DİL MODELLEMEDE BELİRSİZLİK PROBLEMİNİN ETMENLENMİŞ DİLBİLGİSİ İLE GİDERİLMESİ

Zeynep Altan
İstanbul Üniversitesi, Mühendislik Fakültesi,
Bilgisayar Mühendisliği Bölümü 34850- Avcılar, İstanbul.
zaltan@istanbul.edu.tr

Özet: *Bu çalışmada Türkçe tümcelerin sözdizimsel çözümlemesindeki belirsizliği gidermek amacı ile geliştirilen yeni bir çözümleme algoritmasına temel teşkil edecek ağaç yapısı çıkarsanmaktadır. İncelenen tümcelerin tanımlandığı bağlamdan-bağımsız dilbilgisinin genişletilerek değiştirildiği yeni dilbilgisi, bir başka ifade ile etmenlenmiş dilbilgisi, çözümleyicinin karşılaşabileceği belirsizlik durumlarını ortadan kaldırmaktadır. Türkçe kelimelerde özellikle eylemlerin aldıkları eklerle biçimbilimsel analizde oluşan karmaşıklık, çekim ekleri için özel bir yapı tanımlanarak azaltılmıştır. Tümcelerin sözdizimsel analizine ait ağaç yapısı ile, aynı tümcelerin etmenlenmiş dilbilgisi tanımlarına ait ağaç yapıları Visual Basic 6.0 programlama dilinde tasarlanmış ve çözüm ağaçları biçimbilimsel analiz sonuçları ile birlikte arayüzde gösterilmiştir.*

Anahtar Kelimeler: Doğal Dil İşlemleme, Sözdizimsel Çözümleme, Biçimbilimsel Analiz, Belirsizlik, Etmenlenmiş Dilbilgisi.

1. GİRİŞ

Berimsel dilbilimde doğal dilin modellenmesi sırasında pek çok belirsizlik (ambiguity) durumu ile karşılaşılır. Bunlar biçimbilimsel, sözdizimsel, anlamsal, sesbilimsel, edimbilimsel seviyelerde olabilir. İncelenen dilin özellikleri de bu belirsizlik durumlarını etkilemektedir. Türkçe tümceler çözümlenirken en belirgin belirsizlik durumu sözdizimsel-sözlüksel (syntactic-lexical) arayüzde gerçekleşir. Sözdizimsel-anlamsal belirsizliğin de (syntactic-semantic ambiguity) diğeri ile birlikte incelenmesi çözümlemede yapılacak hataları azaltacaktır.

Bu çalışmada Türkçe tümcelerin sözdizimsel analizinden başlanarak, etmenlenmiş bir çözümleme algoritmasına temel teşkil edecek etmenlenmiş dilbilgisi (factored grammar) tanımı verilmekte ve bu dilbilgisine ait ağaç yapısı oluşturularak yeni çözümleme algoritmasını yapısı tanımlanmaktadır. Çalışmada, tümcelerin sözdizimsel çözümlemesi diğer pek çok dilde olduğu gibi Chomsky sıradüzenindeki bağlamdan bağımsız dilbilgisine (context-free-grammar, CFG) göre gerçeklenmektedir. Ayrıca Türkçe'nin sondan eklemeli bir dil olması nedeni ile, biçimbilimsel analizi önem kazanmıştır. Özellikle eylem ifade eden kelimelerin biçimbilimsel analizi çalışmanın ilk kısmının büyük bir bölümünü oluşturmuştur. İncelenen tümce yapılarının biçimbilimsel analizi ve daha sonra sözdizimsel analizi tamamlandıktan sonra, karşılaşılabilecek belirsizlik durumlarının etkilerini en aza indirecek bir çözümleme algoritmasının tasarımı için bağlamdan bağımsız dilbilgisi genişletilmiş ve etmenlenmiş dilbilgisi şeklinde yeniden düzenlenmiştir. Böylece diğer pek çok çözümleme algoritmasına alternatif olabilecek belirli (unambiguous) bir algoritmanın tasarlanması sağlanmıştır.

Türkçe'nin dilbilimsel özelliklerinin zenginliği, Türk dili üzerine yapılan berimsel dilbilim uygulamalarını giderek arttırmaktadır. Bu çalışmanın 2. Bölüm'ünde isim ve eylem türündeki kelimelerin biçimbilimsel analizi ve tümcelerin sözdizimsel analizi gerçekleştirildikten sonra, 3. Bölüm'de geleneksel çözümleme algoritmaları özetlenmekte, etmenlenmiş dilbilgisi kavramı Bölüm 4'de açıklanmaktadır. Geliştirilecek yeni algoritmaya ait tanımlamalar ise 5. Bölüm'de verilmektedir. Biçimbilimsel analizden başlayarak yeni çözümleme algoritmasının tasarımına kadar gerçekleştirilen işlemlerin tümü Visual Basic 6.0 programlama dilinde yazılmıştır.

2. BİÇİMBİLİMSEL VE SÖZDİZİMSEL ÇÖZÜMLEME

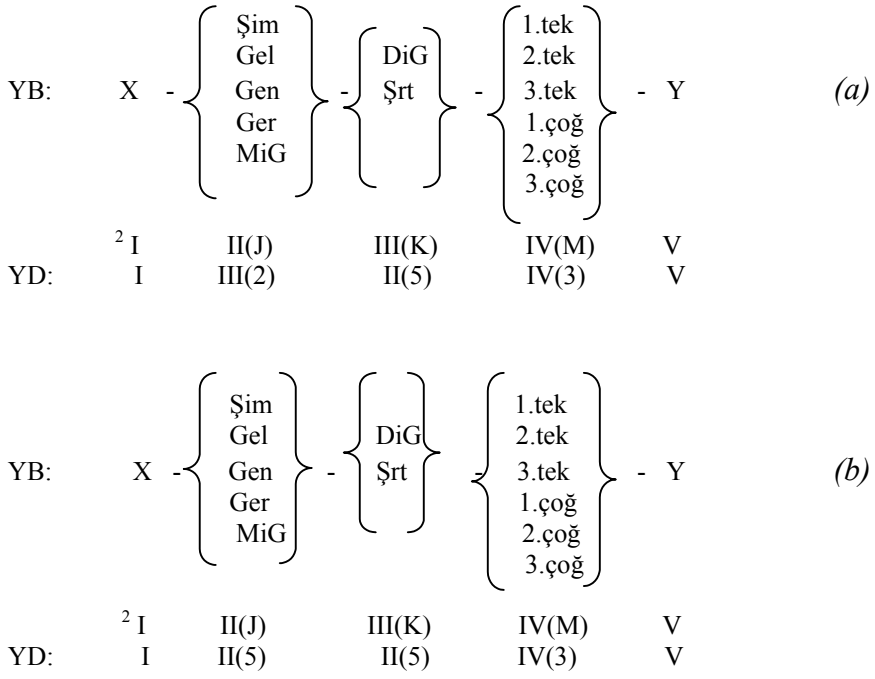
Çalışmanın biçimbilimsel analiz kısmında isim türündeki kelimeler için çekim ekleri ve ad tamlamaları incelenirken, eylem çekimi kiplerine ve bileşik zamanlı eylem oluşturma durumlarına göre incelenmiştir. 9 farklı kipten emir ve istek kipi dışında kalan ekler için, dilbilgisinin dönüşümlü olma özelliği kullanılmıştır. Böylece ağaç yapısı içerisinde kip eklerinin taşınması (move) ve kopyalanması (copy) ile işlemler basitleştirilmiş ve izlenmeleri kolaylaştırılmıştır.

Tablo 1: *Eylem kiplerine ait iki farklı terminal ulamı*

Ek1	Ek2
Şimdiki Zaman (Şim)	Belirli Geçmiş Zaman (DiG)
Gelecek Zaman (Gel)	Dilek Koşul (Şrt)
Geniş Zaman (Gen)	
Gereklilik(Ger)	
Belirsiz Geçmiş Zaman (MiG)	

Dilbilgisi kuralları P: { C → AÖ EÖ, AÖ → Sif AÖ, AÖ → Ad (Eki),
EÖ → (Zarf) EÖ, EÖ → Eyl (Ek1) (Ek2)(KiE) }[1]
olarak tanımlanmış CFG, "akıllı elemanı önceden ayarlasaymış" tümcesinin sözdizimsel analizini gerçekleştirirken, eylem olan "ayarlasaymış" kelimesine ait dilbilgisi kuralı için hem Ek1 hem de Ek2 terminal ulamlarını kullanır. Fakat, *Tablo 1*'e göre Ek1 ile Ek2 ulamlarının yer değiştirmesi gerekmektedir. Bu değişim birbirinin yerine geçme şeklinde olacağı için, bir taşıma ya da kopyalama işlemi gerekmektedir. Yapısal betimleme (YB) eylem kipleri (emir ve istek kipi dışında) için tanımlandığında, yapısal değişim (YD) *Şekil 1-a'*da görüldüğü gibi Ek1 ve Ek2 arasında birer "taşıma" işlemi şeklinde gerçekleşecektir. Eğer sözdizimsel analizi yapılan tümce "küçük çocuğu biraz azarlamışmış" ise, aynı yapısal betimleme için yapısal değişim Ek1'in Ek2'ye kopyalanması şeklinde gerçekleşecektir (*Şekil 1-b*). Böylece, "Üretici Dönüşümsel Dilbilgisi" (Transformational-Generative Grammar) kavramı ile dilbilgisine uygun sonsuz sayıda tümce üretilebilmekte ve dildeki sonlu sayıdaki kural çeşitli dönüşümler gerçekleştirilerek derin yapılardan yüzey yapılara geçiş sağlanmaktadır. Aslında değişik tümce türleri arasındaki eşdeğerlik ilişkileri belirlenerek edinç

[1] ( ) arasında kalan ekler seçimlidir ve kümenin elemanları çalışmada kullanılan dilbilgisine aittir.



Şekil 1: *Tümcelerin derin yapısını (deep structure), yüzey yapıya (surface structure) dönüştüren iki tümce örneği; (a)"akıllı elemanı önceden azarlasaymış" tümcesi için eylemde taşıma işlemi (b)" küçük çocuğu biraz azarlamışmış" tümcesi için eylemde kopyalama işlemi*

biçimselleştirilmektedir *[6]*. Bu simgelenişler İngilizce tümcelerdeki özne-yüklem uyumunun genelleştirilmesi için tanımlanmış olan "Number Agreement T_N" , "Affix Hopping T_{AH}" , "V-Movement T_M" dönüşümlerinin içerdikleri kopyalama ve taşıma işlemlerine benzer olarak geliştirilmiştir *[4]*. *Şekil 2-a* ve *Şekil 2-b*'de yukarıdaki özelliklerin kullanıldığı iki farklı sözdizimsel analiz sonucu görülmektedir.

2.1 Sözdizimsel Analizde Belirsizlik

Doğal dillerin sözdiziminin incelenmesi sırasında karşılaşılan pek çok belirsizlik durumu, son yıllarda hesaplamalı dilbilimde istatistiksel yöntemler üzerindeki çalışmaları arttırmıştır. Karşılaşılabilecek belirsizlik durumları, tümcelere uygun dilbilgisi kurallarının üretilememesi veya tümcelere uygun dilbilgisi kurallarının aynı zamanda dile uygun olmayan başka tümceleri de içermesi şeklinde olabilir. Örneğin: "kurnaz savaş komutanları" deyimi çalışmada kullanılan CFG kuralları ile iki farklı şekilde çözümlenebilir. Çözümleme ağaçlarından biri sıfata iliştirilmiş tamlama yapısında bir türetme gerçekleştirirken, diğerinde ada

---

[2] J=1,2,3,4,5 ; K=1,2 ; M=1,2,3,4,5,6

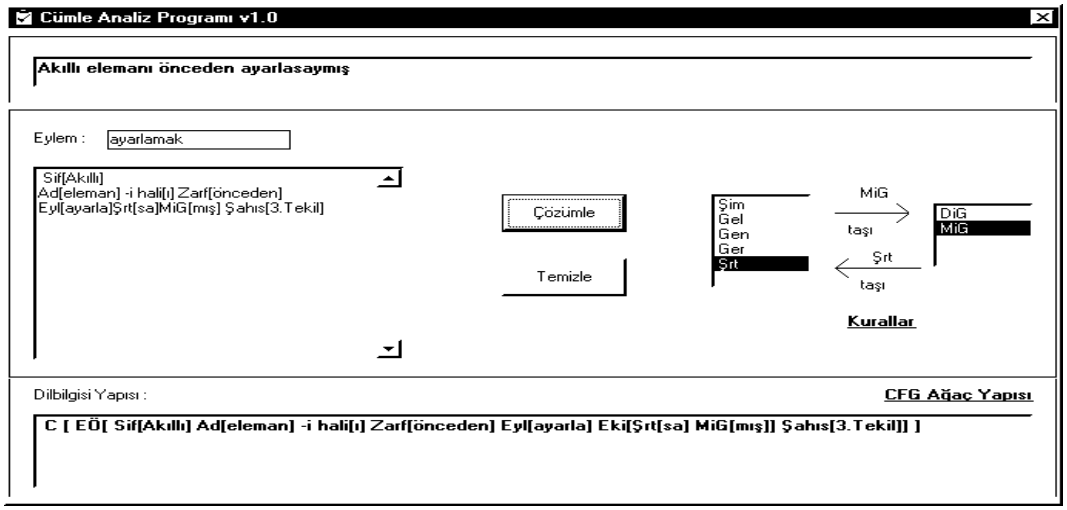
iliştirilmiş bir tamlama ile türetme gerçekleşir. Birinci türetme ağacının hiç bir şekilde doğru olmayan anlamsal yorumu vardır. Çünkü, sadece canlıların kurnaz olabilecekleri (savaşın kurnaz olamayacağı) bir gerçektir. Fakat her zaman bu mantıkla tümceyi veya deyimi belirli hale getiremeyiz. Artımlı bir yorumlama ile, standart anlamsal ve biçimsel simgelem kullanılarak anlamsal belirsizliklerin ortadan kaldırılması da mümkündür.

3. GELENEKSEL ÇÖZÜMLEME ALGORİTMALARI

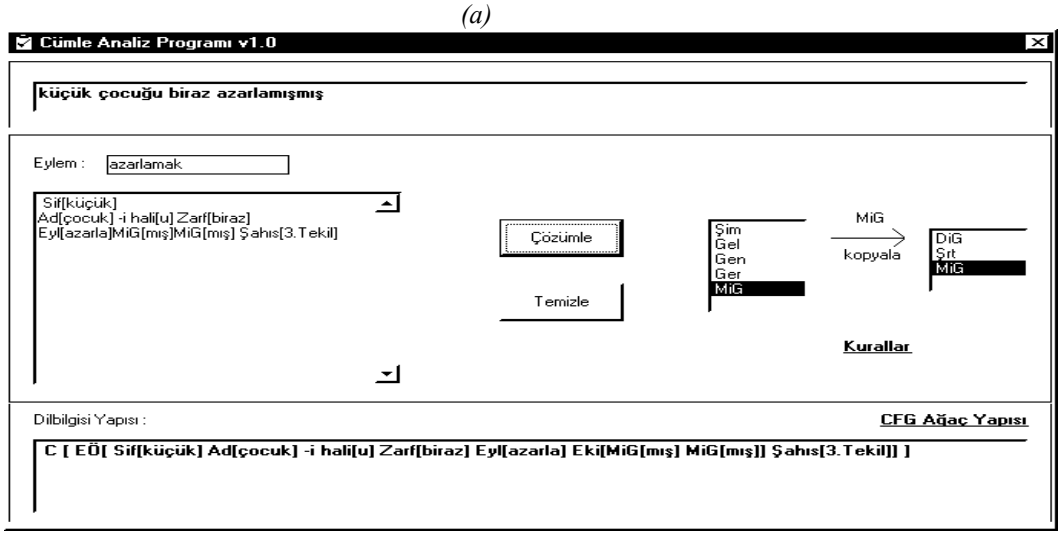
Doğal dillerin modellenmesinde farklı çözümleme algoritmalarının kullanılması mümkün olmasına rağmen, bu algoritmaların bazı dilbilgisi türleri için uygun olmadığı unutmamalıdır. Bunun en belirgin örneği, tepeden-tabana (top-down) çözümleme için dilbilgisi kuralları $A \rightarrow A\alpha$ biçiminde soldan-özyineli olarak verildiğinde, sonsuz sayıda kısmı çözüm elde edilmesi olasılığıdır. Bu durumda tabandan-tepeye (bottom-up) bir çözümleme oluşturan kaydırma-indirgeme (shift- reduce) çözümleyici, aynı zamanda soldan-sağa (left to right) çözümleme özelliği ile alternatif bir analiz türüdür. Earley (1970) etkin bir bağlamdan-bağımsız çözümleme algoritması olarak tepeden-tabana bir kontrol yapısı ile tabandan-tepeye bir çözümleme algoritması geliştirmiştir *[2]*. Burada artımlı olarak dinamik programlama gerçekleştirilmekte, fakat üstel olması gereken karmaşıklık, n çözümlenmiş katarın uzunluğu olmak üzere, belirli dilbilgisi için $O(n^2)$, belirsiz dilbilgisi için $O(n^3)$ mertebesinde elde edilmektedir.

