Model Checking WS 2011: Assignment 3

Institute for Formal Models and Verification, JKU Linz

Due 27.10.2011

Example: Abstracting a Program Fragment

The abstract transition system from part b) of Exercise 8 is used to abstract the code fragment shown below (left). It is assumed that `i`, `n` $\in \mathbb{Z}$ *(integers), i.e. values can not overflow. In the abstraction (right), value* $\star$ *denotes nondeterministic choice. Relational expression* `i <= n` *is replaced by predicate* `a`.

```
assert (i <= n);
lock ();
do {
 i++;
 if (i > n) unlock ();
} while (i <= n);
```

```
Bool a = true;
assert (a);
lock ();
do {
 if (a) a = *;
 if (!a) unlock ();
} while (a);
```

Exercise 9

Let x and y be *4-bit signed integer* variables with Java semantics, i.e. two's complement representation and modular arithmetic with underflow/overflow. Let $a \leftrightarrow$ `((x % 2) == 0)` and $b \leftrightarrow$ `(x < y)` be predicates. Similar to part b) of Exercise 8, a and b together define 4 possible abstract states. Draw an abstract transition system for predicates a and b and actions $\alpha :=$ `x++` and $\beta :=$ `y = y - 2`. How do α and β influence the values of a and b? Make sure to consider all possible combinations of abstract states and transitions. For each transition you draw between abstract states, give *concrete values* for variables x and y as a "witness".

Exercise 10

Using predicates a and b and actions α and β from the abstract transition system in Exercise 9, construct an abstract program statement by statement for the program fragment given below. Replace relational expressions by the corresponding predicate. Make sure that *all possible* transitions between abstract states (see Exercise 9) by executing $\alpha :=$ `x++` and $\beta :=$ `y = y - 2` are *fully* encoded in the abstract program. Your program should have only two boolean variables a and b.

```
assert (x < y);
lock();
while (x < y) {
  if ((x % 2) == 0) x++; else y = y - 2;
  if (x >= y && (x % 2) != 0)
    unlock;
}
```

Exercise 11

Justify your answers to the following questions.

a) Explain why the empty set $\lesssim := \{\}$ is a simulation over any arbitrary LTS $L = (S, I, \Sigma, T)$.

b) Given an LTS L and two simulations $\lesssim_1$ and $\lesssim_2$ over L. Prove that $\lesssim_1 \cup \lesssim_2$ is a simulation over L as well.

c) Draw all different LTS $L = (S, I, \Sigma, T)$ with the restrictions that $I = S = \{1, 2\}$, $\Sigma = \{a\}$ and further $(1, a, 2) \notin T$ and $(2, a, 2) \notin T$.

d) Given the relation $\lesssim := S \times S$. For which LTS of part c) is $\lesssim$ a simulation?

Exercise 12

Compute the maximal simulation $\lesssim$ over the following LTS using the fixpoint algorithm: