To appear in Keith Brown (Ed.): *Encyclopedia of Language and Linguistics, Second Edition.* Oxford: Elsevier.

Head-Driven Phrase Structure Grammar
Linguistic Approach, Formal Foundations, and Computational Realization

Robert D. Levine and W. Detmar Meurers
The Ohio State University

1 Introduction

The theoretical richness, formal rigor and computational versatility of Head-driven Phrase Structure Grammar (HPSG) preclude any kind of in-depth coverage of its content within the confines of an encyclopedia article such as this. Our objectives are accordingly far more modest: we seek to provide a kind of aerial view of the linguistic approach (§2), summarize the formal foundations (§3), and characterize computational work developed based on this paradigm (§4).

2 The linguistic approach

Our discussion is primarily rooted in Pollard and Sag (1994) as the touchstone of the HPSG paradigm, based on earlier exploratory work (Pollard and Sag, 1987). This choice reflects the high degree of formal explicitness with deep and comprehensive coverage achieved in that work, which brought to fruition a line of research originating in the early 1980s. HPSG was heavily influenced by Generalized Phrase Structure Grammar (GPSG, Gazdar et al., 1985), itself the product of a fertile period at the end of the 1970s and into the early 1980s when a number of explicit and comprehensive alternatives to transformational grammar began to emerge. GPSG, Lexical-Functional Grammar (LFG, Bresnan, 1982), a revived, empirically-based version of Categorial Grammar (CG, Ades and Steedman, 1982), and several other approaches all made their appearance, and HPSG has integrated ideas from each of these paradigms, in combination with insights developed in the Government and Binding (GB, Chomsky, 1981) paradigm.

2.1 Fundamentals of the linguistic framework

HPSG rests on two essential components: (i) an explicit, highly structured representation of grammatical categories, encoded as typed feature structures, whose complex geometry is motivated by empirical considerations against the background of theoretical desiderata such as locality; (ii) a set of descriptive constraints on the modeled categories expressing linguistic generalizations and declaratively characterizing the expressions admitted as part of the natural language. The formal foundation of this architecture is discussed in §3.

From a linguistic perspective, the set of descriptive constraints expressing the theory of an HPSG grammar consist of a) a lexicon licensing basic words, b) lexical rules licensing derived words, c) immediate dominance schemata licensing constituent structure, d) linear precedence statements constraining constituent order, and e) a set of grammatical principles expressing generalizations about linguistic objects. The clearest way to illuminate the interaction of these components is to provide a concrete example.

2.1.1 Lexical organization

Consider the verb *put*, for which a partial lexical entry is shown in Figure 1.

$$\begin{bmatrix} \textit{word} & \\ \textsc{phon} & <\textit{put}> \\ \textsc{synsem}|\textsc{local} & \begin{bmatrix} \textsc{category} & \begin{bmatrix} \textsc{head} & \begin{bmatrix} \textit{verb} & \\ \textsc{aux} & - \end{bmatrix} \\ \textsc{valence} & \begin{bmatrix} \textsc{subj} & \left\langle NP_{\boxed{1}} \right\rangle \\ \textsc{comps} & \left\langle NP_{\boxed{2}},\ PP_{\boxed{3}} \right\rangle \end{bmatrix} \end{bmatrix} \\ \textsc{content} & \begin{bmatrix} \textit{put-relation} & \\ \textsc{putter} & \boxed{1} \\ \textsc{thing-put} & \boxed{2} \\ \textsc{destination} & \boxed{3} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Figure 1: A basic lexical entry

As specified by the type in the top-left corner, this Attribute-Value Matrix (AVM) describes objects of type *word*. The PHON attribute here is simply taken to be a list of strings serving as a placeholder for an actual phonological representation for HPSG, as developed by Bird and Klein (1994) and Höhle (1999). The morpho-syntactic information which characterizes local properties of linguistic expressions are specified under the CATEGORY feature which, along with the semantic CONTENT, are identified by the LOCAL part of the SYNSEM value. The subset of CATEGORY properties which are necessarily shared between mother and head daughter in a local tree are packaged together under the HEAD feature.

The VALENCE feature specifies the combinatory potential of lexical items as lists of *synsem* objects (as opposed to lists of *signs*). Thus neither phonological information (specified in PHON), nor the DAUGHTERS feature, which we will see as encoding constituent structure in objects of type *phrase*, can be selected for, incorporating the well-supported generalization that syntactic selection is independent of phonological form and is consistently local.

The specification of the valence features (SUBJ and COMPS) and CONT, specifying the semantic roles assigned by the head, make it possible to lexically associate the valents of a head with the semantic contribution of these valents to the relation it denotes. The boxed numbers indicate token-identity of the values specified (cf. §3). Instead of specifying such linking in each lexical entry, it can be derived from general linking principles (cf. Koenig and Davis, 2003, and references therein). The issue of lexical generalizations brings us to the question how the lexicon is defined.

The basic lexicon of an HPSG theory can be defined by the *Word Principle* shown in Figure 2, where each *Lexical-Entry* is a description of the kind we saw in (1).

$$\textit{word} \rightarrow \textit{Lexical-Entry}_1 \vee \textit{Lexical-Entry}_2 \vee \ldots \vee \textit{Lexical-Entry}_n$$

Figure 2: The Word Principle

While this principle illustrates the basic method for expressing a lexicon as a constraint, a wide range of approaches have been developed which do not take lexical entries of type *word* as primitives, such as realizational approaches to morphology (cf., e.g., Erjavec, 1996; Riehemann, 1998) or word syntax approaches (cf., e.g., Krieger and Nerbonne, 1993). The HPSG architecture also readily supports lexical generalizations through principles expressing constraints on sets of lexical elements, so called *vertical generalizations*, as well as through lexical rules expressing relations between sets of lexical elements, referred to as *horizontal generalizations* (cf. Meurers, 2001; Bouma et al., 2000, and references cited therein).

2.1.2 Phrasal structures

Turning to phrasal structures, consider the tree shown in Figure 3 (note that S, NP, VP, etc. are just abbreviations for AVM specifications; for space reasons, some attribute names are also abbreviated).

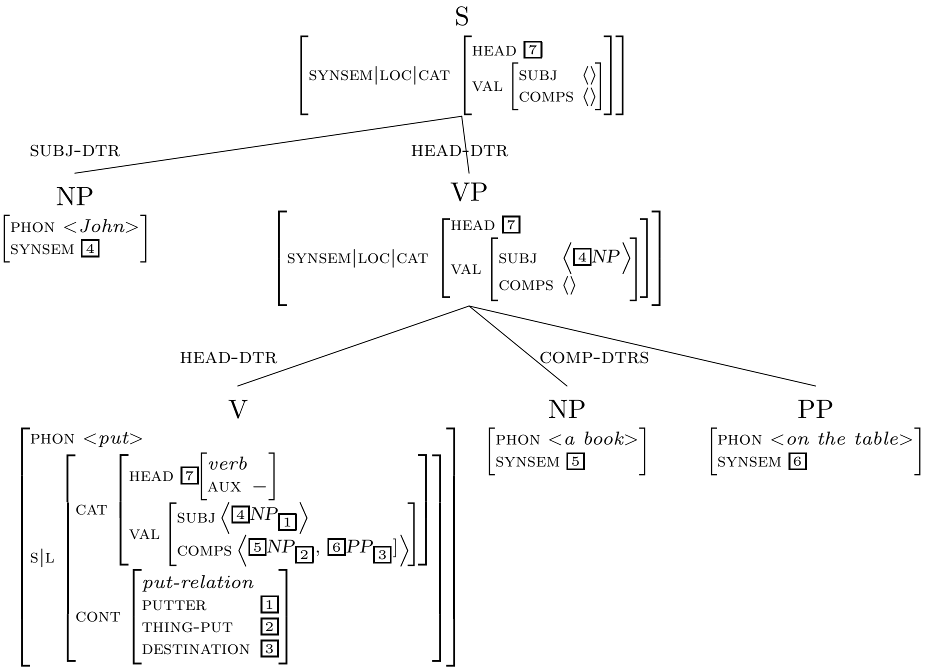


Figure 3: An example for a phrasal construction

As with all grammatical principles in HPSG, generalizations over phrasal structure are expressed as constraints. Figure 4 shows the relevant parts of the Immediate Dominance (ID) Principle, which essentially encodes a version of the X-bar schema (Jackendoff, 1977).

$$\left[\begin{array}{ll} \textit{phrase} & \\ \textsc{dtrs} & \textit{headed-struc} \end{array}\right] \rightarrow$$

$$\left[\begin{array}{ll} \textsc{synsem}|\textsc{loc}|\textsc{cat} & \left[\begin{array}{ll} \textsc{head} & \left(\left[\begin{array}{ll}\textit{verb} & \\ \textsc{inv} & -\end{array}\right] \vee \neg\ \textit{verb}\right) \\ \textsc{val} & \left[\begin{array}{ll}\textsc{subj} & \langle\rangle \\ \textsc{comps} & \langle\rangle\end{array}\right]\end{array}\right] \\ \textsc{dtrs} & \left[\begin{array}{ll}\textit{head-subj-struc} & \\ \textsc{head-dtr} & \textit{phrase}\\ \textsc{subj-dtr} & \textit{sign}\end{array}\right]\end{array}\right] \qquad \text{(Head-Subject)}$$

$$\vee \left[\begin{array}{ll} \textsc{synsem}|\textsc{loc}|\textsc{cat} & \left[\begin{array}{ll} \textsc{head} & \left(\left[\begin{array}{ll}\textit{verb} & \\ \textsc{inv} & -\end{array}\right] \vee \neg\ \textit{verb}\right) \\ \textsc{val} & \left[\begin{array}{ll}\textsc{subj} & \left\langle \textit{synsem} \right\rangle \\ \textsc{comps} & \langle\rangle\end{array}\right]\end{array}\right] \\ \textsc{dtrs} & \left[\begin{array}{ll}\textit{head-comps-struc} & \\ \textsc{head-dtr} & \textit{word}\end{array}\right]\end{array}\right] \qquad \text{(Head-Complement)}$$

$$\vee \ldots$$

Figure 4: Immediate Dominance Principle

The first disjunct of the ID Principle is the Head-Subject schema, which licenses the upper local tree of the example in Figure 3; the lower local tree is licensed by the Head-Complement schema.

