Makale Bilgisi: Özgen, M., Varlıklar, Ö., Yenal, Ö., Özgen, F. C., Birant, Ç. C. (2024). Türkçe İçin Sözdizim Ağacı Üretici. DEÜ Edebiyat Fakültesi Dergisi, Cilt:11, Sayı:2, ss.407-421.	**Article Info:** Özgen, M., Aktaş, Ö., Yenal, Ö., Özgen, F. C., Birant, Ç. C. (2024). Syntax Tree Generator for Turkish. DEU Journal of Humanities, Volume:11, Issue:2, pp. 407-421.
Kategori: Araştırma Makalesi	**Category:** Research Article
DOI: 10.69878/deuefad.1436282	**DOI:** 10.69878/deuefad.1436282
Gönderildiği Tarih: 13.02.2024	**Date Submitted:** 13.02.2024
Kabul Edildiği Tarih: 04.04.2024	**Date Accepted:** 04.04.2024

TÜRKÇE İÇİN SÖZDİZİM AĞACI ÜRETİCİ

Murat Özgen[*], Özlem Varlıklar[**], Fatma Cansel Özgen[***], Öykün Yenal[****], Çağdaş Can Birant[*****]

ÖZ

Bu araştırma, Evrensel Dilbilgisi çerçevesinde ele alınan başat türetimci kuramlar çerçevesinde Türkçe sözdizim ağaçlarını otomatik olarak oluşturmayı hedefleyen bir algoritmanın sonlanmış biçimidir. Bu anlamda Chomsky (1965, 1981, 1995, 2000 ve izleyen çalışmaları) çerçevesinde biçimlenen sözdizimsel araç ve düzenekler, program koduna işlenmiştir. Bulgular arasında yer alan adcıl uyumlu içtümceler çalışma kapsamı dışında bırakılmıştır. Bunun nedeni, adcıl uyum taşıyan tümcelerde yer alan görünümler üzerine alanyazında halihazırda bir uzlaşma sağlanamamış olmasıdır. Bu anlamda, kullanıcılar arayüzün solunda yer alacak olan biçimlendirme çubuğundaki işlemleri kullanarak işlerini görecek olan değişiklikleri ağaçlara uygulayabilecektir. Tümce ağaçlarının son biçimlenmelerini Office Word ya da Latex gibi ortamlara aktarmak isteyen kullanıcılar için farklı kaydet ya da "... biçiminde kaydet" seçenekleri bulunmaktadır.

Anahtar Sözcükler: doğal dil işlemleme, sözdizim, ağaç, arayüz, otomasyon

[*] Doç. Dr., Dokuz Eylül Üniversitesi Edebiyat Fakültesi Dilbilim Bölümü, murat.ozgen@deu.edu.tr, ORCID: 0000-0001-7960-6627
[**] Doç. Dr., Dokuz Eylül Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü, aktas.ozlem@deu.edu.tr, ORCID: 0000-0001-6415-0698
[***] Türkçe Öğretmeni, Adnan Menderes Ortaokulu, Bursa – Karacabey, fatmacanselozgen@gmail.com, ORCID: 0000-0001-8587-1099
[****] Uzman Yazılım Mühendisi, K2M Bilişim, oykun.yenal@gmail.com
[*****]Denetim Uzmanı, GmbH, ccbirant@gmail.com, ORCID: 0000-0001-5443-0098

SYNTAX TREE GENERATOR FOR TURKISH

ABSTRACT

This research is the finalized version of an algorithm that aims to automatically create Turkish syntax trees within the framework of dominant generative theories discussed within the framework of Universal Grammar. In this sense, syntactic tools and mechanisms shaped within the framework of Chomsky (1965, 1981, 1995, 2000 and his subsequent works) have been processed into the program code. Nominal agreeing clauses among the findings were excluded from the scope of the study. The reason for this is that there has not yet been a consensus in the literature on the aspects in nominal agreeing clauses. In this sense, users will be able to apply the necessary changes to the trees using the operations in the formatting bar on the left of the interface. For users who want to transfer the final formatting of sentence trees to environments such as Office Word or Latex, there are options to save as or "Save as...".

Keywords: natural language processing, syntax, tree, interface, automation

1. GİRİŞ

"Türkçe için otomatik sözdizim ağacı üretici"[2] başlıklı projeden üretilen bu araştırma, Evrensel Dilbilgisi çerçevesinde ele alınan başat türetimci kuramlar çerçevesinde Türkçe sözdizim ağaçları oluşturmayı hedeflemektedir. Bu hedef çerçevesinde, kodlanmış olan program, Türkçe basit ve çekimli içtümcelerden oluşan karmaşık tümcelerin içerdiği tüm ilişki ve işlemleri içermektedir. Bu anlamda Chomsky (1965, 1981, 1995, 2000 ve izleyen çalışmaları) çerçevesinde biçimlenen tüm kuramlar ayrıntılı olarak betimlenmiş; bu kuramlardaki sözdizimsel araç ve düzenekler program koduna işlenmiştir. Ayrıca, adı geçen çalışmalar dahilinde yürütülen Türkçe sözdizimi üzerine yapılmış kavramsal ve kuramsal çalışmalardaki bilgiler kapsamlı olarak taranmış; bu çalışmalarda bulgulanan tüm sözdizimsel araç ve işlemler Türkçenin dilbilgisel görünümlerine uygun bir biçimde programın algoritmasına girilmiştir.

Bulgular arasında yer alan adcıl uyumlu içtümceler çalışma kapsamı dışında bırakılmıştır. Bunun nedeni, adcıl uyum taşıyan tümcelerde yer alan görünümler üzerine literatürde halihazırda bir uzlaşma sağlanamamış olmasıdır. Kimi araştırmacılar, bu tümceleri Belirleyici Öbeği kapsamında çözümlerken; kimi çalışmalarda hala Tümleyici Öbeği olarak çözümleme eğilimi gözlemlenmektedir. Biz de bu tartışmalara tarafsız kalarak buna ilişkin görünümleri programın algoritmasına eklemek üzere daha ileriki aşamalara saklayacağız. Programa girilen Türkçe tümceler, Yetinmeci Sözdizim Kuramı ve Evre Kuramı çerçevesinde oluşturulacak iki ayrı düzlemde

[2] Bu proje Dokuz Eylül Üniversitesi Bilim Araştırma Projeleri Birimi tarafından 2019.KB.SOS.002 koduyla maddi olarak desteklenmiştir.

gösterimlenecektir. Kullanıcı programın işleyiş biçiminde ve arayüzünde yer alan bu iki butondan herhangi birini seçerek girdiği basit tümceleri yapılandırabilecektir. Girilen tümcelerdeki fazladan taşıma ve işlemleri göstermek üzere, kullanıcılara ağaç üzerinde değişiklik yapma şansı sunulacaktır. Bu anlamda, arayüzün solunda yer alacak olan biçimlendirme çubuğunda işlemleri kullanarak kullanıcılar işlerini görecek olan değişiklikleri ağaçlara uygulayabilecektir. Tümce ağaçlarının son biçimlenmelerini Office Word ya da Latex gibi ortamlara aktarmak isteyen kullanıcılar için farklı kaydet ya da "... biçiminde kaydet" seçenekleri eklenecektir.

2. Problem: Web-Tabanlı Mevcut Sözdizim Ağacı Programları

Alanyazında İngilizce için mshang_syntax_tree_generator, rssyntaxtree, VISL_tree_structure, pHpSyntaxTree, Lingustic Tree Constructor ve TreeForm programları bulunmaktadır. Bu programların tümü İngilizce tümceler üzerinde işlem görmektedir. Türkçe için henüz böyle bir çalışma gerçekleştirilmemiştir. Araştırmanın en özgün yanı budur.

mshang_syntax_tree_generator, etiketli ayraç gösterimi kullanan çevrimiçi bir uygulamadır. Uygulama, siz yazdıkça ağacı oluşturan ve eksik olabilecek tüm parantezleri kapatmaya çalışan bir kullanıcı arayüzüne sahiptir. Son ağaç resmine sağ tıklayıp "Resmi farklı kaydet" butonunu seçerek bilgisayara kaydetmek mümkündür. Örneğin;

[NP^ Alice]

[NP [N Alice] and [N Bob]]

[S[NP[N Alice]][VP[V is][NP[N'[N a student][PP^ of physics]

[S [X_a Movement] [Y example <a>]]

notasyonunu kullanarak bir ağaç çizmek olanaklıdır. Buradaki en temel handikap ağaç çizebilmek için en temel bilgilerin yanısıra, programın gerektirdiği kod dilini de öğrenmektir. Bu da kullanıcıya fazladan yük bindirmektedir.

Bir diğer program, RSyntaxTree, Yoichiro Hasebe'nin oluşturduğu Ruby programlama dilinde yazılmış bir grafik sözdizimi ağacı üreticidir. RSyntaxTree'nin orijinal versiyonu, “André Esenbach” tarafından phpSyntaxTree'ye dayanmaktaydı. RSyntaxTree'nin komut satırı sürümü GitHub'da mevcuttur. Etiketli köşeli parantez içinde, Editör alanına bir cümle girilir ve ‘PNG Çiz’ düğmesine basılır. Bir sözdizimi ağacının her dalı veya yaprağı bir budağa ait olmalıdır. Bir budak oluşturmak için, açılış dirseğinin hemen yanına bir etiket metni yerleştirilir. Sözgelimi, < > sembolünü kullanarak programda beyaz boşluk içeren budaklar oluşturabilirsiniz. Böylece “Modal < > Aux”, örneğin, “Modal Aux” olarak işlenecektir. Terminal düğümleri ile yaprakları arasındaki bağlayıcıların farklı şekilde çizildiği dört tip vardır (otomatik, üçgen, çubuk ve hiçbiri). Otomatik modda,

bir veya daha fazla boşluk içeren bir yaprak için bir üçgen çizilir (= bir ifade). Bir yaprak boşluk içermediğinde (= tek bir kelime), bunun yerine düz bir çubuk çizilecektir (yaprak sonunda bir ^ sembol içermiyorsa, bu, bir ifade olarak ele alınmasını sağlar). PNG görüntü formatına ek olarak, RSyntaxTree PDF ve SVG oluşturabilir. SVG, Adobe Illustrator, Microsoft Visio ve BOXY SVG gibi üçüncü parti vektör grafik yazılımlarındaki çıktı görüntülerini değiştirmek isteyenler için çok uygundur. 'Simetrize' seçeneği, dallanma budaklarının çizilme şeklini etkiler. Seçenekler Yazı tipi stili, Yazı tipi boyutu, Bağlayıcı yüksekliği gibi seçeneklerdir. Ortaya çıkan görüntünün görünümünü değiştirmek için bu seçeneklerin değerleri değiştirilir. Yine buradaki en temel handikap ağaç çizebilmek için en temel bilgilerin yanısıra, programın gerektirdiği kod dilini de öğrenmektir. Bu da kullanıcıya fazladan yük bindirmektedir.

VISL_tree_structure'ın el etiketli öğretim organı, önce Bjarne le Fevre Jakobsen (1997-2000) ve daha sonra otomatik ek açıklama ve revizyondan sorumlu Eckhard Bick (2000 - ...) tarafından denetlenen Danimarkalı VISL ekibinin devam eden ortak çabasıdır. Bu program, Text corpora'nın, özellikle de Corpus 90/2000 ve Arboretum treebank projelerinin, DSL tarafından sağlanan metin materyali ile yürütülmesiyle ortaya çıkmıştır. Otomatik ("serbest metin") Danca NLP sistemi, çok seviyeli Kısıtlı Dilbilgisel anlaşmazlıklara dayanmaktadır ve Eckhard Bick tarafından Portekizce için benzer bir proje temelinde geliştirilmektedir. Kullanılan biçimbilimsel inceleyici (Danmorf), 1986 Danca-Esperanto Machine Translation sisteminin Kaynak Dilidir. Sistemin sözdizimi, DSL ve Mikro Værkstedet tarafından sağlanan büyük sözcük veri tabanlarının yardımıyla ve Anders Hougaard ve Lone Hegelund tarafından derlenen ek değerlik ve anlam bilgisi ile güçlendirilmiş ve düzeltilmiştir. Sözdizimsel ağaç yapıları oluşturmak için kullanılan PSG-dilbilgisi Eckhard Bick tarafından yazılmıştır ve Martin Carlsen'in cg2tree derleyicisini kullanmaktadır. Danimarkalı Kısıtlayıcı Dilbilgisi şu anda, biçimbilimsel ve sözdizimsel anlam ayrımı için yaklaşık 7,000 bağlamsal kuraldan oluşur ve Danca metninin çalışmasını tam olarak sağlar. Eklenti Kısıtlama Dilbilgisi, diğer şeylerin yanı sıra, üye rollerini de ele alır[3]. Sistemin şimdiki sürümü, Pasi Tapanainen tarafından geliştirilen ve lisanslanan CG-2 kural derleyicisiyle ve Martin Carlsen tarafından yazılan VISL-CG derleyicisiyle çalışır. Yine buradaki en temel handikap ağaç çizebilmek için en temel bilgilerin yanısıra, programın gerektirdiği kod dilini ve kodun içine gömülü aygıtların işleyişlerini de öğrenmektir. Bu da kullanıcıya yine fazladan yük bindirmektedir.

