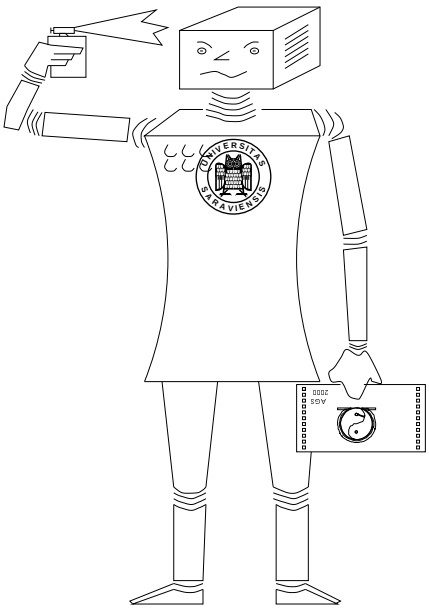
UNIVERSITÄT DES SAARLANDES
FACHBEREICH INFORMATIK
D-66041 SAARBRÜCKEN
GERMANY
WWW: http://www.ags.uni-sb.de/

SEKI Report
ISSN 1437-4447



The Calculemus Autumn School 2002: Course Notes (Part II)

Christoph Benzmüller and Regine Endsuleit (Eds.)

chris@ags.uni-sb.de, endsulei@ira.uka.de

Fachbereich Informatik
Universität des Saarlandes, Saarbrücken, Germany

Fakultät für Informatik
Universität Karlsruhe (TH), Karlsruhe, Germany

SEKI Report SR-02-08

The CALCULEMUS Autumn School 2002: Course Notes (Part II)*

Christoph Benzmüller and Regine Endsuleit (Eds.)

Fachbereich Informatik
Universität des Saarlandes, Saarbrücken, Germany
chris@ags.uni-sb.de

Fakultät für Informatik
Universität Karlsruhe (TH), Karlsruhe, Germany
endsulei@ira.uka.de

Sponsors

CALCULEMUS Autumn School 2002 gratefully acknowledges the sponsorship of the following organizations and companies:

- EU CALCULEMUS Network
  http://www.eurice.de/calculemus/
- EU IST Programme, Future and Emerging Technologies (FET)
  http://www.cordis.lu/ist/fethome.htm
- Dipartimento di Matematica, Universitá di Pisa
  http://www.dm.unipi.it/
- German Research Center for Artificial Intelligence GmbH (DFKI)
  http://www.dfki.de/
- Comune di Pisa
  http://www.comune.pisa.it/doc/cittapisa.htm
- Saar Toto GmbH
  http://www.saartoto.de/
- RIACA Research Institute for Applications of Computer Algebra
  http://www.riaca.win.tue.nl/index.html

*This techreport is supported by the EU training network CALCULEMUS (HPRN-2000-00102) funded in the EU 5th framework.

Contents

1 Alessandro Armando
University of Genua, Italy

Course: Integration of Decision Procedures in Deduction Systems

Course: Constraint Contextual Rewriting & Maple's Simplification Process as Constraint Contextual Rewriting

Tutorial: The RDL System

Integration of Decision Procedures in Automated Reasoning Systems

Alessandro Armando

MRG-Lab

DIST, University of Genova

CALCULEMUS Autumn School

Pisa, Sept 30 – Oct 1, 2002

Decision Procedures in Automated Reasoning

Experience shows that decision procedures are a fundamental ingredient for the construction of state-of-the-art Mechanized Reasoning Systems (e.g. proof-assistants, automated theorem provers, computer algebra systems),

However,

Decision Procedures in Automated Reasoning

Experience shows that decision procedures are a fundamental ingredient for the construction of state-of-the-art Mechanized Reasoning Systems (e.g. proof-assistants, automated theorem provers, computer algebra systems),

However,

to obtain an **effective integration** is a **challange** task.

Decision Procedures in Automated Reasoning – Continued

Main reasons:

- most systems are packaged as stand-alone software with inadequately described interfaces
- most of the research on decision procedures is focused on procedures delivering a 'yes-or-no' answer

Alessandro Armando

Decision Procedures in Automated Reasoning – Continued

Main reasons:

- most systems are packaged as stand-alone software with inadequately described interfaces
- most of the research on decision procedures is focused on procedures delivering a 'yes-or-no' answer

Problem: *lack of comprehensive conceptual and implementational frameworks for the integration of decision procedures in mechanized reasoning systems.*

Plan of the Lectures

The lectures introduce solutions both at the conceptual and at the implementational level.

- **Conceptual level:**
  - Constraint Contextual Rewriting
  - Maple's evalution process as Constraint Contextual Rewriting
- **Implementational level:**
  - Rewrite and Decision procedure Laboratory (RDL)
  - Logic Broker Architecture (LBA)

Constraint Contextual Rewriting

Alessandro Armando

MRG-Lab

DIST, University of Genova

CALCULEMUS Autumn School

Pisa, Sept 30 – Oct 1, 2002

Introduction

- The effective integration of decision procedures in formula simplification is one of the key problems in Automated Reasonir
- Unfortunately the problem is not easy:
  - it is fairly simple to plug a decision procedure inside a prover,
  - but, to obtain an effective integration can be a challenge.
- Problems occur when the decision procedure is asked to solve goals containing symbols which are **interpreted for the prover** but **uninterpreted for the decision procedure**.
- It is thus often the case that the decision procedure can not sol the problem at hand and therefore it is of no help to the prover.

Boyer & Moore's Augmentation Heuristics

- Boyer & Moore devised and implemented a heuristics, called **augmentation**, which extends the information available to the decision procedure with facts encoding properties of the symbols the decision procedure is not aware of.
- In Boyer & Moore's experience the heuristics is crucial to obtain an effective integration (both in speed and in decreased user interaction).

Example: Augmentation

Let us consider the problem of establishing the unsatisfiability of

$$\{n \geq 4,\ n^2 \geq 2^n,\ n^2 \neq 2^n\} \quad ($$

using a decision procedure for total orders

Example: Augmentation

Let us consider the problem of establishing the unsatisfiability of

$$\{n \geq 4,\ n^2 \geq 2^n,\ n^2 \neq 2^n\} \quad (1)$$

using a decision procedure for total orders and the following fact:

$$X \geq 4 \rightarrow X^2 \leq 2^X$$

Example: Augmentation

Let us consider the problem of establishing the unsatisfiability of

$$\{n \geq 4,\ n^2 \geq 2^n,\ n^2 \neq 2^n\}$$

using a decision procedure for total orders and the following fact:

$$X \geq 4 \rightarrow X^2 \leq 2^X$$

We can instatiate X with n and since $n \geq 0$ is in (1) we can exte
(1) with $n^2 \leq 2^n$, thereby obtaining

$$\{n \geq 4,\ n^2 \geq 2^n,\ n^2 \neq 2^n, n^2 \leq 2^n\}$$

Alessandro Armando 3

Example: Augmentation

Let us consider the problem of establishing the unsatisfiability of

$$\{n \geq 4,\ n^2 \geq 2^n,\ n^2 \neq 2^n\} \quad (1)$$

using a decision procedure for total orders and the following fact:

$$X \geq 4 \rightarrow X^2 \leq 2^X$$

We can instatiate X with n and since $n \geq 0$ is in (1) we can extend (1) with $n^2 \leq 2^n$, thereby obtaining

$$\{n \geq 4,\ n^2 \geq 2^n,\ n^2 \neq 2^n, n^2 \leq 2^n\}$$

which is readily found unsatisfiable by the decision procedure.

Alessandro Armando

Boyer & Moore's Augmentation Heuristics (continued)

The problem with the augmentation heuristics is that it *greatly complicates* the integration schema and sophisticated strategies ar needed to control the augmentation activity.

- The complexity of the approach makes it **very difficult** any attempt to modify, extend, reuse, and reason about the resultin integration schema.
- Even the proofs of basic and fundamental properties such as soundness and termination can be so difficult to be impractical.
- This situation has probably discouraged many, thereby preventir a wide use of the approach. (NQTHM/ACL2, Tecton, and **RDL** are the only provers based on Boyer & Moore's original ideas.)

Constraint Contextual Rewriting

- Constraint Contextual Rewriting, $\mathrm{CCR}(X)$ for short, is a generalization of (contextual) rewriting that incorporates the functionalities provided by a decision procedure.
- The services of the decision procedure are characterized **abstractly** (i.e. independently from the theory decided by the decision procedure) and the notation $\mathrm{CCR}(X)$ (by analogy with the $\mathrm{CLP}(X)$ notation) is used to stress this fact.
- By using $\mathrm{CCR}(X)$ as a reference model, the problem of the integration of decision procedures in formula simplification is reduced to the implementation of a decision procedure for the fragment of choice.

Constraint Contextual Rewriting: Applications

- A. Armando, L. Compagna, S. Ranise. RDL—Rewrite and Decision procedure Laboratory. In the Proceedings of the International Joint Conference on Automated Reasoning (IJCAR 2001), 2001.
- A. Armando and C. Ballarin. Maple's Evaluation Process as Constraint Contextual Rewriting. In the Proceedings of the Intl. Symposium on Symbolic and Algebraic Computation (ISSAC'2001), 2001.
- A. Armando, M. Rusinowitch e S. Stratulat. Incorporating Decision Procedures in Implicit Induction. In the Journal of Symbolic Computation, 2002.

Roadmap

- ☑ Introduction
- ☐ From Contextual Rewriting to Constraint Contextual Rewriting
- ☐ Constraint Contextual Rewriting
- ☐ Properties of Constraint Contextual Rewriting
- ☐ RDL
- ☐ Maple's evalution process as Constraint Contextual Rewriting

Contextual Rewriting

Extended form of conditional rewriting whereby information contained in the context of the expression being rewritten is used the rewriting activity.

Let us consider the problem of rewriting the literal p in the clause $\{p\} \cup E$ via a set of conditional rewrite rules, say R.

We call p the focus literal and call the set of the negations of the literals in E the context (denoted by $\overline{E}$).

Key idea: While rewriting the focus literal p it is legal to assume the truth of the literals in the context $\overline{E}$.

Contextual Rewriting: an Example

Let R contain the conditional rule $X \geq 0 \rightarrow \sqrt{X^2} = X$ (1)
and let the clause to be simplified be:

$$\{n \not\geq 0,\ n^2 \neq 2^n,\ \sqrt{2^n} = n\}$$

Alessandro Armando

Contextual Rewriting: an Example

Let R contain the conditional rule $X \geq 0 \rightarrow \sqrt{X^2} = X$ (1)
and let the clause to be simplified be:

$$\{n \not\geq 0,\ n^2 \neq 2^n,\ \sqrt{2^n} = n\}$$

Let $\underbrace{\{n \geq 0,\ n^2 = 2^n\}}_{\text{context}}$ $\underbrace{\sqrt{2^n} = n}_{\text{focus}}$

Alessandro Armando 9

Contextual Rewriting: an Example

Let R contain the conditional rule $X \geq 0 \rightarrow \sqrt{X^2} = X$ (1)
and let the clause to be simplified be:

$$\{n \not\geq 0,\ n^2 \neq 2^n,\ \sqrt{2^n} = n\}$$

Let $\underbrace{\{n \geq 0,\ n^2 = 2^n\}}_{\text{context}}$ $\underbrace{\sqrt{2^n} = n}_{\text{focus}}$

The l.h.s. of (1) does not match with the l.h.s. of the focus.

Alessandro Armando

Contextual Rewriting: an Example

Let R contain the conditional rule $X \geq 0 \rightarrow \sqrt{X^2} = X$ (1)
and let the clause to be simplified be:

$$\{n \not\geq 0,\ n^2 \neq 2^n,\ \sqrt{2^n} = n\}$$

Let $\underbrace{\{n \geq 0,\ n^2 = 2^n\}}_{\text{context}}$ $\underbrace{\sqrt{2^n} = n}_{\text{focus}}$

The l.h.s. of (1) does not match with the l.h.s. of the focus.

However, $n^2 = 2^n$ allows us to rewrite the focus to $\sqrt{n^2} = n$ and hence enables the application of (1).

Contextual Rewriting: an Example

Let R contain the conditional rule $X \geq 0 \rightarrow \sqrt{X^2} = X$ (1)
and let the clause to be simplified be:

$$\{n \not\geq 0,\ n^2 \neq 2^n,\ \sqrt{2^n} = n\}$$

Let $\underbrace{\{n \geq 0,\ n^2 = 2^n\}}_{\text{context}}$ $\underbrace{\sqrt{2^n} = n}_{\text{focus}}$

The l.h.s. of (1) does not match with the l.h.s. of the focus.

However, $n^2 = 2^n$ allows us to rewrite the focus to $\sqrt{n^2} = n$ and hence enables the application of (1).

The focus then can be rewritten to an identity provided that condition $n \geq 0$ holds. (This can be easily verified since $n \geq 0$ occurs in the context.)

From Contextual Rewriting. . .

In Contextual Rewriting the context is used both

1. to rewrite the focus literal (aimed at enabling the application of rewrite rules) and
2. to establish the conditions of conditional rewrite rules.

In both cases the kind of reasoning applied amounts to reasoning about the properties of ground equalities.

Contextual rewriting is thus the results of integrating a "decision procedure" for ground equalities with standard conditional rewritir

. . . to Constraint Contextual Rewriting

- The pattern of interaction between rewriting and the decision procedure does not depend on the theory decided by the decision procedure.
- $\mathrm{CCR}(X)$ is then the result of abstracting contextual rewriting from the theory decided by the decision procedure.
- The traditional notion of contextual rewriting therefore becomes an instance of $\mathrm{CCR}(X)$ whereby X is instantiated by a decision procedure for ground equalities
- New forms of contextual rewriting can be obtained by instantiating X to decision procedures for different decidable theories.

An Example: $\mathbf{CCR}(TO)$

Let R contain the conditional rule $X \geq 0 \rightarrow \sqrt{X^2} = X$ (
and let $\overbrace{\{n \geq 4, n^2 \geq 2^n, n^2 \leq 2^n\}}^{\text{context}}$ $\overbrace{\sqrt{2^n} = n}^{\text{focus}}$

As before, the l.h.s. of (1) does not match with the l.h.s. of the focus. However, now the equation $n^2 = 2^n$ is not directly available in the context.

Such an equality is nevertheless entailed by the context and therefore the decision procedure can rewrite the focus to $\sqrt{n^2} = n$

This step enables the matching of the l.h.s. of (1) and therefore w are left with the problem of establishing $n \geq 0$ which is easily foun to be a consequence of the context by the decision procedure.

Remark

(Constraint) Contextual Rewriting differs from conventional rewriting in two essential ways:

1. Proofs do not have a linear structure.
   - This feature (inherited from conditional rewriting) is due to the presence of subsidiary proofs needed to establish the conditions of conditional rewrite rules.
   - This results in a hybrid notion of proof which combines the linear structure of reduction proofs with the tree-structured proofs of sequent calculi.
2. Due to the dependency from the context, rewriting is a ternary relation and not a binary relation as in conventional rewriting.

Contextual Reduction Systems

Let $\mathcal{L}$ be a set of labels. For all $\ell \in \mathcal{L}$:

- let C_ℓ and E_ℓ be sets of expressions and
- let $\mathcal{S}(C_\ell, E_\ell)$ be the set of sequents of the form $c :: e \underset{\ell}{\longrightarrow} e'$ for all $c \in C_\ell$ and $e, e' \in E_\ell$.

A *contextual reduction system* (CRS) is a structure $\langle \{\mathcal{S}(C_\ell, E_\ell)\}_{\ell \in \mathcal{L}}, \mathcal{R} \rangle$, where $\mathcal{R}$ is a set of *inference rules* of the for

$$(r)\ \frac{c_1 :: e_1 \underset{\ell_1}{\longrightarrow} e'_1 \quad \cdots \quad c_n :: e_n \underset{\ell_n}{\longrightarrow} e'_n}{c :: e \underset{\ell}{\longrightarrow} e'} \text{ if } \mathit{Cond}$$

Contextual Reduction Systems (continued)

An *ℓ-reduction of e_0 to e_m in context c* is an expression of the form $c :: e_0 \xrightarrow[\ell]{\Theta_1} e_1 \xrightarrow[\ell]{\Theta_2} \cdots e_{m-1} \xrightarrow[\ell]{\Theta_m} e_m$ with $m \geq 1$ s.t.

1. either $m = 1$, $e_1 = e_0$, and $\Theta_1 = [\,]$ (called *trivial reduction*)
2. or $c :: e_{i-1} \underset{\ell_i}{\longrightarrow} e_i$ (for $i = 1, \ldots, m$) is the conclusion of an inference rule with premises $c_{1,j} :: e_{1,j} \underset{\ell_{1,j}}{\longrightarrow} e'_{1,j}, \ldots, c_{n_i,j} :: e_{n_i,j} \underset{\ell_{n_i,j}}{\longrightarrow} e'_{n_i,j}$ and the j-th element of Θ_i is an $\ell_{i,j}$-reduction of $e_{i,j}$ to $e'_{i,j}$ in context $c_{i,j}$.

Roadmap

- ☑ Introduction
- ☑ From Contextual Rewriting to Constraint Contextual Rewriting
- ☐ Constraint Contextual Rewriting
- ☐ Properties of Constraint Contextual Rewriting
- ☐ RDL
- ☐ Maple's evalution process as Constraint Contextual Rewriting

Constraint Contextual Rewriting

- Reasoning Specialists
- Clause Simplification
- Rewriting
- Augmentation

Some Definitions. . .

We consider quantifier-free first-order languages and we assume th usual conceptual machinery (e.g. the notion of substitution).

- By Σ, Π (possibly subscripted) we denote finite sets of function and predicate symbols (with their arity), respectively.
- A *signature* is a pair of the form (Σ, Π).
- V (possibly subscribed) denotes a finite set of variables.
- A (Σ, V)*-term* is a term built out of the symbols in Σ and the variables in V in the usual way.

. . . a few more Definitions

- A (Σ, Π, V)*-atom* is either an expression $q(t_1, \ldots, t_n)$ where $q \in \Pi$ and $t_1 \ldots t_n$ are (Σ, V)-terms or one of the propositional constants $true$ and $false$.
- (Σ, Π, V)*-formulae* are built in the obvious way using the standard logical connectives (i.e. $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$).
- A (Σ, Π, V)*-literal* p is either a (Σ, Π, V)-atom, $r(t_1, \ldots, t_n)$, or a negated (Σ, Π, V)-atom, $\neg r(t_1, \ldots, t_n)$.
- A (Σ, Π, V)*-expression* is a (Σ, V)-term or a (Σ, Π, V)-formula.
- A (Σ, Π, V)*-clause* is a disjunction of literals which we indicate as finite set of (Σ, Π, V)-literals.

Abbreviations

- We write (Σ, Π)-atom, (-literal, -expression, -clause) instead of $(\Sigma, \Pi, \emptyset)$-atom (-literal, -expression, -clause).
- If a is an atom, then $\overline{a}$ abbreviates $\neg a$ and $\overline{\neg a}$ stands for a.
- Let Q be a set of literals, then
  - $\overline{Q}$ abbreviates $\{\overline{q} : q \in Q\}$,
  - $Q \rightarrow p$ abbreviates the clause $\overline{Q} \cup \{p\}$, and
  - $\bigwedge Q$ stands for a conjunction of the literals in Q.

Semantics

- If ϕ is a (Σ, Π, V)-formula and Γ is a set of (Σ, Π, V)-formulae, then ϕ is a *logical consequence* of Γ iff $\Gamma \models \phi$, where $\models$ denotes entailment in classical predicate logic with equality.
- A (Σ, Π, V)*-theory* is a set of (Σ, Π, V)-formulae closed under logical consequence.
- If T is a theory, then $\Gamma \models_T \phi$ abbreviates $T \cup \Gamma \models \phi$ and we say that ϕ is T-entailed by Γ.
- A formula ϕ is T*-satisfiable* iff there exists a model of $T \cup \{\phi\}$, and T*-unsatisfiable* otherwise.
- A formula ϕ is T*-valid* iff ϕ is a logical consequence of T or, equivalently, iff $\phi \in T$.

Assumptions

In the following we consider two theories T_c and T_j of signature (Σ_c, Π_c) and (Σ_j, Π_j) respectively s.t.

- $\Sigma_c \subseteq \Sigma_j$,
- $\Pi_c \subseteq \Pi_j$,
- $T_c \subseteq T_j$.

The objective will be to simplify (Σ_j, Π_j)-expressions using:

1. a decision procedure for T_c and
2. a set R of T_j-valid facts.

From Decision Procedures. . .

According to the usual definition, a decision procedure for T_c

- takes a (Σ_c, Π_c)-formula as input and
- returns a 'yes-or-no' answer indicating whether the input formula is T_c-satisfiable or not.

However this definition is seldom adequate in practical applications:

- Efficiency considerations require the procedure to be *incremental*.
- The procedure is often required the ability of "normalizing" any given expression w.r.t. the information available.

. . . to Reasoning Specialists

A *reasoning specialist* is a state-based procedure whose states (call *constraint stores*) are finite sets of (Σ_c, Π_c)-literals represented in some internal form and whose functionalities are:

Functionality		Correctness Condition
`cs-init`(C)	$\Longrightarrow$	$\|C\|$ is T_c-valid

. . . to Reasoning Specialists

A *reasoning specialist* is a state-based procedure whose states (called *constraint stores*) are finite sets of (Σ_c, Π_c)-literals represented in some internal form and whose functionalities are:

Functionality		Correctness Condition
$\texttt{cs-init}(C)$	$\Longrightarrow$	$\|C\|$ is T_c-valid
$\texttt{cs-unsat}(C)$	$\Longrightarrow$	$\|C\|$ is T_c-unsatisfiable

. . . to Reasoning Specialists

A *reasoning specialist* is a state-based procedure whose states (call *constraint stores*) are finite sets of (Σ_c, Π_c)-literals represented in some internal form and whose functionalities are:

Functionality		Correctness Condition
$\texttt{cs-init}(C)$	$\Longrightarrow$	$\|C\|$ is T_c-valid
$\texttt{cs-unsat}(C)$	$\Longrightarrow$	$\|C\|$ is T_c-unsatisfiable
(ax-cs-simp) $\dfrac{}{P :: C \xrightarrow[\text{cs}-\text{simp}]{} C'}$	$\Longrightarrow$	$P, C \models_{T_c} \bigwedge C'$

. . . to Reasoning Specialists

A *reasoning specialist* is a state-based procedure whose states (called *constraint stores*) are finite sets of (Σ_c, Π_c)-literals represented in some internal form and whose functionalities are:

Functionality		Correctness Condition
$\texttt{cs-init}(C)$	$\Longrightarrow$	$\|C\|$ is T_c-valid
$\texttt{cs-unsat}(C)$	$\Longrightarrow$	$\|C\|$ is T_c-unsatisfiable
(ax-cs-simp) $\dfrac{}{P :: C \xrightarrow[\text{cs}-\text{simp}]{} C'}$	$\Longrightarrow$	$P, C \models_{T_c} \bigwedge C'$
(ax-cs-normal) $\dfrac{}{C :: e \xrightarrow[\text{cs}-\text{normal}]{} e'}$	$\Longrightarrow$	$C \models_{T_c} e \sim e'$

Example: a Reasoning Specialist for Total Orders

- Let $\Sigma_c = \Sigma_j$, $\Pi_c = \{\leq, =\}$, and $\Pi_c = \{\leq, <, \leq, <, =\}$.
- Let T_c is a (Σ_c, Π_c)-theory for total orders.
- Constraint stores are finite sets of (Σ_c, Π_c)-literals of the form $t_1 \leq t_2$ or $t_1 \neq t_2$ closed under:

$$(\textit{trans})\ \frac{t_2 \leq t_2 \quad t_2 \leq t_3}{t_1 \leq t_3}$$

For instance,

$\{a \leq b, b \leq c, c \neq a\}$ is a **not** constraint store.

$\{a \leq b, b \leq c, a \leq c, c \neq a\}$ is a constraint store.

Example (continued): a Reasoning Specialist for Total Orders

The functionalities provided by the reasoning specialist are:

- $\texttt{cs-init}(C)$ holds iff $C = \emptyset$.
- $\texttt{cs-unsat}(C)$ holds iff C contains three literals of the form $t_1 \leq t_2$, $t_2 \leq t_1$, and $t_1 \neq t_2$.
- $C :: e \xrightarrow[\text{cs}-\text{normal}]{} e'$ iff $\{s \leq t, t \leq s\} \subseteq C$ and $e' = e[t/s]$.

  For instance:

$$\{\ldots, n^2 \leq 2^n, 2^n \leq n^2, \ldots\} :: \sqrt{2^n} = n \xrightarrow[\text{cs}-\text{normal}]{} \sqrt{n^2} = n$$

Example (continued): a Reasoning Specialist for Total Orde

Let ν be a function associating sets of (Σ_c, Π_c)-literals to (Σ_c, Π_j)-literals defined by:

c	$\nu(c)$	c	$\nu(c)$
$t_1 \leq t_2$	$\{t_1 \leq t_2\}$	$t_1 \neq t_2$	$\{t_1 \neq t_2\}$
$t_1 \not\leq t_2$	$\{t_2 \leq t_1\}$	$t_1 = t_2$	$\{t_1 \leq t_2, t_2 \leq t_1\}$
$t_1 < t_2$	$\{t_1 \leq t_2, t_1 \neq t_2\}$	$t_1 \not\leq t_2$	$\{t_2 \leq t_1, t_1 \neq t_2\}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

- $P :: C \xrightarrow[\text{cs}-\text{simp}]{} C'$ iff C' is the result of adding $\nu(P)$ to C and closing the result w.r.t. $(trans)$.

  For instance:

$$\{2^n < n^2\} :: \{n^2 \leq 2^n\} \xrightarrow[\text{cs}-\text{extend}]{} \{n^2 \leq 2^n, 2^n \leq n^2, 2^n \neq n^2\}$$

Constraint Store Extension

The extension of the constraint store is modeled by:

$$\textit{(cs-simp)}\ \frac{P :: C \xrightarrow[\text{cs}-\text{simp}]{} C'}{P :: C \xrightarrow[\text{cs}-\text{extend}]{} C'}$$

where P is a finite set of (Σ_j, Π_j)-literals.

Note: The $\xrightarrow[\text{cs}-\text{extend}]{}$ relation is introduced for modularity reasons and will be extended later.

Clause Simplification

$$\textit{(cl-true)}\ \frac{}{E \cup \{true\} \xrightarrow[\text{simp}]{} \{true\}}$$

Clause Simplification

$$\textit{(cl-true)}\ \frac{}{E \cup \{true\} \underset{\text{simp}}{\longrightarrow} \{true\}}$$

$$\textit{(cl-false)}\ \frac{}{E \cup \{false\} \underset{\text{simp}}{\longrightarrow} E}$$

Clause Simplification

$$\textit{(cl-true)}\ \frac{}{E \cup \{true\} \underset{\text{simp}}{\longrightarrow} \{true\}}$$

$$\textit{(cl-false)}\ \frac{}{E \cup \{false\} \underset{\text{simp}}{\longrightarrow} E}$$

$$\textit{(cl-simp)}\ \frac{\overline{E} :: C_\circ \underset{\text{cs-extend}}{\longrightarrow} C \quad C :: p \underset{\text{ccr}}{\longrightarrow} p'}{E \cup \{p\} \underset{\text{simp}}{\longrightarrow} E \cup \{p'\}}\ \text{if } \texttt{cs-init}(C_\circ)$$

Example: Clause Simplification

Let $E = \{n \not\geq 0, n^2 \neq 2^n, \sqrt{2^n} = n\}$.

We show that $E \underset{\text{simp}}{\longrightarrow}^* \{true\}$.

Example: Clause Simplification

Let $E = \{n \not\geq 0, n^2 \neq 2^n, \sqrt{2^n} = n\}$.

We show that $E \underset{\text{simp}}{\longrightarrow}^* \{true\}$.

We apply the inference rule *(cl-simp)*

Example: Clause Simplification

Let $E = \{n \not\geq 0, n^2 \neq 2^n, \sqrt{2^n} = n\}$.

We show that $E \xrightarrow[\text{simp}]{}{}^* \{true\}$.

We apply the inference rule *(cl-simp)*

$$E \xrightarrow[\text{simp}]{\left[\{n\geq 0, n^2=2^n\} :: \emptyset \xrightarrow[\text{cs-extend}]{[\cdots]} C \quad C :: \sqrt{2^n}=n \xrightarrow[\text{ccr}]{\Pi} true\right]} \{n \not\geq 0, n^2 \neq 2^n, true\} \cdot$$

where $C = \{0 \leq n, n^2 \leq 2^n, 2^n \leq n^2\}$.

Example: Clause Simplification

Let $E = \{n \not\geq 0, n^2 \neq 2^n, \sqrt{2^n} = n\}$.

We show that $E \xrightarrow[\text{simp}]{}{}^* \{true\}$.

We apply the inference rule *(cl-simp)* and then *(cl-true)*

$$E \xrightarrow[\text{simp}]{\left[\{n\geq 0, n^2=2^n\} :: \emptyset \xrightarrow[\text{cs-extend}]{[\cdots]} C \quad C :: \sqrt{2^n}=n \xrightarrow[\text{ccr}]{\Pi} true\right]} \{n \not\geq 0, n^2 \neq 2^n, t$$

$$\cdots \{n \not\geq 0, n^2 \neq 2^n, true\} \xrightarrow[\text{simp}]{} \{true\}$$

where $C = \{0 \leq n, n^2 \leq 2^n, 2^n \leq n^2\}$.

Rewriting

(cxt-entails) $\dfrac{\{\overline{p}\} :: C \xrightarrow[\text{cs-extend}]{} C'}{C :: p \xrightarrow[\text{ccr}]{} true}$ if p is a (Σ_j, Π_c)-literal and $\texttt{cs-unsat}(C')$

Rewriting

(cxt-entails) $\dfrac{\{\overline{p}\} :: C \xrightarrow[\text{cs-extend}]{} C'}{C :: p \xrightarrow[\text{ccr}]{} true}$ if p is a (Σ_j, Π_c)-literal and $\texttt{cs-unsat}(C')$

(normal) $\dfrac{C :: e \xrightarrow[\text{cs-normal}]{} e'}{C :: e \xrightarrow[\text{ccr}]{} e'}$

Rewriting

$$\textit{(cxt-entails)}\ \frac{\{\overline{p}\} :: C \xrightarrow[\text{cs-extend}]{} C'}{C :: p \xrightarrow[\text{ccr}]{} true}\ \text{if}\ \begin{array}{l} p \text{ is a } (\Sigma_j, \Pi_c)\text{-literal and} \\ \texttt{cs-unsat}(C') \end{array}$$

$$\textit{(normal)}\ \frac{C :: e \xrightarrow[\text{cs-normal}]{} e'}{C :: e \xrightarrow[\text{ccr}]{} e'}$$

$$\textit{(crew)}\ \frac{C :: Q\sigma \xrightarrow[\text{ccr}]{} \emptyset}{C :: s[l\sigma] \xrightarrow[\text{ccr}]{} s[r\sigma]}\ \text{if}\ \begin{array}{l} (Q \rightarrow l = r) \in R \text{ and} \\ \sigma \text{ is a ground substitution s.t.} \end{array}$$

Example: ccr-reduction

The ccr-reduction occurring in the example before is:

$$C :: \sqrt{2^n} = n \xrightarrow[\text{ccr}]{\left[C :: \sqrt{2^n} = n \xrightarrow[\text{cs-normal}]{} \sqrt{n^2} = n\right]} \sqrt{n^2} = n \cdots$$

$$\cdots \sqrt{n^2} = n \xrightarrow[\text{ccr}]{\left[C :: n \geq 0 \xrightarrow[\text{ccr}]{\left[\{n \not\geq 0\} :: C \xrightarrow[\text{cs-extend}]{} C'\right]} true\right]} n = n$$

Example: ccr-reduction

The ccr-reduction occurring in the example before is:

$$C :: \sqrt{2^n} = n \xrightarrow[\text{ccr}]{\left[C :: \sqrt{2^n} = n \xrightarrow[\text{cs-normal}]{} \sqrt{n^2} = n\right]} \sqrt{n^2} = n \cdots$$

$$\cdots \sqrt{n^2} = n \xrightarrow[\text{ccr}]{\left[C :: n \geq 0 \xrightarrow[\text{ccr}]{\left[\{n \not\geq 0\} :: C \xrightarrow[\text{cs-extend}]{} C'\right]} true\right]} n = n$$

$$\cdots n = n \xrightarrow[\text{ccr}]{[\{n \neq n\} :: C \xrightarrow[\text{cs-extend}]{} C'']} true$$

where C' and C'' are readily found T_c-unsatisfiable by `cs-unsat`.

Augmenting the Constraint Store

- When the context is T_j-unsat but not T_c-unsat, then the T_j-unsatisfiability of the context can not possibly be detected by the reasoning specialist.
- The occurrence in the context of (function) symbols interpreted in T_j but not in T_c is the main cause of the problem.
- *Augmentation* extends the context with T_j-valid facts, thereby providing the reasoning specialist with properties of function symbols it is otherwise not aware of.

Augmenting the Constraint Store (continued)

- By adding T_j-valid facts to the context the heuristics aims at generating a T_j-equivalent but T_c-unsat context whose T_j-unsatisfiability can therefore be detected by the reasoning specialist.
- The selection of suitable T_j-valid facts is done by looking throug the available lemmas.

Example: Augmentation

Let T_j be a theory of integer numbers with the usual interpretation of symbols. Let $\Sigma_c = \Sigma_j$ and let Π_c, T_c, and $\xrightarrow[\text{cs-simp}]{}$ be as before. Let the following two formulae be in R (and hence in T_j):

$$X \geq 0 \rightarrow \sqrt{X^2} = X \tag{3}$$

$$X \geq 4 \rightarrow X^2 \leq 2^X \tag{4}$$

Notice that (3) and (4) are in T_j, but not in T_c.

Let $\overbrace{\{n \geq 4, n^2 \geq 2^n\}}^{\text{context}}$ $\overbrace{\sqrt{2^n} = n}^{\text{focus}}$

Unlike the previous examples, $n^2 = 2^n$ is not T_c-**entailed** by the context. However it is T_j-**entailed** by the context.

Example (continued): Augmentation

- Now, if we extend the context with $n^2 \leq 2^n$ we get a new conte that T_c-**entails** $n^2 = 2^n$.
- This enables the decision procedure to rewrite the focus literal t $\sqrt{n^2} = n$ and the reasoning continues as the previous example.
- Notice that the task of establishing that $n^2 = 2^n$ is T_j-**entailed** by the initial context falls largely beyond the scope of a decision procedure for total orders.
- The problem is nevertheless solved thanks to the use of the augmentation heuristics.

Constraint Store Extension with Augmentation

We recall that the activity of extending the constraint store is modeled by:

$$\textit{(cs-simp)}\ \frac{P :: C \xrightarrow[\text{cs}-\text{simp}]{} C'}{P :: C \xrightarrow[\text{cs}-\text{extend}]{} C'}$$

Constraint Store Extension with Augmentation

We recall that the activity of extending the constraint store is modeled by:

$$\textit{(cs-simp)}\ \frac{P :: C \xrightarrow[\text{cs}-\text{simp}]{} C'}{P :: C \xrightarrow[\text{cs}-\text{extend}]{} C'}$$

This is now supplemented with the following rule for augmentation

$$\textit{(aug)}\ \frac{C :: q_1\sigma \xrightarrow[\text{ccr}]{} true \quad \cdots \quad C :: q_n\sigma \xrightarrow[\text{ccr}]{} true \quad \{c\sigma\} :: C \xrightarrow[\text{cs}-\text{simp}]{}}{P :: C \xrightarrow[\text{cs}-\text{extend}]{} C'}$$

if $(\{q_1, \ldots, q_n\} \rightarrow c) \in R$ and
σ is a ground substitution s.t. $c\sigma$ is a (Σ_j, Π_c)-literal

Roadmap

- ☑ Introduction
- ☑ From Contextual Rewriting to Constraint Contextual Rewriting
- ☑ Constraint Contextual Rewriting
- ☐ Properties of Constraint Contextual Rewriting
- ☐ RDL
- ☐ Maple's evalution process as Constraint Contextual Rewriting

Properties of $\mathbf{CCR}(X)$

Theorem 1. [Soundness] *CCR(X) is sound, i.e.*

1. *if* $C :: e \xrightarrow[\text{ccr}]{}{}^{*} e'$, *then* $\|C\| \models_{T_j} e \sim e'$;
2. *if* $P :: C \xrightarrow[\text{cs}-\text{extend}]{}{}^{*} C'$, *then* $P, \|C\| \models_{T_j} \bigwedge \|C'\|$.

Theorem 2. [Termination] *(The version on the paper of) CCR(X) is terminating, i.e. all the reductions of the form:*

- $E_0 \xrightarrow[\text{simp}]{} E_1 \xrightarrow[\text{simp}]{} \cdots$
- $C :: e_0 \xrightarrow[\text{ccr}]{} e_1 \xrightarrow[\text{ccr}]{} \cdots$

have finite size.

Reference paper on CCR

- A. Armando and S. Ranise. Constraint Contextual Rewriting. To appear on the Journal of Symbolic Computation, special issue on First Order Theorem Proving. Peter Baumgartner and Hantao Zhang, Editors.

Conclusions

- $\mathrm{CCR}(X)$ is a generalized form of contextual rewriting which incorporates the functionalities provided by a decision procedure
- $\mathrm{CCR}(X)$ is sound and terminating.
- By using $\mathrm{CCR}(X)$ as a reference model, the problem of the integration of decision procedures in formula simplification is reduced to the implementation of a decision procedure for the fragment of choice.

Maple's Simplification Process as Constraint Contextual Rewriting

Alessandro Armando

MRG-Lab

DIST, University of Genova

CALCULEMUS Autumn School

Pisa, Sept 30 – Oct 1, 2002

The Problem of Simplification

Hard: Not decidable for many domains.

Problem: representation often does not provide sufficient information.

$$\begin{array}{lll} \frac{x^2-1}{x-1} & \rightsquigarrow & x+1 \\ & = & \text{over } \mathbb{Q}(x) \\ & \neq & \text{over } \mathbb{Q} \to \mathbb{Q} \\ & = & \text{for } x \neq 1 \end{array}$$

The Problem of Simplification

Hard: Not decidable for many domains.

Problem: representation often does not provide sufficient information.

$$\begin{array}{lll} \frac{x^2-1}{x-1} & \rightsquigarrow & x+1 \\ & = & \text{over } \mathbb{Q}(x) \\ & \neq & \text{over } \mathbb{Q} \to \mathbb{Q} \\ & = & \text{for } x \neq 1 \\ \sqrt{x^2} & \rightsquigarrow & x \\ & = & x \geq 0 \\ & \neq & x < 0 \\ & ? & x \in \mathbb{C}, \text{consider branch cuts!} \end{array}$$

Maple's Assume Facility

- Maintains a global context of user-provided assumptions.
- Simplification is performed only if conditions can be derived from the context.
- Examples:

$$\begin{aligned}
\texttt{assume}(x > 1, y > 1) &\leadsto \texttt{is}(x + 2y - 3 > 0) = \mathit{true} \\
&\leadsto \sqrt{(x + 2y - 3)^2} = x + 2y - 3 \\
\texttt{assume}(x, \texttt{real}) &\leadsto \texttt{is}(e^x \geq 0) = \texttt{true} \\
&\leadsto \sqrt{(e^x)^2} = e^x
\end{aligned}$$

- Is integrated to Maple's evaluator.

The Property Reasoner

- **Objects:** Maple's objects of computation (e.g.: π, $\sqrt{x}$)
- **Properties:** sets of objects (e.g. $[0, +\infty)$, set of continuous functions)
- They are denoted by object terms and property terms respectively
- If $t \in p$ then we say that t has property p and write $t : p$.

Property Terms

Property Terms are defined in the following way:

(i) Atomic symbols: `real`, `positive`, `monotonic`, ...

(ii) Expressions of the form (l, r), $[l, r)$, $(l, r]$, and $[l, r]$, where l and r are object terms denoting numbers.

(iii) Are built out of simpler property terms using the constructors $\neg$, $\wedge$, and $\vee$.

Property Functions

- Knowledge about functions is encoded in property functions.
- Every function f has an associated property function $\bar{f}$.
- A property function maps properties into properties, e.g.:

$$\begin{aligned}
[0, \infty) \mathbin{\overline{*}} [0, \infty) &= [0, \infty) \\
(-\infty, 0] \mathbin{\overline{*}} (-\infty, 0] &= [0, \infty) \qquad \overline{\sin}([0, \pi]) = [0, 1] \\
[0, \infty) \mathbin{\overline{*}} (-\infty, 0] &= (-\infty, 0] \qquad \overline{\sin}([0, \tfrac{\pi}{3}]) = [0, \tfrac{1}{2}\sqrt{3}] \\
(-\infty, 0] \mathbin{\overline{*}} [0, \infty) &= (-\infty, 0]
\end{aligned}$$

- Property functions satisfy

$$f(t_1, \ldots, t_n) : \bar{f}(p_1, \ldots, p_n) \quad \text{for} \quad t_1 : p_1, \ldots, t_n : p_n.$$

Maple's Evaluation as CCR

The assume facility is a reasoning specialist.
Its integration to the evaluator can be seen as CCR.

First step: regard evaluation functions as rewrite rules.

This is adequate:

- Most evaluation functions perform local transformations on terms.
- Subterms are evaluated recursively.

Property Reasoner as Reasoning Specialist

Constraint store contains property judgements: $\{t_1 : p_1, \ldots, t_n : p_n\}$ or $\vec{t} : \vec{p}$.

- `cs-init`(C) holds if and only if C is the empty set of judgements
- `cs-unsat`(C) holds if and only if $(u : \bot) \in C$ for some term u.
- $P :: C \xrightarrow[\text{cs-simp}]{} C'$ is defined in terms of `assume` and `is`.

(assume) $$\frac{}{\{u : p\} :: (\vec{t} : \vec{p}) \xrightarrow[\text{cs-simp}]{} \{u : p\} \cup (\vec{t} : \vec{p})}$$

(is) $$\frac{(\vec{t} : \vec{p}) :: u \xrightarrow[\text{solve}]{} v[\vec{t}] \qquad v[\vec{p}] \xrightarrow[\text{prop-eval}]{} p_0}{\{\neg(u : p)\} :: (\vec{t} : \vec{p}) \xrightarrow[\text{cs-simp}]{} \{u : \bot\} \cup (\vec{t} : \vec{p})} \text{ if } p_0 \preceq p$$

Property Functions and Augmentation

Augmentation not needed to reconstruct Maple's assume facility.

Weibel and Gonnet's view of property functions:

- Property functions are part of the Property Reasoner.

Proposed view:

- Property functions are lemmas.
- They state properties of functions that are uninterpreted for the reasoning specialist.

$\Longrightarrow$ Weak form of augmentation.

Strengthening the Property Reasoner

Property functions as lemmas.

