Suggested Solutions #2

[Compiled on September 6, 2017]

1. Let *max* be a function that returns the maximal number between two input numbers. Write a specification of *max* as precise as possible.

   - $\{?\}max(x, y)\{?\}$

   Solution.
   $$\{true\}max(x, y)\{(res = x \vee res = y) \wedge res \geq x \wedge res \geq y\}$$

   □

2. Write the specification of a function that concatenates two integer lists. You may define other functions of list and use them in the specification.

   - List of integers is defined as $list ::= nil \mid cons(Int, list)$.

   Solution. Let $concat(xs, ys)$ be a function (call-by-value) that appends a list ys to another list xs. Define a function $size(xs)$ which computes the number of elements in the list xs.

   $$\begin{array}{rcl} size(nil) & = & 0 \\ size(cons(x, xs)) & = & 1 + size(xs) \end{array}$$

   Define a type *option*.

   $$option \quad ::= \quad none \mid some(Int)$$

   The following function can be used to access an element at a specific position of a list.

   $$\begin{array}{rcl} acc(nil, i) & = & none \\ acc(cons(x, xs), 0) & = & some(x) \\ acc(cons(x, xs), i + 1) & = & acc(xs, i) \end{array}$$

   Below is the specification of *concat*.

   $$\begin{array}{lll} \{ & true & \} \\ & concat(xs, ys) & \\ \{ & size(res) = size(xs) + size(ys) & \\ \wedge & (\forall i.(0 \leq i < size(xs) \to acc(res, i) = acc(xs, i))) & \\ \wedge & (\forall j.(0 \leq j < size(ys) \to acc(res, j + n) = acc(ys, j))) & \} \end{array}$$

   For a C-like function $concat(xs, ys)$ with pointers xs and ys, we need logic variables xs_0 and ys_0 quantified by $\exists$ globally to remember the initial lists so that we can describe xs and ys remain unchanged.

   $$\begin{array}{lll} \{ & xs = xs_0 \wedge ys = ys_0 & \} \\ & concat(xs, ys) & \\ \{ & size(res) = size(xs) + size(ys) & \\ \wedge & (\forall i.(0 \leq i < size(xs) \to acc(res, i) = acc(xs, i))) & \\ \wedge & (\forall j.(0 \leq j < size(ys) \to acc(res, j + n) = acc(ys, j))) & \\ \wedge & size(xs) = size(xs_0) \wedge (\forall i.(0 \leq i < size(xs) \to acc(xs, i) = acc(xs_0, i))) & \\ \wedge & size(ys) = size(ys_0) \wedge (\forall j.(0 \leq j < size(ys) \to acc(ys, j) = acc(ys_0, j))) & \} \end{array}$$

   □

3. Complete the proof outline.

$\{x \geq 0 \land y \geq 0 \land \gcd(x,y) = \gcd(m,n)\}$
while $x \neq 0 \land y \neq 0$ **do**
  if $x < y$ **then**
    `x, y := y, x`
  fi;
  `x := x - y`
od
$\{(x = 0 \land y \geq 0 \land y = \gcd(x,y) = \gcd(m,n)) \lor$
$(x \geq 0 \land y = 0 \land x = \gcd(x,y) = \gcd(m,n))\}$

Solution. Let z denote $\gcd(m,n)$.

$\{x \geq 0 \land y \geq 0 \land \gcd(x,y) = z\}$
while $x \neq 0 \land y \neq 0$ **do**
  $\{x \geq 0 \land y \geq 0 \land \gcd(x,y) = z \land x \neq 0 \land y \neq 0\}$
  $\{x \geq 0 \land y \geq 0 \land \gcd(x,y) = z\}$
  if $x < y$ **then**
    $\{x \geq 0 \land y \geq 0 \land \gcd(x,y) = z \land x < y\}$
    `x, y := y, x`
    $\{y \geq 0 \land x \geq 0 \land \gcd(y,x) = z \land y < x\}$
    $\{x \geq 0 \land y \geq 0 \land \gcd(x,y) = z \land x \geq y\}$
  fi;
  $\{x \geq 0 \land y \geq 0 \land \gcd(x,y) = z \land x \geq y\}$
  $\{x - y \geq 0 \land y \geq 0 \land \gcd(x-y,y) = z\}$
  `x := x - y`
  $\{x \geq 0 \land y \geq 0 \land \gcd(x,y) = z\}$
od
$\{x \geq 0 \land y \geq 0 \land \gcd(x,y) = z \land \neg(x \neq 0 \land y \neq 0)\}$
$\{(x = 0 \land y \geq 0 \land y = \gcd(x,y) = z) \lor (x \geq 0 \land y = 0 \land x = \gcd(x,y) = z)\}$

□

4. Compute weakest preconditions.

(a) $wp(x := x+2; y := y-2, x+y = 0)$
(b) $wp(\textbf{if } x < y \textbf{ then } res := y \textbf{ else } res := x \textbf{ fi}, res \geq x \land res \geq y)$

Solution.

