

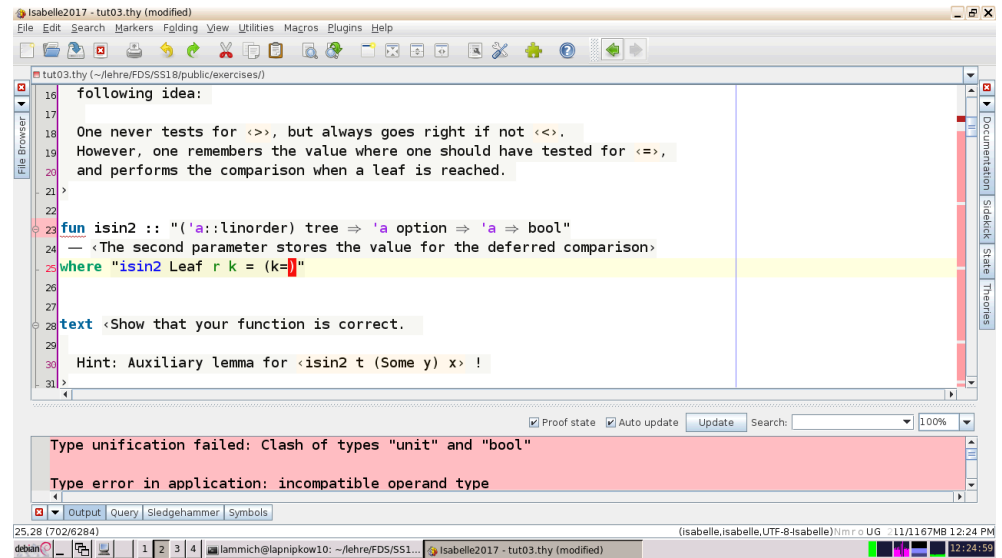
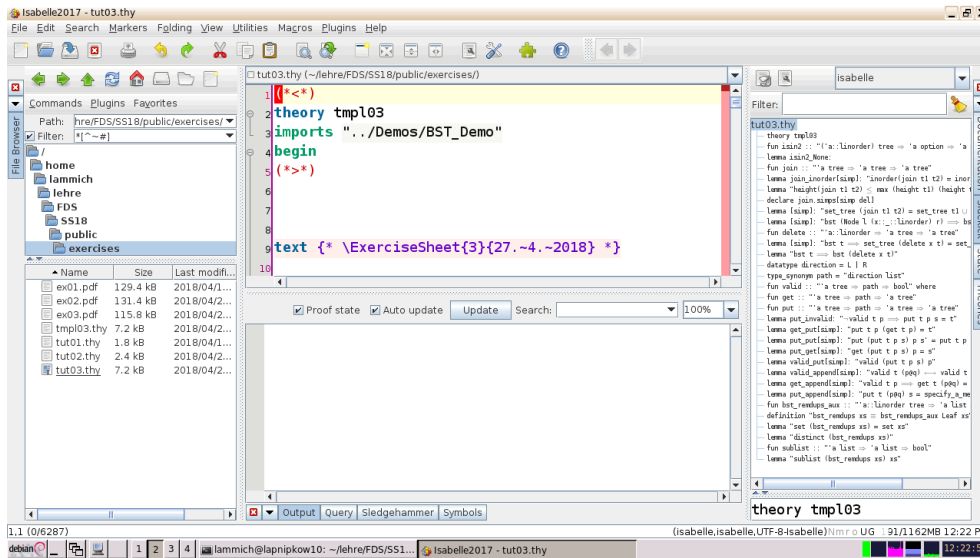
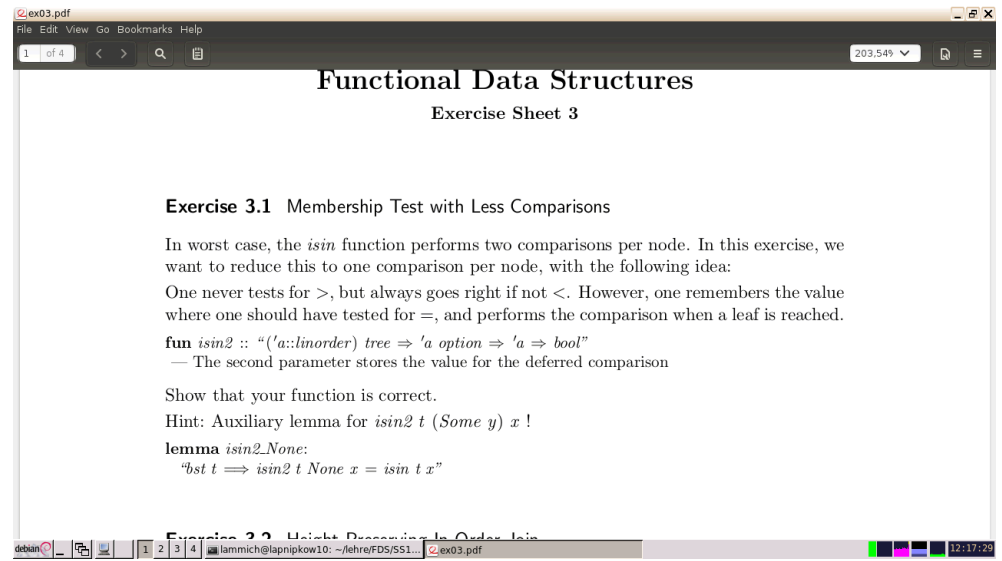
Script generated by TTT

Title: Lammich: FDS Tutorial (27.04.2018)

Date: Fri Apr 27 12:17:29 CEST 2018

Duration: 91:08 min

Pages: 74



Isabelle2017 - tut03.thy (modified)

```

17 One never tests for <>>, but always goes right if not <<>.
18 However, one remembers the value where one should have tested for <=>,
19 and performs the comparison when a leaf is reached.
20
21 >
22
23 fun isin2 :: "('a::linorder) tree => 'a option => 'a => bool"
24   - <The second parameter stores the value for the deferred comparison>
25   where "isin2 Leaf r k = (k=r)"
26
27
28 text <Show that your function is correct.

```

Type unification failed

Type error in application: incompatible operand type

Operator: `op = k :: 'a => bool`

Operand: `r :: 'a option`

25.25 (699/6285) (isabelle,isabelle,UTF-8-isabelle)Nmr o UG 32/1167MB 12:25 PM

Isabelle2017 - tut03.thy

```

17 One never tests for <>>, but always goes right if not <<>.
18 However, one remembers the value where one should have tested for <=>,
19 and performs the comparison when a leaf is reached.
20
21 >
22
23 fun isin2 :: "('a::linorder) tree => 'a option => 'a => bool"
24   - <The second parameter stores the value for the deferred comparison>
25   where "isin2 Leaf r k = (case r of Some a => a=k | None => False)"
26
27
28 text <Show that your function is correct.

```

consts

```

isin2 :: "'a tree => 'a option => 'a => bool"
Missing patterns in function definition:
<math>\lambda v va vb b c. isin2 (v, va, vb) b c</math> = undefined
Found termination order: "{}"

```

25.41 (715/6319) (isabelle,isabelle,UTF-8-isabelle)Nmr o UG 582/1167MB 12:26 PM

Isabelle2017 - tut03.thy

```

19 However, one remembers the value where one should have tested for <=>,
20 and performs the comparison when a leaf is reached.
21
22 >
23 fun isin2 :: "('a::linorder) tree => 'a option => 'a => bool"
24   - <The second parameter stores the value for the deferred comparison>
25   where "isin2 Leaf r k = (case r of Some a => a=k | None => False)"
26
27
28 text <Show that your function is correct.
29
30 Hint: Auxiliary lemma for <math>isin2\ t\ (Some\ y)\ x</math> !