- They can only express relations between properties of the input arguments and the output.
- Relations that **cannot** be expressed:
  - Monotonicity of a function f

$$Y - X : [0, \infty) \Longrightarrow f(Y) - f(X) : [0, \infty).$$

  - "Squeeze" theorem $x \in [a, b] \Longrightarrow g(x) \leq f(x) \leq h(x)$

$$X : [a, b] \Longrightarrow f(X) - g(X) : [0, \infty)$$
$$X : [a, b] \Longrightarrow h(X) - f(X) : [0, \infty)$$

- Augmentation does not suffer from this limitation!

Conclusions

Modular design of the simplifier:

- Evaluation as a set of cooperating reasoning modules with neatly specified interfaces.
- Paves the way to adding other (user-provided) reasoning specialists to Maple.

  $\Longrightarrow$ more powerful simplifier.
- Properties of user-defined functions should be declared rather than programmed.

Reference Paper

- A. Armando and C. Ballarin. Maple's Evaluation Process as Constraint Contextual Rewriting. In the Proceedings of the Intl. Symposium on Symbolic and Algebraic Computation (ISSAC'2001), 2001.

Example: Constraint Contextual Rewriting

Consider a reasoning specialist for total orders and let R contain the rewrite rule

$$X \geq 0 \Longrightarrow \sqrt{X^2} = X.$$

Then $\sqrt{\cos^2 a}$ rewrites to $\cos a$ in context

$$C = \{0 \leq \sin a,\ 0 \neq \sin a,\ \sin a \leq \cos a\}$$

The derivation is

$$C :: \sqrt{\cos^2 a} \xrightarrow[\text{ccr}]{\quad C :: \cos a \geq 0 \xrightarrow[\text{ccr}]{\quad \{\cos a \not\geq 0\} :: C \xrightarrow[\text{cs-simp}]{} C' \quad} \texttt{true} \quad} \cos a$$

where $C' = C \cup \{0 \leq \cos a, \cos a = 0, 0 \neq \cos a\}$, and $\texttt{cs-unsat}(C')$ holds.

Example: Maple's Evaluator as CCR

$\{\neg(x + 2y - 3 : [0, \infty)^{\cdot})\} ::$

$\{\}$

| *(assume)* $x - 1 : [0, \infty)$

$\{x - 1 : [0, \infty)^{\cdot}\}$

| *(assume)* $y - 1 : [0, \infty)$

$\{x - 1 : [0, \infty), y - 1 : [0, \infty)^{\cdot}\}$

| *(is)* contradiction $x + 2y - 3 \xrightarrow{\text{solve}} (x - 1) + 2(y - 1)$, $[0, \infty)^{\cdot} \overline{+2} [0, \infty)^{\cdot} \xrightarrow[\text{prop-eval}]{} [0, \infty) \preceq [0, \infty)^{\cdot}$

$\{x - 1 : [0, \infty), y - 1 : [0, \infty), x + 2y - 3 : \bot\}$

Example: Augmentation

Let R contain the facts:

$$X : [0, \infty) \ \wedge \ Y - X : [0, \infty) \Longrightarrow \sqrt{Y} - \sqrt{X} : [0, \infty)$$

$$X : [-\tfrac{\Pi}{2}, \tfrac{\Pi}{2}] \ \wedge \ Y : [-\tfrac{\Pi}{2}, \tfrac{\Pi}{2}] \ \wedge \ Y - X : [0, \infty) \Longrightarrow \sin Y - \sin X : [0, \infty)$$

and consider the problem of determining whether $\sqrt{\sin y} - \sqrt{\sin x} : [0, \infty)$ in context

$$C = \{x : [0, \tfrac{\Pi}{2}], y : [0, \tfrac{\Pi}{2}], y - x : [0, \infty)\}$$

By *(cxt-entails)* we need to find a C' with

$$\{\neg(\sqrt{\sin y} - \sqrt{\sin x} : [0, \infty))\} :: C \xrightarrow[\text{cs-simp}]{} C'$$

and $\texttt{cs-unsat}(C')$.

Example: Augmentation (2)

$\{\neg(\sqrt{\sin y} - \sqrt{\sin x}) : [0, \infty)^{\cdot}\} ::$
$\{x : [0, \frac{\pi}{2}], y : [0, \frac{\pi}{2}], \ y - x : [0, \infty)\}$

(augment) with $x : [-\frac{\pi}{2}, \frac{\pi}{2}] \ \wedge \ y : [-\frac{\pi}{2}, \frac{\pi}{2}] \ \wedge \ y - x : [0, \infty) \Longrightarrow \sin y - \sin x : [0, \infty)$

$\{\ldots, \sin y - \sin x : [0, \infty)\}$

(augment) with $\sin x : [0, \infty) \ \wedge \ \sin y - \sin x : [0, \infty) \Longrightarrow \sqrt{\sin y} - \sqrt{\sin x} : [0, \infty)$

$\{\ldots, \sqrt{\sin y} - \sqrt{\sin x} : [0, \infty)^{\cdot}\}$

(is) contradiction $\sqrt{\sin y} - \sqrt{\sin x} \xrightarrow[\text{solve}]{} \sqrt{\sin y} - \sqrt{\sin x}$
$[0, \infty)^{\cdot} \xrightarrow[\text{prop-eval}]{} [0, \infty) \preceq [0, \infty)^{\cdot}$

$\{\ldots, \sqrt{\sin y} - \sqrt{\sin x} : \bot\}$

Example: Augmentation (3)

The second application of *(augment)* requires a further invokation of the Property Reasoner.

$$\frac{\dfrac{\dfrac{}{\{x : [0, \frac{\pi}{2}], \ldots\} :: \sin x : [0, \infty) \xrightarrow[\text{ccr}]{} \texttt{true}}}{\{\neg(\sin x : [0, \infty))\} :: \{x : [0, \frac{\pi}{2}], \ldots\} \xrightarrow[\text{cs-simp}]{} \{x : [0, \frac{\pi}{2}], \ldots, \sin x : \bot\}}}{\{x : [0, \frac{\pi}{2}], \ldots\} :: \sin x \xrightarrow[\text{solve}]{} \sin x \qquad \overline{\sin}[0, \tfrac{\pi}{2}] \xrightarrow[\text{prop-eval}]{} [0, 1] \preceq [0, \infty)}$$

RDL

Rewrite and Decision procedure Laboratory

Alessandro Armando

MRG-Lab

DIST, University of Genova

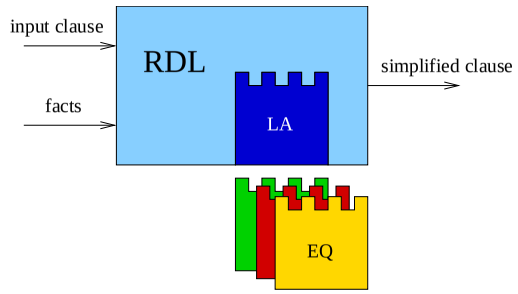
CALCULEMUS Autumn School

Pisa, Sept 30 – Oct 1, 2002

Introduction

- **RDL** simplifies clauses in a quantifier-free first-order logic with equality,
- is based on CCR,
- inherits soundness and termination from CCR,
- allows for the plug&play integration of decision procedures (as long as they comply with some precisely specified requirements), and
- uses *lemma speculation* techniques which extend the scope of the available decision procedures.

RDL: User's View



Example:

Input clause:
$\{\neg list(l),\ \neg a \leq min(l),\ \neg 0 < k,\ a < max(l) + abs(k)\}$

Input Facts: $0 < N \rightarrow abs(N) = N$
$list(X) \rightarrow min(X) \leq max(X)$

Output: $\{true\}$

Basic Reasoning Specialists

The following basic reasoning specialists are available in the current version of **RDL**:

- `eq`: for the quantifier-free theory of equality (based on Shostak's *congruence closure* algorithm),
- `la`: for quantifier-free Presburger Arithmetic (based on *Fourier-Motzkin elimination method*), and
- `eq_la`: for the combination of the previous two theories (obtained by using Nelson & Oppen's combination method).

Lemma Speculation

The following lemma speculation mechanisms are available in the current version of **RDL**:

- `aug`: augmentation, lemmas are instances of available facts.
  - **Pros:** generic (i.e. works for all interpreted symbols)
  - **Cons:** lemmas must be provided by the user
- `aff`: affinization, lemmas are generated on-the-fly
  - **Pros**: fully-automatic
  - **Cons**: generates facts about multiplication over the integers only
- `aug_aff`: combination of the previous two.

Affinization

Problem: Determine the unsatisfiability of

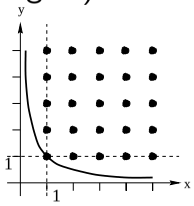
$$\{a < 0, \quad b \geq -a, \quad a * (a + b) \geq 1\}$$

over the integers using a decision procedure for Linear Arithmetic.

Solution:

1. *Affinize hyperbolic inequalities* (over the integers):

$$X * Y \geq 1 \leftrightarrow \begin{pmatrix} X \geq 1 \wedge Y \geq 1 \\ \vee \\ X \leq -1 \wedge Y \leq -1 \end{pmatrix}$$



Affinization

Problem: Determine the unsatisfiability of

$$\{a < 0, \quad b \geq -a, \quad a * (a + b) \geq 1\}$$

over the integers using a decision procedure for Linear Arithmetic.

2. *Compile into lemmas*:

$$X \geq 1 \rightarrow (X * Y \geq 1 \quad \leftrightarrow \quad Y \geq 1)$$

2. *Compile into lemmas*:

$$X \geq 1 \rightarrow (X * Y \geq 1 \;\; \leftrightarrow \;\; Y \geq 1)$$
$$X \leq -1 \rightarrow (X * Y \geq 1 \;\; \leftrightarrow \;\; Y \leq -1)$$

Alessandro Armando

2. *Compile into lemmas*:

$$X \geq 1 \rightarrow (X * Y \geq 1 \;\; \leftrightarrow \;\; Y \geq 1)$$
$$X \leq -1 \rightarrow (X * Y \geq 1 \;\; \leftrightarrow \;\; Y \leq -1)$$
$$\vdots$$

Alessandro Armando 6

2. *Compile into lemmas*:

$$X \geq 1 \rightarrow (X * Y \geq 1 \;\; \leftrightarrow \;\; Y \geq 1)$$
$$X \leq -1 \rightarrow (X * Y \geq 1 \;\; \leftrightarrow \;\; Y \leq -1)$$
$$\vdots$$

3. Use lemmas as rewrite rules to "reduce" the non-linearities, e.g.:

$$\{a < 0, \;\; b \geq -a, \;\; a * (a + b) \geq 1\}$$

Alessandro Armando

2. *Compile into lemmas*:

$$X \geq 1 \rightarrow (X * Y \geq 1 \;\; \leftrightarrow \;\; Y \geq 1)$$
$$X \leq -1 \rightarrow (X * Y \geq 1 \;\; \leftrightarrow \;\; Y \leq -1)$$
$$\vdots$$

3. Use lemmas as rewrite rules to "reduce" the non-linearities, e.g.:

$$\{a < 0, \;\; b \geq -a, \;\; a * (a + b) \geq 1\}$$
$$\Downarrow \qquad \text{if } a \leq -1$$
$$\{a < 0, \;\; b \geq -a, \;\; a + b \leq -1\}$$

2. *Compile into lemmas*:

$$X \geq 1 \rightarrow (X * Y \geq 1 \;\leftrightarrow\; Y \geq 1)$$
$$X \leq -1 \rightarrow (X * Y \geq 1 \;\leftrightarrow\; Y \leq -1)$$
$$\vdots$$

3. Use lemmas as rewrite rules to "reduce" the non-linearities, e.g.:

$$\{a < 0,\quad b \geq -a,\quad a * (a + b) \geq 1\}$$
$$\Downarrow \qquad \text{if } a \leq -1$$
$$\{a < 0,\quad b \geq -a,\quad a + b \leq -1\}$$

4. Invoke the decision procedure for LA.

Specifying the Reasoning Specialist

The reasoning specialist to apply can be specified by means of the following syntax:

$RS ::= DP \mid LS(DP)$
$DP ::= \texttt{eq} \mid \texttt{la} \mid \texttt{eq\_la}$
$LS ::= \texttt{aug} \mid \texttt{aff} \mid \texttt{aug\_aff}$

Examples: `la`,
`aug(la)`,
`aff(eq_la)`,
`aug_aff(eq_la)`,
⋮

RDL problems description

```
description(  Pb,        % Problem name
              Author,    % Problem author
              Note  ).   % Problem description

      input(  Pb,        % Problem name
              CL  ).     % Input clause

expected_output(  Pb,    % Problem name
              RS,        % Reasoning specialist used
              ORD,       % Ordering used
              EXP  ).    % Expected clause
```


RDL problems description (cont'd)

```
   fact(  Pb,            % Problem name
          Lb,            % Lemma name
          Cs,            % Lemma conditions
          Concl  ).      % Lemma conclusion

pred_sym(  Pb,           % Problem name
          P(_,...,_)  ).  % Predicate arity
```

RDL interface predicates

To load the file `F` issue

```
load_problems_file(F).
```


RDL interface predicates

To load the file `F` issue

```
load_problems_file(F).
```

To run **RDL** against problem `Pb` using the reasoning specialist `RS` issue:

```
run(RS,Pb).
```


Zhang's example: `zhang`

Given the lemma

$$A \neq 0 \Rightarrow rem(A * B, A) = 0$$

then the following formula holds

$$x * y = y * z \wedge rem(y * z, x) \neq 0 \Rightarrow x = 0$$

RDL proves the validity of this example in 10 msec using `eq`.

Zhang's example: `zhang_la`

Given the lemma:

$$A > 0 \Rightarrow rem(A * B, A) = 0$$

then the following formula holds over integers,

$$rem(x * y, x) \neq 0 \Rightarrow x \leq 0$$

RDL proves the validity of this example in 10 msec using `la`.

Boyer and Moore's example: bm94

Given the lemma:

$$min(A) \leq max(A)$$

then the following formula holds over integers,

$$l \geq 0 \wedge k > 0 \wedge l \leq min(b) \Rightarrow l < max(b) + k$$

RDL proves the validity of this example in 10 msec using aug(la).

1 step of augmentation is applied.

Boyer and Moore's example: bm102

Given the lemmas:

$$0 < A \Rightarrow B \leq A * B$$
$$0 < ms(C)$$

then the following formula holds over integers,

$$ms(c) + ms(d)^2 + ms(b)^2 <$$
$$ms(c) + ms(b)^2 + 2 * ms(d)^2 * ms(b) + ms(d)^4$$

RDL proves the validity of this example in 30 msec using aug(la).

9 steps of augmentation are applied.

Kapur and Nie's example: kapur_nie_11

Given the lemmas:

$$max(A, B) = A \Rightarrow min(A, B) = B$$
$$p(C) \Rightarrow f(C) \leq g(C)$$

then the following formula holds over integers,

$$(p(x) \wedge z \leq f(max(x, y)) \wedge 0 < min(x, y) \wedge$$
$$x \leq max(x, y) \wedge max(x, y) \leq x)$$
$$\Rightarrow z < g(x) + y$$

RDL proves the validity of this example in 60 msec using aug(la).
2 steps of augmentation are applied.

Bjorner's non-linear example: nl-example-bjorner

The following formula holds over integers,

$$x > 0 \wedge y > 0 \wedge z > 0 \Rightarrow x * y * z + x * y * z * z > 0$$

RDL proves the validity of this problem in 10 msec using aff(la).

3 steps of affinization are applied.

Boyer and Moore's example: `nl-1-termination`

This example differs from the previous Boyer and Moore's example in that the lemma about multiplication is not provided here to the augmentation process.

We only give the lemma encoding the positivity of ms:

$$0 < ms(A)$$

The formula

$$ms(c) + ms(d)^2 + ms(b)^2 <$$
$$ms(c) + ms(b)^2 + 2 * ms(d)^2 * ms(b) + ms(d)^4$$

is proved to be valid in 20 msec using `aug_aff(la)`.

6 steps of augmentation and affinization are applied.

State of the art

- **Acl2/NQTHM** and **Tecton**: rewriting [augmentation [LA]]
- **RDL**: rewriting [lemma speculation [EQ, LA, EQ+LA,. . .]]

State of the art

- **Acl2/NQTHM** and **Tecton**: rewriting [augmentation [LA]]

State of the art

- **Acl2/NQTHM** and **Tecton**: rewriting [augmentation [LA]]
- **RDL**: rewriting [lemma speculation [EQ, LA, EQ+LA,. . .]]
- **Simplify**: matching algorithm [EQ+LA]

State of the art

- **Acl2/NQTHM** and **Tecton**: rewriting [augmentation [LA]]
- **RDL**: rewriting [lemma speculation [EQ, LA, EQ+LA,. . .]]
- **Simplify**: matching algorithm [EQ+LA]
- **SVC** and **STeP**: Propositional/FOL Reasoner [EQ+LA+$\cdots$]

State of the art

- **Acl2/NQTHM** and **Tecton**: rewriting [augmentation [LA]]
- **RDL**: rewriting [lemma speculation [EQ, LA, EQ+LA,. . .]]
- **Simplify**: matching algorithm [EQ+LA]
- **SVC** and **STeP**: Propositional/FOL Reasoner [EQ+LA+$\cdots$]
- **PVS**: [rewriting] [EQ+LA+$\cdots$] [instantiation of lemmas]

Preliminary comparison

System	Problem type			
	eq	**la**	**aug(la)**	**aug_aff(la)**
RDL	100%	100%	100%	100%
Simplify	100%	100%	100%	6%
SVC	100%	100%	0%	11%

where:

- **eq:** ground equality problems (8 pbs)
- **la:** linear arithmetic problem (10 pbs)
- **aug(la):** LA problem with user supplied facts (8 pbs)
- **aug_aff(la):** non LA problem with user supplied facts (35 pbs)

2 Jacques Calmet

University of Karlsruhe, Germany

Course: Introduction to Computer Algebra Systems

Introduction to Computer Algebra Systems

Jacques Calmet

University of Karlsruhe

A – Topics

1. Generalized versus specialized CAS,
2. Data structures, normal and canonical forms,
3. Simplification,
4. Basic concepts in algebraic algorithm design.

B- Literature

Topic 1

Michael J. Wester (ed.) „Computer Algebra Systems - A Practical Guide", John Wiley & Sons (1999).

H. van Hulzen and J. Calmet in „Computer Algebra". Eds B. Buchberger, G.E. Collins and R. Loos, Springer Verlag, 2nd edition (1983).

Topic 2 and 3

B. F. Caviness, „On Canonical Forms and Simplification". PhD thesis, Carnegie-Mellon University, May 1968.

R.J. Fateman, „Essays in Algebraic Simplification". PhD thesis, MIT, Project Mac, April 1972.

Topic 4

J.D. Lipson, „Elements of Algebra and Algebraic Computing". Benjamin/Cummings (1981).

K.O. Geddes, S.R. Czapor and G. Labahn, Algorithms for Computer Algebra), Kluwer Academic Publication (1992).

Many other books are now available. Some authors are: J. von zur Gathen, J.H. Davenport et al. (the first published book on Computer Algebra, R. Zippel, A. Cohen, M. Mignotte, A.G. Akritas), G. Gonnet.

Jacques Calmet, Computer Algebra script (in German) at:
http://iaks-www.ira.uka.de/iaks-calmet/vorlesungen/compalg00/ca96-1.ps.gz

Latest reference

J. Grabmeier, E. Kaltofen & V. Weispfenning

"Computer Algebra Handbook" (2002)

Springer verlag

{1st version in German (a few years ago)}

. 200 contributors

. 63 systems (CD of demos)

. out of 634 pages, 100 for applications.

A brief history

Ref : Calmet - Campbell in Annals of Math. and AI (92)

Phase I until 1971

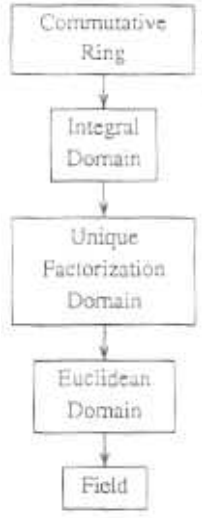
1956 : 2 master thesis (algorithms instead of heuristics symbolic)

1971 : First major conference (Los Angeles)
=> Field is established

- Lisp (John McCarthy)
- IBM support => ACM group SIGSAM (Jean Sammet systems)
- Need for symbolic computation in theoretical physics, General relativity, Celestial mechanics, Engineering ...

  Link to coding theory through finite fields

Table 2.3. Hierarchy of domains.



Notation: Downward pointing arrows indicate that a former domain becomes a latter domain if additional axioms are satisfied.

- Systems delivers result

  - Artificial Intelligence

    Joel Moses - MACSYMA (MIT) - Lisp

  - Particle Physics

    REDUCE (A.C. Hearn) Lisp

    SCHOONSCHIP (T. Veltmann CERN)

    today FORM (in any Physics Dept.)

(Note: A Nobel prize winner)

  - Algorithm (and logic)

    - SAC I (G.E. Collins IBM then Univ of Wisconsin) Fortran

  - Many others ...

Phase II 1971 → 1981

  - Maturation
  - Issues
    - validation (debugging)
    - portability (Lisp, C)
  - C → Maple (K. Geddes et al.)

    (Students rather than users)
  - Commercial system

    Mathematica (S. Wolfram)

- Surveys of systems (63 in 1979) and applications still possible

Phase III 1981 → ≃ 1995

  - Algorithms design
  - Commercially (or academically) established systems
    - Maple, Mathematica, Derive
    - Mupad, Reduce, Magma (Cayley)
    - FORM
    - GAP, Pari ..
  - Personal work stations

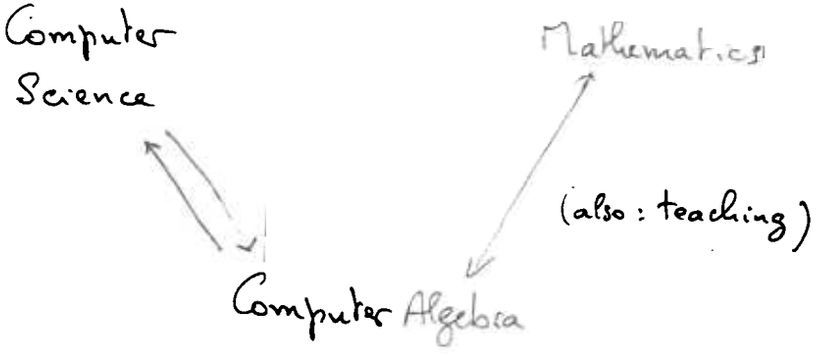
- First truly European conference
  FINANCAM '93 in Marseilles
- ...s
  (Journal
  Applicable

Phase IV

...ing
ECC)...)

...g...
Interoperability
Web technology
Renewed standardization attempts

Deduct...



(The following opinions are personal ones!)

CS <-> CA <-> Math

or

Mechanization of Mathematics

- In CS we work with models (semantics, specifications, programming languages, paradigm for knowledge representations, protocols in communication processes ....)

  or in other words: constraint programming

- In CA these models are hidden in modules (operators acting on a given domain) of algorithms

- Math could be a demanding domain for CS methodology design.

- There is no unique typing model for Math.
  (-> "constraints" again)

European Projects

- Cathode (Network)
- Posso (Network)
- OpenMath (Project)
- Calculemus (Network)

INTAS projects

Project coordinated by

- J. Fitch (Univ. Bath)
- B. Buchberger (?)
- J. Calmet (Over- and under-determined systems of Polyn. & Differ Equations)

Others ??

Algebraic Algorithms

- Computation ≡ { operator + domain }
- Efficiency is required →

1. Representation of "objects" is very important (numbers, polynomials, matrices)

2. Transform (often homomorphically) a problem

   Ex. factorization of integer polynomial → factorization over finite fields

3. "Art of programming"

   Ex: gcd computation through pseudo-resultant sequences

4. Complexity analyses (a highly non-trivial game)

Computer Algebra Systems

Sketch:

(1) Collection / Modules / Libraries of algebraic algorithms

(2) Programming language

=> issues (at least some issues)

1) Openess / availability of algorithm
(Workshop last May in Lyon)
Possible ALDES / SAC2 (Linz)
SACLIB

(2) • Data structures
• Commands (solve, substitute ...)
• simplification : the most important decision of a designer (this is why today most systems look alike!)

[An approach to design a "context"]

5 Possibly many years between algorithm discovery / design and efficient implementation

Ex. • Factorization of polynomial
(Musser's dissertation - Berlekamp's algorithm)
Indefinite integration
(Risch's algorithm - 10 years after: Norman. Davenport's implementation)
Closed-form solution of homogeneous linear ODE's : generic algorithm (Singer then optimality by Ulmer in 1990) - No ~~beyond~~ implementation beyond 2nd order yet.

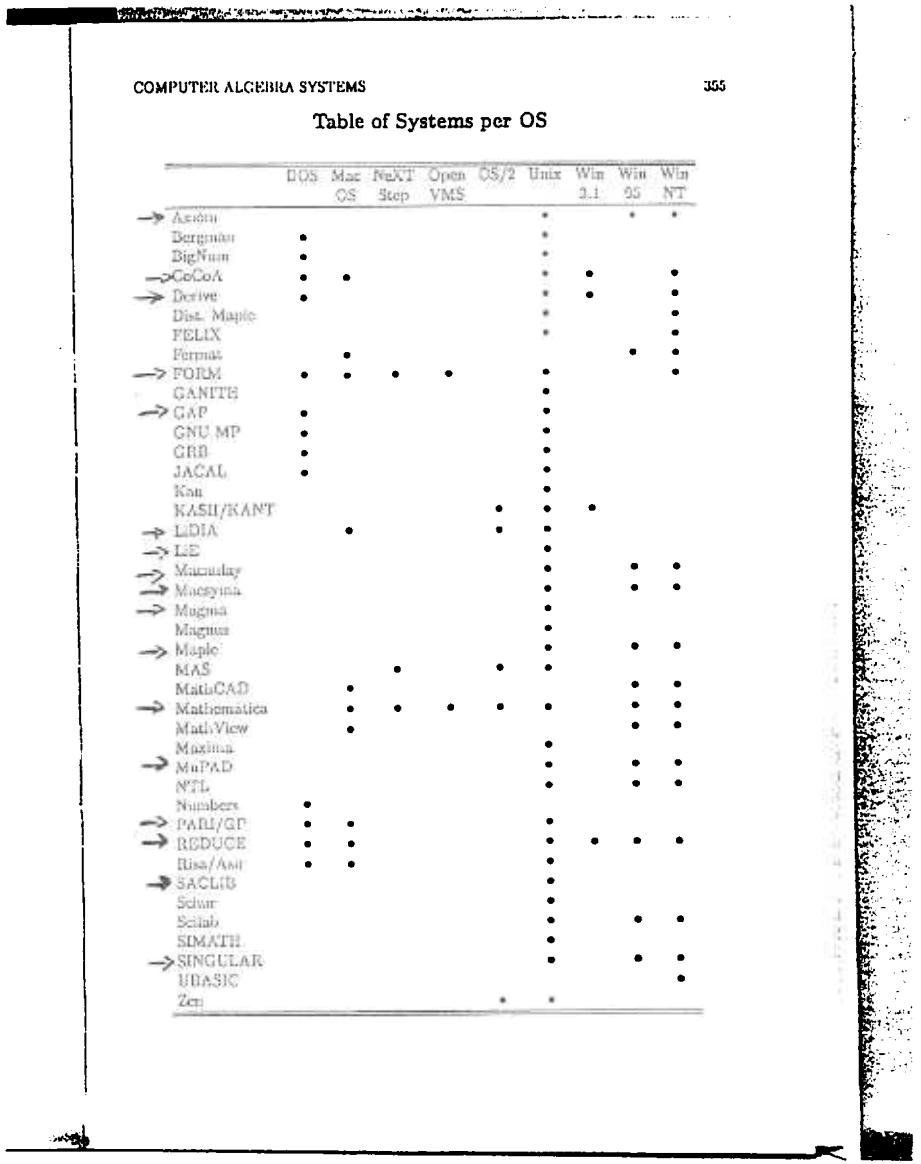
• Groebner bases

General-purpose versus specialized systems

- ∃ systems specialized to most applications
  - Example : tensoss
  - often access restricted (design of wings of plane for instance)
  - Not always well debugged
- For an in-depth, long, simple, tedious application : design your own system.

[Arthur Norman : " I could design a new system every week "]

- you avoid a great deal of code you would never use . "Triming" (software re-use) an existing CAS is never simple,
- you probably know better than the system designer how to conduct your computation



COMPUTER ALGEBRA SYSTEMS 355

Table of Systems per OS

	DOS	Mac OS	NeXT Step	Open VMS	OS/2	Unix	Win 3.1	Win 95	Win NT
→ Axiom						•		•	•
Bergman	•					•			
BigNum	•					•			
→CoCoA	•	•				•	•		•
→ Derive	•					•	•		•
Dist. Maple						•			•
FELIX						•			•
Fermat		•						•	•
→ FORM	•	•	•	•		•			•
GANITH						•			
→ GAP	•					•			
GNU MP	•					•			
GRB	•					•			
JACAL	•					•			
Kan						•			
KASH/KANT					•	•	•		
→ LiDIA		•			•	•			
→ LiE						•			
→ Macsyma						•		•	•
→ Magma						•		•	•
Magnus						•			
→ Maple						•		•	•
MAS			•		•	•			
MathCAD		•						•	•
→ Mathematica		•	•	•	•	•		•	•
MathView		•						•	•
Maxima						•		•	•
→ MuPAD						•		•	•
NTL						•		•	•
Numbers	•								
→ PARI/GP	•	•				•			
→ REDUCE	•	•				•	•	•	•
Risa/Asir	•	•				•			
→ SACLIB						•			
Solvir						•			
Scilab						•		•	•
SIMATH						•			
→SINGULAR						•		•	•
UBASIC									•
Zen					•	•			

A very special system : KENZO

- Algebraic topology
- Author : S. Sergeraert (Grenoble)
  with help from Y. Siret and J Rubio
  (Univ. of Rioja at Logrono)

- also noticeable : typing, specification system
  (only CAS with a non-trivial one)

Note : The concept of infinity in CAS

- Algebra : it plays no role
- { Analysis → Algebra } : difficulties (limits)
- Algebraic geometry : exists but no serious pbs
- Algebraic topology : plays a role in proofs
  and thus design of algorithms

– Capabilities –

#	PROBLEM	Ax	De	Mc	Mp	Mm	Mu	Re
	A. Boolean Logic and Quantifier Elimination							
A1	true and false ⇒ false	•	•	•	•	•	•	•
A2	x or ¬x ⇒ true		•	•	•	•	•	•
A3	x or y or (x and y) ⇒ x or y		•	•	•	•	•	•
A4	x xor y xor y ⇒ x		•		•	•	—	—
A5	(w and x) implies (y and z)		•	—	•	•	—	•
A6	x iff y ⇒ (x and y) or ¬(x or y)	—	—	—	•	—	—	—
A7	x and 1 > 2 ⇒ false		•	ε	•	•	•	∘
A8	$\forall \lambda \in C\ (a\lambda^2 + b\lambda + c = 0$ implies ...)	—	—	—	—	—	—	—
A9	$\exists w \in \mathbb{R}\ \{v > 0$ and $w > 0 \ldots\} \Rightarrow v > 1$	—	—	—	—	—	—	—
A10	$\forall c \in \mathbb{R}\ \{-1 \le c \le 1$ implies ... $\le 4\}$	—	—	—	—	—	—	—
A11	$\forall y \in C\ \{x > 0 \ldots$ implies $\mathrm{Re}\, y < 0\}$	—	—	—	—	—	—	—
A12	$4\omega v > 0$ and $4w(\ldots) > 0 \ldots \Rightarrow w > \lvert W\rvert^{\frac{1}{4}}$	—[2]		ε	$\mathcal{L}$			
A13	$\exists y, n \in C, \exists e \in \mathbb{R} \ni \{\mathrm{Re}\, y > 0$ and ... $\}$	—	—	—	—	—	—	—
	B. Set Theory							
B1	$\{a,b,b,c,c,c\} \bigcup \{d,c,b\} \bigcup \{b,e,b\}$	*	•	•	•	•	•	•
B2	$\{a,b,b,c,c,c\} \bigcap \{d,c,b\} \bigcap \{b,e,b\} \Rightarrow \{b\}$	*	•	•	•	•	•	•
B3	$\{a,b,c,d,c\} - \{b\} \Rightarrow \{a,c,d,c\}$	•	•	•	•	•	•	•
B4	$\{a,b\} \times \{c,d\} \Rightarrow \{ac, ad, bc, bd\}$	a^d	a^d	*	a^d	*	•	a^d
	C. Numbers							
C1	$50! \Rightarrow 3.0414093201 \ldots 0000000000 \cdot 10^{64}$	•	•	•	•	•	•	•
C2	factor(50!) $\Rightarrow 2^{47}3^{22}5^{12}7^{8}11^{4}13^{3} \cdots 47$	•	•	•	•	•	•	*
C3	$10! \Rightarrow 3840$; $9! \Rightarrow 945$	—	—	•	—	•	—	—
C4	$ABC_{16} \Rightarrow 2748_{10}$	—	•		b	•	—	—
C5	$123_{10} \Rightarrow 234_7$	•	•	•	∘	•	—	—
C6	$677_8 \Rightarrow 1BF_{16}$ $(= 447_{10})$	#	•	•	b	•	—	—
C7	$\log_8 32768 \Rightarrow 5$		•	•	•	•		
C8	$5^{-1} \bmod 7 \Rightarrow 3$; $5^{-1} \bmod 6 \Rightarrow 5$	•|*	•	*	•	•	•	•|ε
C9	gcd(1776, 1554, 5698) ⇒ 74	*	•	•	•	•	•	*
C10	$\frac{1}{2} + \cdots + \frac{1}{10} \Rightarrow \frac{4861}{2520}$	•	•	•	•	•	•	•

Originally published in *Computer Algebra Systems: A Practical Guide* edited by Michael J. Wester, John Wiley & Sons, Chichester, United Kingdom, ISBN 0-471-98353-5, xvi+436 pages, 1999.

Note: CA + Logic

Recently: REDLOG (Weispfenning in Passau)

In the past (70's) similar program implemented in Japan. Totally forgotten,

This is not an isolated feature, CA rediscover past work

1) Some packages are simply lost and forgotten

2) Some math results are rediscovered (differential equations domain)

#	PROBLEM	Ax	De	Mc	Mp	Mm	Mu	Re
C11	$N(\frac{1}{7}) \Rightarrow 0.\overline{142857}$	•	—	—	—	•	•	•²
C12	$N(\frac{1}{7})N(\frac{7}{1}) = 0.\overline{142857} \cdot 7 \Rightarrow 1$	•	—	—	—		—	•
C13	$\sqrt[3]{\ldots}\sqrt{1+\frac{\ldots}{1000}} \Rightarrow \sqrt[3]{\ldots}$	•	•	•	•	•	•	•
C14	$\sqrt{2\sqrt{3}+4} \Rightarrow 1+\sqrt{3}$		•	•	•	•	•	•
C15	$\sqrt{14+3\sqrt{3+\cdots}} \Rightarrow 3+\sqrt{2}$		•	•	•	•	•	•
C16	$\sqrt{10+2\sqrt{6}+2\sqrt{10}+2\sqrt{15}} \Rightarrow \sqrt{2}+\cdots$		•					
C17	$\frac{\sqrt[4]{\ldots}+\sqrt{\ldots}}{\sqrt{\ldots}-\sqrt{\ldots}} \Rightarrow 3+2\sqrt{6}$	•	•	•	•	•	•	
C18	$\sqrt{-2+\sqrt{-3}}\sqrt{-2-\sqrt{-3}} \Rightarrow 3$		•	•	•	•	•	
C19	$\sqrt[3]{90+34\sqrt{7}} \Rightarrow 2+\sqrt{7}$		•		•	•	•	
C20	$\frac{[(48+78\sqrt{3})^{1/3}+\ldots]\sqrt{3}}{(48+78\sqrt{3})^{1/3}} \Rightarrow 12$	#	•	#	•	•	•	
C21	$\sqrt[3]{41+29\sqrt{2}} \Rightarrow 1+\sqrt{2}$							
C22	$\frac{(4-4\sqrt{2})\log(3-2\sqrt{2})+(3-2\sqrt{2})\log(17-12\sqrt{2})+\ldots}{48\sqrt{2}-72}$		•	#	#	#		
C23	$2\infty - 3 \Rightarrow \infty$	•	•	∘	•	•	•	—
C24	$2^{\aleph_0} \Rightarrow \aleph_1$	ε	∘	∘	∘	∘	∘	—
	D. Numerical Analysis[3]							
D1	$\frac{0.0}{\sqrt{\ldots}} \Rightarrow 0.0$ (immediate)	•	•	•	•	•		•
D2	$N(e^{-1000000}) \Rightarrow 3.29683 \cdot 10^{-434295}$	•	r	•	•	•	•	•
D3	$N(e^{\pi\sqrt{163}}, 50) \Rightarrow 262537412640768744.0$	•	•	•	•	•	•	•
D4	$\lfloor -\frac{3}{2} \rfloor \Rightarrow -2,\ \lceil -\frac{3}{2} \rceil \Rightarrow -1$	•	•	•	•	•	•	•
D5	compute cubic spline s; $s(3) \Rightarrow \frac{17}{8}$	—	∘	∘	•	c	—	—
D6	translate $p = \sum_{i=1}^{n} a_i x^i$ to Fortran	ε	× | ⊎	× | ⊎	ε	⊎	⊎	⊎ | ⊎
D7	translate $p = \sum_{i=1}^{n} a_i x^i$ to C	—	⊎ | ⊎	⊎	⊎	×	⊎	⊎
D8	Horner's rule applied to $p = \sum_{i=1}^{4} a_i x^i$	—	—	•	•	•	∘	—
D9	translate above to Fortran	•	∘ | •	• | •	•	∘	•	• | ∘
D10	translate above to C	—	⊗ | •	•	•	c	•	•
D11	$\mathrm{flops}(\sum_{i=1}^{n} \prod_{j=1}^{n} f_{ij})$	—	—	•	×	—	—	—
D12	$([-4,2]x+[1,3])^2$ (interval analysis)	—	—	—	—	•[8]	•	—
D13	$\frac{dy}{dt} = \frac{d^2y}{dt^2} \Rightarrow \frac{f_{i+1}-f_i}{\Delta t} = \frac{f_{i+1}-2f_i+f_{i-1}}{(\Delta t)^2}$	—	—	•	—	—	—	—

Originally published in *Computer Algebra Systems: A Practical Guide* edited by Michael J. Wester, John Wiley & Sons, Chichester, United Kingdom, ISBN 0-471-98353-5, xvi+436 pages, 1999.

#	PROBLEM	Ax	De	Mc	Mp	Mm	Mu	Re
	E. Statistics							
E1	mean($[3,7,11,5,19]$) $\Rightarrow 9$	$\circ^{d}$	•	•	•	•	•	$\circ^{d}$
E2	median($[3,7,11,5,19]$) $\Rightarrow 7$	—	—	•	•	•	•	—
E3	quartile$(1,[1,\ldots,8])$ = quantile$(\frac{1}{4},{}^{n})$ $\Rightarrow 2$	—	—	—	•	$\star^{5}$	—	—
E4	mode($[3,7,11,7,3,5,7]$) $\Rightarrow 7$	—	—	—	•	•	—	#
E5	stdev($[1,2,3,4,5]$) $\Rightarrow \sqrt{\frac{5}{2}}$	—	•	•	•	•	•	—
E6	PDF(discrete binomial distribution)	—	•	•	•	•	—	—
E7	CDF(discrete binomial distribution)	—	•	•	•	•	—	—
E8	CDF(continuous normal distribution)	—	•	•	•	•	—	—
E9	hypothesis testing: t distribution	—	$\frac{1}{2}^{6}$	○	○	•	•	—
E10	hypothesis testing: normal distribution	—	○	○	○	⋆	•	—
E11	compute χ^2 statistic by hand $\Rightarrow \frac{1153}{252}$	•	•	•	•	•	•	•
E12	χ^2 test $\Rightarrow 0.46986$	—	⋆	—	⋆	•	•	—
E13	linear regression $\Rightarrow y' \approx 0.7365x + 6.964$	—	•	•	•	•	—	—
E14	multiple linear regression (2 variables)	—	•	⋆	•	•	—	—
E15	multiple linear regression using L_1 norm	—	—	—	—	—	—	—
E16	nonlinear regression: $w = b_0 + b_1 2^{-d/\theta}$	—	—	$\star^{7}$	—	•	—	—
	F. Combinatorial Theory							
F1	$(a)_3 \Rightarrow a(a+1)(a+2)$ (Pochhammer)	—	•	•	•	•	—	•
F2	$\binom{n}{3} \Rightarrow \frac{n(n-1)(n-2)}{6}$	⋆	•	•	⋆	•		⋆
F3	$2^n n!\,(2n-1)!! \Rightarrow (2n)!$ or $\Gamma(2n+1)$	—	—		—		—	—
F4	$2^n n! \prod_{k=1}^{n} 2k-1 \Rightarrow (2n)!$ or $\Gamma(2n+1)$		#	#	•	•	#	#
F5	$\frac{1'(n+1/2)}{\sqrt{\pi}\,n!} \Rightarrow \frac{(2n)!}{2^{2n}n!^2}$ or $\frac{(2n-1)!!}{2^n n!}$							
F6	partitions of $4 \Rightarrow \{4, 2+2, 1+3, \ldots\}$	—	—	—	•	—	—	—
F7	number of partitions of $4 \Rightarrow 5$	—	—	•	•	•	—	—
F8	$S_1(5,2) \Rightarrow -50$ (Stirling numbers)	•	—	—	•	•	—	•
F9	$\phi(1776) \Rightarrow 576$ (Euler's totient fun.)	•	—	•	•	•	•	—
	G. Number Theory							
G1	discover primes 999983 and 1000003	•	⋆	○	•	⋆	•	—
G2	primitive root of $191 \Rightarrow 19$	—	—	—	•	•	•	—

Originally published in *Computer Algebra Systems: A Practical Guide* edited by Michael J. Wester, John Wiley & Sons, Chichester, United Kingdom, ISBN 0-471-98353-5, xvi+436 pages, 1999.