(a)

$$\begin{aligned} & wp(x := x+2; y := y-2, x+y=0) \\ = \; & wp(x := x+2, x+(y-2)=0) \\ = \; & (x+2)+(y-2) = 0 \\ = \; & x+y = 0 \end{aligned}$$

(b)

$$\begin{aligned} & wp(\textbf{if } x < y \textbf{ then } res := y \textbf{ else } res := x \textbf{ fi}, res \geq x \land res \geq y) \\ = \; & \quad ((x<y) \to wp(res := y, res \geq x \land res \geq y)) \\ & \land \; (\neg(x<y) \to wp(res := x, res \geq x \land res \geq y)) \\ = \; & \quad ((x<y) \to (y \geq x \land y \geq y)) \\ & \land \; (\neg(x<y) \to (x \geq x \land x \geq y)) \\ = \; & \mathit{true} \end{aligned}$$

□

5. Conside the following program.

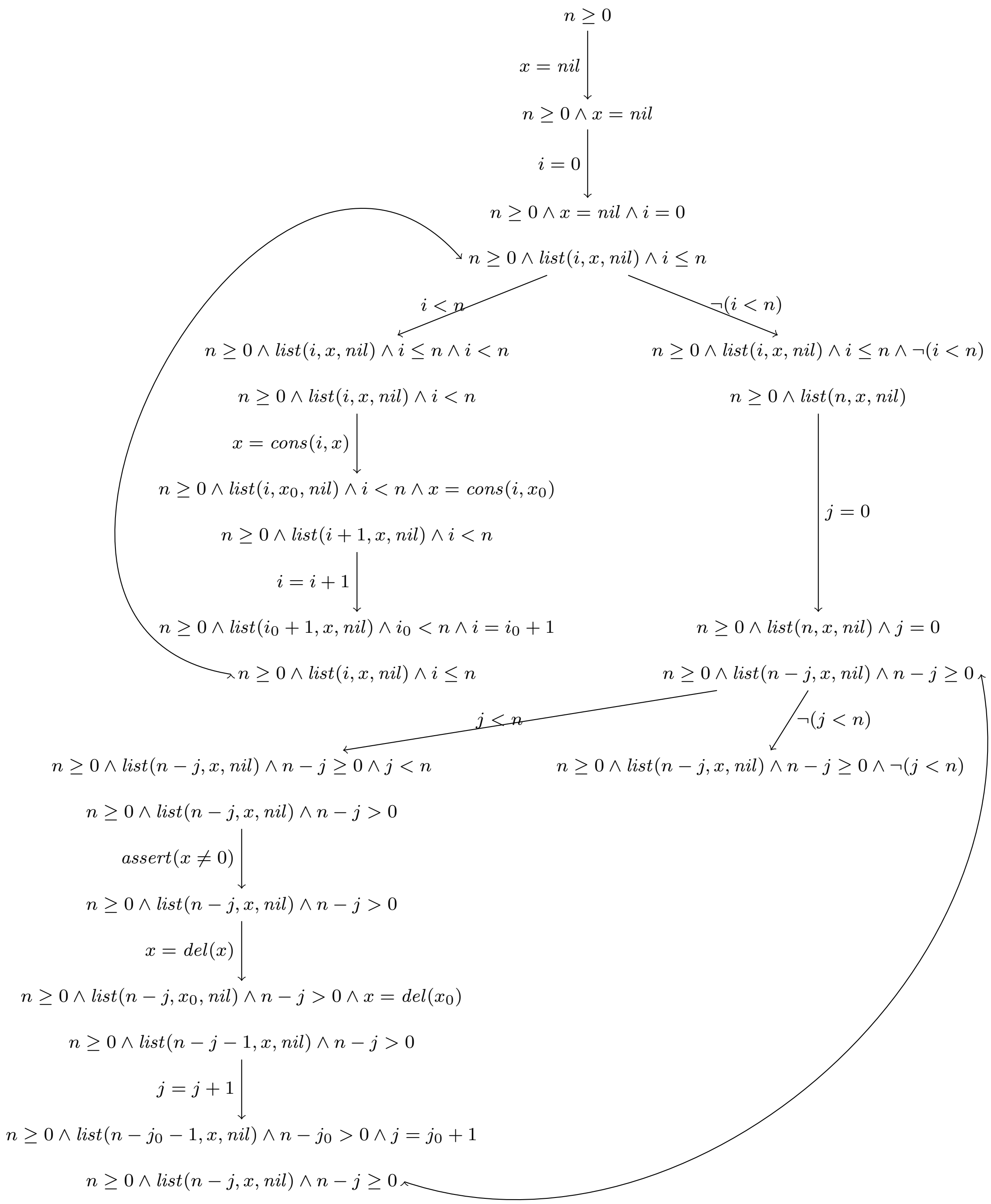
```
x = nil;
i = 0;
while(i < n) {
  x = cons(i, x);
  i = i + 1;
}
j = 0
while(j < n) {
  assert(x != nil)
  x = del(x);
  j = j + 1;
}
```

