July 27, 2004

On the Status of Phrases in Head-driven Phrase Structure Grammar: Illustration by a Totally Lexical Treatment of Extraction

Sylvain KAHANE

Lattice, Université Paris 7 & Modyco, Université paris 10

sk@ccr.jussieu.fr

www.linguist.jussieu.fr/∼skahane

Abstract

This paper proposes a new analysis of extraction in HPSG focused on the syntax-semantics interface. Contrary to previous analyses, our approach is totally lexical: linguistic information is exclusively introduced by the way of lexical descriptions and no phrasal descriptions are needed. Our analysis of *wh*-words is based on an idea of Tesnière, treating them simultaneously as complementizers and pronouns. In this way, the combination of the filler phrase with the rest of the clause involves two simultaneous combinations of words and both phrases act as the head in one of the two combinations.

Our study allows for a precise comparison between HPSG and dependency grammars and poses the problem of the exact role of phrases in the syntax-semantics interface of a linguistic model. We argue that an HPSG without phrasal descriptions is more or less equivalent to a particular implementation of a DG and that the phrase structure only results from a particular choice in words' combination, that is, from a bottom-up covering of the dependency structure.

1. Introduction

The main goal of this paper is a better understanding of the way linguistic units combine to make sentences and particularly the role of phrases in this process. We will compare two families of approaches that have often been opposed (Hudson 1980, Mel'čuk 1988): dependency grammars (DG) and phrase structure grammars (PSG). DGs and PSGs differ by the status they give to phrases: DGs models syntax only in terms of combination of words and dependencies between words, while PSGs models syntax in terms of phrases, namely grouping of words. Nevertheless most of contemporary PSGs–and among them the Head-driven PSG (HPSG; Pollard & Sag 1987, 1994; Sag & Wasow 1999) and all the grammars based on the X-bar Theory (Jackendoff 1977)–are very near to DGs: specifying a head in each phrase makes a PSG more or less equivalent to a DG (Lecerf 1961, Paducheva 1964, Gaifman 1965). Nevertheless, some differences remain that we will explore. And an important question rises: what is exactly the role played by phrases in PSGs if they are strongly equivalent to grammars not considering phrases?

To try to understand the status of phrases in PSG and particularly in HPSG (probably the most achieved formalism for PSGs), we will focus on extraction–a phenomenon whose treatment in previous HPSG

presentations (Pollard & Sag 1994, Sag 1997, Bouma *et al.* 2001) does not fit into the dependency grammar framework: . In these HPSG modelings of extraction, *wh*-words are considered simply as markers and the essence of the construction is attributed to the phrase corresponding to the whole (relative or interrogative) clause. We do not deny that extraction phenomena involve particular constructions (in the sense of Goldberg 1995), but we do not think that these constructions justify using specific phrase descriptions (schemata in HPSG terms) for each of them. It is still possible (and even simpler and more economic) to associate the description of the particularities of these constructions to the lexical markers of these constructions, that is, not to use phrasal descriptions (and even not to consider phrases) and to consider more complex lexical descriptions. We will show how to implement in the HPSG formalism a dependency analysis of extraction, that is, an analysis that does not use phrasal descriptions but only words' combination. We will see that this analysis is descriptively equivalent to traditional analyses (and can inherit from all refinements of such analyses) and has an appreciable elegancy.

It is important to emphasize that this study focuses on the syntax-semantics interface. Phrases are components of the syntax proper, that is, the ordering of words and their grouping in phrases (see Gerdes & Kahane *to appear* for a new characterization of phrases based on these ideas). Not considering phrasal descriptions in the syntax-semantics interface is not just a game. We defend a lexicalist approach, where the structure of a sentence (and more generally of a whole text) is completely determined by the combination of structural descriptions associated to the words (and also by the prosody, not considered here). Our purpose is not to oppose HPSG and DG, but on the contrary to show what each formalism or theorical framework can bring to the other one. Our description of extraction does not fit in pure DG, given that it involves the simultaneous combination of more than two words. We will show how the HPSG formalism allows us to model such phenomena in an elegant way. In this way, the HPSG presented here can be appreciated as a possible formalization of an extended DG.

Section 2 focuses on the core of the grammar (combinations of words with their actants and modifiers) and shows how to adjust an HPSG in order to simulate a DG. Section 3 discusses the treatment of complementizers and relativizers and the interpretation of the SLASH feature in DG terms. Section 4 is the main part of the paper; we propose a new description of extraction based on a double identity of wh-words and involving only words' combinations. Section 5 is devoted to the status of phrases in HPSG. At the end, we will show how the SLASH feature is linked to the underlying phrase structure adopted by HPSG.

Note that this paper is adressed to people working in HPSG as well as those working in other frameworks, dependency grammars or not. For this reason, the presentation is more or less autonomous and does not really suppose the familiarity with the HPSG formalism or with DGs.

2. HPSG as a dependency grammar

In dependency grammars (Hays 1964, Mel'čuk 1988, Hudson 1990), a sentence is built directly by the combination of the words of the sentence, without resorting to chains of words such as phrases. The main conjecture of DGs is that the combination of the words of a sentence can be reduced to pairwise combinations of words. Moreover, DGs suppose that the combination of words is asymmetric: one word is the governor and the other is the dependent. Governor is defined exactly as syntactic head in PSG. The distinction simply lies in the fact that DGs consider only relations between words: a governor governs words, while a head heads a phrase. But if we consider two words that form a phrase, the word that is the head of the phrase is the word that governs the other word. For instance, it is equivalent to say that *apples* is the head of the

phrase *big apples* or to say that *apples* governs *big* in *big apples*.

In many aspects, HPSG can be considered as a DG. Given that in most cases the description of a phrase is reduced to the description of its head, the combination of the head of a phrase with a subphrase can be seen as the combination of two words: the head of the phrase and the head of the subphrase.

In HPSG, linguistic signs such as words are represented by feature structures. In this paper, focusing on the syntax-semantics interface, we do not mention features concerning phonology or word order. In order to be more understandable for HPSG users, we keep usual HPSG notations when it seems possible.

Section 2.1 discusses the interpretation of HPSG word descriptions in terms of DGs; Section 2.2 links phrase descriptions and combination of words in a DG.

2.1. Word descriptions

The structural description of a word in a sentence is a typed-feature structure.[1] All words are of type *word* which is a subtype of the type *grammatical-structure* (abbreviated to *gram-struc*). The relation 'isa' is the relation of inheritance on types: for instance, a feature structure of type *word* inherites all the descriptions of its supertypes and is then the unification of the description attached to the type *word* and the descriptions of its supertypes, that is, *gram-struc* and *feature-struc* (see (2)).

(1) $gram\text{-}struc : \begin{bmatrix} \text{HEAD} & cat \\ \text{VAL} & bag(cat) \\ \text{DEP} & bag(cat) \end{bmatrix}$ isa *feature-struc*

(2) $word : \begin{bmatrix} \text{VAL} \, / \, ebag \\ \text{DEP} \ ebag \end{bmatrix}$ isa *gram-struc*

(3) $word = \begin{bmatrix} \text{HEAD} & cat \\ \text{VAL} \, / \, ebag \\ \text{DEP} \ ebag \end{bmatrix}$

The value of the HEAD feature of *word* is the description of the word itself (which is a feature structure of type *cat*, described in (4)). The VAL(ENCE) feature contains the valence (= subcategorization frame, or government pattern) of the word, that is, a set of slots that will be filled by the syntactic actants of the word.[2] The value of VAL is a bag of feature structures of type *cat*.[3] The empty bag is noted *ebag*. The

---

1. We will not discuss a very important question concerning the units of the syntax-semantics interface. Clearly, words are not these units. Words are the units of the syntax proper, while the units of the semantics-syntax interface are morphemes, which can correspond to a group of words (idioms), as well as to a part of a word (*went* = GO ⊕ past).

2. We use the term *actant* proposed by Tesnière (1959) rather than the term *complement*. In particular, *actants* includes the subject. Contrary to the Chomskyan tradition (partially adopted by HPSG), we do not particularly distinguish the subject from the other actants. Of course the subject is the most salient actant and it can have contrastive properties that oppose it to other actants. But we believe that these properties do not motivate a special status for the subject any more than the properties of the direct object motivate a special status among complements. The case of infinitival verbs is special: they do not realize their subject as their own syntactic actant but it can be realized somewhere else (for instance as the subject of a raising verb). Therefore, the 'subject' of an infinitival verb can be put in a special feature (we can call SUBJ, RAISING or whatever else) whose behavior resembles SLASH rather than VAL. The same feature will be used for adjectives due to the parallelism between the constructions *Peter seems to sleep* and *Peter seems asleep*.

3. A bag is a 'set' whose elements can be repeated, that is a 'list' whose elements are not ordered. HPSG uses lists traditionally. As

default value of VAL is empty.[4] The DEP(ENDENCE) feature contains the bag of its dependents.[5] The DEP of a word is empty as long as it has not combined with other words. When it combines with one of its actants, this actant fills an actant slot, that is, unifies with an element of VAL, which is removed from VAL and put in DEP (see §2.2).

A *cat* is decribed as follows:

$$(4) \qquad cat : \begin{bmatrix} \text{FCT } syntrel \\ \text{POS } pos \\ \text{INFL } [\dots] \\ \text{CONT } cont \end{bmatrix} \text{isa } feature\text{-}struc$$

The feature FCT indicates the syntactic function of the word. Its value is a subtype of the subtype *syntrel*: *root*, *subj*, *dobj*, *iobj*, *obl*, *mod*, etc.[6]

The feature POS indicates the part of speech of the word. Its value is a subtype of the atomic type *pos*: *verb-pos*, *noun-pos*, *adj-pos*, *adv-pos*, etc.

The feature INFL contains information about the inflection of the word, such as mood and tense for verbs or number for nouns. This topic is not in the focus of this paper and INFL will be mentioned only when necessary.

The feature CONT contains the semantic content of the word.[7] For the time being we just mention that CONT contains the list of its semantic arguments as the value of the attribute ARGS:[8]

$$(5) \qquad cont : \begin{bmatrix} \text{ARGS } list(cont) \end{bmatrix}$$

We propose now some subtypes of the type *word*. Words are subdivided into *act(ant)* for *noun* and *verb* and *mod(i)f(ie)r* for *adj(ective)* and *adv(erb)*.[9] Modifiers are generally adjoined to their governor, which is their first semantic argument. They must contain in their description a description of their governor,

---

we do not need order on the values of VAL and DEP, we think it is simpler to consider them as bags, but this is not a serious point.

4. A default value is preceded by the symbol /. It can be changed by a further assignment of value.

5. Bouma *et al.* (2001) introduces a feature DEP(ENDENT)S for the dependents of a word. Contrary to their feature, our DEP feature is only filled when an element is removed from VAL and it does not contain slashed elements. Moreover, we prefer to use the term *dependence* (rather than *dependents*), which is more parallel to *valence*.

6. We make a distinction between syntactic functions and syntactic relations, depending on whether we are speaking about the syntactic function of a word (with respect to its governor) or the syntactic relation between two words. The edges of a dependency tree are labeled by syntactic relations.

7. In fact, we put in CONT only the semantic content of the radical. The semantic content brought by the inflection is put in INFL.

8. Note that we treat CONT as a head feature (see (12)), that is, the CONT value of a phrase is equal to the CONT value of its head word. But, through ARGS, the CONT value of words of a sentence points at other CONT values, giving us a graph, which is always connected. This graph is the semantic structure of sentences considered by Meaning-Text Theory (Žolkovskij & Mel'čuk 1967, Mel'čuk 1988). In the HPSG tradition, the CONT value of a phrase is the set of the CONT values of its daughters. Nothing requires us to pass up and accumulate all the CONT values, just as the percolation of syntactic information is constrained by the Locality Principle (see Section 3.4).

9. Words of type *verb* are necessarily finite verbs. Our treatment of non-finite verbs will be explained in Section 3.1.

which is the value of MOD.[10] We also instantiate the feature MOD of actants by the atomic type *none* that cannot unify with a non-empty value. This ensures that they cannot be used to modify another word (see *mod-val-computation* in §2.2).[11]

(6) **a.** $act : \left[\text{HEAD} \left[\text{MOD } none \right] \right]$ isa *word*

   b. $noun : \left[\text{HEAD} \left[\text{POS } noun\text{-}pos \right] \right]$ isa *act*

   c. $verb : \left[\text{HEAD} \left[\text{POS } verb\text{-}pos \right] \right]$ isa *act*

   d. $trans\text{-}verb : \begin{bmatrix} \text{HEAD} \left[\text{CONT} \left[\text{ARGS } \langle \boxed{1}, \boxed{2} \rangle \right] \right] \\ \text{VAL} \left\{ \begin{bmatrix} \text{FCT } subj \\ \text{POS } noun\text{-}pos \\ \text{CONT } \boxed{1} \end{bmatrix}, \begin{bmatrix} \text{FCT } dobj \\ \text{POS } noun\text{-}pos \\ \text{CONT } \boxed{2} \end{bmatrix} \right\} \end{bmatrix}$ isa *verb*

(7) **a.** $modfr : \begin{bmatrix} \text{HEAD} \begin{bmatrix} \text{FCT } mod \\ \text{CONT} \left[\text{ARGS } \langle \boxed{1}, \ldots \rangle \right] \\ \text{MOD} \left[\text{CONT } \boxed{1} \right] \end{bmatrix} \end{bmatrix}$ isa *word*

   b. $adj : \begin{bmatrix} \text{HEAD} \begin{bmatrix} \text{POS } adj\text{-}pos \\ \text{MOD} \left[\text{POS } noun\text{-}pos \right] \end{bmatrix} \end{bmatrix}$ isa *modfr*

   c. $adv : \begin{bmatrix} \text{HEAD} \begin{bmatrix} \text{POS } adv\text{-}pos \\ \text{MOD} \left[\text{POS } verb\text{-}pos \right] \end{bmatrix} \end{bmatrix}$ isa *modfr*

The word descriptions can be seen as subtrees of the dependency tree. These subtrees can be represented as proposed by Nasr (1995; see also Kahane 2001). A black node corresponds to the current node (= the HEAD value), while a white node corresponds to a requirement, either a syntactic actant (= an element of VAL value) or the governor (= the MOD value). In our diagrams we use the labels N, V, Adv and Adj for feature structures *cat* with $\left[\text{POS } noun\text{-}pos \right]$, $\left[\text{POS } verb\text{-}pos \right]$, etc.

(8) **a. Dependency interpretation of** *trans-verb*
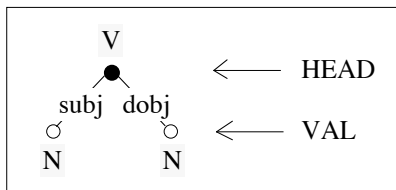
   **b. Dependency interpretation of** *adj*

---
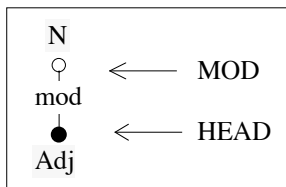
10. Warning: it is not the value of HEAD but the value of MOD taht is the head of the potential phrase containing them.

11. We leave the question of determiners aside. They could be put in the VAL or INFL features of the noun or in a special feature (see the SPR feature of Sag & Wasow (1999)).

In the next section, we will see how word descriptions can be combined to produce a sentence and its representation (namely, its syntactic dependency tree). This will be exemplified with sentence (9a) and its dependency tree (9b).

(9) **a.** *Peter often eats red beans.*

    b. **Dependency tree of (a)**
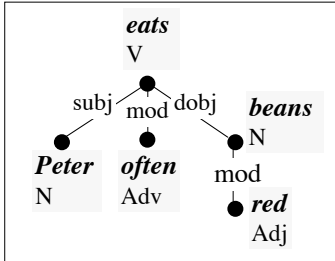
The words of sentence (9) have the following (partial) descriptions:

(10) **a.** *eats:*$\big[$HEAD$\big[$CONT *eat-rel*$\big]\big]$ isa *trans-verb*

    b. *Peter:*$\big[$HEAD$\big[$CONT *Peter-rel*$\big]\big]$ isa *noun*

    c. *beans:*$\big[$HEAD$\big[$CONT *bean-rel*$\big]\big]$ isa *noun*

    d. *red:*$\big[$HEAD$\big[$CONT *red-rel*$\big]\big]$ isa *adj*

    e. *often:*$\big[$HEAD$\big[$CONT *often-rel*$\big]\big]$ isa *adv*

2.2. Combination of words

HPSG comes from the phrase structure grammar tradition, which in turns comes from the Immediate Constituent Analysis (Bloomfield 1933). It does not combine words directly, but it sets them into phrases. A new phrase is obtained by combination of two subphrases (or daughters) called the HEAD-DAUGHTER (= HDTR) and the NON-HEAD-DAUGHTER (= NHDTR). Phrases are subtypes of *gram-struc* like words. Contrary to words, phrases have two additional features for their daughters:

$$(11) \quad phrase: \begin{bmatrix} \text{HDTR } gram\text{-}struc \\ \text{NHDTR} \begin{bmatrix} gram\text{-}struc \\ \text{VAL } ebag \end{bmatrix} \end{bmatrix} \text{ isa } gram\text{-}struc$$

The core of a phrase description, as for a word description, is given in the HEAD feature. HPSG is a Head-driven PSG due to the fact that, in most cases, the description of a phrase comes directly from the description of its HEAD-DTR:[12]

(12) **Head Computation (Head Feature Principle)**

$$hd\text{-}phrase: \begin{bmatrix} \text{HEAD} \boxed{1} \\ \text{HDTR}\big[\text{HEAD}\boxed{1}\big] \end{bmatrix} \text{ isa } phrase$$

Given that the description of a phrase comes only from its head word, a phrase can be identified with its head word. In other words, the combination of two phrases can be interpreted as the combination of two

---

12. A boxed number (such as $\boxed{1}$) indicates that two features share the same value. If these two features receive descriptions separately, these descriptions must be unified to give their common value. If the descriptions cannot be unified, the analysis fails.

words: the HEAD-DAUGHTER is the governor and the NON-HEAD-DAUGHTER is the dependent. More precisely, only the HEAD value of a phrase and its head word are identical; the other values (VAL and DEP) change when the word takes new dependents. What is called a *phrase* in HPSG is, in dependency terms, the description of a word resulting from the combination with some of its dependents. In dependency terms, phrases are not linguistic entities such as signs, but only computational objects used to compute the dependency tree and to ensure that each word is saturated (that is, it consumed its whole valence).

Note that in HPSG it is essential that the NHDTR has an empty VALENCE when it combines with the HDTR (see (15)). In dependency terms, it means that the dependency tree is built bottom up: a word can combine with its governor only if it has finished combining with its dependents.

In order to be more understandable for HPSG users we have kept the usual HPSG notations. In a more dependency style, we would prefer to use NODE instead of HEAD, GOVERNOR instead of HEAD-DAUGHTER and DEPENDENT instead of NON-HEAD-DAUGHTER. But there is another reason to maintain the usual HPSG notations: as said previously, HEAD does not correspond to the description of a single node but rather of a larger part of the dependency tree containing generally only one node but which can contain two or three nodes. This is the case for modifiers, which contain, in their HEAD value, the feature MOD describing their (potential) governor. We will see another case with complementizers in Section 3.

There are two major types of *phrase* structures (or combinations of words) according to whether the NHDTR (= dependent) is an actant or a modifier. These two types differ in the computation of the VALENCE: an actant NHDTR must be removed from the valence of the HDTR, while a modifier does not change its valence. We call them *act-val-computation* and *mod-val-computation*.[13]

(13) **Valence Computation**

    a. *hd-act-phrase* :
$\begin{bmatrix} \text{VAL}\ \boxed{1} \\ \text{HDTR}\begin{bmatrix} \text{VAL}\ \boxed{1} \oplus \{\boxed{2}\} \end{bmatrix} \\ \text{NHDTR}\begin{bmatrix} \text{HEAD}\ \boxed{2} \end{bmatrix} \end{bmatrix}$ isa *hd-phrase*

    b. *hd-mod-phrase* :
$\begin{bmatrix} \text{VAL}\ \boxed{1} \\ \text{HDTR}\begin{bmatrix} \text{HEAD}\ \boxed{2} \\ \text{VAL}\ \boxed{1} \end{bmatrix} \\ \text{NHDTR}\begin{bmatrix} \text{HEAD}\begin{bmatrix} \text{MOD}\ \boxed{2} \end{bmatrix} \end{bmatrix} \end{bmatrix}$ isa *hd-phrase*

Notice that *hd-mod-phrase* cannot be used with a nominal or verbal NHDTR, because it has an empty MOD value that cannot unify with the HEAD value of HDTR. On the contrary, an adjective or adverb can realize an actant, like the adjective *French* in *the French production* or the adverb *here* in *Peter came here*.

To finish, we can explicitly compute the dependency tree with the DEP feature. The computation is very easy: the NHDTR (= the dependent) must be added in the DEP bag of the HDTR (= the governor). We add the following constraint to *hd-phrase*.

(14) **Dependency Computation**

---

13. The symbol $\oplus$ designates the union on bags. It is prefered to the $\cup$ symbol, which could be confused with unification. Union of bags differs from set union by the fact that all the occurrences of an element are counted: $\{1,2\}\oplus\{1,3\}=\{1,1,2,3\}$. Note that in (13), $\boxed{1}$ refers to a bag of *cat* and $\boxed{2}$ to a *cat* due to the definition of VAL and HEAD features (see (1)).

$$\textit{hd-phrase} \Longrightarrow \begin{bmatrix} \text{DEP}\;\boxed{1} \oplus \{\boxed{2}\} \\ \text{HDTR}\,[\,\text{DEP}\;\boxed{1}\,] \\ \text{NHDTR}\,[\,\text{HEAD}\;\boxed{2}\,] \end{bmatrix}$$

Note that the feature FCT of each element of DEP is instantiated, either directly in the description of the word when it is a modifier or via the valence of its governor when it is an actant (due to the fact that an actant necessarily unifies with an element of the valence of its governor (see (13a)).

Finally, after the unification of all the inherited constraints, we obtain for our two types of phrases:

$$(15)\;\textbf{a.}\quad \textit{hd-act-phrase} = \begin{bmatrix} \text{HEAD}\;\boxed{1} \\ \text{VAL}\;\boxed{2} \\ \text{DEP}\;\boxed{3} \oplus \{\boxed{4}\} \\ \text{HDTR}\begin{bmatrix} \text{HEAD}\;\boxed{1} \\ \text{VAL}\;\boxed{2} \oplus \{\boxed{4}\} \\ \text{DEP}\;\boxed{3} \end{bmatrix} \\ \text{NHDTR}\begin{bmatrix} \text{HEAD}\;\boxed{4} \\ \text{VAL}\;\textit{ebag} \end{bmatrix} \end{bmatrix}$$

$$\textbf{b.}\quad \textit{hd-mod-phrase} = \begin{bmatrix} \text{HEAD}\;\boxed{1} \\ \text{VAL}\;\boxed{2} \\ \text{DEP}\;\boxed{3} \oplus \{\boxed{4}\} \\ \text{HDTR}\begin{bmatrix} \text{HEAD}\;\boxed{1} \\ \text{VAL}\;\boxed{2} \\ \text{DEP}\;\boxed{3} \end{bmatrix} \\ \text{NHDTR}\begin{bmatrix} \text{HEAD}\;\boxed{4}\,[\,\text{MOD}\;\boxed{1}\,] \\ \text{VAL}\;\textit{ebag} \end{bmatrix} \end{bmatrix}$$

These two 'phrases' can be interpreted in terms of combinations of words (where the word descriptions are represented as in (8)):

(16) **a.** **Dependency interpretation of** *hd-act-phrase* **(15a)**
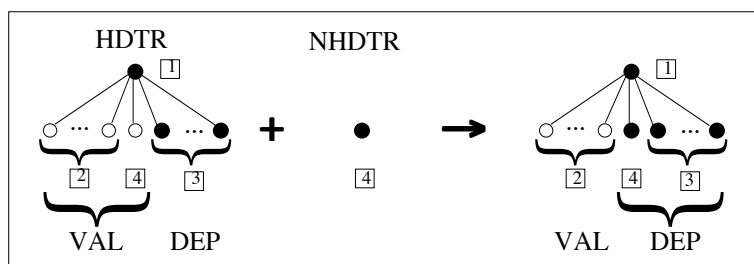
b. **Dependency interpretation of** *hd-mod-phrase* **(15b)**
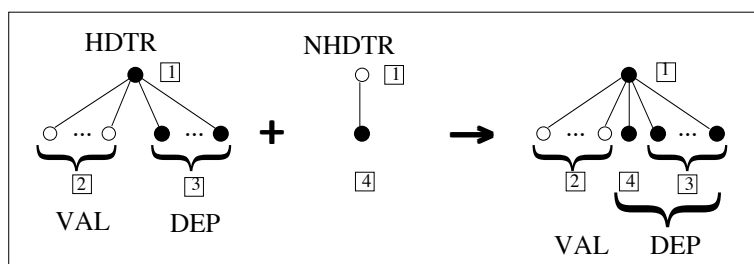
As we can see, both ways of combining are based on the unification of a white node of one structure with a black node of the other. In fact, white and black colors of a node correspond to VALENCE and DEPENDENCE. From this point of view, the potential governor (the MOD value) might be included in the valence.[14] Both word combination operations *hd-act-phrase* and *hd-mod-phrase* can be subsumed by a single word combination operation, illustrated in (17): a white node can be identified with a black node (giving us a black node).

(17) **Word Combination Operation**
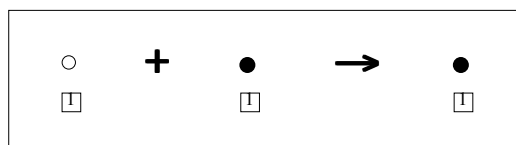
A dependency grammar based on the previous principle has been proposed by Nasr (1995) and extended by Kahane (2001). Such a grammar is inspired by Tree Adjoining Grammars (TAG; Joshi 1987), where each word is associated with an elementary structure, the whole structure being obtained by simple combinations of these elementary structures. But contrary to TAG, which uses two different operations of combinations–substitution and adjoining, it uses a single operation.

A unified treatment of actants and modifiers, comparable to our single operation (17), has been proposed by van Noord & Bouma (1994) (and improved in Bouma *et al.* 2001). The idea is to add all the modifiers of a word in its VAL bag. This can be done by the recursive application of the following lexical rule:[15]

---

14. In the dependency tradition it is usual to include the governor in the valence. Mel'čuk (1988) distinguishes the active valence for the potential dependents from the passive valence for the potential governor.

15. Bouma *et al.* (2001) introduce a special feature for the extended valence they call DEPS. Moreover, they propose a way to compute the various possible values of DEPS without using lexical rules. Although their proposition is more elegant, it remains that a given

(18)   **Modifier Introduction Lexical Rule**

$$
\begin{bmatrix} word \\ \text{HEAD}\,\boxed{1} \\ \text{VAL}\,\boxed{2} \end{bmatrix} \mapsto \begin{bmatrix} word \\ \text{VAL}\,\boxed{2} \oplus \{\, [\,\text{MOD}\,\boxed{1}\,]\,\} \end{bmatrix}
$$

Now, the two ways of the VALENCE computation of (13) are subsumed by the sole *hd-act-phrase* (13a). In case of head-modifier combination, an element $[\,\text{MOD}\,[\,\ldots\,]\,]$ is removed from the VAL of the HDTR.

It is time for an example. In (19), we give the analysis of sentence (9a) *Peter often eats red beans*. In this figure, an HDTR is placed vertically under its mother phrase. We use the abreviation N for the feature structure $[\,\text{HEAD}\,[\,\text{POS } noun\text{-}pos\,]\,]$ (and equivalently V, Adj and Adv).

(19)   **Analysis of sentence (9a)**
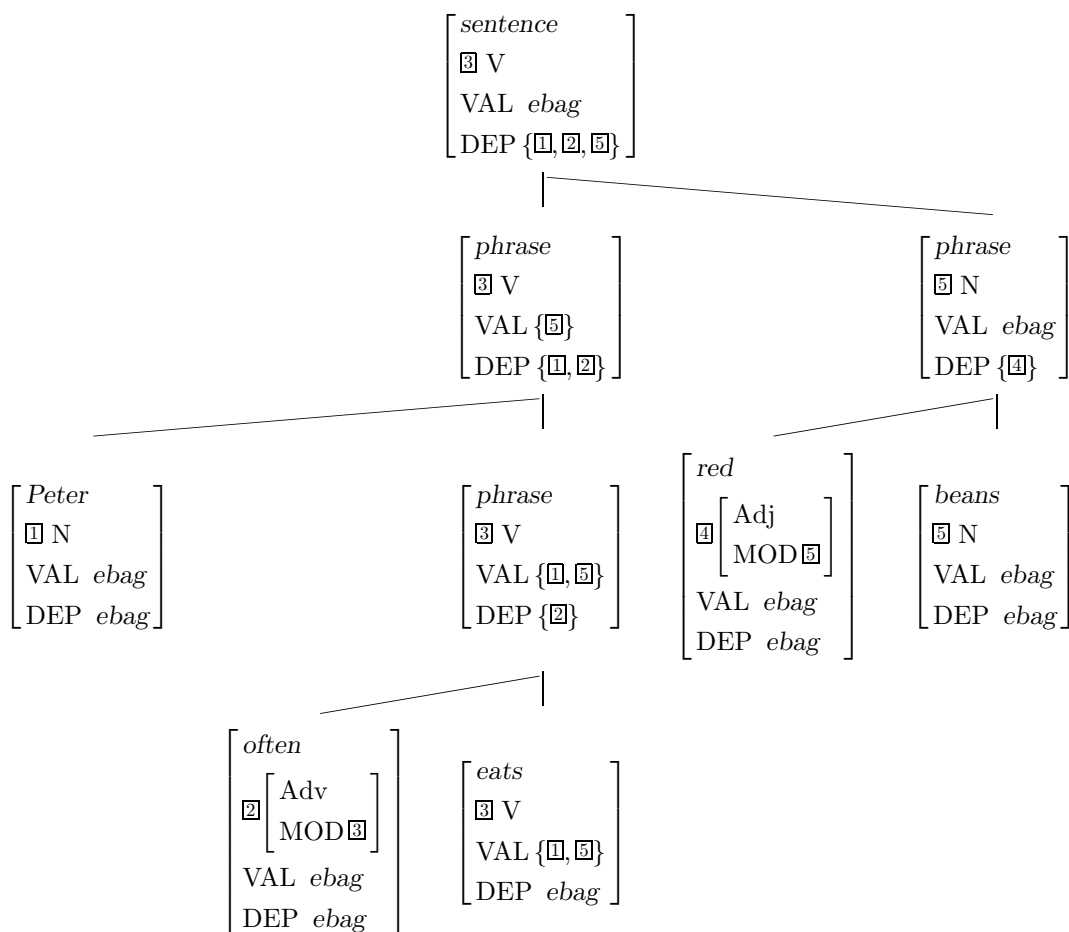
The fact that the root of a sentence must be a finite verb is ensured by imposing the condition that a sentence must be a saturated phrase of subtype *sentence*:

---

word can have numerous elementary structures, which is not convenient from a cognitive point of view nor from a computational point of view. Our dependency grammar avoids this problem.

(20) **a.** *sentence* : $\begin{bmatrix} \text{HEAD} \begin{bmatrix} \text{FCT} & root \\ \text{POS} & verb\text{-}pos \end{bmatrix} \\ \text{VAL} \ ebag \end{bmatrix}$ isa *phrase*

    b. *hd-act-sentence* isa *hd-act-phrase* + *sentence*

    c. *hd-mod-sentence* isa *hd-mod-phrase* + *sentence*

Note that the order in which a word combines with its various dependents does not matter. For instance, in our example, the verb *eats* combines with the adverbial *often*, its subject *Peter* and then its direct object (*red*) *beans*. In Chomskyan tradition, the subject is the last dependent of the verb to combine with it. This is justified by internal considerations of the theory relative to binding. Although traditional HPSGs continue to ensure that the subject is the last dependent to combine with the verb (putting it in a specific feature, SUBJ or SPR, that must be discharged after the rest of the valence), nothing in HPSG requires this, since binding is based on an obliqueness order of the syntactic relations (Pollard & Sag 1994, p. 248).[16]

Another reason to order the combination of a verb with its dependents is to avoid spurious ambiguities. Indeed, our grammar allows three phrase structures for the sentence (9a):

(21) **a.** *((Peter(often eats)) (red beans))*

    b. *(Peter((often eats) (red beans)))*

    c. *(Peter(often (eats (red beans))))*

But from the viewpoint of dependency grammar, this does not matter because these three phrase structures give us the same dependency tree (presented in (9b)).[17]

We will now see how the analysis (19) can be interpreted in terms of dependency grammar. The computation of the dependency tree comes from the DEP values of the maximal projection of each word. The maximal projection of a word is the highest phrase it heads. The computation of the dependency tree can be represented by using the representation we have proposed for word descriptions in (8) and for phrase descriptions in (16). Figure (22) is a copy of (19) where each feature structure is replaced by its representation in terms of dependency. The HPSG analysis in (19) now appears as a particular computation of the dependency tree, namely a bottom-up computation. The figure can be read in this way: for instance, in the bottom of the figure we see the structures associated to *often* and *eats*, and just above the result of the combination of these two structures with the *head-mod-phrase* schema. In these two lexical structures, the features have different values, but after the combination these two features share the same value that is the unification of the two previous values.

---

16. More exactly, Pollard & Sag (1994) does not use syntactic relations explicitly, but orders the actants in VALENCE (called there the SUBCAT list) and use this order as obliqueness order.

17. A third reason to order the combination of a verb with its dependents concerns coordination. The argument is that the subject can be easily factorized, like in *Peter eats red beans and drinks water*. But this argument is not reliable, because it is also possible to factorize the direct object: *Peter likes, and Bill hates, red beans*. Moreover, the contrast between subject and object factorizations can be captured in a different way.

(22)   **Dependency interpretation of (19)**
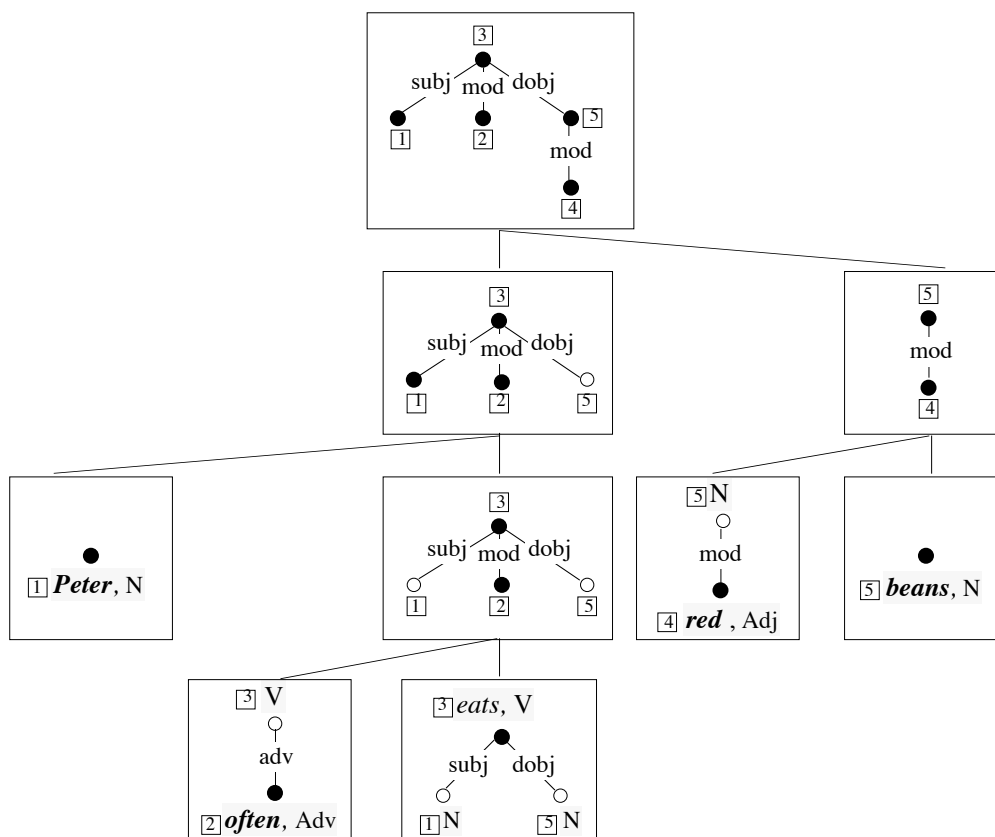
3. Complementizers, relativizers and other transferrers

Tesnière (1959), who is above all known to be the father of modern dependency theory, proposes an original classification of words that we will adopt here. Tesnière postulated that there are only four main parts-of-speech – verb, noun, adjective and adverb – with the following relations:

- verb actants are nouns;

- verb modifiers are adverbs;

- noun dependents (actants or modifiers) are adjectives;

- adjective and adverb dependents are adverbs.

When a word of a part-of-speech other than adjective needs to modify a noun, it must be 'transferred' into an adjective, that is, it must be covered by a special word which masks it and makes it appear as an adjective for its governor. For instance, the transfer of nouns (into adjectives and adverbs) is effected by prepositions.[18]

In Section 3.1, we will see how the Tesnière transfer can be formulated in HPSG. Section 3.2 will study relativizers and introduce the SLASH feature and the constraints on the computation of this feature. Section

---

18. Tesnière's terms for *transfer* and *transferrer* are Fr. *translation* and *translatif* (in French, *translation* only means a translatory movement). The theory of transfer is implicit in the theory of the three ranks of Jespersen (1924).

3.3 proposes a simple solution of the *that*-trace effect in English and the *qui-que* alternation in French. Section 3.4 concerns the representation of transferrers and the Locality Principle.

3.1. Transferrers

Transferrers receive the common POS *t(ransferre)r-pos*. Alongside POS, we will introduce a new feature UPPOS (up part-of-speech) oriented upwards, that is, towards the governor. Plain words–verbs, nouns, adjectives and adverbs–have the same value for both POS and UPPOS:

(23)   *plain-word* : $\left[\text{HEAD} \begin{bmatrix} \text{UPPOS} \boxed{1} \\ \text{POS} \boxed{1} \end{bmatrix} \right]$ isa *word*

The UPPOS of a transferrer is the part-of-speech into which it transfers its actant: its value is of type *main-pos*, which is a supertype for *noun-pos*, *verb-pos*, *adj-pos* and *adv-pos*. The actant of a transferrer will be called the *transferee*.

(24)   *transferrer* : $\begin{bmatrix} \text{HEAD} \begin{bmatrix} \text{UPPOS} \ \textit{main-pos} \\ \text{POS} \ \textit{tr-pos} \end{bmatrix} \\ \text{VAL} \{ [\text{FCT} \ \textit{transf}] \} \end{bmatrix}$ isa *word*

For instance, complementizers such that *that*, *if* or *whether* are transferrers of verb into noun:

(25) **a.**   *verb-transferrer* : $\left[\text{VAL} \{ [\text{UPPOS} \ \textit{verb-pos}] \} \right]$ isa *transferrer*

    b.   *complementizer* : $\left[\text{HEAD} [\text{UPPOS} \ \textit{noun-pos}] \right]$ isa *verb-transferrer*

From a theoretical point of view, our feature UPPOS attributes a common property to a class of words, predicting that they are a 'natural class'–that is, that they should behave alike in some ways. For instance, the subject of the verb *annoy* can be a noun or a *that*-clause, that is, a UPPOS *noun-pos* (*Peter's departure/That Peter leaves annoys me*).

(26)   *annoy* : $\left[\text{VAL} \{ \begin{bmatrix} \text{FCT} \ \textit{subj} \\ \text{UPPOS} \ \textit{noun-pos} \end{bmatrix}, \begin{bmatrix} \text{FCT} \ \textit{dobj} \\ \text{POS} \ \textit{noun-pos} \end{bmatrix} \} \right]$ isa *verb*

Notice that among verbs only finite verbs are *plain-word* and have UPPOS *verb-pos*: infinitival verbs and gerunds are UPPOS *noun-pos* and participles are UPPOS *adj-pos*, because they respectively alternate with nouns (27a,b) and adjectives (27c-e) and never with finite verbs:[19]

(27) **a.**   *Peter wants to drink / wants a beer.*
    b.   *Peter left without sleeping / without his cap.*
    c.   *the cap forgotten by Peter / the red cap*
    d.   *the guy sleeping with his cap / the sleepy guy*
    e.   *Peter is sleeping with his cap / is asleep.*

The noun-transferrers are the prepositions. Note that prepositions can also take infinitival verbs and gerunds as dependent. This is not surprising because these are verbs transferred into nouns.

---

19. As proposed by Tesnière, the copula *to be* in (27e) should be not considered as a plain verb, but rather as a transferrer of adjective into verb.

3.2. Relativizers

Relativizers are transferrers of verbs into adjectives. In other words, a relativizer introduces a clause that can modify a noun, that is, a relative clause:

(28) **a.** *the guy that came yesterday*
     b. *the guy (that) you know*
     c. *the guy (that) you are talking to*

Relativizers are markedly more complicated than complementizers because the noun modified by a relative clause needs to play a semantic role in this clause: it shares its semantic content with a slashed noun. The usual assumption is that the antecedent noun is coindexed with the gap (= the slashed noun). We do not think that indexes are useful here. The use of indices comes from the Chomskyan tradition, where the gap is filled by a trace and traces are considered as some sort of pronoun without phonological realization. Although, following Sag & Fodor (1994), traces have been abandoned in HPSG's analyses of extraction, coindexation has survived. Here we defend a more radical analysis, coming from the Meaning-Text framework (Žolkovskij & Mel'čuk 1967, Kahane & Mel'čuk 1999): the antecedent noun *is* the semantic argument of the governor of the gap. Formally, it means that the CONT values of the antecedent noun and the slashed noun are identical. So relativizers have the following description, where the BIND value must be discharged with a slashed element as explained below.[20]

$$
(29)\quad relativizer: \begin{bmatrix} \mathrm{HEAD} \begin{bmatrix} \mathrm{UPPOS}\ \textit{adj-pos} \\ \mathrm{FCT}\ \textit{mod} \\ \mathrm{MOD}\ \ \mathrm{N}:\boxed{1} \end{bmatrix} \\ \mathrm{BIND}\ \{\,\mathrm{N}:\boxed{1}\,\} \end{bmatrix}\ \text{isa}\ \textit{verb-transferrer}
$$

Slashed elements are part of the set of potential syntactic dependents of a word that are precisely not realized as dependent of their syntactic governor. For this reason, they are put outside of the VALENCE in a special bag called SLASH. The fact that VAL ⊕ SLASH is equal to the bag of potential dependents can be expressed by an equation as in Bouma *et al.* (2001) or ensured by a lexical rule as traditionally done and proposed here. The lexical rule (30) says that each word description with a non-empty VAL can give a new word description where one of the elements of VAL has been moved into the special feature SLASH, which is discharged by a different process than VAL. This lexical rule can be applied recursively.[21]

(30) **Slash Introduction Lexical Rule**

$$
\begin{bmatrix} \textit{word} \\ \mathrm{VAL}\ \boxed{1} \oplus \{\boxed{3}\} \\ \mathrm{SLASH}\ \boxed{2} \end{bmatrix} \mapsto \begin{bmatrix} \textit{word} \\ \mathrm{VAL}\ \boxed{1} \\ \mathrm{SLASH}\ \boxed{2} \oplus \{\boxed{3}\} \end{bmatrix}
$$

In (30), contrary to (15a), an element is removed from VAL without being added to DEP. In fact, a slashed element never appears in the DEP bag of its syntactic governor. We consider that the link between a

---

20. We use the abbreviated notation N : $\boxed{1}$ for the feature structure $\begin{bmatrix} \mathrm{POS}\ \textit{noun-pos} \\ \mathrm{CONT}\ \boxed{1} \end{bmatrix}$.

21. Combined with the Modifier Introduction Lexical Rule (18), the Slash Introduction Lexical Rule allows for modifiers to be slashed and extraction of modifiers to be handled. In the absence of the Modifier Introduction Lexical Rule, a second Slash Introduction Lexical Rule would be necessary.

slashed element and its governor, which we call a *quasi-dependency*, is not a true syntactic dependency. This is justified by two reasons. First, contrary to true dependencies, quasi-dependencies do not obey the tree well-formedness rules of the dependency tree and can form cycles (see (40)). Second, contrary to true dependencies, quasi-dependencies do not intervene in the word order rules due to the fact that a slashed element is not realized in the projection of its governor. In our figures, a quasi-dependency is represented by a dashed arrow. Figure (31) proposes a dependency representation of the Slash Introduction Lexical Rule.

(31)   **Dependency interpretation of the lexical rule (30)**
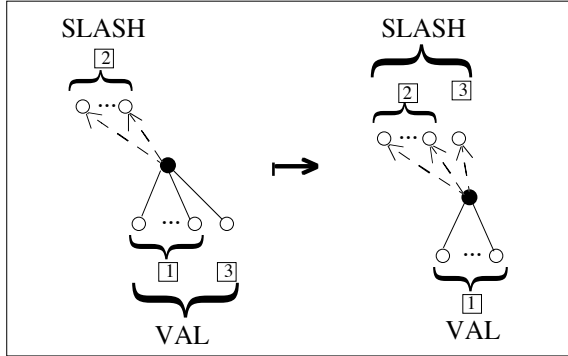
We will now study some constraints on extraction. Two types of constraints must be considered (a third type concerns only the *wh*-extraction and the pied-piping): the elements put into SLASH and the path followed by a slashed element.

Constraints on slashed elements. Not all elements of VALENCE can be slashed. For instance, in French, contrary to English, the transferee of a preposition can never be slashed:

(32)   * *le   type que tu   parles   à*
        the guy   that you   are-talking to

Therefore we introduce a subtype *gaprel* of *syntrel* such that only a syntactic relation that is a subtype of *gaprel* allows for its dependent node to be slashed. For instance, the relation *transf(er)*, between a transferrer and its transferee, will be a subtype of *gaprel* in English but not in French. This limitation on the type of slashed elements is captured by allowing the Slash Introduction Lexical Rule (30) to apply only with an element whose syntactic function is a subtype of *gaprel*. Note that we do not make a particular treatment for subjects and, contrary to Pollard & Sag (1994), we consider that subjects can be slashed and that the relation *subj* is a subtype of *gaprel*. We will come back to subject extraction in the next subsection.

Constraints on the path of a slashed element. The SLASH value of a phrase is the union of the SLASH values of its daughters. If the HDTR has a non-empty BIND, it is unloaded with an element of the SLASH value of the NHDTR. Both properties are expressed by the following constraint:

(33)   **Slash Computation**

$$phrase \Longrightarrow \begin{bmatrix} \text{SLASH} \boxed{1} \oplus \boxed{2} \\ \text{HDTR} \begin{bmatrix} \text{SLASH} \boxed{1} \\ \text{BIND} \boxed{3} \end{bmatrix} \\ \text{NHDTR} \begin{bmatrix} \text{SLASH} \boxed{2} \oplus \boxed{3} \end{bmatrix} \end{bmatrix}$$

Nevertheless, there are well-known constraints on the path a slashed element can follow. These constraints have been first brought to the fore by Ross (1967) in terms of islands, that is, in terms of types of phrases that do not allows for a slashed element to escape.[22] Here, following Kaplan & Zaenen's (1989) LFG description of extraction, we prefer to use syntactic relations to control the path of slashed elements. For this, we define a subtype *liftrel* of *syntrel* such that only a syntactic relation that is a subtype of *liftrel* allows for a slashed element to lift (see Hudson (1990) for a DG description of extraction in terms of lifting, called there raising, and Kahane *et al.* (1998) for a polynomial processing). In other words, a word which does not have a syntactic function of type *liftrel* cannot allow for a slashed element to lift and hence must have an empty SLASH bag. This gives us the following constraint on *gram-struc* (where *nebag* is a supertype for all non empty bags):

(34) **Island Constraint**

$$\begin{bmatrix} \textit{gram-struc} \\ \text{SLASH } \textit{nebag} \end{bmatrix} \Longrightarrow \begin{bmatrix} \text{HEAD} \begin{bmatrix} \text{FCT } \textit{liftrel} \end{bmatrix} \end{bmatrix}$$

For instance, the subject island in (35a) is accounted for by the fact that the syntactic relation *subj* is not a subtype of *liftrel* (and therefore the slashed element cannot raise from the *that*-clause to the verb *annoys*). The fact that the extraction is possible in the impersonal construction (35b) simply comes from the fact that the *that*-clause is no longer subject (the subject is the dummy *it*) but has another syntactic function we call *quasi-subject* and that *quasi-subject* is a subtype of *liftrel*.

(35) **a.** * *Who does that Peter loves annoy you?* (⇐ *That Peter loves her annoys you*)
     b. *Who does it annoy you that Peter loves?*

Moreover, lexical constraints can be easily added to the constraint (34): a particular lexical unit can require one of its actants to have an empty SLASH.

3.3. That-*trace effect and French* qui-que *alternation*

Following our hypothesis that a finite verb cannot be the actant of a verb without being transferred into a noun, we consider that there is, in *that*-less sentential clause and in *that*-less relatives (as in (28)), a verb-transferrer without phonological realization we call the *null transferrer*.[23] We will show how to encode the difference of behavior between *that* and the null transferrer (thus solving the problem of the *that*-trace effect) and to describe the *qui-que* alternation in French.

As we will see, verb-transferrers are sensitive to whether the subject of the verb they transfer is slashed. For this reason, we introduce on finite verbs a feature SUBJ that takes the value + when the subject is in VAL and the value − when it is in SLASH (the changing of the value must be incorporated into by the lexical rule (30) transferring VAL elements into SLASH).[24]

---

22. Evidently, Ross' description was in terms of transformations and not in terms of slashed elements, but the ideas remain the same.

23. Our analysis contrasts with earlier HPSG analyses, which tend to suppose that a verb can subcategorize a clause and that *that* can be present but does not change many things (see, for instance, Sag & Wasow 1999, p. 259).

24. Instead of introducing a special feature to control whether or not the subject is slashed, Bouma *et al.* (2001), following a recent trend in HPSG, keep on a copy of the argument structure of a word (it is the value of ARG-ST) in its HEAD description. In such a case, it is possible to verify whether the subject (the first element of the list ARG-ST) is slashed or not, slashed elements being subtypes

We can now easily encode the contrast between *that* and the null verb-transferrer. First, the null complementizer cannot be used as subject (** Peter leaves annoys me* vs. *That Peter leaves annoys me*). Second, the null relativizer does not allow for the subject of the verb it transfers to be slashed (** The man lives here is a student* vs. *The man that lives here is a student*). Third, like the null relativizer, *that*-complementizer does not allow for the subject of the verb it transfers to be slashed (** Who did Peter claim that left?* vs. *Who did Peter claim left?*). This last phenomenon is known as *that*-trace effect.[25] Note that the null relativizer is possible with a subject extraction as long as it is not the subject of the verb it transfers (*The man Peter claims left is a student*). The following constraints are sufficient to get the right properties:

(36) **a.** *that-complementizer* : $\big[\,\text{VAL}\,\{\,[\,\text{SUBJ}+\,]\,\}\,\big]$ isa *complementizer*

     b. *null-complementizer* : $\big[\,\text{FCT}\ \neg subj\,\big]$ isa *complementizer*

     c. *that-relativizer* isa *relativizer*

     d. *null-relativizer* : $\big[\,\text{VAL}\,\{\,[\,\text{SUBJ}+\,]\,\}\,\big]$ isa *relativizer*

French also has a verb-transferrer which can be both complementizer and relativizer. Contrary to English, this word is a $^{*}k^{W}$-word.[26] It has two alternating forms *qui* and *que*. The traditional analysis says that *qui* is the subject relativizer in (37a) and *que* the object relativizer in (37b) and that only *que* is a complementizer, as in (37c). But this analysis fails in case of subject extraction in a subordinated clause as in (37d), where the relativizer is *que* and the complementizer is *qui*.

(37) **a.** *le type qui vient*

       'the guy that comes'

     b. *le type que tu regardes*

       'the guy that you are watching'

     c. *tu penses que Pierre vient*

       'you think that Peter comes'

     d. *le type que tu penses qui vient*

       'the guy that you think (that) comes'

The solution (adapted from Moreau 1971) is very simple.[27] The words *qui* and *que* are two forms of the same verb-transferrer and this transferrer has the marked form *qui* when the verb it transfers has a slashed

---

of a special type (*gap-ss*). Nevertheless, keeping on the argument structure in the HEAD description of a word is a clear violation of the Locality Principle (see Section 3.4). I prefer to know exactly what information on non-head daughters is useful rather than to allow for the whole information on the dependence to be accessible (especially as, by recursivity, not only the dependence but the whole projection of the word is accessible).

25. See Pollard & Sag (1994, p. 171sqq) for a more complicated analysis where subjects are never slashed and a comparison with the GB treatment.

26. English *wh*-words come from the Indo-European root $^{*}k^{W}$, which have given *qu*-words in French (and *w*-words in German, *ch*-words in Italian, etc.). Contrary to English, French has no verb-transferrers of another family (such as English *that* or German *d*-words coming from the Indo-European root $^{*}to$) and French *qu*-words form a less homogeneous family than English *wh*-words, containing both pure verb-transferrers and pronouns. See, for instance, Haspelmath 1997 or Le Goffic 2003.
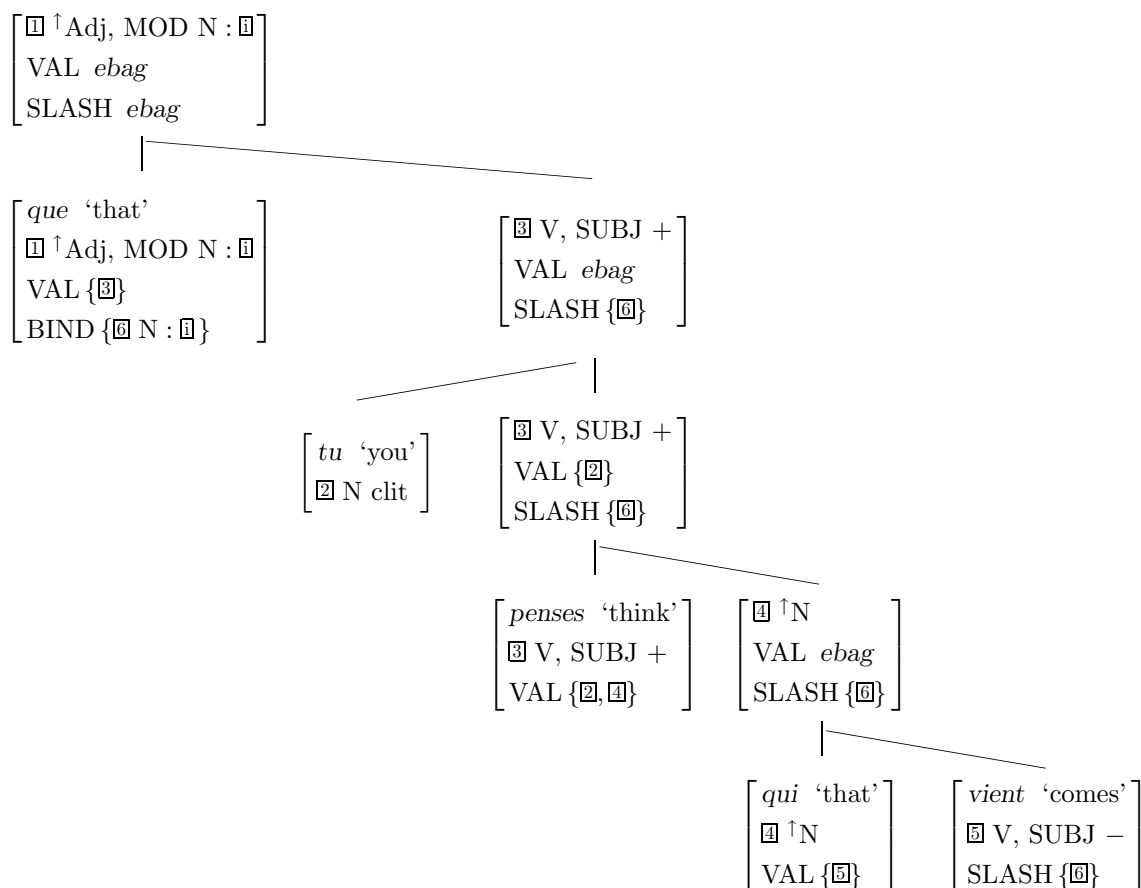
27. The analysis of Moreau (1971), also sketched by Gross (1968, p. 124), has been incorporated by Kayne (1974-75), to whom it is generally attributed. The contribution of Kayne is to defend the thesis that the relativizer and the complementizer *que* are a same word.

subject and the non-marked form *que* otherwise:

(38) **a.** *qui-form-constraint* : $\left[\,\mathrm{VAL}\,\{[\,\mathrm{SUBJ} - \,]\}\,\right]$
　　b. *que-form-constraint* : $\left[\,\mathrm{VAL}\,\{[\,\mathrm{SUBJ} + \,]\}\,\right]$

This gives us the right forms of the transferrer. In (37a), the relativizer is *qui* because it transfers a verb with a slashed subject (which is discharged by the relativizer). In (37b), the relativizer is *que* because it transfers a verb with a canonical subject. In (37c), the complementizer is *que* because it transfers a verb with a canonical subject. And in (37d), the complementizer is *qui* because the verb it transfers (*vient* 'comes') has a slashed subject and the relativizer is *que* because the verb it transfers (*pense* 'think') has a canonical subject. The whole analysis of (37d) is given in (39).

(39)　　**Analysis of the relative clause (37d)**

$$
\begin{bmatrix} \boxed{1}\,^{\uparrow}\mathrm{Adj,\ MOD\ N}:\boxed{i} \\ \mathrm{VAL}\ ebag \\ \mathrm{SLASH}\ ebag \end{bmatrix}
$$

$$
\begin{bmatrix} que\ \text{'that'} \\ \boxed{1}\,^{\uparrow}\mathrm{Adj,\ MOD\ N}:\boxed{i} \\ \mathrm{VAL}\ \{\boxed{3}\} \\ \mathrm{BIND}\ \{\boxed{6}\ \mathrm{N}:\boxed{i}\} \end{bmatrix}
\qquad
\begin{bmatrix} \boxed{3}\ \mathrm{V,\ SUBJ} + \\ \mathrm{VAL}\ ebag \\ \mathrm{SLASH}\ \{\boxed{6}\} \end{bmatrix}
$$

$$
\begin{bmatrix} tu\ \text{'you'} \\ \boxed{2}\ \mathrm{N\ clit} \end{bmatrix}
\qquad
\begin{bmatrix} \boxed{3}\ \mathrm{V,\ SUBJ} + \\ \mathrm{VAL}\ \{\boxed{2}\} \\ \mathrm{SLASH}\ \{\boxed{6}\} \end{bmatrix}
$$

$$
\begin{bmatrix} penses\ \text{'think'} \\ \boxed{3}\ \mathrm{V,\ SUBJ} + \\ \mathrm{VAL}\ \{\boxed{2},\boxed{4}\} \end{bmatrix}
\qquad
\begin{bmatrix} \boxed{4}\,^{\uparrow}\mathrm{N} \\ \mathrm{VAL}\ ebag \\ \mathrm{SLASH}\ \{\boxed{6}\} \end{bmatrix}
$$

$$
\begin{bmatrix} qui\ \text{'that'} \\ \boxed{4}\,^{\uparrow}\mathrm{N} \\ \mathrm{VAL}\ \{\boxed{5}\} \end{bmatrix}
\qquad
\begin{bmatrix} vient\ \text{'comes'} \\ \boxed{5}\ \mathrm{V,\ SUBJ} - \\ \mathrm{SLASH}\ \{\boxed{6}\} \end{bmatrix}
$$

In (40), we give an interpretation of (39) in terms of DG. The representation of the valence of *penses* 'think' is justified in the next subsection. The BIND feature of the relativizer is represented by a dashed arrow that must be unified with one of the dashed arrows of the SLASH bag. This arrow points to the antecedent noun, that is, the governor of the relative clause. This noun will be a semantic argument of the verb introducing the SLASH value (here *vient* 'comes'); this semantic argument has no syntactic counterpart in the clause.

(40)   **Dependency interpretation of (39)**
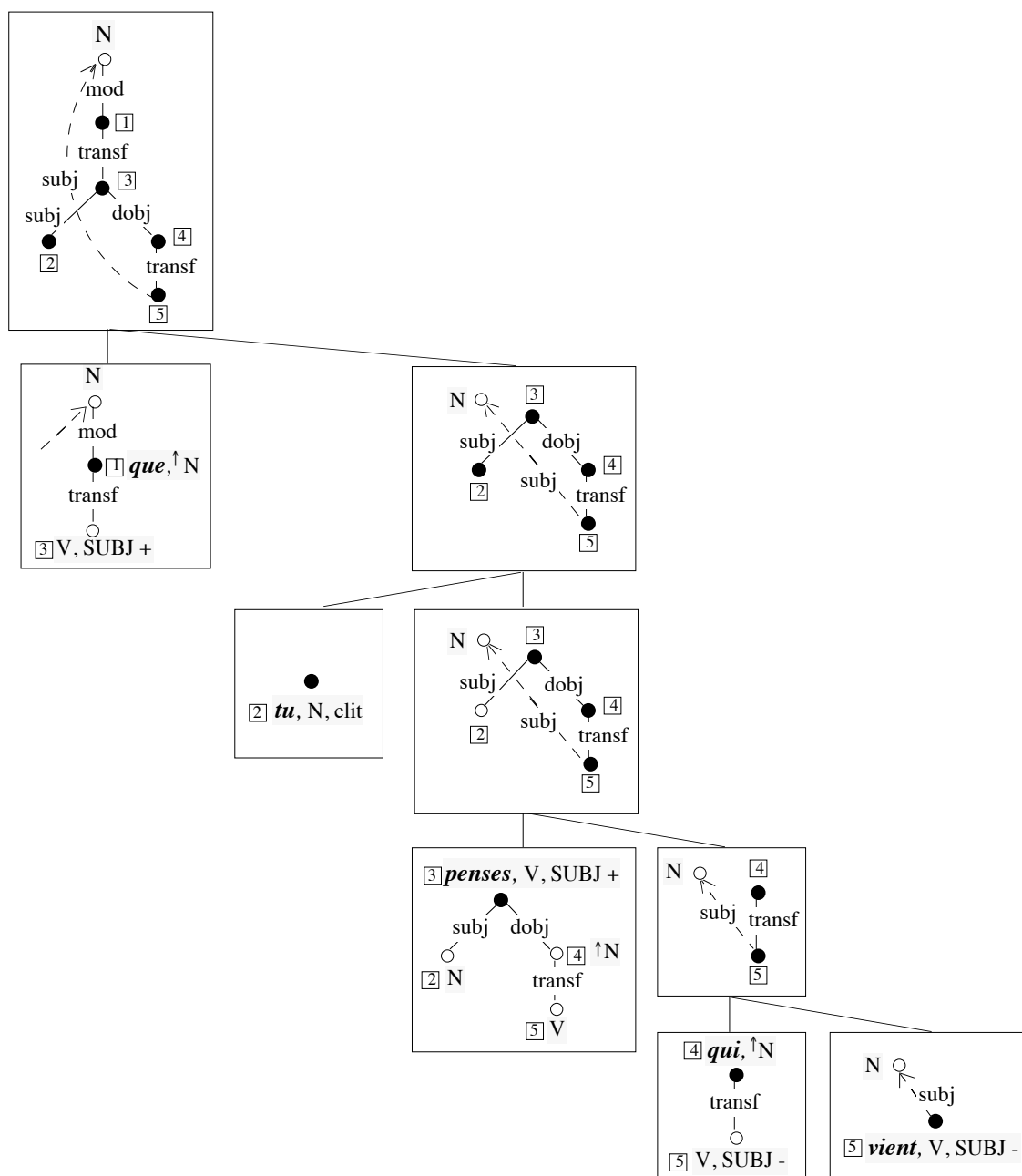
3.4. Transferrers and the Locality Principle

Whether complementizers and transferrers in general are the head of the transferred group or not has been debated for a long time (see, for instance, Pollard & Sag 1994, p. 44). The reason is that many transferrers have no meaning and only play a syntactic role–namely the transfer of the transferee. For meaningless transferrers, the meaning of the combination of the transferrer and the transferee evidently comes from the transferee. Moreover, in some cases, government's information passes through the transferrer: for instance, among verbs that take a *that*-clause as complement, some require that the subordinated verb bears the

indicative mood (as *know* in (40a)), while others require that it bears the subjunctive mood (as *demand* in (40b)).

(41) **a.** *I know that Peter leaves today.*
    b. *I demand that Peter leave immediately.*

HPSG incorporates the Locality Principle (Pollard & Sag 1987, p. 143-44; 1994, p. 23), inherited from the Chomskyan tradition, which ensures that the HDTR cannot impose conditions on the internal structure of the NHDTR. This principle is implemented by the fact that VAL elements are only *cat* and not the whole *gram-struc*, that is, that VAL specifies only the HEAD of the NHDTR and not the whole NHDTR.

Therefore, if we want to pass information through the transferrer to the transferee, we need to raise some features of the transferee into the HEAD of the phrase obtained by the transferrer-transferee combination. One solution, followed for a long time by HPSG, is to introduce a special combination schema (the traditional *head-marker-phrase*). Another solution, more or less equivalent, is to use the standard *head-actant-phrase* but to copy the whole description of the transferee into the transferrer description. For that, we add a feature TRANSF into the HEAD value of a transferrer, whose value is the transferee:

$$(42) \qquad transferrer \Longrightarrow \begin{bmatrix} \text{HEAD}\begin{bmatrix} \text{TRANSF}\ \boxed{1} \end{bmatrix} \\ \text{VAL}\ \{\boxed{1}\} \end{bmatrix}$$

Now we can have the following description for *demand*:

$$(43) \qquad demand: \begin{bmatrix} \text{VAL}\ \left\{ \begin{bmatrix} \text{FCT}\ subj \\ \text{POS}\ noun\text{-}pos \end{bmatrix}, \begin{bmatrix} \text{FCT}\ dobj \\ \text{UPPOS}\ noun\text{-}pos \\ \text{TRANSF}\begin{bmatrix} \text{POS}\ verb\text{-}pos \\ \text{INFL}\begin{bmatrix} \text{MOOD}\ subj\text{-}mood \end{bmatrix} \end{bmatrix} \end{bmatrix} \right\} \end{bmatrix} \text{ isa } verb$$

This amounts more or less to treating the transferrer and the transferee as co-heads.[28] It can be noted that in Pollard & Sag (1994) the main verb of the *that*-clause is treated as its head, while in Sag & Wasow (1999) it is the complementizer *that*. Nevertheless, both analyses are comparable, due to the fact that in both cases information from the two daughters is raised to the mother. Tesnière himself also considered the transferee and the transferrer more or less as co-heads. In his dependency trees, transferrer and transferee are represented side by side in a bubble he calls the transfer nucleus (Fr. *nucléus translatif*), which occupies a node in the dependency tree (Tesnière 1959, p. 46, 56).

In DG's terms, as a result of (43), the direct object valence slot of the verb *demand* is not a single node, but a piece of dependency tree composed of the complementizer and its transferee (see Figure (40), the representation for Fr. *penses* 'think'). Repeating the transferee description in the transferrer HEAD bypasses the Locality Principle. If each word repeats its dependents' descriptions in its own description, by recursivity,

---

28. Coordination is another phenomenon where several daughters, the conjuncts, are treated as co-heads. Coordination cannot be satisfactorily dealt with in a pure dependency approach. Even in very simple examples such as *Peter and Mary are walking* we see that the verb agrees with none of the words *Peter*, *Mary* or *and* but with the result of their combination. In this case, it seems better to clearly introduce a non-lexical node for the coordination as it is done in HPSG. Such analyses have been proposed in dependency frameworks by Petkevič (1995) and Kahane (1997). But contrary to phrase structure interpretations, such a non-lexical node corresponds only to the list of words concerned by the coordination, that is, conjucts and coordinating conjunctions, and not to the whole coordination phrase.

each word will have in its description the description of all the words it dominates, that is, the description of the whole dependency subtree it roots.[29] But, as the repetition of a dependent description is confined to transferrers, we avoid this problem and the Locality Principle is preserved. In fact, the problem comes down to the fact that locality cannot be limited to the direct dependents (and to the governor by way of MOD). It is necessary to allow a deeper control. That is the case for a verb such as *demand* which constrains a dependent of one of its dependents.

4. Wh-extraction

We traditionally call *wh*-extraction an extraction involving a *wh*-word in English (or a similar word in another language). *Wh*-extraction contrasts with *that*-extraction by pied-piping:

(44) **a.** *the person [to whom] they dedicated the building*
    b. *I wonder [to whom] they dedicated the building*

Earlier HPSG analyses of *wh*-word extraction are particularly problematic from the viewpoint of DGs given that they resort to phrasal descriptions. In Section 4.1, we propose a description of extraction directly based on them, but adapted to the notations and notions introduced in the previous sections. In Section 4.2, we propose a new description entirely in terms of combinations of words, that is, without phrasal descriptions.

4.1. An anlysis of extraction based on earlier HPSG analyses

In earlier HPSG analyses (Pollard & Sag 1994; Sag 1997; Ginzburg & Sag 2000; Bouma *et al.* 2001), *wh*-words have rather simple descriptions: they are only distinguished from other pronouns by the fact they duplicate their semantic content in a special feature called REL for relative *wh*-words and QUE for interrogative *wh*-words. To be precise, in traditional HPSG analyses, it is the index rather than the whole semantic content which is duplicated, following the Chomskyan tradition, which, for a long time, did not really consider the meaning. The features REL and QUE are empty for other words.

(45) **a.** *wh-rel-word* : $\begin{bmatrix} \text{HEAD} \begin{bmatrix} \text{CONT} \boxed{1} \end{bmatrix} \\ \text{REL} \boxed{1} \end{bmatrix}$ isa *word*

    b. *wh-int-word* : $\begin{bmatrix} \text{HEAD} \begin{bmatrix} \text{CONT} \boxed{1} \begin{bmatrix} variable \end{bmatrix} \end{bmatrix} \\ \text{QUE} \boxed{1} \end{bmatrix}$ isa *word*

    c. *word* $\Longrightarrow$ $\begin{bmatrix} \text{REL} /none \\ \text{QUE} /none \end{bmatrix}$

Interrogative *wh*-words have a very particular semantic content we call *variable*. We will not discuss this meaning here (see Kahane & Mel'čuk (1999) for one approach).[30] Nevertheless we consider that it is this particular meaning which licenses interrogative clauses in some syntactic positions (the same meaning is

---

29. It could be possible to preserve a sort of Locality Principle by only repeating a part of the dependents' descriptions and to limit the deepth of the dependency subtree reported in a word description.

30. It must be noted that the so-called interrogative *wh*-word are not inherently interrogative and can be found in non-interrogative contexts:

(i)   **a.** *She told me to whom they dedicated the building.*
     b. *I know to whom they dedicated the building.*

introduced by the complementizers *if* and *whether*). On the contrary, relative *wh*-words have no proper
semantic content, because their CONT value is the CONT value of the antecedent noun of the relative
clause. The identification of the two values is realized by the percolation of REL. In the same way, the
semantic value of an interrogative *wh*-word is lifted via QUE in order to license an interrogative clause:[31]

(46)   **Pied-Piping Computation**

$$
phrase \Longrightarrow
\begin{bmatrix}
\text{REL}\ \boxed{1} + \boxed{3} \\
\text{QUE}\ \boxed{2} + \boxed{4} \\
\text{HDTR}
\begin{bmatrix}
\text{REL}\ \boxed{1} \\
\text{QUE}\ \boxed{2}
\end{bmatrix} \\
\text{NHDTR}
\begin{bmatrix}
\text{REL}\ \boxed{3} \\
\text{QUE}\ \boxed{4}
\end{bmatrix}
\end{bmatrix}
$$

All of the complexity of *wh*-extraction is concentrated in the final step of the analysis, the computation of
the whole clause. Recent analyses of extraction in HPSG are based on phrasal descriptions of *wh-relative-
clause* and *wh-interrogative-clause*. These two phrase descriptions inherit from two families of constraints:
head-filler-phrase and *clause*. The *clause* description constrains the HDTR to be a clause, that is a saturated
verb.[32] The *head-filler-phrase* constrains the NHDTR to be combined with an element of the SLASH of
HDTR, that is, to fill a gap. The *wh-rel-cl*, which inherits from both *clause* and *hd-fill-phrase*, constrains
the NHDTR to have a non-empty REL that will be equal to the semantic content of the antecedent noun.

(47) **a.**   *hd-fill-phrase* :
$\begin{bmatrix}
\text{SLASH}\ \boxed{1} \oplus \boxed{2} \\
\text{HDTR}\big[\ \text{SLASH}\ \boxed{1} \oplus \{\boxed{3}\}\ \big] \\
\text{NHDTR}
\begin{bmatrix}
\text{HEAD}\ \boxed{3} \\
\text{SLASH}\ \boxed{2}
\end{bmatrix}
\end{bmatrix}$ isa *phrase*

b.   *clause* :
$\begin{bmatrix}
\text{VAL}\ ebag \\
\text{HDTR}
\begin{bmatrix}
\text{HEAD}\ \text{V} \\
\text{VAL}\ ebag
\end{bmatrix}
\end{bmatrix}$ isa *phrase*

c.   *wh-rel-cl* :
$\begin{bmatrix}
\text{HEAD}
\begin{bmatrix}
\text{UPPOS}\ adj\text{-}pos \\
\text{FCT}\ mod \\
\text{MOD N} : \boxed{1}
\end{bmatrix} \\
\text{NHDTR}\big[\ \text{REL}\ \boxed{1}\ \big]
\end{bmatrix}$ isa *clause* + *hd-fill-phrase*

The *head-filler-phrase* is usually considered as a *headed-phrase* (defined in (12)). It follows, with the *clause*
constraint, that *wh*-clauses are considered as clauses, that is, as phrases headed by a verb. This is not

---

c.   *To whom they dedicated the building does not matter.*

An interrogative *wh*-word introduces a unspecified meaning we can compare to a free mathematical variable. It is the context–the
rising intonation in direct questions or the meaning of the governor verb in indirect questions–that requires to instantiate this variable
and gives the interrogative interpretation.

31. The operation + is defined as follow: *none*+*none*=*none*, *none*+[...]=[...]. The combination of two non-empty values fails. In other
words, only one non empty value is allowed. We do not consider languages that authorize several extraposed *wh*-words.