#	PROBLEM	Ax	De	Mc	Mp	Mm	Mu	Re
G3	$[(a+b)^p \Rightarrow a^p + b^p] \bmod p$ (p prime)	—	—	ε				—
G4	solve$(9x \equiv 15 \bmod 21) \Rightarrow x \equiv 4 \bmod 7$	ε	•	ε	•	•		•
G5	solve$(7x \equiv 22 \bmod 39) \Rightarrow x \equiv 31 \bmod 39$	ε	•	•	•	•		•
G6	solve$(x^2 + x + 4 \equiv 0 \bmod 8)$	ε	—	ε	•	•		•
G7	solve$(x^3 + 2x^2 + 5x + 6 \equiv 0 \bmod 11)$	$\bullet^{m}$	—	•	•	•		$\bullet^{m}$
G8	solve$(x \equiv 7 \bmod 9 \equiv 13 \bmod 23 \equiv \cdots)$	⋆	—	—	⋆	—	⋆	—
G9	solve$([5x + 4y \equiv 6,\ 3x - 2y \equiv 6] \bmod 7)$	•	—	•	ε	•		•
G10	solve$(2x + 3y \equiv 1 \bmod 5, [x, y])$	ε	—	○	ε	○		•
G11	solve$(2x + 3y \equiv 1 \bmod 6, [x, y])$	ε	—	ε	ε	⊗		•
G12	solve$(x^4 + 9 = y^2, \mathbb{Z}) \Rightarrow x, y = 2, 5$	ε		—		—	—	
G13	solve$(x^3 + 4 = y^3, \mathbb{Z}) \Rightarrow x, y = 11, 5$	ε		—		—	—	
G14	solve$([x^2 + y^2 = t^2, t^2 + x^2 = w^2], \mathbb{Z})$	$\mathcal{E}$		—		—	—	—
G15	rational approximation of $\sqrt{3} \Rightarrow \frac{26}{15}$ (e.g.)	•	⋆	•	⋆	•	⋆	○
G16	cf$(3.1415926535) \Rightarrow <3, 7, 15, 1, 292, \ldots>$	•	—	•	•	•	•	•
G17	cf$(\sqrt{23}) \Rightarrow <4, \overline{1, 3, 1, 8}>$	⋆	—	•	•	⋆	⋆	⋆
G18	cf$(\frac{1+\sqrt{5}}{2}) \Rightarrow <\overline{1}>$	⋆	—	⊡	•	⋆	⋆	⋆
G19	cf$(\frac{e^{1/e}-1}{e^{1/e}+1}) \Rightarrow <0, 2x, 6x, 10x, 14x, \ldots>$	—	—	ε	$\mathcal{E}$		$\mathcal{E}$	$\mathcal{E}^{n}$
G20	cf$(\sqrt{x^2+1} - x) \Rightarrow <\overline{2x}>$	—	—	ε	⋆		⋆	$\star^{n}$
	H. Algebra							
H1	$2 \cdot 2^n \Rightarrow 2^{n+1}$		•	$\frac{1}{2}$	•	•	⋆	
H2	$4 \cdot 2^n \Rightarrow 2^{n+2}$		•	○	$\frac{1}{4}$	•		
H3	$(-1)^{n(n+1)} \Rightarrow 1$ ($n \in \mathbb{Z}$)	—		•	•	—		—
H4	factor$(6x - 10) \Rightarrow 2(3x - 5)$	•	•	•	#	•	#	•
H5	univariate gcd $\Rightarrow 1$	•	•	•	•	•	•	•
H6	univariate gcd $\not\Rightarrow 1$	•	•	•	•	•	•	•
H7	multivariate gcd $\Rightarrow 1$ (3 variables)	•	τ	•	•	•	•	$\mathcal{E}$
H8	multivariate gcd $\not\Rightarrow 1$ (3 variables)	•	τ	•	•	•	•	$\mathcal{E}$
H9	gcd$(2x^{n+4} - x^{n+2}, 4x^{n+1} + 3x^n) \Rightarrow x^n$	×	•	•	×	•	×	•
H10	resultant$(3x^4 + \cdots - 2, x^3 - \cdots + 5) \Rightarrow 0$	•	—	•	•	•	•	•

Originally published in *Computer Algebra Systems: A Practical Guide* edited by Michael J. Wester, John Wiley & Sons, Chichester, United Kingdom, ISBN 0-471-98353-5, xvi+436 pages, 1999.

#	PROBLEM	Ax	De	Mc	Mp	Mm	Mu	Re
H11	resultant(expand(p, z), expand(pxq)) ⇒ 0	•	—	•	•	•	•	∠
H12	$\frac{x^3-8}{x^2-4x+4} \Rightarrow \frac{x^2+2x+4}{x-2}$	•	•	•	•	•	•	•
H13	$\frac{x^4-1}{x^2+x+1} \Rightarrow x^{8/3}-1$ [9]	•	•	•	•	•	•	
H14	expand($(x+1)^{20}$) → diff → factor	•	•	•	•	•	•	•
H15	factor(x^3+x^2-7) → expand	•	#	•	•	•	≠	#
H16	factor($x^{100}-1$)	•	•	•	•	•	•	•
H17	factor(expand($(64x^{34}-\cdots)(72x^{60}-\cdots)$))			•	•	•	•	•
H18	factor($4x^4+8x^3+77x^2+18x+53$)	•	•	•	•	•	•	•
H19	let($x^3=2$); $\frac{1}{x-1} \Rightarrow a+1$	•	◦	◦	•	◦		•
H20	let(''); $\frac{x^3+(a-3)x^2-(3a+3)x-8a}{x^3-2} \Rightarrow \frac{x^2-3x-1}{x-a}$	•	◦	◦	•	◦	#	•
H21	let($b^2=2$, $c^2=3$); $(b+c)^4$	•	—	◦	•	◦		—
H22	factor(x^4-3x^2+1 mod 5)	•	—	•	•	•	•	•
H23	factor($x^{13}+x+1$ mod 65537)	•	—	•	•	◦	◦	•
H24	factor(x^3-3x^2+1, RootOf($\phi^2-\phi-1$))	•	•[8]	•	•	◦	◦	—
H25	expand($(x-2y^2+3x^3)^{40}$) → factor	•	◦	•	•	•	•	•
H26	expand($(\sin x-2\cos^3 y+\cdots)^{30}$) → factor		◦	•	•	•	◦	•
H27	factor(expand($(24xy^{10}x^8-\cdots+5)(\ldots)$))	•	#	•	•	•	•	•
H28	expand($\cdots(c^3+x^2)^{10}$) with $c^3+x^2=1$	◦	—	◦	•	◦	◦	◦
H29	factor($4x^2-21xy+10y^2$ mod 3)	∈	—	•	•	#	•	•
H30	factor(x^3+y^3, $i\sqrt{3}$) ⇒ $\frac{x+y}{4}\cdots$	◦	•	•	•	•	◦	#
H31	$\frac{x^5+2x^3+x}{x^4+2x^3+3x^2+2x+1} \Rightarrow \frac{1}{x+1}-\frac{x}{x+1}+\frac{1}{(x+1)^2}$	•	•	•	•	•	•	•
H32	$[ABC-(ABC)^{-1}]ACB$ (noncommuting)	—	•	•			—	×
H33	$[A,B,C]+[B,C,A]+[C,A,B]=0$	—	•	•	•	‡	—	•
	I. Trigonometry							
I1	$\tan\frac{7\pi}{12} \Rightarrow -\frac{\sqrt{6}+1}{\sqrt{6}-1}\frac{\sqrt{6}}{(\sqrt{6}+1)/4} = -\sqrt{1+\frac{3}{\sqrt{4}}}$		•	◦	•	•	#	
I2	$\sqrt{\frac{1+\cos 6}{2}} \Rightarrow -\cos 3$	#	•			•		
I3	$\cos n\pi+\sin\frac{4n-1}{2}\pi \Rightarrow (-1)^n-1$ $(n\in\mathbb{Z})$	—	•	•	•	#	≠	—
I4	$\cos(\pi\cos n\pi)+\sin(\frac{\pi}{2}\cos n\pi)$ $(n\in\mathbb{Z})$	—	•	•	◦	—		—
I5	$\sin([\frac{n^6}{5}+\frac{n^5}{2}+\frac{n^4}{2}-\frac{n}{10}]\pi) \Rightarrow 0$ $(n\in\mathbb{Z})$	—			•	—		—

Originally published in *Computer Algebra Systems: A Practical Guide* edited by Michael J. Wester, John Wiley & Sons, Chichester, United Kingdom, ISBN 0-471-98353-5, xvi+436 pages, 1999.

#	PROBLEM	Ax	De	Mc	Mp	Mm	Mu	Re
I6	$\cos x\,(\sin x)$ $(-3\pi<x<-\frac{17}{4}\pi)$	—	◦		•	—		—
I7	$\frac{\cos 3x}{\cos x} \Rightarrow \cos^2 x-3\sin^2 x$ (or similar)	◦	•	•	•	•	•	•
I8	$\frac{\cos 2x}{\cos x} \Rightarrow 2\cos 2x-1$	‡	•	◦	◦	◦	◦	◦
I9	$\frac{\cos 2x}{\cos x} \Rightarrow \cos^2 x-3\sin^2 x$ (rewrite rules)	•	—	•	•	•	‡	•
I10	simplify($\frac{\tan^3 x+1-\sec^2 x}{\tan^3 x+\cos^2 x-1}$) ⇒ $\frac{0}{0}$, error or ?	⊗[10]	•	×	×	•	•	•
I11	$\lim_{x\to 0}\frac{\tan^3 x+1-\sec^2 x}{\tan^3 x+\cos^2 x-1} \Rightarrow$ error or ?	×	•	•	•	•	•	•
I12	$\frac{d}{dx}\left(\frac{\tan^3 x+1-\sec^2 x}{\tan^3 x+\cos^2 x-1}\right) \Rightarrow$ error or ?	⊗[10]	•	×	×	×	⊞	•
	J. Special Functions[11]							
J1	$B_{18} \Rightarrow -\frac{43,867}{798}$ (Bernoulli number)	•	•	•	•	•	•	•
J2	$\frac{\partial\Gamma(\phi,x)}{\partial\phi} \Rightarrow \frac{\partial(\Gamma(\phi,x)-\Gamma(\phi,x))}{\partial\phi}$ [12]	—	◦	•	•	•	—	×
J3	$\frac{\partial}{\partial x}\operatorname{Gi}x \Rightarrow -\pi^2\operatorname{ai}x\operatorname{Gi}x$	—	—	•	•		—	
J4	$\Gamma(-\frac{1}{2}) \Rightarrow -2\sqrt{\pi}$	α^{π}	•	•	•	•	•	•
J5	$\psi(\frac{1}{3}) \Rightarrow -\gamma-\frac{\pi}{2}\sqrt{3}-\frac{3}{2}\log 3$	α^{π}	#	•	•			
J6	$\mathrm{N}(J_0(1+i)) \approx 0.04158-0.24740i$	•	•	•	•	•	•	•
J7	$J_{-4/3}(\frac{\pi}{3}) \Rightarrow \frac{1}{2}\sqrt{3}$	α^{π}	×	•	•	•		
J8	$J_{7/2}(x) \Rightarrow \sqrt{\frac{2}{\pi x}}(\frac{\sin x}{x}-\cos x)$		$\alpha^{1/2}$	•	•	•		
J9	$\frac{d}{dx}J_0(x) \Rightarrow -J_1(x)$	•	∠	#	•	•	•	•
J10	$P_x^y(0) \Rightarrow \frac{2^y\sqrt{\pi}}{\Gamma(\frac{1-x+y}{2})\Gamma(\frac{1-x-y}{2})}$	—		•	⊞		—	•
J11	$P_3^1(x) \Rightarrow -\frac{3}{2}\sqrt{1-x^2}(5x^2-1)$	—	•	•		•	—	•
J12	$T_{\cos(x)}-2xT_{\cos(x)}+T_{\cos(x)} \Rightarrow 0$	•	∠	◦	•	•	‡	•
J13	$T_n(-1) \Rightarrow (-1)^n$ $(n\in\mathbb{Z}\geq 0)$	∈		•		◦	∠	•
J14	$F(\frac{1}{4},\frac{1}{2},\frac{3}{2};x^2) \Rightarrow \frac{\operatorname{arctanh}x}{x}$	—	⊟	•	•	•	•	—
J15	$F(\frac{3n+3}{2},-\frac{n-3}{2};\frac{3}{2};\sin^2 x) \Rightarrow \frac{\sin 3x}{n\sin x+\cos x}$	—		•	◦	•	•	—
J16	$\zeta'(0) \Rightarrow -\frac{1}{2}\log 2\pi$	—		•	•	•		
J17	$\int_0^3 f(\frac{x+3}{3})\delta(\frac{x-3}{2})-g(x)\delta'(x-1)\,dx$	—	—	◦	•	•		—
J18	define an antisymmetric function f	—	—	•	•	◦	—	•
	K. The Complex Domain							
K1	$[\mathrm{Re}(x+iy),\ \mathrm{Im}(x+iy)]$ $(x,y\in\mathbb{C})^{\mathbb{R}}$	∈	‡	•		•	•	•
K2	$[3-\sqrt{7}+i(6\sqrt{7}-15)]^{1/3} \Rightarrow i$	•	•	•	•	•	•	

Originally published in *Computer Algebra Systems: A Practical Guide* edited by Michael J. Wester, John Wiley & Sons, Chichester, United Kingdom, ISBN 0-471-98353-5, xvi+436 pages, 1999.

#	PROBLEM	Ax	De	Mc	Mp	Mm	Mu	Re
K3	$\left\|\frac{1}{a+ib}\right\| \Rightarrow \frac{1}{\sqrt{a^2+(1/a+b)^2}}$ $(a,b \in \mathbb{R})$	•	•	•	•		•	—
K4	rectform$(\log(3-4i)) \Rightarrow \log 5 + i\tan^{-1}\frac{4}{3}$	•	•	•	•	•	•	
K5	rectform$(\tan(x+iy))$	○	•	•	•	•	•	
K6	$\frac{\sqrt{xy}\,\lvert x\rvert^2}{\sqrt{x}\,\lvert x\rvert} \Rightarrow \frac{\sqrt{xy}}{\sqrt{x}} \Rightarrow \sqrt{y}$ [8]		•	•	•	$*^{14}$	•	•
K7	$\frac{\sqrt{xy}\,\lvert x\rvert^2}{\sqrt{x}\,\lvert x\rvert} \Rightarrow \sqrt{y}$ $(y \geq 0)$	—	•	•		•	•	—
K8	$\sqrt{\frac{1}{x}} = \frac{1}{\sqrt{x}} \Rightarrow 0$ $(x$ is not real negative$)$ [8]	×	$*^{13}$	•	•	•	⊟	×
K9	$\sqrt{\frac{1}{x}} = \frac{1}{\sqrt{x}} \Rightarrow 0$ $(x > 0)$	—	•	•	•	•	•	—
K10	$\sqrt{\frac{1}{x}} \neq \frac{1}{\sqrt{x}} \Rightarrow 0$ $(x < 0)$	—	•	•	•	—		—
K11	$\sqrt{e^x} - e^{x/2} \Rightarrow 0$ $(-\pi < \operatorname{Im} x \leq \pi$, etc.$)$ [8]	•	•	•	•	•	×	×
K12	$\sqrt{e^x} - e^{x/2} \Rightarrow 0$ $(x \in \mathbb{R})$		•	•	•	•	•	•
K13	$\sqrt{e^{2i}} \Rightarrow -e^{2i}$ (principal value)	○	•	•	•	○	•	○
K14	$\log e^x \Rightarrow x$ $(-\pi < \operatorname{Im} x \leq \pi)$ [8]	×	•	•	•	•	⊗	×
K15	$\log e^x \Rightarrow x$ $(x \in \mathbb{R})$	•	•	•	•	•	•	•
K16	$\log e^{10i} \Rightarrow (10-4\pi)i$ (principal value)	×	•	•	•	•	○	
K17	$(xy)^{1/n} - x^{1/n}y^{1/n} \Rightarrow 0$ $(\operatorname{Re} x, y > 0)$ [8]	⊟	•	•	•	$*^{14}$	•	•
K18	$(xy)^{1/n} - x^{1/n}y^{1/n} \Rightarrow 0$ $(y > 0)$	—	•	•	•	$*^{10}$	⊗	—
K19	$(xy)^n - x^n y^n \Rightarrow 0$ $(n \in \mathbb{Z})$	—		•	•	—		—
K20	$\tan^{-1}(\tan x) \Rightarrow x$ $(-\frac{\pi}{2} < x \leq \frac{\pi}{2})$ [8]	⊗	$*^{14}$	•	•	$*^{14}$	•	•
K21	$\tan^{-1}(\tan x) \Rightarrow x$ $(-\frac{\pi}{2} < x < \frac{\pi}{2})$	—		•		—		—
K22	$\tan^{-1}(\tan 10) \Rightarrow 10 - 3\pi$ (princ. value)		•	•	•	•	•	#
K23	$\tan^{-1}[\tan(11+3i)] \Rightarrow 11 - 4\pi + 3i$		?	•	•	•	•	
K24	$\tan^{-1}[\tan(11.0+30.0i)] \approx -1.56637 + 3i$	ℰ		•	×	×	ℰ	
K25	$u = \frac{v+\frac{1}{v}}{2}$; $v + \sqrt{u+1}\sqrt{u-1} \Rightarrow x$ or $\frac{1}{x}$			○	○			
	L. Determining Zero Equivalence							
L1	$\sqrt{997} - (997^2)^{1/3} \Rightarrow 0$	•	•	•	•	•	•	•
L2	$\sqrt{999083} - (999083^3)^{1/3} \Rightarrow 0$	•	•	•	•	•	•	•
L3	$(2^{1/3}+4^{1/3})^3 - 6(2^{1/3}+4^{1/3}) - 6 \Rightarrow 0$	•	•	•	•	•	•	•
L4	$\cos^2 x + \cos x \sin^2 x - \cos x \Rightarrow 0$	•	•	•	•	•	•	#
L5	$\log(\tan(\frac{1}{2}x + \frac{\pi}{4})) - \sinh^{-1}(\tan x) \Rightarrow 0$		#	$\dagger^{17}$	?			

Originally published in *Computer Algebra Systems: A Practical Guide* edited by Michael J. Wester, John Wiley & Sons, Chichester, United Kingdom, ISBN 0-471-98353-5, xvi+436 pages, 1999.

#	PROBLEM	Ax	De	Mc	Mp	Mm	Mu	Re
L6	derivative of above in 0 & above at 0 is 0	#	≠	$*^{17}$	•	○	#	•
L7	$\log \frac{2\sqrt{x^2+1}}{\sqrt{(x+i\sqrt{x^2+1})^3}} \Rightarrow 0$		○	•	•	○	•	•
L8	$[4x+4\sqrt{x}+1]^{\frac{\sqrt{x}}{\sqrt{x}+1}}(2\sqrt{x}+1)^{\frac{x}{\sqrt{x}+1}}\ldots$	•	○			○	•	
L9	$2^{1-x}\Gamma(x)\zeta(x)\cos\frac{\pi x}{2} - \pi^x\zeta(1-x) \Rightarrow 0$	—				•		
	M. Equations							
M1	$\frac{x-1}{x} = (1-\frac{1}{x}) \Rightarrow \frac{x}{2} = 1 = 2$	•		•	•	○	•	•
M2	solve$(3x^3 - 18x^2 + 33x - 19 = 0, \mathbb{R})$	○	•	•	•	○	•	•
M3	solve$(x^4 + x^3 + x^2 + x + 1 = 0)$	?	•	•	•	○	•	•
M4	verify a solution of the above	•	•	•	•	•	•	•
M5	solve$(x^5 - 2x^4 - 4x^3 + 27x^2 - 26x - 23)$							
M6	solve$(x^7 - 1 = 0) \Rightarrow x = \{1, (e^{2\pi i k/7}]_{k=1}^{3}\}$	?	•	•	•	•	?	?
M7	solve$(x^4 - 8x^3 + \cdots - 140x + 46 = 0)$			•	•	•	•	•
M8	solve$(a^{14} + 2a^8 + 1 = x,\ x)$	a^x	a^x	•	a^x	a^x	$*^{14}$	•
M9	solve$(x^{x^{-2}} = x^{x^{-3}}) \Rightarrow x = \{-1, 1\} + C$		a^x	•	a^x	a^x	$*^{18}$	
M10	solve$(e^x = x) \Rightarrow x = -W_n(-1)$ $(n \in \mathbb{Z})$			a^x	x^x	a^x		a^x
M11	solve$(x^x = x) \Rightarrow x = \{-1, 1\}$				○	○		
M12	solve$((x+1)(\sin^3 x + 1)^3 \cos^2 3x = 0)$		$a^{\sin x}$	•	$a^{\sin x}$	$a^{\sin x}$	$a^{\sin}$	•
M13	solve$(\sin x = \cos x) \Rightarrow x = \frac{\pi}{4}[+n\pi]$	a^x	a^x	•	•	a^x	x	•
M14	solve$(\tan x = 1) \Rightarrow x = \frac{\pi}{4}[+n\pi]$	a^x	a^x	•	•	a^x	•	•
M15	solve$(\sin x = \frac{1}{2}) \Rightarrow \frac{\pi}{6}, \frac{5\pi}{6}[+n2\pi, +n2\pi]$	$\frac{\pi}{6}^x$	a^x	•	•	a^x	•	•
M16	solve$(\sin x = \tan x) \Rightarrow 0, 0[+n\pi, -n2\pi]$	$a^{\sin x}$	a^x	•	$a^{\sin x}$	$a^{\sin}$	•	⊗*
M17	solve$(\sin^{-1} x = \tan^{-1} x) \Rightarrow x = [0, 0, 0)$	$a^{\sin}$	$a^{\sin}$	$\frac{\pi}{4}^{\sin}$	$a^{\sin}$	$a^{\sin}$		•
M18	solve$(\cos^{-1} x = \tan^{-1} x) \Rightarrow x = \sqrt{\frac{\sqrt{5}-1}{2}}$	$\frac{\pi}{5}^{\sin}$	•	•	•	○		$\frac{\pi}{4}^{\sin}$
M19	solve$(\frac{x-1}{x+1} = 0) \Rightarrow x = 2$	•	•	•	•	•	•	•
M20	solve$(\sqrt{x^2+1} = x-2) \Rightarrow x = \{\}$	x^x	$a^{\sin}$	•	•	•	•	•
M21	solve$(x + \sqrt{x} = 2) \Rightarrow x = 1$	⊗*	•	•	•	•	•	•
M22	solve$(2\sqrt{x} + 3x^{1/4} - 2 = 0) \Rightarrow x = \frac{1}{16}$	$\frac{\pi}{5}^{\sin}$	•		•	•		•
M23	solve$(x = \frac{1}{\sqrt{1+x^2}}) \Rightarrow \{\sqrt{\frac{\sqrt{5}-1}{2}}, -i\sqrt{\frac{\sqrt{5}+1}{2}}\}$	⊗*		•	•	•	•	○
M24	solve$(\binom{m}{2} 2^k = 1, k) \Rightarrow \log_2 \frac{2}{m(m-1)}$ $(k \in \mathbb{R})$	•	•	•	•	•	•	•

Originally published in *Computer Algebra Systems: A Practical Guide* edited by Michael J. Wester, John Wiley & Sons, Chichester, United Kingdom, ISBN 0-471-98353-5, xvi+436 pages, 1999.

#	PROBLEM	Ax	De	Mc	Mp	Mm	Mu	Re
M25	solve$(ab^x = cd^x, x) \Rightarrow \frac{\log(c/a)}{\log(b/d)}$		•	•	•	⊡		
M26	solve$(\sqrt{\log_b x} = \log_b \sqrt{x}) \Rightarrow x = \{1, e^4\}$	⊗[9]	•	5	•	•	•	
M27	solve$(\log(\cos^{-1}(\sin^{-1}(x^{2/3} - \phi) - 1)) = 2)$	•		9	•	•		•
M28	N(solve$(5x + e^{x-4/7} = 3x^2)$)	—	◦	3	◦	◦	◦	◦
M29	solve$(\lvert x - 1\rvert = 2) \Rightarrow x = \{-1, 3\}$		•	•	•	•	•	•
M30	solve$(\lvert 2x + 5\rvert = \lvert x - 2\rvert) \Rightarrow x = \{-1, -7\}$		•	•	•	•	1	1
M31	solve$(1 - \lvert x\rvert = \max(-x - 2, x - 2)) \Rightarrow \pm\frac{1}{2}$		•		•	•		
M32	solve$(\max(2 - x^2, x) = \max(-x, \frac{x^2}{8}))$	⊗[9]	1		•	•		
M33	solve$(\max(2 - x^2, x) = \frac{x^2}{2}) \Rightarrow x = \{\pm 2, \}$	⊗[9]	1		◦	◦		
M34	solve$((1 + i)x + (2 - i)\bar{x} = -3i, x)$		⊟					
M35	solve$(3x - 2y - z\bar{y} = -3i, [x, y]) \Rightarrow 2 + 3i$			⊟				⊟
M36	solve$(f^3(x) + f(x) - 2 = 0, x)$		•	•	•	•	•	•
M37	solve a 3×3 dependent linear system	•	•	•	•	•	•	•
M38	solve a 189×42 simple linear system	•	•	•	•	×	•	•
M39	solve a 3×3 nonlinear system	•		•	•	⊗	•	•
	N. Inequalities							
N1	is$(e^\pi > \pi^e) \Rightarrow$ true	×	•	•	•	•	•	—
N2	$[$is$(x^4 - x + 1 > 0)$, is$(x^4 - x + 1 > 1)]$	—		◦	⌊•			—
N3	assume$(\lvert x\rvert < 1)$; is$(-1 < x < 1)$	—	—		×	—		—
N4	assume$(x > y > 0)$; is$(2x^2 > 2y^2)$	—	—	•	•	—	•	—
N5	assume$(^*, k > 0)$; is$(kx^2 > ky^2)$	—	—	•	•	—		—
N6	assume$(^*, n > 0)$; is$(kx^n > ky^n)$	—	—			—		—
N7	assume$(x > 1, y \geq x - 1)$; is$(y > 0)$	—	—		•	—	•	—
N8	assume$(x \geq y \geq z \geq x)$; is$(x = y = z)$	—	—	•	•	—	⊟	—
N9	solve$(\lvert x - 1\rvert > 2) \Rightarrow x < -1$ or $x > 3$	—	•	3	•	•	•	5
N10	solve(expand$((x - 1)\cdots(x - 5)) < 0$)	—	•	⊟	•	•	•	•
N11	solve$(\frac{x}{x-3} \leq 3) \Rightarrow x < 3$ or $x \geq 5$	—	•	1	•	•	•	
N12	solve$(\sqrt{x} < 2) \Rightarrow 0 \leq x < 4$	—	•	—	•	⊗	⊗	c
N13	solve$(\sin x < 2) \Rightarrow x \in \mathbb{R}$	—	•	—	⊟[20]	⊟[20]	5	c

Originally published in *Computer Algebra Systems: A Practical Guide* edited by Michael J. Wester, John Wiley & Sons, Chichester, United Kingdom, ISBN 0-471-98353-5, xvi+436 pages, 1999.

#	PROBLEM	Ax	De	Mc	Mp	Mm	Ms	Re
N14	solve$(\lvert\sin x\rvert < 1) \Rightarrow x \neq \frac{\pi}{2}\lvert 1 + n2\pi\rvert$ $(n \in \mathbb{Z})$	—		—		a^r	5	c
N15	solve$(\lvert 2A(\cos t - 1) + 1\rvert \leq 1, A)$	—		—			5	
N16	solve$(A^2(\cos t - 6)^3 \sin^2 t < 0, A)$	—		×			5	
N17	solve$(\lvert x + y > 0, x - y < 3\rvert) \Rightarrow \lvert x\rvert < y$	—		#	#	—		
	O. Vector Analysis							
O1	$\lvert\lvert(1 + i, -2, 3i)\rvert\rvert \Rightarrow \sqrt{15}$	a^d	•	•	•	z^d	•	×
O2	$\langle 2, 2, -3\rangle \times \langle 1, 2, 1\rangle \Rightarrow \langle 11, -5, 4\rangle$	—	•	•	•	•	•	•
O3	$(a \times b)\cdot(c \times d) \Rightarrow (a\cdot c)(b\cdot d) - \cdots$	—		•	—		—	—
O4	$\nabla \times \langle xyz, x^2y^2z^2, y^3z^3\rangle$	—	•	◦	•	•	•	•
O5	$\nabla\cdot(f \times g) \Rightarrow g\cdot(\nabla \times f) - f\cdot(\nabla \times g)$	—			—		—	—
O6	$\nabla\cdot[a_r, a_\theta, a_\phi] \Rightarrow \frac{\partial a_r}{\partial r} + \frac{2a_r}{r} + \frac{1}{r}\frac{\partial a_\theta}{\partial\theta} + \cdots$	—	•	◦	•	•	•	•
O7	$\frac{\partial\vec{r}}{\partial r} \Rightarrow \langle\frac{\partial x}{\partial r}, \frac{\partial y}{\partial r}, \frac{\partial z}{\partial r}\rangle$	—	—	—	—	—	—	—
O8	solve$(\nabla\phi = \langle 2xy, x^2 + 1, 3z^2\rangle, \phi)$	—	•	•	•	—	—	—
O9	solve$(\nabla \times A = \langle 2yz^2 - 2x^2y^2z, xy, \ldots\rangle)$	—	•	•	•	—	•	—
O10	orthogonalize a set of 4 vectors	•	•	•	•	•	—	•
	P. Matrix Theory[21]							
P1	extract superdiagonal of a 3×3 matrix	a^d	a^d	•	a^d	a^d	a^d	a^d
P2	$(2, 3]$-minor of 3×3 matrix $A \Rightarrow \left(\begin{smallmatrix} 1 & 3 \\ 7 & 9 \end{smallmatrix}\right)$	—	•	•	•	—	a^d	•
P3	rand of MATLAB style matrix creation	1	◦	•	•	•	•	◦
P4	diag$(\left(\begin{smallmatrix} 5 & 1 \\ 3 & 4 \end{smallmatrix}\right), 5, \ldots) \Rightarrow$ block diagonal mat.		—	•	•	c	—	1
P5	$\left(\begin{smallmatrix} 7 & 11 \\ 1 & 8 \end{smallmatrix}\right)$ mod $2 \Rightarrow \left(\begin{smallmatrix} 1 & 1 \\ 1 & 0 \end{smallmatrix}\right)$	•	•	•	c	•	c	•
P6	$\frac{d^2}{dt^2}\left(\begin{smallmatrix} \cos t & \sin t \\ -\sin t & \cos t \end{smallmatrix}\right) \Rightarrow -\left(\begin{smallmatrix} \cos t & \sin t \\ -\sin t & \cos t \end{smallmatrix}\right)$	•	•	•	×	•	•	
P7	$\begin{pmatrix} x & y \end{pmatrix}\cdot\left[a\left(\begin{smallmatrix} 1 & 3 & 5 \\ 1 & 4 & 5 \end{smallmatrix}\right) - \left(\begin{smallmatrix} 7 & -6 & 11 \\ -6 & 13 & -13 \end{smallmatrix}\right)\right]$	•	•	•	•	•	•	•
P8	$\lVert\left(\begin{smallmatrix} 1 & -21 \\ -21 & 4 \end{smallmatrix}\right)\rVert_\infty \Rightarrow ?$	—	—	•	•	—	•	—
P9	$\lvert\lvert A\rvert\rvert_F \Rightarrow \sqrt{\frac{a^2+b^2+c^2}{\lvert a\rvert\lvert b\rvert\lvert c\rvert}}$ $(a, b, c \in \mathbb{R})$	—	•	•	•	—	◦	—
P10	$\left(\begin{smallmatrix} 1 & 1 \\ f(x-5)! & 6 \end{smallmatrix}\right)^{-1} \Rightarrow \left(\begin{smallmatrix} \cdot & \cdot \\ \cdot & f(b+6) \end{smallmatrix}\right)$	z^d	z^d	•	•	z^d	a^d	⊡
P11	$\left(\begin{smallmatrix} a & b \\ ab & a^2b \end{smallmatrix}\right)^{-1} \Rightarrow \frac{1}{a^2b-1}\left(\begin{smallmatrix} a & \cdot \\ -1/a & a^2b \end{smallmatrix}\right)$	•	•	•	•	•	•	•
P12	$\left(\begin{smallmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{smallmatrix}\right)^{-1} \Rightarrow \left(\begin{smallmatrix} A_{11}^{-1} & -A_{11}^{-1}A_{12}A_{22}^{-1} \\ 0 & A_{22}^{-1} \end{smallmatrix}\right)$ (blk)	—	⊗	⊗	⊗	—	—	⊟
P13	LU decomposition of a 3×3 poly. mat.	—	—	•	•	×	—	—
P14	reduced row echelon form of a 4×5 mat.	•	•	◦	•	•	•	—

Originally published in *Computer Algebra Systems: A Practical Guide* edited by Michael J. Wester, John Wiley & Sons, Chichester, United Kingdom, ISBN 0-471-98353-5, xvi+436 pages, 1999.

#	PROBLEM	Ax	De	Mc	Mp	Mm	Ma	Re
P15	rank(3 × 4 integer matrix) ⇒ 2	•	•	•	•	o^{14}	•	•
P16	rank$\left(\begin{smallmatrix} 1+\sqrt{2} & 3 \\ 4 & 2+\sqrt{2} \end{smallmatrix}\right)$ ⇒ 1	•	•	×	x	o^{16}	×	•
P17	rank$\left(\begin{smallmatrix} \sin^2\theta & \frac{\sin 2\theta}{2} \\ \frac{\sin 2\theta}{2} & \cos^2\theta \end{smallmatrix}\right)$ ⇒ 1	×	•	×	x	o^{16}	×	×
P18	null_space(3 × 4 matrix) ⇒ 4 × 3 matrix	•	○	•	•	•	•	•
P19	det(4 × 4 Vandermonde matrix)	•	•	•	•	•	•	•
P20	minpoly(4 × 4 matrix) ⇒ $(\lambda-1)^2(\lambda+1)$	x	—	—	•	—	—	—
P21	charpoly(3 × 3 matrix) ⇒ $\lambda=\{1,-2,3\}$	•	•	•	•	•	•	•
P22	eigenvalues$((2-\alpha)I_{100\times100})$ ⇒ $\{2-\alpha\}^{(100)}_{\text{alg,geo}}$	o^{10}	o^{10}	$\frac{1}{2}^{22}$	○	•	x	•
P23	eigenvalues(5 × 5 Wilkinson matrix)	•	•	•	•	•	•	•
P24	eigenvalues(8 × 8 Romer matrix)	•	⊠	•	•	•	•	•
P25	N(eigenvalues(8 × 8 Romer matrix))[23]	⊠	↓	⊡	•	•	x	x
P26	eigenvalues(3 × 3 hypercompanion mat.)	○	⊠	•	•	•	•	•
P27	eigenvalues/vectors(5 × 5 simple matrix)	•		•	•	•	•	⊡
P28	generalised eigenvectors of a 3 × 3 matrix	x	$\frac{1}{2}^{16}$	○	○	⊗	○	○
P29	generalised eigenvectors of a 6 × 6 matrix	x		○	○	⊗	○	○
P30	jordan(3 × 3) ⇒ diag$(\left(\begin{smallmatrix}1&1\\0&1\end{smallmatrix}\right),\left(\begin{smallmatrix}1&1\\0&1\end{smallmatrix}\right),-1)$	—	—	•	•	•	•	•
P31	smith_form$\left(\begin{smallmatrix} x^2 & x^3 \\ x+1 & x^2-1 \end{smallmatrix}\right)$ ⇒ $\left(\begin{smallmatrix} 1 & 0 \\ 0 & x^4-x^3+x^2 \end{smallmatrix}\right)$	—	—	—	•	—	—	•
P32	exp$\left(\begin{smallmatrix} 1 & -3 \\ 2 & 1 \end{smallmatrix}\right)$ ⇒ $\left(\begin{smallmatrix} e\cos 2 & -e\sin 2 \\ e\sin 2 & e\cos 2 \end{smallmatrix}\right)$	—	—	x	•	x	○	—
P33	exp(4 × 4 matrix)	—	—	x	•	x	x	—
P34	sin(6 × 6 Jordan matrix)	—	—	•	○	○		—
P35	sin$(\frac{\pi}{2}[3\times3$ integer matrix$])$ ⇒ I	—	—	•	x	x		—
P36	$\left(\begin{smallmatrix} 10 & 7 \\ 9 & 17 \end{smallmatrix}\right)^{1/3}$ ⇒ $\{\pm\left(\begin{smallmatrix} 2 & 1 \\ 1 & 4 \end{smallmatrix}\right), \pm\frac{1}{\sqrt{3}}\left(\begin{smallmatrix} -1 & 1 \\ 1 & 5 \end{smallmatrix}\right)\}$	—	—	o^{24}	$\frac{1}{2}^{24}$	o^{24}	ε	—
P37	$[3\times3$ non-singular integer matrix$]^{1/2}$	—	—	⊖	$\frac{1}{2}^{24}$	ε	ε	—
P38	$[3\times3$ singular integer matrix$]^{1/3}$	—	—	ε	ε	ε	ε	—
P39	SVD$(\left(\begin{smallmatrix} 1 & 3 & 2 \\ 4 & 6 & 5 \end{smallmatrix}\right)^T)$ ⇒ $(\mathbb{Z}^2)\left(\begin{smallmatrix} \sqrt{7} & 0 & 0 \\ 0 & 1 & 0 \end{smallmatrix}\right)^T(\mathbb{Z}^3)$	—	—	•	o^{16}	$\frac{1}{2}^{16}$	—	$\frac{1}{2}^{16}$
P40	jacobian$(r\cos\theta, r\sin\theta)$ ⇒ $\left(\begin{smallmatrix} \cos\theta & -r\sin\theta \\ \sin\theta & r\cos\theta \end{smallmatrix}\right)$	—	•	•	•	o^{16}	•	•
P41	hessian$(r^2\sin\theta)$ ⇒ $\left(\begin{smallmatrix} 2\sin\theta & 2r\cos\theta \\ 2r\cos\theta & -r^2\sin\theta \end{smallmatrix}\right)$	—	—	—	•	—	•	•
P42	wronskian$(\cos\theta, \sin\theta)$ ⇒ $\left(\begin{smallmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{smallmatrix}\right)$	—	—	•	•	—	—	—
P43	compute jacobian$(r\cos\theta, r\sin\theta)$ by hand	•	•	•	•	•	•	•

Originally published in *Computer Algebra Systems: A Practical Guide* edited by Michael J. Wester, John Wiley & Sons, Chichester, United Kingdom, ISBN 0-471-98353-5, xvi+436 pages, 1999.

#	PROBLEM	Ax	De	Mc	Mp	Mm	Ma	Re
P44	compute hessian$(r^2\sin\theta)$ by hand	•	•	•	•	•	•	•
P45	compute wronskian$(\cos\theta, \sin\theta)$ by hand	•	•	•	1	•	•	•
	Q. Tensor Analysis							
Q1	$\delta^{ik}_{jk} \Rightarrow \delta^i_j\delta^k_k - \delta^i_k\delta^k_j$	—	o^{20}	⊠	—	—	—	—
Q2	$\{\epsilon_{312}, \epsilon_{123}\} \Rightarrow \{-1, 0\}$ (Levi-Civita symbol)	•	•	•	○	•	—	•
Q3	$\left(\begin{smallmatrix} 1 & -2 \\ 3 & 4 \end{smallmatrix}\right) \otimes \left(\begin{smallmatrix} 0 & 5 \\ 6 & 7 \end{smallmatrix}\right)$ (outer product)	•	•	•	•	•	o^6	•
Q4	$\Gamma_{12k} \Rightarrow \frac{1}{2}(\frac{\partial g_{2k}}{\partial x^1} + \frac{\partial g_{1k}}{\partial x^2} - \frac{\partial g_{12}}{\partial x^k})$	—	—	•	o^{27}	—	—	—
Q5	$T^m_{ij,k} \Rightarrow T^m_{i,k} + \Gamma^l_{ik}T^m_{lj} - T^m_{j,k}\Gamma^l_{ik}$	—	—	•	—	—	—	—
Q6	$K^{\ k}_{i\ jkl} + K^{\ k}_{i\ klj} + K^{\ k}_{i\ ljk} \Rightarrow 0$ (Bianchi)	—	—	•	$\frac{1}{2}^{27}$	—	—	—
	R. Sums[24]							
R1	$\sum_{i=1}^n (x_i - \frac{1}{n}\sum_{j=1}^n x_j)^2 \Rightarrow \sum_{i=1}^n x_i^2 - \cdots$		•	○				
R2	least squares derivation		•		•			•
R3	$\sum_k (-1)^k \binom{2n}{k}^2 \Rightarrow (-1)^n\binom{2n}{n}$					—		
R4	$\sum_k \frac{1}{2^k}\binom{n}{k} - \frac{1}{2^{n+1}}\binom{n+1}{k} \Rightarrow \frac{1}{2^{n+1}}\binom{n}{\lfloor n/2\rfloor}$ [25]	•		•	•	—		•
R5	$\sum_k (-1)^k \binom{a+b}{a+k}\binom{b+c}{b+k}\binom{c+a}{c+k} \Rightarrow \frac{(a+b+c)!}{a!\,b!\,c!}$					—		
R6	$\sum_{k=0}^n g_{k+1} - g_k \Rightarrow g_{n+1} - g_0$		•					
R7	$\sum_{k=1}^n k^2 \Rightarrow \frac{n^3+n^2+n}{3}$	•	•	•	•	•	•	•
R8	$\sum_{k=1}^n k^2\binom{n}{k} \Rightarrow n(n+1)2^{n-2}$				⊠	•	•	
R9	$\sum_{k=0}^{10} \frac{1}{k+1}\binom{n}{k} \Rightarrow \frac{2^{n+1}-1}{n+1}$					•	•	
R10	$\sum_{k=0}^r \binom{n}{k}\binom{m}{r-k} \Rightarrow \binom{n+m}{r}$ (Vandermonde)					½	•	×
R11	$\sum_{k=0}^n \binom{n}{k} f_k \Rightarrow f_{2n}$	ε						—
R12	$\sum_{k=1}^n f_k^2 \Rightarrow f_n f_{n+1}$	ε		○				—
R13	$\sum_{k=1}^n \sin kx \Rightarrow \frac{1}{2}\cot\frac{x}{2} - \frac{\cos(\lfloor n+\frac{1}{2}\rfloor x)}{2\sin(x/2)}$		•	•	•	•	•	•
R14	$\sum_{k=1}^n \sin(\lfloor 2k-1\rfloor x) \Rightarrow \frac{\sin^2 nx}{\sin x}$		•		•	•	○	•
R15	$\sum_{k=0}^{\lfloor n/2\rfloor} \binom{n-k}{k} \Rightarrow f_{n+1}$	ε			⊟	½		
R16	$\sum_{k=1}^{100} (\frac{1}{k^3} + \frac{1}{k^2}) \Rightarrow \frac{\pi^2}{2} + \zeta(3)$		•	•	•	•	•	•
R17	$N(\sum_{k=1}^{100} (\frac{1}{k^3} + \frac{1}{k^2}))$ ≈ 2.84099		•	•	•	•	•	•
R18	$\sum_{k=1}^m \frac{1}{2k+1} \Rightarrow \frac{\pi^2}{12} - \frac{1}{2}\log^2 2$				•	•	•	
R19	$\sum_{k=0}^{100} \frac{1}{(3k-1)(3k+1)} \Rightarrow \frac{\pi}{12}\sqrt{3} - \frac{1}{2}\log 3$		•	•	•	•	⊠	
R20	$\sum_{k=0}^{100} \binom{n}{4k} \Rightarrow \frac{1}{4}(2^{n-1} + 2^{n/2}\cos\frac{n\pi}{4})$				○	○		

Originally published in *Computer Algebra Systems: A Practical Guide* edited by Michael J. Wester, John Wiley & Sons, Chichester, United Kingdom, ISBN 0-471-98353-5, xvi+436 pages, 1999.