```

consts

```

isin2 :: "'a tree => 'a option => 'a => bool"
Missing patterns in function definition:
<math>\lambda v va vb b c. isin2 (v, va, vb) b c</math> = undefined
Found termination order: "{}"

```

25.55 (729/6319) (isabelle,isabelle,UTF-8-isabelle)Nmr o UG 143/1243MB 12:27 PM

Isabelle2017 - tut03.thy (modified)

```

19 However, one remembers the value where one should have tested for <=>,
20 and performs the comparison when a leaf is reached.
21
22 >
23 fun isin2 :: "('a::linorder) tree => 'a option => 'a => bool"
24   - <The second parameter stores the value for the deferred comparison>
25   where
26     "isin2 Leaf k = (case k of Some a => a=k | None => False)"
27     | "isin2 (Node l )"
28
29
30 text <Show that your function is correct.

```

Type unification failed: Clash of types `"_ => _"` and `"_ tree"`

Type error in application: incompatible operand type

Operator: `isin2 :: 'a tree => 'a option => 'a => bool`

Operand: `Node l :: ??'a => ??'a tree => ??'a tree`

26.15 (696/6343) Input/output complete (isabelle,isabelle,UTF-8-isabelle)Nmr o UG 15/1243MB 12:28 PM

Isabelle2017 - tut03.thy (modified)

```

19 However, one remembers the value where one should have tested for <=,
20 and performs the comparison when a leaf is reached.
21 >
22 >
23 fun isin2 :: "('a::linorder) tree => 'a option => 'a => bool"
24 — <The second parameter stores the value for the deferred comparison>
25 where
26 "isin2 Leaf rem k = (case rem of Some a => a=k | None => False)"
27 | "isin2 (Node l a r) rem k = {}"
28
29 text <Show that your function is correct.

```

Inner syntax error: unexpected end of input
Failed to parse prop

27.33 (779/6361) (isabelle,isabelle,UTF-8-Isabelle)NmroUG 400/1243MB 12:28 PM

Isabelle2017 - tut03.thy (modified)

```

19 However, one remembers the value where one should have tested for <=,
20 and performs the comparison when a leaf is reached.
21 >
22 >
23 fun isin2 :: "('a::linorder) tree => 'a option => 'a => bool"
24 — <The second parameter stores the value for the deferred comparison>
25 where
26 "isin2 Leaf rem k = (case rem of Some a => a=k | None => False)"
27 | "isin2 (Node l a r) rem k = (if k<a then isin2 l else )"
28
29 text <Show that your function is correct.

```

Inner syntax error
Failed to parse prop

27.52 (798/6388) Input/output complete (isabelle,isabelle,UTF-8-Isabelle)NmroUG 643/1243MB 12:30 PM

Isabelle2017 - tut03.thy (modified)

```

21 >
22 >
23 fun isin2 :: "('a::linorder) tree => 'a option => 'a => bool"
24 — <The second parameter stores the value for the deferred comparison>
25 where
26 "isin2 Leaf rem k = (case rem of Some a => a=k | None => False)"
27 | "isin2 (Node l a r) rem k = (
28   if k<a then isin2 l rem k
29   else if )"
30
31 text <Show that your function is correct.

```

Inner syntax error
Failed to parse prop

29.13 (822/6405) (isabelle,isabelle,UTF-8-Isabelle)NmroUG 2562/13MB 12:30 PM

Isabelle2017 - tut03.thy

```

21 >
22 >
23 fun isin2 :: "('a::linorder) tree => 'a option => 'a => bool"
24 — <The second parameter stores the value for the deferred comparison>
25 where
26 "isin2 Leaf rem k = (case rem of Some a => a=k | None => False)"
27 | "isin2 (Node l a r) rem k = (
28   if k<a then isin2 l rem k
29   else isin2 r (a) k)"
30
31 text <Show that your function is correct.

```

Type unification failed
Type error in application: incompatible operand type
Operator: isin2 :: "('a::linorder) tree => 'a option => 'a => bool
Operand: a :: 'a

29.19 (828/6415) (isabelle,isabelle,UTF-8-Isabelle)NmroUG 37/1267MB 12:31 PM

```

Isabelle2017 - tut03.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
30 text <Show that your function is correct.
31
32 Hint: Auxiliary lemma for <isin2 t (Some y) x !
33
34 Lemma isin2_None:
35 "bst t  $\Rightarrow$  isin2 t None x = isin t x" oops
36
37 text <
38 \Exercise{Height-Preserving In-Order Join}
39
40
41

```

38.1 (1001/6420) (isabelle,isabelle,UTF-8-Isabelle)NmroUG 2/7/1267MB 12:32 PM

```

Isabelle2017 - tut03.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
30 text <Show that your function is correct.
31
32 Hint: Auxiliary lemma for <isin2 t (Some y) x !
33
34 Lemma isin2_None:
35 "bst t  $\Rightarrow$  isin2 t None x = isin t x" oops
36
37 text <
38 \Exercise{Height-Preserving In-Order Join}
39
40
41

```

```

proof (prove)
goal (1 subgoal):
1. bst t  $\Rightarrow$  isin2 t None x = isin t x

```

37.1 (958/6420) (isabelle,isabelle,UTF-8-Isabelle)NmroUG 2/7/1267MB 12:32 PM

```

Isabelle2017 - tut03.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
20 and performs the comparison when a leaf is reached.
21 >
22
23 fun isin2 :: "('a::linorder) tree  $\Rightarrow$  'a option  $\Rightarrow$  'a  $\Rightarrow$  bool"
24 — <The second parameter stores the value for the deferred comparison>
25 where
26 "isin2 Leaf rem k = (case rem of Some a  $\Rightarrow$  a=k | None  $\Rightarrow$  False)"
27 | "isin2 (Node l a r) rem k = (
28   if k<a then isin2 l rem k
29   else isin2 r (Some a) k)"
30
31

```

```

consts
isin2 :: "'a tree  $\Rightarrow$  'a option  $\Rightarrow$  'a  $\Rightarrow$  bool"
Found termination order: "( $\lambda$ p. size (fst p)) <math>*</math>{> {}"

```

29.28 (837/6420) (isabelle,isabelle,UTF-8-Isabelle)NmroUG 2/7/1267MB 12:33 PM

```

Isabelle2017 - tut03.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
21 >
22
23 fun isin2 :: "('a::linorder) tree  $\Rightarrow$  'a option  $\Rightarrow$  'a  $\Rightarrow$  bool"
24 — <The second parameter stores the value for the deferred comparison>
25 where
26 "isin2 Leaf rem k = (case rem of Some a  $\Rightarrow$  a=k | None  $\Rightarrow$  False)"
27 | "isin2 (Node l a r) rem k = (
28   if k<a then isin2 l rem k
29   else isin2 r (Some a) k)"
30
31
32 text <Show that your function is correct.

