

3.2. Diğer Zamansal Mantıklar

LTL üzerine geliştirilmiş ve LTL’de bulunan bütün operatör ve mantığı aynen kullanan bir diğer zamansal mantık, Kesit Mantığı’dır (Choppy Logic) [2]. Bu mantıkta yer alan kesit operatörü (ξ) yardımıyla olaylar, alt olaylara bölünebilir.

Örneğin, $\sigma \equiv p\xi q$ şeklindeki bir gösterimin anlamı σ olayının iki farklı parçadan oluştuğu ve p ’nin q ’dan önce geldiği veya q ’nun, p ’nin hemen ardından geldiğidir. “Ankaradan gelir gelmez alışverişe çıktı” cümlesinde iki olay bulunur (Ankaradan gelmek ve Alışverişe çıkmak). CL mantığı ile bu olaylar, Ankaradan gelme eylemi, bilet almak, otobüse binmek, yolda mola vermek veya İstanbul’a varmak gibi alt parçalara bölünebilir. Buradaki p ve q sembolleri bu alt parçaları oluştururken, σ sembolü “Ankaradan gelmek” eylemini temsil etmektedir.

Hesaplama Ağacı Mantığı’nda (CTL-Computation Tree Logic) zaman, olaylar üzerinde çatallanmalar olan bir ağaç yapısı kullanılarak modellenir [3][4][5]. Burada LTL mantığından farklı olarak A (her zaman) veya E (herhangi bir zaman) ön ekleri bulunmaktadır. CTL mantığına bulunan operatörler (EF, gelecekte bir zamanda; AF, gelecekteki her an için; EP, geçmişteki bir zamanda; AP, geçmişteki

her zaman için; EG, herhangi bir zamanda; AG, her zaman için) dir. LTL’de verdiğimiz örneği CTL için genişletecek olursak Eşitlik 2’yi, k önermesini de eklersek Eşitlik 3’ü yazabiliriz.

$$p \text{ AU } s \wedge q \text{ EU } s \wedge \text{EX } r \quad (2)$$

$$p \text{ AU } s \wedge q \text{ EU } s \wedge \text{EX } r \wedge \text{EG } k \quad (3)$$

Bu anlamda CTL mantığını birinci derece mantık üzerine kurulmuş (First Order Predicate Calculus) kabul edebilir ve mantıkta bulunan operatörler üzerinde bir etki alanı (domain) tanımlama imkanı verdiğini söyleyebiliriz.

Zaman Çatallanmalı Zamansal Mantık (BTTL-Branching Time Temporal Logic) belirsiz gelecek modellerinde alternatif model imkanı sunar ve gelecekteki olayların olasılıksal olarak ifade edilmesine imkan verir [6].

Aralıklı Zaman Mantığı (ITL-Interval Temporal Logic) LTL üzerine kurulmuş, kesikli, doğrusal ve geçmişi belirli, geleceği ise belirsiz bir mantıktır [7][8][9]. Bu mantıkta kullanılan operatörler \square Her zaman (Always); \diamond Bazı zamanlarda (Eventually); O Sonraki (Next) ve ξ Kesit operatörü (Chop) dür.

ITL mantığının getirdiği en büyük fark, olaylar arasındaki zamanın olay sayısı cinsinden belirtilebilmesidir. $Len(n)$ şeklinde sonucu tam sayı olan iki zamansal olay arasında geçen olayların sayısını gösteren bir fonksiyon kullanılır.

“Ali uyandı, giyinip dışarı çıktı ve okula gidene kadar kitap okudu.” cümlesinde (P: uyandı; Q: giyindi; R: okula gitti; S: kitap okudu) dört olay vardır. Bu cümlenin ITL mantığında gösterilimi Eşitlik 4’teki gibidir.

$$P\xi R \rightarrow \text{Ali uyandı ve okula gitti} \quad (4)$$

Aynı zamanda Eşitlik 4 açılarak, ITL mantığında Eşitlik 5 gibi yazılabilir.

$$P \xi Q \xi S \xi R \rightarrow \text{true (doğru)} \quad (5)$$

Aynı eşitliği Len fonksiyonunu kullanarak Eşitlik 6’daki gibi yazabiliriz.

$$Len(2) (P \xi R) \rightarrow \text{true (doğru)} \quad (6)$$

Burada ifade edilen, P ve R olayları arasında iki adet olay olduğudur.

