Solutions to TD9

Reminder:

A *pushdown system (PDS)* is a triple $\mathcal{P} = (P, \Gamma, \Delta)$, where P is a finite set of *control states*, Γ is a finite *stack alphabet*, and $\Delta \subseteq (P \times \Gamma) \times (P \times \Gamma^*)$ is a finite set of *rules*. We write $pA \hookrightarrow qw$ when $((p, A), (q, w)) \in \Delta$. We associate with a PDS $\mathcal{P}$ and an initial configuration $c_0 \in P \times \Gamma^*$ the transition system $\mathcal{T}_{\mathcal{P}} = (\mathit{Con}(\mathcal{P}), \rightarrow, c_0)$, where $\mathit{Con}(\mathcal{P}) = P \times \Gamma^*$ is the set of *configurations*, and $pAw' \rightarrow qww'$ for all $w' \in \Gamma^*$ iff $pA \hookrightarrow qw \in \Delta$. We write $pw \Rightarrow p'w'$ if there is a path from pw to $p'w'$ in $\mathcal{T}_{\mathcal{P}}$.

Let $\mathcal{P}$ be a PDS. A $\mathcal{P}$*-automaton* is a finite automaton $\mathcal{A} = (Q, \Gamma, P, T, F)$, where the alphabet of $\mathcal{A}$ is the stack alphabet Γ, and the initial states of $\mathcal{A}$ are the control states P. It is *normalized* if there are no transitions leading into initial states. We say that $\mathcal{A}$ *accepts* the configuration pw if $\mathcal{A}$ has a path labelled by input w starting at p and ending at some final state. We denote by $\mathcal{L}(\mathcal{A})$ be the set of configurations accepted by $\mathcal{A}$. A set C of configurations is called *regular* if there is some $\mathcal{P}$-automaton $\mathcal{A}$ with $\mathcal{L}(\mathcal{A}) = C$.

Given a set C of configurations of $\mathcal{P}$, we let

$$\mathit{pre}^*(C) = \{c' \mid \exists c \in C : c' \Rightarrow c\} \qquad \mathit{post}^*(C) = \{c' \mid \exists c \in C : c \Rightarrow c'\}$$

If C is regular, then so are $\mathit{pre}^*(C)$ and $\mathit{post}^*(C)$. If $\mathcal{A}$ is a normalized $\mathcal{P}$-automaton accepting C, $\mathcal{A}$ can be transformed into an automaton accepting $\mathit{pre}^*(C)$ by applying the following saturation rule until no transition can be added:

> If $q \xrightarrow{w} r$ currently holds in $\mathcal{A}$ and $pA \hookrightarrow qw$ is a rule in $\mathcal{P}$, then add the transition (p, A, r) to $\mathcal{A}$.

The procedure for $\mathit{post}^*(C)$ is similar.

1 Labelled Pushdown Systems

1. Let $\mathcal{A}_L = (Q, \Sigma, \Delta_L, I, F)$ be the finite automaton which recognises language L. We consider the product of $\mathcal{A}_L$ and $\mathcal{P} = (P, \Gamma, \Delta, \Sigma)$ and obtain a new pushdown system $\mathcal{P}' = (P \times Q, \Gamma, \Delta', \Sigma)$, where Δ' is defined using the following rule:

   $$(p_1, q_1)A \overset{a}{\hookrightarrow} (p_2, q_2)w \text{ in } \mathcal{P}' \text{ if and only if } \ p_1A \overset{a}{\hookrightarrow} p_2w \text{ in } \mathcal{P} \text{ and } q_1 \xrightarrow{a} q_2 \text{ in } \mathcal{A}_L.$$

   We compute $\mathit{pre}^*[L](C)$ as follows. Let C' be the set of configurations $(q, q_f)w$ in $\mathcal{P}'$ such that $qw \in C$ and $q_f \in F$. Then $\mathit{pre}^*[L](C)$ is obtained by computing $\mathit{pre}^*(C') \cap (P \times I) \times \Gamma^*$.

2. Suppose that C is given by a finite automaton $\mathcal{A}_C$ having n_C states. The initial $\mathcal{P}$-automaton corresponding to C' has at most $n_C n_L$ states and the size of the transition system of $\mathcal{P}'$ is at most $|\Delta'| = |\Delta||\Delta_L|$. Thus the time taken to compute the $\mathcal{P}$-automaton for $\mathit{pre}^*(C')$ is $n_C^2 n_L^2 |\Delta||\Delta_L|$. One then just needs to restrict the initial states for this $\mathcal{P}$-automaton to those in $P \times I$.

2 Dickson's Lemma

1. Easy.

2. First note that $(\mathbb{N}, \leq)$ is a wqo: $\leq$ is a total ordering over $\mathbb{N}$, thus $n_i \not\leq n_j$ implies $n_i > n_j$, and any strictly decreasing sequence $n_0 > n_1 > \cdots$ over $\mathbb{N}$ is finite of length at most $n_0 + 1$.

