Int 201: Decision Computation and Language
Tutorial 7

Dr. Chunchuan Lyu

November 7, 2025

Question 1. Draw PDA for language $\{ww^R|w \in \{0,1\}^*\}$, where w^R is reversed of w, and show the language and L(PDA) (strings being accepted by PDA) are equivalent. [1]

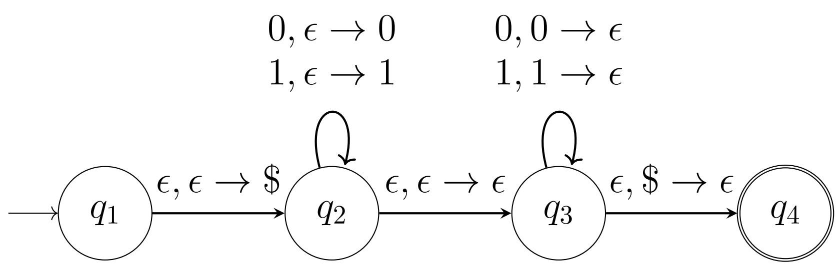

Figure 1: PDA for Q1

Solution 1. We'll prove $L(PDA) = \{ww^R|w \in \{0,1\}^*\}$ by showing both containments.

($\subset$) Let s be any string accepted by the PDA. The acceptance path must:

Push \$ (marker) in $q_1 \to q_2$ Push first part of input in q_2 Move to q_3 without reading input Pop and match second part in q_3 Pop \$ to accept Due to this structure:

Let the first part be a (pushed in q_2) Let the second part be b (matched in q_3) $|a| = |b|$ (since stack must be empty except \$ before q_3) $b[i] = a[|a|-1-i]$ (due to stack LIFO property) Therefore, $b = a^R$ and $s = aa^R \in \{ww^R|w \in \{0,1\}^*\}$

($\supset$) For any $s = ww^R$, the PDA accepts it as follows:

[1]Sets A and B being equivalent means $\forall x, x \in A \implies x \in B$ and $\forall x, x \in B \implies x \in A$

Push \$ marker Push all of w onto stack Enter q_3 when w is complete Match w^R by popping stack (possible since stack contains reversed w) Pop \$ and accept Therefore, $L(PDA) = \{ww^R | w \in \{0,1\}^*\}$

Question 2. Can you draw PDA for language $\{ww | w \in \{0,1\}^*\}$? Can a PDA with two stacks recognize the language $\{ww | w \in \{0,1\}^*\}$? If the answer is yes for any of the questions, draw the PDA, no need for a proof. If the answer is no, give some intuitive explanation. (proof is better, but not necessary).

Solution 2. No, we can't. Intuitively, the string can be arbitrary long, so we need stack for memory, but stack can only be accessed through a last in first out fashion. It cannot recall the first element without popping out everything in between. We will need pumping lemma (next lecture) and PDA CFL equivalence to show this is indeed not CFL.

Yes, we can, as we can reverse the stack element with the second stack. Note that with two stacks, the transition becomes a 5-tuple (tape,pop stack1,pop stack2$\rightarrow$ push stack1, push stack2). We use the following PDA with two stacks:

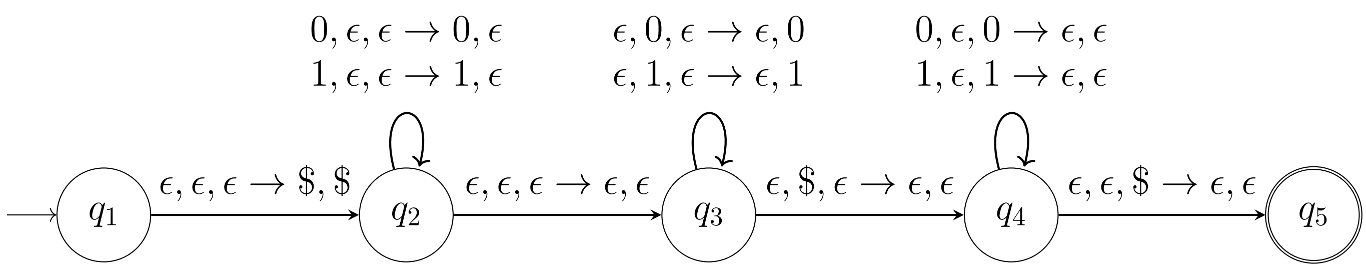

Figure 2: PDA for Q2

Basically, in q_3, we pop all elements from the first stack to the second, essentially simulating a queue with two stacks. This works because the PDA has non-deterministic transition.
[2]

[2]If my memory is correctly, the solution we worked out in the tutorial is different, and I believe that version is incorrect, due to the fact reversed the \$ will be on top of the second stack. Really sorry for missing that.

Question 3. Draw PDA P for language $L = \{a^i b^j c^k | i, j, k \geq 0, i = j \text{ or } j = k\}$, and show the $L = L(P)$. [3]

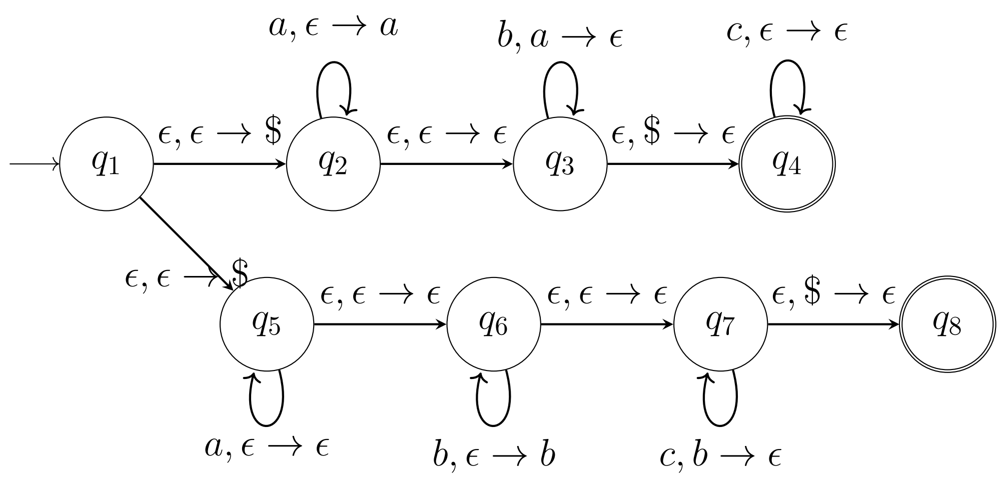

Figure 3: PDA for Q3

Solution 3. Consider the PDA shown in the figure with two main paths: upper path ($q_1 \to q_4$) and lower path ($q_1 \to q_8$).

Let's prove $L(PDA) = \{a^i b^j c^k | i, j, k \geq 0, i = j \text{ or } j = k\}$

($\subset$) For any string accepted by the PDA:

If accepted via q_4 (upper path): Must push i (we don't know what i is, but there is a number) a's using $a, \epsilon \to a$ in q_2 Must pop j b's using $b, a \to \epsilon$ in q_3. Because, pop is possible, so we must have $j \leq i$. Then, we have $j \geq i$ (to see \$ through $\epsilon, \$ \to \epsilon$). [4] In all, we have $j = i$. Last, we can read any number k of c's in q_4 Therefore string is $a^i b^j c^k$ with $i = j$ If accepted via q_8 (lower path): We can read any number i of a's in q_5 Must push j b's using $b, \epsilon \to b$ in q_6 Must pop k c's using $c, b \to \epsilon$ in q_7 with $k = j$ (for same reason as in the upper branch). Therefore string is $a^i b^j c^k$ with $j = k$ Therefore, $L(PDA) = \{a^i b^j c^k | i, j, k \geq 0, i = j \text{ or } j = k\}$

($\supset$) For any string $a^i b^j c^k$ where $i = j$ or $j = k$:

If $i = j$: Use upper path Push i a's in q_2 Match j b's with a's in q_3 ($j = i$) Pop \$ and accept remaining c's in q_4 If $j = k$: Use lower path Skip a's in q_5 Push j b's in q_6 Match k c's with b's in q_7 ($k = j$) Pop \$ and accept in q_8

[3] Note that this is different from the pda in the lecture

[4] Some students told me that I wrote $\geq$ or maybe $\leq$ for both direction, that's a typo, thanks for catching.

Question 4. Complete the proof for the Kleene closure property of CFL. In the sense, that the Kleen closure of language and the language being accepted by the constructed CFG grammar is the same set.

Solution 4. Let L_1 be a CFL with CFG $G_1 = (V_1, \Sigma_1, R_1, S_1)$. Construct $G_2 = (V_1, \Sigma_1, R_1 \cup S \to S_1S|\epsilon, S)$ where S is a new start symbol.

We prove $L(G_2) = L_1^*$ by showing both containments:

($\subset$) Let $w \in L(G_2)$:

If $w = \epsilon$: directly from $S \to \epsilon$, and $\epsilon \in L_1^*$ If $w \neq \epsilon$: Any derivation must use $S \to S_1S$ repeatedly Each S_1 generates some string $w_i \in L_1$ Final S must derive ϵ Thus $w = w_1w_2...w_n$ where each $w_i \in L_1$ Therefore $w \in L_1^*$

($\supset$) Let $w \in L_1^*$:

If $w = \epsilon$: directly derivable using $S \to \epsilon$ If $w = w_1w_2...w_n$ where each $w_i \in L_1$: Use $S \to S_1S$ n times: $S \Rightarrow S_1S \Rightarrow S_1S_1S \Rightarrow ... \Rightarrow S_1S_1...S_1S$ Each S_1 can derive corresponding w_i using rules from R_1 Final S derives ϵ Therefore $w \in L(G_2)$ Therefore, $L(G_2) = L_1^*$, proving the Kleene closure of a CFL is context-free.

(Alternatively, see Waterloo notes for a more lengthy version.)

Xi'an Jiaotong-Liverpool University 西交利物浦大学

Question 5 (Optional)**.** Given the CFG $G = (V, \Sigma, R, S)$:

- V = {S, NP,VP,Det,Nominal,Noun,PP,Preposition,Verb}
- Σ = The, spy, saw, cop, with, a , telescope
- Rules

  S → NP VP

  NP → Det Nominal

  Nominal→ Noun ‖ Nominal PP

  VP→ VP PP‖ Verb NP

  PP → Preposition NP

  Det → The ‖ a

  Noun → spy ‖ cop‖telescope

  Verb → saw

  Preposition → with

Is there a third derivation other than what we found in the lecture for The spy saw a cop with a telescope ?

Solution 5. No, there isn't. The interesting thing is that one might interpret the sentence with the meaning where The spy is with a telescope at hand, but he did not use the telescope to see the cop. However, the restricted grammar structure of CFG will not allow this interpretation. Let's stop here before we sink into the rabbit hole of syntaxsemantics interface. To prove there is no other parse requires some understanding of CFG parsing being solvable by dynamic programming. One can refer to the NLP textbook that explains the CYK algorithm. Here is some code that find all parses.

Question 6 (Optional)**.** Have fun with the notebook. https://github.com/ND-CSE-30151/spring-2024/blob/main/notes/12-pdas.ipynb