Lexical Rules: What are they?

Andrew Bredenkamp, Stella Markantonatou and Louisa Sadler
Department of Language and Linguistics
University of Essex, UK
{andrewb,marks,louisa}@essex.ac.uk

Abstract

Horizontal redundancy is inherent to lexica consisting of descriptions of fully formed objects. This causes an unwelcome expansion of the lexical database and increases parsing time. To eliminate it, direct relations between descriptions of fully formed objects are often defined. These are additional to the (Typed Multiple) Inheritance Network which already structures the lexicon. Many implementations of horizontal relations, however, fail to generate lexical entries on a needs-driven basis, so eliminate neither the problem of lexicon expansion nor that of inefficient parsing. Alternatively, we propose that lexical entries are descriptions of objects open to contextual specification of their properties on the basis of constraints defined within the type system. This guarantees that only those grammatical lexical entries are infered that are needed for efficient parsing. The proposal is extremely modest, making use of only basic inference power and expressivity.

1 Lexical Rules: what are they?

Within the strongly lexical framework of HPSG, **lexical rules** are used to express relations among descriptions—-a kind of indirect "horizontal relatedness" (Pollard & Sag 1987, 209) which can be contrasted with the vertical relations between the type(s) of lexical elements. Type relations are, of course, captured directly as the monotonic (typed) multiple inheritance network itself, which structures the lexicon.

Typical examples of horizontal redundancy in the hierarchical lexicon thus conceived are the Alternation phenomena (e.g. Dative Shift, the Locative Alternation, etc) and word formation phenomena (inflectional and derivational morphology). In fact, Pollard and Sag also refer to declension class membership and similar facts as horizontal relations, and as we shall see, the boundary between vertical and horizontal relations is not immutably fixed once and for all.

The notion of lexical rule is often given some status at the level of linguistic or psychological theory. (Pollard & Sag 1987) make reference to a generative or procedural interpretation of lexical rules as a deductive mechanism which can be deployed on a needs only basis, for example, to generate words from a single base form. The conception of lexical rules as essential generative devices (rather than static statements expressing (sub-) regularities), is shared in much influential work (e.g. (Bresnan 1982), (Pinker 1989)), although it is by no means universal, even within HPSG. Viewed from an implementational perspective, on-the-fly application of lexical rules brings with it a number of distinct advantages which follow from the drastic reduction in the size of the lexical database (lexical construction is less time consuming and parsing time should be reduced as lexical look up is less ambiguous, etc). At first sight then it appears that the benefit of adding an external Lexical Rule component outweighs the disadvantages (external powerful mechanisms). We will first show that their role is less clear than this suggests and certainly more problematic, before suggesting in Section 2 an alternative which eschews any extra mechanisms.

1.1 Horizontal and Vertical Redundancy

The parallel drawn above between vertical relatedness (expressed with the type system) and horizontal relatedness among descriptions of fully formed objects is however rather misleading. Monotonic multiple inheritance networks are most naturally used to represent generalisations over the properties that (groups of) linguistic objects share — inspection of any network will confirm that they are usually deployed to express what is essentially a componential analysis of objects and of the relationship between them (defined on the basis of this analysis). On the other hand, horizontal relations among descriptions (very often modelled by means of lexical rules) are essentially relations holding directly between objects themselves. While this intuition is clear, this is much less adequate an approach for morphological relatedness, where a componential approach may often appear just as natural as an object relatedness view, especially if the formalism includes functionally dependent values, permitting the expression of allomorphic variation and the like. In fact, many putatively horizontal relations may be simply re-expressed within a type hierarchy by viewing them from a componential perspective, obviating the need for expressing them on the "horizontal" dimension which may lead to the use of lexical rules. But this is only possible once one frees oneself from a view of lexical relatedness as something which holds essentially between words (objects which correspond to maximal types, that is types at the bottom of the type hierarchy).

Horizontal relations are perhaps most naturally captured by an extra device (LRs) external to the lexical network and associated inference mechanism – see (Krieger & Nerbonne 1993) and (Calcagno 1995) for recent HPSG proposals. Some recent work ((Meurers 1995) and (Riehemann 1994)) partly departs from this view by expressing relations between objects using the vertical axis (that is, using the type system), but again the starting point is 'complete' lexical objects.

1.2 Why avoid Horizontal Relations?

Horizontal relations have a number of undesirable features as well as requiring an external mechanism. Horizontal relations (between objects) are in principle pretty much unconstrained. Vertical relations are more constrained because they are based on componential analysis, starting out from the set of properties that objects have. On the other hand, any object can be related to any other object by stipulation in an external mechanism. In architectural terms, it is simply accidental (if fortuitous) that lexical rules are often used to relate minimally different objects — they are capable of much more promiscious behaviour.

This state of affairs is amply demonstrated in the literature, which abounds with attempts to constrain horizontal relations by appeal to subsidiary principles (predicate locality in LFG, constraints of a psycholinguistic nature in the work of (Pinker 1989), etc). Horizontal relations must be constrained to account for 'exceptional' behaviour, that is, for those words which do not participate to a given horizontal relation despite the fact that their description makes them appropriate candidates for the relation (verb alternations offer several examples of these situtation, for instance, 'giving' verbs which do not exhibit the so-called 'dative shift' phenomenon).

Modelling of 'exceptional' behaviour leads either to an extreme complexity of the type system or to non-monotonic solutions (Flickinger 1987) because it turns out that certain horizontal relations, usually defined over types, must be blocked for individual objects.

1.3 Implementing Horizontal Relations

Several different implementations of horizontal relations exist. All of them add extra machinery and some add extra expressive power to the core mechanism.

Most frequently, horizontal relations are implemented as unary rules operating at parsing time within a derivational component. Such a component is added to the inheritance machinery for independent reasons, mainly because of the limited expressivity of the type system. With LRs, some lexical entry is considered as 'basic' and all other lexical entries are derived from it introducing otherwise unjustified directionality to the grammar. In addition, the derivational implementation of horizontal relations fails to produce lexical entries as needed, instead, it produces lexical entries according to the system's internal algorithm of searching the rule space. Considerable ambiguity is introduced with unpleasant results for parsing time. Extra machinery for blocking these rules in order to account for exceptional behaviour is also necessary.

Alternatively, LRs may be compiled out but, under this approach too, problems like directionality and the blocking of LRs as well as expensive ambiguity at parsing time remain unsolved.

2 An alternative proposal

In this paper we explore an alternative to horizontal relatedness which exploits the idea that it is often possible to conceive of the linguistic objects in such a way as to eliminate potential sources of ambiguity and additional external mechanisms. To illustrate our approach we will propose an account of a subset of Verb Alternation phenomena which rely on what are essentially underspecified lexical entries. The lexicon will then contain one (verbal) entry and the system will rely only on the existing resources (the type hierarchy) to provide the different interpretations of the predicate which license the distinct complementation patterns. Analysis is incremental and deterministic and the procedure relies mainly on what we will call 'trivial type inference'. In the sections that follow first we discuss the linguistic approach underlying our proposal, second we compare our proposal to existing underspecification approaches and finally, we give some details of the implementation which relies on no special features or external devices.

2.1 Underspecification

We will exemplify our approach by treating a subset of verb alternations which conform to the following general schema (1). These include the so-called *spray/load* (locative) alternation, the *wipe/clear* alternation, the *break/hit* alternation etc (Levin 1993).

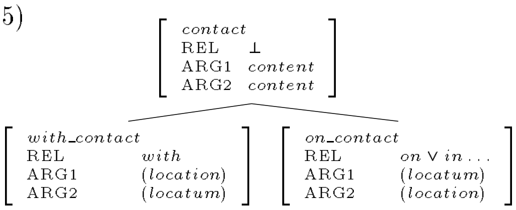
(1) V NP_j [P_1 NP_k] $\rightarrow$ V NP_k [P_2 NP_j]

We adopt the view that verb predicates are open to contextual information (which must be contrasted to the approaches whereby verb predicates are treated as fully formed objects which dictate the exact nature of their dependents). Consider the predicate *load*:

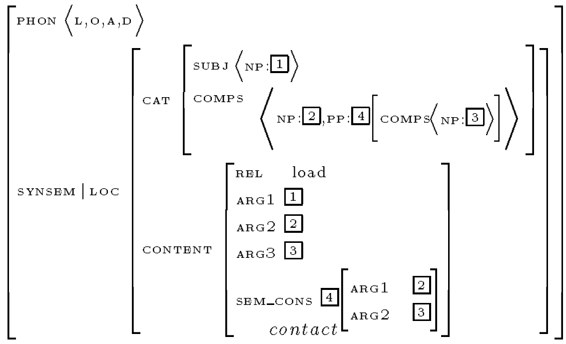
(2) The peasant loaded the horses.
(3) The peasant loaded the horses on the boat.
(4) The peasant loaded the horses with hay.

(2) is ambiguous between (3) and (4) each one of which is not ambiguous. The contextual factor that resolves the ambiguity is the semantics of the head of the prepositional complement which here is taken to specify whether the direct object of the verb is understood as the *location* and the oblique complement as the *locatum* or vice versa. The crucial assumption here is that prepositions have their own semantics, an idea first exploited in (Gawron 1986).

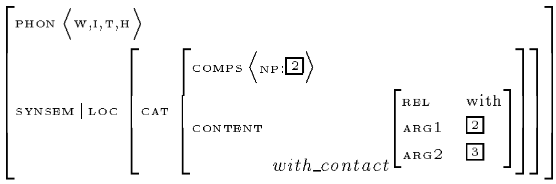
We use HPSG to model our approach. (5) gives the fragment of the type system constraining the values of the SYNSEM|LOC|CONT|NUCLEUS path in the (word) description of prepositions which participate to the locative alternation phenomenon.

(5)


We furthermore assume that the semantics of the predicates include a pointer to the semantics of the prepositional complements they license. This pointer is included as an extra feature of the value of SYNSEM|LOC|CONT|NUCLEUS. This feature we name SEM(ANTIC) CONS(TRAINTS) and we make it appropriate for the same values that the prepositional SYNSEM|LOC|CONT|NUCLEUS is assigned. The lexical entry for *to load* would look as in (6).

(6)


The lexical entry for the preposition *with* is given below:

(7)


(6) is an *underspecified* entry which gets further specified at parsing time when an appropriate PP is attached. For instance, if a *with*-PP is encountered, then an interpretation according to which the *location* surfaces as the direct object of the verb is infered.

There are some theoretical reasons why we have chosen to include a "pointer" to prepositional semantics rather than making it compatible (unifiable) to verbal semantics as Wechsler (Wechsler 1994) has proposed. Firstly, if verbal and prepositional semantics were unifiable then we would not be able to explicitly state in the semantics the relation which each feature structure encodes as there would be a clash of constants (relation names are constants). Secondly, identifying the semantics of verbs with that of prepositions does not allow for expressing certain types of diverse behaviour within the class of alternating verbs. For instance, both *load* and *stuff* show locative alternation, but only the former admits optional PP complements. With *to stuff* the interpretation under which *location* is a direct object admits an optional PP complement (8) while the interpretation under which *locatum* is a direct argument admits an obligatory one (9). Similarly, while both versions of *to load* are related to passive adjectives (*loaded cart, loaded hay*), only the "location" version is related

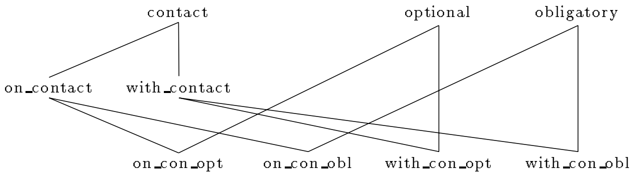


Figure 1: Type system fragment encoding prepositional alternation

to such an adjective in the case of *to stuff* (*stuffed pillow, *stuffed feathers*). The exact treatment of such phenomena, however, goes beyond the scope of our discussion here which concentrates on the use of underspecification.

(8) Mary stuffed the pillow with feathers.
(9) Mary stuffed the feathers into the pillow.

Optionality of PP complements can also be captured easily with this proposal. With *to load* (2), (3) and (4), the PP complement is optional. The grammar must have access to three different versions of *to load*, one with zero PP complements and two with a PP complement participating in the alternation discussed above. One approach would involve defining two lexical rules; an alternative would be to express all three possibilities directly. Both are problematic, of course. Consider the situation when the grammar has two PS rules for VPs, one for discharging a [NP,NP] SUBCAT list and one for discharging a [NP,NP,PP] list. Without harming generality, assume that the bivalent version of *to load* is in the lexicon and two lexical rules generate the trivalent versions. To process a trivalent version, the parser will backtrack on the bivalent version, will use a lexical rule and then, it will either succeed or it will backtrack again and use a second lexical rule.

To avoid this, the following solution may be adopted. First, the type system is augmented to allow for declaring the property of being an optional or an obligatory prepositional complement, as in figure 1.

Second, a PS structure rule is introduced of the following sort:

VP $\rightarrow$ V[SUBCAT[NP,NP,P(optional)P]], NP

Only one trivalent, underspecified version of *to load* is necessary. The parsing of a trivalent version as before would involve backtracking on the rule dealing with optional complements but then the rule dealing with obligatory ones would be chosen and it would succeed anyway.

Only limited inference power is necessary for this set up to work: the system must be able to infer that the unification of a subtype with its supertype is of the type of the subtype. This *'trivial inference power'* is independently needed to deal, for instance, with (10): if NP_1 is a subtype of NP then rule (10) will work only if trivial inference power is available when the sequence NP_1, VP is encountered.

(10) S $\rightarrow$ NP, VP

There are proposals in the literature which build on the idea of using underspecified entries. However, several of them use additional, external powerful mechanisms to simulate type inference. (van Noord & Bouma 1994) use underspecified verb entries and PROLOG delayed evaluation techniques to insert adjuncts in Dutch VPs without using lexical rules which would cater for the necessary variations of the subcategorisation list of verbs. In another proposal using underspecification (Sanfilippo 1995) type inference (feature structure grounding) is simulated by relying on an external mechanism as powerful as PROLOG. In dealing with different complementation pattern phenomena, Sanfilippo constructs type system fragments where the meet of the alternative complements is defined and subtypes verbs according to complement types. Therefore, the information about the alternation is duplicated in the type system as it is encoded both on the complement types and the verb types. The same information is encoded again on a table of clauses which relate a verbal "meet" type with a maximal complement type and a maximal verb type. Such type resolving clauses are provided for each alternation pattern. PS rules are annotated with procedures which pick up the correct verb type resolving clause when the appropriate complement is encountered. Both the clauses and the searching procedures are mechanisms external to the inferencing mechanism that is directly related with the type system. Sanfilippo's approach, though powerful and flexible, seems extravagant for phenomena like verb alternations of the kind discussed here as well as inflection phemomena of the kind discussed in (Krieger & Nerbonne 1993). In such cases the system can take advantage of the fact that type inference can be driven by the combination of the information that is related to two separate strings (preposition and verb, verb ending and verb stem) as is exemplified in our proposal.

Furthermore, in our approach no horizontal relations exist as the lexicon contains only one entry and no other entry is ever generated. Instead, the single lexical entry is interpreted on the fly, each time according to well-specified constraints. Consequently, no ambiguity problems result with a nice effect on parsing time. In this sense, using underspecification defined in the type system is more economic than using lexical rules or a "static" version of underspecification which is defined in the lexicon. For instance, (Krieger & Nerbonne 1993) have used a specialised macro, the so-called distributive (or named) disjunction, in a treatment of German verb inflectional morphology:

While it is true that distributive disjunction does not add any expressive power to the system (though a piece of machinery, the specialised macro, must be supported), if the macro is ever called all the legal combinations are thereby generated and added to the lexicon. In this, the situation is precisely the same as with lexical rules, for in each case, what is provided is simply a compact representation of an ambiguity.

This can be also exemplified from the domain of Verb Alternation phenomena. (11) will generate two lexical entries with an identical PHON string.

(11)

$$\left[\begin{array}{ll} \text{PHON} & \langle \text{L,O,A,D} \rangle \\ \text{SYNSEM}|\text{LOC} & \left[\begin{array}{ll} \text{CAT} & \left\{ \begin{array}{l} \left[\begin{array}{ll}\text{SUBJ} & \langle \text{NP}:\boxed{1}\rangle \\ \text{COMPS} & \langle \text{NP}:\boxed{2}, \text{WITH\_PP}:\boxed{3}\rangle\end{array}\right] \vee \\ \left[\begin{array}{ll}\text{SUBJ} & \langle \text{NP}:\boxed{1}\rangle \\ \text{COMPS} & \langle \text{NP}:\boxed{3}, \text{ON\_PP}:\boxed{2}\rangle\end{array}\right] \end{array}\right\} \\ \text{CONTENT}|\text{NUCLEUS} & \left[\begin{array}{ll} \text{REL} & \text{load} \\ \text{ARG1} & \boxed{1}(\text{CAUSER}) \\ \text{ARG2} & \boxed{2}(\text{LOCATION}) \\ \text{ARG3} & \boxed{3}(\text{LOCATUM})\end{array}\right]\end{array}\right]\end{array}\right]$$

Unlike lexical rules, our approach does not face any blocking problem. A verbal predicate that does not alternate (such as the predicate *to put* (12),(13)), is assigned the apppropriate most specific semantics for its SYNSEM|LOC|CONT|NUCLEUS|SEM.CONS attribute —for *to put* that would be *on-contact* in order to make sure that the *locatum* argument always surfaces as the direct object of the verb predicate.

(12) John put his shoes on the shelf.

(13) *John put the shelf with his shoes.

3 Implementation

The approach described in Section 2 can be implemented in any environment that supports Typed Inheritance because it is monotonic and demands only 'trivial inference power'. For the purposes of experimentation a grammar fragment was implemented in the ALEP system - a lean formalism with a simple inheritance type system, and a simple context free rule backbone. Processing in this system is normally divided into separate structure building and feature decoration rule components, however for our purposes no use was made of this distinction.

For the construction of the VP, a simple rule was used (VP $\rightarrow$ V NP PP), of the following form (14):

(14)

```
ld:{sign=>phrasal:{
        synsem=>synsem:{
                locl=>locl:{
                        cat=>cat:{
                                head=>HEAD,
                                subj=>[SUBJ],
                                comps=>[]},
                        content=>Content}}}}
  <  [
        ld:{sign=>lexical:{
              synsem=>synsem:{
                    locl=>locl:{
                        cat=>cat:{
                          head=>HEAD=>verb:{},
                          subj=>[SUBJ],
                          comps=>[OBJ1,OBJ2]}
                          content=>Content}}}},
        ld:{sign=>phrasal:{
              synsem=>OBJ1=>synsem:{
                          locl=>locl:{
                              cat=>cat:{
                                 head=>noun:{}}}}}},
        ld:{sign=>phrasal:{
                  synsem=>OBJ2=>synsem:{
                              locl=>locl:{
                                  cat=>cat:{
                                     head=>prep:{}}}}}}].
```

The relevant lexical entries for the fragment were as follows. The verbal entry (*load*) subcategorizes for a single NP subject and NP and PP complements (15). This entry has underspecified semantics with respect to the semantic constraints on its second and third arguments (as suggested in (8)). These are provided by (structure sharing with) the SEM_CONSTR feature of third argument, the prepositional phrase (the variable 'Arg3').

(15)

```
load ~
ld:{
 sign=>stem:{
  m_PHON_LEX[load],
  synsem=>synsem:{
     locl=>locl:{
         cat=>cat:{
            head=>verb:{},
                subj=>[synsem:{locl=>locl:{
                            cat=>cat:{
                               head=>nom:{},
                               subj=>[],
                               comps=>[],
                               spr=>[]},
                            content=>Arg1}}],
                comps=>[synsem:{locl=>locl:{
                             cat=>cat:{
                                head=>nom:{},
                                    subj=>[],
                                    comps=>[],
                                    spr=>[]},
                             content=>Arg2}},
                         synsem:{locl=>locl:{
                              cat=>cat:{
                                 head=>prep:{},
                                     subj=>[],
                                     comps=>[],
                                     spr=>[]},
                              content=>Arg3}}]},
         content=>r_psoa:{
                psoa=>arg3_psoa:{
                         rel=>rel:{
                               rel_name=>load},
                               arg1=>Arg1,
                               arg2=>Arg2=>rel_psoa:{
                                     sem_constr=>
                                        loc_alternation:{
                                           arg2_state=>A2S}},
                               arg3=>Arg3=>rel_psoa:{
```

```
                                   sem_constr=>
                                      loc_alternation:{
                                         arg2_state=>A2S,
                                         arg3_state=>A3S}}}}}}}}.
```

The prepositional entries now simply provide the “missing” part of the semantics, namely the *locum/locatum* distinction:

(16)

```
with ~
ld:{
sign=>lexical:{
   m_PHON_LEX[with],
   synsem=>synsem:{
        locl=>locl:{
            cat=>cat:{
               head=>prep:{
                   pform=>with},
                       subj=>[],
                       comps=>[synsem:{
                                 locl=>locl:{
                                     cat=>cat:{
                                        head=>noun:{case=>acc},
                                        subj=>[],
                                        comps=>[]},
                                        content=>inst_psoa:{
                                                  rel=> Rel}}}]},
            content=>inst_psoa:{
                     rel => Rel,
                         sem_constr => with_variant:{
                                         arg2_state=>locatum,
                                         arg3_state=>locum}}}}}}.
```

4 Conclusion

We have shown that horizontal redundancy is inherent to a lexicon consisting of descriptions of fully formed objects. To eliminate horizontal redundancy, direct relations between descriptions of fully formed objects must be defined externally to the Typed Mulitple Inheritance Network or unintuitive solutions must be pursued. Available implementations of horizontal relations fail to satisfy the reasons that dictate their implementation: the on-need generation of lexical entries and efficient parsing. Alternatively, we proposed that lexical entries are descriptions of objects which allow for further contextual specification of their properties on the basis of clearly defined constraints. We have shown that this is an easily implementable proposal even in environments with lean inference power and expressivity because it relies on very basic machinery which is available for independent reasons.

This approach can be adopted whenever information can be distributed among independent surface strings. Under the light of this proposal, many of the phenomena which have been argued in (Pollard & Sag 1987) to justify the horizontal relatedness approach can be viewed as different ‘interpretations’ of a ‘core’ lexical entry according to well-specified types of ‘context’. However, it must be noted here that this is not always a simple task. Roughly speaking, the less specific the contextual information is the more inference power and expressivity is needed to retain the underspecification approach.

References

Joan Bresnan (ed). 1982. *The mental representation of grammatical relations.* MIT Press

Mike Calcagno. 1995. Interpreting lexical rules. In *AQUILEX* Workshop on lexical rules. Cambridge, UK, August 9-11, 1995

Daniel Paul Flickinger. 1987. *Lexical Rules in the Hierarchical Lexicon.* PhD. Stanford University.

Jean Mark Gawron. 1986. Situations and Prepositions. *Linguistics and Philosophy 9*, pp.327–382

Hans-Ulrich Krieger & John Nerbonne. 1993. Feature-BAsed Inheritance Networks for Computational Lexicons. In (eds) Ted Briscoe, Valeria di Paiva and Ann Copstake *Inheritance, Defaults and the Lexicon.* Cambridge University Press, 90—136

Beth Levin. 1993. *English Verb Classes and Alternations. A Preliminary Investigation.* The University of Chicago Press.

Walt Detmar Meurers. 1995. Towards a semantics for lexical rules as used in HPSG. In *AQUILEX* Workshop on lexical rules. Cambridge, UK, August 9-11, 1995

Steven Pinker. 1989. *Learnability and Cognition: The acquisition of argument structure.* MIT Press

Carl Pollard & Ivan A. Sag. 1987. *Information-based Syntax and Semantics. Volume 1. Fundamentals.* Center for the Study of Language and Information

Susanne Riehemann, *Morphology and the Hierarchical Lexicon* (ms.) Stanford University, Stanford.

Antonio Sanfilippo. 1995. Lexical Polymorphism and Word Usage Extensibility. In AQUILEX *Workshop on lexical rules*, Cambridge, UK, August 9–11, 1995

Gertjan van Noord and Gosse Bouma. 1994. Adjuncts and the processing of lexical rules. In *Proceedings of the 15th International Conference on Computational Linguistics (*COLING*)*, Kyoto, 1994

Stephen Mark Wechsler. 1994. Preposition Selection Outside the Lexicon. To appear in *Proceedings of the Thirteenth West Coast Conference on Formal Linguistics*