Aralık Mantığı (IL-Interval Logic), olayın modellemesini zaman aralıkları ile yapmaktadır [10][11]. Aralık Mantığının temel operatörü $[\theta] \alpha$ olup, θ sembolü bir aralığı belirtirken α sembolü IL

için tanımlı olan herhangi bir önermeyi ifade etmektedir. Örneğin, “Ali bütün gün uyudu” (p: Ali uyudu; t : bütün gün) önermesinin IL mantığında gösterilimi Eşitlik 7’deki gibidir.

[t] p (7)

Bu aralık mantığı gösterimize LTL öğelerini de eklersek ve (Ali bütün gün aralıksız uyudu), (Ali bugün bir ara uyudu) gibi iki ayrı önermeyi birbirinden ayırmak istersek, önermeleri sırasıyla [t] □ p ve [t] ◇ p şeklinde yazabiliriz. Ayrıca tanımlanan aralık ve önermeler arasında öncesinde (before), sonrasında (after) ve sırasında (in) gibi ilişkilerde olabilir. Bu ilişkileri de göstermek için ⇒ sembolü kullanılır ve *sırasında* anlamındadır. Örneğimizi bu eklentiler doğrultusunda geliştirecek olursak,

Ali bugün uyudu

Ali bugünden sonra uyudu

Ali bugünden önce uyudu

şeklinde üç farklı aralık/önerme ilişkisi yazabiliriz.

Genişletilmiş Aralık Mantığı (EIL-Extended Interval Logic), Aralık Mantığının (IL) yetersiz kaldığı, özellikle gerçek zamanlı sistemlerde kullanılır [10]. Gerçek zamanlı sistemlerde karşılaşılan problem, bir olayın başlangıç ve bitiş zamanlarının kesin olmasını gerektirmesidir. Bu durumda aralık mantığı yetersiz olur, çünkü bir olayın bir aralıktan önce veya sonra olması kesinlik arz etmez. Bir zaman aralığı ile bir önerme ilişkilendirildiğinde önermenin doğruluğu için bir zamansal koşulun gerçekleşmesi şartı öne sürülebilir. Olayın bitişinde ise bu zamansal şart zaten gerçekleşmiş olmak zorundadır.

Gerçek Zamanlı Aralık Mantığı (RTIL-Real Time Interval Logic), Aralık Mantığının genişletilmiş halidir [12]. Gerçek zamanlı sistemlerde karşılaşılan problem olup, olayın başlangıç ve bitiş zamanının, diğer olayların başlangıç ve bitiş zamanlarını etkilemesidir. Bu durumda bir olayın başladığı ve bittiği zamanın, zamansal mantık üzerinde kesin olarak ifade edilmesi sağlanır.

3.3. Allen Zamansal Mantığı (Allen Interval Logic)

Allen Zamansal Mantığı (AZM) [13], olaylar arasındaki bağlantıları ve olayların sıraları üzerinde çalışır. Örneğin “olay A, olay B’den öncedir” veya “olay A ile olay B aynı zamandır” gibi işlemleri

kullanarak olaylar arasındaki ilişkileri modellemeye çalışır. AZM üzerindeki en temel işlemler, iki parametresi bulunan ve iki olay arasındaki ilişkiyi betimlemeye yarayan işlemlerdir. Bu durumu örnek üzerinden anlatacak olursak:

“Ali, odaya girdikten sonra, masanın üzerindeki elmayı yedi”

cümlesindeki olaylar sıralanacak olursa, Şekil-4’deki durum ortaya çıkacaktır. Şayet cümleler üzerinde daha detaylı bir işleme yapılırsa, Ali’nin ilk başta odanın dışında olduğu, odaya girdikten sonra masanın üzerindeki elmayı alması için masaya yaklaşması gerektiği de çıkarılabilir. Ali’nin masadan elmayı alana kadar, aslında elinde elma olmadığı da doğrudur.

Dışarıda	Girdi	Odanın içerisinde	
Masadan Uzakta		Yaklaştı	Masanın Yanında
Elması yok		Aldı	Elması Var
Yedi			

Şekil-4: Örnek olay-durum ilişkisi

Verilen cümleden, Ali’nin odanın dışındayken elması olmadığı veya Ali’nin masanın yanındayken bir elması olduğu veya Ali’nin elmayı yerken odada olduğu ve masanın yanında olduğu sonuçları da çıkarılabilir.

