The “Mathematical Assistant” Project

Training and Mobility of Researchers TMR

R E S E A R C H   N E T W O R K S

Institut für Algorithmen und Kognitive Systeme
Universität Karlsruhe, Germany

September 1, 1995

subtask **CATS**
Integrating Computer Algebra and Theorem Proving Systems

Outline

MATHEMATICAL ASSISTANT
CATS
INTRODUCTION
RELEVANCE TO OPEN MATH
WORK IN PROGRESS
CONCLUSION
FURTHER RESEARCH

Objectives

- interaction of computer algebra systems and proof development systems
  - correct reasoning in computer algebra
  - integration of symbolic calculators with Isabelle and HOL
  - cooperation between proof-verification and computer algebra (geometry) with ETNA and COCOA
  - proof planning (including proofs by computation) with CLAM-MOLLUSC
  - implementation of decision procedures in theorem provers
  - type theory, term rewriting and computer algebra
  - solving algebraic equations by symbolic computations and verification of constraints
  - extensions of OMRS and OMME
- proof or argument presentation, tutoring systems in mathematics education
  - human interface for working environments at CROAP
  - interfaces between type theory and rewriting
  - human interaction using proof planning ideas with BARNACLE
  - corpus of verified mathematics
  - teaching mathematics with MIZAR
  - comparison of proofs of theorems in different systems

INTRODUCTION

Problem solving in mathematically-based domains:

- ⋆ Computer Algebra Systems (CAS)
- ⋆ Theorem Proving Systems (TPS)

$$\text{Problem} \xRightarrow[\text{Transformation}]{} \text{System Language}$$

$$\begin{array}{lcl} \text{Subproblem}_1 & \Longrightarrow & \text{Maple} \\ \text{Subproblem}_2 & \Longrightarrow & \text{Magma} \\ \text{Subproblem}_3 & \Longrightarrow & \text{HOL} \\ \text{Subproblem}_4 & \Longrightarrow & \text{NAG} \\ & \vdots & \end{array}$$

Different subproblems can be described more naturally by different logics and theories, e.g. HOL, dynamic or temporal logics, ...

Specialized algebraic engines are more efficient and powerful, e.g. Magma, Lie, ...

Goal: common environment which allows interactions between different logics, models, vocabularies, data structures, strategies.

Advantages:

- ⋆ increased efficiency
- ⋆ improved inference capabilities
- ⋆ improved expressiveness
- ⋆ plug & play

RELEVANCE TO OPEN MATH

Probems:

- CAS and TPS are designed as stand-alone systems
- many tools – especially CAS – are copyrighted and do not allow extensions or connections between or to other software systems
- systems based upon special internal (undocumented) representation and usually do not allow any external access.

Languages and standards for interfaces between such systems must be developed.

