

Interactive Formal Verification

5: Logic in Isabelle

Tjark Weber
(Slides: Lawrence C Paulson)
Computer Laboratory
University of Cambridge

Logical Frameworks

Logical Frameworks

- A formalism to represent other formalisms

Logical Frameworks

- A formalism to represent other formalisms
- Support for *natural deduction*

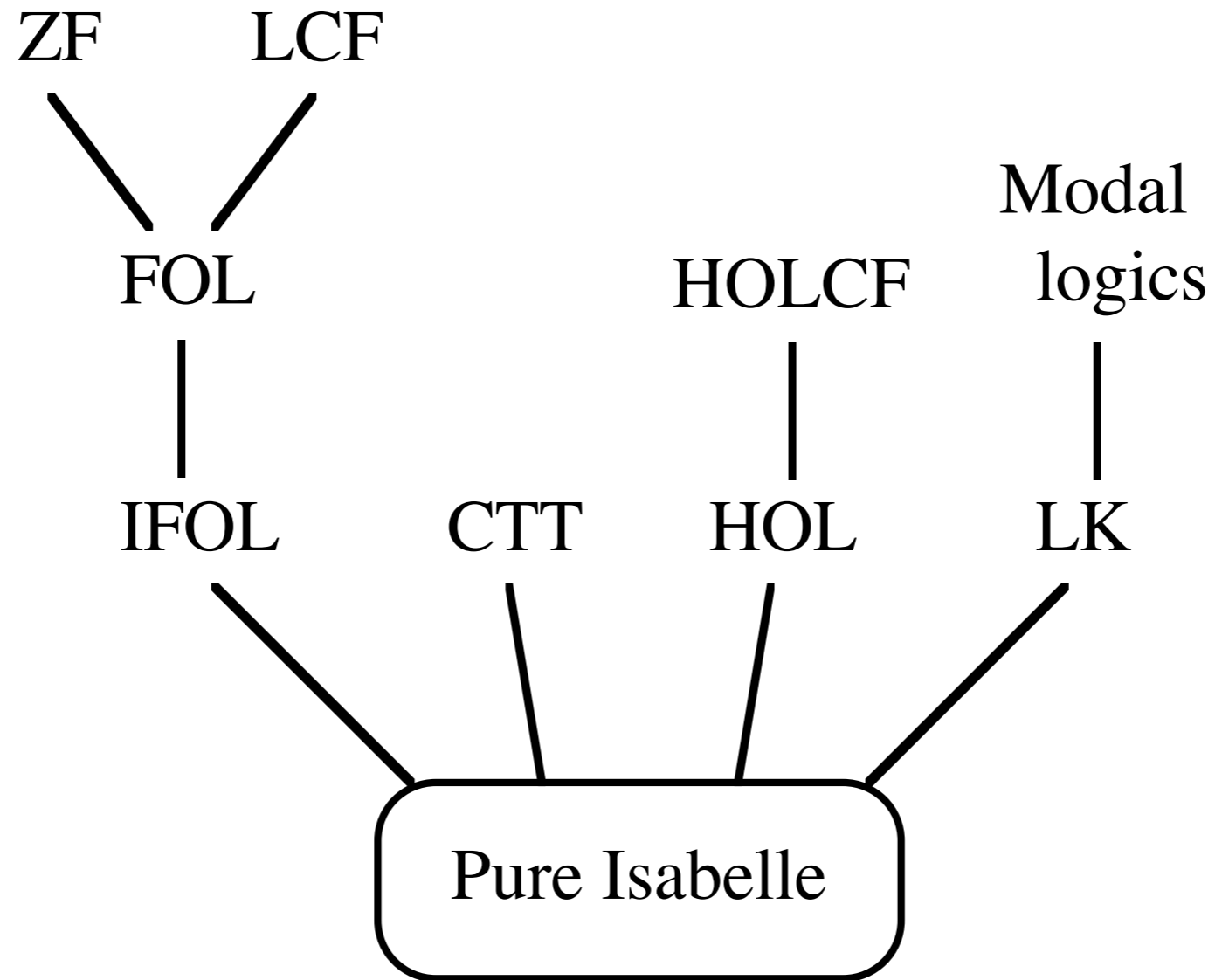
Logical Frameworks

- A formalism to represent other formalisms
- Support for *natural deduction*
- A common basis for implementations

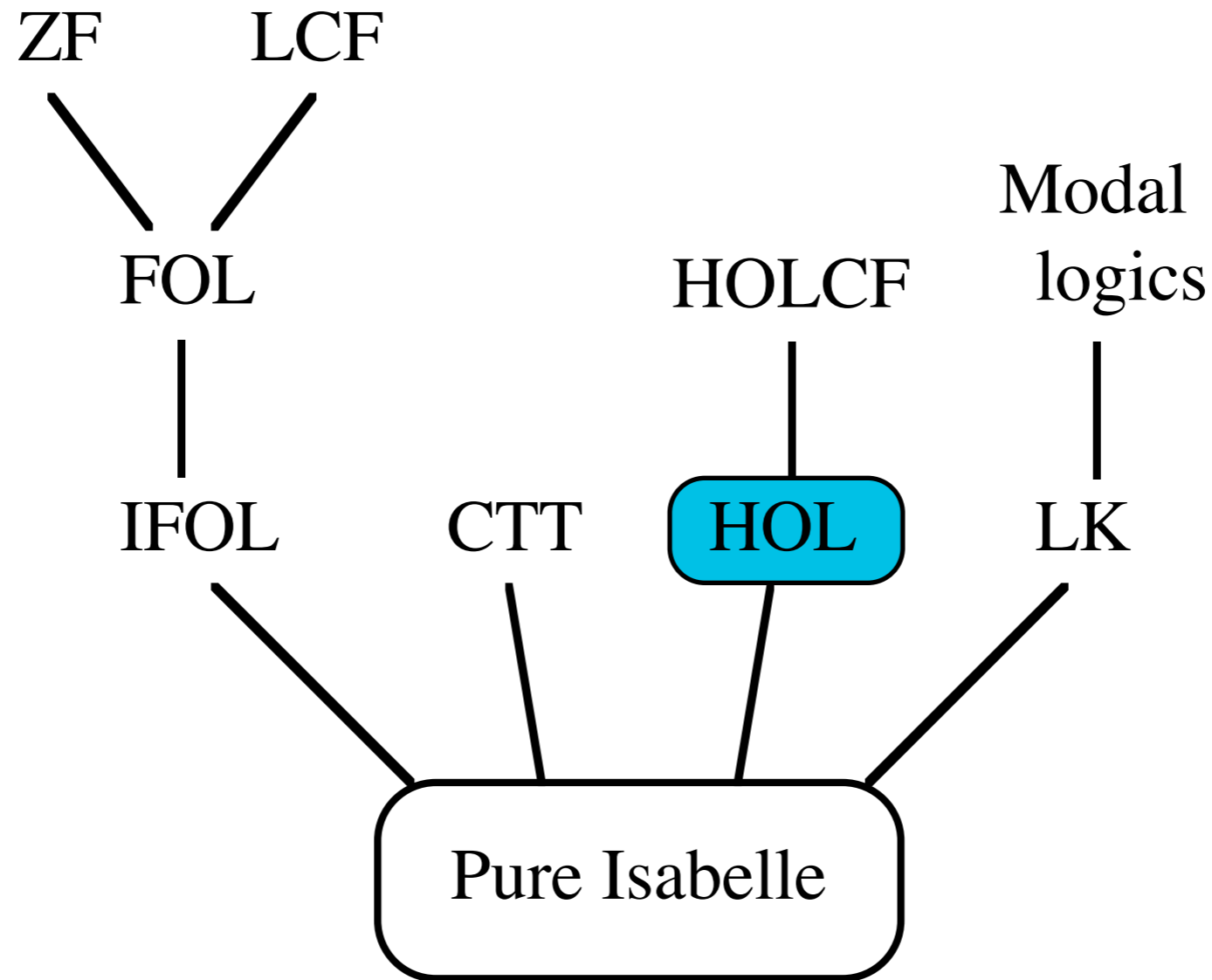
Logical Frameworks

- A formalism to represent other formalisms
- Support for *natural deduction*
- A common basis for implementations
- Type theories are commonly used, but Isabelle uses a simple meta-logic whose main primitives are
 - \Rightarrow (implication)
 - \wedge (universal quantification)

Isabelle's Family of Logics



Isabelle's Family of Logics



Natural Deduction in Isabelle

Natural Deduction in Isabelle

$$\frac{P \quad Q}{P \wedge Q}$$

$$P \Rightarrow (Q \Rightarrow P \wedge Q)$$

Natural Deduction in Isabelle

$$\frac{P \quad Q}{P \wedge Q}$$

$$\frac{P \wedge Q}{P}$$

$$P \Rightarrow (Q \Rightarrow P \wedge Q)$$

$$P \wedge Q \Rightarrow P$$

Natural Deduction in Isabelle

$$\frac{P \quad Q}{P \wedge Q}$$

$$P \Rightarrow (Q \Rightarrow P \wedge Q)$$

$$\frac{P \wedge Q}{P}$$

$$P \wedge Q \Rightarrow P$$

$$\frac{P \wedge Q}{Q}$$

$$P \wedge Q \Rightarrow Q$$

Natural Deduction in Isabelle

$$\frac{P \quad Q}{P \wedge Q}$$

$$P \Rightarrow (Q \Rightarrow P \wedge Q)$$

$$\frac{P \wedge Q}{P}$$

$$P \wedge Q \Rightarrow P$$

$$\frac{P \wedge Q}{Q}$$

$$P \wedge Q \Rightarrow Q$$

$$\frac{P \rightarrow Q \quad P}{Q}$$

$$P \rightarrow Q \Rightarrow (P \Rightarrow Q)$$

Meta-implication

Meta-implication

- The symbol \Rightarrow (or \implies) expresses the relationship between premise and conclusion

Meta-implication

- The symbol \Rightarrow (or \implies) expresses the relationship between premise and conclusion
- ... and between subgoal and goal.

Meta-implication

- The symbol \Rightarrow (or $==>$) expresses the relationship between premise and conclusion
- ... and between subgoal and goal.
- It is distinct from \rightarrow , which is not part of Isabelle's underlying logical framework.

Meta-implication

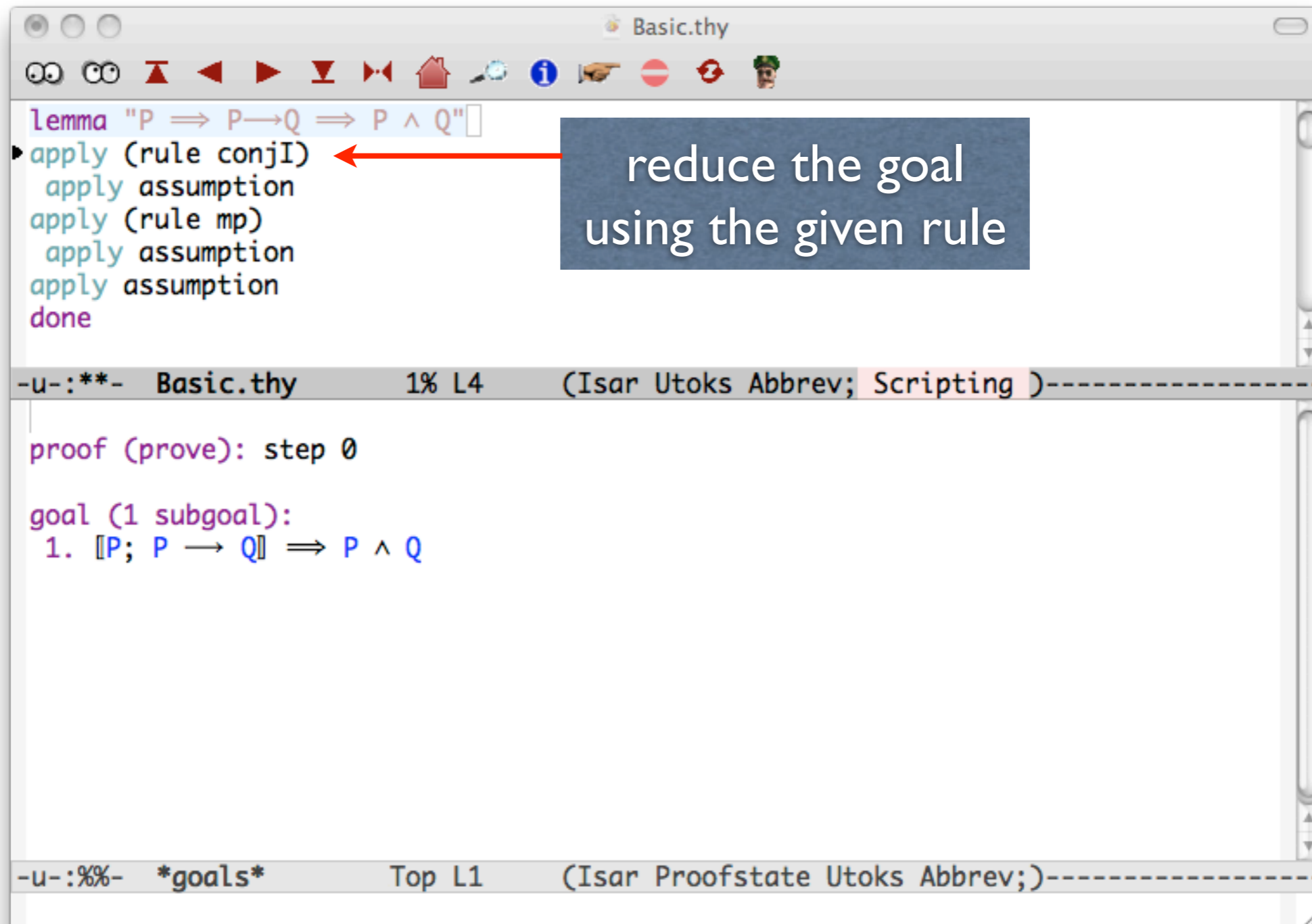
- The symbol \Rightarrow (or $==>$) expresses the relationship between premise and conclusion
- ... and between subgoal and goal.
- It is distinct from \rightarrow , which is not part of Isabelle's underlying logical framework.
- $P \Rightarrow (Q \Rightarrow R)$ is abbreviated as $[[P ; Q]] \Rightarrow R$

A Trivial Proof

```
Basic.thy
lemmas "P ==> P -> Q ==> P ^ Q"
proof (prove): step 0
goal (1 subgoal):
1. [[P; P -> Q]] ==> P ^ Q
done
```

The screenshot shows a proof editor window titled "Basic.thy". The main text area contains a lemma definition and a proof script. The lemma is `lemma "P ==> P -> Q ==> P ^ Q"`. The proof script consists of several steps: `apply (rule conjI)`, `apply assumption`, `apply (rule mp)`, `apply assumption`, `apply assumption`, and `done`. Below the main text area, there is a status bar showing the current position in the proof: `-u-:***- Basic.thy 1% L4 (Isar Utoks Abbrev; Scripting)`. At the bottom, another status bar shows the current goal: `-u-:%%- *goals* Top L1 (Isar Proofstate Utoks Abbrev;)`.

A Trivial Proof



The screenshot shows a proof assistant window titled "Basic.thy". The main text area contains the following code:

```
lemma "P  $\implies$  P  $\longrightarrow$  Q  $\implies$  P  $\wedge$  Q"  
• apply (rule conjI)  
  apply assumption  
  apply (rule mp)  
  apply assumption  
  apply assumption  
done
```

A red arrow points from a blue callout box to the `apply (rule conjI)` line. The callout box contains the text: "reduce the goal using the given rule".

Below the main text area, there are two status bars. The top one shows: `-u-:***- Basic.thy 1% L4 (Isar Utoks Abbrev; Scripting)`. The bottom one shows: `-u-:%%- *goals* Top L1 (Isar Proofstate Utoks Abbrev;)`.

The bottom status bar also indicates the current goal state:

```
proof (prove): step 0  
goal (1 subgoal):  
1.  $\llbracket P; P \longrightarrow Q \rrbracket \implies P \wedge Q$ 
```

Proof by Assumption

```
Basic.thy
lemmas "P ==> P -> Q ==> P ^ Q"
apply (rule conjI)
  apply assumption
  apply (rule mp)
    apply assumption
    apply assumption
  done

-u-:***- Basic.thy 1% L6 (Isar Utoks Abbrev; Scripting )-----

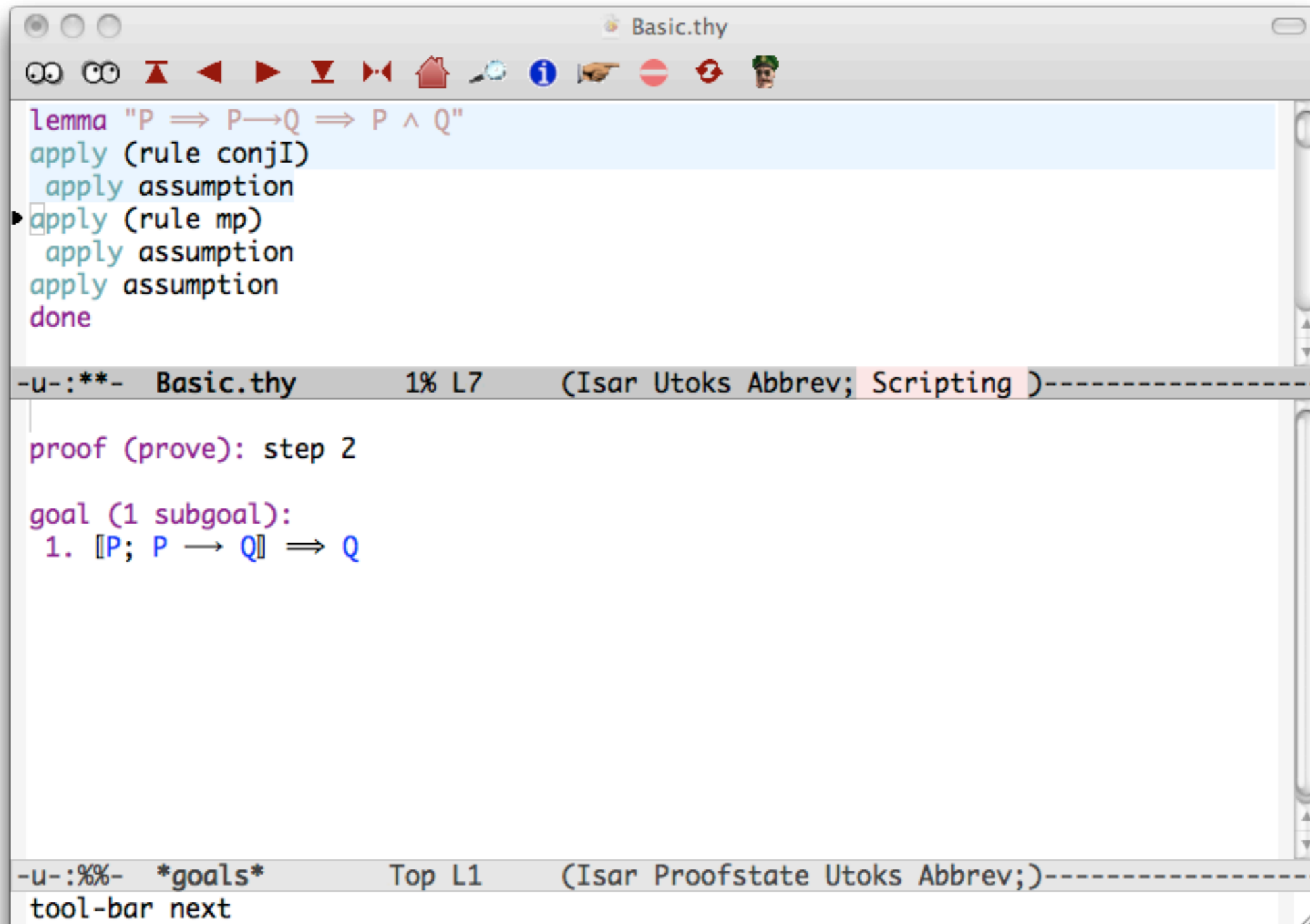
proof (prove): step 1

goal (2 subgoals):
  1. [[P; P -> Q]] ==> P
  2. [[P; P -> Q]] ==> Q

-u-:%%- *goals* Top L1 (Isar Proofstate Utoks Abbrev;)-----
tool-bar next
```

