

Propositional logic

Solution 1

(a)

$$\text{false}(\text{as}(\text{true} \Rightarrow \text{false}) \Leftrightarrow \text{false})$$

(b)

$$\text{true}(\text{as}(\text{false} \Rightarrow \text{false}) \Leftrightarrow \text{true})$$

(c)

$$\text{true}(\text{as}(\text{false} \Rightarrow \text{true}) \Leftrightarrow \text{true})$$

(d)

$$\text{true}(\text{as}(\text{false} \Rightarrow \text{false}) \Leftrightarrow \text{true})$$

(Assuming that pigs can't fly . . .)

Solution 2

(a)

p	q	$p \wedge q$	$(p \wedge q) \Rightarrow p$
t	t	t	t
t	f	f	t
f	t	f	t
f	f	f	t

(b)

p	q	$p \wedge q$	$\neg p$	$\neg p \Rightarrow (p \wedge q)$	$(\neg p \Rightarrow (p \wedge q)) \Leftrightarrow p$
t	t	t	f	t	t
t	f	f	f	t	t
f	t	f	t	f	t
f	f	f	t	f	t

(c)

p	q	$p \Rightarrow q$	$p \wedge (p \Rightarrow q)$	$(p \wedge (p \Rightarrow q)) \Rightarrow q$
t	t	t	t	t
t	f	f	f	t
f	t	t	f	t
f	f	t	f	t

Solution 3

(a)

$$\begin{aligned}
 p \Rightarrow \neg p & \\
 &\Leftrightarrow \neg p \vee \neg p && [\Rightarrow] \\
 &\Leftrightarrow \neg p && [\text{idempotence}]
 \end{aligned}$$

(b)

$$\begin{aligned}
 \neg p \Rightarrow p & \\
 &\Leftrightarrow \neg \neg p \vee p && [\Rightarrow] \\
 &\Leftrightarrow p \vee p && [\neg \neg] \\
 &\Leftrightarrow p && [\text{idempotence}]
 \end{aligned}$$

(c)

$$\begin{aligned}
 p \Rightarrow (q \Rightarrow r) & \\
 &\Leftrightarrow \neg p \vee (q \Rightarrow r) && [\Rightarrow] \\
 &\Leftrightarrow \neg p \vee \neg q \vee r && [\Rightarrow] \\
 &\Leftrightarrow \neg p \vee \neg q \vee r && [\text{associativity}] \\
 &\Leftrightarrow \neg (p \wedge q) \vee r && [\text{De Morgan}] \\
 &\Leftrightarrow p \wedge q \Rightarrow r && [\Rightarrow]
 \end{aligned}$$

(d)

$$\begin{aligned}
 q \Rightarrow (p \Rightarrow r) & \\
 &\Leftrightarrow \neg q \vee (p \Rightarrow r) && [\Rightarrow] \\
 &\Leftrightarrow \neg q \vee \neg p \vee r && [\Rightarrow] \\
 &\Leftrightarrow \neg p \vee \neg q \vee r && [\text{associativity} \wedge \text{commutativity}] \\
 &\Leftrightarrow \neg p \vee (q \Rightarrow r) && [\Rightarrow] \\
 &\Leftrightarrow p \Rightarrow (q \Rightarrow r) && [\Rightarrow]
 \end{aligned}$$

(e)

$$\begin{aligned}
p \wedge q &\Leftrightarrow p && [\Leftrightarrow] \\
&\Leftrightarrow (p \wedge q \Rightarrow p) \wedge (p \Rightarrow p \wedge q) && [\Rightarrow] \\
&\Leftrightarrow (\neg(p \wedge q) \vee p) \wedge (\neg p \vee p \wedge q) && [\text{De Morgan}] \\
&\Leftrightarrow (\neg p \vee \neg q \vee p) \wedge (\neg p \vee p \wedge q) && [\text{associativity } \wedge \text{ comm.}] \\
&\Leftrightarrow (\neg q \vee \neg p \vee p) \wedge (\neg p \vee p \wedge q) && [\text{excluded middle}] \\
&\Leftrightarrow (\neg q \vee \text{true}) \wedge (\neg p \vee p \wedge q) && [\vee \wedge \text{true}] \\
&\Leftrightarrow \text{true} \wedge (\neg p \vee p \wedge q) && [\wedge \wedge \text{true}] \\
&\Leftrightarrow \neg p \vee p \wedge q && [\text{distribution}] \\
&\Leftrightarrow (\neg p \vee p) \wedge (\neg p \vee q) && [\text{excluded middle}] \\
&\Leftrightarrow \text{true} \wedge (\neg p \vee q) && [\wedge \wedge \text{true}] \\
&\Leftrightarrow \neg p \vee q && [\Rightarrow] \\
&\Leftrightarrow p \Rightarrow q
\end{aligned}$$

(f)

$$\begin{aligned}
p \vee q &\Leftrightarrow p && [\Leftrightarrow] \\
&\Leftrightarrow (p \vee q \Rightarrow p) \wedge (p \Rightarrow p \vee q) && [\Rightarrow] \\
&\Leftrightarrow (\neg(p \vee q) \vee p) \wedge (\neg p \vee p \vee q) && [\text{De Morgan}] \\
&\Leftrightarrow (\neg p \wedge \neg q \vee p) \wedge (\neg p \vee p \vee q) && [\text{distribution}] \\
&\Leftrightarrow (\neg p \vee p) \wedge (\neg q \vee p) \wedge (\neg p \vee p \vee q) && [\text{excluded middle}] \\
&\Leftrightarrow \text{true} \wedge (\neg q \vee p) \wedge (\neg p \vee p \vee q) && [\wedge \wedge \text{true}] \\
&\Leftrightarrow (\neg q \vee p) \wedge (\neg p \vee p \vee q) && [\text{associativity}] \\
&\Leftrightarrow (\neg q \vee p) \wedge (\text{true} \vee q) && [\text{excluded middle}] \\
&\Leftrightarrow (\neg q \vee p) \wedge \text{true} && [\vee \wedge \text{true}] \\
&\Leftrightarrow \neg q \vee p && [\wedge \wedge \text{true}] \\
&\Leftrightarrow q \Rightarrow p
\end{aligned}$$

Solution 4

(a) $(p \text{ or } q) \Leftrightarrow ((\text{not } p \text{ or not } q) \text{ and } q)$ is not a tautology. You might illustrate this via a truth table or via a chain of equivalences, showing that the proposition is not equivalent to true. Alternatively, you might try and find a combination of values for which the proposition is false. (In this case, the proposition is false when p and q are both true.)

(b) $(p \text{ or } q) \Leftrightarrow ((\text{not } p \text{ and not } q) \text{ or } q)$ is not a tautology. In this case, the proposition is false when p is true and q is false.

Solution 5

(a)

$\exists d: Dog \bullet gentle(d) \wedge well_t rained(d)$

(b)

$\forall d: Dog \bullet neat(d) \wedge well_t rained(d) \Rightarrow attractive(d)$

(c)

(Requires nested quantifier in implication - parser limitation)

Solution 6

(a)

This is a true proposition: whatever the value of x, the expression $x^2 - x + 1$ denotes a natural number. If we choose y to be this natural number, we will find that p is true.

(b)

This is a false proposition. We cannot choose a large enough value for y such that p will hold for any value of x.

(c)

This is a false proposition. It is an implication whose antecedent part is true and whose consequent part is false.

(d)

This is a true proposition. It is an implication whose antecedent part is false and whose consequent part is true.

Solution 7

(a)

We must define a predicate p that is false for at least one value of x , and is true for at least one other value. A suitable solution would be $p \Leftrightarrow x \neq 1$.

(b)

With the above choice of p , we require only that q is sometimes false when p is true (for else the universal quantification would hold). A suitable solution would be $q \Leftrightarrow x \neq 3$.

Solution 8

(a)

$$\forall x: N \bullet x \geq z$$

Equality**Solution 9**

(d)

$$\begin{aligned} \exists x: N \bullet x = 1 \wedge x > y \vee x = 2 \wedge x > z \\ \Leftrightarrow \exists x: N \bullet x = 1 \wedge x > y \vee \exists x: N \bullet x = 2 \wedge x > z \\ \Leftrightarrow 1 \in N \wedge 1 > y \vee \exists x: N \bullet x = 2 \wedge x > z \\ \Leftrightarrow 1 \in N \wedge 1 > y \vee 2 \in N \wedge 2 > z \\ \Leftrightarrow 1 > y \vee 2 > z \end{aligned}$$

Solution 10

As discussed, the quantifier $\exists!$ can help give rise to a 'test' or 'precondition' to ensure that an application of μ will work.

So, as a simple example, as the proposition

$$\exists_1 n: N \bullet \forall m: N \bullet n \leq m$$

is equivalent to true, we can be certain that the statement

$$\mu n: N \bullet \forall m: N \bullet n \leq m$$

will return a result (which happens to be 0).

Solution 11

(a)

$(\mu a: N \bullet a = a) = 0$ is a provable statement, since 0 is the only natural number with the specified property.

(b)

$(\mu b: N \bullet b = b) = 1$ is not provable. The specified property is true of both 0 and 1, and thus the value of the mu-expression is undefined.

(c)

$(\mu c: N \bullet c > c) = (\mu c: N \bullet c > c)$ is a provable statement. Neither expression is properly defined, but we may conclude that they are equal; there is little else that we can prove about them.

(d)

$(\mu d: N \bullet d = d) = 1$ is not a provable statement. We cannot confirm that 1 is the only natural number with the specified property; we do not know what value is taken by undefined operations.

Solution 12

(Requires mu-operator with expression part - not yet implemented)

(a)

$$\mu m: Mountain \mid \forall n: Mountain \bullet height(n) \leq height(m) \bullet height(m)$$

$$\mu c: Chapter \mid \exists_1 d: Chapter \bullet length(d) > length(c) \bullet length(c)$$

Assuming the existence of a suitable function, $\max: (\mu n: N \bullet n = \max(\{m: N \mid 8 * m < 100.8 * m\}) . 100 - n)$

Deductive proofs

Solution 13

$$\frac{\frac{\frac{\vdash p \wedge (p \Rightarrow q) \top^{[1]}}{p \wedge (p \Rightarrow q)} [\wedge\text{-elim}^{[1]}] \quad \frac{\frac{\frac{p \wedge (p \Rightarrow q)}{p \Rightarrow q} [\wedge\text{-elim}^{[1]}] \quad \frac{p \wedge (p \Rightarrow q)}{p} [\wedge\text{-elim}^{[1]}]}{q} [\Rightarrow \text{elim}]}{p \wedge q} [\wedge \text{intro}]}{p \wedge (p \Rightarrow q) \Rightarrow p \wedge q} [\Rightarrow\text{-intro}^{[1]}]$$

