

# Altsözcüksel Birimlerle Türkçe için Sözcüksel İşlevsel Gramer Geliştirilmesi

Özlem Çetinoğlu ve Kemal Oflazer

Mühendislik ve Doğa Bilimleri Fakültesi  
Sabancı Üniversitesi  
İstanbul, 34956, Türkiye

ozlemc@su.sabanciuniv.edu, oflazer@sabanciuniv.edu

**Özetçe.** Bu bildiri Türkçe'nin karmaşık biçimbilimsel yapısı ve zengin türetme olaylarını ele alırken bir çözüm olarak altsözcüksel birimler kullanmayı incelemekte ve önerilen yaklaşımı Pargram projesi dahilinde gerçekleştirilmekte olan Türkçe Sözcüksel İşlevsel Gramer üzerinden anlatmaktadır. İzlediğimiz yaklaşım sayesinde kurallar daha düzenli ve özlü bir şekilde yazılabilmekte, böylece hem genelleme imkanı arttığı için daha az sayıda olan hem de içerik olarak karmaşık olmayan kurallarla gramer kapsamı genişletilebilmektedir. Üstelik türetmelerin sözcüklere anlambilimsel katkıları programın çalışması sırasında yaratılan PRED değerleri sayesinde sistematik bir biçimde ifade edilebilmektedir. Çalışmamız altsözcüksel birimlerin basit yapım ekleri ile kullanımına yer vermekte daha sonra ettiren yapılar gibi görece daha karmaşık dil olaylarına değinmektedir. Öncelikli amacımız kullandığımız yaklaşımı mümkün olduğunca birbirinden farklı dilbilimsel alanlarda incelemek olduğu için bu bildiriye sayısal bir değerlendirmeye yer verilmemiştir.

## 1 Giriş

Bu bildiriye Pargram projesi<sup>1</sup> dahilinde gerçekleştirilen, Türkçe için geniş kapsamlı Sözcüksel İşlevsel Gramer (Lexical Functional Grammar - LFG) projesinin ana hatlarını sunmayı amaçlıyoruz. Türkçe'nin karmaşık biçimbilimsel yapısı ile sözdizimsel yapısını birleştirebilmek, özellikle de yapım ekleriyle değişen anlambilimsel gösterimleri doğru ve etkin bir biçimde gösterebilmek için *çekimli gruplar* kullanıyoruz.

Çekimli gruplar, kısaca ÇG,  $\hat{DB}$  yani türetme sınırı ile birbirinden ayrılan ve sadece çekimsel özniteliklerden oluşan biçimbirimsel alt-gösterimler olarak tanımlanır. Genel olarak bir ÇG bir biçimbirimden büyük, ama bir sözcükten de küçüktür (sözcüğün hiç yapım eki almadığı ve ÇG'nin sözcüğün tamamına denk geldiği durum hariç). Sözcükler arasındaki sözdizimsel ilişkileri belirleyen de aslen ÇG'ler olmaktadır. Eğer Türkçe için kural yazımında en küçük birim olarak sözcükleri temel alan bir gramer yazılırsa, özellikle türetilmiş sözcüklerde iç içe yapılarıdaki bilgiye ulaşmayı gerektireceği için gramer yazarının işini zorlaştıracaktır. Diğer yandan, temel birim olarak biçimbirimler alınırsa, bu sefer de çekim eklerini de kurallar içine almak gerekeceği için gramerin gereksiz yere şişmesi söz konusudur. ÇG'lerin kullanımı ise bütün sözcüklerin standart bir gösterime sahip olmasını ve karmaşık sözcük yapılarında sözcüğe daha kolay ulaşmayı sağlar.

ÇG'lerin sözcüklerde türetme süreçlerini belirlemesi yapım ekleriyle karşılaşıldığında bunu anlambilimsel gösterime yansıtacak bir mekanizma ihtiyacı doğurur. İngilizce

<sup>1</sup> <http://www2.parc.com/ist1/groups/nlft/pargram/>

için geliştirilen sözcüksel işlevsel gramerde türemiş sözcükler, örneğin *happy* (mutlu) ve *happiness* (mutluluk) sözlüğün bir parçasıdır. Öte yandan *easy* (kolay) ve *easier* (daha kolay) gibi sıfatların sözcüksel gösterimleri arasındaki ilişki *beautiful* (güzel) ve *more beautiful* (daha güzel) sıfatları arasındaki ilişki ile aynı olduğu için *easier* sözcüğü her ne kadar türemiş bir sözcük olsa da Örnek 1’de verilen işlevsel yapı gösterimi *more beautiful* işlevsel yapısına paraleldir.

$$(1) \left[ \begin{array}{l} \text{PRED} \quad \text{'easy'} \\ \text{ADJUNCT} \quad \left[ \text{PRED} \quad \text{'more'} \right] \\ \text{DEG-DIM} \quad \text{pos} \\ \text{DEGREE} \quad \text{comparative} \end{array} \right]$$

Türetmelerin sözlükte gösterimi, görece daha az türetme özelliklerine sahip dillerde uygulanabilir olsa da Türkçe gibi eklemeli dillerde bütün türemiş formların sözlükbirim olarak gramer sözlüğüne girilmesi kesinlikle olası değildir. Dolayısıyla yapıım ekleriyle oluşturulmuş sözcüklerin hem anlambilimsel gösterim açısından belli kurallar ile gösterilmesi hem de bu sürecin algoritmik bir yapıya oturtulması gerekmektedir.

