Theory of Computation

Fall 2014, Midterm Exam Solutions

Date: November 17, 2014

---

(1) (10 pts) If X and Y are any two languages over the same alphabet, the *symmetric difference* $X \triangle Y$ is defined to be the set of strings that are in either X or Y, but not in both. Prove that if X and Y are regular, so is $X \triangle Y$.
Solution: $X \triangle Y = (X \cap \overline{Y}) \cup (Y \cap \overline{X})$. Since regular languages are closed under $\cap, \cup$ and complementation, the result follows.

(2) (10 pts) Given a string $x \in \{0,1\}^*$, let $\overline{x}$ be the bit-wise complement of x (e.g., $\overline{01001} = 10110$). We write x^R to denote the reversal of x (e.g., $01001^R = 10010$). Is the language $L = \{w \mid w = \overline{x} \cdot x^R,\ x \in \{0,1\}^*\}$ context-free? Justify your answer.
Solution: The language can be generated by the following CFG
$G = (V, T, P, S)$, where $P : S \to 0S1 \ \mid 1S0 \ \mid \epsilon$

(3) (10 pts) Let h be the homomorphism $h(a) = 01, h(b) = 0$. Find $h^{-1}(L)$, where $L = (10 + 1)^*$. Explain why.
Solution: First, observe that $\epsilon \in L$ and thus $\epsilon \in h^{-1}(L)$ because by definition of a homomorphism, $h(\epsilon) = \epsilon$. For non-empty $w \in L$, observe that w must begin with a 1. Assume $w = h(v)$. If the first symbol of v is an a, then w would begin with 01; if the first symbol of v is a b, then w would begin with 0. In neither case does w begin with 1, so the assumption $w = h(v)$ cannot hold, so $h^{-1}(w) = \emptyset$. Therefore, $h^{-1}(L) = \{\epsilon\}$.

(4) (10 pts) Prove that the language $L = \{0^n | n = pq \text{ for two primes } p, q\}$ is not regular.
Solution: Assume L is regular. Then by the pumping lemma, there exists a constant c such that, if $w \in L$, $|w| > c$, w can be written as $uvx, r = |v| > 0$, such that $uv^ix \in L$ for all $i \geq 0$. Let p, q such that $n = pq > c$. Then $0^{pq} \in L$ and by the pumping lemma, $0^{pq+ir} \in L$ for all $i \geq 0$. Specifically, for $i = pq$ we obtain $0^{pq(1+r)} \in L$. But $pq(1 + r)$ is not the product of two primes, a contradiction.

(5) (15 pts) Let $L \subseteq \Sigma^*$, and a string $x \in \Sigma^*$.

- The *suffix* language of L with respect to x is defined as $\textit{suffix}(L, x) = \{y \in \Sigma^* \mid xy \in L\}$.
- The *class of suffix languages* of L is $C_{suf}(L) = \{\textit{suffix}(L, x) \mid x \in \Sigma^*\}$.

For example, if $L = \{0^n1^n \mid n \geq 0\}$, then $\textit{suffix}(L, 0)$=$\{0^{n-1}1^n \mid n \geq 1\}$.
Prove that for a language $L \subseteq \Sigma^*$, if $C_{suf}(L)$ is finite, then there exists a DFA $M = (Q, \Sigma, \delta, q_0, F)$ accepting L. To do so, construct M in sufficient detail, and argue that your construction is correct.
Solution: Define $M^L = (Q^L, \Sigma, \delta^L, q_0^L, F^L)$ as follows.

- $Q^L = C_{suf}(L)$
- $q_0^L = \textit{suffix}(L, \epsilon)$
- $F^L = \{\ \textit{suffix}(L, x) \mid \epsilon \in \textit{suffix}(L, x)\}$
- $\delta^L(\textit{suffix}(L, x), a) = \textit{suffix}(L, xa)$

It is not hard to see that δ^L is well defined. It can also be shown that

- For any string $x \in \Sigma^*$, $\hat{\delta}_{M^L}(q_0^L, x) = \textit{suffix}(L, x)$. (By induction)
- $x \in L$ iff $\epsilon \in \textit{suffix}(L, x)$ iff $\textit{suffix}(L, x) \in F^L$
- Hence, $x \in L$ iff M^L accepts x.

