HPSG-Based Model-Theoretic Dependency Grammar

Adam Przepiórkowski (ICS Polish Academy of Sciences and University of Warsaw)

Introduction HPSG is the only major contemporary linguistic framework with an explicit model theory (see, e.g., Richter 2024 and references therein); while Lexical Functional Grammar (LFG) is also sometimes mentioned as being "model-based" (Kaplan 1995: 11), no comprehensive model theory has ever been proposed for LFG (but see Johnson 1988 for an early partial attempt) and its status within the model-theoretic vs. generative-enumerative divide (Pullum and Scholz 2001) is in fact not fully clear (Przepiórkowski 2023).

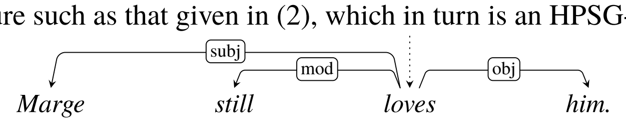
Another major approach whose some manifestations are occasionally claimed to be "constraint-based" – a term sometimes taken to be (near-)synonymous with "model-theoretic" – is the dependency grammar (DG) approach; see, e.g., Hudson 2024: 1546 about Word Grammar (Hudson 1984, 2010). However, unlike in LFG, statements within theoretical-linguistic DGs are often not formalised or – as in the case of Meaning–Text Theory of Mel'čuk 1988, 2009 – they are only partially formalised. While more precise formalisations of constraints are present in computationally-oriented work in 1990s and early 2000s, starting with Maruyama's (1990) Constraint Dependency Grammar, this Natural Language Processing (NLP) work has not influenced theoretical approaches, despite relative linguistic sophistication of some of them, esp., Weighted Constraint Dependency Grammar (Schröder *et al.* 2000; see also By 2004) and Extensible Dependency Grammar (Debusmann *et al.* 2004). Moreover, just as in the case of LFG, none of these theoretical and computational DG approaches developed – or even hinted at – an explicit model theory of the kind assumed in HPSG.

The aim of this paper is to propose such a model theory for dependency grammars. This task is timely, as dependency approaches have been on the rise during the last couple of decades, and they have become clearly dominant both in NLP and in psycholinguistics. Within pre-LLMs NLP, dependency parsing became popular already in late 2000s (see, e.g., Kübler *et al.* 2009), while within the current Large Language Models era, dependency syntax is routinely assumed in investigations of the actual linguistic competence of LLMs (e.g., Hewitt and Manning 2019) and dependency treebanks are used in the construction of linguistic test-suites for LLMs (e.g., Jumelet *et al.* 2026). Similarly, psycholinguistic processes are currently commonly explained with recourse to dependency representations of syntax (e.g., Futrell and Levy 2017 and Futrell *et al.* 2020), and a recent MIT Press syntax textbook aimed at cognitive scientists explicitly advocates dependency syntax (Gibson 2025).

The Model-Theoretic Dependency Grammar (MTDG) proposal sketched below draws heavily on HPSG and, in particular, on the RSRL-based model theory of HPSG (Richter 2004, 2007), but it is not an attempt at turning HPSG into a DG, as postulated in Hudson 2024 (see also Kahane 2009). Rather, it simply adopts – and adapts – various HPSG mechanisms to the needs of formalising DGs.

Models Consider the simple dependency tree in (1). On the proposed approach, it is a shorthand for a fuller structure such as that given in (2), which in turn is an HPSG-like representation of the model depicted in (3).

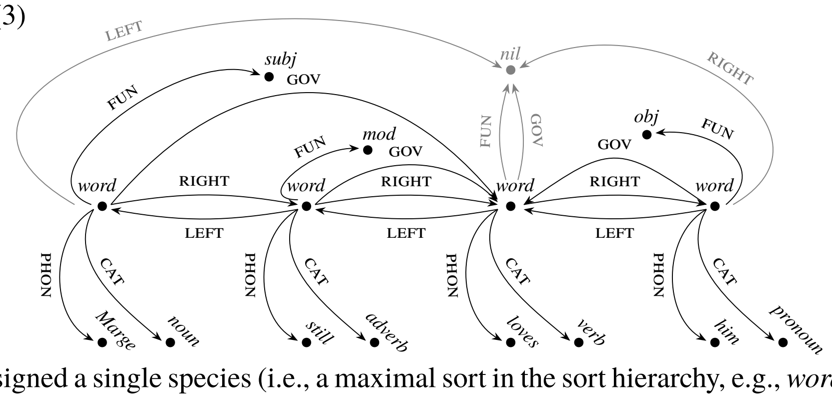
(1)



(2)

	[1]	[2]	[3]	[4]
	word	*word*	*word*	*word*
PHON	*Marge*	*still*	*loves*	*him*
CAT	*noun*	*adverb*	*verb*	*pronoun*
FUN	*subj*	*mod*	*nil*	*obj*
GOV	[3]	[3]	*nil*	[3]
LEFT	*nil*	[1]	[2]	[3]
RIGHT	[2]	[3]	[4]	*nil*

(3)



The structure in (2) represents the bare minimum of the information commonly assumed to be contained in dependency trees: the forms of the words (PHON), their categories (CAT), the actual dependency tree (encoded here with GOV for governor and FUN for the grammatical function, i.e., dependency label), and the linear order of words (encoded here a little redundantly with LEFT and RIGHT, which indicate left and right adjacency).

As in standard HPSG formalised in RSRL, an Attribute-Value Matrix (AVM) representation such as that in (2) corresponds to a specific configuration of objects in a model, as presented in (3). Also as in HPSG, each object in a model is assumed to be assigned a single species (i.e., a maximal sort in the sort hierarchy, e.g., *word* or *noun*), and attributes are interpreted as partial functions on the set of objects in the model. However, unlike in standard RSRL (Richter 2004, 2007), but as proposed within HPSG in Przepiórkowski 2021 and assumed elsewhere in model-theoretic syntax, lean models are adopted in MTDG, each corresponding to a single utterance (rather than the whole language). That is, the configuration shown in (3) represents the complete model of the utterance *Marge still loves him*.

In formal logic, a model is an interpretation in which a given set of formulae is true. The formal definition of interpretation adopted in MTDG is the same as in standard RSRL: it is a quadruple $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$ such that U is a set of objects, S is a function assigning maximal sorts to objects, A is an interpretation of attributes, and R is an interpretation of relation symbols (such as `component` in (6) below).

As in RSRL, the sort hierarchy, attributes, and relation symbols are defined in MTDG in the signature (not presented formally here because of lack of space). For example, (2)–(3) presupposes the existence of maximal sorts *word*, *subj*, *pronoun*, etc., as well as the appropriateness of the attributes PHON, CAT, etc., for the sort *word*. An additional assumption made here is that there is a special maximal sort, *nil*, which is not a subsort of any other sort and which may in principle be the value of any attribute, regardless of the appropriateness conditions stated explicitly in the signature. The role of this special sort is to make attributes optional in principle; when the value of an attribute is *nil*, it is understood as in a sense absent (hence, in grey in (2)–(3)). For example, while the sort *word* is specified for attributes LEFT and RIGHT, the first word in the utterance has no preceding word, so LEFT is not needed there, and similarly for the last word in the utterance and RIGHT. Also, while almost all words have a governor and an appropriate grammatical function, the root word does not, so for this – and only this – word the attributes GOV and FUN appropriate for *word* make no sense. In such cases, the relevant attributes may be specified as having a value of sort *nil*, thus obviating the need for awkward sorts such as *mod-synsem* with subsorts *synsem* and *none* (for values of MOD in Pollard and Sag 1994).

Formulae Also the MTDG language of formulae ("constraints", "principles") is based on that of RSRL. However, as suggested in Przepiórkowski 2021, adopting lean models corresponding to utterances makes it possible to understand quantification in a standard way, as quantification over all objects in the model, rather than as constrained to "components" of the "current object", as in RSRL. This opens way to formulae more closely resembling those of the predicate logic, without the special symbol : for the "currently evaluated object". For example, the Head Feature Principle of HPSG, whose one possible RSRL encoding is given in (4) (Richter 2024: 109), could be formalised with standard quantification and without the special symbol : as in (5).

(4) $(: \sim \textit{headed-phrase}) \rightarrow \exists y\, (: \text{SYNSEM LOCAL CATEGORY HEAD} \approx y \wedge$
$: \text{HEAD-DTR SYNSEM LOCAL CATEGORY HEAD} \approx y)$

(5) $\forall x\ (x \sim \textit{headed-phrase}) \rightarrow \exists y\, (x \text{ SYNSEM LOCAL CATEGORY HEAD} \approx y \wedge$
$x \text{ HEAD-DTR SYNSEM LOCAL CATEGORY HEAD} \approx y)$

This looks like a minor syntactic change, but it has important consequences for how such formulae are understood: they are not anymore true or false *of an object* in a model, as in RSRL, but simply true or false in a given model, as in standard logic.

