**Technische Universität München (I7)** **Summer Semester 2017**
**J. Křetínský / T. Meggendorfer / P. Ashok** **04.–05.04.2017**

# Model Checking – Exercise sheet 12–13

### Exercise 12.1 Abstraction refinement

We consider the following program, over the integer variables $x$ and $y$:

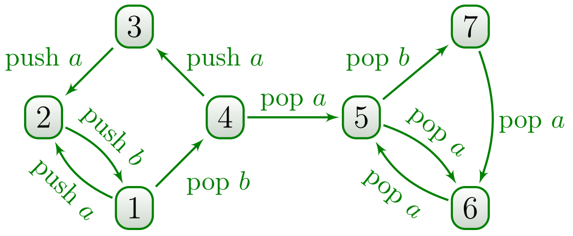
```
if (x >= 0) x = -x;
if (y >= 0) y = -y;
if (x + y > 0) error;
end
```

1. Give the set of configurations of the program (some may not be reachable).
2. Draw the abstract transition system with the predicates $l_1, l_2, l_3, l_4$ and "error".
3. Give a path $\rho$ in the abstract transition system reaching a state where "error" holds.
4. What is the longest prefix (denoted $\rho'$) of $\rho$ that can be concretized ?
5. Denote $q$ the state in the abstract transition system reached by $\rho'$. Give a predicate that separates configurations reachable by $\rho'$ from configurations that admit a successor.
6. Draw the abstract transition system with that additional predicate.
7. How many times does we have to repeat the abstraction refinement technique to exhibit an abstract transition system that does not reach the error state ? Draw that transition system, how many predicates have we introduced ?

### Exercise 12.2 Pre$^*$ in pushdown systems

Consider the following pushdown system, with stack alphabet $\Gamma = \{a, b\}$.

By $\boxed{1} \xrightarrow{\text{push } a} \boxed{2}$, we indicate the presence of transitions $1a \hookrightarrow 2aa$ and $1b \hookrightarrow 2ab$. By $\boxed{4} \xrightarrow{\text{pop } a} \boxed{5}$, we indicate the presence of the transition $4a \hookrightarrow 5$.



1. Let $C = 7b^* = \{7, 7b, 7bb, 7bbb, \ldots\}$. Build the p-automaton accepting pre$^*(7b^*)$.
2. Give the minimal automaton accepting the language of all stacks $w$ such that $1w \in$ pre$^*(C)$.

### Exercise 13.1

Consider the following recursive program:

```
procedure main;
m0: if ? then
call a;
else
call b;
m1: return;
```

```
procedure a;
a0: if ? then
call b;
a1:   call b;
else
call a;
end if;
a2: return;
```

```
procedure b;
b0: if ? then
call a;
b1:   if ? then
call a;
end if;
end if;
b2: return;
```

Recall that `?` denotes a nondeterministic Boolean value.

1. Model the program with a pushdown system.
2. Compute all configurations that can reach the program label `m1`.

### Exercise 13.2

Consider the following recursive program with a global variable `g` and a local variable `l`:

```
boolean g;

procedure main(boolean l);
m0: if l then
call a;
end if;
m1: assert(g == l);
m2: return;
```

```
procedure a();
a0: g := not g;
a1: if not g then
call a;
a2:   call a;
end if;
a3: return;
```

1. Model the program with a pushdown system. Assume that the values of `g` and `l` are not initialized.
2. Compute all configurations that can reach the program label `m2`.
3. Compute all configurations that are reachable from the program label `m0`.