CS138: Automata and Formal Languages
Prof. Stefano Tessaro

UC Santa Barbara
Spring 2018

Homework 7

Posted: Monday, May 21, 2018 – 11:59pm
Due: Wednesday, May 30, 2018 – 2pm (Gradescope)

Task 1 – Pushdown Automata (8 points)

For each of the following languages with alphabet $\Sigma = \{a, b, c\}$, give the transition graph of a non-deterministic pushdown accepter (NPDA) accepting it:

a) $L_1 = \left\{a^i b^j c^k \,:\, i, j \geq 0,\ k = i + j\right\}$

b) $L_2 = \left\{w\ \mathsf{double}(w)^R \,:\, w \in \{a, b\}^*\right\}$, where $\mathsf{double}(a_1 a_2 \ldots a_n) = a_1 a_1 a_2 a_2 \ldots a_n a_n$, and $\mathsf{double}(\lambda) = \lambda$.

c) $L_3 = \{w \in \{a, b\}^* \,:\, n_a(w) = n_b(w) + 2\}$, where recall that $n_a(w)$ and $n_b(w)$ denote the numbers of a's and b's in w, respectively.

d) $L_4 = \{w \in \{a, b\}^* \,:\, n_a(w) < n_b(w)\}$.

Task 2 – Pumping Lemma for Context-Free Languages (4 points)

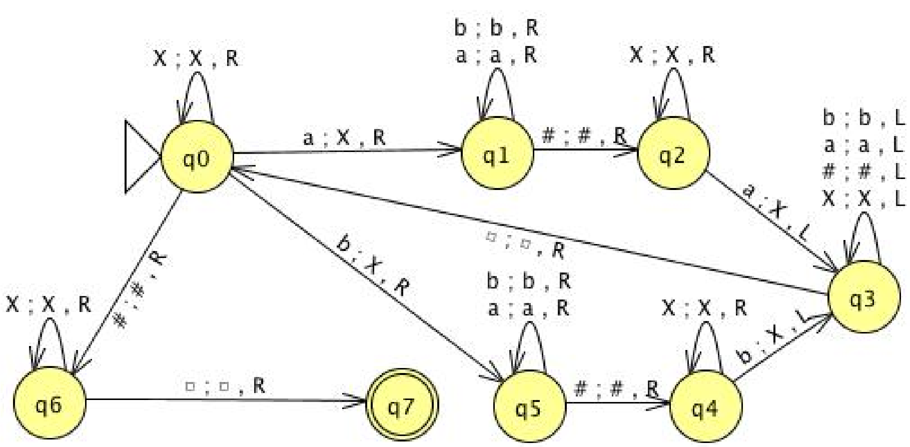
Show that the following languages with alphabet $\Sigma = \{a, b\}$ are not context free using the Pumping Lemma for CFLs.

a) $L_1 = \{a^i b a^i b a^i \,:\, i \geq 0\}$

b) $L_2 = \{w\ \mathsf{double}(w) \,:\, w \in \{a, b\}^*\}$

Task 3 – Turing Machines: Analysis (4 points)

Consider the Turing Machine M defined by the following transition graph, with input alphabet $\Sigma = \{a, b, \#\}$ as well as tape alphabet $\Gamma = \{a, b, \#, X, \Box\}$.



Describe the language $L(M)$ accepted by M.

Hint. "Test" the machine on a few inputs to gain some more intuition.

Task 4 – Turing Machines: Design (8 points)

For each one of the following languages with alphabet $\Sigma = \{0, 1\}$, give the transition graph of a Turing Machine accepting the language:

a) $L_1 = \{1^n 0^n 1^n 0^n \,:\, n \geq 0\}$

b) $L_2 = \{w \in \{0,1\}^* \,:\, n_0(w) = n_1(w)\}$, where $n_0(w), n_1(w)$ are the numbers of 0's and 1's in w.

c) $L_3 = \left\{0^{2^n} \,:\, n \geq 0\right\}$

Task 5 – Turing Computability (6 points)

Turing Machines can also operate as *transducers*, i.e., rather than recognizing a language, they compute a function. In particular, we say that a function $f : \Sigma^* \to \Gamma^*$ is **Turing computable** if there exists a TM $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ such that for all $w \in \Sigma^*$, we have

$$q_q w \overset{*}{\vdash}_M q_f f(w) ,$$

for some final state $q_f \in F$, i.e., when M is started with $w \in \Sigma^*$ on the tape, it terminates in some final state $q_f \in F$ with (1) $f(w)$ on the tape and (2) The head points to the first symbol of $f(w)$.

a) Show that f such that $f(w) = ww$ for all $w \in \{a, b\}^*$ is Turing computable.

b) Show that f' such that $f'(w) = w^R$ for all $w \in \{a, b\}^*$ is Turing computable.

For both **a)** and **b)**, give a TM computing the function.

A general hint: For every NPDA or Turing Machine, explain *why* your machine accepts the strings in the language, and why it does not accept strings outside the language. Also, use the same conventions as the textbook or class for NPDAs and Turing Machines. If you have not yet, this is a good point in time to download JFLAP at *http://www.jflap.org/*. The tool lets you define any Turing Machine and run it on specific inputs, allowing you to validate your intuitions. JFLAP will help you to reduce the amount of errors in your solution, and is therefore heavily recommended!