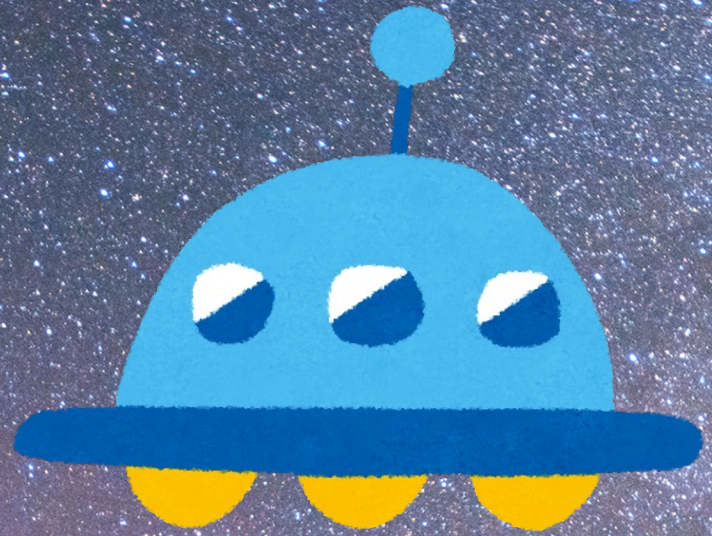
data61/PSL



Abduction Prover in Isabelle(/HOL)

Yutaka Nagashima (the Czech Academy of Sciences)
Daniel Sebastian Goc
x/twitter: YutakangE

✅ high-level talk

✅ installation

LIVE DEMO

(We're almost at the end of this talk.)

3 take-home lessons:

① abduction using modus ponens

② tactics = conjecturing

③ tree search -> graph expansion

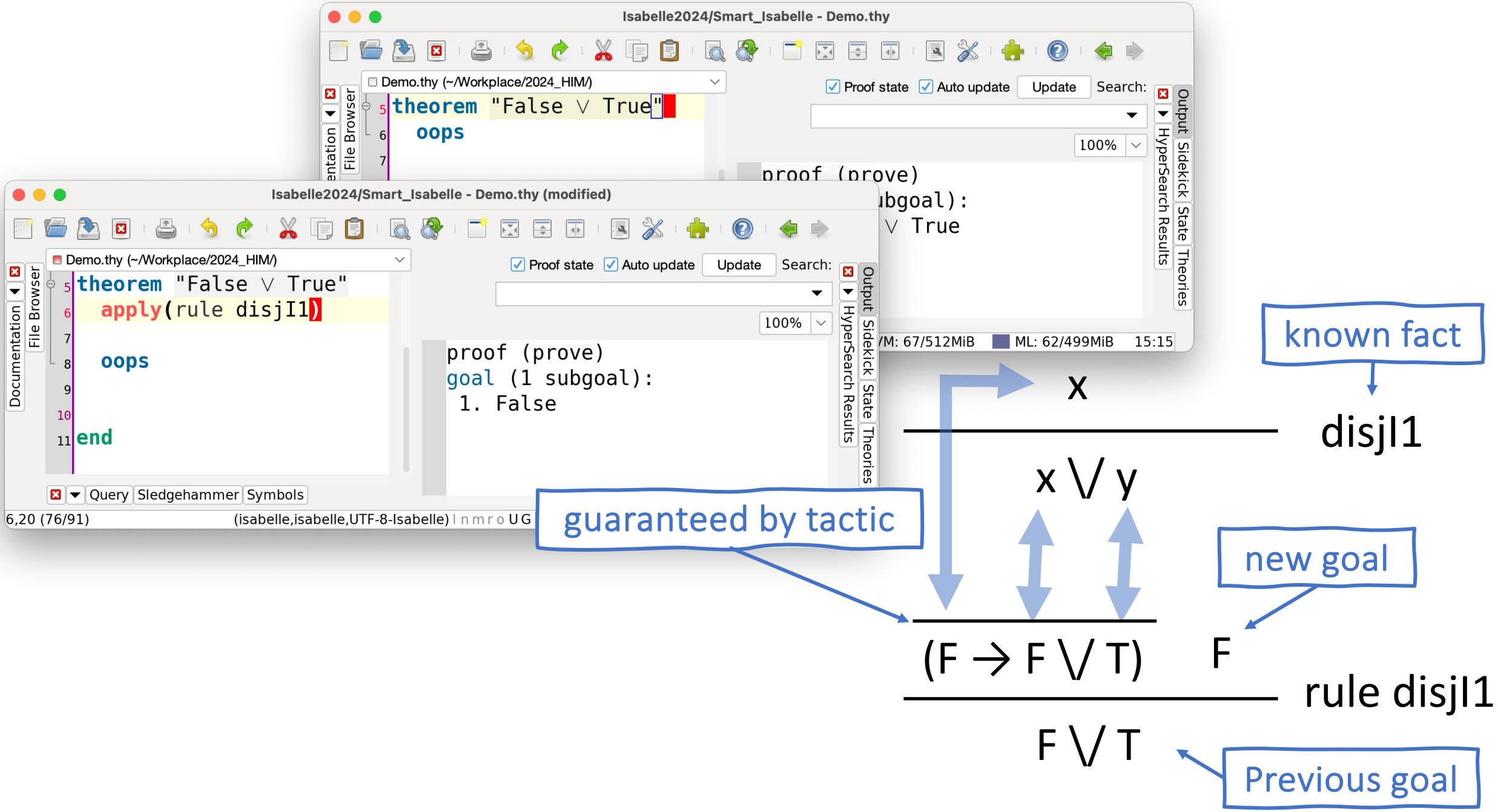
UR : United Reasoning

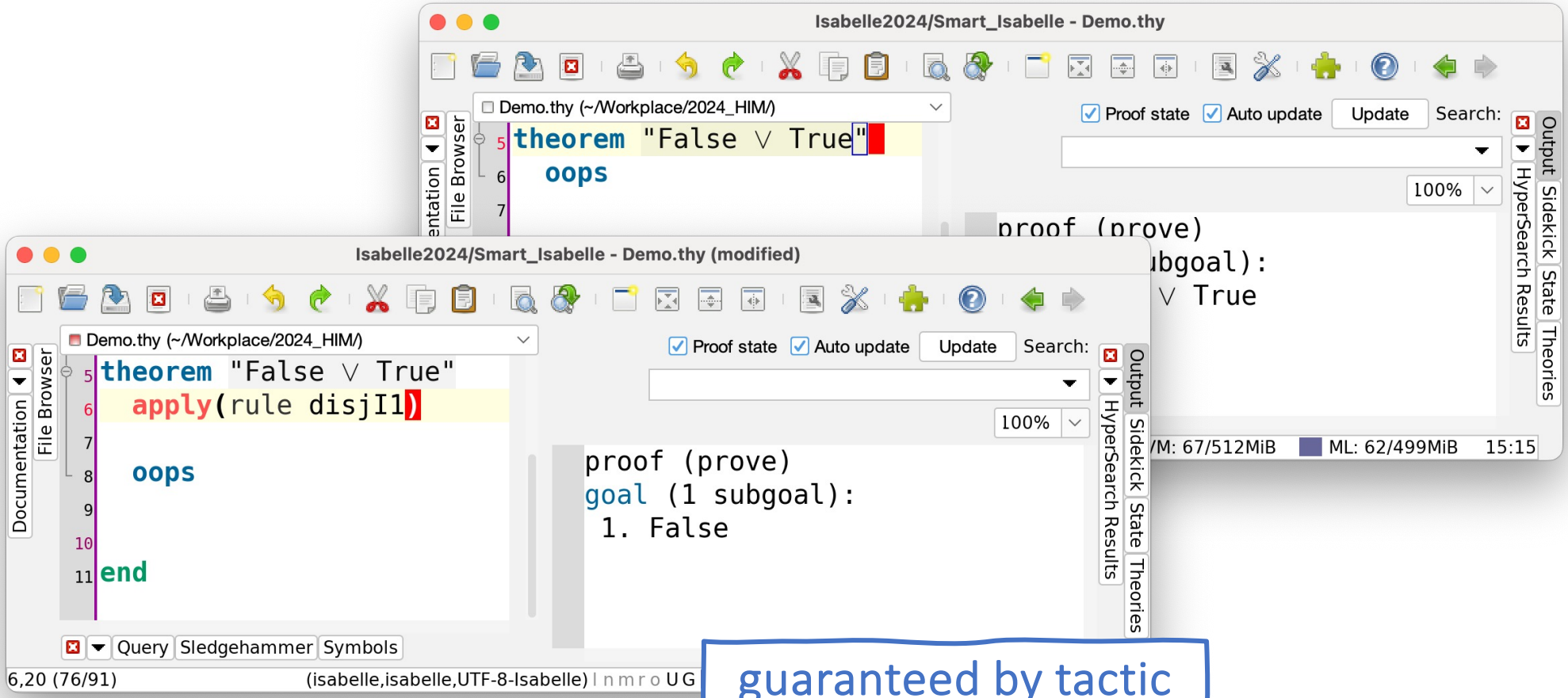
Visit:

https://youtu.be/rXU-lJxP_GI

LIVE DEMO

(We will come back at the end.)

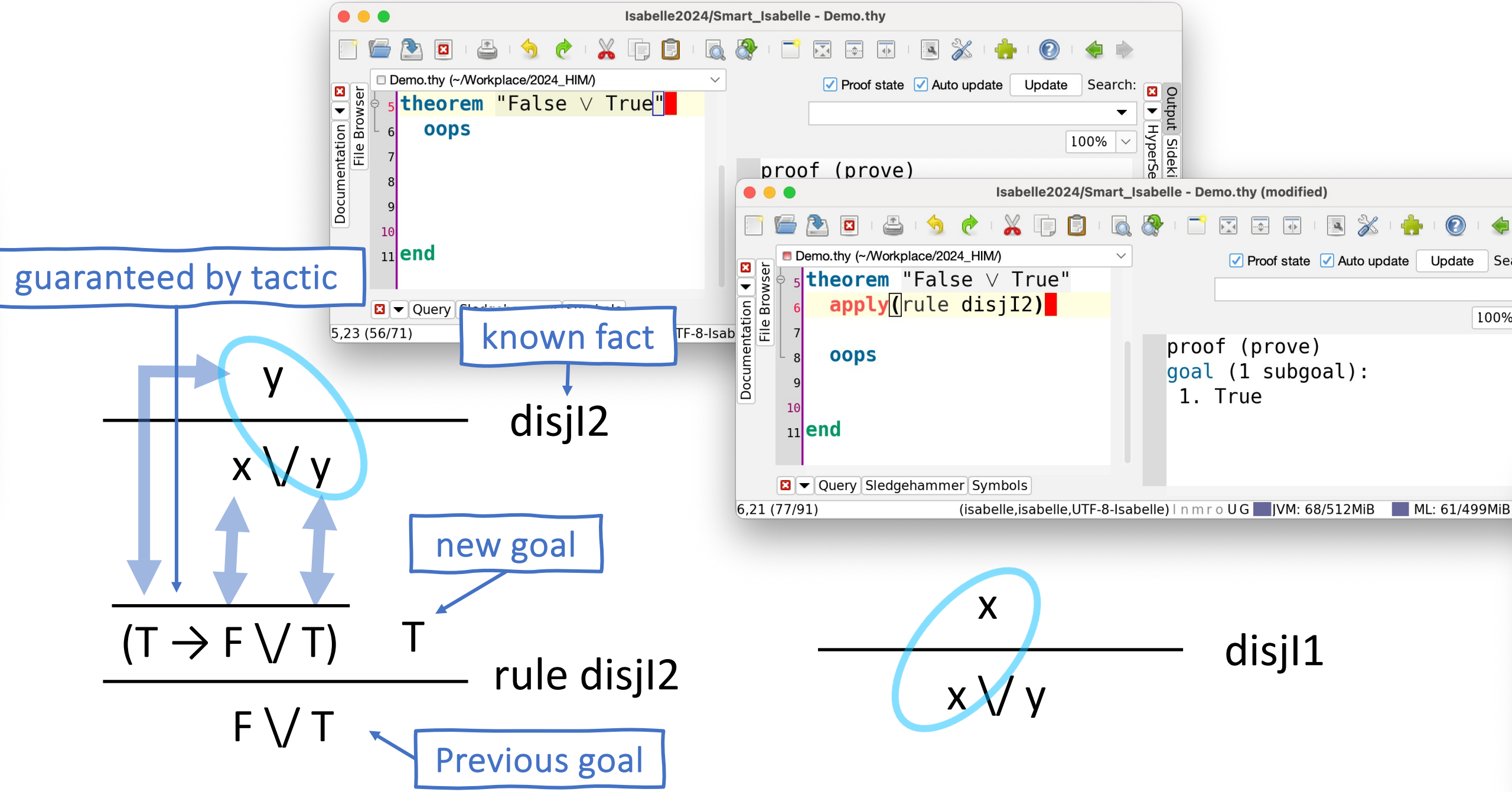

Isabelle2024/Smart_Isabelle - Demo.thy
Demo.thy (~/Workplace/2024_HIM/)
theorem "False ∨ True"
oops
Proof state
Auto update
Update
Search:
100%
Isabelle2024/Smart_Isabelle - Demo.thy (modified)
theorem "False ∨ True"
apply(rule disjI1)
oops
end
proof (prove)
goal (1 subgoal):
1. False
Query
Sledgehammer
Symbols
known fact
disjI1
x
x ∨ y
guaranteed by tactic
new goal
(F → F ∨ T)
F
rule disjI1
F ∨ T
Previous goal

