Computational Models - Exercise #3 solution sketch

1. $L = \left\{ a^i b^j c^k : i > 0 \wedge i > j + k \right\}$. We will prove $L = L(G)$.

   - Let $w \in L$, so there exists $i > 0$, $j, k \geq 0$ so that $w = a^i b^j c^k$ so that $i > j + k$. By applying $S \rightarrow aSc$ for k times, we have $S \rightarrow^\star a^k S c^k$. Now, apply $S \rightarrow A$, so $S \rightarrow^\star a^k A c^k$. Next, apply $A \rightarrow aAb$ for j times and obtain $S \rightarrow^\star a^{j+k} A b^j c^k$. Apply $A \rightarrow aA$ for $i - j - k - 1 \geq 0$ times and get $S \rightarrow^\star a^{i-1} A b^j c^k$. Finally, apply $A \rightarrow a$ and we get $S \rightarrow^\star w$, so $w \in L(G)$.
   - Let $w \in L(G)$. Before proving $w \in L$, let us first prove an auxiliary claim:

     Claim 1 *$L(G_A)$, the grammar restricted to the variable A alone, is contained in $L' = \left\{ a^i b^j : i > 0, i > j \right\}$.*

     Let $s \in L(G_A)$. The proof is by induction on the number of derivation steps used to derive s from A.

     – For a single derivation, the only option is $A \rightarrow a$ and indeed $a\ inL'$.

     – Assume correctness holds for derivations of length $k - 1 \geq 1$ and say $A \rightarrow^\star s$ in k derivation steps. There are two possibilities for the first derivation:

       (a) The first derivation is $A \rightarrow aAb$. Hence, $s = as'b$ and s' is derived from A using $k-1$ steps. By the hypothesis, $s' = a^i b^j$ for some $i > 0$ and $i > j$ so $s = a^{i+1} b^{j+1}$ and still $s \in L'$.

       (b) The first derivation is $A \rightarrow aA$ – very similar to the first item.

     The main proof is again by induction on the number of derivation steps used to derive w from S.

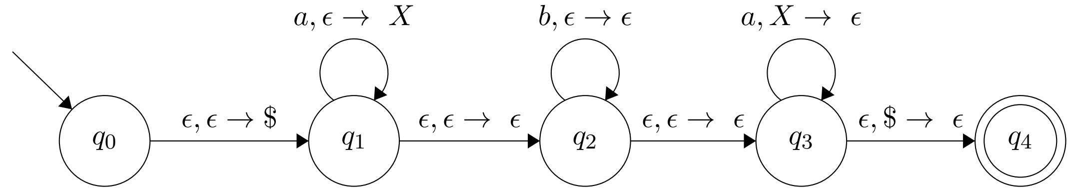
     – A single derivation is not possible. For two derivations, the only option is $S \rightarrow A$ and $A \rightarrow a$ and indeed $a \in L$.

     – Assume correctness holds for derivations of length $k - 1 \geq 2$ and say $S \rightarrow^\star w$ in k derivation steps. There are two possibilities:

       (a) The first derivation is $S \rightarrow aSc$. Hence, $w = aw'c$ and w' is derived from S using $k - 1$ steps. By the hypothesis, $w' = a^i b^j c^k$ for some $i > 0$ and $i > j + k$, so $w = a^{i+1} b^j c^{k+1}$ and still $w \in L$.

       (b) The first derivation is $S \rightarrow A$. By Claim 2, $A \rightarrow^\star w$ implies that $w \in L' \subseteq L$, as desired.

