

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 .

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

Define a type *option*.

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

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

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

Below is the specification of $concat$.

$$\begin{aligned} &\{ \hspace{10em} true \hspace{10em} \} \\ &\hspace{10em} concat(xs, ys) \\ &\{ \hspace{2em} size(res) = size(xs) + size(ys) \\ &\quad \wedge (\forall i. (0 \leq i < size(xs) \rightarrow acc(res, i) = acc(xs, i))) \\ &\quad \wedge (\forall j. (0 \leq j < size(ys) \rightarrow acc(res, j + n) = acc(ys, j))) \hspace{2em} \} \end{aligned}$$

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.

$$\begin{aligned} &\{ \hspace{10em} xs = xs_0 \wedge ys = ys_0 \hspace{10em} \} \\ &\hspace{10em} concat(xs, ys) \\ &\{ \hspace{2em} size(res) = size(xs) + size(ys) \\ &\quad \wedge (\forall i. (0 \leq i < size(xs) \rightarrow acc(res, i) = acc(xs, i))) \\ &\quad \wedge (\forall j. (0 \leq j < size(ys) \rightarrow acc(res, j + n) = acc(ys, j))) \\ &\quad \wedge size(xs) = size(xs_0) \wedge (\forall i. (0 \leq i < size(xs) \rightarrow acc(xs, i) = acc(xs_0, i))) \\ &\quad \wedge size(ys) = size(ys_0) \wedge (\forall j. (0 \leq j < size(ys) \rightarrow acc(ys, j) = acc(ys_0, j))) \hspace{2em} \} \end{aligned}$$

□

3. Complete the proof outline.

```

{x ≥ 0 ∧ y ≥ 0 ∧ gcd(x, y) = gcd(m, n)}
while x ≠ 0 ∧ y ≠ 0 do
  if x < y then
    x, y := y, x
  fi;
  x := x - y
od
{(x = 0 ∧ y ≥ 0 ∧ y = gcd(x, y) = gcd(m, n)) ∨
 (x ≥ 0 ∧ y = 0 ∧ x = gcd(x, y) = gcd(m, n))}

```

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

```

{x ≥ 0 ∧ y ≥ 0 ∧ gcd(x, y) = z}
while x ≠ 0 ∧ y ≠ 0 do
  {x ≥ 0 ∧ y ≥ 0 ∧ gcd(x, y) = z ∧ x ≠ 0 ∧ y ≠ 0}
  if x < y then
    {x ≥ 0 ∧ y ≥ 0 ∧ gcd(x, y) = z ∧ x ≠ 0 ∧ y ≠ 0 ∧ x < y}
    x, y := y, x
    {y ≥ 0 ∧ x ≥ 0 ∧ gcd(y, x) = z ∧ y ≠ 0 ∧ x ≠ 0 ∧ y < x}
    {x - y ≥ 0 ∧ y ≥ 0 ∧ gcd(x - y, y) = z}
  fi;
  {x - y ≥ 0 ∧ y ≥ 0 ∧ gcd(x - y, y) = z}
  x := x - y
  {x ≥ 0 ∧ y ≥ 0 ∧ gcd(x, y) = z}
od
{x ≥ 0 ∧ y ≥ 0 ∧ gcd(x, y) = z ∧ ¬(x ≠ 0 ∧ y ≠ 0)}
{(x = 0 ∧ y ≥ 0 ∧ y = gcd(x, y) = z) ∨ (x ≥ 0 ∧ y = 0 ∧ x = gcd(x, y) = z)}

```

□

4. Compute weakest preconditions.

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

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)

$$\begin{aligned}
& wp(\mathbf{if} \ x < y \ \mathbf{then} \ res := y \ \mathbf{else} \ res := x \ \mathbf{fi}, res \geq x \wedge res \geq y) \\
&= ((x < y) \rightarrow wp(res := y, res \geq x \wedge res \geq y)) \\
&\quad \wedge (\neg(x < y) \rightarrow wp(res := x, res \geq x \wedge res \geq y)) \\
&= ((x < y) \rightarrow (y \geq x \wedge y \geq y)) \\
&\quad \wedge (\neg(x < y) \rightarrow (x \geq x \wedge x \geq y)) \\
&= true
\end{aligned}$$

□

5. Consider the following program.

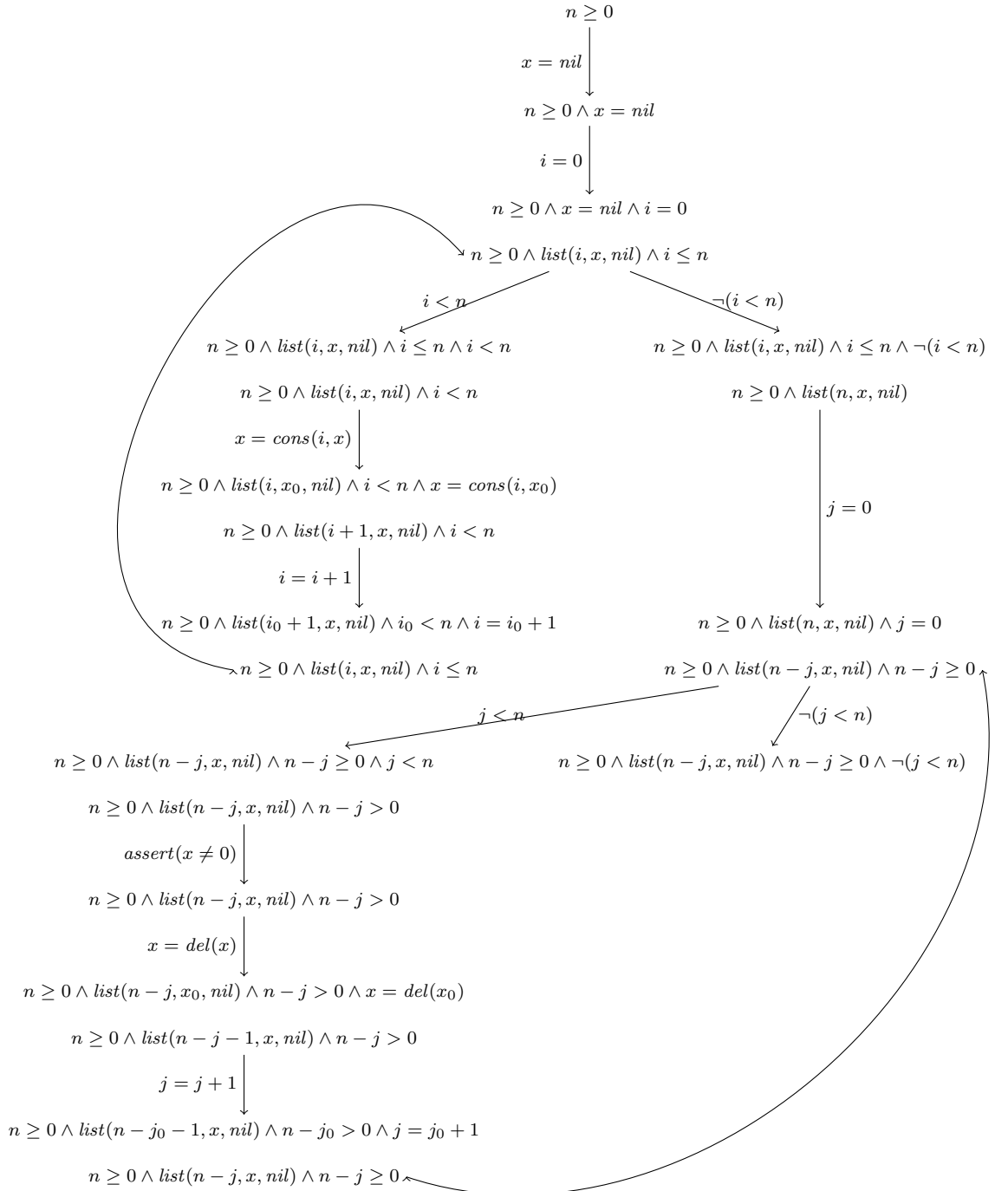
```
x = nil ;
i = 0 ;
while(i < n) {
  x = cons(i, x);
  i = i + 1;
}
j = 0
while(j < n) {
  assert(x != nil)
  x = del(x);
  j = j + 1;
}
```

Assume $n \geq 0$ and

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

Either show that the assertion won't be violated or find a counterexample that violates the assertion. ($list(n, x, y)$: 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 .



□