Meta-Reasoning:
Transcription of an Invited Lecture by

Luigia Aiello

Università di Roma “La Sapienza”
Via Buonarroti 12, 00185 Roma, Italy

This text is an edited version of the lecture on Meta-Reasoning, presented by Luigia Aiello at the Workshop in Sesimbra, Portugal. The lecture was recorded and the transcript was edited afterwards.

My involvement with meta-knowledge and meta-reasoning started, in the early 1970’s, when I visited Stanford and began working with Richard Weyhrauch on the construction of the FOL system. Then Weyhrauch was interested in embedding the capabilities for representing meta-knowledge in his system. The results that came out of this project concerning the use of meta-knowledge were presented in a paper with Mario Aiello. At the IJCAI-75 conference they described the first implementation of meta-reasoning within the FOL system.

Their main goal was to prove properties of programs and to automate deduction in mathematical applications. The motivations they gave for adding the meta-level knowledge to a theorem prover or proof checker were several. Meta-level enabled us to

- represent axiom schemata;
- represent and derive theorem schemata;
- include subsidiary deduction rules;
- represent and prove mathematical theorems.
It has been realized that many mathematical theorems are actually meta-theorems. If you have no possibility of representing meta-knowledge in the system, you have no hope of carrying out the proofs.

Later on I was working in the framework of the FOL system for doing theorem proving within algebra. In 1980, Weyhrauch and I presented a paper at the Automated Deduction Conference in Les Arcs in which I talked about our experiments which used meta-level control to prove theorems in elementary algebra.

The idea behind that experiment was to have a combined use of object-level and meta-level notions. The arithmetic expression was evaluated and simplified at the object-level using rewrite rules; after this, meta-level transformation were performed. Then object-level transformations got it to a normal form.

The meta-level knowledge was used to speed us the proofs. In addition, meta-level enabled us to represent knowledge that could not otherwise be represented. For example, we could represent the rewrite rules for commutativity. This could not be done at the object-level because of the loops in rewriting.

Different Ways of Embedding Meta-knowledge

Now I would like to analyse what has been done in the area of meta-level representation. Some of these ideas have already been presented in my lecture at ECCAI-84. Going through the literature we can analyse various paradigms for representing meta-knowledge in AI systems.

The first paradigm for embedding meta-knowledge in a system is to represent it as a piece of code or a piece of program. You can always program a piece of meta-knowledge in terms of code that does something. But this is a very poor approach to meta-level representation.

The second way of adding meta-knowledge to your system is by means of a meta-level which controls the strategies at the object-level. I call this ML because this is the style used in the system called ML designed by Robin Milner and his colleagues in Edinburgh.