T-79.3001 Logic in computer science: foundations **Spring 2008**
Exercise 12 ([Huth & Ryan, 2000], Chapter 4)
Apr 29–30 and May 2, 2008

Solutions to demonstration problems

Solution to Problem 4

Boolean statements can be represented using basic cases, thus

$$\texttt{a==b} \equiv_{def} \texttt{!(a>b)\&\&!(b>a)}$$

and

$$\texttt{a<b} \equiv_{def} \texttt{b>a}$$
$$\texttt{a!=b} \equiv_{def} \texttt{!(a==b)}$$

We choose $A =$"a>b" and $B =$"b>a" as atomic propositions. This way the statement in item (a) is

$$\neg((\neg A \wedge \neg B) \vee B)$$

and respectivelly, in item (b):

$$\neg(\neg A \wedge \neg B) \wedge \neg B$$

Notice that the sesond proposition is obtained from the first by applying de Morgan's rule and thus the statements are logically equivalent.

Solution to Problem 5

(a) $\models_p$ [x>0] y=x+1 [y>1]

Starting from the postcondition and applying the rule for assignment backwards, we obtain [x+1>1] y=x+1 [y>1]
x>0 is equivalent to x+1>1, and the claim holds.

(b) $\models_p$ [true] y=x; y=x+x+y [y==3*x]

Applying twice the assignment rule, we obtain:

```
[x+x+y==3*x] y=x+x+y [y==3*x]
[x+x+x==3*x] y=x [x+x+y==3*x]
```

and furthermore using the rule for composition:

$$[\texttt{x+x+x==3*x}]\ \texttt{y=x ; y=x+x+y}\ [\texttt{y==3*x}].$$

Statement `x+x+x==3*x` evaluates to true for all integers, and thus the claim holds.

(c) $\models_p$ $[\texttt{x>1}]$ `a=1 ; y=x ; y=y-a` $[\texttt{y>0\&\&x>y}]$

$$[\texttt{y-a>0\&\&x>y-a}]\ \texttt{y=y-a}\ [\texttt{y>0\&\&x>y}]$$
$$[\texttt{x-a>0\&\&x>x-a}]\ \texttt{y=x}\ [\texttt{y-a>0\&\&x>y-a}]$$
$$[\texttt{x-1>0\&\&x>x-1}]\ \texttt{a=1}\ [\texttt{x-a>0\&\&x>x-a}].$$

Now, the latter part of `x-1>0&&x>x-1` evaluates to true for all integers and `x-1>0` is equivalent to `x>1`. Thus the claim holds.

Solution to Problem 6

$$[\texttt{true\&\&!(x>y)}][\texttt{!(x>y)}][\texttt{x<=y}][\texttt{x} == \min(\texttt{x},\texttt{y})]\ \texttt{z=x}\ [\texttt{z} == \min(\texttt{x},\texttt{y})] \text{ and}$$
$$[\texttt{true\&\&(x>y)}][\texttt{(x>y)}][\texttt{y} == \min(\texttt{x},\texttt{y})]\ \texttt{z=y}\ [\texttt{z} == \min(\texttt{x},\texttt{y})].$$

Thus,

```
[true]
if(x>y) then {
    z=y
} else {
    z=x
}  [z == min(x, y)]
```

Solution to Problem 7

First, we need and invariant for the loop. Inspecting the code, we note that the value of variable *z* increases while the value for variable *v* decreases. Moreover, the sum of *z* and *v* stays constant. This constant is obtained for the initila values of *z* and *v*, as thus we have invariant *I*: `z+v==x+y`.

We check that *I* really is an invariant:

$$[\texttt{z+v-1==x+y}]\ \texttt{v=v-1}\ [\texttt{z+v==x+y}]$$
$$[\texttt{z+v==x+y}][\texttt{z+1+v-1==x+y}]\ \texttt{z=z+1}\ [\texttt{z+v-1==x+y}]$$

Thus:

```
[z + v == x + y]
while(!(v == 0)) {
    z = z + 1;
    v = v - 1
} [z + v == x + y && v == 0]
```

Finally, we need to find the preconditions for the assignments before the loop:

```
[z+y==x+y] v=y [z+v==x+y]
[x+y==x+y] z=x [z+y==x+y]
```

Now `x+y==x+y` evaluates to true for all integers.

(b) $\models_t$ `[0<=y] Sum [z==x+y]`

To prove total correctness, we need to make sure the program terminates.

```
[0 <= y] [x + y == x + y && 0 <= y] z = x [z + y == x + y && 0 <= y]
[z + y == x + y && 0 <= y] v = y [z + v == x + y && 0 <= v]
while(!(v == 0)) {
    [z + v == x + y && 0 <= v - 1 && v - 1 < n] z = z + 1;
                        [z + v - 1 == x + y && 0 <= v - 1 && v - 1 < n]
    [z + v - 1 == x + y && 0 <= v - 1 && v - 1 < n] v = v - 1
                        [z + v == x + y && 0 <= v && v < n]
} [z + v == x + y && 0 <= v && v == 0] [z == x + y]
```