holds trivially,
by assumption

Proof by Assumption



```
Basic.thy
lemmas "P ==> P -> Q ==> P ^ Q"
apply (rule conjI)
  apply assumption
  apply (rule mp)
    apply assumption
    apply assumption
  done

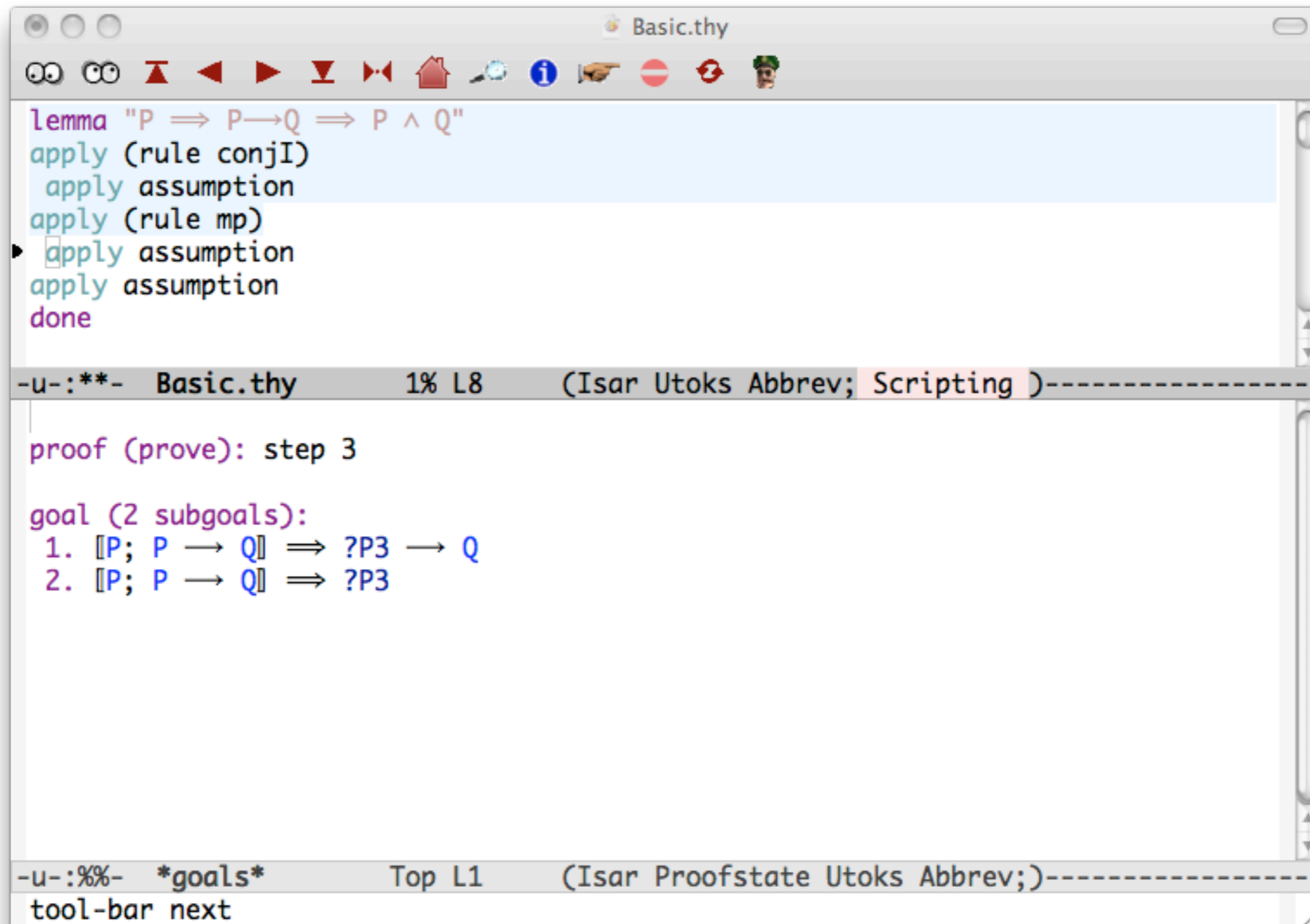
-u-:***- Basic.thy 1% L7 (Isar Utoks Abbrev; Scripting )-----

proof (prove): step 2

goal (1 subgoal):
  1. [[P; P -> Q]] ==> Q

-u-:%%- *goals* Top L1 (Isar Proofstate Utoks Abbrev;)-----
tool-bar next
```

Unknowns in Subgoals



```
Basic.thy
lemmas "P ==> P -> Q ==> P ^ Q"
apply (rule conjI)
  apply assumption
  apply (rule mp)
  apply assumption
done

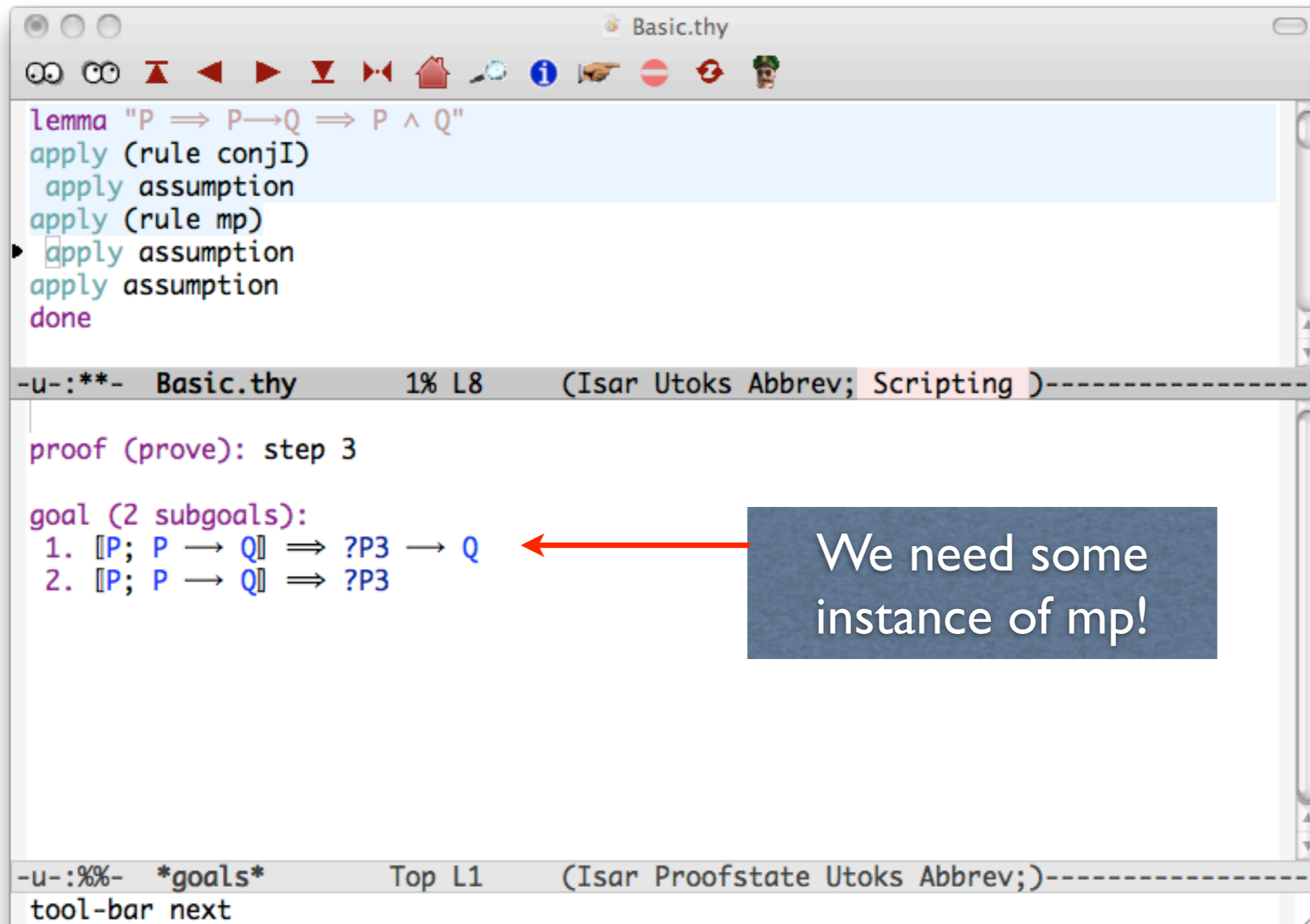
-u-:***- Basic.thy 1% L8 (Isar Utoks Abbrev; Scripting )-----

proof (prove): step 3

goal (2 subgoals):
  1. [[P; P -> Q]] ==> ?P3 -> Q
  2. [[P; P -> Q]] ==> ?P3

-u-:%%- *goals* Top L1 (Isar Proofstate Utoks Abbrev;)-----
tool-bar next
```

Unknowns in Subgoals



```
Basic.thy
lemmas "P ==> P -> Q ==> P ^ Q"
apply (rule conjI)
  apply assumption
  apply (rule mp)
  apply assumption
  apply assumption
done

-u-:***- Basic.thy 1% L8 (Isar Utoks Abbrev; Scripting )-----

proof (prove): step 3

goal (2 subgoals):
  1. [[P; P -> Q]] ==> ?P3 -> Q
  2. [[P; P -> Q]] ==> ?P3

-u-:%%- *goals* Top L1 (Isar Proofstate Utoks Abbrev;)-----
tool-bar next
```

We need some instance of mp!

Unknowns in Subgoals

The screenshot shows the Isabelle/Isar IDE interface. The top window displays the source code for a lemma:

```
lemma "P  $\implies$  P  $\longrightarrow$  Q  $\implies$  P  $\wedge$  Q"  
apply (rule conjI)  
  apply assumption  
  apply (rule mp)  
  apply assumption  
  apply assumption  
done
```

The bottom window shows the proof state after the first two steps:

```
proof (prove): step 3  
goal (2 subgoals):  
1.  $\llbracket P; P \longrightarrow Q \rrbracket \implies ?P3 \longrightarrow Q$   
2.  $\llbracket P; P \longrightarrow Q \rrbracket \implies ?P3$ 
```

Two red arrows point from text boxes to the subgoals:

- A box containing "We need some instance of mp!" points to the first subgoal.
- A box containing "formula placeholder" points to the $?P3$ placeholder in the second subgoal.

The bottom status bar indicates the current view is of the goals: `-u-:%%- *goals* Top L1 (Isar Proofstate Utoks Abbrev;)`

Unknowns and Unification

The screenshot shows a window titled "Basic.thy" with a toolbar at the top. The main text area contains a proof script:

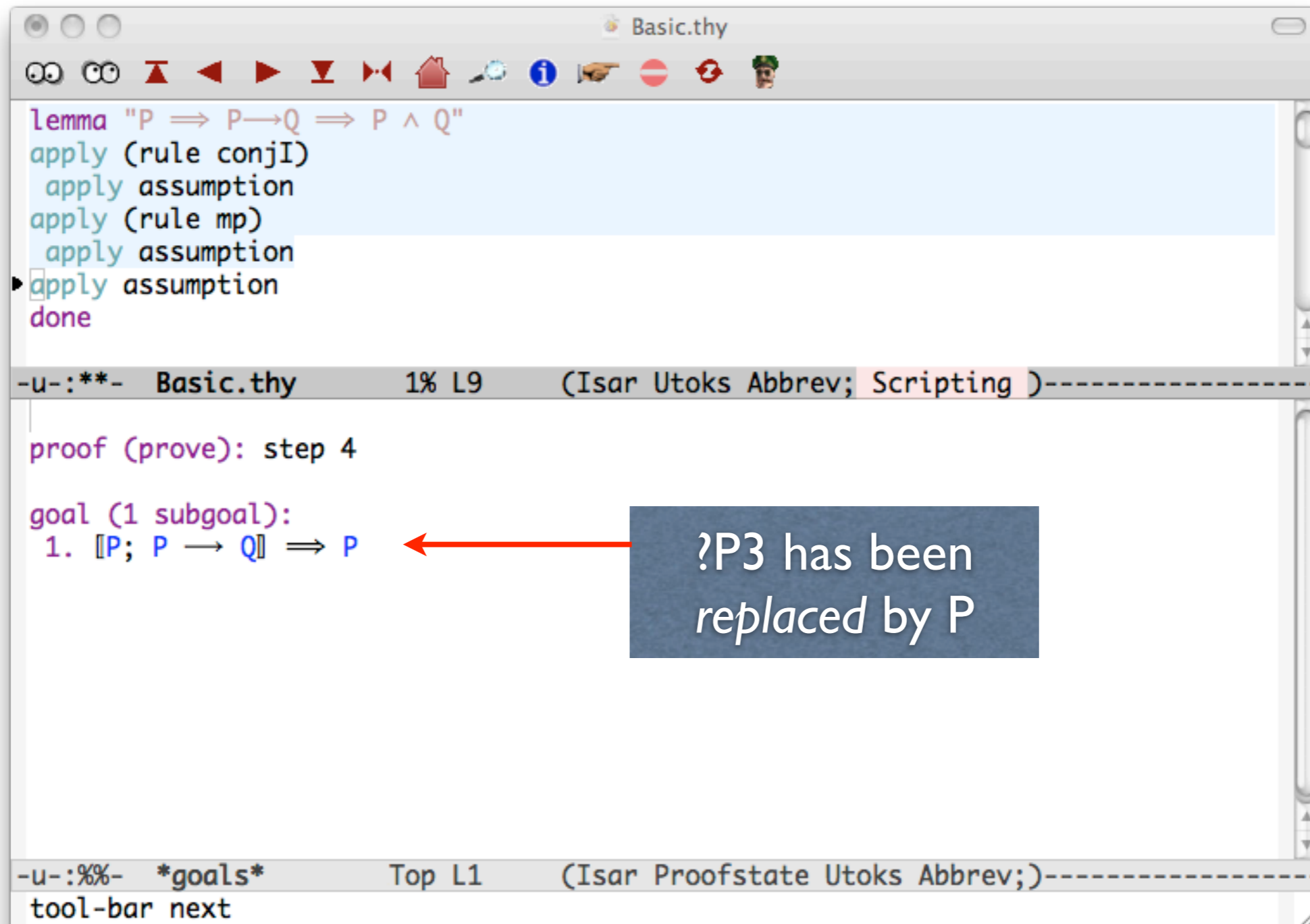
```
lemma "P  $\implies$  P  $\longrightarrow$  Q  $\implies$  P  $\wedge$  Q"  
apply (rule conjI)  
  apply assumption  
  apply (rule mp)  
  apply assumption  
▶ apply assumption  
done
```

Below the script is a status bar with the text: `-u-:***- Basic.thy 1% L9 (Isar Utoks Abbrev; Scripting)`. The bottom section of the window shows the current proof state:

```
proof (prove): step 4  
goal (1 subgoal):  
  1.  $\llbracket P; P \longrightarrow Q \rrbracket \implies P$ 
```

At the very bottom, another status bar reads: `-u-:%%- *goals* Top L1 (Isar Proofstate Utoks Abbrev;)` and a `tool-bar next` button is visible.

