

Abduction Prover for Induction Problems at WAIT2024

Yutaka Nagashima
x/twitter: Yutakange

✓ high-level talk



1



2



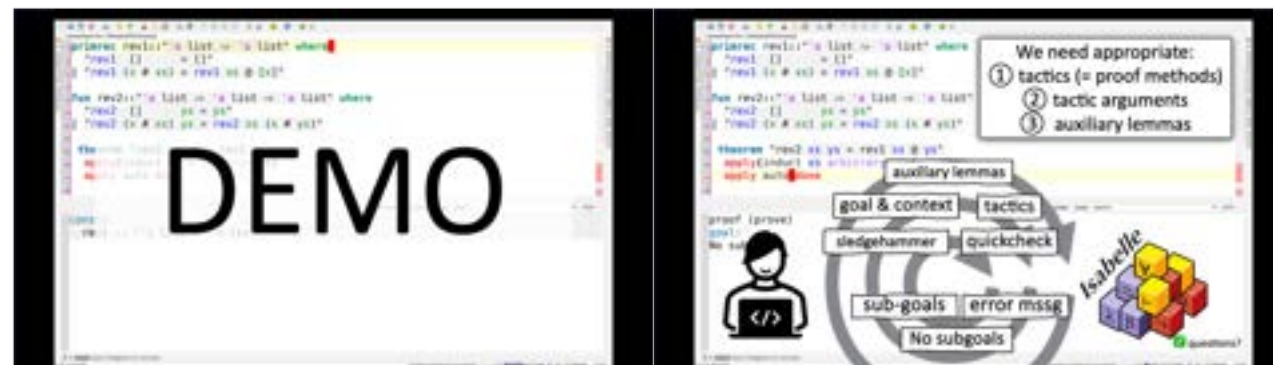
3



4



5



6

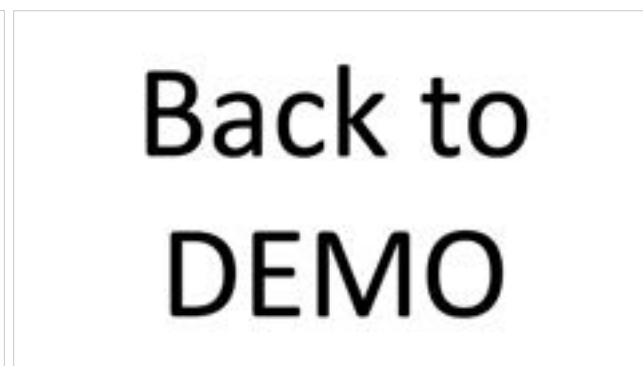
7



8



9



10



11



12

Who am I?



+ AI / PL / (FM)

✓ not obsessed about Isabelle



Who am I?

✓ not obsessed about Isabelle



7th Conference on Artificial Intelligence and Theorem Proving AITP 2022 September 4-9, 2022, Aussois and online, France

The registration is now closed and the list of participants has been fixed. Do not provide any payment data in any other way - it is very likely a scam!

Background

Large-scale semantic processing and strong computer assistance of mathematics and science is our inevitable future. New combinations of AI and reasoning methods and tools deployed over large mathematical and scientific corpora will be instrumental to this task. The AITP conference is the forum for discussing how to get there as soon as possible, and the force driving the progress towards that.

This year AITP will be co-located with a **meeting of WG5** of Cost Action **European Research Network on Formal Proofs**.

Topics

- AI and big-data methods in theorem proving and mathematics
- Collaboration between automated and interactive theorem proving, in particular their AI/ML aspects
- Common-sense reasoning and reasoning in science
- Alignment and joint processing of formal, semi-formal, and informal libraries, Formal Abstracts

3 Secretes of AITP

✓ more of a working hypothesis than a scientific statement



3 things that are not that important.



(Machine)Learning.

compared to representation & search.

🤔 Only one proof is needed for verification.



Tactic Selection.

compared to tactic argument selection.

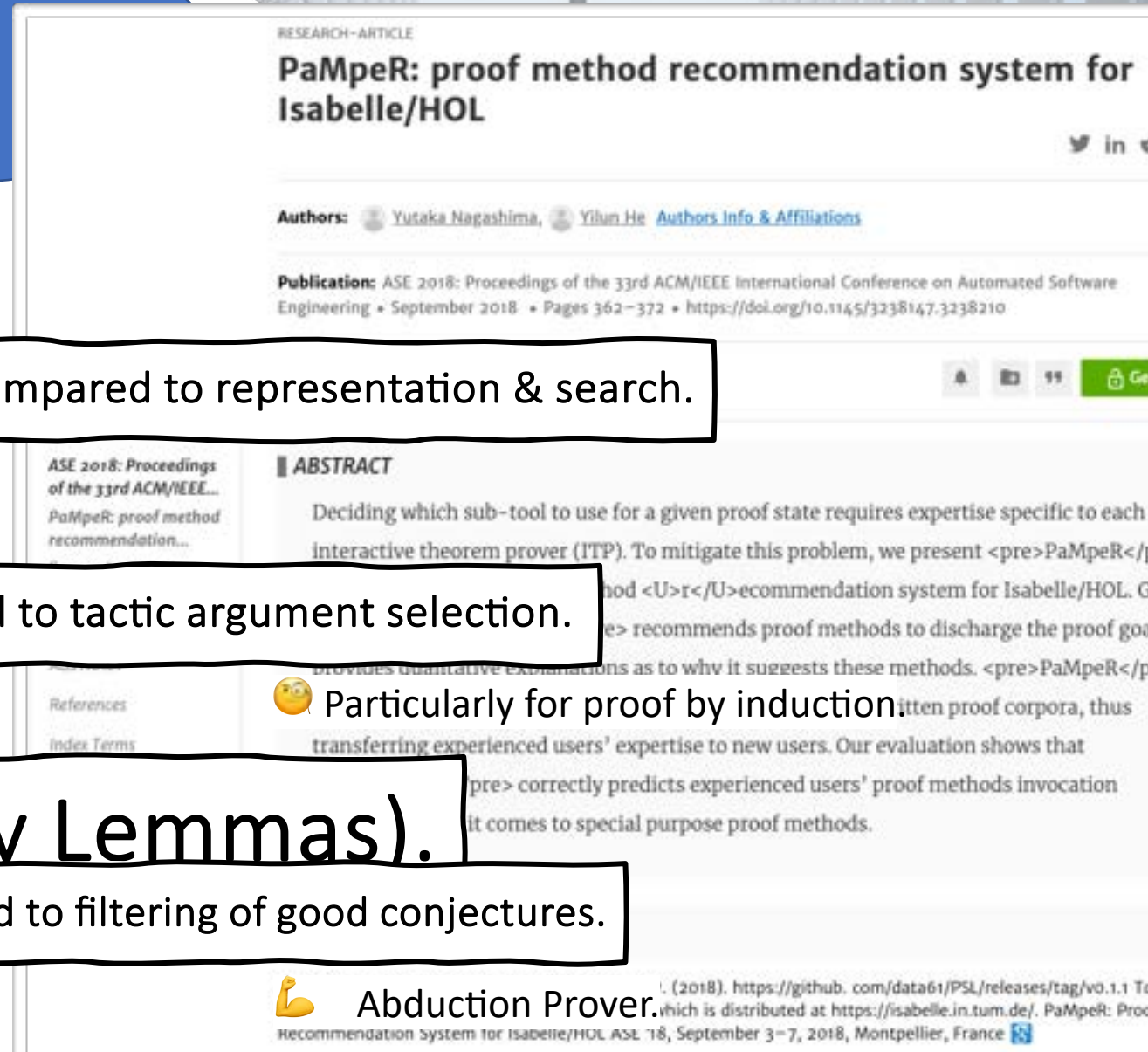
🤔 Useful only for special tactics.
🔨 Just use Sledgehammer.



Conjecturing (Auxiliary Lemmas).

compared to filtering of good conjectures.

🤔 Human creativity?





Abduction Prover.

- Automatic Conjecturing of Auxiliary lemmas
- Automatic Filtering of Conjectures
- Seamless Integration into Isabelle/HOL
- Sledgehammer & Quickcheck & Nitpick Integration
- PSL & Smart Induction Integration
- Native Isabelle Proof Script Generation
- Human-Friendly Proof Script Generation
- (Mostly) Parallel Proof Search
- Best-First Expansion rather than Depth-First Search
- Simultaneous Abduction of Multiple Conjectures
- Multi-Step Abduction of Auxiliary Lemmas
- Template-Based Conjecturing



Q & A

Used OCaml/Haskell
for more than 10 hours?

