0.0.1 Inference

The word "inference" has (at least) two senses. In the first (and narrower) sense, inference means symbolic computation with logical formulas. In the second, inference is understood in a broader, more cognitively-oriented way, namely as the study of the strategies and capacities of human agents for drawing conclusions, often on the basis of relatively limited information. One of the aims of the EGK is to draw these two research traditions closer together. The substantial progress that has been made in the last decade in symbolic computation with logical formulas, and the development of more sophisticated cognitively oriented models, means that this hope now looks realistic.

Computation with logical formulas is a fundamental form of inference: logical inference is well understood, it functions as a benchmark against which other forms of inference can be evaluated, and richer forms of reasoning (e.g. default reasoning and belief revision) often build on standard logical reasoning rather than replacing it. Most important of all, progress in automating various aspects of computation with logical formulas has been substantial. This progress has covered not only the "classic" inference tools (namely, theorem provers) but has lead to the development of alternative approaches (notably model checking and model building) and the development of important new programming paradigms (notably constraint programming).

Saarbrücken has considerable expertise in computation with logical formulas. In particular, Kohlhase, is active in higher-order theorem proving, model generation, and inference for dynamic logic; Siekmann in proof planning, and agent-based distributed theorem proving; and Smolka in the modeling and processing of constraints, and the implementation of constraint programming languages.

Edinburgh has considerable expertise in cognitive and empirically oriented approaches to inference. For example, Fourman works on extracting discourse structures from patterns of automated reasoning; Klein on the role of semantics and inference in robust language processing; Lascarides on default reasoning and its application in natural language semantics; Oberlander on diagrammatic reasoning; and Webber on the development of discourse "test suites" for automatic reasoning systems.

It seems clear from recent work that tools and approaches now available for computing with logical formulas are of relevance to a number of areas in natural language processing. For example, the DORIS system has shown that by making use of state-of-the-art theorem provers and model builders, it is possible to implement linguistically-inspired theories of presupposition. In our view, such examples are merely the tip of the iceberg: much of importance could be achieved by bringing the insights of the automated reasoning and constraint programming communities to bear on the empirical and cognitively oriented theories of inference. The proposed EGK would be an ideal forum for exploring this important new line of work.

Example Thesis Topic: Constraint Programming and Categorial Grammars.

Supervised by Smolka (Saarbrücken) and Steedman (Edinburgh).

Steedman's Combinatorial Categorial Grammar (CCG) is a well-established grammar formalism which has been used to account for the syntactic structure of various human languages and intonation structure. It was originally motivated on psycholinguistic grounds, and recent work by Steedman has also applied CCG to the problem of broad-coverage parsing.

From a computational perspective, CCG can be thought of as providing a *logic* for various levels of linguistic structure. Smolka's Mozart constraint programming system seems to offer a natural environment for developing implementations of CCG. In the past, the Mozart system has been applied in the development of efficient parsers for dependency and D-Tree grammars. Both of these grammar formalisms are highly lexicalized, and the size of syntactic structures is known a priori. CCG does not have this latter property, so designing a CCG parser in Mozart is a non-trivial problem; but the constraints CCG imposes on the way words combine to build larger structures seem to be strong enough to make the task feasible. As well as providing a natural framework in which to develop efficient CCG parsers, implementing CCG in Mozart could throw light on where the complexity of natural language really lies, and on strategies for coping with it.

Additional Thesis Topics

- Understanding Discourse by Model Generation (Kohlhase/Webber)
- A Mathematics Tutoring System Based on Proof Planning (Siekmann/ Moore)
- Type Disciplines for Constraint-based Syntactic and Semantic Processing (Smolka/Fourman)
- Integrating domain reasoning into an HPSG-based grammar for language understanding (Klein/Pinkal)