Sözcüksel İşlevsel Gramer’de [1], sözdizimi iki düzeyde temsil edilmektedir: bileşen yapısı bağlamdan bağımsız öbek yapısı ağacı şeklindedir, işlevsel yapı ise öznitelik-değer çiftlerinin oluşturduğu bir küme olarak tanımlanabilir. Öznitelikler zaman, kişi gibi özellikler olabileceği gibi, özne, nesne gibi işlevler de olabilirler. Bir özniteliğin değeri bir başka öznitelik olabilir, yani iç içe karmaşık yapılar mümkündür. Bileşen yapısında sözdizimsel gösterim işlevsel yapıda ise anlambilimsel gösterim ön plana çıkar bu yüzden bir öbeğin bileşen yapıları her dil için farklılık gösterirken işlevsel yapılarının birbirine benzemesi beklenir.

Bildirinin geri kalanı şu şekilde düzenlenmiştir: Bölüm 2, Türkçe için daha önce yapılmış gramer çalışmalarından ve bildiride bahsedilen çalışmanın benzerlerinden söz etmektedir. Bölüm 3, ÇG’leri kullanmamızın sebeplerini belirtirken Bölüm 4 bu yapıtaşlarının LFG içindeki kullanımını örneklerle açıklamaktadır. Bölüm 5 sistemimizin genel mimarisini ve şu anki durumunu açıklamakta, Bölüm 6 ise sonuçları vermektedir.

## 2 İlgili Çalışmalar

Güngördü ve Oflazer [2] Türkçe için LFG kuramını kullanan oldukça geniş kapsamlı bir gramer çalışması yapmışlardır. Ancak bu çalışma LFG’nin temellerinden birleşme kavramı yerine sözde birleşme kavramı ile gerçekleşmiştir. Daha önemlisi, sözcüklerin gösteriminde daha standart bir yol izlenmiş ve türemiş olsun olmasın her sözcük bir bütün olarak ele alınmış, bu da özellikle birden çok türetme uygulanan sözcüklerde ilgili bilginin iç içe geçmiş yapılarda tutulması sonucu bağımlılık ilişkilerini kurmak için yazılan kuralların karmaşık ve hantal olmasına yol açmıştır.

Türkçe için sözcüklerden daha küçük bir birim kullanılarak gramer geliştirilmesi ile ilgili Bozşahin’in bir çalışması [3] bulunmaktadır. Bu çalışma prototip bir uygulamada çeşitli dilbilim olaylarını temel birim olarak biçimbirim kullanarak Birleşenli Ulamsal Gramer (Combinatory Categorical Grammar - CCG) çatısında gerçeklemiştir. Sistem, biçimdizimlerin işlenmesini de tümce için geliştirilen gramer seviyesinde yapmakta, dilin iki seviyesini tek bir düzlemde çözümlenmektedir. Oflazer’in [4] geliştirdiği bağımlılık grameri ise ÇG’leri bağımlılık ilişkilerinin kurulduğu yapıtaşları olarak kullanmıştır. Çakıcı [5] ise Türkçe ağaç yapılı derlemdeki ÇG’lere dayanan ilişki gösterimlerini temel alarak CCG gramer için otomatik bir şekilde sözlük üreten bir çalışma yapmıştır.

Bizim bu bildiride kullandığımız yönteme benzer şekilde Butt ve King [6] de Urdu dilindeki biçimbirimsel ettirgen yapılar için biçimbirimi de kurallar seviyesine çıkararak bir yöntem seçmişlerdir. Bizim yöntemimizden farklı olarak söz konusu olan bir biçimbirim öbeği değil, tek bir biçimbirimdir. Ayrıca ortaya çıkan ağaç yapısı ve anlambilimsel gösterim de bizim yaklaşımımızla farklılıklar göstermektedir.

