Chapter 2

The evolution of HPSG

Dan Flickinger
Stanford University

Carl Pollard
Ohio State Universtiy

Tom Wasow
Stanford University

HPSG was developed to express insights from theoretical linguistics in a precise formalism that was computationally tractable. It drew ideas from a wide variety of traditions in linguistics, logic, and computer science. Its chief architects were Carl Pollard and Ivan Sag, and its most direct precursors were Generalized Phrase Structure Grammar and Head Grammar. The theory has been applied in the construction of computational systems for the analysis of a variety of languages; a few of these systems have been used in practical applications. This chapter sketches the history of the development and application of the theory.

Introduction

From its inception in 1983, HPSG was intended to serve as a framework for the formulation and implementation of natural language grammars which are (i) linguistically motivated, (ii) formally explicit, and (iii) computationally tractable. These desiderata are reflective of HPSG's dual origins as an academic linguistic theory and as part of an industrial grammar implementation project with an eye toward potential practical applications. Here (i) means that the grammars are intended as scientific theories about the languages in question, and that the analyses the grammars give rise to are transparently relatable to the predictions

Dan Flickinger, Carl Pollard & Tom Wasow. 2018. The evolution of HPSG. in Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook*, 7–40. Berlin: Language Science Press. DOI:??

(empirical consequences) of those theories. Thus HPSG shares the general concerns of the theoretical linguistics literature, including distinguishing between well-formed and ill-formed expressions and capturing linguistically significant generalizations. (ii) means that the notation for the grammars and its interpretation have a precise grounding in logic, mathematics, and theoretical computer science, so that there is never any ambiguity about the intended meaning of a rule or principle of grammar, and so that grammars have determinate empirical consequences. (iii) means that the grammars can be translated into computer programs that can handle linguistic expressions embodying the full range of complex interacting phenomena that naturally occur in the target languages, and can do so with a tolerable cost in space and time resources.

The two principal architects of HPSG were Carl Pollard and Ivan Sag, but a great many other people made important contributions to its development. Many, but by no means all, are cited in the chronology presented in the following sections. There are today a number of groups of HPSG researchers around the world, in many cases involved in building HPSG-based computational systems. While the number of practitioners is relatively small, it is a very active community that holds annual meetings and publishes quite extensively. Hence, although Pollard no longer works on HPSG and Sag died in 2013, the theory is very much alive, and still evolving.

1 Precursors

HPSG arose between 1983 and 1985 from the complex interaction between two lines of research in theoretical linguistics: (i) work on context-free Generative Grammar (CFG) intitiated in the late 1970s by Gerald Gazdar and Geoffrey Pullum, soon joined by Ivan Sag, Ewan Klein, Tom Wasow, and others, resulting in the framework referred to as Generalized Phrase Structure Grammar (GPSG: Gazdar, Klein, Pullum & Sag (1985)); and (ii) Carl Pollard's Stanford dissertation research, under Sag and Wasow's supervision, on Generalized Context-Free Grammar, and more specifically Head Grammar (HG: Pollard (1984)).

1.1 Generalized Phrase Structure Grammar

In the earliest versions of Generative Grammar (Chomsky 1957), the focus was on motivating transformations to express generalizations about classes of sentences. In the 1960s, as generative linguists began to attend more explicitly to meaning, a division arose between those advocating using the machinery of transformations to capture semantic generalizations and those advocating the use of other types of formal devices. This division became quite heated, and was subsequently dubbed "the linguistic wars" (see Newmeyer (1980: Chapter 5)). Much of the work in theoretical syntax and semantics during the 1970s explored ways to constrain the power of transformations (see especially, Chomsky (1973) and Chomsky & Lasnik (1977)), and non-transformational approaches to the analysis of meaning (see especially Montague (1974) and Dowty (1979)).

These developments led a few linguists to begin questioning the central role transformations had played in the syntactic research of the preceding two decades (notably, Bresnan (1978)). This questioning of Transformational Grammar (TG) culminated in a series of papers by Gerald Gazdar, which (in those pre-internet days) were widely distributed as paper manuscripts. The project that they laid out was succinctly summarized in one of Gazdar's later publications (Gazdar 1981: 155) as follows:

> Consider eliminating the transformational component of a generative grammar. (Gazdar 1981: 155)

The framework that emerged became known as Generalized Phrase Structure Grammar; a good account of its development is Ted Briscoe's interview of Gazdar in November 2000.[1]

GPSG developed in response to several criticisms leveled against transformational grammar. First, TG was highly underformalized, to the extent that it was unclear what its claims—and the empirical consequences of those claims—amounted to; CFG, by comparison, was a simple and explicit mathematical formalism. Second, given the TG architecture of a context-free base together with a set of transformations, the claimed necessity of transformations was standardly justified on the basis of arguments that CFGs were insufficiently expressive to serve as a general foundation for NL grammar; but Pullum & Gazdar (1982) showed all such arguments presented up to that time to be logically flawed or else based on false empirical claims. And third, closely related to the previous point, they showed that transformational grammarians had been insufficiently resourceful in exploiting what expressive power CFGs *did* possess, especially through the use of complex categories bearing features whose values might themselves bear features of their own. For example, coordinate constructions and unbounded dependency constructions had long served as prime exemplars of the need for transformations, but Gazdar (1981) was able to show that both kinds of

[1] https://nlp.fi.muni.cz/~xjakub/briscoe-gazdar/, 2018-08-21.

constructions, as well as interactions between them, did in fact yield straightforward analysis within the framework of a CFG.

Gazdar and Pullum's early work in this vein was quickly embraced by Sag and Wasow at Stanford University, both formally inclined former students of Chomsky's, who saw it as the logical conclusion of a trend in Chomskyan syntax toward constraining the tranformational component. That trend, in turn, was a response, at least in part, to (i) the demonstration by Peters & Ritchie (1973) that Chomsky's (1965) Standard Theory, when precisely formalized, was totally unconstrained, in the sense of generating all recursively enumerable languages; and (ii) the insight of Emonds (1976) that most of the transformations proposed up to that time were "structure-preserving" in the sense that the trees they produced were isomorphic to ones that were base-generated. Besides directly addressing these issues of excess power and structure preservation, the hypothesis that NLs were context-free also had the advantage that CFGs were well-known by computer scientists to have decidable recognition problems and efficient parsing algorithms, facts which seemed to have some promise of bearing on questions of the psychological plausibility and computational tractability of the grammars in question.

Aside from serving as a framework for theoretical linguistic research, GPSG also provided the theoretical underpinnings for a natural language processing (NLP) project established in 1981 by Egon Loebner at Hewlett-Packard Laboratories in Palo Alto. This project, which led in due course to the first computer implementation of HPSG, is described below.

1.2 Head Grammar

Pollard, with a background in pure mathematics, Chinese historical phonology, and 1930s–1950s-style American structural linguistics, arrived at Stanford in 1979 with the intention of getting a Ph.D. in Chinese linguistics, but was soon won over to theoretical syntax by Wasow and Sag. He had no exposure to Chomskyan linguistics, but was immediately attracted to the emerging nontransformational approaches, especially the early GPSG papers and the contemporaneous forms of CG in Bach (1979; 1980) and Dowty (1982a,b), in part because of their formal simplicity and rigor, but also because the formalism of CFG was (and is) easy to read as a more technically precise rendering of structualist ideas about syntax (as presented, e.g., in Bloomfield (1933) and Hockett (1958)).

Although Pullum & Gazdar (1982) successfully refuted all published arguments to date that CFGs were inadequate for analyzing NLs, by the following year, Stuart Shieber had developed an argument (published in Shieber (1985)), which was (and remains) generally accepted as correct, that there could not be a CFG that accounted for the cross-serial dependencies in Swiss German; and Chris Culy showed, in his Stanford M.A. thesis (cf. Culy (1985)), that the presence of reduplicative compounding in Bambara precluded a CF analysis of that language. At the same time, Bach and Dowty (independently) had been experimenting with generalizations of traditional A-B (Ajdukiewicz-Bar Hillel) CG which allowed for modes of combining strings (such as reduplication, wrapping, insertion, cliticization, and the like) in addition to the usual concatenation. This latter development was closely related to a wider interest among nontransformational linguists of the time in the notion of discontinuous constituency, and also had an obvious affinity to Hockett's (1954) item-and-process conception of linguistic structure, albeit at the level of words and phrases rather than morphemes. One of the principal aims of Pollard's dissertation work was to provide a general framework for syntactic (and semantic) analysis that went beyond—but not too far beyond—the limits of CFG in a way that took such developments into account.

Among the generalizations of CFG that Pollard studied, special attention was given to HGs, which differ from CFGs in two respects: (i) the role of strings was taken over by headed strings, essentially strings with a designation of one of its words as its head; and (ii) besides concatenation, headed strings can also be combined by inserting one string directly to the left or right of another string's head. An appendix of his dissertation (Pollard 1984) provided an analysis of discontinuous constituency in Dutch, and that analysis also works for Swiss German. In another appendix, Pollard used a generalization of the CKY algorithm to prove that the head languages (HLs, the languages analyzed by HGs) shared with CFLs the property of deterministic polynomial time recognition complexity, but of order n^7, subsequently reduced by Kasami, Seki & Fujii (1989) to n^6, as compared with order n^3 for CFLs. For additional formal properties of HGs, see Roach (1987). Vijay-shanker & Weir (1994) proved that HGs had the same weak generative capacities as three other grammar formalisms (Combinatory Categorial Grammar (Steedman 1987; 1990), Lexicalized Tree-Adjoining Grammar (Shabes 1990), and Linear Indexed Grammar (Gazdar 1988)), and the corresponding class of languages became known as 'mildly context sensitive'.

Although the handling of linearization in HG seems not to have been pursued further within the HPSG framework, the ideas that (i) linearization had to involve data structures richer than strings of phoneme strings, and (ii) the way these structures were linearized had to involve operations other than mere concatenation, were implicit in subsequent HPSG work, starting with Pollard & Sag's (1987) Constituent Order Principle (which was really more of a promissory note than an actual principle). These and related ideas would become more fully fleshed out a decade later within the linearization grammar avatar of HPSG developed by Reape (1996), Reape (1992), and Kathol (1995; 2000). On the other hand, two other innovations of HG, both related to the system of syntactic features, were incorporated into HPSG, and indeed should probably be considered the defining characteristics of that framework, namely the list-valued SUBCAT and SLASH features, discussed below.