Returning to MTDG, formulae can be roughly split into those constraining possible models, those defining the linguistic framework, and those representing specific linguistic analyses.

Model Axioms An example of model axioms is the Component Condition (cf. Richter 2007: 102), which makes sure that all objects in the model are connected via attributes (cf. Przepiórkowski 2021: 169):

(6) Component Condition: there is an object from which all model objects can be reached
$\exists x. \forall y.\, \texttt{component}(y, x)$

The relation `component` is defined in Sailer 2003: 115–116 (and Richter 2007: 102) in such a way that $\texttt{component}(y, x)$ holds iff y can be reached from x by following attributes, i.e., if there exists a sequence of attributes $\text{A}_1, \ldots, \text{A}_n$ ($n \geqslant 0$) such that $x\ \text{A}_1 \ldots \text{A}_n \approx y$. Hence, (6) says that there must be an object that has all model objects as its components. (For example, in (3), each *word* object is such an object and there are no other such objects.) This condition ensures the minimality of models, i.e., the lack of spurious "accidental" objects not integrated in the actual analysis.

Framework Axioms Framework axioms make sure that words are totally ordered and that dependency graphs are of a certain kind (e.g., trees).

Total Word Order Axiom The word order encoded via the attribute RIGHT must be total (i.e., linear):

(7) Total Word Order Axiom
$\forall x. \forall y.\, (x \sim \textit{word} \wedge y \sim \textit{word} \wedge \neg(x \approx y)) \rightarrow (\texttt{precede}(x, y) \veebar \texttt{precede}(y, x))$, where:
$\forall x. \forall y.\, \texttt{precede}(x, y) \leftrightarrow (x \sim \textit{word} \wedge y \sim \textit{word} \wedge$
$(x \text{ RIGHT} \approx y \vee \exists z.\, x \text{ RIGHT} \approx z \wedge \texttt{precede}(z, y)))$

That is, for any two distinct words x and y, either x precedes y or y precedes x (but not both).[1]

[1] $\veebar$ in (7) stands for exclusive *or* (XOR) and may be defined as: $p \veebar q \overset{df}{\equiv} p \leftrightarrow \neg q$.

Additionally, we need to make sure that LEFT defines the converse of RIGHT (see (8)) and that the value of LEFT of the first *word* is *nil* rather than reflexively this *word* itself (no other options are possible, given the signature and axioms (7)–(8)), and similarly for the value of RIGHT of the last *word* (see (9)).

(8) $\forall x.\forall y.\,(x \sim \textit{word} \wedge y \sim \textit{word}) \rightarrow (x\ \textsc{right} \approx y \leftrightarrow y\ \textsc{left} \approx x)$

(9) $\neg\exists x.\, x\ \textsc{right} \approx x \vee x\ \textsc{left} \approx x$

Tree Axioms Assume a dependency approach that only allows trees (so, e.g., no double incoming edges to handle raising, as assumed in Word Grammar; but see the discussion around (20) below), but does not forbid non-projective ("crossing") edges. Three axioms constraining government are needed to formalise such trees.

(10) Root Axiom: exactly one word has no governor
$\exists x.\, \texttt{root}(x) \wedge \forall y.\, \texttt{root}(y) \rightarrow y \approx x,$ where the property `root` is defined as:
$\forall x.\, \texttt{root}(x) \leftrightarrow x\ \textsc{gov} \sim \textit{nil}$; additionally, a *word* has no governor iff it has no grammatical function:
$\forall x.\, x\ \textsc{gov} \sim \textit{nil} \leftrightarrow x\ \textsc{fun} \sim \textit{nil}$

(11) Single Governor Axiom: every word has at most one governor
(built in into the assumed signature: each *word* has GOV whose value is a single *word* or *nil*)

