CS323 Interactive
November 4, 1999

Otter and PVS [and Cyc]
Aarati Parmar

Outline

- Introduction
- Theorem Prover: Otter
- Proof Verification System: PVS
- Cyc (If we have time)
- Questions, Conclusion, Discussion

Introduction

- Theorem Provers are tools to help
  - write logical theories
  - check them for inconsistencies
  - use logical theories
- Proof verification systems are used to
  - verify specifications
  - check that a system is correct
- Cyc: huge knowledge base

Theorem Provers

- Lots of different kinds, specialized
  - Prolog (PTTP)
  - SNARK
  - Boyer-Moore
  - STEP
- For more see
  http://www-formal.stanford.edu/pub/clt/ARS/systems.html

Otter

- Organized Techniques for Theorem-proving and Effective Research
- Resolution-style TP for FOL with =
- Has lots of different resolution strategies:
  - automatic mode: looks at syntax of theory and decides own strategy
  - or can create your own by setting various flags and specifying certain orderings.

Otter: Example -- Socrates

- All men are mortal
- Socrates is a man
- $\models$ Socrates is mortal?

Otter: Example --BlocksWorld

- Simple proof example (p2)
- Green Style Planning:(p7)
  - prove that there exists a situation where goal holds
  - in proving this fact, the existential will get filled in with a plan for s!
- Sad fact: can't do a 2 step plan (Otter not so good at functional theories)

Proof Verification Systems

- Also at http://www-formal.stanford.edu/pub/clt/ARS/systems.html
- Some examples:
  - Lambda
  - SPIN
  - PVS

PVS

- PVS (http://pvs.csl.sri.com/)
- Used to write formal specs and check proofs.
- Been used (Really!) to
  - proof correctness of some hard fault-tolerant algorithms
  - verify microcode for a microprocessor
  - checking Interval Temporal Logic (ITL) proofs

PVS: Using It

- Documentation at `/usr/class/cs323/pvs-2.3/Doc/`
- First thing to do: change context to directory you are in, then open pvs file
- Format:
  - types
  - variables declarations
  - axioms
  - claims

PVS: Using It

- 5 main proof commands you will use:
  - (skolem!)
  - (inst?) [and variations]
  - (prop)
    - (flatten)
    - (split)
  - (assert) [tell pvs to remember equality information]
  - (lemma)

PVS: Using It

- Uses Gentzen's sequent calculus:
  $a_1 \wedge ... \wedge a_n \mid\!- b_1 \vee \ldots \vee b_n$
- Try to get "a"s to entail one of the "b"s
- Claims example
- YSP Example

Cyc

- Knowledge based developed byCyccorp
- http://www.cyc.com
- Neat stuff!
- Inference example

Questions/Conlusions/Discussion

- If you play with otter, pvs, cyc a bit, you will understand it
- Fun to use!
- A great way to check homework problems!
- Common sense problem page: http://www-formal.stanford.edu/leora/cs/
- Interested in Cyc?