Int 201: Decision Computation and Language
Tutorial 8 Solution

Dr. Chunchuan Lyu

November 23, 2023

Question 1. A deterministic PDA have deterministic transition function instead of relations. Can the palindrome language $\{ww^R | w \in \{0,1\}^*\}$ be recognized by a deterministic PDA? (no need for a proof)

Solution 1. No, it cannot. On a intuitive level, as the string we could accept can be arbitrarily long, the memory has to be stored in the stack. This suggests, the DPDA never pops thing out of the stack. However, if this is the case, the DPDA lost access to the elements down the stack, so it cannot know its' a palindrome. On contrary, non-deterministic PDA can have branches that pop out at anytime.

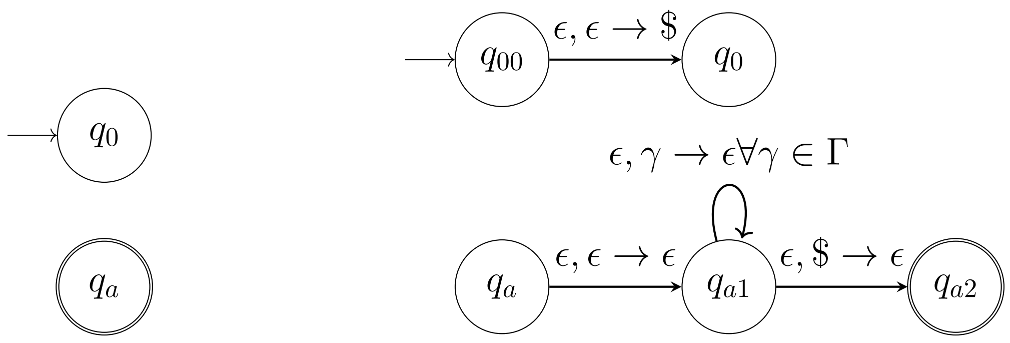
See 5.16 in "Introduction to Languages and the Theory of Computation" by John C. Martin. for a slightly more formal proof.

Question 2. Show that all languages recognized by a regular PDA can be recognized by a PDA that has only one accepting state.

Solution 2. This can be achieved by removing the accepting conditions, and add a new accepting state to which all old accepting states admit an $\epsilon, \epsilon \to \epsilon$ transition.

Clearly, for all strings recognized by the regular PDA, it can transit to the new accepting state in the new PDA. So, $L(PDA_{old}) \in L(PDA_{new})$. On the other hand, all strings accepted by the new pda has to come from an old accepting state through those $\epsilon, \epsilon \to \epsilon$ transitions. So, $L(PDA_{new}) \in L(PDA_{old})$

Question 3. Show that all languages recognized by a regular PDA can be recognized by a PDA that accepts when both its' stack is empty and its' in an accepting state.



q_0
q_a
q_{00}
$\epsilon, \epsilon \to \$$
q_0
$\epsilon, \gamma \to \epsilon \forall \gamma \in \Gamma$
q_a
$\epsilon, \epsilon \to \epsilon$
q_{a1}
$\epsilon, \$ \to \epsilon$
q_{a2}

Figure 1: Left side old PDA; right side PDA accepting in empty stack

Solution 3. As shown in Figure 1, we essentially add a new end of stack symbol and pop everything out after old accepting state.

Clearly, for all strings recognized by the old regular PDA, there is at least one branch of computation where the stack symbol \$ is not be touched during the process of the new PDA. Otherwise, such string would have been rejected by the old PDA for poping when the stack is empty. So, we can enter one of the old accepting state q_a, and consequently enter q_{a2} after popping out everything. We have $L(PDA_{old}) \in L(PDA_{new})$.

On the other hand, to enter one of the new accepting states q_{a2}, the string would have to enter through old accepting state q_a with a \$ at the bottom of the stack that is never popped during processing. This means in the old PDA, the string also went from q_0 to q_a and hence being accepted. So, we have $L(PDA_{new}) \in L(PDA_{old})$.

