# Problem Set for Tutorial 7 — TDDD14/TDDD85

## 1 Pushdown Automata

**Exercise 1.** Consider the following PDA $M$, where

**States:** $\{q_0, q_1, q_2\}$

**Input alphabet:** $\{a, b\}$

**Stack alphabet:** $\{a, \bot\}$

**Initial state:** $q_0$

**Initial stack symbol:** $\bot$

**Final states:** $\{q_2\}$

and the transition relation is

$$\begin{aligned}\delta = \{\ & ((q_0, a, \bot), (q_0, a\bot)),\\ & ((q_0, a, a), (q_0, aa)),\\ & ((q_0, b, a), (q_1, \epsilon)),\\ & ((q_1, b, a), (q_1, \epsilon)),\\ & ((q_1, \epsilon, \bot), (q_2, \epsilon))\quad \}\end{aligned}$$

Find the configurations that describe the actions of the automaton when the following strings are used as input. For each string, also state whether $M$ accepts it by 1) final state and 2) empty stack.

1. $aa$
2. $aabba$
3. $aaabbb$

Is $M$ a deterministic pushdown automaton? In other words, is its next configuration relation a (partial) function?

**Exercise 2.** Construct a deterministic PDA accepting the language $\{a^i b^j \mid 0 \leq i < j\}$.

**Exercise 3.** For each of the following CFG's, construct a PDA which accepts the language generated by the CFG in question. ($S$ is the start symbol, as usual.)

1. $S \Rightarrow aAA$
   $A \Rightarrow aS \mid bS \mid a$

2. $S \Rightarrow aA \mid aBB$
   $A \Rightarrow Ba \mid Sb$
   $B \Rightarrow bAS \mid \epsilon$

## Solutions

**Solution to Exercise 1.** We need to find configurations and acceptance for the strings $aa$, $aabba$, and $aaabbb$. Also, we will determine if $M$ is deterministic.

1. **String:** $aa$
   - $(q_0, aa, \bot) \to (q_0, a, \bot) \to (q_0, a\bot) \to (q_0, \epsilon, aa\bot)$.
   - The final configuration is thus $(q_0, \epsilon, aa\bot)$, and we do not accept since $q_0$ is not accept, and since the stack is not empty.
2. **String:** $aabba$
   - $(q_0, aabba, \bot) \to (q_0, abba, a\bot) \to (q_0, bba, aa\bot) \to (q_1, ba, a\bot) \to (q_1, a, \bot) \to (q_2, a, \bot)$.
   - The final configuration is $(q_2, a, \bot)$. But we do not accept the string since the input string is not exhausted. Formally, recall that in order to accept a string we require a final configuration of the form $(\_, \varepsilon, \_)$.
3. **String:** $aaabbb$
   - $(q_0, aaabbb, \bot) \to (q_0, aabbb, \bot) \to (q_0, abbb, aa\bot) \to (q_0, bbb, aaa\bot) \to (q_0, bb, aa\bot) \to (q_1, b, a\bot) \to (q_1, \varepsilon, \bot) \to (q_2, \varepsilon, \varepsilon)$.
   - The final configuration is $(q_2, \varepsilon, \bot)$ and we accept by final state or empty stack.
4. **Determinism Check:** Yes, it is deterministic, if $((p, x, y), (q, s)) \in \delta$ and $((p, x, y), (q', s')) \in \delta$ then $(q, s) = (q', s')$. Note that if we are in state $q_1$ and read a $b$ then we loop if the head of the stack is $a$, and we are only allowed to proceed to $q_2$ with an $\varepsilon$-transition if the head of the stack is $\bot$.

**Solution to Exercise 2.** Our automaton is $M = (\{q_0, q_1, q_2\}, \{a, b\}, \{a, \bot\}, \delta, q_0, \bot, \{q_2\})$, where $\delta$ is defined below. It accepts $\{a^i b^j \mid 0 \le i < j\}$ by final state. The role of the states is described by

| state | consumed input | stack | |
|---|---|---|---|
| $q_0$ | $a^i$ | $a^i\bot$ | where $i \ge 0$ |
| $q_1$ | $a^i b^j$ | $a^{i-j}\bot$ | where $i \ge j > 0$ |
| $q_2$ | $a^i b^i b^k$ | $\bot$ | where $i \ge 0$, $k > 0$ |

$$
\begin{aligned}
\delta = \{ \quad & ((q_0, a, \bot), (q_0, a\bot)), && ((q_1, b, a), (q_1, \epsilon)), \\
& ((q_0, b, \bot), (q_2, \bot)), && ((q_1, b, \bot), (q_2, \bot)), \\
& ((q_0, a, a), (q_0, aa)), && ((q_2, b, \bot), (q_2, \bot)), \\
& ((q_0, b, a), (q_1, \epsilon)) && \}
\end{aligned}
$$

**Solution to Exercise 3.** The idea is to use the stack to simulate a leftmost derivation of the grammar. If the PDA has read $w$ from the input, the stack contains $\gamma\perp$ and the state is $q_1$ then $S \Rightarrow^* w\gamma$.