Unknowns and Unification



The screenshot shows a theorem prover interface with a window titled "Basic.thy". The main editor contains the following code:

```
lemma "P  $\implies$  P  $\longrightarrow$  Q  $\implies$  P  $\wedge$  Q"  
  apply (rule conjI)  
  apply assumption  
  apply (rule mp)  
  apply assumption  
  apply assumption  
done
```

The interface has a status bar at the bottom of the editor showing "-u-:***- Basic.thy 1% L9 (Isar Utoks Abbrev; Scripting)". Below the editor, a goal state is displayed:

```
proof (prove): step 4  
goal (1 subgoal):  
1.  $\llbracket P; P \longrightarrow Q \rrbracket \implies P$ 
```

A red arrow points from a callout box to the goal. The callout box contains the text: "?P3 has been replaced by P".

At the bottom of the interface, the status bar shows "-u-:%%- *goals* Top L1 (Isar Proofstate Utoks Abbrev;)" and "tool-bar next".

Discharging Assumptions

Discharging Assumptions

$$\frac{\begin{array}{c} [P] \\ \vdots \\ Q \end{array}}{P \rightarrow Q}$$

$$(P \Rightarrow Q) \Rightarrow P \rightarrow Q$$

Discharging Assumptions

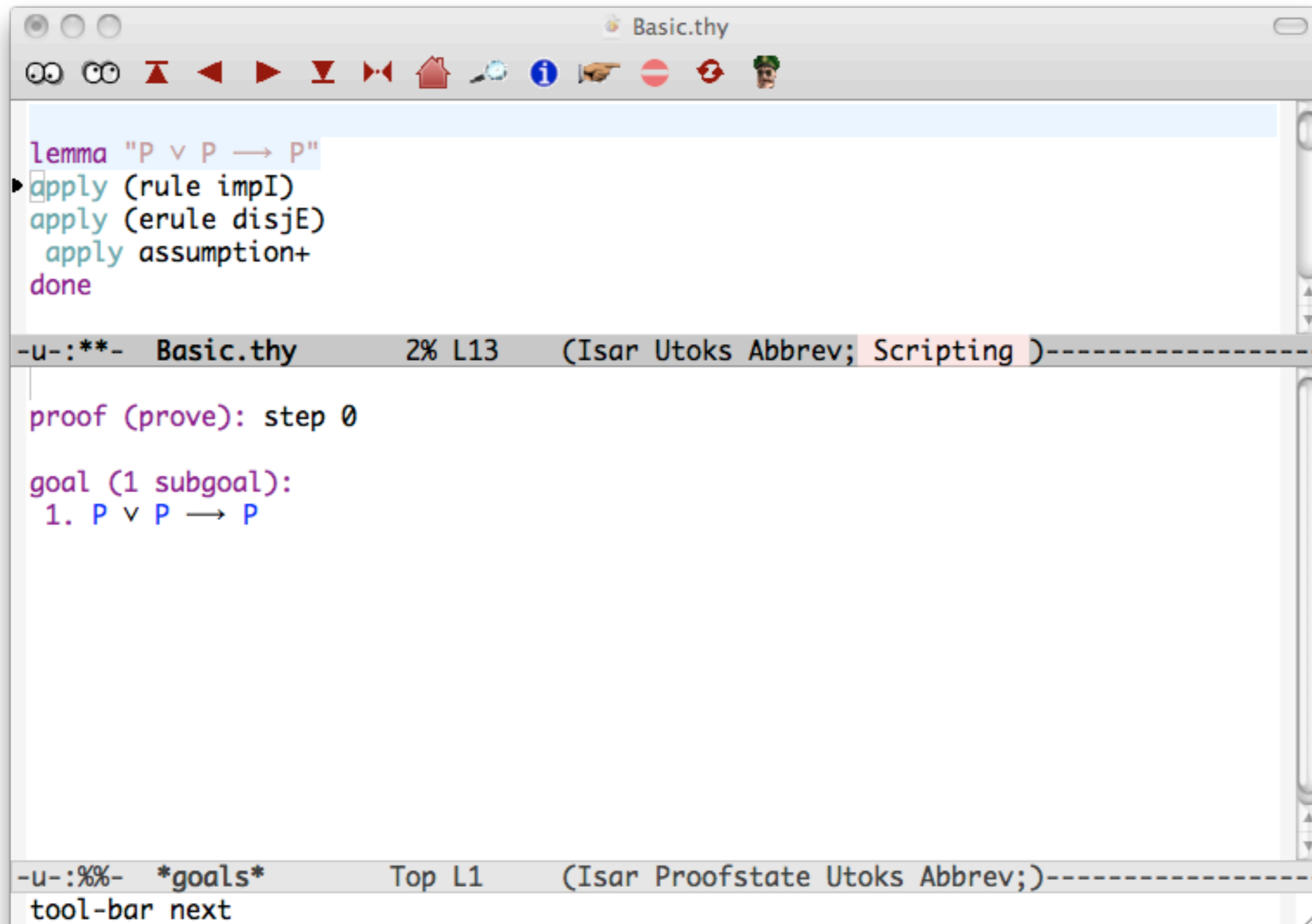
$$\frac{\begin{array}{c} [P] \\ \vdots \\ Q \end{array}}{P \rightarrow Q}$$

$$(P \Rightarrow Q) \Rightarrow P \rightarrow Q$$

$$\frac{P \vee Q \quad \begin{array}{c} [P] \\ \vdots \\ R \end{array} \quad \begin{array}{c} [Q] \\ \vdots \\ R \end{array}}{R}$$

$$\begin{array}{c} \llbracket P \vee Q; P \Rightarrow R; Q \Rightarrow R \rrbracket \Rightarrow \\ R \end{array}$$

A Proof using Assumptions



The screenshot shows a proof assistant window titled "Basic.thy". The main editor contains the following code:

```
lemma "P ∨ P → P"  
  apply (rule impI)  
  apply (erule disjE)  
  apply assumption+  
  done
```

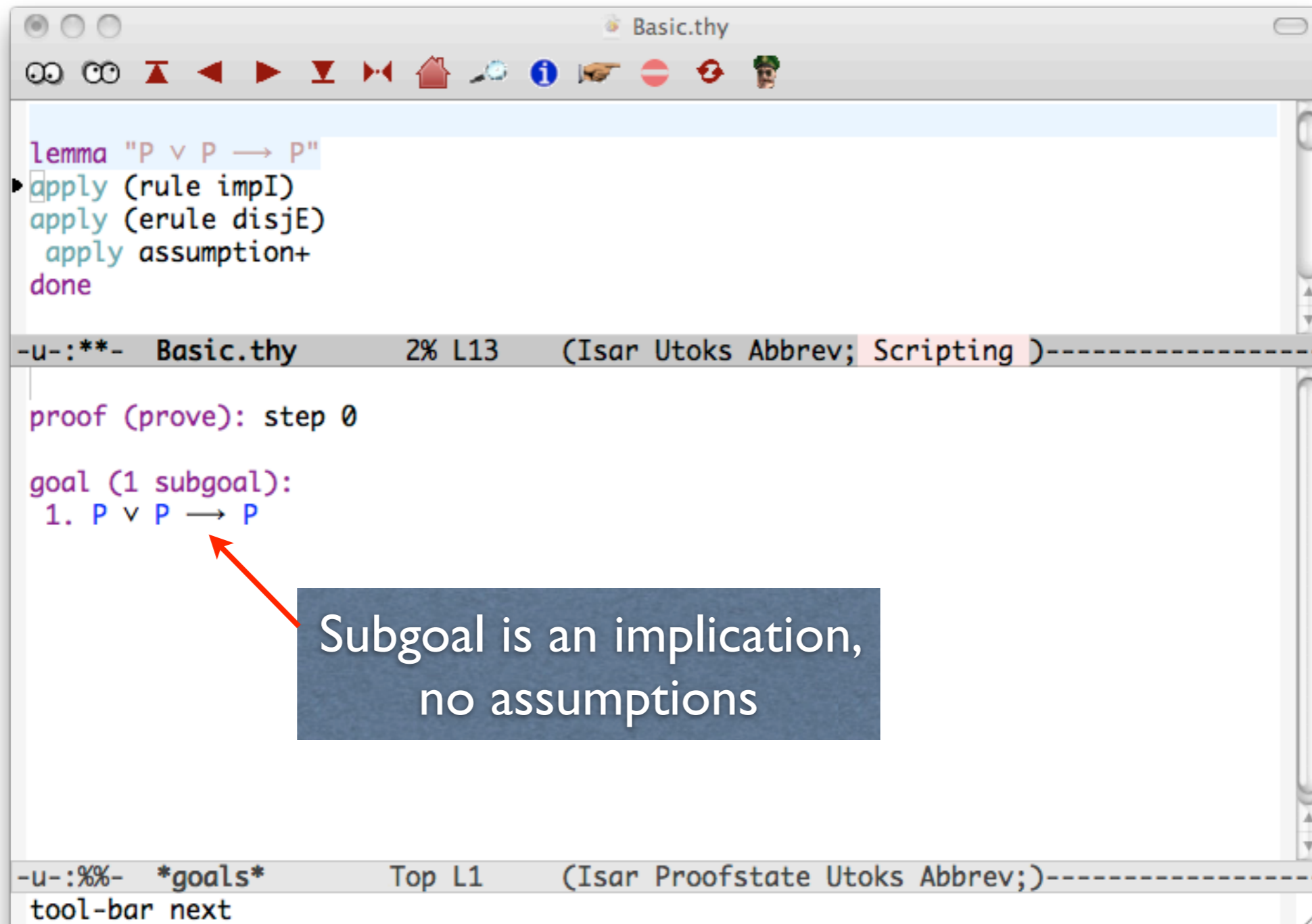
Below the code is a status bar with the text: "-u-:***- Basic.thy 2% L13 (Isar Utoks Abbrev; Scripting)-----".