```

```

consts
isin2 :: "'a tree  $\Rightarrow$  'a option  $\Rightarrow$  'a  $\Rightarrow$  bool"
Found termination order: "( $\lambda$ p. size (fst p)) <math>*</math>{> {}"

```

29.28 (837/6420) (isabelle,isabelle,UTF-8-Isabelle)NmroUG 2/7/1267MB 12:33 PM

```

Isabelle2017 - tut03.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
31 text <Show that your function is correct.
32
33
34 Hint: Auxiliary lemma for <isin2 t (Some y) x> !
35
36
37 Lemma isin2_None:
38 "bst t  $\implies$  isin2 t None k = isin t k"
39 apply (induction t)
40 apply auto
41
42 text <

```

Proof state Auto update Update Search: 100%

```

proof (prove)
goal (8 subgoals):
1.  $\bigwedge t1\ x2\ t2.$ 
  [bst t1; bst t2;  $\forall x \in \text{set\_tree } t1. x < x2$ ;  $\forall x \in \text{set\_tree } t2. x2 < x$ ; isin2 t1 None k; isin t1 k;
  isin2 t2 None k;  $x2 < k$ ;  $\neg k < x2$ ; isin t2 k]
 $\implies$  isin2 t2 (Some x2) k

```

Output Query Sledgehammer Symbols

39.13 (1.030/6450) (isabelle,isabelle,UTF-8-Isabelle)NmroUG 609/1.267MB 12:34 PM

```

Isabelle2017 - tut03.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
31 text <Show that your function is correct.
32
33
34 Hint: Auxiliary lemma for <isin2 t (Some y) x> !
35
36
37 Lemma isin2_None:
38 "bst t  $\implies$  isin2 t None k = isin t k"
39 apply (induction t)
40 apply auto
41
42 text <

```

Proof state Auto update Update Search: 100%

```

  isin2 t2 None k;  $x2 < k$ ;  $\neg k < x2$ ; isin t2 k]
 $\implies$  isin2 t2 (Some x2) k
2.  $\bigwedge t1\ t2.$ 
  [bst t1; bst t2;  $\forall x \in \text{set\_tree } t1. x < k$ ;  $\forall x \in \text{set\_tree } t2. k < x$ ; isin2 t1 None k; isin t1 k;
  isin2 t2 None k; isin t2 k]
 $\implies$  isin2 t2 (Some k) k

```

Output Query Sledgehammer Symbols

39.13 (1.030/6450) (isabelle,isabelle,UTF-8-Isabelle)NmroUG 620/1.267MB 12:34 PM

```

Isabelle2017 - tut03.thy (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
31 text <Show that your function is correct.
32
33
34 Hint: Auxiliary lemma for <isin2 t (Some y) x> !
35
36
37 Lemma
38 "bst t  $\implies$  isin2 t (Some a) k =  $\langle \rangle$ "
39
40 Lemma isin2_None:
41 "bst t  $\implies$  isin2 t None k = isin t k"
42 apply (induction t)

```

Proof state Auto update Update Search: 100%

```

Inner syntax error: unexpected end of input
Failed to parse prop

```

Output Query Sledgehammer Symbols

38.35 (981/6494) (isabelle,isabelle,UTF-8-Isabelle)NmroUG 601/1.267MB 12:36 PM

```

Isabelle2017 - tut03.thy (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
31 text <Show that your function is correct.
32
33
34 Hint: Auxiliary lemma for <isin2 t (Some y) x> !
35
36
37 Lemma
38 "bst t  $\implies$  isin2 t (Some a) k  $\implies$  (isin t k)"
39
40 Lemma isin2_None:
41 "bst t  $\implies$  isin2 t None k = isin t k"
42 apply (induction t)

```

Proof state Auto update Update Search: 100%

```

proof (prove)
goal (1 subgoal):
1. [bst t; isin2 t (Some a) k]  $\implies$  isin t k
Auto Quickcheck found a counterexample:
t =  $\langle \rangle$ 
a = a1
k = a

```

Output Query Sledgehammer Symbols

38.32 (978/6502) Input/output complete (isabelle,isabelle,UTF-8-Isabelle)NmroUG 601/1.267MB 12:37 PM

```

Isabelle2017 - tut03.thy (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
31 text <Show that your function is correct.
32
33
34 Hint: Auxiliary lemma for <isin2 t (Some y) x> !
35
36
37 Lemma
38 "bst t ==> isin2 t (Some a) k ==> (isin t k)"
39
40 Lemma isin2_None:
41 "bst t ==> isin2 t None k == isin t k"
42 apply (induction t)

proof (prove)
goal (1 subgoal):
1. [bst t; isin2 t (Some a) k] ==> isin t k
Auto Quickcheck found a counterexample:
t = {}
a = a_1
k = -

```

```

Isabelle2017 - tut03.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
31 text <Show that your function is correct.
32
33
34 Hint: Auxiliary lemma for <isin2 t (Some y) x> !
35
36
37 Lemma
38 "bst t ==> isin2 t (Some a) k ==> (isin t k)"
39
40 Lemma isin2_None:
41 "bst t ==> isin2 t None k == isin t k"
42 apply (induction t)

proof (prove)
goal (1 subgoal):
1. bst t ==> isin2 t (Some a) k == isin t k
Auto Quickcheck found a counterexample:
t = {}
a = a_1
k = -

```

```

Isabelle2017 - tut03.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
36
37 Lemma
38 "bst t ==> isin2 t (Some a) k ==> (a=k v isin t k)"
39
40 Lemma isin2_None:
41 "bst t ==> isin2 t None k == isin t k"
42 apply (induction t)
43 apply auto
44

proof (prove)
goal (1 subgoal):
1. bst t ==> isin2 t (Some a) k == (a = k v isin t k)
Auto Quickcheck found a counterexample:
t = {{}}, a_1, {}
a = a_2
k = a_2
Evaluated terms:
isin2 t (Some a) k = False

```

```

Isabelle2017 - tut03.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
36
37 Lemma
38 "bst t ==> isin2 t (Some a) k ==> (a=k v isin t k)"
39
40 Lemma isin2_None:
41 "bst t ==> isin2 t None k == isin t k"
42 apply (induction t)
43 apply auto
44

proof (prove)
goal (8 subgoals):
1. ^t1 x2 t2.
[bst t1; bst t2; v xset_tree t1. x < x2; v xset_tree t2. x2 < x; isin2 t1 None k; isin t1 k;
isin2 t2 None k; x2 < k; ~ k < x2; isin t2 k]
==> isin2 t2 (Some x2) k
2. ^t1 t2.
[bst t1; bst t2; v xset_tree t1. x < k; v xset_tree t2. k < x; isin2 t1 None k; isin t1 k;
isin2 t2 None k; isin t2 k]

```

```

Isabelle2017 - tut03.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
34 Hint: Auxiliary lemma for <isin2 t (Some y) x> !
35 >
36
37 Lemma
38 "bst t => ∀x<set_tree t. a<x => isin2 t (Some a) k ↔ (a=k ∨ isin t k)"
39
40 Lemma isin2_None:
41 "bst t => isin2 t None k = isin t k"
42 apply (induction t)

proof (prove)
goal (1 subgoal):
1. [bst t; ∀x<set_tree t. a < x] => isin2 t (Some a) k = (a = k ∨ isin t k)

```

```

Isabelle2017 - tut03.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
35 >
36
37 Lemma
38 "bst t => ∀x<set_tree t. a<x => isin2 t (Some a) k ↔ (a=k ∨ isin t k)"
39
40
41
42 Lemma isin2_None:
43 "bst t => isin2 t None k = isin t k"

proof (prove)
goal (1 subgoal):
1. [bst t; ∀x<set_tree t. a < x] => isin2 t (Some a) k = (a = k ∨ isin t k)

```

```

Isabelle2017 - tut03.thy (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
35 >
36
37 Lemma
38 "bst t => ∀x<set_tree t. a<x => isin2 t (Some a) k ↔ (a=k ∨ isin t k)"
39 apply
40 apply (keyword)
41
42 Lemma isin2_None:
43 "bst t => isin2 t None k = isin t k"

proof (prove)
goal (1 subgoal):
1. [bst t; ∀x<set_tree t. a < x] => isin2 t (Some a) k = (a = k ∨ isin t k)

```

```

Isabelle2017 - tut03.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
44 Lemma isin2_None:
45 "bst t => isin2 t None k = isin t k"
46 apply (induction t)
47 apply (auto simp: aux)
48 done
49
50
51 text <
52 \Exercise{Height-Preserving In-Order Join}

theorem isin2_None: bst ?t => isin2 ?t None ?k = isin ?t ?k

```

```

Isabelle2017 - tut03.thy
File Edit Search Markers Folding View Utilities Magros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
40 apply auto
41 done
42
43
44 Lemma isin2_None:
45 "bst t  $\implies$  isin2 t None k = isin t k"
46 apply (induction t)
47 apply (auto simp: aux)
48 done

theorem isin2_None: bst ?t  $\implies$  isin2 ?t None ?k = isin ?t ?k

```

```

Isabelle2017 - tut03.thy
File Edit Search Markers Folding View Utilities Magros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
26 "isin2 Leaf rem k = (case rem of Some a  $\implies$  a=k | None  $\implies$  False)"
27 "isin2 (Node l a r) rem k = (
28   if k < a then isin2 l rem k
29   else isin2 r (Some a) k)"
30
31
32 text <Show that your function is correct.
33
34 Hint: Auxiliary lemma for <isin2 t (Some y) x> !

consts
isin2 :: "'a tree  $\implies$  'a option  $\implies$  'a  $\implies$  bool"
Found termination order: "( $\lambda$ p. size (fst p)) <+mlex* {}"

```

Functional Data Structures
Exercise Sheet 3

Exercise 3.1 Membership Test with Less Comparisons

In worst case, the *isin* function performs two comparisons per node. In this exercise, we want to reduce this to one comparison per node, with the following idea:

One never tests for $>$, but always goes right if not $<$. However, one remembers the value where one should have tested for $=$, and performs the comparison when a leaf is reached.

fun *isin2* :: "('a::linorder) tree \implies 'a option \implies 'a \implies bool"
 — The second parameter stores the value for the deferred comparison

Show that your function is correct.
 Hint: Auxiliary lemma for *isin2* t (Some y) x !

lemma *isin2_None*:
 "bst t \implies isin2 t None x = isin t x"

```

Isabelle2017 - tut03.thy
File Edit Search Markers Folding View Utilities Magros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
46 apply (induction t)
47 apply (auto simp: aux)
48 done
49
50
51 text <
52 \Exercise{Height-Preserving In-Order Join}
53 Write a function that joins two binary trees such that
54 ■ The in-order traversal of the new tree is the concatenation of the in-order traversals of the original

```


Isabelle2017 - tut03.thy (modified)

```

57 Hint: Once you got the function right, proofs are easy!
58 >
59 fun join :: "'a tree => 'a tree"
60 where
61   "join Leaf t = t"
62 | "join t Leaf = t"
63 | "join (Node l1 a1 r1) (Node l2 a2 r2) = (
64   case join r1 l2 of
65     Leaf => Node l1 a1 (Node Leaf a2 r2)
66   | Node l a r => Node (Node l1 a1 l) a (Node r a2 r2))"
67
68 Lemma join_inorder[simp]: "inorder(join t1 t2) = inorder t1 @ inorder t2"
69 apply
70 apply (keyword)
71 Lemma "height(join t1 t2) <= max (height t1) (height t2) + 1"
72 oops
73
74

```

proof (prove)

69.7 (1.902/6804) (isabelle,isabelle,UTF-8-isabelle)tmr o UG 465.1219MB 1:17 PM

Isabelle2017 - tut03.thy

```

63 "join (Node l1 a1 r1) (Node l2 a2 r2) = (
64   case join r1 l2 of
65     Leaf => Node l1 a1 (Node Leaf a2 r2)
66   | Node l a r => Node (Node l1 a1 l) a (Node r a2 r2))"
67
68 Lemma join_inorder[simp]: "inorder(join t1 t2) = inorder t1 @ inorder t2"
69 apply (induction t1 t2 rule: join.induct)
70 apply auto
71
72 Lemma "height(join t1 t2) <= max (height t1) (height t2) + 1"
73 oops
74

```

proof (prove)

goal (1 subgoal):

1. $\bigwedge l1\ a1\ r1\ l2\ a2\ r2.$
 $\text{inorder (join r1 l2)} = \text{inorder r1 @ inorder l2} \implies$
 $\text{inorder (case join r1 l2 of } \langle \rangle \Rightarrow \langle l1, a1, \langle \rangle, a2, r2 \rangle \mid \langle l, a, r \rangle \Rightarrow \langle \langle l1, a1, l \rangle, a, \langle r, a2, r2 \rangle \rangle) =$
 $\text{inorder l1 @ a1 \# inorder r1 @ inorder l2 @ a2 \# inorder r2}$

70.13 (1.952/6854) (isabelle,isabelle,UTF-8-isabelle)tmr o UG 28/1193MB 1:18 PM

Isabelle2017 - tut03.thy

```

63 "join (Node l1 a1 r1) (Node l2 a2 r2) = (
64   case join r1 l2 of
65     Leaf => Node l1 a1 (Node Leaf a2 r2)
66   | Node l a r => Node (Node l1 a1 l) a (Node r a2 r2))"
67
68 Lemma join_inorder[simp]: "inorder(join t1 t2) = inorder t1 @ inorder t2"
69 apply (induction t1 t2 rule: join.induct)
70 apply auto
71
72 Lemma "height(join t1 t2) <= max (height t1) (height t2) + 1"
73 oops
74

```

proof (prove)

goal (1 subgoal):

1. $\bigwedge l1\ a1\ r1\ l2\ a2\ r2.$
 $\text{inorder (join r1 l2)} = \text{inorder r1 @ inorder l2} \implies$
 $\text{inorder (case join r1 l2 of } \langle \rangle \Rightarrow \langle l1, a1, \langle \rangle, a2, r2 \rangle \mid \langle l, a, r \rangle \Rightarrow \langle \langle l1, a1, l \rangle, a, \langle r, a2, r2 \rangle \rangle) =$
 $\text{inorder l1 @ a1 \# inorder r1 @ inorder l2 @ a2 \# inorder r2}$

70.9 (1.948/6854) (isabelle,isabelle,UTF-8-isabelle)tmr o UG 2/1193MB 1:18 PM

Isabelle2017 - tut03.thy

```

63 "join (Node l1 a1 r1) (Node l2 a2 r2) = (
64   case join r1 l2 of
65     Leaf => Node l1 a1 (Node Leaf a2 r2)
66   | Node l a r => Node (Node l1 a1 l) a (Node r a2 r2))"
67
68 Lemma join_inorder[simp]: "inorder(join t1 t2) = inorder t1 @ inorder t2"
69 apply (induction t1 t2 rule: join.induct)
70 apply (auto split: tree.splits)
71
72 Lemma "height(join t1 t2) <= max (height t1) (height t2) + 1"
73 oops
74

```

proof (prove)

goal:

No subgoals!

70.21 (1.960/6875) (isabelle,isabelle,UTF-8-isabelle)tmr o UG 86/1193MB 1:19 PM

Isabelle2017 - tut03.thy

```

68 lemma join_inorder[simp]: "inorder(join t1 t2) = inorder t1 @ inorder t2"
69   apply (induction t1 t2 rule: join.induct)
70   by (auto split: tree.splits)
71
72 lemma "height(join t1 t2) ≤ max (height t1) (height t2) + 1"
73   apply (induction t1 t2 rule: join.induct)
74   by (auto split: tree.splits)
75
76 text <
77   \Exercise{Implement Delete}
78   Implement delete using the <join> function from last exercise.
79 >

```

theorem height (join ?t1.0 ?t2.0) ≤ max (height ?t1.0) (height ?t2.0) + 1

75.1 (2108/6940) (isabelle.isabelle.UTF-8-isabelle)lmmr o UG 588/1193MB 1:20 PM

Isabelle2017 - tut03.thy

```

63 | "join (Node l1 a1 r1) (Node l2 a2 r2) = (
64   case join r1 l2 of
65     Leaf ⇒ Node r1 a1 (Node Leaf a2 r2)
66     | Node l a r ⇒ Node (Node l1 a1 l) a (Node r a2 r2))"
67
68 lemma join_inorder[simp]: "inorder(join t1 t2) = inorder t1 @ inorder t2"
69   apply (induction t1 t2 rule: join.induct)
70   by (auto split: tree.splits)
71
72 lemma "height(join t1 t2) ≤ max (height t1) (height t2) + 1"
73   apply (induction t1 t2 rule: join.induct)
74   by (auto split: tree.splits)

```

theorem join_inorder: inorder (join ?t1.0 ?t2.0) = inorder ?t1.0 @ inorder ?t2.0

Failed to finish proof:
goal (1 subgoal):
1. $\wedge l1. l1 = \langle \rangle$

66.22 (1782/6940) (isabelle.isabelle.UTF-8-isabelle)lmmr o UG 738/1193MB 1:20 PM

Isabelle2017 - tut03.thy (modified)

```

60 where
61   "join Leaf t = t"
62 | "join t Leaf = t"
63 | "join (Node l1 a1 r1) (Node l2 a2 r2) = (
64   case join r1 l2 of
65     Leaf ⇒ Node l1 a1 (Node Leaf a2 r2)
66     | Node l a r ⇒ Node (Node l1 a1 l) a (Node r a2 r2))"
67
68 lemma join_inorder[simp]: "inorder(join t1 t2) = inorder t1 @ inorder t2"
69   apply (induction t1 t2 rule: join.induct)
70   by (auto split: tree.splits)
71

```

consts
join :: "'a tree ⇒ 'a tree ⇒ 'a tree"
Found termination order: "{λp. size (snd p)} <+mlex*+ {}"

65.23 (1738/6940) (isabelle.isabelle.UTF-8-isabelle)lmmr o UG 51/1170MB 1:20 PM

Isabelle2017 - tut03.thy

```

68 lemma join_inorder[simp]: "inorder(join t1 t2) = inorder t1 @ inorder t2"
69   apply (induction t1 t2 rule: join.induct)
70   by (auto split: tree.splits)
71
72 lemma "height(join t1 t2) ≤ max (height t1) (height t2) + 1"
73   apply (induction t1 t2 rule: join.induct)
74   by (auto split: tree.splits)
75
76 text <
77   \Exercise{Implement Delete}
78   Implement delete using the <join> function from last exercise.
79 >

```

theorem height (join ?t1.0 ?t2.0) ≤ max (height ?t1.0) (height ?t2.0) + 1

75.1 (2108/6940) (isabelle.isabelle.UTF-8-isabelle)lmmr o UG 51/1170MB 1:23 PM

```
Isabelle2017 - tut03.thy
File Edit Search Markers Folding View Utilities Magros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
68 Lemma join_inorder[simp]: "inorder(join t1 t2) = inorder t1 @ inorder t2"
69   apply (induction t1 t2 rule: join.induct)
70   by (auto split: tree.splits)
71
72 Lemma "height(join t1 t2) ≤ max (height t1) (height t2) + 1"
73   apply (induction t1 t2 rule: join.induct)
74   by (auto split: tree.splits)
75
76 text <
77   \Exercise{Implement Delete}
78   Implement delete using the <join> function from last exercise.
79 >
80
81 text <Note: At this point, we are not interested in the implementation details
82 of join any more, but just in its specification, i.e., what it does to trees.
83 Thus, as first step, we declare its equations to not being automatically unfolded.
84 >
85
86 declare join.simps[simp del]
87
75.1 (21.08/6940) (isabelle,isabelle,UTF-8-isabelle)tmr o UG 1170MB 1:23 PM
1 2 3 4 iamlich@lapnikow10: ~/lehre/FDS/SS1... Isabelle2017 - tut03.thy 13:23:24
```

```
Isabelle2017 - tut03.thy
File Edit Search Markers Folding View Utilities Magros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
68 Lemma join_inorder[simp]: "inorder(join t1 t2) = inorder t1 @ inorder t2"
69   apply (induction t1 t2 rule: join.induct)
70   by (auto split: tree.splits)
71
72 Lemma "height(join t1 t2) ≤ max (height t1) (height t2) + 1"
73   apply (induction t1 t2 rule: join.induct)
74   by (auto split: tree.splits)
75
76 text <
77   \Exercise{Implement Delete}
78   Implement delete using the <join> function from last exercise.
79 >
80
81 text <Note: At this point, we are not interested in the implementation details
82 of join any more, but just in its specification, i.e., what it does to trees.
83 Thus, as first step, we declare its equations to not being automatically unfolded.
84 >
85
86 declare join.simps[simp del]
87
81.1 (22.16/6940) (isabelle,isabelle,UTF-8-isabelle)tmr o UG 1170MB 1:24 PM
1 2 3 4 iamlich@lapnikow10: ~/lehre/FDS/SS1... Isabelle2017 - tut03.thy 13:24:49
```

```
Isabelle2017 - tut03.thy
File Edit Search Markers Folding View Utilities Magros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
61 "join Leaf t = t"
62 | "join t Leaf = t"
63 | "join (Node l1 a1 r1) (Node l2 a2 r2) = (
64   case join r1 l2 of
65     Leaf ⇒ Node l1 a1 (Node Leaf a2 r2)
66     | Node l a r ⇒ Node (Node l1 a1 l) a (Node r a2 r2))"
67
68 Lemma join_inorder[simp]: "inorder(join t1 t2) = inorder t1 @ inorder t2"
69   apply (induction t1 t2 rule: join.induct)
70   by (auto split: tree.splits)
71
72 Lemma "height(join t1 t2) ≤ max (height t1) (height t2) + 1"
73   apply (induction t1 t2 rule: join.induct)
74   by (auto split: tree.splits)
75
76 text <
77   \Exercise{Implement Delete}
78   Implement delete using the <join> function from last exercise.
79 >
76.1 (21.09/6940) (isabelle,isabelle,UTF-8-isabelle)tmr o UG 1170MB 1:26 PM
1 2 3 4 iamlich@lapnikow10: ~/lehre/FDS/SS1... Isabelle2017 - tut03.thy 13:26:28
```

```
Isabelle2017 - tut03.thy
File Edit Search Markers Folding View Utilities Magros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
65 Leaf ⇒ Node l1 a1 (Node Leaf a2 r2)
66 | Node l a r ⇒ Node (Node l1 a1 l) a (Node r a2 r2)"
67
68 Lemma join_inorder[simp]: "inorder(join t1 t2) = inorder t1 @ inorder t2"
69   apply (induction t1 t2 rule: join.induct)
70   by (auto split: tree.splits)
71
72 Lemma "height(join t1 t2) ≤ max (height t1) (height t2) + 1"
73   apply (induction t1 t2 rule: join.induct)
74   by (auto split: tree.splits)
75
76 text <
77   \Exercise{Implement Delete}
78   Implement delete using the <join> function from last exercise.
79 >
80
81 text <Note: At this point, we are not interested in the implementation details
82 of join any more, but just in its specification, i.e., what it does to trees.
83 Thus, as first step, we declare its equations to not being automatically unfolded.
84 >
80.1 (22.15/6940) (isabelle,isabelle,UTF-8-isabelle)tmr o UG 1170MB 1:26 PM
1 2 3 4 iamlich@lapnikow10: ~/lehre/FDS/SS1... Isabelle2017 - tut03.thy 13:26:38
```

Isabelle2017 - tut03.thy (modified)

```

79 >
80 >
81 text <Note: At this point, we are not interested in the implementation details
82 of join any more, but just in its specification, i.e., what it does to trees.
83 Thus, as first step, we declare its equations to not being automatically unfolded.
84 >
85 >
86 thm join.simps
87
88 declare join.simps[simp del]
89
90 text <Both, <set_tree> and <bst> can be expressed by the inorder traversal over trees:>

```

88.29 (2508/6956) (isabelle.isabelle.UTF-8-isabelle)tmr o UG 76/1147MB 1:27 PM

Isabelle2017 - tut03.thy (modified)

```

86 thm join.simps
87
88 declare join.simps[simp del]
89
90 text <Both, <set_tree> and <bst> can be expressed by the inorder traversal over trees:>
91
92 thm set_inorder[symmetric] bst_iff_sorted_wrt_less
93
94 text <Note: As @<thm [source] set_inorder> is declared as simp.
95 Be careful not to have both directions of the lemma in the simpset at the
96 same time, otherwise the simplifier is likely to loop.
97
98
99

```

- set_tree ?t = set (inorder ?t)
- bst ?t = sorted_wrt op < (inorder ?t)

92.51 (2649/6956) (isabelle.isabelle.UTF-8-isabelle)tmr o UG 76/1147MB 1:28 PM

Isabelle2017 - tut03.thy (modified)

```

86 thm join.simps
87
88 declare join.simps[simp del]
89
90 text <Both, <set_tree> and <bst> can be expressed by the inorder traversal over trees:>
91
92 thm set_inorder[symmetric] bst_iff_sorted_wrt_less
93
94 text <Note: As @<thm [source] set_inorder> is declared as simp.
95 Be careful not to have both directions of the lemma in the simpset at the
96 same time, otherwise the simplifier is likely to loop.
97
98
99

```

- set_tree ?t = set (inorder ?t)
- bst ?t = sorted_wrt op < (inorder ?t)

92.13 (2611/6956) (isabelle.isabelle.UTF-8-isabelle)tmr o UG 76/1147MB 1:29 PM

Isabelle2017 - tut03.thy (modified)

```

86 thm join.simps
87
88 declare join.simps[simp del]
89
90 text <Both, <set_tree> and <bst> can be expressed by the inorder traversal over trees:>
91
92 thm set_inorder[symmetric] bst_iff_sorted_wrt_less
93
94 text <Note: As @<thm [source] set_inorder> is declared as simp.
95 Be careful not to have both directions of the lemma in the simpset at the
96 same time, otherwise the simplifier is likely to loop.
97
98
99

```

93.1 (2650/6956) (isabelle.isabelle.UTF-8-isabelle)tmr o UG 76/1147MB 1:30 PM

Isabelle2017 - tut03.thy (modified)

```

102 Alternatively, you can write <declare set_inorder[simp_del]> to
103 remove it once and forall.
104 >
105
106 text "For the <sorted_wrt> predicate, you might want to use these lemmas as simp:"
107 thm sorted_wrt_append sorted_wrt_Cons
108
109
110 text <Show that join preserves the set of entries>
111 lemma [simp]: "set_tree (join t1 t2) = set_tree t1 ∪ set_tree t2"
112 oops
113

```

Proof state: 100%

```

• transp ?P ⇒
  sorted_wrt ?P (?xs @ ?ys) = (sorted_wrt ?P ?xs ∧ sorted_wrt ?P ?ys ∧ (∀x∈set ?xs. ∀y∈set ?ys. ?P x y))
• transp ?P ⇒ sorted_wrt ?P (?x # ?xs) = ((∀y∈set ?xs. ?P x y) ∧ sorted_wrt ?P ?xs)

```

109.1 (3259/6956) (isabelle.isabelle.UTF-8-isabelle)tmr o UG 3/11/14 7MB 1:32 PM

Isabelle2017 - tut03.thy (modified)

```

106 text "For the <sorted_wrt> predicate, you might want to use these lemmas as simp:"
107 thm sorted_wrt_append sorted_wrt_Cons
108
109
110 text <Show that join preserves the set of entries>
111 lemma [simp]: "set_tree (join t1 t2) = set_tree t1 ∪ set_tree t2"
112 oops
113
114 text <Show that joining the left and right child of a BST is again a BST:>
115
116 lemma [simp]: "bst (Node l (x:::linorder) r) ⇒ bst (join l r)"
117 oops

```

Proof state: 100%

```

proof (prove)
goal (1 subgoal):
1. set_tree (join t1 t2) = set_tree t1 ∪ set_tree t2

```

111.1 (3311/6956) (isabelle.isabelle.UTF-8-isabelle)tmr o UG 3/11/14 7MB 1:32 PM

Isabelle2017 - tut03.thy (modified)

```

106 text "For the <sorted_wrt> predicate, you might want to use these lemmas as simp:"
107 thm sorted_wrt_append sorted_wrt_Cons
108
109
110 text <Show that join preserves the set of entries>
111 lemma [simp]: "set_tree (join t1 t2) = set_tree t1 ∪ set_tree t2"
112 oops
113
114 text <Show that joining the left and right child of a BST is again a BST:>
115
116 lemma [simp]: "bst (Node l (x:::linorder) r) ⇒ bst (join l r)"
117 oops

```

Proof state: 100%

```

proof (prove)
goal (1 subgoal):
1. set_tree (join t1 t2) = set_tree t1 ∪ set_tree t2

```

110.1 (3260/6956) (isabelle.isabelle.UTF-8-isabelle)tmr o UG 3/11/14 7MB 1:32 PM

Isabelle2017 - tut03.thy

```

106 text "For the <sorted_wrt> predicate, you might want to use these lemmas as simp:"
107 thm sorted_wrt_append sorted_wrt_Cons
108
109
110 text <Show that join preserves the set of entries>
111 lemma [simp]: "set_tree (join t1 t2) = set_tree t1 ∪ set_tree t2"
112
113
114 text <Show that joining the left and right child of a BST is again a BST:>
115
116 lemma [simp]: "bst (Node l (x:::linorder) r) ⇒ bst (join l r)"

```

Proof state: 100%

```

proof (prove)
goal (1 subgoal):
1. set_tree (join t1 t2) = set_tree t1 ∪ set_tree t2

```

111.3 (3313/6952) (isabelle.isabelle.UTF-8-isabelle)tmr o UG 3/11/14 7MB 1:33 PM

```
Isabelle2017 - tut03.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
98 thm join.simps
99
100 declare join.simps[simp del]
101
102 text <Both, <set_tree> and <bst> can be expressed by the inorder traversal over trees:>
103
104 thm set_inorder[symmetric] bst_iff_sorted_wrt_less
105
106 text <Note: As @<thm [source] set_inorder> is declared as simp.
107 Be careful not to have both directions of the lemma in the simpset at the
108 same time, otherwise the simplifier is likely to loop.
109
110
111 • set_tree ?t = set (inorder ?t)
112 • bst ?t = sorted_wrt op < (inorder ?t)
```

```
Isabelle2017 - tut03.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
107
108 text "For the <sorted_wrt> predicate, you might want to use these lemmas as simp:"
109 thm sorted_wrt_append sorted_wrt_Cons
110
111 text <Show that join preserves the set of entries>
112 lemma [simp]: "set_tree (join t1 t2) = set_tree t1 ∪ set_tree t2"
113 apply (simp only: set_inorder[symmetric])
114 apply (simp only: join_inorder)
115
116 text <Show that joining the left and right child of a BST is again a BST:>
117
118 lemma [simp]: "bst (Node l (x:::linorder) r) ⇒ bst (join l r)"
119
120
121 proof (prove)
122 goal (1 subgoal):
123 1. set (inorder t1 @ inorder t2) = set (inorder t1) ∪ set (inorder t2)
```

```
Isabelle2017 - tut03.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
111 text <Show that join preserves the set of entries>
112 lemma [simp]: "set_tree (join t1 t2) = set_tree t1 ∪ set_tree t2"
113 by (simp add: set_inorder[symmetric])
114
115 text <Show that joining the left and right child of a BST is again a BST:>
116
117 lemma [simp]: "bst (Node l (x:::linorder) r) ⇒ bst (join l r)"
118 oops
119
120 text <Implement a delete function using the idea contained in the lemmas above.>
121 fun delete :: "'a::linorder ⇒ 'a tree ⇒ 'a tree"
122 where
123
124
125 proof (prove)
126 goal (1 subgoal):
127 1. bst (l, x, r) ⇒ bst (join l r)
```

```
Isabelle2017 - tut03.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
111 text <Show that join preserves the set of entries>
112 lemma [simp]: "set_tree (join t1 t2) = set_tree t1 ∪ set_tree t2"
113 by (simp add: set_inorder[symmetric])
114
115 text <Show that joining the left and right child of a BST is again a BST:>
116
117 lemma [simp]: "bst (Node l (x:::linorder) r) ⇒ bst (join l r)"
118 oops
119
120 text <Implement a delete function using the idea contained in the lemmas above.>
121 fun delete :: "'a::linorder ⇒ 'a tree ⇒ 'a tree"
122 where
123
124
125 proof (prove)
126 goal (1 subgoal):
127 1. bst (l, x, r) ⇒ bst (join l r)
```

```
Isabelle2017 - tut03.thy
File Edit Search Markers Folding View Utilities Magros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
111 text <Show that join preserves the set of entries>
112 Lemma [simp]: "set_tree (join t1 t2) = set_tree t1 U set_tree t2"
113 by (simp add: set_inorder[symmetric])
114
115 text <Show that joining the left and right child of a BST is again a BST:>
116
117 Lemma [simp]: "bst (Node l (x:::linorder) r) ==> bst (join l r)"
118 oops
119
120 text <Implement a delete function using the idea contained in the lemmas above.>
121 fun delete :: "'a::linorder => 'a tree => 'a tree"
122 where

118.7 (3595/7019) (isabelle.isabelle.UTF-8-isabelle)tmr o UG 109/1104MB 1:37 PM
debian 1 2 3 4 lamlich@lapnikow10: ~/lehre/FDS/SS1... Isabelle2017 - tut03.thy 13:37:35
```

```
Isabelle2017 - tut03.thy
File Edit Search Markers Folding View Utilities Magros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
111 text <Show that join preserves the set of entries>
112 Lemma [simp]: "set_tree (join t1 t2) = set_tree t1 U set_tree t2"
113 by (simp add: set_inorder[symmetric])
114
115 text <Show that joining the left and right child of a BST is again a BST:>
116
117 Lemma [simp]: "bst (Node l (x:::linorder) r) ==> bst (join l r)"
118 oops
119
120 text <Implement a delete function using the idea contained in the lemmas above.>
121 fun delete :: "'a::linorder => 'a tree => 'a tree"
122 where

118.3 (3591/7019) (isabelle.isabelle.UTF-8-isabelle)tmr o UG 109/1104MB 1:38 PM
debian 1 2 3 4 lamlich@lapnikow10: ~/lehre/FDS/SS1... Isabelle2017 - tut03.thy 13:38:11
```

```
Isabelle2017 - tut03.thy
File Edit Search Markers Folding View Utilities Magros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
111 text <Show that join preserves the set of entries>
112 Lemma [simp]: "set_tree (join t1 t2) = set_tree t1 U set_tree t2"
113 by (simp add: set_inorder[symmetric])
114
115 text <Show that joining the left and right child of a BST is again a BST:>
116
117 Lemma [simp]: "bst (Node l (x:::linorder) r) ==> bst (join l r)"
118 apply (simp add: bst_iff_sorted_wrt_less del: join_inorder)
119
120 text <Implement a delete function using the idea contained in the lemmas above.>
121 fun delete :: "'a::linorder => 'a tree => 'a tree"
122 where

proof (prove)
goal (1 subgoal):
1. sorted_wrt op < (inorder l) ^
sorted_wrt op < (inorder r) ^ (∀xa∈set_tree l. xa < x) ^ (∀xa∈set_tree r. x < xa) ==>
sorted_wrt op < (inorder (join l r))

118.61 (3649/7074) (isabelle.isabelle.UTF-8-isabelle)tmr o UG 109/11084MB 1:39 PM
debian 1 2 3 4 lamlich@lapnikow10: ~/lehre/FDS/SS1... Isabelle2017 - tut03.thy 13:39:00
```

```
Isabelle2017 - tut03.thy
File Edit Search Markers Folding View Utilities Magros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
111 text <Show that join preserves the set of entries>
112 Lemma [simp]: "set_tree (join t1 t2) = set_tree t1 U set_tree t2"
113 by (simp add: set_inorder[symmetric])
114
115 text <Show that joining the left and right child of a BST is again a BST:>
116
117 Lemma [simp]: "bst (Node l (x:::linorder) r) ==> bst (join l r)"
118 apply (simp add: bst_iff_sorted_wrt_less del: join_inorder)
119
120 text <Implement a delete function using the idea contained in the lemmas above.>
121 fun delete :: "'a::linorder => 'a tree => 'a tree"
122 where

proof (prove)
goal (1 subgoal):
1. sorted_wrt op < (inorder l) ^
sorted_wrt op < (inorder r) ^ (∀xa∈set_tree l. xa < x) ^ (∀xa∈set_tree r. x < xa) ==>
sorted_wrt op < (inorder (join l r))

118.56 (3644/7074) (isabelle.isabelle.UTF-8-isabelle)tmr o UG 109/11084MB 1:40 PM
debian 1 2 3 4 lamlich@lapnikow10: ~/lehre/FDS/SS1... Isabelle2017 - tut03.thy 13:40:00
```


Isabelle2017 - tut03.thy

```

103 remove it once and forall.
104 >
105 declare set_inorder[simp del]
106
107
108 text "For the <sorted_wrt> predicate, you might want to use these lemmas as simp:"
109 thm sorted_wrt_append sorted_wrt_Cons
110
111 text <Show that join preserves the set of entries>
112 lemma [simp]: "set_tree (join t1 t2) = set_tree t1 ∪ set_tree t2"
113 by (simp add: set_inorder[symmetric])
114

```

proof (prove)
goal (1 subgoal):
1. bst (l, x, r) ⇒ bst (join l r)

106.1 (3166/7057) (isabelle,isabelle,UTF-8-Isabelle)nmr o UG 405.1084MB 1:40 PM

Isabelle2017 - tut03.thy

```

102 Alternatively, you can write <declare set_inorder[simp del]> to
103 remove it once and forall.
104
105 declare set_inorder[simp del]
106
107
108 text "For the <sorted_wrt> predicate, you might want to use these lemmas as simp:"
109 thm sorted_wrt_append sorted_wrt_Cons
110
111 text <Show that join preserves the set of entries>
112 lemma [simp]: "set_tree (join t1 t2) = set_tree t1 ∪ set_tree t2"
113 by (simp add: set_inorder[symmetric])

```

• transp ?P ⇒
sorted_wrt ?P (?xs @ ?ys) = (sorted_wrt ?P ?xs ∧ sorted_wrt ?P ?ys ∧ (∀x∈set ?xs. ∀y∈set ?ys. ?P x y))
• transp ?P ⇒ sorted_wrt ?P (?x # ?xs) = ((∀y∈set ?xs. ?P ?x y) ∧ sorted_wrt ?P ?xs)

109.5 (3255/7057) (isabelle,isabelle,UTF-8-Isabelle)nmr o UG 451.1084MB 1:40 PM

Isabelle2017 - tut03.thy

```

113 by (simp add: set_inorder[symmetric])
114
115 text <Show that joining the left and right child of a BST is again a BST:>
116
117 lemma [simp]: "bst (Node l (x:::linorder) r) ⇒ bst (join l r)"
118 apply (auto simp add: bst_iff_sorted_wrt_less sorted_wrt_append)
119
120 text <Implement a delete function using the idea contained in the lemmas above.>
121 fun delete :: "'a::linorder ⇒ 'a tree ⇒ 'a tree"
122 where
123 "delete _ _ = undefined"
124

```

proof (prove)
goal (1 subgoal):
1. ∧xa xaa.
[xa ∈ set (inorder l); xaa ∈ set (inorder r); sorted_wrt op < (inorder l);
sorted_wrt op < (inorder r); ∀xa∈set_tree l. xa < x; ∀xaa∈set_tree r. x < xaa]
⇒ xa < xaa

118.65 (3653/7079) (isabelle,isabelle,UTF-8-Isabelle)nmr o UG 500.1034MB 1:41 PM

Isabelle2017 - tut03.thy

```

115 text <Show that joining the left and right child of a BST is again a BST:>
116
117 lemma [simp]: "bst (Node l (x:::linorder) r) ⇒ bst (join l r)"
118 apply (auto simp add: bst_iff_sorted_wrt_less sorted_wrt_append set_inorder[symmetric])
119
120 text <Implement a delete function using the idea contained in the lemmas above.>
121 fun delete :: "'a::linorder ⇒ 'a tree ⇒ 'a tree"
122 where
123 "delete _ _ = undefined"
124
125 text <Prove it correct! Note: You'll need the first lemma to prove the second one!>
126 lemma [simp]: "bst t ⇒ set_tree (delete x t) = set_tree t - {x}"

```

proof (prove)
goal (1 subgoal):
1. ∧xa xaa.
[xa ∈ set (inorder l); xaa ∈ set (inorder r); sorted_wrt op < (inorder l);
sorted_wrt op < (inorder r); ∀xa∈set (inorder l). xa < x; ∀xaa∈set (inorder r). x < xaa]
⇒ xa < xaa

118.89 (3677/7102) (isabelle,isabelle,UTF-8-Isabelle)nmr o UG 521.065MB 1:43 PM

```

Isabelle2017 - tut03.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
115 text <Show that joining the left and right child of a BST is again a BST:>
116
117 Lemma [simp]: "bst (Node l (x:::linorder) r) ==> bst (join l r)"
118 apply (auto simp add: bst_iff_sorted_wrt_less sorted_wrt_append set_inorder[symmetric])
119
120 text <Implement a delete function using the idea contained in the lemmas above.>
121 fun delete :: "'a::linorder => 'a tree => 'a tree"
122 where
123   "delete _ _ = undefined"
124
125 text <Prove it correct! Note: You'll need the first lemma to prove the second one! >
126 Lemma [simp]: "bst t ==> set_tree (delete x t) = set_tree t - {x}"

proof (prove)
goal (1 subgoal):
1. ^\xa xaa.
   [xa ∈ set (inorder l); xaa ∈ set (inorder r); sorted_wrt op < (inorder l);
   sorted_wrt op < (inorder r); ∀xa∈set (inorder l). xa < x; ∀xa∈set (inorder r). x < xaa]
   ==> xa < xaa

```

```

Isabelle2017 - tut03.thy (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
115 text <Show that joining the left and right child of a BST is again a BST:>
116
117 Lemma [simp]: "bst (Node l (x:::linorder) r) ==> bst (join l r)"
118 apply (auto simp add: bst_iff_sorted_wrt_less sorted_wrt_append set_inorder[symmetric])
119 rly0
120
121 text <Implement a delete function using the idea contained in the lemmas above.>
122 fun delete :: "'a::linorder => 'a tree => 'a tree"
123 where
124   "delete _ _ = undefined"
125
126 text <Prove it correct! Note: You'll need the first lemma to prove the second one! >

Trying "simp", "auto", "blast", "metis", "argo", "linarith", "presburger", "algebra", "fast", "fastforce", "
Try this: by fastforce
(fastforce: 1 ms; force: 12 ms)

```

```

Isabelle2017 - tut03.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
115 text <Show that joining the left and right child of a BST is again a BST:>
116
117 Lemma [simp]: "bst (Node l (x:::linorder) r) ==> bst (join l r)"
118 apply (auto simp add: bst_iff_sorted_wrt_less sorted_wrt_append set_inorder[symmetric])
119
120 text <Implement a delete function using the idea contained in the lemmas above.>
121 fun delete :: "'a::linorder => 'a tree => 'a tree"
122 where
123   "delete _ _ = undefined"
124
125 text <Prove it correct! Note: You'll need the first lemma to prove the second one! >

proof (prove)
goal (1 subgoal):
1. ^\xa xaa.
   [xa ∈ set (inorder l); xaa ∈ set (inorder r); sorted_wrt op < (inorder l);
   sorted_wrt op < (inorder r); ∀xa∈set (inorder l). xa < x; ∀xa∈set (inorder r). x < xaa]
   ==> xa < xaa

```

```

Isabelle2017 - tut03.thy (modified)
File Edit Search Markers Folding View Utilities Macros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
115 text <Show that joining the left and right child of a BST is again a BST:>
116
117 Lemma [simp]: "bst (Node l (x:::linorder) r) ==> bst (join l r)"
118 apply (auto simp add: bst_iff_sorted_wrt_less sorted_wrt_append set_inorder[symmetric])
119 apply (keyword)
120
121 text <Implement a delete function using the idea contained in the lemmas above.>
122 fun delete :: "'a::linorder => 'a tree => 'a tree"
123 where
124   "delete _ _ = undefined"
125
126 text <Prove it correct! Note: You'll need the first lemma to prove the second one! >

proof (prove)
goal (1 subgoal):
1. ^\xa xaa.
   [xa ∈ set (inorder l); xaa ∈ set (inorder r); sorted_wrt op < (inorder l);
   sorted_wrt op < (inorder r); ∀xa∈set (inorder l). xa < x; ∀xa∈set (inorder r). x < xaa]
   ==> xa < xaa

```

```

Isabelle2017 - tut03.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
115 text <Show that joining the left and right child of a BST is again a BST:>
116
117 lemma [simp]: "bst (Node l (x::_:linorder) r) ==> bst (join l r)"
118   apply (auto simp add: bst_iff_sorted_wrt_less sorted_wrt_append set_inorder[symmetric])
119   apply fastforce
120   done
121
122 text <Implement a delete function using the idea contained in the lemmas above.>
123 fun delete :: "'a::linorder => 'a tree => 'a tree"
124 where
125   "delete _ _ = undefined"
126
theorem bst (?l, ?x, ?r) ==> bst (join ?l ?r)

```

122.1 (3705/7127) (isabelle,isabelle,UTF-8-isabelle)tmr o UG 51.6/065MB 1:45 PM

```

Isabelle2017 - tut03.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
113 by (simp add: set_inorder[symmetric])
114
115 text <Show that joining the left and right child of a BST is again a BST:>
116
117 lemma [simp]: "bst (Node l (x::_:linorder) r) ==> bst (join l r)"
118   apply (auto simp add: bst_iff_sorted_wrt_less sorted_wrt_append set_inorder[symmetric])
119   apply fastforce
120   done
121
122 text <Implement a delete function using the idea contained in the lemmas above.>
123 fun delete :: "'a::linorder => 'a tree => 'a tree"
124 where
125   "delete _ _ = undefined"
126
proof (prove)
goal (1 subgoal):
1. bst (l, x, r) ==> bst (join l r)

```

120.1 (3697/7127) (isabelle,isabelle,UTF-8-isabelle)tmr o UG 72/1046MB 1:47 PM

```

Isabelle2017 - tut03.thy
File Edit Search Markers Folding View Utilities Macros Plugins Help
tut03.thy (~/lehre/FDS/SS18/public/exercises/)
114 text <Show that joining the left and right child of a BST is again a BST:>
115
116 lemma [simp]: "bst (Node l (x::_:linorder) r) ==> bst (join l r)"
117   apply (auto simp add: bst_iff_sorted_wrt_less sorted_wrt_append set_inorder[symmetric])
118   apply fastforce
119   done
120
121 text <Implement a delete function using the idea contained in the lemmas above.>
122 fun delete :: "'a::linorder => 'a tree => 'a tree"
123 where
124   "delete _ _ = undefined"
125
theorem bst (?l, ?x, ?r) ==> bst (join ?l ?r)

```

121.1 (3704/7127) (isabelle,isabelle,UTF-8-isabelle)tmr o UG 03/1046MB 1:48 PM