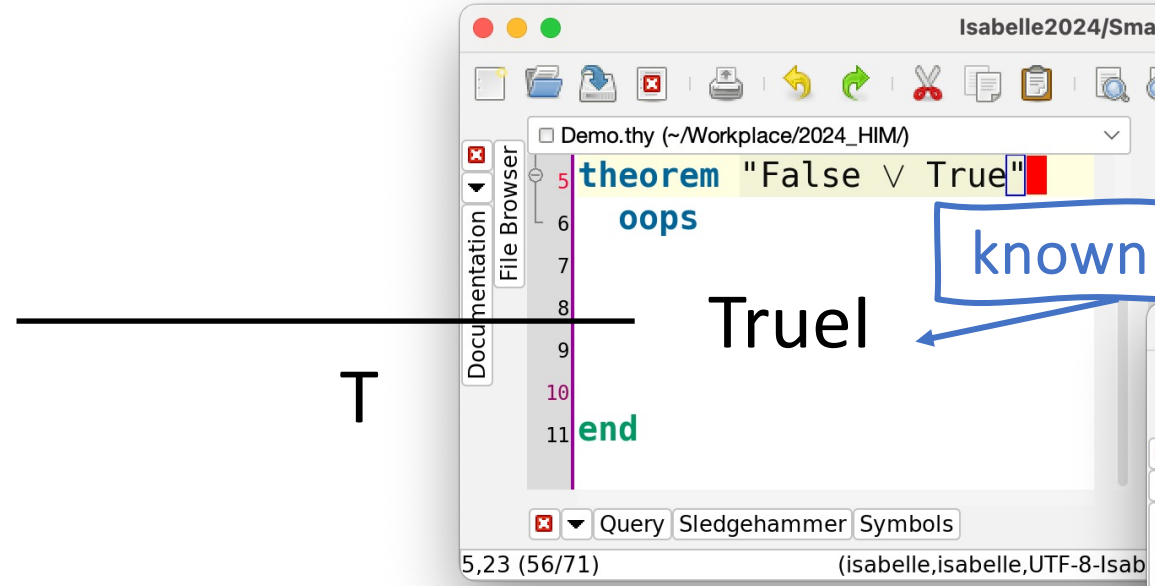


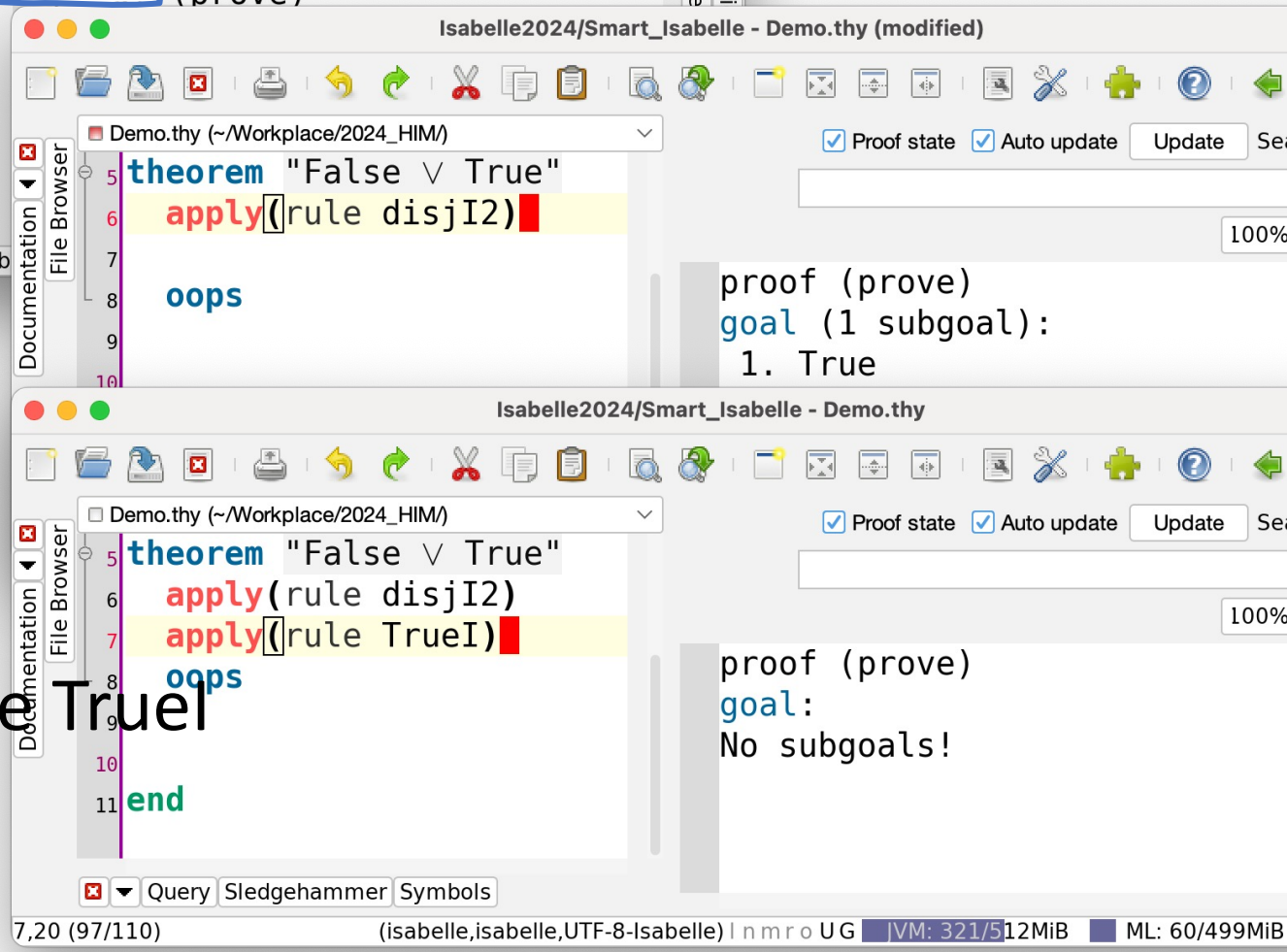
guaranteed by tactic

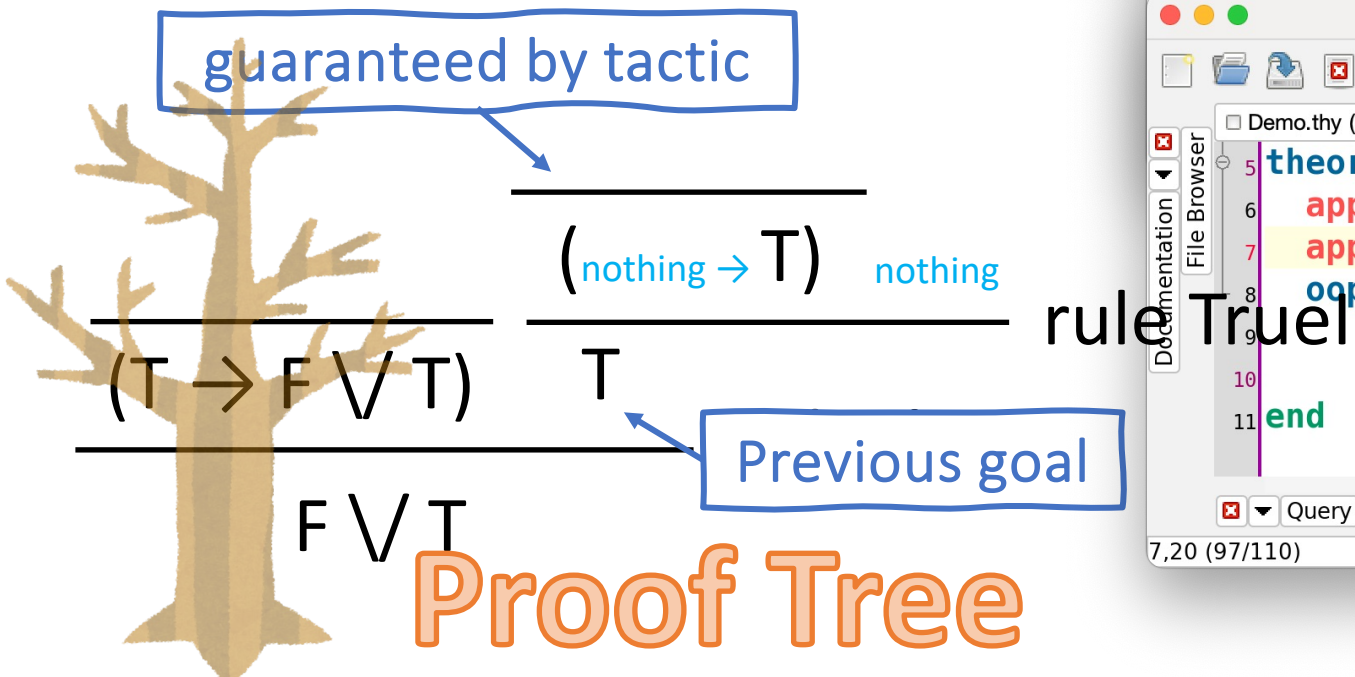
$$\frac{(F \to F \vee T) \qquad F}{F \vee T}$$

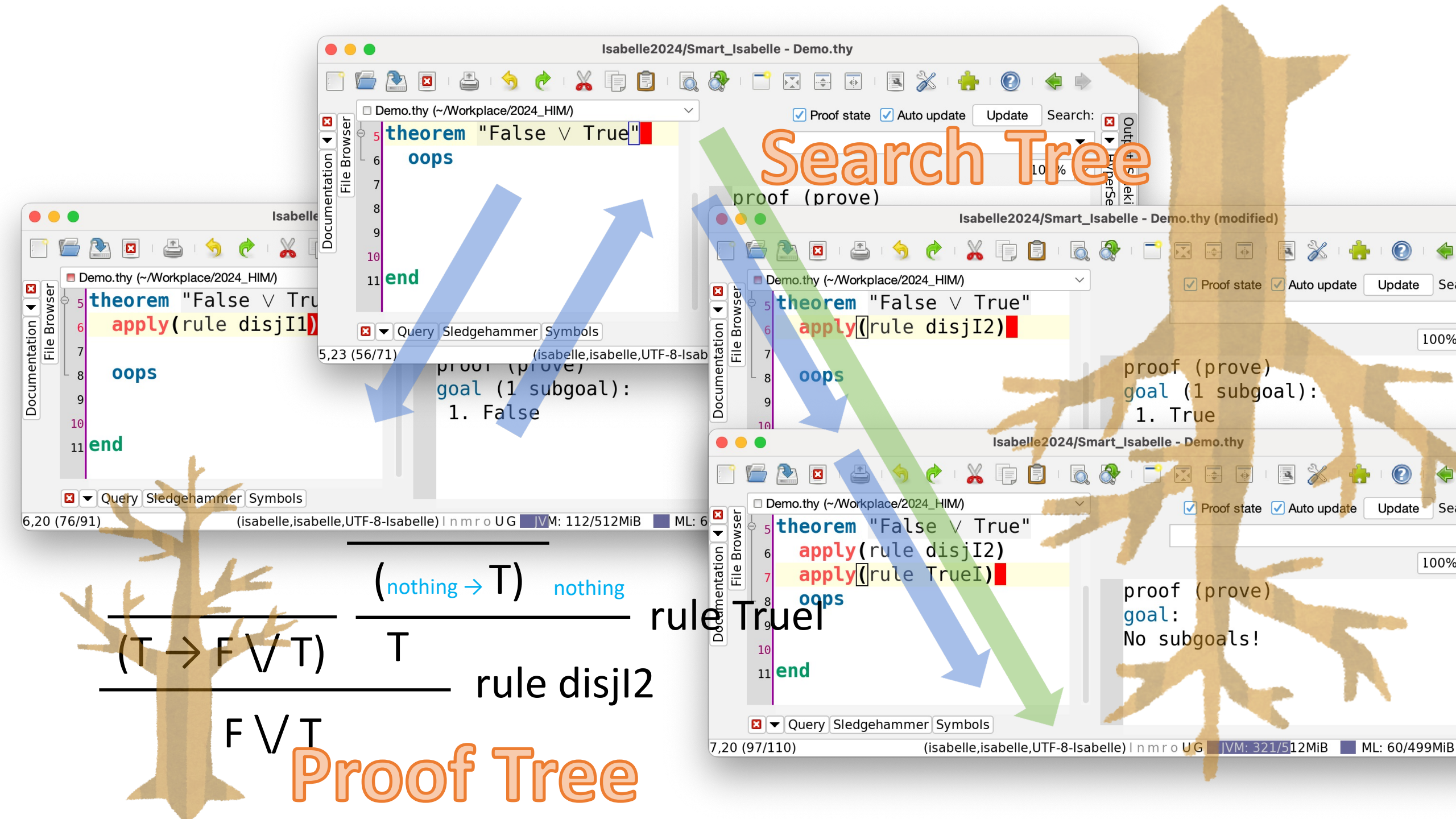
Previous goal


Isabelle2024/Smart_Isabelle - Demo.thy
Demo.thy (~/Workplace/2024_HIM/)
theorem "False ∨ True"
oops
end
Proof state
Auto update
Update
Search:
100%
proof (prove)
Documentation
File Browser
Output
Query
5,23 (56/71)
Isabelle2024/Smart_Isabelle - Demo.thy (modified)
Demo.thy (~/Workplace/2024_HIM/)
theorem "False ∨ True"
apply(rule disjI2)
oops
end
Query
Sledgehammer
Symbols
6,21 (77/91)
(isabelle,isabelle,UTF-8-Isabelle)
JVM: 68/512MiB
ML: 61/499MiB
proof (prove)
goal (1 subgoal):
1. True
guaranteed by tactic
known fact
y
x ∨ y
disjI2
new goal
(T → F ∨ T)
T
rule disjI2
F ∨ T
Previous goal
x
x ∨ y
disjI1


Isabelle2024/Smart_Isabelle - Demo.thy
Demo.thy (~/Workplace/2024_HIM/)
theorem "False ∨ True"
oops
end
Query
Sledgehammer
Symbols
5,23 (56/71)
(isabelle,isabelle,UTF-8-Isab
Proof state
Auto update
Update
Search:
100%
(prove)
known fact
T
Truel


Isabelle2024/Smart_Isabelle - Demo.thy (modified)
Demo.thy (~/Workplace/2024_HIM/)
theorem "False ∨ True"
apply(rule disjI2)
oops
Proof state
Auto update
Update
100%
proof (prove)
goal (1 subgoal):
1. True
Isabelle2024/Smart_Isabelle - Demo.thy
Demo.thy (~/Workplace/2024_HIM/)
theorem "False ∨ True"
apply(rule disjI2)
apply(rule TrueI)
oops
end
Query
Sledgehammer
Symbols
7,20 (97/110)
(isabelle,isabelle,UTF-8-Isabelle)
JVM: 321/512MiB
ML: 60/499MiB
proof (prove)
goal:
No subgoals!


guaranteed by tactic
(nothing → T)
nothing
rule Truel
(T → F ∨ T)
T
Previous goal
F ∨ T
Proof Tree


Search Tree
Isabelle2024/Smart_Isabelle - Demo.thy
Demo.thy (~/Workplace/2024_HIM/)
theorem "False ∨ True"
oops
end
Query Sledgehammer Symbols
Proof state Auto update Update Search:
proof (prove)
goal (1 subgoal):
1. False
Isabelle2024/Smart_Isabelle - Demo.thy (modified)
theorem "False ∨ True"
apply(rule disjI2)
oops
proof (prove)
goal (1 subgoal):
1. True
apply(rule TrueI)
proof (prove)
goal:
No subgoals!
(nothing → T)
nothing
T
rule TrueI
(T → F ∨ T)
T
rule disjI2
F ∨ T
Proof Tree

Search Tree

RESEARCH-ARTICLE

PaMpeR: proof method recommendation system for Isabelle/HOL

Authors: Yutaka Nagashima, Yilun He Authors Info & Affiliations

Publication: ASE 2018: Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering • September 2018 • Pages 362–372 • https://doi.org/10.1145/3238147.3238210

4 152

ASE 2018: Proceedings of the 33rd ACM/IEEE...
PaMpeR: proof method recommendation...
Pages 362–372

← Previous Next →

ABSTRACT
References
Index Terms
Comments

ACM DL DIGITAL LIBRARY

ABSTRACT

Deciding which sub-tool to use
interactive theorem prover (ITP
a <U>p</U>roof <U>m</U>eth
proof state, <pre>PaMpeR</pre
provides qualitative explanatio
generates these recommendati
transferring experienced users'
<pre>PaMpeR</pre> correctly
especially when it comes to spe

References

Faster Smarter Proof by Induction in Isabelle/HOL

Yutaka Nagashima

Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence
Main Track. Pages 1981-1988. https://doi.org/10.24963/ijcai.2021/273

IJCAI 2021 MONTREAL

PDF BibTeX

We present sem_ind, a recommendation tool for proof by induction in Isabelle/HOL. Given an inductive problem, sem_ind produces candidate arguments for proof by induction, and selects promising ones using heuristics. Our evaluation based on 1,095 inductive problems from 22 source files shows that sem_ind improves the accuracy of recommendation from 20.1% to 38.2% for the most promising candidates within 5.0 seconds of timeout compared to its predecessor while decreasing the median value of execution time from 2.79 seconds to 1.06 seconds.

