Introduction to Automated Theorem Proving

CS3234
Lecture 2

Martin Henz and Aquinas Hobor
Outline

• What is an automated theorem prover?

• Advantages and disadvantages

• Introduction to Coq
Automated Theorem Provers

• Computer program that can generate and check mathematical theorems

• Theorems are expressed in some mathematical logic, such as propositional logic, predicate logic, first-order logic, ...

• Many different theorem provers out there: Isabelle/HOL, TWELF, Coq, Metamath, Nuprl, ...
Theorem Prover Overview

- Provided by user (not to scale)

- Provided by developer (not to scale)

Math (in some logic)

Theorem Prover

Provided by user (not to scale)
Differences Between Provers

• The logic the prover uses
  – Isabelle/HOL: Higher Order Logic (HOL)
  – TWELF: Logical Framework (LF)
  – Coq: Calculus of (Co-)Inductive Constructions (CiC)
  – etc.

• Some logics are more powerful (can express and prove more theorems) than others, *e.g.*,  
  – Propositional Logic is usually the weakest  
  – CiC is more powerful than HOL

• More powerful logics can be harder to use
Differences Between Provers

• The task a prover handles
  – All provers can check theorems in their logic
  – Automated proof generation is much harder

• Different provers have different trade-offs between degree of automation and the power of the logic they handle
  – The more powerful the logic, the less automatic generation of proofs
  – Many automated theorem provers are really more automated theorem checkers
### Proof Checking vs. Proof Generation

- Recall that a formal proof is a list of formulas each of which is justified by an axiom or an inference rule applied to earlier formulas

<table>
<thead>
<tr>
<th>Formulas</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>Axiom</td>
</tr>
<tr>
<td>F2</td>
<td>Rule 3 and F1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Theorem</td>
<td>...</td>
</tr>
</tbody>
</table>

- Formal proofs are very easy to check mechanically
  - Just make sure the justifications are applied correctly

- However, proof generation is harder – have to generate a list of formulas, each of which has valid justification, and where the last formula is the desired theorem.
Differences Between Provers

• Practical differences
  – Or, “Oh, yeah – it’s a piece of software!”

• Some provers have:
  – More support if there is trouble (mailing lists)
  – Bigger user base
  – More frequent new versions
  – Tool support
  – Better library support (e.g. built-in definitions of the real numbers, etc.)

• It’s possible to have bugs! Which are more trustworthy?
What does the user provide?

• Depends on which prover!

• All provers: statement of theorem expressed in the logic of the system

• For some provers, this is enough – all you do is give the desired theorem and push “Go”

• Fully automatic provers can’t prove nearly as many theorems as “semiautomatic” provers
What does the user provide?

• So the user must provide some kind of hints that help the prover (often provided in the same file)

• Least useful hint: “A proof exists – search forever until you find it”

• Most useful hint: “Here is the proof: …”
Intermediate Hints

• Most provers take a middle path and require hints between the two extremes
  – Statement of key lemmas (useful intermediate results)
  – Proof outline (how the lemmas connect)
  – Key idea in proof (“prove by induction on n”)
  – Proof script (list of medium-sized steps in the proof)

• One advantage of providing hints is that if the theorem is not provable, the prover can provide better error reporting as to why the proof failed.
  – Most error reporting is still pretty difficult (worse than a typical compiler error report)
Theorem Prover Overview

Provided by user (not to scale)

Statement of Theorem

Hints

Provided by developer (not to scale)

Theorem Prover

12
Library

Many theorems share commonly used definitions and lemmas

• Natural numbers
  – Definition (zero & successor)
  – Facts about naturals (a + b = b + a) and their proofs

• Integers
  – Definition (naturals & negative naturals & zero)
  – Facts about integers (e.g., a + (-a) = 0) and their proofs

• etc.
Theorem Prover Overview

Provided by user (not to scale)

- Statement of Theorem
- Hints

Provided by developer (not to scale)

- Theorem Library
- Theorem Prover
Outline

• What is an automated theorem prover?

• Advantages and disadvantages

• Introduction to Coq
Why are theorem provers used?