Solution 14

In one direction:

$$\frac{\frac{\frac{\overline{p \wedge q} \text{ [derived]}}{\overline{p \wedge q} \text{ [\Rightarrow elim from } 1 \wedge 2\text{]}}}{\overline{p \wedge q} \text{ [\wedge-elim}^{[3]\text{]}}} \quad \frac{\frac{\frac{\frac{\frac{\vdash p \wedge q \Leftrightarrow p^{\neg[1]}}{(p \wedge q \Leftrightarrow p) \Rightarrow (p \Rightarrow q)} \text{ [\Rightarrow-intro}^{[11]\text{]}}}{p \Rightarrow q} \text{ [\Rightarrow-intro}^{[2]\text{]}}}{\frac{\vdash p^{\neg[2]}}{q} \text{ [\wedge-elim}^{[3]\text{]}}}}{\vdash p \wedge q \Leftrightarrow p^{\neg[1]}} \text{ [\Rightarrow elim from } 1 \wedge 2\text{]}$$

and the other:

$$\frac{\frac{\frac{\frac{\Gamma p \wedge q^{\neg[2]} \quad \Gamma p^{\neg[2]}}{p \wedge q \Rightarrow p} [\Rightarrow\text{-intro}^{[2]}] \quad \frac{\frac{\Gamma p^{\neg[3]} \quad \Gamma p \wedge q^{\neg[1]}}{p \Rightarrow p \wedge q} [\Rightarrow\text{-intro}^{[3]}]}{p \wedge q \Leftrightarrow p} [\Leftrightarrow\text{intro}]}{\Gamma p \Rightarrow q^{\neg[1]} \quad p \wedge q \Leftrightarrow p} [\Rightarrow\text{-intro}^{[1]}]}{(p \Rightarrow q) \Rightarrow (p \wedge q \Leftrightarrow p)}$$

We can then combine these two proofs with \Leftrightarrow intro.

Solution 15

[illegible]

Solution 16

In one direction:

[illegible]

In the other:

[illegible]

Solution 17

In one direction:

$$\frac{\frac{\ulcorner p \vee q \wedge r \urcorner^{[3]} \quad \overline{(p \vee q) \wedge (p \vee r)}}{p \vee q \wedge r \Rightarrow (p \vee q) \wedge (p \vee r)} [\vee \text{ elim } \wedge \wedge \text{ intro}]}{p \vee q \wedge r \Rightarrow (p \vee q) \wedge (p \vee r)} [\Rightarrow\text{-intro}^{[3]}]$$

and the other:

$$\frac{\ulcorner (p \vee q) \wedge (p \vee r) \urcorner^{[1]} \quad \ulcorner p \vee q \wedge r \urcorner^{[2]}}{(p \vee q) \wedge (p \vee r) \Rightarrow p \vee q \wedge r} [\Rightarrow\text{-intro}^{[1]}]$$

Solution 18

In one direction:

$$\frac{\ulcorner p \Rightarrow q \urcorner^{[1]} \quad \neg p \vee q}{(p \Rightarrow q) \Rightarrow \neg p \vee q} [\Rightarrow\text{-intro}^{[1]}]$$

and the other:

$$\frac{\frac{\frac{\neg p \vee q}{\neg p \vee q} \quad \frac{\frac{p \Rightarrow q}{p \Rightarrow q}}{\neg p \vee q \Rightarrow (p \Rightarrow q)} [\Rightarrow\text{-intro}^{[4]}] \quad \frac{\neg p \vee q \Rightarrow (p \Rightarrow q)}{\neg p \vee q \Rightarrow (p \Rightarrow q)} [\Rightarrow\text{-intro}^{[3]}]}{\neg p \vee q \Rightarrow (p \Rightarrow q)}$$

Sets and types

Solution 19

(a)

1 in 4, 3, 2, 1 is true.

(b)

1 in 1, 2, 3, 4 is undefined.

(c)

1 in 1, 2, 3, 4 is true.

(d)

The empty set in 1, 2, 3, 4 is undefined.

Solution 20

(a)

$\{1\} \times \{2, 3\}$

is the set (1, 2), (1, 3)

(b)

The empty set cross 2, 3 is the empty set

(c)

$\mathbb{P} \text{ } \textit{emptyset} \times \{1\}$

is the set (emptyset, 1)

(d)

$(1, 2)$ cross $3, 4$ is the set $((1, 2), 3), ((1, 2), 4)$

Solution 21

There are various ways of describing these sets via set comprehensions. Examples are given below.

(a)

$$\{z : Z \mid 0 \leq z \wedge z \leq 100\}$$

(b)

$$\{z : Z \mid z = 10\}$$

(c)

$$\{z : Z \mid z \bmod 2 = 0 \vee z \bmod 3 = 0 \vee z \bmod 5 = 0\}$$

Solution 22

(a)

$$\{n : N \mid n \leq 4 \bullet n^2\}$$

(b)

$$\{n : N \mid n \leq 4 \bullet (n, n^2)\}$$

(c)

$$n : P\ 0, 1$$

(d)

$$n : P\ 0, 1 \text{ — true} . (n, n)$$

Solution 23

(a)

$$\begin{aligned}
x \in a \cap a \\
&\Leftrightarrow x \in a \wedge x \in a \\
&\Leftrightarrow x \in a
\end{aligned}$$

(b)

$$\begin{aligned}
x \in a \cup a \\
&\Leftrightarrow x \in a \vee x \in a \\
&\Leftrightarrow x \in a
\end{aligned}$$

Solution 24

(a)

The set of all pairs of integers is \mathbb{Z} cross \mathbb{Z} . To give it a name, we could write:

$$\text{Pairs} == \mathbb{Z} \times \mathbb{Z}$$

(b)

The set of all integer pairs in which each element is strictly greater than zero could be defined by:

$$\text{StrictlyPositivePairs} == \{ m, n : \mathbb{Z} \mid m > 0 \wedge n > 0 \bullet (m, n) \}$$

(c)

It is intuitive to use a singular noun for the name of a basic type; we define the set of all people by writing:

$$[Person]$$

(d)

The set of all couples could be defined by:

$$\text{Couples} == \{ s : \mathbb{P} \text{ Person} \mid \#s = 2 \}$$

Solution 25

(Requires generic set notation and Cartesian product)

Solution 26

(Requires generic parameters and relation type notation)

Relations**Solution 27**

(a)

The power set of $(0,0)$, $(0,1)$, $(1,0)$, $(1,1)$ is:

$\{\emptyset, \{(0,0)\}, \{(0,1)\}, \{(1,0)\}, \{(1,1)\}, \{(1,0), (1,1)\}, \{(0,0), (0,1)\}, \{(0,1), (1,1)\}, \{(0,1), (1,0)\}, \{(0,$

(b)

$\emptyset, \{(0,0)\}, \{(0,1)\}, \{(0,0), (0,1)\}\}$

(c)

$\{\emptyset\}$

(d)

$\{\emptyset\}$

Solution 28

(a)

$\text{dom } R = \{0, 1, 2\}$

(b)

$\text{ran } R = \{1, 2, 3\}$

(c)

$\{1, 2\} \triangleleft R = \{1 \mapsto 2, 1 \mapsto 3, 2 \mapsto 3\}$

Solution 29

(a)

$$\{2 \mapsto 4, 3 \mapsto 3, 3 \mapsto 4, 4 \mapsto 2\}$$

(b)

$$\{1 \mapsto 3, 2 \mapsto 2, 2 \mapsto 3, 3 \mapsto 1\}$$

(c)

$$\{1 \mapsto 1, 2 \mapsto 2, 2 \mapsto 3, 3 \mapsto 2, 3 \mapsto 3, 4 \mapsto 4\}$$

(d)

$$\{1 \mapsto 4, 2 \mapsto 2, 2 \mapsto 3, 3 \mapsto 2, 3 \mapsto 3, 4 \mapsto 1\}$$

Solution 30

$$\mid \quad \textit{childOf} : \textit{Person} \leftrightarrow \textit{Person}$$

(a)

$$\textit{parentOf} == \textit{childOf}^{-1}$$

This is a good example of how there are many different ways of writing the same thing. An alternative abbreviation is:

$$\textit{parentOf} == \{ x, y : \textit{Person} \mid x \mapsto y \in \textit{childOf} \bullet y \mapsto x \}$$

Or, via an axiomatic definition:

$$\frac{\mid \quad \textit{parentOf} : \textit{Person} \leftrightarrow \textit{Person}}{\mid \quad \textit{parentOf} = \textit{childOf}^{-1}}$$

(b)

$$\textit{siblingOf} == (\textit{childOf} \circ \textit{parentOf}) \setminus \textit{id}$$

(c)

$$\textit{cousinOf} == \textit{childOf} \circ \textit{siblingOf} \circ \textit{parentOf}$$

(d)

$$\textit{ancestorOf} == \textit{parentOf}^+$$

Solution 31

(Requires compound identifiers with operators - $R+$, R^*)

(a)

$$R == \{ a, b : N \mid b = a \vee b = a \}$$

(b)

$$S == \{ a, b : N \mid b = a \vee b = a \}$$

(c)

$$R+ == \{ a, b : N \mid b > a \}$$

(d)

$$R^* == \{ a, b : N \mid b \geq a \}$$

Solution 32

(a)

$$x \mapsto y \in A \triangleleft B \triangleleft R$$

$$\Leftrightarrow x \in A \wedge x \mapsto y \in (B \triangleleft R)$$

$$\Leftrightarrow x \in A \wedge x \in B \wedge x \mapsto y \in R$$

$$\Leftrightarrow x \in A \cap B \wedge x \mapsto y \in R$$

$$\Leftrightarrow x \mapsto y \in A \cap B \triangleleft R$$

(b)

$$x \mapsto y \in R \cup S \triangleright C$$

$$\Leftrightarrow x \mapsto y \in R \cup S \wedge y \in C$$

$$\Leftrightarrow (x \mapsto y \in R \vee x \mapsto y \in S) \wedge y \in C$$

$$\Leftrightarrow x \mapsto y \in R \wedge y \in C \vee x \mapsto y \in S \wedge y \in C$$

$$\Leftrightarrow x \mapsto y \in R \triangleright C \vee x \mapsto y \in S \triangleright C$$

$$\Leftrightarrow x \mapsto y \in (R \triangleright C) \cup (S \triangleright C)$$

Functions

Solution 33

The set of 9 functions:

$$\{\emptysetset, \{(0, 0)\}, \{(0, 1)\}, \{(1, 1)\}, \{(1, 0)\}, \{(0, 0), (1, 1)\}, \{(0, 1), (1, 1)\}, \{(1, 0), (0, 0)\}, \{(0, 1), (1, 0)\}\}$$

(a)

The set of total functions:

$$\{\{(0, 0), (1, 1)\}, \{(0, 1), (1, 1)\}, \{(1, 0), (0, 0)\}, \{(0, 1), (1, 0)\}\}$$

(b)

The set of functions which are neither injective nor surjective:

$$\{\{(0, 1), (1, 1)\}, \{(0, 0), (1, 0)\}\}$$

(c)

The set of functions which are injective but not surjective:

$$\{\emptysetset, \{(0, 0)\}, \{(0, 1)\}, \{(1, 0)\}, \{(1, 1)\}\}$$

(d)

There are no functions (of this type) which are surjective but not injective.

(e)

The set of bijective functions:

$$\{\{(0, 0), (1, 1)\}, \{(0, 1), (1, 0)\}\}$$

Solution 34

(a)

$$\{1 \mapsto a, 2 \mapsto b, 3 \mapsto c, 4 \mapsto b\}$$

(b)

$\{1 \mapsto c, 2 \mapsto b, 3 \mapsto c, 4 \mapsto d\}$

(c)

$\{1 \mapsto c, 2 \mapsto b, 3 \mapsto c, 4 \mapsto b\}$

(d)

$\{1 \mapsto c, 2 \mapsto b, 3 \mapsto c, 4 \mapsto b\}$

Solution 35

(Requires power set notation P and relational image)

(a)

axdef

children : Person \rightarrow P Person

where

children = $p : \text{Person} \rightarrow p \rightarrow \text{parentOf}(\text{--- } p \text{ ---})$

end

(b)

axdef

number_of_grandchildren : Person \rightarrow N

where

number_of_grandchildren = $p : \text{Person} \rightarrow p \rightarrow (\text{parentOf} \circ \text{parentOf})(| p |)$

end

Solution 36

(Requires power set, function types, and ran keyword)

axdef

$\text{number}_{of_drivers} : (Drivers \rightarrow Cars) \rightarrow (Cars \rightarrow N)$

where

forall $r : Drivers \rightarrow Cars$ — $\text{number}_{of_drivers}(r) = c : \text{ran } r.c \mapsto \{ d : Drivers \mid d \mapsto c \in r \}$

end

Sequences

Solution 37

(a)

$\langle a \rangle$

(b)

$\{1 \mapsto a, 2 \mapsto b, 2 \mapsto a, 3 \mapsto c, 3 \mapsto b, 4 \mapsto d\}$

(c)

$\{2 \mapsto b, 3 \mapsto c, 4 \mapsto d\}$

(d)

$\{1, 2, 3, 4\}$

(e)

$\{a, b\}$

(f)

$\{a \mapsto 1, b \mapsto 2, c \mapsto 3, d \mapsto 4\}$

(g)

$\langle a, b \rangle$

(h)

$\{3 \mapsto b\}$

(i)

$\{a\}$

(j)

c

Solution 38

(a)

$$\frac{\left| \begin{array}{l} f : Place \rightarrow \mathbb{P} Place \end{array} \right.}{\forall p : Place \bullet f(p) = \{q : Place \mid p \mapsto q \in \text{ran } trains\}}$$

(b)

$\{p : Place \mid \exists_1 x : \text{dom } trains \bullet trains(x).2 = p\}$

(c)

$(\mu p : Place \mid \forall q : Place \bullet p \neq q \mid \{x : \text{dom } trains \mid trains(x).2 = p\} \mid \{x : \text{dom } trains \mid trains(x).2 = q\})$

(Blocked by: nested quantifiers in mu with multiple pipes - parser ambiguity)

Solution 39

(a)

$\text{large}_{c \text{ oins}} : Collection \rightarrow N$

$\forall c : Collection \bullet \text{large}_{c \text{ oins}}(c) = c(\text{large})$

(Blocked by: underscore in identifier for fuzz compatibility)

(b)

$\text{add}_{\text{coin}} : \text{Collection} * \text{Coin} \rightarrow \text{Collection}$

$\forall c : \text{Collection} \bullet \forall d : \text{Coin} \bullet \text{add}_{\text{coin}}(c, d) = c \cup \llbracket d \rrbracket$

(Blocked by: underscore in identifier and bag union)

Modelling

Solutions 40-52 are work in progress - many require features not yet implemented

Solution 40

(Work in progress - requires semicolon-separated bindings in set comprehensions)

(a)

$\text{hd} : \text{seq}(\text{Title} * \text{Length} * \text{Viewed})$

$\text{cumulative}_{\text{total}} \text{hd} \leq 12000$

$\forall p : \text{ran } \text{hd} \bullet p.2 \leq 360$

Note that $\text{cumulative}_{\text{total}} \text{hd} \text{ is defined in part } (d)$.

(b)

$\{p : \text{ran } \text{hd} \mid p.2 > 120 \bullet p.1\}$

(c)

These can be defined recursively:

viewed $\text{!}i = \text{!}i$

viewed $\text{!}x \text{!}^s = \text{if } x.3 = \text{yes then } \langle x \rangle^v \text{iewed else viewed } s$

or otherwise (omitted - requires semicolon-separated bindings in set comprehension)

(d)

$$\frac{\text{cumulative}_{total} : \text{seq } Title * Length * Viewed \rightarrow N}{\text{cumulative}_{total}(\langle \rangle) = 0 \forall x : Title * Length * Viewed \bullet \forall s : \text{seq } Title * Length * Viewed \bullet \text{cumulative}_{total}(x \bullet s) = \text{cumulative}_{total}(x) + \text{length}(s)}$$

(e)

(mu p : ran hd — $\forall q : \text{ran } hd \bullet p \neq q \rightarrow p.2 \text{!} q.2 \rightarrow p.1$)

(This, of course, assumes that there is a unique element with this property.)

(f)

(f) Omitted - requires semicolon-separated bindings in nested set comprehension

(g)

axdef

$g : \text{seq}(Title * Length * Viewed) \rightarrow \text{seq}(Title * Length * Viewed)$

where

$\forall s : \text{seq } Title * Length * Viewed \bullet g(s) = s \text{!} \{x : \text{ran } s \mid x \neq \text{longest}_{viewed}(s)\}$

end

Where $\text{longest}_{\text{viewed}}$ is defined as

axdef

$\text{longest}_{\text{viewed}} : \text{seq}(\text{Title} * \text{Length} * \text{Viewed}) \rightarrow \text{Title} * \text{Length} * \text{Viewed}$

where

$\forall s : \text{seq} \text{ Title} * \text{Length} * \text{Viewed} \bullet \text{longest}_{\text{viewed}}(s) = (\mu p : \text{ran } s \bullet p.3 = \text{yes} \text{ and } \forall q : \text{ran } s \bullet p \neq q \wedge q.3 = \text{yes} \rightarrow p.2 \leq q.2)$

end

(Blocked by: nested quantifiers in mu expressions - parser limitation)

This, of course, assumes that there is at least one viewed programme (and one of a unique maximum length).

(h)

$$\frac{s : \text{seq} \text{ Title} * \text{Length} * \text{Viewed} \rightarrow \text{seq} \text{ Title} * \text{Length} * \text{Viewed}}{\forall x : \text{seq} \text{ Title} * \text{Length} * \text{Viewed} \bullet \text{items}(s(x)) = \text{items}(x) \wedge \forall i, j : \text{dom } s(x) \bullet i < j \Rightarrow s(x)(i).2 \geq s(x)(j).2}$$

Solution 41

(a)

axdef

records : Year \multimap Table

where

dom(records) = 1993..current

$\forall y: \text{dom } records \bullet \#records(y) \leq 50$

forall y : dom(records) $\multimap \forall e: \text{ran } records(y) \bullet year(e.1) = y$

forall r : ran(records) $\multimap \forall i1, i2: \text{dom } r \bullet i1 \neq i2 \wedge r(i1).1 = r(i2).1 \Rightarrow r(i1).3 \neq r(i2).3$

end

(Blocked by: nested quantifiers in predicates - parser limitation)

(b)

(i)

$\{e: Entry \mid \exists r: \text{ran } records \bullet e \in \text{ran } r \wedge e.3 = 479\}$

ii

$\{e: Entry \mid \exists r: \text{ran } records \bullet e \in \text{ran } r \wedge e.6 > e.5\}$

iii

$\{e: Entry \mid \exists r: \text{ran } records \bullet e \in \text{ran } r \wedge e.7 \geq 70\}$

iv

$\{c: Course \mid \forall r: \text{ran } records \bullet \forall e: \text{ran } r \bullet e.2 = c \Rightarrow e.7 \geq 70\}$

v

y : Year \multimap y in dom records . y \multimap l : Lecturer \multimap c : ran (records y) \multimap
c.4 = l 6

(c)

axdef

where

$$\forall x: Entry \bullet \forall s: seq\ Entry \bullet 479_{courses}(<>) = <> \text{ and } 479_{courses}(< x >^s) = if x.3 = 479 \text{ then } < x >^4 79_{courses}(s) \text{ else } 479_{courses}(s)$$

end

(Blocked by: underscore in identifier - use camelCase for fuzz compatibility)

(d)

$$\left| \forall x: Entry \bullet \forall s: seq\ Entry \bullet total(\langle \rangle) = 0 \wedge total(\langle x \rangle \frown s) = x.5 + total(s) \right|$$

Solution 42

[Person]

axdef

State : P(seq(iseq(Person)))

where

forall s : State — $\forall i, j: \text{dom } s \bullet i \neq j \text{ — } \text{ran}(s(i)) \text{ intersect } \text{ran}(s(j)) =$

end

(Blocked by: nested quantifiers with semicolon bindings - parser limitation)

(b)

axdef

add : N * Person * State \longrightarrow State

where

$\forall n : N \bullet \forall p : Person \bullet \forall s : State \bullet n \in \text{dom } s \wedge p \notin \text{bigcup}(\text{ran } \text{ran } s) \longrightarrow$

add(n, p, s) = s ++ n \longrightarrow s(n) \leq_p >

end

(Blocked by: \longrightarrow operator not implemented)

Solution 43

(a)

(i) forall i : dom bookings $\longrightarrow \forall x, y : \text{bookings}(i) \bullet x \neq y \longrightarrow (\text{x.2}..\text{x.3}) \text{ intersect } (\text{y.2}..\text{y.3}) =$

(ii) forall i : dom bookings $\longrightarrow \forall x : \text{bookings}(i) \bullet \{x.2, x.3\} \text{subsetq } 1..\text{max}(i.1)$

(iii) forall i : dom bookings $\longrightarrow \forall b : \text{bookings}(i) \bullet b.2 \leq b.3$

(iv) This is enforced by the constraint for part (i).