Bu olaylar zincirini modellemek için AZM üzerinde bazı değişkenleri tanımlamamız gerekir. Örneğin, girme (G) eyleminin, odanın dışında (OD) olma durumunu gerektirdiğini ve odaya girme eyleminden sonra odanın içinde (Oİ) olma durumuna geçildiğini söyleyebiliriz. Benzer şekilde masaya yaklaşma (MY) eylemi, masadan uzakta olma (MU) durumunu, masanın yakınında olma (MYO) durumuna çevirmiştir. Ayrıca, elmayı alma (EA) eylemi de elması yok (EY) durumunu, elması var (EV) durumuna çevirmektedir. Bütün bu eylemler aslında yemek (Y) eyleminin ön koşullarıdır.

Bahsedilen bu durum ve şartları birbirine bağlamak mümkündür.


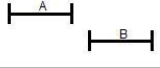
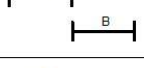
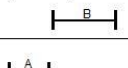



$Meets(OD,G) \wedge Meets(G,Oİ) \wedge During(G,MU) \wedge$
 $During(MU,Oİ) \wedge Meets(MU,MY) \wedge$
 $Meets(MY,MYO) \wedge During(MU,EY) \wedge$
 $During(EA,Oİ) \wedge During(EA,MYO) \wedge$
 $Meets(EY,EA) \wedge Meets(EA,EV) \wedge During(Y,EV) \wedge$
 $During(Y,MYO) \wedge During(Y,Oİ) \wedge Meets(EA,Y)$

Yazılan bu ifade, çıkarılabilecek zamansal ihtimalleri modellemektedir. Bunun yanında bir okuyucu, bu cümleden, Ali'nin henüz elmayı almadan aslında aç olduğu sonucunu da çıkarabilir. Bu durumda modelimiz:

$Occurs(aç, EY) \wedge Holds(aç, EA) \wedge Meets(aç, Y)$

şeklinde olmalıdır. Bu modele göre, Ali'nin henüz elması yokken, Ali açtı, ve Ali elmayı aldığı anda açtı ve Ali'nin aç olma durumu Ali'nin elmayı yemesine kadar sürdü anlamı çıkarılabilir. Modellenen AZM için aşağıdaki örnekler Şekil 5'te görselleştirilmiştir:

“Masanın yanına gidip elmayı alırken Ayşe uyandı. Ayşe uyanırken masanın yanına gidip elmayı almıştım.”

	A ile B eşit B ile A aynı zamanda
	A, B'den önce B, A'dan sonra
	A'yı B takip eder B, A tarafından izlenir
	A ile B çakışır B, A ile çakışır
	A, B'yi başlatır B, A tarafından başlatılır
	A, B'yi bitirir B, A ile biter
	A esnasında B olur B, A'yı kapsar

Şekil-5: Allen Zamansal ilişkileri

Bu işlemleri de kapsayan AZM aşağıdaki şekilde sıralanabilir:

- Before (x,y) or After (y,x) {önce veya sonra}

- Overlaps (x,y) or Overlapped (y,x) {üst üste binmek}
- Meets(x,y) or MetBy(y,x) {bir olayın bitmesi ile başkasının başlaması}
- Contains(x,y) or During (x,y) {bir olayın bir diğerini zamansal olarak kapsaması}
- Starts(x,y) or StartedBy(y,x) {bir olayın bir diğerini başlatması}
- Ends(x,y) or EndedBy(y,x) {bir olayın bir diğerini bitirmesi}
- Equals(x,y) {iki olayın tam olarak aynı zamanda olması}

Şekil 5'te gösterildiği gibi, Allen Zamansal modelinde, 13 farklı ilişki modeli bulunmaktadır. Allen Zamansal Mantığının bütün işlemleri, iki parametre üzerinde çalışmaktadır.

3.5. Reichenbach Zamansal Mantığı

Reichenbach Zamansal Mantığı [14], bu çalışma kapsamında incelenen ve anlatılan tek doğal dil işleme kökenli mantıktır. Bu mantığın kurulması sırasında üç ana unsur ele alınmıştır:

1. olayın geçtiği zaman (E)
2. olayın yazıldığı veya anlatıldığı zaman (S)
3. referans zamanı (R)

Örneğin, “Ali gelip gerçekleri anlattığında çoktan talimat vermişim.” cümlesinde geçen fiilleri ayıklayacak olursak:

(“(Ali gelip gerçekleri anlattığında)R (çoktan talimat vermişim)E.”)S

şeklinde ifade edilir. Gösterilmek istenen bütün cümlelerin şu anda anlatıldığı veya okunduğudur. Talimat verme eylemi tam olarak olayın geçtiği zamanda gerçekleşmiştir. Ali'nin gelip gerçekleri anlatması eylemi ise olayın zamanına referans alınan bir olaydır. Kısaca, bu cümlede bu iki olay arasında bir bağlantı kurulmuştur.