In addition to the Head-Subject and the Head-Complement structures licensed by the two disjuncts shown, Pollard and Sag (1994) assume only a small inventory of general schemata: Head-Adjunct, Head-Filler, Head-Marker, and a flat Head-Subject-Complement schema.

The subcategorization requirements specified under the VALENCE attribute in the lexical entry of *put* are realized as the result of the Valence Principle, which specifies that the valence requirements of a lexical head are identified with the SYNSEM value of the realized daughters, with all unrealized requirements being handed on to the mother of the local tree. This principle essentially is the phrase-structural analogue to Categorial Grammar's treatment of valence satisfaction as combinatory cancellation (though, unlike CG, HPSG's principle does not assume that all valence-driven combinations involve a functor combining with a single argument). In the example in Figure 3, we can see that the valence specification in the lexical entry of *put* requires an NP subject and two complements, an NP and a PP. In the lower tree, the two complements are realized as *a book* ($\boxed{5}$) and *on the table* ($\boxed{6}$). As a result of the Valence Principle, the VP mother of that local tree has an empty COMPS requirement. The SUBJ requirement is inherited unchanged from the lexical verb and realized as *John* ($\boxed{4}$) in the higher local tree, dominated by the fully saturated S.

The percolation of head information along the head projection is the result of the Head-Feature Principle (HFP), shown in Figure 5.

$$\begin{bmatrix} \textit{phrase} \\ \text{\textsc{dtrs}}\ \ \textit{headed-structure} \end{bmatrix} \rightarrow \begin{bmatrix} \text{\textsc{synsem|loc|cat|head}} & \boxed{1} \\ \text{\textsc{dtrs|head-dtr|synsem|loc|cat|head}} & \boxed{1} \end{bmatrix}$$

Figure 5: Head Feature Principle

The HFP is a straightforward adaptation of the Head Feature Convention of Gazdar et al. (1985); it ensures that in any headed structure, the HEAD specifications of a mother and its head daughter are token-identical. In our example in Figure 3, the category *verb* and the $[\text{\textsc{aux}}\ -]$ specification is shared as index $\boxed{7}$ between the lexical head *put* and its VP and S projections.

2.2 Capturing dependencies

Arguably the essential test of a grammatical framework is its ability to capture, parsimoniously and inclusively, the ubiquitous grammatical dependencies of natural languages. Such dependencies are often grouped into two major classes: *local dependencies*, which hold over a limited syntactic domain and frequently depend on properties of a specific class of lexical items, and *non-local dependencies*, which appear to hold over arbitrarily large syntactic distances and are largely independent of the lexicon. In this section, we briefly review, mostly based on Pollard and Sag (1994), how the machinery available in HPSG leads to compact solutions to the problems posed by both dependency types in English.

2.2.1 Local dependencies

Phenomena such as agreement, auxiliary-choice morpho-syntactic dependencies, subject selection, the active/passive relationship, and the so-called Raising/Equi constructions all exemplify what we refer to as local dependencies. In HPSG, these phenomena typically are all handled in terms of the selectional properties of lexical heads. We here sketch the analysis of control constructions as an illustration of this general approach.

Control construction In line with the general HPSG strategy to reject those invisible elements for which there is no empirical evidence, in non-finite constructions, the subject of the embedded verb is analyzed as not locally expressed. There is no need to assume an invisible realization of such subjects, as done in transformational grammar, since HPSG is based on richer linguistic data structures in which lexical specifications, such as subcategorization requirements, are explicitly represented. The questions what is interpreted to be the subject of the non-finite verb, and why verbs selecting non-finite complements differ with respect to what kind of controllers can occur are thus answered through the lexical specification of the control verbs.

A subject-to-subject raising verb like *seem*, for example, will have the partial description in the lexicon given in Figure 6. We see that the subject valence requirement of the subject raising verb is identified with the subject of the verbal complement. Note that the subject is not assigned a semantic role by the raising

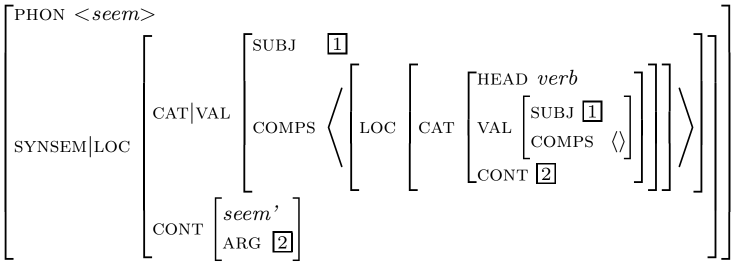


Figure 6: Partial lexical entry for a raising verb

verb. In order to derive the fact that such raising is only permissible if the raised argument is assigned a semantic role elsewhere in the structure, Pollard and Sag (1994, p. 140) specify a general meta principle, the Raising Principle.

Based on the lexical specification in Figure 6, the properties of a raising verb fall out of the interaction with the rest of the architecture; in particular the identification of the subject requirement of the two verbs implies that: If the embedded verb requires a non-referential subject or permits a clausal subject, this is enforced by the raising verb realizing the subject. If the embedded verb has a subject with an idiomatic interpretation, the subject also has that interpretation when realized as argument of a raising verb. In languages where subjectless constructions exist, raising verbs can embed such subjectless complements. And passivization of the non-finite complement results in a paraphrase.

In contrast, the partial lexical specification of a subject-to-subject equi verb like *try* in Figure 7 specifies semantic co-indexing of the subject valence requirement of the subject control equi verb with the subject of the verbal complement. And the subject is assigned a semantic role by the subject control equi verb.

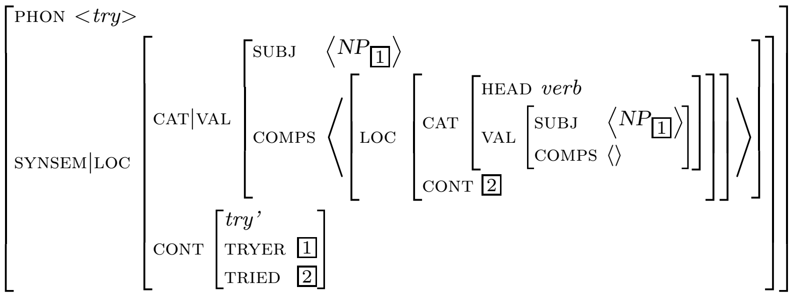


Figure 7: Partial lexical entry for an equi verb

2.2.2 Non-local dependencies

Probably the strongest-seeming case for syntactic transformations was the existence of unbounded dependencies, particularly extraction phenomena. Given that certain valence requirements appear to be satisfied by elements arbitrarily distant from the selecting head, it is intuitively natural to suppose that they were moved there after selection. A derivational account thus seems attractive, even under the anti-transformational stances evident in, e.g., Brame (1976).

The flaws in this line of thinking were made obvious in Gazdar's (1981) pioneering work on an explicit formal characterization of filler/gap linkages in a monostratal framework, which was assimilated into a feature-based account by Bear (1981) and Gazdar et al. (1985). In HPSG, the relevant feature is the set-valued SLASH, which takes as its value the LOCAL specifications of extracted elements.

The Filler-Head schema licenses the local tree realizing the filler as illustrated in Figure 8.

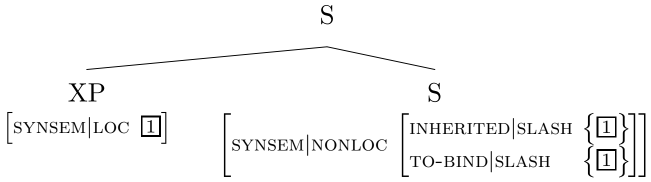


Figure 8: The local tree realizing the filler of a non-local dependency

The Nonlocal Feature Principle has the effect of sharing the SLASH value of the mother with that of some daughter until it reaches the trace, which has the lexical entry shown in Figure 9.

$$\begin{bmatrix} \textsc{phon} & \langle\rangle \\ \textsc{synsem} & \begin{bmatrix} \textsc{loc}\ \boxed{1} \\ \textsc{nonloc}|\textsc{inher}|\textsc{slash}\ \left\{\boxed{1}\right\} \end{bmatrix} \end{bmatrix}$$

Figure 9: Lexical entry for elements not locally realized

This entry identifies the properties of the non-locally realized element specified under SLASH with its own LOCAL specification.