(12) Connectivity Axiom: every (non-root) word must be a descendant of the root
$\forall x.\, x \sim \textit{word} \rightarrow (\texttt{root}(x) \vee \exists y.\, \texttt{root}(y) \wedge \texttt{descendant}(x,y)),$ where the `descendant` is defined as:
$\forall x.\forall y.\, \texttt{descendant}(x,y) \leftrightarrow (x\ \textsc{gov} \approx y \vee \exists z.\, x\ \textsc{gov} \approx z \wedge \texttt{descendant}(z,y))$

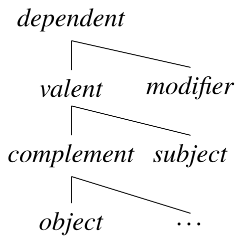
Thus, the Root Axiom in (10) says that there is a single root of the dependency tree, i.e., a *word* without a governor (or grammatical function; see the third formula in (10)). The Single Governor Axiom in (11) does not need to be separately stated, as it follows from the signature. Finally, the Connectivity Axiom in (12) says that every non-root word must be a descendant of (i.e., subordinate to) a root, that is – given (10) – of the unique root.

Depending on a given dependency approach, projectivity axioms may be added or the single governor requirement may be dropped. This latter move would, however, require a change in the signature implicitly assumed in (2)–(3) so that the value of GOV is a list or, more radically, so that GOV is dropped altogether and all dependents are represented via a DEPS list on the governor (more on this below).

Linguistic Principles One advantage of providing an HPSG-based model theory for dependency grammars is that it naturally encodes the taxonomy of dependency relations postulated in Word Grammar (e.g., Hudson 2024: 1546, 1565), namely, as a fragment of the sort hierarchy (see (13) below). This, in turn, makes it possible to concisely state certain generalisations concerning word order and valency.

Word Order The basic generalisation about the English word order is that, to the first approximation, subjects precede their governorns while complements follow them. Given the sort subhierarchy in (13), this can be expressed succinctly by the formulae in (14)–(15), where – as above – *subject* is abbreviated to *subj*, etc.

(13)



(14) $\forall x.\forall y.\,(x\ \textsc{gov} \approx y \wedge x\ \textsc{fun} \sim \textit{subj}) \rightarrow \texttt{precede}(x,y)$

(15) $\forall x.\forall y.\,(x\ \textsc{gov} \approx y \wedge x\ \textsc{fun} \sim \textit{comp}) \rightarrow \texttt{precede}(y,x)$

These principles are meant to provide only an initial illustration of how the proposed formalisation makes it possible to talk about word order. An actual theory of word order would have to make recourse to semantics and information structure (for question formation, topicalisation, etc.), and it would need to leave space for certain sentence processing mechanisms, especially, the well-known principle of dependency length minimisation (see, e.g., Hawkins 1994, Gildea and Temperley 2007, Futrell *et al.* 2020).

Lexicon The simplest HPSG approach to the lexicon, namely the Word Principle (see, e.g., Richter 2024: 101), carries over to the proposed formalisation of DGs:

(16) $\forall x.\, x \sim \textit{word} \rightarrow (\mathrm{LE}_1(x) \vee \mathrm{LE}_2(x) \vee \ldots \vee \mathrm{LE}_n(x)),$ where LE_k is the lexical entry of kth word, e.g.:
$\mathrm{LE}_{666}(x) \overset{df}{\equiv} (x\ \textsc{phon} \sim \textit{loves} \wedge x\ \textsc{cat} \sim \textit{verb}),$ in brief: $\mathrm{LE}_{666} \overset{df}{\equiv} \begin{bmatrix} \textsc{phon} & \textit{loves} \\ \textsc{cat} & \textit{verb} \end{bmatrix}$

Given that MTDG inherits from HPSG the notion of sort hierarchy, also more structured approaches to the lexicon may be formalised (cf. Flickinger 1987).

Valency It is clear that lexical entries such as LE_{666} in (16) are not sufficient as they stand – not only because the current paper is limited to syntax, so they do not say anything about other linguistic levels (especially, semantics), but also because they contain very rudimentary morphosyntax (with no information about case, number, person, etc.) and because they do not encode selectional restrictions. Richer morphosyntax and appropriate agreement principles may be ported from HPSG to MTDG relatively straightforwardly, so we ignore such issues here; valency, on the other hand, requires a more explicit treatment.

