7

A Tour of Grammar Formalisms

Anette Frank

Having worked with a number of grammatical frameworks over many years at varying depth, I have gained an understanding of their similarities and differences, their respective attractiveness and strengths, but also their biases, which relate to the specific architectural choices they make. In this contribution I will highlight some insights I have gained over years of theoretical and applied research on computational grammar in a multilingual context that might be of interest to researchers in this field – if only to see whether their insights line up with mine.

Our choice of LFG as a guiding theoretical framework is what brought Annie and me together. My first encounter with her, dating back to the time when I finished my studies, was related to discussing linking theory in LFG – a research theme Annie has greatly influenced and that still bears many open questions. I have fond memories of a number of years working with Annie at XRCE Grenoble, investigating LFG from many perspectives. In later work I could compare the insights I gained to my experiences with other frameworks, like HPSG and LTAG.

My personal lesson from the synopsis I give below[1] is that none of the frameworks I discuss is the ultimate answer to how to describe natural languages uniformly within a linguistically sound and expressive computational grammar formalism. Still, I hope that these thoughts can contribute to a better understanding of how these frameworks are similar despite their differences, and different despite their similarities.

[1]The ideas I summarize here were presented in a survey talk at the ACL 2007 Workshop *Deep Linguistic Processing*, where I first reflected on the nature of various grammar formalisms, and dimensions of similarities and differences between them.

From Quirky Case to Representing Space: Papers in Honor of Annie Zaenen.
Tracy Holloway King and Valeria de Paiva.

7.1 Characterizing Grammatical Frameworks

The design of a mathematically defined grammar formalism makes strong predictions as to the grammaticality of linguistic constructs. If the grammatical theory that is built on top of is expected to reflect important characteristics of language crosslinguistically, we also expect it to be able to accommodate typologically distinct languages.[2] In this contribution I will investigate the formal and theoretical-linguistic underpinnings of major computational grammar frameworks from different perspectives.[3] A guiding question will be to what extent architectural choices and linguistic assumptions effect linguistic modeling of particular phenomena, within and across languages.

I will concentrate on a selection of grammatical frameworks that have been subject to intensive research in theoretical and computational linguistics: Lexical-Functional Grammar, LFG (Bresnan, 2001), Head-driven Phrase Structure Grammar, HPSG (Pollard and Sag, 1994), (Lexicalized) Tree Adjoining Grammar, (L)TAG (Joshi, 1988, Joshi and Schabes, 1997), and Combinatory Categorial Grammar CCG (Steedman, 2000). They represent major exponents of lexicalized, constraint- and unification-based grammar (especially LFG and HPSG), different types of tree adjunction grammars (TAGs), and CCG as a special type of categorial grammar (CG).[4]

These frameworks have evolved from different linguistic traditions. (C)CG has its roots in Montague Semantics (Dowty et al., 1981). HPSG, LFG and TAGs are grounded in the tradition of Generative Grammar, even though they arose in opposition to this framework, in a 'lexicalist' turn that questioned the transformation-centered views of Chomskyan syntax. Dependency Grammar, DG (Tesnière, 1959), finally, stands in a long tradition of grammar dating from antiquity. It encodes core grammatical concepts, but has not been extensively studied in modern theoretical syntax.

The particular design choices of these frameworks show interesting differences in how they account for general linguistic properties, such as constituency, word order, and valency. This will be illustrated in a concise overview in Section 7.2. Section 7.3 will further analyze differences and similarities of these frameworks by looking at various aspects of comparison, including (i) architectural choices, focusing on representation levels and linguistic concepts, (ii) adoption of special constructs,

---

[2] I deliberately avoid any discussion of 'language universals'.

[3] Given space restrictions, I will assume familiarity with the respective frameworks. For a concise introduction to these frameworks see Müller (2010).

[4] Dependency Grammar (DG) will only briefly be discussed in the conclusion.

and (iii) generalization across languages. In Section 7.4, I will show that we can reach even deeper insights into the strengths or biases of specific formalization choices by examining how they fare with notoriously difficult phenomena that 'strain' core assumptions of syntax and their implementation in a given framework. To this end, we will look at two notoriously difficult phenomena: complex predicates and coordination.[5]

7.2 Grammar Architecture and Formal Constructs

Obvious design choices that characterize a grammatical theory are its general architecture and the formal constructs used to describe linguistic structures.

Head-driven Phrase Structure Grammar, HPSG

In HPSG *all* levels of linguistic descriptions are uniformly encoded in *typed feature structures*, with *unification* and *type inheritance* as the main formal devices. This uniform perspective on the encoding of grammar is complemented with a rich inventory of hierarchically structured linguistic objects and interacting principles. A grammar is defined as a set of *principles* that define linguistic structures, some 'universal', some language-specific, and language-specific *lexicons*. The principles define constraints across different levels of linguistic description. Subcategorization requirements are defined through lexically defined valence lists and principles coordinating their realization and saturation in diverse structural configurations. This includes the treatment of long-distance constructions, which are covered by the interplay of subcategorization, non-local-feature projection and constituency principles, through stepwise projection of *nonloca*l elements from gap to filler position.

The most striking characteristics of HPSG are (i) uniform encoding of linguistic structure in typed feature structures, ranging from phonology to semantics, (ii) free interaction of modular principles across typed structures, which jointly determine grammaticality, and (iii) tight integration of syntax and semantics. The latter is seen most clearly in the collapsed ***synsem*** type that specifies the nature of subcategorized arguments. (iv) In contrast to LFG or DG, HPSG does not treat grammatical functions as primitive concepts in its grammar archictecture.