#	PROBLEM	Ax	De	Mc	Mp	Mm	Mu	Re
R21	$\sum_{k=1}^{\infty} \frac{1}{\sqrt{k(k+1)}(\sqrt{k}+\sqrt{k+1})} \Rightarrow 1$							
R22	$\sum_{n=0}^{\infty}\sum_{k=0}^{\lfloor n/2\rfloor}\binom{n}{2k}\binom{n-k}{n-2k}x^{2k}y^{n-2k}$ $(\lvert xy\rvert<1)$	ε			#		$\mathcal{E}$	
R23	$\sum_{k=0}^{\infty}\sum_{n=k}^{\infty}\frac{n!}{2^n(n-k)!}\binom{n}{k}x^k(xy)^{n-k}$ (*)				•	•		
R24	$\sum_{n=2}^{\infty}\prod_{k=1}^{n}\frac{k}{2k+1} \Rightarrow \frac{1}{3}$		#	#	•	•	#	#
	3. Products							
S1	$\prod_{k=1}^{8}\Gamma(\frac{k}{3}) \Rightarrow \frac{640\pi^4}{2187\sqrt{3}}$		•	#	•	•	•	
S2	$\prod_{k=1}^{n} k \Rightarrow n!$ or $\Gamma(n+1)$		•	•	•	•	•	•
S3	$\prod_{k=1}^{n} x^k \Rightarrow x^{n(n+1)/2}$		•	•		•		•
S4	$\prod_{k=1}^{n-1}\frac{1}{(-1)^k k} \Rightarrow \pi$		•	•	•	•		•
S5	$\prod_{k=1}^{n}\frac{2k-1}{2k} \Rightarrow \frac{1}{2^{2n}}\binom{2n}{n}$		•	•	•	•	•	•
S6	$\prod_{k=1}^{n-1} x^2 - 2x\cos\frac{k\pi}{n} + 1 \Rightarrow \frac{x^{2n}-1}{x^2-1}$							
S7	$\prod_{k=2}^{\infty}\frac{k^2}{k^2+1} \Rightarrow \frac{1}{2}$		30	•	•			•
S8	$\prod_{k=1}^{\infty} 1-\frac{1}{(2k)^2} \Rightarrow \frac{2}{\pi}$		•	•	•	•		②
S9	$\prod_{k=1}^{\infty} 1+\frac{(-1)^{k+1}}{2k-1} \Rightarrow \sqrt{2}$				ε	•		$\mathcal{E}$
S10	$\prod_{k=0}^{\infty}\frac{k!(2k+1)!+1}{k!(k+1)!+(-1)^k} \Rightarrow -1$			•	ε			$\mathcal{E}$
	7. Limits[21]							
T1	$\lim_{n\to+\infty}(1+\frac{1}{n})^n \Rightarrow e$; $\lim_{x\to 0}\frac{1-\cos x}{x^2} \Rightarrow \frac{1}{2}$	•	•	•	•	••	•	•
T2	$\lim_{x\to+\infty}(3^x+5^x)^{1/x} \Rightarrow 5$		$\mathcal{E}$	•	•	×		×
T3	$\lim_{x\to+\infty}\frac{\log x}{\log x+\sin x} \Rightarrow 1$			•	×	•		
T4	$\lim_{x\to+\infty}\frac{1}{x}\left\lvert\exp\left(\frac{ax^{x+1}}{x^x-(x-1)^x}-ex\right)\right\rvert \Rightarrow e^{\frac{1}{2}}$		τ		•	×	•	
T5	$\lim_{x\to+\infty}\frac{x\log x\log^2(xe^x-e^x)}{\log(\log(x^{x^2}+3\exp(e^{\sin x}\ln x)))} \Rightarrow \frac{1}{3}$		τ		•	x	•	②
T6	$\lim_{n\to+\infty}\frac{1}{n}\sqrt[n]{n!} \Rightarrow \frac{1}{e}$		$\mathcal{E}$	•	•	•	$\mathcal{E}$	×
T7	$\lim_{n\to+\infty}\frac{1}{n}\sqrt[n]{\Gamma(n+1)} \Rightarrow \frac{1}{e}$		$\mathcal{E}$	•		×	×	②
T8	$\lim_{x\to+\infty}\frac{\Gamma(x+a)}{\Gamma(x)}e^{-a\log x} \Rightarrow 1$		•	•		•	$\mathcal{E}$	②
T9	$\lim_{k\to\infty} F(1,k;1;\frac{1}{k}) \Rightarrow e$ [22]	—	•	•	•	••	—	
T10	$\lim_{s\to 0}\zeta(s)-\frac{1}{s-1} \Rightarrow \gamma$	—		×	•	••		×
T11	$\lim_{n\to+\infty}\frac{n^x}{x\prod_{k=1}^{n}(1+\frac{x}{k})} \Rightarrow \Gamma(x)$		τ	×	≠	•		②
T12	$\lim_{x\to 0}\frac{2\sqrt{\pi}\int_0^x e^{-u^2}du}{1-e^{-x^2}} \Rightarrow 1$		•	•	•	•×	•	•
T13	$\lim_{x\to 0-}\frac{x}{\lvert x\rvert} \Rightarrow -1$; $\lim_{x\to 0+}\frac{x}{\lvert x\rvert} \Rightarrow 1$	•	•	•	•	••	•	•

Originally published in *Computer Algebra Systems: A Practical Guide* edited by Michael J. Wester, John Wiley & Sons, Chichester, United Kingdom, ISBN 0-471-98353-5, xvi+436 pages, 1999.

#	PROBLEM	Ax	De	Mc	Mp	Mm	Mu	Re
T14	$\lim_{x\to 0+}\tan^{-1}(-\log x) \Rightarrow \frac{\pi}{2}$	•	•	•	•	••	$\mathcal{E}$	
	U. Calculus							
U1	$\frac{d}{dx}\lvert x\rvert \Rightarrow \frac{x}{\lvert x\rvert}$ or $\operatorname{sign}(x)$	•	•	•	•	•	•	•
U2	$\frac{d}{dx}\lvert x\rvert$ (piecewise defined)[23]	•	•••	•••	•••	•$\mathcal{E}$	•$\mathcal{E}$	$\mathcal{E}$
U3	$f(x)=\begin{cases} x^2-1 & (x=1) \\ x^2 & \text{otherwise} \end{cases}$; $f'(1) \Rightarrow 2$	•	×	•	x^{2x}	×	•	•
U4	$\int_{x=0}^{n} x^n \Rightarrow n!$	ε		•			$\mathcal{E}$	×
U5	$\frac{d^2}{dx^2}y(x(t)) \Rightarrow \frac{d^2y}{dt^2}\left(\frac{dx}{dt}\right)^{-2}+\frac{dy}{dx}\frac{d^2x}{dt^2}$	•	•	•	•	•	•	•
U6	$\frac{d}{dx}\int_{g(x)}^{h(x)} f(y)\,dy \Rightarrow f(h(x))\frac{dh}{dx}-f(g(x))\frac{dg}{dx}$	ε	•	•	•	•	•	
U7	$d(V(P,T)) \Rightarrow \frac{\partial V}{\partial P}dP+\frac{\partial V}{\partial T}dT$	—		•	—	•	—	—
U8	solve$(y=\sin xy+z,\frac{\partial z}{\partial x}) \Rightarrow \frac{-y\cos xy}{1+x\cos xy}$	—	•	◦	•	◦	—	—
U9	substitute$(f=g(x^2+y^2),\frac{\partial f}{\partial x}+\frac{\partial f}{\partial y})$		•	◦	•	•	•	
U10	residue$(\frac{z^3}{(z^2-1)(z+1)},z=-1) \Rightarrow -\frac{1}{4}$	—	—	•	•	•	—	•
U11	$(3\,dx+dz)\wedge(3\,dx+dy+dz)\wedge\cdots$	•	—	•	•	—	—	•
U12	$d(3x^2\,dy\wedge dz-5xy^2\,dx\wedge dz-8x\,dx\wedge dy)$	$\mathcal{E}$	—	•	$\mathcal{E}$	—	—	$\mathcal{E}$
U13	minimize$(x^4-x+1) \Rightarrow 1-\frac{3}{4}\sqrt[3]{4}$	—	—	ε	•	x^{2n}	—	—
U14	minimize/maximize$(\frac{1}{x^2+y^2+1},\Gamma)$ $\Rightarrow (0,1)$	—	—	—	×	$\mathcal{E}$	—	—
U15	minimize$(a+bx+cy+dxy,[-1,1]^2)$	—	—	—	⊗	—	—	—
U16	minimize/maximize$(x^2y^2,[-1,1]^2)$	—	—	—	$\mathcal{E}$	—	—	—
U17	maximum(linear function + constraints)	—	—	•	•	•	•	•
	V. Indefinite Integrals[24]							
V1	$\int\lvert x\rvert\,dx \Rightarrow \frac{x\lvert x\rvert}{2}$ $(x\in\mathbb{R})$		⊗	•	⊗	•	⊗	•
V2	$\int\lvert x\rvert\,dx$ (piecewise defined)[25]	⊗	•	••	$\mathcal{E}$•	•	$\mathcal{E}$	
V3	$\int\frac{1}{x^2+2}\,dx \rightarrow$ diff $\rightarrow$ simplify	•	•	•	•	•	•	•
V4	$\int\frac{x^n}{\sqrt{1-x^2}}\,dx \Rightarrow \frac{n\tan^{-1}\frac{x}{\sqrt{1-x^2}}}{\log 2}-\frac{x^{n-1}\sqrt{1-x^2}}{\log 2}$	$\mathcal{E}$	•	◦	×			
V5	$\int\frac{(3x-2)^3}{(2x-1)^{7/3}}\,dx \Rightarrow \frac{-6x^2+14x-41/5}{(2x-1)^{4/3}}$	•	•	•	•	•	•	•
V6	$\int\frac{1}{5e^{4x}-5e^{-x}}\,dx \Rightarrow \frac{1}{2e\sqrt{10}}\log\frac{-3+e^{5x}\sqrt{10}}{-3-e^{5x}\sqrt{10}}$	•	•	•	•	•	◦	•
V7	$\int\frac{\sinh^3 x}{\cosh^2 x}\,dx \Rightarrow -\frac{3x}{2}-\frac{1}{4}\sinh 2x+\tanh x$	•	◦	•	•	•	•	◦
V8	$\int\frac{1}{a+b\cos x}\,dx$ $(a<b)$ [real solution]	•[26]		•	•	•	◦	
V9	$\int\frac{1}{a+b\cos x}\,dx \rightarrow$ diff $\rightarrow$ simplify	•	•	•	•	•	•	•

Originally published in *Computer Algebra Systems: A Practical Guide* edited by Michael J. Wester, John Wiley & Sons, Chichester, United Kingdom, ISBN 0-471-98353-5, xvi+436 pages, 1999.

#	PROBLEM	Ax	De	Mc	Mp	Mm	Mu	Re
V10	$\int \frac{1}{3+2\cos x+4\sin x}\,dx \Rightarrow \frac{1}{5}\log(2+4\tan\frac{x}{2})$	•	⊡	•	•	•	ℰ	•
V11	$\int \frac{1}{1+2\cos x+4\sin x}\,dx \Rightarrow \frac{1}{5}\log\frac{\tan(x/2)+1}{\tan(x/2)+7}$	•	◦	•	◦	•	◦	•
V12	$\int \frac{1}{2+3\cos x+4\sin x}\,dx \Rightarrow -\frac{1}{2+\tan\frac{x}{2}}$	•	◦	◦	•	•	ℰ	◦
V13	$\int \frac{1}{5+3\cos x+4\sin x}\,dx \Rightarrow \frac{1}{\sqrt{11}}\tan^{-1}\frac{3\tan(x/2)+4}{\sqrt{11}}$	•	⊎	⊠	•	•	•	•
V14	$\int \log\lvert x^2-a^2\rvert\,dx \Rightarrow x\log\lvert x^2-a^2\rvert-\cdots$	⊡	⊗	⊠	⊡	ℰ[3]		⊡
V15	$\int x\cos^{-1}\frac{x}{a}\,dx \Rightarrow \frac{3x}{8}+\frac{1}{4}(x^2+a^2)\cos^{-1}\frac{x}{a}$	⊗	•	•	•	•	⊡	•
V16	$\int \cos bx\,\mathrm{Ci}\,2x\,dx \Rightarrow \frac{\sin bx\,\mathrm{Ci}\,2x}{b}-\frac{\mathrm{Si}\,2x+\mathrm{Ci}\,2x}{2b}$	ℰ	∉			•		
V17	$\int \frac{\Gamma(x)\,\psi(x)-\Gamma(x)\frac{d\Gamma(x)}{dx}}{\Gamma^2(x)+\psi^2(x)}\,dx \Rightarrow \frac{1}{2}\frac{\Gamma(x)-\psi(x)}{\Gamma(x)+\psi(x)}$			○		ℰ		
	W. Definite integrals[28]							
W1	$\int_{n-1}^{n+1}\frac{1}{x-a}\,dx \Rightarrow 0$ (principal value)	ε	×	⊎	•	○	•	×
W2	$\int_{n-1}^{n+1}\frac{1}{x-a}\,dx \Rightarrow$ divergent		◦	•	•	•	•	◦
W3	$\int_0^2\sqrt{x+\frac{1}{x}-2}\,dx \Rightarrow \frac{2}{3}$		•	•	•	•	•	⊎
W4	$\int_1^2\sqrt{x+\frac{1}{x}-1}\,dx \Rightarrow \frac{4-\sqrt{3}}{3}$	∗	•	•	•	•	•	•
W5	$\int_0^2\sqrt{x+\frac{1}{x}-1}\,dx \Rightarrow \frac{8-\sqrt{6}}{3}$		•	•	•	•	•	⊎
W6	$\int_{-2\pi}^{-\pi}\frac{\sqrt{2-\cos 2x}}{\sin x}\,dx \Rightarrow \sqrt{3}$		•	×	•	•	•	
W7	$\int_{-\infty}^{\infty}\frac{\cos x}{x^2+a^2}\,dx \Rightarrow \frac{\pi}{a}e^{-a}\quad(a>0)$			•	•	•	•	
W8	$\int_0^\infty\frac{x^{a-1}}{1+x}\,dx \Rightarrow \frac{\pi}{\sin\pi a}\quad(0<a<1)$			•		•	◦	ℰ
W9	$\int_{-\infty}^{\infty}\frac{2x^2}{1+2x+3x^2+4x^3+5x^4}\,dx \Rightarrow -2\pi(\sin\frac{\pi}{3}+\cdots)^{29}$			⊠		×		×
W10	$\int_{-\infty}^{\infty}\frac{1}{(x-a)^2+b^2}\,dx \Rightarrow \frac{2\pi}{b}(1+\cos\frac{2\pi}{3})\cos\frac{2\pi}{3}$	⊤		⊤	×			
W11	$\int_{-1}^{1}\frac{\sqrt{1-x^2}}{1+x^2}\,dx \Rightarrow \pi(\sqrt{2}-1)$	∗	•	•	•	•		×
W12	$\int_{-\infty}^{\infty}xe^{-px^2+2qx}\,dx \Rightarrow \frac{q}{p}\sqrt{\frac{\pi}{p}}e^{q^2/p}\quad(p>0)$		•	•	•	⊡	∉	×
W13	$\int_0^1\frac{1}{\log x}+(\frac{1}{x}-\log(\log\frac{1}{x}))\,dx \Rightarrow 2\gamma$							ℰ
W14	$\int_{-\infty}^{\infty}\frac{\sin x}{x}e^{2ix}\,dx \Rightarrow 0$	ε		•	⊎[29]	•		
W15	$\int_0^1\log\Gamma(x)\cos\pi x\,dx \Rightarrow \frac{1}{12}$							⊤
W16	$\int_{-1}^{1}(1+x)^2P_1(x)P_3(x)\,dx \Rightarrow \frac{16}{35}$	•	•	•	•	•	ℰ	×
W17	$\int_0^\infty e^{-ax}J_0(bx)\,dx \Rightarrow \frac{1}{\sqrt{a^2+b^2}}\quad(a>0)$			ℰ	•	⊡	•	•	•
W18	$\int_0^\infty\left(\frac{J_1(x)}{x}\right)^2dx \Rightarrow \frac{8}{3\pi}$			ℰ		•	•	ℰ	
W19	$\int_0^\infty\mathrm{Ci}\,x\,J_0(2\sqrt{7x})\,dx \Rightarrow \frac{\cos 7-1}{7}$	ℰ	ℰ					—	
W20	$\int_0^1 x^2\mathrm{Li}_3(\frac{1}{x+1})\,dx \Rightarrow \frac{17+\pi^2}{36}+\cdots+\frac{\zeta(3)}{4}$	—							—

Originally published in *Computer Algebra Systems: A Practical Guide* edited by Michael J. Wester, John Wiley & Sons, Chichester, United Kingdom, ISBN 0-471-98353-5, xvi+436 pages, 1999.

#	PROBLEM	Ax	De	Mc	Mp	Mm	Mu	Re
W21	$N(\int_{-\infty}^{t}x^2\mathrm{Li}_3(\frac{1}{x+1})\,dx)=N(t)\approx 0.210883$	—	◦	•	•	•	◦	↯
W22	$s(t)=\begin{cases}1 & (1\le t\le 2)\\ 0 & \text{otherwise}\end{cases}$; $\int_0^\infty s(t)\cos t\,dt$?	×	•	⊎	ℰ	•	•	ℰ	ℰ
W23	$\int_0^1\int_{-\infty}^{\infty}\frac{x^k}{x^2+y^2}\,dy\,dx=\int_{-\infty}^{\infty}\int_0^1\frac{x^k}{x^2+y^2}\,dx\,dy$	⊂⊃	•	•	⊗	↯	◦	↯
W24	$\int_0^1\int_0^1\sqrt{x^2+y^2}\,dx\,dy \Rightarrow \frac{\log(\sqrt{2}+1)+\sqrt{2}}{3}$	ℰ[0]	•	•		•		•
W25	$\int_0^{\pi/2}\int_0^{\pi/2}\frac{\sin x\sin y}{\sqrt{1-\sin^2x\sin^2y}}\,dx\,dy \Rightarrow \frac{2\pi}{3}$	ε	ℰ		•		∉	ℰ
W26	$\int_{-1}^{1}\int_0^2\lvert y-x^2\rvert\,dy\,dx \Rightarrow \frac{56}{15}$	ε	•	•	•	•		≠
W27	$\int_0^1\int_0^{2(1-x/2)}\int_0^{3(1-x/2-y/4)}1\,dz\,dy\,dx \Rightarrow \frac{4\pi}{3}$	•	•	•	•	•	•	•
	X. Series							
X1	$\mathrm{taylor}(\frac{1}{\sqrt{1-x+x^2}},x=3)\Rightarrow 1+\frac{1}{2}x_3^2+\cdots$	•	•	•	•	•	•	•
X2	$\frac{1}{e^x+e^{-x}}\Rightarrow 1-\frac{1}{2}x^2+\cdots$	•	◦	•	◦	•	•	◦
X3	$\frac{\mathrm{taylor}(\sin x,x=0)}{\mathrm{taylor}(\cos x,x=0)}=\mathrm{taylor}(\tan x,x=0)$	•	◦	•	•	•	•	◦
X4	$\log(\mathrm{taylor}(\frac{\sin x}{x},x=0))\Rightarrow -\frac{x^2}{3}-\frac{x^4}{180}-\cdots$	•	◦	◦	◦	◦	◦	◦
X5	$\mathrm{taylor}(f^{(1)}(xz)-g(\ln z)+\int_0^x h(xy)\,dy,\ldots)$	ε	•	○	•	•	∉	○
X6	$\mathrm{taylor}(e^{(A+B)x}-e^{Ax}e^{Bx},\ldots)\quad(AB\neq BA)$	—	×	×	×	—	—	×
X7	$\mathrm{taylor}(\frac{1}{x(e^x-1)},x=0)\Rightarrow\frac{1}{x^2}-\frac{1}{2x}+\frac{1}{12}-\cdots$	•	↯	•	•	•	•	•
X8	$\mathrm{taylor}(\sqrt{\sec x},x=\frac{3\pi}{4})\Rightarrow\frac{1}{\sqrt{2-2\pi/3}}+\cdots$	•		•	•	•	•	•
X9	$\mathrm{taylor}(x^x,x=0)\Rightarrow\sum_{k=0}^{\infty}\frac{(x\log x)^k}{k!}$	•		•	•	•	•	
X10	$\mathrm{taylor}(\log(\sinh x)+\log(\cosh(x+w)))$	•		◦	◦	•	•[29]	•
X11	$\mathrm{taylor}(\log(\sinh x\cosh(x+w)),x=0)$	•		•	•[29]	•	•[29]	
X12	$\mathrm{taylor}((\log x)^n x^{-w},x=1)\Rightarrow\frac{(-1)^n}{n!}-\cdots$	ℰ		•	ℰ		[29]	•
X13	$\mathrm{taylor}(\sqrt{2x^3+1},x=\infty)\Rightarrow\sqrt{2}x+O(\frac{1}{x})$	ε		•	•	•	•[29]	•
X14	$\mathrm{taylor}(\frac{1}{\Gamma(x)},(\frac{2n}{e}),n=\infty)\Rightarrow\frac{1}{\sqrt{4\pi}}+\cdots$	ε		ℰ	ℰ	ε	ℰ	
X15	$\mathrm{taylor}(e^x\int_x^{\infty}t^{\frac{1}{t}}\,dx),x=\infty)\Rightarrow\frac{1}{x}-\cdots$	ε	ℰ	ℰ	•			ℰ
X16	$\mathrm{taylor}(\cos(x+y),x=y=0)\Rightarrow 1-\cdots$	ε	◦	•	◦	•	↯	•
X17	$\mathrm{power\_series}(\log\frac{\sin x}{x},x=0)$	—	—	•	ℰ	◦		
X18	$\mathrm{power\_series}(e^{-x}\sin x,x=0)$	—	—	⊎	•	•	•	⊗
X19	$\mathrm{solve}(x=\sin y+\cos y,y=0,\mathrm{series})$	ℰ	•	•	•	•	◦	•
X20	$\mathrm{padé}(e^{-x},x=0)\Rightarrow\frac{2-x}{2+x}$	•	•	•	•	•	•	•[5]
X21	$\mathrm{fourier}(x,p)\Rightarrow-\frac{2p}{\pi}\sum_{n=1}^{\infty}\frac{(-1)^n}{n}\sin\frac{n\pi x}{p}$	—	•	•	—	•	—	—

Originally published in *Computer Algebra Systems: A Practical Guide* edited by Michael J. Wester, John Wiley & Sons, Chichester, United Kingdom, ISBN 0-471-98353-5, xvi+436 pages, 1999.

#	PROBLEM	As	Dc	Mc	Mp	Mm	Mu	Re
X22	fourier($\lvert x\rvert$, $x \to s$) $\Rightarrow \frac{2}{s^2} - \frac{2\pi}{|s|} \sum_{n=1}^{\infty} \frac{1-(-1)^n}{n^2\pi^2} \cos\frac{n\pi s}{a}$	—	•	•	—	▫	—	—
	Y. Transforms[60]							
Y1	laplace($\cos((\omega-1)t)$, $t \to s$) $\Rightarrow \frac{s}{s^2+(\omega-1)^2}$	•	•	•	•	•	•	•
Y2	inverse Laplace transform of above	•	—	•	•	•	•	▫
Y3	laplace($\sinh\omega t \cosh\omega t$, $t \to s$) $\Rightarrow \frac{\omega}{s^2-4\omega^2}$	•	•	•		•	•	•
Y4	laplace($\operatorname{erf}\left(\frac{a}{\sqrt{t}}\right)$, $t \to s$) $\Rightarrow \frac{1-e^{-2\sqrt{s}a}}{s}$ $(a>0)$	▫			⊗			▫
Y5	laplace($\frac{d^2y}{dt^2} + y = 4[H(t-1) - H(t-2)]$)	#	#	•	•	•	•		
Y6	solve " using inverse Laplace transforms			—	•	•	•	#	—
Y7	laplace($1 + 2\sum_{n=1}^{\infty}(-1)^n H(t-na)$, $t \to s$)	▫		#	#				⊘
Y8	fourier(1, $x \to s$) $\Rightarrow \sqrt{2\pi}\delta(s)$ [61]	▫	▫		•	•	•	—
Y9	fourier(e^{-bx^2}, $x \to s$) $\Rightarrow \frac{e^{-s^2/4b}}{2\sqrt{b}}$ [61]	▫	▫	•	•	•	•	•
Y10	fourier($\lvert x\rvert e^{-\lvert x\rvert a}$, $x \to s$) $\Rightarrow \sqrt{\frac{2}{\pi}}\frac{a^2-s^2}{(a^2+s^2)^2}$ [61]	▫	▫	•		•			
Y11	Mellin_transform($\frac{1}{1-x}$, $x \to s$) $\Rightarrow \pi\cot s\pi$	▫	▫	▫	•	▫		▫
Y12	Mellin_transform($\frac{\ln(x)}{x^2}$, $x \to s$) $\Rightarrow \frac{\Gamma(-s-1)\psi(1)}{\Gamma(s-s/\lambda)}$	▫	▫	▫	•	▫		▫
Y13	Z_transform($H(t-2T)$, $t \to z$) $\Rightarrow \frac{z}{z^2(z-1)}$	—	—	—	•	—	—	—
Y14	Z_transform($H(t-mT)$, $t \to z$) $\Rightarrow \frac{z}{z^m(z-1)}$	—	—	—		—	—	—
	Z. Ordinary Difference and Differential Equations[62]							
Z1	solve($[r_{n+2} - 2r_{n+1} + r_n = 2, \ldots], r_n$)	—	•	•	•	•	•	—
Z2	solve($[r_n = 3r_{n-1} - 2r_{n-2}, \ldots], r_n$)	—	•	•	•	•	•	—
Z3	solve($[r_n = r_{n-1} + r_{n-2}, r_1 = 1, r_3 = 2]$)	—	▫	▫[63]	▫	▫	▫	—
Z4	solve($[r_n = \frac{(a+c)r^{n-1} - (a+c)^{n+1}}{1-r^2} r_{n-1} - \ldots], r_n$)	—	—	▫	▫	▫		—
Z5	solve($\frac{d^2f}{dt^2} + 4f = \sin 2t$, $f(0) = f'(0) = 0$)	•	▫/•	•	•	•	•	½
Z6	above solution using Laplace transforms	½	#	•	•	•	•	
Z7	solve($\frac{dy}{dx} = \frac{x^2}{y(1+x^3)}$, $y(x)$)	•	•/•	•	•	•	•	
Z8	solve($\frac{dy}{dx} = \frac{x}{a} + \frac{a}{y}$) $\Rightarrow y^2(x) = 2x^2\log\lvert Ax\rvert$	⊡	⊡	•	•	⊡	•	⊗	▫
Z9	solve($x^2\frac{dy}{dx} + 2xy = \frac{\sin x}{x}$) $\Rightarrow y = \frac{A-\cos x}{x^2}$	•	•/•	•	•	•	•	•	•
Z10	solve($\frac{dy}{dx} = -\frac{(-1)x}{1+x^2}\frac{\sin x}{\cos y}$, $y(x)$)	•	½/•	•	•	▽			
Z11	solve($\frac{d^2y}{dx^2} + y(\frac{dy}{dx})^2 = 0$, $y(x)$) $\Rightarrow \frac{x^2(x)}{2} \ldots$	▫	½[64]	×	•	•	•	•	
Z12	solve($[$", $y(0) = 0$, $y'(0) = 2], y(x)$)	▫	×	▫	▷[65]	▷[65]		—

Originally published in *Computer Algebra Systems: A Practical Guide* edited by Michael J. Wester, John Wiley & Sons, Chichester, United Kingdom, ISBN 0-471-98353-5, xvi+436 pages, 1999.

#	PROBLEM	Ax	Dc	Mc	Mp	Mm	Mu	Re
Z13	solve($\frac{\partial}{\partial x}y(x,a) = ay(x,a)$, $y(x,\alpha)$)[66]	▫		•	•	▫/•	•	•
Z14	solve($[\frac{d^2y}{dx^2} + k^2y = 0, y(0) = \ldots = 3], y, k$)[66]		#					
Z15	solve($\frac{d^2y}{dx^2}\frac{1}{x} + \frac{1}{x}\frac{dy}{dx} + \frac{1}{x^2}(1 - \frac{n^2}{x^2})y = 0$, $y(x)$)		▫	•	•	•		
Z16	solve($\frac{dy}{dt} = ay(t-1) = 0$, $y(t)$)		—		▫	▫	▫	
Z17	discontinuous ODE[67]	×/▫	▫	▫	▫	×/▫	×/×	▫	▫
Z18	solve($\frac{dy}{dt} + 2i + 5\int_0^t y(\tau)\,d\tau = 10e^{-4t}$, $y(t)$)	▫	—		▫			
Z19	above solution using Laplace transforms	▫	#	•	•	•	#	
Z20	solve($[\frac{dx}{dt} = x - y, \frac{dy}{dt} = x + y]$, $[x(t), y(t)]$)	—	—	•	•	•/▫	•	—
Z21	verify the solution of the above	▫	•	▫	▫	▫/•	▫	•
Z22	solve($[\frac{dx}{dt} = x\left(1 + \frac{x}{\sqrt{x^2+y^2}}\right), \frac{dy}{dt} = x - y]$)	—	—	•	•	•	▫	—
Z23	as above, but one equation at a time	•	▫/•	•	•	•	•	½
Z24	3×3 linear system ($\lambda = 2, 1, 3$)	—	—	•	•	•	•	—
Z25	3×3 linear system ($\lambda = 0, -1 \pm i\sqrt{2}$)	—	—	•	•	▫	•	—
Z26	3×3 linear system ($\lambda = 2, 2, 2$)	—	—	•	•	•	▫	—
Z27	solve($[\frac{d^2y}{dx^2} = 2\omega\frac{dy}{dx}, \frac{d^2z}{dx^2} = -2\omega\frac{dy}{dx} + 3\omega^2 y]$)	—	—	•	•	•	▫	—
	a. Partial Differential Equations[67]							
a1	solve($\frac{\partial^2 f}{\partial x \partial y} = 0$, $f(x,y)$) $\Rightarrow g(x) + h(y)$	—	—		•	⊡	—	•
a2	solve($\frac{\partial w}{\partial t} = \frac{\partial^2 w}{\partial x^2}$, $w(x,t)$) [Heat equation]	—	—	▫[68]	½		—	
a3	substitute($u = \frac{1}{\sqrt{t}}f(\frac{x}{\sqrt{t}})$, $\frac{\partial w}{\partial t} = \frac{\partial^2 w}{\partial x^2}$)	▫	•	▫	▫	▫		▫	▫
a4	substitute($z = x\sqrt{t}$, above)	•	⊗	•	•	•	•	⊗	▫
a5	solve($2\frac{d^2f}{dz^2} + z\frac{df}{dz} + f = 0$, $f(z)$)[69]	▫	▫	•	•	•	•		
a6	solve($\frac{1}{r}\frac{\partial}{\partial r}(r\frac{\partial w}{\partial r}) + \frac{1}{r^2}\frac{\partial^2 w}{\partial \theta^2} = 0$, $w(r,\theta)$)	—	—		▫		—		
	b. Operators							
b1	$f = e^x$, $g = x^2$: $(f+2g)(y) \Rightarrow e^y + 2y^2$	▫	—	•	•	#	•	
b2	$(f*g)(y) \Rightarrow e^{y^2}$	▫	—	•	•	⊗	•	
b3	$L = (D-1)(D+2)$ (operator)	•	—	•	•	•	•	—
b4	$L(f) \Rightarrow D^2f + Df - 2f$ (pure function)	•	—	•	•	½	•	—
b5	$L_y(z(y)) \Rightarrow \frac{d^2z}{dy^2} + \frac{dz}{dy} - 2z(y)$	▫	—	•	•	½	•	—
b6	$L_x(A\sin x^2) \Rightarrow 2A[(1+x)\cos x^2 - \cdots]$	▫	—	•	•	½	•	—
b7	$T = \sum_{n=0}^{2}\frac{(D^n f)(a)}{n!}(x-a)^n$ (operator)	▫	—	•	•	•	•	•

Originally published in *Computer Algebra Systems: A Practical Guide* edited by Michael J. Wester, John Wiley & Sons, Chichester, United Kingdom, ISBN 0-471-98353-5, xvi+436 pages, 1999.

#	PROBLEM	Ax	De	Mc	Mp	Mm	Mu	Ro
b8	$\Gamma(f) \Rightarrow f(a) + (Df)(a)(x-a) + \cdots$	•	—	•	•	○	•	•
b9	$T_{g,b}(g(y)) \Rightarrow g(b) + \frac{dg}{dy}\|_{y=b}(y-b) + \cdots$	•	—	•	•	•	•	•
b10	$T_{s,c}(\sin x) \Rightarrow \sin c + \cos c\,(x-c) - \cdots$	•	—	•	•	•	•	•
b11	define $*$ so that $x * y = \sqrt{x^2+y^2}$	—	—	•	•	•	•	•
b12	$3 * 4 * 12 \Rightarrow 13$ (make $*$ associative)	—	—	•	•	○	○	•
b13	define $\|\ \|$ so that $\|x\| = \sin(x)$	—	—	•	—	—	—	—
	c. Programming and Miscellaneous							
c1	substitute$(a+b=x, (a+b+c)^2+\cdots)$	○	○	•	•	•	○	○
c2	substitute$(\sqrt{x^2+y^2}=r, \frac{x}{\sqrt{x^2+y^2}}) \Rightarrow \frac{x}{r}$	•	○	•	•	•	•	•
c3	change variables in a messy expression	½	○	○	○	○	○	○
c4	$f'g' + f'g \Rightarrow (fg)'$ (rewrite rule)		—	½[50]		•		$\mathcal{L}$
c5	$2(f'g'+f'g') + fg \Rightarrow 2[fg]' + fg$ (*)	$\mathcal{L}$	—	½[50]	½[50]	•		$\mathcal{L}$
c6	multiply two infinite lists together	•	○	•	○	•	○	○
c7	compute Legendre polys directly	•	•	•	•	•	•	•
c8	compute Legendre polys recursively	•[51]	•	•	•[51]	•	•[51]	•
c9	evaluate the 4[19] Legendre poly at 1 [52]	•/•	•	•	•	•	•/•	•/•
c10	define iterative Fibonacci number func.	•	•	•	•	•	•	•
c11	translate to Fortran syntax	ε	○	ε	○	○	ε	
c12	create $[f_0, \ldots, f_{10}] \Rightarrow [0,1,1,2,\ldots,55]$	•	•	•	•	•	•	•
c13	define a simple derivative operator	•		○	○	•	○	•
c14	define p so $p(2+i\sqrt{2}) = 0$, $p(\left(\begin{smallmatrix}1 & -1\\ 1 & 2\end{smallmatrix}\right)) = 0$	•	•	⊗	⊗[53]	⊗	⊗	⊗
c15	define a function as a calculation result	•	•	•	•	•	•	•
c16	display an expression's top-level structure	—	—	•	½	½	½	○
c17	translate $y = \sqrt{\frac{e^x+ix-e^{-x}}{\sqrt{3}x-\sqrt{2}}}$ to TpX/10TpX[54]	•	—	•	•	○	½	•/•
	d. Mathematics versus Computer Science							
d1	(3[10] k local?) $k \leftarrow 1$; $\sum_{k=1}^{4} k \Rightarrow 10$	ε	•	•	ε	•	ε	ε
d2	(3[10] k local?) $k \leftarrow 1$; $\prod_{k=1}^{3} k \Rightarrow 6$	ε	•	•	ε	•	ε	ε
d3	(3[10] k local?) $k \leftarrow 1$; $\lim_{k\to\infty} k \Rightarrow 0$	ε	•	ε	ε	ε	ε	×
d4	(3[10] k local?) $k \leftarrow 1$; $\int_{k=0}^{1} k\,dk \Rightarrow \frac{1}{2}$	ε	•	ε	ε	ε	ε	ε

Originally published in *Computer Algebra Systems: A Practical Guide* edited by Michael J. Wester, John Wiley & Sons, Chichester, United Kingdom, ISBN 0-471-98353-5, xvi+436 pages, 1999.

#	PROBLEM	Ax	De	Mc	Mp	Mm	Mu	Ro
d5	match(exp(y), exp(x))	•	—	•	•	•	•	•
d6	match(e^x, exp(x))	•	—	•	•	•		
d7	match(exp(y), e^x)	•	—	•	•	•		•
d8	match(e^x, e^x)	•	—	•	•	•	•	•
d9	match($\sqrt{y}$, $\sqrt{x}$)	•	—	•	•	•	•	•
d10	match($y^{1/2}$, $\sqrt{x}$)	•	—	•	•	•	•	
d11	match($\sqrt{y}$, $x^{1/2}$)	•	—	•	•	•	•	•
d12	match($y^{1/2}$, $x^{1/2}$)	•	—	•	•	•	•	•
d13	match(iy, ix)		—	•	•	•	•	•
d14	match($\sqrt{-1}y$, ix)	ε	—	•	•	•	•	•
d15	match($(-1)^{1/2}y$, ix)	ε	—	•	•	•	•	•
d16	match(iy, $\sqrt{-1}x$)	ε	—	•	•	•	•	•
d17	match($\sqrt{-1}y$, $\sqrt{-1}x$)	•	—	•	•	•	•	•
d18	match($(-1)^{1/2}y$, $\sqrt{-1}x$)	•	—	•	•	•	•	•
d19	match(iy, $(-1)^{1/2}x$) [55]	$\mathcal{L}$/ε	—	•	•	•	•	•
d20	match($\sqrt{-1}y$, $(-1)^{1/2}x$) [55]	$\mathcal{L}$/•	—	•	•	•	•	•
d21	match($(-1)^{1/2}y$, $(-1)^{1/2}x$) [55]	$\mathcal{L}$/•	—	•	•	•	•	•

Originally published in *Computer Algebra Systems: A Practical Guide* edited by Michael J. Wester, John Wiley & Sons, Chichester, United Kingdom, ISBN 0-471-98353-5, xvi+436 pages, 1999.

Chapter 3 — Simplification

1 - Introduction

Let us consider the example: $\frac{d}{dx}\left(ax + x e^{x^2}\right)$ One uses:

$\forall x \quad u, v, c \qquad \frac{dc}{dx} \to 0 \qquad \frac{dx}{dx} \to 1$

$\frac{d(u+v)}{dx} \to \frac{du}{dx} + \frac{dv}{dx}$, $\frac{d(uv)}{dx} \to u\frac{dv}{dx} + \frac{du}{dx} \cdot v$

$\frac{d(u^v)}{dx} \to v\, u^{v-1} \frac{du}{dx} + (\log u)\, u^v \frac{dv}{dx}$

$\to a.1 + 0.x + 1.e^{x^2} + x\left[x^2 e^{x^2-1}\, 0 + (\log e).e^{x^2}.\{2.x^{2-1} + (\log x).x^2.0\}\right]$

"by hand" $a + e^{x^2} + 2x^2 e^{x^2} = a + (2x^2+1)e^{x^2}$

→ we need algorithms and/or techniques to automatically simplify expressions.

More generally, it is possible to list some simplifications which are neither mandatory or highly useful in computer algebra systems.

1) To suppress parenthesis: $((u)\,v) + w) = uv + w$

2) Identity simplification $0.u \to 0$, $1.u \to u$

$u/1 \to u$, $v^0 \to 1$ $(v \neq 0)$

$0^w \to 0$ $(w > 0)$

$A \cdot I \to A$ (matrices)

3) signs $(-u)(-v^3) = uv^3$ But, one may want: $-(a+b) \to -a-b$

4) Numerical simplifications: $5/8 - 1/8 \to 1/2$; $9! \to 362880$

many are non-trivial $(33282)^{1/2} \sin\left(\frac{13\pi}{6}\right) \to \frac{122}{\sqrt{2}}$

Pb with e^e, e, π, transcendental numbers decidability ?

5) Associative laws $(uv)w + (p+q) \to uvw + p + q$

6) Ordering : polynomial representation for instance.

7) To collect common factors : $u + (2/3)u \to 5/3\, u$

$2^{x+2} \quad 4.2^x$; $e^{5+\log k} \to e^5 e^{\log u} \ldots$

8) Operations on exponents: $(u^w)^v \to u^{vw}$; $(uv)^w \to u^w . v^w$

9) Distributivity laws: $(u+v)w \to uw + vw$

$(p-q)(u+v) \to pu - qu - qv + qv$

10) Power expansions $(a+b)^2 = a^2 + 2ab + b^2$

but: $(1+x)^{100}$??

11) gcd simplification: $\frac{4u^4 + 12u^3 + 12u^2 + 4u}{2u^4\ 2u^3 - 2u^2 + 2u} \to \frac{2u+2}{u-1}$

but $\frac{u^{100} - v^{100}}{u - v} \xrightarrow{??} u^{99} - uv^{98} + \ldots + v^{99}$

12) Rational expressions $\frac{1}{x}\ \frac{1}{y} - \frac{(y-x)}{xy} \to 0$

$x + \frac{x^3}{9} + \frac{x^5}{25} + \frac{x^7}{49} \to (225x^7 + 441x^5 + 1225x^3 + 11025x)/11$

13) Special arguments: $\sin 0 \to 0$ $\cos 0 \to 1$; $\log 1 \to 0$

or $2k\pi i$ $\log e \to 1$ or $2m\pi i + 1$ $\arctan 1 \to \frac{\pi}{4}$ or $\frac{\pi}{4} +$

14) Radical simplification :

$$\sqrt{(x^2-x+1)(x+\sqrt{x-1})} - \left(x+\sqrt{x-1}\,\sqrt{x-\sqrt{x-1}}\right) \to 0$$

$$\left(2^{1/3}+4^{1/3}\right)^3 - 6\left(2^{1/3}+4^{1/3}\right) - 6 \to 0$$

15) Transcendental identities : simplification or complexification of expressions ?

$\cos(-x) = \cos x$; $\cos(u+v)$; $\log(uv) = \log u + \log v$

$$\log\left(e^{2x}+2e^x+1\right) - 2\log\left(e^x+1\right) \to 0$$

$$\log\tan\left(\frac{x}{2}-\frac{\pi}{4}\right) - \sinh^{-1}\tan x \to 0$$

16) polynomial factorization.