The bottom panel shows the current proof state:

```
proof (prove): step 0  
  
goal (1 subgoal):  
  1. P ∨ P → P
```

At the bottom of the window, there is a toolbar and the text: "-u-:%%- *goals* Top L1 (Isar Proofstate Utoks Abbrev;)-----" and "tool-bar next".

A Proof using Assumptions



The screenshot shows a proof assistant window titled "Basic.thy". The main text area contains the following code:

```
lemma "P ∨ P → P"  
  apply (rule impI)  
  apply (erule disjE)  
  apply assumption+  
  done
```

Below the code, a status bar indicates the current position: "-u-:***- Basic.thy 2% L13 (Isar Utoks Abbrev; Scripting)".

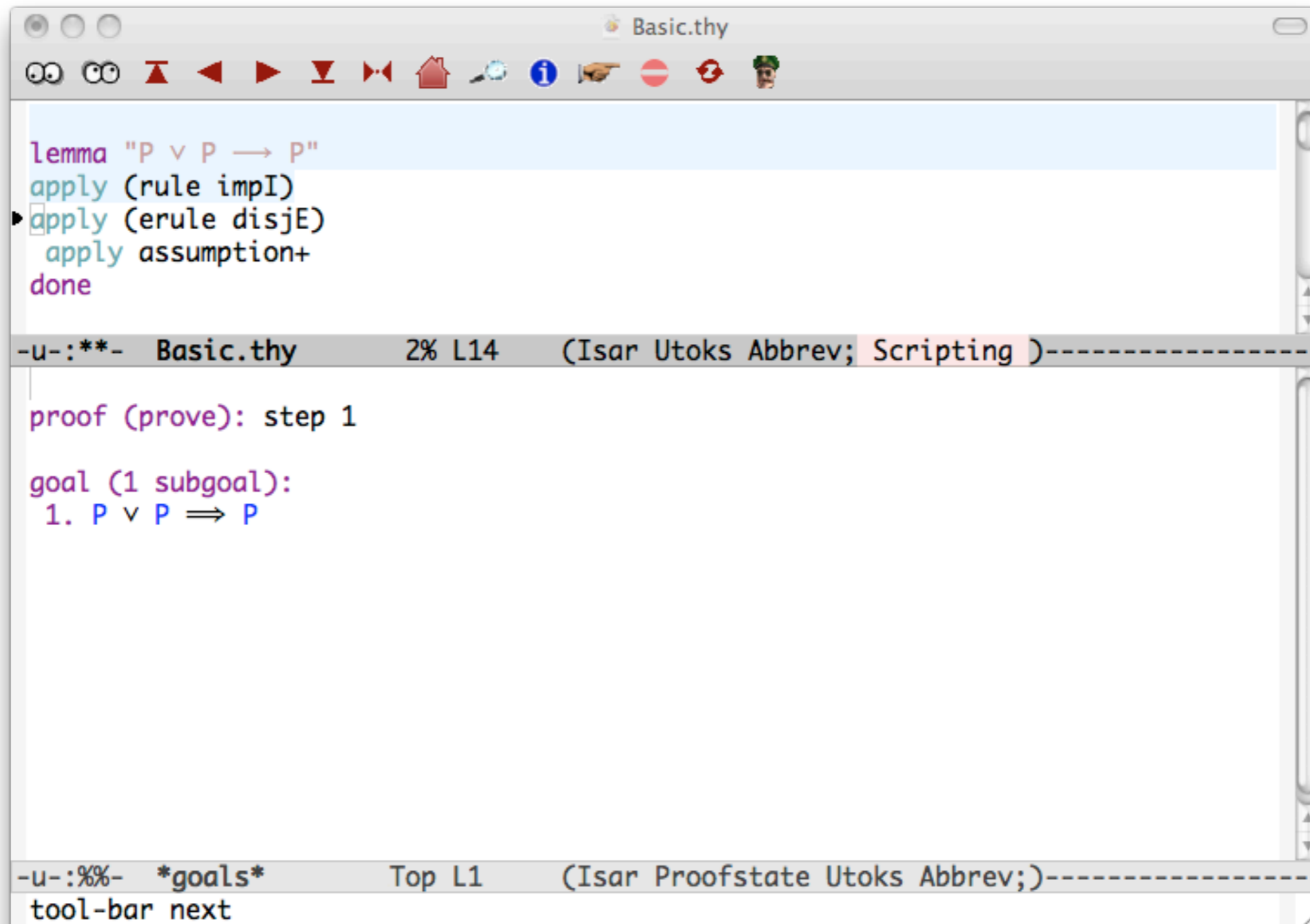
The proof state is shown below the status bar:

```
proof (prove): step 0  
goal (1 subgoal):  
1. P ∨ P → P
```

A red arrow points from a dark blue callout box to the subgoal "1. P ∨ P → P". The callout box contains the text: "Subgoal is an implication, no assumptions".

At the bottom of the window, another status bar shows: "-u-:%%- *goals* Top L1 (Isar Proofstate Utoks Abbrev;)-----
tool-bar next".

After Implies-Introduction



The screenshot shows a proof assistant window titled "Basic.thy". The main editor contains the following code:

```
lemma "P ∨ P → P"  
  apply (rule impI)  
  apply (erule disjE)  
  apply assumption+  
  done
```

The status bar below the editor shows the current position: "-u-:***- Basic.thy 2% L14 (Isar Utoks Abbrev; Scripting)".

The proof state is shown in a separate window below the editor:

```
proof (prove): step 1  
  
goal (1 subgoal):  
  1. P ∨ P ⇒ P
```

The status bar for the proof state shows: "-u-:%%- *goals* Top L1 (Isar Proofstate Utoks Abbrev;)".

At the bottom left, there is a "tool-bar next" label.

After Implies-Introduction

```
lemma "P ∨ P → P"
  apply (rule impI)
  apply (erule disjE)
  apply assumption+
  done
```

Prove P using P ∨ P

```
proof (prove): step 1
  goal (1 subgoal):
    1. P ∨ P ⇒ P
```

goals Top L1 (Isar Proofstate Utoks Abbrev;) tool-bar next

After Implies-Introduction

The screenshot shows a theorem prover interface with the following content:

```
lemma "P ∨ P → P"
  apply (rule impI)
  apply (erule disjE)
  apply assumption+
  done
```

The proof state is shown below:

```
-u-:**- Basic.thy
proof (prove): step 1
goal (1 subgoal):
1. P ∨ P ⇒ P
```

Annotations:

- A blue box with the text "Prove P using P ∨ P" has a red arrow pointing to the goal "1. P ∨ P ⇒ P".
- A blue box with the text "Assumption will be used, then **deleted**" has a red arrow pointing to the goal "1. P ∨ P ⇒ P".

The bottom status bar shows: `-u-:%%- *goals* Top L1 (Isar Proofstate Utoks Abbrev;)-----` and `tool-bar next`.

Disjunction Elimination

The screenshot shows a window titled "Basic.thy" with a toolbar at the top. The main text area contains the following code:

```
lemma "P ∨ P → P"  
apply (rule impI)  
apply (erule disjE)  
▸ apply assumption+  
done
```

Below the code is a status bar: "-u-:**- Basic.thy 2% L15 (Isar Utoks Abbrev; Scripting)-----".

The next section shows the current proof state:

```
proof (prove): step 2  
  
goal (2 subgoals):  
1. P ⇒ P  
2. P ⇒ P
```

At the bottom, another status bar reads: "-u-:%%- *goals* Top L1 (Isar Proofstate Utoks Abbrev;)-----" and "tool-bar next".

Disjunction Elimination

The screenshot shows a theorem prover interface with a window titled "Basic.thy". The main editor contains the following code:

```
lemma "P ∨ P → P"  
  apply (rule impI)  
  apply (erule disjE)  
  apply assumption+  
done
```

A red arrow points from a dark blue callout box containing the text "erule is good with elimination rules" to the `erule disjE` line in the code. Below the code, the interface shows the current goal state:

```
proof (prove): step 2  
  
goal (2 subgoals):  
1. P ⇒ P  
2. P ⇒ P
```

The bottom status bar indicates the current state: `*goals*`, `Top L1`, and `(Isar Proofstate Utoks Abbrev;)`. The text `tool-bar next` is visible at the very bottom.

Disjunction Elimination

The screenshot shows a theorem prover interface with two panels. The top panel displays a lemma and its proof steps:

```
lemma "P ∨ P → P"  
  apply (rule impI)  
  apply (erule disjE)  
  apply assumption+  
done
```

A red arrow points from the text "erule is good with elimination rules" to the `erule disjE` line.

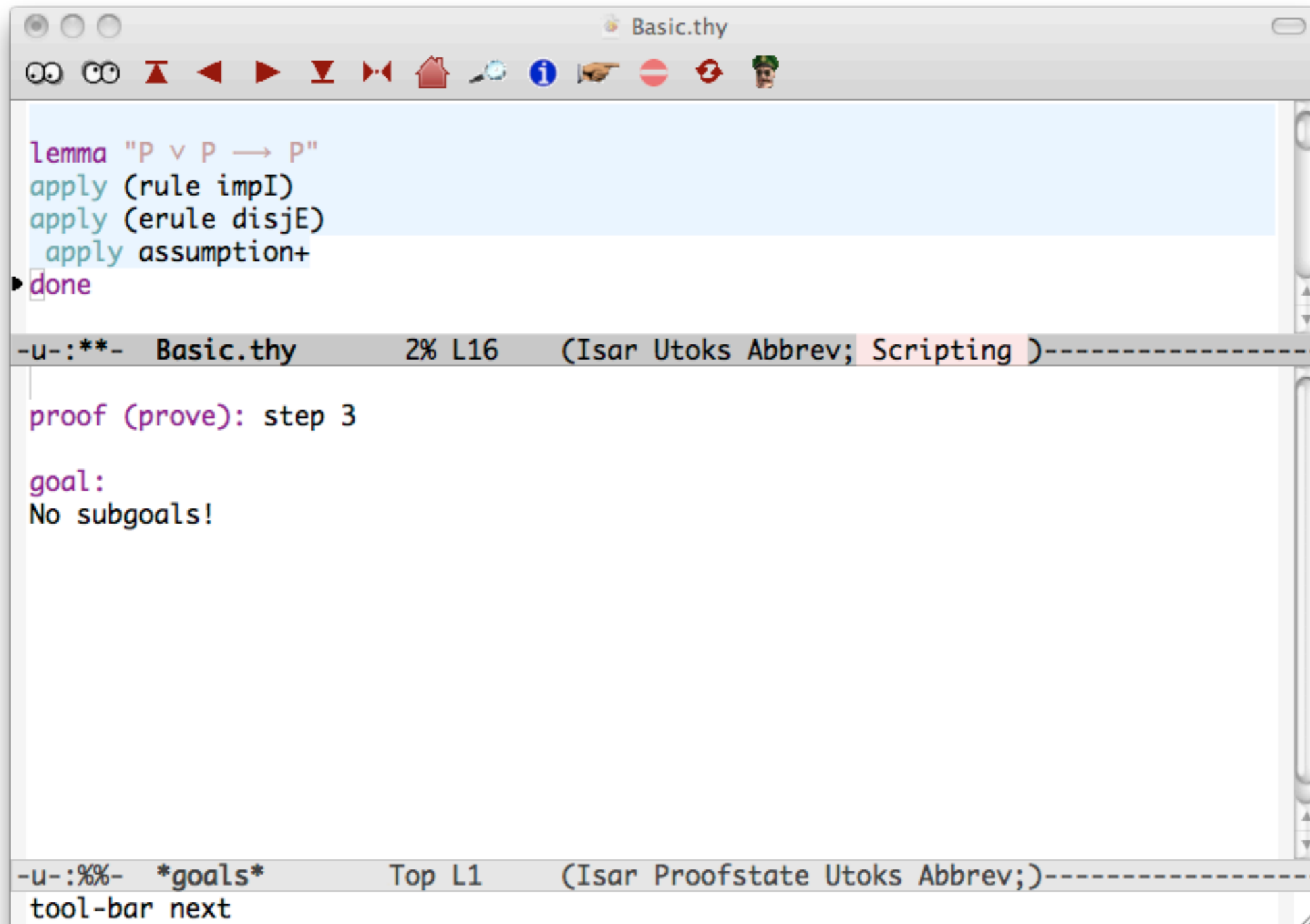
The bottom panel shows the proof state:

```
proof (prove): step 2  
goal (2 subgoals):  
1. P ⇒ P  
2. P ⇒ P
```

A red arrow points from the text "An instance of ?P ∨ ?Q has been found" to the first goal `1. P ⇒ P`.

