Isabelle and Proof General: Preview

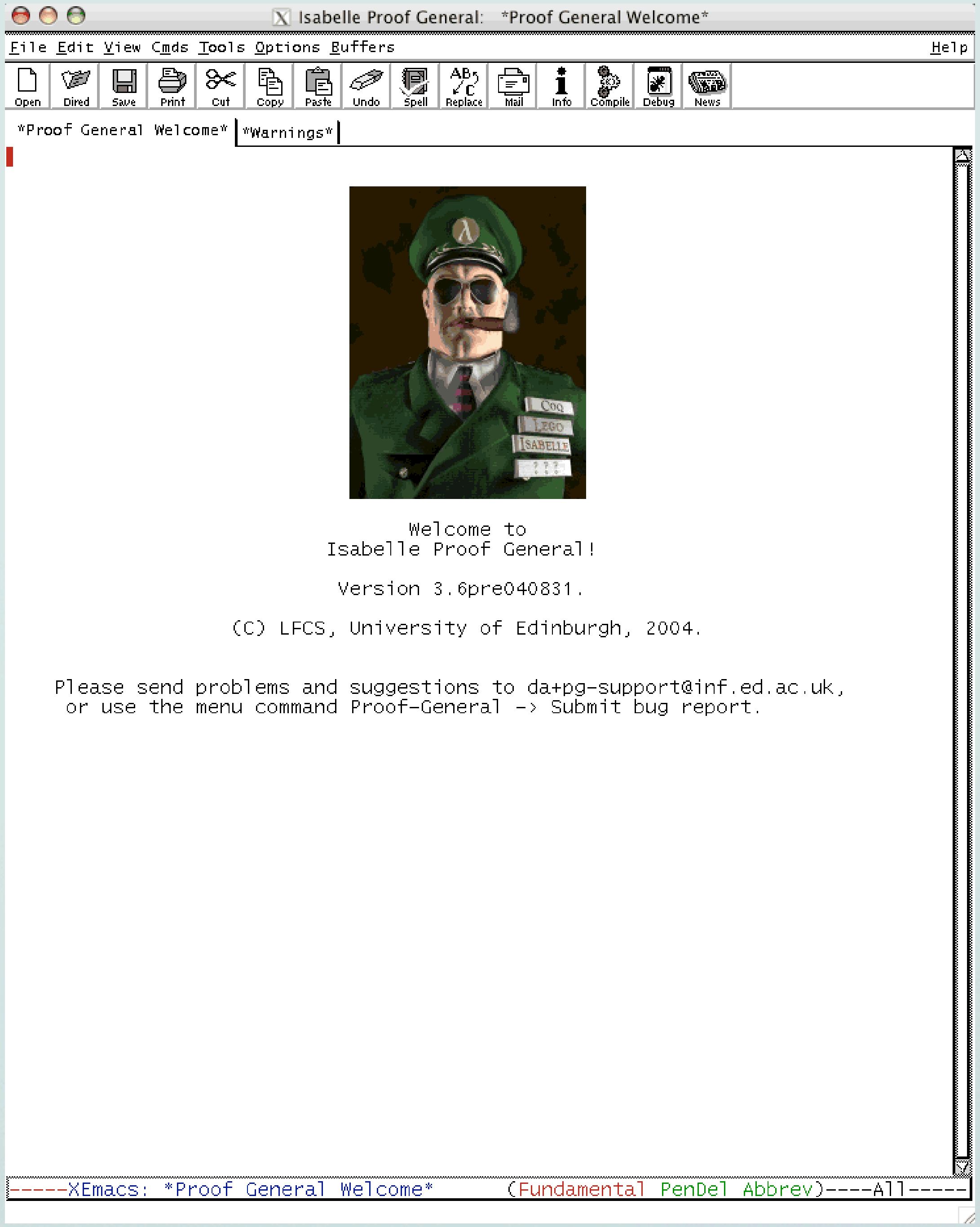
Lawrence C. Paulson
University of Cambridge

Getting Started

- Install Isabelle, following instructions on the download page.
- Install Proof General.
- Proof General requires the editor XEmacs to be installed.
- If you have not used XEmacs before, practice on plain text files before attempting proofs!
- Launch Isabelle from the command line.
- Here, Isabelle has been installed at `/usr/local` and is used to open one of the standard theories.

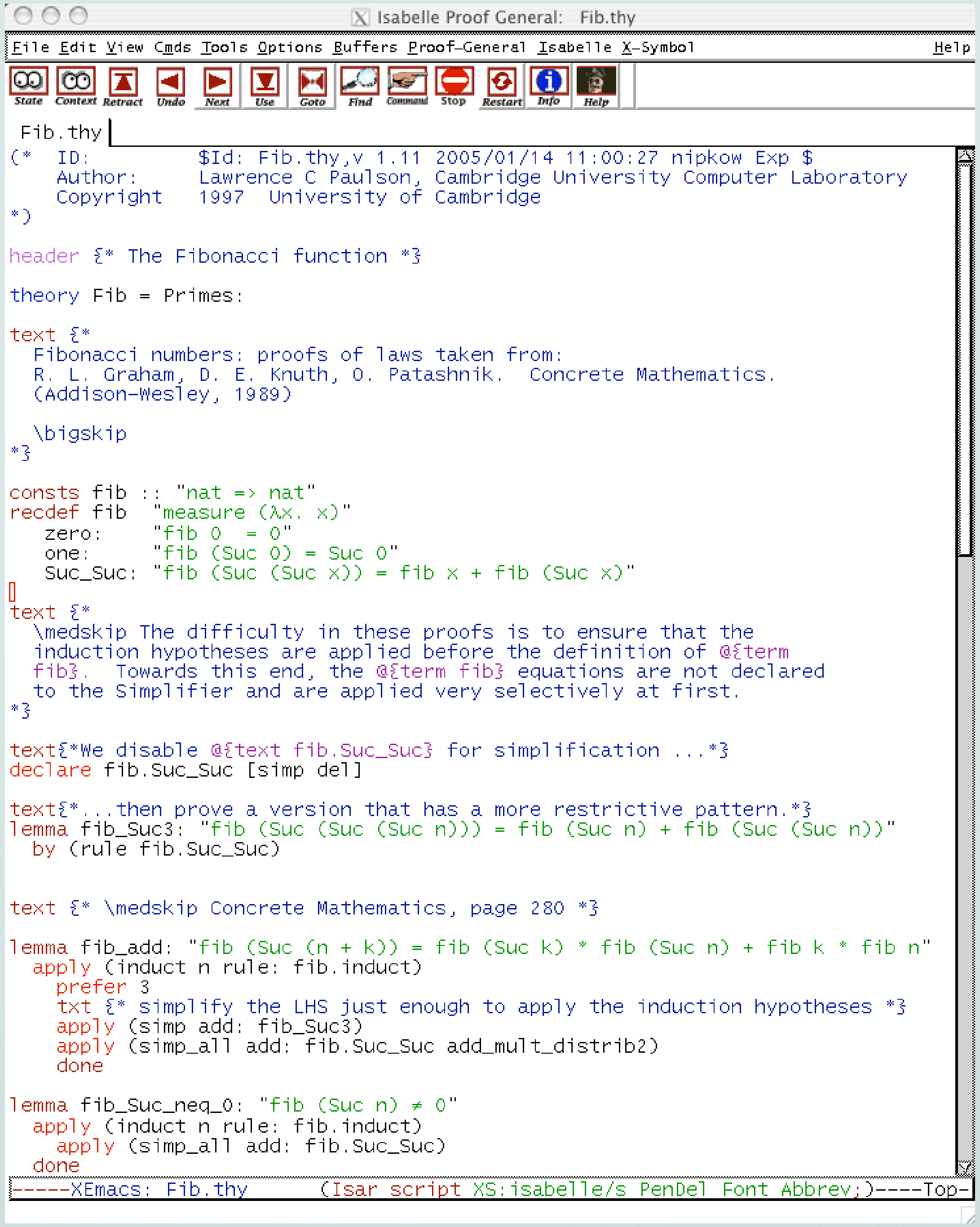
Launching Isabelle at the UNIX command line

Proof General



- Proof General launches within XEmacs.
- If you don't see this splash screen, Proof General is not correctly installed.

