Formal & Computational Aspects of

Dependency Grammar

– Heads, dependents, and dependency structures –

GEERT-JAN M. KRUIJFF

COMPUTATIONAL LINGUISTICS
UNIVERSITY OF THE SAARLAND
SAARBRÜCKEN GERMANY

⟨GJ@COLI.UNI-SB.DE⟩

Goals

- Introduce the core notions of dependency grammar: **head**, **dependent**, and **dependency structures**.
  - What makes a head a head; endo- and exocentric constructions.
  - Problems, non-problems with the head/dependent asymmetry.
  - Dependency structures: trees, or graphs?
- Give a formal definition of dependency grammars after Gaifman (1965) and Robinson (1970), to investigate the generative strength argument.
  - Regular (RDG), context-free (CFDG) and indexed dependency grammars (IDG).
  - Pollard's mildly context-sensitive Head Grammars ($CFDG \subset HG \subset IDG$).

Contents

A remark about constituency vs. dependency

- Constituency (cf. structural linguists like Bloomfield, Harris, Wells) is a *horizontal* organization principle: it groups together **constituents** into **phrases** (larger structures), until the entire sentence is accounted for.
  - Terminal and non-terminal (phrasal) nodes.
  - *Immediate constituency*: constituents need to be adjacent (modelled by CFPSG).
  - *Discontinuous constituency*: meaningful units may not be adjacent (noted already in 1943!).
  - *Flexible constituency*: "phrasal re-bracketing" (associativity: long development in categorial grammar).
- Dependency is an **asymmetrical relation** between a **head** and a **dependent**, i.e. a *vertical* organization principle.

A remark about constituency vs. dependency

- Heads and dependents are related *immediately*, there are no nonterminals.
- Many dependency grammarians have therefore claimed that DG is preferable over constituency: Dependency structures are "more economic" due to their lack of nonterminals, e.g. (Mel'čuk, 1988; Sgall and Panevová, 1989).
- This is a non-argument, for various reasons:
  - Dependency and constituency describe *different dimensions*.
  - A phrase-structure tree is closely related to a *derivation*, whereas a dependency tree rather describes the *product* of a process of derivation.
  - Usually, given a prase-structure tree, we can get very close to a dependency tree by constructing the transitive collapse of headed structures over nonterminals.
- Constituency and dependency are not adversaries, they are complementary notions. Using them together we can overcome the problems that each notion has individually.

The notion of *head*

- What makes a head a head? If we know what makes a head a head, we can also tell what makes a dependent a dependent (namely "everything" that's not a head).
- We can come up with several criteria, including for example the following (Fraser, 1994):
  1. A head determines whether a dependent is optional or obligatory (as modifier of the head), not vice versa.
  2. A head subcategorizes for a dependent, and not vice versa.
  3. The head determines which inflectional form of a dependent occurs, and not vice versa.
  4. The head identifies a meaningful object that the meaning of the dependent contributes to (specifies further), and not vice versa.
- No single criterion provides a necessary and sufficient condition for dependency – witness the many dependency grammar frameworks!

Zwicky's (1985) criteria for being a head

1. **Hyponomy**: A construction as a whole will denote a hyponym (subtype) of the head of the construction, not of the dependents.

   (1) *extremely heavy books* denotes a subtype of *books*, not of *heavy*.

   (2) *dances beautifully* denotes a subtype of *dancing*, not of *beautifully*.

2. **Subcategorization**: A head subcategorizes for a dependent, not vice versa.

   (3) In *She danced a tango*, *dance* subcategorizes for *tango*. It is not a feature of *tango* that it should occur in a construction with *dance*: E.g. *They did a tango*.

3. **Morphosyntactic marking**: Typically, the head bears the inflection indicating the relationship of the construction to other constructions.

   (4) In *extremely heavy books*, the head *books* indicates the plurality of the dependent.

Zwicky's (1985) criteria (cont'd)

4. **Government**: The head determines the inflectional form of a dependent.

   (5) In *dances beautifully*, it is the head *dances* that indicates third person, and which determines the inflection of the subject.

5. **Concord**: The element which other words agree with is the head.

   (6) French *un livre égyptien* ("an Egyptian book") is different from *une livre égyptienne* ("an Egyptian pound"). The different demands on gender agreement reveal not only that we have two different lexemes, but also that *livre* is the head in both cases.

6. **Distribution equivalence**: If there is one element in the construction which has a parallel distribution to the construction as a whole, it is the head.