pHpSyntaxTree, etiketli köşeli parantez gösterim ifadelerinden grafik sözdizimi ağaçları oluşturmanıza olanak tanır. Etiketli dirsek notasyonu

[3] Kısıtlayıcı Dilbilgisi kuramına giriş için bkz. "Fred Karlsson ve diğ., Kısıt Dilbilgisi: Sınırsız metnin ayrıştırılması için dilden bağımsız bir sistem, Berlin 1995".

kullanarak ifade girilir. Girilen sözcük için bir sözdizimi ağacı grafiği oluşturmak için "Çiz" seçeneği tıklanır. Oluşturduğunuz ağaç resmini kopyalayıp belgelerinize yapıştırabilir veya indirmek için ağaç resmini sol tıklatabilirsiniz. Abone parçalarına sahip bir düğüm oluşturmak için, alt karakter kısmı _ karakterini kullanarak kullanılır. Örneğin "NP_1", grafikte NP1 olarak görünecektir. İsteğe bağlı olarak adlandırılan düğümleri (NP1, NP2 vb.) atomatik olarak numaralandırmak için alt sekme seçilir. Siyah beyaz yazıcı dostu bir görüntü oluşturmak için rengi ve / veya düzgün çizgileri devre dışı bırakmak mümkündür. Yine buradaki en temel handikap ağaç çizebilmek için en temel bilgilerin yanısıra, programın gerektirdiği kod dilini ve kodun içine gömülü aygıtların işleyişlerini de öğrenmektir. Bu da benzer biçimde kullanıcıya fazladan yük bindirmektedir.

Linguistic Tree Constructor (LTC), metinden dilsel söz dizimi ağaçları oluşturmak için ücretsiz bir programdır. Kullanıcı, sözdizimsel bir analize giden yolu gösterir ve tıklar. LTC, çok miktarda metnin sözdizimsel analizlerini hızlı bir şekilde üretmek için tasarlanmıştır. Program kendi başına analiz yapamaz. Kullanıcı istediği zaman ağacı çizmek için tamamen özgürdür. Ancak, program ağacın bir ağaç olduğundan ve başka bir çeşit grafik olmadığından emin olmak zorundadır. Buradaki en temel handikap ağaç çizebilmek için en temel bilgilerin yanısıra, programın gerektirdiği kod dilini ve kodun içine gömülü aygıtların işleyişlerini de öğrenmektir.

TreeForm sözdizim ağacı çizim yazılımı, bir Dilbilgisi Sözdizim / Anlambilim ağaç çizim editörüdür. Grafiksel n-ary ağacı çizimi için tasarlanmıştır. Linux kullanıcıları, "java -jar TreeForm.jar" ile kurulum dizin konumundan konsolda TreeForm'u başlatmalıdır. TreeForm Sözdizimi Ağacı Çizim Yazılımı, diğer alt kategoriden Geliştirme kategorisinin bir parçası olan ücretsiz bir yazılım uygulamasıdır. Uygulama şu anda İngilizce olarak mevcuttur ve en son 2008-03-28 tarihinde güncellenmiştir. Program WinXP, WinVista, Win7 x64, Win7 x32 üzerinde yüklenebilir. Bu programdaki en temel handikap ise Windows dışındaki işletim sistemlerini desteklememesidir. Ayrıca, program otomatik ağaç üretmez, ağaçlar üzerindeki tüm işlemler, manuel gerçekleştirilmektedir.

Özetle, alanyazına göz attığımızda, tümceler için ağaç oluşturan programların en temelde iki sorunu olduğu görülür. İlk sorun, bu programların hepsinin İngilizce için kurgulanmış olması ve işleyiş biçimlerinin Türkçe tümceleri desteklememesidir. Ayrıca bu programlarda destek dilinin İngilizce olduğunu düşünsek bile, ağaç çizebilmek için ayrıca bir koda ya da manuel yardıma gereksinim bulunmaktadır. İkinci sorun ise her programın yukarıda sayılan yalnızca bir kuramı çerçeve belirleyerek tümceleri ağaçlandırmasıdır. Bu projenin en temel amacı bu iki sorunu alanyazındaki programlardan bağımsız yepyeni bir program geliştirerek çözmeye çalışmaktır. Bu amaçla öncelikle, her kurama girdi sağlayan Türkçe tümce kuralları ve dilötesi

etiketlerin ayrıntılı dökümü belirlenecek ve bunlar bilgisayar ortamına aktarılacaktır. Daha sonra, Türkçe sözlükçesinde yer alan sözlükbirimlerin etiketleri belirlenip bu etiketlerin denetimi sağlanacaktır. Son olarak, ağaçlandırmaya temel olan sayıltılar ve kısıtlar belirlenip programa işlenecek ve program sonlandırılacaktır. Buradaki en önemli kısıtlardan bir tanesi, programın yalnızca çekimli ve basit tümcelerin ağaçlarını yapılandıracak olmasıdır. Bunun nedeni, çekimsiz içtümcelerden oluşan ve adcıl uyum taşıyan Türkçe tümcelerin alanyazında tartışmaya hala açık birden fazla noktası bulunmasıdır. Elde edilen programın üreteceği sözdizim ağaçlarının yukarıda sözü edilen kuramlar çerçevesinde iki ayrı seçenek sunmasının ilk sorunu çözmesi beklenmektedir. Ortaya çıkacak olan programın Türkçe tümceleri ağaçlandıracak olması ise bu anlamda ikinci sorunu çözecektir. Ayrıca, programın Türkçe tümceleri ağaçlandırması ise Türkçe üzerine yapılan ilk ve tek çalışma olması nedeniyle büyük bir önem taşımaktadır.