File Edit Search Markers Folding View Utilities Macros Plugins Help

TIP_prop_06.thy (~/Workplace/Prod/Prod/)

```
| "t2 (S z2) z = S (t2 z2 z)"

prove property0 :
  "((length (rev (x y z))) = (t2 (length y) (length z)))"


lemma abduced_lemma_tactic_14045356: "var_0 = t2 var_0 Z"
  apply ( induct "var_0" )
  apply (simp_all ) done

lemma abduced_lemma_tactic_14045358: "(⋀a. t2 var_0 a = t2 a var_0) ⟹ S (t2 var_1 var_0) =
  apply ( induct "var_1" "var_0" rule : TIP_prop_06.t2.induct )
  apply (simp_all ) done

lemma abduced_lemma_commutativity_7642214: "t2 var_0 var_1 = t2 var_1 var_0"
  apply ( induct "var_0" arbitrary : var_1 )
  apply (simp_all )
  apply ( simp add : abduced_lemma_tactic_14045356 )
  using abduced_lemma_tactic_14045358 apply force done

lemma abduced_lemma_tactic_13498028: "length var_0 = length (x var_0 nil2)"
  apply ( induct "var_0" )
  apply (simp_all ) done

lemma abduced_lemma_remove_assumption_47001480: "S (length (x var_0 var_1)) = length (x var_0
  apply ( induct "var_0" arbitrary : var_1 )
  apply (simp_all ) done

lemma abduced_lemma_tactic_13498034:
"(⋀a. length (x var_0 a) = length (x a var_0)) ⟹ S (length (x var_1 var_0)) = length (x var_1 (
  apply ( simp add : abduced_lemma_remove_assumption_47001480 ) done

lemma abduced_lemma_composite_commutativity_7642270: "length (x var_0 var_1) = length (x var_1 var_0)"
  apply ( induct "var_0" arbitrary : var_1 )
  apply (simp_all )
  using abduced_lemma_tactic_13498028 apply blast
  apply ( simp add : abduced_lemma_tactic_13498034 ) done

lemma abduced_lemma_tactic_13293202: "length (rev var_0) = length var_0 ⟹ length (x (rev var_0)
  apply ( metis TIP_prop_06.length.simps ( 2 ) TIP_prop_06.x.simps ( 1 ) TIP_prop_06.x.simps ( 2
          abduced_lemma_composite_commutativity_7642270 ) done

lemma abduced_lemma_generalisation_then_extension_7642332: "length (rev var_0) = length var_0"
  apply ( induct "var_0" )
  apply (simp_all )
  apply ( simp add : abduced_lemma_tactic_13293202 ) done

lemma abduced_lemma_identity_7579838: "x var_0 nil2 = var_0"
  apply ( induct "var_0" )
  apply (simp_all ) done

lemma abduced_lemma_generalisation_then_extension_17745958: "length (rev (x var_0 nil2)) = len
  apply ( simp add : abduced_lemma_generalisation_then_extension_7642332 abduced_lemma_identit
  done

lemma abduced_lemma_tactic_33224946: "(⋀a. length (rev (x a var_0)) = t2 (length var_0) (leng
length (rev (x var_1 (cons2 var_2 var_0))) = S (t2 (length var_0) (length var_1))"
  apply ( metis TIP_prop_06.length.simps ( 2 ) TIP_prop_06.x.simps ( 2 ) abduced_lemma_com
  done

lemma abduced_lemma_generalise_by_renaming_7642286: "length (rev (x var_0 var_1)) = t2 (length var_0) (length var_1)"
  apply ( induct "var_1" arbitrary : var_0 )
  apply (simp_all )
  apply ( simp add : abduced_lemma_generalisation_then_extension_17745958 )
  apply ( simp add : abduced_lemma_tactic_33224946 )
  done

lemma original_goal_7579816: "length (rev (x var_0 var_1)) = t2 (length var_0) (length var_1)"
  apply ( simp add : abduced_lemma_commutativity_7642214 abduced_lemma_generalise_by_renaming_7642286 )
  done
```

Query Sledgehammer Symbols

Proof state | Auto update | Update | Search: | 100%

```
theorem
  original_goal_7579816:
    length (rev (x ?var_0.0 ?var_1.0)) =
    t2 (length ?var_0.0) (length ?var_1.0)
```

HyperSearch Results Output Sidekick State Theories

File Browser Documentation

auxiliary lemmas

final proof

Good lemma?

- ✅ readability
- ✅ used lemmas only

$$\frac{(P \to Q) \qquad P}{Q}$$ Modus ponens

1. Not obviously false.
2. Useful to prove the goal.
3. Easy to prove.

try to prove

quickcheck

easier

(conjecture→ goal) conjecture

Modus ponens

goal

difficult

Good lemma?

1. Not obviously false.
2. Useful to prove the goal.
3. Easy to prove.

try to prove

quickcheck

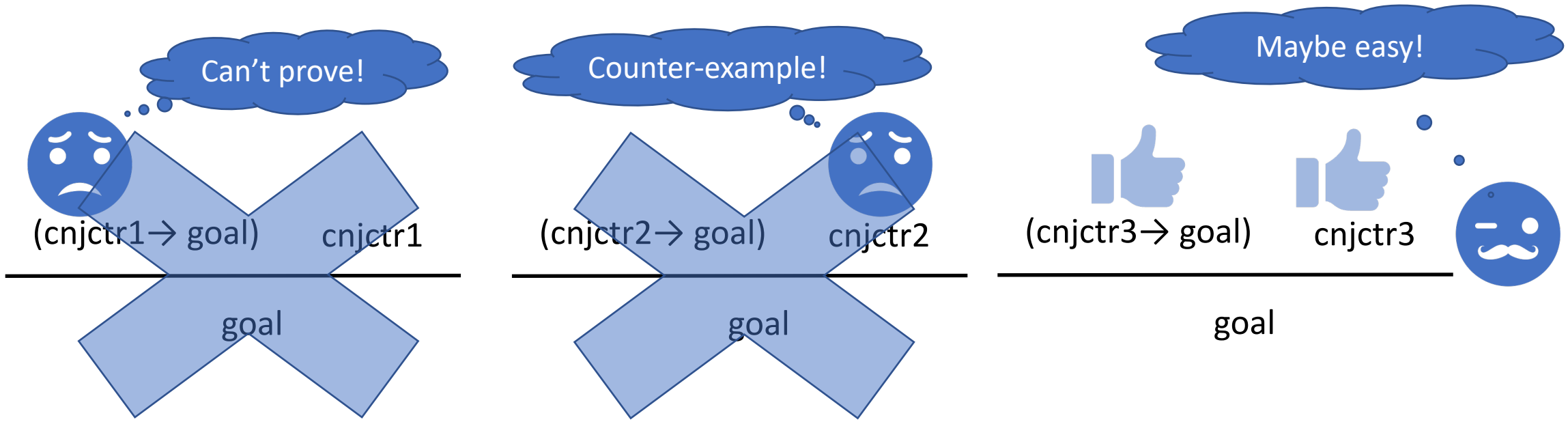
easier

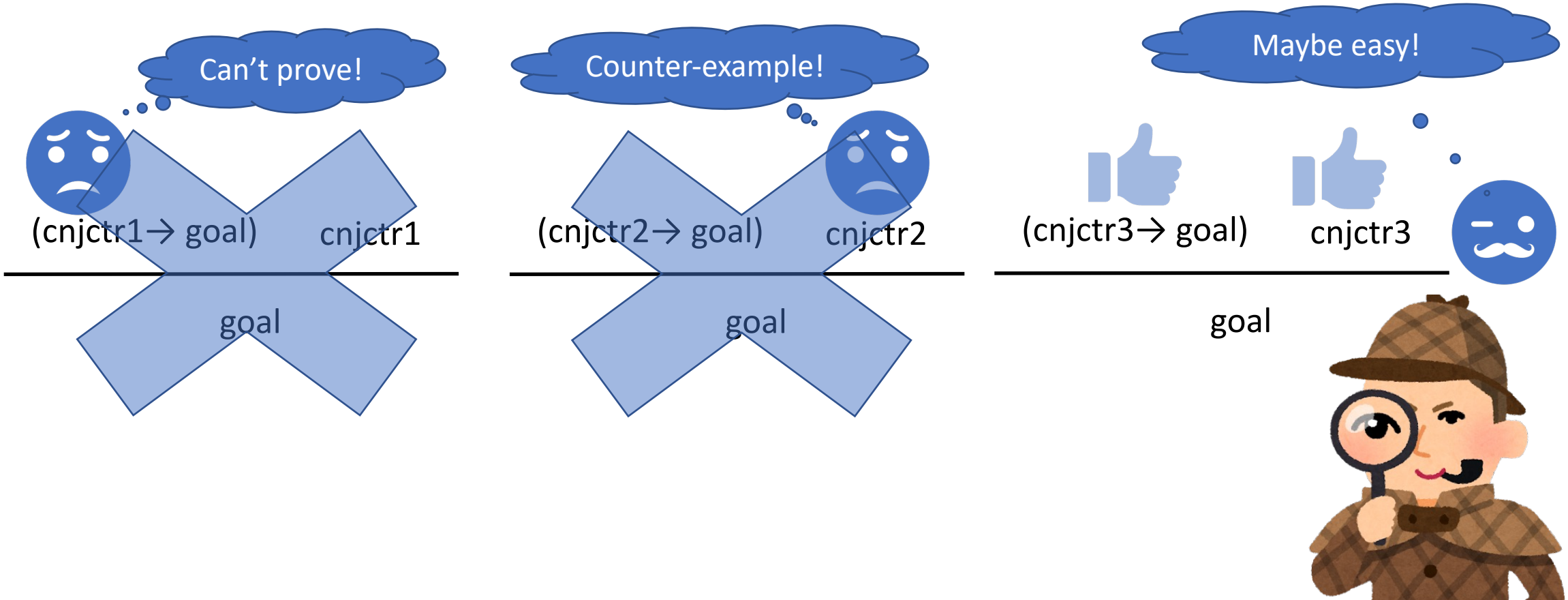
(conjecture→ goal) conjecture

Modus ponens

goal

1. Not obviously false.
2. Useful to prove the goal.
3. Easy to prove.


Can’t prove!
(cnjctr1→ goal)
cnjctr1
goal
Counter-example!
(cnjctr2→ goal)
cnjctr2
goal
Maybe easy!
(cnjctr3→ goal)
cnjctr3
goal


Can’t prove!
(cnjctr1→ goal)
cnjctr1
goal
Counter-example!
(cnjctr2→ goal)
cnjctr2
goal
Maybe easy!
(cnjctr3→ goal)
cnjctr3
goal