   (7) Whenever we find *extremely heavy books*, we also find *books*.

Zwicky's (1985) criteria (cont'd)

7. **Obligatoriness**: The obligatory element in a construction is the head.

   (8) To be of the type of *dances beautifully* there must be a verb, but not necessarily an adverb. Hence the verb is the head.

8. **Characterization**: The head lends the construction its grammatical category.

   (9) It is the verb *dances* that makes *dances beautifully* a Verb Phrase.

- **Question**: If we construct a word group, where does the head come from?
- Bloomfield defined an *endocentric* construction as one whose distribution is identical with one or more of its constituents – that is, if in a composition with a head, the head provides the head of the overall construction.
- *Exocentricity* is defined as the negative of endocentricity.
- For Bloomfield and some others, the heads as defined by Zwicky's criteria Distribution Equivalence and Obligatoriness are "centers".
  - Only those constructions which have the same distribution as the head were considered to have a center, though – in other words, only the endocentric constructions.
  - This still holds in Hudson's Word Grammar, which restricts every phrase to be endocentric. E.g. a gerund cannot be an NP headed by a verb.
  - Exocentric constructions like prepositional phrases or sentences were considered not to have centers.

- From a more functional (categorial) perspective, we can argue that the problem is again the conflation of two different dimensions: Categorial function/argument structure, and dependency. Cf. also (Kruijff, 2001).

- **Example**:

  - Consider the slashes $\{\backslash, /\}$ to bear an indication of *dependency* $\prec, \succ$ pointing from head to dependent; (Moortgat and Morrill, 1991; Moortgat and Oehrle, 1994; Hepple, 1997; Kruijff, 2001).
  - $John \vdash NP + walks \vdash NP \backslash_{\prec} S$ gives us $(John \circ_{\prec} walks) \vdash S$, with $John$ as a dependent of $walks$.
  - For an adverb we don't get $S \backslash S$ but $S \backslash_{\succ} S$: The adverb is the function, taking the verb as an argument – but the dependency shows that the adverb is the dependent, and the verb being composed with is the head of the construction.
  - (Kruijff, 2001) calls $S \backslash_{\succ} S$ an *exocentric category*, and $NP \backslash_{\prec} S$ an *endocentric category*.

Arguments against headedness

- Does *every* construction have a head? Does every construction have to have a *single* head?
- Mel'čuk (1988)(p.24ff) discusses three types of criticism:

1. **Double dependency**: "A wordform can depend simultaneously on two different wordforms; hence, the idea of a unique head is unacceptable."
2. **Mutual dependency**: "The main verb and its grammatical subject are mutually dependent: the verb represents the clause but the subject controls the verb's form."
3. **No dependency**: "Some constructions have no syntactic head, e.g. coordination where there is no dependency between the conjoined items."

Arguments against headedness (cont'd)

- **Double dependency**

  > "A wordform can depend simultaneously on two different wordforms; hence, the idea of a unique head is unacceptable."

- **Examples**

  (10) *Wash the dish* **clean**.

  The word *clean* supposedly depends on *the dish* as well as on *wash*.

  (11) a. *He pounded the box* **flat**.
  b. *He sanded the stick* **smooth**.

  (12) a. *We heard Mary* **singing**.
  b. *The watchdog heard us* **coming**.

- Even though semantically the (ad)verb may depend on the directly preceding noun and the verbal head, there is only one syntactic dependency.

Arguments against headedness (cont'd)

- **Mutual dependency**

> "The main verb and its grammatical subject are mutually dependent: the verb represents the clause but the subject controls the verb's form."

- **Examples**

(13) *The child* **is** *playing* versus *The children* **are** *playing*

"The verb governs the sentence, but the subject determines the form of the verb".

- There is a difference between morphological and syntactic dependency.

Arguments against headedness (cont'd)

- **No dependency**

  "Some constructions have no syntactic head, e.g. coordination where there is no dependency between the conjoined items."

- **Examples**

(14) *He stood up and gave me the letter.*
$\neq$ *He gave me the letter and stood up.*

(15) *Go to bed, or I'll spank you!.*
$\neq$ **I'll spank you, or go to bed!*

(16) *not only a good worker but also a nice man*
$\neq$ *not only a nice man but also a good worker*

- Semantically symmetric, but not (always) syntactically reversible.

Problems with dependency

- Phenomena like coordination and apposition, but possibly also multiword expressions, do present a problem for dependency as such.
- People have solved these problems to some extent.
  - Involving meaning-related features in syntax.
  - Delineating the class of problematic constructions, and proposing a uniform solution to dealing with these constructions.
- Particularly when looking at coordination, this results in interesting new perspectives though!

Problems with dependency (cont'd)

- We can distinguish constituent/non-constituent (Sag et al., 1985) from dependent/independent coordination constructions (Panevová, 1974).
- In some languages, only particular inequal-dependent coordinations are allowed (even though they are constituent-coordinations).
- For example, in Czech, coordinating Aim and Concession is questionable, and Cause and Condition is ungrammatical (Panevová, 1974)(p.14).
- An explanation for the (un)grammaticality of such coordinations can be based on the *interpretative import* of dependency relations.

Dependency structures

- If we put heads and dependents together, we get **dependency structures**.
- Different constraints that can hold for dependency structures:

1. **There should be one independent element.**
   In a dependency structure, every word must depend on some other word with the exception of one element – the root.
2. **The dependency structures must be connected.**
   All the words should be connected by one and the same structure.
3. **Every dependent must have a unique head.**
   Each dependent must depend on exactly one head, except for the root.
4. **Heads and dependents must be adjacent.**
   The way the expression's words are connected as heads and dependents in the dependency structure must project to a linear order of words that matches the expression.

Trees or graphs?

- Theories consider dependency structures to be *trees* or *graphs*, depending on how they apply the above constraints.
- If we relax the "unique head" constraint, we obtain graphs rather than trees.
- The dependency grammar in the EUROTRA system (Johnson et al., 1985), or Hudson's Word Grammar (1984; 1990) allow for dependents to have multiple heads.
- Done for semantic reasons, e.g. to deal with subject sharing in "Kathy likes writing."

Projective or non-projective structures?

- Relaxing the adjacency/projectivity constraint is relevant primarily at the level of surface structure.
- Non-projectivity at the level of surface structure enables one to give a uniform account of local/long-distance dependencies. For languages with a freer word order projectivity is thus hard to maintain.
- Dependency grammars allowing for non-projective trees are for example the computational formalisms (closely tied to parsing) of Covington (1990), Holán, Kuboň and Plátek (Holan et al., 2001; Kuboň, 2001), Duchier and Debusmann (2001), or the logical formalism of Kruijff (2001).
- To some extent, see also Hudson's Word Grammar, or Sgall et al's Functional Generative Description (1986; 1995).
  - Their structures are mostly concerned with meaning so the issue of adjacency only arises in the generation.
  - FGD's tectogrammatical representations *are* projective, but to "deep word order".

A formal definition of dependency grammars

- The literature contains numerous formalizations of dependency grammars, derived from the work by Robinson (1970) and Gaifman (1965).
- Here we look at different forms that DG rules can take.
  - We want to get a feel for what a grammar looks like.
  - Chomsky showed that the generative strength of a PSG depends on the forms that its rules take – we can do the same for DG (Fraser, 1994).
  - By showing we can move beyond CFG this way, we defuse the old (mistaken) criticism against DG's adequacy.

A formal definition of dependency grammars

- We do not deal with the computational complexity of *recognition* here.
  - See (Neuhaus and Bröker, 1997) for general results ($\mathcal{NP}$-completeness), matching similar complexity proofs for the frameworks developed in (Duchier and Debusmann, 2001) or (Kruijff, 2001).
  - See (Baldridge and Kruijff, 2002) for a polynomially parseable, mildly context sensitive grammar with a dependency-based semantics.
  - See also work by Kuboň *et al.*, e.g. (Holan et al., 2001).

Gaifman's (1965) formalism

- Rules in Gaifman's (1965) formalism take three basic forms.

  (1) $X(Y_1, \ldots, Y_i, *, Y_{i+1}, \ldots, Y_n)$
  A word of category X governs words of categories $Y_1, \ldots, Y_n$, occurring in the given order. The position of X (as the head) is indicated by the $*$.

  (2) $X(*)$
  A word of category X may occur without any dependents.

  (3) $*(X)$
  A word of category X may occur without depending on any other element.

- The trees resulting from (1) are essentially ordered trees of depth one, whereby nodes are labelled.

- As proven by Gaifman, (1)+(2)+(3) generate context-free languages. The rules formalize **context-free dependency grammars**.

Gaifman's (1965) formalism (cont'd)

- Trees are combined by identifying a head of category Y with a dependent (of another head!) with the matching category.
  - Gaifman's formulization thus gives us projective dependency trees, where the order of the dependents is completely specified by the rules.
- Implicit in (1)-(3) is a linear use of heads:
  - Each rule may apply at most once to each head word (Fraser, 1994).
  - Generalizing this to every *word*, we can call this the *single application constraint*.
  - In DGs where this constraint applies, the number of dependents a head may have is bound by the grammar (lexicon) and is necessarily finite.
  - Without this constraint, there need not be an upper bound to the number of dependents a head may have.

Regular dependency grammars

- We can replace rules of type (1) by rules of the following forms:

(1r1) $X(Y_1, *)$

(1r2) $X(*, Y_1)$

- Together with rules (2) and (3), we obtain a grammar in which each head can have at most one dependent.
- The set of languages that (1r1)+(1r2)+(2)+(3) generate is properly contained in the language generated by Gaifman's original formulation.
- (1r1)+(1r2)+(2)+(3) gives a Gaifman-style formalization of **regular dependency grammars**:
  - (1r1)+(1r2)+(2)+(3) generate exactly the regular languages.
  - These grammars cannot generate the string-copying language $a^n b^n$.
  - (1r1)+(2)+(3) corresponds to left-linear regular PSGs, (1r2)+(2)+(3) corresponds to right-linear regular PSGs.

Stronger dependency grammars

- We can also obtain more powerful dependency grammars on the basis of (1) – we are interested in dependency grammars that are stronger than context-free.
- From the time that the Chomsky hierarchy was first established, people supposed that the natural languages, when considered as string sets, would be somewhere between context-free and context-sensitive:
  - Context-sensitive languages are too inclusive, e.g. they include $\{a^n b^{n!}\}$ or $\{a^n \ : n \ is \ prime\}$.
  - Context-free languages are too weak (Shieber, 1985).
- Research in formal language theory has spawned a family of grammars that generates "mildly context-sensitive" languages (Aravind Joshi).
- We look at *indexed dependency grammars*, based on indexed grammars, and at Pollard's *Head Grammars*.

Formulating indexed dependency grammars

- Indexed grammars were invented by Aho (1968), and are >CF.
- According to Hopcroft & Ullman (1979)(p.389), "of the many generalizations of context-free grammars that have been proposed, a class called 'indexed' appears the most natural in that it arises in a wide variety of contexts."
- An indexed grammars differs from a CFG in that nonterminal symbols may carry a sequence of indices chosen from a given finite set, and its productions allow for adding or removing these indices during a derivation – the index sequence functions like a pushdown stack (Partee et al., 1990)(p.536).
- In linguistics, Gazdar (1988) was the first to exploit (linear!) indexed grammars (LIGs) for modelling long-distance dependencies.

Formulating indexed dependency grammars (cont'd)

- **Sketch of the landscape**:
- If we restrict LIGs to being either left-linear (LLIG) or right-linear (RLIG) then we stay with context-free languages (Aho, 1968).
- R/LLIGs have strong connections to Schabes & Waters' (1995) Tree-Insertion Grammars (TIGs), cf. (Michaelis and Wartena, 1999). TIGs are context-free.
- TIGs are related to TAGs; TAGs generate mildly context sensitive languages, just like LIGs (Vijayashanker and Weir, 1994). TAGs led to D-Tree grammars (Rambow et al., 2001).
- We explore:

  $CFG = RLIG = LLIG$ (Aho, 1968)
  $\subset$ $Head\ Grammar = TAG = LIG$ (Vijayashanker and Weir, 1994)
  $\subset$ $IDG \subseteq IG$ (Fraser, 1994)+(Partee et al., 1990)
  $\subset$ CSG (Partee et al., 1990).

Dependency grammars with stacks

- To define indexed dependency grammars, we need *stacks*.
- Assume a **stack**: $[\,]$ denotes the empty stack, $[i]$ is a stack containing the single symbol i, and $[i, \ldots]$ is the stack with i as its topmost element.
- We can now define *push* and *pop* rules.
  - A push rule (1push*) copies the contents of a stack from head to dependent, and pushes i onto the top of the dependent's stack.

    (1push1) $H[\ldots](\ldots, D[i, \ldots], \ldots, *, \ldots)$

    (1push2) $H[\ldots](\ldots, *, \ldots, D[i, \ldots], \ldots)$
  - A pop rule pops the top of a head's stack and then copies the remainder of the head's stack to the dependent's stack.