Taken together, the lexical entry and the general principles yield a structure for a sentence such as *What do you think Robin said?* as in Figure 10. Note that *pace* Baker (1978), apart from the schema realizing the filler, the sentence is licensed by the same general ID schemata also used in non-extraction examples.

Independent support for the inherently local propagation of SLASH feature specifications as the vehicle for UDC dependencies comes from work on languages which visibly mark filler/gap pathways, as discussed at length in Zaenen (1983) and Hukari and Levine (1995). Such languages record by morpho- or phono-syntactic means the presence of each intermediate clausal step along the filler/gap pathway.

2.2.3 Other dependencies

In addition to the local and non-local dependencies, so-called middle distance dependencies arising in the context of coherence/verb-raising in the Germanic languages and of restructuring phenomena in Romance—an issue we return to briefly in the next section when discussing argument attraction as the lexical specification underlying an analyses of these phenomena.

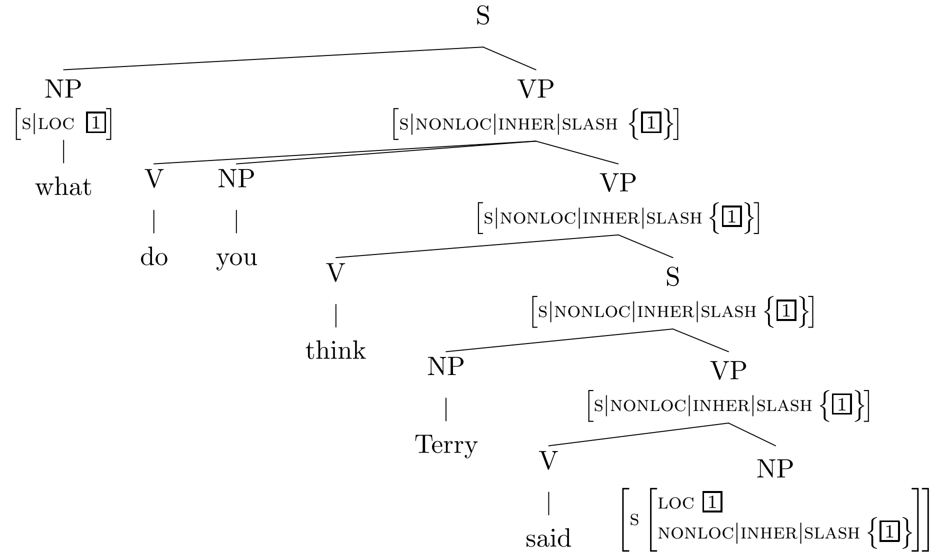


Figure 10: An example for a non-local dependency

Finally, another kind of dependency relation, treated in considerable detail in Pollard and Sag (1992, 1994), involves the syntactic conditions which require, allow or forbid coindexation of referring constituents. These conditions, collectively referred to as the HPSG Binding Theory, are formulated in HPSG not in terms of syntactic configuration but rather in terms of an obliqueness ordering that is lexically specified as a part of the selection properties.

2.3 Strands of research

During the past decade, a vast literature in the HPSG framework has addressed empirical and theoretical issues arising in a wide range of languages. The scale of the enterprise allows only the most cursory overview; interested readers will find many of these issues taken up in breadth and detail in Kathol and Przepiórkowski (To appear) and extensive on-line bibliographies of HPSG literature can, e.g., be found at http://www.cl.uni-bremen.de/HPSG-Bib/ as well as at http://www.sfs.uni-tuebingen.de/hpsg/library.html.

Beginning with the cross-linguistic breadth of coverage, a large collection of volumes has been devoted to the analysis of Romance languages (cf., e.g. Balari and Dini, 1997; Monachesi, 1999), Slavic languages (cf., e.g. Borsley and Przepiórkowski, 1999), German (cf., e.g. Nerbonne et al., 1994; Kiss, 1995; Kathol, 2000; Müller, 1999, 2002; Meurers, 2000; Meurers and Kiss, 2001; Holler, 2001; De Kuthy, 2002), Japanese (cf., e.g. Iida, 1995), Welsh (cf., e.g. Borsley and Jones, 2005), and dozens of other languages. In addition, innumerable article in journals, conference proceeding and collections address the relevance of HPSG analyses to an understanding of complex phenomena in a variety of languages.

A good example of the latter is the analysis of clause union and restructuring phenomena in terms of argument attraction (Hinrichs and Nakazawa, 1994), which has had significant impact on analyses of languages as diverse as French and Korean. Similarly, the linearization-based approach to discontinuous constituency first proposed as a key to reconciling German constituent structure with the facts of word order in that language (Reape, 1996; Kathol, 2000; Müller, 1999) has been widely applied to problems of discontinuity in a variety of construction types in English, French, Portuguese Japanese, and Warlpiri. Another example is the introduction of a limited degree of non-locality of selection into HPSG as a result of the work on case assignment in German reported in Meurers (2000, ch. 10), which has proven useful in discussions of quite different phenomena, such as *tough* constructions in various languages (cf., e.g., Levine, 2001; Amaral, 2005).

Other work addresses questions of the grammatical architecture, such as the interaction of syntax, information structure, and intonation (Engdahl and Vallduví, 1994; De Kuthy, 2002; De Kuthy and Meurers, 2003) and the integration of a range of semantic architectures into HPSG, such as Underspecified DRT (Frank and Reyle, 1995), Minimal Recursion Semantics (Copestake et al., To appear), Constraint Language for Lambda Structures (Egg et al., 2001), and Lexical Resource Semantics (Richter, 2004a), which support innovative analyses of longstanding natural language puzzles, such as those posed by idioms (Sailer, 2003).

As should be expected of an active research community, the HPSG paradigm also includes significant diversity and disagreement about research strategies and analyses. One such issue concerns the choice between lexical and constructional approaches. Pollard and Sag (1994) account for the complex properties of relative clauses using null functional heads; Sag (1997) instead essentially revives the treatment of these adjuncts in Gazdar et al. (1985) by proposing to account for their behavior in terms of constructional characteristics inherited in a complex type hierarchy from super-types. This approach has been further explored in Ginzburg and Sag (2001) and other work which increasingly links certain quarters within HPSG to the program of Construction Grammar (Fillmore and Kay, 1999). There is no consensus within HPSG that this is the optimal move, however, and research which minimizes reliance on structural types continues to emphasize the head-driven, lexical aspect of HPSG (cf., e.g., Müller, 2004). In a directly related issue, two principle strands of HPSG have developed with respect to the use of types. In one strand, new types are introduced for any class of objects to be generalized over and these types are arranged in large, cross-classifying taxonomies. The other strand emphasizes minimal linguistic ontologies in that new types are only introduced if they are necessary to refer to language properties that cannot otherwise be distinguished in the model (cf. Meurers, 2001; Koenig and Davis, 2003). Another source of diversity lies in the analysis of unbounded dependencies. Traceless versions of HPSG have been pursued in the last part of Pollard and Sag (1994), Bouma et al. (2001), and Ginzburg and Sag (2001), but this line of analysis is not universally accepted (cf., e.g., Levine, 2003; Müller, 2002, ch. 6.2.5.1).

The foregoing discussion, while inevitably very incomplete, should give the reader an idea of the creative ferment that has characterized HPSG over the past decade. Nor has this innovative momentum been confined to the empirical/analytic side; it has, as we shall discuss below, also made itself evident in the formal and computational aspects of the theory, to which we now turn.

3 HPSG from a formal perspective

3.1 A declarative characterization of natural languages

The first step of a scientific approach to any empirical subject is the modeling of the domain. Models of empirically observable objects are established to capture the relevant properties of those objects. The theories then make reference to these models and express generalizations about which of the potential models actually exist and how their properties relate.

Correspondingly, Head-Driven Phrase Structure Grammar as a scientific approach to language specifies every grammar to have two components: the signature and the theory (in a formal sense). The *signature* of an HPSG grammar defines the ontology ('declaration of what exists'): which kind of objects are distinguished, and which properties of which objects are modeled. It consists of the type hierarchy and the appropriateness conditions, defining which type has which appropriate attributes with which appropriate values. The *theory* of an HPSG grammar is a set of description language statements, often referred to as the constraints. The theory essentially singles out a subset of the objects declared in the signature, namely those which are grammatical. A linguistic object is admissible with respect to a theory iff it satisfies each of the descriptions in the theory and so does each of its substructures. In the following, we take a closer look at these two components of an HPSG grammar.

3.1.1 Modeling the domain

What do the mathematical structures used as models for HPSG theories look like, and how are they related to the linguistic objects? Pollard and Sag (1994, p. 8) require the HPSG architecture to use typed feature structures (TFS) as models of linguistic objects. More precisely, they argue for modeling types (vs. tokens) of total linguistic objects (as opposed to partial information about them) and therefore assume the models to be totally well-typed and sort-resolved feature structures.