Applications with connection of modules

⋆ multi-logic prover

⋆ hardware- and software verification

⋆ proofs with arithmetic

⋆ different abstraction levels

⋆ program transformation, planning systems

⋆ intelligent agents, natural language systems

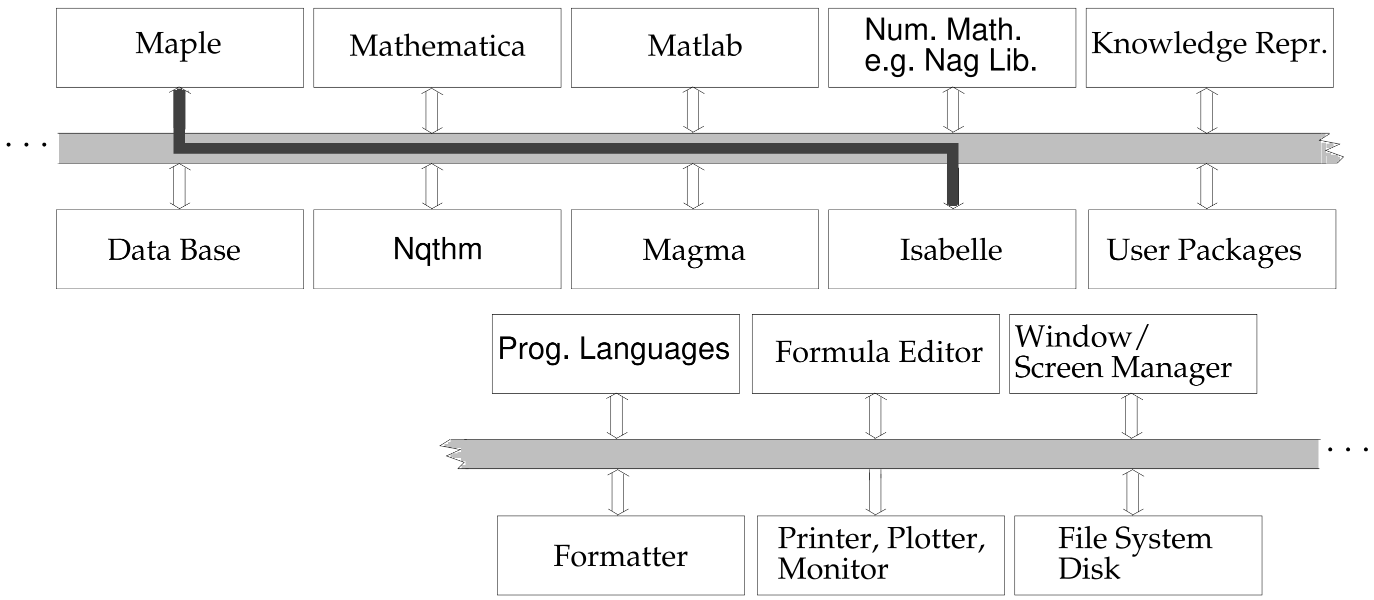
⋆ ...

WORK IN PROGRESS

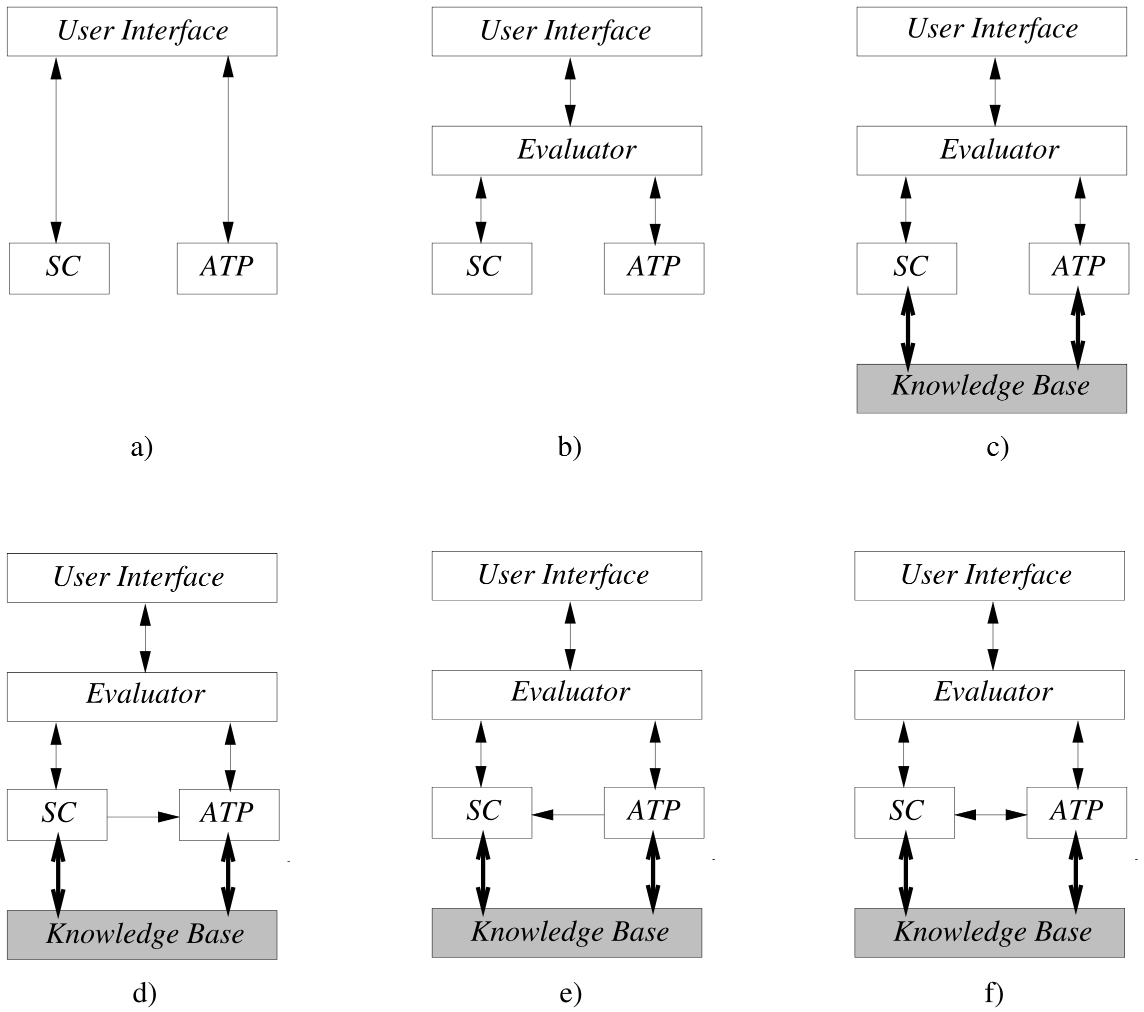
Environment for mathematical computing:

- reasoning capabilities and symbolic computations
- tools for visualization, editing, ...
- extensible by users (not only by experts)
- construction of complex systems by fitting together already existing tools and systems
- supports different levels of cooperation
- based on different logics, formalisms, data structures, interfaces, interaction capabilities
- "learning" modules
- ...

Mathematic bus (similar SCSI) with semantic for cooperation:



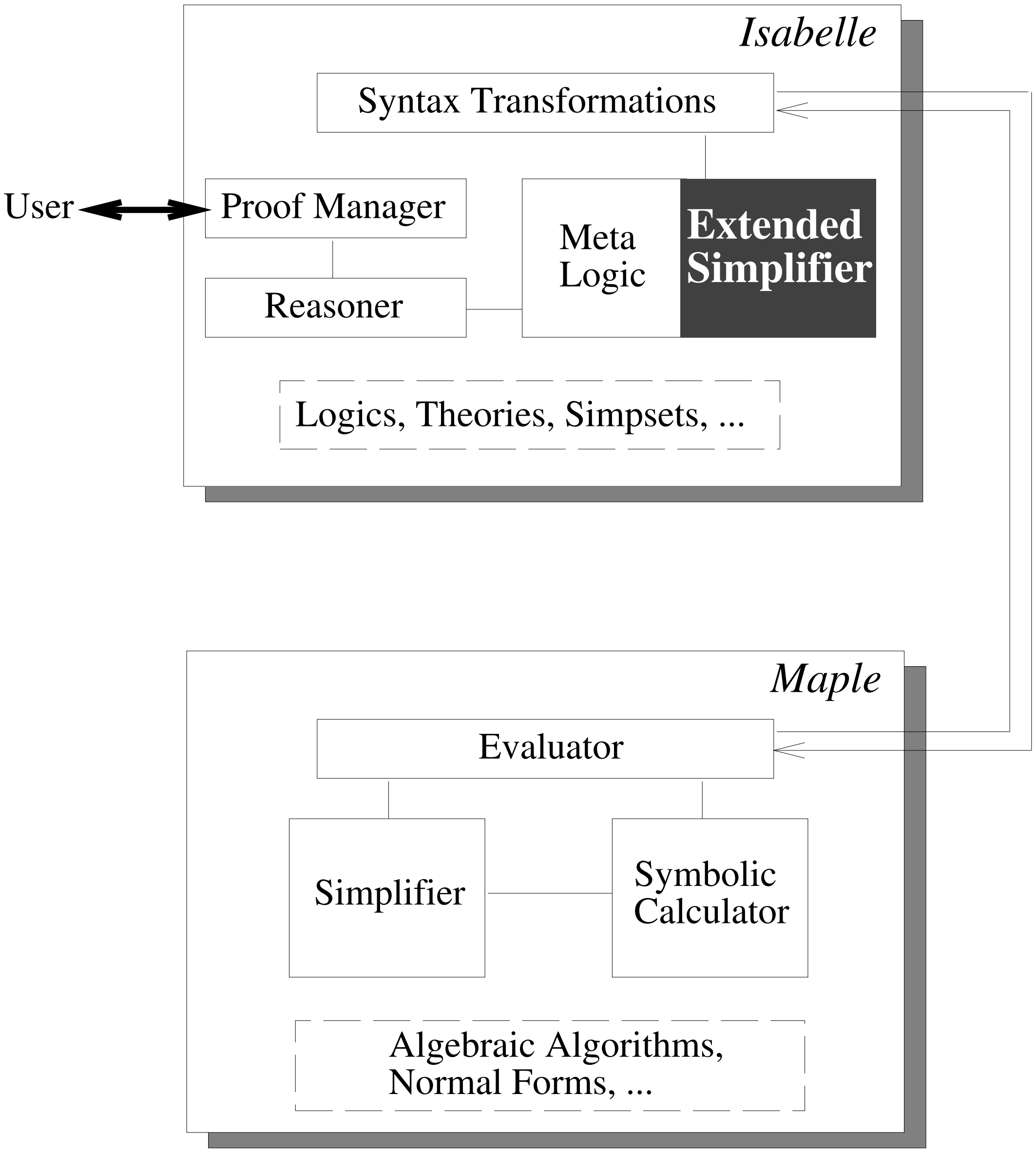
Types of Cooperation



Criteria for distinction of cooperations: master/slave relation, common knowledge base, CAS only as decision procedure, oracles, black boxes, extensible, exchangeable, trusting:

- ⋆ complete trust
  $CAS \vdash A = B$
- ⋆ partial trust
  accept results of CAS in interactive proofs, verify all computed results to accept as theorems
- ⋆ no trust
  CAS as oracle, verify immediately (only verification needed)

Example: Isabelle $\Leftrightarrow$ Maple



Examples:

$$1 + 9 * 11 = 100$$

$$\sum_{n=0}^{k} (2\,n + 1) = (k+1)^2$$

$$n : \mathtt{Nat} \quad \wedge \quad 5 \leq n \implies n^5 \leq 5^n$$

$$\vdots$$

OMRS

Specification and Combination of Reasoning Components
Open Mechanized Reasoning Systems

Stanford J. McCarthy, C. Talcott,
IRST F. Giunchiglia, P. Pecchiari

Composition of complex systems from existing tools, inclusion of modules, composition of provers with different logics by "plug and play", e.g.

prover	=	tautology + rewriter + simplifier + unifier + ...
enhanced-prover	=	add-decider(prover,decider,use-specification)

$\{$Prover $\cong$ logical service$\}$
$+$ $\{$Computing unit $\cong$ algorithmic service$\}$
$=$ $\{$Module $\cong$ mathematical service$\}$

Development of a framework for specification and structuring of modules in form of mathematical services.
Extension of *Reasoning Structures* and *Handles* for CAS.