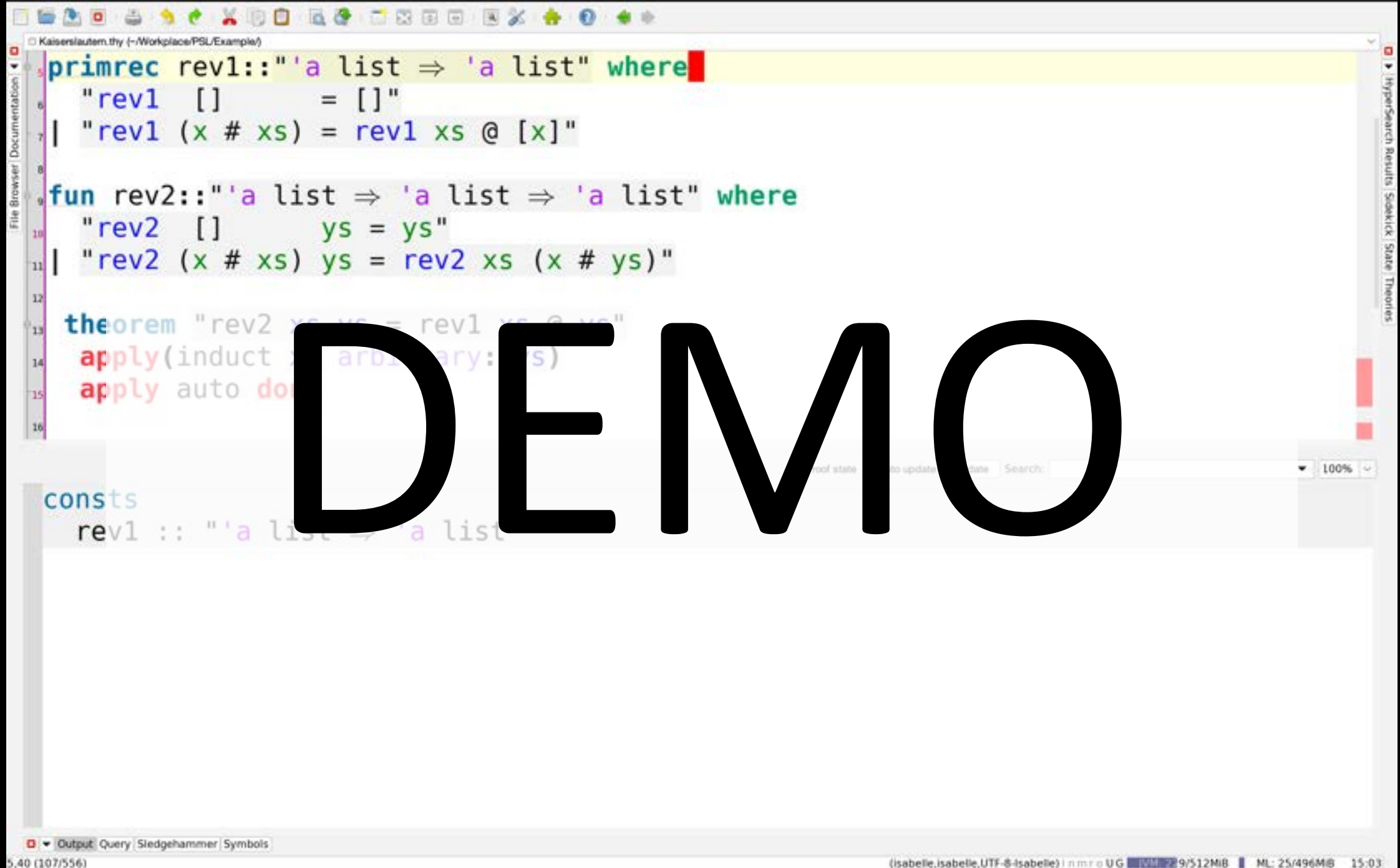
90 % ?

Used an Isabelle/Coq/LEAN
for more than 10 hours?

70 % ?

Invoked Sledgehammer
at least once?

70 % ?



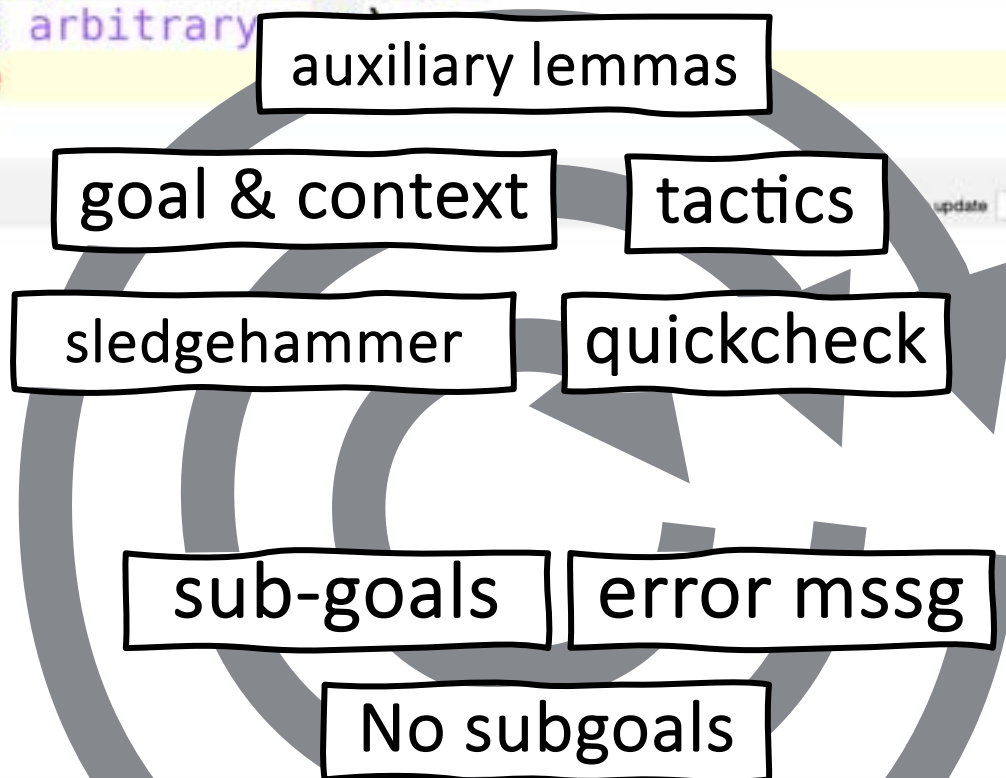
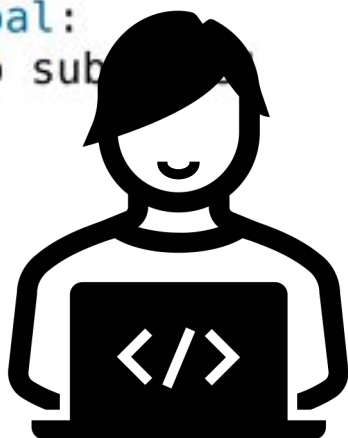

```

Kaiserslautern.thy (~\Workplace\PSL\Example\
5 primrec rev1:: "'a list ⇒ 'a list" where
6   "rev1 [] = []"
7   | "rev1 (x # xs) = rev1 xs @ [x]"
8
9 fun rev2:: "'a list ⇒ 'a list ⇒ 'a list"
10  "rev2 [] ys = ys"
11  | "rev2 (x # xs) ys = rev2 xs (x # ys)"
12
13 theorem "rev2 xs ys = rev1 xs @ ys"
14   apply (induct xs arbitrary)
15   apply auto done
16

```

- We need appropriate:
- ① tactics (= proof methods)
 - ② tactic arguments
 - ③ auxiliary lemmas

proof (prove)
goal:
No sub



✓ questions?

We need appropriate:

- ① tactics (= proof methods)
- ② tactic arguments
- ③ auxiliary lemmas



Conjecturing is not that important if we can ...



Good lemma?

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



We need appropriate:

- ① tactics (= proof methods)
- ② tactic arguments
- ③ auxiliary lemmas



Conjecturing is not that important if we can ...



Fail early.



Good lemma?

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



We need appropriate:

- ① tactics (= proof methods)
- ② tactic arguments
- ③ auxiliary lemmas



Conjecturing is not that important if we can ...



Fail early.

$(P \rightarrow Q)$

P

Modus ponens

Q



Good lemma?

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



We need appropriate:

- ① tactics (= proof methods)
- ② tactic arguments
- ③ auxiliary lemmas



Conjecturing is not that important if we can ...



Fail early.

try to prove

quickcheck

easier

(lemma \rightarrow goal)

lemma

Modus ponens

goal

difficult



Good lemma?