3. Ağaç Üretici Algoritmanın Kuramsal Artalanı

Öncesinde de söz edildiği gibi, Chomsky (1965, 1981, 1995, 2000 ve izleyen çalışmaları) çerçevesinde biçimlenen tüm kuramlar ayrıntılı olarak betimlenecek; bu kuramlardaki sözdizimsel araç ve düzenekler program koduna işlenecektir. Yetinmeci Çizginin (Minimalist Program) temel önerisi, dilin *sesletimsel-algısal* (articulatory-perceptual) ve *kavramsal-sezgisel* (conceptual-intentional) arakesitlerince dayatılan sınırlamalara en uygun çözüm olduğudur (Chomsky, 1995). Bu arakesitlerce dayatılan sınırlamalar sonucunda, tümceler işlemlenir; üretilir ve depolanır. İşlemlenen ve üretilen bilgilerin kısa-süreli bellekte depolanmalarının üzerinde birtakım sınırlamalar bulunması, sözdizimsel işlemlerin kapsamlarını da sınırlayabilir. Chomsky (MI; DbP; BEA; TFLD AUGB ve OP)[4] , bu sınırlamaları dikkate alarak *Genişletilmiş Ölçünlü Kuram* (Extended Standard Theory), *Yönetim ve Bağlama Kuramı*'ndan (Government & Binding Theory) köklü bir ayrılıştan söz eder. Buna göre, sözdizimsel türetimler, *evre* (phase) adı verilen çoğalımlı işlemlerden oluşmaktadır. Her evre, ayrı bir *sözlüksel alt-dizin*den (lexical sub-array) yapılandırılır. Chomsky, evrelerin önermeli doğasından yola çıkarak TümÖ (Tümleyici Öbeği - Complementizer Phrase) ve *e**Ö'lerin (*v**P) evre oluşturduklarını belirtir. *e** bu bağlamda, tam üye yapısını taşıyan işlevsel bir baştır (Chomsky, OP: 143). TümÖ bütün bir *önermesel karmaşığı* (propositional complex) gösterirken, *e**Ö ise, bütün bir *üye karmaşığı*nı (argument complex) anlatmaktadır. Buna göre, Tüm ve *e** evre başları iken, ZÖ'nün (Zaman Öbeği - Tense Phrase) başı Z ile EÖ'nün (Eylem Öbeği -

[4] Chomsky'nin 2000 ve sonrasında izleyen çalışmaları için kullanılan kısaltmalar şu biçimdedir: Chomsky (2000) 'Minimalist Inquiries': **MI**; Chomsky (2001) 'Derivation by Phase': **DbP**; Chomsky (2004) 'Beyond Explanatory Adequacy': **BEA**; Chomsky (2005) 'Three Factors in Language Design': **TFLD**; Chomsky (2006) 'Approaching UG from below': **AUGB**; Chomsky (2008) 'On Phases': **OP**

Verb Phrase) başı E evre başları değildir. Bir sözlüksel alt-dizinden yapılandırılan evreler, Döngüsel Aktarım (Cyclic Transfer) aracı kullanılarak daha önce sözü edilen sesletimsel-algısal ve kavramsal-sezgisel arakesitlere gönderilir. Bu sayede, hem türetim üzerindeki işlem yükü azalır hem de arakesitlere çoklu erişim sağlanır.

Program algoritmasına kodlamalar yukarıda sözü edilen dilbilgisi mimarisi çerçevesinde gerçekleşecektir. Sözgelimi, *Çizgisel Örtüşme Beliti*'nin 'ÇÖB' (Linear Correspondence Axiom 'LCA') Japonca, Türkçe, Ö(zne)-N(esne)-E(ylem) dizilişini açıklamada ilk bakışta birtakım sorunlar yaşadığı düşünülür. Bu durum üzerine alanyazında birçok çalışma bulunmaktadır. Bu çalışmalardan bir bölümü ÇÖB'ü desteklemekteyken (Lee, 2007; Pearson, 2001; Zwart, 1994 ve 2002), bir bölümü ÇÖB'ün yapılanmasında birtakım iyileştirmeler gerektiğini ya da bu tür dillerin özünde Ö-N-E dizilişli olduğunu savlamaktadır (Kelepir, 1996; Kural, 1997; Stepanov, 2007 vb.). Bu çalışmalar içerisinde Türkçe üzerine yapılmış doğrudan üç çalışma bulunmaktadır (Kelepir, 1996; Kural, 1997; Zwart, 2002).

Bir diğer görünüm uyum çalışmalarıdır. Zidani-Eroğlu (1997), Moore (1998) ve Özsoy (2001), belirtme durumu belirtili öznelerin anatümceye yükseldiğini ve burada durumlarını yetkilendirildiğini belirtir. Adcıl uyum konusunda ise, Keskin (2009), adcıl uyum alanındaki *tamlayan* (genitive) ve *belirtme* (accusative) durumlarının *Bel* (D) tarafından gerçekleştirildiğini belirtir. Programa girilen Türkçe tümceler, Yetinmeci Sözdizim Kuramı ve Evre Kuramı çerçevesinde oluşturulacak iki ayrı düzlemde gösterimlenecektir. Kullanıcı programın işleyiş biçiminde ve arayüzünde yer alan bu beş butondan herhangi birini seçerek girdiği basit tümceleri yapılandırabilecektir.

Programın artalanını oluşturan Üretici Dilbilgisi tarihsel açıdan belirli dönemlere ayrılır. Dönüşümsel kuramın ilk biçimi Chomsky'nin 1957 (C57) yılında yayımladığı *Syntactic Structures* adlı yapıtıyla ortaya atılmıştır. Daha sonra, 1965 yılında yayımlanan *Aspects of the Theory of Syntax* (C65) adlı yapıtla, kuramın daha bütüncül bir görünümünü sunan Ölçünlü Kuram gelmektedir. Kuramda 1965 sonrasında da önemli denebilecek gelişmeler yaşanmıştır. Daha sonraları 1981 yılında da *Lectures on Government and Binding: The Pisa Lectures* adlı yapıtla *İlkeler ve Değiştirgenler Kuramı* ve sonrasında Yetinmeci Kuram gelmektedir. Son yıllarda ise, Yetinmeci Kuram kendi içerisindeki değişimler ve çıkarsamalar ile son biçimi olan Evre Kuramı'na dönüşmüştür.