Which information and capabilities are needed by a module to produce a mathematical service? How can useful and well-defined communication, integration and interoperationability capabilities be achieved?

$\rightsquigarrow$ Open question, no definite answer ...

Open Mechanized Mathematics Environment

Mathematical services contain information:

- ⋆ logic, algorithms, control,
- ⋆ bidirectional communication mechanisms,
- ⋆ when , how, why can/should it be used,
- ⋆ which other services are needed or can be used, ...

Such services must be described at different levels: traditional sequences, data structures, inference algorithms, control information, interaction capabilities with protocols and translations into internal representation.

Non trivial problem:
Analysis of structures and protocols for specification of interaction and development of mechanisms for ***incremental and restartable*** deduction and computation.

OMRS	Open Mechanized Reasoning System
OMCS	Open Mechanized Computation System
OMME	Open Mechanized Mathematics Environment

$$\begin{array}{rcl} \mathit{Reasoning\ Theory} & = & \mathit{Sequents} + \mathit{Rules} \\ \mathit{Reasoning\ System} & = & \mathit{Reasoning\ Theory} + \mathit{Control} \\ \textbf{OMRS} & = & \mathit{Reasoning\ System} + \mathit{Interaction} \\ \textbf{OMCS} & = & \mathit{Symbolic\ Calculator} + \mathit{Interaction} \\ \textbf{OMME} & = & \textbf{OMRS}^* + \textbf{OMCS}^* \end{array}$$

Reasoning Theory

Goal: description of ***reasoning structures*** on an abstract level, applicable to (almost) all provers.

Sequences

Pairs of sets or lists of formulae and goal formulae (Natural Deduction), or consideration of contexts (Nelson-Oppen), diagrams, place holder (lazy evaluation). Sequence System:

$$Ssys = <S, C, \models, I, \_[\_]>$$

Rules

Inference rules are n-ary functions (with constraints), allow schematic variables, backward (NuPrl), forward (FOL), both backward and forward (HOL, GETFOL).

$$\begin{aligned} Rule[Ssys] &= \{R \subseteq (S^* \times S \times P_\omega(C)) \mid (\forall <\bar{s}, s, \tilde{c}> \in R) \\ &\qquad (\forall i \in I)(<\bar{s}, s, \tilde{c}>[i] \in R)\} \\ Rset[Ssys, Id] &= [Id \rightarrow Rule[Ssys]] \end{aligned}$$

Examples (ND):

$$\frac{}{\{A\} \vdash A} ASS \qquad \frac{\Gamma, \neg A \vdash \perp}{\Gamma \vdash A} \perp_c \qquad \frac{\Gamma_1 \vdash A \qquad \Gamma_2 \vdash \neg A}{\Gamma_1 \cup \Gamma_2 \vdash \perp} \neg E$$

Reasoning Theory

Representation of a formal system or theory, specification of a set of sequences and a set of rules.

$$Rth = <Ssys, Id, \bar{r}>$$

where $\bar{r} \in Rset[Ssys, Id]$.

Combination of ***Reasoning Theories***.

Result

Many systems use different representations of information usually in the context of different logics $\leadsto$ inference rules with premisses and conclusions of different types needed, e.g.

- bridge rules in multi-language systems
- Reflection rules in FOL und NuPrl
- Meta rules in Boyer-Moore logic
- Inference rules in Hyperproof

Problem:
Representation and access to proofs and derivations ...

Reasoning Structures and Derivations

Representation of intermediate steps during proof construction; fragments can be schematic and/or temporary.

Horizontal (type of construction, abstraction and re-use) and vertical flexibility (visibility).

$$Rs_0[Rth, SN, LN] = \{<Sn, Ln, g, sg, sL, lL>\}$$

with

- $\star$ $Sn \in P_\omega(SN)$, set of sequence nodes
- $\star$ $Ln \in P_\omega(LN)$, set of connection nodes
- $\star$ $g : [Ln \to Sn]$, provides appropriate goal sequence
- $\star$ $sg : [Ln \to Sn^*]$, provides appropriate goal of sequence nodes
- $\star$ $sL : [Sn \to S]$, marks sequence nodes
- $\star$ $lL : [Ln \to [Id \times P_\omega(C)]]$, marks connection nodes

For every $rs \in Rs_0$ there exists a graph: nodes $Sn \cup Ln$ and edges $\{(ln, g(ln)) \mid ln \in Ln\} \cup \{(sn, ln) \mid ln \in Ln \ \wedge \ sn \in sg(ln))\}$.

Extension of the definition to vertical flexibility, derivations and proofs. Definition of operations on reasoning structures:

Primitive operations: `mtrs, addS, linkR, solveC, linkN`

Inference rules: forward and backward chaining strategies

Contexts

What kinds of ***mathematical services*** can be integrated?
NO black boxes (deciders, CAS, ...)!

- facilities for communication
- access to provisional internal structures
- accept and compute information incrementally
- deal with "unknown" objects

We introduce the notion of local ***context*** by extending handles (CAROLYN TALCOTT, Stanford):

context: explicitly represented knowledge of a symbolic calculator or reasoning system.

Let $c_1, c_2, \ldots$ range over contexts, $\exists$ pre-order $\trianglelefteq$:

$c_1 \trianglelefteq c_2 \Longleftrightarrow c_2$ is reachable by c_1 by adding new information

$ask(c, q)$ example interrogations are

(i) is formula f of q entailed by c

(ii) is attribute a of q computable with context c

(iii) solve q by computation with context c
The answer is $\in \{yes, no, unknown, external\}$.

$tell(c, k)$ gives a new context by extending a given context c with additional information k ($c \trianglelefteq tell(c, k)$).

$c_1 \cup c_2$, $\neg c$, $c_1 \cap c_2$, $extract(c)$**:** composition, negation, conjunction and extraction (externalize)

Computational Structures

complete trust

$$\frac{}{\vdash A = B}\ \texttt{CAS\_1} \qquad \frac{}{\vdash B}\ \texttt{CAS\_2}$$

bridging rules, e.g.:

$$\frac{\Gamma_1, A, \Gamma_2 \vdash \Gamma_3 \quad \vdash A = B}{\Gamma_1, B, \Gamma_2 \vdash \Gamma_3} =_1$$

Example:

$$\frac{}{\vdash (n+1)^5 = n^5 + 5\,n^4 + 10\,n^3 + 10\,n^2 + 5\,n + 1}\ \texttt{CAS\_1}$$

$$\frac{}{\vdash 5^{(n+1)} = 5 * 5^n}\ \texttt{CAS\_1}$$

$$a \in \texttt{Nat}, b \in \texttt{Nat}, c \in \texttt{Nat}, d \in \texttt{Nat}\ \frac{a \leq b, c \leq d,}{a + c \leq b + d}\ \texttt{le\_add}$$

$$a \in \texttt{Nat}, b \in \texttt{Nat}, c \in \texttt{Nat}, d \in \texttt{Nat}\ \frac{a \leq b, c \leq d,}{a * c \leq b * d}\ \texttt{le\_mult}$$

Solution $\rightsquigarrow$ extended reasoning structure with computations.

Conclusion

Open heterogeneous environment for mathematical computing

- ⋆ combination of different systems is very useful
- ⋆ many advantages and applications
- ⋆ few research groups are active in this field
- ⋆ languages and standards for interfaces are lacking

- ⋆ classification of different cooperation possibilities
- ⋆ example: Isabelle ⇔ Maple
- ⋆ formal definition of mathematical services
- ⋆ adaptation of sequences, rules, reasoning theory and structures

- ⋆ goal: environment in form of intelligent bus systems
- ⋆ problems with black box behaviour
- ⋆ extension and application to algorithmic systems is useful and needed