2 The HP NLP project

Work on GPSG culminated in the 1985 book *Generalized Phrase Structure Grammar* by Gazdar, Klein, Pullum, and Sag. During the writing of that book, Sag taught a course on the theory, with participation of his co-authors. The course was attended not only by Stanford students and faculty, but also by linguists from throughout the area around Stanford, including the Berkeley and Santa Cruz campuses of the University of California, as well as people from nearby industrial labs. One of the attendees at this course was Anne Paulson, a programmer from Hewlett-Packard (HP) Laboratories in nearby Palo Alto, who had some background in linguistics from her undergraduate education at Brown University. Paulson told her supervisor at HP Labs, Egon Loebner, that she thought the theory could be implemented and might be turned into something useful. Loebner, a multi-lingual polymathic engineer, had no background in linguistics, but he was intrigued, and invited Sag to meet and discuss setting up a natural language processing project at HP. Sag brought along Gazdar, Pullum, and Wasow. This led to the creation of the project that eventually gave rise to HPSG. Gazdar, who would be returning to England relatively soon, declined the invitation to be part of the new project, but Pullum, who had taken a position at the University of California at Santa Cruz (about an hour's drive from Palo Alto), accepted. So the project began with Sag, Pullum, and Wasow hired on a part-time basis to work with Paulson and two other HP programmers, John Lamping and Jonathan King, to implement a GPSG of English at HP Labs. J. Mark Gawron, a linguistics graduate student from Berkeley who had attended Sag's course, was very soon added to the team.

The initial stages consisted of the linguists and programmers coming up with a notation that would serve the purposes of both. Once this was accomplished, the linguists set to work writing a grammar of English in Lisp to run on the DEC-20 mainframe computer that they all worked on. The first publication coming out of this project was a 1982 Association for Computational Linguistics paper

(Gawron et al. 1982). The paper's conclusion (p. 80) begins:

> What we have outlined is a natural language system that is a direct implementation of a linguistic theory. We have argued that in this case the linguistic theory has the special appeal of computational tractability (promoted by its context-freeness), and that the system as a whole offers the hope of a happy marriage of linguistic theory, mathematical logic, and advanced computer applications. (Gawron et al. 1982: 80)

This goal was carried over into HPSG.

It should be mentioned that the HP group was by no means alone in these concerns. The early 1980s was a period of rapid growth in computational linguistics (due at least in part to the rapid growth in the power and accessibility of computers). In the immediate vicinity of Stanford and HP Labs, there were at least two other groups working on developing natural language systems that were both computationally tractable and linguistically motivated. One such group was at the Xerox Palo Alto Research Center, where Ron Kaplan and Joan Bresnan (in collaboration with a number of other researchers, notably Martin Kay) were developing Lexical Functional Grammar; the other was at SRI International, where a large subset of SRI's artificial intelligence researchers (including Barbara Grosz, Jerry Hobbs, Bob Moore, Hans Uszkoreit, Fernando Pereira, and Stuart Shieber) worked on natural language. Thanks to the founding of the Center for the Study of Language and Information (CSLI) at Stanford in the early 1980s, there was a great deal of interaction among these three research groups. Although some aspects of the work being done at the three non-Stanford sites were proprietary, most of the research was basic enough that there was a fairly free flow of ideas among the three groups about building linguistically motivated natural language systems.

Other projects seeking to develop theories that combined computational tractability with linguistic motivation were also underway outside of the immediate vicinity of Stanford, notably at the Universities of Pennsylvania and Edinburgh. Aravind Joshi and his students were working on Tree Adjoining Grammars (Joshi et al. 1975; Joshi 1987), while Mark Steedman and others were developing Combinatory Categorial Grammar (Steedman 1987; 1990).

During the first few years of the HP NLP project, several Stanford students were hired as part-time help. One was Pollard, who was writing his doctoral dissertation under Sag's supervision. Ideas from his thesis work played a major role in the transition from GPSG to HPSG. Two other students who became very important to the project were Dan Flickinger, a doctoral student in linguistics, and Derek Proudian, who was working on an individually-designed undergraduate major when he first began at HP and later became a master's student in computer science. Both Flickinger and Proudian became full-time HP employees after finishing their degrees. Over the years, a number of other HP employees also worked on the project and made substantial contributions. They included Susan Brennan, Lewis Creary, Marilyn Friedman (now Walker), Dave Goddeau, Brett Kessler, Joachim Laubsch, and John Nerbonne. Brennan, Walker, Kessler, and Nerbonne all later went on to academic careers at major universities, doing research dealing with natural language processing.

The HP NLP project lasted until the early 1990s. By then, a fairly large and robust grammar of English had been implemented. The period around 1990 combined an economic recession with what has sometimes been termed an "AI winter" – that is, a period in which enthusiasm and hence funding for artificial intelligence research was at a particularly low ebb. Since NLP was considered a branch of AI, support for it waned. Hence, it was not surprising that the leadership of HP Labs decided to terminate the project. Flickinger and Proudian came to an agreement with HP that allowed them to use the NLP technology developed by the project to launch a new start-up company, which they named Eloquent Software. They were, however, unable to secure the capital necessary to turn the existing system into a product, so the company never got off the ground.

3 The emergence of HPSG

A few important features of GPSG that were later carried over into HPSG are worth mentioning here. First, GPSG borrowed from Montague the idea that each phrase structure rule was to be paired with a semantic rule providing a recipe for computing the meaning of the mother from the meanings of its daughters (Gazdar 1981: 156); this design feature was shared with contemporaneous forms of Categorial Grammar (CG) being studied by such linguists as Emmon Bach (Bach 1979; 1980) and David Dowty (Dowty 1982a,b). Second, the specific inventory of features employed in GPSG for making fine-grained categorial distinctions (such as case, agreement, verb inflectional form, and the like), was largely preserved, though the technical implementation of morphosyntactic features in HPSG was somewhat different. And third, the SLASH feature, which originated in Gazdar's (1981) derived categories (e.g. S/NP), and which was used to keep track of unbounded dependencies, was generalized in HPSG to allow for multiple unbounded dependencies (as in the notorious violins-and-sonatas example in (1) below). As will be discussed, this SLASH feature bears a superficial—and misleading—resemblance to the Categorial Grammar connectives written as '/' and '\'. On the other hand, a centrally important architectural feature of GPSG absent from HPSG (and from HG) was the device of metarules, higher-order rules used to generate the full set of context-free phrase structure rules (PSRs) from an initial inventory of basic PSRs. Among the metarules were ones used to introduce non-null SLASH values and propagate them upward through trees to a position where they were discharged by combination with a matching constituent called a filler (analogous to a *wh*-moved expression in TG).

A note is in order about the sometimes confusing use of the names *Head Grammar (HG)* and *HPSG*. Strictly speaking, HG was a specific subtype of generalized CFG developed in Pollard's dissertation work, but the term *HG* did not appear in academic linguistic publications with the exception of the Pollard & Sag (1983) WCCFL paper, which introduced the distinction between head features and binding features (the latter were incorporated into GPSG under the name *foot features*). In the summer of 1982, Pollard had started working part time on the HP NL project; and the term *HPSG* was first employed (by Pullum) in reference to an extensive reworking by Pollard and Paulson of the then-current HP GPSG implementation, incorporating some of the main features of Pollard's dissertation work in progress, carried out over the summer of 1983, while much of the HP NLP team (including Pullum and Sag) was away at the LSA Institute in Los Angeles. The implication of the name change was that whatever this new system was, it was no longer GPSG.

Once this first HPSG implementation was in place, the NLP work at HP was considered to be within the framework of HPSG, rather than GPSG. After Pollard completed his dissertation, he continued to refer to *HG* in invited talks as late as autumn 1984; but his talk at the (December 1984) LSA Binding Theory Symposium used *HPSG* instead, and after that, the term *HG* was supplanted by *HPSG* (except in publications by non-linguists about formal language theory). One additional complication is that until the Gazdar, Klein, Pullum & Sag (1985) volume appeared, GPSG and HPSG were developing side by side, with considerable interaction. Pollard, together with Flickinger, Wasow, Nerbonne, and others, did HPSG; Gazdar and Klein did GPSG; and Sag and Pullum worked both sides of the street.

HPSG papers, about both theory and implementation, began to appear in 1985, starting with Pollard's WCCFL paper *Phrase structure grammar without metarules* (Pollard 1985), and his paper at the Categorial Grammar conference in Tucson (Pollard 1988), comparing and contrasting HPSG with then-current versions of Categorial Grammar due to Bach, Dowty, and Steedman. These were followed by a trio of ACL papers documenting the current state of the HPSG implementation at HP Labs: Creary & Pollard (1985), Flickinger, Pollard & Wasow (1985), and Proudian & Pollard (1985). Of those three, the most significant in terms of its influence on the subsequent development of the HPSG framework was the second, which showed how the lexicon could be (and in fact was) organized using multiple-inheritance knowledge representation; Flickinger's Stanford dissertation (Flickinger 1987) was an in-depth exploration of that idea.

4 Early HPSG

Setting aside implementation details, early HPSG can be characterized by the following architectural features:

Elimination of metarules Although metarules were a central feature of GPSG, they were also problematic: Uszkoreit & Peters (1982) had shown that if metarules were allowed to apply to their own outputs, then the resulting grammars were no longer guaranteed to generate CFLs; indeed, such grammars could generate all recursively enumerable languages. And so, in GPSG, the closure of a set of base phrase structure rules (PSRs) under a set of metarules was defined in such a way that no metarule could apply to a PSR whose own derivation involved an application of that metarule. This definition was intended to ensure that the closure of a finite set of PSRs remained finite, and therefore still constituted a CFG.

