Exam Automated Reasoning

Friday, 1 Februari 2008, 9 - 12 h.

NB. The exam will be corrected and graded by humans, not by a computer. Therefore, you need not to bother too much about the syntactical peculiarities of PVS and Promela. You may use English or Dutch when making the exam.

After correction, you may consult your exam via the teachers of the course (Wim Hesselink and Gerard Renardel).

1. We consider a function $f : \mathbb{N} \to \mathbb{N}$.

   (a) Complete the following definitions in PVS.

   ```
   f: VAR [nat -> nat]

   monotonic?(f): bool = ...
   injective?(f): bool = ...
   ```

   Recall that
   f is called monotonic if for all m, $n \in \mathbb{N}$ with $m \leq n$ we have $f(m) \leq f(n)$;
   f is called injective if for all m, $n \in \mathbb{N}$ we have that $f(m) = f(n)$ implies $m = n$.

   (b) Formulate a lemma that asserts that, if f is monotonic and injective, every $n \in \mathbb{N}$ satisfies $n \leq f(n)$. Give a mathematical proof for this lemma (i.e. not in PVS).
   Hints for the proof:
   (i) use induction over n;
   (ii) you may use that $n + 1 \leq f(n + 1)$ follows from $\neg f(n + 1) \leq n$.

   (c) Formulate the lemma of the previous section in PVS. Show how to prove it in PVS by means of a list of subsequent proof commands and the resulting sequents. Use only proof commands from the following list:

   ```
   assert, flatten, split, skolem!, lift-if,
   case, replace, induct, inst, expand
   ```

   In the latter five cases, you have to provide a full list of parameters to the proof command.

2. There are N players, numbered $0 \leq p < N$. There is a shared variable x of type byte, initially equal to 1. Given is some unknown function

```
f(byte x, byte p, bit a)
```

The players play a game that ends when $x = 0$. As long as $x \neq 0$, player p can execute $x :=$ f(x, p, a) for arbitrary $a = 0$ or 1. The winner is the player that executes $x := 0$. The players are allowed to act in arbitrary order, but always at least one player must act.

Model this in Promela by means of an uninterpreted function f.

(a) How can you test (for given f) that player 0 will never win the game?

(b) How can you test that the game always terminates with some winner?

(c) How can you test that the game never terminates with a winner?

(d) How can you *ensure* that, in every infinite game, player 0 acts infinitely often? (Do not impose any restriction on function f.)

(e) How can you *test* that, in every infinite game, player 0 acts infinitely often?

(f) Take $N = 3$ and give a function f that satisfies the condition of (e) and fails the tests for (a), (b) and (c), without imposing weak fairness. For simplicity, you need not to worry about the size of the state space.