## CAS 734 Winter 2005

# 09 Symbolic Computation in Formal Proofs

Instructor: W. M. Farmer

Revised: 23 March 2005

# Kinds of Symbolic Computation

- Tactics.
- Decision procedures.
- Simplification procedures.
- Problem-solving procedures.
- Hybrid procedures.

# Tactics

- A **tactic** is a program that reduces a goal in a proof to a set of subgoals by applying the rules of inference of the proof system.

- Issues of concern:
  - Transparency: Are the intermediate steps taken by a tactic accessible?
  - Coverage: What kind of techniques can be effectively formalized as tactics?

- Examples:
  - Logical simplification.
  - Applying theorems.
  - Performing proof by induction.

# Decision Procedures

- A **decision procedure** for a set $\mathcal{S}$ of formulas of a theory $T$ is an algorithm that answers whether or not $T \models A$ is true for each $A \in \mathcal{S}$.

- Issues of concern:
  - Coverage: How big is $\mathcal{S}$?
  - Efficiency: How fast is the algorithm?
  - Correctness: How trustworthy is the algorithm?
  - Combination: How can different decision procedures be combined?

- Techniques:
  - Binary decision diagrams (BDDs).
  - Automated theorem proving techniques based on, for example, resolution and semantic tableaux.
  - Term rewriting.
  - Quantifier elimination.

# Quantifier Elimination

- A **quantifier elimination** method for a set $\mathcal{S}$ of formulas of a theory $T$ consists of:
  - A set $\mathcal{B}$ of **basic formulas** such that it is easy to answer whether or not $T \models B$ for each $B \in \mathcal{B}$. ($\mathcal{B}$ is often a set of quantifier-free formulas.)
  - An algorithm that, given a formula $A \in \mathcal{S}$, produces a boolean combination $C$ of members of $\mathcal{B}$ such that $T \models A \Leftrightarrow C$. (The key step in the algorithm "eliminates quantifiers".)

- Decision procedures based on quantifier elimination:
  - Additive number theory (Presburger 1929).
  - Multiplicative number theory (Skolem 1930).
  - Real closed fields (Tarski 1948).
  - Algebraically closed fields (Tarski 1948).

# Simplification Procedures

- A **simplification procedure** for a set $\mathcal{S}$ of expressions of a theory $T$ is an algorithm that, given an expression $E \in S$, returns a "simpler" expression $E'$ such that $T \models E = E'$.

- Issues of concern:
  - Simplicity: How is simplicity measured?
  - Order of operation: What is done first, second, etc.?
  - Persistence: When does the algorithm give up?

- Examples:
  - Logical simplification.
  - Arithmetic evaluation.
  - Function application evaluation.
  - Term rewriting.
  - Algebraic simplification (using cancellation and collecting like terms).

# Computational Domains

- A **computational domain** is a set of data structures that represents a set of mathematical elements (such as the integers) and a set of operations on the data structures that implement mathematical functions (such as addition and multiplication).

- The Axiom system has a sophisticated programming language for constructing computational domains.

# Computational Models

- A **computational model** is a set of simplification procedures for a theory $T$ that use the data structures and operations of a computational domain $D$.
  - The simplification procedures utilize a bidirectional mapping from part of the language of $T$ to the language of $D$.

- One domain can serve several computational models.

- The IMPS theory h-o-real-arithmetic has two computational models, one for arithmetic over the integers and one for arithmetic over the rational numbers.

# Problem-Solving Procedures

- A **problem-solving procedure** for a set $\mathcal{S}$ of existential formulas of a theory $T$ is an algorithm that, given a problem represented as a formula $\exists\, x \,.\, A$ in $\mathcal{S}$, returns a solution represented as an expression $E$ such that $T \models A[x \mapsto E]$.

- Issues of concern:
  - Multiple solutions: Which solution should be chosen if there is more than one?
  - Representation: How should a set of solutions be represented?

- Techniques:
  - Unification.
  - Logic programming.

# Hybrid Procedures

- A **hybrid procedure** for a theory $T$ is an algorithm that combines decision, simplification, and problem-solving techniques.

- Example: IMPS simplifier.

- Advantages:
  - One big procedure replaces several little procedures.
  - The procedure can provide partial solutions.

- Disadvantages:
  - Difficult to design and implement due to competing goals.
  - The procedure can produce solutions that have both desirable and undesirable attributes.