    (1pop1) $H[i, \ldots](\ldots, D[\ldots], \ldots, *, \ldots)$

    (1pop2) $H[i, \ldots](\ldots, *, \ldots, D[\ldots], \ldots)$

- The indexed dependency grammar we present here (Fraser, 1994) generates the language $a^n b^n c^n$ (>CF):

$$
\begin{array}{ll}
(a[a]) & b[a,\ldots](, b[\ldots]) \\
c[a](*) & c[a,\ldots](*, c[\ldots]) \\
b[a](*) & a[\ldots](*, a[a,\ldots]) \\
 & a[\ldots](*, b[\ldots], c[\ldots])
\end{array}
$$

- Using an IDG, we can generate the CFLs by simply omitting the indices (stacks).
- Indices also enable us to define right- or left-linear IDGs, just like in Aho's original proof for $LinIL = CFL$ in (1968); see also (Partee et al., 1990)(p.540).
- At the same time, $IL \subset CSL$: "The indexed languages thus provide us, at least for the moment, with a kind of upper bound for syntactic phenomena." (Gazdar & Pollum)

“Mildly context-sensitive dependency grammars”

- Pollard (1984) defines **Head Grammars**, which present an extension of CFGs.
- Vijayashanker & Weir (1994) show that Head Grammars generate exactly the same class of string languages as TAG does, i.e. the mildly context sensitive languages.
- In Head Grammar, each string contains a distinguished symbol called the *head* of that string.
- Formally, a headed string is a tuple of a string and a natural number indicating the position of the head in the string.
- Head Grammar has *concatenation* and *wrapping* operators to compose new strings.

- Two headed strings can be *concatenated*, in which case either the left or the right head becomes the head of the resulting string.a
- *Wrapping* operations separate a string at its head, and then insert another string.

  There are four wrapping operations: the second string is inserted eitehr left or right of the head of the first string, and the head of either the first or the second string becomes the head of the resulting string.
- Rules take the form $A \rightarrow \alpha_1$ or $A \rightarrow f(\alpha_1 ... \alpha_n)$ with f a concatenation or wrapping operator, e.g.
  - $C_1(w_1 \uparrow w_2, u_1 \uparrow u_2) \longrightarrow w_1 \uparrow w_2 u_1 u_2$
  - $C_2(w_1 \uparrow w_2, u_1 \uparrow u_2) \longrightarrow w_1 w_2 u_1 \uparrow u_2$
  - $W(w_1 \uparrow w_2, u_1 \uparrow u_2) \longrightarrow w_1 u_1 \uparrow u_2 w_2$

In conclusion

- We have seen criteria that establish interpretations of the basic notions of dependency grammar: heads, dependents, and dependency structures.
- Dependency grammar frameworks have often adopted different versions of these criteria: dependency structures as *graphs* or *trees* (uniqueness of head), projectivity.
- After Gaifman and Robinson, we constructed formal dependency grammars whose strength varied from regular to greater than context-free – thus, defusing the old argument that dependency grammars *exactly* generate the context-free languages (Gross, 1964) and were therefore inadequate.

References

Alfred V. Aho. 1968. Indexed grammars – an extension of context-free grammars. *Journal of the Association for Computing Machinery*, 15:647–671.

Jason Baldridge and Geert-Jan M. Kruijff. 2002. Coupling CCG and hybrid logic dependency semantics. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*, Philadelphia, Pennsylvania.

Michael A. Covington. 1990. Parsing discontinuous constituents in dependency grammar. *Computational Linguistics*, 16:234–236.

Denys Duchier and Ralph Debusmann. 2001. Topological dependency trees: A constraint-based account of linear precedence. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL 2001)*, Toulouse, France.

Norman M. Fraser. 1994. Dependency grammar. In R.E. Asher, editor, *The Encyclopedia of Language and Linguistics*, pages 860–864. Pergamon Press, Oxford, United Kingdom.

Haim Gaifman. 1965. Dependency systems and phrase-structure systems. *Information and Control*, 8(3):304–337.

Gerald Gazdar. 1988. Applicability of indexed grammars to natural language. In Uwe Reyle and Christian Rohrer, editors, *Natural Language Parsing and Linguistic Theories*, pages 69–94. D. Reidel Publishing Company, Dordrecht, Boston, London.

Maurice Gross. 1964. On the equivalence of models of language used in the fields of mechanical translation and information retrieval. *Information Storage and Retrieval*, 2(1):43–57.