C57, dilbilgisinin üç tür kurallar düzleminden oluştuğunu kabul eder. Bu kurallar düzlemi; öbek yapı kuralları, dönüşüm kuralları ve biçimsesbilim kurallarıdır. Kuram, dönüşümler ile kurallar düzlemini ön plana çıkardığından dönüşümsel dilbilgisi olarak da anılmaktadır. C57 kuramının ilk bileşeni öbek

yapı kurallarını düzenleyen öbek yapı bileşenidir. Bu kurallar düzlemi, doğal konuşucuların sözcükleri ayırma, bunların kategorilerini bilme, sözcükler arasındaki öbeksel ilişkileri tanıma gibi tümce yapısı bilgilerine dayanan ve tümce üreten kurallardır. Öbek yapı kuralları, yeniden yazma kuralları anlamına gelmektedir. Yeniden yazma işlemi, öbek belirleyicilerini tümdengelimli olarak yeniden tanımlamak demektir. Yeniden tanımlanan her öbek için ulamsal kurallar işlerken, bu ulamsal kuralları sözcüklere bağlayan sözlüksel kurallar bulunmaktdr

Chomksy Dilbilgisinin ilk biçimlenmesi olan C57 ile standart biçimlenmesi olarak kabul edilen C65'e kadar geçen süre içinde, söz konusu dilbilgisi hakkındaki tartışmalar daha çok dönüşümler üzerine yoğunlaşmıştır. İlgili tartışmaların yaygınca benimsenen; bu nedenle C65'e etki eden ve birbiri ile etkileşimde olan iki önemli yönü bulunmaktadır. Birincisi, dönüşümlerin yapı koruyucu olması gerekliliği, diğeri biçemsel olanlar dışındaki dönüşümlerin tümünün zorunlu dönüşümler olarak kabul edilmesi gerekliliğidir. Chomsky'nin 1965 yılında yayımladığı *Aspects of the Theory of Syntax* adlı yapıtıyla Ölçünlü Kuram olarak adlandırılan biçimini alan C65, C57'ye nazaran daha net bir dilbilgisi sunuluşu vermektedir. C65, dilbilgisini bileşenlere ayırmaktadır. C57'nin öbek yapıya yapışık olarak gördüğü sözcükler, öbek yapıdan ayrı ama yapılanmış bir bileşen olan sözlükçede listelenmektedir. Bu bileşen, basitçe, doğal konuşucuların sözcük hakkındaki bütün sezgisel dil bilgilerinin bir araya toplanıp biçimlendiği bileşendir. Öbek yapı kuralları C57 ile aynı kalmak üzere sözlükçe ile beraber taban bileşeninde toplanmıştır. Üretici olan bu sözdizimsel bileşene karşılık yalnızca yorumlayıcı olan anlamsal ve sessel bileşen ayrımı yapılmıştır.

1970'lerden 1980'lerin başına kadar, kuramda önemli bazı değişikliklere gidilmiştir. Bu değişiklikler pek çok yazarca Genişletilmiş Ölçünlü Kuram (GÖK) olarak adlandırılmaktadır. GÖK, dönüşümlü dilbilgisinin 1970'lerin ortalarındaki sürümüdür ve daha önceki Ölçünlü Kuramdan birtakım yönlerle ayrılmaktadır. Bu kuramda, sözdizimi yapısı anlambilimsel yorumu içine almakta ve Chomsky (1971) ile kısmen Jackendoff (1972)'ye dayanmaktadır. GÖK'ün kurama getirdiği pek çok yenilikten biri sözdizimsel sunuluşlara izlerin girmesidir. Bu, dönüşümlerin yapı koruyucu olmaları ilkesine bağlı bir değişikliktir. Bunun dışında, ilke olarak Aşamalı-X Sözdizimi kabul edilmiştir. GÖK ile birlikte, kuralların belirlenmesinden daha çok kurallardaki kısıtların belirlenmesi gereği ortaya çıkmış ve kısıt terimi önem taşımaya başlamıştır. Buna bağlı olarak da dilbilgisine bir dönüşüm aracı olarak *α-taşıma* kavramı getirilmiş, böylece de dönüşüm kurallarının gücünde önemli ölçüde azalma oluşmuştur. Ayrıca, daha önceki yüzey yapının derecesinin azalması sonucu tüm dönüşümleri izleyen ancak süzgeçlerden ve silme kurallarından önce gelen bir düzey olan sığ yapı kavramı ortaya atılmıştır. Önemli bir değişim de Mantıksal Biçim olarak adlandırılan büsbütün yeni sunuluş düzeyinin yer almasıdır.

Chomskyci dilbilgisinin önemli bir halkasını oluşturan Yönetme ve Bağlama Kuramının (YBK) ilk sunuluşu 1981 yılında yayımlanan *Lectures on Government and Binding* adlı yapıttır. Sonraları, İlkeler ve Değiştirgenler Kuramı (İDK) olarak da adlandırılan bu kuram, dilbilgisini ilkeler ve değiştirgenlerin 'yuvalandığı' modüllerden oluşmuş bir model olarak ele almıştır. İDK'de içinde kuralların bulunduğu bileşen kavramı atılmış, yerine sunuluş kavramı getirilmiştir. Bu sunuluşların içinde kurallar yerine ilkeler ve değiştirgenler bulunmaktadır. İDK'de sözdizimi sunuluşunun yanı sıra, anlambilim ve sesbilim sunuluşları da yer almaktadır. Daha önceki kuramlardaki derin yapı ve yüzey yapı düzlemlerinin adı İDK'de d-yapı ve y-yapı olarak ele alınmıştır. Bunun nedeni, GÖK öncesi kuramlara göre, d-yapının daha soyut olmasından ve y-yapının GÖK öncesi kuramlardan farklı olarak kendisinden sonra da sunuluş düzlemlerinin bulunmasından, yani dilbilgisinin son sunuluşu olmamasından kaynaklanmaktadır. D-yapı ile y-yapı arasındaki bağ, α-taşımalar ile sağlanmaktadır.