try to prove

quickcheck

easier

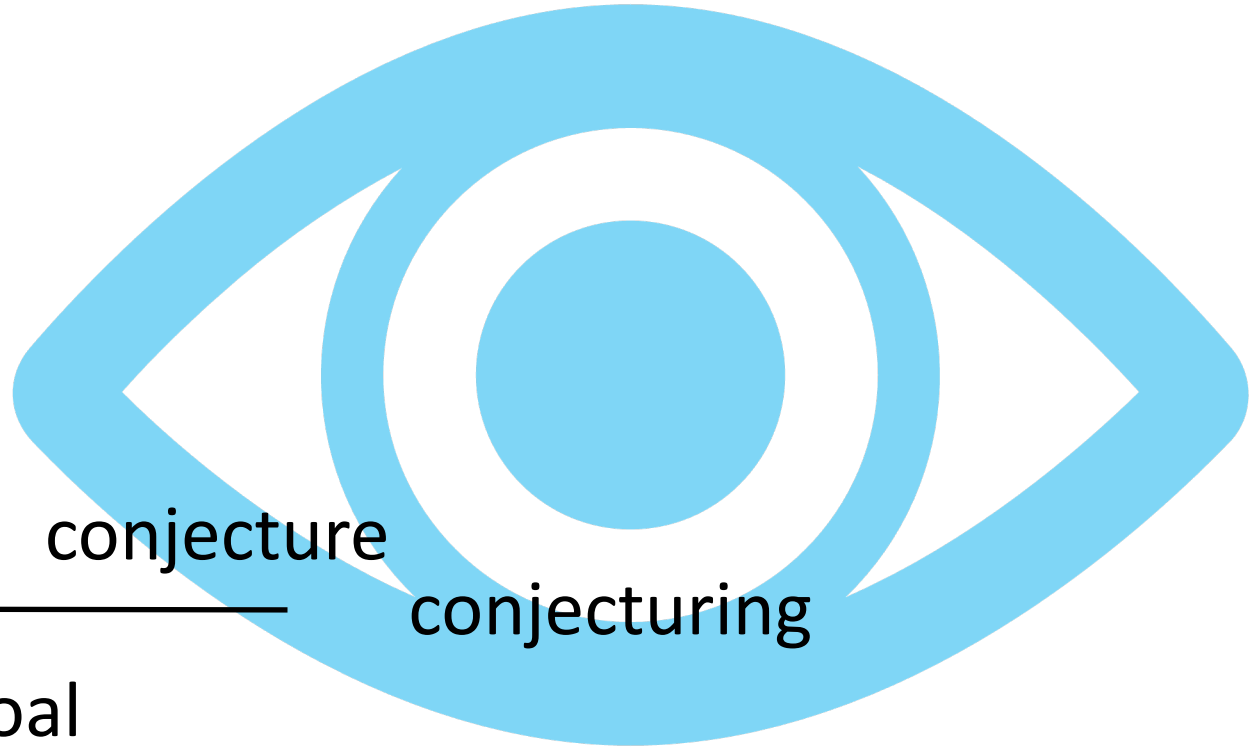
(conjecture→ goal) conjecture

conjecturing

goal

difficult

1. Not obviously false.
2. Useful to prove the goal.
3. Easy to prove.

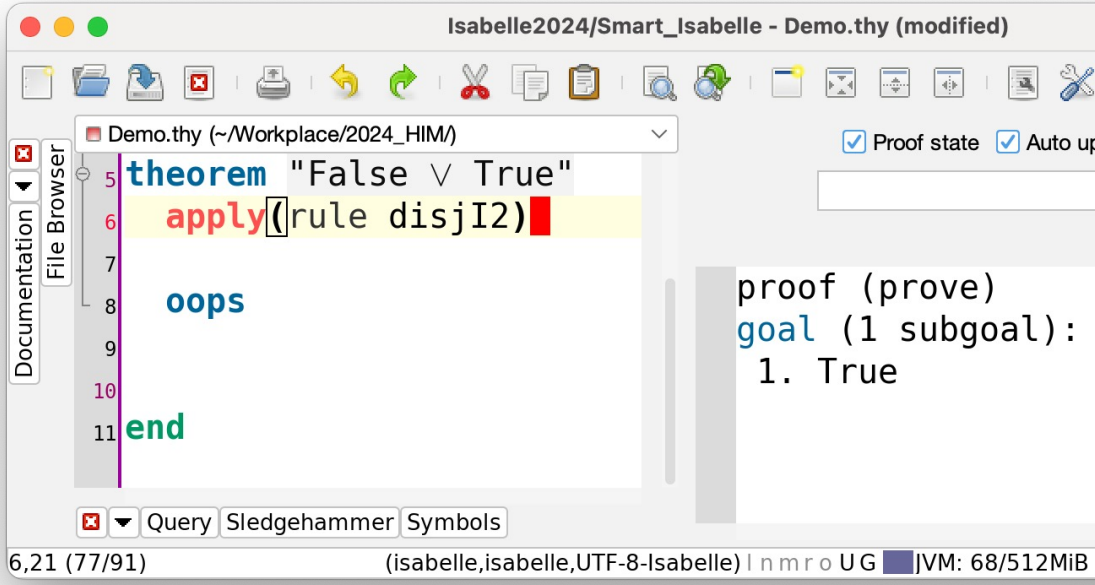


(conjecture→ goal)　conjecture

conjecturing

goal

$$\frac{(T \to F \vee T) \quad T}{F \vee T}\ \text{rule disjI2}$$



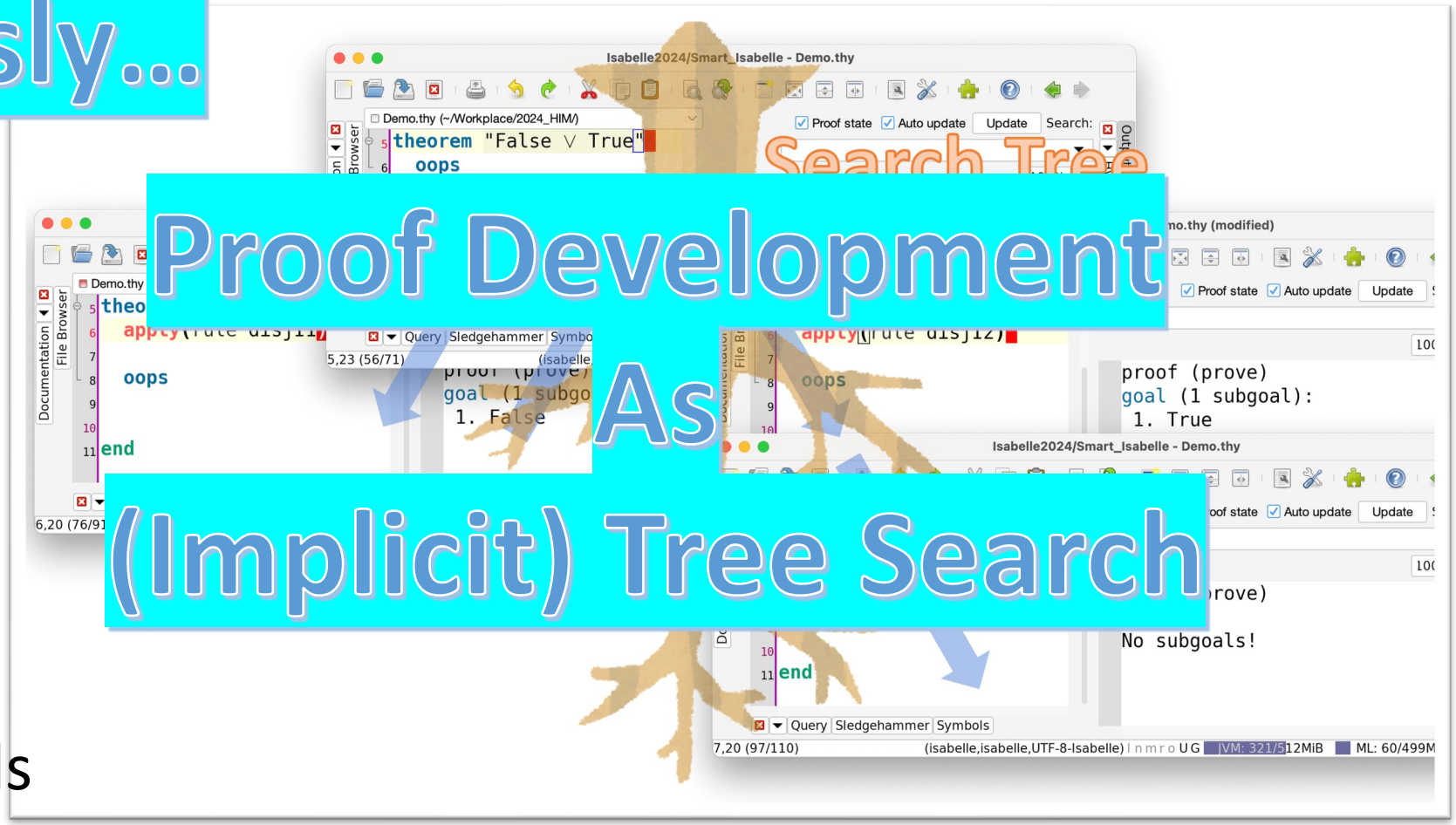
$$\frac{(\text{conjecture} \to \text{goal}) \quad \text{conjecture}}{\text{goal}}\ \text{conjecturing}$$

$$\frac{(\text{subgoal} \rightarrow \text{goal}) \quad \text{subgoal}}{\text{goal}}\ \text{tactic application}$$

$$\frac{(\text{conjecture} \rightarrow \text{goal}) \quad \text{conjecture}}{\text{goal}}\ \text{conjecturing}$$

tactic application as conjecturing

Previously...



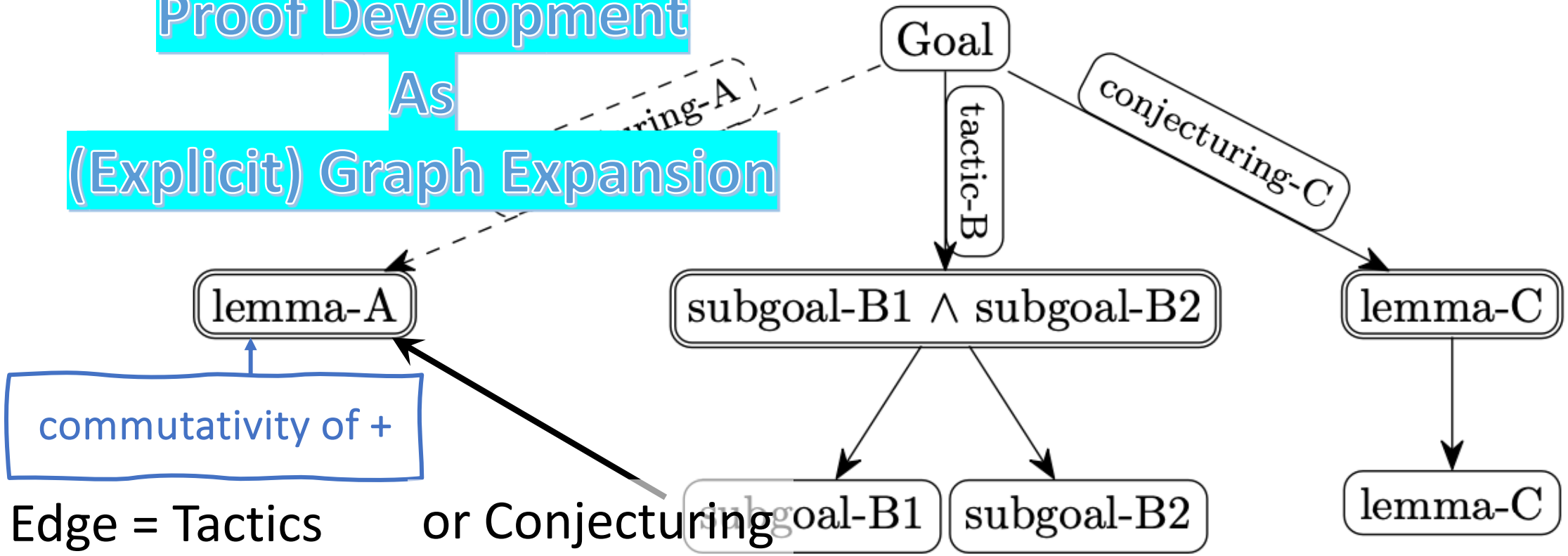
Edge = Tactics

Node = Subgoals

Aim: to find a path from the root to the node representing proof completion.

tactic application as conjecturing

Proof Development
As
(Explicit) Graph Expansion



Edge = Tactics or Conjecturing

Node = Subgoals or Auxiliary Lemma

Aim: to find a portion of the graph that represents a proof of the original goal.

LIVE DEMO

(We're almost at the end of this talk.)

File Edit Search Markers Folding View Utilities Macros Plugins Help

TIP_prop_06.thy (~/Workplace/Prod/Prod/)

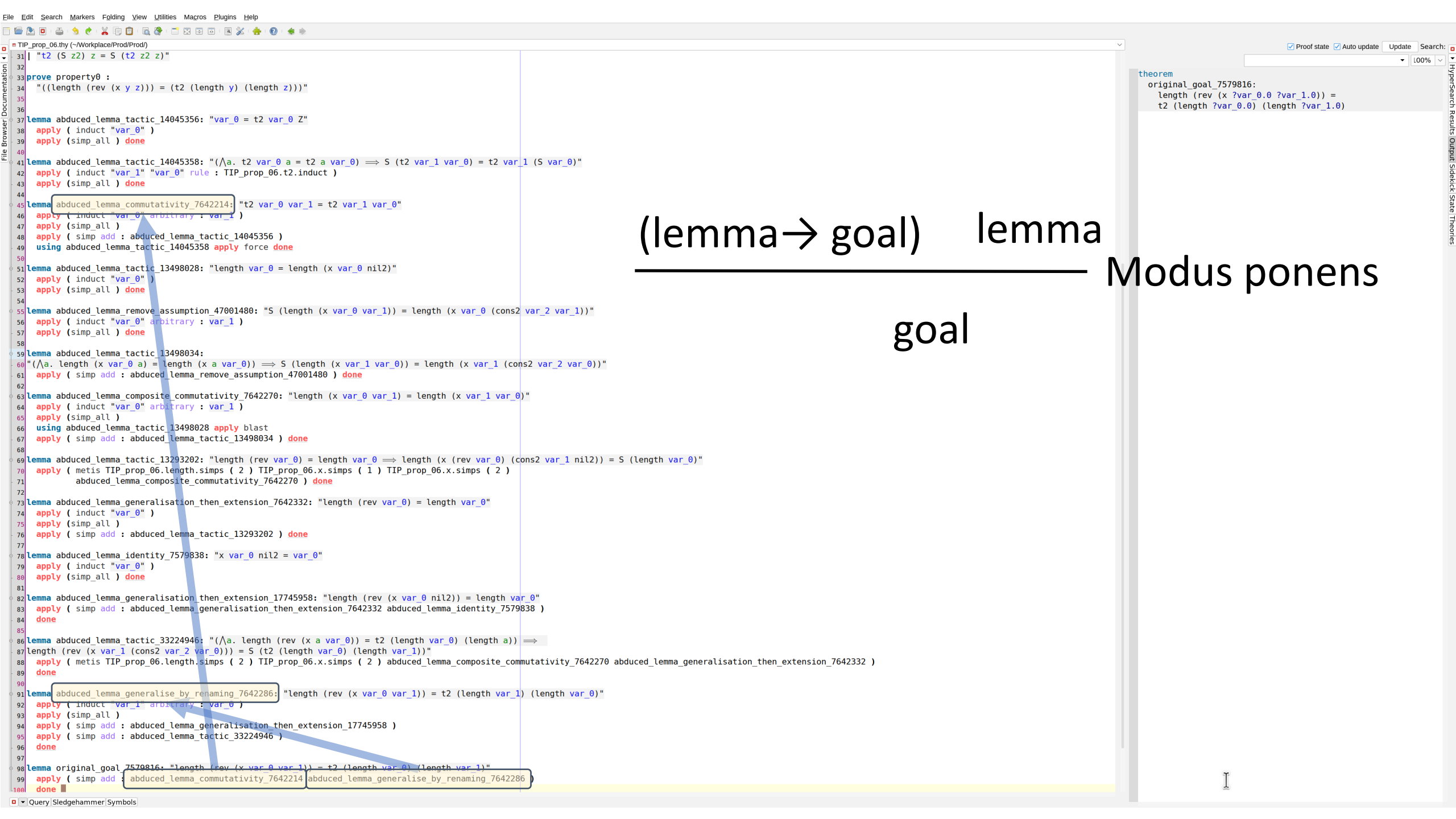
```
| "t2 (S z2) z = S (t2 z2 z)"

prove property0 :
  "((length (rev (x y z))) = (t2 (length y) (length z)))"


lemma abduced_lemma_tactic_14045356: "var_0 = t2 var_0 Z"
  apply ( induct "var_0" )
  apply (simp_all ) done

lemma abduced_lemma_tactic_14045358: "(⋀a. t2 var_0 a = t2 a var_0) ⟹ S (t2 var_1 var_0) = t2 var_1 (S var_0)"
  apply ( induct "var_1" "var_0" rule : TIP_prop_06.t2.induct )
  apply (simp_all ) done

lemma abduced_lemma_commutativity_7642214: "t2 var_0 var_1 = t2 var_1 var_0"
  apply ( induct "var_0" arbitrary : var_1 )
  apply (simp_all )
  apply ( simp add : abduced_lemma_tactic_14045356 )
  using abduced_lemma_tactic_14045358 apply force done

lemma abduced_lemma_tactic_13498028: "length var_0 = length (x var_0 nil2)"
  apply ( induct "var_0" )
  apply (simp_all ) done

lemma abduced_lemma_remove_assumption_47001480: "S (length (x var_0 var_1)) = length (x var_0 (cons2 var_2 var_1))"
  apply ( induct "var_0" arbitrary : var_1 )
  apply (simp_all ) done

lemma abduced_lemma_tactic_13498034:
"(⋀a. length (x var_0 a) = length (x a var_0)) ⟹ S (length (x var_1 var_0)) = length (x var_1 (cons2 var_2 var_0))"
  apply ( simp add : abduced_lemma_remove_assumption_47001480 ) done

lemma abduced_lemma_composite_commutativity_7642270: "length (x var_0 var_1) = length (x var_1 var_0)"
  apply ( induct "var_0" arbitrary : var_1 )
  apply (simp_all )
  using abduced_lemma_tactic_13498028 apply blast
  apply ( simp add : abduced_lemma_tactic_13498034 ) done

lemma abduced_lemma_tactic_13293202: "length (rev var_0) = length var_0 ⟹ length (x (rev var_0) (cons2 var_1 nil2)) = S (length var_0)"
  apply ( metis TIP_prop_06.length.simps ( 2 ) TIP_prop_06.x.simps ( 1 ) TIP_prop_06.x.simps ( 2 )
          abduced_lemma_composite_commutativity_7642270 ) done

lemma abduced_lemma_generalisation_then_extension_7642332: "length (rev var_0) = length var_0"
  apply ( induct "var_0" )
  apply (simp_all )
  apply ( simp add : abduced_lemma_tactic_13293202 ) done

lemma abduced_lemma_identity_7579838: "x var_0 nil2 = var_0"
  apply ( induct "var_0" )
  apply (simp_all ) done

lemma abduced_lemma_generalisation_then_extension_17745958: "length (rev (x var_0 nil2)) = length var_0"
  apply ( simp add : abduced_lemma_generalisation_then_extension_7642332 abduced_lemma_identity_7579838 )
  done

lemma abduced_lemma_tactic_33224946: "(⋀a. length (rev (x a var_0)) = t2 (length var_0) (length a)) ⟹
length (rev (x var_1 (cons2 var_2 var_0))) = S (t2 (length var_0) (length var_1))"
  apply ( metis TIP_prop_06.length.simps ( 2 ) TIP_prop_06.x.simps ( 2 ) abduced_lemma_composite_commutativity_7642270 abduced_lemma_generalisation_then_extension_7642332 )
  done

lemma abduced_lemma_generalise_by_renaming_7642286: "length (rev (x var_0 var_1)) = t2 (length var_1) (length var_0)"
  apply ( induct "var_1" arbitrary : var_0 )
  apply (simp_all )
  apply ( simp add : abduced_lemma_generalisation_then_extension_17745958 )
  apply ( simp add : abduced_lemma_tactic_33224946 )
  done

lemma original_goal_7579816: "length (rev (x var_0 var_1)) = t2 (length var_0) (length var_1)"
  apply ( simp add : abduced_lemma_commutativity_7642214 abduced_lemma_generalise_by_renaming_7642286 )
  done
```