Here, totally well-typed means that a) "what attribute labels can appear in a feature structure is determined by its sort; this fact is the reflection within the model of the fact that what attributes ... an empirical object has depends on its ontological category." (Pollard and Sag, 1994, p. 18) and b) "every feature that is appropriate for the sort assigned to that node is actually present." (ibid.). Sort-resolvedness requires that every node is assigned a most specific type as value. Note that *type* and *sort* are often used synonymously; the same is true for *attribute* and *feature*.

It is important to realize that these requirements are not merely technical stipulations; they are a direct consequence of the decision that feature structures should serve as total models of linguistic objects. It is not uncontroversial within the HPSG community whether total models of linguistic objects are what is needed for linguistics. Problems for such total models arise in the analysis of coordination (Pollard and Sag, 1994, p. 203, fn. 39; Sag, 2003) and arguments for and against total models arise from the formalization of lexical rules and regarding the representation of unrealized arguments (Pollard, 2001; Meurers, 2001, p. 180–182). At the same time, we will see in the next section that only total models support the classical negation needed for formulating the implicational constraints standardly used to express grammatical principles in HPSG.

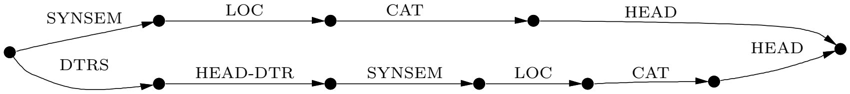
3.1.2 Expressing the theory

Having clarified how linguistic objects are modeled, we can turn to the question of how to express the theory and how the theory characterizes the set of well-formed linguistic objects.

An HPSG theory is specified using a specific description language, sometimes written down in the form of *Attribute-Value Matrices (AVM).* The description language makes it possible to express that a feature structure has a certain type (e.g., *noun*), that the value of an attribute of a feature structure satisfies a given description (e.g., [CASE *nominative*]), or that the values of two attribute paths are *token identical.* Complex descriptions are obtained by combining descriptions with the help of conjunction, disjunction and negation; in the AVM notation, conjunction is implicit.

The notion of token identity means that two paths point to the same node in the feature structure model; this is illustrated in (1) with an AVM specifying token identity of HEAD values (as specified by the HFP discussed in §2.1.2) and a small graph showing the relevant part of a feature structure denoted by this description.

(1) a.
$$\begin{bmatrix} \textsc{synsem}|\textsc{loc}|\textsc{cat}|\textsc{head} & \boxed{1} \\ \textsc{dtrs}|\textsc{head-dtr}|\textsc{synsem}|\textsc{loc}|\textsc{cat}|\textsc{head} & \boxed{1} \end{bmatrix}$$

b.


Another identity notion sometimes referred to is *type identity*: Two feature structures are type identical iff they are of the same type, the same attributes are defined on both feature structures, and the feature structures which are the values of the attributes are type identical.

Turning to the second question, what does it mean for a theory to specify admissible feature structures? An HPSG theory simply is a set of descriptions which are interpreted as being true or false of a feature structure in the domain. A feature structure is admissible with respect to a certain theory iff it satisfies each of the descriptions in the theory and so does each of its substructures. The descriptions which make up the theory are also called *constraints*, since these descriptions constrain the set of feature structures which are admissible with respect to the theory compared to the domain of feature structures specified by the signature. Note that the term constraint has been used for many different purposes—we will only use 'constraint' to mean 'description which is part of the theory'.

3.2 Formalizing the HPSG setup

The setup of HPSG characterized in the previous section is summed up in Figure 11. The usual symbols for description language operators are used: conjunction ($\wedge$), disjunction ($\vee$) and negation ($\neg$). Type assignment and path equality are noted as "$\sim$" and "$=$", respectively.

$$\underset{(\sim, = \,;\ \wedge, \vee, \neg)}{\text{Descriptions}} \quad \xrightarrow[\textit{Interpretation}]{} \quad \text{TFS} \quad \xleftrightarrow[\substack{\textit{One-to-one}\\ \textit{Correspondence}}]{} \quad \text{Linguistic object}$$

Figure 11: The setup desired in Pollard and Sag (1994)

Several logics have been proposed to provide the formal foundations for this setup. There essentially are two families of logics dealing with this task: the *Kasper-Rounds* logics (Rounds and Kasper, 1986; Moshier and Rounds, 1987; Carpenter, 1992; Copestake, 1993) and the *Attribute-Value* logics (Johnson, 1988; Smolka, 1988; King, 1989). In the following we focus on two prominent representatives in the context of HPSG, the Kasper-Rounds logic defined in Carpenter (1992) and the Attribute-Value logic of King (1989). Figure 12 shows the setup proposed in Carpenter (1992), with "$\neq$" as notation for path inequality.

$$\underset{(\sim, =, \neq \,;\ \wedge, \vee)}{\text{Descriptions}} \quad \xrightarrow[\textit{Interpretation}]{} \quad \text{TFS} \quad \xleftrightarrow[\substack{\textit{One-to-one}\\ \textit{Correspondence}}]{} \quad \text{Partial information}$$

Figure 12: The setup of Carpenter (1992)

The descriptions of Carpenter (1992) describe typed feature structures modeling partial information. Presumably there is a further level, in which the partial information is related to the linguistic objects, which is left undefined. This setup can also be seen to underly an early model of HPSG (Pollard and Sag, 1987). Since partial information is modeled, not total linguistic objects, Carpenter (1992) does not require the typed feature structures to be well-typed or sort-resolved.

The difference between the total models of linguistic objects assumed in HPSG since Pollard and Sag (1994) and the models of partial information of Carpenter (1992) has important consequences. Moshier and Rounds (1987) show that in the setup of a Kasper-Rounds logic, full classical negation as part of the description language destroys subsumption monotonicity on typed feature structures—a property which, as Carpenter and others argue, has to be upheld if feature structures are to function as models of partial information. Carpenter (1992, ch. 15, p. 233) goes one step further by stating that even for the formulation of implicational constraints with type antecedents (i.e., type negation), subsumption monotonicity cannot be achieved. The description language of Carpenter (1992) therefore only contains path-inequations, a weaker form of negation.

Turning to Attribute Value Logics, King (1994, 1999) shows that his Speciate Reentrant Logic (SRL) can provide the setup we saw in Figure 11, i.e., the setup envisaged for HPSG in Pollard and Sag (1994). Descriptions are given a set theoretic interpretation: The interpretation of a description is a set of objects, i.e., an object satisfies a description iff it is in the denotation of that description. For example, the description *word* denotes the set of all objects of type *word*; and the description [CASE *nominative*] denotes the set of all objects for which the partial function CASE is defined such that the value of that function is an object of type *nominative*. Conjunction, disjunction, and negation as operations on descriptions are interpreted as set intersection, set union, and set complement, respectively.

While King's SRL captures the basic HPSG setup, one conceptual difference is that Pollard and Sag (1994) envisage modeling *abstract* linguistic objects (i.e., types) whereas SRL models *concrete* linguistic objects (i.e., tokens). But Pollard (1999) distances himself from the earlier emphasis on types and adopts King's denotation function in place of an approach based on feature structure satisfaction and admission.

King's SRL falls short, however, in failing to provide relational expressions. Richter (2000) addresses this issue by defining the Relational Speciate Re-entrant Language (RSRL) which adds a bounded form of quantification and relations to the inventory of expressions. He shows in detail how this formal language can provide a direct and transparent formalization of the English HPSG in Pollard and Sag (1994), which turns out to require more relations than was traditionally assumed. Richter (2000, 2004a) can also be recommended as a comprehensive overview of the complex issues involved in formalizing HPSG.

4 Processing with HPSG-based grammars

4.1 Motivation

There are at least three motivations for implementing and processing with HPSG-based grammars. From a *linguistic perspective*, such work can provide important feedback on the empirical adequacy of the linguistic analyses, on the explicitness, completeness, and compatibility of the linguistic theories integrated in one grammar, and on the rigid and complete formalization of the linguistic architecture. From a *computational perspective*, grammar implementation can stimulate and test the development of systems which support parsing and generation with HPSG-based grammars. Finally, from an *engineering perspective*, deep processing with HPSG-based grammars is potentially useful for applied human language technology as, e.g., illustrated by the speech-to-speech machine translation project VerbMobil (http://verbmobil.dfki.de/).

Depending on the perspective adopted, research on HPSG-based processing has emphasized different issues; it is important to note, however, that even from an engineering perspective the value added by basing grammar writing on HPSG as a linguistic paradigm derives from the insights provided by an active linguistic research community. The development of implementation systems which support a transparent and tractable way of implementing a grammar close to the linguistic theory thus is of particular importance for HPSG-based computational work. In the same vein, a thorough documentation of grammars (cf., e.g., Meurers, 1994; Gregory, 1997; Müller, 1999), including discussions motivating the differences between the linguistic theory and the implementation is an essential component of such work.

4.2 Frameworks for HPSG-based processing

Since the early 90s, a range of systems has been developed that support processing with HPSG-based grammars. Classifying these systems according to the means used for expressing grammars, one can distinguish systems which are a) constraint-based in the sense that grammars for these systems consist of implicational statements constraining the domain directly, b) relation-based in the sense that they make use of a relational level for organizing the grammar, i.e., the relational extension of the constraint language, or c) a combination of the two. We here focus on providing an overview of the key theoretical distinctions relevant for characterizing the different systems; a detailed system comparison can be found in Bolc et al. (1996).

4.2.1 Systems directly constraining the domain

The Typed Feature structure System (TFS, Emele and Zajac, 1990) arguably was the first system that supported a direct implementation of HPSG theories (cf., also, Matiasek, 1994). TFS grammars consist of a set of implicational statements with type antecedents. Any description can be entered as a query, and the system returns a more specific description such that the description and each of its parts satisfies each constraint in the theory.

The advantage of this approach is that the organization of constraints in TFS is the same as that in the HPSG architecture: an HPSG theory is just a set of constraints, some of which happen to constrain lexical objects, while others constrain phrasal objects, etc. The TFS system only supports implicational statements with type antecedent though, not general negation, so that principles with complex antecedents cannot be directly encoded.

In line with the credo for reversible processing architectures (cf., Van Noord, 1993; Neumann, 1998; Minnen, 2001, and references therein), processing in TFS always amounts to applying constraints to whatever specification is given as the input—there is no specialized parsing or generation algorithm. This also leads us to the main disadvantage of the TFS approach: Since every linguistic constraint is treated in the same way, including type constraints on recursively defined data structures, TFS can run into significant control problems, which result in problems of efficiency and termination.

4.2.2 Systems based on a relational backbone

In the Comprehensive Unification Formalism (CUF, Dörre and Eisele, 1991; Dörre and Dorna, 1993), a theory is expressed using definite clauses as a relational extension of the description language (cf., Jaffar and Lassez, 1987; Höhfeld and Smolka, 1988). CUF thus essentially is a relational programming language like PROLOG (Colmerauer, 1993), but with feature terms as arguments of the relations instead of first order terms. While CUF provides some advanced features, such as complex coroutining capabilities, related approaches have focused on providing a lean integration of typed feature structures into Prolog such as GULP (Covington, 1994) or ProFIT (Erbach, 1995b).

An HPSG theory is implemented in such a setup by rewriting it as a logic program. In the resulting reorganized grammar, the recursive constraints are encoded on a different level (relations) than the linguistic data structure constraints (arguments to the relations). A query to the system is a call to one of the relations, and the system returns the instantiation of the arguments required by the called relation. Whether such a query results in parsing or generation (or some other kind of processing) depends entirely on how the grammar was encoded as a logic program and which processing regime is applied to it.

Compared to the computational setup exemplified by TFS, in which the constraints on the domain are expressed directly, the advantage of the relational approach is that the grammar writer determines the order of execution of goals by the way in which one encodes the grammar as a definite clause program. The definite clause encoding also allows the efficient processing techniques of logic programming to be used, e.g., clause indexing, Earley deduction, goal freezing, and specialized algorithms for parsing or generation can be encoded in this setup (cf., e.g., Erbach, 1995a, and references therein).

The disadvantage of the relational systems is that the organization and expression of a grammar is entirely different from the organization of an HPSG theory as a set of constraints. A grammar with a relational backbone in general is only related to the original linguistic HPSG theory on an intuitive level (unless the relational encoding is the result of compilation of a set of constraints, cf. Götz and Meurers, 1995).

Phrase structure as a specialized relational backbone Probably the largest class of systems, such as ALE (Carpenter and Penn, 1994), LKB (Copestake, 1993), and Troll (Gerdemann and Götz, 1996), has opted for a particular relational approach: phrase structure grammars. Phrase structure is a relation demanding a fixed number of daughters (at runtime) and having a designated argument (in HPSG typically encoded under PHON) that satisfies the condition that the list of tokens covered by the mother is the concatenation of that covered by all its daughters. The key advantage of a phrase structure based approach is that the most important recursive structure, syntactic constituency, is singled out and encoded in a format that readily supports efficient algorithms for parsing, which exploit the formal properties of the phrase structure relation for efficient indexing, memoization and other optimizations, instead of relying on resolution to tackle general definite clause programs. Some of the systems also make it possible to combine phrase structure grammars with definite clause programs, so that, e.g., a phrase structure rule can refer to a general definite relation for concatenating or shuffling lists.

The phrase structure based approach shares the general disadvantage of the relational encoding: that the organization and formulation of the grammar is different from that of the linguistic theory. Despite its somewhat confusing name, a Head-Driven Phrase Structure Grammar does not include phrase structure rules; as we saw in §3, an HPSG theory is a set of constraints, one of which typically encodes a set of immediate dominance schemata licensing syntactic structure, as discussed in §2.1. When one recodes such immediate dominance schemata into phrase structure rules, the specific restrictions inherent to phrase structure mentioned above have the effect that each immediate dominance constraint of the original linguistic theory results in a possibly very large set of phrase structure rules.

4.2.3 Combining paradigms

As mentioned in the previous section, a relational backbone is advantageous for efficient processing. Implicational constraints, on the other hand, support a more modular, underspecified encoding of grammars by making use of the elaborate structure that typed feature structures impose on the modeled domain. Several HPSG-based processing systems have thus been developed to combine the advantages of both encodings, such as the (new) ALE (Carpenter and Penn, 1994), TDL (Krieger, 1995), ConTroll (Götz and Meurers, 1997), the (new) LKB (Copestake, 2002), Hdrug (van Noord and Bouma, 1997), or TRALE (Meurers et al., 2002; Penn, 2004). This development is reminiscent of the integration of the paradigms of (constraint) logic programming, functional programming, and object-oriented programming in general purpose programming languages, such as LIFE (Aït-Kaci and Podelski, 1991) and OZ (http://www.mozart-oz.org/).

How the paradigms are combined in the HPSG-implementation systems differs from system to system, reflecting the linguistic or engineering focus under which they have been developed. Let us illustrate this with three examples:

The ConTroll system was developed to reflect the linguistic and formal architecture of HPSG as closely as possible. It supports interleaved processing of implicational constraints and relational expressions controlled through delay statements and prioritized deterministic computation (Götz and Meurers, 1997). The implicational constraints can have complex antecedents, and the system strictly enforces inferences based on the appropriateness conditions formulated in the signature (Gerdemann, 1995). Constraints are enforced lazily (Götz and Meurers, 1999) in a way that ensures that every description returned as a result has at least one instantiation that is most specific with respect to the signature defined, i.e., every solution returned by the system describes a non-empty set of linguistic objects. But the system lacks a phrase structure backbone for efficient processing with large scale grammars.

At the other end of the spectrum, the LKB is a system emphasizing efficient processing with a phrase structure backbone and basic type constraints (Copestake, 2002). The system supports efficient processing of large scale grammars and is at the heart of a rich array of natural language engineering efforts (cf., e.g., Oepen et al., 2000, and several articles mentioned in the next section). In support of efficiency, the system has a lean design: The description language does not include disjunction, negation, path-inequation or macros; the basic type constraints are not intended for recursive processing, and the system does not support definite relations so that recursive processing is limited to the phrase structure component. To avoid costly computation, constraints and appropriateness conditions are not strictly enforced, in the sense that a solution returned is not guaranteed to have any specific instantiations that are well-formed.

The TRALE system attempts to cover a middle ground by combining the definite relations and efficient phrase structure backbone of the ALE system (Penn and Munteanu, 2003) with implicational constraints with complex antecedents and enforcing some, but not all, of the inferences possible through appropriateness. In support of efficient processing, implicational constraints are applied only when the antecedent subsumes the description being processed (Penn, 2004), in contrast to ConTroll, where the interpretation of negation is classical, potentially enforcing costly exploration of many disjuncts (unless a delay statement is specified to obtain the interpretation adopted by TRALE).

4.3 Strands of research

HPSG-based processing is an active field of research, so that we want to use the remainder of this section to point out some of the strands of research.

In line with the current emphasis on human language technology and applied research, a major strand of work on HPSG-based processing is advancing the efficiency, robustness, and applicability of deep processing. This focus includes work on abstract machine compilation (cf., e.g., Wintner, 1997; Miyao et al., 2000), an efficient reimplementation of a lean, phrase structure based system in C++ (Callmeier, 2000), grammar compilation techniques (cf., e.g., Minnen et al., 1995; Minnen, 2001; Meurers and Minnen, 1997), and a wide range of algorithmic improvements and techniques (cf., e.g., Kiefer et al., 1999; Oepen et al., 2000; Munteanu and Penn, 2004).

A related line of work relieves the HPSG-based deep processing of some of the processing burden by combining it with shallow processing techniques (cf., e.g., Frank et al., 2003) or filtering with extracted context-free grammars (cf., e.g., Kiefer and Krieger, 2000; Torisawa et al., 2000). Some of this work is organized around the Deep Thought project (http://www.project-deepthought.net/), which investigates such hybrid methods for the purpose of information extraction. Stochastic extensions of HPSG have been developed (Brew, 1995; Abney, 1997) and such a combination of statistical and linguistic insights is, e.g., explored in the Delph-IN collaboration (http://www.delph-in.net/).

