Purely Functional Algorithm Specification
Exercises Day 4

Jan van Eijck
CWI & ILLC, Amsterdam

ESSLLI, Opole, August 9, 2012

homepages.cwi.nl/~jve/courses/12/esslli12/

```
module Exerc4

where
import List
import While
import Assert
import Reasoning (update,updates)
import GraphsAlgs
```

Exercise about Reachability

```
reachable :: Eq a => [(a,a)] -> a -> [a]
reachable g x = reachable' g [x] [x]

reachable' :: Eq a => [(a,a)] -> [a] -> [a] -> [a]
reachable' g = while2
   (\ current _ -> not (null current))
   (\ current marked -> let
     (y,rest) = (head current, tail current)
     newnodes = [ z  | (u,z) <- g, u == y,
                       notElem z marked   ]
     current' = rest ++ newnodes
     marked'  = marked ++ newnodes
  in
     (current', marked'))
```

Exercise 1 How can this algorithm be tested? Can you find a reasonable assertion or a reasonable step invariant?

Alternative Representation

Exercise 2 Another way to implement a graph $G = (V, E)$ is as a list of vertices (a list of type $[a]$) together with an edge function (edge matrix), i.e. a function of type $a \to a \to$ Bool. Implement the reachable and reachable' functions using this alternative representation. The type declarations are:

```
reachable1 :: Eq a => ([a], a -> a -> Bool) -> a -> [a]

reachable1' :: Eq a => ([a],a->a->Bool) -> [a] -> [a] -> [a]
```

Exercise about Connectedness

Exercise 3 Write a function `cyclic :: Eq a => [(a,a)] -> Bool` that checks whether a list of edges has cycles. A cycle is a path $x \to \cdots \to x$, for some node x.

Same Exercise, Different Representation

Exercise 4 Write a version of `isConnected` in terms of reflexive transitive closure of the edge list of a graph.

Invariant for Reachability Algorithm

Extending the notation xR, let $CR = \bigcup_{x \in C} xR$. In terms of this, a loop invariant for

```
reachable' E C M
```

can be expressed as:

$$xE^* = CE^+ \cup M.$$

Exercise 5 Check that this invariant holds for the step function of `reachable'`. Deduce that the return value of `reachable'` satifies $xE^* = M$.

Exercise 6 Write an assertive version of `reachable'` that uses this invariant.

Exercise About Minimum Spanning Trees

Exercise 7 Let G be a symmetric, undirected weighted graph. Suppose all edges have different positive weights. Show that the minimum spanning tree of G is unique.

Exercise about Breadth First Search

Exercise 8 Find a reasonable assertion for `bfs`, and use this to write an assertive version `bfsA`.

Belman-Ford

Exercise 9 Can you give a proof that the check for negative cycles at the end of the Belmann-Ford algorithm is actually correct?

Exercise 10 Find other suitable assertions to wrap around `bfLoop`.

Exercise 11 Look up Yen's improvement of the Bellman-Ford algorithm, in [1] or on Wikipedia. Implement it.

References

[1] Jin Y. Yen. An algorithm for finding shortest routes from all source nodes to a given destination in general networks. Quarterly of Applied Mathematics, 27:526–530, 1970.