Review of *Computational Linguistics and Formal Semantics*, edited by Michael Rosner and Roderick Johnson, Cambridge University Press, 1992, 321 pp.

This is a collection of excellent papers from a workshop chaired by the editors held in Lugano at IDSIA, an institute of the Dalle Molle Foundation. I believe the workshop was held in September, 1988, but the book is not explicit.

The "Formal Semantics" (FS) of the title refers to the theories of NL meaning derived from model theory and in standard use in theoretical linguistics. The title (and the preface) suggests a thematic cohesion among papers which is actually lacking. Only some of the papers actually focus on relating FS to computational linguistics. There are honorable exceptions, however (which this review pays the most attention to), and the papers are generally of excellent quality (on Computational Linguistics *or* Formal Semantics).

"Unification" by Martin Kay is a lucid introduction to feature-based linguistic description, with prolog examples included. It is additionally interesting for Kay's history of unification-based theories, which he traces, not to the use of feature-based linguistic theories, but rather to the difficulties using a single grammatical specification for parsing and generation in ATN's.

"Representations and Interpretations" by Jens Erik Fenstad, Tore Langholm and Espen Vestre is best read as two papers. There is first a 40-pp overview of the mathematics of feature-structures (graphs), feature description languages (terms), and the attempt to find canonical data structures for these which allow the definition of efficient algorithms. This half of the paper is marred by incoherence in the only complicated example (2.48-49 on p.52), but it is one of the best articles at an accessible level on this still very current topic. An updated version with discussion of typed theories would be welcome.

The second half of Fenstad et al.'s paper is a 24-pp. development of situation semantics. It first introduces a proof theory for a situation-theoretic logic using a generalization of a Gentzen calculus (needed to deal with strong vs. weak negation simultaneously). The theorem prover is put to use in a question-answering system, but the report on the nonsemantic details of the system (morphology, parser, generator) unfortunately leave little room for the development of a number of intriguing points on the potential and implementation of situation theory for computational semantics. These concern the prolog implementation of partial logics, the treatment of quantifiers and questions, and how complete and partial information may be integrated.

Perhaps the most interesting suggestion here is that situation theory could provide an improved characterization of pragmatic notions such as relevance and informativeness. It might have enhanced the programmatic goals of this book (delineating the division of labor between linguistics and artificial intelligence; see below) if Fenstad et al. had elaborated on such points since they potentially bear on the question of where labor is to be divided. Fenstad et al.'s ideas clearly come from linguistics and they clearly encroach on the usual border, i.e. the one which charges linguistics with specifying conventional content and AI with interpreting it (see below for further discussion).

"Syntactic categories and semantic type" by Barbara H. Partee advances the argument that the strict typing commonly found in Montague grammar must be relaxed to admit polymorphism. Partee first demonstrates that the typing postulated for transitive verbs must be modified in order to account for the truth conditions of conjoined transitive verbs, and proceeds to note that this leaves conjunctions of unlike-typed verbs unexplained. This motivates a very general account of (infinite) semantic polymorphism and some remarks about how to control the process postulating the minimally required types.

"Fine Structure in Categorial Semantics" by Johan van Benthem begins from the overly general account in categorial grammar and asks how restrictions might be mathematically interesting and linguistically motivated. He leaves the impression that the mathematicians have anticipated many of the linguistic questions—and that little new mathematics is required to model NL phenomena.

"Properties, propositions and semantic theory" by Raymond Turner is a tutorial on property theory, which postulates that properties are primitive individuals rather than defined entities (defined by extension or intension). The ideas are motivated both by the need to make finer-grained distinctions than is possible in possible worlds semantics and also by cardinality paradoxes which arise on views where properties are defined as sets of individuals (relative to possible worlds).

"Algorithms for semantic interpretation" by Per-Kristian Halvorsen is a brief overview explaining the use of feature formalisms for semantic interpretation—especially the relaxation of compositionality and the exploitation of semantic underspecification.