### 3 Alt Birim Olarak Çekimli Gruplar

Türkçe, bir dizi yapım ve çekim ekinin kökün ardından belli bir sıraya uyarak eklendiği sonradan eklemeli bir dildir [7]. Sözdizim seviyesinde ise her ne kadar genel sıralama özne-nesne-yüklem şeklinde ise de tüm ögeler tümce içinde farklı yerlerde olabilirler. Türkçe'nin biçimdizimi ise oldukça karmaşıktır. Bir sözcük birden çok türetme içerebilir ve isim ya da fiil kökünden türetililecek sözcük sayısı kuramsal olarak sonsuzdur. Tipik bir Türkçe metinde ortalama biçimbirim sayısı kök de dahil edilirse 3-4 iken sözcük başına ortalama 1.23 türetme düşmektedir. İlgeç, belirteç gibi ek almayan sözcükler çıkartıldığında ise bu rakam 2'ye yaklaşmaktadır.

Bir sözcüğün biçimbirimsel çözümlemesi her biçimbirime karşılık bir etiket gelecek şekilde oluşturulmuş dizilerle ifade edilebilir. Biçimbirimsel çözümleyicimizde  $\hat{DB}$  etiketi bizim çekimli grupları belirlemede kullandığımız türetme sınırlarını göstermektedir. Eğer Türkçe'deki biçimbirimsel bilgiyi

$$kök + \hat{CG}_1 + \hat{DB} + \hat{CG}_2 + \hat{DB} + \dots + \hat{DB} + \hat{CG}_n$$

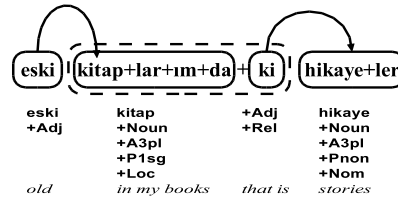
şeklinde gösterirsek, her  $\hat{CG}_i$  bir çekim ekinin etiketlerinden oluşan bir diziye karşılık gelmektedir. Örneğin aşağıda belirtilen biçimbirimsel çözümleyici çıktısına sahip *gerçekleştirilebilir* sözcüğü için altı  $\hat{CG}$  bulunmaktadır. Bu  $\hat{CG}$ 'ler aşağıdaki şekildedir:

gerçek+Adj  $\hat{DB}$ +Verb+Become  $\hat{DB}$ +Verb+Caus  $\hat{DB}$ +Verb+Pass+Pos  
symbol94DB+Verb+Able+Aor  $\hat{DB}$ +Adj+Zero

- |                 |                   |
|-----------------|-------------------|
| 1. +Adj         | 4. +Verb+Pass+Pos |
| 2. +Verb+Become | 5. +Verb+Able+Aor |
| 3. +Verb+Caus   | 6. +Adj+Zero      |

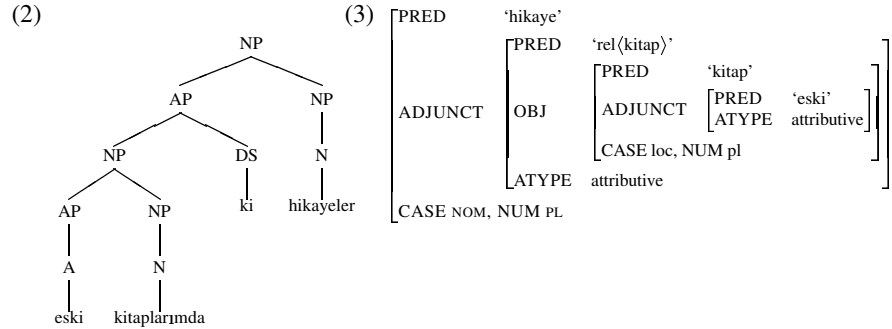
Sözcüklerin  $\hat{CG}$ 'ler ile tanımlanması sözdizimsel ilişkilerin doğru bir şekilde gösterimini oldukça kolaylaştırmıştır. Bağımlı sözcükten baş sözcüğe doğru olan işlevsel bağımlılık aslında bağımlı sözcüğün son  $\hat{CG}$ 'sinden baş sözcüğün bir  $\hat{CG}$ 'sine bir bağlantıdır.

*Eski kitaplarımdaki hikayeler* isim öbeğini bir örnek olarak verirsek, bu öbeğin işlevsel bağımlılıkları Şekil 1'deki gibi gösterilebilir.



Şekil 1. *Eski kitaplarımdaki hikayeler* isim öbeğinin işlevsel bağımlılıkları

Küçük dikdörtgenler ÇG'leri, noktalı dikdörtgen ise sözcük sınıırı göstermektedir. Görüldüğü gibi, her sözcüğün son ÇG'si, kendinden sonra gelen bir sözcüğün ÇG'lerinden birine bağlıdır. Aynı isim öbeğinin bileşen yapısı Örnek 2'de, işlevsel yapısı Örnek 3'te gösterilmiştir. Bileşen yapısı gösteriminde her ÇG bir düğüme karşılık gelmektedir, dolayısıyla gramer içindeki kurallar ÇG'leri temel olarak yazılmıştır. Eğer bir ÇG kök biçimbirimi içeriyorsa, ağaçta o ÇG'ye karşılık gelen düğüm kökün sözdizimsel türü ile adlandırılır. Diğer ÇG'ler ise içeriklerinden bağımsız olarak DS şeklinde bir düğümler gösterilirler



