Chapter 3

Formal Background

Frank Richter

Goethe Universität Frankfurt

This chapter provides a very condensed introduction to a formalism for Pollard & Sag (1994) and explains its fundamental concepts. It pays special attention to the model-theoretic meaning of HPSG grammars. In addition it points out some links to other, related formalisms, such as feature logics of partial information, and to related terminology in the context of grammar implementation platforms.

1 Introduction

The two HPSG books by Pollard and Sag (Pollard & Sag 1987; 1994) do not present grammar formalisms with the intention to provide precise definitions. Instead they refer to various inspirations in the logics of typed feature structures or in predicate logic, informally characterize the intended formalisms, and explain them as they are used in concrete grammars of English. Pollard & Sag (1994) further clarifies their intentions in an appendix which lists most (but not all) of the components of their grammar of English explicitly, and summarizes most of their core assumptions. With this strategy, both books leave room for interpretation.

There are a number of challenges with reviewing the formal background of HPSG. Some of them have to do with the long or complicated publication history of relevant articles and books, some with the effects of grammar implementation platforms, which have their own underlying formalisms and influence the way in which linguists think and talk about their grammars with their own terminology and notational conventions. Salient examples include notations for phrase structure rules, the treatment of lexical representations or the lexicon, mechanisms for lexical rules, and notations for default values, among many other devices.

Frank Richter. 2019. Formal Background. In Stefan Müller, Anne Abeillé, Robert D. Borsley & Jean-Pierre Koenig (eds.), *Head-Driven Phrase Structure Grammar: The handbook*, 45–65. Berlin: Language Science Press. DOI:??

Many of these notations are well-known in the HPSG community. Even if they do not strictly belong to a formal syntax, there is usually a way to interpret their convenient notations in formalisms intended for, or at least compatible with, HPSG. However, their re-interpretation can deviate to a larger or lesser degree from what their users have in mind when they write grammars. These differences may be sometimes subtle and sometimes significant, but they entail that the meaning of the notations when adapted to a formalism is not what their users might have deduced from the behavior of a given implementation platform for parsing or generation. Similarly, terminology that belongs to the computational environment of implementations is often transferred to grammar theory, and again, a re-interpretation in terms of the underlying formalism can sometimes be trivial and sometimes nearly impossible, and different available choices of re-interpretations might have significantly different effects.

It is not only the flexibility of the informal AVM notation and its convenient notational enhancements (for lexical rules, decorated sort hierarchies, phrase structure trees, etc.) but also early changes in underlying foundational assumptions and terminology that one needs to be aware of when reviewing HPSG's formal background. When first presented in a book in 1987, HPSG was conceived of as a *unification-based* grammar theory, a name, the authors explain, which "arises from the algebra that governs partial information structures" (Pollard & Sag 1987: 7). This algebra was populated by partial feature structures with unification as a fundamental algebraic operation. In the framework envisioned seven years later in Pollard & Sag (1994), that algebra did not exist anymore, feature structures were no longer partial but total objects in models of a logical theory, and unification was no longer defined in the new setting (as the relevant algebra was gone). However, most of the notation and considerable portions of the terminology of 1987 remain with us to this day, such as the *types* of feature structures (for the *sorts* of 1994, when the term *type* was used for a different concept, to be discussed below), the pieces of information (for 1987-style feature structures) or even the word *unification*, which took on a casual life of its own without the underlying original algebra in which it had been defined. Occasionally these words still have a concrete technical interpretation in the language of grammar implementation environments or in their run-time system, which may reinforce their use in the community despite their lack of meaning in the standard formalism of HPSG. Implementation platforms may also add their own convenient technical and notational devices, implicitly inviting linguists to import them into their theoretical grammar writing.

This handbook article cannot disentangle the history and relationships between the various formalisms leading to an explication of the 1994 version of HPSG, nor of those that existed and still exist in parallel. It sets out to clarify the terminology and structure of a formalism for Pollard & Sag (1994) and presents a canonical formalism of the final version of HPSG in Pollard & Sag (1994). It will only occasionally point out some of the differences to its 1987 precursor where the older terminology is still present in current HPSG papers and may be confusing to an audience unaware of the different usages of terms which mean different things in different contexts.

The main sources of the present summary are the construction of a model theory for HPSG by King (1999) and Pollard (1999), and their synoptic reconstruction on the basis of a comprehensive logical language for HPSG, *Relational Speciate Re-entrant Language* (RSRL) by Richter (2004), including the critique and extensions sketched in Richter (2007).

2 Signatures and descriptions

As logical theories of entities in a domain of objects, HPSG grammars consist of two main components. First, a logical signature, which provides the symbols for describing the domain of interest, in this case a natural language. And second, an exact delineation of all and only the legitimate entities in the denotation of the grammar, written as a collection of statements about their configuration. These statements are descriptions within a logical language and are composed from logical constants, variables, quantifiers, brackets and the symbols provided by the signature. They are variously known to linguists as principles of grammar, constraints, or rules. In the following, I will use the term *principles* to designate these statements. As we will see, they must be of a certain form in order to do the job they are meant to perform, and linguists often use abbreviatory conventions for conceptually distinguished groups of principles, such as grammar rules, lexical entries, or lexical rules. From a logical perspective, then, a grammar is a pair consisting of a signature and a collection of principles. The appendix of Pollard & Sag (1994) provides an early example in HPSG of this conception.

Signatures in HPSG go beyond supplying non-logical symbols for descriptions, they impose additional restrictions on the organization of the non-logical symbols. These restrictions ultimately have an effect on how the domain of described objects is structured. Let us first investigate the two most prominent sets of non-logical symbols, sorts and attributes. The set of *sort* symbols is arranged in a *sort hierarchy*, and that sort hierarchy is in turn connected to the set of *attribute* symbols (also known as *features*). The sort hierarchy is a partial order, and attributes def. partial order?

are declared *appropriate to* sorts in the sort hierarchy. This appropriateness declaration must not be entirely random: if an attribute is declared appropriate to some sort, it must also be declared appropriate to all its subsorts. This requirement is known as *feature inheritance.* Moreover, for each sort σ and attribute ϕ such that ϕ is appropriate to σ, some other sort σ' is *appropriate for* ϕ *at* σ. In other words, a certain attribute value (σ') is declared appropriate for ϕ at σ. These attribute values must not be completely random either: For any subsort of σ, the value of an appropriate feature ϕ of σ is of course also appropriate to that subsort (by feature inheritance), but in addition, the value of ϕ at that subsort must be at least as specific as it is at σ. This means the value is either σ' or a subsort thereof. It may not be less specific, or, to put it differently, it may not be a supersort of σ'.

pictures?, incl. to vs for (JP)

Some sorts in the sort hierarchy enjoy a special status by being *maximally specific.* They are called *species.* Species are sorts without subsorts. Sorts that are maximally specific and lack any appropriate attribute receive a special name and are called *atomic* sorts or simply *atoms.*