Question 4. Show that all languages recognized by a regular PDA can be recognized by a PDA where all transitions either push or pop the stack but not do both.

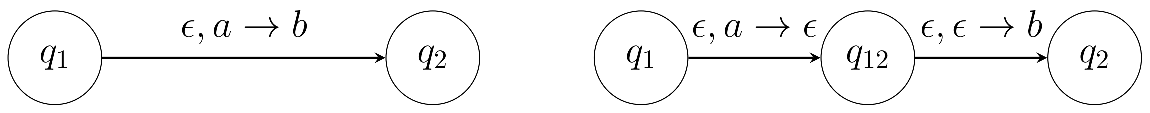


Figure 2: Left side old PDA; right side PDA that don't push and pop at the same time

Solution 4. We replace every pop and push transition with an pop transition and push transition through an intermediate state. Clearly, all transitions that went from q_1 to q_2 starts and ends with the same stack content in both PDAs.

Question 5. Convert this PDA to CFG.

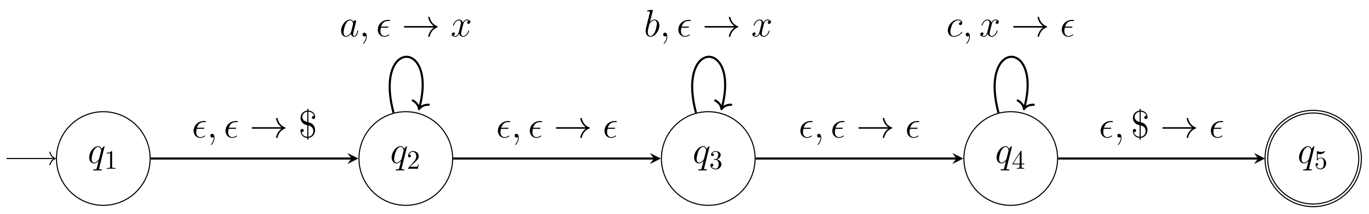

$a, \epsilon \to x$
$b, \epsilon \to x$
$c, x \to \epsilon$
$\epsilon, \epsilon \to \$$
$\epsilon, \epsilon \to \epsilon$
$\epsilon, \epsilon \to \epsilon$
$\epsilon, \$ \to \epsilon$
q_1
q_2
q_3
q_4
q_5

Figure 3: PDA for Q6

Solution 5. We could use the construction in the equivalence relation. Here, we list only the symbols that are useful.$S = A_{15}$

- $\forall i, A_{ii} \to \epsilon$
- $A_{15} \to A_{24}$
- $A_{24} \to aA_{24}c|A_{23}A_{34}$
- $A_{23} \to A_{33}$
- $A_{34} \to bA_{34}c|A_{44}$

Or we understand this PDA generates $\{a^n b^m c^{m+n} | m, n \geq 0\}$, and write

- $S \to aSc|B$
- $B \to bBc|\epsilon$

Question 6. Show that the language $\{ww|w \in \{0,1\}^*\}$ is not context-free by using the pumping lemma.

Solution 6. Proof by contradiction: Assume $\{ww|w \in \{0,1\}^*\}$ is CFL, pumping lemma says that there is a pumping length p. Take $s = 0^p 1^p 0^p 1^p$, clearly s is in the language, and $|s| > p$. Let us show none of the decomposition of $s = uvxyz$ will satisfy the pumping lemma. Again, the key is to rely on $|vxy| \leq p$. Clearly, if vxy lies entirely on the left or right part of s, the pumping will destroy the symmetry. So, we must have

- $u = 0^p 1^{p-m}$

- $vxy = 1^m0^n$
- $z = 0^{p-n}1^p$

for $m+n \leq p$. If we are going to pump down to uxz, for the middle part, we either lose some 1 or lose some 0, as $|vz| \geq 1$. In other words, we end up with $uxz = 0^p1^i0^j1^p$, where either $i < p$ or $j < p$. So, we must either mismatch the 0s or 1s, and hence break the symmetry. So, we get a contradiction.