İşlevsel yapı oluşturulmasında yapım eklerinin anlambilimsel gösterimlerinin büyük rolü vardır. Yüzey gösteriminde bir biçimbirime karşılık gelsin gelmesin hemen hemen tüm yapım ekleri, işlevsel yapı içinde bir nesne (OBJ) özelliğine sahip olurlar ve bu özellik türetilen sözcüğün asıl kökünü temsil eden işlevsel yapı ile birleşir. Yapım ekinin anlambilimsel gösterimi ise PRED özelliği içinde yer alır ve OBJ özelliği ile ilgili bilgi de içerdiği için programın çalışması sırasında oluşturulur. Öte yandan PRED özelliği her ne kadar yapım ekinin anlamsal gösterimini içerse de, bu gösterim oluşturulan sözcüğün gerçek anlamından sapmalar gösterebilir. Bunun sebebi bazı yapım eklerinin birden çok anlama gelmesidir, bu tür yapım ekleri için gerekli olan anlamsal ayrıştırma bizim çalışmamızda göz önüne alınmamıştır.

Bileşen yapısındaki her bir düğüm ayrı bir işlevsel yapıya karşılık gelmektedir. Böylece işlevsel bağımlılık ile ilgili kurallar yazılırken sadece ÇG'lere ulaşmak yeterlidir. Örneğin *eski kitaplarımdaki* öbeğini çözümlmek için önce *eski kitaplarımda* öbeğini çözümlen bir kural kullanılmış ve bir isim öbeği olduğu görülmüş, bir isim öbeğine *-ki* eki ekleyerek sıfat öbeği elde edilebileceğine dair ikinci kural kullanılmıştır. Alternatif yöntemde sözcüklerin bir bütün olarak işlevsel yapıda temsil edildiğini düşünelim. Böyle bir durumda, "sıfatların önlerine sıfat belirteni almaları ancak sıfat isim öbeğinden türediyse mümkündür" şeklindeki kural gerçekleşirken "türemiş sıfatın kökü bir isim midir" şeklinde bir kontrol gereklidir. *Kitaplarımdaki* sözcüğü işlevsel yapıda iki düzeyli temsil edilecek ve içteki düzeyde *kitap* kökü bulunacaktır dolayısıyla kuralın içteki düzeye ulaşması gerekir. Tek yapım ekine sahip bir sözcükte bu yaklaşım çok sorun çıkarmasa da örneğin *elbisesindeki* sözcüğü gibi birden çok yapım ekine sahip bir sözcük söz konusu olursa *elbise* kökü içten dördüncü düzeyde bulunacaktır. Bu da farklı yapıda fakat benzer işlevli sözcükler için yazılması gereken kuralların çok daha karmaşık olması anlamına gelir. Üstelik sözcükler bütün olarak ele alındığında *kitaplarımdaki* sözcüğü bir sıfattır ve *eski kitaplarımdaki* tamlamasının bileşen yapısını çizmeye çalışırsak iki sıfatı bir araya getirerek bir sıfat tamlaması elde ederiz ki bu asıl anlatmak istediğimiz ilişkiyi yanlış şekilde aktarmak olur.

## 4 Çekimli Grupların Gramerde Kullanımı

Çekimli grupları kullanma amacımızı ve bileşen yapısı ve işlevsel yapı oluştururken nasıl kullandıklarımızı bir önceki bölümde açıklamaya çalıştık. Bu bölümde ise bu yaklaşımı kullanarak çeşitli dilbilim olaylarını nasıl gerçekleştirdiğimizi anlatmak istiyoruz.

### 4.1 İsim-fiiller ve sıfat-fiiller

Türkçe’de fiilimsiler, fiillerin yapım ekleri olarak isim soylu hale getirilmeleri ile oluşturulur. Fiil aldığı yapım ekine göre isim, sıfat ya da zarf olarak kullanılabilir. Örnek 4’te verilen basit tümce bundan sonraki örneklerin temelini oluşturacaktır.

(4) Kız adamı aradı.

Örnek 5’te basit tümcenin *söyledi* fiilinin nesnesi olacak şekilde bir isim-fiil haline dönüştürüldüğünü görüyoruz. Fiilimsiyi oluşturan bütün sözcükler, fiil sonundaki yapım ekine kadar normal bir tümce gibi çözümlenmiştir. Yapım ekinin içinde bulunduğu ÇG ise, hal eki, kişi eki gibi özelliklerin işlevsel yapıya eklenmesini sağlamış, böylece türetilmiş isim öbeği, basit bir isimden oluşan bir isim öbeğinden farksız şekilde ana tümcenin yüklemi olan *söyledi* fiilinin nesnesi (OBJ) görevini üstlenebilmiştir (Örnek 6).

