LOGIC PROGRAMMING

(Abstract only)

S. Tarnlund

Rapporteur: Mr. M. Koutny

Abstract:

We shall take up logic programming as a general computational formalism. In particular, we shall present logic programming with respect to three criteria - methodology, efficiency and applicability.

Our methodology part contains first a discussion about logic and programming. This leads to theories of logic programming of which Kowalski's theory was the first. Colmerauer's programming language, Prolog, has a close relationship to this theory. We will point out what problems this theory solved and what problems it did not solve, e.g. the notion of a correct program and to prove that a program is correct. In fact, to solve these problems a new logic theory of Clark and Tarnlund was presented.

Our efficiency part contains a short presentation of efficient logic programming languages on sequential machines, followed by a presentation of parallel logic programming languages.

The final part, on applications of logic programming, will discuss interesting applications of logic programming.

DISCUSSION

Professor Randell asked how in teaching the students PROLOG the cut operator should be explained. Prof. Tarnlund answered that he would essentially explain it in terms of what is happening in the machine.

Professor Sintzoff pointed out that in justifying the claim that logic programming is computationally adequate it is not enough to say that one can simulate any Turing machine, but one should also consider whether programming logic is structurally adequate.

Professor Randell asked whether he agreed with the Japanese Fifth Generation Project's assumption that special hardware would be necessary in order to evaluate PROLOG programs efficiently. Professor Tarnlund replied that there exist very efficient PROLOG systems and in his opinion special hardware might be needed to support parallelism rather than to evaluate sequential PROLOG programs efficiently.

DISCUSSION

The discussion during the lecture concentrated on the differences between proofs in (pure) mathematics and proofs of programs. Professor Gries asked whether one should always demand proofs of programs which take care of every detail instead of partially relying on common sense. Professor Tarnlund answered that such an approach may be acceptable in mathematics but in the proofs of programs it would be too dangerous. It was then said that proofs of programs are too often done in a way which does not take into account the proper development of data structures. Professor Hoare pointed out that proofs in mathematics are constructed in such a way that one can check them locally which is a great advantage, while in order to understand one line of the program's code one must consider the whole program which makes the development of "context-free" proofs a hard task.

After the lecture the discussion was focused on the problem of the logical relationship between the program and its specification. The discussion was summarized by Professor Pnueli who emphasized the role of minimal fixpoints in the Kowalski's system and, as a consequence, in the understanding of the relationship between programs and specifications.