*Proving, 13th International Joint Conference on Artificial Intelligence, Chambery, Initiating Talk, p. 61, 1993.*

# $\mathcal{LEMMA}$ or Applying AI Techniques to Computer Algebra

Karsten Homann
Universität Karlsruhe
Institut für Algorithmen und Kognitive Systeme
Am Fasanengarten 5 · 76128 Karlsruhe · Germany
E-mail: homann@ira.uka.de

## 1 Introduction

$\mathcal{LEMMA}$[1] is a knowledge based environment which allows to integrate computer algebra and theorem proving.

This environment is based upon a hybrid knowledge representation system, MANTRA[2] which is semantically unified and sound. The adoption of a four-valued semantics ensures the decidability of inference algorithms. In addition, it provides communication among the usual knowledge paradigms and may be used as a rule-based system to model the procedural knowledge of a domain.

## 2 The Environment

$\mathcal{LEMMA}$ is designed and implemented stepwise. Among the available modules is a specification language for symbolic mathematical computations which enables to consider horizontal and vertical polymorphisms and to perform program transformations into executable code. Another module defines mathematical knowledge through domains of computation, operators and their properties. This is the kernel of a computer algebra subsystem.

## 3 Future Research

At it's present stage, one of our research activities is the valuation of suitable proving methods for the design of a knowledge-based mathematical deduction system. We developed an object-oriented implementation of a prover based on Girard's linear logic which we chose because of it's constructiveness. This efficient (not considering the exponentials) implementation is based on a structure called *Pool* which corresponds to semantic tableaux.

One of the ideas is to further develop the prover to a system similar to $PN2$, the second-order companion of $F$. With such kind of system, one has a natural translation of the usual data types. At this point, two questions have to be discussed:

- What are the criteria for valuating theorem proving methods for symbolic mathematical computations? Some of them are efficiency, completeness, soundness, decidability, constructiveness, naturality ...
- How can they be integrated into computer algebra systems? The environment must support interactions between the algebraic subsystem and the theorem prover. One idea is to use MANTRA's knowledge bases as common knowledge memory which is then distributed by some heuristics.

[1] $\mathcal{L}$earning $\mathcal{E}$nvironment for $\mathcal{M}$athematic and $\mathcal{M}$athematical $\mathcal{A}$pplications
[2] Modular Assertional, semantic Network and Terminological Representation Approach