Bağımlılık hesaplamasında zorlandığımız noktalar

Some dependency nuts to crack computationally

Cem Bozşahin

Computer Eng., and Cognitive Science, ODTÜ (METU)

(currently visiting Boğaziçi Univ. Cognitive Science)

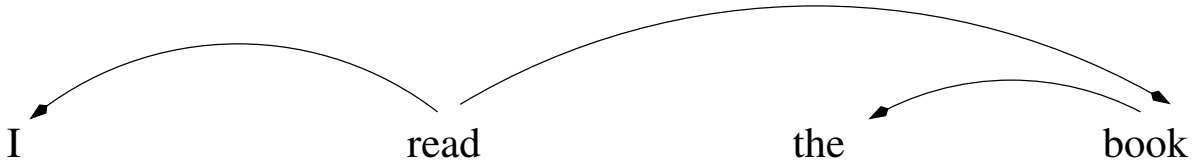
May 9, 2011, İstanbul Technical University (İTÜ)

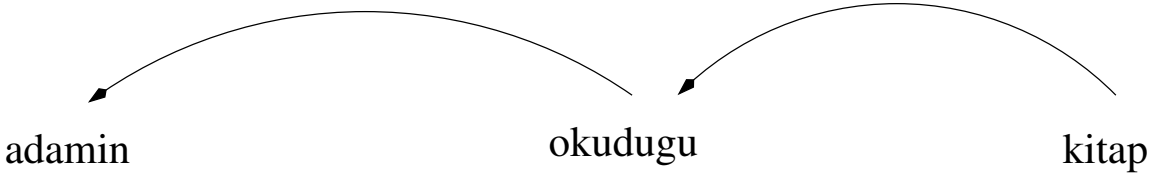
Since Kuhlmann and Nivre (2006), we know that **Dependency parsing** must strike a balance

between expressivity and complexity

Projectivity, single-headedness, planarity, multi-planarity,

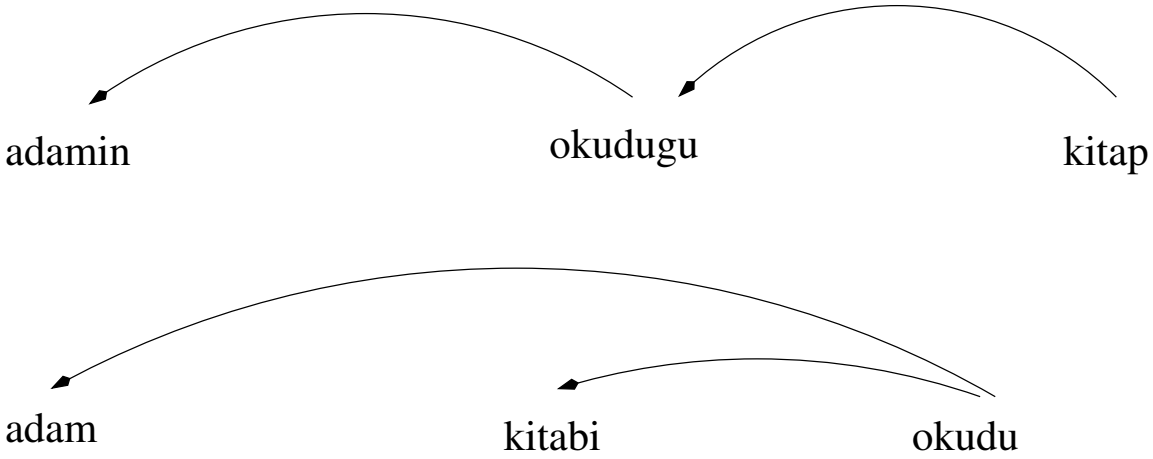
well-nestedness, gap degree, edge degree.



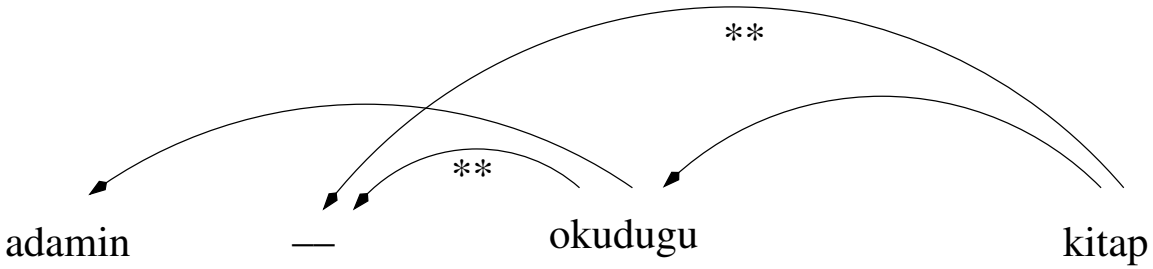


Tesnière (1959); Mel'čuk (1988)

Dependency grammars are sensitive to structure:

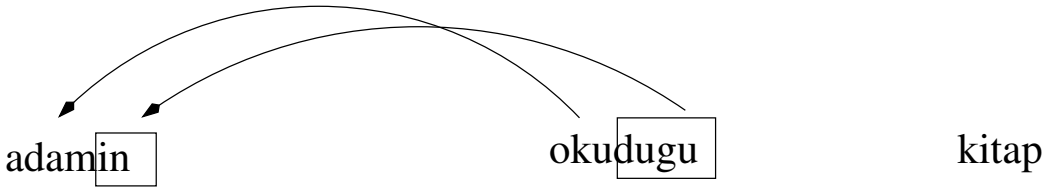


They are also sensitive to **surface** structure:



Dependencies are between words and observable items (end of sentence etc.)

