CAS 734 Winter 2005

05 Styles of Formal Proof

Instructor: W. M. Farmer

Revised: 1 February 2005

Forward Reasoning

- An assertion is proved by reasoning forward from the assumptions to the assertion.

- Forward reasoning is done by:
  - Applying rules of inference or
  - Applying assumptions in the form of implications (**forward chaining**).

- IMPS supports:
  - High-level forward reasoning with its **theory development mechanism**.
  - Low-level forward reasoning with **sequents**.

Backward Reasoning

- An assertion is proved by reasoning backward from the assertion to the assumptions.

- Backward reasoning is done by:
  - Applying rules of inference in reverse or
  - Applying assumptions in the form of implications in reverse (**backward chaining**).

- IMPS support backward reasoning with the **deduction graph mechanism**.

Reasoning by Contraposition

- An implication $A \Rightarrow B$ is proved by proving its **contrapositive** $\neg B \Rightarrow \neg A$.

- IMPS supports reasoning by contraposition with the **contrapose proof command**.

Reasoning by Contradiction

- An assertion is proved by assuming the negation of the assertion and then proving a **contradiction**.

- Reasoning by contradiction is a special case of reasoning by contraposition where the implication has the form $\mathsf{T} \Rightarrow B$.

- IMPS supports reasoning by contradiction with the **contrapose proof command**.

Equational Reasoning

- An assertion is proved by repeated equality substitution.
- Assumptions in the form of universally quantified [conditional] equations are used as [conditional] rewrite rules.
- A proof by equational reasoning looks like

$$E_1 = E_2 = \cdots = E_n.$$

- IMPS supports:
  - Equational reasoning with the **force-substitution proof command** and the **rewrite mechanism** in the IMPS **simplifier**.
  - Conditional equational reasoning with **theorem macetes**.

Algebraic Reasoning

- Let $\mathcal{R}$ be a set of functions that map expressions to expressions called **computation rules**.

- A **computation** in $\mathcal{R}$ is a finite sequence $C = \langle E_1, \ldots, E_n \rangle$ of expressions such that, for all i with $1 \leq i < n$, there is some $r \in \mathcal{R}$ such that $E_{i+1} = r(E_i)$.

- An assertion is proved by creating an appropriate computation.
  - For example, if the rules in $\mathcal{R}$ map expressions to strictly bigger expressions, then $E_1 < E_n$ is proved by a computation $\langle E_1, \ldots, E_n \rangle$ in $\mathcal{R}$.

- Equational reasoning is a special case of algebraic reasoning.

- IMPS supports algebraic reasoning with **compound macetes**.

Existential Instantiation

- An assertion $\exists x \,.\, A$ is proved by constructing an expression c and proving $A[x \mapsto c]$.

- Existential instantiation is **problem solving**.

- **Logic programming** is automated existential instantiation.

- IMPS provides very little support for existential instantiation.

Model Checking

- An assertion is disproved by constructing a counterexample for it or proved by showing that it is true in every model.

- Contemporary model checkers applying model checking to a theory specifying a finite state machine whose models are the possible states of the machine.

- IMPS provides no support for model checking.

Induction

- An assertion in the form of a universal statement is proved by employing an **induction principle**.
  - The induction principle reduces the assertion to a **base case** and an **induction case**.
  - Often the induction principle must be applied to a stronger assertion in order to have a sufficiently strong **induction hypothesis**.

- IMPS supports induction with the **induction command** that applies an **inductor** consisting of:
  1. An induction principle.
  2. Heuristics to handle the base and induction cases.

Ways of Reducing Proof Complexity

- Definitions.
  - IMPS supports several powerful **definition principles** and allows **local definitions** to be created in proofs by using the **cut command** with existential statements.
- Lemmas.
  - IMPS allows theorems inside and outside the theory to be applied directly or via macetes and **local lemmas** to be created in proofs using the **cut command**.
- Computation.
  - IMPS supports several kinds of computation in proofs with **simplification** and the **macetes mechanism**.
- Local contexts.
  - Reasoning in IMPS is systematically performed relative to the **local context**.