Lexical-Functional Grammar, LFG

LFG's architecture encodes a system of *functional projections* that distinguish constituent, functional and semantic structure as independent

[5]Müller (2010) offers a by-far more rigorous description and comparison of grammatical frameworks than what is possible within the scope of this article. Our main novel contribution is related to the discussion in Section 7.4.

levels of grammatical description. Each level is encoded using an individually motivated formalism: tree structures for constituency, and attribute-value (feature) structures for the encoding of functional and morphosyntactic properties and subcategorization. Principles governing grammatical wellformedness are stated on individual levels (most prominently, f-structure), but also across levels, constraining structure-to-function correspondences, or argument linking. This co-description architecture accommodates non-isomorphism between structures, especially word order variation and discontinuity in surface structure. For *non-local* dissociations of constituency and functional embedding, as in long-distance dependencies, LFG adopts *functional uncertainty* as a formal device that bridges the (potentially unlimited) dissociation of argument realization in the mapping between c- and f-structure.

The most striking characteristics of LFG are (i) its distributed projection architecture, which makes it possible to (ii) study and process syntax independently from semantics, (iii) its strong focus on grammatical functions as a primitive concept for (crosslingual) grammatical description, and (iv) the dissociation of context-free surface constituency encoding vs. functional representation in feature structures.

Lexicalized Tree Adjoining Grammar, LTAG

LTAG shares with LFG the encoding of surface syntactic properties in constituent tree structures and a modular interface to semantic representation so that syntax can be defined and processed independently from semantics. Its grammar architecture is based on *tree adjunction* as the central mechanism for structure composition. The grammar is built from lexicalized *elementary trees (etrees)* that are composed by *substitution* and *adjunction operations*. The latter is not restricted to syntactic modification, but serves as a general device for *factoring recursion* – one of the most prominent characteristics of language and a guiding principle for finite grammatical description. *etrees* fulfill two functions: they encode *argument structure* and they pre-define surface properties that account for order variation, diatheses such as the passive, or *wh-* and relative clause constructions. A wide variation of such etree variants is organized in automatically generated *tree families*.

The most striking characteristics of LTAG are (i) its formalization of recursion in terms of adjunction applied to tree fragments, i.e. *tree adjunction*. By this move, it does not require additional devices for capturing long-distance dependencies. (ii) LTAG offers *derived* and *derivation trees* as parallel syntactic structures. The derivation tree records the history of tree compositions and traditionally serves as the basis for semantic projection. (iii) LTAG does not adopt grammatical functions as a central notion of grammar. (iv) The theory puts less emphasis on constraints that govern the internal structure of etrees, and thus on the shape of the resulting derived tree. This would be possible by adopting core X′-principles, similar to LFG.

Combinatory Categorial Grammar, CCG

CCG differs from the previous frameworks in that it is strongly influenced by semantics, notably Montague Grammar. It employs a small number of *syntactic composition operations*: i.a., forward and backward application, composition and type raising. Syntactic composition is guided by semantic composition that operates in parallel with syntax. *Syntactic categories* are either atomic or complex categories that encapsulate the way the category can be embedded in its constructional context by syntactic/semantic composition rules. In this way, syntax is modeled as a composition process driven by complex categories that externalize their constructional context, rather than by traditional construction-specific rules. *Type raising* in conjunction with composition accounts for *non-local dependencies* and other non-standard constructions such as raising and coordination. In contrast to LTAG, which derives dependency-like structures from the history of derivations, CCG records and outputs predicate-argument dependencies as defined in lexical types. Saturation of argument structure is achieved by deriving the target category.

The most striking characteristics of CCG are (i) that syntax and semantics are highly intertwined, with syntax merely considered a side-process running in parallel with semantic composition. (ii) Syntactic categories can be type-raised to complex categories that encapsulate their constructional context. (iii) The categories are defined to reflect core syntactic properties, but the derived constituents can diverge considerably from traditional assumptions.

7.3 Architectural Choices and Linguistic Modeling

Looking at these characteristics, we can map out similarities and differences between the frameworks along various dimensions (cf. Fig.1).

Architecture and Main Focus

LFG and LTAG assume a clear *separation between syntax and semantics*, while for HPSG and CCG it is more difficult to dissociate semantics from syntax. We characterize this as *modular vs. integrated architectures*. At the same time, HPSG and CCG are fundamentally different in that HPSG models syntax in a highly articulated structure representing core syntactic properties and wellformedness constraints, while CCG mainly defines valid surface structures in a generative process driven by syntactic/semantic compositions. That is, CCG's focus is on *semantics and the syntactic encoding of argument structure*, including a proper treatment of core syntactic constructions, such as binding, reflexivization, control or raising and the like. HPSG takes a somewhat broader look at grammar. Its core theory encompasses an articulated representation of linguistic objects that is constrained by general interacting *syntactic and semantic principles of composition*. Syntax and semantics are strongly intertwined and can only be described jointly.[6]

LFG and LTAG syntax is more clearly dissociated from semantics. Here, syntax is conceived of as an independent grammatical system. Both have been coupled with various semantic representation layers and diverse semantics construction architectures.[7]

Formal Devices, Representation and Generalization

Looking at formal devices, HPSG employs a rich formalization using *typed feature structures*, with sophisticated encoding of linguistic objects and structures that are constrained by interacting principles. The rich representation of structural layers and interacting principles readily accounts for the treatment of long-distance dependencies in a feature propagation analysis that is reminicent of dislocation analyses in Generative Grammar. While 'classical' HSPG analyses control surface order by way of phrase structure schemata, *linearization-based accounts* allow for dissociation of surface realization and phrasal constituent structure by way of independent linearization constraints.