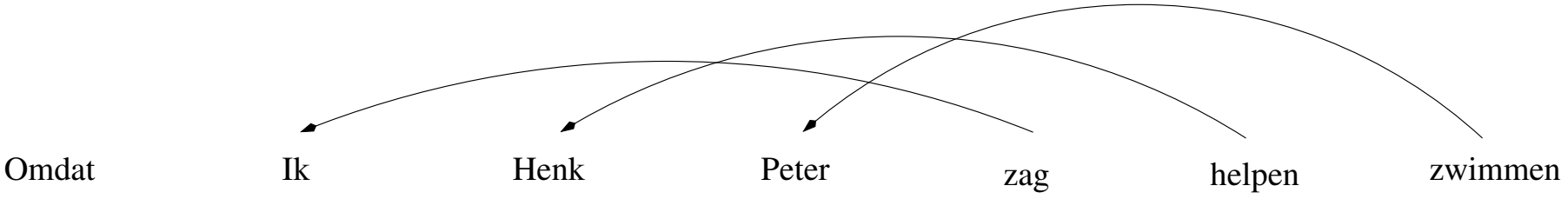
A closer look at dependencies might reveal some crossings:



These are within words, hence not so critical to parsing.

(see Eryiğit, Nivre and Oflazer 2008 for Turkish dependency parsing)

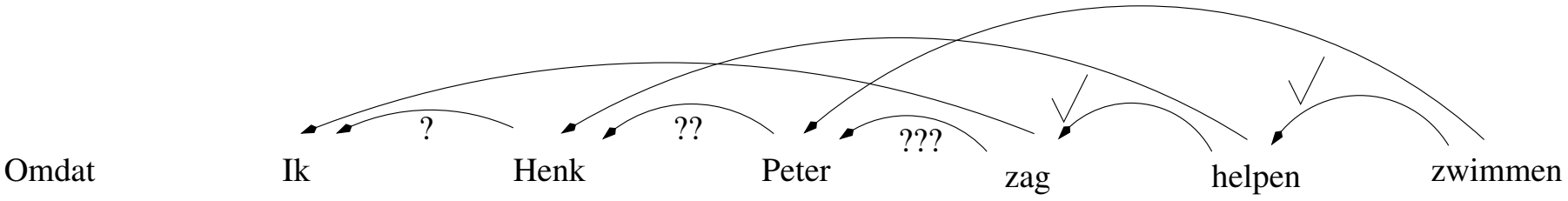
Dutch can cross many dependencies across words and phrases:



'because I saw Henk help Peter swim ...'

This example violates **projectivity**. Same for Swiss-German (Shieber, 1985)

Projectivity restored if we could assume the following:

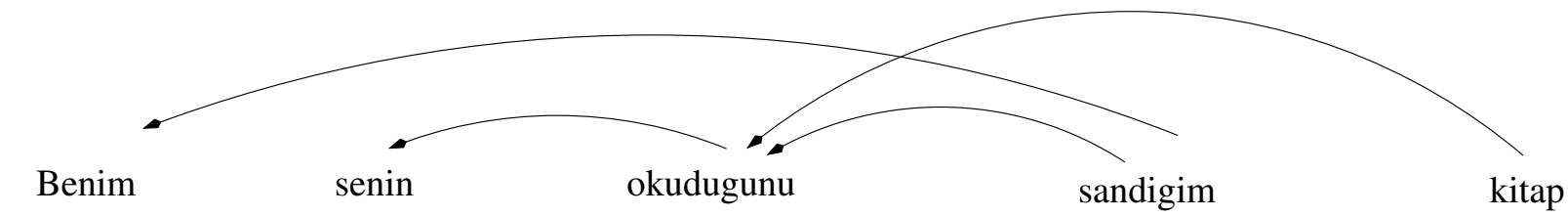


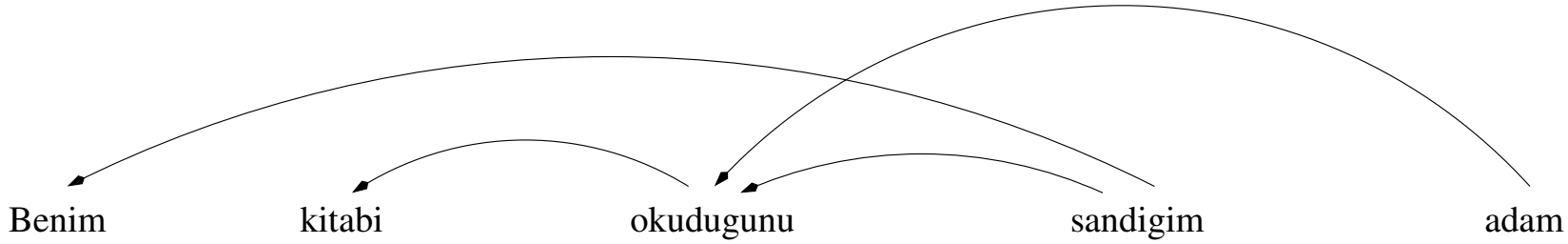
'because I saw Henk help Peter swim ...'

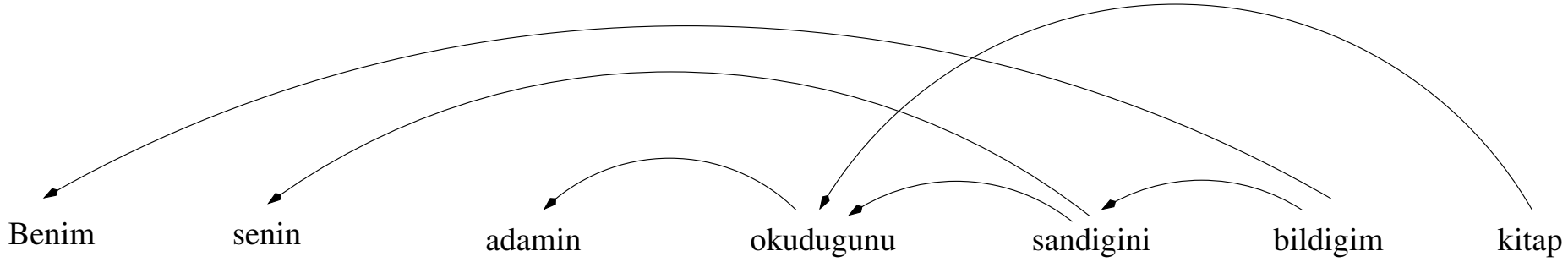
?: doubts for empirical base.

These are lexically mediated by Dutch verbs.

Turkish verbs do not cross lexically, but they can cross at surface structure:

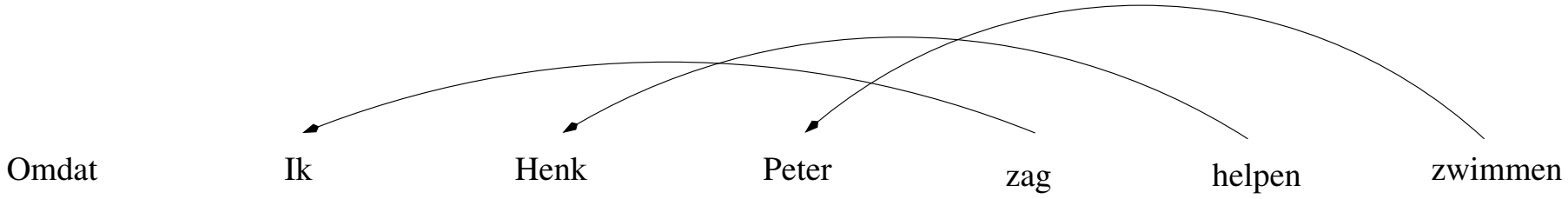

Benim
senin
okudugunu
sandigim
kitap


Benim
kitabi
okudugunu
sandigim
adam


Benim
senin
adamin
okudugunu
sandigini
bildigim
kitap

Single-headedness: every node has at most one head.

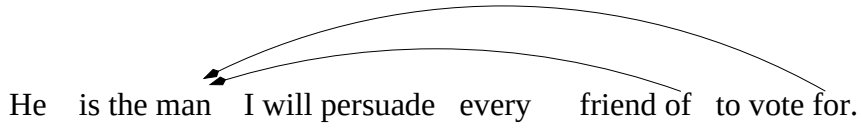
OK so far even when projectivity is not:

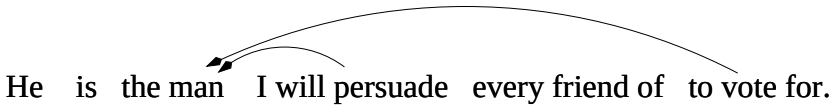


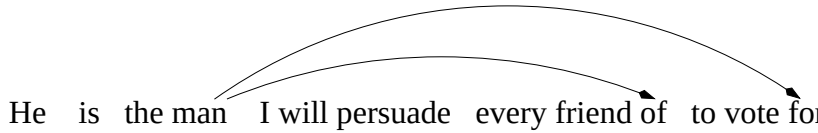
'because I saw Henk help Peter swim ...'

Not always satisfied in surface structure:

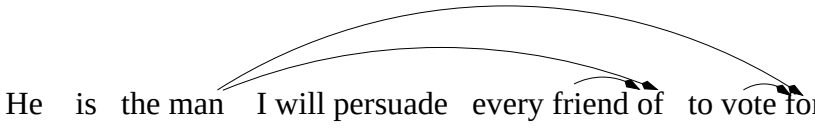
Baldridge (2002)







??



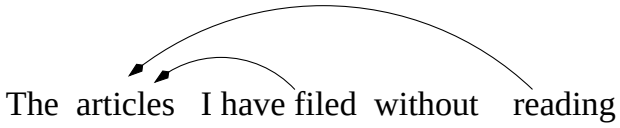
Some choice of heads across languages:

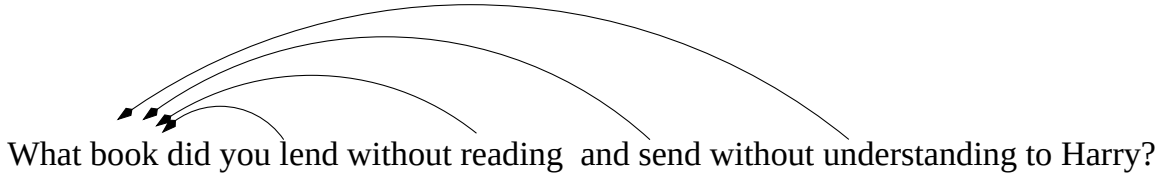
Table 2. *Annotation style (choice of head). VG = Verb group (Aux = Auxiliary verb, MV = Main verb); AP = Adpositional phrase (Ad = Adposition, N = Nominal head); NP = Noun phrase (Det = Determiner, N = Noun); SC = Subordinate clause (Comp = Complementizer, V = Verb); Coord = Coordination (CC = Coordinating conjunction, $Conj_1$ = First conjunct, $Conj_n$ = Last conjunct); NA = Not applicable*

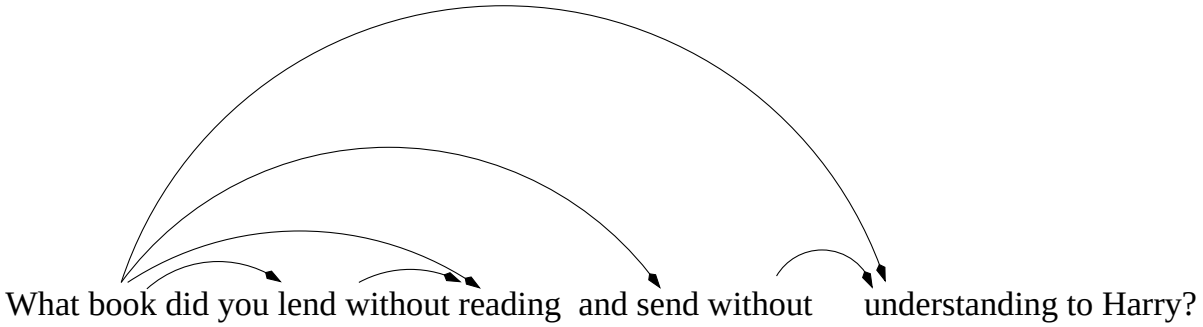
Language	VG	AP	NP	SC	Coord
Bulgarian	MV	Ad	N	Comp	$Conj_1$
Chinese	Aux	Ad	N	Comp	$Conj_1$/$Conj_n$
Czech	MV	Ad	N	V	CC
Danish	Aux	Ad	Det	Comp	$Conj_1$
Dutch	Aux	Ad	N	Comp	CC
English	Aux	Ad	N	Comp	$Conj_1$/$Conj_n$
German	Aux	Ad	N	V	$Conj_1$
Italian	MV	Ad	Det	Comp	$Conj_1$
Swedish	Aux	Ad	N	Comp	$Conj_1$
Turkish	NA	(Ad)	N	NA	$Conj_n$

Nivre et al. (2007)

The problem does not appear to be construction-specific: (Steedman, p.c.)







It is not language-specific either:

Kuhlmann and Nivre (2006): well-nestedness and some n-discontinuity provide a **good fit** with linguistic data.

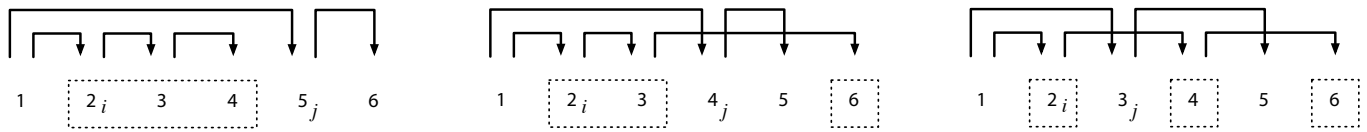
Hockenmaier and Steedman (2007) seem to suggest otherwise.

Lots of dependencies in Penn Treebank are unbounded and not projective.

Dutch and Swiss-German cannot be well-nested or n-planar.

A DG is m-**planar** if it can be split into m planar graphs.

A DG is **well-nested** if no disjoint trees interleave.



(a) $gd = 0, ed = 0, wn+$ (b) $gd = 1, ed = 1, wn+$ (c) $gd = 2, ed = 1, wn-$

Figure 3: Gap degree, edge degree, and well-nestedness

Kuhlmann and Nivre (2006)

Why these assumptions?

They make dependencies efficiently parsable.

Cognitive science of language

Kuhlmann and Nivre (2006): well-nestedness and some n-discontinuity provide a **good fit** with linguistic data.

Fitting the data seems to depend on our **semantic radar**.

How much semantics do we expect to get out from parsing?

How can competence grammars of adults and 10-year-olds keep up with intricate dependencies?

If parsing is a **reflex**, what kind of all-around computation can deliver this performance?

We need an explanatory theory of

grammars,

parsing,

dependency,

constituency.

Combinatory Categorial Grammar

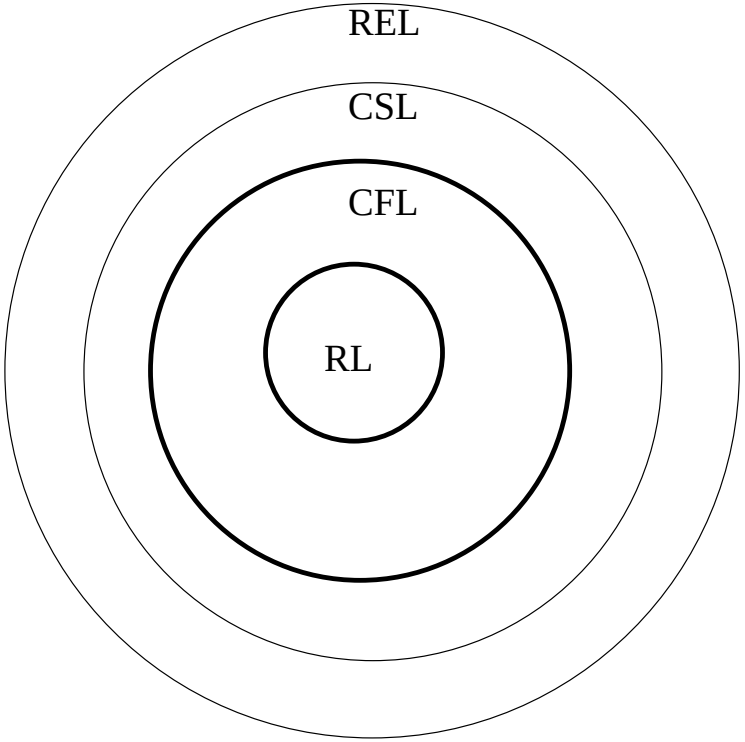
Steedman (2000)

Dependency is a relation of **predicate-argument** structure.

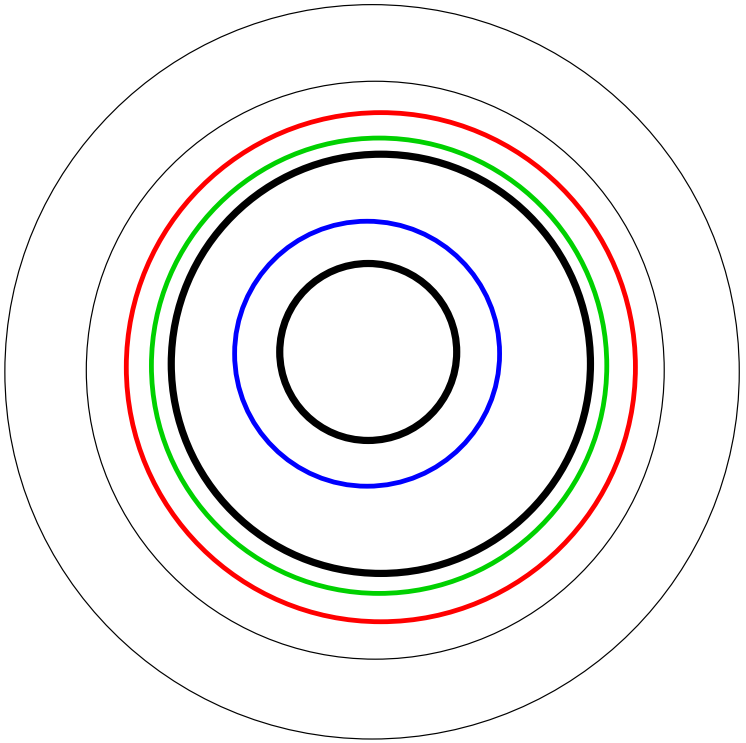
Constituency is a relation of **syntactic** structure.

All of these can be mediated **lexically** by **combinatory types**.

