Abstract This paper discusses how the English Resource Grammar (ERG) captures the optionality of certain complements of verbs based on a single lexical entry coupled with an ontology of markings distinguishing optional from obligatory as well as unrealized from realized elements. Subject-head and head-complement structures are modified accordingly, but due to the lack of a possibility to express and use relational goals in grammars implemented in the LKB system, the ERG encoding falls short of the goal of treating optional complements in a general way. Instead, it requires two new types of 'auxiliary' phrases which are otherwise unmotivated. We show that the problem can be overcome by using a recursive relation selecting a member from a list. The use of a lean implementation platform not supporting such relational goals, such as the LKB, thus results in a loss of generality of the grammars that can be expressed, which undermines the closeness of the implemented grammar to current linguistic analyses as one of the hallmarks of HPSG-based grammar implementation. The case study presented in this paper thus supports the position argued in Götz and Meurers (1997) that a system for the implementation of HPSG-based grammars should include both universal implicational principles as well as definite clauses over feature terms.

1 Introduction

The English Resource Grammar (ERG) developed by the LinGO project[1] is a freely available, broad-coverage, HPSG-based grammar of English (Flickinger et al. 2000), which is implemented in the LKB system (Copestake 2002). The grammar contains a wealth of analyses of English phenomena, many of which have not received particular attention in generative linguistics. In this short paper we want to investigate the ERG analysis of optional arguments, an issue with a linguistic basis that is relevant for grammar implementation in general. Based on a discussion of the treatment of optionality proposed in Flickinger (2000) and how it was implemented in the ERG, the paper is intended to contribute to a discussion of the choices involved in implementing HPSG analyses, and how those choices are determined by the options for expressing grammars in a given implementation platform.

2 Optional complements and their treatment in the ERG

In a paper discussing grammar writing techniques intended to improve the efficiency of processing with such grammars, Flickinger (2000) includes a sketch of a proposal for the analysis of verbs with optional complements. The empirical issue is illustrated by the sentences in (1).

[1]http://lingo.stanford.edu/

(1) a. Kim bet Tom five dollars that they hired Cindy.
b. Kim bet Tom five dollars.
c. Kim bet Tom that they hired Cindy.
d. Kim bet five dollars that they hired Cindy.
e. Kim bet five dollars.
f. Kim bet that they hired Cindy.
g. Kim bet Tom.
h. Kim bet.

In sentence (1a), the verb *bet* takes a subject *Kim* and three complements, the NPs *Tom* and *five dollars*, as well as the sentential complement *that they hired Cindy*. The other sentences in (1) exemplify that each of those three complements is optional. In (1b)–(1d) one of the complements is missing, in (1e)–(1g) two complements are missing, and in (1h) none of the complements of *bet* are realized.

The brute-force method for licensing these structures would be to posit eight independent lexical entries for *bet*, one for each of the environments exemplified above. But this would miss the generalization that *bet* has three complements, each of which can be realized or not. Following Flickinger (2000), the ERG takes this generalization into account and posits only the single lexical entry shown in figure 1.[2] The key aspect here is the specification of the complement requirements

$$\left[\begin{array}{l} \textit{non\_derived\_word} \\ \text{PHON } \langle \textit{bet} \rangle \\ \text{S} \left[\begin{array}{l} \textit{canonical\_synsem} \\ \text{L}|\text{C} \left[\begin{array}{l} \text{HEAD } \textit{verb} \\ \text{V} \left[\begin{array}{ll} \text{COMPS} & \left\langle \left[\begin{array}{l} \textit{synsem} \\ \text{L}|\text{C}|\text{H} \left[\begin{array}{l} \textit{nominal} \\ \text{CASE } \textit{acc} \end{array}\right] \\ \text{OPT } \textit{plus} \end{array}\right], \left[\begin{array}{l} \textit{synsem} \\ \text{L}|\text{C}|\text{H} \left[\begin{array}{l} \textit{nominal} \\ \text{CASE } \textit{acc} \end{array}\right] \\ \text{OPT } \textit{plus} \end{array}\right], \left[\begin{array}{l} \textit{synsem} \\ \text{L}|\text{C}|\text{H} \left[\begin{array}{l} \textit{verbal} \\ \text{VFORM } \textit{fin} \end{array}\right] \\ \text{OPT } \textit{plus} \end{array}\right] \right\rangle \\ \text{SUBJ} & \left\langle \left[\begin{array}{l} \textit{synsem} \\ \text{L}|\text{C}|\text{H } \textit{nominal} \end{array}\right] \right\rangle \end{array}\right] \end{array}\right] \end{array}\right] \end{array}\right]$$

Figure 1: Lexical entry for *bet*

on the COMPS list. The list contains three elements, each of which is marked as optional with the help of an attribute OPT(IONAL) appropriate for *synsem* objects.