One possibility, close to traditional HPSG, is to add a single VALENCE list encoding the head's selectional restrictions, so that LE_{666} may be represented as in (17). (Note that, as grammatical functions are represented on VAL members, the usual division of valency into SUBJ and COMPS is not needed here.) Given that VAL elements are of sort *word* (since they have attributes appropriate for *word*), they are subject to the framework axioms stated above, including the Total Word Order Axiom in (7). This automatically ensures that VAL elements belong to the string of words constituting the modelled utterance. Also, since VAL elements have different grammatical functions (FUN values), they necessarily correspond to different words. However, an additional principle is needed to ensure that VAL information is compatible with GOV information – valence elements specified by a given head must have this head as their governor:

(17) $$\begin{bmatrix} \textsc{phon} & \textit{loves} \\ \textsc{cat} & \textit{verb} \\ \textsc{val} & \left\langle \begin{bmatrix} \textsc{cat} & \textit{noun} \\ \textsc{fun} & \textit{subj} \end{bmatrix}, \begin{bmatrix} \textsc{cat} & \textit{noun} \\ \textsc{fun} & \textit{obj} \end{bmatrix} \right\rangle \end{bmatrix}$$

(18) $\forall x.\forall y.\,\mathtt{member}(x, y\ \textsc{val}) \rightarrow x\ \textsc{gov} \approx y,$ where member is defined as usual:
$\forall x.\forall y.\,\mathtt{member}(x, y) \leftrightarrow (x \approx y\ \textsc{first} \vee \mathtt{member}(x, y\ \textsc{rest}))$

Note that (18) is a one-way implication, as adjuncts (modifiers) satisfy its consequent (they have a governor), but not its antecedent (they are not on this governor's VAL list). In MTDG, just as in HPSG, adjuncts specify their governors directly in their lexical entries, here via the GOV attribute (so no separate MOD attribute is needed), as shown in the lexical entry in (19). Another principle could make sure that all – and only – elements of VAL (i.e., arguments) have FUN of type *valent* (see (13)), while other words with non-*nil* GOV value (i.e., adjuncts) have FUN of type *mod*.

(19) $$\begin{bmatrix} \textsc{phon} & \textit{still} \\ \textsc{cat} & \textit{adverb} \\ \textsc{gov} & \textit{verb} \\ \textsc{val} & \textit{nil} \end{bmatrix}$$

Another possibility is to employ the "adjuncts-as-complements" approach, where some (Bouma *et al.* 2001) or all (Przepiórkowski 1999, 2016) traditional adjuncts appear on a VAL-like list, sometimes called DEPS (dependents). On that latter approach, GOV would not be needed, as DEPS would be sufficient to encode the dependency tree. For example, the AVM representation of the running example (1) would now be as in (20).

(20) $$\boxed{1}\begin{bmatrix} \textit{word} & \\ \textsc{phon} & \textit{Marge} \\ \textsc{cat} & \textit{noun} \\ \textsc{deps} & \textit{nil} \\ \textsc{left} & \textit{nil} \\ \textsc{right} & \boxed{2} \end{bmatrix} \boxed{2}\begin{bmatrix} \textit{word} & \\ \textsc{phon} & \textit{still} \\ \textsc{cat} & \textit{adverb} \\ \textsc{deps} & \textit{nil} \\ \textsc{left} & \boxed{1} \\ \textsc{right} & \boxed{3} \end{bmatrix} \boxed{3}\begin{bmatrix} \textit{word} & \\ \textsc{phon} & \textit{loves} \\ \textsc{cat} & \textit{verb} \\ \textsc{deps} & \left\langle \begin{bmatrix} \textit{subj} & \\ \textsc{dep} & \boxed{1} \end{bmatrix}, \begin{bmatrix} \textit{obj} & \\ \textsc{dep} & \boxed{4} \end{bmatrix}, \begin{bmatrix} \textit{mod} & \\ \textsc{dep} & \boxed{2} \end{bmatrix} \right\rangle \\ \textsc{left} & \boxed{2} \\ \textsc{right} & \boxed{4} \end{bmatrix} \boxed{4}\begin{bmatrix} \textit{word} & \\ \textsc{phon} & \textit{him} \\ \textsc{cat} & \textit{pronoun} \\ \textsc{deps} & \textit{nil} \\ \textsc{left} & \boxed{3} \\ \textsc{right} & \textit{nil} \end{bmatrix}$$

This move would require some changes in the principles and signature assumed above – dropping attributes GOV and FUN, adding *list*-valued DEPS appropriate for *word*, adding *word*-valued DEP appropriate for *dependent* (cf. (13)), redefinition of `root` and `descendant` relations in (10) and (12), explicit encoding of the Single Governor Axiom in (11) if still desired (see immediately below), and reformulation of the word order principles (14)–(15) – as well as adding explicit principles to the effect that distinct DEPS members correspond to different *word* objects. However, this move would also make it possible to adopt dependency graphs more complex than trees, as assumed (for raising, relative clauses, etc.) for example in Word Grammar (see, e.g., Hudson 1984: 83–84) or – within NLP – in Enhanced Universal Dependencies (Schuster and Manning 2016).