Parsing this way is **Linear-indexed** (a proper subset of polynomially parsable languages—MCSL; Joshi 1985)



Chomsky hierarchy



Chomsky−Joshi−Aho−Gazdar hierarchy

Linear languages

Linear−indexed languages

Mildly context−sensitive languages

Constituency

Examples from Bozsahin (2011a)

I (begin (to (try (to (avoid (reading Kafka before sleep.))))))

(I wonder) (who Kafka might have liked) (and what Wittgenstein might have written.)

(((((((((I begin) to) try) to) avoid) reading) Kafka) before) sleep.)

(I begin to try to avoid), (and you should refrain from), (reading Kafka before sleep.)

Gelin-e ben-im uyu-duğ-um-u, damad-a Ahmet'in çalış-tığ-ı-nı söyle-miş.
Bride-DAT I-AGR.1s sleep-COMP-1s-ACC groom-DAT A-AGR.3s work-COMP-3s-ACC tell-PERF
lit. 'S/he told the bride that I am sleeping and the groom that Ahmet is working.'

See Bozsahin (2002); Çakıcı (2008); Steedman (2010); Bozsahin (2011b) for CCG parsing of Turkish

[Ev-de-ki-nin-ki adam-a], [salon-da-ki çocuğ-a] sarıl-mış.

house-LOC-ki-GEN-ki man-DAT room-LOC-ki child-DAT hug-PERF

lit. ‘The one in the house’s one hugged the man, and the one in the room the child.’

e.g. ‘The friend/acquaintance of the one in the house hugged the man, and the one in the room the child.’

Dependency in constituents

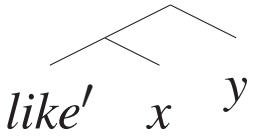
A function depends on its arguments. (dependency)

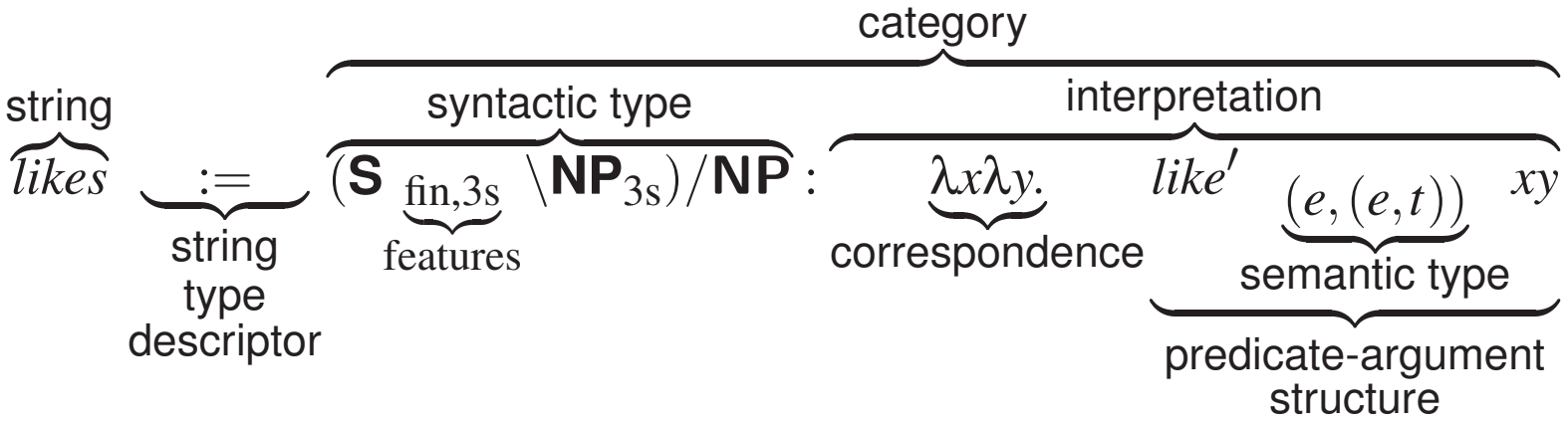
Juxtaposition xy means 'x depends on y'.

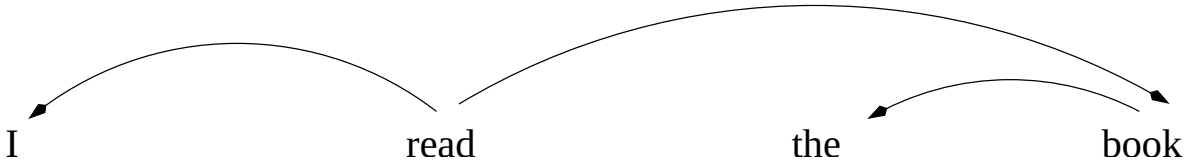
Kafka liked Milena. $like' \, milena' \, kafka'$

How dependency and constituency are put together: **Syntactic type**

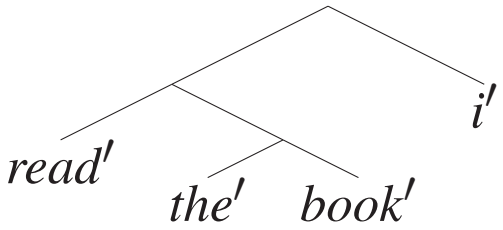
likes := $(\mathbf{S}_{\text{fin,3s}} \backslash \mathbf{NP}_{\text{3s}})/\mathbf{NP} : \lambda x \lambda y.like' xy$







I	read	the	book
NP: i'	**(S\NP)/NP**: $\lambda x \lambda y.read' xy$	**NP/N**: the'	**N**: $book'$
		NP: $the' book'$ (>)	
	S\NP: $\lambda y.read'(the' book')y$ (>)		
S: $read'(the' book')i'$ (<)			



Violation of single-headness is predicted:

persuade every friend of to vote for

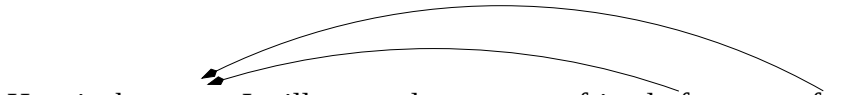
$$\dfrac{\dfrac{\overline{\textbf{(S\NP)/VP/NP}} \qquad \overline{\textbf{VP/NP}}}{\textbf{(S\NP)/NP}}\,{>}\,\textbf{S}}{}$$

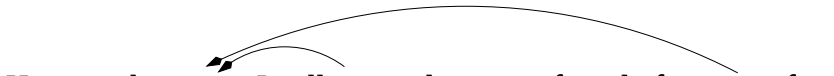
Schönfinkel's (1920/1924) $\mathbf{S} \stackrel{def}{=} \lambda f \lambda g \lambda x.fx(gx)$

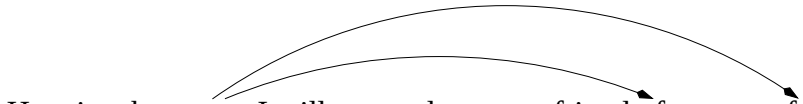
$\mathbf{S}pefo'tvf' = \lambda x.pefo'x(tvf'x)$

$$\begin{array}{lll} \textbf{(X/Y)/Z} & \textbf{Y/Z} & \rightarrow \quad \textbf{X/Z} \\ f & g & \end{array}$$

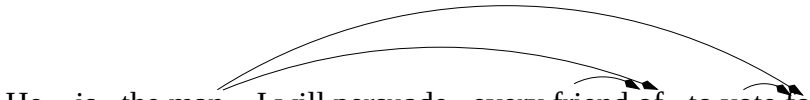
Steedman (2000) predicts **BTS** system to capture all NL dependency and constituency.


He is the man I will persuade every friend of to vote for.

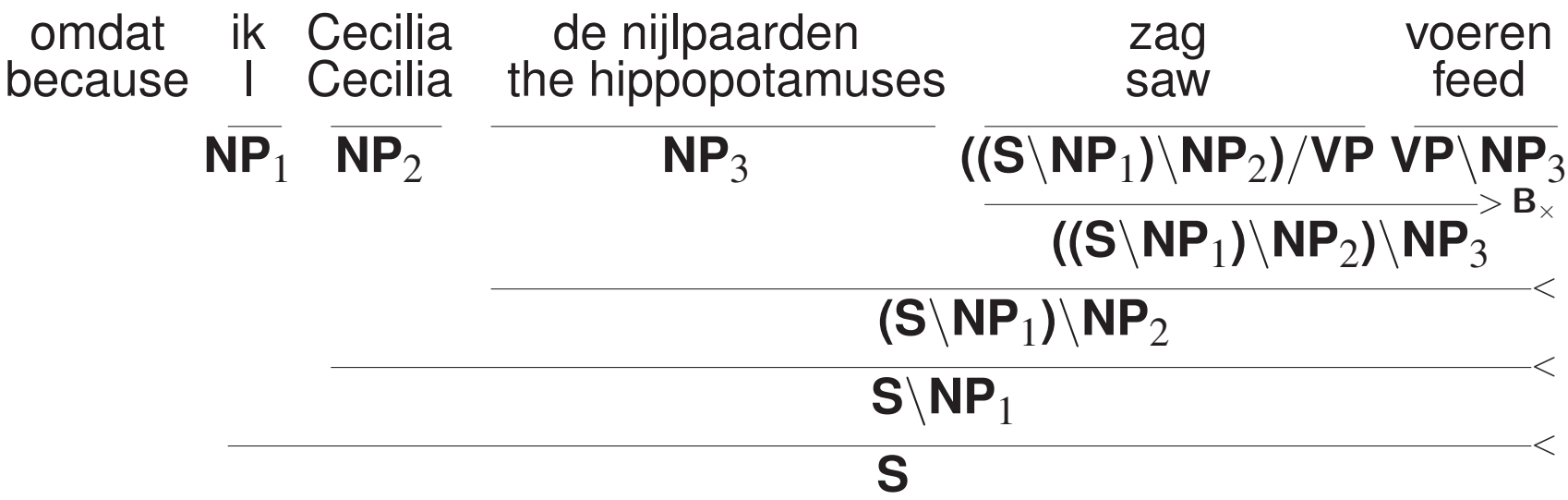

He is the man I will persuade every friend of to vote for.