Dolayısıyla Reichenbach analizine göre verilen cümlede 3 farklı zamanın sıralaması Eşitlik 8'deki gibi olacaktır:

$$E < R < S \quad (8)$$

Olayın geçtiği zaman, Referans zamanından çok önce olmuştur, benzer şekilde olayın referans zamanı ise olayın anlatıldığı şimdiki zamandan öncedir.

Bu örnekten yola çıkarak Reichenbach 13 farklı zaman dizilimi alternatifi getirmiştir. Bu dizilimlere Çizelge 1’de yer verilmiştir.

Çizelge1 Reichenbach zaman ilişkileri

İlişki	Reichenbach Zaman İsmi (Tense Name)	İngilizce Zaman İsmi	Türkçe Karşılığı
E<R<S	Anterior past	Past perfect	Hikaye Geçmiş Zaman
E=R<S	Simple past	Simple past	Geçmiş Zaman
R<E<S			
R<S=E	Posterior past	YOK	Gelecek Zamanın Mazisi
R<S<E			
E<S=R	Anterior present	Present perfect	Rivayet Geçmiş Zaman
S=R=E	Simple present	Simple present	Geniş Zaman
S=R	Posterior present	Simple future	YOK
R<E			
S<E<R			YOK
S=E<R	Anterior future	Future perfect	YOK
E<S<R			
S<R=E	Simple future	Simple future	Gelecek Zaman
S<R<E	Posterior future		

4. TimeML

TimeML, anlatılan zamansal mantıkların kullanıldığı, makine tarafından işlenebilir bir işaretleme dilidir [15]. TimeML, 3 seviyede incelenebilir.

- Olay Seviyesi
- Sinyal Seviyesi
- Bağlantı Seviyesi

En basitten, karmaşığa doğru verilen bu sıralar, TimeML dilindeki çeşitli etiketlere karşılık gelir. Olay seviyesinde, işaretlenecek olan olay veya zaman kelime grupları belirtilir. Örneğin, “geldim”, “gittim” gibi temel kelimeler veya kelime grupları bu seviyede yer alır.

Sinyal seviyesinde, olay seviyesinde işaretlenen kelime veya kelime gruplarının etkinliği veya uzunluğu belirtilir. Örneğin, “iki saatte gittim” gibi

bir ifadede bulunan “gittim” kelimesi olay belirtirken, “iki saatte” ifadesi, bu olayın uzunluğunu belirtmektedir. Dolayısıyla sinyal seviyesinde gösterilen bu durum, olay seviyesinin üstüne geçirilen bir kılıf gibi düşünülebilir.

Bağlantı seviyesinde ise, sinyallenmiş olaylar arasındaki bağlantılar gösterilir. Örneğin, “yola çıktım ve iki saatte gittim” cümlesindeki “yola çıkmak” ve “iki saatte gitmek” şeklinde geçen iki ayrı olay arasında önce / sonra ilişkisi bulunmaktadır. Bu durumda bu iki olay arasındaki bağlantıyı gösteren bir etiket eklenir.

TimeML, olay seviyesinde Reichenbach Zamansal Mantığından faydalanırken, bağlantı seviyesinde Allen Zamansal Mantığından faydalanır.

4.1. Zamansal Kayıtlar (yaprak seviyesi)

<EVENT>: En alt seviyedeki etiketlerdir ve olayları tutmak için kullanılır. Genel olarak her olay bir fiildir ve DTD, (Document Type Definition) kaydı aşağıda verilmiştir:

```
attributes ::= eid class
eid ::= ID
{eid ::= EventID
EventID ::= e<integer>}
class ::= 'OCCURRENCE' |
'PERCEPTION'
| 'REPORTING' | 'ASPECTUAL'
| 'STATE' | 'I_STATE' | 'I_ACTION'
```

Tanımdan da anlaşılacağı üzere bir olayın ayırt edici bir ID değeri bulunmalı ve bu değer bir tam sayı olmalıdır. Ayrıca, bir olay tanımlanan sınıflardan (Class) sadece bir tanesine dahil olabilir.