1. $M = (\{q_0, q_1, q_2\}, \{a, b\}, \{a, b, S, A, \perp\}, \delta, q_0, \perp, \{q_2\})$, where

$$\begin{aligned}\delta = \{\ & ((q_0, \epsilon, \perp), (q_1, S\perp)),\\ & ((q_1, \epsilon, S), (q_1, aAA)),\\ & ((q_1, \epsilon, A), (q_1, aS)),\\ & ((q_1, \epsilon, A), (q_1, bS)),\\ & ((q_1, \epsilon, A), (q_1, a)),\\ & ((q_1, a, a), (q_1, \epsilon)),\\ & ((q_1, b, b), (q_1, \epsilon)),\\ & ((q_1, \epsilon, \perp), (q_2, \epsilon))\ \ \}.\end{aligned}$$

2. $M = (\{q_0, q_1, q_2\}, \{a, b\}, \{a, b, S, A, B, \perp\}, \delta, q_0, \perp, \{q_2\})$, where

$$\begin{aligned}\delta = \{\ & ((q_0, \epsilon, \perp), (q_1, S\perp)), && ((q_1, a, a), (q_1, \epsilon)),\\ & ((q_1, \epsilon, S), (q_1, aA)), && ((q_1, b, b), (q_1, \epsilon)),\\ & ((q_1, \epsilon, S), (q_1, aBB)), && ((q_1, \epsilon, \perp), (q_2, \epsilon)),\\ & ((q_1, \epsilon, A), (q_1, Ba)),\\ & ((q_1, \epsilon, A), (q_1, Sb)),\\ & ((q_1, \epsilon, B), (q_1, bAS)),\\ & ((q_1, \epsilon, B), (q_1, \epsilon))\ \ \}\end{aligned}$$

Both automata accept by final state.

## 2 The Pumping Lemma for Context-Free Languages

**Exercise 4.** Show that the following languages are not context-free:

1. $L_1 = \{\, a^j b^k a^l \mid 0 < j < k < l \,\}$,
2. $L_2 = \{\, w \in \{a, b, c\}^* \mid$ $w$ has an equal number of $a$'s, $b$'s and $c$'s $\}$,
3. $L_3 = \{\, ww \mid w \in \{a, b\}^* \,\}$.

## Solutions

**Solution to Exercise 4.**

1. Assume that $L_1$ is context-free. Then the pumping lemma holds. According to the lemma, there exists a number $n$ such that if a string $z$, not shorter than $n$, is in $L_1$ (i.e. $|z| \geq n$, $z \in L_1$) then $z$ can be split into five strings $u, v, w, x, y$:

   $$z = uvwxy$$

   such that

   $$|vx| \geq 1, \qquad |vwx| \leq n, \qquad uv^i wx^i y \in L_1 \text{ for all } i \geq 0$$

   We show that this leads to a contradiction. Take

   $$z = a^n b^{n+1} a^{n+2} \in L_1.$$

   Then there exist strings $u, v, w, x, y$ satisfying the conditions above. We have two possibilities:

   (a) $vwx$ does not overlap with the initial $a^n$. In other words, $u = a^n u'$ (for some $u'$). Take $i = 0$. Then $uv^0 wx^0 y = uwy = a^n b^k a^l$, for some $k$ and $l$. The string $a^n b^k a^l$ is shorter than $a^n b^{n+1} a^{n+2}$ (as $|vx| \geq 1$), hence $k < n + 1$ or $l < n + 2$ (or both). In both cases $n < k < l$ is impossibe, so $uwy \notin L_1$ and we have a contradiction.

   (b) Otherwise $vwx$ overlaps with the initial $a^n$. So it does not overlap with the final $a^{n+2}$, as $|vwx| \leq n$. In other words $y = y' a^{n+2}$, for some $y'$. Take $i = 2$. If $uv^2 wx^2 y$ is not of the form $a^j b^k a^l$ then $uv^2 wx^2 y \notin L_1$, contradiction. If $uv^2 wx^2 y = a^j b^k a^l$ then $l = n + 2$ but $j > n$ or $k > n + 1$ (as $|vx| \geq 1$). Thus $j < k < l$ does not hold and $uv^2 wx^2 y \notin L_1$. Contradiction.

   This completes the proof. As usually in proofs with pumping lemmas, choosing an appropriate string $z$ was crucial. For instance, if $z = ab^n a^{2n}$ then we may take $v = w = \epsilon$, $x = a$, $u = ab^n$ and and we do not obtain contradiction.

2. We know (from the lecture) that the language $M = \{ a^i b^i c^i \mid i \geq 0 \}$ is not context-free. Notice that $L_2 \cap a^* b^* c^* = M$. Remember that the intersection of a context-free language with a regular language is context-free. So if $L_2$ were context-free then $M$ would also be context-free. The latter is not true, so $L_2$ is not context-free.