Assume $n \geq 0$ and

- $\mathit{list}(0, x, x)$ for all x
- $\mathit{list}(0, x, z) \rightarrow x = z$
- $x = \mathit{cons}(a, b) \wedge \mathit{list}(n, b, z) \leftrightarrow \mathit{list}(n+1, x, z)$
- $\mathit{list}(n, x, z) \wedge y = \mathit{del}(x) \wedge n > 0 \rightarrow \mathit{list}(n-1, y, z)$
- $\mathit{list}(n, x, z) \wedge n > 0 \rightarrow x \neq \mathit{nil}$

Either show that the assertion wont be violated or find a counterexample that violates the assertion. ($\mathit{list}(n, x, y)$: x points to a list ended at y with length n.)

Solution. Logic variables x_0, i_0, and j_0 are quantified implicitly by $\exists$.


$n \geq 0$
$x = nil$
$n \geq 0 \wedge x = nil$
$i = 0$
$n \geq 0 \wedge x = nil \wedge i = 0$
$n \geq 0 \wedge list(i, x, nil) \wedge i \leq n$
$i < n$
$\neg(i < n)$
$n \geq 0 \wedge list(i, x, nil) \wedge i \leq n \wedge i < n$
$n \geq 0 \wedge list(i, x, nil) \wedge i \leq n \wedge \neg(i < n)$
$n \geq 0 \wedge list(i, x, nil) \wedge i < n$
$n \geq 0 \wedge list(n, x, nil)$
$x = cons(i, x)$
$n \geq 0 \wedge list(i, x_0, nil) \wedge i < n \wedge x = cons(i, x_0)$
$j = 0$
$n \geq 0 \wedge list(i + 1, x, nil) \wedge i < n$
$i = i + 1$
$n \geq 0 \wedge list(i_0 + 1, x, nil) \wedge i_0 < n \wedge i = i_0 + 1$
$n \geq 0 \wedge list(n, x, nil) \wedge j = 0$
$n \geq 0 \wedge list(i, x, nil) \wedge i \leq n$
$n \geq 0 \wedge list(n - j, x, nil) \wedge n - j \geq 0$
$j < n$
$\neg(j < n)$
$n \geq 0 \wedge list(n - j, x, nil) \wedge n - j \geq 0 \wedge j < n$
$n \geq 0 \wedge list(n - j, x, nil) \wedge n - j \geq 0 \wedge \neg(j < n)$
$n \geq 0 \wedge list(n - j, x, nil) \wedge n - j > 0$
$assert(x \neq 0)$
$n \geq 0 \wedge list(n - j, x, nil) \wedge n - j > 0$
$x = del(x)$
$n \geq 0 \wedge list(n - j, x_0, nil) \wedge n - j > 0 \wedge x = del(x_0)$
$n \geq 0 \wedge list(n - j - 1, x, nil) \wedge n - j > 0$
$j = j + 1$
$n \geq 0 \wedge list(n - j_0 - 1, x, nil) \wedge n - j_0 > 0 \wedge j = j_0 + 1$
$n \geq 0 \wedge list(n - j, x, nil) \wedge n - j \geq 0$

$\square$