   The set $(\mathbb{N} \uplus \{\omega\}, \leq)$ is also totally ordered. Consider a sequence $n_0 > n_1 > \cdots$ over $\mathbb{N} \uplus \{\omega\}$. If for some index i, $n_i = \omega$, then $i = 0$ and for all $j > 0$, $n_j < \omega$, thus the sequence $n_1 > n_2 > \cdots$ is over $\mathbb{N}$ and is finite.

3. We start with the following claim.

   Claim 1. *Let $(a_n)_{n\geq 0}$ be a sequence that does not contain any non-decreasing subsequence. There exist $i_0 < i_1 < i_2 < \dots$ such that $a_{i_0} \not\leq a_{i_k}$ for all k.*

   Proof. We suppose on the contrary that for all $i_0 < i_1 < \dots$, there exists $k > 0$ such that $a_{i_0} \leq a_{i_k}$. We can then easily construct an non-decreasing subsequence $a_{j_0} \leq a_{j_1} \leq \dots$ inductively: starting with $a_{j_0} = a_0$, given a_{j_ℓ}, we can always find $a_{j_{\ell+1}}$ such that $a_{j_\ell} \leq a_{j_{\ell+1}}$. This contradicts the condition that there is no non-decreasing subsequence of $(a_n)_{n\geq 0}$. □

   Suppose $(a_n)_{n\geq 0}$ is a sequence that does not contain any non-decreasing subsequence. We will construct an infinite sequence that violates the wqo condition. A first application of the claim gives a subsequence $(a_{i_j})_{j\geq 0}$ such that $a_{i_0} \not\leq a_{i_j}$ for all $j > 0$. As $(a_{i_j})_{j\geq 0}$ cannot be non-decreasing we apply the claim once again to $(a_{i_j})_{j\geq 1}$ (note the $j \geq 1$ instead of $j \geq 0$), and so on. This gives us a way of obtaining an non-decreasing subsequence of $(a_n)_{n\geq 0}$; a contradiction.

   Alternative proof that is direct: Fix the infinite sequence $a_0 a_1 \cdots$ and consider the set $M = \{i \in \mathbb{N} \mid \forall j > i, a_i \not\leq a_j\}$. If M were infinite, then the infinite sequence $a_{i_0} a_{i_1} \cdots$ for $i_0 < i_1 < \cdots$ in M would verify in particular $a_{i_j} \not\leq a_{i_k}$ for all $j < k$, contradicting the fact that $(A, \leq)$ is a wqo. Thus M is bounded and any a_i with $i > \max M$ selects an element a_j with $j > i > \max M$ and $a_i \leq a_j$, i.e. can start an infinite increasing subsequence.

4. For any sequence $((a_n, b_n))_{n\geq 0}$, by the previous question, we can choose a subsequence $((a_{i_n}, b_{i_n}))_{n\geq 0}$ such that $(a_{i_n})_{n\geq 0}$ is non-decreasing. Finally by the definition of a wqo, one can find m, ℓ such that $b_{i_m} \leq b_{i_\ell}$.

3 Data-flow Analysis

1. We show how to convert the given example into a pushdown system. The general approach follows the same idea.

$$\begin{array}{ll}
\langle s, e_{init}\rangle \xhookrightarrow{x=5} \langle s, n_1\rangle & \langle s, e_p\rangle \xhookrightarrow{y=2x} \langle s, n_3\rangle \\
\langle s, n_1\rangle \xhookrightarrow{call(p)} \langle s, e_p n_2\rangle & \langle s, n_3\rangle \xhookrightarrow{x=0} \langle s, x_p\rangle \\
\langle s, n_2\rangle \xhookrightarrow{skip} \langle s, x_{init}\rangle & \langle s, x_{init}\rangle \xhookrightarrow{return} \langle s, \varepsilon\rangle \\
\langle s, n_1\rangle \xhookrightarrow{skip} \langle s, x_{init}\rangle & \langle s, x_p\rangle \xhookrightarrow{return} \langle s, \varepsilon\rangle
\end{array}$$

2. We can construct a regular automaton from the flow graph as follows. The set of states is N. The set of transitions is initially given by the edges in E. We then modify it as follows: For edges in E of the form $(n_1, call(p), n_2)$, we remove the transition $n_1 \xrightarrow{call(p)} n_2$ and add the transitions $n_1 \xrightarrow{call(p)} e_p$ and $x_p \xrightarrow{\varepsilon} n_2$. This corresponds to our intuition for how a program execution proceeds. Finally, the node n is defined as the initial state and n' is defined as the final state of the automaton.

   The language of the automaton intersected with the language $(A \setminus D_v)^* R_v$ describes the sequences of actions that can happen between n and n'.

3. Let $L_{n,n'}$ be the regular language obtained in the last question. Define $L = \bigcup_{n,n' \in N} L_{n,n'}$. Then the set of nodes where n is live is obtained from $pre^*[L](Q \times \Gamma^*)$ by determining the initial states of its $\mathcal{P}$-automaton from which the final states are reachable.