LFG, with its system of parallel projections and especially its *dissociation of constituency and functional structure*, allows for a very flexible encoding of surface realization within and across languages. Non-local realization of arguments is mediated by *functional uncertainty* – an equivalent to the unbounded feature passing devices of HPSG-like formalisms that operates on the level of f-structure.[8] [9]

HPSG and LFG share a constraint-based view of grammatical structure with articulated representations and principles of wellformedness

[6]See for instance the semantic construction algebra of Copestake et al. (2001).

[7]For LFG see the co-description vs. description-by-analysis architectures (Halvorsen and Kaplan, 1988) and the resource-logic account (Dalrymple, 2001, Crouch et al., 2001). For LTAG see Kallmeyer and Joshi (1999) or Gardent and Kallmeyer (2003) and synchronous TAG (Shieber and Schabes, 1990).

[8]Advantages of functional vs. constituent-based constraints on extraction and binding have been discussed in Kaplan and Zaenen (1995) and Dalrymple (1993).

[9]Discontinuous phrases marked by morphological case as discussed in Nordlinger (1998) can be resolved in HPSG using a feature-passing device similar to long-distance dependencies (Bender, 2008). This is largely equivalent to the LFG analysis using functional uncertainty.

operating on them. LTAG and CCG take a more generative perspective on syntax, with sparser representational devices. Linguistic and constructional properties of words and phrases are captured in a theory of complex lexical categories, or complex encoding of families of etrees, which are carefully designed to generate valid surface structures for a given language or as a basis for semantic construction from predicate-argument structures.

LTAG encapsulates argument structure in *etrees* and applies *tree adjunction* as its main syntactic composition operation. Since etrees must express a wide variety of structures, an important line of research pursues a 'meta grammar' approach as a general framework for describing and factoring TAG grammars that offers an abstract level of grammatical description for defining the set of admissible etrees for a given language.[10] In LTAG, due to tree adjunction as a general compositional device, no additional devices are required to account for non-local dependencies.[11] Yet this specific take on recursion comes at the price of an asymmetry between adjunction as a recursion building process as opposed to adjunction as a structural indicator of linguistic modification, as traditionally assumed in X′ syntax.[12]

In CCG, type raising and composition account for a wide spectrum of constructions, including long-distance dependencies. Type-raised categories may also be used to encode notions of case, in terms of externalized structural configurations. Argument structure is defined by complex lexical categories, in terms of the arguments they specify. From these lexical definitions full-fledged dependency representations can be derived in parsing (Clark et al., 2002). Thus, the syntactic formalism proves homogeneous and representationally sparse and offers great variety in structural exponence of syntactic properties and semantic content. While syntacticians do not find traditional notions of constituency and projectivity in the syntactic derivation structure, core syntactic properties and constructions are modeled in a lexicon theory of complex categories.[13]

[10] See i.a. Candito (1996), Doran et al. (2000), Crabbé et al. (2012).

[11] Though this requires careful definition of lexical or tree families, see above.

[12] Since LTAG's preferred structure for semantic construction is the derivation tree, this lack of discrimination between modification and complementation has implications for the projection of semantics from syntax. Alternatively, semantic construction can be based on the derived tree. See Frank and van Genabith (2001), Gardent and Kallmeyer (2003) and Cimiano et al. (2007) for more detail.

[13] See Steedman and Baldridge (2011) on the encoding of binding, extraction, raising and control, and gapping.

Cross-linguistic Language Modeling

As seen above, the different formalisms have differing foci in expressing and representing linguistic structure and generalizations. This may have an impact on insights gained by cross-linguistic language modeling.

The *principle-driven formalism of HPSG* offers strong formal support for cross-linguistic syntactic description, which is documented by the *Grammar Matrix* and its extension to grammar fragments for a great variety of languages (Bender et al., 2002). Generalizations can be defined using type inheritance, as well as language-specific parameterizations of general principles (for constituent order, case, etc.).

LFG's focus is on *f-structure as an independent level of grammatical description*. Consequently, the theory draws important cross-linguistic generalizations linked to the concept of grammatical functions. This includes argument realization in linking theory and constraints observed in extraction and binding constructions.[14] Less prominent have been its generalizations regarding constituency and mapping principles to f-structure (Bresnan, 2001).[15] Multilingual grammar development in the ParGram project has proven that f-structure can offer a pivot for aligning grammars cross-linguistically, requiring little variation across typologically diverse languages (King et al., 2005).

In both frameworks, the encoding of interactions between word order, constituency and morphological marking has led to important insights into the grammar of nonconfigurational languages and morphological marking strategies across languages.

Research in LTAG and CCG is restricted to a smaller community. Accordingly the range of multilingual studies is less diverse.[16] However, it has been shown, by wide-coverage treebank-based grammar induction and parsing of corpora in different languages, that these formalisms are able to analyze a wide range of linguistic constructions.[17]

7.4 Straining Theories

The grammatical frameworks under discussion show considerable differences in how they encode grammatical concepts, most importantly argument structure and its interplay with surface realization. Yet the consequences of these formalization choices are limited, as long as we

[14]See Bresnan and Zaenen (1990), Dalrymple (1993), Kaplan and Zaenen (1995) and Butt et al. (1997).

[15]But see the formalization of mapping principles in treebank-based LFG grammar induction (Frank et al., 2001).

[16]See e.g. Kroch and Joshi (1985), Becker et al. (1991), Kinyon et al. (2006) for LTAG and the overview in Steedman and Baldridge (2011).

[17]See references above and the overview in van Genabith et al. (2006).

concentrate on core constructions and ignore questions of personal taste or adherence to traditional notions of grammatical description.

In fact, it is by looking at linguistic phenomena that 'strain' general assumptions about grammar encoding that we can gain more insight into possible biases of particular formalization choices. I will thus take a closer look at two phenomena related to argument structure realization that present true challenges to any grammatical framework, and reflect on their way of handling these. For exposition I will concentrate on their analyses in LFG and HPSG. But our observations will bring out further aspects of formalization choices that clearly differentiate LFG from HPSG, and also LTAG from CCG.

Complex Predicates (such as causatives or coherently constructed infinitive embedding verbs) are subject to intensive research in HPSG and LFG. Linguistic evidence calls for an analysis in terms of *clause union* or *argument composition* that conflates the arguments of two predicates into a monoclausal structure to account for the argument relation changes and surface realizations characterized as *long scrambling*.

Complex predicates present a particular challenge for LFG: in order to account for their monoclausal properties, two lexical predicates need to be turned into a single predicator with redefined argument characteristics. Butt (1995) and Alsina (1996) employ a *restriction operator* that constructs a joint predicate 'on the fly' in syntax. This causes a disruption in the functional projection and leads to problems in defining relation changing processes (e.g., reflexivization, passivization) that need to apply in the lexicon. Frank (1996), and more recently Bouma and Kuhn (2009), therefore proposes an alternative analysis with lexical rules that (re)define the involved predicates as co-predicators in the lexicon, where relation changing processes can apply in the usual way.

These contrastive approaches reveal a bias in LFG's grammar architecture: syntactic arguments that may be realized in dissociated phrasal structures are integrated into complete, fully specified f-structure nuclei by means of functional head projection rules. This mechanism allows for an elegant analysis of local and nonlocal surface realization phenomena by means of f-structure equations defined over functional paths. At the same time, this characteristic of LFG makes it difficult to accommodate dynamic changes of argument structures in complex predicate formation.

In HPSG we do not find a layer comparable to f-structure that represents the complete syntactic argument structure of a clause. Argument structure is essentially defined in the lexicon's SUBCAT list, where it is directly linked to the semantic representation. In syntactic composition, the SUBCAT list is *redefined* in each phrasal projection to record saturation and the realization of arguments. In this architecture complex predicate formation can be defined through *argument composition in syntax*, yet *(pre)defined in the lexicon*, as proposed by Hinrichs and Nakazawa (1994): the SUBCAT list of the main predicate attracts the arguments of a co-predicate into its own argument list. This produces a joint argument structure as soon as the co-predicate is encountered in syntax. At the same time, lexical syntactic processes can apply to the incorporated or the composed SUBCAT list in the lexicon.[18]

(Asymmetric) Coordination presents another challenge related to argument realization, as it typically involves factorization of one or more arguments that are shared between coordinated predicates.

Coordination is handled in similar ways in LFG and HPSG, yet here, the differences we highlighted above favor LFG's way of coding syntax by means of a monostratal and fully connected f-structure representation. This can be observed by looking at a special type of *asymmetric* coordination that is frequently observed in German, and illustrated in (1).[19] The puzzle this construction presents is that the joint subject of the coordinated sentential phrases is deeply embedded within the first conjunct (German is V2), but seems to be accessible for binding the subject gap in the second conjunct. This construction presents a true challenge for any theory that is based on notions of constituency.

(1) Im Park sitzen Leute und erzählen Geschichten.
in the park sit people and recount stories
People are sitting in the park and are recounting stories.

LFG and HPSG both account for shared arguments in coordination by joint reference to a single argument (cf. Fig. 2). In HPSG (upper left), this is encoded in the lexical entry of the coordinating conjunction: the non-consumed arguments of all coordinated phrases are coindexed with the arguments on the phrase's SUBCAT list. This allows for standard coordination structures with a shared subject realized outside the coordinated VP. In LFG, a SUBJECT that is realized outside of the

[18] This difference between LFG's and HPSG's representation architecture becomes apparent in Andrews and Manning (1999)'s reformulation of LFG in a *spreading information* account that dissociates the contribution of different feature types into separate layers. Here, LFG's uniform $\uparrow=\downarrow$ head projection rules are dissociated according to feature types f: $\uparrow_f=\downarrow_f$. This mimics an HPSG-like architecture, with the possibility of redefining individual features in phrasal composition, and thus allows for argument composition along the lines of Hinrichs and Nakazawa (1994).

[19] See Höhle (1983), Wunderlich (1988), Steedman (1990), Kathol (1999), and Frank(2002, 2006) for more detailed discussion of this construction.

coordinated VP is defined in the conjoined phrase's f-structure. From there it is *distributed* into the f-structures of the coordinated phrases and thus fills their respective subcategorization requirements (cf. Dalrymple (2001)). Neither of these standard analyses for coordination accounts for cases of asymmetric coordination as in (1).

Frank (2002, 2006) motivates an analysis of asymmetric coordination in analogy to modal subordination constructions. In this analysis the first sentential conjunct licenses an *extension of its discourse-functional domain* that includes the second conjunct. Operators that can perform such domain extensions are *grammaticalized discourse functions*, here the SUBJect. Domain extension is defined by asymmetric projection of the SUBJect from the first conjunct's clausal node to the coordinated phrase (Fig. 2, middle left). From there, the SUBJect is distributed to the second conjunct by applying LFG's distribution mechanism.

Could a similar analysis be designed for HPSG? This is not possible without further ado, precisely because HPSG does *not*, in contrast to LFG, offer an integrated monostratal syntactic argument structure where all arguments 'float' up and down along syntactic head projection lines. In HPSG's coordinated phrase for (1), the subject of the first conjunct is not accessible from the SUBCAT list of the first conjunct to 'fill' the open subject slot of the second conjunct (cf. Fig. 2): The conjuncts are symmetric with regard to their constituent phrases, but asymmetric regarding saturation. One way of solving this problem is to resort to the ARG-ST list, usually employed for expressing binding constaints, that represents a copy of the complete SUBCAT list, as defined in the lexical entry (cf. Pollard and Sag (1994)).[20]

Lesson I. Levels of Representation. In sum, by looking at exceptional linguistic structures that strain basic assumptions of linguistic formalization, a principled difference in the representation architecture of LFG and HPSG shows up. LFG's encoding of a complete clausal nucleus in functional structure defines its interface to semantics (Dalrymple, 2001) and also offers great flexibility in accessing argument functions non-locally along head projections. The latter turns out as an advantage in the case of asymmetric coordination, yet as a problem

[20]Indeed, related problems have been faced in the description of Germanic V2 and generally, verb initial constructions. Borsley (1989) proposed a *double slash* feature DSL that makes the *complete* lexical SUBCAT list available along the head projection, mimicking local head movement. A similar mechanism had been proposed for CG by Jacobson (1987). Thus, one could adopt the ARG-ST or the DSL mechanism to make the SUBCAT list (and with it the subject) available along the head projection, to make it accessible from the second conjunct. Technically, this opens the way for an analysis along the lines of Frank (2002, 2006), yet it needs to be integrated with HPSG's core analysis of coordination.

when trying to integrate complex predicate constructions in a monoclausal syntactic representation. HPSG, by contrast, lacks an integrated syntactic representation layer. Syntactic arguments defined in the SUBCAT list are directly linked to semantics. The SUBCAT feature is discharged stepwise, as arguments are realized syntactically. This way of specifying and controlling clausal argument structure explains the greater flexibility of HPSG in accounting for complex predicates: composed argument structures can be built on the fly, without requiring full integration into a monoclausal syntactic structure. Yet, the lack of such a representational layer is what prevents non-local access to arguments along the head projection, and thus the binding of subject gaps in asymmetric coordination structures.

Lesson II: Argument Encoding and Surface Realization in LTAG & CCG. This observation brings us back to LTAG and CCG. These theories offer sparser formalizations than LFG and HPSG in terms of representational devices. Do they fare better with these exceptional construction types?

LTAG, with its free encoding of argument structure in etrees, could be expected to flexibly accommodate structural asymmetries in coordination. But for LTAG it is the *factorization* of arguments in coordination that challenges its strongest assumption: the encoding of *full argument structures* in etrees. The problem is illustrated in Fig. 2 (lower, right) for symmetric VP coordination: LTAG has to cope with multi-rooted derived structures in parsing, and needs to focus on derivation structure to derive valid argument and semantic structures.[21]

CCG bears a strong resemblance to the way arguments are processed in HPSG. Argument structure is defined in lexical types, i.e. families of complex categories that account for diverse structural realizations. The stepwise reduction of complex categories to infer a clausal category is similar to the reduction of the SUBCAT list, as is the composition of the encountered arguments into full argument structures in a concurrently processed semantic structure. CCG shows even stronger flexibility than HPSG, in that it does not encode a rich system of general principles of linguistic structure, especially, phrasal structure. The free application of composition operations may produce structures that do not correspond to traditional notions of phrase structure.[22] In fact, Steedman (1990) shows how a special *decomposition operation* detaches the em-

[21] Sarkar and Joshi (1996) propose a *conjoin* operator to merge identical nodes. This approach is further developed in recent work by Banik (2004), Seddah (2008) and Lichte and Kallmeyer (2010).

[22] Steedman and Baldridge (2011) motivate such exceptional phrase structures as a natural way of integrating information structure with semantics.

bedded subject in asymmetric coordination constructions and makes it accessible as a shared subject in the second conjunct (cf. Fig. 2, upper right).[23]

In sum, when it comes to coordination, CCG's discharging processes for argument structure realization prove to be highly flexible, whereas LTAG suffers from a more rigid encoding of full argument structures in etrees. In this respect, it bears similarities to LFG's representation of clausal nuclei in terms of subcategorized grammatical functions, yet at the level of phrase-structural encoding. Finally, for sake of completeness, let us note that nonlocal argument serialization in complex predication constructions has been studied extensively in (MC)TAG and CCG.[24] While it has been assumed that the LTAG and CCG formalisms are equivalent in terms of serialization capacities, recently Hockenmaier and Young (2008) established that there are configurations that can be generated with CCG that cannot be generated with TAG (see also Kuhlmann et al. (2010)).

7.5 Conclusions

Beyond the aspects of linguistic modeling proper, formal design choices have implications for grammar engineering and processing complexity, as well as techniques for grammar induction and automatic disambiguation.[25] All the frameworks under discussion have developed sophisticated grammar engineering platforms and efficient parsing techniques, including stochastic disambiguation. It is with techniques for automatic grammar induction from treebanks that we can again observe that different views on grammar constructs and detail of representation are clearly reflected in the proposed techniques: articulated frameworks like HPSG and LFG require considerable ingestion of linguistic knowledge to define finer-grained distinctions or linguistic principles not reflected in classic treebanks, whereas, at least theoretically, algorithms for LTAG and CCG grammar inducion can rely on leaner methods.

This survey tries to shed some light on similarities and differences among grammatical frameworks in how their particular take on the formalization of linguistic concepts is reflected in different foci of research as well as potential biases in the formalization of syntactic phenomena.

[23] Note that the *decomposition* operator is not in the scope of constructors generally considered in CCG formalizations.

[24] See e.g. Becker et al. (1991), Rambow (1994), Joshi et al. (2000), and Steedman and Baldridge (2011).

[25] These aspects could not be discussed in this contribution, but are integrated in Fig. 1 for completeness. For an overview regarding the generative capacity of the respective formalisms see e.g. Müller (2010).

These stand out most clearly in the treatment of special constructs that strain the borders of general syntactic principles.

From our observations we may conclude that HPSG's take on grammar is the most articulated one and is the most closely related to traditional structure-oriented, GB-style notions of syntax, through its traditional take on non-local dependencies and its principle-driven account of grammar formalization. CCG, LTAG and LFG each adopt specific assumptions and constructs, with LFG being closely related to HPSG in offering a representation- and constraint-based theory of syntax that stays close to traditional notions of syntactic description. LFG is special in choosing grammatical functions as its main descriptive device, and is thus close to Dependency Grammar, a framework that is seeing a strong revival, supported by efficient parsing algorithms. It has been shown in recent multilingual parsing challenges that dependency-based syntactic analysis is applicable to many languages without major adjustments. It offers a lean representational view on syntax that is close to LFG's f-structure representation with all its strengths and weaknesses, yet little emphasis on surface constituency. CCG maybe in fact turn out to be the most versatile and flexible grammar framework, one that is capable of bridging large discrepancies between surface form and semantic encoding, across a wide variety of languages and constructions. However, this enormous flexibility needs to be paired with the cautious statement of linguistic constraints that restrict the space of possible structures to those that are (cross-linguistically) grammatical and adhere to linguistic constraints on the association of form and meaning.

Acknowledgements. Thanks go to Claire Gardent and Julia Hockenmaier, as well as two anonymous reviewers for insightful comments on an earlier version of this paper. Any errors are my own responsibility.

References

Alsina, Alex. 1996. *The Role of Argument Structure in Grammar*. CSLI Publications.

Andrews, Avery D. and Christopher D. Manning. 1999. *Complex Predicates and Information Spreading in LFG*. CSLI Publications.

Banik, Eva. 2004. Semantics of VP coordination in LTAG. In *Proceedings of the Seventh International Workshop on Tree Adjoining Grammar and Related Formalisms, TAG+7*, pages 118–125.

Becker, Tilman, Aravind K. Joshi, and Owen Rambow. 1991. Long-distance scrambling and Tree Adjoining Grammars. In *Proceedings of the Fifth Conference of the European Chapter of the Association for Computational Linguistics*, pages 21–26.

	CCG	HPSG	LFG	LTAG
architecture	integrated	integrated	modular	modular
main focus	semantics	syntax & semantics	f-structure	derivation tree
main constructs	function composition	principles & unification	functional projection	tree composition
theory-specific devices	type raising	word order domains	functional uncertainty	tree adjunction
languages	E, G, NL, Tur, ...	many (Matrix)	many (Pargram)	E, Zh, F, I, G, ..
means of abstraction	lexical families	types & inheritance	templates & macros	tree families
processing efficiency	–	packing	packing	–
grammar induction	yes	yes	yes	yes
statistical selection	supertagging	supertagging	–	supertagging
statistical disambiguation	yes	yes	yes	yes

FIGURE 1 Facets and Dimensions across Frameworks

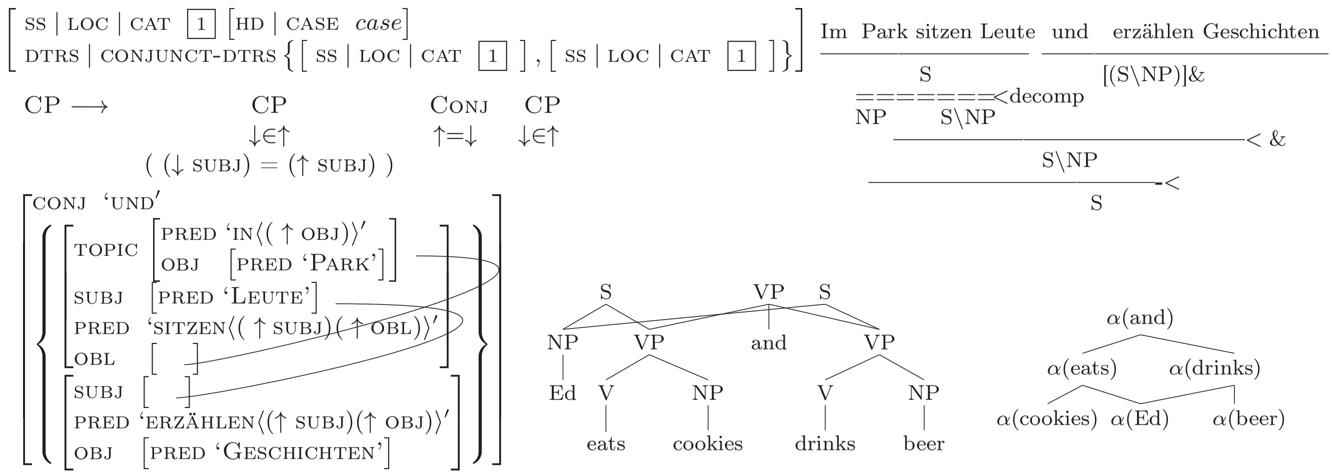

FIGURE 2 Cross-framework Analyses of Coordination: left: HPSG (upper), LFG (lower); right: CCG (upper), LTAG (lower)

Bender, Emily M. 2008. Radical non-configurationality without shuffle operators: An analysis of Wambaya. In S. Müller, ed., *The Proceedings of the 15th International Conference on Head-Driven Phrase Structure Grammar*, pages 6–24. CSLI Publications.

Bender, Emily M., Dan Flickinger, and Stephan Oepen. 2002. The Grammar Matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, pages 8–14.

Borsley, Robert D. 1989. Phrase-structure grammar and the barriers conception of clause structure. *Linguistics* pages 843–863.

Bouma, Gerlof and Jonas Kuhn. 2009. On the split nature of the Dutch latencausative. In M. Butt and T. H. King, eds., *Proceedings of the LFG'09 Conference*, pages 167–187. CSLI Publications.

Bresnan, Joan. 2001. *Lexical-Functional Syntax*. Blackwell Publishers.

Bresnan, Joan and Annie Zaenen. 1990. Deep unaccusativity in LFG. In K. Dziwirek, P. Farrell, and E. Mejias-Bikandi, eds., *Grammatical Relations. A Cross-Theoretical Perspective*, pages 45–57. CSLI Publications.

Butt, Miriam. 1995. *The Structure of Complex Predicates in Urdu*. CSLI Publications.

Butt, Miriam, Mary Dalrymple, and Anette Frank. 1997. An architecture for linking theory in LFG. In M. Butt and T. H. King, eds., *Proceedings of the LFG-97 Conference*, CSLI Online Publications.

Candito, Marie-Helene. 1996. A principle-based hierarchical representation of LTAGs. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING)*, pages 194–199.

Cimiano, Philipp, Anette Frank, and Uwe Reyle. 2007. UDRT-based semantics construction for LTAG — and what it tells us about the role of adjunction in LTAG. In *Proceedings of the 7th International Workshop on Computational Semantics, IWCS-7*.

Clark, Stephen, Julia Hockenmaier, and Mark Steedman. 2002. Building deep dependency structures using a wide-coverage CCG parser. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 327–334. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics.

Copestake, Ann, Alex Lascarides, and Dan Flickinger. 2001. An algebra for semantic construction in constraint-based grammars. In *Proceedings of the 39th ACL*.

Crabbé, Benoit, Denis Duchier, Claire Gardent, Joseph Leroux, and Yannik Parmentier. 2012. XMG: eXtensible MetaGrammar. *Computational Linguistics* to appear.

Crouch, Richard, Anette Frank, and Josef van Genabith. 2001. Glue, underspecification and translation. In H. C. Bunt, ed., *Computing Meaning Volume 2*. Kluwer Academic Publishers.

Dalrymple, Mary. 1993. *The Syntax of Anaphoric Binding*. CSLI Lecture Notes. CSLI Publications.

Dalrymple, Mary. 2001. *Lexical-Functional Grammar*, vol. 34 of *Syntax and Semantics*. Academic Press.

Dalrymple, Mary, Ronald M. Kaplan, John T. Maxwell III, and Annie Zaenen, eds. 1995. *Formal Issues in Lexical-Functional Grammar*. CSLI Lecture Notes, No. 47. CSLI Publications.

Doran, Christy, Beth Ann Hockey, Anoop Sarkar, B. Srinivas, and Fei Xia. 2000. Evolution of the XTAG system. In A. Abeillé and O. Rambow, eds., *Tree Adjoining Grammars: Formalisms, Linguistic Analysis and Processing*, pages 371–404. CSLI Publications.

Dowty, David R., Robert E. Wall, and Stanley Peters. 1981. *Introduction to Montague Semantics*. Reidel, Dordrecht.

Frank, Anette. 1996. A note on complex predicate formation: Evidence from auxiliary selection, reflexivization, passivization and past participle agreement in French and Italian. In M. Butt and T. H. King, eds., *Proceedings of the 1st LFG Conference*. CSLI Online Publications.

Frank, Anette. 2002. A (discourse) functional analysis of asymmetric coordination. In *Proceedings of LFG 2002*, pages 174–196.

Frank, Anette. 2006. A (discourse-) functional analysis of asymmetric coordination. In M. Butt, M. Dalrymple, and T. H. King, eds., *Intelligent Linguistic Architectures: Variations on themes by Ronald M. Kaplan*, pages 259–285. CSLI Publications.

Frank, Anette, Louisa Sadler, Josef van Genabith, and Andy Way. 2001. From treebank resources to LFG f-structures. Automatic f-structure annotation of treebank trees and CFGs extracted from treebanks. In A. Abeille, ed., *Building and Using Syntactically Annotated Corpora*. Kluwer Academic Publishers.

Frank, Anette and Josef van Genabith. 2001. LL-based semantics for LTAG — and what it teaches us about LFG and LTAG. In *Proceedings of LFG 2001*, pages 104–126.

Gardent, Claire and Laura Kallmeyer. 2003. Semantic construction in F-TAG. In *Proceedings of the 10th Conference of the European Chapter for the Association of Computational Linguistics, EACL 2003*, pages 123–130.

Halvorsen, Per-Kristian and Ronald M. Kaplan. 1988. Projections and semantic description in Lexical-Functional Grammar. In *Proceedings of the International Conference on Fifth Generation Computer Systems*, pages 11–16. Reprinted in: Mary Dalrymple et al. 1995. eds.: *Formal Issues in Lexical Functional Grammar*, CSLI Publications.

Hinrichs, Erhard and Tsuneko Nakazawa. 1994. Linearizing finite AUX in German complex VPs. In J. Nerbonne, K. Netter, and C. Pollard, eds., *German in Head-Driven Phrase Structure Grammar*, vol. 46 of *CSLI Lecture Notes*. CSLI Publications.

Hockenmaier, Julia and Peter Young. 2008. Non-local scrambling: The equivalence of TAG and CCG revisited. In *Proceedings of The Ninth International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+9)*.

Höhle, Tilman. 1983. Subjektlücken in Koordinationen. Unpublished manuscript, University of Cologne.

Jacobson, Pauline. 1987. Phrase structure, grammatical relations, and discontinuous constituents. In G. J. Huck and A. E. Ojeda, eds., *Discontinuous Constituency*, vol. 20 of *Syntax and Semantics*, pages 27–69. Academic Press.

Joshi, Aravind K. 1988. Tree Adjoining Grammars. In D. Dowty, L. Karttunen, and A. Zwicky, eds., *Natural Language Parsing*. Cambridge University Press.

Joshi, Aravind K., Tilman Becker, and Owen Rambow. 2000. Complexity of scrambling: A new twist to the competence-performance distinction. In A. Abeillé and O. Rambow, eds., *Tree Adjoining Grammars: Formalisms, Linguistic Analysis and Processing*, pages 167–181. CSLI Publications.

Joshi, Aravind K. and Yves Schabes. 1997. Tree Adjoining Grammars. In A. Salomma and G. Rosenberg, eds., *Handbook of Formal Languages and Automata*. Springer Verlag.

Kallmeyer, Laura and Aravind K. Joshi. 1999. Factoring predicate argument and scope semantics: Underspecified semantics with LTAG. In P. Dekker, ed., *Proceedings of the 12th Amsterdam Colloquium*, pages 169–174.

Kaplan, Ronald M. and Annie Zaenen. 1995. Long-distance dependencies, constituent structure, and functional uncertainty. In M. Dalrymple, R. Kaplan, J. M. III, and A. Zaenen, eds., *Formal Issues in Lexical Functional Grammar*, CSLI Lecture Notes, pages 137–165. CSLI Publications.

Kathol, Andreas. 1999. Linearization vs. phrase structure in German coordination constructions. *Cognitive Linguistics* 10(4):303–342.

King, Tracy Holloway, Martin Forst, Jonas Kuhn, and Miriam Butt. 2005. The feature space in parallel grammar writing. *Research on Language and Computation* 3(2-3):139–163.

Kinyon, Alexandra, Owen Rambow, Tatjana Scheffler, SinWon Yoon, and Aravind K. Joshi. 2006. The metagrammar goes multilingual: A crosslinguistic look at the V2-phenomenon. In *Proceedings of The Eighth International Workshop on Tree Adjoining Grammar and Related Formalisms, TAG+8*, pages 17–24.

Kroch, Anthony S. and Aravind K. Joshi. 1985. The linguistic relevance of Tree Adjoining Grammar. Tech. rep., University of Pennsylvania.

Kuhlmann, Marco, Alexander Koller, and Giorgio Satta. 2010. The importance of rule restrictions in CCG. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 534–543.

Lichte, Timm and Laura Kallmeyer. 2010. Gapping through tag derivation trees. In *Proceedings of the 10th International Workshop on Tree-Adjoining Grammar and related Formalisms (TAG+10)*.

Müller, Stefan. 2010. *Grammatiktheorie*. No. 20 in Stauffenburg Einführungen. Tübingen: Stauffenburg Verlag.

Nordlinger, Rachel. 1998. *Constructive Case: Evidence from Australian Languages*. CSLI Publications.

Pollard, Carl and Ivan Sag. 1994. *Head-Driven Phrase Structure Grammar*. The University of Chicago Press.

Rambow, Owen. 1994. *Formal and Computational Aspects of Natural Language Syntax*. Ph.D. thesis, University of Pennsylvania.

Sarkar, Anoop and Aravind K. Joshi. 1996. Coordination in Tree Adjoining Grammars: Formalization and implementation. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING)*, pages 610–615.

Seddah, Djamé. 2008. The use of MCTAG to process elliptic coordination. In *Proceedings of The Ninth International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+9)*.

Shieber, Stuart M. and Yves Schabes. 1990. Synchronous Tree-Adjoining Grammars. In *Proceedings of the 13th International Conference on Computational Linguistics (COLING)*, pages 253–258.

Steedman, Mark. 1990. Gapping as constituent coordination. *Linguistics and Philosophy* 13:207–264.

Steedman, Mark. 1996. *Surface Structure and Interpretation*. No. 30 in Linguistic Inquiry Monograph. MIT Press.

Steedman, Mark. 2000. *The Syntactic Process*. MIT Press.

Steedman, Mark and Jason Baldridge. 2011. Combinatory Categorial Grammar. In R. Borsley and K. Borjars, eds., *Non-Transformational Syntax: Formal and Explicit Models of Grammar*. Wiley-Blackwell.

Tesnière, Lucien. 1959. *Éléments de syntaxe structurale*. Éditions Klincksieck.

van Genabith, Josef, Julia Hockenmaier, and Yusuke Miyao. 2006. Treebank-based acquisition of LFG, HPSG and CCG resources. ESSLI 2006 Advanced Course.

Wunderlich, Dieter. 1988. Some problems of coordination in German. In U. Reyle and C. Rohrer, eds., *Natural Language Parsing and Linguistic Theories*, pages 289–316. Reidel.