(Blocked by: nested quantifiers - parser limitation)

(b)

(i) $\{i : \text{dom } bookings \mid i.1 = Banbury \bullet i.2\}$

(ii) $i : \text{dom } bookings \multimap i.1 = Banbury \text{ and } \exists b : bookings(i) \bullet 50 \in b.2 \dots b.3$

(iii) $r : \text{Room}; s : \mathbb{N} \multimap \exists i : \text{dom } bookings \bullet i.1 = r \wedge i.2 = s. (r, s)$

(iv) $r : \text{Room} \multimap \exists i : \text{dom } bookings \bullet i.1 = r \multimap (\text{bookings}(i))_i = 10$

(Blocked by: semicolon bindings in set comprehensions and nested quantifiers)

Free types and induction

Solution 44

The two cases of the proof are established by equational reasoning: the first by

$$\text{reverse } (j_i^t) = \text{reverset}[cat.1a] = (\text{reverset})^< > [cat.1b]$$

$$\text{where } cat.1a \text{ is } j_i^s = sandcat.1biss^< > = s$$

and the second by

$$\text{reverse} ((\text{ixl}^u)^t) = \text{reverse}(< x >^{\text{u}^t})[\text{cat.2}]$$

$$= \text{reverse} (\text{u}^t)^{< x >} [\text{reverse.2}]$$

$$= (\text{reverse t}^{\text{reverse u}})^{< x >} [\text{anti} - \text{distributive}]$$

$$= \text{reverse t}^{\text{reverse u}^{< x >}} [\text{cat.2}]$$

$$= \text{reverse t}^{\text{reverse}(< x >^{\text{u}})} [\text{reverse.2}]$$

Solution 45

The base case:

$$\text{reverse} (\text{reverse iil}) = \text{reverse iil} [\text{reverse.1}] = \text{iil} [\text{reverse.1}]$$

The inductive step:

$$\text{reverse} (\text{reverse} (\text{ixl}^t))$$

$$= \text{reverse} ((\text{reverse t})^{< x >}) [\text{reverse.2}]$$

$$= \text{reverse} (\text{ixl}^{\text{reverse t}})^{\text{reverse}(\text{reverset})} [\text{anti} - \text{distributive}]$$

$$= \text{reverse} (\text{ixl}^{< x >})^{\text{reverse}(\text{reverset})} [\text{cat.1}]$$

$$= ((\text{reverse iil})^{< x >})^{\text{reverse}(\text{reverset})} [\text{reverse.2}]$$

$$= (i \cdot x)^r \text{reverse}(\text{reverset})[\text{reverse}.1]$$

$$= i \cdot x^r \text{reverse}(\text{reverset})[\text{cat}.1]$$

$$= i \cdot x^t [\text{reverse}(\text{reverset}) = t]$$

Solution 46

(a)

$$\text{count} : \text{Tree} \rightarrow \mathbb{N}$$

$$\text{count stalk} = 0$$

$$\forall n : N \bullet \text{count}(\text{leaf } n) = 1$$

$$\forall t1, t2 : \text{Tree} \bullet \text{count}(\text{branch}(t1, t2)) = \text{count } t1 + \text{count } t2$$

(Blocked by: recursive free types and pattern matching)

(b)

$$\text{flatten} : \text{Tree} \rightarrow \text{seq } \mathbb{N}$$

$$\text{flatten stalk} = i$$

$$\forall n : N \bullet \text{flatten}(\text{leaf } n) = i \cdot n$$

$$\forall t1, t2 : \text{Tree} \bullet \text{flatten}(\text{branch}(t1, t2)) = \text{flatten } t1 \mathbin{\cdot} \text{flatten } t2$$

(Blocked by: recursive free types and pattern matching)

Solution 47

First, exhibit the induction principle for the free type:

$\mathbb{P} \text{ stalk}$ and $(\forall n: N \bullet P(\text{leaf } n))$ and $(\forall t1, t2: Tree \bullet \mathbb{P} t1 \wedge \mathbb{P} t2 \Rightarrow \mathbb{P} \text{branch}(t1, t2))$

implies $\forall t: Tree \bullet \mathbb{P} t$

This gives three cases for the proof:

$(\text{flatten stalk}) = \text{count stalk} = 0 = \text{count stalk} [\text{count}]$

(Remaining cases omitted - require equational reasoning with recursive functions)

Supplementary material : assignment practice

Solution 48

$\text{songs} : F \text{ SongId}$

$\text{users} : F \text{ UserId}$

$\text{playlists} : \text{PlaylistId} \rightarrow F \text{ Playlist}$

$\text{playlistOwner} : \text{PlaylistId} \rightarrow F \text{ UserId}$

$\text{playlistSubscribers} : \text{PlaylistId} \rightarrow F1 \text{ UserId}$

$\forall i : \text{dom } \textit{playlists} \bullet \text{ran } \textit{playlists}(i) \text{subseq songs}$

$\text{dom}(\textit{playlistOwner}) \text{subseq dom}(\textit{playlists})$

$\text{ran}(\textit{playlistOwner}) \text{subseq users}$

$\text{dom}(\textit{playlistSubscribers}) \text{subseq dom}(\textit{playlists})$

$\forall i : \text{dom } \textit{playlistSubscribers} \bullet \textit{playlistSubscribers}(i) \text{subseq users}$

$\forall i : \text{dom } \textit{playlists} \bullet \textit{playlistOwner}(i) \in \textit{playlistSubscribers}(i)$

(Blocked by: F and F1 finite set types not implemented)