Tepeden-tabana en soldan türetmeler herhangi bir son giren ilk çıkar otomat (push down automata, PDA) ile ifade edilebilir. Bu otomat, yığın içeren bir sonlu durum otomatıdır. Yığın ise son girenin ilk çıkması koşulu ile, girişlerin ilave edildiği (push) ve kaldırıldığı (pop) bir veri yapısıdır. Tepeden-tabana çözümleme algoritması giriş katarındaki sonraki sembolü (look-ahead symbol) işaretlediği için, çözümleyici tahmini ulamlar oluşturmaz ve sonraki kelime ile bağlantılı kısmi ağaç yapıları üretmek ister. Böylece birbirleri ile bağlantıları öncedeki adımlarda tamamlanmış bir kısmi analiz ve yeni bir sözcük için, mevcut yapıdan yeni bir yapıya genişlemenin mümkün olup olmadığı berimsel olarak araştırılır. Dilbilgisi dönüşümleri, uygulanan çözümleme yöntemine göre tanımlanır. Eğer tepeden-tabana bir çözümleme gerçekleştiriliyorsa, bir ebeveyn ulam ve bu ulama ait genişleyen kural herhangi bir çocuğundan önce bildirilirken, tabandan tepeye çözümlemede ebeveyn ulam ve genişleyen kural tüm çocuklardan sonra tanımlanmalıdır. Sol-köşeden başlayan standart çözümleyicilerde ise, bir ebeveyn ulamı ve buna ait kural en sol çocuğu tamamladıktan sonra, fakat diğer tüm çocuklardan önce bildirilir. *Şekil 3,* bu betimlemelere ait çözümleme algoritmaları için düğüm noktalarının işleme giriş sıralarını göstermektedir.

Bu çalışmada geliştirilen tahmini çözümleme algoritması, herhangi bir giriş katarı için o katarın ait olduğu dilbilgisi kurallarından oluşturulan etmenlenmiş türetme ağacına göre yazılmıştır. Çözümleyici öncelikle tanımlanan etmenlenmiş dilbilgisine göre, artımlı olarak soldan soğa hareket etmekte ve herhangi bir düğüme ait ağaç yapısı kontrol edilmeden önce, o düğümden önceki tüm düğümlerin ağaç kontrollerinin tamamlanmış olması şartı aranmaktadır.



(a)



(b)

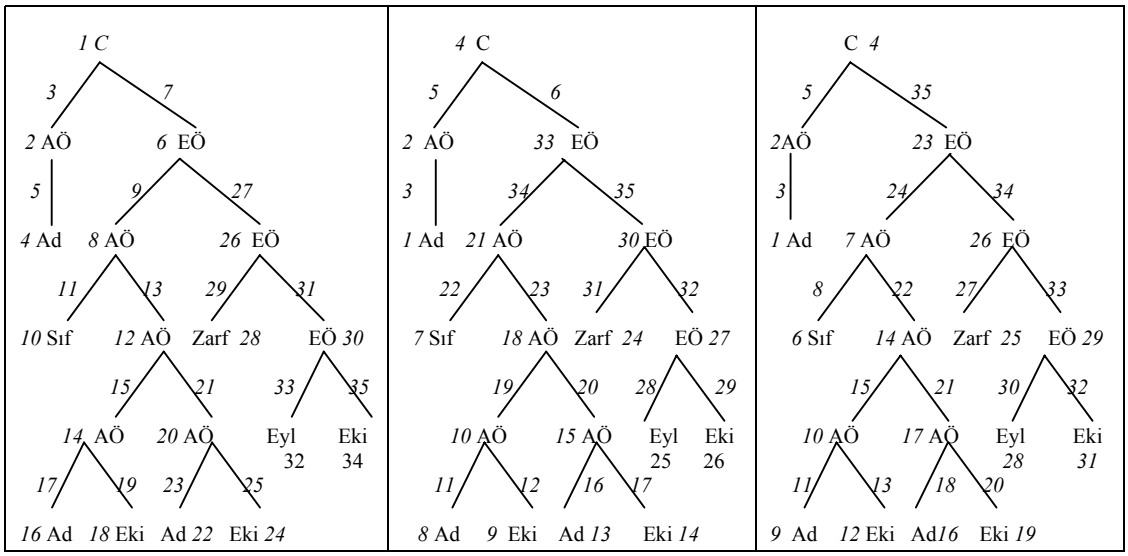
Şekil 2: *(a) Akıllı elemanı önceden ayarlasaymış" (b) "Küçük çocuğu biraz azarlamışmış" tümcelerinin sözdizimsel çözümleme sonuçları ve eylemlerin biçimbilimsel çözümlemesindeki yapısal değişimin (YD) gösterimi*

