Propositional logic

Solution 1

(a)

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

(b)

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

(c)

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

(d)

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

(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)

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

(b)

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

(c)

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

(d)

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

(e)

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

(f)

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

Solution 4

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

Solution 5

(a)

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

(b)

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

(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 ¿ 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 ¿ 3.

Solution 8

(a)

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

Equality

Solution 9

(d)

$$
\begin{aligned}
&\exists x \colon N \bullet x = 1 \land x > y \lor x = 2 \land x > z \\
&\quad \Leftrightarrow \exists x \colon N \bullet x = 1 \land x > y \lor \exists x \colon N \bullet x = 2 \land x > z \\
&\quad \Leftrightarrow 1 \in N \land 1 > y \lor \exists x \colon N \bullet x = 2 \land x > z \\
&\quad \Leftrightarrow 1 \in N \land 1 > y \lor 2 \in N \land 2 > z \\
&\quad \Leftrightarrow 1 > y \lor 2 > z
\end{aligned}
$$

Solution 10

As discussed, the quantifier exists1 can help give rise to a 'test' or 'precondition' to ensure that an application of mu 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)

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

(b)

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

(c)

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

Deductive proofs

Solution 13

$$\cfrac{\ulcorner p \wedge (p \Rightarrow q) \urcorner^{[1]} \qquad \cfrac{\cfrac{p \wedge (p \Rightarrow q)}{p}\,[\wedge\text{-elim}^{[1]}] \qquad \cfrac{\cfrac{p \wedge (p \Rightarrow q)}{p \Rightarrow q}\,[\wedge\text{-elim}^{[1]}] \quad \cfrac{p \wedge (p \Rightarrow q)}{p}\,[\wedge\text{-elim}^{[1]}]}{q}\,[\Rightarrow \text{elim}]}{p \wedge q}\,[\wedge \text{ intro}]}{p \wedge (p \Rightarrow q) \Rightarrow p \wedge q}\,[\Rightarrow\text{-intro}^{[1]}]$$

Solution 14

In one direction:

$$\cfrac{\ulcorner p \wedge q \Leftrightarrow p \urcorner^{[1]} \qquad \cfrac{\ulcorner p \urcorner^{[2]} \qquad \cfrac{\cfrac{\cfrac{}{p \wedge q}\,[\text{derived}]}{p \wedge q}\,[\Rightarrow \text{elim from } 1 \wedge 2]}{q}\,[\wedge\text{-elim}^{[3]}]}{p \Rightarrow q}\,[\Rightarrow\text{-intro}^{[2]}]}{(p \wedge q \Leftrightarrow p) \Rightarrow (p \Rightarrow q)}\,[\Rightarrow\text{-intro}^{[1]}]$$

and the other:

$$\cfrac{\ulcorner p \Rightarrow q \urcorner^{[1]} \qquad \cfrac{\cfrac{\ulcorner p \wedge q \urcorner^{[2]} \quad \ulcorner p \urcorner^{[2]}}{p \wedge q \Rightarrow p}\,[\Rightarrow\text{-intro}^{[2]}] \qquad \cfrac{\ulcorner p \urcorner^{[3]} \quad \ulcorner p \wedge q \urcorner^{[1]}}{p \Rightarrow p \wedge q}\,[\Rightarrow\text{-intro}^{[3]}]}{p \wedge q \Leftrightarrow p}\,[\Leftrightarrow \text{ intro}]}{(p \Rightarrow q) \Rightarrow (p \wedge q \Leftrightarrow p)}\,[\Rightarrow\text{-intro}^{[1]}]$$

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

Solution 15

$$\dfrac{\ulcorner (p \Rightarrow q) \wedge \neg\, q \urcorner^{[1]} \qquad \dfrac{\ulcorner p \urcorner^{[2]} \qquad \dfrac{\dfrac{\ulcorner p \Rightarrow q \urcorner^{[1]} \quad \ulcorner p \urcorner^{[2]}}{q}\,[\Rightarrow \text{ elim}] \qquad \ulcorner \neg\, q \urcorner^{[1]}}{\mathit{false}}\,[\text{false intro}]}{\neg\, p}\,[\text{false-elim}^{[2]}]}{(p \Rightarrow q) \wedge \neg\, q \Rightarrow \neg\, p}\,[\Rightarrow\text{-intro}^{[1]}]$$