2. (a) The PDA for L_1:

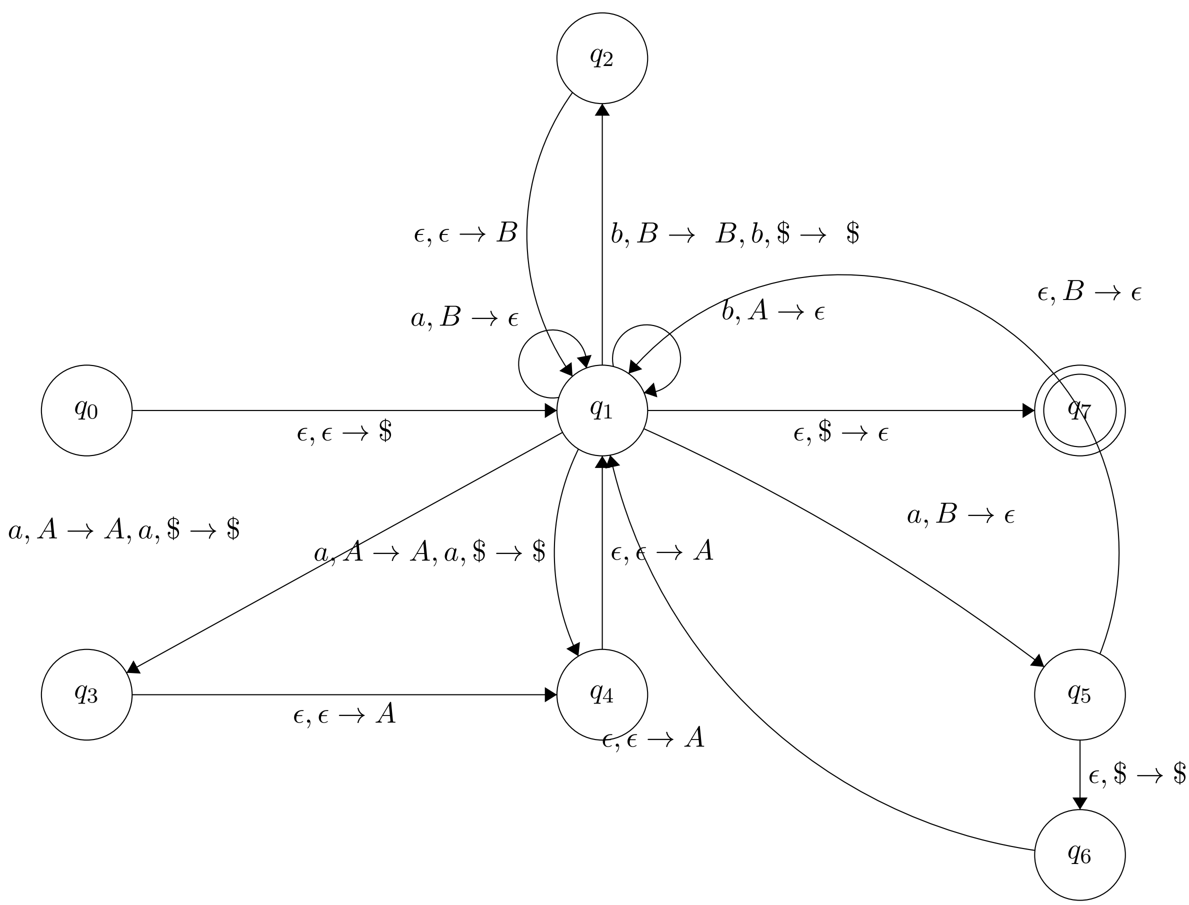


A formal definition: $P = (\{q_0, q_1, q_2, q_3, q_4\}, \{a, b, \epsilon\}, \{\$, X\}, \delta, q_0, \{q_4\})$, where:

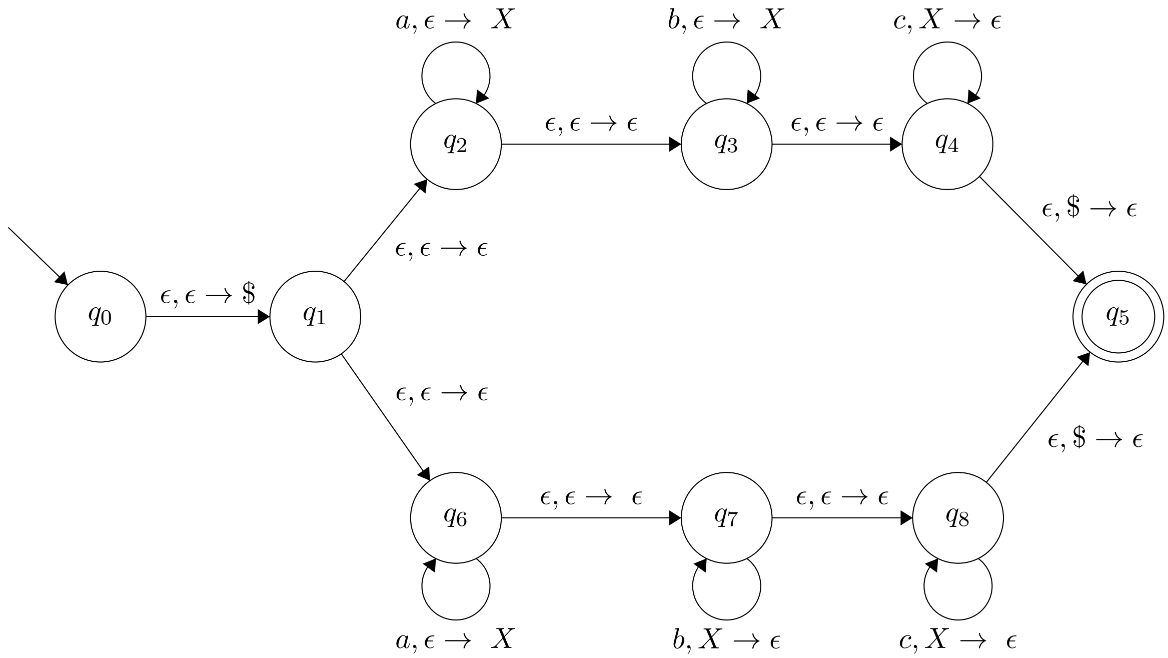
$$\begin{aligned}
\delta(q_0, \epsilon, \epsilon) &= \{(q_1, \$)\} \\
\delta(q_1, a, \epsilon) &= \{(q_1, X)\} \\
\delta(q_1, \epsilon, \epsilon) &= \{(q_2, \epsilon)\} \\
\delta(q_2, b, \epsilon) &= \{(q_2, \epsilon)\} \\
\delta(q_2, \epsilon, \epsilon) &= \{(q_3, \epsilon)\} \\
\delta(q_3, a, X) &= \{(q_3, \epsilon)\} \\
\delta(q_3, \epsilon, \$) &= \{(q_4, \epsilon)\},
\end{aligned}$$

and $\emptyset$ elsewhere.

(b) The PDA for L_2:



(c) The PDA for L_3:



3. (a) $G_4 = (\{S\}, \{a, b\}, R, S)$ where R has the single rule:

$$S \to SaSaSbS \mid SaSbSaS \mid SbSaSaS \mid \epsilon.$$

This grammar is ambiguous – find a word with two parse trees.

(b) $G_5 = (\{S, L, R, A, B\}, \{a, b, \$\}, \tilde{R}, S)$ where $\tilde{R}$ has the rules:

$$\begin{aligned} S &\to BL \mid RB \\ L &\to BL \mid A \\ R &\to RB \mid A \\ A &\to BAB \mid \$ \\ B &\to a \mid b \end{aligned}$$

In CNF, the grammar has the following rules:

$$\begin{aligned} S &\to BL \mid RB \\ L &\to BL \mid BC \mid \$ \\ R &\to RB \mid BC \mid \$ \\ A &\to BC \mid \$ \\ B &\to a \mid b \\ C &\to AB \end{aligned}$$

4. Assume to the contrary that L_6 is context-free and let ℓ be the promised pumping constant, and we can assume w.l.o.g. that $\ell \geq 10$ (why?). Take $s = a^\ell b^\ell c^\ell \in L_6$ and obviously $|s| \geq \ell$.

Consider a general decomposition $s = uvxyz$ such that $|vy| > 0$ and $|vxy| \leq \ell$. As $|vxy| \leq \ell$, the string vxy cannot contain both a-s and c-s. Obviously this holds for vy as well. We consider the two possibilities:

(a) $\#_a(vy) > 0$. Thus, $\#_c(vy) = 0$, and consider the pump $w = uv^2xy^2z$. It then holds that $\#_c(w) = \#_c(s) = \ell$. Also,

$$\#_a(w) = \#_a(s) + \#_a(vy) > \ell = \#_c(w),$$

in contradiction to $w \in L_6$.