Proof state | Auto update | Update | Search: | 100%

```
theorem
  original_goal_7579816:
    length (rev (x ?var_0.0 ?var_1.0)) =
    t2 (length ?var_0.0) (length ?var_1.0)
```

File Browser Documentation | HyperSearch Results Output Sidekick State Theories

Query Sledgehammer Symbols


File Edit Search Markers Folding View Utilities Macros Plugins Help
TIP_prop_06.thy (~/Workplace/Prod/Prod/)
Proof state
Auto update
Update
Search:
100%
theorem
original_goal_7579816:
length (rev (x ?var_0.0 ?var_1.0)) =
t2 (length ?var_0.0) (length ?var_1.0)
"t2 (S z2) z = S (t2 z2 z)"
prove property0 :
"((length (rev (x y z))) = (t2 (length y) (length z)))"
lemma abduced_lemma_tactic_14045356: "var_0 = t2 var_0 Z"
apply (induct "var_0")
apply (simp_all) done
lemma abduced_lemma_tactic_14045358: "(⋀a. t2 var_0 a = t2 a var_0) ⟹ S (t2 var_1 var_0) = t2 var_1 (S var_0)"
apply (induct "var_1" "var_0" rule : TIP_prop_06.t2.induct)
apply (simp_all) done
lemma abduced_lemma_commutativity_7642214: "t2 var_0 var_1 = t2 var_1 var_0"
apply (induct "var_0" arbitrary : var_1)
apply (simp_all)
apply (simp add : abduced_lemma_tactic_14045356)
using abduced_lemma_tactic_14045358 apply force done
lemma abduced_lemma_tactic_13498028: "length var_0 = length (x var_0 nil2)"
apply (induct "var_0")
apply (simp_all) done
lemma abduced_lemma_remove_assumption_47001480: "S (length (x var_0 var_1)) = length (x var_0 (cons2 var_2 var_1))"
apply (induct "var_0" arbitrary : var_1)
apply (simp_all) done
lemma abduced_lemma_tactic_13498034:
"(⋀a. length (x var_0 a) = length (x a var_0)) ⟹ S (length (x var_1 var_0)) = length (x var_1 (cons2 var_2 var_0))"
apply (simp add : abduced_lemma_remove_assumption_47001480) done
lemma abduced_lemma_composite_commutativity_7642270: "length (x var_0 var_1) = length (x var_1 var_0)"
apply (induct "var_0" arbitrary : var_1)
apply (simp_all)
using abduced_lemma_tactic_13498028 apply blast
apply (simp add : abduced_lemma_tactic_13498034) done
lemma abduced_lemma_tactic_13293202: "length (rev var_0) = length var_0 ⟹ length (x (rev var_0) (cons2 var_1 nil2)) = S (length var_0)"
apply (metis TIP_prop_06.length.simps (2) TIP_prop_06.x.simps (1) TIP_prop_06.x.simps (2)
abduced_lemma_composite_commutativity_7642270) done
lemma abduced_lemma_generalisation_then_extension_7642332: "length (rev var_0) = length var_0"
apply (induct "var_0")
apply (simp_all)
apply (simp add : abduced_lemma_tactic_13293202) done
lemma abduced_lemma_identity_7579838: "x var_0 nil2 = var_0"
apply (induct "var_0")
apply (simp_all) done
lemma abduced_lemma_generalisation_then_extension_17745958: "length (rev (x var_0 nil2)) = length var_0"
apply (simp add : abduced_lemma_generalisation_then_extension_7642332 abduced_lemma_identity_7579838)
done
lemma abduced_lemma_tactic_33224946: "(⋀a. length (rev (x a var_0)) = t2 (length var_0) (length a)) ⟹
length (rev (x var_1 (cons2 var_2 var_0))) = S (t2 (length var_0) (length var_1))"
apply (metis TIP_prop_06.length.simps (2) TIP_prop_06.x.simps (2) abduced_lemma_composite_commutativity_7642270 abduced_lemma_generalisation_then_extension_7642332)
done
lemma abduced_lemma_generalise_by_renaming_7642286: "length (rev (x var_0 var_1)) = t2 (length var_1) (length var_0)"
apply (induct "var_1" arbitrary : var_0)
apply (simp_all)
apply (simp add : abduced_lemma_generalisation_then_extension_17745958)
apply (simp add : abduced_lemma_tactic_33224946)
done
lemma original_goal_7579816: "length (rev (x var_0 var_1)) = t2 (length var_0) (length var_1)"
apply (simp add : abduced_lemma_commutativity_7642214 abduced_lemma_generalise_by_renaming_7642286)
done
Query Sledgehammer Symbols
(lemma→ goal) lemma
goal
Modus ponens

File Edit Search Markers Folding View Utilities Macros Plugins Help
TIP_prop_06.thy (~/Workplace/Prod/Prod/)
"t2 (S z2) z = S (t2 z2 z)"
prove property0 :
"((length (rev (x y z))) = (t2 (length y) (length z)))"
lemma abduced_lemma_tactic_14045356: "var_0 = t2 var_0 Z"
apply (induct "var_0")
apply (simp_all) done
lemma abduced_lemma_tactic_14045358: "(⋀a. t2 var_0 a = t2 a var_0) ⟹ S (t2 var_1 var_0) = t2 var_1 (S var_0)"
apply (induct "var_1" "var_0" rule : TIP_prop_06.t2.induct)
apply (simp_all) done
lemma abduced_lemma_commutativity_7642214: "t2 var_0 var_1 = t2 var_1 var_0"
apply (induct "var_0" arbitrary : var_1)
apply (simp_all)
apply (simp add : abduced_lemma_tactic_14045356)
using abduced_lemma_tactic_14045358 apply force done
lemma abduced_lemma_tactic_13498028: "length var_0 = length (x var_0 nil2)"
apply (induct "var_0")
apply (simp_all) done
lemma abduced_lemma_remove_assumption_47001480: "S (length (x var_0 var_1)) = length (x var_0 (cons2 var_2 var_1))"
apply (induct "var_0" arbitrary : var_1)
apply (simp_all) done
lemma abduced_lemma_tactic_13498034:
"(⋀a. length (x var_0 a) = length (x a var_0)) ⟹ S (length (x var_1 var_0)) = length (x var_1 (cons2 var_2 var_0))"
apply (simp add : abduced_lemma_remove_assumption_47001480) done
lemma abduced_lemma_composite_commutativity_7642270: "length (x var_0 var_1) = length (x var_1 var_0)"
apply (induct "var_0" arbitrary : var_1)
apply (simp_all)
using abduced_lemma_tactic_13498028 apply blast
apply (simp add : abduced_lemma_tactic_13498034) done
lemma abduced_lemma_tactic_13293202: "length (rev var_0) = length var_0 ⟹ length (x (rev var_0) (cons2 var_1 nil2)) = S (length var_0)"
apply (metis TIP_prop_06.length.simps (2) TIP_prop_06.x.simps (1) TIP_prop_06.x.simps (2)
abduced_lemma_composite_commutativity_7642270) done
lemma abduced_lemma_generalisation_then_extension_7642332: "length (rev var_0) = length var_0"
apply (induct "var_0")
apply (simp_all)
apply (simp add : abduced_lemma_tactic_13293202) done
lemma abduced_lemma_identity_7579838: "x var_0 nil2 = var_0"
apply (induct "var_0")
apply (simp_all) done
lemma abduced_lemma_generalisation_then_extension_17745958: "length (rev (x var_0 nil2)) = length var_0"
apply (simp add : abduced_lemma_generalisation_then_extension_7642332 abduced_lemma_identity_7579838)
done
lemma abduced_lemma_tactic_33224946: "(⋀a. length (rev (x a var_0)) = t2 (length var_0) (length a)) ⟹
length (rev (x var_1 (cons2 var_2 var_0))) = S (t2 (length var_0) (length var_1))"
apply (metis TIP_prop_06.length.simps (2) TIP_prop_06.x.simps (2) abduced_lemma_composite_commutativity_7642270 abduced_lemma_generalisation_then_extension_7642332)
done
lemma abduced_lemma_generalise_by_renaming_7642286: "length (rev (x var_0 var_1)) = t2 (length var_1) (length var_0)"
apply (induct "var_1" arbitrary : var_0)
apply (simp_all)
apply (simp add : abduced_lemma_generalisation_then_extension_17745958)
apply (simp add : abduced_lemma_tactic_33224946)
done
lemma original_goal_7579816: "length (rev (x var_0 var_1)) = t2 (length var_0) (length var_1)"
apply (simp add : abduced_lemma_commutativity_7642214 abduced_lemma_generalise_by_renaming_7642286)
done
Query Sledgehammer Symbols
Proof state
Auto update
Update
Search:
100%
theorem
original_goal_7579816:
length (rev (x ?var_0.0 ?var_1.0)) =
t2 (length ?var_0.0) (length ?var_1.0)
HyperSearch Results Output Sidekick State Theories
File Browser Documentation
(lemma→ goal)
lemma
goal
Modus ponens

File Edit Search Markers Folding View Utilities Macros Plugins Help

TIP_prop_06.thy (~/Workplace/Prod/Prod/)

```
| "t2 (S z2) z = S (t2 z2 z)"

prove property0 :
  "((length (rev (x y z))) = (t2 (length y) (length z)))"


lemma abduced_lemma_tactic_14045356: "var_0 = t2 var_0 Z"
  apply ( induct "var_0" )
  apply (simp_all ) done

lemma abduced_lemma_tactic_14045358: "(⋀a. t2 var_0 a = t2 a var_0) ⟹ S (t2 var_1 var_0) = t2 var_1 (S var_0)"
  apply ( induct "var_1" "var_0" rule : TIP_prop_06.t2.induct )
  apply (simp_all ) done

lemma abduced_lemma_commutativity_7642214: "t2 var_0 var_1 = t2 var_1 var_0"
  apply ( induct "var_0" arbitrary : var_1 )
  apply (simp_all )
  apply ( simp add : abduced_lemma_tactic_14045356 )
  using abduced_lemma_tactic_14045358 apply force done

lemma abduced_lemma_tactic_13498028: "length var_0 = length (x var_0 nil2)"
  apply ( induct "var_0" )
  apply (simp_all ) done

lemma abduced_lemma_remove_assumption_47001480: "S (length (x var_0 var_1)) = length (x var_0 (cons2 var_2 var_1))"
  apply ( induct "var_0" arbitrary : var_1 )
  apply (simp_all ) done

lemma abduced_lemma_tactic_13498034:
"(⋀a. length (x var_0 a) = length (x a var_0)) ⟹ S (length (x var_1 var_0)) = length (x var_1 (cons2 var_2 var_0))"
  apply ( simp add : abduced_lemma_remove_assumption_47001480 ) done

lemma abduced_lemma_composite_commutativity_7642270: "length (x var_0 var_1) = length (x var_1 var_0)"
  apply ( induct "var_0" arbitrary : var_1 )
  apply (simp_all )
  using abduced_lemma_tactic_13498028 apply blast
  apply ( simp add : abduced_lemma_tactic_13498034 ) done

lemma abduced_lemma_tactic_13293202: "length (rev var_0) = length var_0 ⟹ length (x (rev var_0) (cons2 var_1 nil2)) = S (length var_0)"
  apply ( metis TIP_prop_06.length.simps ( 2 ) TIP_prop_06.x.simps ( 1 ) TIP_prop_06.x.simps ( 2 )
          abduced_lemma_composite_commutativity_7642270 ) done

lemma abduced_lemma_generalisation_then_extension_7642332: "length (rev var_0) = length var_0"
  apply ( induct "var_0" )
  apply (simp_all )
  apply ( simp add : abduced_lemma_tactic_13293202 ) done

lemma abduced_lemma_identity_7579838: "x var_0 nil2 = var_0"
  apply ( induct "var_0" )
  apply (simp_all ) done

lemma abduced_lemma_generalisation_then_extension_17745958: "length (rev (x var_0 nil2)) = length var_0"
  apply ( simp add : abduced_lemma_generalisation_then_extension_7642332 abduced_lemma_identity_7579838 )
  done

lemma abduced_lemma_tactic_33224946: "(⋀a. length (rev (x a var_0)) = t2 (length var_0) (length a)) ⟹
length (rev (x var_1 (cons2 var_2 var_0))) = S (t2 (length var_0) (length var_1))"
  apply ( metis TIP_prop_06.length.simps ( 2 ) TIP_prop_06.x.simps ( 2 ) abduced_lemma_composite_commutativity_7642270 abduced_lemma_generalisation_then_extension_7642332 )
  done

lemma abduced_lemma_generalise_by_renaming_7642286: "length (rev (x var_0 var_1)) = t2 (length var_1) (length var_0)"
  apply ( induct "var_1" arbitrary : var_0 )
  apply (simp_all )
  apply ( simp add : abduced_lemma_generalisation_then_extension_17745958 )
  apply ( simp add : abduced_lemma_tactic_33224946 )
  done

lemma original_goal_7579816: "length (rev (x var_0 var_1)) = t2 (length var_0) (length var_1)"
  apply ( simp add : abduced_lemma_commutativity_7642214 abduced_lemma_generalise_by_renaming_7642286 )
  done
```

