Automata Theory and Computability

Assignment 3 (Ultimate Periodicity, Parikh's Theorem, Pushdown Systems)

(Due on Mon 5 Nov 2018)

1. Consider the two definitions of ultimate periodicity below.

   (a) A subset X of $\mathbb{N}$ is *ultimately periodic* if there exist $n_0 \geq 0$, $p \geq 1$ in $\mathbb{N}$, such that for all $m \geq n_0$,

   $$m \in X \text{ iff } m + p \in X.$$

   (b) A subset X of $\mathbb{N}$ is *ultimately periodic* if there exist $n_0 \geq 0$, $p \geq 1$ in $\mathbb{N}$, such that for all $m \geq n_0$,

   $$m \in X \implies m + p \in X.$$

   Show that the two definitions are equivalent.

2. Using results about regular languages and ultimate periodicity, or otherwise, show that if $L \subseteq \{a\}^*$, then L^* must be regular.

3. Consider the language $L = \{a^n b a^{n^2} \mid n \geq 1\}$.

   (a) Use the Pumping Lemma for context-free languages to show that L is not context-free.

   (b) Your friend wants to prove that L is not context-free using the following idea:

   > Suppose L is context-free. Then by Parikh's theorem, there is a regular language L' that is letter-equivalent to L. But no such L' can be regular.

   Give as clear a reason (ideally a proof) why the last statement is true.

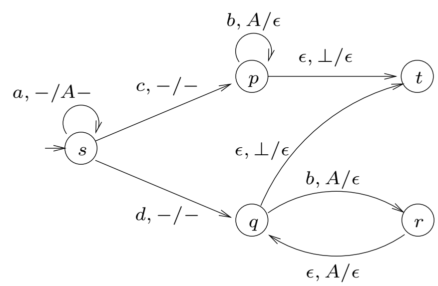
4. Consider two PDAs P_1 and P_2 with n_1 and n_2 states respectively that accept by empty stack. Construct a PDA with $n_1 + n_2 + O(1)$ states that accepts the CFL $L(P_1) \cdot L(P_2) \cup L(P_2) \cdot L(P_1)$. Briefly explain why your construction is correct. (A formal proof is not needed.)

5. Consider the CFG G with the productions

   $$\begin{array}{lcl} S & \to & Aa \\ A & \to & AA \mid aAb \mid \epsilon, \end{array}$$

   where S is the start-symbol. You can easily construct a recursive automaton for $L(G)$ with 8 states using the procedure explained in class. Show two different recursive automata with 7 states that accept $L(G)$.

6. Consider the PDA M below which runs on the input alphabet $A = \{a, b, c, d\}$ and stack alphabet $\Gamma = \{\perp, A\}$, and accepts by empty stack. The set of control states is $P = \{s, p, q, r, t\}$.



(a) Describe the language accepted by the PDA.

(b) Describe the set of configurations (control state, stack contents) which are reachable from the initial configuration $s\perp$, *and* from which the PDA can go on to empty its stack. Describe this as a regular expression describing strings in $P \cdot \Gamma^*$.

(c) Use the algorithm for *post** given in Fig. 4.2 on page 134 of Wolfgang Thomas's notes on Applied Automata (available on the course page), to compute a P-automaton accepting *post*$^*(\{s\perp\})$ for the pushdown system $\mathcal{P}$ corresponding to M.

(d) Similarly, use the algorithm done in class to compute a P-automaton for *pre*$^*(\{t\epsilon\})$.

(e) Compute a P-automaton for part (6b) above.