Yönetme ve Bağlama Kuramının çeşitli diller üzerine yapılmış dilbilim çalışmalarına önemli etkileri olmuştur. Özellikle birinci dil ve ikinci dil edinimi çalışmalarında bu kuramdaki 'ilkeler' ve 'değiştirgen'ler (İ&D) önemli açıklamalar getirmiştir, pek çok çalışmayla da bu düşünce desteklenmiştir. Ancak 1980'lerin sonunda, Chomsky (1993, 1995) ve Chomsky & Lasnik (1993) çeşitli dilsel işlemlerin gerçek doğasını yeniden gözden geçirmiştir, böylece işlemlerin pek çoğuna yeni açılardan bakılmış ve ekonomik olmayanlar reddedilmiştir. YB, bugüne kadar ele alınan en başarılı İ&D Kuramı olduğundan Yetinmeci bakış açısı için temel oluşturmaktadır. İ&D Kuramında, dil dizgesi çeşitli sunuluş düzlemleri bulunmaktaydı. Bu sunuluş düzlemleri, DY, YY, MB ve SB'ydi. Bu düzeylerden edim dizgeleri olan Söyleyiş Algısal (S-A) ve Kavramsal Sezgisel (K-S) dizgelere girdi sağlayan dilsel düzlemler MB ve SB'nin vazgeçilmez düzlemler olduğu kabul edilip DY ve YY düzlemleri terk edilmiştir. Bu durumda, yetinmeci bir anlayışla tüm dil dizgesinin ilke ve değiştirgenleri MB ya da SB'de okunabilir durumdadır. İ&D Kuramı yetinmeci bakış açısıyla son biçimine, görünümüne kavuştuktan sonra, kuram üzerine araştırmalar devam etmiş; yetinmeci anlayış çerçevesinde kurama daha büyük yenilikler getirilmiştir. Yetinmeci anlayış dilbilgisinin T-tasarımına karşı, daha basit, daha yetinmeci bir tasarım oluşturulmuştur. Buna göre, Sözlükçeden sözlüksel diziye atanan sözcükler, İçsel ve Dışsal Birleştir aracı kullanılarak birleştirilir ve evreler oluşturulur. Oluşturulan evreler Aktar aracı kullanılarak SESsel bileşkeye ve ANLAMsal bileşkeye gönderilir.

Kurama adını veren *evre* (phase) kavramı bu yeniliklerden en önemlisidir. MI, DbP, BEA, OP çalışmalarında Chomsky, türetimlerin bütüncül bir biçimde değil, evreler oluşturarak ilerlediğini varsaymaya başlamıştır. Evre, döngüsel aktarım birimi olarak, sözdizim ve dışsal dizgeler arasındaki ilişkiyi iki yönüyle en uygun biçime getirir:

(i) belirli aralıklarla uyguladığı türetimsel bilgileri `unutma' yoluyla işlemsel ağırlığı azaltarak ve
(ii) tek-döngülü üretimdeki gereksiz içsel düzeyleri ortadan kaldırarak.

Chomsky, evrelerin önermeli doğasından yola çıkarak TümÖ ve *e**Ö'lerin aşama oluşturduklarını belirtir. TümÖ bütün bir tümcesel karmaşığı gösterirken, *e**Ö ise, bütün bir konusal karmaşığı anlatmaktadır. Bir evre içerisindeki tüm sözdizimsel işlemler sona erdikten sonra, evre, döngüsel aktarım ile anlamsal bileşkeye aktarılır ve bu evre, daha fazla sözdizimsel işlem için girilemez ve geçirimsiz duruma gelir. Chomsky bu koşula *Evre Girimsizliği Koşulu* (EGK) adını vermiştir. Evrelerin sonraki aşamalarda geçirimsiz olmaları durumunda dilsel birimlerin özelliklerin durum, GYİ (=EPP) gibi biçimsel özellikleri nasıl eşlediği sorusu gündeme gelecektir. Chomsky, bu sorunu ortadan kaldırmak için UYUŞUM aracını kuramın içine yerleştirmiştir. Z ya da e başı yorumlanamaz özelliklere [-*y*Ö] sahiptir. Özellik, hedefi arayan bir yoklayıcıdır, yani bu ikisi uyumu sağlamak için özelliklerini karşılaştırmaktadır. Uyuşum işlemi yoklayıcı ve hedefin yorumlanamaz özelliklerinin silinmesiyle sonuçlanmaktadır. Chomsky tüm sözdizimsel işlemlerin yalnızca evre başları tarafından gerçekleştirildiğinden söz etmektedir. Sözgelimi, öznenin GYİ özelliğini eşlemesi için gerçekleştirdiği Ü-taşıma işlemi aslında evre başı olan T tarafından gerçekleştirilmektedir. Buna göre, yorumlanamaz özellikler sözlükçede değerlendirilmemiş biçimde bulunmaktadır ve bu özellikler dar sözdizimde değerlendirilip silinmelidir. Aksi takdirde, bu özellikler arakesitlerde çakışmalara neden olacaktır. Chomsky'nin bu çıkarımlarından yola çıkılarak kuramın içerisine Özellik Devri Düzeneği adı verilen bir işleyiş süreci yerleştirilmiştir. Bu sürece göre, evre başları olan T ve e başlarından özellikleri devralan T ve E'ler gerekli özellikleri Uyuşum aracını kullanarak eşler, değerlendirir ve siler. Özellikleri değerlendirilip tamamlanan bu evre Aktar aracıyla arakesitlerelere gönderilir.

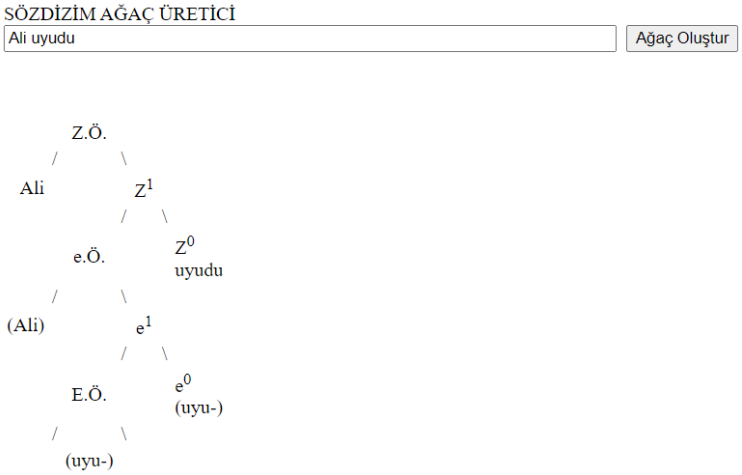
4. Program Basamakları

24 aylık aşamada meydana gelen pandemi koşulları ve bu koşulların yarattığı yeni süreçler nedeniyle 20 aylık iş paketine karşılık gelen bir çalışma gerçekleştirilmiştir. Araştırmacıların eğitim – öğretim ve idari işlerindeki sorumlulukları nedeniyle ortaya çıkan ve BAP komisyonu tarafından kabul edilen süre uzatma talebimiz sayesinde proje hedeflendiği şekliyle tamamlanmış, program işlerliğini kazanmıştır.

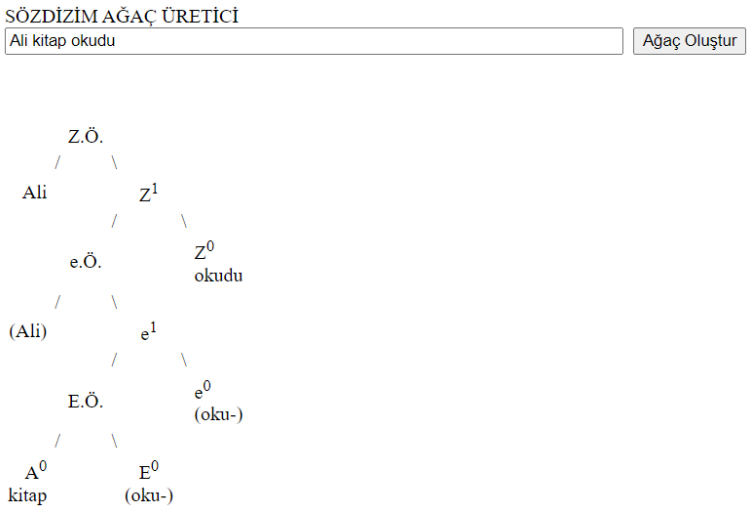
Yazılım, aşamalar halinde gerçekleştirilmiş ve basitten zora doğru süreçlendirilmiştir. Gerek tümce içindeki sözcük sayısı gerekse öbek niteliklerinin nicel olarak artması üzerine şekillenen süreçlere dair kod geliştirilmiştir. İlk aşamada kurguladığımız yazılıma basit ve tek sözcüklü öbeklerden oluşan tümceler girilmiş ve çıkan sonucun başarılı olduğu

görülmüştür. Yukarıda da belirtildiği şekliyle aşamalar sırasıyla belirlenmiş ve bu aşamalara istinaden program çıktıları aşağıdaki şekilde elde edilmiştir:

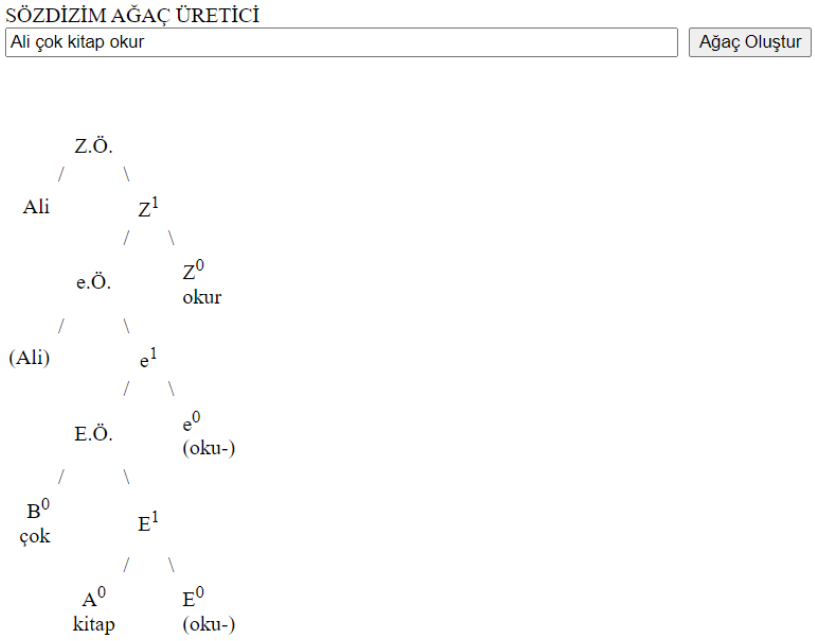
- Geçişsiz bir tümce (Görsel 1),
- Tekgeçişli bir tümce (Görsel 2),
- Tekgeçişli ve belirteçli bir tümce (Görsel 3)
- Çiftgeçişli bir tümce (Görsel 4)



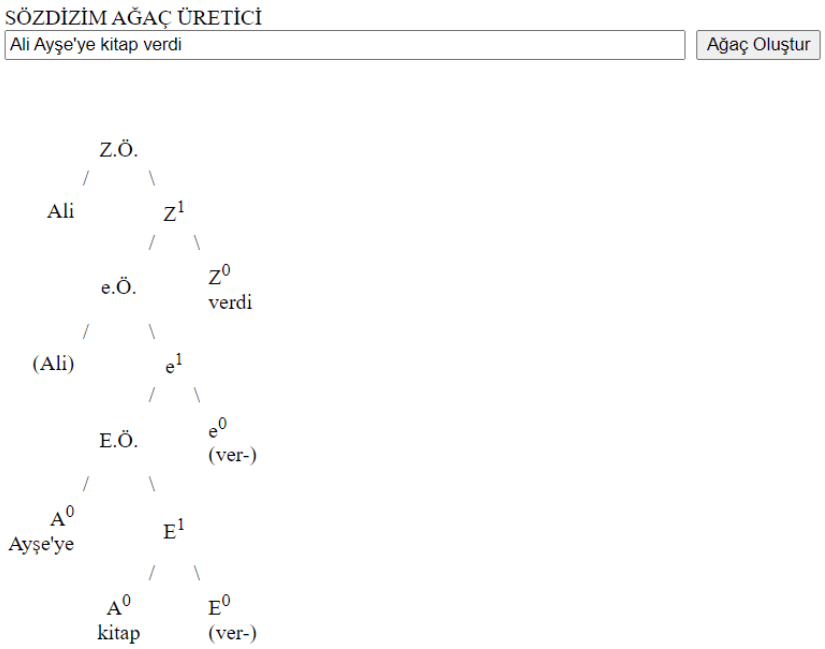
Görsel 1. "Ali uyudu" tümcesinin ağacı



Görsel 2. "Ali kitap okudu" tümcesinin ağacı



Görsel 3. "Ali çok kitap okur" tümcesinin ağacı



Görsel 4. "Ali Ayşe'ye kitap verdi" tümcesinin ağacı

Ağaç türetici algoritma için tümcenin öğelerini bulmak taban olarak alınmıştır. Bir tümce içinde yer alabilecek özne, yüklem, nesne (belirtili – belirtisiz), dolaylı tümleç ve belirteç yapılarını tanımlamak bir ağacın çatılarını çizebilmek için en önemli olgudur. Her ne kadar algoritmada bu öğelerin sözcük ya da öbek olabileceği ihtimali üzerinde durulsa da, kod geliştirme aşamasında test yapılabilmesi ve projenin taahhüt edilen sürede tamamlanabilmesi için bu öğelerin sözcük öbeklerinden değil sözcüklerden oluştuğu kabul edilerek yapılmıştır. Bunun yanı sıra tümcelerin bilgi yapılarına göre çalkalanmış biçimleri değil, olağan sözcük dizilişinde bulunduğu varsayılmıştır. Bu nedenle ilk sözcük Özne, son sözcük ise yüklem olarak öngörülmüştür. Bu öğeler dışındaki sözcükler için ise otomatik olarak ağaç çizilmiştir.