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



We need appropriate:

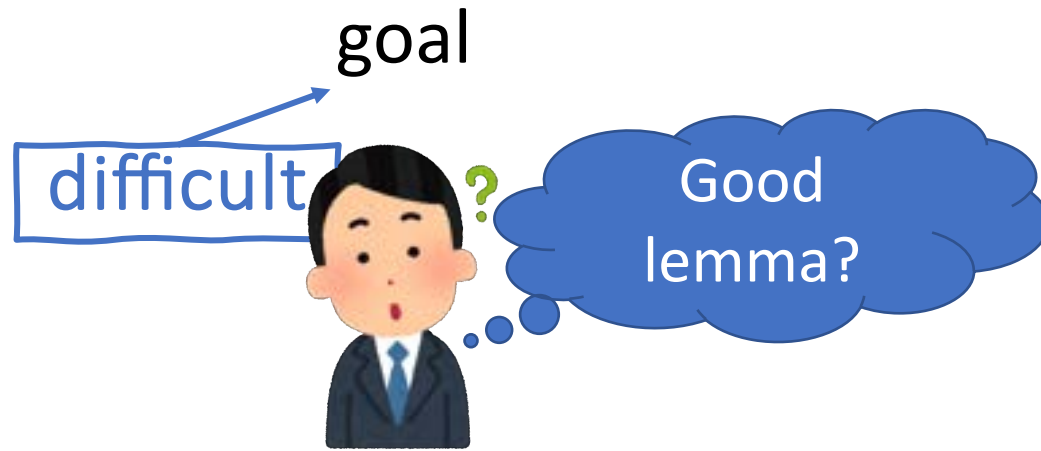
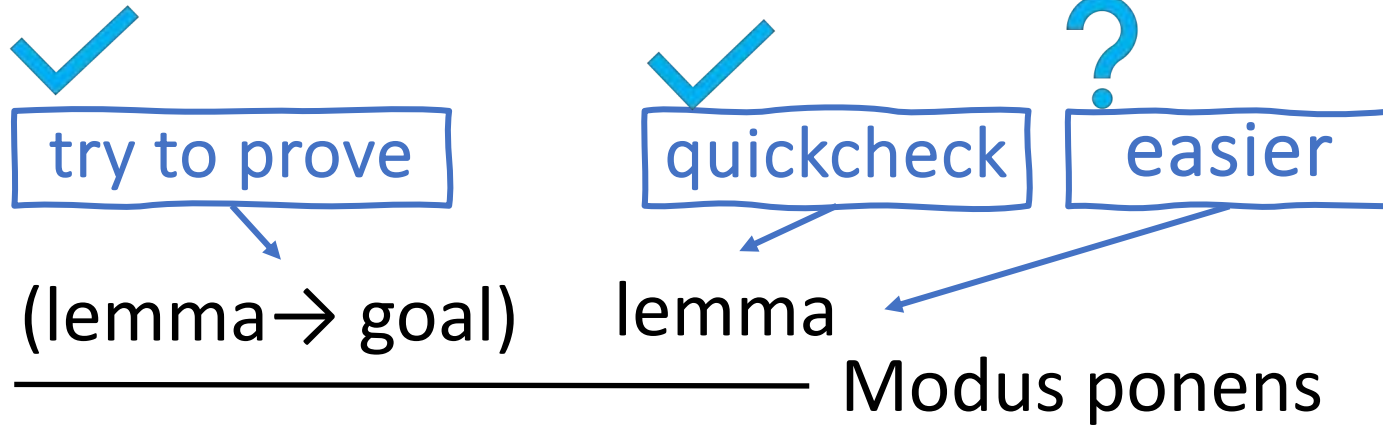
- ① tactics (= proof methods)
- ② tactic arguments
- ③ auxiliary lemmas



Conjecturing is not that important if we can ...



Fail early.



Good lemma?

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



We need appropriate:

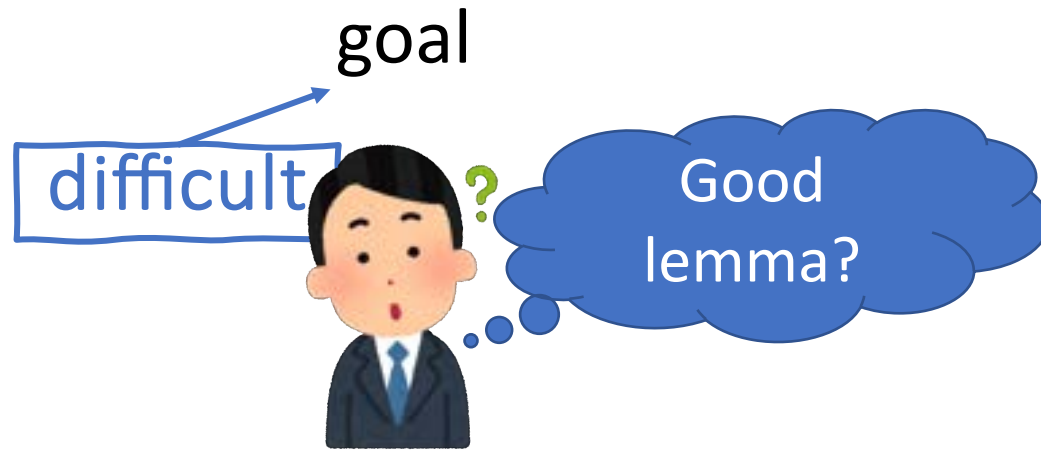
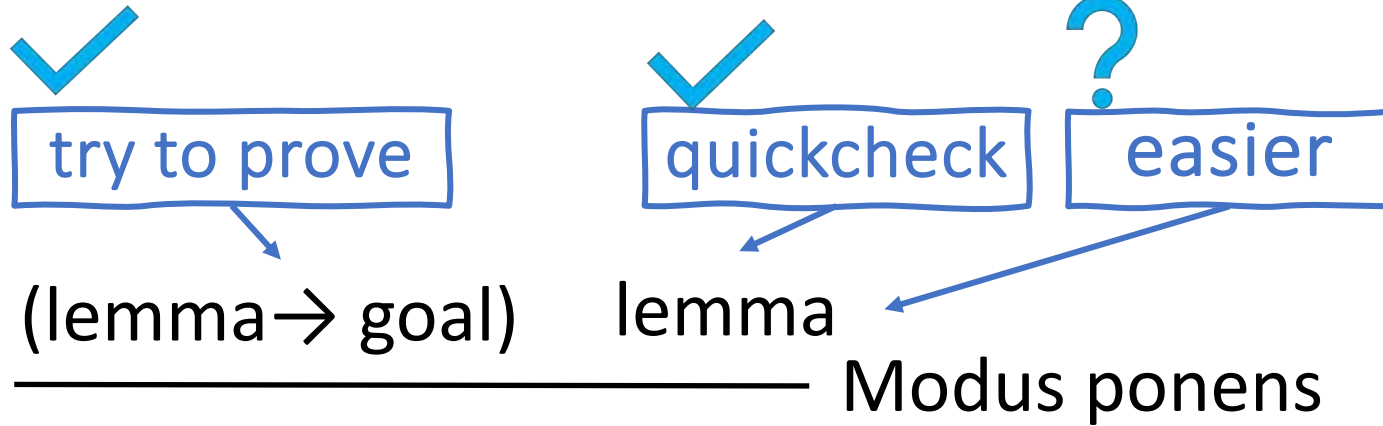
- ① tactics (= proof methods)
- ② tactic arguments
- ③ auxiliary lemmas



Conjecturing is not that important if we can ...



Fail early.




Good lemma?

