

## Defaults and lexical prototypes

Robert Malouf, San Diego State University

Inheritance hierarchies play at least two distinct roles in Head-Driven Phrase Structure Grammar. First, they are used to establish a basic common metatheory for language description, by specifying an ontology of linguistic objects, their types, and their properties. At this level, for example, we would state the fact that there is a feature called *SUBJ* whose value must be a list of signs, or that the feature *HEAD* has a sub-feature *INV*, whose value can be + or -. While important, this level is not especially interesting from a purely linguistic point of view, and the descriptions at this level hardly vary across languages (Bender et al., 2002).

The other role for inheritance in HPSG is to describe regularities and sub-regularities in a specific language's inventory of lexical and phrasal constructions (e.g., Flickinger, 1987). At this level, we capture the fact that classes of words can be the same in some ways and different in others. Patterns of sameness can be reified as supertypes, while differences are instantiated on lower types in the hierarchy. For example, both transitive verbs and intransitive verbs take subjects, but only transitive verbs take complements. In a hierarchical representation for this, the type *verb* could specify the subject-selecting property of verbs in general, but leave complement-selecting properties unspecified. Subtypes of *verb* would then add more detail, specifying the complement-selecting properties of particular kinds of verbs.

As a general device for representing linguistic knowledge, inheritance hierarchies have proven to be very powerful. However, while strict inheritance hierarchies can in principle be scaled up to capture a large portion of the lexicon of a language (Copestake and Flickinger, 2000; Flickinger et al., 2000), the resulting hierarchies are extremely complex. This is especially true for categories exhibiting family resemblances, in which the members share many properties but which have no necessary or sufficient conditions for membership.

Lexical inheritance hierarchies can be greatly simplified by allowing default constraints which can be overridden by more specific types (e.g., Ginzburg and Sag, 2001; Sag et al., 2002). In addition, Malouf (2000, 2003) argues that default inheritance hierarchies can be used to indirectly model prototype effects of the kind discussed by Rosch (1978) and Croft (1991), among many others. But, introducing default overriding allows the possibility that a member of a category might conceivably share no properties in common with the other members, raising questions about what it really means for something to belong to a category.

The core of the problem is that both inheritance hierarchies and default overriding are mechanisms for describing systems of objects that are both the same and different. It is unclear what kind of empirical arguments could be adduced to distinguish between strict and default inheritance in a formal system which allows both. The standard solution to this problem is to either prohibit default overriding completely, or to allow it when convenient only as an abbreviatory device.

In this talk, I will sketch an alternative, namely, eliminating inheritance in favor of default overriding. In this kind of prototype-based model, lexical classes are represented as a fully specified prototypical member, and the other members of the categories are represented as

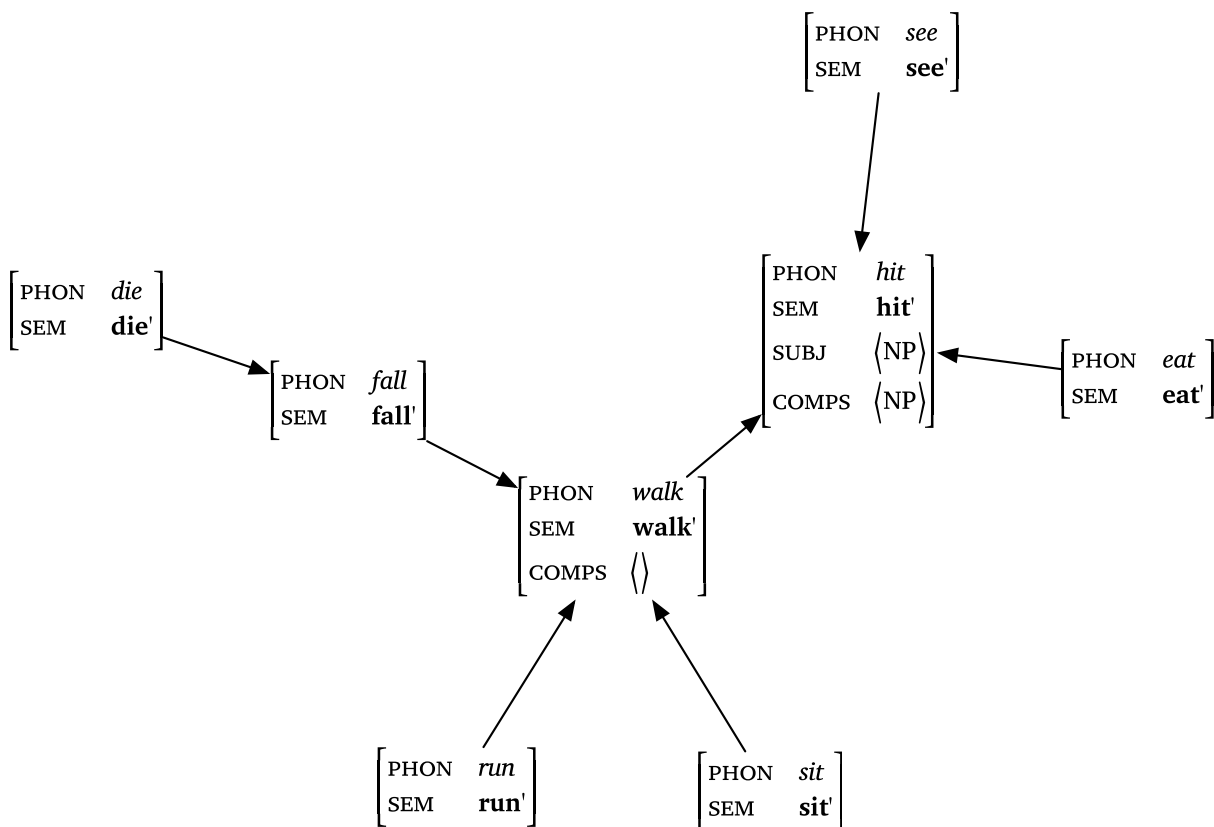


Figure 1: A prototype-based model of the lexicon

extensions or modifications of that central member. In a prototype-based model of the lexicon, we might have a fully specified lexical entry for a prototypical verb, say, *hit* (see Figure 1). Other transitive verbs would then be defined in relation to the prototype. A less prototypical lexical item, such as the intransitive verb *walk*, is specified for any ways in which it differs from the prototype. A particular item may be both an extension and a prototype: in Figure 1, the unergative verb *walk* is an extension of transitive *hit*, but the prototype for the unaccusative verb *fall*.

The relation between a more prototypical item and a less prototypical item is formally very much like default inheritance, but unlike inheritance it does not imply that one item is a more specific case of the other. Following the computer science literature (Lieberman, 1986), we can call this relation **delegation** to distinguish it from inheritance in type hierarchies. Any attempt to resolve the value of a feature which isn't specified is delegated to the prototype. In Figure 1, *eat* delegates any unspecified features to *hit*. In effectively, *eat* inherits its SUBJ value from the lexical entry for *hit*, but there is no sense in which *eat* is a member of the class *hit*; both *eat* and *hit* are full-fledged lexical entries.

Like inheritance-based models, this kind of prototype-based model has a long and successful

history as a formal knowledge-representation strategy (Dony et al., 1992; Blaschek, 1994; Craig, 2002). This talk will offer a specification of a formal mechanism for prototype-based representation for lexical and phrasal constructions in the context of HPSG, and will discuss some of the analytic questions that this type of model raises. In particular, I will address the consequences a move like this would have for analyses of mixed category constructions. Mixed categories, like gerunds or participles, seem to be simultaneously members of more than one category. One device for representing such hybrids is an extension to prototype-based models to allow multiple delegation, where a single item can be an extension of more than one prototype.

In addition, prototype-based theories have the potential to interface in a natural way with psycholinguistic view of lexical representations like exemplar-based models (Exemplar-learning et al., 2006; Baayen, 2007). Indeed, since lexical model organized around prototypes depends on making specific claims about which forms are prototypical and which are extensions, a fully elaborated theory will require an integration of morphosyntactic constraints with other dimensions of language use, such as frequency. This may also have important consequences for the theory.

## References

- Baayen, R. H. (2007). Storage and computation in the mental lexicon. In Jarema, G. and Libben, G., editors, *The Mental Lexicon: Core Perspectives*, pages 81–104. Elsevier.
- Bender, E. M., Flickinger, D., and Oepen, S. (2002). The grammar matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, Taipei, Taiwan.
- Blaschek, G. (1994). *Object-Oriented Programming with Prototypes*. Springer-Verlag, Berlin.
- Copetake, A. and Flickinger, D. (2000). An open source grammar development environment and broad-coverage English grammar using HPSG. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC-2000)*, Athens.
- Craig, I. (2002). *The Interpretation of Object-oriented Programming Languages*. Springer Verlag, London.
- Croft, W. (1991). *Syntactic Categories and Grammatical Relations*. University of Chicago Press.
- Dony, C., Malenfant, J., and Cointe, P. (1992). Prototype-based languages: From a new taxonomy to constructive proposals and their validation. In *Proceedings of ACM OOPSLA'92*, pages 201–217.
- Exemplar-learning, schematization in a usage-based account of syntactic acquisition Kirsten Abbot-Smith, and Tomasello, M. (2006). Exemplar-learning and schematization in a usage-based account of syntactic acquisition. *The Linguistic Review*, 23(3):275–290.

- Flickinger, D. (1987). *Lexical Rules in the Hierarchical Lexicon*. PhD thesis, Stanford University.
- Flickinger, D., Copestake, A., and Sag, I. A. (2000). HPSG analysis of English. In Wahlster, W., editor, *Verbmobil: Foundations of Speech-to-Speech Translation*, pages 255–264. Springer, Berlin.
- Ginzburg, J. and Sag, I. A. (2001). *Interrogative Investigations*. CSLI Publications.
- Lieberman, H. (1986). Using prototypical objects to implement shared behavior in object oriented systems. In *Proceedings of First ACM Conference on Object-Oriented Programming Systems, Languages and Applications*, Portland, OR.
- Malouf, R. (2000). *Mixed Categories in the Hierarchical Lexicon*. CSLI Publications.
- Malouf, R. (2003). Cooperating constructions. In Francis, E. and Michaelis, L., editors, *Mismatch: Form-function Incongruity and the Architecture of Grammar*, pages 403–424. CSLI Publications, Stanford.
- Rosch, E. (1978). Principles of categorization. In Rosch, E. and Lloyd, B. B., editors, *Cognition and Categorization*, pages 24–48. Lawrence Erlbaum, Hillsdale.
- Sag, I. A., Baldwin, T., Bond, F., Copestake, A. A., and Flickinger, D. (2002). Multiword expressions: A pain in the neck for NLP. In *CICLing '02: Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*.