Note that, in either variant, no attempt is made to ensure the locality of selection; unlike in HPSG, where only *synsem* objects are selected for, here valency lists are assumed to be lists of *word* objects (VAL) or *dependent*s containing such *word* objects (DEPS). That is, the governor has access to the dependent's PHON, as well as all structural and linear information about this dependent. This is a conscious design decision. First, selectional locality constraints do not seem to be a concern in dependency approaches, at least not to the extent that influenced the design of broadly understood "generative" formalisms. Second, within RSRL-based HPSG, it is possible to express constraints violating such selectional locality anyway, so this difference between HPSG and MTDG is aesthetic rather than substantial.[2] Third, this "non-local" phonetic and configurational information may turn out useful in analyses of idiomatic expressions. Finally, last but not least, an attempt to constrain selectional information would – just as in HPSG – lead to more complex, more deeply embedded structures, which would be against the spirit of dependency approaches, where the simplicity of representations is important.

Conclusion As mentioned in the Introduction, the recent syntax textbook Gibson 2025 aimed at cognitive scientists argues for dependency syntax. Importantly, it attempts to provide constraints on dependency structures, i.e., it seems to allude to a model-theoretic DG. Unfortunately, this attempt is not really successful, to a large extent because of the lack of any explicit model theory. The thesis of the current paper is that such a model theory for DGs may be fruitfully and relatively straighforwardly based on that developed for HPSG.

[2] However, Sign-Based Construction Grammar (Boas and Sag 2012) attempts to "solve" this "problem".