• Very high assurance due to mechanical checking

• When possible, automatic proof generation can significantly improve program development
  – Earlier detection of bugs
  – Better code/design coverage than testing
  – Frequently tools can locate errors faster than debugging
High assurance

• In general, highest assurance that there are no mistakes in proof
  – Checkers are very thorough: don’t get tired, don’t get bored, don’t make mistakes
  – If anything, the problem is the opposite – trying to convince a checker that a true thing is true can be frustrating.

• Used in areas where correctness is critical
  – Aerospace
  – Defense
  – etc.
Right tool for the job...

Better at some kinds of tasks than others

- **Best:** proving behavior of real programs

- **Bad:**
  - Cryptography: often we rely on guesses ($P = NP$ ?)
  - Pure math: Tools and libraries are not practical yet
  - Design: How to prove one user interface better than another?
Uses for proof generation

For problems that are simpler, proof generation is very useful as well

• Type inference & checkers (e.g., ML, Java, C#)

• Safety of web applications (e.g., Java)

• Static analysis tools
  – Buffer overrun analysis
  – Safety property analysis
Disadvantages of Automated Theorem Proving

For proof generation:

- Only useful for certain kinds of “simple” problems
- Tools are frequently very difficult to develop
- Often can have very bad worst-case running time
  - e.g., Hindley-Milner type inference is $O(2^n)$
  - Sometimes the average running time is much better
Disadvantages of Automated Theorem Proving

For proof checking:

• Developing the hints / proof by hand can be very labor-intensive

• It can be very difficult to formalize correctness
  – “correct” operating system?
  – “correct” web browser?
  – “correct” compiler?

• Learning curve to use systems can be steep
One more advantage... they are fun to use!

• A bit like writing software in a scripting language

  “Building such scripts is surprisingly addictive, in a videogame kind of way...”
  - Xavier Leroy

• The advantage of never having to worry about bugs in the finished product

• Can work on math at 3 AM without fear
Outline

• What is an automated theorem prover?

• Advantages and disadvantages

• Introduction to Coq
Coq

• Theorem prover developed in France
  – Name is the French word for rooster (The French have a distressing lack of regard for the way their words sound in other languages.)
  – Lots of library & tool support
  – Large user base

• Calculus of (Co-)Inductive Constructions (CiC)
  – We will just use a small portion

• Available on the web at http://coq.inria.fr/
  – Windows, MacOS, Linux/UNIX
Tactic-based system

• CiC is quite powerful, so automatic proof generation is quite limited

• Instead, a user provides hints in the form of proof scripts

• Proof scripts are lists of tactics, which guide Coq in generating the proof
CoqIDE
CoqIDE

Proof Script

Current Goal

Error Reporting
Demonstration

Note: this presentation as well as the script file we will go over now will be available to you online.
Homework, due next week

A file has been added to the online course workbin.

Hints:

• **Start early.** I am very unlikely to help with installation problems the night before the homework is due. Also, the learning curve for Coq can be steep.

• I am available for help most afternoons. Feel free to knock on my door or write me email if you have questions.

• You are encouraged to go over the script we did in class before starting the homework.
Homework, due next week

A file has been added to the online course workbin.

Hints:

• **Start early.** I am very unlikely to help with installation problems the night before the homework is due. Also, the learning curve for Coq can be steep.

• I am available for help most afternoons. Feel free to knock on my door or write me email if you have questions.

• You are encouraged to go over the script we did in class before starting the homework.

If I could make it flash bright orange I would.
Individual Work Only

Because of the nature of the machine-checked part of this assignment, you should not collaborate with any of your classmates. The handwritten part of the assignment can be discussed with your classmates.

Do the machine-checked part on your own.

If you have questions about this, please email me.
Hints

1. **Start Early**
2. We have a local copy of the downloads (much faster!) on the website
3. Email Aquinas with questions if you get stuck
4. Tutorials:
   2. [http://cel.archives-ouvertes.fr/docs/00/33/44/28/PDF/coq-hurry.pdf](http://cel.archives-ouvertes.fr/docs/00/33/44/28/PDF/coq-hurry.pdf)