Query Sledgehammer Symbols

Proof state | Auto update | Update | Search: | 100%

```
theorem
  original_goal_7579816:
    length (rev (x ?var_0.0 ?var_1.0)) =
    t2 (length ?var_0.0) (length ?var_1.0)
```

HyperSearch Results Output Sidekick State Theories

File Browser Documentation

$$\frac{(\text{lemma} \rightarrow \text{goal}) \quad \text{lemma}}{\text{goal}}\ \text{Modus ponens}$$

graph instead of tree!

File Edit Search Markers Folding View Utilities Macros Plugins Help

TIP_prop_06.thy (~/Workplace/Prod/Prod/)

```
| "t2 (S z2) z = S (t2 z2 z)"

prove property0 :
  "((length (rev (x y z))) = (t2 (length y) (length z)))"


lemma abduced_lemma_tactic_14045356: "var_0 = t2 var_0 Z"
  apply ( induct "var_0" )
  apply (simp_all ) done

lemma abduced_lemma_tactic_14045358: "(⋀a. t2 var_0 a = t2 a var_0) ⟹ S (t2 var_1 var_0) = t2 var_1 (S var_0)"
  apply ( induct "var_1" "var_0" rule : TIP_prop_06.t2.induct )
  apply (simp_all ) done

lemma abduced_lemma_commutativity_7642214: "t2 var_0 var_1 = t2 var_1 var_0"
  apply ( induct "var_0" arbitrary : var_1 )
  apply (simp_all )
  apply ( simp add : abduced_lemma_tactic_14045356 )
  using abduced_lemma_tactic_14045358 apply force done

lemma abduced_lemma_tactic_13498028: "length var_0 = length (x var_0 nil2)"
  apply ( induct "var_0" )
  apply (simp_all ) done

lemma abduced_lemma_remove_assumption_47001480: "S (length (x var_0 var_1)) = length (x var_0 (cons2 var_2 var_1))"
  apply ( induct "var_0" arbitrary : var_1 )
  apply (simp_all ) done

lemma abduced_lemma_tactic_13498034:
"(⋀a. length (x var_0 a) = length (x a var_0)) ⟹ S (length (x var_1 var_0)) = length (x var_1 (cons2 var_2 var_0))"
  apply ( simp add : abduced_lemma_remove_assumption_47001480 ) done

lemma abduced_lemma_composite_commutativity_7642270: "length (x var_0 var_1) = length (x var_1 var_0)"
  apply ( induct "var_0" arbitrary : var_1 )
  apply (simp_all )
  using abduced_lemma_tactic_13498028 apply blast
  apply ( simp add : abduced_lemma_tactic_13498034 ) done

lemma abduced_lemma_tactic_13293202: "length (rev var_0) = length var_0 ⟹ length (x (rev var_0) (cons2 var_1 nil2)) = S (length var_0)"
  apply ( metis TIP_prop_06.length.simps ( 2 ) TIP_prop_06.x.simps ( 1 ) TIP_prop_06.x.simps ( 2 )
          abduced_lemma_composite_commutativity_7642270 ) done

lemma abduced_lemma_generalisation_then_extension_7642332: "length (rev var_0) = length var_0"
  apply ( induct "var_0" )
  apply (simp_all )
  apply ( simp add : abduced_lemma_tactic_13293202 ) done

lemma abduced_lemma_identity_7579838: "x var_0 nil2 = var_0"
  apply ( induct "var_0" )
  apply (simp_all ) done

lemma abduced_lemma_generalisation_then_extension_17745958: "length (rev (x var_0 nil2)) = length var_0"
  apply ( simp add : abduced_lemma_generalisation_then_extension_7642332 abduced_lemma_identity_7579838 )
  done

lemma abduced_lemma_tactic_33224946: "(⋀a. length (rev (x a var_0)) = t2 (length var_0) (length a)) ⟹
length (rev (x var_1 (cons2 var_2 var_0))) = S (t2 (length var_0) (length var_1))"
  apply ( metis TIP_prop_06.length.simps ( 2 ) TIP_prop_06.x.simps ( 2 ) abduced_lemma_composite_commutativity_7642270 abduced_lemma_generalisation_then_extension_7642332 )
  done

lemma abduced_lemma_generalise_by_renaming_7642286: "length (rev (x var_0 var_1)) = t2 (length var_1) (length var_0)"
  apply ( induct "var_1" arbitrary : var_0 )
  apply (simp_all )
  apply ( simp add : abduced_lemma_generalisation_then_extension_17745958 )
  apply ( simp add : abduced_lemma_tactic_33224946 )
  done

lemma original_goal_7579816: "length (rev (x var_0 var_1)) = t2 (length var_0) (length var_1)"
  apply ( simp add : abduced_lemma_commutativity_7642214 abduced_lemma_generalise_by_renaming_7642286 )
  done
```

Query Sledgehammer Symbols

Proof state | Auto update | Update | Search: | 100%

```
theorem
  original_goal_7579816:
    length (rev (x ?var_0.0 ?var_1.0)) =
    t2 (length ?var_0.0) (length ?var_1.0)
```

HyperSearch Results Output Sidekick State Theories

File Browser Documentation

$$\frac{(\text{lemma} \rightarrow \text{goal}) \quad \text{lemma}}{\text{goal}}\ \text{Modus ponens}$$

File Edit Search Markers Folding View Utilities Macros Plugins Help

TIP_prop_06.thy (~/Workplace/Prod/Prod/)

```
| "t2 (S z2) z = S (t2 z2 z)"

prove property0 :
  "((length (rev (x y z))) = (t2 (length y) (length z)))"


lemma abduced_lemma_tactic_14045356: "var_0 = t2 var_0 Z"
  apply ( induct "var_0" )
  apply (simp_all ) done

lemma abduced_lemma_tactic_14045358: "(⋀a. t2 var_0 a = t2 a var_0) ⟹ S (t2 var_1 var_0) = t2 var_1 (S var_0)"
  apply ( induct "var_1" "var_0" rule : TIP_prop_06.t2.induct )
  apply (simp_all ) done

lemma abduced_lemma_commutativity_7642214: "t2 var_0 var_1 = t2 var_1 var_0"
  apply ( induct "var_0" arbitrary : var_1 )
  apply (simp_all )
  apply ( simp add : abduced_lemma_tactic_14045356 )
  using abduced_lemma_tactic_14045358 apply force done

lemma abduced_lemma_tactic_13498028: "length var_0 = length (x var_0 nil2)"
  apply ( induct "var_0" )
  apply (simp_all ) done

lemma abduced_lemma_remove_assumption_47001480: "S (length (x var_0 var_1)) = length (x var_0 (cons2 var_2 var_1))"
  apply ( induct "var_0" arbitrary : var_1 )
  apply (simp_all ) done

lemma abduced_lemma_tactic_13498034:
"(⋀a. length (x var_0 a) = length (x a var_0)) ⟹ S (length (x var_1 var_0)) = length (x var_1 (cons2 var_2 var_0))"
  apply ( simp add : abduced_lemma_remove_assumption_47001480 ) done

lemma abduced_lemma_composite_commutativity_7642270: "length (x var_0 var_1) = length (x var_1 var_0)"
  apply ( induct "var_0" arbitrary : var_1 )
  apply (simp_all )
  using abduced_lemma_tactic_13498028 apply blast
  apply ( simp add : abduced_lemma_tactic_13498034 ) done

lemma abduced_lemma_tactic_13293202: "length (rev var_0) = length var_0 ⟹ length (x (rev var_0) (cons2 var_1 nil2)) = S (length var_0)"
  apply ( metis TIP_prop_06.length.simps ( 2 ) TIP_prop_06.x.simps ( 1 ) TIP_prop_06.x.simps ( 2 )
        abduced_lemma_composite_commutativity_7642270 ) done

lemma abduced_lemma_generalisation_then_extension_7642332: "length (rev var_0) = length var_0"
  apply ( induct "var_0" )
  apply (simp_all )
  apply ( simp add : abduced_lemma_tactic_13293202 ) done

lemma abduced_lemma_identity_7579838: "x var_0 nil2 = var_0"
  apply ( induct "var_0" )
  apply (simp_all ) done

lemma abduced_lemma_generalisation_then_extension_17745958: "length (rev (x var_0 nil2)) = length var_0"
  apply ( simp add : abduced_lemma_generalisation_then_extension_7642332 abduced_lemma_identity_7579838 )
  done

lemma abduced_lemma_tactic_33224946: "(⋀a. length (rev (x a var_0)) = t2 (length var_0) (length a)) ⟹
length (rev (x var_1 (cons2 var_2 var_0))) = S (t2 (length var_0) (length var_1))"
  apply ( metis TIP_prop_06.length.simps ( 2 ) TIP_prop_06.x.simps ( 2 ) abduced_lemma_composite_commutativity_7642270 abduced_lemma_generalisation_then_extension_7642332 )
  done

lemma abduced_lemma_generalise_by_renaming_7642286: "length (rev (x var_0 var_1)) = t2 (length var_1) (length var_0)"
  apply ( induct "var_1" arbitrary : var_0 )
  apply (simp_all )
  apply ( simp add : abduced_lemma_generalisation_then_extension_17745958 )
  apply ( simp add : abduced_lemma_tactic_33224946 )
  done

lemma original_goal_7579816: "length (rev (x var_0 var_1)) = t2 (length var_0) (length var_1)"
  apply ( simp add : abduced_lemma_commutativity_7642214 abduced_lemma_generalise_by_renaming_7642286 )
  done
```

Query Sledgehammer Symbols

Proof state | Auto update | Update | Search: | 100%

```
theorem
  original_goal_7579816:
    length (rev (x ?var_0.0 ?var_1.0)) =
    t2 (length ?var_0.0) (length ?var_1.0)
```

HyperSearch Results Output Sidekick State Theories

File Browser Documentation

$$\frac{(\text{lemma} \rightarrow \text{goal}) \quad \text{lemma}}{\text{goal}}\ \text{Modus ponens}$$

already checked it was proved

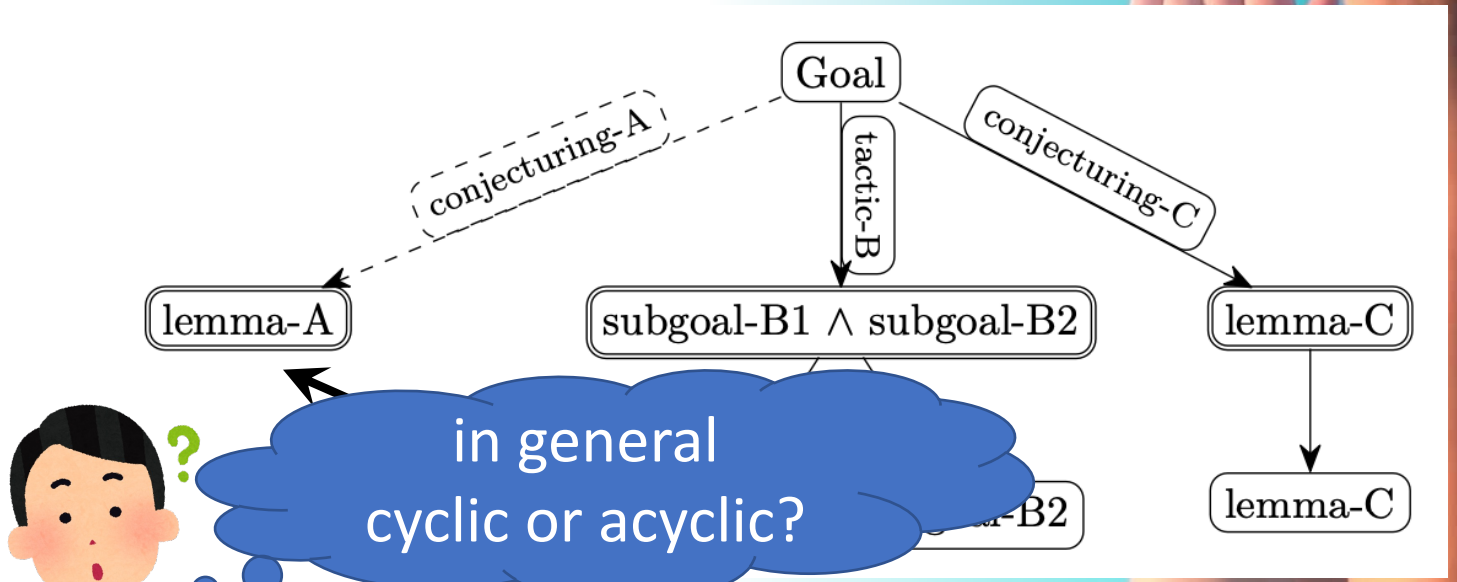
because it was used here.

graph instead of tree!

data61/PSL

Q & A

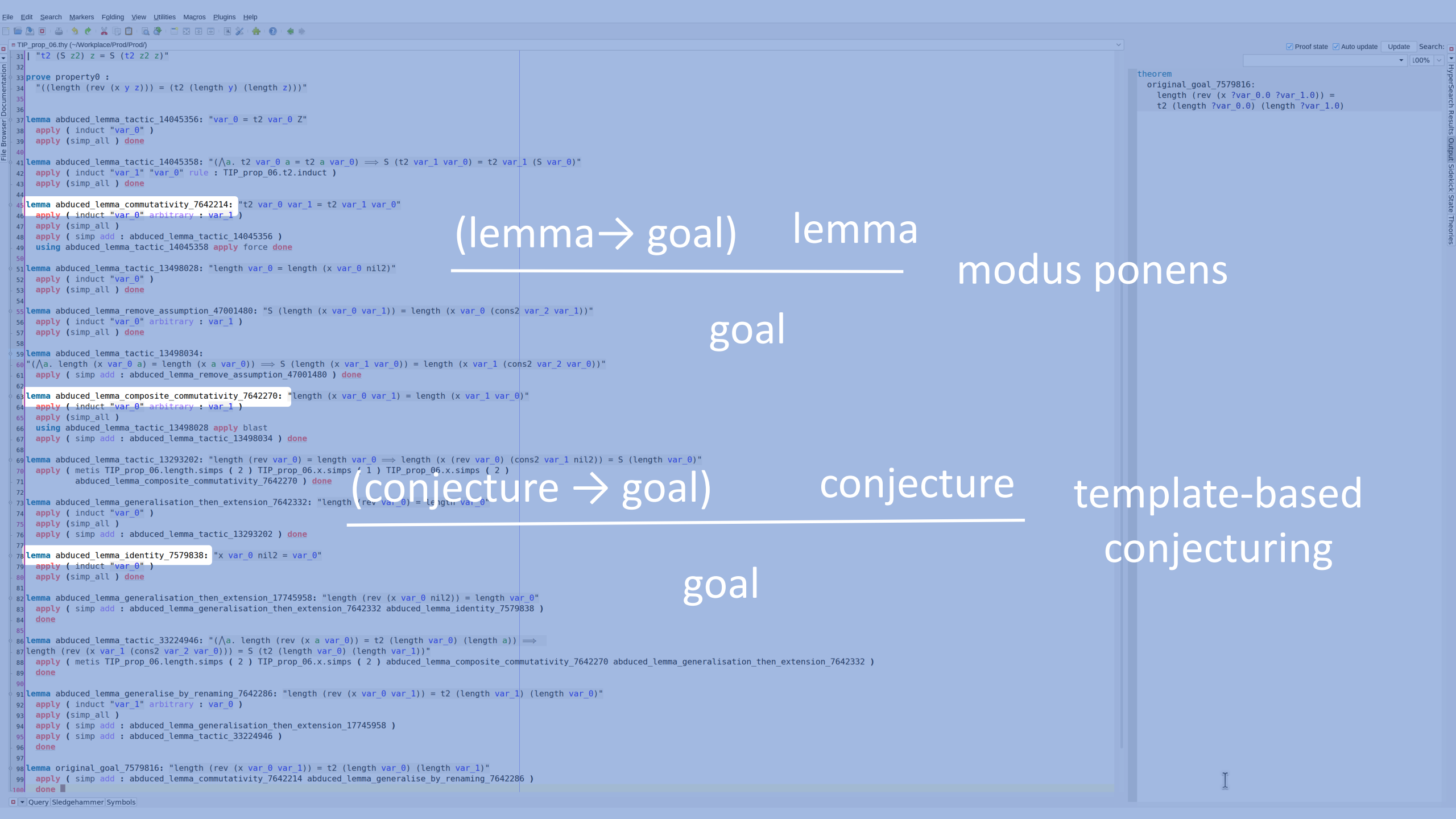
3 take-home lessons:

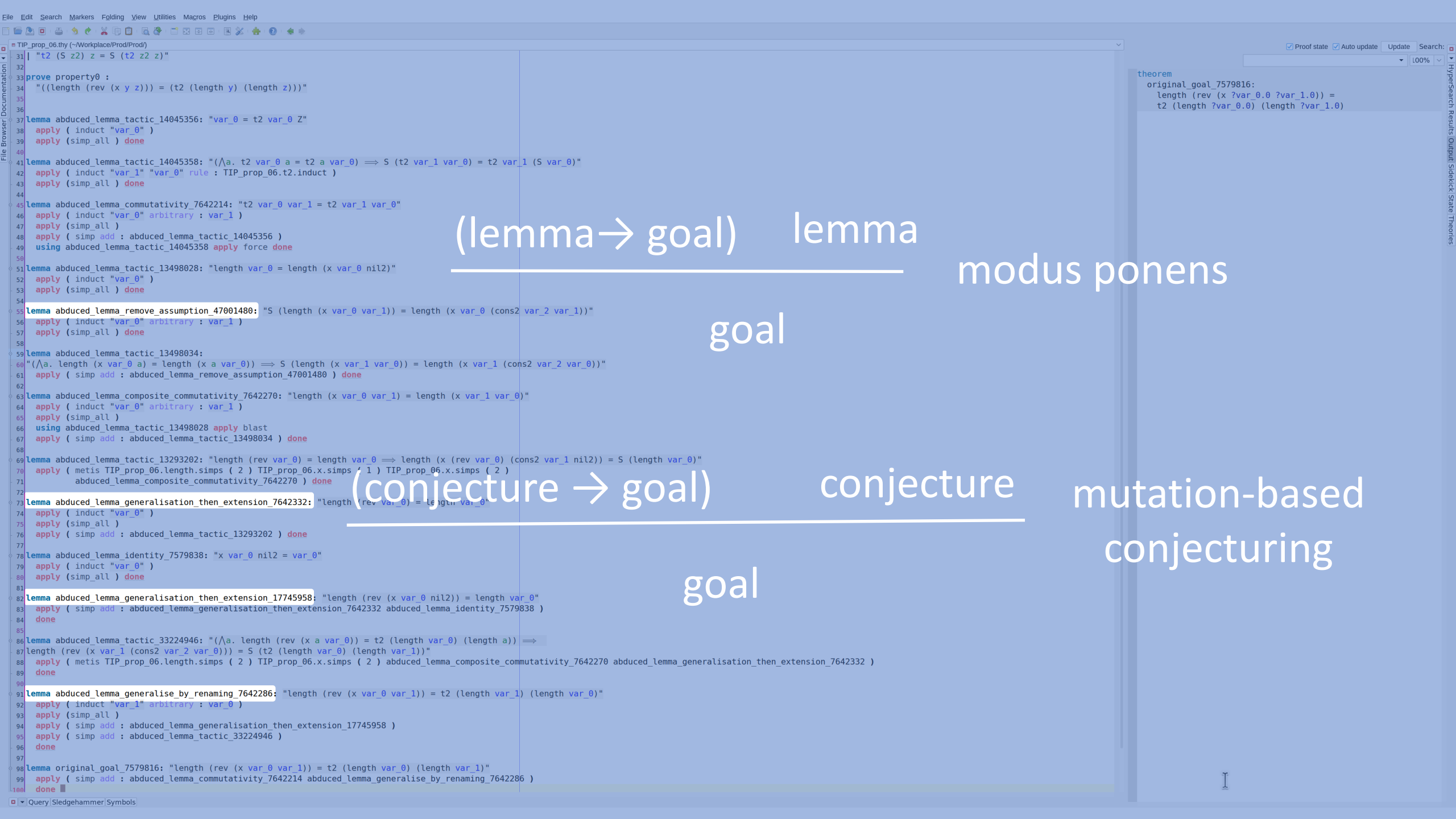


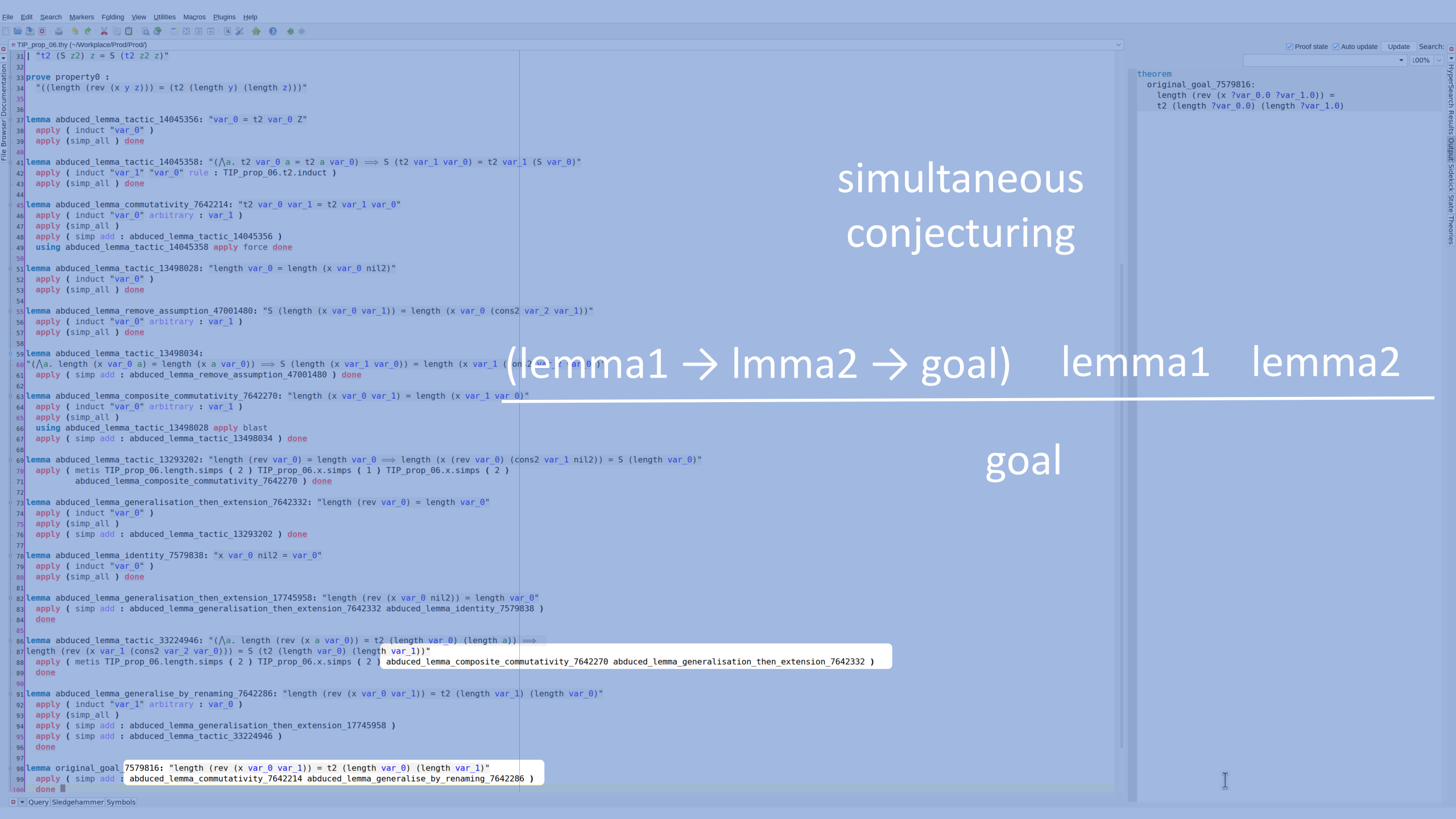
- Domain-Specific Language
- Smart Induction
- Evaluation Results
- Parallelism
- Many-Step Abduction
- Simultaneous Abduction
- Definitinal Quantifiers


MATHWORKS
1984
INTERNS SPEAK MATLAB


File Edit Search Markers Folding View Utilities Macros Plugins Help
TIP_prop_06.thy (~/Workplace/Prod/Prod/)
Proof state
Auto update
Update
Search:
100%
"t2 (S z2) z = S (t2 z2 z)"
prove property0 :
"((length (rev (x y z))) = (t2 (length y) (length z)))"
lemma abduced_lemma_tactic_14045356: "var_0 = t2 var_0 Z"
apply (induct "var_0")
apply (simp_all) done
lemma abduced_lemma_tactic_14045358: "(⋀a. t2 var_0 a = t2 a var_0) ⟹ S (t2 var_1 var_0) = t2 var_1 (S var_0)"
apply (induct "var_1" "var_0" rule : TIP_prop_06.t2.induct)
apply (simp_all) done
lemma abduced_lemma_commutativity_7642214: "t2 var_0 var_1 = t2 var_1 var_0"
apply (induct "var_0" arbitrary : var_1)
apply (simp_all)
apply (simp add : abduced_lemma_tactic_14045356)
using abduced_lemma_tactic_14045358 apply force done
lemma abduced_lemma_tactic_13498028: "length var_0 = length (x var_0 nil2)"
apply (induct "var_0")
apply (simp_all) done
lemma abduced_lemma_remove_assumption_47001480: "S (length (x var_0 var_1)) = length (x var_0 (cons2 var_2 var_1))"
apply (induct "var_0" arbitrary : var_1)
apply (simp_all) done
lemma abduced_lemma_tactic_13498034:
"(⋀a. length (x var_0 a) = length (x a var_0)) ⟹ S (length (x var_1 var_0)) = length (x var_1 (cons2 var_2 var_0))"
apply (simp add : abduced_lemma_remove_assumption_47001480) done
lemma abduced_lemma_composite_commutativity_7642270: "length (x var_0 var_1) = length (x var_1 var_0)"
apply (induct "var_0" arbitrary : var_1)
apply (simp_all)
using abduced_lemma_tactic_13498028 apply blast
apply (simp add : abduced_lemma_tactic_13498034) done
lemma abduced_lemma_tactic_13293202: "length (rev var_0) = length var_0 ⟹ length (x (rev var_0) (cons2 var_1 nil2)) = S (length var_0)"
apply (metis TIP_prop_06.length.simps (2) TIP_prop_06.x.simps (1) TIP_prop_06.x.simps (2)
abduced_lemma_composite_commutativity_7642270) done
lemma abduced_lemma_generalisation_then_extension_7642332: "length (rev var_0) = length var_0"
apply (induct "var_0")
apply (simp_all)
apply (simp add : abduced_lemma_tactic_13293202) done
lemma abduced_lemma_identity_7579838: "x var_0 nil2 = var_0"
apply (induct "var_0")
apply (simp_all) done
lemma abduced_lemma_generalisation_then_extension_17745958: "length (rev (x var_0 nil2)) = length var_0"
apply (simp add : abduced_lemma_generalisation_then_extension_7642332 abduced_lemma_identity_7579838)
done
lemma abduced_lemma_tactic_33224946: "(⋀a. length (rev (x a var_0)) = t2 (length var_0) (length a)) ⟹
length (rev (x var_1 (cons2 var_2 var_0))) = S (t2 (length var_0) (length var_1))"
apply (metis TIP_prop_06.length.simps (2) TIP_prop_06.x.simps (2) abduced_lemma_composite_commutativity_7642270 abduced_lemma_generalisation_then_extension_7642332)
done
lemma abduced_lemma_generalise_by_renaming_7642286: "length (rev (x var_0 var_1)) = t2 (length var_1) (length var_0)"
apply (induct "var_1" arbitrary : var_0)
apply (simp_all)
apply (simp add : abduced_lemma_generalisation_then_extension_17745958)
apply (simp add : abduced_lemma_tactic_33224946)
done
lemma original_goal_7579816: "length (rev (x var_0 var_1)) = t2 (length var_0) (length var_1)"
apply (simp add : abduced_lemma_commutativity_7642214 abduced_lemma_generalise_by_renaming_7642286)
done
theorem
original_goal_7579816:
length (rev (x ?var_0.0 ?var_1.0)) =
t2 (length ?var_0.0) (length ?var_1.0)
(lemma→ goal)
lemma
goal
modus ponens
(sub-goal → goal)
sub-goal
goal
tactic
application
Query Sledgehammer Symbols
File Browser Documentation
HyperSearch Results Output Sidekick State Theories