In addition to sorts and attributes, a signature provides relation symbols. Well known examples are a ternary `append` relation and a binary `member` relation, but grammars may also require relations such as (often ternary) `shuffle` and binary `o-command`. Each relation symbol comes with a positive natural number for the number of arguments, its *arity.*

Putting all of this together, we obtain a definition of signatures:

Definition 1 Σ *is a* signature *iff*
Σ *is a septuple* $\langle S, \sqsubseteq, S_{max}, A, F, R, Ar\rangle$,
$\langle S, \sqsubseteq\rangle$ *is a partial order,*
$S_{max} = \{\sigma \in S \mid \textit{for each } \sigma' \in S, \textit{if } \sigma' \sqsubseteq \sigma \textit{ then } \sigma = \sigma'\}$,
A *is a set,*
F *is a partial function from* $S \times A$ *to* S,
for each $\sigma_1 \in S$, *for each* $\sigma_2 \in S$, *for each* $\phi \in A$,
  if $F(\sigma_1, \phi)$ *is defined and* $\sigma_2 \sqsubseteq \sigma_1$
  then $F(\sigma_2, \phi)$ *is defined and* $F(\sigma_2, \phi) \sqsubseteq F(\sigma_1, \phi)$,
R *is a finite set, and*
Ar *is a total function from* R *to the positive integers.*

The partial order $\langle S, \sqsubseteq\rangle$ is the sort hierarchy, and the set of sorts S, just like the set of attributes A, can in principle be infinite. In actual grammars it is finite, and in HPSG grammars it is also assumed that S contains a top element, which is a sort that subsumes all other sorts in the sort hierarchy. S_{max} is the set of maximally specific sorts, which will play a prominent role in the semantics of descriptions. F is a function for fixing the appropriateness conditions on attributes and attribute values, and the conditions on that function reflect HPSG's restrictions on feature declarations. F is called the *(feature) appropriateness function*. The last two lines of the definition provide the set of relation symbols with their arity. We assume that relations are at least unary.

Relations in HPSG often express relationships between lists (`append`, `shuffle`) or sets (`union`, `intersection`). In order to give grammarians the flexibility that some of the applications of these relations in the literature starting with Pollard & Sag (1994) require, RSRL adds dedicated sorts and attributes with a fixed interpretation to every signature. They can be thought of as a more flexible treatment of lists along their regular explicit encoding in HPSG (usually with attributes FIRST and REST, and sorts *list*, *elist*, and *nelist*, but of course the exact naming does not matter). Informally, the extra symbols act very much like sorts and attributes for lists. In order to integrate the reserved new sort symbols with any signature a linguist might specify, a distinguished sort *metatop* serves as unique top element of the extended signature. The extension is defined for any signature by adding reserved *pseudo-sorts* and *pseudo-attributes* and structuring the expanded sort hierarchy in the desired way:

add picture (Stefan)

Definition 2 *For each signature* $\Sigma = \langle S, \sqsubseteq, S_{max}, A, F, R, Ar \rangle$,
$\widehat{S} = S \cup \{chain, echain, nechain, metatop\}$,
$\widehat{\sqsubseteq} = \sqsubseteq \cup \{\langle echain, chain \rangle, \langle nechain, chain \rangle\} \cup \left\{ \langle \sigma, \sigma \rangle \mid \sigma \in \widehat{S} \backslash S \right\} \cup \left\{ \langle \sigma, metatop \rangle \mid \sigma \in \widehat{S} \right\}$,
$\widehat{S}_{max} = S_{max} \cup \{echain, nechain\}$, *and*
$\widehat{A} = A \cup \{\dagger, \triangleright\}$.

Apart from the non-logical constants from (expanded) signatures and some logical symbols, we also need a countably infinite set of variables, which will be symbolized by V.

For expository reasons the syntax of descriptions, to be introduced next, does not employ attribute-value matrices (AVMs), the common lingua franca of constraint-based grammar formalisms. The reasons are twofold: Most importantly, although AVMs provide an extremely readable and flexible notation, they are quite cumbersome to define as a rigorous logical language which meets all the expressive needs of HPSG. Some of this awkwardness in explicit definitions derives from the very flexibility and redundancy in notation that makes AVMs perfect for everyday linguistic practice. Second, the original syntax of (R)SRL is, by contrast, extremely easy to define, and, as long as it is not used for descriptions as complex as they occur in real grammars, its expressions are still transparent for everyone who is familiar with AVMs. Readers who want to explore how our description syntax relates to a formal syntax of AVMs are referred to Richter (2004) for details and a correspondence proof.

The definition of the syntax of descriptions proceeds in two steps, quite similar to first-order predicate logic. We will first introduce terms and then build formulae and descriptions from terms. Terms are essentially what is known to linguists as *paths*, sequences of attributes:

Definition 3 *For each signature* $\Sigma = \langle S, \sqsubseteq, S_{max}, A, F, R, Ar \rangle$, T^{Σ} *is the smallest set such that*
$: \in T^{\Sigma}$,
for each $v \in V$, $v \in T^{\Sigma}$,
for each $\phi \in \widehat{A}$ *and each* $\tau \in T^{\Sigma}$, $\tau\phi \in T^{\Sigma}$.

Simply put, sequences of attributes (including the two pseudo-attributes † and ▹) starting either with the colon or a single variable are Σ terms. Equipped with terms, we can immediately proceed to formulae, the penultimate step on the way to descriptions. There are three kinds of simple formulae: Formulae that assign a sort to the value of a path, formulae which state that two paths have the same value (*structure sharing*, in linguistic terminology), and relational formulae. Complex formulae can be built from these by existential and universal quantification, negation, and the classical binary logical connectives.

Definition 4 *For each signature* $\Sigma = \langle S, \sqsubseteq, S_{max}, A, F, R, Ar \rangle$, D^{Σ} *is the smallest set such that*
for each $\sigma \in \widehat{S}$, *for each* $\tau \in T^{\Sigma}$, $\tau \sim \sigma \in D^{\Sigma}$,
for each $\tau_1, \tau_2 \in T^{\Sigma}$, $\tau_1 \approx \tau_2 \in D^{\Sigma}$,
for each $\rho \in R$, *for each* $x_1, \ldots, x_{Ar(\rho)} \in V$, $\rho(x_1, \ldots, x_{Ar(\rho)}) \in D^{\Sigma}$,
for each $x \in V$, *for each* $\delta \in D^{\Sigma}$, $\exists x \delta \in D^{\Sigma}$, *(analogous for* $\forall$*)*
for each $\delta \in D^{\Sigma}$, $\neg\delta \in D^{\Sigma}$,
for each $\delta_1, \delta_2 \in D^{\Sigma}$, *and* $(\delta_1 \wedge \delta_2) \in D^{\Sigma}$. *(analogous for* $\vee, \rightarrow, \leftrightarrow$*)*

Finally, *FV* is a function that determines for every Σ term and Σ formula the set of variables that occur free in them.

Definition 5 *For each signature* $\Sigma = \langle S, \sqsubseteq, S_{max}, A, F, R, Ar \rangle$,
$FV(:) = \{\}$,
for each $x \in V$, $FV(x) = \{x\}$,
for each $\tau \in T^{\Sigma}$, *for each* $\phi \in \widehat{A}$, $FV(\tau\phi) = FV(\tau)$,

for each $\tau \in T^{\Sigma}$, *for each* $\sigma \in \widehat{S}$, $FV(\tau \sim \sigma) = FV(\tau)$,
for each $\tau_1, \tau_2 \in T^{\Sigma}$, $FV(\tau_1 \approx \tau_2) = FV(\tau_1) \cup FV(\tau_2)$,
for each $\rho \in R$, *for each* $x_1, \ldots, x_{Ar(\rho)} \in V$, $FV(\rho(x_1, \ldots, x_{Ar(\rho)})) = \{x_1, \ldots, x_{Ar(\rho)}\}$,
for each $\delta \in D^{\Sigma}$, *for each* $x \in V$, $FV(\exists x \delta) = FV(\delta) \setminus \{x\}$, *(analogous for* $\forall$*)*,
for each $\delta \in D^{\Sigma}$, $FV(\neg\delta) = FV(\delta)$,
for each $\delta_1, \delta_2 \in D^{\Sigma}$, $FV((\delta_1 \wedge \delta_2)) = FV(\delta_1) \cup FV(\delta_2)$. *(analogous for* $\vee, \rightarrow, \leftrightarrow$*)*

Informally, an occurrence of a variable is free in a Σ term or a Σ formula if it is not bound by a quantifier. We single out Σ formulae without free occurrences of variables as a kind of formula of special interest and reserve the term Σ *description* for them:

Definition 6 *For each signature* Σ, $D_0^{\Sigma} = \left\{\delta \in D^{\Sigma} | FV(\delta) = \{\}\right\}$.

D_0^{Σ} is the set of Σ descriptions. When a signature is fixed by the context, or when the exact signature is irrelevant in the discussion, we can simply speak of *descriptions* instead of Σ descriptions. Descriptions are the syntactic units that linguists use in grammar writing. Grammars, as we will see in Section 4, are written by declaring a signature and stating a set of descriptions. But before we can investigate grammars and what they mean, we have to explain the meaning of signatures and of descriptions.

3 Meaning of signatures and descriptions

Descriptions of RSRL are interpreted similar to expressions of classical logics such as first order logic, except that they are not evaluated as true or false in a given structure; instead, they denote collections of structures.

Defining the meaning of descriptions begins with delineating the structures which interpret signatures. In particular, species and attributes must receive a meaning, which should be tied to the HPSG-specific intentions behind sort hierarchies and feature declarations; and so must relation symbols, whose interpretation should heed their arity. Due to some extra restrictions which will ultimately be put on the interpretation of relation symbols (to meet intuitions of grammarians) and whose formulation requires a notion of term interpretation, we start with *initial interpretations*. They will be refined in a second step to full interpretations (Definition 13).

Some additional notation is needed in the next definition. If S is a set, S^* is the set of all finite sequences (or n-tuples) of elements of S. S^+ is the same set without the empty sequence. $\overline{S}$ is short for the set $S \cup S^*$.

Definition 7 *For each signature* $\Sigma = \langle S, \sqsubseteq, S_{max}, A, F, R, Ar \rangle$, I *is an* initial Σ interpretation *iff*
$\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$,
U *is a set,*
S *is a total function from* U *to* S_{max},
A *is a total function from* A *to the set of partial functions from* U *to* U,
for each $\phi \in A$ *and each* $u \in \mathsf{U}$
if $\mathsf{A}(\phi)(u)$ *is defined*
then $F(\mathsf{S}(u), \phi)$ *is defined, and* $\mathsf{S}(\mathsf{A}(\phi)(u)) \sqsubseteq F(\mathsf{S}(u), \phi)$, *and*
for each $\phi \in A$ *and each* $u \in \mathsf{U}$,
if $F(\mathsf{S}(u), \phi)$ *is defined then* $\mathsf{A}(\phi)(u)$ *is defined,*
R *is a total function from* R *to the power set of* $\bigcup_{n \in \mathbb{N}} \overline{\mathsf{U}}^n$, *and*
for each $\rho \in R$, $\mathsf{R}(\rho) \subseteq \overline{\mathsf{U}}^{Ar(\rho)}$.

Initial Σ interpretations are quadruples consisting of four components. The first three of them will remain unchanged in full Σ interpretations (Definition 13). The elements of U are entities which populate the universe of structures. Their ontological status has been debated fiercely in HPSG, and will be discussed in Sections 4–5. For the moment, assume that they are either linguistic objects or appropriate abstractions thereof. S assigns each object in the universe a species, which is another way of saying that each object is of exactly one maximally specific sort. This is what is known as the property of being *sort-resolved*. The attribute interpretation function A interprets each attribute symbol as a (partial) function that assigns an object of the universe to an object of the universe, and as such it obeys the restrictions of the feature declarations of the signature, embodied in the function F: Attributes are defined on all and only those objects u_1 which have a species to which the attributes are appropriate according to F; and the object which u_1 is mapped to by the attribute must in turn be of a species which is appropriate for the attribute (at the species of u_1). This is what is known as the property of interpreting structures of being *totally well-typed*. Originally both of these properties of interpreting structures were formulated with respect to so-called *feature structures*, but, as we will see below, this conception of interpreting structures for grammars was soon given up for philosophical reasons.[1] The relation interpretation function R finally interprets n-ary relation symbols as sets of n-tuples of objects. However, there is an additional option, which makes the definition look more complex: An object in an n-tuple may in fact not be an

[1]Of course, the informal term *feature structure* is still alive among linguists, and in a technical sense they are essential constructs for implementation platforms.

atomic object, it can alternatively be an n-tuple of objects itself. These n-tuples in argument positions of relations will be described as *chains* with the extra symbols, pseudo-sort and pseudo-attributes, which were added to signatures in Definition 2 above. As pointed out there, chains are additional constructs which give grammarians the flexibility to use (finite) lists in all the ways in which they are put in relations in actual HPSG grammars.

Since chains are provided by an extension of the set of sort symbols and attributes (Definition 2), the interpretation of the additional symbols must be defined separately. This is very simple, since these symbols behave essentially analogous to the conventional sort and attribute symbols of HPSG's list encoding.

Definition 8 *For each signature* $\Sigma = \langle S, \sqsubseteq, S_{max}, A, F, R, Ar \rangle$, *for each initial* Σ *interpretation* $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$,
$\widehat{\mathsf{S}}$ *is the total function from* $\overline{\mathsf{U}}$ *to* $\widehat{S}$ *such that*
for each $u \in \mathsf{U}$, $\widehat{\mathsf{S}}(u) = \mathsf{S}(u)$,
for each $u_1, \ldots, u_n \in \mathsf{U}$, $\widehat{\mathsf{S}}(\langle u_1, \ldots, u_n \rangle) = \begin{cases} echain & \text{if } n = 0, \\ nechain & \text{if } n > 0 \end{cases}$, *and*
$\widehat{\mathsf{A}}$ *is the total function from* $\widehat{A}$ *to the set of partial functions from* $\overline{\mathsf{U}}$ *to* $\overline{\mathsf{U}}$ *such that*
for each $\phi \in A$, $\widehat{\mathsf{A}}(\phi) = \mathsf{A}(\phi)$,
$\widehat{\mathsf{A}}(\dagger)$ *is the total function from* U^+ *to* U *such that for each* $\langle u_0, \ldots, u_n \rangle \in \mathsf{U}^+$,
$\widehat{\mathsf{A}}(\dagger)(\langle u_0, \ldots, u_n \rangle) = u_0$, *and*
$\widehat{\mathsf{A}}(\triangleright)$ *is the total function from* U^+ *to* U^* *such that for each* $\langle u_0, \ldots, u_n \rangle \in \mathsf{U}^+$,
$\widehat{\mathsf{A}}(\triangleright)(\langle u_0, \ldots, u_n \rangle) = \langle u_1, \ldots, u_n \rangle$.

$\widehat{S}$ is the *expanded species assignment function*, and $\widehat{A}$ is the *expanded attribute interpretation function*. The pseudo-species symbols *echain* and *nechain* label empty chains and non-empty chains, respectively. Given a non-empty chain, the pseudo-attribute † picks out its first member, corresponding to the function of the FIRST attribute on non-empty lists. Conversely, ▹ cuts off the first element of a non-empty chain and returns the remainder of the chain, as does the standard attribute REST for lists.

In addition to attributes, terms may also contain variables. Term interpretation thus requires a notion of *variable assignments* in (initial) interpretations.

Definition 9 *For each signature* Σ, *for each initial* Σ *interpretation* $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$, $\mathsf{G}_\mathsf{I} = \overline{\mathsf{U}}^V$ *is the smallest set of variable assignments in* I.

An element of G_I (the set of total functions from the set of variables to the set of objects and chains of objects of U) will be notated as g, following a convention frequently observed in predicate logic. With variable assignments in (initial) interpretations, variables denote objects in the universe U and chains of objects of the universe.

Definition 10 *For each signature* $\Sigma = \langle S, \sqsubseteq, S_{max}, A, F, R, Ar\rangle$, *for each initial* Σ *interpretation* $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R}\rangle$, *for each* $g \in \mathsf{G}_\mathsf{I}$, T^g_I *is the total function from* T^Σ *to the smallest set of partial functions from* U *to* $\overline{\mathsf{U}}$ *such that for each* $u \in \mathsf{U}$,

$\mathsf{T}^g_\mathsf{I}(:)(u)$ *is defined and* $\mathsf{T}^g_\mathsf{I}(:)(u) = u$,

for each $v \in V$, $\mathsf{T}^g_\mathsf{I}(v)(u)$ *is defined and* $\mathsf{T}^g_\mathsf{I}(v)(u) = g(u)$,

for each $\tau \in T^\Sigma$, *for each* $\phi \in \widehat{A}$,

$\mathsf{T}^g_\mathsf{I}(\tau\phi)(u)$ *is defined iff* $\mathsf{T}^g_\mathsf{I}(\tau)(u)$ *is defined and* $\widehat{\mathsf{A}}(\phi)(\mathsf{T}^g_\mathsf{I}(\tau)(u))$ *is defined, and if* $\mathsf{T}^g_\mathsf{I}(\tau\phi)(u)$ *is defined then* $\mathsf{T}^g_\mathsf{I}(\tau\phi)(u) = \widehat{\mathsf{A}}(\phi)(\mathsf{T}^g_\mathsf{I}(\tau)(u))$.

T^g_I is called the *term interpretation function under* I *under* g. Σ terms either start with a variable or with the special symbol colon (':'). The colon denotes the identity function. Interpreted on any object, it returns that object. If a term τ starts with the colon, its term interpretation so to speak starts at the object u to which it is applied ($\mathsf{T}^g_\mathsf{I}(\tau)(u)$) and, if each subsequent attribute in τ is defined on the object to which the interpretation of the earlier attribute(s) took us, the term interpretation will yield the object reached by the last attribute. When a Σ term starts with a variable v, the given variable assignment g will determine the starting point of interpreting the sequence of attributes ($g(v)$). Of course, variables may be assigned chains of objects, in which case the symbols of the expanded attribute set can be used to navigate the elements of the chain.

The set of objects which are reachable from a single given object in an interpretation by following sequences of attribute interpretations is important for the way in which quantification is conceived by grammarians, it plays a role in thinking about which objects can in principle stand in a relation, and it is crucial for explicating different notions of the meaning of grammars. Definition 11 captures this idea:

Definition 11 *For each signature* $\Sigma = \langle S, \sqsubseteq, S_{max}, A, F, R, Ar\rangle$, *for each initial* Σ *interpretation* $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R}\rangle$, *for each* $u \in \mathsf{U}$,

$$\mathsf{C}^u_\mathsf{I} = \left\{ u' \in \mathsf{U} \;\middle|\; \begin{array}{l} \textit{for some } g \in \mathsf{G}_\mathsf{I}, \\ \textit{for some } \pi \in A^*, \\ \quad \mathsf{T}^g_\mathsf{I}(:\pi)(u) \textit{ is defined, and} \\ \quad u' = \mathsf{T}^g_\mathsf{I}(:\pi)(u) \end{array} \right\}.$$

C^u_I is the set of components of u in I. The purpose of C^u_I is to capture the set of all objects that are reachable from some object u in the universe by following a path of interpreted attributes. Thinking of these configurations as graphs, the set of components of u in I is the set of nodes that can be reached by following any sequence of arrows starting from u. This corresponds to how linguists normally conceive of the substructures of some structured object.[2] The set of components of objects is used in two ways in the definitions of full interpretations and description denotations: It restricts the set of objects that are permitted in relations, and it provides the domain of quantification in quantificational expressions of the logical language.

According to Definition 7 of initial interpretations, relation symbols are simply interpreted as tuples of objects (and chains of objects) in the universe of interpretation. However, HPSGians have a slightly more restricted notion of relations: For them, relations hold between objects that occur within a sign (or a similar kind of larger linguistic structure), they are not relations between objects that occur in separate (unconnected) signs. The following notion of *possible relation tuples in an interpretation* captures this intuition.

Definition 12 *For each signature* $\Sigma = \langle S, \sqsubseteq, S_{max}, A, F, R, Ar \rangle$, *for each initial* Σ *interpretation* $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$,

$$\mathsf{RT}_\mathsf{I} = \bigcup_{n \in \mathbb{N}} \left\{ \langle u_1, \ldots, u_n \rangle \in \overline{\mathsf{U}}^n \;\middle|\; \begin{array}{l} \textit{for some } u \in \mathsf{U}, \\ \textit{for each } i \in \mathbb{N}, 1 \leq i \leq n \\ \quad u_i \in \overline{\mathsf{C}_\mathsf{I}^u} \end{array} \right\}.$$

RT_I is the set of possible relation tuples in I. Possible relation tuples in an initial interpretation are characterized by the existence of some object in the interpretation from which each object in a relation tuple can be reached by a sequence of attribute interpretations. In case an argument in a tuple is a chain, then the objects on the chain are thus restricted.

The notion of *full interpretations* integrates this restriction on possible relations, keeping everything else unchanged from initial interpretations:

Definition 13 *For each signature* $\Sigma = \langle S, \sqsubseteq, S_{max}, A, F, R, Ar \rangle$, *for each initial* Σ *interpretation* $\mathsf{I}' = \langle \mathsf{U}', \mathsf{S}', \mathsf{A}', \mathsf{R}' \rangle$, *for the set of possible relation tuples in* I', $\mathsf{RT}_{\mathsf{I}'}$, $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$ *is a full* Σ *interpretation iff*
$\mathsf{U} = \mathsf{U}'$, $\mathsf{S} = \mathsf{S}'$, $\mathsf{A} = \mathsf{A}'$, *and* R *is a total function from* R *to the power set of* $\mathsf{RT}_{\mathsf{I}'}$, *and for each* $\rho \in R$, $\mathsf{R}(\rho) \subseteq \left(\mathsf{RT}_{\mathsf{I}'} \cap \overline{\mathsf{U}}^{Ar(\rho)} \right)$.

[2] Phrasing this more carefully, the object itself is not structured, but there is a structure generated by the object by following the arrows, or more technically, by the composition of functions which interpret attribute symbols.

It can be checked that variable assignments in initial interpretations and sets of components of objects in initial interpretations are the same as in corresponding full interpretations with the same universe, species interpretation and attribute interpretation functions, since variable assignments and sets of components of objects do not depend on the interpretation of relations. From now on all of the above will be used with respect to full interpretations, and full interpretations will simply be called interpretations.

The following definition of Σ formula denotation needs a notation for modifying variable assignments with respect to the value of designated variables. For any variable assignment $g \in \mathsf{G_I}$, for $g' = g[x \mapsto u]$, g' is just like g except that g' maps variable x to object u.

Definition 14 *For each signature $\Sigma = \langle S, \sqsubseteq, S_{max}, A, F, R, Ar\rangle$, for each (full) Σ interpretation $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R}\rangle$, for each $g \in \mathsf{G_I}$, D^g_I is the total function from D^Σ to the power set of U such that*

$$\textit{for each } \tau \in T^\Sigma, \textit{for each } \sigma \in \widehat{S}, \mathsf{D}^g_\mathsf{I}(\tau \sim \sigma) = \left\{ u \in \mathsf{U} \;\middle|\; \begin{array}{l} \mathsf{T}^g_\mathsf{I}(\tau)(u) \textit{ is defined, and} \\ \widehat{\mathsf{S}}\left(\mathsf{T}^g_\mathsf{I}(\tau)(u)\right) \mathrel{\widehat{\sqsubseteq}} \sigma \end{array} \right\},$$

$$\textit{for each } \tau_1, \tau_2 \in T^\Sigma, \mathsf{D}^g_\mathsf{I}(\tau_1 \approx \tau_2) = \left\{ u \in \mathsf{U} \;\middle|\; \begin{array}{l} \mathsf{T}^g_\mathsf{I}(\tau_1)(u) \textit{ is defined,} \\ \mathsf{T}^g_\mathsf{I}(\tau_2)(u) \textit{ is defined, and} \\ \mathsf{T}^g_\mathsf{I}(\tau_1)(u) = \mathsf{T}^g_\mathsf{I}(\tau_2)(u) \end{array} \right\},$$

$$\textit{for each } \rho \in R, \textit{for each } x_1, \ldots, x_{Ar(\rho)} \in V,$$
$$\mathsf{D}^g_\mathsf{I}\left(\rho(x_1, \ldots, x_{Ar(\rho)})\right) = \left\{ u \in \mathsf{U} \;\middle|\; \left\langle g(x_1), \ldots, g(x_{Ar(\rho)})\right\rangle \in \mathsf{R}(\rho) \right\},$$

$$\textit{for each } x \in V, \textit{for each } \delta \in D^\Sigma, \mathsf{D}^g_\mathsf{I}(\exists x \delta) = \left\{ u \in \mathsf{U} \;\middle|\; \begin{array}{l} \textit{for some } u' \in \overline{\mathsf{C}^u_\mathsf{I}} \\ u \in \mathsf{D}^{g[x \mapsto u']}_\mathsf{I}(\delta) \end{array} \right\},$$

$$\textit{for each } x \in V, \textit{for each } \delta \in D^\Sigma, \mathsf{D}^g_\mathsf{I}(\forall x \delta) = \left\{ u \in \mathsf{U} \;\middle|\; \begin{array}{l} \textit{for each } u' \in \overline{\mathsf{C}^u_\mathsf{I}} \\ u \in \mathsf{D}^{g[x \mapsto u']}_\mathsf{I}(\delta) \end{array} \right\},$$

$$\textit{for each } \delta \in D^\Sigma, \mathsf{D}^g_\mathsf{I}(\neg\delta) = \mathsf{U} \backslash \mathsf{D}^g_\mathsf{I}(\delta),$$
$$\textit{for each } \delta_1, \delta_2 \in D^\Sigma, \mathsf{D}^g_\mathsf{I}((\delta_1 \wedge \delta_2)) = \mathsf{D}^g_\mathsf{I}(\delta_1) \cap \mathsf{D}^g_\mathsf{I}(\delta_2)$$
$$\textit{for each } \delta_1, \delta_2 \in D^\Sigma, \mathsf{D}^g_\mathsf{I}((\delta_1 \vee \delta_2)) = \mathsf{D}^g_\mathsf{I}(\delta_1) \cup \mathsf{D}^g_\mathsf{I}(\delta_2)$$
$$\textit{for each } \delta_1, \delta_2 \in D^\Sigma, \mathsf{D}^g_\mathsf{I}((\delta_1 \rightarrow \delta_2)) = \left(\mathsf{U} \backslash \mathsf{D}^g_\mathsf{I}(\delta_1)\right) \cup \mathsf{D}^g_\mathsf{I}(\delta_2), \textit{ and}$$
$$\textit{for each } \delta_1, \delta_2 \in D^\Sigma,$$
$$\mathsf{D}^g_\mathsf{I}((\delta_1 \leftrightarrow \delta_2)) = ((\mathsf{U} \backslash \mathsf{D}^g_\mathsf{I}(\delta_1)) \cap (\mathsf{U} \backslash \mathsf{D}^g_\mathsf{I}(\delta_2))) \cup (\mathsf{D}^g_\mathsf{I}(\delta_1) \cap \mathsf{D}^g_\mathsf{I}(\delta_2)).$$

