Workshop Testing and Formal Methods, Week 2, With Answers

September 11, 2017

```
> module Workshop2Answers where
> import Data.List
```

```
> infix 1 -->
```

```
> (-->) :: Bool -> Bool -> Bool
> p --> q = (not p) || q
```

```
> forall = flip all
```

This workshop is about understanding fundamental concepts in algorithm specification and algorithm design.

The focus is on pre- and postcondition specifications.

The first exercise uses a sudoku example that you will encounter further on in the course.

1. A sudoku is a 9×9 matrix of numbers in $\{1, \ldots, 9\}$, possibly including blanks, satisfying certain constraints. A *sudoku problem* is a sudoku containing blanks, but otherwise satisfying the sudoku constraints. The sudoku solver transforms the problem into a solution.

Give a Hoare triple for a sudoku solver. If the solver is called P, the Hoare triple consists of

$$\{\text{precondition}\}$$
$$P$$
$$\{\text{postcondition}\}$$

The precondition of the sudoku solver is that the input is a correct sudoku problem.

The postcondition of the sudoku solver is that the transformed input is a solution to the initial problem.

State the pre- and postconditions as clearly and formally as possible.

Answer

First find an appropriate representation for sudoku problems. We will assume this is `grid[i][j]` where `grid` is an array of arrays.

A position (i, j) is instantiated if `grid[i][j]` is not equal to `nil`.

A grid is correct if they satisfy the following constraints:

- For all i in $1, \ldots, 9$: all j, k in $1, \ldots, 9$ for which (i, j) and (i, k) are instantiated satisfy `grid[i][j]` in $1, \ldots, 9$, `grid[i][k]` in $1, \ldots, 9$, and if j and k are different then `grid[i][j]` and `grid [i][k]` are different.
- For all j in $1, \ldots, 9$: all i, k in $1, \ldots, 9$ for which (i, j) and (k, j) are instantiated satisfy `grid[i][j]` in $1, \ldots, 9$, `grid[k][j]` in $1, \ldots, 9$, and if i and k are different then `grid[i][j]` and `grid[k][j]` are different.
- Similarly for the nine blocks ...

Precondition: Input grids have to be correct.

Postcondition: Output grids have to be correct, fully instantiated, and instantiated values from the input have to be preserved.

A sudoku solving program is correct if it maps correct input grids to correct output grids, and moreover it holds for each position (i, j) that if the position is instantiated in the input, its grid value has not changed in the output.

2. Suppose $\{p\}\ f\ \{q\}$ holds for some function $f : a \to a$ and a pair of properties p and q.

Recall the meaning of $\{p\}\ f\ \{q\}$:

For every possible argument x for f it is the case that if x has property p then $f(x)$ has property q.

- If p' is stronger that p, does it follow that $\{p'\}\ f\ \{q\}$ still holds?

    - If p' is weaker that p, does it follow that $\{p'\}\ f\ \{q\}$ still holds?
    - If q' is stronger that q, does it follow that $\{p\}\ f\ \{q'\}$ still holds?
    - If q' is weaker that q, does it follow that $\{p\}\ f\ \{q'\}$ still holds?

Answer

- If p' is stronger that p, does it follow that $\{p'\}\ f\ \{q\}$ still holds? Yes, strengthening the precondition preserves truth of the Hoare assertion.
- If p' is weaker that p, does it follow that $\{p'\}\ f\ \{q\}$ still holds? No. This step broadens the range of possible tests for the function, and some of these tests may not be appropriate.
- If q' is stronger that q, does it follow that $\{p\}\ f\ \{q'\}$ still holds? No. This step strengthens the requirement on the output of the function, and tests on f that are in accordance with the new specification may be too severe.
- If q' is weaker that q, does it follow that $\{p\}\ f\ \{q'\}$ still holds? Yes, weakening the postcondition preserves truth of the Hoare assertion.

3. A function of type $a \to a$ (a unary function with arguments and values of the same type) can be tested with test properties of the type $a \to \text{Bool}$ or of the type $a \to a \to \text{Bool}$.

We will consider test properties of type $a \to \text{Bool}$.

Define the following predicate on test properties:

```
> stronger, weaker :: [a] -> (a -> Bool) -> (a -> Bool) -> Bool
> stronger xs p q = forall xs (\ x -> p x --> q x)
> weaker   xs p q = stronger xs q p
```

Predict the output of the following Haskell tests:

```
test1 = stronger [1..10] (\ x -> even x && x > 3) even
test2 = stronger [1..10] (\ x -> even x || x > 3) even
test3 = stronger [1..10] (\ x -> (even x && x > 3) || even x) even
test4 = stronger [1..10] even (\ x -> (even x && x > 3) || even x)
```

Answer

The outcomes are:

```
*Workshop2Answers> test1
True
*Workshop2Answers> test2
False
*Workshop2Answers> test3
True
*Workshop2Answers> test4
True
```

4. Which of the following properties is stronger?

- $\lambda x \mapsto x = 0$ and $\lambda x \mapsto x \geq 0$
- $\lambda x \mapsto x \neq 0$ and $\lambda x \mapsto x > 3$
- $\lambda x \mapsto x \neq 0$ and $\lambda x \mapsto x < 3$
- $\lambda x \mapsto x^3 + 7x^2 \geq 3$ and $\lambda x \mapsto \bot$
- $\lambda x \mapsto x \geq 2 \vee x \leq 3$ and $\lambda x \mapsto x \geq 2$
- $\lambda x \mapsto x \geq 2 \wedge x \leq 3$ and $\lambda x \mapsto x \geq 2$

Answer

- $\lambda x \mapsto x = 0$ and $\lambda x \mapsto x \geq 0$. $\lambda x \mapsto x = 0$ is stronger.
- $\lambda x \mapsto x \neq 0$ and $\lambda x \mapsto x > 3$. $\lambda x \mapsto x > 3$ is stronger.
- $\lambda x \mapsto x \neq 0$ and $\lambda x \mapsto x < 3$. Neither is stronger than the other.
- $\lambda x \mapsto x^3 + 7x^2 \geq 3$ and $\lambda x \mapsto \bot$. $\lambda x \mapsto \bot$ is stronger.
- $\lambda x \mapsto x \geq 2 \vee x \leq 3$ and $\lambda x \mapsto x \geq 2$. $\lambda x \mapsto x \geq 2$ is stronger.
- $\lambda x \mapsto x \geq 2 \wedge x \leq 3$ and $\lambda x \mapsto x \geq 2$. $\lambda x \mapsto x \geq 2 \wedge x \leq 3$ is stronger.

5. Implement all properties from the previous question as Haskell functions of type `Int -> Bool`. Note: this is a pen and paper exercise: just write out the definitions. If you have a computer, this allows you to check your answers to the previous exercise, on some small domain like $[(-10)..10]$.

Answer

```
domain = [(-10)..10]

t1 = stronger domain (\x -> x==0) (\ x -> x >= 0)
t2 = weaker domain (\x -> x/=0) (\ x -> x > 3)
t3 = stronger domain (\x -> x/=0) (\ x -> x < 3)
t3' = weaker  domain (\x -> x/=0) (\ x -> x < 3)
t4 = weaker domain (\x -> x^3 + 7*x^2 >= 3) (\x -> False)
t5 = weaker domain (\x -> x>=2 || x<=3) (\x -> x>=2)
t6 = stronger domain (\x -> x>=2 && x<=3) (\x -> x>=2)
```

The outcomes:

```
*Workshop2Answers> t1
True
*Workshop2Answers> t2
True
*Workshop2Answers> t3
False
*Workshop2Answers> t3'
False
*Workshop2Answers> t4
True
*Workshop2Answers> t5
True
*Workshop2Answers> t6
True
```

Now that we know what weaker and stronger means, we can talk about the weakest property p for which

$$\{p\}\ f\ \{q\}$$

holds, for a given function f and a given postcondition property q.

Example: the weakest p for which

$$\{p\}\lambda x \mapsto 2 * x + 4\ \ \{\lambda x \mapsto 0 \leq x < 8\}$$

holds is $\lambda x \mapsto -2 \leq x < 2$.

Note: $\lambda x \mapsto 0 \leq x < 8$ has to hold. The recipe for finding out when that is the case is as follows.

Use the function $\lambda x \mapsto 2 * x + 4$ as a *substitution*: substitute the right-hand side $2 * x + 4$ for x in the postcondition q to get the weakest precondition, and simplify.

6. Work out the weakest preconditions for the following triples. You may assume that the variables range over integers.

- $\{\cdots\}\ \lambda x \mapsto x{+}1\ \{\lambda x \mapsto 2x - 1 = A\}$
- $\{\cdots\}\ \lambda x \mapsto x * x + 1\ \{\lambda x \mapsto x = 10\}$
- $\{\cdots\}\ \lambda x \mapsto x{+}y\ \{\lambda x \mapsto x{-}y = 7\}$
- $\{\cdots\}\ \lambda x \mapsto x{+}y\ \{\lambda x \mapsto x \geq y\}$
- $\{\cdots\}\ \lambda x \mapsto -x\ \{\lambda x \mapsto x \geq 0\}$

Answer

- $\{\lambda x \mapsto 2x + 1 = A\}\ \lambda x \mapsto x{+}1\ \{\lambda x \mapsto 2x - 1 = A\}$
- $\{\lambda x \mapsto x^2 = 9\}\ \lambda x \mapsto x * x + 1\ \{\lambda x \mapsto x = 10\}$ The precondition simplifies to $\lambda x \mapsto x = 3 \vee x = -3$.
- $\{\lambda x \mapsto x = 7\}\ \lambda x \mapsto x{+}y\ \{\lambda x \mapsto x{-}y = 7\}$
- $\{\lambda x \mapsto x \geq 0\}\ \lambda x \mapsto x{+}y\ \{\lambda x \mapsto x \geq y\}$
- $\{\lambda x \mapsto x \leq 0\}\ \lambda x \mapsto -x\ \{\lambda x \mapsto x \geq 0\}$

7. Show the following (again, you may assume that the variables range over integers):

- $\{\lambda n \mapsto x = n^2\}\ \lambda n \mapsto n{+}1\ \{\lambda n \mapsto x = (n{-}1)^2\}$
- $\{\lambda x \mapsto A = x\}\ \lambda x \mapsto x{+}1\ \{\lambda x \mapsto A = x{-}1\}$

- $\{\, \lambda x \mapsto x \geq 0 \,\}\ \lambda x \mapsto x{+}y\ \{\, \lambda x \mapsto x \geq y \,\}$
- $\{\, \lambda x \mapsto 0 \leq x < 100 \,\}\ \lambda x \mapsto x{+}1\ \{\, \lambda x \mapsto 0 \leq x \leq 100 \,\}$
- $\{\, \lambda n \mapsto x = (n{+}1)^2 \wedge n = A \,\}\ \lambda n \mapsto n{+}1\ \{\, \lambda n \mapsto x = n^2 \wedge n = A{+}1 \,\}$

Answer

The correct method is to calculate the weakest precondition from the assignment and the postcondition, and next check that this weakest precondition is weaker than the precondition mentioned in the Hoare triple.

- $\{\, \lambda n \mapsto x = n^2 \,\}\ \lambda n \mapsto n{+}1\ \{\, \lambda n \mapsto x = (n{-}1)^2 \,\}$. Weakest precondition of $\lambda n \mapsto n{+}1\ \{\, \lambda n \mapsto x = (n'{-}1)^2 \,\}$ is $\lambda n \mapsto x = n^2$. This proves that the Hoare triple is correct.
- $\{\, \lambda x \mapsto A = x \,\}\ \lambda x \mapsto x{+}1\ \{\, \lambda x \mapsto A = x{-}1 \,\}$. Weakest precondition of $\lambda x \mapsto x{+}1\ \{\, \lambda x \mapsto A = x{-}1 \,\}$ is $\lambda x \mapsto A = x$. This proves that the Hoare triple is correct.
- $\{\, \lambda x \mapsto x \geq 0 \,\}\ \lambda x \mapsto x{+}y\ \{\, \lambda x \mapsto x \geq y \,\}$. Weakest precondition of $\lambda x \mapsto x{+}y\ \{\, \lambda x \mapsto x \geq y \,\}$ is $\lambda x \mapsto x \geq 0$. This proves that the Hoare triple is correct.
- $\{\, \lambda x \mapsto 0 \leq x < 100 \,\}\ \lambda x \mapsto x{+}1\ \{\, \lambda x \mapsto 0 \leq x \leq 100 \,\}$. Weakest precondition of $\lambda x \mapsto x{+}1\ \{\, \lambda x \mapsto 0 \leq x \leq 100 \,\}$ is $\lambda x \mapsto -1 \leq x < 100$. The property $\lambda x \mapsto 0 \leq x < 100$ is *stronger* than the weakest precondition, so the Hoare triple is correct.
- $\{\, \lambda n \mapsto x = (n{+}1)^2 \wedge n = A \,\}\ \lambda n \mapsto n{+}1\ \{\, \lambda n \mapsto x = n^2 \wedge n = A{+}1 \,\}$. Weakest precondition of $\lambda n \mapsto n{+}1\ \{\, \lambda n \mapsto x = n^2 \wedge n = A{+}1 \,\}$ is $\lambda n \mapsto x = (n{+}1)^2 \wedge n = A$. This proves that the Hoare triple is correct.

Note: if you have difficulty in simplifying algebraic expressions, perhaps you should watch some videos on basic algebra on Khan Academy. See the Wikipedia Entry on Khan Academy for further information.