References

Boas, H. and Sag, I. A., eds. (2012). *Sign-Based Construction Grammar*. CSLI Publications. **Bouma, G., Malouf, R., and Sag, I. A.** (2001). Satisfying constraints on extraction and adjunction. *Natural Language and Linguistic Theory*, **19**(1), 1–65. **By, T.** (2004). English dependency grammar. In *Proceedings of the Workshop on Recent Advances in Dependency Grammar*, pp. 64–69. COLING. **Debusmann, R., Duchier, D., and Kruijff, G.-J. M.** (2004). Extensible dependency grammar: A new methodology. In *Proceedings of the Workshop on Recent Advances in Dependency Grammar*, pp. 70–76. COLING. **Flickinger, D.** (1987). *Lexical Rules in the Hierarchical Lexicon*. Ph.D. dissertation, Stanford University. **Futrell, R. and Levy, R.** (2017). Noisy-context surprisal as a human sentence processing cost model. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2009): Volume 1, Long Papers*, pp. 688–698. **Futrell, R., Levy, R. P., and Gibson, E.** (2020). Dependency locality as an explanatory principle for word order. *Language*, **96**(2), 371–412. **Gibson, E.** (2025). *Syntax: A Cognitive Approach*. MIT Press. **Gildea, D. and Temperley, D.** (2007). Optimizing grammars for minimum dependency length. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pp. 184–191. **Hawkins, J. A.** (1994). *A Performance Theory of Order and Constituency*. Cambridge University Press. **Hewitt, J. and Manning, C. D.** (2019). A structural probe for finding syntax in word representations. In J. Burstein, C. Doran, and T. Solorio, eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4129–4138. Association for Computational Linguistics. **Hudson, R.** (1984). *Word Grammar*. Blackwell. **Hudson, R.** (2010). *An Introduction to Word Grammar*. Cambridge University Press. **Hudson, R.** (2024). HPSG and Dependency Grammar. In Müller *et al.* (2024), pp. 1531–1580. **Johnson, M.** (1988). *Attribute-Value Logic and the Theory of Grammar*. CSLI Publications. **Jumelet, J., Weissweiler, L., Nivre, J., and Bisazza, A.** (2026). MultiBLiMP 1.0: A massively multilingual benchmark of linguistic minimal pairs. *Transactions of the Association for Computational Linguistics*, **14**, 193–216. **Kahane, S.** (2009). On the status of phrases in Head-driven Phrase Structure Grammar: Illustration by a fully lexical treatment of extraction. In Polguère and Mel’čuk (2009), pp. 111–150. **Kaplan, R. M.** (1995). The formal architecture of Lexical-Functional Grammar. In M. Dalrymple, R. M. Kaplan, J. T. Maxwell, III, and A. Zaenen, eds., *Formal Issues in Lexical-Functional Grammar*, pp. 7–27. CSLI Publications. **Kübler, S., McDonald, R., and Nivre, J.** (2009). *Dependency Parsing*. Morgan & Claypool. **Maruyama, H.** (1990). Structural disambiguation with constraint propagation. In *28th Annual Meeting of the Association for Computational Linguistics*, pp. 31–38. Association for Computational Linguistics. **Mel’čuk, I.** (1988). *Dependency Syntax: Theory and Practice*. The SUNY Press. **Mel’čuk, I.** (2009). Dependency in natural language. In Polguère and Mel’čuk (2009), pp. 1–110. **Müller, S., Abeillé, A., Borsley, R. D., and Koenig, J.-P.**, eds. (2024). *Head-Driven Phrase Structure Grammar: The Handbook*. Language Science Press, 2nd edition. **Polguère, A. and Mel’čuk, I.**, eds. (2009). *Dependency in Linguistic Description*. John Benjamins. **Pollard, C. and Sag, I. A.** (1994). *Head-driven Phrase Structure Grammar*. Chicago University Press / CSLI Publications. **Przepiórkowski, A.** (1999). *Case Assignment and the Complement-Adjunct Dichotomy: A Non-Configurational Constraint-Based Approach*. Ph.D. dissertation, Universität Tübingen. **Przepiórkowski, A.** (2016). How *not* to distinguish arguments from adjuncts in LFG. In D. Arnold, M. Butt, B. Crysmann, T. H. King, and S. Müller, eds., *Proceedings of the Joint 2016 Conference on Head-driven Phrase Structure Grammar and Lexical Functional Grammar*, pp. 560–580. CSLI Publications. **Przepiórkowski, A.** (2021). Three improvements to the HPSG model theory. In S. Müller and N. Melnik, eds., *Proceedings of the HPSG 2021 Conference*, pp. 165–185. Frankfurt/Main University Library. **Przepiórkowski, A.** (2023). LFG and HPSG. In M. Dalrymple, ed., *Handbook of Lexical Functional Grammar*, pp. 1861–1918. Language Science Press. **Pullum, G. K. and Scholz, B. C.** (2001). On the distinction between model-theoretic and generative-enumerative syntactic frameworks. In P. de Groote, G. Morrill, and C. Retoré, eds., *Logical Aspects of Computational Linguistics: 4th International Conference*, pp. 17–43. Springer-Verlag. **Richter, F.** (2004). *A Mathematical Formalism for Linguistic Theories with an Application in Head-Driven Phrase Structure Grammar*. Ph.D. dissertation (defended in 2000), Universität Tübingen. **Richter, F.** (2007). Closer to the truth: A new model theory for HPSG. In J. Rogers and S. Kepser, eds., *Model-Theoretic Syntax at 10*, pp. 99–108. European Summer School on Logic, Language and Information (ESSLLI 2007). **Richter, F.** (2024). Formal background. In Müller *et al.* (2024), pp. 93–131. **Sailer, M.** (2003). *Combinatorial Semantics and Idiomatic Expressions in Head-driven Phrase Structure Grammar*. Ph.D. dissertation (defended in 2000), Universität Tübingen. **Schröder, I., Menzel, W., Foth, K., and Schulz, M.** (2000). Modeling dependency grammar with restricted constraints. *Traitement Automatique des Langues (T.A.L.)*, **41**(1), 97–126. **Schuster, S. and Manning, C. D.** (2016). Enhanced English Universal Dependencies: An improved representation for natural language understanding tasks. In N. Calzolari, K. Choukri, T. Declerck, M. Grobelnik, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, and S. Piperidis, eds., *Proceedings of the Tenth International Conference on Language Resources and Evaluation, LREC 2016*, pp. 2371–2378. European Language Resources Association.