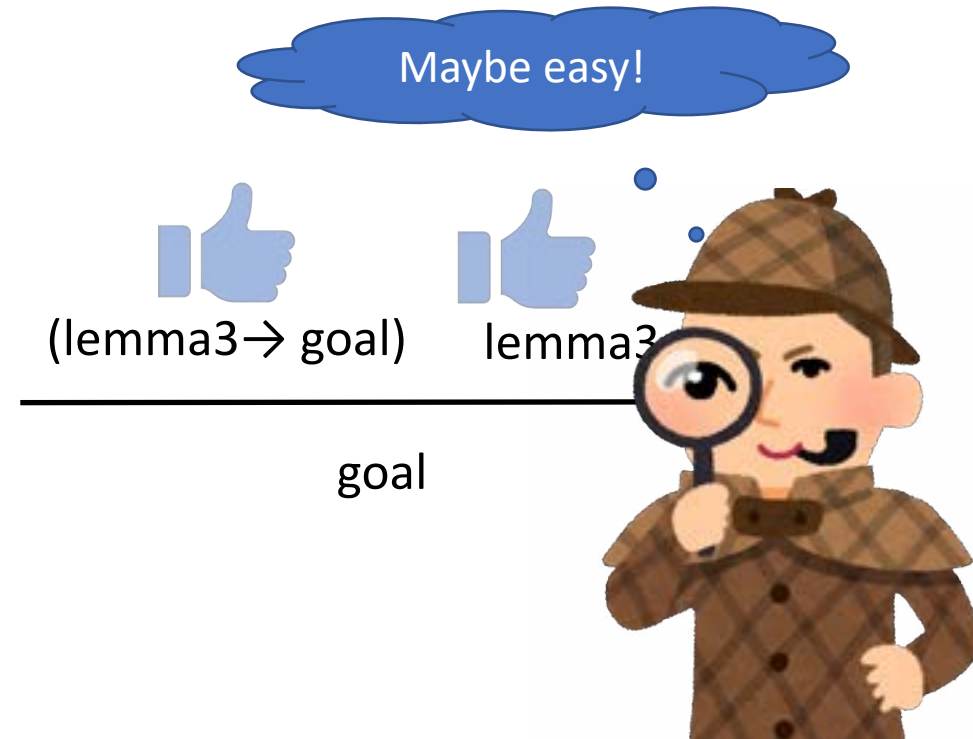
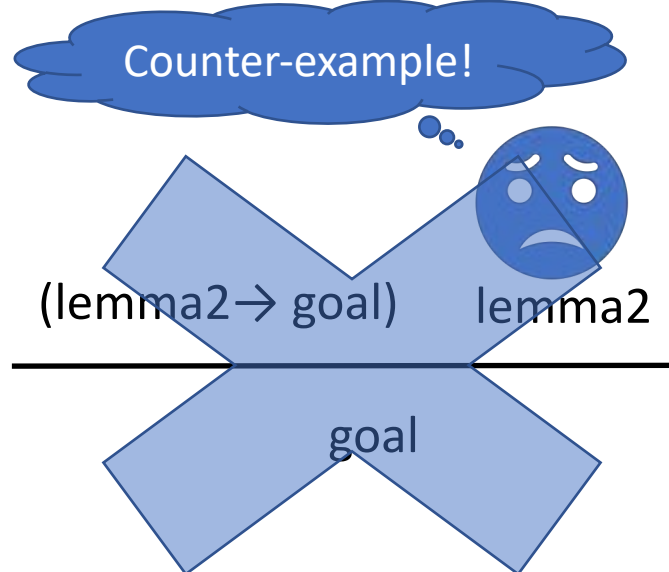
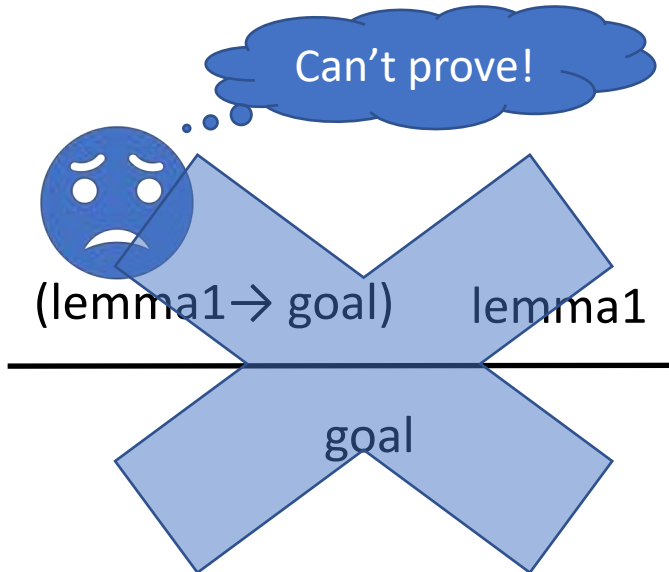
- ✓ 1. Not obviously false.
- ✓ 2. Useful to prove the goal.
- ? 3. Easy to prove.



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

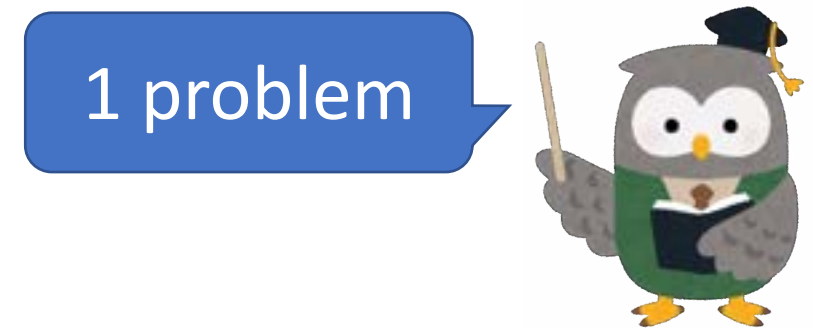


- ✓ 1. Not obviously false.
 - ✓ 2. Useful to prove the goal.
 - ? 3. Easy to prove.
- 
- A cartoon owl wearing a graduation cap and holding a book and a pointer.



We need appropriate:

- ① tactics (= proof methods)
- ② tactic arguments
- ③ auxiliary lemmas





lemma abduction

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

tactic application

$$\frac{(\text{new-goal} \rightarrow \text{goal}) \quad \text{new-goal}}{\text{goal}}$$

proof by induction

$$\frac{(\text{base-case} \ \& \ \text{step-case} \rightarrow \text{goal}) \quad \text{base-case} \quad \text{step-case}}{\text{goal}}$$

Back to

DEMO

UR ■ United Reasoning



```

10 | "t2 (5 z2) z = 5 (t2 z2 z)"
11
12 prove property0 :
13   "((length (rev (x y z))) = (t2 (length y) (length z)))"
14
15 lemma abducted lemma_tactic 14045356: "var_0 = t2 var_0 2"
16   apply [ induct "var_0" ]
17   apply [ simp all ] done
18
19 lemma abducted lemma_tactic 14045358: "(!a, t2 var_0 a = t2 a var_0) ==> 5 (t2 var_1 var_0) = t2 var_1 (5 var_0)"
20   apply [ induct "var_1" "var_0" rule : TIP_prop_06.t2.induct ]
21   apply [ simp all ] done
22
23 lemma abducted lemma_commutativity_7642214: "t2 var_0 var_1 = t2 var_1 var_0"
24   apply [ induct "var_0" arbitrary : var_1 ]
25   apply [ simp all ]
26   apply [ simp all : abducted lemma_tactic 14045356 ]
27   using abducted lemma_tactic 14045358 apply force done
28
29 lemma abducted lemma_tactic 13498028: "length var_0 = length (x var_0 nil2)"
30   apply [ induct "var_0" ]
31   apply [ simp all ] done
32
33 lemma abducted lemma_remove_assumption 47001400: "5 (length (x var_0 var_1)) = length (x var_0 (cons2 var_2 var_1))"
34   apply [ induct "var_0" arbitrary : var_1 ]
35   apply [ simp all ] done
36
37 lemma abducted lemma_tactic 13498034:
38   "(!a, length (x var_0 a) = length (x a var_0)) ==> 5 (length (x var_1 var_0)) = length (x var_1 (cons2 var_2 var_0))"
39   apply [ simp all : abducted lemma_remove_assumption 47001400 ] done
40
41 lemma abducted lemma_composite_commutativity_7642270: "length (x var_0 var_1) = length (x var_1 var_0)"
42   apply [ induct "var_0" arbitrary : var_1 ]
43   apply [ simp all ]
44   using abducted lemma_tactic 13498028 apply blast
45   apply [ simp all : abducted lemma_tactic 13498034 ] done
46
47 lemma abducted lemma_tactic 13293202: "length (rev var_0) = length var_0 ==> length (x (rev var_0) (cons2 var_1 nil2)) = 5 (length var_0)"
48   apply [ metis TIP_prop_06.length.simps { 2 } TIP_prop_06.x.simps { 1 } TIP_prop_06.x.simps { 2 }
49         abducted lemma_composite_commutativity_7642270 ] done
50
51 lemma abducted lemma_generalisation_then_extension_7642332: "length (rev var_0) = length var_0"
52   apply [ induct "var_0" ]
53   apply [ simp all ]
54   apply [ simp all : abducted lemma_tactic 13293202 ] done
55
56 lemma abducted lemma_identity_7579838: "x var_0 nil2 = var_0"
57   apply [ induct "var_0" ]
58   apply [ simp all ] done
59
60 lemma abducted lemma_generalisation_then_extension_17745958: "length (rev (x var_0 nil2)) = length var_0"
61   apply [ simp all : abducted lemma_generalisation_then_extension_7642332 abducted lemma_identity_7579838 ]
62   done
63
64 lemma abducted lemma_tactic 33224946: "(!a, length (rev (x a var_0)) = t2 (length var_0) (length a)) ==>
65   length (rev (x var_1 (cons2 var_2 var_0))) = 5 (t2 (length var_0) (length var_1))"
66   apply [ metis TIP_prop_06.length.simps { 2 } TIP_prop_06.x.simps { 2 } abducted lemma_composite_commutativity_7642270
67         abducted lemma_generalisation_then_extension_7642332 ]
68   done
69
70 lemma abducted lemma_generalise_by_renaming_7642286: "length (rev (x var_0 var_1)) = t2 (length var_1) (length var_0)"
71   apply [ induct "var_1" arbitrary : var_0 ]
72   apply [ simp all ]
73   apply [ simp all : abducted lemma_generalisation_then_extension_17745958 ]
74   apply [ simp all : abducted lemma_tactic 33224946 ]
75   done
76
77 lemma original_goal_7579836: "length (rev (x var_0 var_1)) = t2 (length var_0) (length var_1)"
78   apply [ simp all : abducted lemma_commutativity_7642214 abducted lemma_generalise_by_renaming_7642286 ]
79   done

```

theorem
original goal 7579836:
length (rev (x ?var_0 0 ?var_1 0)) =
t2 (length ?var_0 0) (length ?var_1 0)

Q & A

We need appropriate:

- ① tactics (= proof methods)
- ② tactic arguments
- ③ auxiliary lemmas

These 3 things are
not that important.



(Machine)Learning.

compared to representation & search.



Tactic Selection.

compared to tactic argument selection.



Conjecturing (Auxiliary Lemmas).

compared to filtering of good conjectures.

