CS/ECE 374 ✧ Fall 2016

Homework 3

Due Tuesday, September 20, 2016 at 8pm

Groups of up to three people can submit joint solutions. Each problem should be submitted by exactly one person, and the beginning of the homework should clearly state the Gradescope names and email addresses of each group member. In addition, whoever submits the homework must tell Gradescope who their other group members are.

---

1. For each of the following regular expressions, describe or draw two finite-state machines:

   - An NFA that accepts the same language, obtained using Thompson's recursive algorithm
   - An equivalent DFA, obtained using the incremental subset construction. For each state in your DFA, identify the corresponding subset of states in your NFA. Your DFA should have no unreachable states.

   (a) $(\mathtt{00} + \mathtt{11})^*(\mathtt{0} + \mathtt{1} + \varepsilon)$

   (b) $\mathtt{1}^* + (\mathtt{01})^* + (\mathtt{001})^*$

2. Give context-free grammars for the following languages, and clearly explain how they work and the role of each nonterminal. Grammars can be very difficult to understand; if the grader does not understand how your construction is intended to generate the language, then you will receive no credit.

   (a) In any string, a ***block*** (also called a ***run***) is a maximal non-empty substring of identical symbols. For example, the string `0111000011001` has six blocks: three blocks of `0`s of lengths 1, 4, and 2, and three blocks of `1`s of lengths 3, 2, and 1.

   Let L be the set of all strings in $\{\mathtt{0},\mathtt{1}\}^*$ that contain two blocks of `0`s of equal length. For example, L contains the strings `01101111` and `01001011100010` but does not contain the strings `000110011011` and `00000000111`.

   (b) $L = \{w \in \{\mathtt{0},\mathtt{1}\}^* \mid w \text{ is } \textit{not} \text{ a palindrome}\}$.

3. Let $L = \{\mathtt{0}^i\mathtt{1}^j\mathtt{2}^k \mid k = i + j\}$.

   (a) Show that L is context-free by describing a grammar for L.

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

Solved problem

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

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

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

Solution: (Hopefully you recognized this as a more advanced version of HW0 problem 3.) We separately prove $L \subseteq L(G)$ and $L(G) \subseteq L$ as follows:

Claim 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 $S \rightsquigarrow SS \rightsquigarrow^* w$. 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 $S \rightsquigarrow 00S1 \rightsquigarrow^* w$. 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 $S \rightsquigarrow 1S00 \rightsquigarrow^* w$. 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 $S \rightsquigarrow 0S1S1 \rightsquigarrow^* w$. 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. □

[1]Alternatively: Consider the *shortest* derivation of w, and assume $\Delta(x) = 0$ for every string $x \in L(G)$ such that $|x| < |w|$.

[2]Alternatively: Suppose the *shortest* derivation of w begins $S \rightsquigarrow SS \rightsquigarrow^* w$. Then $w = xy$ for some strings $x, y \in L(G)$. Neither x or y can be empty, because otherwise we could shorten the derivation of w. Thus, x and y are both shorter than w, so the induction hypothesis implies. . . . We need some way to deal with the decompositions $w = \varepsilon \bullet w$ and $w = w \bullet \varepsilon$, which are both consistent with the production $S \to SS$, without falling into an infinite loop.

Claim 2. *$L \subseteq L(G)$; that is, G generates every binary string with exactly twice as many* `0`*s as* `1`*s.*

Proof: As suggested by the hint, for any string u, let $\Delta(u) = \#(\mathtt{0}, u) - 2\#(\mathtt{1}, u)$. 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 `0`s as `1`s. Assume that G generates every binary string x that is shorter than w and has twice as many `0`s as `1`s. 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 $w = \mathtt{00}x\mathtt{1}$ 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 $S \to \mathtt{00}S\mathtt{1}$ implies $w \in L(G)$.
  - Suppose $\Delta_i < 0$ for all $0 < i < |w|$. A symmetric argument to the previous case implies $w = \mathtt{1}x\mathtt{00}$ for some binary string x with $\Delta(x) = 0$. The induction hypothesis implies $x \in L(G)$, and thus the production rule $S \to \mathtt{1}S\mathtt{00}$ 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 $w = \mathtt{0}x\mathtt{1}y\mathtt{0}$ for some binary strings x and y, where $|\mathtt{0}x\mathtt{1}| = i$.

    We easily observe that $\Delta(x) = \Delta(y) = 0$, so the inductive hypothesis implies $x, y \in L(G)$, and thus the production rule $S \to \mathtt{0}S\mathtt{1}S\mathtt{0}$ 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).