Mark Hepple. 1997. A dependency-based approach to bounded and unbounded movement. In *Proceedings of the Fifth Meeting on Mathematics of Language (MOL-5)*. DFKI-D-97-02.

Tomáš Holan, Vladislav Kuboň, Karel Oliva, and Martin Plátek. 2001. Word-order relaxations & restrictions within a dependency grammar. In *Proceedings of International Workshop on Parsing Technologies*, pages pp.237–240, Taiwan. Tsinghua University Press.

J.E. Hopcroft and J.D. Ullman. 1979. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, Reading, Massachusetts.

Richard Hudson. 1984. *Word Grammar*. Basil Blackwell, Oxford, United Kingdom.

Richard Hudson. 1990. *English Word Grammar*. Basil Blackwell, Oxford, United Kingdom.

Rod Johnson, Maghi King, and Louis des Tombe. 1985. EUROTRA: A multilingual system under development. *Computational Linguistics*, 11:155–169.

Geert-Jan M. Kruijff. 2001. *A Categorial-Modal Logical Architecture of Informativity: Dependency Grammar Logic & Information Structure*. Ph.D. thesis, Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic, April.

Vladislav Kuboň. 2001. *Problems of Robust Parsing of Czech*. Ph.D. thesis, Faculty of

Mathematics and Physics, Charles University, Prague, Czech Republic.

Igor A. Mel'čuk. 1988. *Dependency Syntax: Theory and Practice*. SUNY Press, Albany NY.

Jens Michaelis and Christian Wartena. 1999. Ligs with reduced derivation sets. In Gosse Bouma, Erhard Hinrichs, Geert-Jan M. Kruijff, and Richard T. Oehrle, editors, *Constraints and Resources in Natural Language Syntax and Semantics*, pages 263–279. CSLI Publications, Stanford CA.

Michael Moortgat and Glyn Morrill. 1991. Heads and phrases: Type calculus for dependency and constituent structure. Unpublished manuscript. Available from http://www-lsi.upc.es/˜morrill/.

Michael Moortgat and Richard T. Oehrle. 1994. Adjacency, dependency and order. In *Proceedings of the Ninth Amsterdam Colloquium*.

Peter Neuhaus and Norbert Bröker. 1997. The complexity of recognition of linguistically adequate dependency grammars. In *Proceedings of the 35th Annual Meeting of the ACL and 8th Conference of the EACL*, pages 337–343, Madrid, Spain.

Jarmila Panevová. 1974. On verbal frames in functional generative description I. *Prague Bulletin of Mathematical Linguistics*, 22:3–40.

Barbara H. Partee, Alice Ter Meulen, and Robert E. Wall. 1990. *Mathematical Methods in Linguistics*. Kluwer Academic Publishers, Dordrecht, Boston, London.

Vladimír Petkevič. 1995. A new formal specification of underlying structures. *Theoretical Linguistics*, 21(1):7–61.

Carl Pollard. 1984. *Generalized Phrase Structure Grammars, Heads Grammars, and Natural Language*. Ph.D. thesis, Stanford University. Ph.D. thesis.

Owen Rambow, K. Vijay-Shanker, and David Weir. 2001. D-tree substitution grammars. *Computational Linguistics*, 27(1):87–122.

Jane J. Robinson. 1970. Dependency structures and transformational rules. *Language*, 46(2):259–285.

Ivan A. Sag, Gerald Gazdar, Thomas Wasow, and Stephan Weisler. 1985. Coordination and how to distinguish categories. *Natural Language and Linguistic Theory*, 3:117–171.

Yves Schabes and Richard C. Waters. 1995. Tree Insertion Grammar: A cubic-time, parsable formalism that lexicalizes context-free grammar without changing the trees produced. *Computational Linguistics*, 21(4):479–513.

Petr Sgall and Jarmila Panevová. 1989. Dependency syntax - a challenge. *Theoretical Linguistics*, 15(1):73–86.

Petr Sgall, Eva Hajičová, and Jarmila Panevová. 1986. *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects*. D. Reidel Publishing Company, Dordrecht, Boston, London.

Stuart Shieber. 1985. Evidence against the context-freeness of natural language. *Linguis-*

tics and Philosophy, 8:333–343.

K. Vijayashanker and David Weir. 1994. The equivalence of four extensions of context-free grammar. *Mathematical Systems Theory*, 27:511–546.