4.2. Olaylar arası bağlantı seviyesi

TimeML için olaylar arası bağlantıların modellenmesi Allen Zamansal Mantığından esinlenerek hazırlanmıştır. AZM için tanımlı olan ilişki yapısı aşağıda verilmiştir:

```
lid ::= LinkID
eventInstanceID ::= EventInstanceID
timeID ::= TimeID
signalID ::= SignalID
```

```

relatedToEventInstance ::=
EventInstanceID
relatedToTime ::= TimeID
relType ::= 'BEFORE' | 'AFTER' |
'INCLUDES' | 'IS_INCLUDED' |
'DURING' | 'DURING_INV' |
'SIMULTANEOUS' | 'IAFTER' |
'IBEFORÉ' |
'IDENTITY' | 'BEGINS' | 'ENDS' |
'BEGUN_BY' | 'ENDED_BY'

```

İki olay arasındaki bağlantının kurulmasında bu yapıya uygun bir tanım gerekmektedir ve ayrıca ilişki tipi AZM için tanımlı olan tiplerden (relType) olmalıdır. Bu ilişki tipleri Çizelge 2’de örnekleri ile birlikte verilmiştir.

Çizelge2 TimeML ilişki tipleri

relType	Örnek
BEFORE	Geldi ve beni gördü.
AFTER	
INCLUDES	Ali, Ankara’ya geçen Perşembe vardı.
IS_INCLUDED	
DURING	Ali, iki yıllığına öğrenci oldu.
DURING_INV	
SIMULTANEOUS	
IAFTER	Uçak dağa çarpınca bütün yolcular öldü.
IBEFORÉ	
IDENTITY	Ali Bursa’ya arabası ile giderken köfte yedi.
BEGINS	Ali, 1980’den beri ders veriyor.
ENDS	Ali, saat 7’ye kadar okuldaydı.
BEGUN_BY	
ENDED_BY	

4.3. Örnek TimeML dosyası

TimeML dilinin örnek bir cümle için nasıl çalıştığı bu bölümde açıklanmaktadır. Örnek cümlemiz, “Masanın yanına gidip elmayı alırken Ayşe uyandı” olarak verilmiş olsun. Bu cümlemin TimeML olarak işaretlenmiş hali:

```

Masanın yanına <EVENT eid="e1"
class="OCCURRENCE">gidip</EVENT>

```

```

elmayı <EVENT eid="e2"
class="OCCURRENCE">alırken</EVENT>
Ayşe
<EVENT eid="e3"
class="OCCURRENCE">uyandı</EVENT>.
<MAKEINSTANCE eventID="e1"
eiid="ei1" tense="PAST"
aspect="NONE" polarity="POS"
pos="UNKNOWN"/>
<MAKEINSTANCE eventID="e2"
eiid="ei2" tense="PAST"
aspect="NONE" polarity="POS"
pos="UNKNOWN"/>
<MAKEINSTANCE eventID="e3"
eiid="ei3" tense="Hikaye"
aspect="NONE" polarity="POS"
pos="UNKNOWN"/>
<TLINK lid="12" relType="BEFORE"
eventInstanceID="ei1"
relatedToEventInstance="ei2"
origin="USER"/>
<TLINK lid="11"
relType="IS_INCLUDED"
eventInstanceID="ei3"
relatedToEventInstance="ei2"
origin="USER"/>

```

şekindedir. Cümlede sadece olay ve olaylar arasındaki bağlantıları belirten kelimeler işaretlenmiş, bunun dışındaki kelimeler serbest olarak bırakılmıştır. Ayrıca, dosyanın sonunda olaylar arasındaki ilişkileri gösteren TLINK etiketleri yer almaktadır.

5. Türkçe Zamansal Mantık

Her olayda bir zaman ve her dilde de bu zamanın ifade edilme şekli farklılıklar gösterebilir. Türkçe dilinde de doğal olarak bir zamansal mantık kullanılmaktadır. Örneğin, Türkçe’deki geçmiş zamanın rivayet ve hikaye şeklinde ayrılması, kişinin olayı bilmesi veya öğrenmesi arasındaki farkı belirler. Böyle bir fark bulunmayan Latin ailesi dillerini kullanan kişilerin, ürettikleri mantıkta bu durum eksik kalmaktadır.

5.1. Reichenbach Zamansal Mantığı ve Türkçe

Bu çalışmada çözüm olarak, TimeML dilinin üzerine kurulu olan Reichenbach Zamansal Mantığında değişiklik yapılması gerekmiştir. Bu değişiklik Çizelge 3’te sunulmuştur.

Çizelge3 Reichenbach Zamansal Mantığının Türkçeye Uygulanması

İlişki	Örnek
B,E<R<S	Dün Geleceğim dediydi
Ö,E<R<S	Dün Geleceğim demişti
B,E=R<S	Geliyorum dedi
Ö,E=R<S	Geliyorum demiş
B,R<E<S	Geleceğim dedi
Ö,R<E<S	Geleceğim demiş
B,R<S=E	Geldim diyor
Ö,R<S=E	Geldim diyor
B,R<S<E	Geldim diyecek
Ö,R<S<E	Geldim diyecek
B,E<S=R	Şimdi döneceğini söyleyiydi
Ö,E<S=R	Şimdi döneceğini söylemişti
B,S=R=E	Geliyorum diyor
Ö,S=R=E	Geliyorum diyor
B,S=R<E	Şimdiye dönecek
Ö,S=R<E	Şimdiye dönecek
B,S<E<R	Geleceğim diyecek
Ö,S<E<R	Geleceğim diyecek
B,S=E<R	Geleceğim diyor
Ö,S=E<R	Geleceğim diyor
B,E<S<R	Yarın geleceğini söyleyiydi
Ö,E<S<R	Yarın geleceğini söylemişti
B,S<R=E	Geliyorum diyecek
Ö,S<R=E	Geliyorum diyecek
B,S<R<E	Geldim diyecek
Ö,S<R<E	Gelmişim diyecek

Çizelge 3’te bulunan B fonksiyonu, bilinen ve Ö fonksiyonu ise öğrenilen durumu ifade etmektedir.

Türkçede desteklenmeyen bazı durumlar teorik olarak mümkün olsa da Çizelge 3’te gösterilmemiştir. Ayrıca, Çizelge 3’te görüldüğü gibi bazı mantıksal durumlar Türkçede de aynı anlama gelmektedir. Örneğin, “Geliyorum diyor” şeklindeki bir cümlede gelme eyleminin, anlatan kişi tarafından biliniyor olup, olmaması anlatımı değiştirmez. Bu şekilde anlam değişmeyen durumlar elendiğinde, Türkçe için genişletilen Reichenbach Zamansal Mantığının Çizelge 4’te veri hali 19 alternatif içermektedir.

Çizelge4 Reichenbach Zamansal Mantığının Türkçeye Uygun Hali

İlişki	Örnek
B,E<R<S	Dün Geleceğim dediydi
Ö,E<R<S	Dün Geleceğim demişti
B,E=R<S	Geliyorum dedi
Ö,E=R<S	Geliyorum demiş
B,R<E<S	Geleceğim dedi
Ö,R<E<S	Geleceğim demiş
R<S=E	Geldim diyor
R<S<E	Geldim diyecek
B,E<S=R	Şimdi döneceğini söyleyiydi
Ö,E<S=R	Şimdi döneceğini söylemişti
S=R=E	Geliyorum diyor
S=R<E	Şimdiye dönecek
S<E<R	Geleceğim diyecek
S=E<R	Geleceğim diyor
B,E<S<R	Yarın geleceğini söyleyiydi
Ö,E<S<R	Yarın geleceğini söylemişti
S<R=E	Geliyorum diyecek
B,S<R<E	Geldim diyecek
Ö,S<R<E	Gelmişim diyecek

Reichenbach Zamansal Mantığına yapılan bu ilave Türkçe’deki zamansal gösterim zenginliğinden kaynaklanmaktadır.

5.2. Allen Zamansal Mantığının Geliştirilmesi

Türkçede bulunan ve bazı olayları belirten fiillerin tekrarlanmasından oluşan ifadeler de Allen Zaman Mantığında gösterilememektedir. Örneğin, “Arabanın sinyali, yanıp sönyordu” cümlesinde iki ayrı fiil bulunmaktadır. Yanmak ve sönmek fiilleri arasındaki ilişki maalesef Allen Zamansal Mantığı tarafından modellenemez. Bunun sebebi tekrarlı olayların Türkçe’de birbirinin tersi fiillerle gösterilmesine karşılık, Latin dil grubunda benzer bir yaklaşımın bulunmamasıdır. Örneğin, İngilizce’de bu tip ifadeler tek bir fiil ile karşılanır.