D^g_I is the Σ *formula interpretation function with respect to* I *under a variable assignment, g, in* I. Sort assignment formulae, $\tau \sim \sigma$, denote sets of objects on which the attribute path τ is defined and leads to an object u' of sort σ. If σ is not a species, the object u' must be of a maximally specific subsort of σ. Path equations of the form $\tau_1 \approx \tau_2$ hold of an object u when path τ_1 and path τ_2 lead to the same object u'. And an n-ary relational formula $\rho(x_1, \ldots, x_n)$ denotes a set of objects such that the n-tuples of objects (or chains of objects) assigned to the variables x_1 to x_n are in the denotation of the relation ρ. This means that a relational formula either denotes the entire universe U or the empty set, depending on the variable assignment g in I. Applying the idea behind this definition of the interpretation of relational formulae to a typical HPSG grammar, each well formed phrase described by an HPSG grammar is such that append holds of exactly those triples of lists (and chains) in that phrase of which it is supposed to hold according to the grammarian's definition of append. This guarantees that the use of append in grammar principles has the effect it is supposed to have. How exactly it is achieved will become clear when we introduce the meaning of grammars.

Negation is interpreted as set complement of the denotation of a formula, conjunction and disjunction of formulae as set intersection and set union of the denotation of two formulae, respectively. The meaning of implication and bi-implication follows the pattern of classical logic and could alternatively be defined on the basis of negation and disjunction (or conjunction) alone. Quantificational expressions are special in that they implement the idea of restricted quantification by referring to the set of components of objects in I. An existentially quantified formula, $\exists x \delta$, denotes the set of objects u such that there is at least one component (or chain of components) u' of u, and interpreting x as u' leads to δ describing u. With universal quantification, the corresponding condition must hold for *all* components (or chains of components) of the objects u in the denotation of the quantified formula. Again turning to the application of these definitions of formula denotations in grammar writing, the intuition is that linguists quantify over the components of grammatical structures (sentences, phrases), and not over a universe of objects that may include unrelated sentences and grammatical structures, or components thereof: a certain kind of object exists within a given structure, or all objects in a certain structure fulfill certain conditions.

A standard proof shows that the denotation of Σ formulae without free occurrences of variables, i.e. the denotation of Σ descriptions, is independent of the initial choice of variable assignment. For Σ descriptions we can thus define a simpler Σ *description denotation function with respect to an interpretation* I, D_I:

Definition 15 *For each signature* $\Sigma = \langle S, \sqsubseteq, S_{max}, A, F, R, Ar \rangle$, *for each (full)* Σ *interpretation* $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$, D_I *is the total function from* D_0^Σ *to the power set of* U *such that* $\mathsf{D}_\mathsf{I}(\delta) = \left\{ u \in \mathsf{U} \,\middle|\, \text{for each } g \in \mathsf{G}_\mathsf{I}, u \in \mathsf{D}_\mathsf{I}^g \right\}$.

For each description δ, D_I returns the set of objects in the universe of I that are described by δ. With Σ descriptions and their denotation as sets of objects we have everything in place to symbolize all grammar principles of a grammar such as the one presented by Pollard & Sag (1994) in logical notation and to receive and interpretation as intended by the authors. A comprehensive logical rendering of their grammar of English can be found in Appendix C of Richter (2004). Moreover, as shown there, all syntactic constructs of the logical languages above are necessary to achieve that goal without reformulating the grammar.

4 Meaning of grammars

Grammars comprise sets of descriptions, the principles of grammar. These sets of principles are often called *theories* in the context of logical languages for HPSG, although this terminology can occasionally be confusing.[3] Theories, i.e. sets of descriptions, are symbolized with θ. A grammar is simply a theory together with a signature:

Definition 16 Γ *is a* grammar *iff*
Γ *is a pair* $\langle \Sigma, \theta \rangle$, *where* Σ *is a signature, and* $\theta \subseteq D_0^\Sigma$.

Essentially, theories denote the conjunction of their descriptions, except that theories can, in principle (and contrary to deliberate linguistic convention), be infinite:

Definition 17 *For each signature* Σ, *for each* Σ *interpretation* $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$, Θ_I *is the total function from the power set of* D_0^Σ *to the power set of* U *such that for each* $\theta \subseteq D_0^\Sigma$,
$\Theta_\mathsf{I}(\theta) = \left\{ u \in \mathsf{U} \,\middle|\, \textit{for each } \delta \in \theta, u \in \mathsf{D}_\mathsf{I}(\delta) \right\}$.

Θ_I is the *theory denotation function with respect to* I. A theory consisting of a set of descriptions holds of every object u in the universe exactly if every description in the theory holds of u. In short, a theory denotes the set of objects that are described by everything in the theory. They do not violate any restriction the theory expresses in one of its descriptions.

A first approximation to the meaning of grammars is provided by the notion of a Γ model, a model of a grammar Γ:

[3] The problem with this term is that it can be argued that theories, defined this way, do not constitute what would traditionally be called a *theory of a language*, since many central aspects of a theory in the latter sense are not embodied in that kind of formalized theory.

Definition 18 *For each grammar* $\Gamma = \langle \Sigma, \theta \rangle$*, for each* Σ *interpretation* $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$*,*
I *is a* Γ model *iff* $\Theta_{\mathsf{I}}(\theta) = \mathsf{U}$.

A Γ model is an interpretation $\mathsf{I} = \langle \mathsf{U}, \mathsf{S}, \mathsf{A}, \mathsf{R} \rangle$ in which every description in the theory of grammar Γ is true of every object in the interpretation's universe U. In other words, each object in the interpretation fulfills all conditions which are imposed by the grammar principles. There is no object in a Γ model that violates any principle.

Linguists use grammars to make predictions about the grammatical structures of languages. In classical generative terminology, a grammar undergenerates if there are grammatical structures it does not capture. It overgenerates if it permits structures that are deemed ungrammatical. It is uncontroversial that an appropriate notion of the meaning of a grammar should support linguists in making such predictions with their grammars. However, the previous notion of Γ models is not strong enough for this purpose. To see this, suppose there is a signature Σ which is fit to describe the entire English language, and there is a theory θ which expresses correctly all and only what there is to say about English. A $\langle \Sigma, \theta \rangle$ model may then consist of a structure of the single sentence *Elon is frying fish.* This follows from the definition of Γ models because any appropriate grammar of English describes all objects of which this sentence consists. But this one-sentence-model is obviously too small for being a good candidate for the English language, because English contains much more than this single sentence. In arbitrary models it cannot be seen if a grammar undergenerates or overgenerates.

King (1999)'s *exhaustive models* are a possibility to define the meaning of grammars in such a way that they reflect the basic expectations of generative linguists. The underlying intuition is to choose a maximal model which contains a congruent copy of any configuration of objects which can be found in some model of the grammar.

Definition 19 *For each grammar* $\Gamma = \langle \Sigma, \theta \rangle$*, for each* Σ *interpretation* I,
I *is an* exhaustive Γ model *iff*
I *is a* Γ *model, and*
for each $\theta' \subseteq D_0^{\Sigma}$*, for each* Σ *interpretation* I'*,*
if I' *is a* Γ *model and* $\Theta_{\mathsf{I}'}(\theta') \neq \emptyset$ *then* $\Theta_{\mathsf{I}}(\theta') \neq \emptyset$*.*

Any grammar with a non-empty model also has a non-empty exhaustive model. In addition to being a model of a given grammar $\Gamma = \langle \Sigma, \theta \rangle$, an exhaustive Γ model I has the property that each arbitrarily chosen set of descriptions θ' which denotes anything at all in any Γ model also denotes something in I. An alternative algebraic way to characterize this requirement is to say that any configuration of objects in any Γ model has a congruent counterpart in an exhaustive Γ model. At the same time, since an exhaustive model is from a special class of *models*, if a description in θ does not describe some object in a Γ interpretation I', then this object in I' cannot have a counterpart in an exhaustive Γ model.

This is sufficient to capture relevant grammar-theoretic notions of linguistics: A grammar Γ of a language $\mathcal{L}$ overgenerates iff an exhaustive Γ model contains configurations that are not (congruent to) grammatical expressions in $\mathcal{L}$; it undergenerates iff an exhaustive Γ model does not contain configurations which are (congruent to) grammatical expressions in $\mathcal{L}$.

5 Alternative conceptions of the meaning of grammars

The exhaustive models of Section 4 provide a first solid notion of the meaning of HPSG grammars, but they adopt a very particular conceptualization of the ontological status of the structures in the denotation of grammars. This section outlines three additional alternative ways to define the meaning of HPSG grammars with different foundational assumptions.

The theory of exhaustive models from the previous section decidedly supposes that a grammar denotes a *token* model of a language $\mathcal{L}$. According to this theory, it is actual well-formed linguistic tokens which are described by a grammar. For any occurrence of an utterance of $\mathcal{L}$ in the real world, the intended exhaustive model contains the utterance itself. As a consequence, if there are several occurrences of the same expression type (such as different occurrences of *Elon is frying fish*), the intended exhaustive language model contains the relevant number of copies of the expression, namely all its past and future occurrences. However, it is clear that most conceivable well-formed expressions of any given human language were never produced and never will be. Since an exhaustive model must contain all potential well-formed expressions of a language which obey the principles of grammar, the theory of exhaustive models must admit *potential tokens* for those utterances which never occur in the intended exhaustive model. If token models are already suspicious (or unacceptable) to some linguists, models comprising non-actual tokens are even more contentious. Alternative theories of the meaning of grammars have been formulated to avoid these consequences of exhaustive models.

The feature logical description languages of HPSG support alternative theories of the meaning of grammars without the need to change the syntax and semantics of descriptions. There exist four alternative ways to determine the meaning of HPSG grammars, none of which means that a given grammar of the form $\langle \Sigma, \theta \rangle$ in the sense of Definition 16 has to be abandoned to adopt the alternative interpretation of the meaning of grammars. The reason exhaustive models were presented first above is simply that they require the fewest additional definitions to make them fully explicit.

In addition to King's theory suggesting that the meaning of a grammar is an exhaustive model containing the set of potentially non-actual utterance tokens of a language (henceforth referred to as T1), there are three other proposals in the literature. In chronological order these are: (T2) a theory which proposes that grammars should be interpreted as a set of linguistic types, formalized as a set of abstract feature structures of a certain kind (Pollard & Sag 1994); (T3) the interpretation of grammars as collections of unique mathematical idealizations isomorphic to actual language tokens, outlined in Pollard (1999); and (T4) a theory of normalized HPSG grammars (by systematically adding signature extensions and structural axioms to given grammars) which denote *minimal* exhaustive models containing configurations of objects structurally identical to the well-formed utterances of a language, sketched in Richter (2007).

The most traditional view of the meaning of HPSG grammars is the only one that refers back in its formalization to a specialized variant of classical feature structures. T2 of Pollard & Sag (1994) proposes that a grammar denotes a set of mathematical representations of types of linguistic events. The main intuition is that the object types abstract away from individual circumstances of token occurrences. The object types model linguistic token expressions in the sense that an object type conventionally corresponds to one grammatical expression of a language. The expressions of a language are observed as tokens in the real world, for example as occurrences of sentences like *Elon is frying fish*). The postulated intuitive correspondence is not explicated further, but it is expected that a trained linguist will recognize which object type a linguistic token encountered in the real world corresponds to. This loose and informal relation between the denotation of a grammar (mathematical objects serving as object types) and the domain of empirically measurable events (utterances of grammatical expression of a language) is strongly criticized by King (1999), who argues that it is far from clear how a linguist would recognize the correspondence and if two linguists would reliably agree on it. Falsification of the predictions of a grammar would therefore become unnecessarily hard: The proponents of a grammar could argue that their grammar is correct because the correspondence between observed utterances and the object types admitted by the grammar was not the one assumed by their detractors. An utterance supposedly not predicted by the grammar could be argued to correspond to an object type which another linguist did not think it corresponded to, and an object type that one linguist says corresponds to an ungrammatical token utterance (thus claiming that the grammar overgenerates) could be claimed to correspond to a grammatical token utterance instead. In addition to this weakness of the relation between object types and the domain of empirically accessible data, object types have been objected to as being ontologically dubious and in any case superfluous.

From a technical perspective, the abstract feature structures of T2 can be thought of as abstractions of configurations of objects under a root node. The idea of a root node in a concrete feature structure (conventionally depicted as a graph) corresponds to the object u in sets of components of an u in an interpretation I as introduced in Definition 11. The abstract feature structures used as mathematical representations of object types, however, are not concrete feature structures, since two concrete feature structures could be isomorphic, in violation of the idea of object types without duplicates. The necessary abstract feature structures are usually constructed set-theoretically by representing each node v as equivalence classes of paths that lead to v from the root node, a labeling function which assigns sorts to these abstract nodes in accordance with the feature appropriateness function of the signature, and a treatment of nodes in relations as tuples of abstract nodes. An abstract feature structure satisfaction function defines what it means for a feature structure to satisfy a description, which is then elaborated in the notion of grammars admitting sets of abstract feature structures.