File Edit Search Markers Folding View Utilities Macros Plugins Help
TIP_prop_06.thy (~/Workplace/Prod/Prod/)
Proof state
Auto update
Update
Search:
100%
"t2 (S z2) z = S (t2 z2 z)"
prove property0 :
"((length (rev (x y z))) = (t2 (length y) (length z)))"
lemma abduced_lemma_tactic_14045356: "var_0 = t2 var_0 Z"
apply (induct "var_0")
apply (simp_all) done
lemma abduced_lemma_tactic_14045358: "(⋀a. t2 var_0 a = t2 a var_0) ⟹ S (t2 var_1 var_0) = t2 var_1 (S var_0)"
apply (induct "var_1" "var_0" rule : TIP_prop_06.t2.induct)
apply (simp_all) done
lemma abduced_lemma_commutativity_7642214: "t2 var_0 var_1 = t2 var_1 var_0"
apply (induct "var_0" arbitrary : var_1)
apply (simp_all)
apply (simp add : abduced_lemma_tactic_14045356)
using abduced_lemma_tactic_14045358 apply force done
lemma abduced_lemma_tactic_13498028: "length var_0 = length (x var_0 nil2)"
apply (induct "var_0")
apply (simp_all) done
lemma abduced_lemma_remove_assumption_47001480: "S (length (x var_0 var_1)) = length (x var_0 (cons2 var_2 var_1))"
apply (induct "var_0" arbitrary : var_1)
apply (simp_all) done
lemma abduced_lemma_tactic_13498034:
"(⋀a. length (x var_0 a) = length (x a var_0)) ⟹ S (length (x var_1 var_0)) = length (x var_1 (cons2 var_2 var_0))"
apply (simp add : abduced_lemma_remove_assumption_47001480) done
lemma abduced_lemma_composite_commutativity_7642270: "length (x var_0 var_1) = length (x var_1 var_0)"
apply (induct "var_0" arbitrary : var_1)
apply (simp_all)
using abduced_lemma_tactic_13498028 apply blast
apply (simp add : abduced_lemma_tactic_13498034) done
lemma abduced_lemma_tactic_13293202: "length (rev var_0) = length var_0 ⟹ length (x (rev var_0) (cons2 var_1 nil2)) = S (length var_0)"
apply (metis TIP_prop_06.length.simps (2) TIP_prop_06.x.simps (1) TIP_prop_06.x.simps (2)
abduced_lemma_composite_commutativity_7642270) done
lemma abduced_lemma_generalisation_then_extension_7642332: "length (rev var_0) = length var_0"
apply (induct "var_0")
apply (simp_all)
apply (simp add : abduced_lemma_tactic_13293202) done
lemma abduced_lemma_identity_7579838: "x var_0 nil2 = var_0"
apply (induct "var_0")
apply (simp_all) done
lemma abduced_lemma_generalisation_then_extension_17745958: "length (rev (x var_0 nil2)) = length var_0"
apply (simp add : abduced_lemma_generalisation_then_extension_7642332 abduced_lemma_identity_7579838)
done
lemma abduced_lemma_tactic_33224946: "(⋀a. length (rev (x a var_0)) = t2 (length var_0) (length a)) ⟹
length (rev (x var_1 (cons2 var_2 var_0))) = S (t2 (length var_0) (length var_1))"
apply (metis TIP_prop_06.length.simps (2) TIP_prop_06.x.simps (2) abduced_lemma_composite_commutativity_7642270 abduced_lemma_generalisation_then_extension_7642332)
done
lemma abduced_lemma_generalise_by_renaming_7642286: "length (rev (x var_0 var_1)) = t2 (length var_1) (length var_0)"
apply (induct "var_1" arbitrary : var_0)
apply (simp_all)
apply (simp add : abduced_lemma_generalisation_then_extension_17745958)
apply (simp add : abduced_lemma_tactic_33224946)
done
lemma original_goal_7579816: "length (rev (x var_0 var_1)) = t2 (length var_0) (length var_1)"
apply (simp add : abduced_lemma_commutativity_7642214 abduced_lemma_generalise_by_renaming_7642286)
done
theorem
original_goal_7579816:
length (rev (x ?var_0.0 ?var_1.0)) =
t2 (length ?var_0.0) (length ?var_1.0)
(lemma → goal)
lemma
goal
modus ponens
(conjecture → goal)
conjecture
goal
template-based
conjecturing
Query Sledgehammer Symbols
File Browser Documentation
HyperSearch Results Output Sidekick State Theories


File Edit Search Markers Folding View Utilities Macros Plugins Help
TIP_prop_06.thy (~/Workplace/Prod/Prod/)
Proof state
Auto update
Update
Search:
"t2 (S z2) z = S (t2 z2 z)"
prove property0 :
"((length (rev (x y z))) = (t2 (length y) (length z)))"
lemma abduced_lemma_tactic_14045356: "var_0 = t2 var_0 Z"
apply (induct "var_0")
apply (simp_all) done
lemma abduced_lemma_tactic_14045358: "(⋀a. t2 var_0 a = t2 a var_0) ⟹ S (t2 var_1 var_0) = t2 var_1 (S var_0)"
apply (induct "var_1" "var_0" rule : TIP_prop_06.t2.induct)
apply (simp_all) done
lemma abduced_lemma_commutativity_7642214: "t2 var_0 var_1 = t2 var_1 var_0"
apply (induct "var_0" arbitrary : var_1)
apply (simp_all)
apply (simp add : abduced_lemma_tactic_14045356)
using abduced_lemma_tactic_14045358 apply force done
lemma abduced_lemma_tactic_13498028: "length var_0 = length (x var_0 nil2)"
apply (induct "var_0")
apply (simp_all) done
lemma abduced_lemma_remove_assumption_47001480: "S (length (x var_0 var_1)) = length (x var_0 (cons2 var_2 var_1))"
apply (induct "var_0" arbitrary : var_1)
apply (simp_all) done
lemma abduced_lemma_tactic_13498034:
"(⋀a. length (x var_0 a) = length (x a var_0)) ⟹ S (length (x var_1 var_0)) = length (x var_1 (cons2 var_2 var_0))"
apply (simp add : abduced_lemma_remove_assumption_47001480) done
lemma abduced_lemma_composite_commutativity_7642270: "length (x var_0 var_1) = length (x var_1 var_0)"
apply (induct "var_0" arbitrary : var_1)
apply (simp_all)
using abduced_lemma_tactic_13498028 apply blast
apply (simp add : abduced_lemma_tactic_13498034) done
lemma abduced_lemma_tactic_13293202: "length (rev var_0) = length var_0 ⟹ length (x (rev var_0) (cons2 var_1 nil2)) = S (length var_0)"
apply (metis TIP_prop_06.length.simps (2) TIP_prop_06.x.simps (1) TIP_prop_06.x.simps (2)
abduced_lemma_composite_commutativity_7642270) done
lemma abduced_lemma_generalisation_then_extension_7642332: "length (rev var_0) = length var_0"
apply (induct "var_0")
apply (simp_all)
apply (simp add : abduced_lemma_tactic_13293202) done
lemma abduced_lemma_identity_7579838: "x var_0 nil2 = var_0"
apply (induct "var_0")
apply (simp_all) done
lemma abduced_lemma_generalisation_then_extension_17745958: "length (rev (x var_0 nil2)) = length var_0"
apply (simp add : abduced_lemma_generalisation_then_extension_7642332 abduced_lemma_identity_7579838)
done
lemma abduced_lemma_tactic_33224946: "(⋀a. length (rev (x a var_0)) = t2 (length var_0) (length a)) ⟹
length (rev (x var_1 (cons2 var_2 var_0))) = S (t2 (length var_0) (length var_1))"
apply (metis TIP_prop_06.length.simps (2) TIP_prop_06.x.simps (2) abduced_lemma_composite_commutativity_7642270 abduced_lemma_generalisation_then_extension_7642332)
done
lemma abduced_lemma_generalise_by_renaming_7642286: "length (rev (x var_0 var_1)) = t2 (length var_1) (length var_0)"
apply (induct "var_1" arbitrary : var_0)
apply (simp_all)
apply (simp add : abduced_lemma_generalisation_then_extension_17745958)
apply (simp add : abduced_lemma_tactic_33224946)
done
lemma original_goal_7579816: "length (rev (x var_0 var_1)) = t2 (length var_0) (length var_1)"
apply (simp add : abduced_lemma_commutativity_7642214 abduced_lemma_generalise_by_renaming_7642286)
done
theorem
original_goal_7579816:
length (rev (x ?var_0.0 ?var_1.0)) =
t2 (length ?var_0.0) (length ?var_1.0)
(lemma→ goal)
lemma
goal
modus ponens
(conjecture → goal)
conjecture
goal
mutation-based
conjecturing
Query Sledgehammer Symbols
File Browser Documentation
HyperSearch Results Output Sidekick State Theories


File Edit Search Markers Folding View Utilities Macros Plugins Help
TIP_prop_06.thy (~/Workplace/Prod/Prod/)
"t2 (S z2) z = S (t2 z2 z)"
prove property0 :
"((length (rev (x y z))) = (t2 (length y) (length z)))"
lemma abduced_lemma_tactic_14045356: "var_0 = t2 var_0 Z"
apply (induct "var_0")
apply (simp_all) done
lemma abduced_lemma_tactic_14045358: "(⋀a. t2 var_0 a = t2 a var_0) ⟹ S (t2 var_1 var_0) = t2 var_1 (S var_0)"
apply (induct "var_1" "var_0" rule : TIP_prop_06.t2.induct)
apply (simp_all) done
lemma abduced_lemma_commutativity_7642214: "t2 var_0 var_1 = t2 var_1 var_0"
apply (induct "var_0" arbitrary : var_1)
apply (simp_all)
apply (simp add : abduced_lemma_tactic_14045356)
using abduced_lemma_tactic_14045358 apply force done
lemma abduced_lemma_tactic_13498028: "length var_0 = length (x var_0 nil2)"
apply (induct "var_0")
apply (simp_all) done
lemma abduced_lemma_remove_assumption_47001480: "S (length (x var_0 var_1)) = length (x var_0 (cons2 var_2 var_1))"
apply (induct "var_0" arbitrary : var_1)
apply (simp_all) done
lemma abduced_lemma_tactic_13498034:
"(⋀a. length (x var_0 a) = length (x a var_0)) ⟹ S (length (x var_1 var_0)) = length (x var_1 (cons2 var_2 var_0))"
apply (simp add : abduced_lemma_remove_assumption_47001480) done
lemma abduced_lemma_composite_commutativity_7642270: "length (x var_0 var_1) = length (x var_1 var_0)"
apply (induct "var_0" arbitrary : var_1)
apply (simp_all)
using abduced_lemma_tactic_13498028 apply blast
apply (simp add : abduced_lemma_tactic_13498034) done
lemma abduced_lemma_tactic_13293202: "length (rev var_0) = length var_0 ⟹ length (x (rev var_0) (cons2 var_1 nil2)) = S (length var_0)"
apply (metis TIP_prop_06.length.simps (2) TIP_prop_06.x.simps (1) TIP_prop_06.x.simps (2)
abduced_lemma_composite_commutativity_7642270) done
lemma abduced_lemma_generalisation_then_extension_7642332: "length (rev var_0) = length var_0"
apply (induct "var_0")
apply (simp_all)
apply (simp add : abduced_lemma_tactic_13293202) done
lemma abduced_lemma_identity_7579838: "x var_0 nil2 = var_0"
apply (induct "var_0")
apply (simp_all) done
lemma abduced_lemma_generalisation_then_extension_17745958: "length (rev (x var_0 nil2)) = length var_0"
apply (simp add : abduced_lemma_generalisation_then_extension_7642332 abduced_lemma_identity_7579838)
done
lemma abduced_lemma_tactic_33224946: "(⋀a. length (rev (x a var_0)) = t2 (length var_0) (length a)) ⟹
length (rev (x var_1 (cons2 var_2 var_0))) = S (t2 (length var_0) (length var_1))"
apply (metis TIP_prop_06.length.simps (2) TIP_prop_06.x.simps (2) abduced_lemma_composite_commutativity_7642270 abduced_lemma_generalisation_then_extension_7642332)
done
lemma abduced_lemma_generalise_by_renaming_7642286: "length (rev (x var_0 var_1)) = t2 (length var_1) (length var_0)"
apply (induct "var_1" arbitrary : var_0)
apply (simp_all)
apply (simp add : abduced_lemma_generalisation_then_extension_17745958)
apply (simp add : abduced_lemma_tactic_33224946)
done
lemma original_goal_7579816: "length (rev (x var_0 var_1)) = t2 (length var_0) (length var_1)"
apply (simp add : abduced_lemma_commutativity_7642214 abduced_lemma_generalise_by_renaming_7642286)
done
Query Sledgehammer Symbols
Proof state Auto update Update Search:
100%
theorem
original_goal_7579816:
length (rev (x ?var_0.0 ?var_1.0)) =
t2 (length ?var_0.0) (length ?var_1.0)
File Browser Documentation
HyperSearch Results Output Sidekick State Theories
simultaneous
conjecturing
(lemma1 → lmma2 → goal)
lemma1
lemma2
goal