Exercise 12.1

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

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

(b) Give the minimal automaton accepting the language of all stacks $w$ such that $1w \in \text{pre}^*(L)$.

Exercise 12.2

Consider the following recursive program, where $?$ denotes a nondeterministic Boolean value:

```plaintext
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;

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

Exercise 12.3
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 12.1

(a) First note that the transitions of the pushdown system are as follows:

1a → 2a
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 $\mathcal{P}$-automaton for $L$:
We apply the algorithm to compute $\text{pre}^*(L)$ on the above automaton $\mathcal{A}$. More precisely, if $q \xrightarrow{w} r$ in $\mathcal{A}$ and if the pushdown system contains a rule $pA \rightarrow qw$, then we add a transition $p \xrightarrow{A} r$ to $\mathcal{A}$. 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 $\mathcal{P}$-automaton$^1$:

$^1$Blue and magenta are only used to help distinguishing $a$ and $b$-transitions.
(b) We are interested in the language accepted by the $P$-automaton obtained in (a) starting from control-state 1. By removing the control-states which are non reachable, we obtain the following automaton:
By determinizing and minimizing the above automaton, we derive:

![Automaton Diagram]

Solution 12.2
(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:

- $m_0 \rightarrow a_0 m_1 | b_0 m_1$
- $m_1 \rightarrow \varepsilon$
- $a_0 \rightarrow b_0 a_1 | a_0 a_2$
- $a_1 \rightarrow b_0 a_2$
- $a_2 \rightarrow \varepsilon$
- $b_0 \rightarrow a_0 b_1 | b_2$
- $b_1 \rightarrow a_0 b_2 | b_2$
- $b_2 \rightarrow \varepsilon$

Legend:
- $p$
(b) We are looking for \( \text{pre}^*(L) \) where \( L = p \ m_1 \Gamma^* \). We construct the following \( \mathcal{P} \)-automaton for \( L \):

\begin{center}
\begin{tikzpicture}
  \node (p) [shape=circle,draw] {\( p \)};
  \node (m1) [shape=circle,draw, below right of=p] {\( m_1 \)};
  \node (g) [shape=circle,draw, above right of=m1] {\( \Gamma \)};
  \draw (p) edge [->] node [near start] {} (m1);
  \draw (m1) edge [->] node [midway] {} (g);
\end{tikzpicture}
\end{center}

By applying the algorithm to compute \( \text{pre}^*(L) \), we derive the following \( \mathcal{P} \)-automaton:

\begin{center}
\begin{tikzpicture}
  \node (p) [shape=circle,draw] {\( p \)};
  \node (m0m1) [shape=circle,draw, below right of=p] {\( m_0, m_1 \)};
  \node (g) [shape=circle,draw, above right of=m0m1] {\( \Gamma \)};
  \draw (p) edge [->] node [near start] {} (m0m1);
  \draw (m0m1) edge [->] node [midway] {} (g);
\end{tikzpicture}
\end{center}

**Solution 12.3**

(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

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

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

\begin{center}
\begin{tikzpicture}
  \node (g0) [shape=circle,draw] {\( g_0 \)};
  \node (g1) [shape=circle,draw, below right of=g0] {\( g_1 \)};
  \node (a0a1) [shape=circle,draw, above right of=g0] {\( a_0 \rightarrow a_1 \)};
  \node (a0a1) [shape=circle,draw, above right of=g1] {\( a_0 \rightarrow a_1 \)};
  \draw (g0) edge [->] node [near start] {} (g1);
  \draw (g1) edge [->] node [near start] {} (g0);
  \draw (g0) edge [->] node [near start] {} (a0a1);
  \draw (g1) edge [->] node [near start] {} (a0a1);
  \node (m0l0) [shape=circle,draw, below right of=g0] {\( m_0, \ell_0 \)};
  \node (m0l1) [shape=circle,draw, below right of=g1] {\( m_0, \ell_1 \)};
  \node (m1l0) [shape=circle,draw, below right of=g0] {\( m_1, \ell_0 \)};
  \node (m1l1) [shape=circle,draw, below right of=g1] {\( m_1, \ell_1 \)};
  \node (m2l0) [shape=circle,draw, below right of=g0] {\( m_2, \ell_0 \)};
  \node (m2l1) [shape=circle,draw, below right of=g1] {\( m_2, \ell_1 \)};
  \node (a0a2) [shape=circle,draw, below right of=g0] {\( a_0 \rightarrow a_2 \)};
  \node (a0a3) [shape=circle,draw, below right of=g0] {\( a_0 \rightarrow a_3 \)};
  \node (a2a3) [shape=circle,draw, below right of=g1] {\( a_2 \rightarrow a_3 \)};
  \node (a3e) [shape=circle,draw, below right of=g1] {\( a_3 \rightarrow \varepsilon \)};
  \node (a1e) [shape=circle,draw, below right of=g1] {\( a_1 \rightarrow \varepsilon \)};
  \node (a3e) [shape=circle,draw, below right of=g1] {\( a_3 \rightarrow \varepsilon \)};
  \node (a2e) [shape=circle,draw, below right of=g1] {\( a_2 \rightarrow \varepsilon \)};
  \node (a2e) [shape=circle,draw, below right of=g1] {\( a_2 \rightarrow \varepsilon \)};
  \node (a1e) [shape=circle,draw, below right of=g1] {\( a_1 \rightarrow \varepsilon \)};
  \node (a1e) [shape=circle,draw, below right of=g1] {\( a_1 \rightarrow \varepsilon \)};
  \draw (m0l0) edge [->] node [near start] {} (m0l0);
  \draw (m0l0) edge [->] node [near start] {} (m0l1);
  \draw (m0l1) edge [->] node [near start] {} (m0l0);
  \draw (m0l1) edge [->] node [near start] {} (m0l1);
  \draw (m1l0) edge [->] node [near start] {} (m1l0);
  \draw (m1l0) edge [->] node [near start] {} (m1l1);
  \draw (m1l1) edge [->] node [near start] {} (m1l0);
  \draw (m1l1) edge [->] node [near start] {} (m1l1);
  \draw (m2l0) edge [->] node [near start] {} (m2l0);
  \draw (m2l0) edge [->] node [near start] {} (m2l1);
  \draw (m2l1) edge [->] node [near start] {} (m2l0);
  \draw (m2l1) edge [->] node [near start] {} (m2l1);
  \draw (a0a2) edge [->] node [near start] {} (a0a2);
  \draw (a0a2) edge [->] node [near start] {} (a0a3);
  \draw (a0a3) edge [->] node [near start] {} (a0a2);
  \draw (a0a3) edge [->] node [near start] {} (a0a3);
  \draw (a2a3) edge [->] node [near start] {} (a2a3);
  \draw (a2a3) edge [->] node [near start] {} (a2a3);
  \draw (a3e) edge [->] node [near start] {} (a3e);
  \draw (a3e) edge [->] node [near start] {} (a3e);
  \draw (a2e) edge [->] node [near start] {} (a2e);
  \draw (a2e) edge [->] node [near start] {} (a2e);
  \draw (a1e) edge [->] node [near start] {} (a1e);
  \draw (a1e) edge [->] node [near start] {} (a1e);
\end{tikzpicture}
\end{center}
(b) We are looking for pre\(^*\)(L) where \(L = (g_0 + g_1) ((m_2, \ell_0) + (m_2, \ell_1))\Gamma^*\). We construct the following \(\mathcal{P}\)-automaton for \(L\):

![Diagram](attachment:diagram.png)

By applying the algorithm to compute pre\(^*\)(L), we derive the following \(\mathcal{P}\)-automaton:

![Diagram](attachment:diagram2.png)

(c) We are looking for 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 \(L\):

![Diagram](attachment:diagram3.png)
By applying the algorithm to compute $\text{post}^*(L)$, we derive the following $\mathcal{P}$-automaton: