Book Reviews

Inheritance, Defaults, and the Lexicon

Ted Briscoe, Valeria de Paiva, and Ann Copestake (editors)
(University of Cambridge)

Cambridge: Cambridge University
Press (Studies in Natural Language
Processing, edited by Branimir
Boguraev), 1993 (actual publication
date: April 1994), vii + 298 pp.
Hardbound, ISBN 0-521-43027-5, $59.95

Reviewed by
Walter Daelemans
Tilburg University

Inheritance is an inference mechanism originally defined for semantic network formalisms. It allows redundancy to be removed from the description of a domain by factoring out reusable chunks of representation. Such a chunk is given a name, and instances of it are replaced by a link to it. For example, instead of repeating for canaries, sparrows, and doves that they have skin, feathers, wings and that they fly, we define a node *bird* encapsulating these properties and connect canary, sparrow, and dove nodes via an inheritance link to this bird node. This saves us for this example almost 50% in the number of nodes needed. Default inheritance is needed to represent exceptions in a natural way by making possible the overriding of inherited properties. In the famous penguin example, inheritance of the information "can fly" has to be blocked by the specification that penguins, even if they inherit from the bird node, cannot fly. Inheritance is called simple or tree-based when any node is allowed to inherit information from only one other node; it is called multiple if nodes may inherit information from more than one other node, possibly giving rise to conflicts in the inherited information. This problem can be solved either by refusing to infer conflicting information (orthogonal multiple inheritance), or by computing a priority ordering on the ancestor nodes in which one of the conflicting values is given precedence.

The need to encode an increasing amount of linguistic knowledge in the lexicon has increased its potential size considerably and has made the structure and organization of the lexicon a central issue in computational linguistics. It is not surprising, therefore, that computational linguists have turned to inheritance for removing redundancy from their computational lexicons and to have lexicon organization reflect linguistic generalizations. This book contains a collection of articles about such applications of inheritance. It faithfully follows the proceedings of an Acquilex workshop on this topic held at Cambridge in April 1991, with one additional paper by Lynne Cahill. Acquilex is an Esprit Basic Research action, funded by the European Community. The aim of the project is to exploit machine-readable dictionaries in the construction of lexicons for natural language processing. The book addresses the issue of lexical representation rather than acquisition, however.

For such a project, a lexical representation language (LRL) featuring inheritance is needed. Some motivations for such an LRL for the design of computational lexicons

to be used with monostratal, lexicalist theories of grammar are briefly discussed in the introduction by Ted Briscoe. It is apparently unattractive for computational linguists simply to use an existing object-oriented or frame-based programming language with inheritance for the implementation of computational lexicons (see Daelemans and De Smedt 1994, for a counter example). Instead, new LRLs are defined that do, however, incorporate many features of these more general programming languages. In "Skeptical and credulous default unification with applications to templates and inheritance," for example, Bob Carpenter shows how feature systems may be extended with a notion of skeptical default unification (default information is added only when it does not conflict with other applicable default information) to implement template and default inheritance systems. The topological sorting algorithm in CLOS (the object-oriented system of Common Lisp; Steele 1990) is used to resolve conflicts between inconsistent information inherited in multiple inheritance. "A practical approach to multiple default inheritance for unification-based lexicons," by Graham Russell, Afzal Ballim, John Carroll, and Susan Warwick-Armstrong, describes default inheritance in ELU, a unification grammar development environment. An object-oriented approach is used here as well, with classes, methods (macro calls), and again the topological sorting of CLOS to compute the parent priorities in multiple inheritance. Rémi Zajac, in "Issues in the design of a language for representing linguistic information based on inheritance and feature structures," explicitly addresses some of the connections between object-oriented approaches and TFS, a typed feature-structure language developed at the University of Stuttgart that is intended to reconcile object-oriented with logic programming. The language does not allow default inheritance, however.

Typed feature structures, with or without default inheritance, have become the mainstream LRL in computational linguistics. Typing feature structures allows limited error checking and classification of feature structures (by type checking and type inferencing, respectively). A partial ordering of types allows non-default inheritance. As shown in the article by Carpenter, TFS systems can be extended with default inheritance using default unification. Hans-Ulrich Krieger and John Nerbonne, in "Feature-based inheritance networks for computational lexicons," show that such a formalism is even expressive enough to represent lexical rules, for which other approaches use special machinery. An inflectional paradigm, for example, is represented as a feature structure and can participate in the same inheritance relations that relate word classes to lexemes. Distributed disjunction is used to associate forms with syntactic properties.

A serious contender to typed feature structures as LRL of choice is DATR (Evans and Gazdar 1990). In normal use, DATR enforces orthogonal multiple inheritance (information inherited from different "parents" should not contain conflicts). In "Prioritised multiple inheritance in DATR," Roger Evans, Gerald Gazdar, and Lionel Moser show that prioritized multiple inheritance of the CLOS type discussed earlier can be described in DATR without making syntactic and semantic additions to the language. In "Some reflections on the conversion of the TIC lexicon into DATR," Lynne Cahill describes experiences with porting a moderately large lexicon (1000 words) written in POP11 to DATR in order to improve its portability. She provides some comparisons of storage needed (DATR is slightly better) and compilation and lookup processing time (DATR is worse). It is unclear whether generalizations can be made on the basis of this anecdotal experiment, but it does provide insight into what a DATR lexicon might look like.

The inclusion in this volume of "Norms or inference tickets? A frontal collision between intuitions," by Michael Morreau, is slightly puzzling. He presents an argument that two types of nonmonotonic reasoning—normative reasoning (as in circumscription) and default reasoning (as in default logic)—cannot be combined within one

theory. It would have been helpful if the relevance of this result for approaches to nonmonotonic reasoning in the lexicon would have been explained.

The latter half of the book is taken up by several articles describing the Acquilex approach to lexical representation. The project has chosen a typed feature-structure language augmented with default inheritance as its LRL. In "The Acquilex LKB: An introduction," by Ann Copestake, Antonio Sanfilippo, Ted Briscoe, and Valeria de Paiva, an overview of the formalism is given, and it is convincingly argued that the type checking and classification that are possible with the authors' LRL are useful when building up lexicons incrementally. Lexical rules in this formalism are feature structures representing relationships between lexical signs and can be used to generate new lexical signs. Lexical rules are used, as expected, to describe productive lexical processes in morphology and semantics. The LRL is formally described in "Types and constraints in the LKB," by Valeria de Paiva, and in "LKB encoding of lexical knowledge," Antonio Sanfilippo provides an analysis of the English verb system using the LRL. The analysis is provided without much linguistic argumentation but it provides an excellent illustration of the kind of structures that arise when using the LRL. An article by Ann Copestake, "Defaults in lexical representation," shows how the LRL is extended with a default inheritance mechanism. Again, default unification is used to implement default inheritance. A clarifying comparison of the Acquilex default unification to alternative definitions is provided. In "Untangling definition structure into knowledge representation," by Piek Vossen and Ann Copestake, finally, a number of problems with the semiautomatic acquisition of semantic taxonomies from MRDs is discussed, with their partial solution by extending the syntax of the LRL. Some of these changes also affect inheritance. Three appendices list a number of Acquilex-related publications, the syntax of the description language of the Acquilex lexical knowledge base, and references to available software.

The book is very well edited and forms on the whole a coherent collection. It will be required reading for anyone who wants to get acquainted with what is going on in computational lexicology, especially in the design of lexical representation formalisms. It is a welcome follow-up to the recent special issue of *Computational Linguistics* on inheritance in natural language processing at large (volume 18, numbers 2 and 3, 1992), in which most of the articles were concerned with the use of inheritance in the lexicon as well. Some of the articles in both collections even overlap. Readers interested in developments in the field *after* this *CL* special issue may be disappointed, however. Most of the papers in the book do not differ very much from the version in the Acquilex proceedings. This leads in some cases to the paradoxical situation that the work described in this book (available only in early 1994) was already out of date when the 1992 *CL* special issue appeared. The article by Russell et al., for example, lacks the scope and examples of Russell et al. 1992, and the TFS system described in the article by Zajac lacks the completeness of the description in Zajac 1992.

Which lexical representation formalism is preferable is still an open research question as some of the complexities, limitations, and subtleties of the different proposals become apparent. Current research, as evidenced in this book, focuses on special-purpose default inheritance formalisms or limited extensions of unification-based feature-structure formalisms. This is acceptable because until recently lexical semantics and pragmatics have remained relatively undeveloped. More sophisticated theories in this area (e.g., Pustejovsky 1991) and descriptions of less-studied languages may soon require more sophisticated lexical inference processes, however.

References

Daelemans, Walter, and De Smedt, Koenraad (1994). "Inheritance in an object-oriented representation of linguistic categories." In *International Journal of Human-Computer Studies* (special issue on object-oriented approaches in artificial intelligence and human-computer interaction). In press.

Evans, Roger, and Gazdar, Gerald (1990). *The DATR Papers*, Cognitive Science Research Paper CSRP 139, School of Cognitive and Computing Sciences, University of Sussex.

Pustejovsky, James (1991). "The generative lexicon." *Computational Linguistics* **17**(4):409–441.

Russell, Graham; Ballim, Afzal; Carroll, John; and Warwick-Armstrong, Susan (1992). "A practical approach to multiple default inheritance for unification-based lexicons." *Computational Linguistics* **18**(3):311–338.

Steele, Guy L., Jr. (1990). *Common Lisp: The Language*, second edition. Bedford, Massachusetts: Digital Press.

Zajac, Rémi (1992). "Inheritance and constraint-based grammar formalisms." *Computational Linguistics* **18**(2):159–182.

Walter Daelemans is a senior researcher at the Institute for Language Technology and AI (ITK, Tilburg University) and teaches computational linguistics and artificial intelligence courses at the University of Antwerp. His main research interests are in linguistic knowledge representation and induction of linguistic knowledge from data (machine learning of natural language). Daelemans's address is ITK, Tilburg University, P.O. Box 90153, 5000 LE Tilburg, The Netherlands; E-mail: walter.daelemans@kub.nl