In figure 2 on the next page we see the structure that is licensed for a sentence in which none of the optional complements are realized, i.e., sentence (1h). The entry of *bet* can construct as the head daughter of a head-subject phrase even though it has not yet realized its complements. This is possible since, different from the traditional HPSG analysis (Pollard and Sag 1994), the head daughter is not required to be saturated, i.e., have a a COMPS value of type *e_list*. Instead, the COMPS value of the head daughter is required to be of type *o_list*, which is a (potentially empty)

[2]Here and in the following figures, only the specifications relevant to the issue of optionality are shown. For space reasons, attribute names are sometimes abbreviated by their first letter.

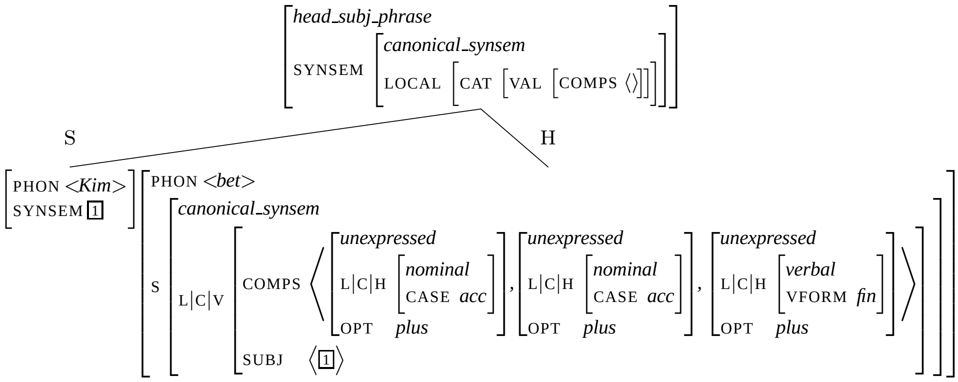

Figure 2: A sentence with three unrealized complements

list of elements, all of which are optional ([OPT *plus*]) and *unexpressed*; the relevant type constraint and parts of the list hierarchy are shown in figure 3.

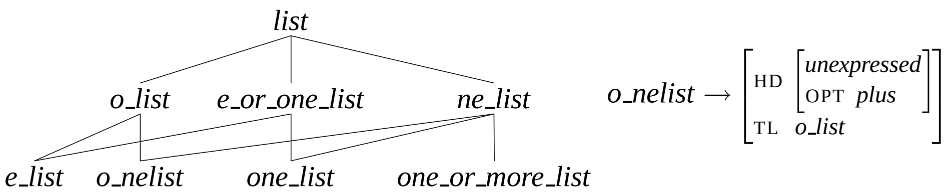

Figure 3: Introducing and constraining the *o_nelist* subtype

In plain words, a sign is understood to be saturated for complements if it either has discharged all its complement requirements (the traditional requirement) or has only optional complement requirements left, which are marked as *unexpressed*.

Adding head-complement phrases to the picture, one can also license (1b) and (1g), which are sentences in which one or two complements are realized and the other complements, which are more oblique than the ones that are realized, are missing.[3] Figure 4 shows the relevant aspects of the definition of head-complement phrases in the ERG. Note that it is always the first element of the COMPS list that is realized as the *non_head_dtr* of such a phrase.

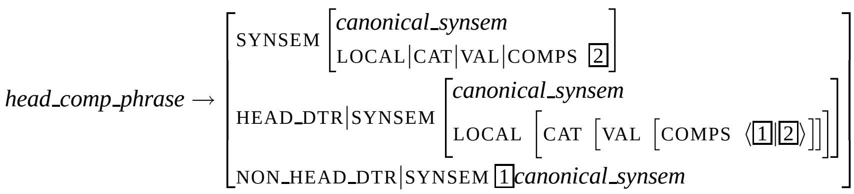

Figure 4: The realization of COMPS requirements in the head-complement rule

[3] The COMPS is ordered by obliqueness, with the least oblique complement being the first element.

Exemplifying the two types of phrases we have discussed so far, figure 5 shows the structure that the ERG assigns to the sentence (1g).

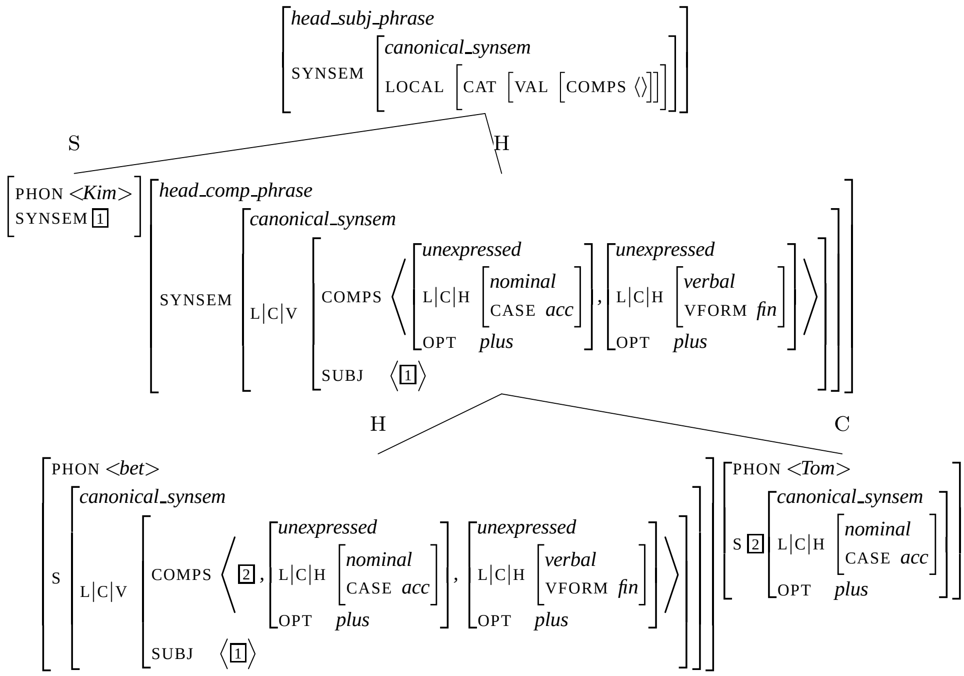

Figure 5: A sentence in which the two most oblique complements are not realized

The lower tree is an instance of a *head_comp_phrase*, in which the first subcategorization requirement on COMPS, namely the NP *Tom* bearing the tag $\boxed{2}$, is realized. The *head_subj_phrase* on top is licensed just as in the previous example, marking the remaining optional elements on the COMPS list of the head daughter *bet Tom* as unexpressed.

Since the *head_comp_phrase* in the ERG always realizes the first element of the COMPS list, a problem arises if one wants to license a sentence in which the least oblique complement, i.e., the first element on the COMPS list is optional and not realized. Note that this is not an accidental oversight in the formulation of the rule licensing *head_comp_phrases* in the ERG; rather it is a consequence of the fact that the LKB system does not support relational goals as attachment to phrase structure rules. In HPSG linguistics such relational goals are used extensively, most prominently to concatenate valence or phonology lists using the append relation, which in the AVM notation is often specified using the $\oplus$ infix-operator. We will see in the next section that when such relational goals are included in the expressive means available to the grammar writer, one can express the proper generalization for the optional argument case: the *head_comp_phrase* realizes the first requirement on COMPS which is not marked as unrealized optional element. In the ERG as implemented in the LKB system, the problem is addressed by introducing additional types of phrases which eliminate the unrealized optional subcategorization requirements from the front of the COMPS list in order to bring the requirement intended to be realized to the first position of the COMPS list. For this purpose, in addition to the ordinary *head_comp_phrases*, the ERG uses two additional rules: the *head_opt_comp_phrase* which eliminates one optional complement from the front of the COMPS list, and the *head_opt_two_comp_phrase* which eliminates the first two complement requirements from the COMPS list. Further additional phrases would be needed if the grammar had COMPS lists longer than three; this also makes the approach inappropriate for languages exhibiting coherence or restructuring phenomena (e.g., German, Dutch, and the Romance languages) given that under the standard HPSG argument-attraction analyses of those languages, the number of elements on COMPS is not bounded in the lexicon.

Figure 6 illustrates the structure licensed for sentence (1e), in which only the second most oblique complement is realized.