4. ETMENLENMİŞ DİLBİLGİSİ

Etmenlenmiş dilbilgisini tanımlamadan önce, incelenen tümcelerde kullanılan CFG tanımı verilmelidir. Bağlamdan bağımsız dilbilgisi CFG G=(N,∑,P,S), terminal olmayan semboller kümesi N, terminal veya terminal öğeleri sembollerini veren ∑ sonlu kümesi, S başlangıç sembolü ve $A \rightarrow \alpha$ ($A \in N$, $\alpha \in (N \cup \sum)^*$) biçimindeki kurallar dizisinden oluşur. Bu kurallar yeniden yazılabilir kurallardır. Çünkü herhangi bir kuralın sol tarafındaki terminal olmayan sembol A, kuralın sağ tarafındaki α ile (Λ- katarı da dahil olmak üzere) yer değiştirmektedir *[3]*.

Tepeden-tabana bir çözümleme algoritması için herhangi bir adımda yığının en üstünde "AÖ" olsun ve sonraki sembol de "Ad" olsun. Bu durumda kullanılan dilbilgisine ait $AÖ \rightarrow$ Ad Eki ve $AÖ \rightarrow$Ad kurallarından hangisinin

seçilebileceğine ilişkin hiç bir bilgi yoktur. Bu tür belirsizlik durumlarını yok etmek için dilbilgisinin etmenlenmesi (grammar factorization) bir çözüm şeklidir. Böylece, tepeden-tabana çözümleyici için AÖ-Ad genişlemesi tanımlanır ve bu durum, bir sonraki terminal ulamı (ön-terminal[3]) giriş katarındaki sonraki sembolde bulunduğu zaman gerçekleşir. Dilbilgisinin faktörlenmesi, her kuralın sağ tarafındaki öğelerin en sağdan gruplanması ile oluşur ve soldan etmenleme (left factorization) olarak adlandırılır.



Şekil 3: *(a) tepeden-tabana (b) tabandan-tepeye (c) sol köşeden standart çözümlemede düğüm noktalarının yerleşimleri*

Bir CFG, G=(N,∑,P,S)'nin soldan etmenlenmiş dönüşümü yine öyle bir bağlamdan-bağımsız dilbilgisidir ki, $G_1=(N_1,\sum,P_1,S)$ olmak üzere terminal olmayan küme $N_1= N \cup \{A - B:\ A\in N,\ B\in(N\cup\sum)^+\}$ şeklinde tanımlanır *[5]*. Yeni terminal olmayan sembol A–B simgelenişinde A, bir kuralın sol tarafını ifade ederken; B bu kuralın çocuk ulamlarının dizildiği kuralın sol tarafını tanımlar. Böylece: herhangi bir $A - B \Rightarrow^{*} \gamma$ cümlesel biçiminin mevcut olması $A \Rightarrow^{*} B\gamma$ cümlesel biçimine bağlıdır. Soldan etmenlenmiş dilbilgisi kuralları P_1, bağlamdan bağımsız dilbilgisi kuralları P cinsinden 4 farklı şekilde tanımlanabilir:

i) Her P: $A\rightarrow B\gamma$ kuralı için P_1: $A\rightarrow B$ $A–B$
ii) Her P: $A\rightarrow \alpha B$ kuralı için P_1: $A\text{-}\alpha \rightarrow B$ $A\text{-}\alpha B$
iii) Her P: $A\rightarrow \alpha$ kuralı için P_1: $A - \alpha\rightarrow \wedge$
iv) Her P: $A\rightarrow \wedge$ kuralı için P_1: $A \rightarrow \wedge$

*Tablo2'*de **"**çocuk değerli kitabın sayfalarını sabahleyin yırtmış" tümcesine ait CFG kuralları ve bu tümcenin etmenlenmiş dilbilgisi yapısının nasıl türetildiği görülmektedir. *Şekil 4*'te ise aynı tümceye ait çözümleme sonuçları verilmektedir.

[3] Bir terminal öğesinin ebeveynine ön-terminal adı verilir (pre-terminal). Sadece bulunduğu konumdaki terminal öğesinin ebeveynidir *[1]*. Örneğin, Penn Treebank ön-terminalleri POS (tümcenin öğeleri) bileşenleri olarak kullanır ve diğer terminal olmayan bileşenlerini hariç tutar.

5. ETMENLENMİŞ ÇÖZÜMLEME ALGORİTMASI

Etmenlenmiş Çözümleme (EÇ) Algoritmasını oluştururken terminal olmayan sembollerin etmenleme düzeylerine dikkat etmek gerekir. Bu nedenle de etmenlenmiş dilbilgisinin her adımda kontrol edilmesi önemlidir. EÇ Algoritması uygulandığı dilbilgisinin ağaç yapısını doğru olarak yansıtır.