Benzer şekilde Türkçe’de bir fiilin sürekliliğini ifade etmek için aynı fiilin tekrarlanması da söz konusudur. Örneğin, “Ali vitrinlere bakıp bakıp hayıflanıyordu” cümlesine ”bakıp bakıp” ifadesi, bakma eyleminin sürekliliğini ifade etmektedir.

TimeML dilinin, Türkçe’den farklı olarak eksik olan bazı özellikleri de mevcuttur. Örneğin, bir olayın kendisine atıfta bulunması durumu TimeML dilinde bulunmaz. “Bu laf kulağına küpe olsun” cümlesi

Türkçe'ye özgü bir zamansal mantık taşımamakla birlikte, cümle, yine kendisine atıfta bulunmaktadır. Bu durum Allen Zamansal Mantığı'nda bulunmadığı gibi mevcut TimeML tarafından da desteklenmemektedir.

Bu iki durumun TimeML tarafından kapsanması için, Allen Zamansal Mantığı üzerine kurulu olan mevcut ilişki tiplerine iki ek yapılması yeterlidir. Bu durum Çizelge 5'te gösterilmektedir.

Çizelge5 TimeML Türkçe'yi Kapsayan İlişki Tipleri

relType	Örnek
BEFORE	Geldi ve beni gördü.
AFTER	
INCLUDES	Ali, ankaraya geçen Perşembe
IS_INCLUDED	
DURING	Ali, iki yılığına öğrenci oldu.
DURING_INV	
SIMULTANEOUS	
I_AFTER	Uçak dağa çarpınca bütün
IBEFORE	
IDENTITY	Ali bursaya arabası ile giderken
BEGINS	Ali, 1980'den beri ders veriyor.
ENDS	Ali, saat 7'ye kadar okuldaydı.
BEGUN_BY	
ENDED_BY	
Tekrarlatır	Damlayadamlaya göl olur.
Atıf	Bu söylediklerimi unutma

5.3. Yokluğun İfadesi

Farklı bir problem de zamansal ifadelerden bazılarının gösteriminin imkansızlığıdır.

“ -- Ne zaman geliyorsun?
-- Asla! ”

diyaloğunda geçen “Asla” kelimesi veya Türkçe'de bulunan “Hiçbir zaman” gibi ifadelerin gösterimi veya diğer olaylarla bağlanması imkansızdır. TimeML içerisinde bulunan ve sinyal seviyesinde bir fiile eklenen durum, bu durumdan farklıdır. Bu durumu örnek ile açıklayacak olursak:

“Ali, okula gitti, derse girdi ve hiç not tutmadan eve döndü”

cümlesinde “not tutmamak” olarak geçen fiil aslında olumsuz fiildir ve bir önceki örnekte bulunan, “Hiçbir zaman” ile karıştırılmamalıdır. “Hiçbir

zaman not tutmamak” ile kastedilen durum, okula gitmiş, derse girmiş ve eve dönmeye önce not tutmamak ile karıştırılmamalıdır. TimeML, ikinci durum için çözüm üretirken, bahsedilen “Asla” anlamı için bir çözüm sunmamaktadır.

Bu durumda, bir çözüm, fiilin olumsuzluğunun her zamana çevrilmesidir. Örneğin, “Ali hiçbir zaman yalan söylemez” cümlesinin modellenmesi sırasında, “Her zaman” için “Ali yalan söylemez” modellenmesi önerilebilir. Yani, yok olan bir şeyin olumsuzunun her zaman var olduğu iddia edilebilir. Bu iddia kısmen problemi çöze de varlık felsefesinde bulunan ve varlığın, yokluğun tersi olmadığı iddiası ile çelişir. Kısaca yokluğun ifadesi, varlığın tersi ile gösterilemez¹ ve bu durumu TimeML dilinde göstermek için ilave bir gösterim eklenmesi en kolay ve başarılı çözümdür. Bu ilave gösterime göre, bütün yokluğa bağlı olaylar, yokluk kavramı ile ilişki tipine göre modellenenecektir. Bu durumu gösteren bazı çözümler aşağıda sunulmuştur:

“Tanrı, Adem'i yoktan yarattı”

Bu durumda, yaratma eylemi, yok kavramına göre sonra (Next) ilişkisi içerisinde yer alır.

“Uzay boşluğunda, kendiliğinden yetişen bir bitki hiçbir zaman olamaz”

Bu durumda, yetiştirme olayı, hiçbir zaman gerçekleşemez ve yokluk ile içerir (INCLUDES) ilişkisine sahiptir.