   (A proof using the pumping lemma is also possible; take for instance $z = a^n b^n c^n$).

3. To show that $L_3$ is not a context-free language, we show that the pumping lemma does not hold for $L_3$. To do this, for every number $n$ we have to find a string $z \in L_3$, $|z| \geq n$ such that for every splitting of $z$ into five pieces

   $$z = uvwxy, \quad \text{where } |vx| \geq 1 \text{ and } |vwx| \leq n$$

   some of the strings $uv^i wx^i y$ ($i = 0, 1, \ldots$) are not in $L_3$.

Let us try with

$$z = a^n b^n a^n b^n \in L.$$

Consider an arbitrary splitting as above. If $|vx|$ is odd then $uv^0wx^0y$ has an odd length and thus is not in $L_3$. So it remains to consider the case of $|vx|$ being even. Notice that $3n \leq |uv^0wx^0y| \leq 4n - 1$. We have three possibilities:

(a) $vwx$ is contained in the first half of $z$ (so $y = y'a^n b^n$, for some $y'$). Then the last symbol of the first half of $uv^0wx^0y$ is $a$ (we removed some symbols from the first half of $z$ and "the middle moved to the right"). Thus $uv^0wx^0y$ is not in $L_3$ (as the last symbol of its second half is $b$).

(b) $vwx$ is contained in both halves of $z$. So it begins with a $b$ and ends with an $a$. Thus $v$ contains (at least one) $b$ or $x$ contains (at least one) $a$. Thus the first half of $uv^0wx^0y$ has fewer $b$'s than the second one[a] or the second half of $uv^0wx^0y$ has fewer $a$'s than the first one. Hence $uv^0wx^0y$ is not in $L_3$.

(c) $vwx$ is contained in the second half of $z$ (so $u = a^n b^n u'$, for some $u'$). This case is symmetric to case 3a. The first symbol of the second half of $uv^0wx^0y$ is $b$ and $uv^0wx^0y \notin L_3$.

[a] As the halves of $uv^0wx^0y$ are not shorter than $1.5n$, the first half begins with $a^n$ and the second half ends with $b^n$.

## 3 Advanced and Exam Like Exercises

**Exercise 5.** Consider the PDA $\langle Q, \Sigma, \Gamma, \delta, s, \bot, F\rangle$ where

- $Q = \{q_0\}$,
- $\Sigma = \{a, b\}$,
- $\Gamma = \{a, \bot\}$,
- $\delta = \{((q_0, a, \bot), (q_0, a\bot)), ((q_0, a, a), (q_0, aa)), ((q_0, b, a), (q_0, \varepsilon))\}$,
- $s = q_0$ is the start state,
- $\bot$ is the initial stack symbol, and
- $F = \{q_0\}$ is the set of final states.

The machine accepts by final state.

1. Which of the following strings over $\{a, b\}$ are accepted by the PDA?
   (a) $aa$.
   (b) $ba$.
   (c) $aabb$.
   (d) $abba$.
   (e) $abab$.
2. What is the language recognized by the PDA?

**Exercise 6.** Prove that the language $L = \{a^i b^j c^{ij} \mid 0 < i < j\}$ is not context-free by using the pumping lemma for context-free languages.

## Solutions

**Solution to Exercise 5.**

1. $aa$, $aabb$, and $abab$.
2. All strings over $\{a, b\}$ where each occurence of a $b$ has a matching $a$ somewhere to the left in the string.

**Solution to Exercise 6.** We provide a proof sketch which illustrates the most important case distinctions but does not go into every case in detail.

To prove that the language $L = \{a^i b^j c^{ij} \mid 0 < i < j\}$ is not context-free, we use the (inverted) pumping lemma for context-free languages. Let $p \geq 1$ be arbitrary and consider the string $s = a^p b^{p+1} c^{(p)(p+1)} \in L$. Let $s = uvwxy$ be a partitioning where $|vwx| \leq p$ and $|vx| \geq 1$. We observe that since $|vwx| \leq p$, $vwx$ can contain at most two different types of symbols (either $a$s and $b$s, or $b$s and $c$s).

We then consider three cases based on $vwx$:

1. $vwx$ consists of a single symbol): in this case $uv^2wx^2y$ is not included in the language.
2. $vwx$ consists of only $a$'s and $b$'s: then pumping $v$ and $x$ changes the number of $a$'s or $b$'s, but does not change the number of $c$'s accordingly. Thus, $uv^2wx^2y$ would have either more $a$'s or $b$'s without the corresponding number of $c$'s to match the condition $c^{ij}$, where $i$ and $j$ are the counts of $a$ and $b$ respectively.
3. $vwx$ consists of only $b$'s and $c$'s: then pumping $v$ and $x$ changes the number of $b$'s or $c$'s without the corresponding adjustment to $a$'s. If $p = 1$ then we choose $i = 0$ since then $uv^0wx^0y$ is not included in the language. For $p > 1$ one can prove that we can always find $i \geq 2$ such that $uv^iwx^iy$ is not in the language.