Giriş tümcesinin biçimbilimsel analizi çalışmanın ilk kısmında tamamlandığı için, giriş katarının uzunluğu hesaplanırken herhangi bir kelimenin ek alması durumunda, bu ek(ler) ayrı birer sembol olarak değerlendirilmektedir. Bu durum isim ve eylem işlevi gören sözcükler için geçerlidir. Örneğin "çocuk değerli kitabın sayfalarını sabahleyin yırtmış" tümcesi 9 uzunlukta bir katardır ve ω_0^8 şeklinde ifade edilir. Giriş katarının uzunluğu algoritmanın dallanacağı düzey sayısını belirler. EÇ Algoritmasının genel ifadesi

$$\text{EÇ}\ [i, j, l] \leftarrow [\ KT;\ YK;\ \omega_k ..... \omega_n]$$

3-lüleri ile tanımlanır. Algoritmanın ilk değişkeni

i=1 ise, j. kısmı türetmenin sol tarafının işlemleneceği;
i=2 ise, j. kısmi türetmenin sağ tarafının işlemleneceği;
i indisinin "0i" olarak simgelenmesi halinde ise, mevcut durumun tekli (unary) alt ağacının işlemleneceği betimlenir. Algoritmanın ikinci değişkeni j, kısmi türetmenin düzeyini ifade eder. Üçüncü değişken ise, öngörme (prediction) ve eşleştirme (matching) olarak iki farklı işlev gerçekleştirmek üzere, sırası ile a ve b sembollerini işlemler.

Bu tanımlamalara göre:

EÇ[01, 4, a] bildirimi etmenlenmiş bir çözümleme ağacına göre, ebeveyn ağacın dördüncü düzeydeki birli kısmi türetme ağacını öngörme olarak işlemler. EÇ[2, 4,a] bildirimi ise, ebeveyn ağacın dördüncü düzeydeki ikinci, yani sağ taraftaki kısmi türetme ağacının öngörme olarak işlemleneceğini açıklar.

EÇ $[i, j, l] \leftarrow [KT; YK; \omega_k ....\omega_n]$ simgelenişinin sağ tarafındaki ilk değişken KT kısmi türetme kuralını, ikinci değişken YK mevcut durumda yığın içinde bulunan (terminal ve terminal olmayan) sembollerin tümünü, son değişken ise kalan giriş katarını simgeler. Sembollerin yığına atılması ve çekilmesi diğer çözümleme algoritmalarında olduğu gibi son giren ilk çıkar mantığı ile gerçekleşmektedir. Çalışmanın önceki adımında etmenlenmiş dilbilgisi ağacının çıkarımında kullanılan değişkenler, algoritmanın her bir adımının simgelenişine yardımcı olmaktadır. Bir başka ifade ile algoritmanın sağ tarafını belirlemek için, etmenlenmiş kısmi türetme ağacının oluşturulduğu bilgisayar kodunun değişkenleri ile eşleştirme yapılır. *G0* başlangıç düğümü, *G01* "0.düzeyin" sol kısmi türemesi, *G02* sağ kısmi türetmesi olmak üzere

$$\text{EÇ}[-, 0, -] \leftarrow [C^+ \rightarrow C\ \ C^+\text{-}C;\ \ C\ \ \ C^+\text{-}C;\ \ w_0^8\ ]$$

bildirimi ile başlayan EÇ Algoritması,

```
Set Treekok = SentesTree.Nodes.Add(, , "G0", "C+", root, root2)
Set Treelevel = SentesTree.Nodes.Add(Treekok, tvwChild, "G01", "C", level, level2) Set
Set Treelevel = SentesTree.Nodes.Add(Treekok, tvwChild, "G02", "C+-C",level, level2)
```

kod parçasının değişkenlerini kullanmaktadır. *G011* "1. düzeyin" sol kısmi türetmesi, *G012* de aynı düzeyin sağ kısmi türetmesi olmak üzere

$$\text{EÇ}[1,1,a] \leftarrow [C \rightarrow \text{AÖ}\ \ C\text{- AÖ};\ \ \text{AÖ}\ \ C\text{- AÖ}\ \ C^{+}\text{- C};\ \text{çocuk}\ w_0^{7}\ ]$$

bildirimi de

```
Set Treekok = SentesTree.Nodes(2)
Set Treelevel = SentesTree.Nodes.Add(Treekok, tvwChild, "G011", "AÖ", level, level2)
Set Treelevel = SentesTree.Nodes.Add(Treekok, tvwChild, "G012", "C-AÖ",level, level2)
```

kodundan yararlanarak yazılmıştır. Benzer şekilde aşağıdaki kod parçasında *G01211* değişkeni ile 3.düzeyin sol kısmi türetmesi, *G012112* değişkeni ile de 4. düzeyin sağ kısmi türetmesi simgelenmektedir ve

```
Set Treekok = SentesTree.Nodes(8)
Set Treelevel = SentesTree.Nodes.Add(Treekok, tvwChild, "G01211", "AÖ", level, level2)
Set Treelevel = SentesTree.Nodes.Add(Treekok, tvwChild, "G01212", "EÖ-AÖ",level, level2)
Set Treekok = SentesTree.Nodes(10)
Set Treelevel = SentesTree.Nodes.Add(Treekok, tvwChild, "G012111", "Sif", level, level2)
Set Treelevel = SentesTree.Nodes.Add(Treekok, tvwChild, "G012112", "AÖ-Sif", level, level2)
```

kodundan yararlanarak, ilgili algoritma ifadeleri aşağıdaki gibi oluşturulmuştur:

$\text{EÇ}[1,3,a] \leftarrow [\text{EÖ} \rightarrow \text{AÖ}\ \ \text{EÖ-AÖ};\ \ \text{AÖ}\ \ \text{EÖ-AÖ}\ \ \text{C-AÖ, EÖ}\ \ C^{+}\text{-C};\ \text{kırmızı}\ w^{6}]$