(5) Manav kızın adamı aradığını söyledi.

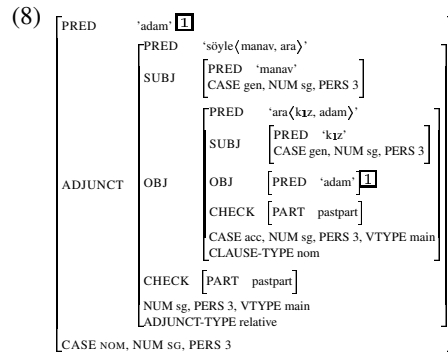
(6)

PRED	'söyle<manav, ara>'
SUBJ	[PRED 'manav' CASE nom, NUM sg, PERS 3]
	[PRED 'ara<kız, adam>'
	SUBJ [PRED 'kız' CASE gen, NUM sg, PERS 3]
OBJ	OBJ [PRED 'adam' CASE acc, NUM sg, PERS 3]
	CHECK [PART pastpart]
	CASE acc, NUM sg, PERS 3, VTYPE main [CLAUSE-TYPE nom]
TNS-ASP	[TENSE past]
	NUM SG, PERS 3, VTYPE MAIN]

Sıfat-fiiller de bu mekanizmaya benzer şekilde çözümlenmişlerdir. Türkçe’de sıfat-fiiller, isimleri niteleyen boşluklu tümceler şeklinde yer alırlar. Kaplan ve Zaenen [8] uzun mesafeli bağımlılık ilişkileri için genel bir yaklaşım önermişler, yaptıkları genişletmelerle LFG notasyonunda kısıt kuralları yazılırken sadece basit özellikler değil düzenli ifadeler yazmaya da olanak sağlamışlardır. Böylelikle sonsuz sayıda ayırtımlı sonlu bir ifade ile yazmak mümkün olmuştur. Notasyondaki bu kolaylık, boşluğa sahip tümleçlerin boşluklarını dolduran sözcüklerin işlevsel yapıda hangi derinlikte olduğuna bakılmaksızın genel birleştirme kuralları yazılmasını sağlamıştır. Örnek 7’nin de gösterdiği tümcede, fiilimsiyi oluşturan tümce, *ara-* fiili bir nesne beklemesine rağmen boşluklu bir şekilde incelenmiş ve *-dığı* eki ile karşılaşılnca tümcenin işlevsel yapısı bir fiilimsiyeye dönüştürülmüştür. Daha sonra ise işlevsel belirsizlik kuralı uyarınca fiilimsideki boşluk, fiilimsinin nitelediği isim ile birleştirilmiştir. Örnek 8 tüm tümcenin işlevsel yapısını göstermekte ve numaralandırılmış kutular birbiriyle birleştirilen alt işlevsel yapıları işaretlemektedir.

(7) Manavın kızın [ ]<sub>i</sub> aradığını söylediği adam<sub>i</sub>

Örnek 7, Örnek 5 tümcesini bir sıfat-fiil olarak içermektedir, dolayısıyla iki tümcenin de işlevsel yapılarına daha yakından bakarsak ne kadar benzediklerini görebiliriz. Örnek 8'deki niteleyici (ADJUNCT) Örnek 6'nın bütün işlevsel yapısıyla neredeyse aynıdır ve TNS-ASP ve ADJUNCT-TYPE özelliklerinde farklılık göstermektedir. Kurallar düzeyinde duruma bakarsak, filimsi de tam tümce de aynı çekirdek tümce kuralı ile incelenmekte, daha sonra tümcenin tam bir tümce olup olmadığına sonlu fiil koşulu denetlenerek karar verilmektedir. Bu koşul Örnek 5'te TENSE özelliğinin varlığı ile sağlandığı için çekirdek tümce tam bir tümcedir. Aslında sıfat fiil de *-dı ğı* ekinin işlevsel yapıya kattığı PART özelliğinin 'pastpart' değeriyle, geçmiş zamana ait bir olayı belirten zaman bilgisi içermektedir.



#### 4.2 Ettirgen Fiiller

Türkçe fiil biçimimizi fiiller için birden çok ettirgen çatıyı yapım ekleriyle ifade edebilme olanağı sağlar. Her ne kadar fiillerin alabileceği ettirgen türetmeler kuramsal olarak sınırlı değilse de, pratikte 2-3 türetmeden fazlası kullanılmaz. Bu tür türetmeler de biçimbirimsel çözümleme sırasında türetme sınırlarıyla ayrılan etiket dizileri ile gösterildikleri için ÇG'ler kullanılarak kurallar yazılmasına uygundur. Örneğin, *aradı* sözcüğünün biçimbirimsel çözümlemesi (ara+Verb+Pos+Past+A3sg) iken, fiilin ettirgen hali *arattı* için çözümleme (ara+Verb^DB+Verb+Caus+Pos+Past+A3sg) şeklindedir.