The Theory File

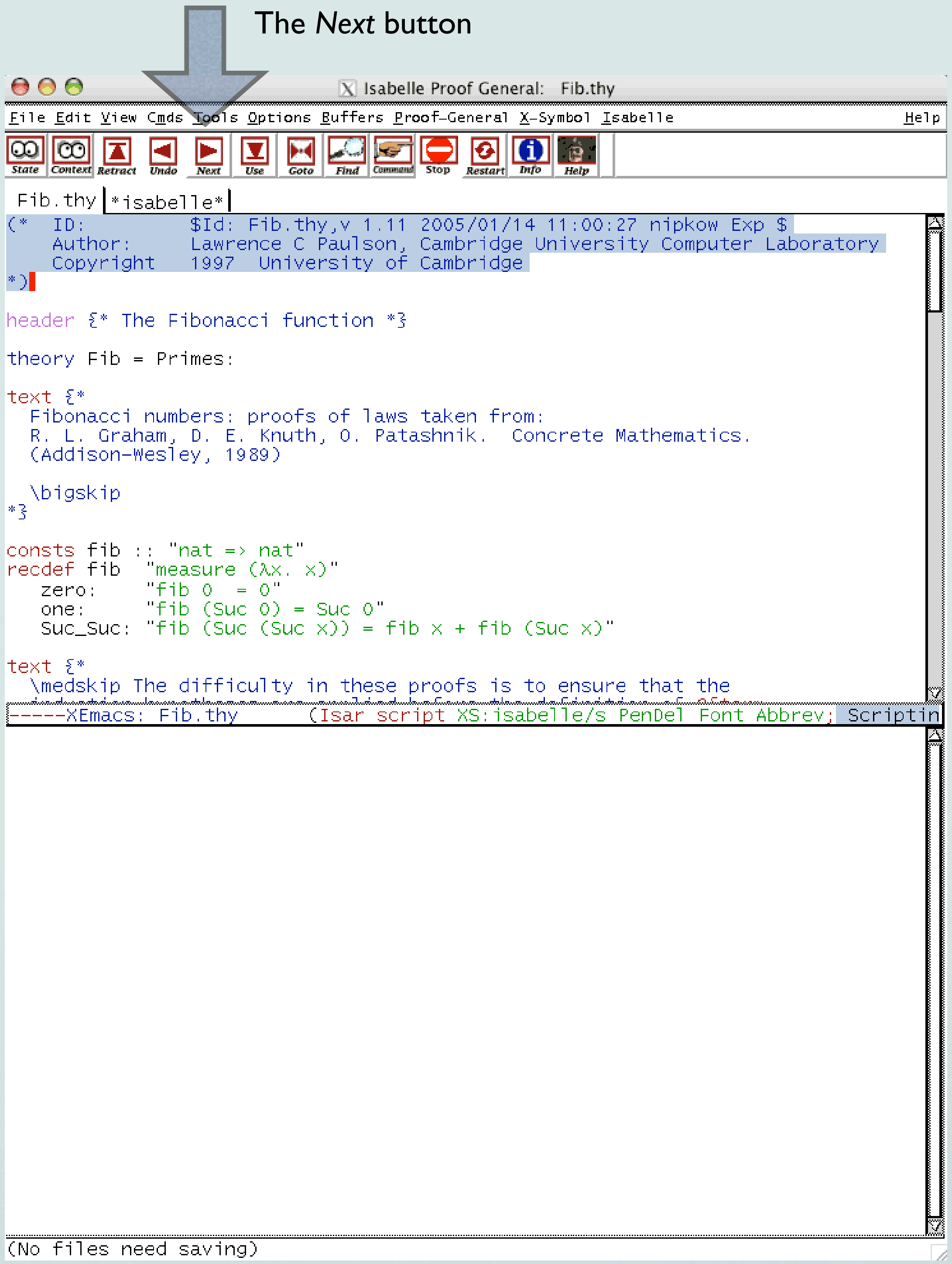


- The theory opens in Proof General.
- Theory files visited from XEmacs also open in Proof General.
- Syntax colouring distinguishes constants, types, keywords, etc.
- The toolbar gives quick access to basic proof operations.
- This theory defines the Fibonacci function and proves theorems about it.

Basic Navigation

- A theory file contains definitions, proofs, LaTeX markup, and general commands.
- Clicking on *Next* starts Isabelle and processes the first item: a comment.
- Repeated clicks on *Next* step through the definitions.
- Proof General highlights material that has been processed in blue.

The *Next* button



Jumping Ahead

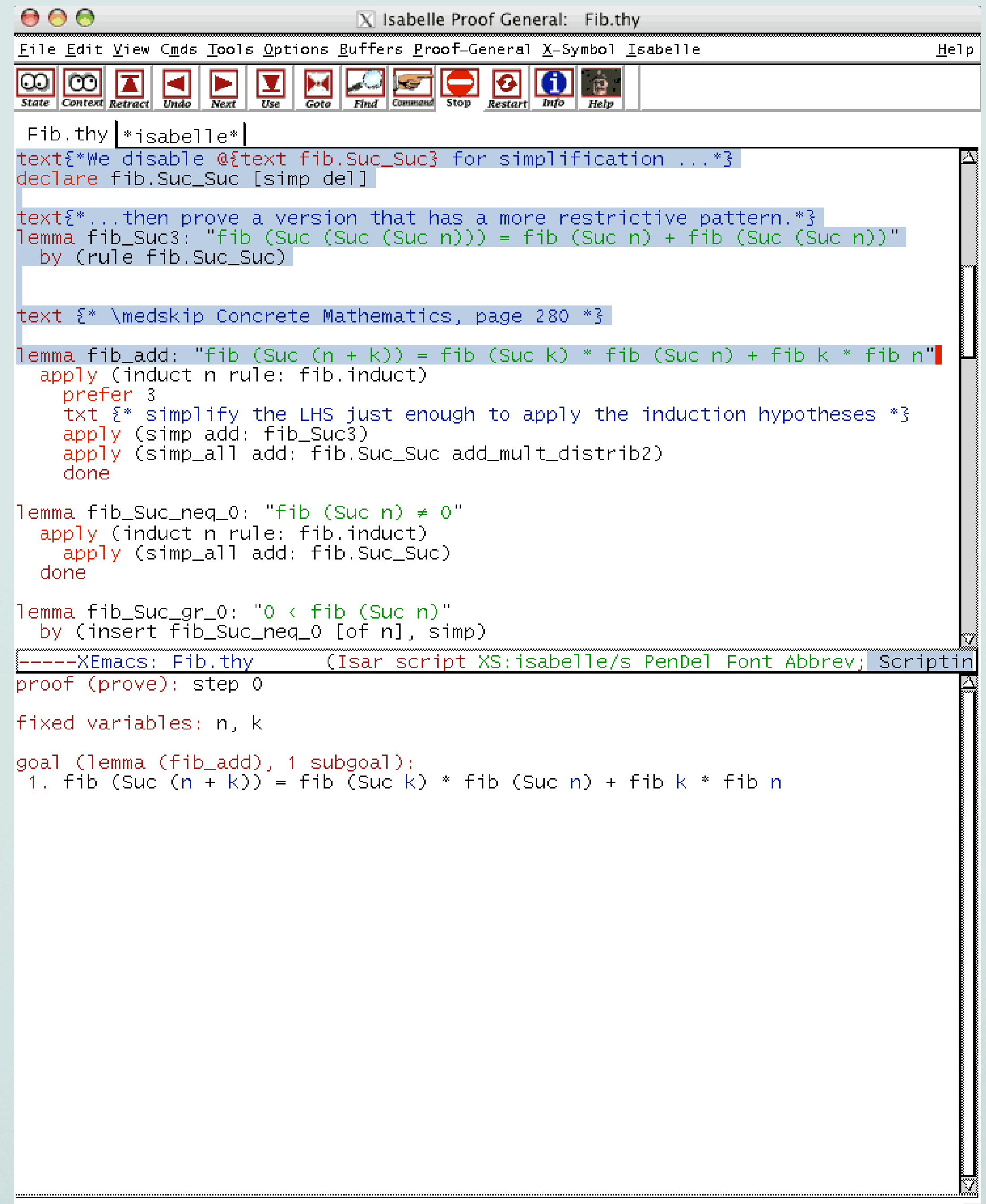
The *Undo* button

The *Goto* button

- You can click anywhere in the theory and then click on *Goto*.
- *Goto* can even go backward, undoing declarations and commands. (To undo a single command, use the *Undo* button.)
- The `header` command is processed quickly, but the `theory` command refers to another theory.
- While Isabelle is working, Proof General highlights the corresponding text in pink.

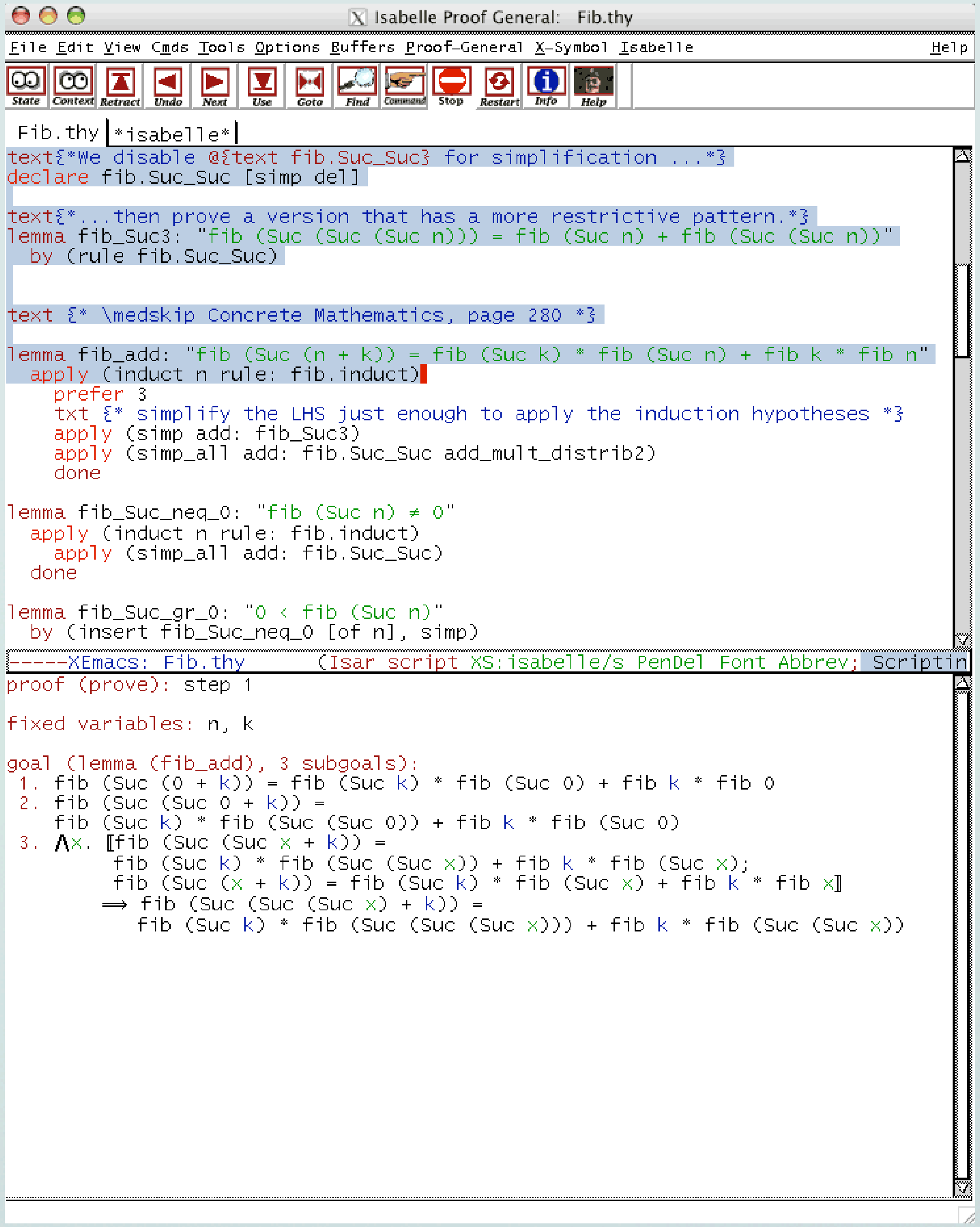
Running a Proof

- We are about to replay a small proof relating the Fibonacci function, addition and multiplication.
- Processing the `lemma` command displays one subgoal in the proof window.
- The commands `lemma`, `theorem` and `corollary` are essentially equivalent.