$\text{EÇ}[1,4,a] \leftarrow [\text{AÖ} \rightarrow \text{Sıf}\ \ \text{AÖ-Sıf};\ \text{Sıf}\ \ \text{AÖ- Sıf}\ \ \text{EÖ-AÖ}\ \ \text{C-AÖ, EÖ}\ \ C^{+}\text{-C};\ \text{kırmızı}\ w_0^{6}]$

$\text{EÇ}[01,5,a] \leftarrow [\text{Sıf} \rightarrow \text{kırmızı};\ \text{kırmızı}\ \ \text{AÖ-Sıf}\ \ \text{EÖ-AÖ}\ \ \text{C-AÖ, EÖ}\ \ C^{+}\text{-C};\ \text{kırmızı}\ w_0^{6}]$

$\text{EÇ}[01,5,b] \leftarrow [\ -\ ;\ \text{AÖ- Sıf}\ \ \text{EÖ- AÖ}\ \ \text{C-AÖ, EÖ}\ \ C^{+}\text{-C};\ \text{kitabın}\ w_0^{5}\ ].$

Son iki ifadeden görüldüğü gibi, tekli bir türetme ağacı tanımlandığında bunu mutlaka bir eşleştirme işlemi izlemelidir. Etmenlenmiş türetme ağacını oluşturan Visul Basic kodunun değişkenleri kullanılarak, algoritmada yığına atma ve yığından çekme işlemleri sırasında hiç bir belirsizlik problemi ile karşılaşmadan EÇ Algoritmasına devam edilebilmektedir.

6. SONUÇ

Bu çalışmada Türkçe tümcelerin sözdizimsel çözümlemesindeki belirsizliği önlemek amacı ile yeni bir dilbilgisi olarak etmenlenmiş dilbilgisi tanımlanmış ve bu dilbilgisine göre geliştirilen çözümleme algoritmasının temel yapısı modellenmiştir. Etmenlenmiş Çözümleme Algoritmasının bilgisayar ortamında işleyişinin dilbilgisi kurallarına uygun tümceler üzerinde test edilmesi işlemleri devam etmektedir. Bu algoritma, tüm etmenlenmiş CFG tanımlamalarına uygun tümcelere uygulandığında, Türkçe tümcelerin çözümlenmesindeki belirsiz durumlarını önleyecek bir çözümleme yöntemi geliştirilmiş olacaktır. Ayrıca düzeylerin dinamik olarak yaratılması algoritmanın en önemli özelliklerinden

biridir. Çalışmada kullanılan veritabanı isim, sıfat, eylem ve zarf bildiren kelimeler için Türk Dil Kurumu sözlüğünden titiz bir çalışma sonucunda elde edildiği için, tanımlanan dilbilgisine uygun ve anlamlı olan çok çeşitli tümcenin çözümlenmesi mümkün olacaktır.

Tablo 2: "*çocuk değerli kitabın sayfasını sabahleyin yırtmış" tümcesini çözümleyen CFG kuralları ve bu tümceye ait etmenlenmiş dilbilgisi yapısı*

CFG yapısı	**Etmenlenmiş Dilbilgisi Yapısı**
C → AÖ EÖ	C+ → C C+-C
AÖ → AÖ AÖ	C → AÖ C-AÖ
AÖ → Sif AÖ	AÖ → Ad AÖ-Ad
AÖ → Ad (Eki)	AÖ-Ad → /\
EÖ → Zarf EÖ	C-AÖ → EÖ C-AÖ, EÖ
EÖ → AÖ EÖ	EÖ → AÖ EÖ-AÖ
EÖ → Eyl (Eki)	AÖ →Sif AÖ-Sif
	AÖ-Sif → AÖ AÖ-Sif, AÖ
	AÖ → AÖ AÖ-AÖ
	AÖ → Ad A-Ad
	AÖ-Ad → Eki AÖ-Ad, Eki
	AÖ-Ad, Eki → /\
	AÖ-AÖ → AÖ AÖ-AÖ, AÖ
	AÖ → Ad AÖ-Ad
	AÖ-Ad → Eki AÖ-Ad, Eki
	AÖ-Ad, Eki →/\
	AÖ-AÖ, AÖ → /\
	AÖ-Sif, AÖ → /\
	EÖ-AÖ → EÖ EÖ-AÖ, EÖ
	EÖ → Zarf EÖ-Zarf
	EÖ-Zarf → EÖ EÖ-Zarf, EÖ
	EÖ → Eyl EÖ-Eyl
	EÖ-Eyl → Eki EÖ-Eyl, Eki
	EÖ-Eyl, Eki → /\
	EÖ-Zarf, EÖ → /\
	EÖ-AÖ, EÖ → /\
	C-AÖ, EÖ → /\
	C+-C → </s>

Tablo 1'de örnek olarak alınan ve çözümlenecek olan tümceye ait kelimeler "isim", "sıfat","zarf" ve "eylem" görevi gören tüm sözcükler için oluşturulmuş veri tabanından alınmaktadır. Bu veri tabanı her dilbilgisi türü için Türk Dil Kurumu sözlüğünden alfabetik sırada düzenlenmiştir. Ayrıca, "AÖ" ve "EÖ" sembollerine (terminal olmayan sembollere) ait "Eki" terminal ulamları biçimbilimsel analiz sonucunda elde edilmektedir. Aşağıda bu kelimelerin, kullandığımız CFG kurallarına ilave edilmiş şekli görülmektedir:

Ad → Çocuk
Sif →değerli
Ad → kitap
Eki → ın
Ad → sayfa
Eki → sını
Zarf → sabahleyin
Eyl → yırt
Eki → mış

Teşekkür: Çalışmanın uygulama kısmı Yüksek Lisans öğrencim Mustafa Çelikpençe'nin katkıları ile geliştirilmiştir.