6. Sonuçlar ve Öneriler

Bu çalışmada, zamansal gösterimler için kullanılan ve varlıkbilim (ontoloji) veya doğal dil işleme çalışmaları için kullanılabilir olan temel zamansal mantıklar incelenmiş, TimeML işaretleme dili ve bu dile temel teşkil eden Reichenbach ve Allen Zamansal Mantıkları açıklanmış ve Latin dil aileleri gözetilerek üretilen bu zamansal mantık ve işlemeli dillerin aslında Türkçede, daha zengin olan zamansal yapıyı taşıyamadıkları gösterilmiştir.

Çözüm olarak, mevcut TimeML çalışmaları ile uyumlu olması açısından, bazı etiketlere, ilave bazı özellikler eklenmesi önerilmiştir.

¹ Lütfen, çalışmanın doğal dil ile sınırlı olduğunu ve dilde bulunan her ismin, varlık olduğunu, aksi halde, “olmayan bir şeyin” isminin olamayacağını hatırlayınız

Kaynakça

- [1] Pnueli, A. (1971) The temporal logic of programs. In Proceedings of the 18th Annual Symposium on Foundations of Computer Science (Providence, RI). IEEE, New York, pp. 46-57.
- [2] Rosner, R. And Pnueli, A. (1986). A choppy logic. In Proceedings of the First IEEE Symposium on Logic in Computer Science, IEEE Press, Piscataway, NJ, 306-313.
- [3] Clarke, E. M., Emerson, E. A., And Sistla, A. P., (1986). Automatic verification of finite-state concurrent systems using temporal logic specifications. ACM Trans. Program. Lang. Syst. 8, 2 (Apr. 1986), 244-263.
- [4] Clarke, E. M. And Grumberg, O. (1987). The model checking problem for concurrent systems with many similar processes. In Proceedings of the International Conference on Temporal Logic in Specification (Altrincham, UK, Apr. 8-10), B. Banieqbal, H. Barringer, and A. Pnueli, Eds. Springer Lecture Notes in Computer Science Springer-Verlag, New York, NY, 188-201.
- [5] Stirling, C. (1987). Comparing linear and branching time temporal logics. In Proceedings of the International Conference on Temporal Logic in Specification (Altrincham, UK, Apr. 8-10), B. Banieqbal, H. Barringer, and A. Pnueli, Eds. Springer Lecture Notes in Computer Science Springer-Verlag, New York, NY, 1-20.
- [6] Ben-Ari, M., Pnueli, A., And Manna, Z. (1983). The temporal logic of branching time. Acta Inf. 20.
- [7] Halpern, J., Manna, Z., And Moszkowski, B. (1983). A hardware semantics based on temporal intervals. In Proceedings of the Tenth Colloquium on Automata Languages and Programming (Barcelona, Spain, July), J. Diaz, Ed. Springer-Verlag, New York.
- [8] Moszkowski, B. And Manna, Z. (1984). Reasoning in interval logic. In Proceedings of the ACM/NSF/ONR Workshop on Logics of Programs, Springer-Verlag, Secaucus, NJ, 371-384.
- [9] Moszkowski, B. C. (1986). Executing temporal logic programs. Ph.D. Dissertation. Cambridge University Press, New York, NY.
- [10] Melliar-Smith, P. M. (1987). Extending interval logic to real time systems. In Proceedings of the Conference on Temporal Logic Specification (UK, Apr.), B. Banieqbal, H. Barringer, and A. Pnueli, Eds. Springer-Verlag, New York, 224-242.
- [11] Schwartz, R. L., Melliar-Smith, P. M., And Vogt, F. H. (1983). An interval logic for higherlevel temporal reasoning. In Proceedings of the Second ACM Symposium on Principles of Distributed Computing (Aug.), ACM Press, New York, NY, 173-186.
- [12] Razouk, R. And Gorlick, M. (1989). Real-time interval logic for reasoning about executions of real-time programs. SIGSOFT Softw. Eng. Notes 14, 8 (Dec. 1989), 10-19.
- [13] Allen, J. F. (1984). A General Model of Action and Time. Artificial Intelligence 23, 2, July 1984.
- [14] Reichenbach, H., (1947). Elements of Symbolic Logic, New York: Macmillan
- [15] Pustejovsky, J., Castaño, J., Ingria, R., Aurí, R., Gaizauskas, R., Setzer, A. And Katz, G., (2003). TimeML: Robust Specification of Event and Temporal Expressions in Text. IWCS-5, Fifth International Workshop on Computational Semantics.