Solution 16

In one direction:

$$\dfrac{\ulcorner p \wedge (q \vee r) \urcorner^{[1]} \qquad \dfrac{\ulcorner q \vee r \urcorner^{[1]} \qquad \dfrac{\dfrac{\ulcorner p \urcorner^{[1]} \quad \dfrac{}{q}\,[\text{case assumption}]}{p \wedge q}\,[\wedge \text{ intro}]}{p \wedge q \vee p \wedge r}\,[\vee \text{ intro}] \qquad \dfrac{\dfrac{\ulcorner p \urcorner^{[1]} \quad \dfrac{}{r}\,[\text{case assumption}]}{p \wedge r}\,[\wedge \text{ intro}]}{p \wedge q \vee p \wedge r}\,[\vee \text{ intro}]}{p \wedge q \vee p \wedge r}\,[\vee\text{-elim}^{[2]}]}{p \wedge (q \vee r) \Rightarrow p \wedge q \vee p \wedge r}\,[\Rightarrow\text{-intro}^{[1]}]$$

In the other:

$$\dfrac{\ulcorner p \wedge q \vee p \wedge r \urcorner^{[3]} \qquad \dfrac{\ulcorner case1 \vee case2 \urcorner^{[3]} \qquad \begin{array}{c} \dfrac{\dfrac{\dfrac{}{p}\ [\wedge \text{ elim}]}{q \vee r}\ [\vee \text{ intro}]}{p \wedge (q \vee r)}\ [\wedge \text{ intro}] \\ \Big| \end{array} \qquad \begin{array}{c} \dfrac{\dfrac{\dfrac{}{p}\ [\wedge \text{ elim}]}{q \vee r}\ [\vee \text{ intro}]}{p \wedge (q \vee r)}\ [\wedge \text{ intro}] \\ \Big| \end{array}}{p \wedge (q \vee r)}\ [\vee\text{-elim}^{[4]}]}{p \wedge q \vee p \wedge r \Rightarrow p \wedge (q \vee r)}\ [\Rightarrow\text{-intro}^{[3]}]$$

Solution 17

In one direction:

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

and the other:

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

Solution 18

In one direction:

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

and the other:

$$\dfrac{\ulcorner \neg\, p \vee q \urcorner^{[3]} \qquad \dfrac{\ulcorner p \urcorner^{[4]} \qquad \ulcorner q \urcorner^{[3]}}{p \Rightarrow q}\ [\Rightarrow\text{-intro}^{[4]}]}{\neg\, p \vee q \Rightarrow (p \Rightarrow q)}\ [\Rightarrow\text{-intro}^{[3]}]$$

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)

$\mathbb{P}\ \mathit{emptyset} \times \{1\}$

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)

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

(b)

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

(c)

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

Solution 22

(a)

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

(b)

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

(c)

n : P 0, 1

(d)

n : P 0, 1 — true . (n, n)

Solution 23

(a)

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

(b)

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

Solution 24

(a)

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

Pairs == Z $\times Z$

(b)

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

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

(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:

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

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)

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

(b)

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

(c)

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

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

$$\mid \; \mathit{childOf} : \mathit{Person} \leftrightarrow \mathit{Person}$$

(a)

parentOf == childOf^{-1}

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

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

Or, via an axiomatic definition:

$$\begin{array}{|l} \mathit{parentOf} : \mathit{Person} \leftrightarrow \mathit{Person} \\ \hline \mathit{parentOf} = \mathit{childOf}^{-1} \end{array}$$

(b)

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

(c)

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

(d)

ancestorOf == 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+ == $\{a, b \colon N \mid b > a\}$

(d)

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

Solution 32

(a)

$$
\begin{aligned}
& x \mapsto y \in A \lhd B \lhd R \\
& \qquad \Leftrightarrow x \in A \wedge x \mapsto y \in (B \lhd R) \\
& \qquad \Leftrightarrow x \in A \wedge x \in B \wedge x \mapsto y \in R \\
& \qquad \Leftrightarrow x \in A \cap B \wedge x \mapsto y \in R \\
& \qquad \Leftrightarrow x \mapsto y \in A \cap B \lhd R
\end{aligned}
$$

(b)

$$
\begin{aligned}
& x \mapsto y \in R \cup S \rhd C \\
& \qquad \Leftrightarrow x \mapsto y \in R \cup S \wedge y \in C \\
& \qquad \Leftrightarrow (x \mapsto y \in R \vee x \mapsto y \in S) \wedge y \in C \\
& \qquad \Leftrightarrow x \mapsto y \in R \wedge y \in C \vee x \mapsto y \in S \wedge y \in C \\
& \qquad \Leftrightarrow x \mapsto y \in R \rhd C \vee x \mapsto y \in S \rhd C \\
& \qquad \Leftrightarrow x \mapsto y \in (R \rhd C) \cup (S \rhd C)
\end{aligned}
$$

Functions

Solution 33

The set of 9 functions:

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

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

(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 -¿ P Person

where

children = p : Person . p —-¿ parentOf(— p —)

end

(b)

axdef

number$_o f_g randchildren : Person- > N$

where

number$_o f_g randchildren = p : Person.p \mid - > (parentOfo9parentOf)(\mid p \mid)$

end

Solution 36

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

axdef

number$_of_drivers : (Drivers <-> Cars) -> (Cars -> N)$

where

forall r : Drivers ¡-¿ Cars — number$_of_drivers(r) = c : ran r . c | -> \{$ 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)

$$\begin{array}{|l}
f : Place \rightarrow \mathbb{P}\ Place \\
\hline
\forall\, p : Place \bullet f(p) = \{q : Place \mid p \mapsto q \in \operatorname{ran} trains\}
\end{array}$$

(b)

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

(c)

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

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

Solution 39

(a)

large$_c$$oins : Collection - > N$

$\forall\, c : Collection \bullet large_c oins(c) = c(large)$

(Blocked by: underscore in identifier for fuzz compatibility)

(b)

$\text{add}_c oin : Collection * Coin- > Collection$

$\forall c\colon Collection \bullet \forall d\colon Coin \bullet add_c oin(c, d) = c \cup [\![d]\!]$

(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)

hd : seq(Title * Length * Viewed)

$\text{cumulative}_t otalhd <= 12000$

$\forall p\colon \operatorname{ran} hd \bullet p.2 \leq 360$

Note that $\text{cumulative}_t otalisdefinedinpart(d)$.

(b)

$\{p\colon \operatorname{ran} hd \mid p.2 > 120 \bullet p.1\}$

(c)

These can be defined recursively:

viewed ¡¿ = ¡¿

viewed ¡x¿ $^{s} = ifx.3 = yesthen < x >^{v} iewedselsevieweds$

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

(d)

$cumulative_{t}otal :$ seq $Title * Length * Viewed \rightarrow N$

$cumulative_{t}otal(\langle\rangle) = 0 \forall x\colon Title * Length * Viewed \bullet \forall s\colon$ seq $Title * Length * Viewed \bullet cumulative_{t}ot$

(e)

(mu p : ran hd — $\forall q\colon$ ran $hd \bullet p \neq q$— p.2 ¿ q.2 — 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 : seq(Title * Length * Viewed) -¿ seq(Title * Length * Viewed)

where

$\forall s\colon$ seq $Title{*}Length{*}Viewed{\bullet}g(s) = s$—¿ $\{x\colon$ ran $s \mid x \neq longest_{v}iewed(s)\}$

end

Where longest$_v iewedisdefinedas$

axdef

longest$_v iewed : seq(Title * Length * Viewed) + - > Title * Length * Viewed$

where

$\forall s :$ seq $Title * Length * Viewed \bullet longest_v iewed(s) =$ ($\mu\, p :$ ran $s \bullet p.3 = yes$and $\forall q :$ ran $s \bullet p \neq q \wedge q.3 = yes$— p.2 ¿ 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)

$$s : \text{seq } Title * Length * Viewed \rightarrow \text{seq } Title * Length * Viewed$$

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

Solution 41

(a)

axdef

records : Year -—¿ Table

where

dom(records) = 1993..current

$\forall y\colon \mathrm{dom}\, \mathit{records} \bullet \# \mathit{records}(y) \leq 50$

forall y : dom(records) — $\forall e\colon \mathrm{ran}\, \mathit{records}(y) \bullet \mathit{year}(e.1) = y$

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

end

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

(b)

(i)