5. SONUÇ

Ağaç üretici programın başarısı, internet ortamında herkese servis verebilen bir sözdizim ağaç üretici yazılım prototipinin gerçekleştirilmiş olması ile ölçülmektedir. Şu ana kadar gerçekleştirilen çalışmalara göz atıldığında, projenin bütününün tamamlandığı görülür. Daha önce de söz edildiği gibi programa daha fazla tümce girerek testler yapılmış ve son düzenlemelerin de koda uygulanması ile program tamamlanmıştır.

Projenin süreç olarak tamamlanmasının yanı sıra geliştirme çalışmaları devam etmekte olup yukarıda belirtilen öğe tespit yazılımının sözcük öbekleri üzerine de çalışması hedeflenmiştir. Bu hedefe ulaşılmasının sonrasında, söz konusu yazılım, son kullanıcıya bir web uygulaması olarak sunulacaktır. Yazılımın geliştirilmesine özellikle Dilbilim paydaşları tarafından oluşacak geribildirimler ışığında devam edilecek; iyileştirme çalışmaları çıkan sonuçlar temelinde farklı yazılım altyapılarında sürdürülecektir. Ayrıca, sonraki aşamada adcıl yantümcelerin de değerlendirileceği bir yazılım geliştirilmesi hedeflenmektedir.

ÇIKAR ÇATIŞMASI BEYANI

Yazarların herhangi bir çıkara dayalı ilişkisi bulunmamaktadır.

ETİK ONAY/KATILIMCI ONAMI

Makale kapsamında katılımcı kullanılmamıştır.

MADDİ DESTEK

Bu program Dokuz Eylül Üniversitesi Bilim Araştırma Projeleri Birimi tarafından 2019.KB.SOS.002 koduyla maddi olarak desteklenmiştir.

YAZAR KATKILARI

Bu araştırma ve araştırma ile ilgili tüm aşamalar 5 (beş) yazar tarafından eşit olarak yürütülmüştür.

KAYNAKÇA

Aktaş, Ö. & Çebi, Y. (2013). Rule-based sentence detection method (RBSDM) for Turkish, *International Journal of Language and Linguistics*. Vol. 1, No. 1, 2013, pp. 1-6. 10.11648/j.ijll.20130101.11

Birant, Ç. C. (2008). *Root-suffix seperation of Turkish words*. [Unpublished MA Thesis]. Dokuz Eylul University.

Chomsky, N. (1957). *Syntactic structures*. Mouton de Gruyter.

Chomsky, N. (1965). *Aspects of the theory of syntax* . MIT press.

Chomsky, N. (1971). Deep structure, surface structure and semantic interpretation. *Semantics: foundational issues* içinde (s. 193-216). Routledge.

Chomsky, N. (1981). *Lectures on government and binding: The Pisa lectures*. Walter de Gruyter.

Chomsky, N. (1993). A minimalist program for linguistic theory. *The view from building 20: Essays in linguistics in honor of Sylvain Bromberger* içinde, (s. 1–52). MIT Press.

Chomsky, N. (1995). *The minimalist program.* MIT Press.

Chomsky, N. (2000). Minimalist inquiries: the framework. *Step by step* içinde (s. 89-156). MIT Press.

Chomsky, N. (2001). Derivation by phase. *Ken Hale: A life in language* içinde (s. 1-52). MIT Press.

Chomsky, N. (2004). Beyond explanatory adequacy. *Structures and beyond* içinde (s. 104-131). Oxford University Press.

Chomsky, N. (2005). Three factors in language design. *Linguistic inquiry 36: 1*. 1-22. 10.1162/0024389052993655

Chomsky, N. (2006). *Approaching UG from below.* Ms.: MIT.

Chomsky, N. (2008). On phases. *Foundational issues in linguistics theory* içinde. (s. 133-166) MIT Press.

Jackendoff, R. (1972). *Semantic interpretation in Generative Grammar*. MIT Press.

Karlsson, F. (1995). The formalism and environment of Constraint Grammar Parsing. *Constraint grammar. a language ındependent system for parsing unrestricted text* içinde (s.41-89) Mouton de Gruyter.

Kelepir, M. (1996). *The implications of antisymmetry theory of syntax on Turkish word order*. [Unpublished MA Thesis]. Boğaziçi Üniversitesi.

Keskin, C. (2009). *Subject agreement-dependency of accusative case in Turkish or jump-starting grammatical machinery*. LOT Publishing.

Kural, M. (1997). Postverbal constituents and Linear Correspondence Axiom in Turkish. *The Linguistic Inquiry* (28), 498-519. https://www.jstor.org/stable/4178988

Lee, J.-S. (2007). Deriving SOV from SVO in Korean. *The Linguistic Association of Korea Journal, 15* (3), 1-20.

Moore, J. (1998). Turkish copy-raising and A-chain locality. *Natural language & linguistic theory*, *16*(1), 149-189.

Özsoy, A. S. (2001). On 'small'clauses, other 'bare'verbal complements and feature checking in Turkish. *The verb in Turkish* içinde (s. 213-237). *John Benjamins*.

Pearson, M. (2001). *The clause structure of Malagasy: A minimalist approach*. [Unpublished PhD Dissertation]. UCLA, Department of Linguistics.

Stepanov, A. (2007). The end of CED? Minimalism and extraction domains. *Syntax, 10* (1), 80–126. https://doi.org/10.1111/j.1467-9612.2007.00094.x

Zidani-Eroğlu, L. (1997). Exceptionally case-marked NPs as matrix objects. *Linguistic inquiry*, *28*, 219-230. https://www.jstor.org/stable/4178975

Zwart, C. J. (1994). Dutch is head-initial. *The Linguistic review* (11), 377-406.

Zwart, C. J. (2002). The antisymmetry of Turkish. *Generative Grammar in Geneva* (3), 23-36.