(b) $\#_a(vy) = 0$. As $|vy| > 0$ then either $\#_b(vy) > 0$ or $\#_c(vy) > 0$. Consider the pump $w = uv^0xy^0z = uxz$. It then holds that

$$\#_a(w) = \#_a(s) - \#_a(vy) = \#_a(s) = \ell,$$

and also either $\#_b(w) = \#_b(s) - \#_b(vy) < \ell$ or in a similar way, $\#_b(w) < \ell$, in contradiction to $w \in L_6$.

We have a contradiction in each possibility, so L_6 is not context-free.

5. Let $A = (Q, \Sigma, \delta, q_0, F)$. We define $M = (\{q_i, q_M, q_f\}, \Sigma, \Gamma, \delta', q_i, \{q_f\})$ where the idea is to simulate the transitions of the DFA using the stack (as we cannot use multiple states). That is, $\Gamma = Q$ and δ' contains the following transitions:

   - $\delta'(q_i, \epsilon, \epsilon) = \{(q_M, q_0)\}$.
   - For every $q \in Q$ and $\sigma \in \Sigma$, $\delta'(q_M, \sigma, q) = \{(q_M, \delta(q, \sigma))\}$.
   - For every $q \in F$, $\delta'(q_M, \epsilon, q) = \{(q_f, \epsilon)\}$.

   We now prove that $L(A) = L(M)$.

   - Let $w \in L(M)$, so $\hat{\delta}'(q_M, w, \epsilon) \cap (F \times \Gamma^\star) \neq \emptyset$. Our stack size at every step is at most 1 so we reach q_f only with an empty stack. Thus, $(q_f, \epsilon) \in \hat{\delta}'(q_M, w, \epsilon)$ and again by our construction, this implies that there exists $q_{f,A} \in F$ such that $(q_M, q_{f,A}) \in \hat{\delta}'(q_M, w, \epsilon)$ (as the last step is an epsilon transition that pops an accepting state). We have the following claim:

     Claim 2 *If $(q_M, q) \in \hat{\delta}'(q_M, w, \epsilon)$ then $\hat{\delta}(q_0, w) = q$.*

     The proof of the claim is by induction on $|w|$ using the inductive definition of $\hat{\delta}'$ for PDAs. Fill in the details yourselves.

     Now, using the claim, as $\hat{\delta}(q_0, w) \in F$, $w \in L(A)$ and we are finished.

   - Let $w \in L(A)$, so $\hat{\delta}(q_0, w) = q_{f,A} \in F$. Note that for every $q \in Q$ and $\sigma \in \Sigma$ we can always apply the transition $(q_M, \delta(q, \sigma)) \in \delta'(q_M, \sigma, q)$ and this transition shortens the input by 1. Thus, when we are at configuration (q_M, q_0) we can apply the aforementioned transition $|w|$ times and reach a configuration (q_M, q) for some $q \in Q$. By Claim 1, $q = \hat{\delta}(q_0, w) = q_{f,A}$, so the configuration is in fact $(q_M, q_{f,A})$ and we can now apply the rule $(q_f, \epsilon) \in \delta'(q_M, \epsilon, q_{f,A})$ and so $w \in L(M)$.

6. (a) The claim is false. Let $L_1 = \{a\}^\star$ and $L_2 = \{b\}^\star$ – both are regular. However, $Even(L_1, L_2) = \left\{a^i b^i : i \geq 0\right\}$, which is not.

   (b) The claim is false. Let $L_1 = \{c\}^\star$, which is regular, and $L_2 = \left\{a^i b^i : i \geq 0\right\}$, which is context-free. It holds that

$$L = Even(L_1, L_2) = \left\{c^{2i} a^i b^i : i \geq 0\right\},$$

   and we will prove that L is not context-free. Assume towards contradiction that it is, and define a homomorphism $h : \{a, b, c\} \to \{a, b, c\}^\star$ by $h(a) = cc$, $h(b) = a$, $h(c) = b$. It holds that $h^{-1}(L) = \left\{a^i b^i c^i : i \geq 0\right\}$ (prove it), which is known to be non context-free. This contradicts the fact that CFLs are closed under inverse homomorphism.

7. (a) The claim is false. Let $L = L((ab)^\star)$, which is of course regular. However, $Sort(L) = \left\{a^i b^i : i \geq 0\right\}$ (verify this to yourself) which is not regular.

   (b) The claim is true. The idea is, given a DFA for L, to construct a CFG that imitates the DAF's derivations, however performs the derivation according to the ordering of Σ. Formally: Given a DFA $M = (Q, \Sigma = \{a, b\}, \delta, q_0, F)$, construct $G = (Q, \Sigma, R, q_0)$, where R is as follows. For every q, let $\delta(q, a) = q_1$ and $\delta(q, b) = q_2$. Then, R has the rule $q \to aq_1 \mid q_2 b$. Also, for every $q_f \in F$ we have the rule $q_f \to \epsilon$. We now prove that $Sort(L) = L(G)$.

   - Let $w = a^i b^j \in Sort(L)$. Then, there exists a word s of length $n = i + j$ such that $sort(s) = w$. As $\hat{\delta}(q_0, s) = q_f \in F$, there exist $q_1, \ldots, q_{n-1}$ such that $\delta(q_0, s_1) = q_1$, $\delta(q_1, s_2) = q_2$,..., $\delta(q_{n-1}, s_n) = q_f$. To show that $q_0 \to^\star w$ in G, simply apply the rules according to the states and the specific character. Namely, for $1 \leq i \leq n$, the i-th rule to apply is given by $q_{i-1} \to aq_i$ if $s_i = a$ and $q_{i-1} \to q_i b$ if $s_i = b$. Finally, apply $q_f \to \epsilon$. All these rules are valid, and it is easy to see that since $\#_a(s) = i$ and $\#_b(s) = j$, the derived word will indeed be $a^i b^j$, so $w \in L(G)$.
   - Let $w \in L(G)$ and assume $|w| = n$. As every derivation rule which is not of the form $q \to \epsilon$ adds exactly one literal and one variable, we get $q_0 \to^\star w$ after exactly $n + 1$ derivations, where the last one is $q_f \to \epsilon$ for some $q_f \in F$ and the first n derivations involve the set of variables $\{q_0, \ldots, q_{n-1}, q_n = q_f\}$. That is, for $1 \leq i \leq n$, the i-th derivation is either $q_{i-1} \to aq_i$ or $q_{i-1} \to q_i b$. Let s be the concatenation of the n literals that appear throughout the derivation. By the construction of G, $\delta(q_{i-1}, s_i) = q_i$, so overall $\hat{\delta}(q_0, s) = q_f$ and $s \in L$. It holds that $w = sort(s)$ (why?) so $w \in Sort(L)$ and we are finished.

   (c) The claim is false. The proof is similar to (a) by using $L = L((abc)^\star)$.

8. Note that an algorithm is a process that halts on every input and returns the correct answer.

   (a) Construct an equivalent CFG G such that $L(G) = L(M)$. The condition is equivalent to asking whether $L(G)$ is infinite, for which we saw an algorithm. We prove this claim:

   - If L is infinite there is no bound on the length of its words, and as L is also context-free then by the Pumping Lemma there exists a long enough word that satisfies those requirements.
   - Let w be such that there exists a decomposition $w = uvxyz$ that satisfies $|vy| \geq 1$ and $uv^i xy^i z \in L(G)$ for every $i \geq 0$. As $vy \neq \epsilon$, $L(G)$ is infinite.

(b) Let ℓ be a pumping constant, which we can easily compute (how?). First we check if $L(G)$ is infinite by the algorithm shown in class. If it is, we return *false*. Otherwise, we enumerate over all words of length at most ℓ (why is it sufficient?) and count how many of them are in $L(G)$ using the algorithm we saw in class. We return *true* iff the count is 2017.