$\{e\colon \mathit{Entry} \mid \exists r\colon \mathrm{ran}\, \mathit{records} \bullet e \in \mathrm{ran}\, r \wedge e.3 = 479\}$
ii
$\{e\colon \mathit{Entry} \mid \exists r\colon \mathrm{ran}\, \mathit{records} \bullet e \in \mathrm{ran}\, r \wedge e.6 > e.5\}$
iii
$\{e\colon \mathit{Entry} \mid \exists r\colon \mathrm{ran}\, \mathit{records} \bullet e \in \mathrm{ran}\, r \wedge e.7 \geq 70\}$
iv
$\{c\colon \mathit{Course} \mid \forall r\colon \mathrm{ran}\, \mathit{records} \bullet \forall e\colon \mathrm{ran}\, r \bullet e.2 = c \Rightarrow e.7 \geq 70\}$
v

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

(c)

axdef

where

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

end

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

(d)

$\forall x : Entry \bullet \forall s : \text{seq } Entry \bullet total(\langle\rangle) = 0 \land total(\langle x \rangle \frown s) = x.5 + total(s)$

Solution 42

$[Person]$

axdef

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

where

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

end

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

(b)

axdef

add : N * Person * State -—¿ State

where

$\forall n\colon N \bullet \forall p\colon \mathit{Person} \bullet \forall s\colon \mathit{State} \bullet n \in \operatorname{dom} s \land p \notin \bigcup \operatorname{ran}\operatorname{ran} s$—

add(n, p, s) = s ++ n —-¿ s(n) $^{<}p >$

end

(Blocked by: -—¿ operator not implemented)

Solution 43

(a)

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

(ii) forall i : dom bookings — $\forall x\colon \mathit{bookings}(i) \bullet \{x.2, x.3\}$subseteq 1..max(i.1)

(iii) forall i : dom bookings — $\forall b\colon \mathit{bookings}(i) \bullet b.2 \leq b.3$

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

(Blocked by: nested quantifiers - parser limitation)

(b)

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

(ii) i : dom bookings — i.1 = Banbury and $\exists\, b : bookings(i) \bullet 50 \in b.2 \,..\, b.3$

(iii) r : Room; s : N — $\exists\, i : \operatorname{dom} bookings \bullet i.1 = r \land i.2 = s.$ (r, s)

(iv) r : Room — $\exists\, i : \operatorname{dom} bookings \bullet i.1 = r$— (bookings(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

reverse (¡¿ $^{t}) = reverset[cat.1a] = (reverset)^{<} > [cat.1b]$

where cat.1a is ¡¿ $^{s} = sandcat.1biss^{<} >= s$

and the second by

reverse ((¡x¿ $^{u})^{t}) = reverse(< x >^{(} u^{t}))[cat.2]$

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

= (reverse t $^{r}everseu)^{<}x > [anti - distributive]$

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

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

Solution 45

The base case:

reverse (reverse ¡¿) = reverse ¡¿ [reverse.1] = ¡¿ [reverse.1]

The inductive step:

reverse (reverse (¡x¿ ^{t}))

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

= reverse (¡x¿) $^{r}everse(reverset)[anti - distributive]$

= reverse (¡x¿ $^{<} >)^{r}everse(reverset)[cat.1]$

= ((reverse ¡¿) $^{<}x >)^{r}everse(reverset)[reverse.2]$

$= (¡¿ ^{<} x >)^r everse(reverset)[reverse.1]$

$=$ ¡x¿ $^r everse(reverset)[cat.1]$

$=$ ¡x¿ $^t[reverse(reverset) = t]$

Solution 46

(a)

count : Tree -¿ N

count stalk = 0

$\forall n \colon N \bullet count$(leaf n) = 1

$\forall t1, t2 \colon Tree \bullet count(branch(t1, t2)) = count$t1 + count t2

(Blocked by: recursive free types and pattern matching)

(b)

flatten : Tree -¿ seq N

flatten stalk = ¡¿

$\forall n \colon N \bullet flatten$(leaf n) = ¡n¿

$\forall t1, t2 \colon Tree \bullet flatten(branch(t1, t2)) = flatten$t1 $^f lattent2$

(Blocked by: recursive free types and pattern matching)

Solution 47

First, exhibit the induction principle for the free type:

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

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

This gives three cases for the proof:

(flatten stalk) = ¡¿ [flatten] = 0 [] = count stalk [count]