The status bar at the bottom of the interface reads: `-u-:%%- *goals* Top L1 (Isar Proofstate Utoks Abbrev;)-----`

The Final Step



The screenshot shows a window titled "Basic.thy" with a toolbar at the top. The main text area contains a proof script:

```
lemma "P ∨ P → P"  
  apply (rule impI)  
  apply (erule disjE)  
  apply assumption+  
done
```

Below the script, a status bar indicates the current position: "-u-:**- Basic.thy 2% L16 (Isar Utoks Abbrev; Scripting)".

The lower section of the window shows the execution state:

```
proof (prove): step 3  
  
goal:  
No subgoals!
```

At the bottom, another status bar shows: "-u-:%%- *goals* Top L1 (Isar Proofstate Utoks Abbrev;)".

The Final Step

```
lemma "P ∨ P → P"
  apply (rule impI)
  apply (erule disjE)
  apply assumption+
done
```

+ applies a method one or more times

```
proof (prove): step 3
goal:
No subgoals!
```

tool-bar next

Quantifiers

Quantifiers

$$\frac{P(t)}{\exists x. P(x)}$$

$$P(x) \Rightarrow \exists x. P(x)$$

Quantifiers

$$\frac{P(t)}{\exists x. P(x)}$$

$$P(\mathbf{x}) \Rightarrow \exists \mathbf{x}. P(\mathbf{x})$$

$$\frac{\exists x. P(x) \quad \begin{array}{c} [P(x)] \\ \vdots \\ Q \end{array}}{Q}$$

$$\llbracket \exists \mathbf{x}. P(\mathbf{x}) ; \bigwedge \mathbf{x}. P(\mathbf{x}) \Rightarrow Q \rrbracket \Rightarrow Q$$

Quantifiers

$$\frac{P(t)}{\exists x. P(x)}$$

$$P(x) \Rightarrow \exists x. P(x)$$

$$\frac{\exists x. P(x)}{Q}$$

$[P(x)]$
 \vdots
 Q

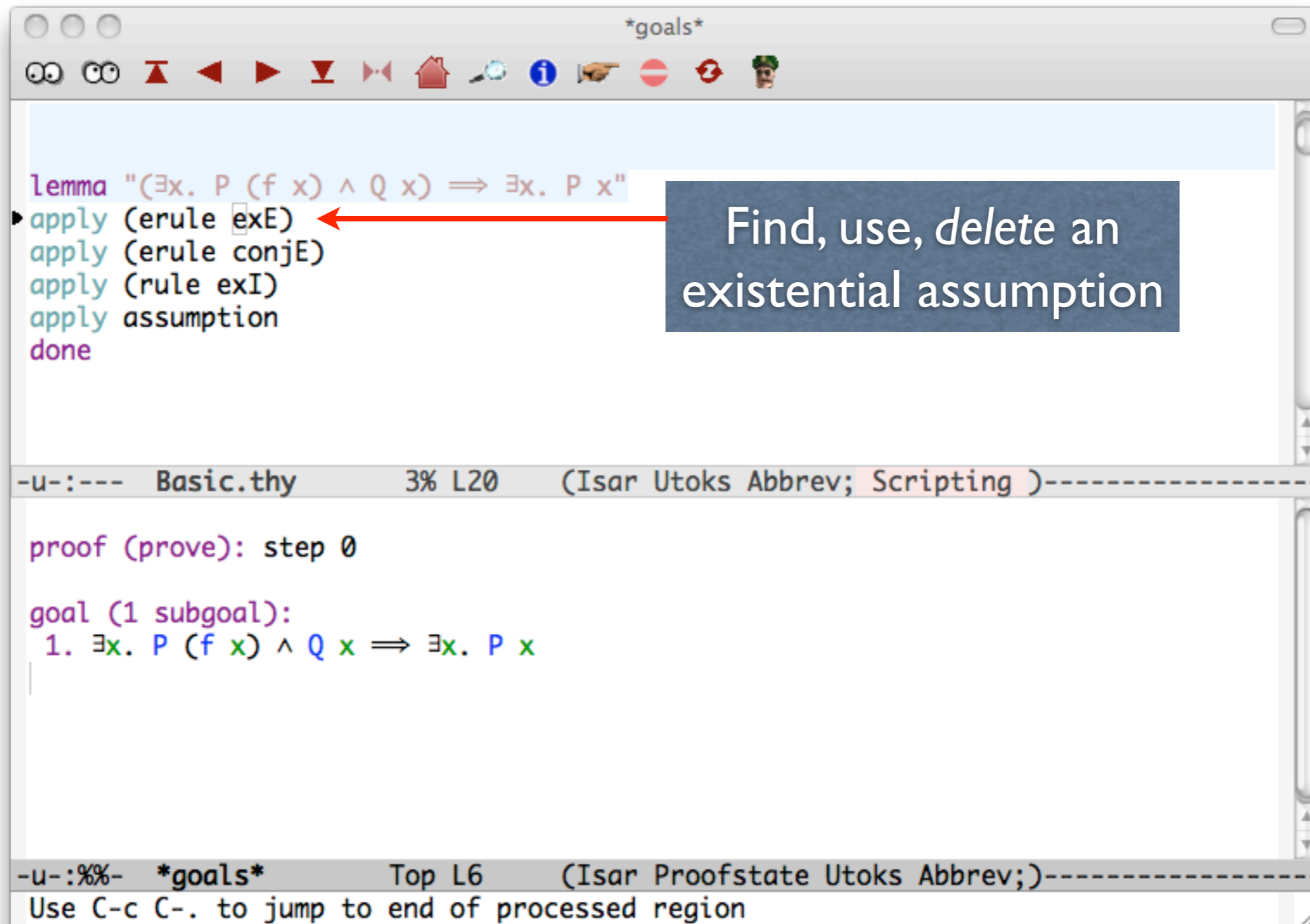
$$\llbracket \exists x. P(x) ; \forall x. P(x) \Rightarrow Q \rrbracket \Rightarrow Q$$

meta-universal quantifier
states the variable condition

A Tiny Quantifier Proof

```
*goals*  
lemmas  
  lemma "( $\exists x. P (f x) \wedge Q x$ )  $\implies \exists x. P x$ "  
  apply (erule exE)  
  apply (erule conjE)  
  apply (rule exI)  
  apply assumption  
  done  
-u-:--- Basic.thy      3% L20      (Isar Utoks Abbrev; Scripting )-----  
proof (prove): step 0  
goal (1 subgoal):  
  1.  $\exists x. P (f x) \wedge Q x \implies \exists x. P x$   
-u-:%%- *goals*      Top L6      (Isar Proofstate Utoks Abbrev;)-----  
Use C-c C-. to jump to end of processed region
```

A Tiny Quantifier Proof



The screenshot shows a proof assistant window titled '*goals*'. The main area contains a lemma and its proof steps. A red arrow points from a text box to the 'exE' rule in the proof steps.

```
lemma "( $\exists x. P (f x) \wedge Q x$ )  $\implies \exists x. P x$ "  
• apply (erule exE) ←  
  apply (erule conjE)  
  apply (rule exI)  
  apply assumption  
done
```

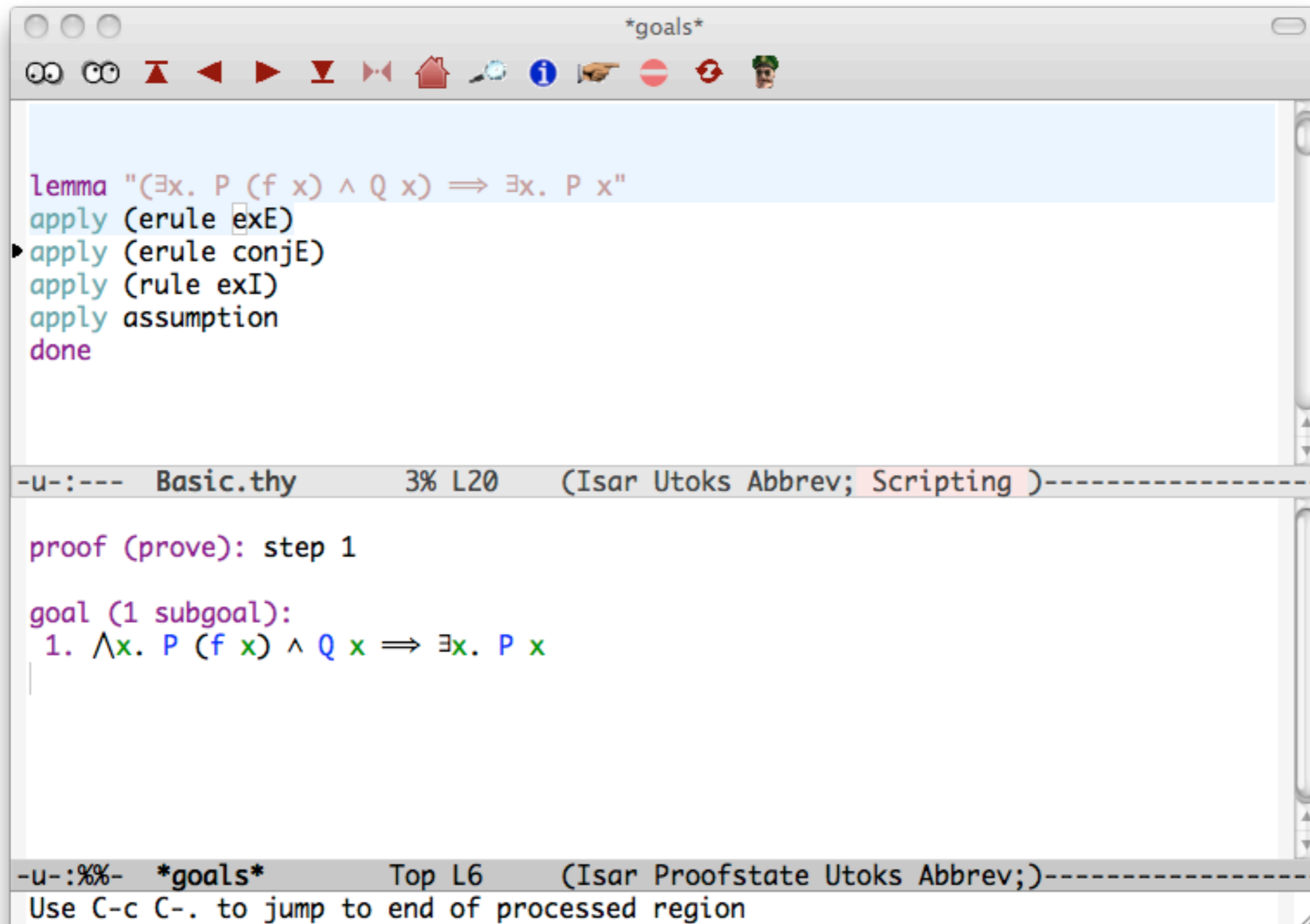
Find, use, *delete* an existential assumption