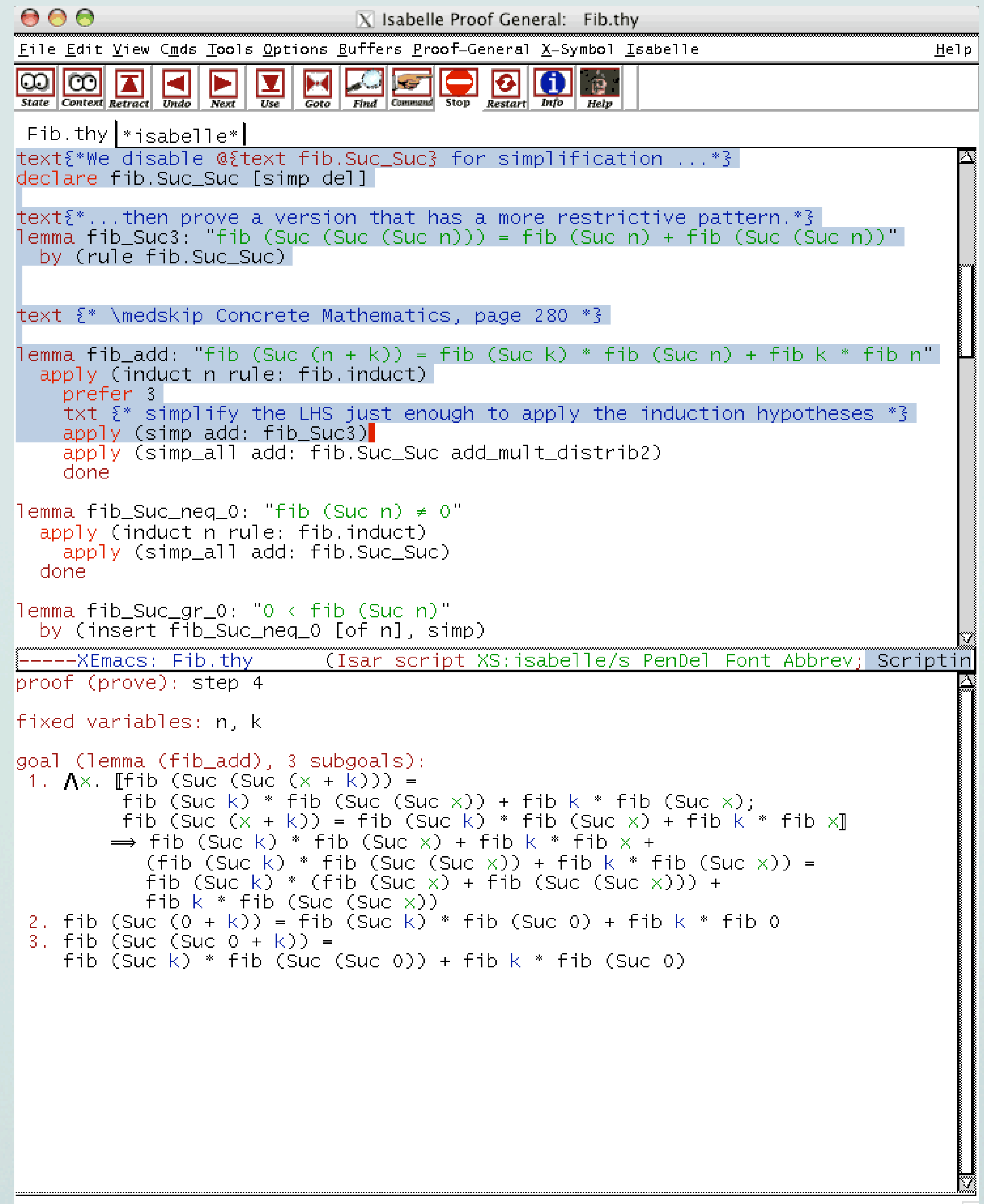


Performing Induction

- The first command performs induction on *n* using the rule `fib.induct`.
- Isabelle produced this rule while processing the recursive definition of the Fibonacci function.
- The proof window now displays three subgoals.



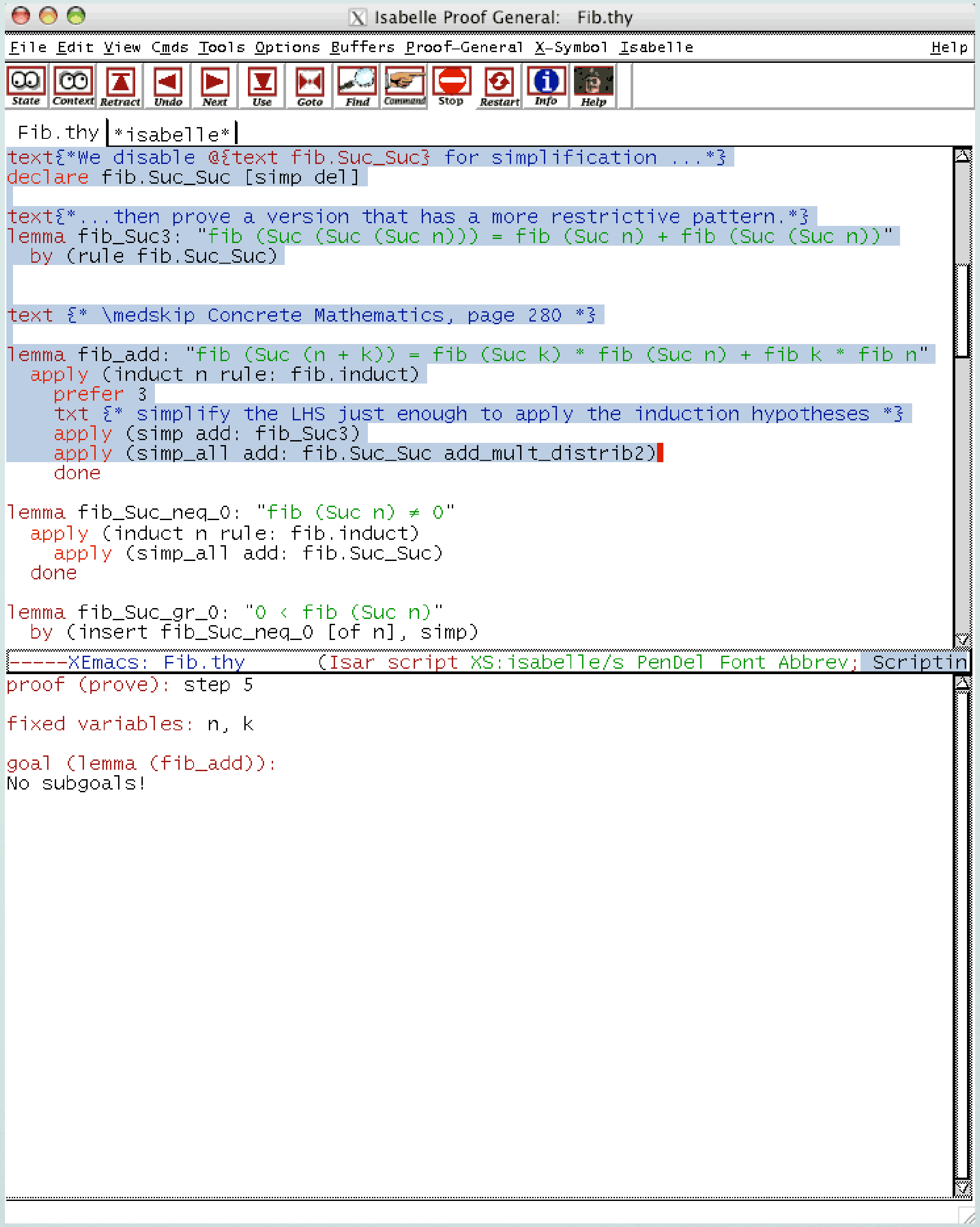
A Rewriting Step



- The third subgoal is selected: `prefer 3`.
- Then, it is simplified with the help of a rewrite rule called `fib_Suc3`.
- This subgoal is still rather complicated!