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

Supplementary material : assignment practice

Solution 48

$[\mathit{SongId}, \mathit{UserId}, \mathit{PlaylistId}, \mathit{Playlist}]$

$\mathit{songs} : \mathbb{F}\ \mathit{SongIdusers} : \mathbb{F}\ \mathit{UserIdplaylists} : \mathit{PlaylistId} \nrightarrow \mathit{PlaylistplaylistOwner} : \mathit{PlaylistId} \nrightarrow \mathit{UserIdpla}$

$\forall\, i : \operatorname{dom} \mathit{playlists} \bullet \operatorname{ran} \mathit{playlists}(i) \mathit{subseteqsongs} \operatorname{dom} \mathit{playlistOwnersubseteq} \operatorname{dom} \mathit{playlists} \operatorname{ran} \mathit{playlistOu}$

Solution 49

axdef

hated : UserId +-¿ F SongId

loved : UserId +-¿ F SongId

where

dom(hated) subseteq users

$\forall\, i :\ \mathrm{dom}\, hated \bullet hated(i)$subseteq songs

dom(loved) subseteq users

$\forall\, i :\ \mathrm{dom}\, loved \bullet loved(i)$subseteq songs

forall i : dom(hated) union dom(loved) — hated(i) intersect loved(i) = emptyset

end

(Blocked by: union in quantifier domain - parser issue)

Solution 50

(a)

abbrev
A == users $\setminus \bigcup \mathrm{ran}\, playlistSubscribers$

(b)

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

(c)

C == (mu u : dom(loved) — $\forall\, v :\ \mathrm{dom}\, loved \bullet u \neq v$— (loved(u)) ¿ (loved(v)))

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

(d)

D == (mu s : songs — $\forall t\colon songs \bullet s \neq t$— $\{u\colon UserId \mid s \in loved(u)\}$ ¿ $\{u\colon UserId \mid t \in loved(u)\}$)

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

Solution 51

(a)

Let's first define two helper functions:

loveHateScore : SongId +-¿ N

forall i : songs — $\{u\colon UserId \mid i \in loved(u)\}$ ¿= $\{u\colon UserId \mid i \in hated(u)\}$ $\Rightarrow$

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

and

forall i : songs — $\{u\colon UserId \mid i \in loved(u)\}$ ¡ $\{u\colon UserId \mid i \in hated(u)\}$ $\Rightarrow$

loveHateScore(i) = 0

$$\begin{array}{|l} playlistCount : SongId \nrightarrow N \\ \hline \forall i\colon songs \bullet playlistCount(i) = \#\{p\colon \mathrm{dom}\, playlist \mid i \in \mathrm{ran}\, playlist(p)\} \end{array}$$

We then have:

$length : SongId \pfun Npopularity : SongId \pfun N$

dom $lengthsubseteqsongs$ dom $popularitysubseteqsongs \forall i : songs \bullet popularity(i) = loveHateScore(i) + p$

(b)

mostPopular : SongId

(exists1 i : songs — $\forall j : songs \bullet i \neq j$— popularity(i) ¿ popularity(j)) $\Rightarrow$

mostPopular = (mu i : songs — $\forall j : songs \bullet i \neq j$— popularity(i) ¿ popularity(j))

and

not (exists1 i : songs — $\forall j : songs \bullet i \neq j$— popularity(i) ¿ popularity(j)) $\Rightarrow$

mostPopular = nullSong

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

(c)

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

Solution 52

(a)

premiumPlays : seq(Play) -¿ seq(Play)

premiumPlays(¡¿) = ¡¿

forall x : Play; s : seq(Play) —

premiumPlays(¡x¿ ^{s}) $=< x >^{p} remiumPlays(s) if userStatus(x.2) = premium$

premiumPlays(s) if userStatus(x.2) = standard

(Note: Uses camelCase for fuzz compatibility)

(b)

standardPlays : seq(Play) -¿ seq(Play)

standardPlays(¡¿) = ¡¿

forall x : Play; s : seq(Play) —

standardPlays(¡x¿ ^{s}) $=< x >^{s} tandardPlays(s) if userStatus(x.2) = standard$

standardPlays(s) if userStatus(x.2) = premium

(Note: Uses camelCase for fuzz compatibility)

(c)

cumulativeLength : seq(Play) -¿ N

cumulativeLength(¡¿) = 0

forall x : Play; s : seq(Play) —

cumulativeLength(¡x¿ ^{s}) $= length(x.1) + cumulativeLength(s)$

(Note: Uses camelCase for fuzz compatibility)