A further attractive practical aspect of Halvorsen's work (shared by Fenstad et al. and Rupp et al.) is the fact that their semantics representations are writtten in the same formalism as their syntax. As anyone with experience in practical development knows, one of the most time-consuming problems in system maintenance is the existence of multiple representational systems, which inevitably mean varied data structures, more interfaces, complexity of interaction, decreased modifiability, and increased training times for new users. All of these problems increase when the different representational systems do not have entirely well-defined domains—when, e.g., there are different lexical, syntactic and semantic representations and where there are competing accounts of a given phenomena, something which is not uncommon. The feature-based formulations manipulate syntax and semantics in exactly the same way, eliminating these difficulties.

"Situation Schemata and Linguistic Representation" by C.J. Rupp, Roderick Johnson and Michael Rosner is an excellent companion piece to Halvorsen's. They discuss the feature formalism as a METALANGUAGE for interpretation, further emphasizing its difference from the Montague grammar approach, and they provide a very useful illustration of the techniques in a fragment used in a prototype for machine translation. The metalanguage view makes it clear that the approach comes at the cost of some indirection, but Rupp et al. find engineering reasons for preferring the approach—essentially it insulates some parts of the system from changes in others.

"Application-oriented computational semantics" by Sergei Nirenburg and Christine Defrise is a further view from the perspective of machine translation, which, next to natural language understanding, is probably the most important application of computational semantics. But Nirenburg and Defrise are attempting to bring FS to bear not in prototypes, but in genuine applications, so their perspective is new. They catalogue a number of interpretation problems in enough detail to sober any overzealous hopes that all-encompassing solutions to applications problems are likely to come from FS, and they emphasize their own pragmatic openness to solutions from any provenance. They see FS as the supplier of "microtheories" for areas such as tense or quantification. This is all encouraging coming from an applications group (at CMU and Brussels). But the system they sketch in the body of the paper appears ill-suited to the accommodation of resources from linguistic semantics. The representation language, in particular, is a mixture of so many levels that it one would despair of finding a consequence relation—syntactic or semantic on it. A comparison to the papers by Fenstad et al., Halvorsen, and Rupp et al. suggests that the latter, in adapting the feature language of syntactic description to semantic purposes, have a better design for "NL in the large".

"Form and Content in Semantics" by Yorick Wilks assesses the role of FS in computational semantics most directly and most negatively. The contribution is an imaginative (and at times entertaining) reconstruction of debates on the role of logic in AI with appropriate morals drawn about the superiority of commonsense semantics (CSS) over FS for NLP. A sample:

> [...] recent discussions in AI concerned with inheritance systems, in particular, tend to confirm my hunch on this issue, in that the most useful ones at the moment, such as Touretsky's [*sic*], are CSS systems, and attempts to formalize systems fully has made them intractable. (p.263)

But Touretzky's contribution (Touretzky 1986) was exactly the formalization of inheritance as originally used in Fahlman's NETL! It indeed turned up inconsistencies (and proposed remedies) but in general the work might be taken as a paradigm of how formal theory and practical application cooperate to mutual benefit (cf. Thomason 1992 for a history of this effort). Wilks's last aside, about formalization degrading performance makes as much sense as criticism ruining art. Formalization clarifies what a system does, but has no effect on its algorithmic complexities.

Wilks spends two pages misrepresenting FS as offering nothing but "symbol-to-symbol transformations", where its actual task is to relate symbols to mathematical structures, e.g., propositional logic expressions to the elements of a boolean algebra, lambda calculus expressions to functions of various types, etc. FS does this in oreder to characterize the relation of logical consequence. Its practical utility lies in *defining* a consequence relation that we might otherwise intuit broadly, but fail to define precisely. And a well-defined consequence relation is useful if you wish to build systems which infer.

But Wilks misses this point, goes on to conflate decidability and definability (p.265), savage the issue of scope ambiguity (whose existence he seems to deny, p.266), and then heap abuse on McDermott, apparently for agreeing with him too unenthusiastically. It is all written with the flippant venom of a bright undergrad, and, if it were only about someone else's field, one might enjoy reading it in *New York Review of Books*.