He is the man I will persuade every friend of to vote for.

??


He is the man I will persuade every friend of to vote for.

Violation of projectivity and planarity is predicted:



'..because I saw Cecilia feed the hippopotamuses.'

$$\mathbf{X/Y}\quad \mathbf{Y\backslash Z}\quad \rightarrow\quad \mathbf{X\backslash Z} \qquad (>\mathbf{B}_{\times})$$

Curry and Feys's (1958) $\mathbf{B} \stackrel{def}{=} \lambda f \lambda g \lambda x.f(gx)$

Vijay-Shanker and Weir (1993) proved that CCG is

polynomially parsable

efficiently parsable.

Clark and Curran (2007); Hockenmaier and Steedman (2007) show wide-coverage parsing using CCG.

Conclusion

Dependency grammars and CCG are closely related.

Both assume that dependency arises from **surface** structure.

Surface structure (constituency) can be predicted by **combinators**.

Crossing and multi-headed dependencies are hard to compute in DG.

Dependency structures are much easier to annotate.

(labels are prone to errors)

They are translatable to CCG categories.

*References

Aho, Alfred. 1968. “Indexed Grammars—An Extension of Context-Free Grammars.” *Communications of the Association for Computing Machinery*, 15, 647–671.

Baldridge, Jason. 2002. *Lexically Specified Derivational Control in Combinatory Categorial Grammar*. Ph.D. thesis, University of Edinburgh.

Bozsahin, Cem. 2002. “The Combinatory Morphemic Lexicon.” *Computational Linguistics*, 28, 2, 145–176.

Bozsahin, Cem. 2011a. *Combinatory Linguistics*. Berlin and New York: Mouton. In review.

Bozsahin, Cem. 2011b. “Words and Types in Grammar.” *Linguistic Inquiry*, in review.

Çakıcı, Ruken. 2008. *Wide-coverage Parsing for Turkish*. Ph.D. thesis, University of Edinburgh.

Clark, Stephen, and James R. Curran. 2007. “Wide-Coverage Efficient Statistical Parsing with CCG and Log-Linear Models.” *Computational Linguistics*, 33, 4, 493–552.

Curry, Haskell Brooks, and Robert Feys. 1958. *Combinatory Logic*. Amsterdam: North-Holland.

Eryiğit, Gülşen, Joakim Nivre, and Kemal Oflazer. 2008. “Dependency Parsing of Turkish.” *Computational Linguistics*, 34, 3, 357–389.

Gazdar, Gerald. 1988. “Applicability of Indexed Grammars to Natural Languages.” In Uwe Reyle and Christian Rohrer, eds., *Natural Language Parsing and Linguistic Theories*, 69–94. Dordrecht: Reidel.

Hockenmaier, Julia, and Mark Steedman. 2007. “CCGbank: a Corpus of CCG Derivations and Dependency Structures Extracted from the Penn Treebank.” *Computational Linguistics*, 33, 3, 356–396.

Joshi, Aravind. 1985. “How Much Context-sensitivity is Necessary for Characterizing Structural Descriptions: Tree Adjoining Grammars.” In David Dowty, Lauri Karttunen, and Arnold Zwicky, eds., *Natural Language Parsing*, 206–250. Cambridge: Cambridge University Press.

Kuhlmann, Marco, and Joakim Nivre. 2006. “Mildly non-projective dependency structures.” In *Proc. of COLING-ACL*, 507–514. Sydney.

Mel’čuk, Igor A. 1988. *Dependency syntax: theory and practice*. Albany, NY: State Univ. of New York Press.

Nivre, Joakim, Johan Hall, Jens Nilsson, Atanas Chanev, Gülşen Eryiğit, Sanda Kübler, Svetoslav Marinov, and Erwin Marsi. 2007. “MaltParser: A language-independent

system for data-driven dependency parsing." *Natural Language Engineering*, 13, 2, 95–135.

Schönfinkel, Moses Ilyich. 1920/1924. "On the Building Blocks of Mathematical Logic." In Jan van Heijenoort, ed., *From Frege to Gödel*. Harvard University Press, 1967. Prepared first for publication by H. Behmann in 1924.

Shieber, Stuart. 1985. "Evidence against the Context-Freeness of Natural Language." *Linguistics and Philosophy*, 8, 333–343.

Steedman, Mark. 2000. *The Syntactic Process*. Cambridge, MA: MIT Press.

Steedman, Mark. 2010. "Gramplus: Grammar-based robust natural language processing." FP7 Project (Ideas), Edinburgh University, 2010–2013.

Tesnière, Lucien. 1959. *Éléments de syntaxe structurale*. Paris: Editions Klincksieck.

Vijay-Shanker, K., and David Weir. 1993. "Parsing Some Constrained Grammar Formalisms." *Computational Linguistics*, 19, 591–636.