32. Ginzburg & Sag (2000) consider different types of clauses for relatives and interrogatives. We do not retain this idea, which is an
attempt to preserve the fact that interrogative clauses are *hd-phrase*. See our solution below.

compatible with the Chomskyan CP analysis of *wh*-clauses nor with the Tesnière transfer theory, which predicts that an interrogative clause, which is a verb actant, must be UPPOS N and that a relative clause, which modifies a noun, must be UPPOS Adj. The analysis can be adapted to our theoretical framework by considering that *head-filler-phrase* is not a *headed-phrase* but a verb-transfered phrase:[33]

$$(48) \qquad \textit{hd-fill-phrase} \Longrightarrow \begin{bmatrix} \text{HEAD}\begin{bmatrix} \text{TRANSF} \boxed{1} \end{bmatrix} \\ \text{HDTR}\begin{bmatrix} \text{HEAD} \boxed{1} \end{bmatrix} \end{bmatrix}$$

This alternative analysis is particularly adapted for the treatment of interrogative clauses. We can now propose that the semantic content of a *wh-int-clause* comes from the *wh*-word:

$$(49) \qquad \textit{wh-int-cl} : \begin{bmatrix} \text{HEAD}\begin{bmatrix} \text{UPPOS} \;\; \textit{noun-pos} \\ \text{CONT} \boxed{1} \end{bmatrix} \\ \text{NHDTR}\begin{bmatrix} \text{QUE} \boxed{1} \end{bmatrix} \end{bmatrix} \;\; \text{isa} \; \textit{clause} \; + \; \textit{hd-fill-phrase}$$

The presence of a *variable* meaning allows us to ensure the right distribution of the interrogative clause. Moreover, as argued extensively by Kahane & Mel'čuk (1999), the semantic content of the *wh*-word is the semantic argument of the verb governing the interrogative clause.

The variant of Sag's (1997) analysis we have proposed brings us back to the analysis of relatives proposed in Pollard & Sag (1994, p. 212), where *wh-rel-clause* is obtained by the combination of the *hd-fill-clause* with a null relativizer (with a substantial difference, given that the relativizer is not considered as a head). This analysis does not discharge a slashed element in the *hd-fill-phrase*, but instead discharges it with the BIND feature of the null relativizer (which is the same null relativizer as in *that*-less relatives). Such an analysis, inspired by Chomskyan analyses (see Sag (1997, p. 433) for a criticism), is not very motivated for languages other than English where the deletion of the relativizer is not possible.

Let us come to our point: what are the drawbacks of these analyses?

First, these analyses use complex *ad hoc* phrasal descriptions. Much information is added via *wh-rel-clause* and *wh-int-clause* and we do not know from which words it comes and why. Independently of the fact that such an approach is incompatible with a DG, we can suppose we have lost generalizations somewhere.

Second, we can observe that the semantic content of *wh*-clauses does not come from the HDTR but from the *wh*-word: the semantic content is lifted via the features REL or QUE and the distributional properties of the clause (the fact that the relative clause can modify a noun or the fact that the interrogative can be the object of an interrogative verb) are no longer controlled by the HDTR (the main verb of the clause).

In fact, the combination of the filler daughter and the head daughter (the rest of the relative clause) depends, for the filler daughter, both on its head (which will govern the gap) and on the presence of a wh-word in this phrase (which controls the distribution of the clause) and, for the head daughter, both on its head (the main verb of the clause) and on the word that governs the gap (that is, that has introduced the content of the slash feature). We will show that this phrase description effectively involves four words and the simultaneous combination of two couples of words: the *wh*-word with the main verb of the relative/interrogative clause and the governor of the gap with the head of the filler phrase.

---

33. In the *head-filler-phrase* the two subphrases are called HDTR and NHDTR, but these notations are not very signifcant here, providing that no one actually is a head daughter.

4.2. A more lexical treatment of wh-extraction

The treatment of *wh*-extraction we propose is again based on an original idea of Tesnière, which has been succesfully implemented in a dependency unification grammar based on Meaning-Text Theory (Kahane 1996, 1997, 2001), in TAG (Kahane *et al.* 2000) and in DTG, a TAG-related formalism, (Candito & Kahane 1998).[34] Tesnière (1959, p. 561) postulates that a *wh*-word has two roles, a transferrer role and a pronominal role, and it behaves as two words simultaneously. In Tesnière's dependency trees, the *wh*-word occupies two nodes. For instance, in (50b), the French relative pronoun *qui* occupies two nodes called *qu-* and *qui*.[35] The *qu-* node of the pronoun *qui* corresponds to the transferrer role and is common to all French *qu*-words (pronouns and pure transferrers). In the same way, an English *wh*-word occupies two nodes; all *wh*-words them have a common part *wh-* which behaves as the pure transferrer *that*.

(50) **a.**   *la   personne à   qui   je veux parler*
              the person   to whom I   want to-speak

   b.   Dependency tree à la Tesnière for (a)
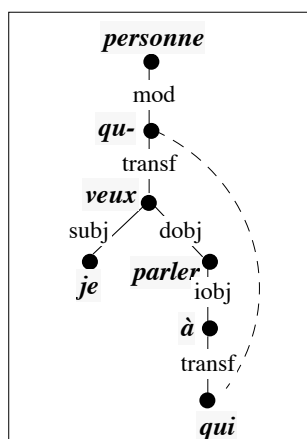
In order to introduce two descriptions of nodes in the description of a *wh*-pronoun, we add a feature SUPERHEAD (SRHD), whose value is a *word*. The pronominal part of the *wh*-word is associated to the core of word description (= the main HEAD feature), while the transferrer part of the *wh*-word is associated to the superhead (= the HEAD feature of SRHD). By default, a word has an empty SRHD.

(51) **a.**   *wh-word* : $\begin{bmatrix} pronoun \\ \text{SRHD } \textit{verb-transferrer} \end{bmatrix}$ isa *word*

   b.   *word* $\Longrightarrow \begin{bmatrix} \text{SRHD } /none \end{bmatrix}$

---

34. Questions similar to the questions studied here arise in Lexicalized TAG. LTAG is a fully lexicalized grammar which does not use phrasal descriptions: all of the information must be attached to elementary structures (= word descriptions). In traditional analyses in TAG, the ability of relative clauses to modify a noun (given by the *wh-rel-clause* constraint in HPSG) is attached to the main verb of the relative clause (Kroch & Joshi 1986; Joshi 1987; XTAG 1995), strongly increasing the number of elementary structures associated with a verb entry. A simpler and more powerful analysis treating *wh*-words as transferrers, comparable to the analysis proposed here, is possible (Kahane *et al.* 2000).

35. The relative pronoun *qui* is a different word from the verb-transferer *que/qui* presented Section 3.2. This one is +human and can be used only after a preposition.

There are four subtypes of *wh-word* corresponding to relative pronouns, interrogative pronouns in indirect questions, interrogative pronouns in direct questions and pronouns in so-called headless relatives.[36] Only the two first subtypes are considered here:

(52) **a.** *rel-wh-word* :
$$\begin{bmatrix} \text{HEAD}\,[\,\text{CONT}\ \boxed{i}\,] \\[4pt] \text{SRHD}\begin{bmatrix} \text{HEAD}\begin{bmatrix} \text{FCT}\ \textit{mod} \\ \text{UPPOS}\ \textit{adj-pos} \\ \text{MOD}\ \text{N}:\boxed{i} \end{bmatrix}\end{bmatrix} \end{bmatrix}\ \text{isa } \textit{wh-word}$$

b. *int-wh-word* :
$$\begin{bmatrix} \text{HEAD}\,[\,\text{CONT}\ \boxed{i}\,[\,\textit{variable}\,]\,] \\[4pt] \text{SRHD}\begin{bmatrix} \text{HEAD}\begin{bmatrix} \text{UPPOS}\ \textit{noun-pos} \\ \text{CONT}\ \boxed{i} \end{bmatrix}\end{bmatrix} \end{bmatrix}\ \text{isa } \textit{wh-word}$$

It can be noted that the SUPERHEAD of *rel-wh-word* and *int-wh-word* is similar to *relativizer* and *complementizer*, respectively. The semantic cohesion between the relative or interrogative clause and the rest of the sentence is ensured by the *wh*-word and the meaning $\boxed{i}$ which appears in the two-head descriptions of the word and hence will make the connection between the two parts of the sentence.

We will now show how the above description of *wh*-words can be interpreted in terms of dependency grammars. The two parts of a *wh*-word (the main HEAD and the HEAD of SRHD) correspond to two nodes in the dependency tree, as proposed by Tesnière. We propose in Figures (53) graphical representations of *int-wh-word* and *rel-wh-word*. The dashed line linking two nodes indicates that these two nodes are two parts of the same word and that they must be merged at the morphological level.

It must be noted that our analysis (and therefore Tesnière's description of *wh*-words) does not imply that we consider a *wh*-word to be the merging of two words: a *wh*-word is well and truly a single word. We simply consider that the *wh*-word has two roles–transferrer and pronoun–and it is why it occupies two positions in the dependency tree. Even these two positions are not very significant: what is important in our analysis is the fact that the *wh*-words is governed in two different ways; it is only because we impose to the dependency structure to be a tree (and each node of the structure to have at most one governor) that we need to associate the *wh*-word to two nodes. Note also that the constraints on extraction can be in a large part imputed to the double role of the *wh*-word: it is because a single word must both transfer the clause and fill a gap that the way between the head of the clause and the gap is not free. Ideally the gap should be governed by the head of the transferred clause, allowing for the *wh*-word to fill its two roles directly and it is not so far from the truth (as argued in Kahane 2001).

(53) **a. Dependency interpretation of** *int-wh-word*
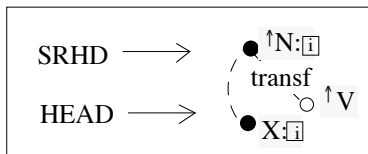
b. Dependency interpretation of *rel-wh-word*

---

36. The so-called headless relatives are traditionaly described as relatives with an empty antecedent. There are good reasons to consider that they are more closely related to indirect interrogatives than relatives (LeGoffic 1993).
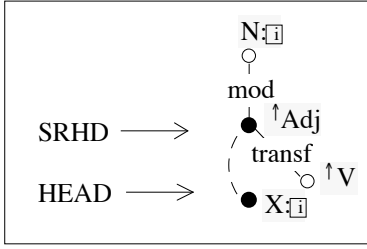
Our analysis works in the following way. The feature SUPERHEAD merges the features QUE and REL of the earlier analyses.[37] Like them, it is a foot feature:

(54)   **Superhead Computation**

$$hd\text{-}phrase \Longrightarrow \begin{bmatrix} \text{SRHD}\,\boxed{1} + \boxed{2} \\ \text{HDTR}\big[\,\text{SRHD}\,\boxed{1}\,\big] \\ \text{NHDTR}\big[\,\text{SRHD}\,\boxed{2}\,\big] \end{bmatrix}$$

Only certain syntactic relations allow pied-piping. To control that, we define a subtype of *syntrel* called *whrel*. A word which does not have a syntactic function of type *whrel* cannot allow for SUPERHEAD to percolate and hence must have an empty SUPERHEAD. This gives us the following constraint on *gram-struc*:

(55)   **Pied-Piping Constraint**

$$\begin{bmatrix} gram\text{-}struc \\ \text{SRHD } verb\text{-}transferrer \end{bmatrix} \Longrightarrow \big[\,\text{HEAD}\big[\,\text{FCT } whrel\,\big]\,\big]$$

To combine the filler phrase with the rest of the clause, we use a *filler-gap-phrase*. When this phrase is a *hd-phrase*, it gives the traditional *hd-fill-phrase*, which is still needed for topicalization. As the traditional *hd-fill-phrase*, the *fill-gap-phrase* constrains the NHDTR to be combined with an element of HDTR|SLASH and thus to fill a gap.[38]

(56) **a.** *fill-gap-phrase* : $\begin{bmatrix} \text{SLASH}\,\boxed{1} \oplus \boxed{2} \\ \text{HDTR}\big[\,\text{SLASH}\,\boxed{1} \oplus \{\boxed{3}\}\,\big] \\ \text{NHDTR}\begin{bmatrix} \text{HEAD}\,\boxed{3} \\ \text{SLASH}\,\boxed{2} \end{bmatrix} \end{bmatrix}$ isa *phrase*

   b. *hd-fill-phrase* isa *fill-gap-phrase* + *hd-phrase*

We introduce another subtype of *filler-gap-phrase* we call *superhead-filler-phrase* which is not headed by the HDTR but by the SUPERHEAD (see (57a)). Moreover, the SUPERHEAD takes the HDTR as actant (as

---

37. Pollard & Sag (1994, p. 159) justify introducing two distinct features QUE and REL instead of a single *wh*-feature by the following contrast between interrogative and relative clauses:

(i)   **a.**   *This is the farmer pictures of whom appeared in Newsweek.*

     b. * *Pictures of whom appeared in Newsweek?*

Whatever the reasons of this contrast, it can still be handled in our analysis given that *rel-wh-word* and *int-wh-word* have distinct SUPERHEAD values.

38. We continue to take the filler phrase as NHDTR, but it is clear that it is once again not very significant.

transferee to be exact). Therefore, *srhd-fill-phrase* inherits from a Valence Computation schema comparable to the standard *hd-act-phrase* in (13a), but this one involves SUPERHEAD and HDTR, rather than HDTR and NHDTR (see (57b)).

(57) **a. Alternative Head Computation (Head from Superhead)**

$$
\textit{srhd-fill-phrase} : \begin{bmatrix} \text{HEAD}\,\boxed{1} \\ \text{SRHD } \textit{none} \\ \text{NHDTR}\big[\,\text{SRHD}\big[\,\text{HEAD}\,\boxed{1}\,\big]\,\big] \end{bmatrix} \text{isa } \textit{fill-gap-phrase}
$$

b. Alternative Valence Computation (Superhead-Head Combination)

$$
\textit{srhd-fill-phrase} \Longrightarrow \begin{bmatrix} \text{VAL}\,\boxed{1} \\ \text{HDTR}\begin{bmatrix} \text{HEAD}\,\boxed{2} \\ \text{VAL } \textit{ebag} \end{bmatrix} \\ \text{NHDTR}\big[\,\text{SRHD}\big[\,\text{VAL}\,\boxed{1}\oplus\{\boxed{2}\}\,\big]\,\big] \end{bmatrix}
$$

Let us explain what happens: the two parts of a *wh*-word act alternatively as head. At the beginning, for the pied-piping, only the pronominal head is recognized and the SUPERHEAD is simply percolated (as a SLASH feature). At the last step, the SUPERHEAD becomes the head and the extracted phrase (= filler phrase) acts as a verb-transferrer. The extracted phrase in a relative becomes equivalent to the relativizer (29):

(58) $\quad \begin{bmatrix} \text{HEAD}\,\boxed{2} \\ \text{SRHD}\big[\,\text{HEAD}\,\boxed{1}\,\big] \end{bmatrix} \equiv \begin{bmatrix} \text{HEAD}\,\boxed{1} \\ \text{BIND}\,\boxed{2} \end{bmatrix}$

The unification of the three constraints of *srhd-fill-phrase* gives us (59).

(59) $\quad \textit{srhd-fill-phrase} = \begin{bmatrix} \text{HEAD}\,\boxed{1} \\ \text{VAL}\,\boxed{6} \\ \text{SLASH}\,\boxed{4}\oplus\boxed{5} \\ \text{SRHD } \textit{none} \\ \text{HDTR}\begin{bmatrix} \text{HEAD}\,\boxed{2} \\ \text{VAL } \textit{ebag} \\ \text{SLASH}\,\boxed{4}\oplus\{\boxed{3}\} \end{bmatrix} \\ \text{NHDTR}\begin{bmatrix} \text{HEAD}\,\boxed{3} \\ \text{VAL } \textit{ebag} \\ \text{SLASH}\,\boxed{5} \\ \text{SRHD}\begin{bmatrix} \text{HEAD}\,\boxed{1} \\ \text{VAL}\,\boxed{6}\oplus\{\boxed{2}\} \end{bmatrix} \end{bmatrix} \end{bmatrix}$

Note that, given that SUPERHEAD is a verb-transferrer, we obtain the equivalent of the *clause* constraint (47), that is, that $\boxed{6} = \textit{ebag}$ and $\boxed{2} = \big[\,\text{UPPOS } \textit{verb-pos}\,\big]$. We give only the last step of the computation involving the *srhd-fill-phrase*.

From the DG point of view, *srhd-fill-phrase* corresponds to two simultaneous combination of words: the NHDTR's SUPERHEAD with the HDTR and the NHDTR with an element of HDTR|SLASH:
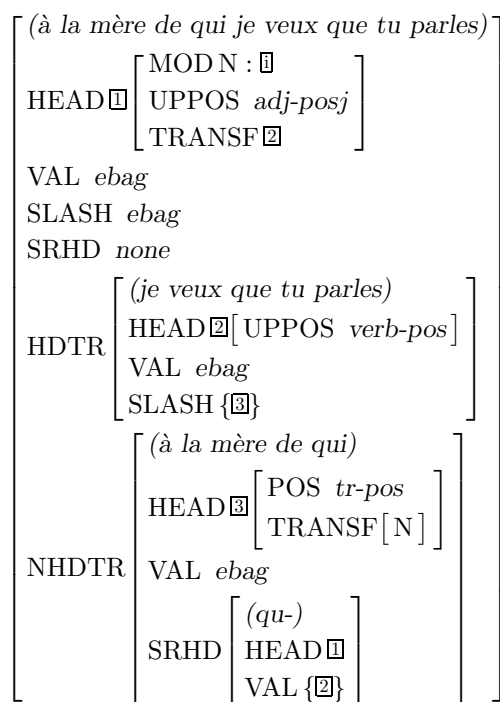
(60) **Dependency interpretation of** *srhd-fill-phrase*
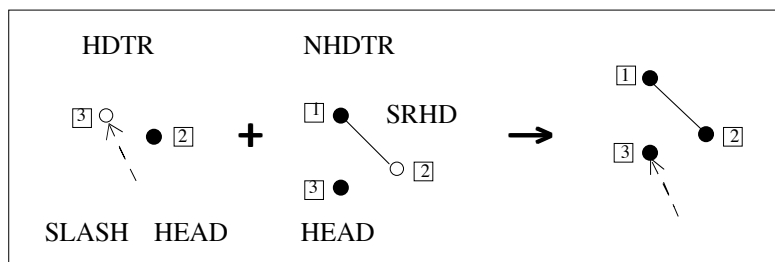
We will exemplify the *srhd-fill-phrase* with the computation of the French relative clause in (61). The analysis of (61) is given in (62) and its dependency interpretation, in (63).

(61)　　*la　personne à　la　mère　　de qui　　je veux que　tu　parles*
　　　　　the person　　to the mother of　whom I　want that you speak

　　　　　'the person whose mother I want you to speak to'

(62)　　**An example of** *srhd-fill-phrase*

$$
\begin{bmatrix}
\textit{(à la mère de qui je veux que tu parles)} \\
\text{HEAD}\,\boxed{1}\begin{bmatrix} \text{MOD N}:\boxed{1} \\ \text{UPPOS } \textit{adj-posj} \\ \text{TRANSF}\,\boxed{2} \end{bmatrix} \\
\text{VAL } \textit{ebag} \\
\text{SLASH } \textit{ebag} \\
\text{SRHD } \textit{none} \\
\text{HDTR}\begin{bmatrix} \textit{(je veux que tu parles)} \\ \text{HEAD}\,\boxed{2}\begin{bmatrix} \text{UPPOS } \textit{verb-pos} \end{bmatrix} \\ \text{VAL } \textit{ebag} \\ \text{SLASH } \{\boxed{3}\} \end{bmatrix} \\
\text{NHDTR}\begin{bmatrix} \textit{(à la mère de qui)} \\ \text{HEAD}\,\boxed{3}\begin{bmatrix} \text{POS } \textit{tr-pos} \\ \text{TRANSF}\begin{bmatrix} \text{N} \end{bmatrix} \end{bmatrix} \\ \text{VAL } \textit{ebag} \\ \text{SRHD}\begin{bmatrix} \textit{(qu-)} \\ \text{HEAD}\,\boxed{1} \\ \text{VAL } \{\boxed{2}\} \end{bmatrix} \end{bmatrix}
\end{bmatrix}
$$

We will finish this section by recapitulating our treatment of constraints on extraction and pied-piping. We have defined three subtypes of syntactic relations: *gaprel*, *slashrel* and *whrel*; *gaprels* allow for the dependent to be a gap, that is, the dependent to be slashed; *slashrels* allow the extraction, that is, an non empty SLASH to percolate; and *whrel* allow the pied-piping, that is, a non empty SUPERHEAD to percolate. For the major syntactic relations of English, we obtain the inheritance graph (64):

(63) **Dependency interpretation of (62)**
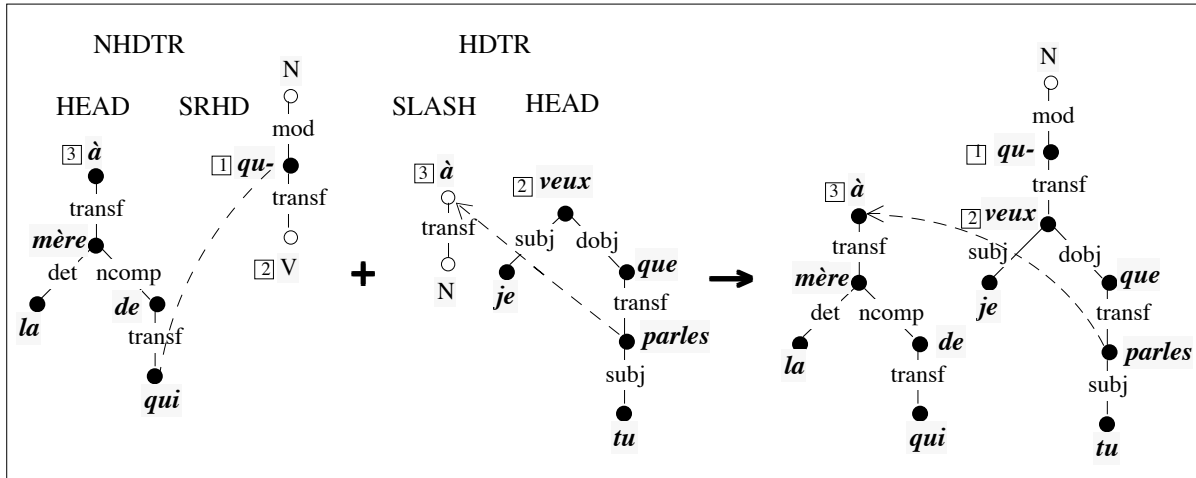
(64) **Syntactic relations' inheritance graph for English**
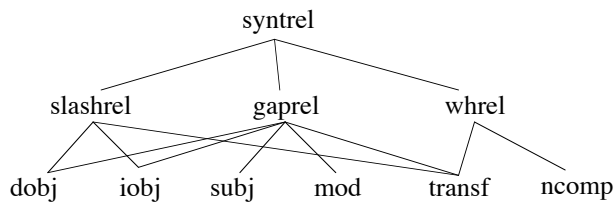
This completes the presentation of our alternative treatment of extraction in HPSG. We hope that we have convinced the reader that this description using only the combination of word descriptions and no phrasal descriptions works at least as well as earlier approaches.

5. Status of phrases in the syntax-semantics interface of HPSG

In this paper, we have argued that, with no substantial modifications, the syntax-semantics interface of HPSG can be viewed as a true dependency grammar, that is, a grammar where all of the information about sentences comes from the information on words and their combination.[39] We have shown that even puzzling phenomena such as extraction can be described just in terms of combinations of words. In such a grammar, no linguistic information is attached to objects other than words and in particular, no linguistic information is attached to objects such as phrases. (It is no longer the case in the syntax-phonology interface, where linearization and grouping for prosody are handled.)

Adopting our view of syntax, it can be said that syntactic phrases are simply the result of particular word combinations procedures. Indeed, HPSG is based on a bottom-up process: HPSG allows for a word to combine with a dependent word only when this dependent has combined with all of its own dependents (and

---

39. As said previously, it would be better to handle morphemes rather than words (including multi-word morphemes such as idioms) and to obtain word descriptions by combinations of morpheme descriptions. This is another point, which cannot be discussed here, where we think that the architecture of HPSG is not completely convenient, due to the fact that grammatical morphemes are dealt with by lexical rules and therefore are not considered as signs.

its VAL is empty). In other words, the underlying tree of an HPSG analysis is built from the leaves to the root. Such a procedure imposes the restriction that a word can combine with its governor only when its projection has been processed. Consequently, we claim that syntactic phrases are simply the result of a particular way of processing and that HPSG contains in itself this way of processing, which might be avoided.

It must be noted that a syntactic "representation" is always linked to processing. The syntactic dependency tree stores how words combine each other; a syntactic dependency witnesses the combination of two words and indicates which one is the HDT (= governor) and which one is the NHDTR (= dependent). Nevertheless the dependency treee is less procedural than the phrase structure tree: it does not store in which order words have been combined during the process. And we think that this last information is not relevant from a linguistic point of view.

We will now see that, even with a formalism such as HPSG, it is possible to use a strategy other than the bottom-up process. For instance, dealing with a sentence such as *Peter wants to eat beans*, we can begin to combine *wants* and *to eat*. For that, we must allow for a word (here *wants*) to combine with a dependent (here *to eat*) whose valence is not filled. This can be done with a schema such as (65).[40]

(65)   *hd-act-phrase* : $\begin{bmatrix} \text{VAL}\,\boxed{1} \oplus \boxed{3} \\ \text{HDTR}\big[\,\text{VAL}\,\boxed{1} \oplus \{\boxed{2}\}\,\big] \\ \text{NHDTR}\begin{bmatrix} \text{HEAD}\,\boxed{2} \\ \text{VAL}\,\boxed{3} \end{bmatrix} \end{bmatrix}$ isa *hd-phrase*

Here we do not require the VAL(ENCE) of NHDTR to be empty; this valence is added to the valence of HDTR. This idea directly comes from categorial grammars (see the Forward Partial Combination rule of Ades & Steedman 1982, p. 527). The SLASH feature has been introduced to allow such a combination; we distinguish the VAL coming from HDTR from the VAL coming from NHDTR, which is put in SLASH. Such a schema should be constrained in the same way as SLASH introduction and SLASH percolation is constrained. It could be useful for dealing with phenomena such as gapping (*Peter wants to eat beans and Mary, spaghetti*) or word order in languages such as German, where verbs can form a cluster, the order of their dependents being free (*dass Peter Bohnen essen will* 'that Peter beans eat wants' vs. *dass Bohnen Peter essen will*).[41]

The use of phrases has been defended for the treatment of word order. But in non-configurational languages such as Russian or German, observable groupings do not correspond to syntactic phrases (= projections of heads) and another structure must be invoked. For example, in the description of German, a structure other than phrase structure, the topological model, is used to compute word order (see Kathol 1995 for a formalization in HPSG and Donohue & Sag 1999 for an application to Warlpiri). We do not deny that it could be useful to introduce particular groupings to compute word order. But in any case, the introduction of a particular type of grouping must be assumed only for the treatment of word order and prosody.

We will conclude by showing that even the SLASH feature, which seems to be indispensable for the HPSG analysis of extraction, results from the underlying process of HPSG, which consists of combining words

---

40. It is not the first time one proposes to introduce such a schema in HPSG. See, for instance, the handling of auxiliaries of Hinrichs & Nakazawa (1994).

41. Word order is actually never free. But it can depend on factors such as information structure and be not constrained by purely syntactic properties such as the height of the complements *Peter* and *Bohnen* 'beans' in the dependency tree (contrary to Dutch, which is a closely related language).

according to the traditional phrase structure. Consider the topicalized sentence in (66). In HPSG, following the traditional X-bar Theory, the sentence is obtained by a combination of words giving us the phrase structure indicated in (66a).[42]

(66) **a.** *[Sandy (I know [you said (Peter hates)])]*

 b. *[([(Sandy) I know] you said) Peter hates]*

But it is possible to obtain the sentence by a combination of words giving us the grouping in (66b). In this case, we must first combine *Sandy* and *I know*. We can allow that, but as *Sandy* is not a syntactic dependent of *know*, it will be added in a special feature we call VIS(ITOR) (Hudson (1990, 2000) uses a similar relation to link the main verb with an extracted phrase). A new visitor is introduced with the *head-visitor-phrase*. In this schema, NHDTR|HEAD does not combine with HDTR|VAL but is simply added to the VIS(ITOR) bag:
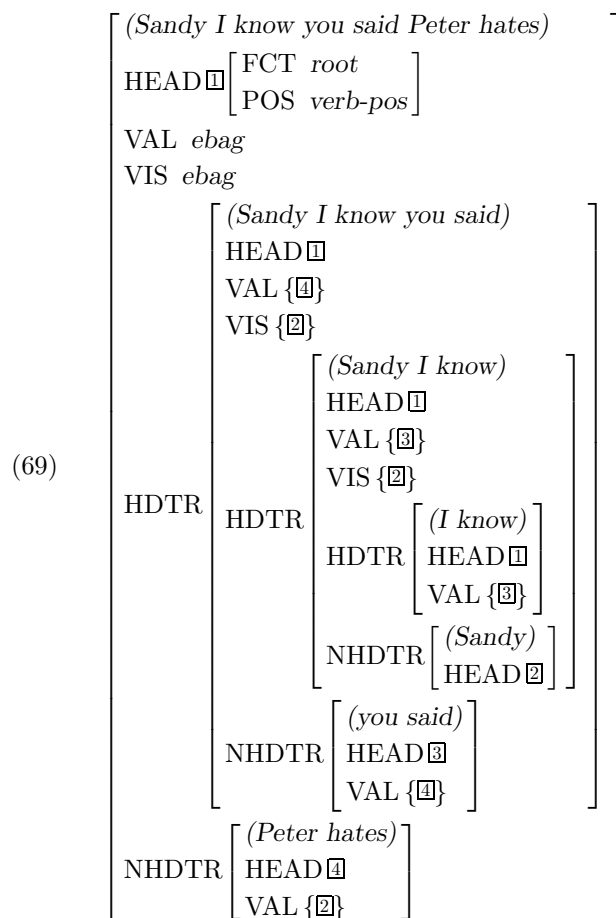
$$
(67) \quad \textit{hd-vis-phrase}: \begin{bmatrix} \text{VAL}\,\boxed{1} \\ \text{VIS}\,\boxed{2} \oplus \{\boxed{3}\} \\ \text{HDTR}\begin{bmatrix} \text{VAL}\,\boxed{1} \\ \text{VIS}\,\boxed{2} \end{bmatrix} \\ \text{NHDTR}\begin{bmatrix} \text{HEAD}\,\boxed{3} \end{bmatrix} \end{bmatrix} \quad \text{isa } \textit{hd-phrase}
$$

Next, the visitor is lifted to the verb governing the gap and fills the gap using the following schema *generalized-head-actant-phrase*:

$$
(68) \quad \textit{g-hd-fill-phrase}: \begin{bmatrix} \text{VAL}\,\boxed{1} \oplus \boxed{3} \\ \text{VIS}\,\boxed{4} \\ \text{HDTR}\begin{bmatrix} \text{VAL}\,\boxed{1} \oplus \{\boxed{2}\} \\ \text{VIS}\,\boxed{4} \oplus \boxed{5} \end{bmatrix} \\ \text{NHDTR}\begin{bmatrix} \text{HEAD}\,\boxed{2} \\ \text{VAL}\,\boxed{3} \oplus \boxed{5} \end{bmatrix} \end{bmatrix} \quad \text{isa } \textit{hd-phrase}
$$

This schema generalizes *hd-act-phrase* (65) (it is equal when $\boxed{4} = \textit{ebag}$ and $\boxed{5} = \textit{ebag}$). It allows us to lift visitors ($\boxed{4} \neq \textit{ebag}$) and fill gaps ($\boxed{5} \neq \textit{ebag}$). When *Sandy I know* and *you said* are combined, $\boxed{4} = \{Sandy\}$ and $\boxed{5} = \textit{ebag}$ and, when *Sandy I know you said* and *Peter hates* are combined, $\boxed{4} = \textit{ebag}$ and $\boxed{5} = \{Sandy\}$. The major lines of the VISITOR-like analysis of (66) are given in (69).

---

42. X-bar Theory encodes, via the phrase structure and the c-command, the syntactic relations between words. But X-bar phrase structure is not useful for that in HPSG, where the syntactic relations between words are encoded in VAL and MOD. We hope that this point, extensively developed in Section 2, is absolutely clear for the reader.

(69)

$$
\begin{bmatrix}
\textit{(Sandy I know you said Peter hates)} \\
\text{HEAD}\,\boxed{1}\begin{bmatrix}\text{FCT } \textit{root} \\ \text{POS } \textit{verb-pos}\end{bmatrix} \\
\text{VAL } \textit{ebag} \\
\text{VIS } \textit{ebag} \\
\text{HDTR}\begin{bmatrix}
\textit{(Sandy I know you said)} \\
\text{HEAD}\,\boxed{1} \\
\text{VAL }\{\boxed{4}\} \\
\text{VIS }\{\boxed{2}\} \\
\text{HDTR}\begin{bmatrix}
\textit{(Sandy I know)} \\
\text{HEAD}\,\boxed{1} \\
\text{VAL }\{\boxed{3}\} \\
\text{VIS }\{\boxed{2}\} \\
\text{HDTR}\begin{bmatrix}\textit{(I know)} \\ \text{HEAD}\,\boxed{1} \\ \text{VAL }\{\boxed{3}\}\end{bmatrix} \\
\text{NHDTR}\begin{bmatrix}\textit{(Sandy)} \\ \text{HEAD}\,\boxed{2}\end{bmatrix}
\end{bmatrix} \\
\text{NHDTR}\begin{bmatrix}\textit{(you said)} \\ \text{HEAD}\,\boxed{3} \\ \text{VAL }\{\boxed{4}\}\end{bmatrix}
\end{bmatrix} \\
\text{NHDTR}\begin{bmatrix}\textit{(Peter hates)} \\ \text{HEAD}\,\boxed{4} \\ \text{VAL }\{\boxed{2}\}\end{bmatrix}
\end{bmatrix}
$$

This last analysis has the advantage of allowing us to parse the sentence incrementally (that is, to build its dependency tree and its semantic structure as soon as possible) and is therefore better motivated cognitively. Contrary to the SLASH analysis, no lexical rule (or an equivalent device) is needed to introduce the VISITOR. Finally, this analysis is compatible with the SUPERHEAD analysis presented in Section 4.2; it simply need to replace the *head-visitor-phrase* by a *superhead-visitor-phrase* where NHDTR|SRHD takes HDTR as actant (and NHDTR becomes a VISITOR).

SLASH and VISITOR are linked to two different word groupings corresponding to different processes in the combination of the words: in one case, the gap is lifted up to the filler by way of SLASH, while in the other, the filler is lowered down to the gap by the way of VISITOR. Kahane (2001) proposes a formalization that subsumes these two analyses. This more generic formalization is less procedural and does not presuppose any particular word grouping. Such a formalism is associative, in the mathematical sense of the term: it allows for the dependency to be built in whatever direction, while in the traditional HPSG the dependency tree is necessary built bottom-up, that is, from the leaves to the root.[43] The implementation of an associative analysis necessitates a serious revision of the foundations of HPSG.

---

43. Rather than lift an element from the verb introducing the gap to the main verb, Kahane (2001) analyzes the extraction path – the string of verbs (or verb-equivalent forms) – as a syntactic unit (called a *verbal nucleus*).

6. Conclusion

For HPSG users, the main result of this paper is certainly the demonstration that, in the same way than the *head-daughter-phrase* can be interpreted as tha combination of two words, the traditional *head-filler-phrase* (the schema of combination of a filler phrase and a clause with a gap) can be interpreted as a schema involving two simultaneous combinations of words where the two daughters act as heads, each one in a different combination. The filler phrase, by the way of the *wh*-word, controls the distribution of the clause (allowing it to be a relative or an interrogative clause and to modify a noun or to occupy an actancial position of an interrogative verb) and is thus the true head of the clause, but, simultaneously, it fills a gap in the clause and is thus governed by the clause. We have solved this problem by giving a double role to the *wh*-word, a pronominal role and a verb-transferrer role. Contrary to earlier analyses of extraction in HPSG, we do not introduce special schemata for each type of extraction and we do not allow for direct introduction of linguistic information in a phrasal schema.

Our study proves that phrasal descriptions are not needed in the analysis of extraction and moreover that the recourse to phrasal description masks what combinations of words are involved in extraction. From a theoretical point of view, we think that purely linguistic data might be introduced in word descriptions and that phrasal schemata might only encode how subphrases combine together.[44] This gives us a totally lexicalist version of the syntax-semantics interface of HPSG, where the phrase structure simply stores the way in which words have combined. As a consequence, phrases have lost any linguistic status; they only result from a particular procedure in words' combination and another procedure will give us another grouping of words. Of course, it is possible to give a theoretical status to the phrase structure considered by traditional HPSG and inherited from X-bar Theory: a phrase is the maximal projection of a word, that is, a word with all the phrases headed by its dependents. It is even easy to introduce another phrases by considering partial projection involving only a part of the dependents (excluding for example the subject or whatever else). What we say is that such a phrase structure is simply a by-product of the dependency structure and does not play any particular role other than a procedural one.

For DG users, the main result is a proposal of an implementation of a DG in the well-defined formalism of HPSG. To be exact, the grammar proposed here is not a genuine DG–it allows for a dependency structure that is not a true tree, which results of the double role of the *wh*-word. Such an analysis needs an extension of traditional DG formalisms and it is interesting to see that the formalism of HPSG allows elegant solutions.

For all, we hope that this paper proposes a better understanding of both HPSG and DG and particularly the way they model extraction.

Acknowledgements

---

44. It is not exactly the case that all constructions, in the sense of Goldberg (1995), are lexically marked and in some case some linguistic material can not be easily associated to a particular word. It is the case of *that*-less relative in English. But even in such cases, it seems preferable to introduce a "lexical" sign without phonological realization, rather to introduce a specific phrasal schema.

References

Ades Anthony E. & Mark J. Steedman, 1982, "On the order of words", *Linguistics and Philosophy*, *4*, 517-58.

Bloomfield Leonard, 1933, *Language*, New York.

Bouma Gosse, Robert Malouf & Ivan A. Sag, 2001, "Satisfying Constraints on Extraction and Adjuction", *Natural Language and Linguistic Theories*.

Candito Marie-Hélène & Sylvain Kahane, 1998, "Defining DTG derivations to get semantic graphs", *Proceedings TAG+4 Workshop*, Philadelphie, 25-28.

Donohue Cathryn & Ivan Sag, 1999, "Domains in Warlpiri", *HPSG 1999*, Edinburgh.

Gaifman Haim, 1965, "Dependency systems and phrase-structure systems", *Information and Control*, *18*, 304-37 ; Technical Report, Rand Corporation, 1961, RM-2315.

Gerdes Kim & Kahane Sylvain, to appear, "Phrasing it differently", Meaning-Text papers in honour of the 70th birthday of Igor Mel'čuk, Benjamins.

Ginzburg Jonathan & Ivan Sag, 2001, *Interrogative Investigations: The Form, Meaning, Use of English Interrogatives*, CSLI Publications/University of Chicago Press.

Goldberg Adele E., 1995, *A Construction Grammar Approach to Argument Structure*, University of Chicago Press.

Gross Maurice, 1968, *Grammaire transformationnelle du français. Syntaxe du verbe*, Larousse, Paris.

Jespersen Otto, 1924, *The Philosophy of Grammar*, Chicago U. Press, Chicago.

Joshi Aravind, 1987, "Introduction to Tree Adjoining Grammar", in A. Manaster Ramer (ed.), *The Mathematics of Language*, Benjamins, 87-114.

Haspelmath Martin, 1997, *Indefinite Pronouns*, Clarendon Press, Oxford.

Hays David, 1964, "Dependency theory: A formalism and some observations", *Language*, *40:4*, 511-25.

Hinrichs Erhard & Tsuneko Nakasawa, 1994, "Linearizing AUXs in German Verbal Complexes", in J. Nerbonne, K. Netter, C. Pollard (eds.), *German In Head-driven Phrase Structure Grammar*, CSLI Publications, Stanford, 11-38.

Hudson Richard, 1980, "Constituency and dependency", *Linguistics*, *18*, 179-198.

Hudson Richard, 1990, *English Word Grammar*, Oxford, Blackwell.

Hudson Richard, 2000, "Discontinuity", in S. Kahane (ed.), *Dependency Grammars*, *T.A.L.*, *41:1*, Hermès, Paris.

Kahane Sylvain, 1996, "If HPSG were a dependency grammar ...", *Proceedings TALN'96*, Marseille, 45-49.

Kahane Sylvain, 1997, "Bubble Trees and Syntactic Representations", in Becker & Krieger (eds), *Proc. 5th Meeting of the Mathematics of Language (MOL5)*, DFKI, Saarbrücken, 70-76.

Kahane Sylvain, 2001, "A Fully Lexicalized Grammar for French Based on meaning-Text Theory", in A. Gelbukh (ed.), *Computational Linguistics adn Intelligent Text Processing*, *CICLing 2001 Proceedings*, Springer ,18-31.

Kahane Sylvain, Marie-Hélène Candito & Yannick de Kercadio, 2000, "An Alternative Description of Extractions in TAG", *Proceedings of TAG+5 Workshop*, Paris, 123-28.

Kahane Sylvain & Mel'čuk Igor, 1999, "La synthèse sémantique ou la correspondance entre graphes sémantiques et arbres syntaxiques. Le cas des phrases à extraction en français contemporain", *T.A.L.*, *40:2*, 25-85.

Kahane Sylvain, Alexis Nasr & Owen Rambow, 1998, "Pseudo-Projectivity: A polynomially parsable non-projective Dependency Grammar", *Proceedings COLING-ACL'98*, Montréal, 646-52.

Kaplan Ronald & Annie Zaenen, 1989, "Long Distance Dependencies, Constituent Structure and Functional Uncertainty", in A. Baltin & A. Kroch (éds), *Alternative Conceptions of Phrase Structure*, Univ. of Chicago Press, 17-42.

Kathol Andreas, 1995, *Linearization-Based German Syntax*, Ph.D. thesis, The Ohio State University.

Kayne Richard, 1974-75, "French Relative *que*", *Recherches Linguistiques II*, 40-61, et *III*, 27-92.

Kroch Anthony & Aravind Joshi, 1986, "Analysing Extraposition in a TAG", in G. Huck & A. Ojeda (eds.), *Discontinuous Constituents*, Syntax and Semantics 20, Academic Pres, 107-49.

Kroch Anthony, 1987, "Unbounded Dependencies and Subjacency in a Tree Adjoining Grammar", in A. Manaster Ramer (ed.), *The Mathematics of Language*, Benjamins, 143-71.

Lecerf Yves, 1961, "Une représentation algébrique de la structure des phrases dans diverses langues naturelles", *C. R. Académie des Sciences de Paris*, *252*, 232-34.

LeGoffic Pierre, 2003, *Morphèmes en $*k^W$*, Verbum.

Mel'čuk Igor, 1988, *Dependency Syntax: Theory and Practice*, State Univ. of New York Press, Albany.

Moreau Marie-Lyne, 1971, "L'homme que je crois qui est venu ; qui, que : relatifs et conjonctions", *Langue Française*, *11*, 77-90.

Nasr Alexis, 1995, "A Formalism and a Parser of Lexicalized Dependency Grammars", *4th Int. Workshop on Parsing Technologies*, State Univ. of NY Press.

Padučeva Elena V., 1964, "O sposobax predstavlenija sintaksičeskoj struktury predloženija" [On methods of representing the syntactic structure of a sentence], *Voprosy jazykoznanija* [Questions of linguistics], *2*, 99-113.

Petkevič Vladimir, 1995, "A New Formal Specification of Underlying Representations", *Theoretical Linguistics*, *21*, 7-61.

Pollard Carl & Ivan A. Sag, 1987, *Information-Based Syntax and Semantics*, CSLI Publications, Stanford.

Pollard Carl & Ivan A. Sag, 1994, *Head-Driven Phrase Structure Grammar*, CSLI Series, Univ. Chicago Press.

Ross John, 1967, *Constraints on Variables in Syntax*, Ph.D. Thesis, MIT (printed as *Infinite Syntax!*, Reidel, Dordrecht, 1985).

Sag Ivan A., 1997, "English Relative Clause Constructions", *Journal of Linguistics*, *33*, 431-84.

Sag Ivan A. & Janet Dean Fodor, 1994, "Extractions Without Traces", in *Proceedings of the West Coast Conference on Formal Lingusitics*, *13*, Stanford, 365-84.

Sag Ivan A. & Thomas Wasow, 1999, *Syntactic Theory: A Formal Introduction*, CSLI Publications, Stanford.

Tesnière Lucien, 1959, *Éléments de syntaxe structurale*, Kliencksieck, Paris.

van Noord Gertjan & Gosse Bouma, 1994, "Adjuncts and the Processing of Lexical Rules", *COLING'94*, Kyoto, 250-56.

XTAG Research Group, 1995, "A Lexicalized TAG for English", Technical report IRCS 95-03, University of Pennsylvania (On line updated version).

Žolkovskij Aleksandr K. & Mel'čuk Igor A., 1967, "O semantičeskom sinteze" [On Semantic Synthesis (of Texts)], *Problemy Kibernetiky* [Problems of Cybernetics], *19*, 177-238.