Meaning theory T3 is positioned against the theory of object types for equivalence classes of linguistic tokens, T2, and against the idea of employing actual and non-actual linguistic tokens in intended exhaustive models, T1. With T3 Pollard (1999) is strictly opposed to a notion of meaning which employs tokens rather than some form of mathematical idealization as fundamental to grammatical meaning, and he finds the concept of non-actual tokens inacceptable and self-contradictory. At the same time, the *strong generative capacity* of HPSG grammar rejects T2's ontological commitment to object types and instead strengthens the relationship between the structures in the denotation of a grammar and empirically observable token expressions. Fundamental assumptions of this theory are that no two structures in the collection denoted by a grammar are structurally isomorphic, and that each utterance token in the language which is judged grammatical finds a structurally isomorphic counterpart in the grammar's strong generative capacity. With the second requirement, T3 tightens the relationship between observables and the mathematical model, establishing a much stricter link between the predictions of a grammar and the domain of empirical phenomena than the abstract feature structure models of Pollard & Sag (1994) offers with its reference to an intuitive correspondence.

With respect to the technical details, T3 is spelled out on the basis of models (Definition 18),[4] offering three alternative ways of characterizing the strong generative capacity of a grammar. In the terminology presented above, the structures in Pollard's models can be understood as pairs of interpretations I together with a root node u whose set of components constitute I's universe. The objects in C^u_I are all defined as canonical representations by a construction employing equivalence classes of attribute paths originating at the root node: Given a grammar Γ, its strong generative capacity can be obtained by taking the set of all such canonical representations whose interpretations are Γ models. By construction, they are all pairwise non-isomorphic, and with their internal structure they can be assumed to be structurally isomorphic to grammatical utterance tokes of a language, in contrast to the abstract feature structures of Pollard & Sag (1994). The relationship to exhaustive models can be understood by noting that the canonical representations in the strong generative capacity can be abstracted from each exhaustive model.

One of the main tenets of the theories of the meaning of grammars as sets of abstract feature structures and as mathematical idealizations in the strong generative capacity is that there is a one-to-one correspondence either of object types or of mathematical idealizations to grammatical utterances in a language. Richter (2007) investigates the models of existing HPSG grammars such as the fragment of English developed in Pollard & Sag (1994) and notes that the theories of meaning T2 and T3 will necessarily lead to a one-to-many relationship between grammatical utterances and structures in the denotation of the grammar: one token utterance leads to more than one structure in the grammar denotation. Informally, the reason for that is that for both theories of meaning each structure which corresponds to a grammatical utterance entails the presence of a potentially large number of further structures. For the strong generative capacity, these additional structures come from the substructural nodes in the mathematical idealization of an utterance which, by design, must function as root nodes of admissible structures. But these additional structures are not mathematical idealizations of empirically observable grammatical utterances. In fact, many of the structures present in the strong generative capacity do not correspond to structures which can occur in grammatical utterances at all. For example, there

[4] Pollard (1999) is in fact based on Speciate Re-entrant Logic (SRL), King's precursor of RSRL, but a straightforward extension to full RSRL is provided in Richter (2004).

are many structures under *synsem* nodes in the denotation of grammars that cannot occur as the SYNSEM value of signs, because the grammars impose structural restrictions on signs which are incompatible with the shape of these configurations under a *synsem* object. Not only are they met-empirical, they are even explicitly excluded from empirical events. As argued by Richter (2007), this problem even extends to expressions which have the form of finite sentences containing extraction sites of long distnace dependencies. These structures may contain configurations that are impossible in any structure which also contains a filler linked to the extraction site. In other words, there are no overt fillers to complete these idealizations in a token utterance, but these structures are contained in the grammar denotation, including their potentially observable phonetic string and a meaning representation, they are predicted to occur as observable linguistic data.

In response to these problems, T4 develops *normal form grammars*, presented as signature and theory extensions applicable to any HPSG grammar. The basic idea behind the canonical grammar extension is to partition the denotation of grammars into utterances and making sure that every connected configuration of objects in a grammar's denotation is isomorphic to a token utterance in a language. It is then shown that for T2 and T3 this extension is insufficient to establish the intended one-to-one correspondence between observable utterances and object types (T2) or mathematical idealizations (T3), because the structures predicted by T2 and T3 will still comprise separate structures corresponding to each substructure. However, normal form grammars allow the definition of *minimal* exhaustive models, because normal form grammars can be shown to have exhaustive models which only contain non-isomorphic configurations of objects with the additional property that each of these configurations corresponds to a grammatical utterance. Proposal T4 is not forced to make any assumptions about the ontological status of its minimal exhaustive models of normal form grammars, since they do not have to be defined as a particular kind of mathematical structure (nor is this option excluded if it is desired). With T3, T4 shares the commitment to providing an isomorphic structure to each grammatical utterance of a given language rather than just a corresponding linguistic type.

HPSG is among a small group of grammar formalisms with a very precise outline of its formal foundations. It is exceptional with its alternative characterizations of the meaning of grammars based on one and the same set of core definitions of the syntax and semantics of its descriptive devices. This common core of philosophically different approaches to the scientific descripton of human languages makes their respective advantages and disadvantages comparable within one single framework, and it renders the discussion of very abstract concepts unusually concrete. Alternative approaches to grammatical meaning based on different views of the nature of scientific description of an empirical domain can be investigated and compared with a degree of detail that is hardly achieved elsewhere in linguistics.

References

King, Paul. 1999. Towards truth in Head-Driven Phrase Structure Grammar. In Valia Kordoni (ed.), *Tübingen studies in Head-Driven Phrase Structure Grammar* (Arbeitsberichte des SFB 340 No. 132), 301–352. Tübingen: Universität Tübingen. http://www.sfs.uni-tuebingen.de/sfb/reports/berichte/132/132abs.html, accessed 2018-2-25.

Pollard, Carl J. 1999. Strong generative capacity in HPSG. In Gert Webelhuth, Jean-Pierre Koenig & Andreas Kathol (eds.), *Lexical and Constructional aspects of linguistic explanation* (Studies in Constraint-Based Lexicalism 1), 281–298. Stanford, CA: CSLI Publications.

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.

Richter, Frank. 2004. *A mathematical formalism for linguistic theories with an application in Head-Driven Phrase Structure Grammar*. Universität Tübingen Phil. Dissertation (2000). https://publikationen.uni-tuebingen.de/xmlui/handle/10900/46230, accessed 2018-2-25.

Richter, Frank. 2007. Closer to the truth: A new model theory for HPSG. In James Rogers & Stephan Kepser (eds.), *Model-theoretic syntax at 10 – Proceedings of the ESSLLI 2007 MTS@10 Workshop, August 13–17*, 101–110. Dublin: Trinity College Dublin.