Örnek 9 basit bir tümceyi ve tümcenin ettirgen halini örneklerken, bu tümcelemlerin işlevsel yapıları sırasıyla Örnek 10 ve 11'de verilmiştir. Ettirgen hale getirilmiş bir fiili çözümlemek, bir öncekine oranla daha büyük olan ve PRED özelliği özne (SUBJ), nesne (OBJ) ve tümleç (XCOMPLEMENT) bekleyen bir işlevsel yapı yaratmak anlamına gelmektedir. İlk tümcenin işlevsel yapısının tamamı ikinci tümcenin tümleci konumdadır. Ettirgen fiilin nesnesi ilk fiili gösteren işlevsel yapının içindeki özne ile birleştirilmiştir. Eğer ilk fiil geçişli ise, bu fiilin nesnesi, ettirgen fiilde ikinci nesne (OBJTH) görevini alır. Örnek tümcelerde açıkça görülmesi de önemli olan tümcede nesne ve önceki özneyi doğru hal ekini denetleyerek bulmaktır, çünkü serbest sözcük sırası kuralı gereği tümcedeki öbeklerin görevlerini hal ekleri belirtmektedir.

- (9) a. Kız adamı aradı.  
b. Manav kıza adamı arattı.

(10)	<table border="0"> <tr> <td style="padding-right: 10px;">[</td> <td style="padding-right: 10px;">PRE</td> <td style="padding-right: 10px;">‘ara(kız, adam)’</td> <td style="padding-right: 10px;">]</td> </tr> <tr> <td style="padding-right: 10px;">SUBJ</td> <td style="padding-right: 10px;">[</td> <td style="padding-right: 10px;">PRE ‘kız’</td> <td style="padding-right: 10px;">]</td> </tr> <tr> <td></td> <td></td> <td style="padding-right: 10px;">CASE nom</td> <td></td> </tr> <tr> <td></td> <td></td> <td style="padding-right: 10px;">NUM sg, PERS 3</td> <td></td> </tr> <tr> <td style="padding-right: 10px;">OBJ</td> <td style="padding-right: 10px;">[</td> <td style="padding-right: 10px;">PRE ‘adam’</td> <td style="padding-right: 10px;">]</td> </tr> <tr> <td></td> <td></td> <td style="padding-right: 10px;">CASE acc</td> <td></td> </tr> <tr> <td></td> <td></td> <td style="padding-right: 10px;">NUM sg, PERS 3</td> <td></td> </tr> <tr> <td style="padding-right: 10px;">TNS-ASP</td> <td style="padding-right: 10px;">[</td> <td style="padding-right: 10px;">TENSE past</td> <td style="padding-right: 10px;">]</td> </tr> <tr> <td></td> <td></td> <td style="padding-right: 10px;">NUM sg, PERS 3, VTYPE MAIN</td> <td></td> </tr> </table>	[	PRE	‘ara(kız, adam)’	]	SUBJ	[	PRE ‘kız’	]			CASE nom				NUM sg, PERS 3		OBJ	[	PRE ‘adam’	]			CASE acc				NUM sg, PERS 3		TNS-ASP	[	TENSE past	]			NUM sg, PERS 3, VTYPE MAIN		(11)	<table border="0"> <tr> <td style="padding-right: 10px;">[</td> <td style="padding-right: 10px;">PRE</td> <td style="padding-right: 10px;">‘caus(manav, kız, adam, ara(kız, adam))’</td> <td style="padding-right: 10px;">]</td> </tr> <tr> <td style="padding-right: 10px;">SUBJ</td> <td style="padding-right: 10px;">[</td> <td style="padding-right: 10px;">PRE ‘manav’</td> <td style="padding-right: 10px;">]</td> </tr> <tr> <td style="padding-right: 10px;">OBJ</td> <td style="padding-right: 10px;">[</td> <td style="padding-right: 10px;">PRE ‘kız’</td> <td style="padding-right: 10px;">1]</td> </tr> <tr> <td style="padding-right: 10px;">OBJTH</td> <td style="padding-right: 10px;">[</td> <td style="padding-right: 10px;">PRE ‘adam’</td> <td style="padding-right: 10px;">2]</td> </tr> <tr> <td style="padding-right: 10px;">XCOMP</td> <td style="padding-right: 10px;">[</td> <td style="padding-right: 10px;">PRE ‘ara(kız, adam)’</td> <td style="padding-right: 10px;">]</td> </tr> <tr> <td></td> <td style="padding-right: 10px;">SUBJ</td> <td style="padding-right: 10px;">[</td> <td style="padding-right: 10px;">PRE ‘kız’</td> <td style="padding-right: 10px;">]</td> </tr> <tr> <td></td> <td></td> <td style="padding-right: 10px;">CASE dat, NUM sg, PERS 3</td> <td style="padding-right: 10px;">1]</td> </tr> <tr> <td></td> <td style="padding-right: 10px;">OBJ</td> <td style="padding-right: 10px;">[</td> <td style="padding-right: 10px;">PRE ‘adam’</td> <td style="padding-right: 10px;">]</td> </tr> <tr> <td></td> <td></td> <td style="padding-right: 10px;">CASE acc, NUM sg, PERS 3</td> <td style="padding-right: 10px;">2]</td> </tr> <tr> <td></td> <td></td> <td style="padding-right: 10px;">VTYPE main</td> <td></td> </tr> <tr> <td style="padding-right: 10px;">TNS-ASP</td> <td style="padding-right: 10px;">[</td> <td style="padding-right: 10px;">TENSE past</td> <td style="padding-right: 10px;">]</td> </tr> <tr> <td></td> <td></td> <td style="padding-right: 10px;">NUM sg, PERS 3, VTYPE MAIN</td> <td></td> </tr> </table>	[	PRE	‘caus(manav, kız, adam, ara(kız, adam))’	]	SUBJ	[	PRE ‘manav’	]	OBJ	[	PRE ‘kız’	1]	OBJTH	[	PRE ‘adam’	2]	XCOMP	[	PRE ‘ara(kız, adam)’	]		SUBJ	[	PRE ‘kız’	]			CASE dat, NUM sg, PERS 3	1]		OBJ	[	PRE ‘adam’	]			CASE acc, NUM sg, PERS 3	2]			VTYPE main		TNS-ASP	[	TENSE past	]			NUM sg, PERS 3, VTYPE MAIN	
[	PRE	‘ara(kız, adam)’	]																																																																																						
SUBJ	[	PRE ‘kız’	]																																																																																						
		CASE nom																																																																																							
		NUM sg, PERS 3																																																																																							
OBJ	[	PRE ‘adam’	]																																																																																						
		CASE acc																																																																																							
		NUM sg, PERS 3																																																																																							
TNS-ASP	[	TENSE past	]																																																																																						
		NUM sg, PERS 3, VTYPE MAIN																																																																																							
[	PRE	‘caus(manav, kız, adam, ara(kız, adam))’	]																																																																																						
SUBJ	[	PRE ‘manav’	]																																																																																						
OBJ	[	PRE ‘kız’	1]																																																																																						
OBJTH	[	PRE ‘adam’	2]																																																																																						
XCOMP	[	PRE ‘ara(kız, adam)’	]																																																																																						
	SUBJ	[	PRE ‘kız’	]																																																																																					
		CASE dat, NUM sg, PERS 3	1]																																																																																						
	OBJ	[	PRE ‘adam’	]																																																																																					
		CASE acc, NUM sg, PERS 3	2]																																																																																						
		VTYPE main																																																																																							
TNS-ASP	[	TENSE past	]																																																																																						
		NUM sg, PERS 3, VTYPE MAIN																																																																																							