So, for example, the metarule STM1 was used in GPSG to convert a PSR into another PSR one of whose daughters is [+NULL] (informally speaking, a 'trace'), and feature cooccurrence restrictions (FCRs) guaranteed that such daughters would bear a SLASH value, and that this SLASH value would also appear on the mother. Unfortunately, the finite closure definition described above does not preclude the possibility of derived PSRs whose mother carries multiple, in fact unboundedly many SLASH values (e.g. NP/NP, (NP/NP)/NP, etc.). And this in turn leads to an infinite set of PSRs, outside the realm of CF-ness (see Ristad (1986)). Of course, one could rein in this excess power by imposing another FCR that disallows categories of the form (X/Y)/Z; but then there is no way to analyze sentences containing a constituent with two undischarged unbounded dependencies, such as the VP complement of *easy* in the following example:

(1) Violins this finely crafted, even the most challenging sonatas are easy to [play _ on _].

GPSG avoided this problem by not analyzing such examples. In HPSG (Pollard 1985), by contrast, such examples were analyzed straightforwardly by replacing GPSG's category-valued SLASH feature with one whose values were lists (or sets) of categories. This approach still gave rise to an infinite set of rules, but since maintaining context-freeness was no longer at stake, this was not seen as problematic. The infinitude of rules in HPSG arose not through a violation of finite closure (since there were no longer any metarules at all), but because each of the handful of schematic PSRs (see below) could be directly instantiated in an infinite number of ways, given that the presence of list-valued features gave rise to an infinite set of categories.

Lexical rules GPSG, generalizing a suggestion of Flickinger (1983), constrained metarules to apply only to PSRs that introduced a lexical head. Pollard (1985) took this idea a step further, noting that many proposed metarules could be reformulated as lexical rules that (among other effects) operated on the subcategorization frames (encoded by the SUBCAT feature discussed below) of lexical entries. The idea of capturing some linguistic generalizations by means of rules internal to the lexicon had been explored by generative grammarians since Jackendoff (1975); and lexical rules of essentially the kind Pollard proposed were employed by Bach (1983), Dowty (1978), and others working in Categorial Grammar. Examples of constructions handled by metarules in GPSG but in HPSG by lexical rules included sentential extraposition, subject extraction, and passive. Flickinger, Pollard & Wasow (1985) argued for an architecture for the lexicon that combined lexical rules with multiple inheritance using a frame-based knowledge representation system, on the basis of both overall grammar simplicity and efficient, easily modifiable implementation.

CG-like treatment of subcategorization In GPSG, subcategorization was treated by an integer-valued feature called SUBCAT that in effect indexed each lexical item with the rule that introduced and provided its subcategorization frame; e.g. *weep* was listed in the lexicon with SUBCAT value 1 while *devour* was listed with SUBCAT value 2, and then PSRs of roughly the form in (2)

(2) VP → V[SUBCAT 1]
    VP → V[SUBCAT 2] NP

guaranteed that lexical heads would have the right kinds of complements. In HPSG, by contrast, the SUBCAT feature directly characterized the grammatical arguments selected by a head (not just the complements, but the subject too) as a list of categories, so that e.g. *weep* was listed as V[SUBCAT <NP>] but *devour* as V[SUBCAT <NP,NP>] (where the first occurrence of NP refers to the object and the second to the subject). This treatment of argument selection was inspired by Categorial Grammar, where the same verbs would have been categorized as NP\S and (NP\S)/NP respectively;[2] the main differences are that (i) the CG treatment also encodes the directionality of the argument relative to the head, and (ii) in HPSG, all the arguments appear on one list, while in CG they are ‘picked up’ one at a time, with as many connectives (/ or \) as there are arguments. In particular, as in the CG of Dowty (1982c), the subject was defined as the last argument, except that in HPSG, ‘last’ now referred to the rightmost position on the SUBCAT list, not to the most deeply embedded connective. In HPSG, this ordering of the categories on the SUBCAT list was related not just to CG, but also to the traditional grammatical notion of obliqueness, and also to the accessibility hierarchy of Keenan & Comrie (1977).

Schematic rules Unlike CFG, but like CG, HPSG had only a handful of schematic rules. For example, in Pollard (1985), a substantial chunk of English ‘local’ grammar (i.e. leaving aside unbounded dependencies) was handled by three rules: (i) a rule (used for subject-auxiliary inversion) that forms a sentence from an inverted (+INV) lexical head and all its arguments; (ii) a rule that forms a phrase from a head with SUBCAT list of length > 1 together with all its non-subject arguments; and (iii) a rule that forms a sentence from a head with a SUBCAT value of length one together with its single (subject) argument.

List- (or set-) valued SLASH feature The list-valued SLASH was introduced in Pollard (1985) to handle multiple unbounded dependencies, instead of the GPSG category-valued SLASH (which in turn originated as the *derived categories* of Gazdar (1981), e.g. S/NP). In spite of the notational similarity, though, the PSG SLASH is not an analog of the CG slashes / and \(though HPSG’s SUBCAT is, as explained above). In fact, HPSG’s SLASH has no analog in the kinds of CGs being developed by Montague semanticists such as Bach (1979; 1980) and Dowty (1982a) in the late 1970s and early 1980s, which followed the CGs of Bar-Hillel (1954) in having only rules for eliminating (or canceling) slashes as in (3):

(3) $\frac{\mathrm{A}\ \ \mathrm{A}\backslash\mathrm{B}}{\mathrm{B}} \qquad \frac{\mathrm{B/A}\ \ \mathrm{A}}{\mathrm{B}}$

[2] We adhere to the Lambek convention for functor categories, so that expressions seeking to combine with an A on the left to form a B are written ‘A\B’ (not ‘B\A’).

To find an analog to HPSG's SLASH in CG, we have to turn to the kinds of CGs invented by Lambek (1958), which unfortunately were not yet well-known to linguists (though that would soon change starting with Lambek's appearance at the 1985 Categorial Grammar conference in Tucson). What sets apart grammars of this kind (and their elaborations by Moortgat (1989), Oehrle et al. (1988), Morrill (1994), and many others), is the existence of rules for hypothetical proof (not given here), which allow a hypothesized category occurrence introduced into a tree (thought of as a proof) to be discharged.

In the Gentzen style of natural deduction (see Pollard (2013)), hypothesized categories are written to the left of the symbol |- (turnstile), so that the two slash elimination rules above take the following form (where Γ and Δ are lists of categories, and comma represents list concatenation as in (4):

(4) $$\frac{\Gamma \vdash \mathrm{A} \quad \Delta \vdash \mathrm{A}\backslash\mathrm{B}}{\Gamma,\Delta \vdash \mathrm{B}} \qquad \frac{\Gamma \vdash \mathrm{B}/\mathrm{A} \quad \Delta \vdash \mathrm{A}}{\Gamma,\Delta \vdash \mathrm{B}}$$

These rules propagate hypotheses (analogous to linguists' traces) downward through the proof tree (downward because logicians' trees are upside down with the conclusion ('root') at the bottom). In HPSG notation, these same rules can be written as one rule (since SUBCAT is nondirectional) in (5):

(5) $$\frac{\mathrm{B[SUBCAT} <\ldots,\mathrm{A}>,\mathrm{SLASH}\ \Gamma] \quad \mathrm{A[SLASH}\ \Delta]}{\mathrm{B[SUBCAT} <\ldots>][\mathrm{SLASH}\ \Gamma,\Delta]}$$

This in turn is a special case of an HPSG principle first known as the Binding Inheritance Principle (BIP) and later as the Nonlocal Feature Principle (binding features included SLASH as well as the features QUE and REL used for tracking undischarged interrogative and relative pronouns). The original statement of the BIP (Pollard 1986) treated SLASH as set- rather than list-valued):

> The value of a binding feature on the mother is the union of the values of that feature on the daughters.

For example, the doubly-gapped VP in the violins-and-sonatas example in (1) is analyzed in HPSG roughly as is shown in Figure 1 and essentially the same way in Lambek-style CG:

(6) $$\frac{\dfrac{\overset{\text{play}}{\vdash ((\mathrm{NP}\backslash\mathrm{S})/\mathrm{PP})/\mathrm{NP}} \quad \overset{t}{\mathrm{NP} \vdash \mathrm{NP}}}{\mathrm{NP} \vdash (\mathrm{NP}\backslash\mathrm{S})/\mathrm{PP}} \qquad \dfrac{\overset{on}{\vdash \mathrm{PP}/\mathrm{NP}} \quad \overset{t}{\mathrm{NP} \vdash \mathrm{NP}}}{\mathrm{NP} \vdash \mathrm{PP}}}{\mathrm{NP},\mathrm{NP} \vdash \mathrm{NP}\backslash\mathrm{S})}$$

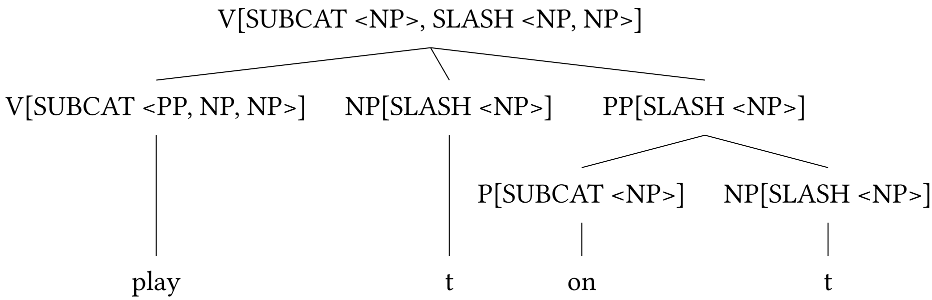

Figure 1: *play on* as part of *Violins this finely crafted, even the most challenging sonatas are easy to play on.*

Aside from the binary branching of the Lambek analysis, the main difference is that HPSG traces of the form A[SLASH <A>] correspond to Lambek axioms of the form A |- A, which is the standard mechanism for introducing hypotheses in Gentzen-style natural deduction.

An overview and elaboration of early HPSG is provided by the two books Pollard & Sag (1987) and Pollard & Sag (1994). Confusingly, the former is called *Information-Based Syntax and Semantics, Volume 1: Fundamentals*, and the second simply *Head-Driven Phrase Structure Grammar* (not *Information-Based Syntax and Semantics, Volume 2*). The reason for the title change had to do with a change in the underlying mathematical theory of feature structures. In the first book, following work in theoretical computer science by Rounds & Kasper (1986) and Moshier & Rounds (1987), feature structures were treated as data structures that supplied partial information about the linguistic objects being theorized about; this perspective in turn was based on Scott's (1982) mathematical theory of computation in terms of what he called information systems. Subsequently, Paul King persuaded Pollard and Sag that it was more straightforward to distinguish between feature structures, thought of as formal models of the linguistic objects, and feature descriptions or formulas of feature logic, which provided partial information about them, as described in his Manchester dissertation (King 1989). Although the formal issues involved in distinguishing between the two approaches are of interest in their own right, they seem not to have had a lasting effect on how theoretical linguists used HPSG, or on how computational linguists implemented it. As for subject matter, Pollard & Sag (1987) was limited to the most basic notions, including syntactic features and categories (including the distinction between head features and binding features); subcategorization and the distinction between arguments and adjuncts (the latter of which necessitated one more rule schema beyond the three proposed by Pollard (1985)); basic principles of grammar (especially the Head Feature Principle and the Subcategorization Principle); the obliqueness order and constituent ordering; and the organization of the lexicon by means of a multiple inheritance hierarchy and lexical rules. Pollard & Sag (1994) used HPSG to analyze a wide range of phenomena that had figured prominently in the syntactic literature of the 1960s–1980s, including agreement, expletive pronoun constructions, raising, control, filler-gap constructions (including island constraints and parasitic gaps); so-called binding theory (the distribution of reflexive pronouns, nonreflexive pronouns, and non-pronominal NPs), and scope of quantificational NPs.

5 Theoretical Developments

Three decades of vigorous work since Pollard & Sag (1987) developing the theoretical framework of HPSG receive detailed discussion throughout the present volume, but we highlight here two significant stages in that development. The first is in Chapter 9 of Pollard & Sag (1994), where a pair of major revisions to the framework presented in the first eight chapters are adopted, changing the analysis of valence and of unbounded dependencies. Following **Borsley87**; Borsley (1988; 1989; 1990), Pollard and Sag move to distinguish subjects from complements, and further to distinguish subjects from specifiers, thus replacing the single SUBCAT attribute with SUBJ, SPR, and COMPS. This formal distinction between subjects and complements enabled an improved analysis of unbounded dependencies, eliminating traces altogether by introducing three lexical rules for the extraction of subjects, complements, and adjuncts respectively. It is this revised analysis of valence constraints that came to be viewed as part of the standard HPSG framework, though issues of valence representation crosslinguistically remain a matter of robust debate.

The second notable stage of development was the introduction of a type hierarchy of *constructions* as descriptions of phrasal feature structures, employed first by Sag (1997) in a richly detailed analysis of a wide variety of relative clause phenomena in English. This extension from the lexicon of the use of descriptions of typed feature structures organized in hierarchies to syntactic rules preserved the ability to express general principles holding for rule schemata while also enabling expression of idiosyncratic properties of phrases. In Borsley & Abeillé (2018), Chapter 1 of this volume, the version of the framework with this extended use of types is termed *Construction-based HPSG*, including further elaboration by Ginzburg & Sag (2000) to a comprehensive analysis of interrogatives in English.

6 The LinGO Project

In the early 1990s, a consortium of research centers in Germany secured funding from the German government for a large project in spoken language machine translation, called Verb*mobil* (Wahlster 2000), which aimed to combine a variety of methods and frameworks in a single implemented state-of-the-art demonstrator system. Grammars of German and English were to be implemented in HPSG, to be used both for parsing and for generation in the translation of human-human dialogues, with a German grammar initially implemented by Pollard and Tibor Kiss at IBM in Heidelberg, later replaced by one developed at the German AI Research Center (DFKI), coordinator for the Verb*mobil* project. The DFKI contracted in 1993 with Sag at CSLI to design and implement the English grammar, with Flickinger brought over from HP Labs to help lead the effort, forming a new research group at CSLI initially called ERGO (for English Resource Grammar Online), later generalized to the name LinGO (Linguistic Grammars Online). Early LinGO members included Wasow and linguistics graduate student Rob Malouf, who authored the initial implementation of the English Resource Grammar (ERG), along with two other linguistics graduate students: Kathryn Campbell-Kibler, who contributed to the development of the lexicon, and Tony Davis, who helped in refining the lexical type hierarchy.

During the first of the two four-year phases of the Verb*mobil* project, the focus was on designing and implementing core syntactic and semantic analyses, initially using the DISCO/PAGE platform (Uszkoreit et al. 1994) developed at the DFKI, and largely informed by the framework presented in Pollard & Sag (1994). However, a more computationally useful semantic formalism emerged, called Minimal Recursion Semantics (MRS: Copestake, Flickinger, Pollard & Sag (2005)), which Ann Copestake, formerly of the European ACQUILEX project, helped to design. Copestake also expanded the LKB system (Copestake 2002) which had been used in ACQUILEX, to serve as the grammar development environment for the LinGO project, including both a parser and a generator for typed feature structure grammars.

The second four years of the Verb*mobil* project emphasized development of the generation capabilities of the ERG, along with steady expansion of linguistic coverage, and elaboration of the MRS framework. LinGO contributors in this phase, in addition to Sag, Wasow, Flickinger, Malouf, and Copestake, included Stanford Linguistics graduate students Emily Bender and Susanne Riehemann, along with a regular visitor and steady contributor from the DFKI, Stephan Oepen. Verb*mobil* had meanwhile added Japanese alongside German (Müller & Kasper 2000) and

English (Flickinger, Copestake & Sag 2000) for more translation pairs, giving rise to another relatively broad-coverage HPSG grammar, JaCY, authored by Melanie Siegel at the DFKI (Siegel 2000). Work continued at the DFKI, of course, on the German HPSG grammar, written by Stefan Müller, adapted from his earlier Babel grammars (Müller 1999), and with semantics contributed by Walter Kasper.

Before the end of Verbmobil funding in 2000, the LinGO project had already begun to diversify into other application and research areas using the ERG, including over the next several years work on augmented/adaptive communication, multiword expressions, and hybrid processing with statistical methods, variously funded by the National Science Foundation, the Scottish government, and industrial partners including IBM and NTT. At the turn of the millenium, Flickinger joined the software start-up boom, co-founding YY Software funded through substantial venture capital to use the ERG for automated response to customer emails for e-commerce companies. YY produced the first commercially viable software system using an HPSG implementation, processing email content in English with the ERG and the PET parser (Callmeier 2000) which had been developed by Ulrich Callmeier at the DFKI, as well as in Japanese with JaCY, further developed by Siegel and by Bender. While technically capable, the product was not commercially successful enough to enable YY to survive the bursting of the dot-com bubble, and it closed down in 2003. Flickinger returned to the LinGO project with a considerably more robust ERG, and soon picked up the translation application thread again, this time using the ERG for generation in the LOGON Norwegian-English machine translation project based in Oslo.

7 Research and Teaching Networks

The first international conference on HPSG was held in 1993 in Columbus, Ohio, in conjunction with the Linguistic Society of America's Summer Institute. The conference has been convened every year since then, with locations in Europe, Asia, and North America. Two of these annual meetings have been held jointly with the annual Lexical Functional Grammar conference, in 2000 in Berkeley and in 2016 in Warsaw. Proceedings of these conferences since 2000 are available on-line from CSLI Publications.[3] Since 2003, HPSG researchers in Europe have frequently held a regional workshop in Bremen, Berlin, Frankfurt, or Paris, to foster informal discussion of current work in HPSG. These follow in the footsteps of European HPSG workshops starting with one on German grammar, held in

[3] http://csli-publications.stanford.edu/HPSG/, 2018-08-21.

Saarbrücken in 1991, and including others in Edinburgh and Copenhagen in 1994, and in Tübingen in 1995.

In 1994, the HPSG mailing list was initiated,[4] and from 1996 to 1998, the electronic newsletter, the HPSG Gazette,[5] was distributed through the list, with its function then taken over by the HPSG mailing list.

Courses introducing HPSG to students became part of the curriculum during the late 1980s and early 1990s at universities in Osaka, Paris, Saarbrücken, Seoul, and Tübingen, along with Stanford and OSU. Additional courses came to be offered in Bochum, Bremen, Carnegie-Mellon, Göttingen, Heidelberg, Jena, and Potsdam. Summer courses and workshops on HPSG have also been offered since the early 1990s at the LSA Summer Institute in the U.S., including a course by Sag and Pollard on binding and control in 1991 in Santa Cruz, and at the European Summer School in Logic, Language and Information (ESSLLI), including a course by Pollard in Saarbrücken in 1991 on HPSG, a workshop in Colchester in 1992 on HPSG, a workshop in Prague in 1996 on Romance (along with two HPSG-related student papers at the first-ever ESSLLI student session), and courses in 1998 in Saarbrücken on Germanic syntax, grammar engineering, and unification-based formalisms, in 2001 on HPSG syntax, in 2003 on linearization grammars, and more since. Also in 2001, a Scandinavian summer school on constraint-based grammar was held in Trondheim.

Several HPSG textbooks have been published, including at least Borsley (1991; 1996), Sag & Wasow (1999), Sag, Wasow & Bender (2003), Müller (2007a; 2013a)), Müller (2016), Kim (2016), and Levine (2017).

8 Implementations and Applications of HPSG

The first implementation of a grammar in the HPSG framework emerged in the Hewlett-Packard Labs natural language project, for English, with a lexical type hierarchy (Flickinger, Pollard & Wasow 1985), a set of grammar rules that provided coverage of core syntactic phenomena including unbounded dependencies and coordination, and a semantic component called Natural Language Logic (Laubsch & Nerbonne 1991). The corresponding parser for this grammar was implemented in Lisp (Proudian & Pollard 1985), as part of a system called HP-NL (Nerbonne & Proudian 1987) which provided a natural language interface for querying relational databases. The grammar and parser were shelved when HP Labs terminated their natural language project in 1991, leading Sag and Flickinger to begin

[4] Its archives can be found at https://hpsg.hu-berlin.de/HPSG/MailingList.