-u-:--- Basic.thy 3% L20 (Isar Utoks Abbrev; Scripting)-----

```
proof (prove): step 0  
goal (1 subgoal):  
1.  $\exists x. P (f x) \wedge Q x \implies \exists x. P x$   
|
```

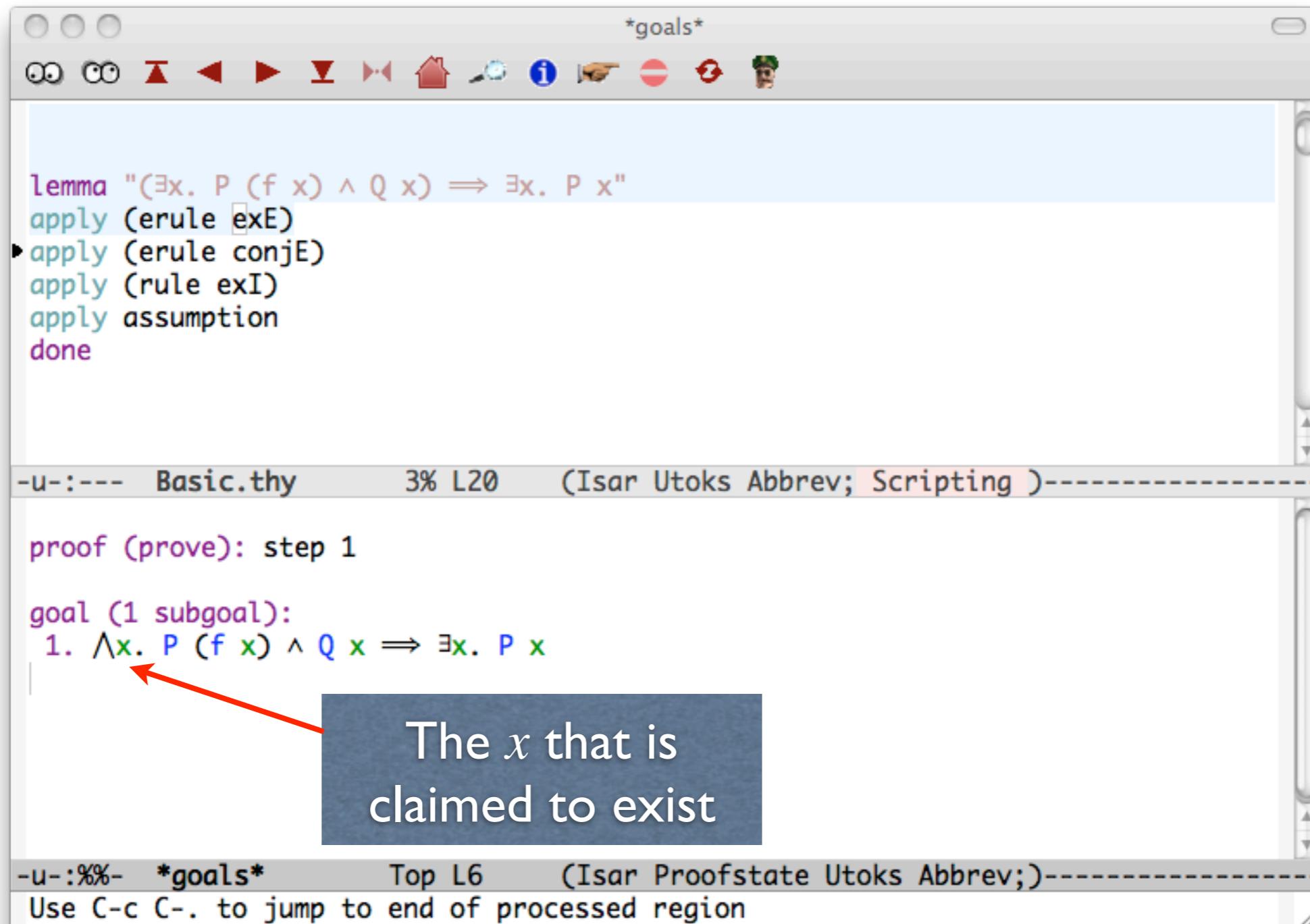
-u-:%%- *goals* Top L6 (Isar Proofstate Utoks Abbrev;)-----
Use C-c C-. to jump to end of processed region

Conjunction Elimination



```
*goals*  
lemmas "( $\exists x. P (f x) \wedge Q x$ )  $\implies \exists x. P x$ "  
apply (erule exE)  
apply (erule conjE)  
apply (rule exI)  
apply assumption  
done  
-----  
-u-:--- Basic.thy      3% L20      (Isar Utoks Abbrev; Scripting )-----  
proof (prove): step 1  
goal (1 subgoal):  
1.  $\wedge x. P (f x) \wedge Q x \implies \exists x. P x$   
-----  
-u-:%%- *goals*      Top L6      (Isar Proofstate Utoks Abbrev;)-----  
Use C-c C-. to jump to end of processed region
```

Conjunction Elimination



The screenshot shows a proof assistant window titled '*goals*' with a toolbar at the top. The main text area contains the following code:

```
lemma "( $\exists x. P (f x) \wedge Q x$ )  $\implies \exists x. P x$ "  
apply (erule exE)  
• apply (erule conjE)  
apply (rule exI)  
apply assumption  
done
```

Below the code is a status bar: `-u-:--- Basic.thy 3% L20 (Isar Utoks Abbrev; Scripting)-----`

The next section shows a proof step:

```
proof (prove): step 1  
goal (1 subgoal):  
1.  $\wedge x. P (f x) \wedge Q x \implies \exists x. P x$ 
```

An orange arrow points from a blue callout box to the x in the goal statement. The callout box contains the text: "The x that is claimed to exist".

At the bottom, another status bar reads: `-u-:%%- *goals* Top L6 (Isar Proofstate Utoks Abbrev;)-----`
Use C-c C-. to jump to end of processed region

Conjunction Elimination

```
lemma "( $\exists x. P (f x) \wedge Q x$ )  $\implies \exists x. P x$ "  
apply (erule exE)  
• apply (erule conjE) ←  
apply (rule exI)  
apply assumption  
done
```

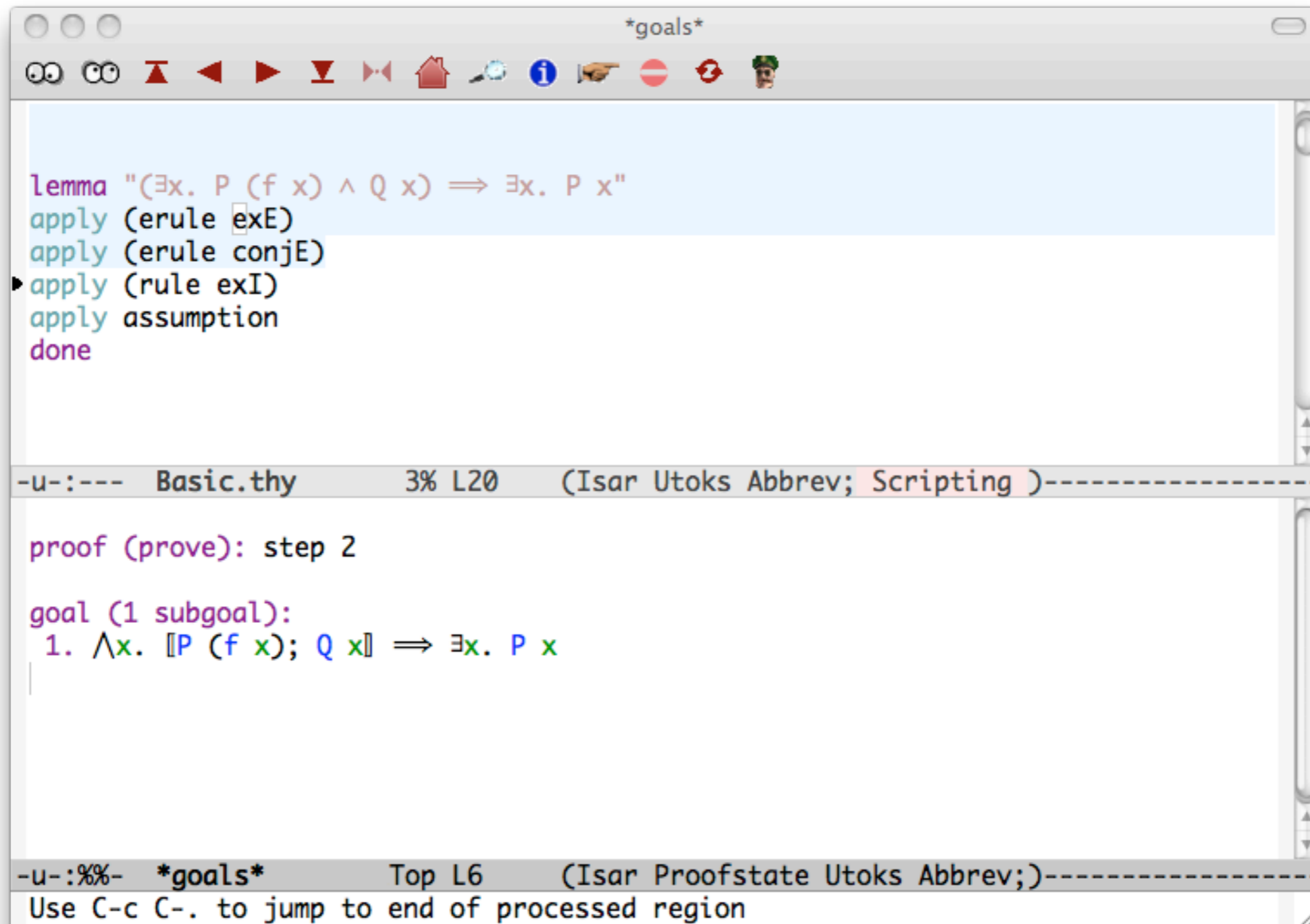
Find, use, delete a conjunctive assumption

```
-u-:--- Basic.thy 3% L20 (Isar Utoks Abbrev; Scripting )-----  
  
proof (prove): step 1  
  
goal (1 subgoal):  
1.  $\wedge x. P (f x) \wedge Q x \implies \exists x. P x$   
|  
|
```