## 5 Gramerin Güncel Durumu

Türkçe LFG grameri XLE (Xerox Linguistic Environment) [9] kullanılarak gerçekleştirilmektedir. XLE geniş ölçekli gramerler yazmayı kolaylaştırmak amacıyla geliştirilmiştir ve öncözümleyici, biçimbirimsel çözümleyici, işaretleyici gibi araçların bağımsız bir şekilde geliştirilip bütünleştirilmesine olanak sağlamaktadır. Bizim gramerimizde de biçimbirimsel çözümleme ve bileşik sözcükler için sonlu durumlu deęiştiriciler sisteme eklenmiştir. Sonlu durumlu birimler sözcüğün bütün olası biçimbirimsel çözümlerini vermekte, anlambilimsel işaretleme ise köklerin yer aldığı sözlük bölümünde yapılmaktadır.

Şu ana kadar geliştirilen gramer, serbest sözcük sırasından, uzun mesafeli ilişkilere birçok önemli dil olayını gerçeklemiştir ve özellikle isim öbekleri konusunda oldukça kapsamlı bir alt gramere sahiptir.

ÇG’lerle türetme yapan kurallar haricindeki diğer kurallar, mesela isim tamlaması kuralı ÇG’lerden bağımsızdır. Yani sözcükler ister basit ister türemiş olsun aynı kural içinde ele alınırlar. Yine de Türkçe’nin bazı ilginç dil olaylarını ele almada ÇG kullanmak yetersiz kalmaktadır. Örneğin, bağlaç kullanılan eşlemlerde en son sözcük grubunun çekim eklerinin tüm grup üzerinde etkisi vardır. *Ertelenmiş ekleme* [10] adı verilen bu olayı Örnek 12a ve 12b ile gösterebiliriz. Görüldüğü gibi *kadın* sözcüğünün sonundaki hal eki aslında *adam* sözcüğünün de hal ekidir.