[5] http://www.sfs.uni-tuebingen.de/~gazette, 2018-08-21.

the LinGO project and development of the English Resource Grammar at Stanford.

By this time, grammars in HPSG were being implemented in university research groups for several other languages, using a variety of parsers and grammar engineering platforms for processing typed feature structure grammars. Early platforms included the DFKI's DISCO system (Uszkoreit et al. 1994) with a parser and graphical development tools, which evolved to the PAGE system; the ALE system (Franz 1990; Carpenter & Penn 1996), which evolved in Tübingen to TRALE (Meurers et al. 2002; Penn 2004); and Ann Copestake's LKB (Copestake 2002) which grew out of the ACQUILEX project. Other early systems included ALEP within the Eurotra project (Simpkins & Groenendijk 1994), ConTroll at Tübingen (Götz & Meurers 1997), CUF at IMS in Stuttgart (Dörre & Dorna 1993), CL-ONE at Edinburgh (Manandhar 1994), TFS also at IMS (Emele 1994), ProFIT at the University of Saarland (Erbach 1995), Babel at Humboldt University in Berlin (Müller 1996), and HDrug at Groningen (van Noord & Bouma 1997).

Relatively early broad-coverage grammar implementations in HPSG, in addition to the English Resource Grammar at Stanford (Flickinger 2000), included one for German at the DFKI (Müller & Kasper 2000) and one for Japanese (Jacy: Siegel (2000)), all used in the Verbmobil machine translation project; a separate German grammar (Müller 1996; 1999); a Dutch grammar in Groningen (Bouma, van Noord & Malouf 2001); and a separate Japanese grammar in Tokyo (Miyao et al. 2005). Moderately large HPSG grammars were also developed during this period for Korean (Kim & Yang 2003) and Polish (Mykowiecka, Marciniak, Przepiórkowski & Kupść 2003).

In 1999, research groups at the DFKI, Stanford, and Tokyo set up a consortium called DELPH-IN (Initiative for Deep Linguistic Processing in HPSG), to foster broader development of both grammars and platform components, described in Oepen, Flickinger, Tsujii & Uszkoreit (2002). Over the next two decades, substantial DELPH-IN grammars were developed for Norwegian, Portuguese, and Spanish, along with moderate-coverage grammars for Bulgarian, Greek, Hausa, Hebrew, Indonesian, Mandarin Chinese, Thai, and Wambaya, all described at http://delph-in.net. Several of these grammars are based on the Grammar Matrix (Bender, Flickinger & Oepen 2002), a starter kit generalized from the ERG and Jacy for rapid prototyping of HPSG grammars, along with a much larger set of coursework grammars.[6]

Broad-coverage grammars developed in the TRALE system (Meurers et al. 2002; Penn 2004) include German (Müller 2007a), Danish (Müller & Ørsnes 2013),

[6] http://moin.delph-in.net/MatrixTop, 2018-08-21.

and Persian (Müller 2010). Other TRALE grammars include Mandarin Chinese (Müller & Lipenkova 2013), Georgian (Abzianidze 2011), Maltese (Müller 2009), Spanish (Machicao y Priemer 2015), and Yiddish (Müller & Ørsnes 2011). Development of grammars in TRALE is supported by the Grammix system (Müller 2007b); Müller (2015) provides a summary of this family of grammar implementations.

These grammars and systems have been used in a wide variety of applications, primarily as vehicles for research in computational linguistics, but also for some commercial software products. Research applications already mentioned include database query (HP Labs) and machine translation (Verb*mobil* and LOGON), with additional applications developed for use in anthology search (Schäfer, Kiefer, Spurk, Steffen & Wang 2011), grammar tutoring in Norwegian (Hellan, Bruland, Aamot & Sandøy 2013), ontology acquisition (Herbelot & Copestake 2006), virtual robot control (Packard 2014), visual question answering (Kuhnle & Copestake 2017), and logic instruction (Flickinger 2017), among many others. Commercial applications include e-commerce customer email response (for YY Software), and grammar correction in education (for Redbird Advanced Learning, now part of McGraw-Hill Education: Suppes, Flickinger, Macken, Cook & Liang (2012)).

For most practical applications, some approximate solution to the challenge of parse selection (disambiguation) must be provided, so several of the DELPH-IN grammars, including the ERG, follow the approach of Oepen, Flickinger, Toutanova & Manning (2004), which uses a manually-annotated treebank of sentences parsed by a grammar to train a statistical model which is applied at run-time to identify the most likely analysis for each parsed sentence. These treebanks can also serve as repositories of the analyses intended by the grammarian for the sentences of a corpus, and some resources, notably the Alpino Treebank (Bouma, van Noord & Malouf 2001), include analyses which the grammar may not yet be able to produce automatically.

9 Prospects

As we noted early in this chapter, HPSG's origins are rooted in the desire simultaneously to address the theoretical concerns of linguists and the practical issues involved in building a useful natural language processing system. In the decades since the birth of HPSG, the mainstream of work in both theoretical linguistics and NLP developed in ways that could not have been anticipated at the time. NLP is now dominated by statistical methods, with almost all practical applications making use of machine learning technologies. It is hard to see any influence of research by linguists in most NLP systems. Mainstream grammatical theory, on the other hand, is now dominated by the Minimalist Program (MP), which is too vaguely formulated for a rigorous comparison with HPSG.[7] Concern with computational implementation plays virtually no role in MP research; see Müller (2016) for a discussion.

It might seem, therefore, that HPSG is further from the mainstream of both fields than it was at its inception, raising questions about how realistic the objectives of HPSG are. We believe, however, that there are grounds for optimism.

With regard to implementations, there is no incompatibility between the use of HPSG and the machine learning methods of mainstream NLP. Indeed, as noted above, HPSG-based systems that have been put to practical use have necessarily included components induced via statistical methods from annotated corpora. Without such components, the systems cannot deal with the full variety of forms encountered in usage data. On the other hand, existing NLP systems that rely solely on machine learning from corpora do not exhibit anything that can reasonably be called understanding of natural language. Current technologies for machine translation, automatic summarization, and various other linguistic tasks fall far short of what humans do on these tasks, and are useful primarily as tools to speed up the tasks for the humans carrying them out. Many NLP researchers are beginning to recognize that developing software that can plausibly be said to understand language will require representations of linguistic structure and meaning like those that are the stock in trade of linguists.

Evidence for a renewed interest in linguistics among NLP researchers is the fact that major technology companies with natural language groups have recently begun (or in some cases, resumed) hiring linguists, and increasing numbers of new linguistics PhDs have taken jobs in the software industry.

In the domain of theoretical linguistics, it is arguable that the distance between HPSG and the mainstream of grammatical research (that is, MP) has narrowed, given that both crucially incorporate ideas from Categorial Grammar (see Retoré & Stabler (2004), Berwick & Epstein (1995), and Müller (2013b) for comparisons between MP and CG). Rather than trying to make that argument, however, we will point to connections that HPSG has made with other work in theoretical linguistics. Perhaps the most obvious of these is the work of Peter Culicover and Ray Jackendoff on what they call *Simpler Syntax*. Their influential 2005 book with that title (Culicover & Jackendoff 2005) argues for a theory of grammar that

[7] Most work in MP is presented without precise definitions of the technical apparatus, but Edward Stabler and his collaborators have written a number of papers aimed at formalizing MP.See in particular Collins & Stabler (2016).

differs little in its architecture and motivations from HPSG.

More interesting are the connections that have been forged between research in HPSG and work in Construction Grammar (CxG). Fillmore (1988: 36) characterizes the notion of *construction* as "any syntactic pattern which is assigned one or more conventional functions in a language, together with whatever is linguistically conventionalized about its contribution to the meaning or use of structures containing it." Among the examples that construction grammarians have described at length are *the Xer, the Yer* (as in *the older I get, the longer I sleep*), *X let alone Y* (as in *I barely got up in time to eat lunch, let alone cook breakfast*), and *What's X doing Y?* (as in *What's this scratch doing in the table?*). As noted above and in Müller (2018), Chapter 37 of this volume, HPSG has incorporated the notion of construction since at least the late 1990s.

Nevertheless, work that labels itself CxG tends to look very different from HPSG. This is in part because of the difference in their origins: many proponents of CxG come from the tradition of Cognitive Grammar or typological studies, whereas HPSG's roots are in computational concerns. Hence, most of the CxG literature is not precise enough to allow a straightforward comparison with HPSG, though the variants called Embodied Construction Grammar and Fluid Construction Grammar have more in common with HPSG; see Müller (2017) for a comparison. In the last years of his life, Ivan Sag sought to unify CxG and HPSG through collaboration with construction grammarians from the University of California at Berkeley, particularly Charles Fillmore, Paul Kay, and Laura Michaelis. They developed a theory called *Sign-Based Construction Grammar* (SBCG), which would combine the insights of CxG with the explicitness of HPSG. Sag (2012: 70) wrote, "To readers steeped in HPSG theory, SBCG will no doubt seem like a minor variant of constructional HPSG." Indeed, despite the name change, the main feature of SBCG that differs from HPSG is that it posits an inheritance hierarchy of constructs, which includes feature structure descriptions for such partially lexicalized multi-word expressions as *Ved X's way PP*, instantiated in such VPs as *ad-libbed his way through a largely secret meeting*. While this is a non-trivial extension to HPSG, there is no fundamental change to the technical machinery. In fact, it has been a part of the LinGO implementation for many years.

That said, there is one important theoretical issue that divides HPSG and SBCG from much other work in CxG. That issue is locality. To constrain the formal power of the theory, and to facilitate computational tractability, SBCG adopts what Sag (2012: 150) calls "Constructional Localism" and describes as follows: "Constructions license mother-daughter configurations without reference to embedding or embedded contexts." That is. like phrase structure rules, constructions must be characterized in terms of a mother node and its immediate daughters. At first glance, this seems to rule out analyses of many of the examples of constructions provided in the CxG literature. But Sag (2012: 150) goes on to say, "Constructional Localism does not preclude an account of nonlocal dependencies in grammar, it simply requires that all such dependencies be locally encoded in signs in such a way that information about a distal element can be accessed locally at a higher level of structure."