The x that is claimed to exist

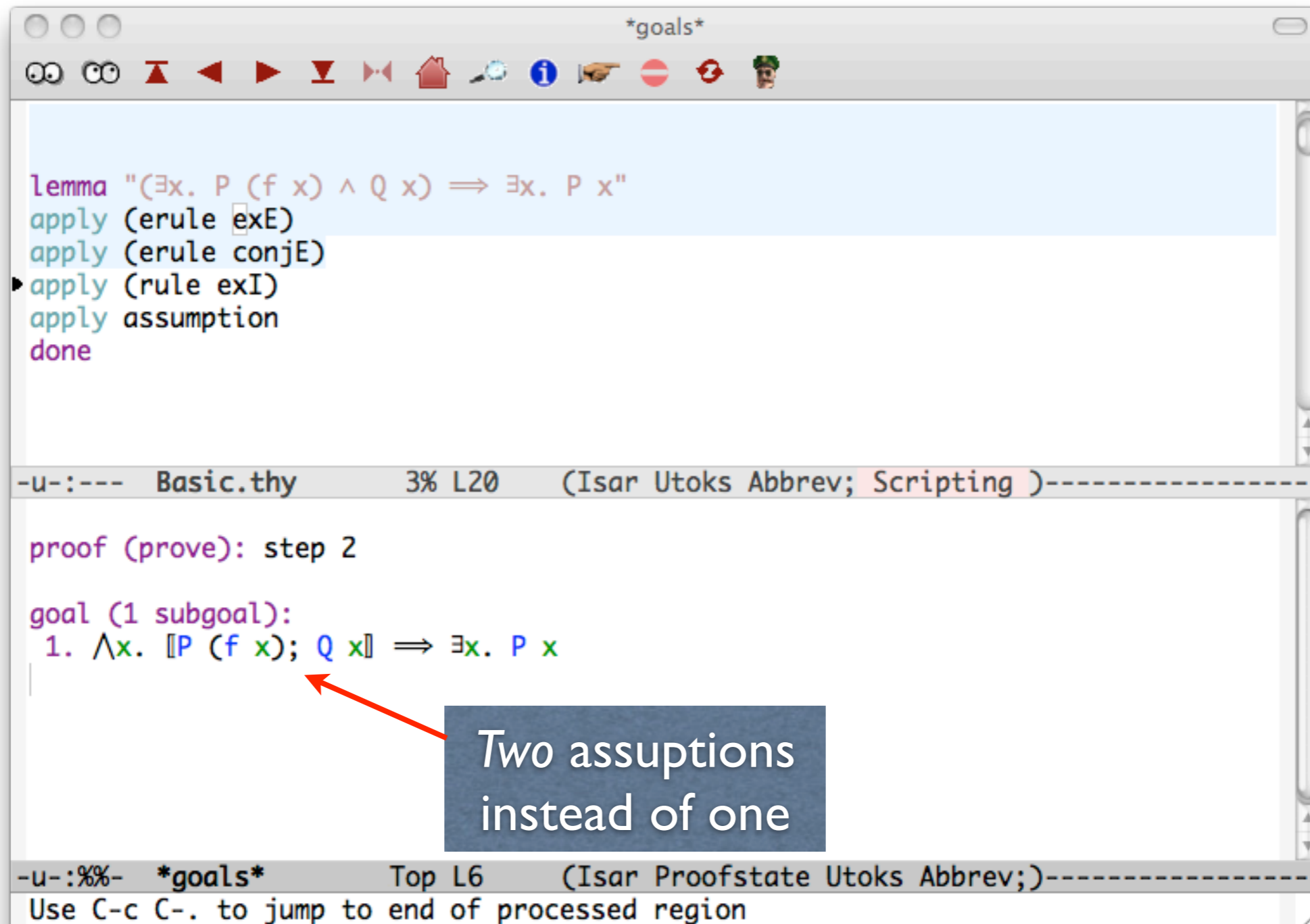
```
-u-:%%- *goals* Top L6 (Isar Proofstate Utoks Abbrev;)-----  
Use C-c C-. to jump to end of processed region
```

Now for \exists -Introduction



```
*goals*  
lemmas "( $\exists x. P (f x) \wedge Q x$ )  $\implies \exists x. P x$ "  
apply (erule exE)  
apply (erule conjE)  
apply (rule exI)  
apply assumption  
done  
-----  
-u-:--- Basic.thy          3% L20    (Isar Utoks Abbrev; Scripting )-----  
proof (prove): step 2  
goal (1 subgoal):  
1.  $\wedge x. [P (f x); Q x] \implies \exists x. P x$   
-----  
-u-:%%- *goals*          Top L6    (Isar Proofstate Utoks Abbrev;)-----  
Use C-c C-. to jump to end of processed region
```

Now for \exists -Introduction



```
lemma "( $\exists x. P (f x) \wedge Q x$ )  $\implies \exists x. P x$ "
  apply (erule exE)
  apply (erule conjE)
  apply (rule exI)
  apply assumption
  done
```

-u-:--- Basic.thy 3% L20 (Isar Utoks Abbrev; Scripting)-----

```
proof (prove): step 2
  goal (1 subgoal):
  1.  $\wedge x. [P (f x); Q x] \implies \exists x. P x$ 
```

-u-:%%- *goals* Top L6 (Isar Proofstate Utoks Abbrev;)-----
Use C-c C-. to jump to end of processed region

Two assumptions instead of one

Now for \exists -Introduction

```
lemma "( $\exists x. P (f x) \wedge Q x$ )  $\implies$   $\exists x. P x$ "
apply (erule exE)
apply (erule conjE)
apply (rule exI)
apply assumption
done
```

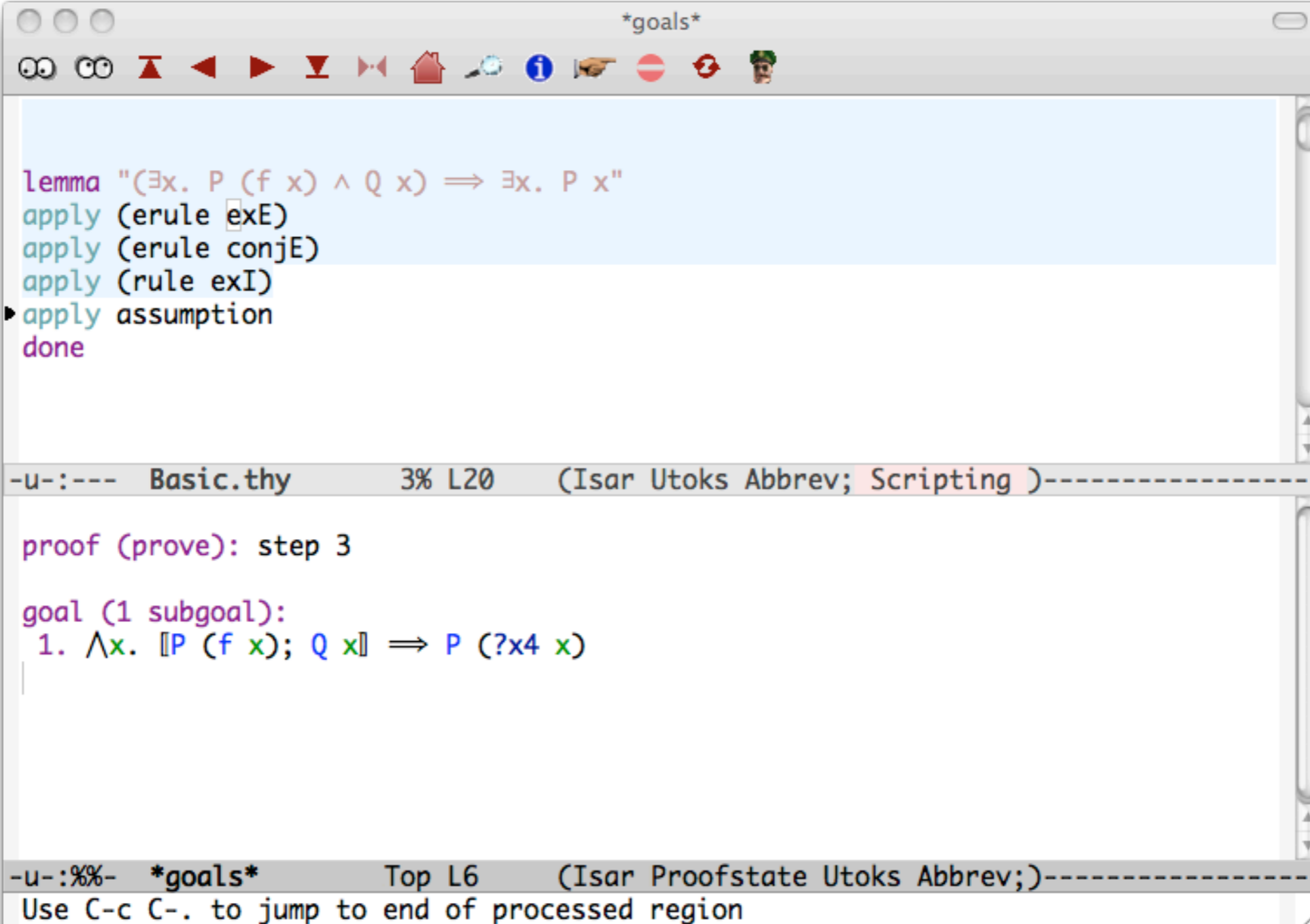
Apply the rule exI

```
proof (prove): step 2
goal (1 subgoal):
1.  $\wedge x. [P (f x); Q x] \implies \exists x. P x$ 
```

Two assumptions instead of one

```
-u-:%%- *goals* Top L6 (Isar Proofstate Utoks Abbrev;)-----
Use C-c C-. to jump to end of processed region
```

An Unknown for the Witness



```
*goals*
lemmas "(∃x. P (f x) ∧ Q x) ⇒ ∃x. P x"
apply (erule exE)
apply (erule conjE)
apply (rule exI)
▶ apply assumption
done

-u-:--- Basic.thy      3% L20  (Isar Utoks Abbrev; Scripting )-----

proof (prove): step 3

goal (1 subgoal):
  1. ∧x. [P (f x); Q x] ⇒ P (?x4 x)

-u-:%%- *goals*      Top L6  (Isar Proofstate Utoks Abbrev;)-----
Use C-c C-. to jump to end of processed region
```

An Unknown for the Witness

```
lemma "( $\exists x. P (f x) \wedge Q x$ )  $\implies \exists x. P x$ "
apply (erule exE)
apply (erule conjE)
apply (rule exI)
▶ apply assumption
done
```

-u-:--- Basic.thy 3% L20 (Isar Utoks Abbrev; Scripting)-----

```
proof (prove): step 3
goal (1 subgoal):
1.  $\wedge x. [P (f x); Q x] \implies P (?x4 x)$ 
```

-u-:%%- (state Utoks Abbrev;)-----

Use C-c C-. to jump to end of processed region

Proof by assumption will unify these two terms

Done!

The screenshot shows a window titled '*goals*' with a toolbar at the top. The main area contains a proof script for a lemma. The script is as follows:

```
lemma "( $\exists x. P (f x) \wedge Q x$ )  $\implies \exists x. P x$ "  
  apply (erule exE)  
  apply (erule conjE)  
  apply (rule exI)  
  apply assumption  
done
```

Below the script, a status bar indicates the current file is 'Basic.thy' at line 3, column 20, with the text '(Isar Utoks Abbrev; Scripting)'. The next section shows the result of the proof:

```
proof (prove): step 4  
goal:  
No subgoals!  
|
```

At the bottom, another status bar shows the current proof state as '*goals*' at line 6, column 1, with the text '(Isar Proofstate Utoks Abbrev;)'. A note at the very bottom says 'Use C-c C-. to jump to end of processed region'.