Model Checking – Exercise sheet 13

Exercise 13.1

Consider the pushdown system below, with stack alphabet $\Gamma = \{a, b\}$ where $\xrightarrow{\text{push } a}$ indicates the presence of transitions $1a \rightarrow 2aa$ and $1b \rightarrow 2ab$, and $\xrightarrow{\text{pop } a}$ indicates the presence of transition $4a \rightarrow 5$.

Let $L = 7b^* = \{7, 7b, 7bb, 7bbb, \ldots\}$. Construct the $\mathcal{P}$-automaton accepting $\text{pre}^*(L)$.

Exercise 13.2

Consider the following recursive program, where $\? \text{ denotes a nondeterministic Boolean value:}$$$

```
procedure main;

m0: if ? then
    call a;
else
    call b;

m1: return;

procedure a;

a0: if ? then
    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;

(a) Model the program with a pushdown system.
(b) Compute all configurations that can reach the program label m1.

Exercise 13.3
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;

(a) Model the program with a pushdown system, where the values of g and l are not initialized.
(b) Compute all configurations that can reach the program label m2.
(c) ⭐ Compute all configurations that are reachable from the program label m0.
Solution 13.1
First note that the transitions of the pushdown system are as follows:

1a → 2aa
1b → 2ab
1b → 4
2a → 1ba
2b → 1bb
3a → 2aa
3b → 2ab
4a → 3aa
4b → 3ab
4a → 5
5b → 7
5a → 6
6a → 5
7a → 6.

We are looking for \text{pre}^*(L) where \(L = 7b^*\). We construct the following \(P\)-automaton for \(L\):
We apply the algorithm to compute $\text{pre}^*(L)$ on the above automaton $A$. More precisely, if $q \xrightarrow{w} r$ in $A$ and if the pushdown system contains a rule $pA \rightarrow qw$, then we add a transition $p \xrightarrow{A} r$ to $A$. For example, this is the case for $7 \xrightarrow{\varepsilon} 7$ and $5b \rightarrow 7$, so we add the transition $5 \xrightarrow{b} 7$: 
By repeatedly doing the same with rules $4a \rightarrow 5$, $6a \rightarrow 5$, $5a \rightarrow 6$, $7a \rightarrow 6$ and $1b \rightarrow 4$, we obtain:
From rule $2a \rightarrow 1ba$, we add the following (orange) transition:
We repeat the process until we derive the following $P$-automaton$^\ddagger$:

![Diagram](image)

**Solution 13.2**

(a) Since the program has no global variable, the pushdown system has a single control-state, say $p$. The stack alphabet is $\Gamma = \{m_0, m_1, a_0, a_1, a_2, b_0, b_1, b_2\}$. The resulting pushdown system is:

---

$^\ddagger$Blue and magenta are only used to help distinguishing $a$ and $b$-transitions.
(b) We are looking for \( \text{pre}^*(L) \) where \( L = p_{m_1}\Gamma^* \). We construct the following \( \mathcal{P} \)-automaton for \( L \):

\[
\begin{align*}
m_0 & \rightarrow a_0m_1 | b_0m_1 \\
m_1 & \rightarrow \varepsilon \\
a_0 & \rightarrow b_0a_1 | a_0a_2 \\
a_1 & \rightarrow b_0a_2 \\
a_2 & \rightarrow \varepsilon \\
b_0 & \rightarrow a_0b_1 | b_2 \\
b_1 & \rightarrow a_0b_2 | b_2 \\
b_2 & \rightarrow \varepsilon
\end{align*}
\]
By applying the algorithm to compute \( \text{pre}^*(L) \), we derive the following \( \mathcal{P} \)-automaton:

![Diagram](image)

**Solution 13.3**

(a) Since the program has a global boolean variable \( g \), the pushdown system has two control-states \( g_0 \) and \( g_1 \) representing respectively \( g = \text{false} \) and \( g = \text{true} \). The stack alphabet is

\[
\Gamma = \{(m_0, \ell_0), (m_0, \ell_1), (m_1, \ell_0), (m_1, \ell_1), (m_2, \ell_0), (m_2, \ell_1), a_0, a_1, a_2, a_3, \perp\}
\]

where \( \perp \) stands for an error, and \((m_i, \ell_j)\) stands for location \( m_i \) of \texttt{main} with \( \ell = \text{true} \) if \( j = 1 \), and \( \ell = \text{false} \) if \( j = 0 \). The resulting pushdown system is:

![Diagram](image)

(b) We are looking for \( \text{pre}^*(L) \) where \( L = (g_0 + g_1) ((m_2, \ell_0) + (m_2, \ell_1)) \). We construct the following \( \mathcal{P} \)-automaton for \( L \):
By applying the algorithm to compute $\text{pre}^*(L)$, we derive the following $\mathcal{P}$-automaton:

(c) We are looking for $\text{post}^*(L)$ where $L = (g_0 + g_1) ((m_0, \ell_0) + (m_0, \ell_1)) \Gamma^*$. We construct the following $\mathcal{P}$-automaton for $L$:
By applying the algorithm to compute $\text{post}^*(L)$, we derive the following $\mathcal{P}$-automaton: