HW 4: Extra problems

Instructor: Timothy M. Chan, Haitham Hassanieh, and Sariel Har-Peled.

CS/ECE 374: Algorithms & Models of Computation, Spring 2019 Version: **1.0**

1 Let L be an arbitrary regular language.

1.A. Prove that the language $palin(L)\{w \mid ww^R \in L\}$ is also regular.

1.B. Prove that the language $drome(L)\{w \mid w^R w \in L\}$ is also regular.

2 Suppose F is a fooling set for a language L. Argue that F cannot contain two distinct string x, y where both are not prefixes of strings in L.

3 Prove that the language $\{0^i 1^j \mid \gcd(i, j) = 1\}$ is not regular.

4 Consider the language $L = \{w : |w| = 1 \mod 5\}$. We have already seen that this language is regular. Prove that any DFA that accepts this language needs at least 5 states.

5 Consider all regular expressions over an alphabet Σ. Each regular expression is a string over a larger alphabet $\Sigma' = \Sigma \cup \{\emptyset\text{-Symbol}, \epsilon\text{-Symbol}, +, (,)\}$. We use $\emptyset$-Symbol and ϵ-Symbol in place of $\emptyset$ and ϵ to avoid confusion with overloading; technically one should do it with $+, (,)$ as well. Let R_Σ be the language of regular expressions over Σ.

5.A. Prove that R_Σ is not regular.

5.B. Prove that R_Σ is a CFL by giving a CFG for it.

6 Regular languages?

6.A. Prove that the following languages are not regular by providing a fooling set. You need to prove an infinite fooling set and also prove that it is a valid fooling set.

6.A.i. $L = \{0^k 1^k ww \mid 0 \le k \le 3, w \in \{0, 1\}^+\}$.

6.A.ii. Recall that a block in a string is a maximal non-empty substring of identical symbols. Let L be the set of all strings in $\{0, 1\}^*$ that contain two blocks of 0s of equal length. For example, L contains the strings 01101111 and 01001011100010 but does not contain the strings 000110011011 and 00000000111.

6.A.iii. $L = \{0^{n^3} \mid n \ge 0\}$.

6.B. Suppose L is not regular. Show that $L \cup L'$ is not regular for any finite language L'. Give a simple example to show that $L \cup L'$ is regular when L' is infinite.

7 Describe a context free grammar for the following languages. Clearly explain how they work and the role of each non-terminal. Unclear grammars will receive little to no credit.

7.A. $\{a^i b^j c^k d^\ell \mid i, j, k, \ell \ge 0 \text{ and } i + \ell = j + k\}$.

7.B. $L = \{0, 1\}^* \setminus \{0^n 1^n \mid n \ge 0\}$. In other words the complement of the language $\{0^n 1^n \mid n \ge 0\}$.

8 Let $L = \{0^i 1^j 2^k \mid k = 2(i + j)\}$.

8.A. Prove that L is context free by describing a grammar for L.

8.B. Prove that your grammar is correct. You need to prove that if $L \subseteq L(G)$ and $L(G) \subseteq L$ where G is your grammar from the previous part.

Solved problem

9 Let L be the set of all strings over $\{0, 1\}^*$ with exactly twice as many 0s as 1s.

9.A. Describe a CFG for the language L.

(**Hint:** For any string u define $\Delta(u) = \#(0, u) - 2\#(1, u)$. Introduce intermediate variables that derive strings with $\Delta(u) = 1$ and $\Delta(u) = -1$ and use them to define a non-terminal that generates L.)

Solution:

$S \to \varepsilon \mid SS \mid 00S1 \mid 0S1S0 \mid 1S00$

9.B. Prove that your grammar G is correct. As usual, you need to prove both $L \subseteq L(G)$ and $L(G) \subseteq L$.

(**Hint:** Let $u_{\leq i}$ denote the prefix of u of length i. If $\Delta(u) = 1$, what can you say about the smallest i for which $\Delta(u_{\leq i}) = 1$? How does u split up at that position? If $\Delta(u) = -1$, what can you say about the smallest i such that $\Delta(u_{\leq i}) = -1$?)

Solution:

We separately prove $L \subseteq L(G)$ and $L(G) \subseteq L$ as follows:

Claim 4.1. *$L(G) \subseteq L$, that is, every string in $L(G)$ has exactly twice as many 0s as 1s.*

Proof: As suggested by the hint, for any string u, let $\Delta(u) = \#(0, u) - 2\#(1, u)$. We need to prove that $\Delta(w) = 0$ for every string $w \in L(G)$.

Let w be an arbitrary string in $L(G)$, and consider an arbitrary derivation of w of length k. Assume that $\Delta(x) = 0$ for every string $x \in L(G)$ that can be derived with fewer than k productions.[1] There are five cases to consider, depending on the first production in the derivation of w.

- If $w = \varepsilon$, then $\#(0, w) = \#(1, w) = 0$ by definition, so $\Delta(w) = 0$.
- Suppose the derivation begins $S \to SS \to^* w$. Then $w = xy$ for some strings $x, y \in L(G)$, each of which can be derived with fewer than k productions. The inductive hypothesis implies $\Delta(x) = \Delta(y) = 0$. It immediately follows that $\Delta(w) = 0$.[2]
- Suppose the derivation begins $S \to 00S1 \to^* w$. Then $w = 00x1$ for some string $x \in L(G)$. The inductive hypothesis implies $\Delta(x) = 0$. It immediately follows that $\Delta(w) = 0$.
- Suppose the derivation begins $S \to 1S00 \to^* w$. Then $w = 1x00$ for some string $x \in L(G)$. The inductive hypothesis implies $\Delta(x) = 0$. It immediately follows that $\Delta(w) = 0$.

- Suppose the derivation begins $S \to 0S1S1 \to^* w$. Then $w = 0x1y0$ for some strings $x, y \in L(G)$. The inductive hypothesis implies $\Delta(x) = \Delta(y) = 0$. It immediately follows that $\Delta(w) = 0$.

In all cases, we conclude that $\Delta(w) = 0$, as required. ■

Claim 4.2. *$L \subseteq L(G)$; that is, G generates every binary string with exactly twice as many 0s as 1s.*

Proof: As suggested by the hint, for any string u, let $\Delta(u) = \#(0, u) - 2\#(1, u)$. For any string u and any integer $0 \le i \le |u|$, let $\boldsymbol{u_i}$ denote the ith symbol in u, and let $\boldsymbol{u_{\le i}}$ denote the prefix of u of length i.

Let w be an arbitrary binary string with twice as many 0s as 1s. Assume that G generates every binary string x that is shorter than w and has twice as many 0s as 1s. There are two cases to consider:

- If $w = \varepsilon$, then $\varepsilon \in L(G)$ because of the production $S \to \varepsilon$.
- Suppose w is non-empty. To simplify notation, let $\Delta_i = \Delta(w_{\le i})$ for every index i, and observe that $\Delta_0 = \Delta_{|w|} = 0$. There are several subcases to consider:
  - Suppose $\Delta_i = 0$ for some index $0 < i < |w|$. Then we can write $w = xy$, where x and y are non-empty strings with $\Delta(x) = \Delta(y) = 0$. The induction hypothesis implies that $x, y \in L(G)$, and thus the production rule $S \to SS$ implies that $w \in L(G)$.
  - Suppose $\Delta_i > 0$ for all $0 < i < |w|$. Then w must begin with 00, since otherwise $\Delta_1 = -2$ or $\Delta_2 = -1$, and the last symbol in w must be 1, since otherwise $\Delta_{|w|-1} = -1$. Thus, we can write $w = 00x1$ for some binary string x. We easily observe that $\Delta(x) = 0$, so the induction hypothesis implies $x \in L(G)$, and thus the production rule $S \to 00S1$ implies $w \in L(G)$.
  - Suppose $\Delta_i < 0$ for all $0 < i < |w|$. A symmetric argument to the previous case implies $w = 1x00$ for some binary string x with $\Delta(x) = 0$. The induction hypothesis implies $x \in L(G)$, and thus the production rule $S \to 1S00$ implies $w \in L(G)$.
  - Finally, suppose none of the previous cases applies: $\Delta_i < 0$ and $\Delta_j > 0$ for some indices i and j, but $\Delta_i \ne 0$ for all $0 < i < |w|$.

    Let i be the smallest index such that $\Delta_i < 0$. Because Δ_j either increases by 1 or decreases by 2 when we increment j, for all indices $0 < j < |w|$, we must have $\Delta_j > 0$ if $j < i$ and $\Delta_j < 0$ if $j \ge i$.

    In other words, there is a *unique* index i such that $\Delta_{i-1} > 0$ and $\Delta_i < 0$. In particular, we have $\Delta_1 > 0$ and $\Delta_{|w|-1} < 0$. Thus, we can write $w = 0x1y0$ for some binary strings x and y, where $|0x1| = i$.

    We easily observe that $\Delta(x) = \Delta(y) = 0$, so the inductive hypothesis implies $x, y \in L(G)$, and thus the production rule $S \to 0S1S0$ implies $w \in L(G)$.

In all cases, we conclude that G generates w. ■

Together, Claim 1 and Claim 2 imply $L = L(G)$.

Rubric: 10 points:

- part (a) = 4 points. As usual, this is not the only correct grammar.
- part (b) = 6 points = 3 points for $\subseteq$ + 3 points for $\supseteq$, each using the standard induction template (scaled).