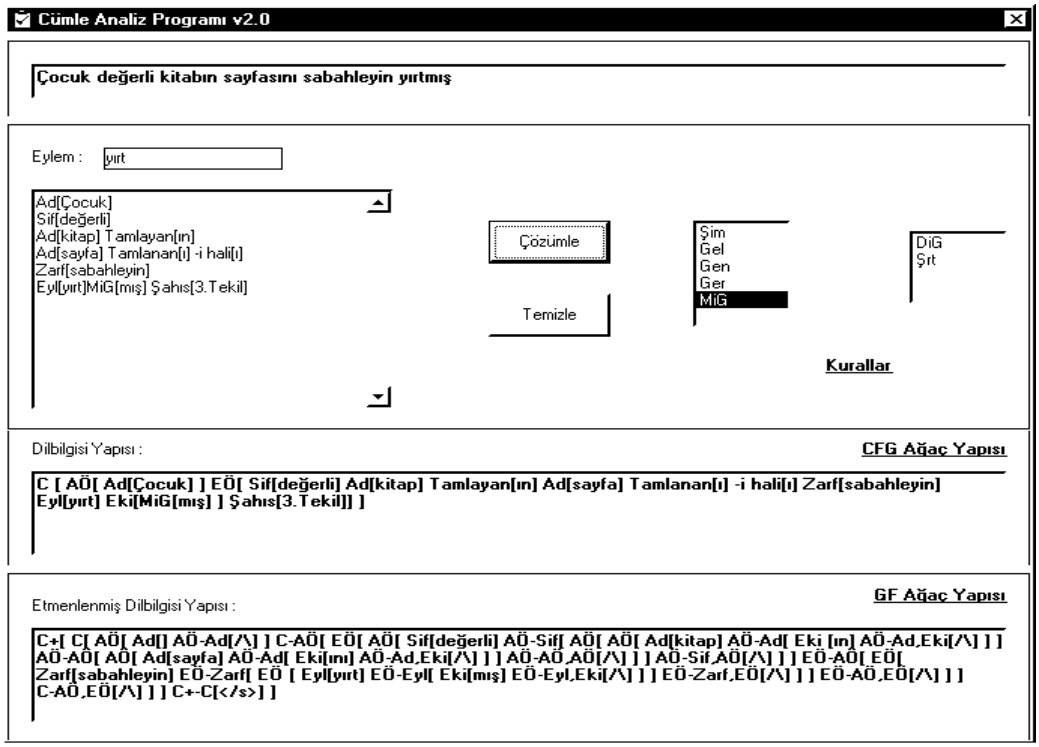
REFERENCES

[1] Aho A.V., Sethi R., Ullman J.D., 1986. "Compilers, Principles, Techniques, and Tools", Addison Wesley, Mass.

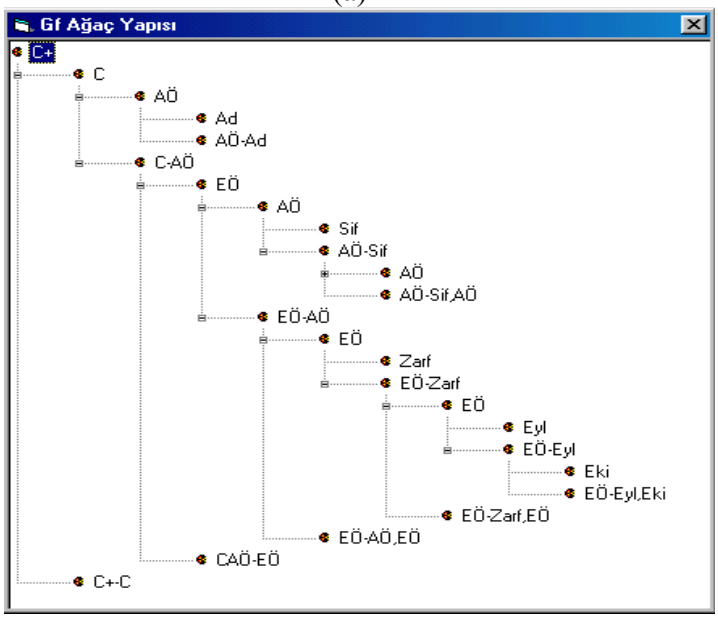
[2] Altan Z., 2001. "Visual Basic Application of the Earley Algorithm", Second International Conference on Electrical and Electronics Engineering (Electric-Control) , pp. 372-375, Bursa.

[3] Altan Z., 2001. "Formal Diller ve Soyut Makineler Ders Kitabı", İ.Ü. Yayın.

[4] Krulee G.K.,1991. “Computer Processing of Natural Language”,Prentice Hall.
[5] Roark B., 2001. “Probabilistic Top-Down Parsing and Language Modeling”, Association for Computational Linguistics, Vol. 27, Num.2, pp. 249-277.
[6]Vardar B. , 1998. “Dilbilimin Temel Kavram ve İlkeleri”,”Multilingual.



(a)



(b)

Şekil 4*(a) “çocuk değerli kitabın sayfasını sabahleyin yırtmış” tümcesine ait sözdizimsel çözümleme sonuçları (b) aynı cümlenin etmenlenmiş dilbilgisine göre türetilmiş ağaç yapısı*