Another strand of research emphasizes and explores the relation of linguistic theory and grammar implementation (cf., e.g. Hinrichs et al., 2004). Work in this strand includes research exploring the expressive means needed for transparent and modular grammar design (cf., e.g., Meurers et al., 2003; Penn and Hoetmer, 2003; Penn, 2004), the development of a core architecture for multi-lingual grammar development (Bender et al., 2002), and research developing processing regimes for linearization-based HPSG (cf., e.g., Kasper et al., 1998; Daniels and Meurers, 2004; Penn and Haji-Abdolhosseini, 2003)—a paradigm providing modular and compact analyses for languages with relatively free word order (Müller, 2004). In the context of the MiLCA consortium (http://milca.sfs.nphil.uni-tuebingen.de/A4/), the relation between HSPG as linguistic formalisms, its formal foundations, and computational realization is explored from a pedagogical perspective (Richter, 2004b). This relation is also the topic of the CoGETI network (http://www.cl.uni-heidelberg.de/forschungsprojekte/cogeti).

A further strand of research concerns the integration of grammar implementation and computational semantics, as, e.g., reflected in the papers from the workshop on Semantics in Grammar Engineering included in HPSG (2004) and the HPSG Ellip project (http://semantics.phil.kcl.ac.uk/ellip/hpsg_ellip.html).

Finally, while we have focused on language processing from the perspective of computational linguistics, we should mention in closing that the HPSG paradigm has also been used for psycholinguistic research on human sentence processing. HPSG-based models of human language comprehension are, for example, explored in Semantics-Oriented Unification-based Language Processing (SOUL, Konieczny, 1996), which integrates constraints from linguistic, conceptual, and discourse knowledge.

References

Abney, S. P. (1997). Stochastic Attribute-Value Grammars. *Computational Linguistics* 23(4), 597–618.

ACL (2003). *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*. Sapporo, Japan. http://www.aclweb.org/anthology/.

ACL (2004). *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*. Barcelona, Spain. http://www.aclweb.org/anthology/.

Ades, A. and Steedman, M. (1982). On the Order of Words. *Linguistics and Philosophy* 5, 519–558.

Aït-Kaci, H. and Podelski, A. (1991). Towards a Meaning of LIFE. In J. Maluszynski and M. Wirsing (eds.), *Programming Language Implementation and Logic Programming: Proc. of the 3rd International Symposium PLILP'91, Passau*, Berlin, Heidelberg: Springer, pp. 255–274.

Amaral, L. A. (2005). The Double Subcategorization of *Tough*-Adjectives in Brazilian Portuguese. In R. D. Levine and W. D. Meurers (eds.), *Locality of Grammatical Relationships*, Columbus, Ohio: Dept. of Linguistics, The Ohio State University, no. 58 in OSU Working Papers in Linguistics. http://ling.osu.edu/publications/osu_wpl/.

Baker, C. L. (1978). *Introduction to Generative-Transformational Grammar*. New York: Prentice-Hall.

Balari, S. and Dini, L. (eds.) (1997). *HPSG in Romance*. Stanford, CA: CSLI Publications.

Bear, J. (1981). Gaps as Syntactic Features. Master's thesis, University of Texas at Austin.

Bender, E. M., Flickinger, D. and Oepen, S. (2002). The Grammar Matrix: An Open-Source Starter-Kit for the Rapid Development of Cross-Linguistically Consistent Broad-Coverage Precision Grammars. In J. Carroll, N. Oostdijk and R. Sutcliffe (eds.), *Procedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*. Taipei, Taiwan, pp. 8–14. http://www-csli.stanford.edu/~bender/papers/gee03.pdf.

Bird, S. and Klein, E. (1994). Phonological Analysis in Typed Feature Systems. *Computational Linguistics* 20(3), 455–491.

Bolc, L., Czuba, K., Kupść, A., Marciniak, M., Mykowiecka, A. and Przepiórkowski, A. (1996). *A Survey of Systems for Implementing HPSG Grammars*. Technical Report 814, IPI PAN (Institute of Computer Science, Polish Academy of Sciences), Warsaw, Poland. http://www.ipipan.waw.pl/~adamp/Papers/1996-survey/.

Borsley, R. and Jones, B. M. (2005). *Welsh Negation and Grammatical Theory*. Bangor: University of Wales Press.

Borsley, R. D. and Przepiórkowski, A. (1999). *Slavic in Head-Driven Phrase Structure Grammar*. Stanford, CA: 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. ftp://csli-ftp.stanford.edu/linguistics/sag/bms-nllt.ps.

Bouma, G., van Eynde, F. and Flickinger, D. (2000). Constraint-based Lexicons. In F. van Eynde, D. Gibbon and I. Schuurman (eds.), *Lexicon Development for Speech and Language Processing*. Dordrecht: Kluwer.

Brame, M. (1976). *Conjectures and Refutations in Syntax and Semantics*. New York: North-Holland.

Bresnan, J. (ed.) (1982). *The Mental Representation of Grammatical Relations*. Cambridge, MA: MIT Press.

Brew, C. (1995). Stochastic HPSG. In EACL (1995), pp. 83–89.

Callmeier, U. (2000). PET - a platform for experimentation with efficient HPSG processing techniques. *Natural Language Engineering* 6(1), 99–107.

Carpenter, B. (1992). *The Logic of Typed Feature Structures – With Applications to Unification Grammars, Logic Programs and Constraint Resolution*, vol. 32. Cambridge, UK: Cambridge University Press.

Carpenter, B. and Penn, G. (1994). *ALE – The Attribute Logic Engine, User's Guide, Version 2.0.1, December 1994*. Tech. rep., Computational Linguistics Program, Philosopy Department, Carnegie Mellon University.

Chomsky, N. (1981). *Lectures on Government and Binding*. Dordercht: Foris.

Colmerauer, A. (1993). The Birth of Prolog. In *The Second ACM-SIGPLAN History of Programming Languages Conference*. ACM SIGPLAN Notices, pp. 37–52.

Copestake, A. (1993). *The Compleat LKB*. Technical report 316, University of Cambridge Computer Laboratory. http://www-csli.stanford.edu/~aac/papers/compleat.ps.gz.

Copestake, A. (2002). *Implementing Typed Feature Structure Grammars*. Stanford, CA: CSLI Publications.

Copestake, A., Flickinger, D., Sag, I. and Pollard, C. (To appear). Minimal Recursion Semantics: an introduction. *Research on Language and Computation* .

Covington, M. A. (1994). *GULP 3.1: An extension of Prolog for Unification-Based Grammar*. Research Report AI-1994-06, Artificial Intelligence Center, The University of Georgia, Athens, Georgia. http://www.ai.uga.edu/~mc/ai199406.pdf.

Daniels, M. and Meurers, D. (2004). A grammar formalism and parser for linearization-based HPSG. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING-04)*. Geneva, Switzerland, pp. 169–175.

De Kuthy, K. (2002). *Discontinuous NPs in German — A Case Study of the Interaction of Syntax, Semantics and Pragmatics*. Stanford, CA: CSLI Publications.

De Kuthy, K. and Meurers, W. D. (2003). The secret life of focus exponents, and what it tells us about fronted verbal projections. In HPSG (2003b), pp. 97–110. http://ling.osu.edu/~dm/papers/dekuthy-meurers-hpsg03.html.

Dörre, J. and Dorna, M. (1993). CUF – a formalism for linguistic knowledge representation. In J. Dörre (ed.), *Computational aspects of constraint based linguistic descriptions I*. DYANA-2 Deliverable R1.2.A, Uni Stuttgart, pp. 1–22. http://www.illc.uva.nl/DYANA/R1.2.A/Doerre-Dorna.pdf.

Dörre, J. and Eisele, A. (1991). *A Comprehensive Unification-Based Grammar Formalism*. DYANA Deliverable R3.1.B, Universität Stuttgart. http://elib.uni-stuttgart.de/opus/volltexte/1999/95/.

EACL (1995). *Proceedings of the Seventh Conference on European Chapter of the Association for Computational Linguistics*. http://acl.ldc.upenn.edu/E/E95/.

Egg, M., Koller, A. and Niehren, J. (2001). The Constraint Language for Lambda Structures. *Journal of Logic, Language and Information* 10(4), 457–485.

Emele, M. and Zajac, R. (1990). Typed Unification Grammars. In H. Karlgreen (ed.), *Proceedings of the 13th Conference on Computational Linguistics (COLING-90)*. Helsinki. http://www.aclweb.org/anthology/C90-3052.

Engdahl, E. and Vallduví, E. (1994). Information Packaging and Grammar Architecture: A constraint-based approach. In E. Engdahl (ed.), *Integrating Information Structure into Constraint-based and Categorial Approaches*, University of Edinburgh, DYANA-2 Deliverable R1.3.B, pp. 39–79. http://www.illc.uva.nl/DYANA/R1.3.B/Engdahl-Vallduvi.pdf.

Erbach, G. (1995a). Bottom-Up Earley Deduction for Preference-Driven Natural Language Processing. Ph.D. thesis, Universität des Saarlandes. http://www.coli.

uni-sb.de/~erbach/pub/diss/.