But the paper also asks the wrong questions. Wilks lists application areas to which he would like to apply semantics (p.263); these include "large-scale" lexical ambiguity and problems "collected around the notions of expertise, plans, intentions, [and] goals" (among others). It is clear that FS alone is hopelessly insufficient in dealing with these (but still necessary for many since it best provides the meaning of NL expressions about these matters). The lack of complete solutions is beside the point, however. We don't dismiss theories because of areas to which they do not apply. FS is valuable because of the areas to which it *does apply*: it has produced treatments of NL conjunction, negation, quantification, anaphora, temporal and locative reference, collective and distributive predication, mass terms, comparatives, presupposition, focus, and the syntax/semantics interface. Even where they are incomplete (e.g., anaphora resolution is never sufficiently treated by purely logical means), these treatments deliver constraints inaccessible to alternative approaches. Wilks's advocacy of CSS *instead of* FS might be tested against these phenomena—the proper areas of application for FS.

Although Wilks recommends CSS as a semantic theory at the beginning of his essay, his contribution closes with an admonition to get back to programming in order to derive NL semantic theory from successful programs (*à la* Scott-Strachey "denotational semantics"), but one might hope that theory would inform practice, and not merely legitimize it retrospectively. Wilks's reminder that application is important is certainly *à propos* in discussing computational semantics. But we should not be trapped into a false dichotomy between theory and practice.

"Epilogue: on the relation between computational linguistics and formal semantics" by Margaret King warns us from facile formulations of the issues separating AI and linguistics views of semantics and closes the book with a reminder from Pollard and Sag 1987 that debates about relative superiority of two approaches may be infected by the different perspectives being taken.

So where is computational semantics if this book may be taken as indicative?

In their preface Rosner and Johnson explain that the workshop was part of a project with the aim of promoting more "constructive interaction" between

Computational Linguistics (CL) and Artificial Intelligence (AI). FS may play the role of an intermediary here, it is suggested, defining meaning representations and suggesting canonical interpretations. The task of relating representations to syntactic expression would then fall to CL, and that of interpreting representations in a larger context of use and communication to AI.

This view of computational semantics is of course standard in much earlier, very well respected work, e.g., that of Scha 1976, Hobbs and Rosenschein 1978, et al. 1982, Schubert and Pelletier 1982, Halvorsen 1983 or Hirst 1987. It would nonetheless be useful to see its consequences systematically developed and elaborated on—after all, a great deal of work, practical and theoretical, depends on it, but that will await another book.

It makes as much good sense to exploit the results of linguistic research in semantics as it does in syntax or morphology—more so, perhaps, given the rather higher standards of precision of semanticists. But linguistic semantics does not furnish a characterization of the INTERPRETATION of utterances in use, which is what one finally needs for natural language understanding (NLU) applications—rather, it (mostly) provides a characterization of CONVENTIONAL CONTENT, that part of meaning determined by linguistic form. Interpretation is not determined by form, however, nor by its derivative content. In order to interpret correctly, one must exploit further knowledge sources and processes that are not studied linguistically and probably are not linguistic at all: domain knowledge, common sense, communicative purpose, extralinguistic task, assumptions of interlocutors about each other. AI probably has the best theories about how to make sense of this, and certainly the best theories about how to make sense of it computationally (but see Partee's contribution for references to linguitic work on context change potential in dynamic logic and discourse representation theory).

So the division of labor looks like it makes sense. Then why aren't we understanding better?

There are difficulties on both sides, but this is a book on FS, so let's concentrate on that. Here we see a fundamental equivocation in using the word "theory" as in "applying semantic theory", and sensitivity to it leads to an appreciation of the greatest difficulty in applying FS in practical systems. This is the equivocation between "an abstract study" and "a complete account". FS provides a theory of NL meaning only in the first sense. In the second sense, it might be regarded as a theory *in statu nascendi*, but it is not a closed body of doctrine. There are innumerable areas (types of lexical items, grammatical structures, contextual dependence) where it has not been developed or at best only addressed in a single article or thesis. There are even more areas where no consensus exists.