(6) (10 pts) Let $L = a^*b^*$. Prove formally that any DFA accepting L must have at least two final states.
Solution: Let $M = (Q, \{a, b\}, \delta, q_0, F)$ be a DFA accepting L. Since $\epsilon \in L$, $q_0 \in F$. Also since $ab \in L$, $\hat{\delta}(q_0, ab) = p_1 \in F$. We claim that $p_1 \neq q_0$. Assume, otherwise, that $p_1 = q_0$. Consider the word $abab$. $\hat{\delta}(q_0, abab) = \hat{\delta}(\hat{\delta}(q_0, ab), ab) = \hat{\delta}(p_1, ab) = \hat{\delta}(q_0, ab) = p_1 \in F$ – a contradiction since $abab \notin L$.

(7) (15 pts) A context-free grammar $G = (V, \Sigma, P, S)$ is called a linear grammar, if each production rule is of the following form: $A \to a \ \mid \ aB \mid \ Bb \ \mid \ aBb \ \mid \ \epsilon$, where $a, b \in \Sigma$ and $A, B \in V$. A PDA M is one-turn if its stack can only change its direction (i.e., from increasing to decreasing) once. That is, once a one-turn PDA pops a symbol from its stack, it will never push a symbol in the remainder of the computation. You may assume that at each step, a PDA M always pushes or pops a symbol, and accepts a string upon reaching a

final state with an empty stack.
Prove that the language accepted by a one-turn PDA can be generated by a linear grammar. (Hint: Given a one-turn PDA M, construct an equivalent linear grammar G.)
Solution: Without loss of generality, we consider a pushdown automaton that has a single accept state q_{accept} and empties the stack before accepting. Moreover, its transition either pushes or pops a stack symbol at any time. Let $P = (Q, \Sigma, \Gamma, \delta, q_0, \{q_{\text{accept}}\})$. Define a linear grammar $G = (V, \Sigma, R, S)$ where

- $V = \{A_{pq} : p, q \in Q\}$, $S = A_{q_0, q_{accept}}$; and
- R has the following rules:
  - For each $p, q, r, s \in Q$, $t \in \Gamma$, and $a, b \in \Sigma_\epsilon$, if $(r, t) \in \delta(p, a, \epsilon)$ and $(q, \epsilon) \in \delta(s, b, t)$, then $A_{pq} \longrightarrow aA_{rs}b \in R$.
  - For each $p \in Q$, $A_{pp} \longrightarrow \epsilon \in R$.

(8) (10 pts) Is the following language context-free? Justify your answer.

$$\{xy \mid x, y \in \{0,1\}^*, \text{ and } |x| = |y| \text{ but } x \neq y\}$$

Solution: It is not hard to see that the langauge can be expressed as $\{\underbrace{w_1}_{k}\, x\, \underbrace{v_1 w_2}_{l+k}\, y\, \underbrace{v_2}_{l} \mid |w_1| = |w_2| = k, |v_1| = |v_2| = l, x \neq y, w_1, w_2, v_1, v_2 \in \{0,1\}^*, x, y \in \{0,1\}\}$. We can design a PDA to do the following:

(a) read w_1 and push the length (i.e., the number k) into the stack
(b) (nondeterministically) read x and keep the value in the finite state control
(c) read $v_1 w_2$ while popping the stack; once the stack becomes empty, push the length of the remaining string into the stack (the stack now contains l symbols)
(d) (nondeterministically) read y and and compare it against x; if $x = y$, then reject,
(e) read v_2 and pop the stack
(f) after reading all the input symbols, **ACCEPT** if the stack is empty.

(9) (10 pts) Consider the operation X/Y=$\{w \mid \exists y \in Y, wy \in X\}$. Prove that if X is regular, Y is an arbitrary language, then X/Y is always regular. (Hint: Suppose FA M accepts X, construct an FA M' to accept X/Y.)
Proof:

Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA accepting X. We define $F' = \{q \in Q \mid \exists y \in Y, \hat{\delta}(q, y) \in F\}$. Then $M' = (Q, \Sigma, \delta, q_0, F')$ accepts exactly X/Y. Hence, X/Y is regular.