Automatic Conjecturing of Auxiliary lemmas

Automatic Filtering of Conjectures

Seamless Integration into Isabelle/HOL

Sledgehammer & Quickcheck & Nitpick Integration

PSL & Smart Induction Integration

Native Isabelle Proof Script Generation

Human-Friendly Proof Script Generation

(Mostly) Parallel Proof Search

Best-First Expansion rather than Depth-First Search

Simultaneous Abduction of Multiple Conjectures

Multi-Step Abduction of Auxiliary Lemmas

Template-Based Conjecturing


```
The GUI Search Markers Folding View Utilities Macros Theories help
[Icons]
File Browser Documentation
0 TIP_prop_06.thy (WindowsProdProd)
30 | "t2 (S z2) z = S (t2 z2 z)"
31
32
33 prove property0 :
34   "((length (rev (x y z))) = (t2 (length y) (length z)))"
35
36
37 lemma abducted lemma_tactic_14045356: "var_0 = t2 var_0 2"
38   apply (induct "var_0" )
39   apply (simp all ) done
40
41 lemma abducted lemma_tactic_14045358: "(!s. t2 var_0 s = t2 s var_0) ==> S (t2 var_1 var_0) = t2 var_1 (S var_0)"
42   apply (induct "var_1" "var_0" rule : TIP_prop_06.t2.induct )
43   apply (simp all ) done
44
45 lemma abducted lemma_commutativity_7642214: "t2 var_0 var_1 = t2 var_1 var_0"
46   apply (induct "var_0" arbitrary : var_1 )
47   apply (simp all )
48   apply (simp add : abducted lemma_tactic_14045356 )
49   using abducted lemma_tactic_14045358 apply force done
50
51 lemma abducted lemma_tactic_13498028: "length var_0 = length (x var_0 nil2)"
52   apply (induct "var_0" )
53   apply (simp all ) done
54
55 lemma abducted lemma_remove_assumption_47001400: "S (length (x var_0 var_1)) = length (x var_0 (cons2 var_2 var_1))"
56   apply (induct "var_0" arbitrary : var_1 )
57   apply (simp all ) done
58
59 lemma abducted lemma_tactic_13498034:
60   "(!s. length (x var_0 s) = length (x s var_0)) ==> S (length (x var_1 var_0)) = length (x var_1 (cons2 var_2 var_0))"
61   apply (simp add : abducted lemma_remove_assumption_47001400 ) done
62
63 lemma abducted lemma_composite_commutativity_7642270: "length (x var_0 var_1) = length (x var_1 var_0)"
64   apply (induct "var_0" arbitrary : var_1 )
65   apply (simp all )
66   using abducted lemma_tactic_13498028 apply blast
67   apply (simp add : abducted lemma_tactic_13498034 ) done
68
69 lemma abducted lemma_tactic_13293202: "length (rev var_0) = length var_0 ==> length (x (rev var_0) (cons2 var_1 nil2)) = S (length var_0)"
70   apply (metis TIP_prop_06.length.simps [ 2 ] TIP_prop_06.x.simps [ 1 ] TIP_prop_06.x.simps [ 2 ]
71     abducted lemma_composite_commutativity_7642270 ) done
72
73 lemma abducted lemma_generalisation_then_extension_7642332: "length (rev var_0) = length var_0"
74   apply (induct "var_0" )
75   apply (simp all )
76   apply (simp add : abducted lemma_tactic_13293202 ) done
77
78 lemma abducted lemma_identity_7579838: "x var_0 nil2 = var_0"
79   apply (induct "var_0" )
80   apply (simp all ) done
81
82 lemma abducted lemma_generalisation_then_extension_17745958: "length (rev (x var_0 nil2)) = length var_0"
83   apply (simp add : abducted lemma_generalisation_then_extension_7642332 abducted lemma_identity_7579838 )
84   done
85
86 lemma abducted lemma_tactic_33224946: "(!s. length (rev (x s var_0)) = t2 (length var_0) (length s)) ==>
87   length (rev (x var_1 (cons2 var_2 var_0))) = S (t2 (length var_0) (length var_1))"
88   apply (metis TIP_prop_06.length.simps [ 2 ] TIP_prop_06.x.simps [ 2 ] abducted lemma_composite_commutativity_7642270
89     abducted lemma_generalisation_then_extension_7642332 )
90   done
91
92 lemma abducted lemma_generalise_by_renaming_7642286: "length (rev (x var_0 var_1)) = t2 (length var_1) (length var_0)"
93   apply (induct "var_1" arbitrary : var_0 )
94   apply (simp all )
95   apply (simp add : abducted lemma_generalisation_then_extension_17745958 )
96   apply (simp add : abducted lemma_tactic_33224946 )
97   done
98
99 lemma original_goal_7579816: "length (rev (x var_0 var_1)) = t2 (length var_0) (length var_1)"
100   apply (simp add : abducted lemma_commutativity_7642214 abducted lemma_generalise_by_renaming_7642286 )
101   done
```

(lemma \rightarrow goal) lemma

goal

(sub-goal \rightarrow goal) sub-goal

goal

Modus ponens

tactic

```
theorem
  original goal_7579816:
    length (rev (x tvar_0 0 tvar_1 0)) =
      t2 (length tvar_0 0) (length tvar_1 0)
```

```

30 | "t2 (S z2) z = S (t2 z2 z)"
31
32
33 prove property0 :
34   "((length (rev (x y z))) = (t2 (length y) (length z)))"
35
36
37 lemma abduced_lemma_tactic_14045356: "var_0 = t2 var_0 2"
38   apply (induct "var_0" )
39   apply (simp_all ) done
40
41 lemma abduced_lemma_tactic_14045358: "(!x. t2 var_0 x = t2 x var_0) ==> S (t2 var_1 var_0) = t2 var_1 (S var_0)"
42   apply (induct "var_1" "var_0" rule : TIP_prop_06.t2.induct )
43   apply (simp_all ) done
44
45 lemma abduced_lemma_commutativity_7642214: "t2 var_0 var_1 = t2 var_1 var_0"
46   apply (induct "var_0" arbitrary : var_1 )
47   apply (simp_all )
48   apply (simp add : abduced_lemma_tactic_14045356 )
49   using abduced_lemma_tactic_14045358 apply force done
50
51 lemma abduced_lemma_tactic_13498028: "length var_0 = length (x var_0 nil2)"
52   apply (induct "var_0" )
53   apply (simp_all ) done
54
55 lemma abduced_lemma_remove_assumption_47001400: "S (length (x var_0 var_1)) = length (x var_0 (cons2 var_2 var_1))"
56   apply (induct "var_0" arbitrary : var_1 )
57   apply (simp_all ) done
58
59 lemma abduced_lemma_tactic_13498034:
60   "(!x. length (x var_0 x) = length (x x var_0)) ==> S (length (x var_1 var_0)) = length (x var_1 (cons2 var_2 var_0))"
61   apply (simp add : abduced_lemma_remove_assumption_47001400 ) done
62
63 lemma abduced_lemma_composite_commutativity_7642270: "length (x var_0 var_1) = length (x var_1 var_0)"
64   apply (induct "var_0" arbitrary : var_1 )
65   apply (simp_all )
66   using abduced_lemma_tactic_13498028 apply blast
67   apply (simp add : abduced_lemma_tactic_13498034 ) done
68
69 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)"
70   apply (metis TIP_prop_06.length.simps [ 2 ] TIP_prop_06.x.simps [ 1 ] TIP_prop_06.x.simps [ 2 ]
71     abduced_lemma_composite_commutativity_7642270 ) done
72
73 lemma abduced_lemma_generalisation_then_extension_7642332: "length (rev var_0) = length var_0"
74   apply (induct "var_0" )
75   apply (simp_all )
76   apply (simp add : abduced_lemma_tactic_13293202 ) done
77
78 lemma abduced_lemma_identity_7579816: "x var_0 nil2 = var_0"
79   apply (induct "var_0" )
80   apply (simp_all ) done
81
82 lemma abduced_lemma_generalisation_then_extension_17745958: "length (rev (x var_0 nil2)) = length var_0"
83   apply (simp add : abduced_lemma_generalisation_then_extension_7642332 abduced_lemma_identity_7579816 )
84   done
85
86 lemma abduced_lemma_tactic_33224946: "(!x. length (rev (x x var_0)) = t2 (length var_0) (length x)) ==>
87   length (rev (x var_1 (cons2 var_2 var_0))) = S (t2 (length var_0) (length var_1))"
88   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 )
89   done
90
91 lemma abduced_lemma_generalise_by_renaming_7642286: "length (rev (x var_0 var_1)) = t2 (length var_1) (length var_0)"
92   apply (induct "var_1" arbitrary : var_0 )
93   apply (simp_all )
94   apply (simp add : abduced_lemma_generalisation_then_extension_17745958 )
95   apply (simp add : abduced_lemma_tactic_33224946 )
96   done
97
98 lemma original_goal_7579816: "length (rev (x var_0 var_1)) = t2 (length var_0) (length var_1)"
99   apply (simp add : abduced_lemma_commutativity_7642214 abduced_lemma_generalise_by_renaming_7642286 )
100  done