Solution 49

$\text{hated} : \text{UserId} \rightarrow \mathcal{F} \text{ SongId}$

$\text{loved} : \text{UserId} \rightarrow \mathcal{F} \text{ SongId}$

$\text{dom}(\text{hated}) \text{subseq users}$

$\forall i : \text{dom } \text{hated} \bullet \text{hated}(i) \text{subseq songs}$

$\text{dom}(\text{loved}) \text{subseq users}$

$\forall i : \text{dom } \text{loved} \bullet \text{loved}(i) \text{subseq songs}$

forall i : $\text{dom}(\text{hated}) \cup \text{dom}(\text{loved}) \rightarrow \text{hated}(i) \cap \text{loved}(i) = \text{emptyset}$

(Blocked by: F finite set type not implemented)

Solution 50

(a)

$A == \text{users} \text{ bigcup } (\text{ran}(\text{playlistSubscribers}))$

(Blocked by: bigcup operator not implemented)

(b)

$B == \{p : \text{dom } \text{playlistSubscribers} \mid \# \text{playlistSubscribers}(p) \geq 100\}$

(c)

$C == (\mu u : \text{dom}(\text{loved}) \multimap \forall v : \text{dom } \text{loved} \bullet u \neq v \multimap (\text{loved}(u)) \wedge (\text{loved}(v)))$

(Blocked by: nested quantifiers in mu - parser ambiguity)

(d)

$D == (\mu s : \text{songs} \multimap \forall t : \text{songs} \bullet s \neq t \multimap \{u : \text{UserId} \mid s \in \text{loved}(u)\} \wedge \{u : \text{UserId} \mid t \in \text{loved}(u)\})$

(Blocked by: nested quantifiers in mu - parser ambiguity)

Solution 51

(a)

Let's first define two helper functions:

$\text{loveHateScore} : \text{SongId} \rightarrow \mathbb{N}$

forall $i : \text{songs} \multimap \{u : \text{UserId} \mid i \in \text{loved}(u)\} \not\subseteq \{u : \text{UserId} \mid i \in \text{hated}(u)\}$
 \Rightarrow

$\text{loveHateScore}(i) = \{u : \text{UserId} \mid i \in \text{loved}(u)\} - \{u : \text{UserId} \mid i \in \text{hated}(u)\}$

and

forall $i : \text{songs} \multimap \{u : \text{UserId} \mid i \in \text{loved}(u)\} \cap \{u : \text{UserId} \mid i \in \text{hated}(u)\}$
 \Rightarrow

$\text{loveHateScore}(i) = 0$

$$\frac{\text{playlistCount} : \text{SongId} \rightarrow N}{\forall i : \text{songs} \bullet \text{playlistCount}(i) = \#\{p : \text{dom playlist} \mid i \in \text{ran playlist}(p)\}}$$

We then have:

$$\frac{\text{length} : \text{SongId} \rightarrow N \text{ popularity} : \text{SongId} \rightarrow N}{\text{dom length} \subseteq \text{songs} \text{ dom popularity} \subseteq \text{songs} \forall i : \text{songs} \bullet \text{popularity}(i) = \text{loveHateScore}(i) + p}$$

(b)

$\text{mostPopular} : \text{SongId}$

$(\text{exists1 } i : \text{songs} \multimap \forall j : \text{songs} \bullet i \neq j \multimap \text{popularity}(i) \not\subseteq \text{popularity}(j)) \Rightarrow$

$\text{mostPopular} = (\mu i : \text{songs} \multimap \forall j : \text{songs} \bullet i \neq j \multimap \text{popularity}(i) \not\subseteq \text{popularity}(j))$

and

$\text{not } (\text{exists1 } i : \text{songs} \multimap \forall j : \text{songs} \bullet i \neq j \multimap \text{popularity}(i) \not\subseteq \text{popularity}(j))$
 \Rightarrow

mostPopular = nullSong

(Blocked by: nested quantifiers in mu - parser ambiguity)

(c)

playlistsContainingMostPopularSong == $\{i : \text{dom } playlists \mid \text{mostPopular} \in \text{ran } playlists(i)\}$

Solution 52

(a)

premiumPlays : seq(Play) \rightarrow seq(Play)

premiumPlays(i) = i

forall x : Play; s : seq(Play) —

premiumPlays(i s) = $\langle x \rangle^p \text{ premiumPlays}(s) \text{ if } \text{userStatus}(x.2) = \text{premium}$

premiumPlays(s) if $\text{userStatus}(x.2) = \text{standard}$

(Note: Uses camelCase for fuzz compatibility)

(b)

standardPlays : seq(Play) \rightarrow seq(Play)

standardPlays(i) = i

forall x : Play; s : seq(Play) —

$\text{standardPlays}(x.s) = \langle x \rangle^s \text{standardPlays}(s) \text{ if } \text{userStatus}(x.2) = \text{standard}$

$\text{standardPlays}(s) \text{ if } \text{userStatus}(x.2) = \text{premium}$

(Note: Uses camelCase for fuzz compatibility)

(c)

$\text{cumulativeLength} : \text{seq}(\text{Play}) \rightarrow \mathbb{N}$

$\text{cumulativeLength}(\langle \rangle) = 0$

forall $x : \text{Play}; s : \text{seq}(\text{Play})$ —

$\text{cumulativeLength}(x.s) = \text{length}(x.1) + \text{cumulativeLength}(s)$

(Note: Uses camelCase for fuzz compatibility)