The areas of linguistic theory that have been most enthusiastically accepted in CL (e.g., GPSG) have been unusual not only in their computational properties, but also it their willingness to provide extended and detailed analyses of substantial NL fragments. While early Montague grammar attempted comprehensive fragments, more recent linguistic semantics has not, and this makes it harder to apply.

If FS had very comprehensive and detailed theories of NL meaning, the only issue for most pragmatically oriented systems builders would be how and not whether to use them. (Conversely, if we had excellent working systems based on AI principles, formal semanticists would wish to study them on their own terms.) In the current situation, computational linguists who wish to apply FS will have to work in it to some extent. This can be fruitful for theory as well as practice, but it slows things down.

There can be substantive debate about how best to integrate the linguistic and nonlinguistic tasks in NLU, but it is unproductive to characterize the issue as a simple choice between AI or linguistic views. There is something of a consensus along the lines sketched by Rosner and Johnson, but the contemporary consensus is also noteworthy for its silence about the actual computation with semantics: it is one thing to map a syntactic form to (or from) a meaning representation, quite another to characterize and control the processes of inference one may need. There are further important issues almost never raised. What kinds of inference do we need for the semantic tasks of NLU? How sensible is the difference in methodologies between the CL and AI camps: if the CL/AI arrangement entails handing a carefully derived logical characterization of meaning over to a marker-passing algorithm whose properties are logically opaque, well how much sense does it really make? (I am aware that "neater" alternatives are available, but the situation is not uncommon that logical representations are massaged by very "scruffy" techniques.) Can one usefully characterize the expressive capacity of a domain of discourse—say that needed to obtain schedule information for trains or planes—and thus restrain overly sensitive linguistic semantics in a principled way?

There are linguistic and cognitive motivations for pursuing semantics computationally, since both linguistics and cognitive science benefit from the exact modelling possible using computers. And there is the practical motivation of wanting successful NLU systems. All of these purposes might be enhanced by further attention to the issued raised by Rosner and Johnson.

The book lacks an index although one is promised in the table of contents, but is otherwise competently produced. The overall high quality of the contributions should make it valuable to all computational linguists interested in semantics.

References

et al., M. G. 1982. The GPSG Demo System. Technical report, Hewlett-Packard Laboratories.

Halvorsen, P. K. 1983. Semantics for Lexical Functional Grammars. *Linguistic Inquiry* 14(3):567–615.

Hirst, G. 1987. *Semantic Interpretation and the Resolution of Ambiguity.* Studies in Natural Language Processing. Cambridge: Cambridge University Press.

Hobbs, J., and S. Rosenschein. 1978. Making Computaional Sense of Montague's Intensional Logic. *Artificial Intelligence* 9:287–306.

Pollard, C., and I. Sag. 1987. *Information-Based Syntax and Semantics, Vol.I.* Stanford: CSLI.

Scha, R. 1976. Semantic Types in PHILQA1. In *COLING 76 Preprints.* Ottawa.

Schubert, L. K., and F. J. Pelletier. 1982. From English to Logic: Context-Free Computation of Conventional Logic Translations. *Journal of the Association for Computational Linguistics* 165–176.

Thomason, R. 1992. NETL and Subsequent Path-Based Inheritance Theory. *Computers and Mathematics with Applications* 23:179–204. reprinted in F.Lehmann (ed.) *Semantic Networks in Artificial Intelligence*, Pergamon Press 1992.

Touretzky, D. 1986. *The Mathematics of Inheritance Systems.* San Mateo, CA: Morgan Kaufmann.

John Nerbonne has written on semantics in constraint-based theories (HPSG) and is the co-developer (with Joachim Laubsch) of $\mathcal{N}LL$, a software package for semantic representation, syntax/semantics interfaces, and interfacing to NLU applications. His address is alfa-informatica, P.O.Box 716, Oude Kijk in 't Jatstraat, Rijksuniversiteit Groningen, NL 9700 AS Groningen, The Netherlands, and nerbonne@let.rug.nl.