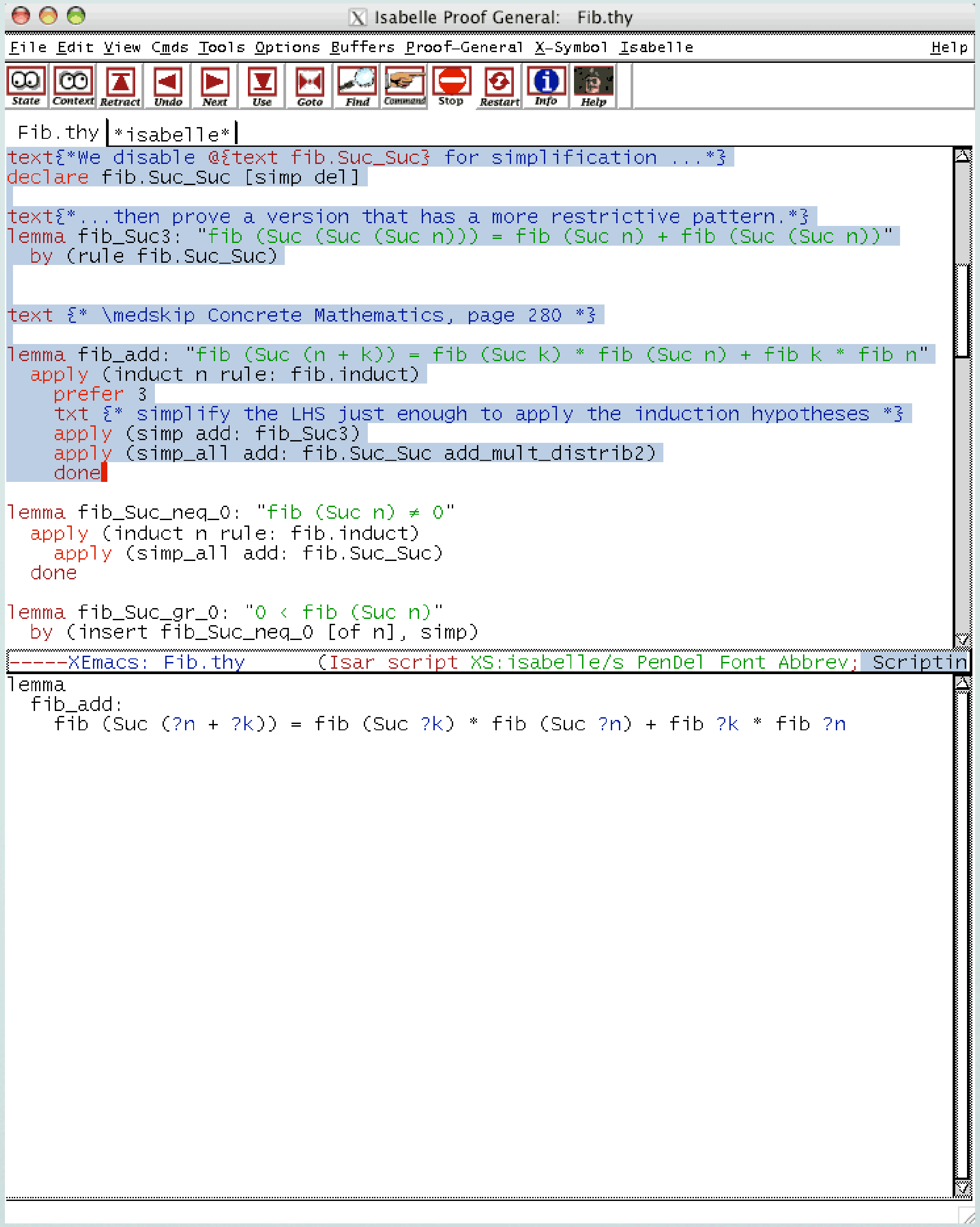
Finishing the Proof

- Next, all three subgoals are simplified, with the help of the rewrite rules shown.
- The simplifier automatically includes hundreds of other rewrite rules, as well as various decision procedures.
- This time, no subgoals remain.



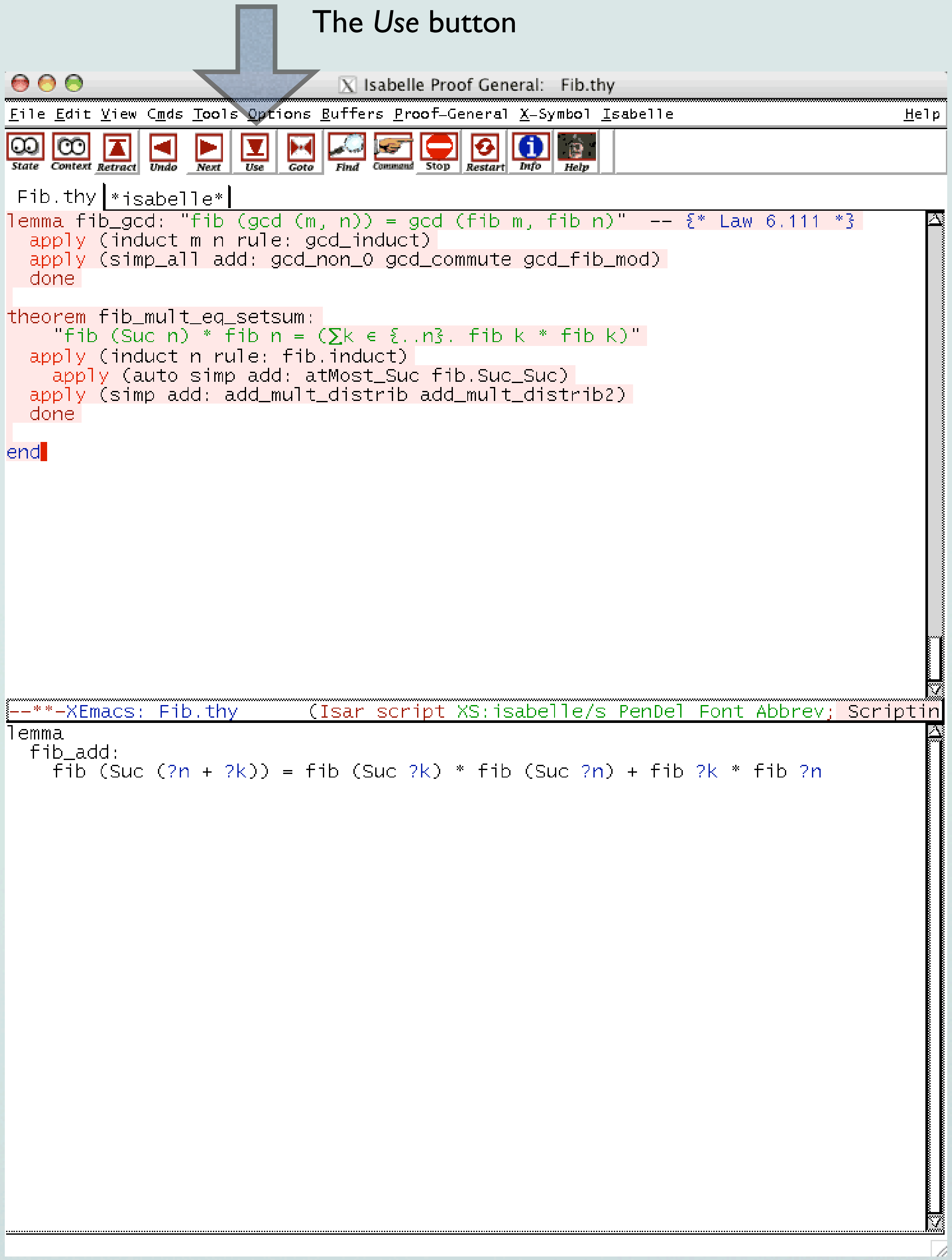
Storing the Theorem

- The done command causes Isabelle to accept the proof, storing the theorem.
- If you were proving this theorem for the first time, you would try various commands right in the editor buffer. You would use *Undo* frequently!
- Once you have succeeded, the file will hold the final version of your proof.
- Using *Undo* on a done command moves the cursor above its proof. Isabelle "forgets" the theorem.



Processing a Theory

The *Use* button

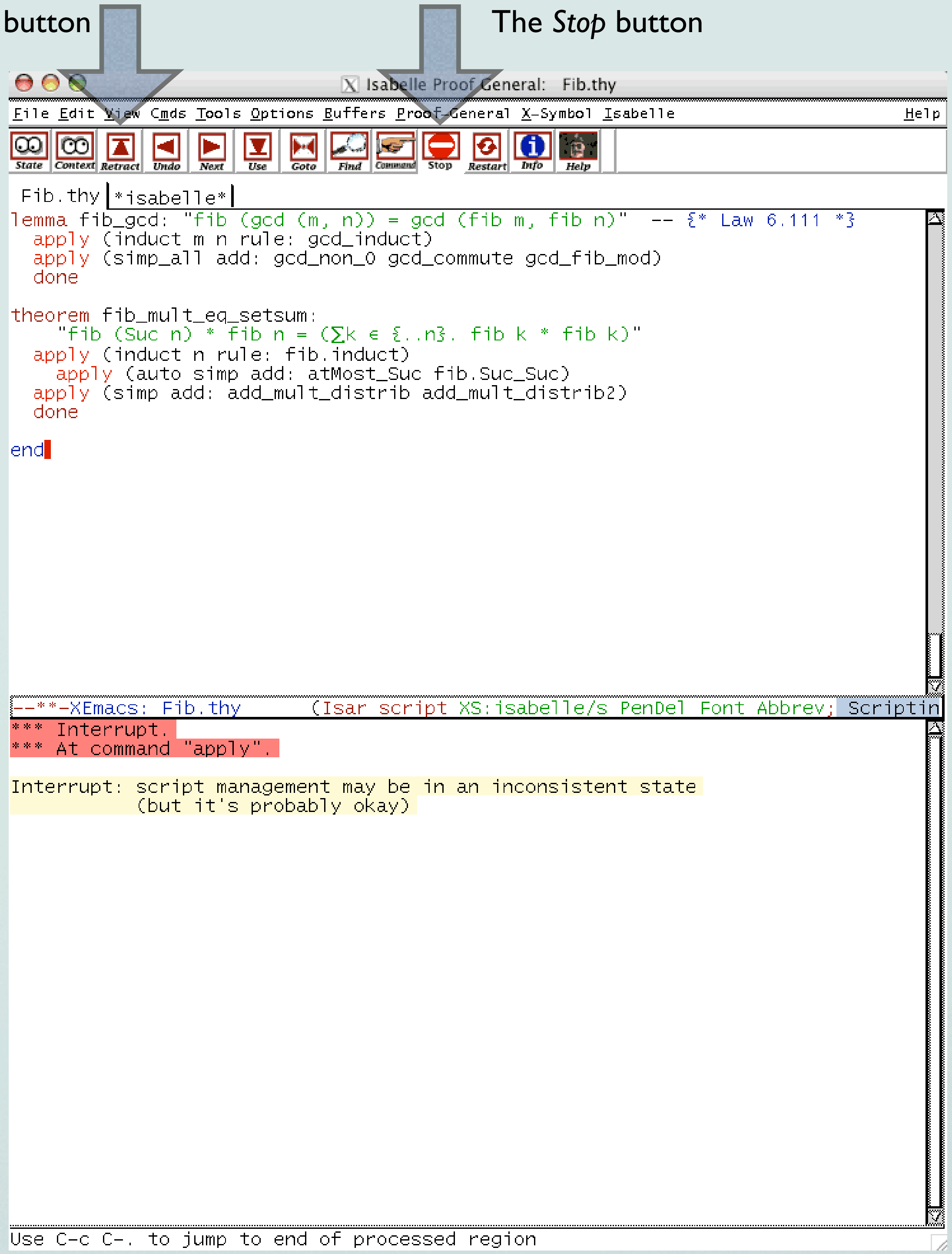


- To run a theory right to the end, click on the *Use* button.
- Now the rest of the theory appears in pink until Isabelle can process it.

Stop!

The *Revert* button

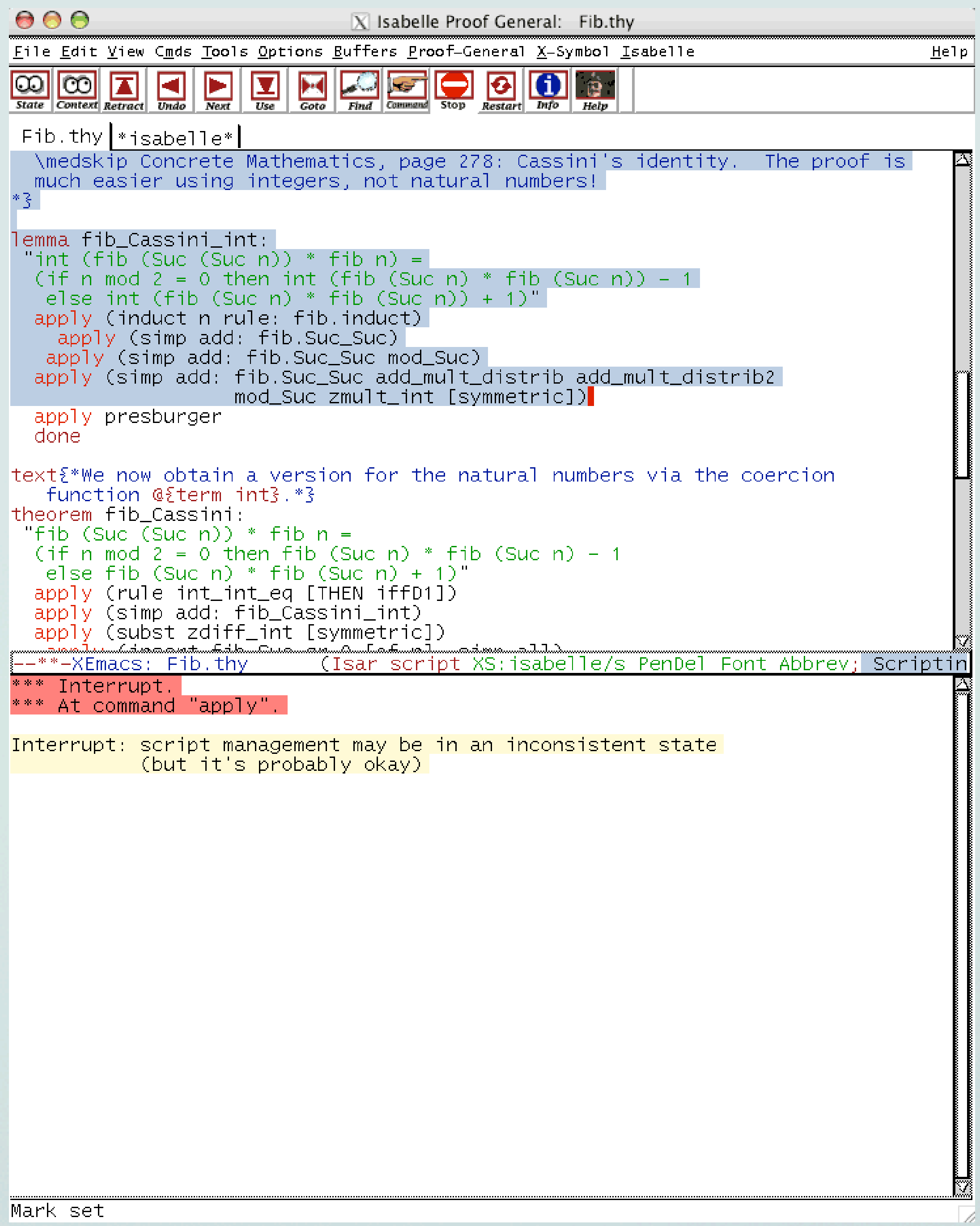
The *Stop* button



- Proof taking too long? Simplifier's looping? Clicked the wrong button? Just click on *Stop*.
- If things behave weirdly after this, perhaps Proof General has got out of sync with Isabelle.
- To get back into sync, use *Goto* to go back to the start of the current proof.
- You can use *Revert* to go back to the top of the theory file.

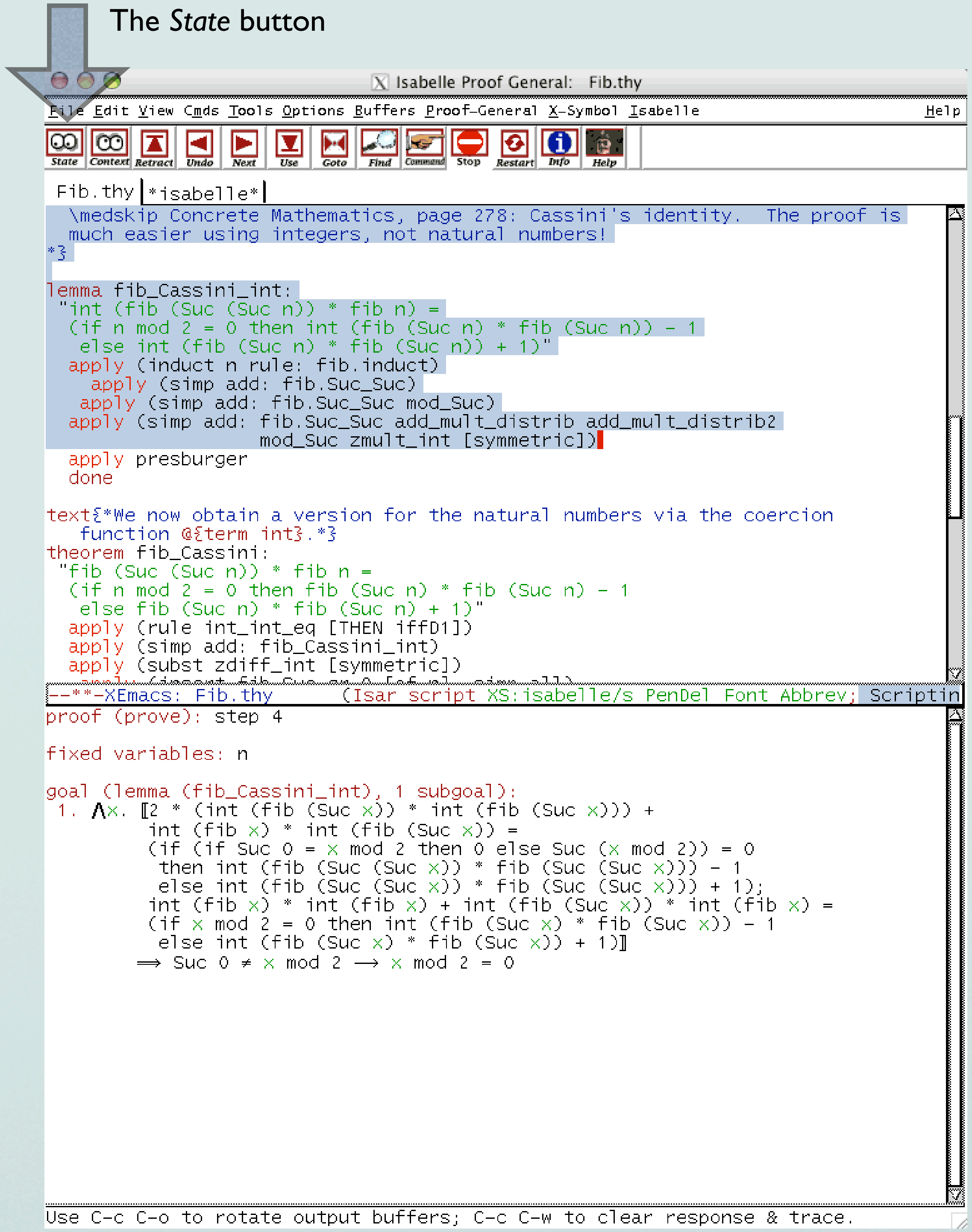
Where Am I?

- If a proof fails—or is interrupted—in a long theory file, how do we locate the critical spot?
- You could simply scroll through the file until you find the end of the blue region.
- To jump right there, use the menu item `Proof General > Goto Locked End`. The key combination `CTRL/C-.` does the same thing.
- The proof was interrupted during a call to `presburger`, an arithmetic decision procedure.



The Proof State

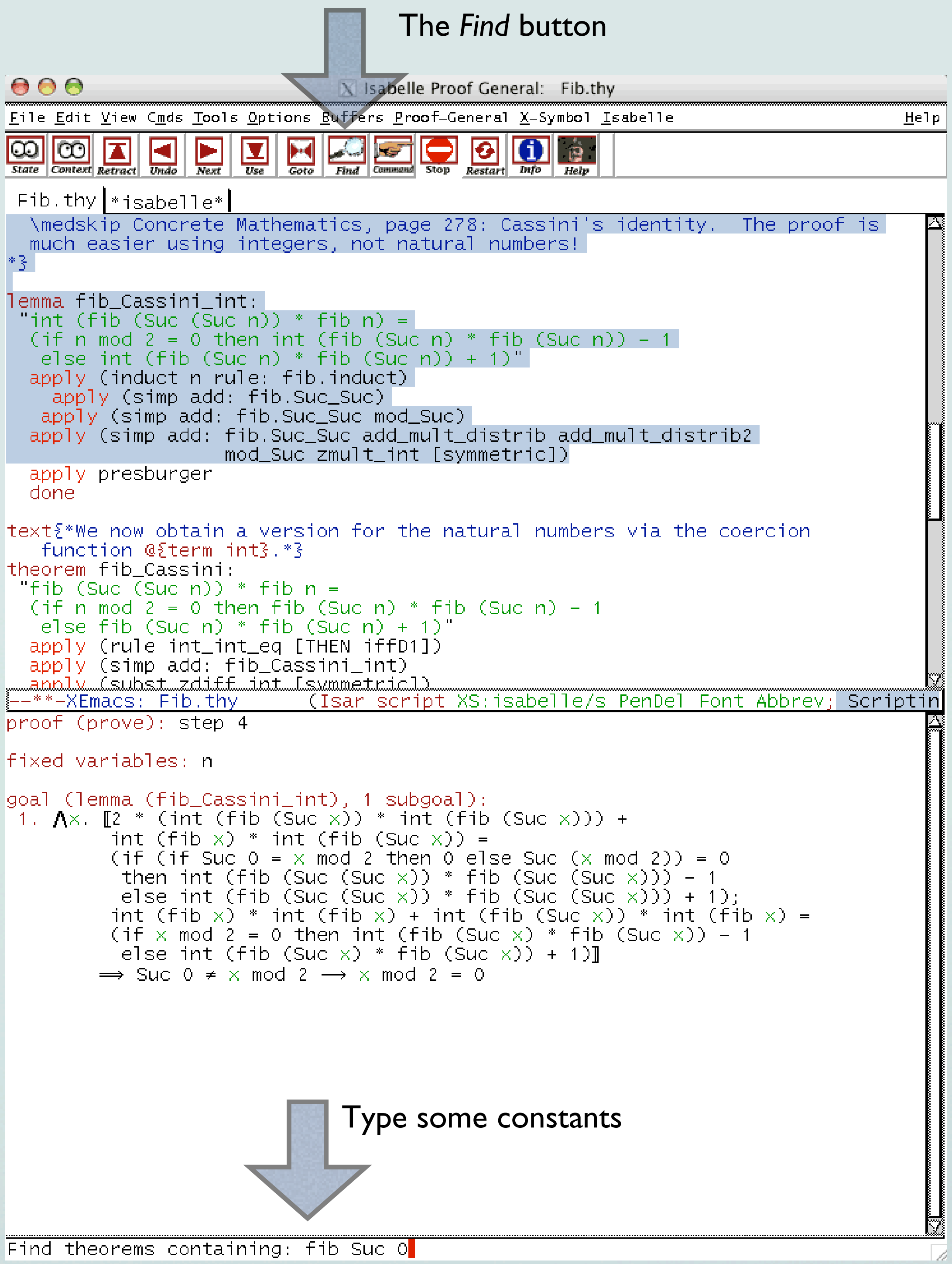
The *State* button



- Clicking on the *State* button reveals the proof state at the given point.
- Here, there was one subgoal left when the proof was interrupted.

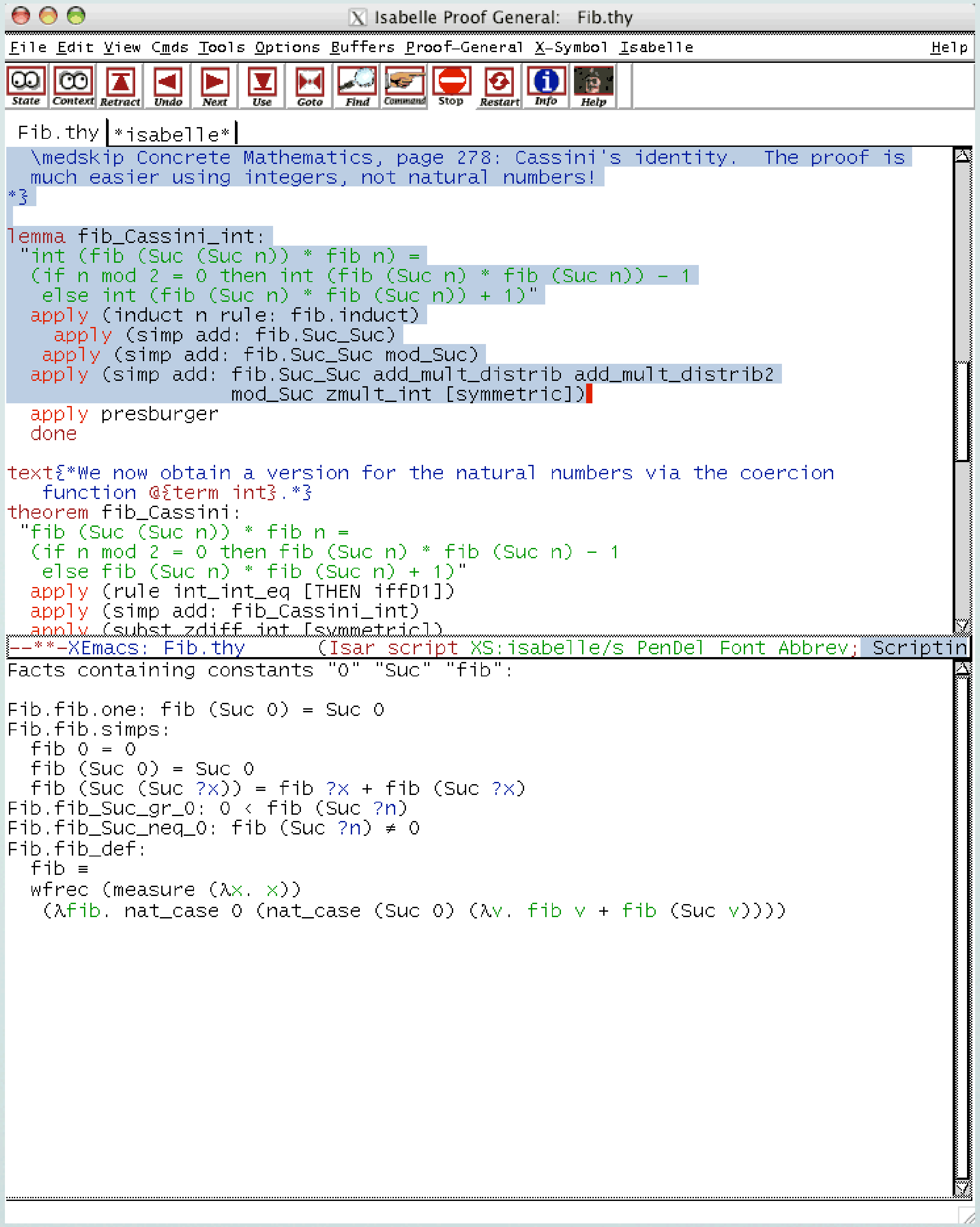
Finding Theorems

- Isabelle provides thousands of lemmas. How do you find the ones you need? One way is to click the *Find* button.
- Then, type some constants—or entire terms—into the XEmacs minibuffer.
- Type the term "(_+_)*_ = _", including the quotation marks, to see all theorems containing an instance of this pattern.
- The pattern "_+_" matches all terms containing the infix operator + because _ matches any term.



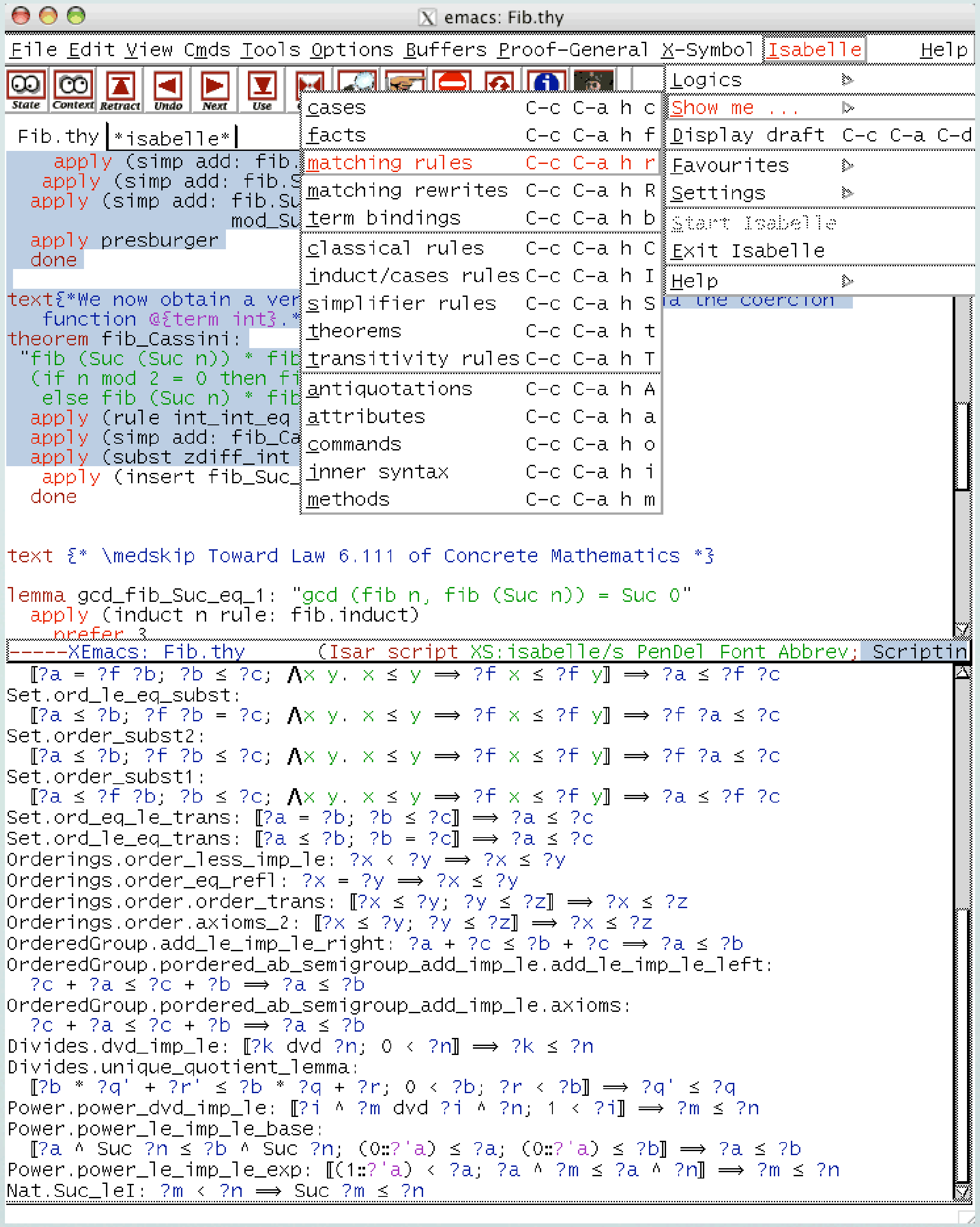
Theorems Found

- The response buffer lists the theorems containing *all* of the listed constants.
- If you are lucky, there will be just a few rather than hundreds!
- The more patterns you type, the fewer theorems will be displayed.
- During the search, variables mentioned in the current goal are viewed as constants.



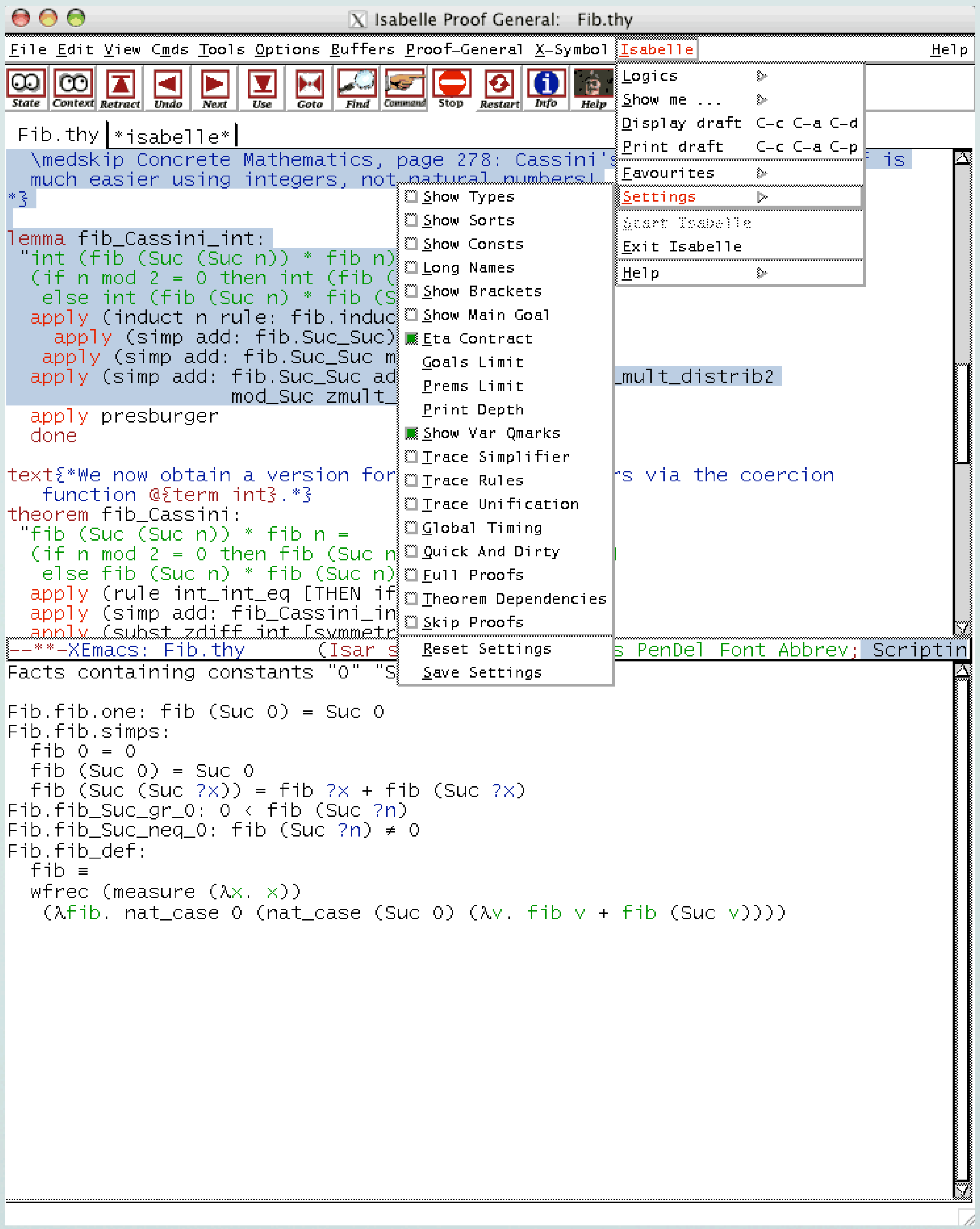
The Isabelle Menu

- The `Isabelle` menu gives access to Isabelle commands and information.
- `Isabelle > Show me...` provides other ways of finding theorems: `matching rules` and `matching rewrites`.
- In the example, the current subgoal has the form $x \leq y$, and `matching rules` displays all known theorems that can prove a conclusion of that form.



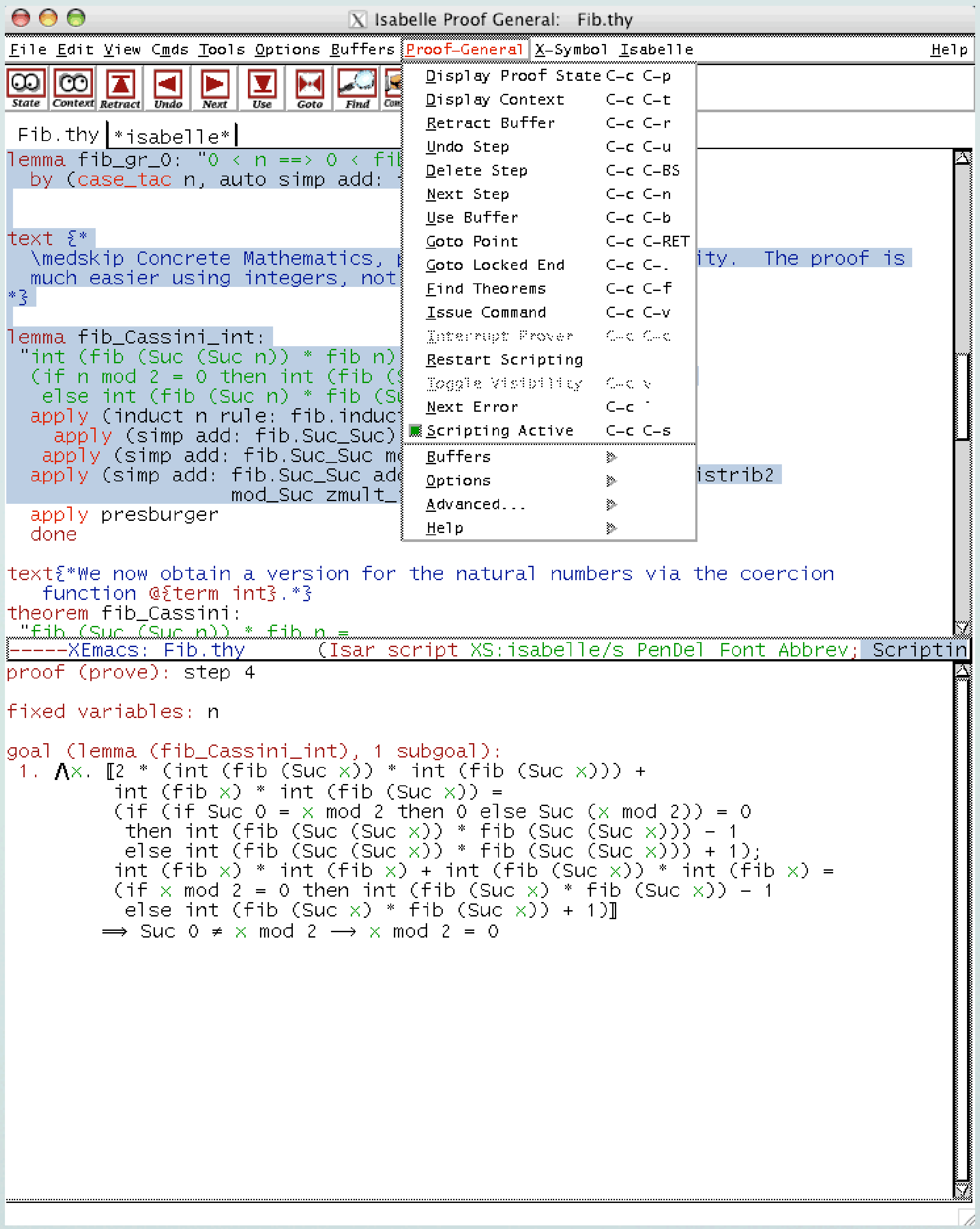
Settings

- The menu `Isabelle > Settings` can request the display of types, execution times, and various traces.
- There are printing options to suit special situations, such as enormous subgoals.
- Use `Show Types` and `Show Sorts` to cause more type information to be displayed.
- The various `Show` options make the output more verbose, but more explicit, and are helpful for diagnosing problems.



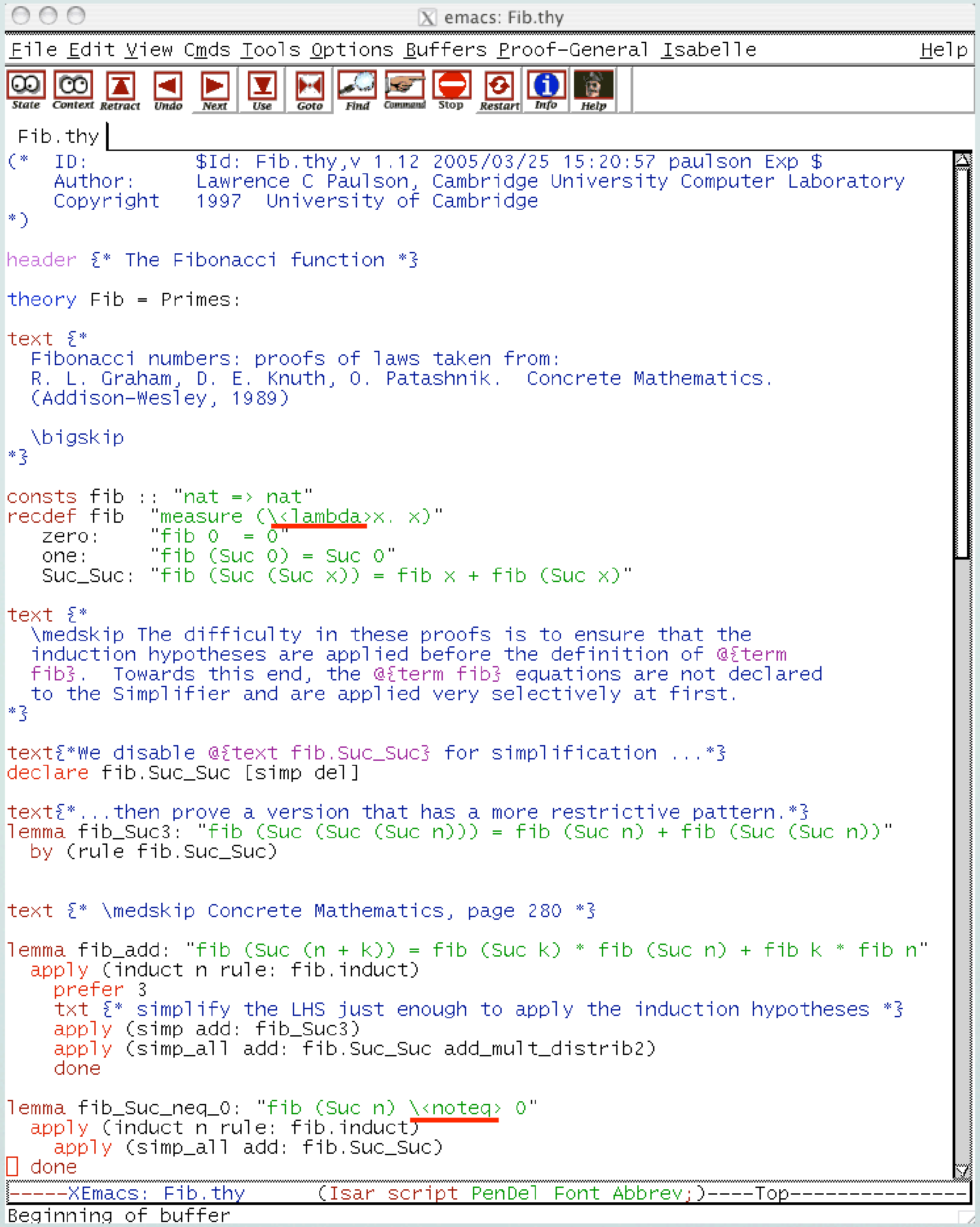
The PG Menu

- The `Proof General` menu gives access to many commands.
- The main commands are available from the toolbar. A notable exception is `Goto Locked End`.
- Choose `Proof General > Buffers > Trace` to see tracing output.



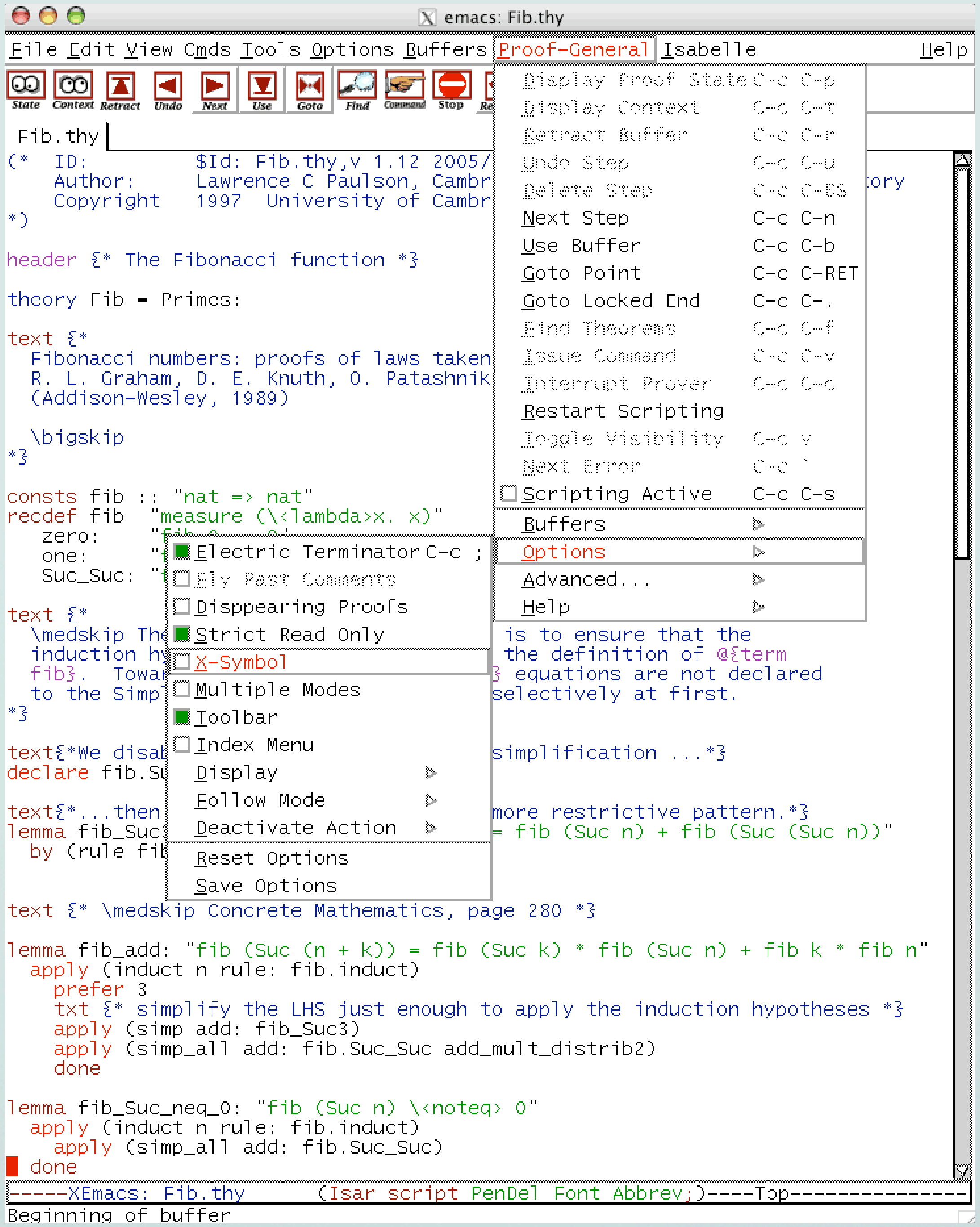
Mathematical Symbols

- Proof General uses the X-Symbol package to display mathematical symbols such as λ ≤ ≠ ∈ ∉ ∪ and ∩.
- The package is included with Proof General, but may need to be switched on.
- If X-Symbol mode is off, Proof General will display ASCII escape sequences, as shown on the right.



Enabling Symbols

- To enable X-Symbol mode, select the menu item `Proof General > Options > X-Symbol`.
- Then, make this setting permanent using `Proof General > Options > Save Options`.
- Take the time to explore the many other options and settings on offer.



Go Forth and Prove!

- Try out this theory yourself: you will find it in `src/HOL/NumberTheory/Fib.thy`.
- For more information on Isabelle, read the documentation.
- For more information on Proof General, see its user manual.
- Have fun!