Plus many, many other possible simplifying manipulations such as: taking derivatives, differentiating, integrating, matrix operations, equation solutions, power series expansions

Simplification problem ranges from trivial questions ($a+x$ or $x+a$) to utmost difficult ones (such as: is $\sin(\pi+e)$ a rational ?).

2 - Classification of systems based upon simplification

Simplification plays the most important part in system design. Indeed, when the designer has decided on the following points:

- data representations and structures,
- what change of representations are automatically performed,
- which such changes are under user's control,
- what are the additional simplification procedures the system includes

nothing really important is left to be decided upon.

The classical examples are: $(x+y)^{1000}$ → keep ? → expand ?

$x^{1000} - y^{1000}$ → keep ? → factorize ?

It exists a classification of the CAS systems (CAS), due to J. Moses, which is based upon their simplification capabilities and procedures. In fact the criteria are : is an expression given by a user modified by the CAS and to which extent.

1) Radicals : the expression is radically modified and is transformed into a well defined internal representation. As a result, these can only manipulate well defined classes of expressions (polynomials, rational functions, truncated power series, Poisson series, ...). For each of these classes, there is a canonical form

→ no algorithmic simplification. Simplification simply amounts to write expressions in canonical forms.

For instance, for multivariate polynomials, a precedence ordering is imposed upon the variables ($x > y > z > \dots$) and a polynomial is seen as a polynomial with respect to the leading variable.

$$3x^2y^2 - 2x^2yz^3 + 5x^2z^2 + 4x - 6y^3z + yz^3 + 3y^2 + z^4 + 1$$

is represented as:

$$\left(3y^2 - (2z^3)\,y + 5z^2\right)x^2 + (4)\,x + \left((-6z+1)\,y^3 + (z)\,y^2 + z^4 + 1\right)$$

Difficulties arise for : $(1+x)^{1000}$ which will be expanded, which is

- rational function system : $\frac{P}{P'} = \frac{\text{canonical form}}{\text{canonical form}}$

A radical system combines sums of quotients into a quotient

and looks for the gcd's of numerators and denominators

For instance $\frac{2x+3}{x^2+2x+5} - \frac{1}{3x+2} + \frac{3x}{x^2+2x+6}$

will be combine into one quotient which is not the solution one usually wants.

Radical systems have been extended to more general expressions such as exponentials, log., trigonometric functions. For instance a "canonical" expression is introduced for $\sin x$

$$\sin x = \frac{e^{ix} - e^{-ix}}{2i}$$

This leads to: $\int \sin x \cos^{10} x \, dx = \int \frac{1}{2i}\left(y - \frac{1}{y}\right) \frac{1}{2^{10}} \left(y + \frac{1}{y}\right)^{10} \frac{dy}{iy}$

which is not convenient to use.
Examples of such CAS have to be found in "old generation" CAS such as SAC-1, ALPAK.

2) <u>New Left</u> Such a CAS will not necessary transform an expression into its canonical representation.
⇒ users have some control which is obtained by not imposing canonical representations in a systematic way.
Examples: REDUCE, ALTRAN
Such a system handles general data types and expressions. They are easy to extend. An idea behind these capabilities is to "label" some data types.
For instance: 1) $x e^x + x^2 \sin x$. The "labels" $y = e^x$, $z = \sin x$ are introduced and give $xy + x^2 z$

2) $\frac{e^{2x} + e^x}{e^x} \to \frac{y+z}{z}$ with $y = e^{2x}$ $z = e^x$.

For these CAS, simplification consists in canonical forms and specialized representations.

3) <u>Liberals</u>: They are based upon a very general representation of expressions. Simplifications are achieved by transformations close to those performed in hand calculations.
In general they:
- gather terms in sums, exponents in products
- have rules for $0 \wedge 1$.
- suppress some operators (i.e. $a+(b+c) \to a+b+c$, $a+(-b) \to a-b$)
- simplify (often, not always) for special arguments: ($\sin 2\pi \to 0$, $e^{2\log y + x} \to y^2 e^x$)

They differ from radicals and new left CAS since:
a) as for new left CAS: expansion is upon request,
b) $\frac{a}{b} + \frac{c}{d} \not\to \frac{n}{db}$ If this is required, the gcd is not automatically computed,
c) un-simplified forms are possible ($1.\sin x + 0.\cos x$) but not during the computation itself,
d) Expressions such as e^x, $x!$, $\sum_{i=0}^{n} c_i x^i$ are not represented by and substituted to labels,
e) an information has a local meaning. This means that $\sin$ in an expression may have a meaning different from $\sin(x)$ in another part of the same expression.

<u>Drawbacks</u>: - inefficiency (execution time but also storage)
<u>Advantage</u>: - "natural" style for expressions.

4) Conservatives. Main idea: it is not possible to imagine a simplification procedure which is optimal for all cases
→ few means to automatically simplify → help the user to design is own simplification system.
All CAS based upon this philosophy have all disappeared because simplification is always difficult to implement and always expensive.

5) Catholics. They offer several representations for an expression. Main idea when a technique does not work another is called in ⇒ the user must be capable of easely switching representations.
(MACSYMA, Scratchpad-1)

Drawback: system grows rapidly and is very large.

Example of a special representation which is picked up for computations in the field of Celestial Mechanics: Poisson series

$$\varphi(x,y) = \sum \{ P_j \cos(j \cdot y) + Q_j \sin(j \cdot y) \}$$

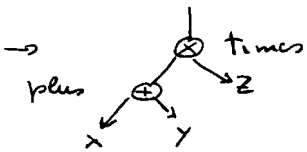
$\underline{x} = (x_1, \dots, x_m)$: variables in polynomials ; $j = j_1, \dots, j_n$: integers
$\underline{y} = (y_1, \dots, y_n)$: trigonometric variables
Q_j, P_j polynomials in x_i with rational or floating-point int. coefficients

A last remark about the simplification based upon representations is that if a conservative CAS is unable to realize that $\sqrt{6} = \sqrt{2}\sqrt{3}$, it is not always ruled out to prove with a "radical" system that $1 = \sqrt{1}\sqrt{1} = \sqrt{1} = \sqrt{(-1)(-1)} = \sqrt{-1}\sqrt{-1} = i \cdot i = -1$
(in the framework of algebraic numbers) ⇒ great care is required when designing a simplifier.

3. Simplification in systems

Simplification is usually performed at two levels of abstractions.
1) by algorithmic methods which are based upon well understood mathematical methods. They will be covered in the following section.
2) "Simple" simplification by programming the handling of expressions that we describe shortly in this section.

Many of the "easy" simplifications that are listed in section 1, must be performed at a very simple level, and very easely, when manipulating expressions.
It must be remembered that most often expressions are represented by (as in Lisp) a tree structure.
For instance $(X+Y) * Z$ →



each of these basic operators (*, + ...) has well defined properties it is thus natural to have for the functions for * and + functions which perform the simplification relevant for each them. In practice TIMES is replaced by $\leq$TIMES (Simple product command) which performs both the * operation and applies the possible simplification rules.
This idea can be extended to special cases. For instance it is possible to flag expressions in order to know if they correspond to factored polynomials, square-free factorization, gcd free rational fractions ... for instance. A continuous flow of information is thus available (cheaply) throughout a computation.

In a similar way one introduces SPLUS, SEXPT, SQUOTIENT as well as simplifiers for absolute values, trigonometric functions, logarithms....

There is another simplification which is rather easy to implement: the simplification for common sub-expressions. The idea consists in memorizing only one expression when the simplifier gives back the original expression.

The general "philosophy" of such an approach is that a simplifier can be design as a command driven transformer of canonical expressions (or forms).

Associated to the simplifier, one has a lexicographical ordering. This is required to get (for instance) $a+b \to a+b \to \geq b$.

User's driven simplification: How to allow the feature which will a simplifier of the above mentioned form?

This is readily realized through two possibilities:

(i) To suppress the built-in simplification rules by redefining S(OPERATOR) along new simplification rules.

(ii) To make use of a command which is usually available (whatever the acronym for it is): DECLARE. This allows to introduce the properties of commutativity (as an example one may have DECLARE(F, COMMUTATIVE) $\Rightarrow$ [F(A,B) - F(B,A) $\to$ 0]) additivity, anti-symmetry, linearity, associativity (left, right or both), parity ....

Such a simplifier is present in Macsyma. In addition, generalized simplifier packages are available for: canonical rational expressions, sums simplification, Taylor's series, trigonometric expressions, Poisson's series....

Another technique for simplification present in MACSYMA is the following ones..

Pattern-matching identification

So far we have consider syntactic (or lexicographic) identification. We now turn our attention towards semantic identification.

This is done by relying on the predicate concept.

Example: Let us consider $SIGINT(x) = \begin{cases} -1 & x < 0 \\ 0 & x = 0 \\ 1 & x > 0 \end{cases}$ (Signature of an integer).

Now: NEGATPRED(x): if SIGINT(x) = -1 then true else false

The DECLARE command permits to associate a form to a predicate. For instance

DECLARE (name, predicate ($arg_1, \ldots, arg_n$))

ex: DECLARE (A, NEGATPRED(X)).

The advantage of semantic identification is to be able to make use very simply of the properties of operators (such as addition is commutative with 0 identity) to identify among the different parts of a form and the presence of this form in an expression. For instance, for a polynomial one may find a unique form P with immediate identification of the coefficients.

Identification definition is made by a command of the type DEFMATCH which defines a new program (predicate) which

will be enforced only if a particular semantic form is identified.

DEFMATCH (program-name, form, var-form$_1$ .... var-form$_n$)

Example (of a fictious program)

- Defmatch (linear, $ax+b$, x)
- linear ($3y+4$, y) → $b=4$, $a=3$, $x=y$
- linear ($3y+4+x$, y) → $b=x+4$, $a=3$, $x=y$

Identification algorithm (taken from Macsyma)

Def: A unidentified variable (or non-paired variable) in a form is a declared variable without an assigned value.

A form p is compared to an expression e by identification of p to e, { if success → non-paired variables of p are assigned a value; if failure → False.

Def: A form where all variables are paired is a fixed form (constants, numbers, atomic-name ....)

Def: A form is anchored if once all the fixed parts are removed from the instantiated expression of the form:

either (i) the remaining part of the form consists of an isolated non-paired variable which is neither in a sum nor in a product.

or (ii) there is at least a fixed sub-part of the form such that any expression which represents it can be splitted into at least two parts, each one corresponding to an anchored sub-part of the original form.

The main goal of the algorithm is to identify anchored forms. When a form p identifies an expression e, the algorithm performs the following steps:

I If a form p is fixed, p identifies/recognizes the expression e iff p-e is 0 after simplification (by using canonical simplification).

II If p is a sum, Σa_i, all fixed a_i's are removed from e and the remaining a_i's are examined as follows.

(A) if a_i is a product with more than one non-paired variable ⇒ ambiguity (see (E))

(B) if a_i is a product of a declared variable, v, and a fixed form f, then the predicate of v is applied to the coefficients of f in e. If false ⇒ identification fails, otherwise → continue

(C) If a_i is a non-paired variable ⇒ it must be the only non-paired a_i. (Possibility to have selectors)

(D) If a_i is an exponentiation: several possibilities (fixed basis, fixed exponent, none fixed)

(d_1) fixed basis: one searches in e an exponentiation operator with same basis. If true → exponents are compared. If false → exponent 1 is possible → test (basis = sum → subtracted from e for test for exp. 1)

(d_2) fixed exponent: one looks for an exponentiation operator with same exponent. If true → basis forms are compared. If false and exponent < 0 (integer), one substracts 1 from e

and the basis form is compared to 1 (when no denominator). Otherwise : comparison to 0 (exponent > 0)

Meaning : $a + 1/b$ is identified to $x + a$ (for $a = x$, $b = 1$)
and to x ($a = x-1$, $b = 1$)
$a + b^2$ is identified to x ($b = 0$, $a = x$)

(d3) None fixed : one looks for an identification (see E)

(d4) Special case : one looks for an exponentiation which is the only element of the non-paired sum ($y^x + A$ where A is identified and subtracted).
if B (basis) is fixed : one looks also for B ** E paired to 1
" E (exponent) " " 0.

(E) a_i is a specific function (sin, ...) : the first occurrence of this function is looked for, then test on the arguments

(F) if the name for a_i is not paired one searches to identify a_i to all the functions (even +, *, **) as in E.

III p is a product, Πa_i, one substitutes "divide" to "substract" and "product" to "sum" in II (except for II-A & B since products in products are not considered).

V p is an exponentiation. → similar to II-9 : d_1, d_2 & d_4.
When neither basis nor exponent is fixed (d_3), one proceeds as follows : a) e is 1 → p is compared to 1 ** 0
b) e is 0 → p " 0 ** 1
c) e is not a power → p compared to e ** 1
d) e is a power → respective basis & expon. are compared.

V p is a specific function : comparison after search in a table, search for the number of the arguments and identification of these arguments

VI p is a non-specificied function Proceeds as in V ~~case~~ w in addition a test on the function name.

VII p is an unidentified atomic variable ("atom" in Lisp)
⇒ p is compared to e

It is possible to mix and interleave these operations at any level of arbitrary depth. Thus algorithm is exhaustif : for ~~all~~ any expression which is syntactically accepted, a procedure of pattern matching and identification is defined.
Such an algorithm always terminates.

4. Zero equivalence problem

We have already realized (and this will be confirmed in the next section) that simplification is not a well defined problem.
There is however a sub-problem which is well defined : the problem of recognizing when an expression is equivalent to zero. This is the zero equivalence problem.
But, even this sub problem is quite difficult.
The main question is : for which classes of functions is the zero equivalence problem decidable ?
The answer is : the problem is not always decidable.

Theorem (Richardson) Let $\mathcal{R}$ be the class of expressions made of:

(1) rational numbers and π,

(2) a variable, x.

(3) the operations: addition ($+$), product ($*$), composition ($\circ$)

(4) sinus and absolute value ($|\cdot|$).

Then: for all expressions $E \in \mathcal{R}$, the predicate "E is identically zero" is recursively undecidable.

The proof of this theorem requires to use the theorem of Matiyasevič which shows that Hilbert's 10th problem is not solvable (see footnote)

Theorem (Matiyasevič, 1970): It exists a set of polynomials over the integers $\mathcal{P} = \{P(x_1, \dots, x_n)\}$ such that $\forall P \in \mathcal{P}$ the predicate "$\exists$ non-negative integers $a_1, \dots, a_n$ such that $P(a_1, \dots, a_n) = 0$" is recursively undecidable.

Let us examine how this theorem applies to the case of univariate polynomials.

Notations: if $f: \mathbb{R} \to \mathbb{R}$ is a function, $f^{(n)}(x)$ means: $f^{(0)}(x) = x$, $f^{(n+1)}(x) = f(f^{(n)}(x))$ for $n \geq 0$

Lemma 1: Let $h(x) = x \sin(x)$, $g(x) = x \sin(x^3)$. Then, $\forall$ $a_1, a_2, \dots a_n \in \mathbb{R}$ and $\forall$ $0 < \varepsilon < 1$, $\exists$ b such that $\forall$ $1 \leq k \leq n$ $\quad |h(g^{(k-1)}(b)) - a_k| < \varepsilon$

Footnote: Hilbert's Pb 10 (Diophantine problem). Does it exist an algorithm to decide if a multivariate integer polynomial ($\mathbb{Z}$) has integer solutions? (i.e. solutions of $P = 0$)

Sketch of the proof: proof is by induction on n. Richardson proves that given 2 real numbers a_1 and a_2, $\exists$ $b > 0$ such that $|h(b) - a_1| < \varepsilon$ and $g(b) = a_2$. Now, suppose that the lemma is true for n. Then, it exists b' such that $|h(b') - a_2| < \varepsilon$, $|h(g(b')) - a_3| < \varepsilon$, ... $|h(g^{(n-1)}(b')) - a_{n+1}| < \varepsilon$. Then, it exists b such that $|h(b) - a_1| < \varepsilon$ and $g(b) = b'$ and the result is valid for $(n+1)$ ∎

Meaning of this lemma: any finite collection of real numbers can be represented by a real number with an arbitrary position. This method is certainly not unique → one assumes that any n-tuple of real numbers $(x_1, \dots, x_n)$ can be (approximately) represented by a real number x. This leads to

Definition: A function $F(x_1, \dots, x_n) \in \mathcal{R}$ is dominated by $G(x_1, \dots, x_n) \in \mathcal{R}$ if for any real nb $x_1, \dots, x_n$

(i) $G(x_1, \dots, x_n) > 1$

and (ii) $\forall$ $\Delta_1, \Delta_2, \dots \Delta_n \in \mathbb{R}$ such that $|\Delta_i| < 1$,

$$G(x_1, \dots, x_n) > F(x_1 + \Delta_1, \dots, x_n + \Delta_n)$$

Lemma 2: $\forall$ $F \in \mathcal{R}$ it exists a dominant function G.

Proof is by induction on the number of operators in F. For instance one applies the following transformations on F, while assuming that f, f_1, f_2 are dominated by g, g_1, g_2 respectively. Then

if $F = f_1 + f_2$ → $G = g_1^2 + g_2^2 + 2$

if $F = f_1 * f_2 \longrightarrow G = (g_1^2+2)*(g_2^2+2)$

if $F = x \longrightarrow G = x^2+2$

if $F = \sin(f) \longrightarrow G = 2$ ∎

Then Richardson proves the following important theorem.

Theorem: For all $P \in \mathcal{S}$, it exists $F \in \mathcal{R}$ such that

(i) $\exists$ a n-tuple of non-negative integers $\vec{a} = (a_1, \dots, a_n)$ such that $P(\vec{a}) = 0$ iff

(ii) $\exists$ $\vec{b} = (b_1, \dots, b_n)$ $b_i \geq 0$, b_i real number such that $F(\vec{b}) < 0$.

The proof is very technical and is skipped. The proof of Richardson's theorem follows up by using techniques and results from Logic. One shows that: for $F(x) \in \mathcal{R}$, if it is possible to decide that $F(x) \equiv 0$, then one may decide that $\exists\, b \in \mathbb{R}$ such that $G(b) < 0$ with $F(x) = |G(x)| - G(x)$. One shows that these two statements are contradictory.

---

the question Richardson asks himself is: is it possible to decide if an expression is equivalent to zero in an interval of real numbers? The proof, sketched above, relies on a reduction method which consists in considering the zero equivalence problem for less complex expressions. For Richardson, the complexity degree is lexicographic, not algebraic (i.e. e^{e^x} is more complex than e^x)

in fact the complexity degree (definition) is irrelevant provided it is kept fixed.

This theorem of Richardson shows an undecidability result. The following one demonstrates a decidability feature.

Decision algorithm (Richardson)

Consider the class of functions generated by:

(1) the rationals and π
(2) the variable x
(3) $+$, $*$ and quotient $(/)$
(4) Log $|u|$ and $\exp(u)$ (u: any function in Russ

It is possible to prove that the predicate "$F(x) \equiv 0$ is true / false" amounts to prove that a constant expression (depending on F) is zero. $\Rightarrow$ it exists a decision procedure (not really an algorithm) for this class.

A brief sketch of this procedure is as follows.

Let F a function in the class above. F is decomposed into sub expressions (in a certain way rather arbitrarily) and they are ordered "by complexity". We assume that if y_1 is a sub-expression of y_2, then y_1 is of complexity less than y_2

Let us suppose that y is the sub-expression of F of greatest complexity and $y = \log(u)$

F is decomposed into: $F = a_n y^n + a_{n-1} y^{n-1} + \dots + a_1 y + a_0$

the decision procedure is applied to a_n.

If $a_n \equiv 0$, then this decision procedure is applied to

$$F_1 = a_{n-1} y^{n-1} + \dots + a_0$$

which of lesser complexity than F and

$$a_n \equiv 0 \Longrightarrow [F \equiv 0 \Leftrightarrow F_1 \equiv 0]$$

if $a_n \neq 0$ then

$$F_2 = \frac{F}{a_n} = y^n + \frac{a_{n-1}}{a_n} y^{n-1} + \cdots + \frac{a_0}{a_n}$$

Then :

$$F_3 = F_2' = n y^{n-1} y' + \cdots + \frac{(a_n a_0' - a_0 a_n')}{a_n^2}$$

Then : $F_2 \equiv 0 \Rightarrow F_3 \equiv 0$ and $F_3 \equiv 0 \Rightarrow F_2 \equiv$ constant

The procedure proceeds along this scheme. It is based upon the fact that the derivative is less complex than the function y.

For instance, if $y = \log(u) \rightarrow y' = \frac{u'}{u}$ and (u^{-1} and u') being less complex than y, the procedure may proceed.

But, if $y = e^u$ this is no longer true Let us examine this case :

$$F = a_n y^n + \cdots + a_0 \qquad y = \exp(u)$$

Let us apply the decision procedure to a_0

if $a_0 \equiv 0 \rightarrow F_1 = a_n y^n + \cdots + a_1 y = Q y$

if $a_0 \neq 0 \qquad F_2 = F/a_0$

$$F_2' = \left(\frac{a_n' a_0 - a_n a_0' + n a_n a_0 u'}{a_0^2} \right) y^n + \cdots$$

$$+ \frac{a_1' a_0 - a_1 a_0' + a_1 a_0 u'}{a_0^2} y$$

Thus $F_2'/y \equiv 0 \rightarrow F \equiv$ constant (and F_2'/y is of lesser complexity). The procedure thus reduces to the question of to determine if the value of an expression for a constant is 0.

Since this problem is in its full generality undecidable, this is not really an algorithm but a procedure valid for the class of functions which is under consideration here

5 An attempt to formalize the problem of simplification

The simplification problem shows two aspects :

1) To get equivalent but simpler objects
2) To compute canonical representations (see next chapter) for equivalent objects.

Let T be a class of objects (expressions). For instance : terms (of 1.st order), logical formulas (a reduced class a class of programs) ...

Let $\sim$ be an equivalence relation on T (functional equivalence, equality coming from axioms, congruence modulo an ideal)

1) The pb of finding simpler equivalent objects consists in finding an effective procedure S ($S : T \rightarrow T$) which satisfies the following requirements $\forall t \in T$

(i) $S(t) \sim t$

(ii) $S(t) \leq t$

where "$\leq$" refers to the complexity concept which has been selected (for instance : "shorter", "less memory"...) In fact (and obviously) $S(t) \leq t$ is required for <u>all</u> $t \in T$.

2) "Canonical simplification" consists in finding a procedure S (canonical simplifier) for $\sim$ ($S : T \rightarrow T$) which verifies for all $s, t \in T$

(i) $S(t) \sim t$

(ii) $s \sim t \Rightarrow S(s) \sim S(t)$

$S(t)$: canonical form of t.

Example: a trivial canonical simplifier is to transform elementary arithmetic operations into polynomials.

$\frac{1}{2}(2x+2)(x-1) \rightarrow x^2-1$

Problems 1 & 2 are not totally independent. Indeed, a canonical simplifier defines a measure of "simplicity". Also, a definition for simplicity may induce a canonical simplifier. On the opposite, a canonical form may produce expressions which are not the simplest possible ones. For instance, the canonical transformation for $(x+1)^2$ is (x^2+2x+1).

In CAS the problem of designing a canonical simplifier is basic since it is directly linked to:

- effective calculations in various algebraic domains
- the problem of effectively deciding on equivalence relations

In practice, a canonical simplifier immediately generates an algorithm to decide on equivalences and to compute in the algebraic domain defined by this equivalence.

Theorem (Canonical simplification and equivalence decidability)

Let T be a set of decidable objects, ν an equivalence relation on T. Then, ν is decidable iff it exists a canonical simplifier S for ν

Example: Let us consider an abelian semi-group defined by the generators a, b, c, f, ρ and the definition relations

$a\rho = c^2\rho \qquad b\rho = c\rho \qquad \rho = f$.

The (rewrite) rules $\rho \rightarrow f \quad cf \rightarrow bf \quad b^2f \rightarrow af$ constitute a canonical simplifier (this may be proven)

$a^5 b\, c^3 f^2 \rho^3$ and $a^5 b^2 c^2 \rho^5$ represent the same element of the semi-group since their canonical forms are identical. $a^7 f^5$

$(a^5 b\, c^3 f^2 \rho^3 \xrightarrow{cf \rightarrow bf} a^5 b^3 c f^2 \rho^3 \xrightarrow{b^2 f \rightarrow af} a^6 bc f^2 \rho^3$

$\xrightarrow{cf \rightarrow bf} a^6 b^2 f^2 \rho^3 \xrightarrow{b^2 f \rightarrow af} a^7 f^2 \rho^3 \xrightarrow{\rho \rightarrow f} a^7 f^5$

and similarly for $a^5 b^2 c^2 \rho^5$)

But $c\rho^2$ and $c^2\rho$ are 2 different elements because their canonical forms, bf^2 and af, are distinct

$c\rho^2 \xrightarrow{\rho \rightarrow f} cf^2 \longrightarrow bf^2 \; ; \; c^2\rho \rightarrow c^2 f \rightarrow cbf \rightarrow b^2 f \rightarrow af$

Theorem (Canonical simplification and computation)

Let T be a decidable set, R a binary operation which is calculable on T and ν an equivalence relation on T which is a congruence with respect to R. Let us assume that S is a canonical simplifier for ν. Define:

$\mathrm{Rep}(T) := \{ t \in T \mid S(t) = t \}$ set of canonical representatives

$R'(s,t) := S(R(s,t)) \quad \forall s,t \in \mathrm{Rep}(T)$.

Then $(\mathrm{Rep}(T), R')$ is isomorphic to $(T/\nu, R/\nu)$, $\mathrm{Rep}(T)$ is decidable and R' is computable. //

(One may use the following notation $R/\nu\,(C_s, C_t) := C_{R(s,t)}$

C_t: congruence class of t with respect to ν)

This theorem means that when one possesses a canonical simplifier for an equivalence relation which is a congruence with respect to a computable operation, one can "master" algorithmically the structure of the factors linked to R.

Example : The rewrite-rules $x^3 \to x^2$, $x^2y \to x^2$ define a canonical simplifier S on $T = \mathbb{Q}[x,y]$ for the congruence relation modulo the ideal I generated by the 2 polynomials $x^3y - x^3$ and $x^3y - x^2$ (This can be demonstrated).

Rep (T) : set of polynomials $f \in \mathbb{Q}[x,y]$ such that no monomial in f is a multiple of x^3 or x^2y.

—|

Since some functional equivalences are undecidable, one introduces concepts of simplification which are weaker than canonical:

- normal simplification (or 0 equivalence)
- regular simplification

Normal simplification A computable application of T in T is said to be a normal simplifier (or of 0-equivalence) for the equivalence relation $\sim$ in T iff $\forall\, t \in T$:

(i) $S(t) \sim t$

and (ii) $t \sim 0 \Longrightarrow S(t) = S(0)$

One may prove the properties:

1) a canonical simplifier is a normal simplifier

2) if $\exists$ M, operation in T | $s \sim t \Leftrightarrow M(s,t) \sim 0$

then [$\exists$ normal simplifier] $\Rightarrow$ [$\exists$ canonical simpl. for $\sim$]

Regular simplification is used in a context where transcendental functions (exp., log, ...) are used. It is only necessary for our purposes to know that a regular simplifier guarantees that all non-rational terms in the simplified expression are algebraically independent.

A. Canonical simplification for polynomials and rational expressions.

A1 Polynomials a/ Let $\mathcal{R}$ be the set of all terms of the form $a_n x^n + \dots + a_1 x + a_0$ $(n \geq 0)$ $a_i \in \mathbb{R}$, $a_n \neq 0$

$\Rightarrow \mathcal{R} \cup \{0\}$ form a system of canonical forms for $\mathcal{R}[x]$

b/ multivariate case : $x^{j} \doteq x_1^{j_1} \dots x_n^{j_n}$

$\{ a_{i_1} x^{i_1} + \dots + a_{i_m} x^{i_m} \}$ and $\{0\}$ form a system of canonical forms for $\mathcal{R}[x_1, \dots, x_n]$

c/ One may also use for polynomials a canonical recursive form

ex. $(3x^2+5)y^2 + (x-3)y + (x^3 - 2x + 2)$

Polynomials over groups Let G be a group, $\mathcal{R}$ the set of forms $a_0 x^{n_1} a_1 x^{n_2} \dots a_{r-1} x^{n_r} a_r$ where $r \in \mathbb{Z}$, $r \geq 0$ $n_i \in \mathbb{Z}$, $n_i \geq 0$, $(a_i \in G \quad i = 0, 1, \dots r)$ and $(a_i \neq 1$ for $i = 1, 2, \dots, r-1) \Rightarrow \mathcal{R}$ is a system of canonical forms for $G[x]$.

- It is also possible to define canonical forms for : lattices, boolean algebras ...

(See the book of Lausch & Nöbauer, "Algebra of polynomials" where "canonical form" is called "normal form")

A2 Rational expressions

One looks for a canonical simplifier for the equivalence

Simplification of radicals

1) Richard Zippel (1985)
Simplification of Expressions involving Radicals
JSC, 1, pp 189-210

2) David J. Jeffrey and Albert D Rich (1999)
Simplifying Square Roots of Square Roots by Denesting
In "Computer Algebra Systems: A Practical Guide
M. J. Wester Ed., Wiley 1999

relation defined in the domain:
$R[x_1, \ldots, x_n] \times (R[x_1, \ldots, x_n] - \{0\})$

One distinguishes among the cases where rational expressions are
not quotient of polynomials $\left(x - \frac{x}{x^2}\right)/(x+1)$
in this case one uses rational arithmetic in algebraic domains and gcd's.
quotient of polynomials $(x^2-1)/(x-1)$
one relies on gcd's.

B - Canonical simplification of expressions with radicals.

variables ($x_1, \ldots, x_n$), constants, arithmetic functions ($+, -, *, /$) and root sign $\sqrt[n]{}$ (or non integer exponents, s^n)
=> . It exists a canonical simplification algorithm for non overlapping radicals ($\sqrt{\sqrt{x}}$ is ruled out)

. when expressions have overlapping radicals $\sqrt{\sqrt{}}$ factorial results are known only.

C Transcendental functions

Several classes of expressions are defined from variables (x_i), constants, π, e, arithmetic functions ($+, -, *, /$, absolute value), exp, log, sin, error function (erf)

No general solution
Some irreducibility to canonical form results (Richardson's)

3 Olga Caprotti
(RISC Linz, Austria)

Tutorial: OpenMath

Introductory Tutorial

Olga Caprotti
RISC-Linz, Research Institute for Symbolic Computation

Johannes Kepler University, Linz, Austria

September 28, 2002

Pisa Summer School 2002
September - October 2002
Pisa, Italy

-

Copyright © 2002 Olga Caprotti

Goal

"*OpenMath* is a standard for the communication of semantically rich mathematical objects allowing them to be *exchanged* between computer programs, *stored* in databases, or *published* on the WWW"

- **semantically unambiguous**
- **extensible**
- **machine-readable, machine independent**

History

***OpenMath* initial series of workshops held in**

- **Zurich (1993)**
- **Oxford (1994)**
- **Amsterdam, Copenhagen (1995)**
- **Bath, Dublin, Zurich (1996)**
- **Nice, Yorktown Heights (1997)**
- **Berlin, Tallahassee (1998)**

Objectives and basic design of *OpenMath* document was produced (published as Abbott_Leeuwen_Strotmann_98)

Society

Worldwide OpenMath activities are coordinated within the OpenMath Society [http://monet.nag.co.uk/cocoon/openmath/society/index.html], based in Helsinki, Finland:

- **workshop organization**
- **webpage maintainance**
- **discussion forum**
- **standard promotion**

The OpenMath Standard

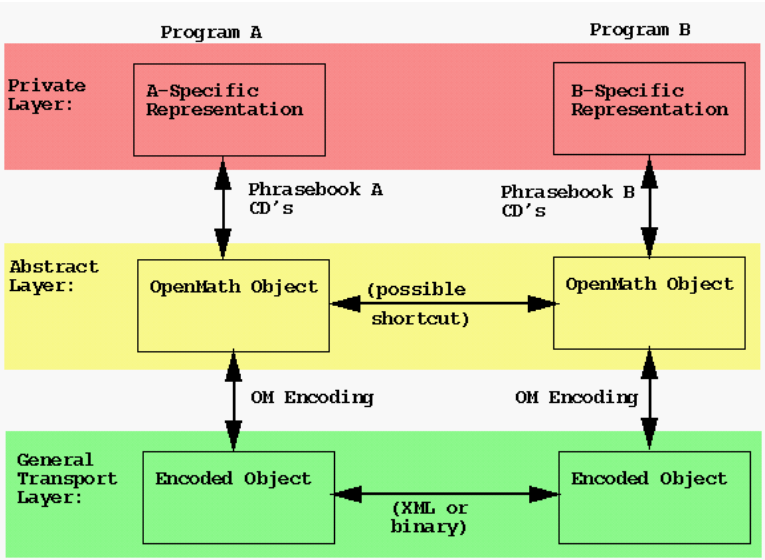
The OpenMath Standard [http://monet.nag.co.uk/cocoon/openmath/standard/] describes the structure of OpenMath objects, encodings and content dictionaries. Version 1.0 is available as

- Pdf [http://monet.nag.co.uk/cocoon/openmath/standard/omstd.pdf]
- Docbook+SVG+MathML XML [http://monet.nag.co.uk/cocoon/openmath/standard/xml/omstd.xml]
- XHTML+MathML+SVG [http://monet.nag.co.uk/cocoon/openmath/standard/xml/omstd.html]

Architecture

The architecture of *OpenMath* and summarizes the interactions among the different *OpenMath* components.

Figure 1. OpenMath Architecture



There are three layers of representation of a mathematical object.

- private: internal representation used by an application
- abstract: representation as an *OpenMath* object
- communication: translation of the *OpenMath* object representation to a stream of bytes

An application usually manipulates the mathematical objects using its internal representation, but can convert them to *OpenMath* objects for communicating their byte stream representation.

Abstract Objects

- **Basic Objects [omstd/omstd.xml#sec_basic]**
- **Compound Objects [omstd/omstd.xml#sec_compound]**

Encodings

Two encodings map between *OpenMath* objects and a low level byte stream representation that can be easily exchanged between processes (via almost any communication method) or stored and retrieved from files.

- **ISO 646:1983 characters (ascii) and is an xml application. [1] (DTD [omobj.dtd], XSD [omobj.xsd])**
- **binary encoding meant to be used when the compactness of the encoding is important**

[1]xml markup uses only ascii characters, however OpenMath strings may use arbitrary Unicode/ISO 10646:1988 characters

The xml Encoding

This encoding has been designed with two main goals in mind:

1. **to provide an encoding that uses the most common character set (so that it can be easily included in most documents and transport protocols) and that is both readable and writable by a human.**
2. **to provide an encoding that can be included (embedded) in xml documents.**

Phrasebooks

Programs manipulating OpenMath objects according to certain directives:

- **typically, evaluation using some mathematical package**
- **note: convert between OpenMath and the package syntax**

Mathematical Software

Mathematical software, (e.g. Mathematica, Gap, CoCoA, Coq and R), is accessed via JAVA service wrappers of the RIACA OpenMath library that provide TCP/IP socket interface

OpenMath aims at supporting electronic communication and manipulation of mathematical content by abstracting from specific syntax to concentrate on an abstract representation of mathematics.

Mathematical services based on OpenMath can be developed *independently* of the software package used for computing.

Libraries and Tools

A brief survey of what is out there...

RIACA OpenMath Library

Java library providing classes

- **to represent abstract OpenMath objects, nl.tue.win.riaca.openmath.lang [projects/om/lib/build/doc/api/nl/tue/win/riaca/openmath/lang/package-summary.html]**
- **read and write objects in XML encoding, nl.tue.win.riaca.openmath.io [projects/om/lib/build/doc/api/nl/tue/win/riaca/openmath/io/package-summary.html]**
- **encode/decode e.g. to MathML, nl.tue.win.riaca.openmath.codec [projects/om/lib/build/doc/api/nl/tue/win/riaca/openmath/codec/package-summary.html]**
- **basic Phrasebooks, e.g. to Mathematica, Gap, nl.tue.win.riaca.openmath.phrasebook [projects/om/lib/build/doc/api/nl/tue/win/riaca/openmath/phrasebook/package-summary.html]**

OpenMath JSP Tag Library

The Riaca OpenMath tag library [projects/taglibs/build/taglibs/index.html] project has for objective the creation of several OpenMath related JSP tag libraries.

Other libraries

- C, C++ library available from INRIA ftp://ftp-sop.inria.fr/safir/OM
- Corba library available from LBA http://www.mrg.dist.unige.it/~nembo/
- OmdocJdom library available from ActiveMath http://www.activemath.org/~ilo/java-doc/ [http://www.activemath.org/~ilo/]

XML Tools

Editors for working with XML encoding

- JOME, JavaBean OpenMath Editor from http://mainline.essi.fr/jome/jome-en.html
- QMath [http://www.activemath.org/~paul/AuthoringComments/soft/QMath/index.html] converts its standard input (in QMath syntax) to the standard output (in OMdoc)
- XML editors based on DTDs (emacs psgmlx-mode!)
-

Markup Languages

- OMDoc [http://www.mathweb.org/omdoc/], Open Mathematical Documents, allows to represent the semantics and structure of various kinds of mathematical documents, from articles, textbooks, to mathematical knowledge bases.
- MathBook [http://www.riaca.win.tue.nl/public/mathbook/index.html], a mixture of DocBook with OpenMath, is used for developing interactive books. It comes with a range of tools (mainly Java). (mathbook.xsd [mathbook.xml])
- Mathbroker [http://poseidon.risc.uni-linz.ac.at:8080/index.html], generalization of WSDL [http://www.w3.org/TR/wsdl] to mathematical services, is a language for describing the mathematical problems solved by the services

Mathematical Services

- JavaMath API [http://javamath.sourceforge.net/] enables mathematical programs in Java to use the computational capabilities of existing compute engines.
- OpenXM [http://www.openxm.org] (Open message eXchange for Mathematics) infrastructure for mathematical software providing a protocol for interactive distributed computation (currently Risa/Asir, Kan/sm1, OpenMath/XML)
- MathWeb-SB [http://www.mathweb.org/mathweb/index.html], a software bus for MathWeb connects a wide-range of mathematical services
- IAMC [http://icm.mcs.kent.edu/research/iamc.html] Internet Accessible Mathematical Computation

Workbenches

- **OpenMath Shell [http://www.riaca.win.tue.nl/products/openmath/shell/index.html], a mathematical workbench based on Java webstart**
- **OMSW [http://www-sop.inria.fr/cafe/Manuel.Bronstein/cathode/software.html], OpenMath Worksheet and OMD (OpenMath Dispatcher) is a client/server system for offering easy access to OpenMath - based computation over the internet (out-of-date?).**

Symbolic Integration Service

- **constructs the OpenMath object for the indefinite integral of the input**
- **invokes the (Mathematica) Phrasebook to evaluate it**
- **returns the resulting OpenMath object in XML encoding and displays it using MathML**

Examples

Next we give examples on how to handle OpenMath using the ROML library and tools. More examples can be found in the software documentation.

A JSP Page

We construct a JSP page using the ROML taglibs to manipulate the XML encoding and to produce presentation-MathML for display.

```
<% response.setContentType( "text/xml" );%>
<%@ taglib uri="/WEB-INF/tlds/omcore-taglib.tld" prefix="omcore" %
<%@ taglib uri="/WEB-INF/tlds/util-taglib.tld" prefix="util" %>

<?xml-stylesheet type="text/xsl" href="style/pmathml.xsl"?>
<html xmlns:pref="http://www.w3.org/2002/Math/preference"
      xmlns="http://www.w3.org/1999/xhtml"
      pref:renderer="css">
```

We also need to use some Java OpenMath classes.

```
<%@page import="java.io.*,
                antlr.*,
                nl.tue.win.riaca.openmath.lang.*,
```

```
                nl.tue.win.riaca.openmath.io.*,
                at.ac.uni_linz.risc.perseus.*,
                at.ac.uni_linz.risc.openmath.expressions.parser.*"
%>
```

From a parser, we get a string that represents the OpenMath object in XML. This is how it is converted to an abstract OpenMath object.

```
String      tObj = walker.expr(tAST);
OMXMLReader tReader = new OMXMLReader( tObj );
OMObject    tObject = tReader.readObject();
OMObject    tAppl = (OMObject) tObject;
```

Now we are ready to display the OpenMath encoding:

```
<p><table border="0" cellspacing="5" cellpadding="5" width="95%" bgcol
   <tr>
```

Service Integrator

We use the ROML packages to provide the basic functions for constructing OpenMath objects, performing I/O operations are

```
....
import nl.tue.win.riaca.openmath.lang.*;
import nl.tue.win.riaca.openmath.io.*;
```

To connect to a computer algebra system we need the phrasebook package:

```
import nl.tue.win.riaca.mathematica.phrasebook.MathematicaSock
```

We construct the indefinite integral of the input expression by making a lambda expression in the integration variable (*x*) and then applying the in-

```
    <td><omcore:omtomathml id="Str" name="<%=tAppl%>" mode="displa
        The indefinite integral of
        <math display="block" xmlns="http://www.w3.org/1998/Math/M
         is
         <omcore:xmlomtojavaom id="resObj"><%=ret%></omcore:xmlomt
         <omcore:omtomathml id="oRes" name="<%=resObj%>" mode="dis
         <math display="block" xmlns="http://www.w3.org/1998/Math/
         <%=oRes%></math></p>
     </td>

</tr>
</table>
```

The full java source code [demo.jsp] for the JSP page.

tegration function to it.

```
OMBinding tIntegrandExpr = new OMBinding();
tIntegrandExpr.setBinder(new OMSymbol("fns1","lambda"));
Vector tBVars = new Vector();
tBVars.addElement(new OMVariable("x"));
tIntegrandExpr.setBoundVariables(tBVars);
tIntegrandExpr.setBody(fInputExpr);

OMApplication tIntegrand = new OMApplication();
tIntegrand.setHead(new OMSymbol("calculus1","int"));
tIntegrand.addElement( tIntegrandExpr );
```

Putting it all together, with the initialization of the phrasebook, we get the complete code.

```
public class  Integrator
    {

private String  mMethod ="EVAL";
```

```
private int mPort = 1235;
private String mHost =".....tue.nl";
private MathematicaSocketPhrasebook tMathematicaSocketPhrasebook
                  = new MathematicaSocketPhrasebook(mHost,mPort);

public String IndefInt(OMObject fInputExpr)
{
    String tOmContent = "";


    // prepare for output of XML encoding
    StringWriter tStringWriter = new StringWriter();
    OMXMLWriter tXMLWriter = new OMXMLWriter( tStringWriter );

    // construct the integrand expression wrt variable x
    OMBinding tIntegrandExpr = new OMBinding();
    tIntegrandExpr.setBinder(new OMSymbol("fns1","lambda"));
    Vector tBVars = new Vector();
    tBVars.addElement(new OMVariable("x"));
    tIntegrandExpr.setBoundVariables(tBVars);
    tIntegrandExpr.setBody(fInputExpr);

    OMApplication tIntegrand = new OMApplication();
    tIntegrand.setHead(new OMSymbol("calculus1","int"));
    tIntegrand.addElement( tIntegrandExpr );

    Vector tArguments = new Vector();
    tXMLWriter.writeObject(tIntegrand);
    tArguments.addElement( tStringWriter.toString() );
    try{
    tOmContent =
        tMathematicaSocketPhrasebook.perform( mMethod , tArgu
    }
    catch (Exception w)
        {System.out.println(w);}
    return tOmContent;
}

...
```

Conclusions

OpenMath is a technology support the electronic manipulation and exchange of mathematical objects.

Several related technologies are built on OpenMath.

The OpenMath Community is more and more becoming a reality.

References

http://www.openmath.org

http://poseidon.risc.uni-linz.ac.at:8080/ MSDL

http://perseus.risc.uni-linz.ac.at:8080/openmath test services (WSDL + OpenMath)

http://riaca.win.tue.nl/ OpenMath software

Q&A

Any questions?

4 James Davenport
University of Bath, England

Evening Talk: Computer Algebra or Computer Mathematics?

Course: Solving Systems of Polynomials: Theory and Practice

Computer Algebra or Computer Mathematics?

James H. Davenport[a]

October 8, 2002

Department of Computer Science,
University of Bath, Bath BA2 7AY, England
J.H.Davenport@bath.ac.uk

[a]The author was partially supported by the European OpenMath Thematic Network.

Scope

- "polynomial-type" systems: Axiom, Macsyma/Maxima, Maple Mathematica and Reduce.
- We quote Maple, but do not intend to condemn it.
- I do not ignore systems such as GAP, KANT, PARI and Magma: rather the thesis of this talk is (largely) irrelevant to them.

Strengths

- These algebra systems perform massive computations (Delaunay's 20-year computation of the orbit of the moon can now be done in under a second)
- They incorporate extremely sophisticated algorithms: integration, ordinary differential equations and Gröbner bases
- G.H. Hardy on integration: "There is no known process [algorithm], and there is reason to believe that no such can be given".

  Solved (Risch) since indefinite integration can be viewed as an algebraic processs, essentially anti-differentiation.

But these systems do not please the users

- Some of this is because users are never pleased;
- Some of this is because users have unrealistic expectations.
- but some of it is because there is a mismatch between what these systems do, and what users think they do. In particular, the users think that the systems are doing mathematics, whereas typically the systems are only doing that part of the mathematics that is algebra.

A little history

- The scope of the computer algebra systems of the 1960s was unashamedly limited to algebra: they did the tedious algebra, but it was the task of the user, generally an expert in the relevant mathematics, to see that the algebra made sense.
- As computers spread, and particularly with the advent of personal computers, these systems became available to a much wider audience, and the assumption that they were merely "algebra engines" and all the mathematical knowledge belonged to the user, had to be challenged.
- Mathematica is now explicitly sold as a "computer mathematics system". Do we know what "computer mathematics" is?
- Does "computer mathematics" make sense as one subject, or is "computer **R**" different from "computer **C**"?

Algebraic Numbers and their Fields

- The common way of constructing $\mathbf{Q}(\sqrt{2})$ algebraically is as $K := Q[\alpha]/(\alpha^2 - 2)$: the quotient of a polynomial ring by a principal ideal.
- In K, just as in $\mathbf{R}$, the equation $x^2 = 2$ has two roots, α and $-\alpha$. However, nothing says whether α corresponds to $\sqrt{2} = 1.4142$ or $-\sqrt{2} = -1.4142$.
- K as we have defined it is not an *ordered field*.
- How does K embed into $\mathbf{R}$?

For more general algebraic numbers, this information is no longer implicit in the definition of the roots of a polynomial. For example, the polynomial $f(x) := x^4 - 10x^2 + 1$ has four real roots, and we might have to code one possible embedding of $\mathbf{Q}$ extended by a root of this polynomial as

$$\mathbf{Q}[\alpha]/(\alpha^4 - 10\alpha^2 + 1) \wedge \alpha \in [3, 4].$$

The other possible embeddings have $\alpha \in [0, 1]$, $\alpha \in [-1, 0]$ and $\alpha \in [-4, -3]$. Once we have this interval information, a bisection process can refine the interval sufficiently that any two different elements of $\mathbf{Q}[\alpha]/(\alpha^4 - 10\alpha^2 + 1)$ can be compared.

There are several other approaches to distinguishing the real roots of a polynomial, e.g. by Thom's Lemma, but we will not compare these in detail. In these approaches, the abstract algebraic extension, as an unordered field, is modeled as $\mathbf{Q}[\alpha]/(f(\alpha))$, with algebra and equality testing done in this algebraic domain, and the choice of "which root" is only important when the ordering properties of the field are invoked. Nevertheless, a purely algebraic approach has to be blended with some numeric information in order to model the user's mental image of "*this* root of $f(x) := x^4 - 10x^2 + 1$".

The Numeric Approach

An alternative approach is to model the field as a subset of $\mathbf{R}$, with α being represented by a "sufficiently accurate" numerical approximation. Possible models include the use of continued fractions and B-adic approximations for various bases B. This seems to model the ordered field structure correctly, and in one sense it does. However, in computer algebra, we also take for granted the notion of equality, and that is what this approach does not do. If we try to compare $\sqrt{2}^2$ with 2, we will find that, no matter how much precision we call for, the answer is always "uncertain".

The solution to this problem seems to be that an element of $\mathbf{Q}(\alpha) \subset \mathbf{R}$ must be modeled with both its numerical properties and its algebraic properties. One way of doing this is to combine a numerical approximation with a minimal, or at least defining, polynomial, so an algebraic α is represented as $\langle \overline{\alpha}, f(\alpha) \rangle$, where $\overline{\alpha}$ is the approximation-producing equivalent of α. In the example in the previous paragraph, $\sqrt{2}$ would be represented by $\langle \overline{\sqrt{2}}, x^2 - 2 \rangle$, $\sqrt{2}^2$ by $\langle \overline{\sqrt{2}^2}, x^4 - 8x^2 + 16 \rangle$, and $\sqrt{2}^2 - 2$ by $\langle \overline{\sqrt{2}^2 - 2}, x^4 + 8x^3 + 16x^2 \rangle$. This last polynomial admits $x = 0$ as a root, and numerical evaluation of $\overline{\sqrt{2}^2 - 2}$, combined with Mahler's lower bound on non-zero roots of a polynomial will show that this *has* to be zero.

However we do it, we have to blend the numeric approach with some algebraic information in order to model the user's mental image of "this number $3.1462\ldots$ *is* also a root of $f(x) := x^4 - 10x^2 + 1$".

How to Model a Number Field

While we mention minimal polynomials above, this is in practice a very inefficient means of computing, and one should certainly compute in a tower of extensions. We should note this as a typical example of mathematical cleanliness ("without loss of generality, we may assume that $\alpha_1, \ldots, \alpha_k$ all lie in a given field $\mathbf{Q}(\beta)$") versus computational efficiency. There are several reasons for the practical use of towers.

1. Primitive elements tend to introduce a lack of sparsity. For example, the field $\mathbf{Q}(\sqrt{2}, \sqrt{3}, \sqrt{5}, \sqrt{7})$ has a primitive element (viz. $\beta := \sqrt{2} + \sqrt{3} + \sqrt{5} + \sqrt{7}$) whose minimal polynomial is $\beta^{16} - 136\beta^{14} + \cdots - 5596840\beta^2 + 46225$.
2. As one can see from the example above, they also tend to induce coefficient growth.
3. Primitive elements tend to place one in the "most general setting": one could be subtracting $\sqrt{2}$ from itself, but because one had mentioned $\sqrt{3}$, one was dealing in a more complex world, where the generator was $\alpha : \alpha^4 - 10\alpha^2 + 1 = 0$, and to check that $\sqrt{2} - \sqrt{2} = 0$, one has to satisfy oneself that one has the root $\beta = 0$ of $\beta^{16} - 32 * \beta^{14} + \cdots + 4096\beta^8$ rather than of $\beta^4 - 8\beta^2$.
4. From the point of view of programming a computer algebra system, the field is not generally given in advance: the user can

introduce a new algebraic element, i.e. grow the field of definition, at any time. This requires an elaborate data structure to convert elements on-the-fly from the old presentation to the new one, even though that conversion may not be necessary.

Local or global towers

Line	Code	Tower
1	`a:=sqrt(2)`	$\sqrt{2}$
2	`b:=a+sqrt(3)`	$\sqrt{2}, \sqrt{3}$
3	`c:=a+sqrt(5)`	`???`

If `???` is $\sqrt{2}, \sqrt{3}, \sqrt{5}$, then we have adopted the "global tower" approach, using the last tower even though $\sqrt{3}$ is irrelevant to `c`. We are then working in a larger tower than is necessary. For example, if line 2 was a typographical error, and line 3 were `b:=a+sqrt(5)`, from then on we would be working in a field of twice the degree necessary — an expensive price to pay for a simple typing error.

Local Towers

If `???` is $\sqrt{2}, \sqrt{5}$, then we have adopted the "local tower" approach, using the tower of the input (merger of the towers of the inputs, in general) to build on. This leads to much smaller towers, and avoids the "typing error penalty" of the other approach. However, the operation of merging towers can prove interesting.

Line	Code	Tower
1	`a:=sqrt(2)`	$\sqrt{2}$
2	`b:=a+sqrt(3)`	$\sqrt{2}, \sqrt{3}$
3	`c:=a+sqrt(6)`	$\sqrt{2}, \sqrt{6}$
4	`d:=b+c`	`merge(`$\sqrt{2}, \sqrt{3}$`,`$\sqrt{2}, \sqrt{6}$`)`

It is clear that, *algebraically*, the merged tower is $\sqrt{2}, \sqrt{3}$ (which is isomorphic to $\sqrt{2}, \sqrt{6}$), but we have no idea whether $\sqrt{6}$, in this tower, is $\sqrt{2}\sqrt{3}$ or $-\sqrt{2}\sqrt{3}$.

In line with our thesis, that one has to know the embedding into $\mathbf{R}$ as well as the algebraic information, the code fragment should be written as below, where `d` becomes $2\sqrt{2} + \sqrt{3} + \sqrt{2}\sqrt{3}$.

Line	Code	Tower
1	`a:=[sqrt(2),[1,2]]`	$\sqrt{2} \in [1,2]$
2	`b:=a+[sqrt(3),[1,2]]`	$\sqrt{2} \in [1,2], \sqrt{3} \in [1,2]$
3	`c:=a+[sqrt(6),[2,3]]`	$\sqrt{2} \in [1,2], \sqrt{6} \in [2,3]$
4	`d:=b+c`	$\sqrt{2} \in [1,2], \sqrt{3} \in [1,2]$

It should be noted that it is also possible to have a "lazy" tower merging process, in which one can build an unreduced tower.

Line	Code	Tower
1	`a:=[sqrt(2),[1,2]]`	$\sqrt{2} \in [1,2]$
2	`b:=a+[sqrt(3),[1,2]]`	$\sqrt{2} \in [1,2], \sqrt{3} \in [1,2]$
3	`c:=a+[sqrt(6),[2,3]]`	$\sqrt{2} \in [1,2], \sqrt{6} \in [2,3]$
4	`d:=b+c`	$\sqrt{2} \in [1,2], \sqrt{3} \in [1,2], \sqrt{6} \in [2,3]$
5	`if c-a=a*(b-a) ...`	$\sqrt{2} \in [1,2], \sqrt{3} \in [1,2], \sqrt{6} \in [2,3]$

In this case, `d` becomes $2\sqrt{2}+\sqrt{3}+\sqrt{6}$, and it is only at line 5 that we discover that $\sqrt{6} = \sqrt{2}\sqrt{3}$.

- In the algebraic approach, testing whether a number is zero is tantamount to testing whether it is a zero divisor, and, appliying the extended Euclidean algorithm to inverting $\gamma - \alpha\beta$ subject to $\gamma^2 - 6$, $\beta^2 - 3$ and $\alpha^2 - 2$, we soon find that $\gamma\beta\alpha - 6$ is a zero-divisor, with cofactor $\gamma\beta\alpha + 6$. Which of these is "right" depends on the numeric information for α, β and γ.
- In the numerical approach, since numerical computation with the approximations to `c-a` and `a*(b-a)` does not decide the equality after a suitable point, we switch to a symbolic test of equality via Mahler's inequality. At this point, we can change the minimal polynomial for $\sqrt{6}$, from $z^2 - 6$ to $z - \sqrt{2}\sqrt{3}$. Note that, in this case, had we been asked the same question, but with $\sqrt{6} \in [-3,-2]$, we would not have bothered to construct the (reducible) defining polynomial (which is of course the same), since numerical tests would have shown us that $\sqrt{6} - \sqrt{2}\sqrt{3} \in [-3,-7]$, and is therefore nonzero.

The principal advantage of the "lazy tower" method is that one avoids all factorisation of polynomials over algebraic number fields. Instead, a factorisation is only detected when it is thrust in our face by the user's computations. Although theoretically in polynomial time, factorisation of polynomials over algebraic number fields is very expensive. The drawback is that, if, say, one starts with $\mathbf{Q}(\sqrt{2},\sqrt{8})$, then, until the redundancy is detected, one is working in extensions of twice the necessary degree. We should note that there is no need to "split" and pursue multiple computations, since the numeric information tells us which branch to pursue.

Moral. The moral of this section is two-fold. The first is that it is necessary, to capture the users' requirements, to model algebraic numbers both as elements of an abstract algebraic extension, and with an embedding into $\mathbf{R}$ (or $\mathbf{C}$). Secondarily, we have learnt that the simplest situation for abstract mathematics (a primitive element) is neither efficient nor suitable for practical computation.

Elementary Transcendental Functions

These functions are normally considered to be exp and its inverse ln, the six trigonometric functions and their inverses, and the six hyperbolic functions and their inverses. For the purposes of this paper, we will class exp, the six trigonometric functions and the six hyperbolic functions together as the *forward functions*, and the remainder as the *inverse functions*. The forward functions are relatively straight-forward: they are many–one, continuous functions $\mathbf{C} \to \mathbf{C}$ (and restrict to similar functions $\mathbf{R} \to \mathbf{R}$). These functions, or at least most of them, are built into computer algebra systems, as well as the numeric languages for which computer algebra systems often generate code, and are used in a variety of ways.

Calculus We expect to integrate, differentiate, and solve differential equations with expressions involving these functions. In particular, we expect $\int \frac{1}{x}\,\mathrm{d}x$ to return $\ln x$ (or possibly $\ln|x|$).

"Simplification" We expect to see expressions involving the elementary functions "simplified" — whatever that might mean, and simplification is, in general, in the eye of the beholder. For example, we might expect $\exp(a)\exp(b) \to \exp(a+b)$, and many people would expect its converse $\ln(a) + \ln(b) \to \ln(ab)$ to happen, even though the branch cuts mean that this is not true over $\mathbf{C}$.

Symbolic Evaluation We expect to see $\ln 1 \to 0$ and $\sin\frac{\pi}{2} \to 1$. Whether we expect to see $\tan\frac{\pi}{2} \to \infty$ or $+\infty$ is a moot point.

Numeric Evaluation We expect to be able to evaluate these functions at floating-point values (in $\mathbf{R}$ or $\mathbf{C}$) to yield

floating-point results.

Furthermore, the user wishes to perform any or all of these operation on an expression built from elementary functions, and know that he is getting consistent results.

What Need a System Know

Indefinite Calculus By this phrase, we mean differentiation, indefinite integration and indefinite solution of ordinary differential equations. Here all that is required is a knowledge of the differential-algebraic properties of the functions, e.g. $(\exp f(x))' = f'(x)\,(\exp f(x))$ or $(\ln f(x))' = \frac{f'(x)}{f(x)}$. From this point of view, $\int \exp x = \exp x$, and whether $\exp(x)$ is e^x or $2e^x$ is irrelevant, and where the branch cuts of any actual function $\ln : \mathbf{C} \to \mathbf{C}$ lie even more so. Indeed, in differential algebra, there is no concept of "evaluating" x at all: x is merely the base symbol whose derivative is 1.

Definite Calculus By this phrase, we mean definite integration and definite solution of ordinary differential equations. Sometimes these are solved numerically, in which case we are really back to numeric evaluation, but more often they are solved by indefinite methods followed by instantiation. At this

stage the actual meaning of the functions, as $\mathbf{R} \to \mathbf{R}$ (or possibly $\mathbf{C} \to \mathbf{C}$) matters: after all $\int_0^1 e^x\,\mathrm{d}x$ is suddenly very different from $\int_0^1 2e^x\,\mathrm{d}x$. Furthermore branch cuts in the integrand and the (indefinite) integral become very important. In certain cases the Lazard/Rioboo/Trager formulation of the integral can help, but in general there is no substitute for a complete analysis of these branch cuts.

A simple example is given by the following integral in the complex plane: $\int_P^{\overline{P}} \frac{1}{z}\,\mathrm{d}z$, where P is the point on the unit circle $\frac{-1+i}{\sqrt{2}}$. An indefinite integration followed by substitution gives $\ln\overline{P} - \ln P$, which is indeed what Maple 8 returns. A naïve evaluation of this would give $\frac{-3}{4}\pi i - \frac{3}{4}\pi i = -\frac{3}{2}\pi i$. However, standard complex variable theory tells us that integrating along the straight line between P and $\overline{P}$ is the same as integrating along the arc, when we are integrating something of absolue value 1 along an arc of length $\frac{\pi}{2}$, so the answer is clearly $\frac{\pi}{2}i$. So we have two answers: $-\frac{3}{2}\pi i$ and $\frac{\pi}{2}i$. The second is correct, the first ignored the fact that the standard branch cut for logarithm, the negative real axis, intersects the path of integration, so that our integral, as expressed by the standard form of ln, is not continuous. The resolution is to choose a different branch cut, and therefore, if $\ln P = \frac{3}{4}\pi i$, $\ln\overline{P}$ has to be $\frac{5}{4}\pi i$, and the correct solution is restored.

Simplification/Symbolic Evaluation Purely algebraic simplification, as in Maple's `simplify(...,symbolic)`, can be achieved from the differential-algebraic definitions and appropriate values of constants. For example, consider $\ln f + \ln g = \ln fg$. Differentiating this equation gives $\frac{f'}{f} + \frac{g'}{g}$ on the left-hand side, and $\frac{(fg)'}{fg}$ on the right-hand side, and these are clearly equal. Hence we can deduce that $\ln f + \ln g = c + \ln fg$ for some constant c. Unfortunately, the meaning of the word "constant" here is "something that differentiates to zero" rather than "something that takes on the same value as the variables range through $\mathbf{C}$". The value of this "constant" is given by equation (3), and depends on the imaginary parts of $\ln f$ and $\ln g$.

However, this sort of simplification says very little about the actual simplification rules as applied to functions $\mathbf{C} \to \mathbf{C}$ (or $\mathbf{R} \to \mathbf{R}$). There is a close relation between precisely what definitions are adopted for the elementary functions, and precisely what simplifications are valid.

- With the standard definitions of
$$\arctan(z) = \frac{1}{2i}\left(\ln(1+iz) - \ln(1-iz)\right), \qquad (1)$$
and, as functions $\mathbf{C} \to \mathbf{C}$, $\arcsin(z) \neq \arctan\frac{z}{\sqrt{1-z^2}}$, since they differ on the branch cuts $(-\infty,-1)\cup(1,\infty)$.
- Conversely, for Derive's definition of arctan, $\arcsin(z) = \underbrace{\arctan}_{\text{Derive}}\frac{z}{\sqrt{1-z^2}}$, and $\underbrace{\arctan}_{\text{Derive}}(z) \neq \frac{1}{2i}\left(\ln(1+iz) - \ln(1-iz)\right)$ since they differ on the branch cuts. What is true is that

$$\underbrace{\arctan}_{\text{Derive}}(z) = \overline{\frac{1}{2i}\left(\ln(1+i\overline{z}) - \ln(1-i\overline{z})\right)}.$$

- Of course, arctan and $\underbrace{\arctan}_{\text{Derive}}$ agree on $[-1,1]$, i.e. as (partial) functions $\mathbf{R} \to \mathbf{R}$.

Numeric Evaluation Here again, the numeric value is critically dependent on the branch cuts chosen. A further problem, intrinsic to floating-point, is that numeric evaluation near branch cuts is inevitably unstable: a minor change in real or imaginary part can push the problem the wrong side of the cut, and give completely the "wrong" answer. The risk is reduced for elementary functions, since the branch cuts for these lie along the axes (and therefore the "signed zeros" approach works for these branch cuts), but not eliminated.

Hence computer algebra systems need to know:

1. differential-algebraic information;
2. abstract simplification information (including symbolic evaluation);
3. branch cut information;
4. numerical evaluation information.

In theory, all four should be consistent, but a look at the last two lines of the next table shows that it is perfectly possible for them not to be.

Item (1) is easily encoded inside the calculus packages or equivalent, and similarly item (4) is easily encoded in the numerical evaluation package.

Different values of $\operatorname{arccot}(-1)$

Source	Detail	$\operatorname{arccot}(-1)$	Comments
AS	1st printing	$3\pi/4$	inconsistent
AS	9th printing	$-\pi/4$	
GR	5th edition	?	inconsistent
CRC	30th edition	$3\pi/4$	inconsistent
Maple	V release 5	$3\pi/4$	
Axiom	2.1	$3\pi/4$	
Mathematica		$-\pi/4$	
Maxima	5.5	$-\pi/4$	
Reduce	3.4.1	$-\pi/4$	in floating point
Matlab	5.3.0	$-\pi/4$	in floating point
Matlab	5.3.0	$3\pi/4$	symbolic toolbox

The note "inconsistent" means that, although the source quotes, or clearly lets be inferred, a value for $\operatorname{arccot}(-1)$, there are enough inconsistencies in the definition of arccot that one could infer a different value. For GR, $3\pi/4$ and $-\pi/4$ are equally inferrable.

Simplification and Branch Cuts

Items (2–3) are more complicated. It is not even clear what would be meant by a formal encoding of "branch cut information" — however, it should include both the location of the branch cut and the values that the function takes on the branch cut. For ln, AS resorts to a mixture of words (for the former) and the inequality $-\pi < \Im \ln z \leq \pi$ (for the latter). For elementary functions, two solutions have recently been proposed.

- CJ introduces the *unwinding number* $\mathcal{K}$, defined by

$$\mathcal{K}(z) = \frac{z - \ln \exp z}{2\pi i} = \left\lceil \frac{\Im z - \pi}{2\pi} \right\rceil \in \mathbf{Z}. \tag{2}$$

  The branch cut information associated with each simplification can then be encoded in terms of additional unwinding number terms, as in:

$$\ln(z_1 z_2) = \ln z_1 + \ln z_2 - 2\pi i \mathcal{K}(\ln z_1 + \ln z_2); \tag{3}$$

$$\arcsin z = \arctan \frac{z}{\sqrt{1 - z^2}} + \pi\mathcal{K}(-\ln(1 + z)) - \pi\mathcal{K}(-\ln(1 - z)). \tag{4}$$

One drawback of this is that the notation is unfamiliar to most users, and is not in calculus texts, so a system which used it internally would probably have to convert it into a more user-friendly form on output — quite a challenge in practice. One advantage that such a system might have is that it restricts comparatively easily to $\mathbf{R}$. For example, in equation(3), if $\ln z_1$ and $\ln z_2$ are real, then $\mathcal{K}(\ln z_1 + \ln z_2)$ is automatically zero. It is also possible (at least for a human) to tell from the form of the $\mathcal{K}$ terms whether they are ruling out the simple identity for a whole region (as in equation (3)), or just specific branch cuts (as in equation (4)).

- CDJW proposes to encode all elementary functions in terms of exp and ln. If we treat the branch cut for ln as a built-in primitive, then all other elementary branch cuts can be expressed in terms of this one. One major drawback to this scheme by itself is that the simplifier would then have to do a lot of reasoning to derive, and ensure the appropriate correction terms in, simplification rules. While it is possible to derive equations such as (4) quasi-automatically, the process is complicated and can only be guaranteed to work in a restricted set of cases.

  However, we could still take this approach as a base, in that the primitive would indeed be the branch cut for ln, and any added rules, whether encoded via $\mathcal{K}$ or in other ways, would be verifiable (semi-manually?) to be consistent with this primitive.

Moral. The moral of this section is that a full interpretation of a function such as arctan involves several different kinds of

information, differential-algebraic, simplification, branch cuts and numeric, which have to be present consistently in a system for it to be able to model a user's full range of expectations about that function. It should also be noted that, while "abstract" differential-algebraic simplification can be achieved, in terms of deciding whether two expressions are "equal up to constants", the problem for actual functions $\mathbf{R} \to \mathbf{R}$ or $\mathbf{C} \to \mathbf{C}$ is much harder and in general undecidable.

R or C?

Aslaksen asks: "Can your computer do complex analysis?", to which the answer generally is "not very well". An equally relevant question would be "Can your computer do real analysis, or does it insist on (trying to do) complex analysis?". In this section, we explore the difference.

The issue

We have already seen that $\ln(a) + \ln(b) \to \ln(ab)$ is true over $\mathbf{R}$ (by which we mean that not only $a, b \in \mathbf{R}$ but also $\ln(a), \ln(b) \in \mathbf{R}$) but not over $\mathbf{C}$. In general, most computer algebra systems:

- are unclear[a] (certainly to the user, and all too often in the minds of the authors) whether they are working in $\mathbf{R}$ or $\mathbf{C}$;
- provide few or no means for controlling this behaviour. The Maple `assume` facility, while very powerful, does not really control this — it can control, *inter alia*, whether particular variables are to be interpreted in $\mathbf{R}$ or $\mathbf{C}$, as in `assume(a,real)`, but this is insufficient, since $\ln(a)$ might still not be real (indeed $a = b = -1$ is a counter-example to

[a] An honest assessment is given in the Maxima 5.5 documentation: "Variable: **DOMAIN**. Default: [REAL] — if set to COMPLEX, SQRT(X2̂) will remain SQRT(X2̂) instead of returning ABS(X). The notion of a "domain" of simplification is still in its infancy, and controls little more than this at the moment."

$\ln(a) + \ln(b) \to \ln(ab)$). One would need to add that a and b were positive. In a more complicated case, it might be very difficult to work out the assumptions on the input variables that would guarantee that all the intermediate functions were real, and there is still no guarantee that Maple would make those deductions as well.

A common argument is that, since $\mathbf{C}$ is closed under the elementary functions, it is the "natural" domain, and that to work in $\mathbf{R}$ would leave systems open to variants on the

$$1 = \sqrt{1} = \sqrt{(-1)\cdot(1)} = \sqrt{-1}\cdot\sqrt{-1} = \sqrt{-1}^2 = -1$$

paradox. This is a reasonable argument for making $\mathbf{C}$ the default, which we do not challenge, but seems like a poor argument for making the semantics of $\mathbf{R}$ inaccessible, or at least very hard to access.

But $\mathbf{R} \subset \mathbf{C}$

It is often assumed that, since $\mathbf{R} \subset \mathbf{C}$, this doesn't really matter, since everything that is true in $\mathbf{C}$ will still be true in $\mathbf{R}$. For identities, this is indeed true: since $\exp(a)\exp(b) \to \exp(a+b)$ is true over $\mathbf{C}$, it is true over $\mathbf{R}$. However, this piece of glib reasoning misses the following points.

1. It may give unexpected results to the user: having seen Maple, say, simplify $\ln 2 + \ln 3 \to \ln 6$ (which Maple can do since it knows that the lns involved are positive, so the branch cuts don't interfere), the user is frustrated by the fact that $\ln(a) + \ln(b) \not\to \ln(ab)$.
2. Although $\mathbf{R} \subset \mathbf{C}$, the same is not true of their topological completions: $\mathbf{R} \cup \{-\infty, +\infty\} \not\subset \mathbf{C} \cup \{\infty\}$. For example, over $\mathbf{R}$ one can define tan to be a bijection between $[-\frac{\pi}{2}, \frac{\pi}{2}]$ and $[-\infty, +\infty]$, but over $\mathbf{C}$, $\tan(-\frac{\pi}{2}) = \tan(\frac{\pi}{2}) = \infty$.

3. The "natural" values of some intrinsically multivalued functions may be different. For example, over $\mathbf{C}$ it is natural to choose $\sqrt[3]{-1}$ to be $\frac{1-i\sqrt{3}}{2} = \exp(\frac{1}{3}\ln(-1))$, but over $\mathbf{R}$ it is natural to choose it to be -1. Now, Maple does provide the function `surd`: $\mathbf{R} \times \mathbf{N} \to \mathbf{R} \cup$"square root of negative" to solve this problem, but nevertheless the issue crops up about once a month in the Maple news group.

4. Any statement that requires an existence argument, e.g. a proof that $f \neq g$ via Maple's `testeq` [17], which relies on finding a z such that $f(z) \neq g(z)$, may not be valid over $\mathbf{R}$, since the necessary z may not be in $\mathbf{R}$. For example, $\arcsin(z) \neq \arctan\frac{z}{\sqrt{1-z^2}}$ as functions $\mathbf{C} \to \mathbf{C}$, since, for example, $\arcsin(2) \approx \frac{\pi}{2} - 1.317i$, whereas $\arctan\frac{2}{\sqrt{-3}} \approx -\frac{\pi}{2} - 1.317i$. However, as (partial) functions $\mathbf{R} \to \mathbf{R}$, they do agree, since all the counter-examples, although real values of z, involve complex values of $\arcsin z$.

The Users' Reality

It is the author's experience, both first-hand and from reading the Maple user group and news group, that the majority of users (if they are concerned at all about the field of definition) are concerned with $\mathbf{R}$ rather than $\mathbf{C}$. The conclusion from Maple users could be considered biased, since those who are happy with $\mathbf{C}$, the semantics of Maple, may well not comment; however the conclusion from the author's own experience is probably free from such biases. In particular, when working with the GENTRAN FORTRAN generation package in Reduce, it was obvious (and conversations with the author confirmed this) that the complex part of this was much less heavily exercised than the real.

The reason for this is obvious. The vast majority of physical equations involve real variables. Indeed many physical quantities (masses, resistances, concentrations etc.) are constrained to be non-negative, and the Maple user group is often advising people on

how to ensure this.

Those users who do venture into complex variables, e.g. for the conformal mapping solution of 2-dimensional fluid dynamics problems are often unaware of the pitfalls of complex simplification, and wish to use the rules inherited from the reals. Algebra systems often stop them, but do not explain why they are doing so.

The following excerpt from a posting by Fateman in the Maple news group in 1993 expresses the view that the majority of users of the "polynomial-type" computer algebra systems today are hoping that they are using "computer mathematics" systems.

> I believe that for many readers of this newsgroup, those who are interested in the problems of general algorithmic manipulation of mathematical objects in CAS, the point is really this:
>
> You cannot tell the sign of RootOf(x^ 2-1) (which could be either +1 or -1). As a consequence, an enormous range of computations that you could do with the specific number 1 or the specific number -1 are excluded.
>
> When an algebraist claims "any root will do" he/she has focussed on a set of operations that represent only a portion of what a CAS can do. What is sin(rootof(x^ 2-1))? The algebraist might say, oh, I didn't mean you could do THAT!
>
> While I think I have some appreciation for the importance

of algebra as the basis for the CAS of today, most people "out there" are probably NOT asking algebraic questions. Most people who wish to find the roots of a univariate polynomial are generally NOT seeking algebraic answers. They are seeking a set of complex floating-point numbers. If they are offered algebraic factoring programs instead of root-finding programs, they will be wasting time, and will be unhappy with the results.

Of course, some people ARE interested in the algebraic questions, and it is nice to be able to answer them, too. However, it may be a good idea to address those issues in a different fashion. Some people advocate building a separate CAS in the service of algebraists, number theorists, etc. Although I think it is good to have algebraic facilities available in a general system – to use them when needed, I think it is a mistake to view algebraic answers as "the most correct" or when they are useless for the purposes intended for the answers.

Conclusions

We can make various deductions about the inadequacy of current computer algebra systems in meeting users' requirements for computer mathematics.

- Even in the case of algebraic numbers, $\alpha \mid \alpha^2 = 2$ does not fully capture the user's intuitive $\sqrt{2}$. One needs some way of combining the algebraic information with numerical information $\alpha \approx 1.4142\ldots$. We point out a couple of solutions which have been discussed, but neither have made their way into main-stream computer algebra systems yet. Specialised software, e.g. to do Cylindrical Algebraic Decomposition has incorporated these ideas for years, typically in the $\alpha \mid \alpha^2 = 2 \wedge \alpha \in [1, 2]$ or "$\alpha \mid \alpha^2 = 2$ and α is the second root (from $-\infty$) of this equation" formulations.
- Elementary functions have many uses in mathematics, and

many ways to manipulate them in algebra systems. Systems are not necessarily consistent (see Appendix) in this, and the information required to define both the algebraic and analytic behaviours of these functions, in a consistent way, is not well-defined.

- Systems are much better at not applying a simplification (for good reason) than they are in explaining *why* it was not applied.
- Computer algebra systems have not come to terms with the $\mathbf{R}/\mathbf{C}$ dichotomy.

Questions for Mathematicians

1. Is it possible to produce a theory of the elementary functions *as they are in computation*: single-valued functions $\mathbf{C} \setminus \{\text{singularities}\} \to \mathbf{C}$? The following standard responses are not necessarily adequate, for the reasons given.
   - Many analysts would urge one to "consider the Riemann surface". Unfortunately, in the case of $\arctan x + \arctan y \stackrel{?}{=} \arctan\left(\frac{x+y}{1-xy}\right)$, $\arctan x$ is defined on one surface, $\arctan y$ on a second, and $\arctan\left(\frac{x+y}{1-xy}\right)$ on a third. It is not clear how "considering the Riemann surface" solves practical questions such as this identity.
   - In a similar vein, one is urged to consider multivalued definitions (denoted Ln etc.) for the inverses of exp etc. Simple examples with Ln and Arctan look fairly convincing. However, there are difficulties in practice, e.g.

$\mathrm{Arcsin}(x) - \mathrm{Arcsin}(x) =$

$\{2n\pi \mid n \in \mathbf{Z}\} \cup \{2\arcsin(x) - \pi + 2n\pi \mid n \in \mathbf{Z}\} \cup \{\pi - 2\arcsin(x) + 2n\pi \mid n \in \mathbf{Z}\},$

which depends on x, so the algebra of multivalued functions is non-trivial. Several useful identities also become more difficult, e.g. $\arcsin z \stackrel{?}{=} \arctan \frac{z}{\sqrt{1-z^2}}$ (true except on the branch cuts) becomes the more complicated

$$\mathrm{Arcsin}(z) \cup \mathrm{Arcsin}(-z) = \mathrm{Arctan}\left(\frac{z}{\mathrm{Sqrt}(1-z^2)}\right).$$

2. Is it possible to produce a "calculus of branches"? This might be more promising, since it might extend to non-elementary functions, for which there may be no simple explanation of the branch structure in the form $+2n\pi$, as in the case of the Lambert W function.

Questions for Computer Algebra System Designers

1. Should there be (and we have argued for) an explicit choice between the semantics of $\mathbf{R}$ and $\mathbf{C}$? This could either be global, e.g. a massive reworking of Maxima's `DOMAIN` switch, or local, e.g. a command such as `realsimplify`.
2. Is it possible to explain to users why simplifications *don't* happen?
3. Could some interactivity be built in? This is trickier than it looks: Macsyma used to be notorious for asking questions of the form "is `very large expression` positive, negative or zero?".
4. Below we observe that a command like `LOGCONTRACT` does not keep any record of what is being assumed in the process, e.g. $\ln a + \ln b \to \ln(ab)$ assumes that $-\pi < \Im(\ln a + \ln b) \leq \pi$. Maybe such a record should be kept, and either displayed to the user, or maybe even fed to some kind of verifier. Such

verification would need to take account of any `assume` facility [37]. Also, the verifier would need to have the same meanings of the elementary functions as the algebra system: not entirely obvious in view of the table in the appendix.

5. Generalising the above idea, is the manipulation of these functions best left to an algebra system, or is interaction between an algebra system and theorem provers required?

Questions for User Education

1. It is quite clear from the questions asked in, e.g., the Maple news group, that many users are ignorant about, or puzzled by aspects of, the analysis of $\mathbf{C}$. We have argued above that some of these problems could be remedied by explicitly offering $\mathbf{R}$ instead. However, this is not a complete solution, and one has to ask how one explains the issues to the user, who is currently faced with a choice between:
   - A refusal by the system to "do the right thing", e.g. $\ln a + \ln b \not\to \ln(ab)$, with no explanation attached;
   - A command like `LOGCONTRACT`, which will do what the user believes to be the right thing, but with no explanation of what the user is assuming in the process — the verification community would think of these as the user's proof obligations.

2. How does an algebra system explain to the user "valid except on these branch cuts" or "valid outside this region"? We have already said that an explanation in terms of unwinding numbers is unlikely to make sense to most users in the current state of complex variable education. In one complex variable, one can imagine the system drawing a diagram (though issues of scale would be interesting), but in more than one variable it is hard to see how to do it.

References

[1] Abbott,J.A., Bradford,R.J. & Davenport,J.H., Factorisation of Polynomials: Old Ideas and Recent Results. Proc. "Trends in Computer Algebra" (ed. R. Janssen), Springer Lecture Notes in Computer Science 296 (Springer-Verlag, Berlin-Heidelberg-New York-Tokyo, 1988), pp. 81–91. MR 89f:12001.

[2] Abramowitz,M. & Stegun,I., Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables. US Government Printing Office, 1964. 10th Printing December 1972.

[3] Aslaksen,H., Can your computer do complex analysis?. In: *Computer Algebra Systems: A Practical Guide* (M. Wester ed.), John Wiley, 1999. `http://www.math.nus.edu.sg/aslaksen/helmerpub.shtml`.

[4] Boehm,H. & Cartwright,R., Exact Real Arithmetic: Formulating Real Numbers as Functions. Chapter 3 of D.A. Turner (ed.), *Research Topics in Functional Programming*, Addison-Wesley, 1990, pp. 43–64.

[5] Bradford,R.J., Corless,R.M., Davenport,J.H., Jeffrey,D.J. & Watt,S.M., Reasoning about the Elementary Functions of Complex Analysis. *Annals of Mathematics and Artificial Intelligence* **36** (2002) pp. 303–318.

[6] Bradford,R.J. & Davenport,J.H., Towards Better Simplification of Elementary Functions. Proc. ISSAC 2002 (ed. T. Mora), ACM Press, New York, 2002, pp. 15–22.

[7] Collins,G.E., Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition. Proc. 2nd. GI Conference Automata Theory & Formal Languages (Springer Lecture Notes in Computer Science 33) pp. 134–183. MR **55** (1978) #771.

[8] Corless,R.M., Davenport,J.H., Jeffrey,D.J. & Watt,S.M., "According to Abramowitz and Stegun". *SIGSAM Bulletin* **34** (2000) 2, pp. 58–65.

[9] Corless,R.M. & Jeffrey,D.J., The Unwinding Number. *SIGSAM Bulletin* **30** (1996) 2, pp. 28–35.

[10] Coste,M. & Roy,M.F., Thom's Lemma, the Coding of Real Algebraic Numbers and the Computation of the Topology of Semi-Algebraic Sets. *J. Symbolic Comp.* **5** (1988) pp. 121–129. Also *Algorithms in Real Algebraic Geometry* (ed. D.S. Arnon and B. Buchberger), Academic Press, London, 1988.

[11] Davenport,J.H., Equality in Computer Algebra and beyond. To appear in *J. Symbolic Comp.*

[12] Delaunay, Ch., Théorie du mouvement de la lune. Extract from the Comptes Rendus de l'Académie des Sciences, Vol. LI (1860).

[13] Della Dora,J., DiCrescenzo,C. & Duval,D., About a new Method for Computing in Algebraic Number Fields. Proc. EUROCAL 85, Vol. 2 (Springer Lecture Notes in Computer Science Vol. 204, Springer-Verlag, 1985) pp. 289–290.

[14] Dicrescenzo,C. & Duval,D., Algebraic Extensions and Algebraic Closure in SCRATCHPAD II. Proc. ISSAC 1988 (ed. P. Gianni), Springer Lecture Notes in Computer Science 358, Springer-Verlag, 1989, pp. 440–446.

[15] Duval,D., Algebraic numbers: an example of dynamic evaluation. *J. Symbolic Comp.* **18** (1994) pp. 429–445.

[16] Gates,B.L., A Numeric Code Generation Facility for Reduce. Proc. SYMSAC 86, ACM, New York, 1986, pp. 94–99.

[17] Gonnet,G.H., New Results for Random Determination of Equivalence of Expressions. Proc. SYMSAC 86, ACM, New York, 1986, pp. 127–131.

[18] Gradshteyn,I.S. & Ryzhik,I.M. (ed A. Jeffrey), Table of Integrals, Series and Products. 5th ed., Academic Press, 1994.

[19] Hur,N. & Davenport,J.H., An Exact Real Arithmetic with Equality Determination. Proc. ISSAC 2000 (ed. C. Traverso), ACM, New York, 2000, pp. 169–174. MR 2002c:11171.

[20] IEEE Standard 754 for Binary Floating-Point Arithmetic. IEEE Inc., 1985.

[21] Jeffrey, D.J., Integration to obtain expressions valid on domains of maximum extent. Proc. ISSAC 1993 (ed. M. Bronstein), ACM, New York, 1993, pp. 34–41.

[22] Jeffrey,D.J., Hare,D.E.G. & Corless,R.M., Unwinding the branches of the Lambert W function. *Mathematical Scientist* **21** (1996) pp. 1–7.

[23] Kahan,W., Branch Cuts for Complex Elementary Functions. *The State of Art in Numerical Analysis* (ed. A. Iserles & M.J.D. Powell), Clarendon Press, Oxford, 1987, pp. 165–211.

[24] Lazard,D. & Rioboo,R., Integration of Rational Functions — Rational Computation of the Logarithmic Part. *J. Symbolic Comp.* **9** (1990) pp. 113–115. MR 91h:68091.

[25] Lenstra,A.K., Factoring Polynomials over Algebraic Number Fields. Proc. EUROCAL 83, Springer Lecture Notes in Computer Science 162, Springer-Verlag, 1983, pp. 245–254.

[26] Mahler,K., An Inequality for the Discriminant of a Polynomial. *Michigan Math. J.* **11** (1964) pp. 257–262.

[27] Ménissier-Morain,V., Arithmétique exacte, conception, algorithmique et performances d'une implémentation informatique en précision arbitraire. Thèse, Université Paris 7, Dec. 1994.

[28] Mulders,T., A note on subresultants and the Lazard/Rioboo/Trager formula in rational function integration. *J. Symbolic Comp.* **24** (1997) pp. 45–50. MR 98c:26001.

[29] Rich,A.D. & Stoutemyer,D.R., Capabilities of the MUMATH-79 Computer Algebra System for the INTEL-8080 Microprocessor. Proc. EUROSAM 79 (Springer Lecture Notes in Computer Science 72, Springer-Verlag, Berlin-Heidelberg-New York) pp. 241–248.

[30] Richardson,D., Some Unsolvable Problems Involving Elementary Functions of a Real Variable. *J. Symbolic Logic* **33** (1968) pp. 514–520.

[31] Rioboo,R., Real algebraic closure of an ordered field, implementation in Axiom. Proc. ISSAC 1992 (ed. P.S. Wang), ACM, New York, 1992, pp. 206–215.

[32] Rioboo,R., Towards Faster Real Algebraic Numbers. Proc. ISSAC 2002 (ed. T. Mora), ACM, New York, 2002, pp. 221–228.

[33] Risch,R.H., Algebraic Properties of the Elementary Functions of Analysis. *Amer. J. Math.* **101** (1979) pp. 743–759. MR 81b:12029.

[34] Stoutemyer,D.R., Crimes and Misdemeanors in the Computer Algebra Trade. *Notices AMS* **38** (1991) pp. 779–785.

[35] Stoutemyer,D.R., Should computer algebra programs use ln x or ln $|x|$ as their antiderivation of $1/x$. *DERIVE Newsletter* **26** (Jun 1997) pp. 3–6.

[36] Vuillemin,J.E., Exact real computer arithmetic with continued fractions. *IEEE Trans. Computing* **39** (1990) pp. 1087–1105.

[37] Weibel,T. & Gonnet,G.H., An Assume Facility for CAS with a Sample Implementation for Maple. Proc. DISCO '92 (Springer Lecture Notes in Computer Science 721, ed. J.P. Fitch), Springer, 1993, pp. 95–103.

[38] Wolfram,S., *The Mathematica Book*. Wolfram Media/C.U.P., 1999.

[39] Zwillinger,D. (ed.), *CRC Standard Mathematical Tables and Formulae*. 30th. ed., CRC Press, Boca Raton, 1996.

Solving systems of polynomials:
Theory and Practice.

James H. Davenport
Department of Computer Science
University of Bath, U.K.
J.H.Davenport@bath.ac.uk

What you already know.

- One variable, any degree
  α is a root of $f_1, \ldots, f_n \Leftrightarrow \alpha$ is a root of $\gcd(f_1, \ldots, f_n)$.
  Euclid's algorithm is generally thought of as repeated division, but think of it as repeated subtraction: if $f = a_n x^n + \cdots a_0$ and $g = b_m x^m + \cdots b_0$ $(n \geq m)$, then $\{f, g\} \to \{g, h\}$, where $h := f - \frac{a_n}{b_m} x^{n-m} g$.
  Terminates since $n + m$ decreasing.
- degree one, any number of variables
  Linear equations — Gaussian elimination.

A few disclaimers

- A vast field, and I don't pretend to know all the literature
- Terminology is not always settled
- I only consider the commutative case: see [40, 41, 49] for the non-commutative case and a vast literature (e.g. [42, 51]) for the skew (e.g. differential) case: $(\delta_x x)f = (x\delta_x + 1)f$
- Opinions vary, and implementation details really matter
- Different examples show different properties

Preserving sparsity — 1

Consider the matrix

$$\begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & \ldots & a_{1,n} \\ a_{2,1} & a_{2,2} & 0 & 0 & \ldots & 0 \\ a_{3,1} & 0 & a_{3,3} & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ldots & \vdots \\ a_{n,1} & 0 & 0 & 0 & \ldots & a_{n,n} \end{pmatrix}$$

After eliminating the first column ($O(n^2)$ work), the resulting $(n-1) \times (n-1)$ matrix is completely dense ($O((n-1)^3)$ work). So even though we had only $O(n)$ non-zeros, we still do $O(n^3)$ work.

Preserving sparsity — 2

If, however, we permute the rows, as

$$\begin{pmatrix} a_{2,1} & a_{2,2} & 0 & 0 & \dots & 0 \\ a_{3,1} & 0 & a_{3,3} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \dots & \vdots \\ a_{n,1} & 0 & 0 & 0 & \dots & a_{n,n} \\ a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} & \dots & a_{1,n} \end{pmatrix}$$

then clearing out the first column is only $O(n)$ work, and the resulting $(n-1) \times (n-1)$ matrix is of the same shape, so the total work is $O(n^2)$.

Preserving sparsity — 3

Of course, we were lucky here.

- In an exact setting, methods such as Lanczos can produce a solution in time $O(nN)$, where N is the number of non-zero entries.
- Sparse Gaussian Elimination [39] is a method which tries to reduce n without increasing N (too much).

Combinations of these methods have enabled the **Z**-factorers to solve 750000^2 matrices over $\mathbf{F}_2$, and similarly 100000^2 over $\mathbf{F}_p$ ($p \approx 10^{100}$), typically with 10–20 entries/row.

In the g.c.d. case, we naturally preserve *some* sparsity.

Z or **Q** (R or K)

The algorithms are naturally set, and described, over a field (say **Q**). However, in practice this leads to expensive fractional arithmetic.

Matrices Fraction-free elimination [9, 24] works over **Z** by cross-multiplying, but removes all "generic" factors. $O(n)$ growth in coefficients, rather than $O(2^n)$.

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \to \begin{pmatrix} a & b & c \\ 0 & ae-bd & af-cd \\ 0 & ah-gb & ai-gc \end{pmatrix} \to \begin{pmatrix} a & b & c \\ 0 & ae-bd & af-cd \\ 0 & 0 & Z \end{pmatrix}$$

where $Z = (ae-bd)(ai-gc) - (ah-gb)(af-cd)$ and a factor of a can be cancelled throughout the row. At the next stage, we seem to have total degree 6, but a factor of $(ae-bd)$ cancels, so total degree 4.

g.c.d Several algorithms.

- Integer (cross-multiply) — exponential growth;
- Primitive — remove all common **Z**-factors at each step — expensive g.c.d.s, especially in more variables;
- Sub-resultant [12, 18] — remove all generic common factors — dense case = fraction-free matrix equivalent.

The importance of order

g.c.d Implicit in Euclid is the fact that we can find the terms of highest degree.

matrices We can permute the columns (change the order of the variables), but we must do so consistently:

$$\begin{aligned} 2x + 3y + 4z &= 1 \\ 2y + 5x + 3z &= 2 \\ 2z + 7y + 3x &= 9 \end{aligned}$$

does not yield a matrix directly.

The general case: definitions.

K a field (any characteristic)

$x_1, \ldots, x_n$ commuting variables

$f_1, \ldots, f_k \in K[x_1, \ldots, x_n]$. f_i is to be thought of as the equation $f_i = 0$.

$(f_1, \ldots, f_k)$ "the **ideal** generated by $f_1, \ldots, f_k$" $= \{\sum_{i=1}^{k} g_i f_i \mid g_i \in K[x_1, \ldots, x_n]\}$. If $h \in (f_1, \ldots, f_k)$, then $h = 0$ is a consequence of $f_i = 0$

$V(I)$ "the variety of the ideal I" = set of zeros of (every polynomial in) I in $\overline{K}$.

monomial in $K[x_1, \ldots, x_n]$ is a power product $x_1^{a_1} \ldots x_n^{a_n}$: $a_i \in \mathbf{N}$, e.g. $x^2y^0z^1 = x^2z$.

The problem

Given $f_1, \ldots, f_k$, find "the solutions" to $f_1 = 0, \ldots, f_k = 0$, or some information about the solutions, e.g. how many:

none the equations are inconsistent

finite how many solutions are there?

$$\begin{aligned} a + b + c + d + e + f + g \\ ab + bc + cd + de + ef + fg + ga \\ abc + bcd + \cdots + fga + gab \\ abcd + bcde + \cdots + gabc \\ abcde + bcdef + \cdots + gabcd \\ abcdef + \cdots + gabcde \\ abcdefg - 1 \end{aligned}$$

has 924 roots [8].

infinite what dimension/how many free variables are there? These questions may not have a simple answer: $\{(x-1)(y-1), (y^2-1)\}$ has as solutions the line $y = 1$ and $(x = 1, y = -1)$: mixed dimension 1 and 0. xy has the lines $x = 0$ and $y = 0$: both variables are free, but not simultaneously.

Some questions

- Solutions in what field? For the moment, we will assume $\overline{K}$, the algebraic closure of K.
- What do we mean by "find"? In some cases obvious: $\{(x^2-1),(x-1)(y-1),(y^2-1)\}$ gives three solutions: $(x=1,y=1)$, $(x=1,y=-1)$,$(x=-1,y=1)$. But $\{(x^2-1),(y^2-1)\}$ also has $(x=-1,y=-1)$, and $\{(x-1)(y-1),(y^2-1)\}$ has the line $y=1$ and $(x=1,y=-1)$.
- Do such lower-dimensional solutions matter?
- Does the order of the variables matter?
- If there are infinitely many solutions, how do we describe them?
- What are f_i: random, structured, symmetric etc.?

Ordering the monomials of $K[x_1,\ldots,x_n]$

We will call the ordering $<$, even though it has *nothing* to do with any numeric ordering. For sanity, we want $<$ to be:

- compatible with multiplication: $\forall i : m_1 < m_2 \Rightarrow x_i m_1 < x_i m_2$;
- Well-founded: $1 = x_1^0 \ldots x_n^0$ is the least monomial.

Such an ordering is called *admissible*. We always assume this.
In these lectures, we assume as given that $x_1 > x_2 > \cdots > x_n$.

Possible orderings

lex "lexicographic": if m_i has a higher power of x_1 than m_j, then $m_i > m_j$. If equal, use powers, of x_2, then x_3 ....

deglex if m_i has a higher total degree than m_j, then $m_i > m_j$. If equal, use lexicographic to break ties.

degrevlex if m_i has a higher total degree than m_j, then $m_i > m_j$. If equal, use lexicographic to break ties in reverse, i.e. $m_i >_{\text{lex}} m_j \Rightarrow m_i <_{\text{degrevlex}} m_j$.

elimination Use one order on $x_1 \ldots, x_k$. If there's a tie, use another order on $x_{k+1}, \ldots, x_n$.

"Total degree" can be weighted.
Any ordering (including others) can be defined by a matrix of weights.

Isn't degrevlex just deglex with a reverse ordering of variables?

- deglex: $x > y > z$
  $x^3 > x^2y > x^2z > xy^2 > xyz > xz^2 > y^3 > y^2z > yz^2 > z^3$
- degrevlex: $z > y > x$
  $x^3 > x^2y > xy^2 > y^3 > x^2z > xyz > y^2z > xz^2 > yz^2 > z^3$

Notice that y^3 comes much earlier in degrevlex.

Fundamental definitions

for $p \neq 0 \in K[x_1, \ldots, x_n]$ with a fixed ordering.

$\mathrm{lm}(p)$ "leading monomial of p": the greatest monomial with a non-zero coefficient in p.

$\mathrm{lc}(p)$ "leading coefficient of p": the coefficient of $\mathrm{lm}(p)$ in p.

$\mathrm{lt}(p)$ "leading term of p": $\mathrm{lc}(p)\,\mathrm{lm}(p)$.

$\mathrm{red}(p)$ "reductum of p": $p - \mathrm{lt}(p)$.

The S-polynomial

Let f, g be non-zero in $K[x_1, \ldots, x_n]$ with a fixed ordering.
Let $l = \mathrm{lcm}(\mathrm{lm}(f), \mathrm{lm}(g))$
$\left(\mathrm{lcm}(\prod x_i^{a_i}, \prod x_i^{b_i}) = \prod x_i^{\max(a_i,b_i)}\right)$.
$S(f,g) = \frac{l}{\mathrm{lt}(f)} f - \frac{l}{\mathrm{lt}(g)} g$.
This simplifies to $S(f,g) = \frac{l}{\mathrm{lt}(f)}\,\mathrm{red}(f) - \frac{l}{\mathrm{lt}(g)}\,\mathrm{red}(g)$.
This generalises one step of Gaussian elimination:
$l = \mathrm{lm}(f) = \mathrm{lm}(g) =$ variable being eliminated.
This generalises one subtraction step of Euclid:
$l = x^{\max(\deg(f),\deg(g))}$.

Reduction

If $\mathrm{lm}(g) | \mathrm{lm}(f)$, then we can reduce f by g to get $h = f - \frac{lt(f)}{\mathrm{lt}(g)} g$.
We write $f \xrightarrow{g} h$.
If G is a set of polynomials, and there is a $g \in G$ with $\mathrm{lm}(g) | \mathrm{lm}(f)$, and $f \xrightarrow{g} h$, then we write $f \xrightarrow{G} h$.
N.B. g may not be unique in G with $\mathrm{lm}(g) | \mathrm{lm}(f)$, and therefore the definition of $\xrightarrow{G}$ may not be unique.
It is possible that $f \xrightarrow{G} h_1 \xrightarrow{G} h_2 \ldots \xrightarrow{G} h_n$, where h_n is not reducible by any element of G. Then we write $f \xrightarrow{G}{}^{*} h_n$. Again, this may not be unique, but terminates since $<$ is well-founded.
We define $\mathrm{NF}_G(f)$, the *normal form* of f with respect to G, to be the result of applying $\xrightarrow{G}{}^{*}$ to every monomial of f, not just the leading one. Again, it may not be unique.
G is said to be *auto-reduced* if $\forall g \in G : g = \mathrm{NF}_{G \setminus \{g\}}(g)$.

Gröbner bases [13, 10]

Let G be a set of polynomials generating an ideal I. G is said to be a *Gröbner basis* of I if any of the following equivalent conditions is satisfied.

1. NF_G is always unique (Church-Rosser).
2. $f \in I \Leftrightarrow NF_G(f) = 0$.
3. $\forall f, g \in G : S(f,g) \xrightarrow{G}{}^{*} 0$.
4. The ideal generated by $\{\mathrm{lm}(g) \mid g \in G\}$ is equal to the ideal generated by $\{\mathrm{lm}(f) \mid f \in I\}$.

For a fixed ordering, an ideal always has a unique auto-reduced monic Gröbner basis.

What can a Gröbner basis tell us?

- $G = \{1\} \Leftrightarrow$ no solutions.
- Every x_i occurs alone (to some power) as a leading monomial in $G \Leftrightarrow$ finitely many solutions.
- The number of monomials irreducible by $\stackrel{G}{\rightarrow}$ is the number of solutions (counted with multiplicity).

Examples

- $\{(x^2-1),(x-1)(y-1),(y^2-1)\}$
  x^2 and y^2 occur as leading monomials, so finitely many solutions.
  $1, x, y$ irredudcible monomials, therefore three solutions.
- $\{(x^2-1),(y^2-1)\}$
  xy is now irreducible, so now four solutions.
- $\{(x-1)(y-1),(y^2-1)\}$
  has no power of x as leading monomial, therefore infinitely many solutions.

And a lex basis?

Assuming only finitely many solutions, a lex basis will be of the form

$p_n(x_n)$	Determines the x_n co-ordinates of solutions
$p_{n-1,1}(x_n, x_{n-1})\ldots$	For each value of x_n
	Determine the x_{n-1} co-ordinates of solutions
$p_{n-2,1}(x_n, x_{n-1}, x_{n-2})\ldots$	For each value of (x_n, x_{n-1})
	Determine the x_{n-2} co-ordinates of solutions

Theorem [30, 36]: For given values of $x_n, \ldots, x_{k+1}$, the lowest degree (in x_k) of the $p_{k,i}$ whose leading coefficient in x_k does not vanish does all the determining of possible values of x_k.
Partial generalisation in [27].

If Gröbner bases are that useful
how do I compute one?

Recall characterisation 3: $\forall f, g \in G : S(f,g) \stackrel{G}{\longrightarrow}{}^* 0$.
$S(f,g)$, and h where $S(f,g) \stackrel{G}{\longrightarrow}{}^* h$ are both in the ideal generated by G.
So there is no harm in adding h to G, whereupon $S(f,g) \stackrel{G}{\longrightarrow}{}^* h \stackrel{h}{\rightarrow} 0$.
Of course, this means there are more S-polynomials to consider, since G has grown.

More formally: Buchberger's Algorithm [13]

Input $S \subset K[x_1, \ldots, x_n]$ with ordering
Output G a Gröbner basis for ideal generated by S
$G := S$; $m := |G|$
$P := \{(g_i, g_j) \mid 1 \leq i < j \leq m\}$
While $P \neq \emptyset$
    Pick (f, g) from P (and remove from P)
    Let $S(f, g) \xrightarrow{G}{}^{*} h$
    If $h \neq 0$ then
        $P := P \cup \{(g_i, h) \mid 1 \leq i \leq n\}$
        $G := G \cup \{h\}$; $m := m + 1$
return G

Clearly, if it returns, G is a Gröbner basis, by criterion 3. It does terminate, since, every time G grows, so does the ideal generated by $\{\mathrm{lm}(g_i)\}$, and an increasing sequence of ideals is finite (Noether).

Pretty under-specified

- Which (f, g) do I pick?
  Everyone agrees such that total degree $\mathrm{lcm}(\mathrm{lm}(f), \mathrm{lm}(g))$ is least
  Beyond this, heuristics vary, e.g.:
  - $\mathrm{lcm}(\mathrm{lm}(f), \mathrm{lm}(g))$ is least with $<$
  - "oldest" f and g (in some way)
  - "Sugar" strategies
- $\xrightarrow{G}{}^{*}$ is probably not unique — which option?
- Should we add h or $NF_G(h)$?

In practice, 99%+ $S(f, g) \xrightarrow{G}{}^{*} 0$

- If $\mathrm{lm}(f)$, $\mathrm{lm}(g)$ have no variables in common, then always $S(f, g) \xrightarrow{G}{}^{*} 0$.
- If $\exists h \in G$, $S(f, h)$ and $S(g, h)$ computed and $\mathrm{lcm}(\mathrm{lm}(f), \mathrm{lm}(h)), \mathrm{lcm}(\mathrm{lm}(g), \mathrm{lm}(h)) \mid \mathrm{lcm}(\mathrm{lm}(f), \mathrm{lm}(g))$, then always $S(f, g) \xrightarrow{G}{}^{*} 0$ [14].
- Graph-theoretic generalisation [8].

* Gröbner trace algorithms [48] — simulate mod p first.

* A different algorithm (F_5) [25].

A brief outline of Gröbner trace algorithms.

Given $H = \{f_1, \ldots, f_k\} \subset \mathbf{Z}[x_1, \ldots, x_n]$.

Choose A prime p, certainly not dividing any $\mathrm{lc}(f_i)$.

Compute A Gröbner basis G of H in $(\mathbf{Z}/p\mathbf{Z})[x_1, \ldots, x_n]$

Keeping track of which $S(g_i, g_j) \xrightarrow{G}{}^{*} h \neq 0$.

Augment H with these $S(g_i, g_j) \xrightarrow{G}{}^{*} h \neq 0$ (over $\mathbf{Z}$).

Discard those polynomials that were discarded over $\mathbf{Z}/p\mathbf{Z}$.

Check that new H' is a Gröbner basis: every $S(h_i, h_j) \xrightarrow{H'}{}^{*} 0$.

Check that every f_i reduces to zero (i.e. right ideal).

Variants on Gröbner trace algorithms.

- Work modulo two (or more) primes, but don't duplicate:

  Represent $f = \sum_i (c_i, d_i) m_i$: m_i monomials, c_i coefficient of m_i modulo p_1, d_i modulo p_2.

- Work modulo a prime and in floats.
- If the ground ring is $\mathbf{Z}[y_1, \ldots, y_m]$, work modulo p and evaluations $y_i \to v_i \in (\mathbf{Z}/p\mathbf{Z})$

  being careful that the evaluation does not destroy leading coefficients (at least).

A brief outline of F_5

Given G a Gröbner base, and g

and the history of computing G (syzygies of G)

Compute a Gröbner base of $G \cup \{g\}$

considering only S-poynomials between those that depend on g and old ones;

using the syzygies of G to reject any redundant computations.

Faugère claims that this only generates pairs which $\stackrel{G}{\longrightarrow}{}^* 0$ in exceptional circumstances.

So I need a lex GB?

- You do not know that there are only finitely many solutions (I is zero-dimensional).
- lex GB are, theoretically and practically, much more expensive to compute: lex means "decrease the degree in x_1 irrespective of the degrees of $x_2, \ldots, x_n$".

In practice (with some theory) degrevlex are the cheapest to compute. So can we just convert into lex afterwards? Main idea in [26], see also [17, 47].

The FGLM Algorithm [26]

Input G: a Gröbner basis of zero-dimensional I w.r.t. $<_{\text{in}}$.
Output G': a Gröbner basis of I w.r.t. $<_{\text{lex}}$.
$G' := \emptyset$
Until G' is a zero-dimensional GB
  Enumerate the monomials with $<_{\text{lex}}$: $m_0 = 1, m_1 = x_n, m_2 = x_n^2, \ldots$
    letting $m_i \stackrel{G}{\longrightarrow}{}^* r_i$
    When $\exists$ a linear relation among r_i
      Add corresponding relation among m_i to G'
  Continue enumerating $m_{k+1} = x_{n-1}, m_{k+2} = x_{n-1}x_n \ldots$
    (ignoring monomials reducible ($<_{\text{lex}}$) by G')

The incremental linear algebra is tricky.

Fine print of these algorithms

- We have assumed ordinary polynomial arithmetic: it may be better (fewer temporaries) to do lazy shifted subtraction [**?**].
- Memory management really matters:
  - Locality of reference
  - Cost of garbage collection
- Parallelism — difficult, but not impossible [5, 50].
- Over **Z** or **Q** — often **Q**, but ...
- Storage of monomials matters.

Gröbner Factorisation

Normally, we produce G a Gröbner base of I.
If, when we are about to do $G := G \cup \{h\}$, we spot $h = h_1 h_2$, we can do two calculations, with $G_1 := G \cup \{h_1\}$ and $G_2 := G \cup \{h_2\}$.
This produces bases G_1 and G_2 of ideals I_1 and I_2 such that $I = I_1 \cap I_2$, and so $V(I) = V(I_1) \cup V(I_2)$.
Random polynomials never factorise.
But in systems with symmetry, there is always factorisation [29].

pro lower degrees — much of G may reduce modulo h_i.

con destroys strategy — have to start again.

Very useful when relevant [21].

Example of Gröbner Factorisation

$$I = (a+b+c+d,\ ab+bc+cd+da,\ abc+bcd+cda+dab,\ abcd-1)$$

$$\begin{aligned} G \quad = \{ \quad & a+b+c+d,\ b^2+2bd+d^2,\ bc-bd+c^2\ d^4+cd-2d^2,\ bd^4 \\ & -b+d^5-d,\ c^3\ d^2+c^2d^3-c-d,\ c^2d^6-c^2\ d^2-d^4+1\} \end{aligned}$$

But $I = I_1 \cap I_2$ where $G_1 = \{a+c,\ b+d,\ cd+1\}$ and
$G_2 = \{a+c,\ b+d,\ cd-1\}$.
Which is easier to understand?

Dimension > 0

- As we have seen, can be hard to understand.
- $V(\{(x-1)(y-1), y^2-1\})$ is line $y=1$ and $(y=-1, x=1)$ — a mixed-dimensional variety.
- It is possible to use Gröbner bases to compute an equi-dimensional decomposition: $I = I_1 \cap \cdots I_n$ where $V(I_k)$ is purely of dimension k. But requires Gröbner with more than n variables.
- Even if not mixed, can be hard to describe: in (xy), it is not true simply to say "x is free and $y = 0$", or *vice versa*: both are true.
- Can solve by rotating axes, e.g. $x = \frac{\xi+\eta}{2}, y = \frac{\xi-\eta}{2}$ gives $\xi^2 - \eta^2$, where ξ is free and $\eta = \pm\xi$ (or *vice versa*).
- But this tends to make everything dense.

Triangular Sets [7, 6]

An alternative approach is provided by "triangular sets", which, roughly speaking, say

- system of equations defining non-free variables in terms of free ones
- Except on this subvariety, where
  - system of equations defining non-free variables in terms of free ones
  - Except on this subvariety, where ...

E.g. x is free and $y = 1$ except on $x = 1$ where $y = \pm 1$.

Resultants

$\mathrm{res}_x(p(x,y), q(x,y)) = r(y)$, where the roots α_i of $r(y)$ (in $\overline{K}$) are precisely those values of y such that $p(x, \alpha_i)$ and $q(x, \alpha_i)$ have common roots.
So, at least in dimension zero, this could replace Gröbner bases for two variables.
Except that $r(y)$ has repeated roots if two elements of $V((p, q))$ have the same y-value, so can be of higher degree.
Also, may not be good with sparsity (but see [4]).
Surely, by taking repeated resultants, we can generalise to more variables.

Resultants: example in three variables

$p_1 = x^2 + y^2 + z^2, p_2 = (x-1)^2 + 2*(y-1)^2 + 3*(z-2)^2, p_3 = (x-2)^2 + 3*(y-4)^2 + 6*(z-6)^2$.
$q_{12} = \mathrm{res}_x(p1, p2)$, $q_{13} = \mathrm{res}_x(p1, p3)$, $r = \mathrm{res}_y(q_{12}, q_{13})$.
$r = z^{16} - 384z^{15} + \cdots + 3482273444524437760$
Gröbner gives us
$[512x - z^4 + \cdots,\ 16y - z^2 + 48z - 238,\ z^8 - 192z^7 + \cdots + 2105746960]$.
r is $(z^8 - 192z^7 + \cdots)$ times a spurious polynomial, since q_{12} and q_{13} can have intersections which do not correspond to a common x-value.

Are resultants useless?

Not at all, merely that one has to look directly at multi-polynomial (or multi-variate) resultants. It is recursive use of resultants that is dangerous.
This is a very active area of research [3].
There is much debate on the best way to compute them, and even on what the definition should be: Sylvester, Macauley, Dixon [16], Bezout, multihomogenous [23] etc.

Complexity Theory

Has done a great deal of work in this area. However

- it is "worst-case", which seems not to reflect practice, e.g. sparse and/or structured input;
- too many results ignore coefficient growth, which in practice is often a major problem;
- Some key results, e.g. polynomial-time solving of zero-dimensional systems [31], rely on data structures such as straight-line programs, which are currently not part of practical systems (but maybe should be).

Introduction to straight-line programs

- Represent $(x+y)^9$ by:

  $l_1 := x$, $l_2 := y$, $l_3 := l_1 + l_2$, $l_4 := l_3 * l_3$, $l_5 := l_4 * l_4$, $l_6 := l_5 * l_5$, $l_7 := l_6 * l_3$.
- Addition/multiplication is $O(1)$.
- Need upper bounds on degrees.
- Leading term is hard: essentially needs full interpolation

  (important work on sparse interpolation [**?**, BOT,KL,KLL]
- May want to keep track of first few terms in usual format as well.

Suppose I don't want solutions in $\overline{K}$

I just want the solutions in K, or in a ring R which generates K.

$K = \mathbf{R}$ To be described.

$K = \mathbf{Q}$ No known algorithm: even $y^2 = x^3 + ax + b$ unknown (unless assume Birch–Swinnerton-Dyer conjecture).

$K = \mathbf{Z}$ Impossible.

$K = \mathbf{F}_p$ Easy (at least in principle) since the solutions generate a finite extension of F_p.

Solutions over $\mathbf{R}$

If there are only finitely many solutions over $\mathbf{C}$, then clearly there are only finitely many over $\mathbf{R}$. In principle easy to find.
$W(X) = (x+1)(x+2)\cdots(x+20) = x^{20} + 210x^{19} + \cdots + 20!$ has 20 real roots. However, $W(x) + 2^{-23}x^{19}$ has only 10 [54], so the number of real roots can be a very unstable property.
In a lex GB, the polynomial in x_n will probably have very high degree and large coefficients.
A real root α of a polynomial f can be represented by:

Descartes An interval enclosing α which, when transformed to $[0, \infty)$, shows there is only one root in it;

Sturm An interval enclosing α in which the Sturm sequence of f shows there is only one root;

Thom A sequence of signs of $f'(\alpha), f''(\alpha)$ etc.

A lot of research in this area [32].

If $V(I)$ is not zero-dimensional.

Let $V_{\mathbf{R}}(I)$ be the set of real zeros of every polynomial in I.

I	$\dim V(I)$	$\dim V_{\mathbf{R}}(I)$	
x^2+y^2-1	1	1	circle
x^2+y^2	1	0	two complex lines
x^2+y^2+1	1	$\emptyset$	

Furthermore, describing $V_{\mathbf{R}}$ is more difficult. In the first case x is free, but only in $[-1,1]$, and in general one needs inequalities as well as equalities: i.e. semi-algebraic sets.

Cylindrical Algebraic Decomposition [19, 20]

Input A variable ordering $x_1 > x_2 > \cdots > x_n$
Polynomials $f_1, \ldots, f_m \in \mathbf{Q}[x_1, \ldots, x_n]$

Output A Cylindical Algebraic Decomposition of $\mathbf{R}[x_1, \ldots, x_n]$
x_1-axis $= (-\infty = \alpha_0, \alpha_1) \cup \{\alpha_1\} \cup (\alpha_1, \alpha_2) \cup \{\alpha_2\} \cup \cdots \cup (\alpha_N, \infty = \alpha_{N+1})$
Such that Above each $\{\alpha_i\}$, x_2 is divided similarly
Above each (α_i, α_{i+1}) are curves $g_{i,1}(x_1, x_2), \ldots, g_{i,n_i}(x_1, x_2)$:
Well-defined (conditions on x_2), non-intersecting
Such that Above each $(\alpha_i, \beta_{i,j})$, x_3 is divided as x_1
Above each $\alpha_i \times (\beta_{i,j}, \beta_{i,j+1})$ are curves $h_{i,j,1}(\alpha_i, x_2, x_3)$ . .
Above each $(\alpha_i, \alpha_{i+1}) \times (\beta_{i,j}, \beta_{i,j+1})$ are
surfaces $\hat{h}_{i,j,1}(x_1, x_2, x_3) \ldots$
etc.
Together with a sample point in each cell

Uses

- Allows one to determine the truth/falsity of all statements of the form $\overset{\forall}{\exists} x_1 \overset{\forall}{\exists} x_2 \ldots$ boolean combination of $f_i \overset{<}{\geq} 0$, by testing truth at sample points.
- Very powerful - in many ways too powerful.
- Size can be doubly exponential in n [22], and $n = 4$ stretches most implementations.
- In particular, those cells with $f_1 = \cdots f_n = 0$ describe $V_{\mathbf{R}}$
- But it may still take a lot of work to determine the topology: one component described by many cells.

Much other research in this area

- Low-degree cases: [35, 52, 53]
- Low-dimension cases:
  - Curves: [1]
  - General Dimension 2:[2]
  - Surfaces: [28]
- Robotics: [43, 44, 45, 46]
- Complexity theory
  - Connected components in sub-exponential time: [15, 34]
  - inequalities in sub-exponential time: [33]

References

[1] Arnon,D.S. & McCallum,S., A Polynomial-time Algorithm for the Topological Type of a Real Algebraic Curve. Proc. EUROCAL 85, Vol. 2 (Springer Lecture Notes in Computer Science Vol. 204, Springer-Verlag, 1985) pp. 275–276.

[2] Alonso,M.E. & Raimondo,M., The Computation of the Topology of a Planar Semialgebraic Set. *Rend. Sem. Math. Univ. Politecnica Torino* **46** (1988) pp. 327–342.

[3] D'Andrea,C. & Dickenstein,A., Explicit formulas for the multivariate resultant *J. Pure Appl. Algebra* **164** (2001) pp. 59–86.

[4] D'Andrea,C. & Emiris,I.Z., Hybrid Sparse Resultant Matrices for Bivariate Polynomials. *J. Symbolic Comp.* **33** (2002) pp. 587–608. `http://www.idealibrary.com/links/doi/10.1006/jsco.2002.0524`

[5] Attardi,G. & Traverso,C., Strategy-accurate Parallel Buchberger Algorithms. *J. Symbolic Comp.* **21** (1996) pp. 411–425.

[6] Aubry,P. & Moreno Maza,M., Triangular Sets for Solving Polynomial Systems: A Comparison of Four Methods. *J. Symbolic Comp.* **28** (1999) pp. 125–154.

[7] Aubry,P., Lazard,D. & Moreno Maza,M., On the Theories of Triangular Sets. *J. Symbolic Comp.* **28** (1999) pp. 105–124.

[8] Backelin,J. & Fröberg,R., How we proved that there are exactly 924 cyclic 7-roots. Proc. ISSAC 1991 (ed. S.M. Watt), ACM, New York, pp. 103–111.

[9] Bareiss,E.H., Sylvester's Identity and Multistep Integer-preserving Gaussian Elimination. *Math. Comp.* **22** (1968) pp. 565–578. Zbl. 187,97.

[10] Becker,T. & Weispfeninng,V. (with H. Kredel), Groebner Bases. A Computational Approach to Commutative Algebra. Springer Verlag, Graduate Texts in Mathematics 141, 1993.

[11] Ben-Or,M. & Tiwari,P., A deterministic algorithm for sparse multivariate polynomial interpolation. Proc. 20th. Symp. Theory of Computing, ACM, 1988, pp. 301–309.

[12] Brown,W.S., On Euclid's Algorithm and the Computation of Polynomial Greatest Common Divisors. *J. ACM* **18** (1971) pp. 478–504. MR **46** (1973) #6570.

[13] Buchberger,B., Ein Algorithmus zum Auffinden des Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal. Ph.D. Thesis, Math. Inst., University of Innsbruck, 1965.

[14] Buchberger,B., A Criterion for Detecting Unnecessary Reductions in the Construction of Groebner Bases. Proc. EUROSAM 79 (Springer Lecture Notes in Computer Science 72, Springer-Verlag, Berlin-Heidelberg-New York) pp. 3–21. Zbl. 417.68029. MR 82e:14004.

[15] Canny,J.F., Grigoriev,D.Yu. & Vorobjov,N.N.,Jr., Finding Connected Components of a Semialgebraic Set in Subexponential Time. *AAECC* **2** (1992) pp. 217–238.

[16] Chtcherba,A.D. & Kapur,D., On the Efficiency and Optimality of Dixon-based Resultant Methods. Proc. ISSAC 2002 (ed. T. Mora), ACM Press, New York, 2002, pp. 29–36.

[17] Collart,S., Kalkbrener,M. & Mall,D., Converting Bases with the Gröbner Walk. *J. Symbolic Comp.* **24** (1997) pp. 465–469. MR 98f:68006.

[18] Collins,G.E., Subresultants and Reduced Polynomial Remainder Sequences. *J. ACM* **14** (1967) pp. 128–142.

[19] Collins,G.E., Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition. Proc. 2nd. GI Conference Automata Theory & Formal Languages (Springer Lecture Notes in Computer Science 33) pp. 134–183. MR **55** (1978) #771.

[20] Collins,G.E., Quantifier elimination by cylindrical algebraic decomposition — twenty years of progess. Quantifier Elimination and Cylindrical Algebraic Decomposition (ed B.F. Caviness & J.R. Johnson, Springer, Wien-New York, 1998) pp. 8–23.

[21] Davenport,J.H., Looking at a set of equations. Bath Computer Science Technical Report 87–06, October 1987. `http://staff.bath.ac.uk/masjhd/87-06.pdf`

[22] Davenport,J.H. & Heintz,J., Real Quantifier Elimination is Doubly Exponential. *J. Symbolic Comp.* **5** (1988) pp. 29–35. Also Algorithms in Real Algebraic Geometry (ed. D.S. Arnon and B. Buchberger), Academic Press, London, 1988. Zbl. 663.03015. MR 89g:03009.

[23] Dickenstein,A. & Emiris,I., Multihomogeneous Resultant Matrices. Proc. ISSAC 2002 (ed. T. Mora), ACM Press, New York, 2002, pp. 46–54.

[24] Dodgson,C.L., Condensation of determinants, being a new and brief method for computing their algebraic value. *Proc. Roy. Soc. Ser. A* **15** (1866) pp. 150–155.

[25] Faugère,J.-C., A New Efficient Algorithm for Computing Gröbner Bases Without Reduction to Zero (F_5). Proc. ISSAC 2002 (ed. T. Mora), ACM Press, New York, 2002, pp. 75–83.

[26] Faugère,J.C., Gianni,P., Lazard,D. & Mora,T., Efficient Computation of Zero-Dimensional Gröbner Bases by Change of Ordering. *J. Symbolic Comp.* **16** (1993) pp. 329–344. MR 94k:68095.

[27] Fortuna,E., Gianni,P. & Trager,B., Degree reduction under specialization. *J. Pure Appl. Algebra* **164** (2001) pp. 153–163.

[28] Fortuna,E., Gianni,P., Parenti,P. & Traverso,C., Computing the Topology of a Real Algebraic Surface. Proc. ISSAC 2002 (ed. T. Mora), ACM Press, New York, 2002, pp. 92–100.

[29] Gatermann,K., Symbolic Solutions of Polynomial Equation Systems with Symmetry. Proc. ISSAC 1990 (ed. S. Watanabe & M. Nagata), ACM, 1990, pp. 112–119.

[30] Gianni,P., Properties of Gröbner bases under specializations. Proc. EUROCAL 87 (Springer Lecture Notes in Computer Science 378, Springer-Verlag, Berlin-Heidelberg-etc., 1989), pp. 293–297. MR 91g:13032.

[31] Giusti,M., Heintz,J., Morais,J.E. & Pardo,L.M., When Polynomial Equation Systems can be Solved Fast. Springer Lecture Notes in Computer Science 948(1995), pp. 205–231.

[32] Gonzalez-Vega,L., Rouillier,F., Roy,M.-F. & Trujilo,G., Symbolic Recipes for Real Solutions. Some Tapas of Computer Algebra vol. 4 (ed. A.M. Cohen, Cuypers,H. & Sterk,M.,), Springer-Verlag, 1999, pp. 121–167.

[33] Grigoriev,D.Yu. & Vorobjov,N.N.,Jr., Solving Systems of Polynomial Inequalities in Subexponential Time. *J. Symbolic Comp.* **5** (1988) pp. 37–64. Also Algorithms in Real Algebraic Geometry (ed. D.S. Arnon and B. Buchberger), Academic Press, London, 1988.

[34] Grigoriev,D.Yu. & Vorobjov,N.N.,Jr., Counting connected components of a semi-algebraic set in subexponential time. *Computational Complexity* **2** (1992) pp. 133–184.

[35] Hong,H., Quantifier Elimination for Formulas Constrained by Quadratic Equations. Proc. ISSAC 1993 (ed. M. Bronstein, ACM, 1993) pp. 264–274.

[36] Kalkbrener,M., Solving systems of algebraic equations by using Gröbner bases. Proc. EUROCAL 87 (Springer Lecture Notes in Computer Science 378, Springer-Verlag, Berlin-Heidelberg-etc., 1989), pp. 282–292.

[37] Kaltofen,E. & Lakshman,Y., Improved Sparse Multivariate Polynomial Interpolation Algorithms. Proc. ISSAC 1988 (ed. P. Gianni), Springer Lecture Notes in Computer Science 358, Springer-verlag, 1989, pp. 467–474.

[38] Kaltofen,E., Lee,W. & Lobo,A.A., Early Termination in Ben-Or/Tiwari Sparse Interpolation and a Hybrid of Zippel's Algorithm. Proc. ISSAC 2000 (ed. C. Traverso), ACM, New York, 2000, pp. 192–201.

[39] LaMacchia,B.A. & Odlyzko,A.M., Solving Large Sparse Linear Systems over Finite Fields. Proc. CRYPTO '90 (ed. A.J. Menezes & S.A. Vanstone) Springer Lecture Notes in Computer Science 537, Springer-Verlag, 1991, pp. 109–133.

[40] Mora,T., Groebner Bases for Non-commutative Polynomial Rings. Proc. AAECC-3 (Springer Lecture Notes in Computer Science 229, Springer-Verlag, 1986) pp. 353–362.

[41] Mora,T., An introduction to commutative and non-commutative Gröbner bases. *Theor. Comp. Sci.* **134** (1994) pp. 131–173.

[42] Reid,G.J., Wittkopf,A.D. & Boulton,A., Reduction of systems of nonlinear partial differential equations to simplified involutive forms. *Eur. J. Appl. Math.* **7**(1996) pp. 604–635.

[43] Schwartz,J.T. & Sharir,M., On the "Piano-Movers" Problem: I. The Case of a Two-dimensional Rigid Polygonal Body Moving Amidst Polygonal Barriers. *Comm. Pure Appl. Math.* **36** (1983) pp. 345–398.

[44] Schwartz,J.T. & Sharir,M., On the "Piano-Movers" Problem: II. General Techniques for Computing Topological Properties of Real Algebraic Manifolds. *Adv. Appl. Math.* **4** (1983) pp. 298–351.

[45] Schwartz,J.T. & Sharir,M., On the "Piano-Movers" Problem: III. Coordinating the Motion of Several Independent Bodies: The Special Case of Circular Bodies Moving Amidst Polygonal Barriers. *Int. J. Robot. Res.* **2** (1983) pp. 46–75.

[46] Schwartz,J.T. & Sharir,M., On the "Piano-Movers" Problem: V. The Case of a Rod Moving in Three-Dimensional Space Amidst Polyhedral Obstacles. *Comm. Pure Appl. Math.* **37** (1984) pp. 815–848.

[47] Tran,Q.-N., A Fast Algorithm for Gröbner Basis Conversion and its Applications. *J. Symbolic Comp.* **30** (2000) pp. 451–467. MR 2001k:13043.

[48] Traverso,C., Gröbner trace algorithms. Proc. ISSAC 1988 (ed. P. Gianni), Springer Lecture Notes in Computer Science 358, Springer-Verlag, Berlin-Heidelberg-New York-Tokyo, 1989, pp. 125–138.

[49] Ufnarovski,V., Introduction to noncommutative Gröbner bases theory. Gröbner bases and applications (Linz, 1998), London Math.

Soc. Lecture Note Ser., 251, Cambridge Univ. Press, Cambridge, 1998, pp. 259–280. MR 2000i:16051.

[50] Vidal.J.-P., The Computation of Groebner Bases on a Shared Memory Multiprocessor. Proc. DISCO '90 (Springer Lecture Notes in Computer Science Vol. 429, ed. A. Miola) pp. 81–90.

[51] Weispfenning,V., Differential Term-Orders. Proc. ISSAC 1993 (ed. M. Bronstein, ACM, 1993) pp. 245–253.

[52] Weispfenning,V., Quantifier elimination for real algebra — the cubic case. Proc. ISSAC 1994 (ACM, New York, 1994), pp. 258–263.

[53] Weispfenning,V., Quantifier elimination for real algebra — the quadratic case and beyond. *AAECC* **8** (1997) pp. 85–101.

[54] Wilkinson,J.H., The Evaluation of the Zeros of Ill-conditioned Polynomials. *Num. Math.* **1** (1959) pp. 150–166, 167–180.

5 Tom Kelsey
University of St.Andrews, Scotland

Tutorial: The MAPLE-PVS System

Maple-PVS Tutorial

Tom Kelsey
University of St Andrews

$\leftarrow * \rightarrow$

Calculemus Autumn School $\leftrightarrow$Pisa $\leftrightarrow$ September 2002

Overview

- Introduction to MaplePVS
- Naïve use
- Adding formal checks to Maple procedures
- Summary

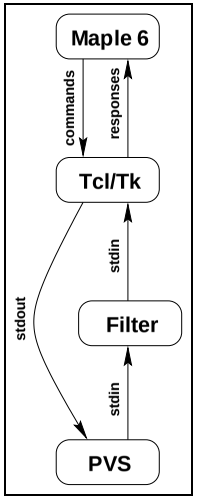
Design

- Maple6+ permits calls to **C** and **FORTRAN** routines
- PVS augmented with real analysis and transcendental function libraries at St Andrews
- Aim: run PVS as a black box tool providing formal checks for Maple expressions

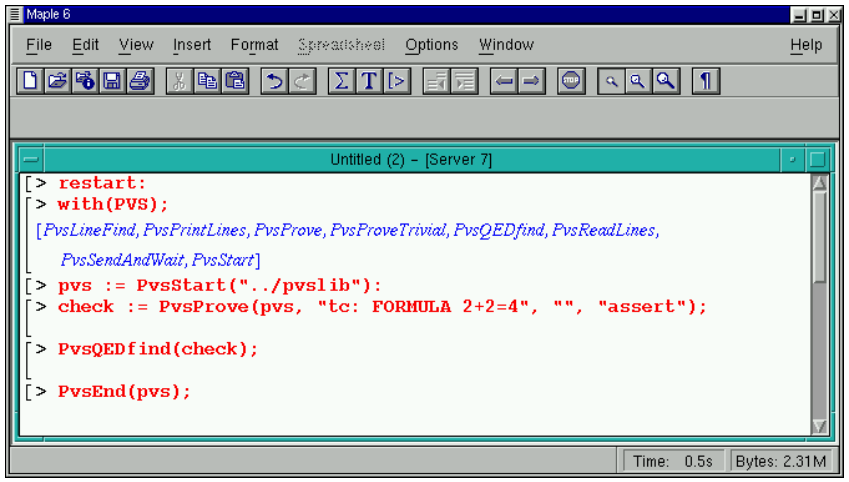
Architecture

Within a Maple session

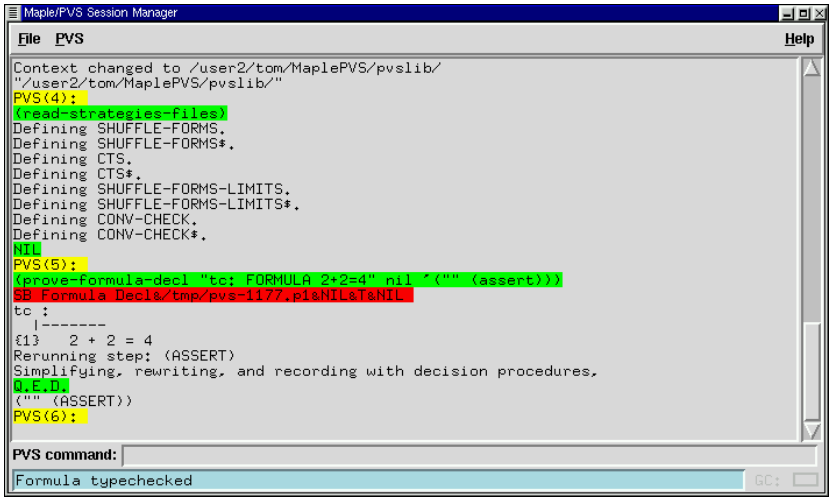
- use **C** interface functions
  - start/end PVS subprocesses
  - send/receive information
- use Tcl/Tk windows
  - display options
  - monitor PVS progress
  - interact if necessary
- PVS acts as a black box



Maple screenshot



Tcl/Tk screenshot



Using MaplePVS – proof attempts

$$\forall x, y \in \mathbb{R}^+ \quad |x - y| = |\sqrt{x} + \sqrt{y}||\sqrt{x} - \sqrt{y}|$$

```
> PvsProve(pvs,
> "sqrt_eq:LEMMA FORALL(x,y:posreal):
> abs(root(x,2)-root(y,2))*abs(root(x,2)+root(y,2))
> = abs(x-y)",
> "roots","then (skolem!)(use "sqrt_difference")
> (use "abs_mult")(assert)");
```


Built-in strategies for convergence and continuity

Is $\frac{1}{|\pi - 6\cos(x)|}$ continuous over $\mathbb{R}$?

```
> PvsProve(pvs, "g:  LEMMA FORALL(y:real):
> continuous(lambda (x:real) :
> 1/exp(pi - abs(6*cos(x))),y)",
> "top_analysis", "cts");
```

Improving Maple using MaplePVS

1. Identify suitable procedures
   - continuity/monotonicity checks
   - initial value problems
2. Analyse the Maple source code
3. Add formal check(s)
4. Use the results to
   - replace incorrect Maple output
   - aid method selection

Floating point discontinuity in Maple

`fdiscont` attempts to find discontinuities of a function over the reals.

- returns a list of numeric ranges
- numeric resolution has a default value of 10^{-3}
- adequate for many input functions

  ... `fdiscont` can be fooled by dense oscillatory functions (such as $sin(500x)$ on $0 \ldots \pi$) - if features are found that are not expected, the resolution, `res`, should be made smaller ...

Floating point discontinuity in MaplePVS

```
PVSfdiscont := proc pvs, f::algebraic, arange::(name = range))
   thm := ( "forall(y(ls,rs)) : continuous(lambda(vars:(ls,rs)): f, y)")
   tctest := PvsTypecheck(pvs, thm, top_analysis) ;
   if not PvsTCfind(pvs, tctest) then
      error "formula fails to typecheck in PVS"
   elif
      PvsQEDfind(PvsProve(pvs, g:FORMULA thm,top_analysis, cts))
   then
      return []
   else
      fdiscont(f, var = ls..rs, 1/100, newton = true)
   end if
end proc
```


Comparison

```
> PVSfdiscont(pvs,sin(750*x), x=0..Pi/2);
                              [ ]
> fdiscont(sin(750*x), x=0..Pi/2);
      [.939969669603023528...940875175375470940]
```

Algebraic discontinuity in Maple

The `discont` procedure finds discontinuities in expressions over $\mathbb{R}$.

For example, the discontinuities for $tan(x)$ are given as

$$\left\{ Z\pi + \frac{1}{2}\pi \right\}$$

where $Z \in \mathbb{Z}$.

It can be fooled by supplying rational trigonometric functions, such as

$$\frac{1}{cos(x)+2} \quad .$$

Algebraic discontinuity in MaplePVS

```
PVSdiscont := proc(pvs, f::algebraic, var)
   thm := "forall(y:real): continuous(lambda(var :real): f , y)";
   tctest := PvsTypecheck(pvs, thm, top_analysis) ;
   if not PvsTCfind(pvs, tctest) then
      error "formula fails to typecheck in PVS"
   elif
      PvsQEDfind(PvsProve(pvs, g:FORMULA thm,top_analysis, cts))
   then
      return {}
   else
      discont(f)
   end if
end proc
```


Initial Value Problems

$$y'(x) = r(x) - q(x)y(x), \quad y(a) = \eta, \quad x \in [a,b] \tag{1}$$

We can check requirements on inputs:

1. $r(x)$ and $q(x)$ are continuous over $[a,b]$;
2. $r(x) - q(x)y(x)$ is continuous, Lipschitz, differentiable over $[a,b]$.

Use to fine tune the Maple `dsolve` procedure giving $y(x)$ and check:

1. $y'(x) - r(x) + q(x)y(x) = 0$;
2. $y(a) = \eta$;
3. $y(x)$ has removable poles, non-removable branch points and/or is itself continuous.

IVPs in MaplePVS

```
qsolve := proc(pvs r, q, a, b, η)
z1 := PvsProve(pvs,
   "g: FORMULA FORALL (v:I[a,b]) : continuous(lambda (x:I[a,b]) : r(x), v)",
   "top_analysis", "cts");
z2 := PvsProve(pvs,
   "g: FORMULA FORALL (v:I[a,b]) : continuous(lambda (x:I[a,b]) : q(x), v)",
   "top_analysis", "cts");
if not (PvsQEDfind(z1) and PvsQEDfind(z2)) then ERROR(`invalid input`)
        ⋮                    ⋮                    ⋮
```

. . . if the input is OK, then . . .

else

sol := dsolve({diff(y(x), x) = $r(x) - q(x)$y(x), y(a) = η}, y(x));

$diffsol$:= diff(sol, x);

$z3$:= PvsProve(pvs,
"g: FORMULA FORALL (v:l[a,b]) : diffsol(v) = r(v) - q(v)*sol(v)",
"top_analysis", "grind");

$z4$:=
PvsProve(pvs, "g: FORMULA sol(a) = eta", "top_analysis", "grind");

$z5$:= PvsProve(pvs,
"g: FORMULA FORALL (v:l[a,b]): continuous(lambda (x:l[a,b]) : sol(x),v)",
"top_analysis", "cts")

⋮ ⋮ ⋮

. . . check the validity of the Maple solution

⋮ ⋮ ⋮

fi;

if not (PvsQEDfind($z4$) **and** PvsQEDfind($z5$) **and** PvsQEDfind($z6$))
then ERROR(‘*invalid solution*’)
else
return sol
fi
end

Other properties

Monotonicity: $f'(x) > 0 \quad \forall x \in (a, b)$

OddMap: $g(x) = -g(-x) \quad \forall x \in [a, b]$

EvenMap: $h(x) = h(-x) \quad \forall x \in [a, b]$

Is $f(x) > g(x)$ in the interval $[a, b]$?

Summary

MaplePVS works without the real analysis libraries

Continuity and convergence strategies add expressivity

Augment existing Maple procedures

Develop new procedures, optimised by formal checks

Tcl/Tk tool support for refining strategies and procedures

6 Manfred Kerber

University of Birmingham, England

Course: Partiality in Deduction and Computation

Partiality in Deduction and Computation

Manfred Kerber
School of Computer Science
The University of Birmingham
Birmingham, B15 2TT, England
e-mail: M.Kerber@cs.bham.ac.uk
WWW: http://www.cs.bham.ac.uk/~mmk/
ftp://ftp.cs.bham.ac.uk/pub/authors/M.Kerber/Talks/02-calculemus.pdf
ftp://ftp.cs.bham.ac.uk/pub/authors/M.Kerber/Talks/02-calculemus.ps.gz

Overview

- Classical logic revisited, problems
- Problems with logic and computation
- Problems with partial functions, examples
- Solutions
  - Set theory
  - Sorts
  - Partial functions (two-valued vs three-valued logic)

Classical Logic

- Constant symbols like 2 or socrates. Denote an object.
- Function symbols like $+$ or father. Denote a (total) function.
- Predicate symbols like $\geq$ or Mortal. Denote a relation.

Everything assumed to denote something, function symbols are assumed to be **total**, that is, there is no input to which there is no answer.
For $+$ this means e.g. that there is no overflow due to a sum getting too big.

Problems with the Classical Approach

Standard Expressions	Problematic Expressions
$1/2$	$1/0$
$s(p(2))$	$s(p(0))$
$2+3$	$mary + peter$
$4*5$	$anne * john$
$6+9>3$	$mary + peter > 3$
$Mortal(mother(socrates))$	$Mortal(father(2))$

Hägar the Horrible

© by Dik Browne

Levels of Program Behaviour

1. Correct behaviour: correct output for any input
   But, sometimes no "correct" behaviour possible, e.g. $1/0$.
2. Crash: $1/0$ results in crash (or exception).
3. Non-termination: no output for some input could define division `x/y` by

```
a := 0 ;
b := x ;
while b >= y do
    b := b - y;
    a := a + 1
 end
```

4. Faulty output: the system gives a wrong answer for some input: $1/0 \mapsto 0$.

Hoare Triples

Axiom Schemes: (A, B, C, D formulae)

SKIP $\{A\}\,\mathrm{skip}\,\{A\}$

ASS $\left\{A^t_x\right\}\,\mathrm{x := t}\,\{A\}$ $\quad A^t_x$ replace free x in A by t

Rules:

SEQ $$\frac{\{A\}\,\mathrm{S_1}\,\{B\} \qquad \{B\}\,\mathrm{S_2}\,\{C\}}{\{A\}\,\mathrm{S_1; S_2}\,\{C\}}$$

IF $$\frac{\{A \wedge C\}\,\mathrm{S_1}\,\{B\} \qquad \{A \wedge \neg C\}\,\mathrm{S_2}\,\{B\}}{\{A\}\,\mathbf{if}\ \mathrm{C}\ \mathbf{then}\ \mathrm{S_1}\ \mathbf{else}\ \mathrm{S_2}\ \mathbf{fi}\,\{B\}}$$

WHILE $$\frac{\{A \wedge C\}\,\mathrm{S}\,\{A\}}{\{A\}\,\mathbf{while}\ \mathrm{C}\ \mathbf{do}\ \mathrm{S}\ \mathbf{od}\,\{A \wedge \neg C\}}$$ $A =$ loop invariant

WEAK $$\frac{A \to B \qquad \{B\}\,\mathrm{S}\,\{C\} \qquad C \to D}{\{A\}\,\mathrm{S}\,\{D\}}$$

What to do with "undefined" expressions?

Naive usage of $\frac{1}{0}$ in two-valued logic: term can be instantiated in $\forall n.n \cdot 0 = 0$, hence $\frac{1}{0} \cdot 0 = 0$.

$\frac{1}{0}$ equal to 0 or not (two-valued, tertium non datur):
Program fragment P which consists of the assignment to an undefined expression x:=1/0. Assume $\frac{1}{0} = 0$ to be false.

The Hoare triple $\left\{\frac{1}{0} \neq 0\right\}\,\mathrm{x{:=}1/0}\left\{x \neq 0\right\}$, which is an instance of the *assign* axiom-schema, can be weakened (by the weakening rule) to the valid Hoare triple

$$\{\mathrm{T}\}\,\mathrm{x{:=}1/0}\,\{x \neq 0\}\,.$$

Three-valued Hoare (preliminary version)

[Joint work with Viviana Bono (Torino)]

Axiom Schemes: (A, B, C, D formulae)

D	
0	1
u	0
1	1

SKIP^3 $\quad \{A\}\,\mathrm{skip}\,\{A\}$

ASS^3 $\quad \left\{\mathbf{Def}(t) \wedge A^t_x\right\}\,\mathrm{x := t}\,\{\mathbf{Def}(t) \wedge \mathbf{Def}(x) \wedge A\}$

Rules:

SEQ^3 $\quad \dfrac{\{A\}\,\mathrm{S}_1\,\{B\} \qquad \{B\}\,\mathrm{S}_2\,\{C\}}{\{A\}\,\mathrm{S}_1;\mathrm{S}_2\,\{C\}}$

IF^3 $\quad \dfrac{A \to \mathbf{D}(C) \qquad \{A \wedge C\}\,\mathrm{S}_1\,\{B\} \qquad \{A \wedge \neg C\}\,\mathrm{S}_2\,\{B\}}{\{A\}\,\mathrm{if\ C\ then\ S}_1\ \mathrm{else\ S}_2\ \mathrm{fi}\,\{B\}}$

WHILE^3 $\quad \dfrac{A \to \mathbf{D}(C) \qquad \{A \wedge C\}\,\mathrm{S}\,\{A\}}{\{A\}\,\mathrm{while\ C\ do\ S\ od}\,\{A \wedge \neg C\}}$ $\quad A =$ loop invariant

WEAK^3 $\quad \dfrac{A \to B \qquad \{B\}\,\mathrm{S}\,\{C\} \qquad C \to D}{\{A\}\,\mathrm{S}\,\{D\}}$

Example from Computer Algebra

Which answer do we expect from a CAS:

consider: $f(x) = \frac{x^2-1}{x-1}$ and $g(x) = x + 1$

$f(0) = g(0)$	**YES**
$f(0) = g(1)$	**NO**
$f(x) = g(x)$	**Depends** on context, e.g. • space of meromorphic functions: **YES** • all real numbers: **not defined for** $x = 1$ • for $\mathbb{R}\backslash\{1\}$: **YES**
$f = g$	**Depends**
$f(1) = g(1)$	f **is undefined for** $x = 1$
$\forall x. x \neq 1 \to f(x) = g(x)$	**YES**

Some Examples with Maple

```
I    f := x-> (x^2 -1)/(x-1);
                  2
                 x  - 1
O    f := x -> ------
                 x - 1

I    g := x-> x+1;

O    g := x -> x + 1

I    verify(f(0),g(0));

O    true

I    verify(f(0),g(1));

O    false
```


Some Examples with Maple (Cont'd)

```
I    verify(f(x),g(x));

O    false

I    verify(f(1),g(1));

O    Error, (in f) numeric exception: division by zero

I    verify(f,g);

O    false
```

All seems fine!

Some Examples with Maple (Cont'd)

```
I     h:= x-> (x+1)*(x-1)/(x-1);
O     h := x -> x + 1
I     verify(h(0),g(0));
O     true
I     verify(h(0),g(1));
O     false
I     verify(h(x),g(x));
O     true
I     verify(h(1),g(1));
O     true
```


Some Examples with Maple (Cont'd)

```
I     verify(h,g);
O     false
I     k:= x-> x+1;
O     k := x -> x + 1
I     verify(h,k);
O     false
I     x = 1 or verify(f(x),g(x));
O     false
```

Take great care, results may be surprising!

Main Approaches for Treatment of Partiality

1. Ignore the Problem.
2. Deal with partiality outside logic, set theory.
3. Syntactic exclusion of "dirty" expressions.
4. Partiality modelled by partial functions and total predicates.
5. Partiality modelled by partial functions and partial predicates.

Now, these approaches in more detail.

Ignore the Problem

Approach: A value is assigned to $\frac{1}{0}$, either a fixed value (e.g. 0) or an undetermined one.

Example: assign to non-sense expressions like father(2) either a concrete value (like 0) or an indefinite value (some natural number, but we don't know which one).

Advantage: Does not require any adaptation.

Disadvantage: In both cases it is necessary to tolerate undesired theorems, in the first case, for instance, $\frac{1}{0} = 0$, or in the second case from $0 \cdot x = 0$ the instance $0 \cdot \frac{1}{0} = 0$.

Summary: This approach is not satisfying, if such theorems are unwanted, which is normally the case in mathematics.

Set Theory

Approach: Take sets as primitives, define relations as subsets of Cartesian products, and define function as left-total right-unique relations. Partial function as right-unique relation.

Example: $/ = \{(1, 2, 0.5), (4, 2, 2), (6, 2, 3), \ldots\}$

Advantage: Well established, well understood, adequate.

Disadvantage: Unintuitive, since abandons idea of a function with input and output. Not well suited for many practical applications.

Summary: Theoretically sound, practically difficult to use.

Syntactic Exclusion

Approach: $\frac{x}{0}$ is treated as syntactically ill-formed, for instance, by using a sorted logic, in which the domain of the $\frac{x}{y}$ function is defined to be $\mathbb{R} \times \mathbb{R}_0$.

Example: rule out cases like s(father(2)), mary + peter, anne * john, i.e., non-sense expressions are syntactically excluded.

Advantage: If applicable very adequate. Reflects the usual way of handling undefined expressions in mathematics: **Assure that all expressions are defined before beginning to reason.**

Disadvantage: Not decidable. For instance, consider $\frac{1}{f(x)}$. Necessary to exclude this expression for those x where $f(x) = 0$; might be not computable at all.

Summary: Remedies the problem of partiality in (many important) cases but does not provide a full solution.

Partial Functions – Total Predicates

Approach: All atomic expressions containing a meaningless term are considered as false.

Example: Bottom element, $\perp$, and **two truth values**, atoms containing non-sense terms are evaluated to false.

Advantage: Problems that accompany treating a third truth value are avoided. Partial functions can be handled within the classical two-valued framework.

Disadvantage: Partly unintuitive results: e.g. elementary arithmetic $\forall x, y, z . z = \frac{x}{y} \rightarrow x = y * z$ is a theorem of such systems (for $y = 0$, the formula $z = \frac{x}{0}$ evaluates to F). However, mathematical answer: true for $y \neq 0$.

Summary: Two-valued approach, partly adequate.

Partial Functions – Partial Predicates

Approach: As in previous approach terms like $\frac{1}{0}$ are not defined and semantically either uninterpreted or interpreted by some error element. Atomic formulae, containing such an undefined term are not interpreted by a truth value (true or false) at all or are interpreted by a third truth value (undefined).

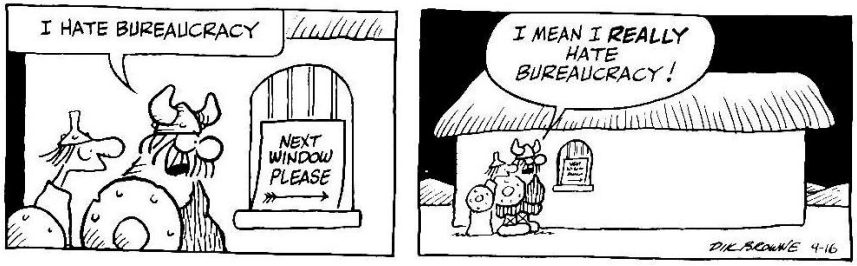
Example: Bottom element, $\perp$, and **three truth values**

Advantage: Partiality adequately modelled.

Disadvantage: Partly unintuitive, formalisation in three-valued logic. Introduce $\perp$ for denoting meaningless individuals and a third truth value P, standing for the "undefined" (meaningless) truth value.

Summary: Three-valued approach, adequate, but at first unfamiliar.

More Hägar the Horrible

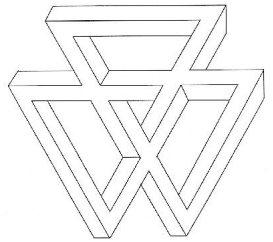


© by Dik Browne

End of lecture 1

Next lecture: Three-valued approach in more depth.

Partiality in Deduction and Computation (Cont'd) A Three-Valued Approach to Partiality



A mechanisation to Three-Valued Kleene Logic

[Joint work with Michael Kohlhase (CMU)]

- Motivation
- Three-valued syntax, semantics
- Relativisation
- Tableau calculus
- Relationship between 3-valued and 2-valued systems
- Strategy result

Motivation

Try to avoid unwanted theorems in classical 2-valued logic

Theorem? $\forall x_{\mathbb{R}}, y_{\mathbb{R}}, z_{\mathbb{R}} z = \frac{x}{y} \rightarrow x = y * z$ (TH)

in 2-valued logic YES:
Critical instance $z = \frac{x}{0} \rightarrow x = 0 * z$ evaluates to T for all x, z
since $z = \frac{x}{0} \mapsto \mathrm{F}$

in 3-valued Kleene logic NO:
Critical instance $1 = \frac{1}{0} \rightarrow 1 = 0 * 1$ evaluates to P
since $1 = \frac{1}{0} \mapsto \mathrm{P}$ and $1 = 0 * 1 \mapsto \mathrm{F}$

More precise theorem:

$$\forall x_{\mathbb{R}}, y_{\mathbb{R}_0}, z_{\mathbb{R}} z = \frac{x}{y} \rightarrow x = y * z \qquad \text{(TH')}$$

Mechanisation by:

3-valued sorted tableau calculus (also resolution calculus developed)

Partial Functions

What is the semantic status of atomic formulae containing meaningless expressions like $\frac{0}{0}$ as in $\frac{0}{0} = 0$?

Possible answers:

- **false** (reduction to classical logic)
- do **not** assign **any truth value** (e.g., intuitionistic logic)
- **undefined**, **paradoxical** (three-valued logic)

[Kleene52]: **"strong three-valued logic"** for partial recursive predicates on the set of natural numbers. intuitive meaning of the **third truth value** "**undefined**", "**unknown**", "**undefinable**", "**problematic**", "**paradoxical**", abbreviated as P.

Multi-Valued Logics

- **General Approach:** [Łukasiewicz20, Post21]
- **Resolution Calculus etc.:**
  [Carnielli87, Hähnle92, Baaz/Fermüller92]

$$\frac{A^{\alpha}[s] \quad A^{\beta}[t]}{*} \quad \sigma(s) = \sigma(t),\ \mathrm{mgu}(\sigma),\ \text{and } \alpha \neq \beta,\ \alpha, \beta \text{ truth values}$$

- **Can Kleene logic processed this way?**
  - **strict functions and predicates**
    (if $\mathcal{I}(t) = \bot$, then $\mathcal{I}(f(t)) = \bot$ and $\mathcal{I}(P(t)) = \mathtt{P}$)
  - **but** in Kleene Logic **quantifiers don't range** over all individuals, not **over** $\bot$
  - **Sort techniques** [Weidenbach]

Syntax

- **Symbols**
  - **sort symbols** $S, T, \ldots \in Sorts$, with $\mathcal{D} \in Sorts$ and $Sorts$ finite
  - **sorted variables** $x_S, y_T, \ldots \in \mathrm{Vars}$
  - **function symbols** $f, g, \ldots \in \mathrm{Funcs}$
  - **predicate symbols** $P, Q, \ldots \in \mathrm{Preds}$
- **Terms:** $s, t, \ldots$
  - constants, variables and function application
- **Atoms:** A, B
  - proper atom: $P(t^1, \ldots, t^n)$
  - **sort atom:** $t \lessdot S$
- **compound formulae:** $A \wedge B, \neg A, \mathbf{D}A, \forall x_S A$
- called one-sorted if $Sorts = \{\mathcal{D}\}$

Semantics

- **strict semantics for terms** with respect to $\bot$
- **3 Truth values** $\mathtt{F}, \mathtt{P}, \mathtt{T}$
- $\mathcal{I}_\varphi(P(t^1, \ldots, t^n)) = \mathtt{P}$,
  if $(\mathcal{I}_\varphi(t^1), \ldots, \mathcal{I}_\varphi(t^n)) \notin \mathrm{Dom}(\mathcal{I}(P))$, in particular, if $\mathcal{I}(t^i) = \bot$.
- **Connectives:**

$\wedge$	F	P	T
F	F	F	F
P	F	P	P
T	F	P	T

$\neg$	
F	T
P	P
T	F

D	
F	T
P	F
T	T

- **Quantification:** $\mathcal{I}_\varphi(\forall x_S.A) := \widetilde{\forall}(\{\mathcal{I}_{\varphi,[a/x]}(A) \mid a \in \mathcal{I}(S)\})$

$$\widetilde{\forall}(\sigma) := \begin{cases} \mathtt{T} & \text{for } \sigma = \{\mathtt{T}\} \text{ or } \sigma = \emptyset \\ \mathtt{P} & \text{for } T = \{\mathtt{T}, \mathtt{P}\} \text{ or } \sigma = \{\mathtt{P}\} \\ \mathtt{F} & \text{for } \mathtt{F} \in \sigma \end{cases}$$

- **Model:** $\mathcal{M} = (\mathcal{A}, \mathcal{I}) \models A$, iff $\mathcal{I}_\varphi(A) = \mathtt{T}$

Refutation by Tableau Calculus

Let A be a formula, then A^α is the corresponding **labelled formula**, for $\alpha \in \{\mathrm{F}, \mathrm{P}, \mathrm{T}\}$

Show

$$\{A_1, \ldots, A_n\} \models B$$

by **refuting**

$$A_1^{\mathrm{T}}, \ldots, A_n^{\mathrm{T}}, B^{\mathrm{PF}}$$

"Quartum non datur"-Principle

multi-labels stand for sets, e.g.

$$B^{\mathrm{PF}} = B^{\{\mathrm{PF}\}} = B^{\mathrm{P}} \vee B^{\mathrm{F}}$$

"Tertium non datur" vs "Quartum non datur"

Tertium non datur:

- To prove: $\{A_1, \ldots, A_n\} \models B$
- Assume assumptions $A_1, \ldots, A_n$
- Show that under assumptions B **can't be false**, i.e., bring $A_1^{\mathrm{T}}, \ldots, A_n^{\mathrm{T}}, B^{\mathrm{F}}$ to **contradiction**

Quartum non datur:

- To prove: $\{A_1, \ldots, A_n\} \models B$
- Assume assumptions $A_1, \ldots, A_n$
- Show that under assumptions B **can't be false and can't be undefined**, i.e., bring $A_1^{\mathrm{T}}, \ldots, A_n^{\mathrm{T}}, B^{\mathrm{F}}$ to **contradiction and** bring $A_1^{\mathrm{T}}, \ldots, A_n^{\mathrm{T}}, B^{\mathrm{P}}$ to **contradiction**

Computational Price

- Instead of **one** contradiction proof, it seems that we have to make two contradiction proofs.
- However, we will see that under certain quite general assumptions traditional proofs can be mapped to proofs in the three-valued framework, i.e., in many important cases, **no additional cost**.

Tableau Rules

$$\frac{(A \wedge B)^{\mathrm{PT}}}{\begin{array}{c} A^{\mathrm{PT}} \\ B^{\mathrm{PT}} \end{array}} \qquad \frac{(A \wedge B)^{\mathrm{P}}}{\begin{array}{c} A^{\mathrm{PT}} \\ B^{\mathrm{PT}} \\ A^{\mathrm{P}} \mid B^{\mathrm{P}} \end{array}} \qquad \frac{(A \wedge B)^{\mathrm{FP}}}{A^{\mathrm{FP}} \mid B^{\mathrm{FP}}}$$

$$\frac{(\forall x_S.A)^{\mathrm{PT}}}{[y_S/x_S]A^{\mathrm{PT}}} \qquad \frac{(\forall x_S.A)^{\mathrm{P}}}{\begin{array}{c} [f(y^1, \ldots, y^n)/x_S]A^{\mathrm{P}} \\ [y_S/x_S]A^{\mathrm{PT}} \\ (f(y^1, \ldots, y^n) \leqslant S)^{\mathrm{T}} \end{array}} \qquad \frac{(\forall x_S.A)^{\mathrm{FP}}}{\begin{array}{c} [f(y^1, \ldots, y^n)/x_S]A^{\mathrm{FP}} \\ (f(y^1, \ldots, y^n) \leqslant S)^{\mathrm{T}} \end{array}}$$

$$\frac{(t \leqslant \mathcal{D})^{\mathrm{P}}}{*} \qquad \frac{(t \leqslant S)^{\mathrm{P}}}{(t \leqslant \mathcal{D})^{\mathrm{F}}} \qquad \frac{\begin{array}{c} A^\alpha \\ B^\beta \end{array}}{* \mid \mathcal{C}(\sigma)}\ \sigma \qquad \frac{\begin{array}{c} C^\gamma \\ (t \leqslant \mathcal{D})^{\mathrm{F}} \end{array}}{* \mid \mathcal{C}(\sigma)}\ \sigma$$

$$\sigma(A) = \sigma(B) \qquad \mathcal{C}(\sigma) = ((t_1 \leqslant S_1) \wedge \ldots \wedge (t_n \leqslant S_n))^{\mathrm{FP}}$$

Properties

- **Soundness**
  - If tableau constructed from $A_1^{\mathsf{T}}, \ldots, A_n^{\mathsf{T}}, B^{\mathsf{PF}}$ can be **closed** then $\{A_1, \ldots, A_n\} \models B$
- **Completeness**
  - If $\{A_1, \ldots, A_n\} \models B$ then tableau constructed from $A_1^{\mathsf{T}}, \ldots, A_n^{\mathsf{T}}, B^{\mathsf{PF}}$ can be **closed**.
- show with methods from [Weidenbach94]

Example

Consider $\{\mathsf{A1}, \mathsf{A2}, \mathsf{A3}, \mathsf{A4}, \mathsf{A5}\} \models \mathsf{T}$ with

A1 $\forall x{:}\mathbb{R}.\, x \neq 0 \rightarrow x{:}\mathbb{R}_0$

A2 $\forall x{:}\mathbb{R}_0.\, \frac{1}{x}{:}\mathbb{R}_0$

A3 $\forall x{:}\mathbb{R}_0.\, x^2 > 0$

A4 $\forall x{:}\mathbb{R}.\forall y{:}\mathbb{R}.\, x - y{:}\mathbb{R}$

A5 $\forall x{:}\mathbb{R}.\forall y{:}\mathbb{R}.\, x - y = 0 \rightarrow x = y$

T $\forall x{:}\mathbb{R}.\forall y{:}\mathbb{R}.\, x \neq y \rightarrow \left(\frac{1}{x-y}\right)^2 > 0$

A **tableau proof for a consequent** $\Phi \models T$ is a **closed tableau constructed from** $\Phi^{\mathsf{T}} \cup \left\{T^{\mathsf{FP}}\right\}$, i.e. here from $\mathsf{A1}^{\mathsf{T}}$, $\mathsf{A2}^{\mathsf{T}}$, $\mathsf{A3}^{\mathsf{T}}$, $\mathsf{A4}^{\mathsf{T}}$, $\mathsf{A5}^{\mathsf{T}}$, T^{FP}

Relativisation to One-Sorted Logic

☞ $\Re^{Sorts}$ as identity on terms and atoms (forget sort information), homomorphic on connectives, and

$$\Re^{Sorts}(\forall x{:}S.\Phi) := \forall x{:}\mathcal{D}.S(x) \rightarrow \Re^{Sorts}(\Phi)$$

☞ For any of the new predicate symbols S we add the axiom: $\forall x{:}\mathcal{D}.\mathbf{D}S(x)$. Set of these axioms is denoted by $\Re^{Sorts}(\Sigma)$

Relativisation to Unsorted Logic

☞ Define $\Re^{\mathcal{D}}$ as identity on terms and proper atoms (forget sort information)

☞ $\Re^{\mathcal{D}}(t{:}\mathcal{D}) := \mathcal{D}(t)$

☞ $\Re^{\mathcal{D}}(\forall x{:}\mathcal{D}.A) := \forall x.\mathcal{D}(x) \rightarrow \Re^{\mathcal{D}}(A)$

☞ Signature axioms $\Re^{\mathcal{D}}(\Sigma)$ like

$\forall x_1, \ldots, x_n.P^n(x_1, \ldots, x_n) \vee \neg P^n(x_1, \ldots, x_n) \rightarrow (\mathcal{D}(x_1) \wedge \ldots \wedge \mathcal{D}(x_n))$
$\forall x_1, \ldots, x_n.\mathcal{D}(f(x_1, \ldots, x_n)) \rightarrow (\mathcal{D}(x_1) \wedge \ldots \wedge \mathcal{D}(x_n))$

for predicate symbols $P \in \mathsf{Preds}^n$ and
for function symbols $f \in \mathsf{Funcs}^n$, furthermore

$$\forall x.\mathcal{D}(x) \vee \neg\mathcal{D}(x)$$

Example Relativisation

$\Re^{Sorts} \circ \Re^{\mathcal{D}}$ of $\{A1, A2, A3, A4, A5\} \models T$ is

$\{R1, R2, R3, R4, R5, R^{\mathbb{R}}, R^{\mathbb{R}_0}, R^{=}, R^{>}, R^{-}, R^{/}, R^{2}, \mathcal{D}^{\mathbf{D}}\} \models RT$ with:

R1 $\forall x.\mathcal{D}(x) \rightarrow (\mathbb{R}(x) \rightarrow (x \neq 0 \rightarrow \mathbb{R}^*(x)))$

R2 $\forall x.\mathcal{D}(x) \rightarrow (\mathbb{R}^*(x) \rightarrow \mathbb{R}^*(\frac{1}{x}))$

R3 $\forall x.\mathcal{D}(x) \rightarrow (\mathbb{R}^*(x) \rightarrow x^2 > 0)$

R4 $\forall x.\mathcal{D}(x) \rightarrow (\mathbb{R}(x) \rightarrow (\forall y.\mathcal{D}(y) \rightarrow (\mathbb{R}(y) \rightarrow \mathbb{R}(x-y))))$

R5 $\forall x.\mathcal{D}(x) \rightarrow (\mathbb{R}(x) \rightarrow (\forall y.\mathcal{D}(y) \rightarrow (\mathbb{R}(y) \rightarrow (x - y = 0 \rightarrow x = y))))$

RT $\forall x.\mathcal{D}(x) \rightarrow (\mathbb{R}(x) \rightarrow (\forall y.\mathcal{D}(y) \rightarrow (\mathbb{R}(y) \rightarrow (x \neq y \rightarrow \left(\frac{1}{x-y}\right)^2 > 0))))$

The set of signature axioms $\Re^{\mathcal{D}}(\Sigma \cup Sorts^*) \cup \Re^{\mathcal{D}}(\Re^{Sorts}(\Sigma))$:

$R^{\mathbb{R}}$ $\forall x.\mathcal{D}(x) \rightarrow \mathbf{D}\mathbb{R}(x)$

$R^{\mathbb{R}_0}$ $\forall x.\mathcal{D}(x) \rightarrow \mathbf{D}\mathbb{R}_0(x)$

$R^{=}$ $\forall x, y.(x = y \vee x \neq y) \rightarrow \mathcal{D}(x) \wedge \mathcal{D}(y)$

$R^{>}$ $\forall x, y.(x > y \vee x \not> y) \rightarrow \mathcal{D}(x) \wedge \mathcal{D}(y)$

R^{-} $\forall x, y.\mathcal{D}(x-y) \rightarrow \mathcal{D}(x) \wedge \mathcal{D}(y)$

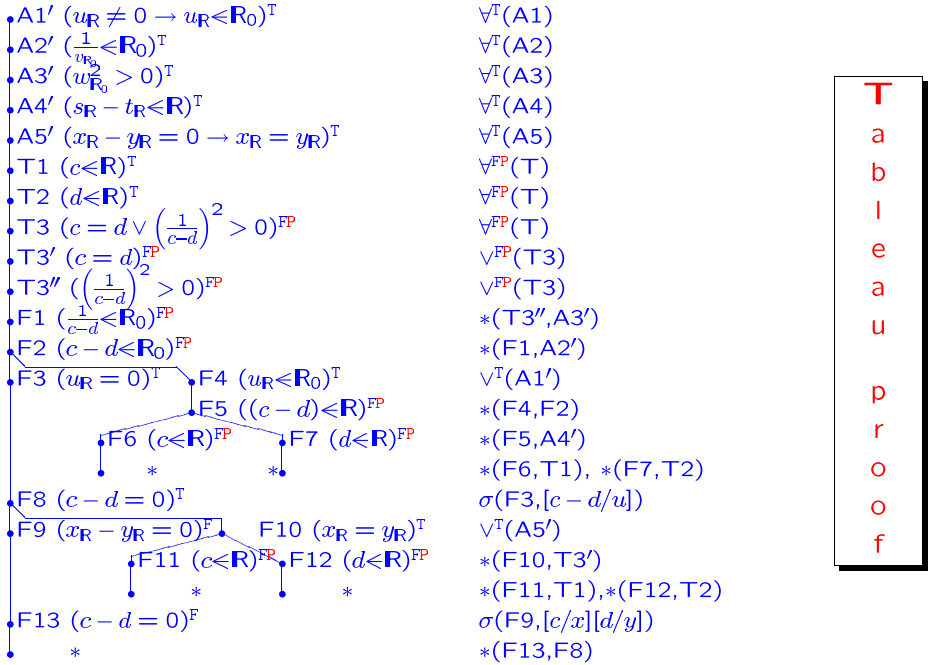
$R^{/}$ $\forall x.\mathcal{D}(\frac{1}{x}) \rightarrow \mathcal{D}(x)$

R^{2} $\forall x.\mathcal{D}(x^2) \rightarrow \mathcal{D}(x)$

$\mathcal{D}^{\mathbf{D}}$ $\forall x.\mathcal{D}(x) \vee \neg\mathcal{D}(x)$

Sorted vs Unsorted Logic

- Unsorted System is **easier to understand**, possible to define semantics of sorted system in unsorted system
- Representation in unsorted system **much less natural**
- Because of larger formulae, **much less efficient automation**
- **Complicated connection between 3-valued and 2-valued system**



Conservativity Theorem

Each tableau proof (using compound rules only) for a "normal" problem $\Phi \models T$ (no $\mathbf{D}$ in Φ and T) in **3-valued logic** can be isomorphically transformed into a tableau proof in **2-valued logic**.

Eliminate $\mathtt{P}$ in the rules.

$$\frac{(A \wedge B)^{\mathrm{PT}}}{\begin{array}{c} A^{\mathrm{PT}} \\ B^{\mathrm{PT}} \end{array}} \quad \rightsquigarrow \quad \frac{(A \wedge B)^{\mathrm{T}}}{\begin{array}{c} A^{\mathrm{T}} \\ B^{\mathrm{T}} \end{array}}$$

Question: Does the reverse hold for 3-valued theorems?
Answer: No

Counterexample:

Proof for $A \models A \vee (B \vee \neg B)$

(A) $(A)^{\mathrm{T}}$	
(T) $(A \vee (B \vee \neg B))^{\mathrm{F}}$	
(F1) $(A)^{\mathrm{F}}$	$\vee^{\mathrm{F}}$(T)
(F2) $(B \vee \neg B)^{\mathrm{F}}$	$\vee^{\mathrm{F}}$(T)
(F3) $(B)^{\mathrm{F}}$	$\vee^{\mathrm{F}}$(F2)
(F4) $(\neg B)^{\mathrm{F}}$	$\vee^{\mathrm{F}}$(F2)
(F5) $(B)^{\mathrm{T}}$	$\neg^{\mathrm{F}}$(F4)
$*$	$*$(F5,F3)

☞ This proof in the two-valued system **cannot be lifted** to a proof in the three-valued system. **Why?**

☞ **Usual theorems** employ preconditions as weak as possible and consequences as strong as possible. For instance, in a mathematical context we would expect theorems like A, $B \vee \neg B$, $A \wedge (B \vee \neg B)$, but not the one above.

Strategy Result

☞ Note from example the general principle: **How to generate a 2-valued proof that is not liftable to a 3-valued proof? close the tableau by two complementary formulae, which both stem from the theorem**
i.e. closure in 2-valued, but not in 3-valued setting.

Theorem [Strategy]:

Let $\mathcal{STR}$ be the control strategy never to close a tableau on two formulae both stemming from the theorem , then each 2-valued tableau proof found using $\mathcal{STR}$ can be lifted to a 3-valued tableau proof for the theorem.

Conclusion

☞ **3-valued logic** for informal mathematical reasoning with partial functions, generalises Kleene's system.

☞ **Combination of**

- **Many-valued logics** [Hähnle,Baaz,Fermüller] for the correct treatment of the undefined
- **Sorted logics** with dynamic sorts [Weidenbach] for the adequate treatment of the defined

☞ Soundness, Completeness

☞ Conservativity

☞ Derivational complexity close to case with total functions.

☞ Strategy

☞ **Adequate Treatment of partiality is possible.**

Paradoxes and Partiality

© by Dik Browne