- (12) a. Kız adam ve kadını aradı.  
b. Kız [adam ve kadın]-ı aradı.

Ertelenmiş ekleme, sözcükleri ÇG’lerden bölmek şeklindeki yaklaşımla ters düşen bir durumdur çünkü ismin hal eki çekim eki olmasına rağmen, yapım ekleri gibi sözcüğü değil tüm grubu etkilemektedir.

## 6 Sonuçlar ve İleri Konular

Bu bildiri Türkçe için geliştirilmekte olan geniş kapsamlı LFG gramerinin, çekimli gruplar olarak adlandırdığımız altsözcüksel birimleri temel alarak gerçekleştirilmesini anlatmaktadır. Kurallarda temel birim olarak ÇG’lerin alınmasıyla hem anlambilimsel

olarak daha doğru bir gösterim söz konusu olmuş hem de Türkçe'nin üretken türetme biçimbirimi düzenli bir şekilde sözdizim düzeyine aktarılmıştır

Öncelikli hedefimiz halen yazmakta olduğumuz grameri mümkün olduğu kadar genişletmek ve bunu yaparken sözlüğü de gramerin gerektirdiği şekilde etiketlemektir. Şu anda kullandığımız gramer sözlüğü biçimbirimsel çözümleyicinin kullandığı kök sözlüğünün küçük bir alt kümesini içermektedir.

Bir sonraki adımda ise, gramer belli bir kapsamın üzerine çıktığında, Türkçe Ağaç Yapılı Derlem'den [11] yararlanmak ve [12, 13] çalışmalarında olduğu gibi istatistiki bilgilerle, ulaşılan çözümleri en iyi hale getirmek gibi hedeflerimiz bulunmaktadır.

**Ek Bilgi:** Yapılan çalışma, 105E021 numaralı “Türkçe için Geniş Kapsamlı Sözcüksel İşlevsel Gramer Tabanlı Tümce Çözümleyicisi Geliştirilmesi ve Akıllı Kitap Ortamına Entegrasyonu” projesi dahilinde TÜBİTAK tarafından desteklenmektedir.

## Kaynakça

1. Kaplan, R.M., Bresnan, J.: Lexical-functional grammar: A formal system for grammatical representation. In Bresnan, J., ed.: *The Mental Representation of Grammatical Relations*. MIT Press, Cambridge, MA (1982) 173–281
2. Güngördü, Z., Oflazer, K.: Parsing Turkish using the Lexical Functional Grammar formalism. *Machine Translation* **10**(4) (1995) 515–544
3. Bozşahin, C.: The combinatory morphemic lexicon. *Computational Linguistics* **28**(2) (2002) 145–186
4. Oflazer, K.: Dependency parsing with an extended finite-state approach. *Computational Linguistics* **29**(4) (2003) 515–544
5. Çakıcı, R.: Automatic induction of a CCG grammar for Turkish. In: *Proceedings of the ACL Student Research Workshop*, Ann Arbor, Michigan, Association for Computational Linguistics (2005) 73–78
6. Butt, M., King, T.H.: Restriction for morphological valency alternations: The Urdu causative. In: *Proceedings of The 10th International LFG Conference*, Bergen, Norway, CSLI Publications (2005)
7. Oflazer, K.: Two-level description of Turkish morphology. *Literary and Linguistic Computing* **9**(2) (1994) 137–148
8. Kaplan, R.M., Zaenen, A.: Long-distance dependencies, constituent structure, and functional uncertainty. In Baitin, M., Kroch, A., eds.: *Alternative Conceptions of Phrase Structure*. University of Chicago Press, Chicago (1988)
9. Maxwell III, J.T., Kaplan, R.M.: An efficient parser for LFG. In Butt, M., King, T.H., eds.: *The Proceedings of the LFG '96 Conference*, Rank Xerox, Grenoble (1996)
10. Kabak, B.: Turkish suspended affixation. *Linguistics* **45** (2007) (to appear).
11. Oflazer, K., Say, B., Hakkani-Tür, D.Z., Tür, G.: Building a Turkish treebank. In Abeille, A., ed.: *Building and Exploiting Syntactically-annotated Corpora*. Kluwer Academic Publishers (2003)
12. Frank, A., Sadler, L., van Genabith, J., Way, A.: From treebank resources to LFG f-structures: automatic f-structure annotation of treebank trees and CFGs extracted from treebanks. In Abeille, A., ed.: *Treebanks*. Kluwer Academic Publishers, Dordrecht (2003)
13. O'Donovan, R., Burke, M., Cahill, A., van Genabith, J., Way, A.: Large-scale induction and evaluation of lexical resources from the Penn-II and Penn-III Treebanks. *Computational Linguistics* **31**(3) (2005) 329–365