```

theorem

original goal_7579816:

```

length (rev (x tvar_0 0 tvar_1 0)) =
t2 (length tvar_0 0) (length tvar_1 0)

```

```

10 | "t2 (S z2) z = S (t2 z2 z)"
11
12 prove property0 :
13   "((length (rev (x y z))) = (t2 (length y) (length z)))"
14
15
16 lemma abducted lemma_tactic 14045356: "var_0 = t2 var_0 2"
17   apply [ induct "var_0" ]
18   apply [ simp all ] done
19
20 lemma abducted lemma_tactic 14045358: "(!x. t2 var_0 x = t2 x var_0) ==> S (t2 var_1 var_0) = t2 var_1 (S var_0)"
21   apply [ induct "var_1" "var_0" rule : TIP_prop_06.t2.induct ]
22   apply [ simp all ] done
23
24 lemma abducted lemma_commutativity_7642214: "t2 var_0 var_1 = t2 var_1 var_0"
25   apply [ induct "var_0" arbitrary : var_1 ]
26   apply [ simp all ]
27   apply [ simp all : abducted lemma_tactic 14045356 ]
28   using abducted lemma_tactic 14045358 apply force done
29
30 lemma abducted lemma_tactic 13498028: "length var_0 = length (x var_0 nil2)"
31   apply [ induct "var_0" ]
32   apply [ simp all ] done
33
34 lemma abducted lemma_remove_assumption 47001400: "S (length (x var_0 var_1)) = length (x var_0 (cons2 var_2 var_1))"
35   apply [ induct "var_0" arbitrary : var_1 ]
36   apply [ simp all ] done
37
38 lemma abducted lemma_tactic 13498034:
39   "(!x. length (x var_0 x) = length (x var_0)) ==> S (length (x var_1 var_0)) = length (x var_1 (cons2 var_2 var_0))"
40   apply [ simp all : abducted lemma_remove_assumption 47001400 ] done
41
42 lemma abducted lemma_composite_commutativity_7642270: "length (x var_0 var_1) = length (x var_1 var_0)"
43   apply [ induct "var_0" arbitrary : var_1 ]
44   apply [ simp all ]
45   using abducted lemma_tactic 13498028 apply blast
46   apply [ simp all : abducted lemma_tactic 13498034 ] done
47
48 lemma abducted lemma_tactic 13293202: "length (rev var_0) = length var_0 ==> length (x (rev var_0) (cons2 var_1 nil2)) = S (length var_0)"
49   apply [ metis TIP_prop_06.length.simps { 2 } TIP_prop_06.x.simps { 1 } TIP_prop_06.x.simps { 2 }
50         abducted lemma_composite_commutativity_7642270 ] done
51
52 lemma abducted lemma_generalisation_then_extension_7642332: "length (rev var_0) = length var_0"
53   apply [ induct "var_0" ]
54   apply [ simp all ]
55   apply [ simp all : abducted lemma_tactic 13293202 ] done
56
57 lemma abducted lemma_identity_7579838: "x var_0 nil2 = var_0"
58   apply [ induct "var_0" ]
59   apply [ simp all ] done
60
61 lemma abducted lemma_generalisation_then_extension_17745958: "length (rev (x var_0 nil2)) = length var_0"
62   apply [ simp all : abducted lemma_generalisation_then_extension_7642332 abducted lemma_identity_7579838 ]
63   done
64
65 lemma abducted lemma_tactic 33224946: "(!x. length (rev (x x var_0)) = t2 (length var_0) (length x)) ==>
66   length (rev (x var_1 (cons2 var_2 var_0))) = S (t2 (length var_0) (length var_1))"
67   apply [ metis TIP_prop_06.length.simps { 2 } TIP_prop_06.x.simps { 2 } abducted lemma_composite_commutativity_7642270
68         abducted lemma_generalisation_then_extension_7642332 ]
69   done
70
71 lemma abducted lemma_generalise_by_renaming_7642286: "length (rev (x var_0 var_1)) = t2 (length var_1) (length var_0)"
72   apply [ induct "var_1" arbitrary : var_0 ]
73   apply [ simp all ]
74   apply [ simp all : abducted lemma_generalisation_then_extension_17745958 ]
75   apply [ simp all : abducted lemma_tactic 33224946 ]
76   done
77
78 lemma original_goal_7579836: "length (rev (x var_0 var_1)) = t2 (length var_0) (length var_1)"
79   apply [ simp all : abducted lemma_commutativity_7642214 abducted lemma_generalise_by_renaming_7642286 ]
80   done

```

theorem

original goal_7579836:

```

length (rev (x tvar_0 0 tvar_1 0)) =
t2 (length tvar_0 0) (length tvar_1 0)

```


goal

```
File Edit Search Windows Fileings View Utilities Macros (F9) Help
TIP_prop_06.thy (WorkplaceProd.thy)
10 | "t2 (S z) z = S (t2 z z)"
11
12
13 prove property0 :
14   "((length (rev (x y z))) = (t2 (length y) (length z)))"
15
16
17 lemma abducted lemma_tactic 14045356: "var_0 = t2 var_0 2"
18   apply [ induct "var_0" ]
19   apply [ simp all ] done
20
21 lemma abducted lemma_tactic 14045358: "(!a. t2 var_0 a = t2 a var_0) ==> S (t2 var_1 var_0) = t2 var_1 (S var_0)"
22   apply [ induct "var_1" "var_0" rule: TIP_prop_06.t2.induct ]
23   apply [ simp all ] done
24
25 lemma abducted lemma_commutativity_7642214: "t2 var_0 var_1 = t2 var_1 var_0"
26   apply [ induct "var_0" arbitrary: var_1 ]
27   apply [ simp all ]
28   apply [ simp add: abducted lemma_tactic 14045356 ]
29   using abducted lemma_tactic 14045358 apply force done
30
31 lemma abducted lemma_tactic 13498028: "length var_0 = length (x var_0 nil2)"
32   apply [ induct "var_0" ]
33   apply [ simp all ] done
34
35 lemma abducted lemma_remove_assumption 47001400: "S (length (x var_0 var_1)) = length (x var_0 (cons2 var_2 var_1))"
36   apply [ induct "var_0" arbitrary: var_1 ]
37   apply [ simp all ] done
38
39 lemma abducted lemma_tactic 13498034:
40   "(!x. 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))"
41   apply [ simp add: abducted lemma_remove_assumption 47001400 ] done
42
43 lemma abducted lemma_composite_commutativity_7642270: "length (x var_0 var_1) = length (x var_1 var_0)"
44   apply [ induct "var_0" arbitrary: var_1 ]
45   apply [ simp all ]
46   using abducted lemma_tactic 13498028 apply blast
47   apply [ simp add: abducted lemma_tactic 13498034 ] done
48
49 lemma abducted lemma_tactic 13293202: "length (rev var_0) = length var_0 ==> length (x (rev var_0) (cons2 var_1 nil2)) = S (length var_0)"
50   apply [ metis TIP_prop_06.length.simps [ 2 ] TIP_prop_06.x.simps [ 1 ] TIP_prop_06.x.simps [ 2 ]
51         abducted lemma_composite_commutativity_7642270 ] done
52
53 lemma abducted lemma_generalisation_then_extension 7642332: "length (rev var_0) = length var_0"
54   apply [ induct "var_0" ]
55   apply [ simp all ]
56   apply [ simp add: abducted lemma_tactic 13293202 ] done
57
58 lemma abducted lemma_identity 7579838: "x var_0 nil2 = var_0"
59   apply [ induct "var_0" ]
60   apply [ simp all ] done
61
62 lemma abducted lemma_generalisation_then_extension 17745958: "length (rev (x var_0 nil2)) = length var_0"
63   apply [ simp add: abducted lemma_generalisation_then_extension 7642332 abducted lemma_identity 7579838 ]
64   done
65
66 lemma abducted lemma_tactic 33224940: "(!a. length (rev (x a var_0)) = t2 (length var_0) (length a)) ==>
67   length (rev (x var_1 (cons2 var_2 var_0))) = S (t2 (length var_0) (length var_1))"
68   apply [ metis TIP_prop_06.length.simps [ 2 ] TIP_prop_06.x.simps [ 2 ] abducted lemma_composite_commutativity_7642270
69         abducted lemma_generalisation_then_extension 7642332 ]
70   done
71
72 lemma abducted lemma_generalise_by_renaming 7642200: "length (rev (x var_0 var_1)) = t2 (length var_1) (length var_0)"
73   apply [ induct "var_1" arbitrary: var_0 ]
74   apply [ simp all ]
75   apply [ simp add: abducted lemma_generalisation_then_extension 17745958 ]
76   apply [ simp add: abducted lemma_tactic 33224940 ]
77   done
78
79 lemma original goal 7579836: "length (rev (x var_0 var_1)) = t2 (length var_0) (length var_1)"
80   apply [ simp add:
81         abducted lemma_commutativity_7642214
82         abducted lemma_generalise_by_renaming 7642200 ]
83   done
```