Erbach, G. (1995b). ProFIT: Prolog with Features, Inheritance, and Templates. In EACL (1995). http://www.aclweb.org/anthology/E95-1025.

Erjavec, T. (1996). Unification, Inheritance, and Paradigms in the Morphology of Natural Languages. Ph.D. thesis, University of Ljubljana. http://nl.ijs.si/et/Thesis/.

Estival, D., Lavelli, A., Netter, K. and Pianesi, F. (eds.) (1997). *Proceedings of the Workshop "Computational Environments for Grammar Development and Linguistic Engineering (ENVGRAM)" held in conjunction with the 35th Annual Meeting of the ACL and 8th Conference of the EACL*. Madrid: Universidad Nacional de Educación a Distancia.

Fillmore, C. and Kay, P. (1999). Grammatical constructions and linguistic generalizations: the *what's x doing y?* construction. *Language* 75, 1–33.

Frank, A., Becker, M., Crysmann, B., Kiefer, B. and Schäfer, U. (2003). Integrated Shallow and Deep Parsing: TopP meets HPSG. In ACL (2003), pp. 104–111. http://www.aclweb.org/anthology/P03-1014.

Frank, A. and Reyle, U. (1995). Principle Based Semantics for HPSG. In EACL (1995). http://acl.ldc.upenn.edu/E/E95/.

Gazdar, G. (1981). Unbounded dependencies and coordinate structure. *Linguistic Inquiry* 12, 155–84.

Gazdar, G., Klein, E., Pullum, G. K. and Sag, I. A. (1985). *Generalized phrase structure grammar*. Cambridge, Massachusetts: Harvard University Press.

Gerdemann, D. (1995). Open and Closed World Types in NLP Systems. In *Proceedings of the DGfS Fachtagung Computerlinguistik*. Düsseldorf, Germany. http://www.sfs.uni-tuebingen.de/~dg/open_world.ps.

Gerdemann, D. and Götz, T. (1996). Troll manual. SFB 340, SfS, Universität Tübingen.

Ginzburg, J. and Sag, I. (2001). *Interrogtive Investigations*. Stanford, CA: CSLI Publications.

Götz, T. and Meurers, W. D. (1995). Compiling HPSG Type Constraints into Definite Clause Programs. In *Proceedings of the 33nd Annual Meeting of the Association for Computational Linguistics (ACL-95)*. Cambridge, MA, pp. 85–91. http://www.aclweb.org/anthology/P95-1012.

Götz, T. and Meurers, W. D. (1997). Interleaving Universal Principles and Relational Constraints over Typed Feature Logic. In *Proceedings of the 35th Annual Meeting of the ACL and 8th Conference of the EACL*. Madrid, pp. 1–8. http://www.aclweb.org/anthology/P97-1001.

Götz, T. and Meurers, W. D. (1999). The importance of being lazy – Using lazy evaluation to process queries to HPSG grammars. In Webelhuth et al. (1999), pp. 249–264.

Gregory, H. (1997). A ProFIT grammar and reconstruction procedure for ellipsis resolution. HPSG Ellip Project, SOAS, University of London, http://semantics.phil.kcl.ac.uk/ellip/papers/manual.ps.gz.

Hinrichs, E. W., Meurers, W. D. and Wintner, S. (2004). Linguistic Theory and Grammar Implementation: Introduction to this Special Issue. *Research on Language and Computation* 2(2), 155–163.

Hinrichs, E. W. and Nakazawa, T. (1994). Linearizing AUXs in German Verbal Complexes. In Nerbonne et al. (1994), pp. 11–37.

Höhfeld, M. and Smolka, G. (1988). *Definite relations over constraint languages*. LILOG technical report 53, IBM Deutschland GmbH. http://www.ps.uni-sb.de/Papers/paper_info.php?label=LR-53.

Höhle, T. N. (1999). An Architecture for Phonology. In Borsley and Przepiórkowski (1999), pp. 61–90.

Holler, A. (2001). Empirische und theoretische Aspekte der Grammatik der weiterführenden Relativsätze. Ph.D. thesis, Universität Tübingen.

HPSG (2003a). *Proceedings of the 9th International Conference on Head-Driven Phrase Structure Grammar*. J.-B. Kim and S. Wechsler (eds.). Stanford, CA: CSLI Publications. http://cslipublications.stanford.edu/HPSG/3/.

HPSG (2003b). *Proceedings of the 10th International Conference on Head-Driven Phrase Structure Grammar*. S. Müller (ed.). Stanford, CA: CSLI Publications. http://cslipublications.stanford.edu/HPSG/4/.

HPSG (2004). *Proceedings of the 11th International Conference on Head-Driven Phrase Structure Grammar*. S. Müller (ed.). Stanford, CA: CSLI Publications. http://cslipublications.stanford.edu/HPSG/5/.

Hukari, T. E. and Levine, R. D. (1995). Adjunct Extraction. *Journal of Linguistics* 31, 195–226.

Iida, M. (1995). *Context and Binding in Japanese*. Stanford, CA: CSLI Publications.

Jackendoff, R. (1977). *X-bar Syntax: A Study of Phrase Structure*. Cambridge, MA: MIT Press.

Jaffar, J. and Lassez, J.-L. (1987). Constraint Logic Programming. In ACM (ed.), *Proceedings of the 14th ACM Symposium on Principles of Programming Languages, Munich*. pp. 111–119.

Johnson, M. (1988). *Attribute-value Logic and the Theory of Grammar*, vol. 16. Stanford, CA: CSLI Publications.

Kasper, R. T., Calcagno, M. and Davis, P. C. (1998). Know When to Hold 'Em: Shuffling Deterministically in a Parser for Nonconcatenative Grammars. In *Proceedings of the 17th International Conference on Computational Linguistics (COLING) and the 36th Annual meeting of the ACL*. Montreal, Canada, pp. 663–669.

Kathol, A. (2000). *Linear Syntax*. Oxford: Oxford University Press. Published version of 1995 Ph. D. thesis, Ohio State University.

Kathol, A. and Przepiórkowski, A. (To appear). Advanced topics in HPSG. In R. Borsley and K. Börjars (eds.), *Nontransformational Syntax*, Oxford: Blackwell.

Kiefer, B. and Krieger, H.-U. (2000). A Context-Free Approximation of Head-Driven Phrase Structure Grammar. In *Proceedings of the Sixth International Workshop on Parsing Technologies (IWPT 2000)*. pp. 135–146.

Kiefer, B., Krieger, H.-U., Carroll, J. and Malouf, R. (1999). A Bag of Useful Techniques for Efficient and Robust Parsing. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL-99)*. College Park, MD: ACL, pp. 473–480. http://www.aclweb.org/anthology/P99-1061.

King, P. J. (1989). A logical formalism for head-driven phrase structure grammar. Ph.D. thesis, University of Manchester.

King, P. J. (1994). *An Expanded Logical Formalism for Head-Driven Phrase Structure Grammar*. No. 59 in Arbeitspapiere des SFB 340, Universität Tübingen. http://www.sfs.uni-tuebingen.de/sfb/reports/berichte/59/59abs.html.

King, P. J. (1999). Towards Truth in Head-Driven Phrase Structure Grammar. In V. Kordoni (ed.), *Tübingen Studies in Head-Driven Phrase Structure Grammar*, No. 132 in Arbeitspapiere des SFB 340, Universität Tübingen, pp. 301–352. http://www.sfs.uni-tuebingen.de/sfb/reports/berichte/132/132abs.html.

Kiss, T. (1995). *Infinitive Komplementation*. Tübingen: Max Niemeyer Verlag. Published version of 1992 Ph. D. thesis, Bergische Universität – Gesamthochschule Wuppertal.

Koenig, J.-P. and Davis, A. (2003). Semantically Transparent Linking in HPSG. In HPSG (2003b), pp. 222–235.

Konieczny, L. (1996). Human Sentence Processing: a Semantics-Oriented Parsing Approach. Ph.D. thesis, Universität Freiburg. http://cognition.iig.uni-freiburg.de/team/members/konieczny/publ/DissLars.pdf.

Krieger, H.-U. (1995). *TDL*—A Type Description Language for Constraint-Based Grammars. Foundations, Implementation, and Applications. Ph.D. thesis, Universität des Saarlandes.

Krieger, H.-U. and Nerbonne, J. (1993). Feature-Based Inheritance Networks for Computational Lexicons. In T. Briscoe, A. Copestake and V. de Paiva (eds.), *Inheritance, Defaults, and the Lexicon*, New York: Cambridge University Press, pp. 90–136.

Levine, R. (2001). *Tough* constructions and the extraclausal propagation of argument descriptions. In *Proceedings of the 7th International Conference on Head-Driven Phrase Structure Grammar*, D. Flickinger and A. Kathol (eds.). Stanford, CA: CSLI Publications.

Levine, R. (2003). Adjunct valents: cumulative scoping adverbial constructions and impossible descriptions. In HPSG (2003a).

Matiasek, J. (1994). Conditional constraints in a CLP-based HPSG implementation. In H. Trost (ed.), *KONVENS '94*. Vienna, Austria, pp. 230–239.