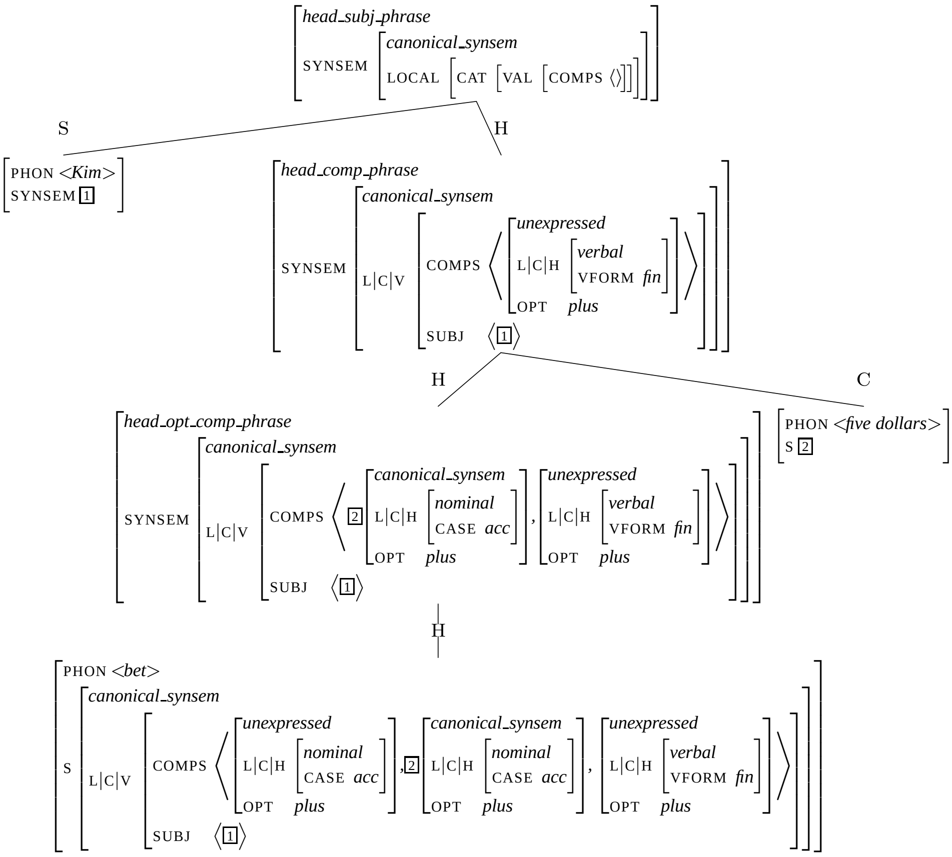

Figure 6: A sentence in which only the second most-oblique object is realized

The unary structure at the bottom of the tree is an instance of the additional *head_opt_comp_phrase*, whose purpose is the elimination of the first complement requirement, an unexpressed optional object NP, in order to bring the requirement $\boxed{2}$ to the front of the COMPS list. That complement (*five dollars*) is then realized in the *head_comp_phrase* dominating the *head_opt_comp_phrase*.

2.1 Capturing the missed generalization

We saw above that the ERG analysis of optional complements requires three different head-complement rules since in the LKB system, in which this grammar is implemented, there is no way to express the relevant generalization: that one wants to realize the first element on the COMPS list that is not an optional argument marked as unrealized. The revised head_complement rule in figure 7 shows how the intended generalization can be expressed using an append relation ($\oplus$) to state that the element $\boxed{1}$ to be realized can be preceded by an *o_list*, i.e., a list of unrealized optional elements.

$$
\textit{head\_comp\_phrase} \rightarrow \left[\begin{array}{l} \textsc{synsem} \left[\begin{array}{l} \textit{canonical\_synsem} \\ \textsc{local}|\textsc{cat}|\textsc{val}|\textsc{comps}\ \boxed{2} \end{array}\right] \\ \textsc{head\_dtr}|\textsc{synsem} \left[\begin{array}{l} \textit{canonical\_synsem} \\ \textsc{local}|\textsc{cat}|\textsc{val}|\textsc{comps}\ \textit{o\_list} \oplus \langle \boxed{1}\boxed{2} \rangle \end{array}\right] \\ \textsc{non\_head\_dtr}|\textsc{synsem}\ \boxed{1}\textit{canonical\_synsem} \end{array}\right]
$$

Figure 7: Generalized COMPS realization in a revised head-complement rule

In a grammar including this revised *head_complement_phrase* instead of the original one from the ERG we saw in figure 4, the types and definitions for *head_opt-_comp_phrases* and *head_opt_two_comp_phrases* are no longer needed.

Interestingly, the LKB encoding of the ERG using a *head_complement_phrase* plus the two 'auxiliary' phrase types that unearth the first realized complement requirement can be seen as the result of unfurling the first three calls to the append ($\oplus$) relation in the revised *head_complement_phrase* defined in figure 7, i.e., the LKB encoding can result from a compilation step taking the more general encoding as its input. This means that the issue of enabling the grammar writer to express the full generalization with the recursive relation in the English grammar is independent of the question of the relative efficiency of parsing systems with and without run-time support for relational goals.

3 Summary

This paper investigated the issue of optional arguments in the ERG, the largest HPSG-based grammar for English currently available, and an excellent collection of analyses of many aspects of English syntax that deserves to be documented and discussed to further progress. Such a discussion is particularly relevant given that, as argued by Copestake and Flickinger (2000), one of the hallmarks distinguishing the ERG from other grammar implementation efforts such as those around the

Alvey Natural Language Tools (Briscoe et al. 1987) is its connection to HPSG as an active linguistic framework.

The discussion of the ERG analysis of optional arguments in this paper showed how the ERG captures the optionality of arguments through the use of a single lexical entry coupled with an ontology of markings distinguishing optional from obligatory as well as unrealized from realized elements. Subject-head and head-complement structures are modified accordingly, but due to the lack of a possibility to express relational attachments to phrase structure rules in grammars implemented in the LKB system, the ERG analysis falls short of the goal of treating optional arguments in a general way. Instead, it requires two new types of 'auxiliary' phrases which are otherwise unmotivated. The focus on a very lean system without relational goal attachments to phrase structure rules thus results in a loss of generality of the grammars that can be expressed, which undermines the closeness of the ERG to linguistic theory as one of its key aspects. We showed that the problem can be overcome when recursive relations are added to the expressive means available to the grammar writer. This supports the position argued in Götz and Meurers (1997) that a system for the implementation of HPSG-based grammars should include both universal implicational principles as well as definite clauses over feature terms. A further case study which makes precise in what sense such a setup supports more modular, transparent, and compact grammars can be found in Meurers et al. (2003).

The revised treatment of optionality proposed in this paper is part of an ongoing reimplementation of the ERG in the TRALE system (Meurers et al. 2002).[4]

References

Briscoe, Ted, Claire Grover, Bran Boguraev and John Carroll (1987). A formalism and environment for the development of a large grammar of English. In *Proceedings of the 10th International Jointt Conference on Artificial Intelligence (IJCAI-87)*. Milan, Italy.

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

Copestake, Ann and Dan Flickinger (2000). An open source grammar development environment and broad-coverage English grammar using HPSG. In Maria Gavrilidou, George Carayannis, Stella Markantonatou, Stelios Piperidis and Gregory Steinhauer (eds.), *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC-00)*, Athens.

Flickinger, Dan (2000). On building a more efficient grammar by exploiting types. *Natural Language Engineering* 6(1), 15–28.

Flickinger, Dan, Ann Copestake and Ivan A. Sag (2000). HPSG Analysis of English. In Wolfgang Wahlster (ed.), *Verbmobil: Foundations of Speech-to-Speech Translation*, Berlin: Springer, Artificial Intelligence, pp. 254–263.

[4] http://www.sfs.uni-tuebingen.de/hpsg/sysen.html

Götz, Thilo and Walt Detmar Meurers (1997). Interleaving Universal Principles and Relational Constraints over Typed Feature Logic. In *Proceedings of the 35th Annual Meeting of the ACL and 8th Conference of the EACL*. Madrid, pp. 1–8. http://ling.osu.edu/˜dm/papers/acl97.html.

Hinrichs, Erhard W. and Tsuneko Nakazawa (1994). Linearizing AUXs in German Verbal Complexes. In John Nerbonne, Klaus Netter and Carl Pollard (eds.), *German in Head-Driven Phrase Structure Grammar*, Stanford, CA: CSLI Publications, no. 46 in CSLI Lecture Notes, pp. 11–37.

Meurers, W. Detmar, Kordula De Kuthy and Vanessa Metcalf (2003). Modularity of grammatical constraints in HPSG-based grammar implementations. In *Proceedings of the ESSLLI '03 workshop "Ideas and Strategies for Multilingual Grammar Development"*. Vienna, Austria. http://ling.osu.edu/˜dm/papers/meurers-dekuthy-metcalf-03.html.

Meurers, W. Detmar, Gerald Penn and Frank Richter (2002). A Web-based Instructional Platform for Constraint-Based Grammar Formalisms and Parsing. In Dragomir Radev and Chris Brew (eds.), *Effective Tools and Methodologies for Teaching NLP and CL*. pp. 18–25. http://ling.osu.edu/˜dm/papers/acl02.html.

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