(lemma → goal) lemma
Modus ponens
goal

```
Proof state    Auto update    Update    Search
theorem
original goal 7579836:
length (rev (x tvar_0.0 tvar_1.0)) =
t2 (length tvar_0.0) (length tvar_1.0)
```

```
The Editor Search Markers Folding View Utilities Macros (F8) Help
File Edit Search Markers Folding View Utilities Macros (F8) Help
TIP_prop_06.thy (WorkplaceProd.thy)
10 | "t2 (S z1) z = S (t2 z2 z)"
11
12
13 prove property0 :
14   "((length (rev (x y z))) = (t2 (length y) (length z)))"
15
16
17 lemma abduced_lemma_tactic_14045356 "var_0 = t2 var_0 2"
18   apply (induct "var_0")
19   apply (simp_all) done
20
21 lemma abduced_lemma_tactic_14045358 "{(x a. t2 var_0 a = t2 a var_0) ==> S (t2 var_1 var_0) = t2 var_1 (S var_0)}"
22   apply (induct "var_1" "var_0" rule: TIP_prop_06.t2.induct)
23   apply (simp_all) done
24
25 lemma abduced_lemma_commutativity_7642214 "t2 var_0 var_1 = t2 var_1 var_0"
26   apply (induct "var_1" arbitrary: var_0)
27   apply (simp_all)
28   apply (simp add: abduced_lemma_tactic_14045356)
29   using abduced_lemma_tactic_14045358 apply force done
30
31 lemma abduced_lemma_tactic_13498028 "length var_0 = length (x var_0 nil2)"
32   apply (induct "var_0")
33   apply (simp_all) done
34
35 lemma abduced_lemma_remove_assumption_47001400 "S (length (x var_0 var_1)) = length (x var_0 (cons2 var_2 var_1))"
36   apply (induct "var_0" arbitrary: var_1)
37   apply (simp_all) done
38
39
40 lemma abduced_lemma_tactic_13498034:
41   "{(x. 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))}"
42   apply (simp add: abduced_lemma_remove_assumption_47001400) done
43
44 lemma abduced_lemma_composite_commutativity_7642270 "length (x var_0 var_1) = length (x var_1 var_0)"
45   apply (induct "var_0" arbitrary: var_1)
46   apply (simp_all)
47   using abduced_lemma_tactic_13498028 apply blast
48   apply (simp add: abduced_lemma_tactic_13498034) done
49
50 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)"
51   apply (metis TIP_prop_06.length.simps [2] TIP_prop_06.x.simps [1] TIP_prop_06.x.simps [2]
52     abduced_lemma_composite_commutativity_7642270) done
53
54 lemma abduced_lemma_generalisation_then_extension_7642332 "length (rev var_0) = length var_0"
55   apply (induct "var_0")
56   apply (simp_all)
57   apply (simp add: abduced_lemma_tactic_13293202) done
58
59 lemma abduced_lemma_identity_7579838 "x var_0 nil2 = var_0"
60   apply (induct "var_0")
61   apply (simp_all) done
62
63 lemma abduced_lemma_generalisation_then_extension_17745958 "length (rev (x var_0 nil2)) = length var_0"
64   apply (simp add: abduced_lemma_generalisation_then_extension_7642332 abduced_lemma_identity_7579838)
65   done
66
67 lemma abduced_lemma_tactic_33224946 "{(x a. length (rev (x a var_0)) = t2 (length var_0) (length a)) ==>
68   length (rev (x var_1 (cons2 var_2 var_0))) = S (t2 (length var_0) (length var_1))"
69   apply (metis TIP_prop_06.length.simps [2] TIP_prop_06.x.simps [2] abduced_lemma_composite_commutativity_7642270
70     abduced_lemma_generalisation_then_extension_7642332)
71   done
72
73 lemma abduced_lemma_generalise_by_renaming_7642286 "length (rev (x var_0 var_1)) = t2 (length var_1) (length var_0)"
74   apply (induct "var_1" arbitrary: var_0)
75   apply (simp_all)
76   apply (simp add: abduced_lemma_generalisation_then_extension_17745958)
77   apply (simp add: abduced_lemma_tactic_33224946)
78   done
79
80 lemma original_goal_7579816 "length (rev (x var_0 var_1)) = t2 (length var_0) (length var_1)"
81   apply (simp add: abduced_lemma_commutativity_7642214 abduced_lemma_generalise_by_renaming_7642286)
82   done
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100 done
```

(lemma \rightarrow goal) lemma
Modus ponens
goal

```
Proof state Auto update Update Search
100%
theorem
original goal 7579816:
length (rev (x tvar_0.0 tvar_1.0)) =
t2 (length tvar_0.0) (length tvar_1.0)
```



```
The Editor Search Markers Folding View Utilities Macros (F8) Help
TIP_prop_06.thy (WorkplaceProd.thy)
11 | "t2 (S z) z = S (t2 z z)"
12
13 prove property0 :
14   "((length (rev (x y z))) = (t2 (length y) (length z)))"
15
16 lemma abducted lemma_tactic_14045356: "var_0 = t2 var_0 2"
17   apply (induct "var_0" )
18   apply (simp all ) done
19
20 lemma abducted lemma_tactic_14045358: "(!a. t2 var_0 a = t2 a var_0) ==> S (t2 var_1 var_0) = t2 var_1 (S var_0)"
21   apply (induct "var_1" "var_0" rule : TIP_prop_06.t2.induct )
22   apply (simp all ) done
23
24 lemma abducted lemma_commutativity_7642214: "t2 var_0 var_1 = t2 var_1 var_0"
25   apply (induct "var_0" arbitrary : var_1 )
26   apply (simp all )
27   apply (simp add : abducted lemma_tactic_14045356 )
28   using abducted lemma_tactic_14045358 apply force done
29
30 lemma abducted lemma_tactic_13498028: "length var_0 = length (x var_0 nil2)"
31   apply (induct "var_0" )
32   apply (simp all ) done
33
34 lemma abducted lemma_remove_assumption_47001400: "S (length (x var_0 var_1)) = length (x var_0 (cons2 var_2 var_1))"
35   apply (induct "var_0" arbitrary : var_1 )
36   apply (simp all ) done
37
38 lemma abducted lemma_tactic_13498034:
39   "(!x. 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))"
40   apply (simp add : abducted lemma_remove_assumption_47001400 ) done
41
42 lemma abducted lemma_composite_commutativity_7642270: "length (x var_0 var_1) = length (x var_1 var_0)"
43   apply (induct "var_0" arbitrary : var_1 )
44   apply (simp all )
45   using abducted lemma_tactic_13498034 apply blast
46   apply (simp add : abducted lemma_tactic_13498034 ) done
47
48 lemma abducted lemma_tactic_13293202: "length (rev var_0) = length var_0 ==> length (x (rev var_0) (cons2 var_1 nil2)) = S (length var_0)"
49   apply (metis TIP_prop_06.length.simps [ 2 ] TIP_prop_06.x.simps [ 1 ] TIP_prop_06.x.simps [ 2 ]
50     abducted lemma_composite_commutativity_7642270 ) done
51
52 lemma abducted lemma_generalisation_then_extension_7642332: "length (rev var_0) = length var_0"
53   apply (induct "var_0" )
54   apply (simp all )
55   apply (simp add : abducted lemma_tactic_13293202 ) done
56
57 lemma abducted lemma_identity_7579838: "x var_0 nil2 = var_0"
58   apply (induct "var_0" )
59   apply (simp all ) done
60
61 lemma abducted lemma_generalisation_then_extension_7642332: "length (rev (x var_0 var_1)) = length var_0"
62   apply (simp add : abducted lemma_generalisation_then_extension_7642332 abducted lemma_identity_7579838 )
63   done
64
65 lemma abducted lemma_tactic_33224946: "(!a. length (rev (x a var_0)) = t2 (length var_0) (length a)) ==>
66   length (rev (x var_1 (cons2 var_2 var_0))) = S (t2 (length var_0) (length var_1))"
67   apply (metis TIP_prop_06.length.simps [ 2 ] TIP_prop_06.x.simps [ 2 ]
68     abducted lemma_composite_commutativity_7642270 abducted lemma_generalisation_then_extension_7642332 )
69   done
70
71 lemma abducted lemma_generalise_by_renaming_7642286: "length (rev (x var_0 var_1)) = t2 (length var_1) (length var_0)"
72   apply (induct "var_1" arbitrary : var_0 )
73   apply (simp all )
74   apply (simp add : abducted lemma_generalisation_then_extension_17745958 )
75   apply (simp add : abducted lemma_tactic_33224946 )
76   done
77
78 lemma original_goal_7579836: "length (rev (x var_0 var_1)) = t2 (length var_0) (length var_1)"
79   apply (simp add : abducted lemma_commutativity_7642214 abducted lemma_generalise_by_renaming_7642286 )
80   done
81
```

(lemma \rightarrow goal) lemma
Modus ponens
goal

```
Proof state Auto update Update Search
= 100%
theorem
original goal 7579836:
length (rev (x tvar_0.0 tvar_1.0)) =
t2 (length tvar_0.0) (length tvar_1.0)
```



```
The Editor Search Markers Folding View Utilities Macros (F8) Help
File Browser Documentation
TIP_prop_06.thy (WorkplaceProd.thy)
11 | "t2 (S z1) z = S (t2 z1 z)"
12
13 prove property0 :
14   "((length (rev (x y z))) = (t2 (length y) (length z)))"
15
16 lemma abducted lemma_tactic_14045356: "var_0 = t2 var_0 2"
17   apply (induct "var_0" )
18   apply (simp_all ) done
19
20 lemma abducted lemma_tactic_14045358: "(!a. t2 var_0 a = t2 a var_0) ==> S (t2 var_1 var_0) = t2 var_1 (S var_0)"
21   apply (induct "var_1" "var_0" rule : TIP_prop_06.t2.induct )
22   apply (simp_all ) done
23
24 lemma abducted lemma_commutativity_7642214: "t2 var_0 var_1 = t2 var_1 var_0"
25   apply (induct "var_0" arbitrary : var_1 )
26   apply (simp_all )
27   apply (simp add : abducted lemma_tactic_14045356 )
28   using abducted lemma_tactic_14045358 apply force done
29
30 lemma abducted lemma_tactic_13498028: "length var_0 = length (x var_0 nil2)"
31   apply (induct "var_0" )
32   apply (simp_all ) done
33
34 lemma abducted lemma_remove_assumption_47001400: "S (length (x var_0 var_1)) = length (x var_0 (cons2 var_2 var_1))"
35   apply (induct "var_0" arbitrary : var_1 )
36   apply (simp_all ) done
37
38 lemma abducted lemma_tactic_13498034:
39   "(!x. 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))"
40   apply (simp add : abducted lemma_remove_assumption_47001400 ) done
41
42 lemma abducted lemma_composite_commutativity_7642270: "length (x var_0 var_1) = length (x var_1 var_0)"
43   apply (induct "var_0" arbitrary : var_1 )
44   apply (simp_all )
45   using abducted lemma_tactic_13498020 rule blast
46   apply (simp add : abducted lemma_tactic_13498034 ) done
47
48 lemma abducted lemma_tactic_13293202: "length (rev var_0) = length var_0 ==> length (x (rev var_0) (cons2 var_1 nil2)) = S (length var_0)"
49   apply (metis TIP_prop_06.length.simps [ 2 ] TIP_prop_06.x.simps [ 1 ] TIP_prop_06.x.simps [ 2 ]
50         abducted lemma_composite_commutativity_7642270 ) done
51
52 lemma abducted lemma_generalisation_then_extension_7642332: "length (rev var_0) = length var_0"
53   apply (induct "var_0" )
54   apply (simp_all )
55   apply (simp add : abducted lemma_tactic_13293202 ) done
56
57 lemma abducted lemma_identity_7579838: "x var_0 nil2 = var_0"
58   apply (induct "var_0" )
59   apply (simp_all ) done
60
61 lemma abducted lemma_generalisation_then_extension_17745958: "length (rev (x var_0 nil2)) = length var_0"
62   apply (simp add : abducted lemma_generalisation_then_extension_7642332 abducted lemma_identity_7579838 )
63   done
64
65 lemma abducted lemma_tactic_33224946: "(!a. length (rev (x a var_0)) = t2 (length var_0) (length a)) ==>
66   length (rev (x var_1 (cons2 var_2 var_0))) = S (t2 (length var_0) (length var_1))"
67   apply (metis TIP_prop_06.length.simps [ 2 ] TIP_prop_06.x.simps [ 2 ] abducted lemma_composite_commutativity_7642270
68         abducted lemma_generalisation_then_extension_7642332 )
69   done
70
71 lemma abducted lemma_generalise_by_renaming_7642286: "length (rev (x var_0 var_1)) = t2 (length var_1) (length var_0)"
72   apply (induct "var_1" arbitrary : var_0 )
73   apply (simp_all )
74   apply (simp add : abducted lemma_generalisation_then_extension_17745958 )
75   apply (simp add : abducted lemma_tactic_33224946 )
76   done
77
78 lemma original_goal_7579836: "length (rev (x var_0 var_1)) = t2 (length var_0) (length var_1)"
79   apply (simp add : abducted lemma_commutativity_7642214 abducted lemma_generalise_by_renaming_7642286 )
80   done
81
```

(lemma \rightarrow goal) lemma
Modus ponens
goal

```
Proof state Auto update Update Search
100%
theorem
original goal 7579836:
length (rev (x tvar_0 0 tvar_1 0)) =
t2 (length tvar_0 0) (length tvar_1 0)
```

```
File Edit Search Windows Fileings View Utilities Macros (F9) Help
TIP_prop_06.thy (WorkspaceProd.thm)
10 | "t2 (S z2) z = S (t2 z2 z)"
11
12
13 prove property0 :
14   "((length (rev (x y z))) = (t2 (length y) (length z)))"
15
16
17 lemma abducted lemma_tactic_14045356: "var_0 = t2 var_0 2"
18   apply [ induct "var_0" ]
19   apply [ simp_all ] done
20
21 lemma abducted lemma_tactic_14045358: "(!a. t2 var_0 a = t2 a var_0) ==> S (t2 var_1 var_0) = t2 var_1 (S var_0)"
22   apply [ induct "var_1" "var_0" rule: TIP_prop_06.t2.induct ]
23   apply [ simp_all ] done
24
25 lemma abducted lemma_commutativity_7642214: "t2 var_0 var_1 = t2 var_1 var_0"
26   apply [ induct "var_0" arbitrary: var_1 ]
27   apply [ simp_all ]
28   apply [ simp add: abducted lemma_tactic_14045356 ]
29   using abducted lemma_tactic_14045358 apply force done
30
31 lemma abducted lemma_tactic_13498028: "length var_0 = length (x var_0 nil2)"
32   apply [ induct "var_0" ]
33   apply [ simp_all ] done
34
35 lemma abducted lemma_remove_assumption_47001400: "S (length (x var_0 var_1)) = length (x var_0 (cons2 var_2 var_1))"
36   apply [ induct "var_0" arbitrary: var_1 ]
37   apply [ simp_all ] done
38
39 lemma abducted lemma_tactic_13498034:
40   "(!x. 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))"
41   apply [ simp add: abducted lemma_remove_assumption_47001400 ] done
42
43 lemma abducted lemma_composite_commutativity_7642270: "length (x var_0 var_1) = length (x var_1 var_0)"
44   apply [ induct "var_0" arbitrary: var_1 ]
45   apply [ simp_all ]
46   using abducted lemma_tactic_13498034 apply blast
47   apply [ simp add: abducted lemma_tactic_13498034 ] done
48
49 lemma abducted lemma_tactic_13293202: "length (rev var_0) = length var_0 ==> length (x (rev var_0) (cons2 var_1 nil2)) = S (length var_0)"
50   apply [ metis TIP_prop_06.length.simps [ 2 ] TIP_prop_06.x.simps [ 1 ] TIP_prop_06.x.simps [ 2 ]
51         abducted lemma_composite_commutativity_7642270 ] done
52
53 lemma abducted lemma_generalisation_then_extension_7642332: "length (rev var_0) = length var_0"
54   apply [ induct "var_0" ]
55   apply [ simp_all ]
56   apply [ simp add: abducted lemma_tactic_13293202 ] done
57
58 lemma abducted lemma_identity_7579838: "x var_0 nil2 = var_0"
59   apply [ induct "var_0" ]
60   apply [ simp_all ] done
61
62 lemma abducted lemma_generalisation_then_extension_17745958: "length (rev (x var_0 nil2)) = length var_0"
63   apply [ simp add: abducted lemma_generalisation_then_extension_7642332 abducted lemma_identity_7579838 ]
64   done
65
66 lemma abducted lemma_tactic_33224946: "(!a. length (rev (x a var_0)) = t2 (length var_0) (length a)) ==>
67   length (rev (x var_1 (cons2 var_2 var_0))) = S (t2 (length var_0) (length var_1))"
68   apply [ metis TIP_prop_06.length.simps [ 2 ] TIP_prop_06.x.simps [ 2 ] abducted lemma_composite_commutativity_7642270
69         abducted lemma_generalisation_then_extension_7642332 ]
70   done
71
72 lemma abducted lemma_generalise_by_renaming_7642286: "length (rev (x var_0 var_1)) = t2 (length var_1) (length var_0)"
73   apply [ induct "var_1" arbitrary: var_0 ]
74   apply [ simp_all ]
75   apply [ simp add: abducted lemma_generalisation_then_extension_17745958 ]
76   apply [ simp add: abducted lemma_tactic_33224946 ]
77   done
78
79 lemma original goal_7579836: "length (rev (x var_0 var_1)) = t2 (length var_0) (length var_1)"
80   apply [ simp add: abducted lemma_commutativity_7642214 abducted lemma_generalise_by_renaming_7642286 ]
81   done
82
```

(lemma \rightarrow goal) lemma

Modus ponens

goal

already proved

```
Proof state Auto update Update Search
100%
theorem
original goal 7579836:
length (rev (x tvar_0.0 tvar_1.0)) =
t2 (length tvar_0.0) (length tvar_1.0)
```