Fillmore (1988: 35) wrote:

> Construction grammars differ from phrase-structure grammars which use *complex symbols* and allow the *transmission of information* between lower and higher structural units, in that we allow the direct representation of the required properties of subordinate constituents. (Should it turn out that there are completely general principles for predicting the kinds of information that get transmitted upwards or downwards, this may not be a real difference.) (Fillmore 1988: 35)

SBCG is committed to the position alluded to in the parenthetical sentence in this quote, namely, that general principles of information transmission within sentences makes it possible to insist on Constructional Localism. See Müller (2018), Chapter 37 of this volume for a much more detailed discussion.

Finally, another point of convergence between work in HPSG and other work in both theoretical linguistics and NLP is the increasing importance of corpus data. In the early years of the HP NLP project, the methodology was the same as that employed in almost all work in theoretical syntax and semantics: the grammar was based entirely on examples invented by the researchers. At one point during the decade of the HP NLP project, Flickinger, Pullum, and Wasow compiled a list of sentences intended to exemplify many of the sentence types that they hoped the system would eventually be able to analyze. That list, 1328 sentences long, continues to be useful as a test suite for the LinGO system and various other NLP groups. But it does not come close to covering the variety of sentence forms that are found in corpora of speech and various written genres. As the goals of the HPSG implementations have broadened from database query to dealing with "language in the wild", the use of corpora to test such systems and motivate extensions to them has increased. This parallels a development in other areas of linguistics, which have also increasingly made use of large on-line corpora as sources of data and tests of their theories. This is a trend that we expect will continue.

In short, there are signs of convergence between work on HPSG and work in other areas, and it seems plausible to think that the market for HPSG research will grow in the future.

References

Abzianidze, Lasha. 2011. *An HPSG-based formal grammar of a core fragment of Georgian implemented in TRALE*. Charles University in Prague MA thesis.

Bach, Emmon. 1979. Control in Montague Grammar. *Linguistic Inquiry* 4(10). 515–531.

Bach, Emmon. 1980. In defense of passive. *Linguistics and Philosophy* (3). 297–342.

Bach, Emmon. 1983. On the relationship between Word-Grammar and Phrase-Grammar. *Natural Language and Linguistic Theory* 1(1). 65–89.

Bar-Hillel, Yehoshua. 1954. Logical syntax and semantics. *Language* 30(2). 230–237. DOI:10.2307/410265

Bender, Emily M., Daniel P. Flickinger & Stephan Oepen. 2002. The Grammar Matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In John Carroll, Nelleke Oostdijk & Richard Sutcliffe (eds.), *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, 8–14. Taipei, Taiwan.

Berwick, Robert C. & Samuel David Epstein. 1995. On the convergence of 'Minimalist' Syntax and Categorial Grammar. In Anton Nijholt, Giuseppe Scollo & Rene Steetskamp (eds.), *Algebraic methods in language processing*, 143–148. Enschede: University of Twente. http://eprints.eemcs.utwente.nl/9555/01/twlt10.pdf, accessed 2018-9-25.

Bloomfield, Leonard. 1933. *Language*. London: George Allen & Unwin.

Borsley, Robert D. 1988. Subjects, complements, and specifiers in HPSG. unpublished paper, University of Wales, Bangor.

Borsley, Robert D. 1989. Phrase-Structure Grammar and the Barriers conception of clause structure. *Linguistics* 27(5). 843–863.

Borsley, Robert D. 1990. Welsh passives. In Martin J. Ball, James Fife, Erich Poppe & Jenny Rowland (eds.), *Celtic linguistics: readings in the bretonic languages, a festschrift for t. arwynwatkins* (Current Issues in Linguistic Theory 68), 89–107. Amsterdam: John Benjamins Publishing Co.

Borsley, Robert D. 1991. *Syntactic theory: A unified approach*. London: Edward Arnold.

Borsley, Robert D. 1996. *Modern Phrase Structure Grammar* (Blackwell Textbooks in Linguistics 11). Oxford: Blackwell Publishers Ltd.

Bouma, Gosse, Gertjan van Noord & Robert Malouf. 2001. Alpino: wide-coverage computational analysis of Dutch. In Walter Daelemans, Khalil Sima'an, Jorn Veenstra & Jakub Zavrel (eds.), *Computational linguistics in the Netherlands 2000: Selected papers from the Eleventh CLIN Meeting* (Language and Computers 37). Amsterdam/New York, NY: Rodopi.

Bresnan, Joan. 1978. A realistic Transformational Grammar. In Morris Halle, Joan Bresnan & George A. Miller (eds.), *Linguistic theory and psychological reality*, 1–59. Cambridge, MA: MIT Press.

Callmeier, Ulrich. 2000. PET: A platform for experimentation with efficient HPSG processing techniques. *Natural Language Engineering* 6(1). 99–108. Special Issue on Efficient Processing with HPSG: Methods, Systems, Evaluation.

Carpenter, Bob & Gerald Penn. 1996. Efficient parsing of compiled typed attribute value logic grammars. In Harry Bunt & Masaru Tomita (eds.), *Recent advances in parsing technology* (Text, Speech and Language Technology 1), 145–168. Dordrecht: Kluwer Academic Publishers.

Chomsky, Noam. 1957. *Syntactic structures* (Janua Linguarum / Series Minor 4). The Hague/Paris: Mouton.

Chomsky, Noam. 1965. *Aspects of the theory of syntax*. Cambridge, MA: MIT Press.

Chomsky, Noam. 1973. Conditions on transformations. In Stephen R. Anderson & Paul Kiparsky (eds.), *A festschrift for Morris Halle*, 232–286. New York: Holt, Rinehart & Winston.

Chomsky, Noam & Howard Lasnik. 1977. Filters and control. *Linguistic Inquiry* 8. 425–504.

Collins, Chris & Edward Stabler. 2016. A Formalization of Minimalist Syntax. *Syntax* 19(1). 43–78.

Copestake, Ann. 2002. *Implementing typed feature structure grammars*. Stanford, CA: CSLI.

Copestake, Ann, Daniel P. Flickinger, Carl J. Pollard & Ivan A. Sag. 2005. Minimal Recursion Semantics: An introduction. *Research on Language and Computation* 3(2–3). 281–332. DOI:10.1007/s11168-006-6327-9

Creary, Lewis G. & Carl J. Pollard. 1985. A computational semantics for natural language. In *Proceedings of the 23rd annual meeting of the association for computational linguistics*, 172–179. Chicago, Illinois, USA: Association for Computational Linguistics. http://www.aclweb.org/anthology/P85-1022. DOI:10.3115/981210.981232

Culicover, Peter W. & Ray S. Jackendoff. 2005. *Simpler Syntax*. Oxford: Oxford University Press.

Culy, Christopher. 1985. The complexity of the vocabulary of Bambara. *Linguistics and Philosophy* 8. 345–351.

Dörre, Jochen & Michael Dorna. 1993. *CUF: A formalism for linguistic knowledge representation*. DYANA 2 deliverable R.1.2A. Stuttgart, Germany: IMS.

Dowty, David R. 1978. Governed transformations as lexical rules in a Montague Grammar. *Linguistic Inquiry* 9(3). 393–426.

Dowty, David R. 1979. *Word meaning and Montague Grammar* (Synthese Language Library 7). Dordrecht: D. Reidel Publishing Company.

Dowty, David R. 1982a. Grammatical relations and Montague Grammar. In Pauline Jacobson & Geoffrey K. Pullum (eds.), *The nature of syntactic representation* (Synthese Language Library 15). Dordrecht: D. Reidel Publishing Company.

Dowty, David R. 1982b. More on the categorial analysis of grammatical relations. In Annie Zaenen (ed.), *Subjects and other subjects: Proceedings of the Harvard Conference on the Representation of Grammatical Relations*, 115–153. Bloomington: Indiana University Linguistics Club.

Dowty, David R. 1982c. More on the categorial analysis of grammatical relations. In Annie Zaenen (ed.), *Subjects and other subjects: Proceedings of the Harvard Conference on the Representation of Grammatical Relations*, 115–153. Bloomington: Indiana University Linguistics Club.

Emele, Martin C. 1994. The typed feature structure representation formalism. In *Proceedings of the international workshop on sharable natural language resources, August 10–11, 1994. Ikoma, Nara, Japan.*

Emonds, Joseph E. 1976. *A transformational approach to English syntax*. New York: Academic Press.

Erbach, Gregor. 1995. ProFIT: prolog with features, inheritance and templates. In Steven P. Abney & Erhard W. Hinrichs (eds.), *Proceedings of the Seventh Conference of the European Chapter of the Association for Computational Linguistics*, 180–187. Dublin: Association for Computational Linguistics.

Fillmore, Charles J. 1988. The mechanisms of "Construction Grammar". In Shelley Axmaker, Annie Jaisser & Helen Singmaster (eds.), *Proceedings of the 14th Annual Meeting of the Berkeley Linguistics Society*, 35–55. Berkeley, CA: Berkeley Linguistics Society.

Flickinger, Daniel. 1983. Lexical heads and phrasal gaps. In *Proceedings of the west coast conference on formal linguistics*, 89–101. Stanford.

Flickinger, Daniel. 2000. On building a more efficient grammar by exploiting types. *Natural Language Engineering* 6(01). 15–28.

Flickinger, Daniel. 2017. Generating English paraphrases from logic. In Martijn Wieling, Gosse Bouma & Gertjan van Noord (eds.), *From semantics to dialectometry: festschrift in honour of john nerbonne*, 99–108. Springer.

Flickinger, Daniel P. 1987. *Lexical rules in the hierarchical lexicon*. Stanford University dissertation.

Flickinger, Daniel P., Ann Copestake & Ivan A. Sag. 2000. HPSG analysis of English. In Wolfgang Wahlster (ed.), *Verbmobil: Foundations of speech-to-speech translation* (Artificial Intelligence), 254–263. Berlin: Springer Verlag.

Flickinger, Daniel P., Carl J. Pollard & Thomas Wasow. 1985. Structure-sharing in lexical representation. In William C. Mann (ed.), *Proceedings of the Twenty-Third Annual Meeting of the Association for Computational Linguistics*, 262–267. Chicago, IL.

Franz, Alex. 1990. *A parser for HPSG*. Laboratory for Computational Linguistics Report CMU-LCL-90-3. Carnegie Mellon University: Laboratory for Computational Linguistics.

Gawron, Jean Mark, Jonathan King, John Lamping, Egon Loebner, E. Anne Paulson, Geoffrey K. Pullum, Ivan A. Sag & Thomas Wasow. 1982. Processing English with a Generalized Phrase Structure Grammar. In *Proceedings of the 20th annual meeting of the Association for Computational Linguistics*, 74–81.

Gazdar, Gerald. 1981. Unbounded dependencies and coordinate structure. *Linguistic Inquiry* 12. 155–184.

Gazdar, Gerald. 1988. Applicability of Indexed Grammars to natural language. In Uwe Reyle & Christian Rohrer (eds.), *Natural language parsing and linguistic theories* (Studies in Linguistics and Philosophy 35), 69–94. Dordrecht/Boston/Lancaster/Tokyo: D. Reidel Publishing Company.

Gazdar, Gerald, Ewan Klein, Geoffrey K. Pullum & Ivan A. Sag. 1985. *Generalized Phrase Structure Grammar*. Cambridge, MA: Harvard University Press.

Ginzburg, Jonathan & Ivan A. Sag. 2000. *Interrogative investigations: The form, meaning, and use of English interrogatives* (CSLI Lecture Notes 123). Stanford, CA: CSLI Publications.

Götz, Thilo & Detmar Meurers. 1997. The ConTroll system as large grammar development platform. In *Proceedings of the Workshop on Computational Environments for Grammar Development and Linguistic Engineering*, 88–97. Madrid: COLIPS Publications.

Hellan, Lars, Tore Bruland, Elias Aamot & Mads H. Sandøy. 2013. A Grammar Sparrer for Norwegian. In *Proceedings of the Nordic Conference on Computational Linguistcs (NoDaLiDa)*, 435–441. Oslo, Norway.

Herbelot, Aurélie & Ann Copestake. 2006. Acquiring Ontological Relationships from Wikipedia using RMRS. In *Proceedings of the ISWC 2006 workshop on web content*. Athens, Georgia.

Hockett, Charles F. 1954. Two models of grammatical description. *Word* 10. 210–234.

Hockett, Charles F. 1958. *A course in modern linguistics*. New York: Macmillan.

Jackendoff, Ray S. 1975. Morphological and semantic regularities in the lexikon. *Language* 51(3). 639–671.

Joshi, Aravind K. 1987. Introduction to Tree Adjoining Grammar. In Alexis Manaster-Ramer (ed.), *The mathematics of language*, 87–114. Amsterdam: John Benjamins Publishing Co.

Joshi, Aravind K., Leon S. Levy & Masako Takahashi. 1975. Tree Adjunct Grammar. *Journal of Computer and System Science* 10(2). 136–163.

Kasami, Tadao, Hiroyuki Seki & Mamoru Fujii. 1989. Generalized context-free grammars and multiple context-free grammars. *Systems and Computers in Japan* 20(7). 43–52.

Kathol, Andreas. 1995. *Linearization-based German syntax*. Ohio State University dissertation.

Kathol, Andreas. 2000. *Linear syntax*. New York, Oxford: Oxford University Press.

Keenan, Edward L. & Bernard Comrie. 1977. Noun phrase accessibility and Universal Grammar. *Linguistic Inquiry* 8(1). 63–99.

Kim, Jongbok. 2016. *The syntactic structures of Korean: A Construction Grammar perspective*. Cambridge, UK: Cambridge University Press.

Kim, Jong-Bok & Jaehyung Yang. 2003. Korean phrase structure grammar and its implementations into the LKB system. In Dong Hong Ji & Kim Teng Lua (eds.), *Proceedings of the 17th Pacific Asia Conference on Language, Information and Computation*, 88–97. National University of Singapore: COLIPS Publications.

King, Paul. 1989. *A logical formalism for Head-Driven Phrase Structure Grammar*. University of Manchester dissertation.

Kuhnle, Alexander & Ann A. Copestake. 2017. Shapeworld - A new test methodology for multimodal language understanding. *CoRR* abs/1704.04517. http://arxiv.org/abs/1704.04517.

Lambek, Joachim. 1958. The Mathematics of Sentence Structure. *The American Mathematical Monthly* 65(3). 154–170.

Laubsch, Joachim & John Nerbonne. 1991. *An Overview of NLL*. Tech. rep. Hewlett-Packard Laboratories.

Levine, Robert D. 2017. *Syntactic analysis: An HPSG-based approach*. Cambridge, UK: Cambridge University Press.

Machicao y Priemer, Antonio. 2015. *SpaGram: An implemented grammar fragment of Spanish*. Ms. Humboldt Universität zu Berlin. In Preparation.

Manandhar, Suresh. 1994. *User's Guide for CL-ONE*. Technical Report. University of Edinburgh.

Meurers, Walt Detmar, Gerald Penn & Frank Richter. 2002. A web-based instructional platform for constraint-based grammar formalisms and parsing. In Dragomir Radev & Chris Brew (eds.), *Effective tools and methodologies for teaching NLP and CL*, 18–25. Association for Computational Linguistics. Proceedings of the Workshop held at 40th Annual Meeting of the Association for Computational Linguistics. Philadelphia, PA.

Miyao, Yusuke, Takashi Ninomiya & Jun'ichi Tsujii. 2005. Corpus-oriented grammar development for acquiring a Head-Driven Phrase Structure Grammar from the Penn Treebank. In Keh-Yih Su, Oi Yee Kwong, Jn'ichi Tsujii & Jong-Hyeok Lee (eds.), *Natural language processing IJCNLP 2004* (Lecture Notes in Artificial Intelligence 3248), 684–693. Berlin: Springer Verlag.

Montague, Richard. 1974. *Formal philosophy: Selected papers of Richard Montague*. Richmond H. Thomason (ed.). Yale University Press.

Moortgat, Michael. 1989. *Categorical investigations: Logical and linguistic aspects of the Lambek Calculus* (Groningen Amsterdam Studies in Semantics 9). Dordrecht/Cinnaminson, U.S.A.: Foris Publications.

Morrill, Glyn V. 1994. *Type Logical Grammars: Categorial logic of signs*. Dordrecht: Kluwer Academic Publishers.

Moshier, M. Drew & William C. Rounds. 1987. A logic for partially specified data structures. In *Proceedings of the 14th ACM Symposium on Principles of Programming Languages*, 156–167. Munich, Germany.

Müller, Stefan. 1996. The Babel-System: An HPSG fragment for German, a parser, and a dialogue component. In *Proceedings of the Fourth International Conference on the Practical Application of Prolog*, 263–277. London.

Müller, Stefan. 1999. *Deutsche Syntax deklarativ: Head-Driven Phrase Structure Grammar für das Deutsche* (Linguistische Arbeiten 394). Tübingen: Max Niemeyer Verlag.

Müller, Stefan. 2007a. *Head-Driven Phrase Structure Grammar: Eine Einführung*. 1st edn. (Stauffenburg Einführungen 17). Tübingen: Stauffenburg Verlag.

Müller, Stefan. 2007b. The Grammix CD Rom: A software collection for developing typed feature structure grammars. In Tracy Holloway King & Emily M. Bender (eds.), *Grammar Engineering across Frameworks 2007* (Studies in Computational Linguistics ONLINE). Stanford, CA: CSLI Publications. http://csli-publications.stanford.edu/GEAF/2007/, accessed 2018-2-25.

Müller, Stefan. 2009. A Head-Driven Phrase Structure Grammar for Maltese. In Bernard Comrie, Ray Fabri, Beth Hume, Manwel Mifsud, Thomas Stolz & Martine Vanhove (eds.), *Introducing Maltese linguistics: Papers from the 1st International Conference on Maltese Linguistics (Bremen/Germany, 18–20 October, 2007)* (Studies in Language Companion Series 113), 83–112. Amsterdam: John Benjamins Publishing Co.

Müller, Stefan. 2010. Persian complex predicates and the limits of inheritance-based analyses. *Journal of Linguistics* 46(3). 601–655. DOI:10.1017/S0022226709990284

Müller, Stefan. 2013a. *Grammatiktheorie*. 2nd edn. (Stauffenburg Einführungen 20). Tübingen: Stauffenburg Verlag. https://hpsg.hu-berlin.de/~stefan/Pub/grammatiktheorie.html, accessed 2018-3-20.

Müller, Stefan. 2013b. Unifying everything: Some remarks on Simpler Syntax, Construction Grammar, Minimalism and HPSG. *Language* 89(4). 920–950. DOI:10.1353/lan.2013.0061

Müller, Stefan. 2015. The CoreGram project: Theoretical linguistics, theory development and verification. *Journal of Language Modelling* 3(1). 21–86. DOI:10.15398/jlm.v3i1.91

Müller, Stefan. 2016. *Grammatical theory: From Transformational Grammar to constraint-based approaches* (Textbooks in Language Sciences 1). Berlin: Language Science Press. DOI:10.17169/langsci.b25.167

Müller, Stefan. 2017. Head-Driven Phrase Structure Grammar, Sign-Based Construction Grammar, and Fluid Construction Grammar: Commonalities and differences. *Constructions and Frames* 9(1). 139–174. DOI:10.1075/cf.9.1.05mul

Müller, Stefan & Walter Kasper. 2000. HPSG analysis of German. In Wolfgang Wahlster (ed.), *Verbmobil: foundations of speech-to-speech translation* (Artificial Intelligence), 238–253. Berlin: Springer Verlag.

Müller, Stefan & Janna Lipenkova. 2013. ChinGram: A TRALE implementation of an HPSG fragment of Mandarin Chinese. In Huei-ling Lai & Kawai Chui (eds.), *Proceedings of the 27th Pacific Asia Conference on Language, Information, and Computation (PACLIC 27)*, 240–249. Taipei, Taiwan: Department of English, National Chengchi University.

Müller, Stefan & Bjarne Ørsnes. 2011. Positional expletives in Danish, German, and Yiddish. In Stefan Müller (ed.), *Proceedings of the 18th International Conference on Head-Driven Phrase Structure Grammar, University of Washington, U.S.A.* 167–187. Stanford, CA: CSLI Publications.

Müller, Stefan & Bjarne Ørsnes. 2013. *Danish in Head-Driven Phrase Structure Grammar* (Empirically Oriented Theoretical Morphology and Syntax). Berlin: Language Science Press. In preparation.

Mykowiecka, Agnieszka, Malgorzata Marciniak, Adam Przepiórkowski & Anna Kupść. 2003. An implementation of a generative grammar of Polish. In Peter Kosta, Joanna Blaszczak, Jens Frasek, Ljudmila Geist & Marzena Zygis (eds.), *Investigations into formal Slavic linguistics: contributions of the fourth European conference on formal description of Slavic languages – FDSL IV held at Potsdam University, November 28–30, 2001*, 271–285. Frankfurt am Main: Peter Lang.

Nerbonne, John & Derek Proudian. 1987. *The HP-NL System*. Tech. rep. Hewlett-Packard Laboratories. URL: http://urd.let.rug.nl/nerbonne/papers/Old-Scans/HP-NL-System-Nerbonne-Proudian-1987.pdf.

Newmeyer, Frederick J. 1980. *Linguistic theory in America: The first quarter-century of transformational generative grammar*. New York: Academic Press.

Oehrle, Richard, Emmon Bach & Deirdre Wheeler (eds.). 1988. *Categorial Grammars and natural language structures*. Dordrecht: D. Reidel Publishing Company.

Oepen, Stephan, Daniel P. Flickinger, Kristina Toutanova & Christopher D. Manning. 2004. LinGO redwoods: A rich and dynamic treebank for HPSG. *Research on Language and Computation* 2(4). 575–596.

Oepen, Stephan, Daniel P. Flickinger, Jun-ichi Tsujii & Hans Uszkoreit. 2002. *Collaborative language engineering: A case study in efficient grammar-based processing* (CSLI Lecture Notes 118). Stanford, CA: CSLI Publications.

Packard, Woodley. 2014. UW-MRS: leveraging a deep grammar for robotic spatial commands. *SemEval 2014*. 812.

Penn, Gerald. 2004. Balancing clarity and efficiency in typed feature logic through delaying. In Donia Scott (ed.), *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), main volume*, 239–246. Barcelona, Spain.

Peters, Stanley & R. W. Ritchie. 1973. On the generative power of Transformational Grammar. *Information Sciences* 6(C). 49–83.

Pollard, Carl J. 1984. *Generalized Phrase Structure Grammars, Head Grammars, and natural language*. Stanford University dissertation.

Pollard, Carl J. 1985. Phrase Structure Grammar without metarules. In Jeffrey Goldberg, Susannah MacKaye & Michael Wescoat (eds.), *WCCFL 4: The proceedings of the Fourth West Coast Conference on Formal Linguistics*, 246–. Stanford Linguistics Association.

Pollard, Carl J. 1986. HPSG tutorial. Handout for tutorial presented at ICOT in Tokyo, in July 1986.

Pollard, Carl J. 1988. Categorial Grammar and Phrase Structure Grammar: An excursion on the syntax-semantics frontier. In Richard Oehrle, Emmon Bach & Deirdre Wheeler (eds.), *Categorial Grammars and natural language structures*, 391–415. Dordrecht: D. Reidel Publishing Company.

Pollard, Carl J. 2013. Traces exist (hypothetically)! https://web.stanford.edu/dept/linguistics/structures_evidence/website/docs/slides/pollard.pdf, accessed 2018-9-10. Presented at the Workshop on Structure and Evidence in Linguistics, Stanford.

Pollard, Carl J. & Ivan A. Sag. 1983. Reflexives and reciprocals in english: an alternative to the binding theory. In *WCCFL83*, 189–203.

Pollard, Carl J. & Ivan A. Sag. 1987. *Information-based syntax and semantics* (CSLI Lecture Notes 13). Stanford, CA: CSLI Publications.

Pollard, Carl J. & Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar* (Studies in Contemporary Linguistics). Chicago: The University of Chicago Press.

Proudian, Derek & Carl J. Pollard. 1985. Parsing Head-Driven Phrase Structure Grammar. In William C. Mann (ed.), *Proceedings of the Twenty-Third Annual Meeting of the Association for Computational Linguistics*, 167–171. Chicago, IL.

Pullum, Geoffrey K. & Gerald Gazdar. 1982. Natural languages and context-free languages. *Linguistics and Philosophy* 4(4). 471–504. DOI:10.1007/BF00360802

Reape, Mike. 1992. *A formal theory of word order: A case study in West Germanic*. University of Edinburgh dissertation.

Reape, Mike. 1996. Getting things in order. In Harry Bunt & Arthur van Horck (eds.), *Discontinuous constituency* (Natural Language Processing 6), 209–253. Berlin: Mouton de Gruyter. Published version of a Ms. dated January 1990.

Retoré, Christian & Edward Stabler. 2004. Generative Grammars in Resource Logics. *Research on Language and Computation*. 3–25.

Ristad, Eric Sven. 1986. Computational complexity of current GPSG theory. In Alan W. Biermann (ed.), *Proceedings of the Twenty-Fourth Annual Meeting of the Association for Computational Linguistics*, 30–39. Columbia University, New York.

Roach, Kelly. 1987. Formal properties of head grammars. In Alexis Manaster-Ramer (ed.), *Mathematics of language*, 293–348. Amsterdam: John Benjamins.

Rounds, William C. & Robert T. Kasper. 1986. A complete logical calculus for record structures representing linguistic information. In *Proceedings of the 15th Annual IEEE Symposium on Logic in Computer Science*, 38–43. Cambridge, MA.

Sag, Ivan A. 1997. English relative clause constructions. *Journal of Linguistics* 33(2). 431–484.

Sag, Ivan A. 2012. Sign-Based Construction Grammar: An informal synopsis. In Hans C. Boas & Ivan A. Sag (eds.), *Sign-based Construction Grammar* (CSLI Lecture Notes 193), 69–202. Stanford, CA: CSLI Publications.

Sag, Ivan A. & Thomas Wasow. 1999. *Syntactic theory: A formal introduction.* Stanford, CA: CSLI.

Sag, Ivan A., Thomas Wasow & Emily M. Bender. 2003. *Syntactic theory: A formal introduction.* 2nd edn. (CSLI Lecture Notes 152). Stanford, CA: CSLI Publications.

Schäfer, Ulrich, Bernd Kiefer, Christian Spurk, Jörg Steffen & Rui Wang. 2011. The ACL Anthology Searchbench. In *Proceedings of the ACL-HLT 2011 system demonstrations*, 7–13. Portland, Oregon, USA.

Scott, Dana S. 1982. Domains for denotational semantics. In M. Nielson & E. M. Schmidt (eds.), *International colloquium on automata, languages and programs, volume 140 of lecture notes in computer science*, 577–613. Springer Verlag.

Shabes, Yves. 1990. *Mathematical and computational aspects of lexicalized grammars.* University of Pennsylvania dissertation.

Shieber, Stuart M. 1985. Evidence against the context-freeness of natural language. *Linguistics and Philosophy* 8(3). 333–343.

Siegel, Melanie. 2000. HPSG analysis of Japanese. In Wolfgang Wahlster (ed.), *Verbmobil: Foundations of speech-to-speech translation* (Artificial Intelligence), 264–279. Berlin: Springer Verlag.

Simpkins, Neil & Marius Groenendijk. 1994. *The ALEP project.* Technical Report. CEC, Luxembourg: Cray Systems.

Steedman, Mark J. 1987. Combinatory grammars and parasitic gaps. *Natural Language and Linguistic Theory* 5(3). 403–439. DOI:10.1007/BF00134555

Steedman, Mark J. 1990. Gapping as constituent coordination. *Linguistics and Philosophy* 13(2). 207–265.

Suppes, Patrick, Dan Flickinger, Elizabeth Macken, Jeanette Cook & L. Liang. 2012. Description of the EPGY Stanford University online courses for Mathematics and the Language Arts. In *Proceedings of the international society for technology in education.* San Diego, California.

Uszkoreit, Hans, Rolf Backofen, Stephan Busemann, Abdel Kader Diagne, Elizabeth A. Hinkelman, Walter Kasper, Bernd Kiefer, Hans-Ulrich Krieger, Klaus

Netter, Günter Neumann, Stephan Oepen & Stephen P. Spackman. 1994. DISCO—An HPSG-based NLP system and its application for appointment scheduling. In Makoto Nagao (ed.), *Proceedings of COLING 94*, 436–440. Kyoto, Japan: Association for Computational Linguistics.

Uszkoreit, Hans & Stanley Peters. 1982. Essential variables in metarules. Paper presented at the annual meeting of the Linguistic Society of America.

van Noord, Gertjan & Gosse Bouma. 1997. HDRUG: A flexible and extendible development environment for natural language processing. In *Proceedings of the workshop "Computational Environments for Grammar Developmentand Linguistic Engineering (ENVGRAM)" held in conjunction with the 35th Annua lMeeting of the ACL and 8th Conference of the EACL*, 91–98. Madrid, Spain: Universidad Nacional de Educación a Distancia.

Vijay-shanker, K. & David J. Weir. 1994. The equivalence of four extensions of context-free grammars. *Mathematical Systems Theory* 27. 511–546.

Wahlster, Wolfgang (ed.). 2000. *Verbmobil: Foundations of speech-to-speech translation* (Artificial Intelligence). Berlin: Springer Verlag.