Meurers, W. D. (1994). On Implementing an HPSG Theory – Aspects of the Logical Architecture, the Formalization, and the Implementation of Head-Driven Phrase Structure Grammars. In E. W. Hinrichs, W. D. Meurers and T. Nakazawa (eds.), *Partial-VP and Split-NP Topicalization in German – An HPSG Analysis and its Implementation*, No. 58 in Arbeitspapiere des SFB 340, Universität Tübingen, pp. 47–155. http://ling.osu.edu/~dm/on-implementing.html.

Meurers, W. D. (2000). *Lexical Generalizations in the Syntax of German Non-Finite Constructions*. No. 145 in Arbeitspapiere des SFB 340. Universität Tübingen. (= Ph. D. thesis, Universität Tübingen, 1999). http://w210.ub.uni-tuebingen.de/dbt/volltexte/2000/118.

Meurers, W. D. (2001). On expressing lexical generalizations in HPSG. *Nordic Journal of Linguistics* 24(2), 161–217. Special issue on "The Lexicon in Linguistic Theory". http://ling.osu.edu/~dm/papers/lexical-generalizations.html.

Meurers, W. D. and Kiss, T. (eds.) (2001). *Constraint-Based Approaches to Germanic Syntax*. Stanford, CA: CSLI Publications.

Meurers, W. D., Kuthy, K. D. and Metcalf, V. (2003). Modularity of grammatical constraints in HPSG-based grammar implementations. In M. Siegel, F. Fouvry, D. Flickinger and E. Bender (eds.), *Proceedings of the ESSLLI '03 workshop "Ideas and Strategies for Multilingual Grammar Development"*. Vienna, Austria. http://ling.osu.edu/~dm/papers/meurers-dekuthy-metcalf-03.html.

Meurers, W. D. and Minnen, G. (1997). A Computational Treatment of Lexical Rules in HPSG as Covariation in Lexical Entries. *Computational Linguistics* 23(4), 543–568. http://www.aclweb.org/anthology/J97-4003.

Meurers, W. D., Penn, G. and Richter, F. (2002). A Web-based Instructional Platform for Constraint-Based Grammar Formalisms and Parsing. In D. Radev and C. Brew (eds.), *Effective Tools and Methodologies for Teaching NLP and CL*. pp. 18–25. http://www.aclweb.org/anthology/W02-0103.

Minnen, G. (2001). *Efficient Processing with Constraint-Logic Grammars Using Grammar Compilation*. Stanford, CA: CSLI Publications.

Minnen, G., Gerdemann, D. and Götz, T. (1995). Off-line Optimization for Earley-style HPSG Processing. In EACL (1995).

Miyao, Y., Makino, T., Torisawa, K. and Tsujii, J.-I. (2000). The LiLFeS abstract machine and its evaluation with the LinGO grammar. *Natural Language Engineering* 6(1), 47–61.

Monachesi, P. (1999). *A lexical approach to Italian cliticization*, vol. 84. Stanford, CA: CSLI Publications.

Moshier, M. A. and Rounds, W. C. (1987). A Logic for Partially Specified Data Structures. In *Proceedings of the 14th ACM Symposium on Principles of Programming Languages. München, Germany*. pp. 156–167.

Müller, S. (1999). *Deutsche Syntax deklarativ. Head-Driven Phrase Structure Grammar für das Deutsche*. Tübingen: Max Niemeyer Verlag.

Müller, S. (2002). *Complex Predicates: Verbal Complexes, Resultative Constructions, and Particle Verbs in German*. Stanford, CA: CSLI Publications.

Müller, S. (2004). Continuous or Discontinuous Constituents? A Comparison between Syntactic Analyses for Constituent Order and Their Processing Systems. In Hinrichs et al. (2004), pp. 209–257.

Müller, S. (2004). Phrasal or Lexical Constructions? A Reply to Goldberg and Jackendoff. Ms. Universität Bremen. http://www.cl.uni-bremen.de/~stefan/Pub/phrasal.html.

Munteanu, C. and Penn, G. (2004). Optimizing Typed Feature Structure Grammar Parsing through Non-Statistical Indexing. In ACL (2004).

Nerbonne, J., Netter, K. and Pollard, C. (eds.) (1994). *German in Head-Driven Phrase Structure Grammar*. Stanford, CA: CSLI Publications.

Neumann, G. (1998). Interleaving Natural Language Parsing and Generation Through Uniform Processing. *Artificial Intelligence* 99, 121–163.

Oepen, S., Flickinger, D., Uszkoreit, H. and Tsujii, J.-I. (2000). Introduction to the Special Issue on Efficient Processing with HPSG: Methods, Systems, Evaluation. *Natural Language Engineering* 6(1).

Penn, G. (2004). Balancing Clarity and Efficiency in Typed Feature Logic through Delaying. In ACL (2004).

Penn, G. and Haji-Abdolhosseini, M. (2003). Topological Parsing. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL-03)*. Budapest, Hungary. http://www.aclweb.org/anthology/E03-1039.

Penn, G. and Hoetmer, K. (2003). In Search of Epistemic Primitives in the English Resource Grammar. In HPSG (2003b).

Penn, G. and Munteanu, C. (2003). A Tabulation-Based Parsing Method that Reduces Copying. In ACL (2003).

Pollard, C. (1999). Strong Generative Capacity in HPSG. In Webelhuth et al. (1999), pp. 281–297.

Pollard, C. (2001). Cleaning the HPSG Garage: Some Problems and some Proposals.

Ms. Summer School on Constraint-based Grammar. Trondheim, Norway. http://www.ling.hf.ntnu.no/HPSG2001/summer_school/cp_garage.ps.

Pollard, C. and Sag, I. A. (1987). *Information-based Syntax and Semantics, Vol. 1: Fundamentals*. Stanford, CA: CSLI Publications.

Pollard, C. and Sag, I. A. (1992). Anaphors in English and the Scope of Binding Theory. *Linguistic Inquiry* 23(2), 261–303.

Pollard, C. and Sag, I. A. (1994). *Head-Driven Phrase Structure Grammar*. Chicago, IL: University of Chicago Press.

Reape, M. (1996). Getting Things in Order. In H. Bunt and A. van Horck (eds.), *Discontinuous Constituency*, Berlin, New York: Mouton de Gruyter.

Richter, F. (2000). A Mathematical Formalism for Linguistic Theories with an Application in Head-Driven Phrase Structure Grammar. Ph.D. thesis, Universität Tübingen. http://w210.ub.uni-tuebingen.de/dbt/volltexte/2004/1203/.

Richter, F. (2004a). Foundations of Lexical Resource Semantics. Habilitationsschrift, Universität Tübingen.

Richter, F. (2004b). *A Web-based Course in Grammar Formalisms and Parsing*. Textbook, MiLCA project A4, SfS, Universität Tübingen. http://milca.sfs.uni-tuebingen.de/A4/Course/PDF/gramandpars.pdf.

Riehemann, S. Z. (1998). Type-Based Derivational Morphology. *The Journal of Comparative Germanic Linguistics* 2, 49–77.

Rounds, W. C. and Kasper, R. T. (1986). A Complete Logical Calculus for Record Structures Representing Linguistic Information. In *Proceedings of the 15th Annual IEEE Symposium on Logic in Computer Science, Cambridge, MA*. pp. 38–43.

Sag, I. (1997). English relative clause constructions. *Journal of Linguistics* 33, 431–484.

Sag, I. A. (2003). Coordination and Underspecification. In HPSG (2003a), pp. 267–291.

Sailer, M. (2003). *Combinatorial Semantics and Idiomatic Expressions in Head-Driven Phrase Structure Grammar*. No. 161 in Arbeitspapiere des SFB 340. Universität Tübingen. (= Ph. D. thesis, Universität Tübingen, 2000). http://w210.ub.uni-tuebingen.de/dbt/volltexte/2003/916/.

Smolka, G. (1988). *A Feature Logic with Subsorts*. LILOG technical report 33, IBM Deutschland GmbH. http://www.ps.uni-sb.de/Papers/abstracts/LR-33.html.

Torisawa, K., Nishida, K., Miyao, Y. and Tsujii, J. (2000). An HPSG parser with CFG filtering. *Natural Language Engineering* 6(1), 63–80.

Van Noord, G. (1993). Reversibility in Natural Language Processing. Ph.D. thesis, University of Utrecht, Utrecht, The Netherlands. http://odur.let.rug.nl/~vannoord/papers/diss/diss.html.

van Noord, G. and Bouma, G. (1997). Hdrug, A Flexible and Extendible Development Environment for Natural Language Processing. In Estival et al. (1997).

Webelhuth, G., Koenig, J.-P. and Kathol, A. (eds.) (1999). *Lexical and Constructional Aspects of Linguistic Explanation*. Stanford, CA: CSLI Publications.

Wintner, S. (1997). An Abstract Machine for Unification Grammars. Ph.D. thesis, Technion, Israel Institute of Technology, Haifa, Israel. http://arxiv.org/abs/cmp-lg/9709013.

Zaenen, A. (1983). On syntactic binding. *Linguistic Inquiry* 14, 469–504.