CS/ECE 374 Sec A ✧ Spring 2025
Homework 4
Due Wednesday, Feb 19th, 2025 at 9am

- You can work in a group of up to **three** students. Read the instructions on the course website for additional details.
- **Submit your solutions electronically on the course Gradescope site as PDF files.** Submit a separate PDF file for each numbered problem. If you plan to typeset your solutions, please use the LaTeX solution template on the course web site. If you must submit scanned handwritten solutions, please use a black pen on blank white paper and a high-quality scanner app (or an actual scanner, not just a phone camera).
- Late submissions will be accepted (for 75% credit) until midnight the day of the deadline.

☞ Some important course policies ☜

- **You may use any source at your disposal**—paper, electronic, or human—but you ***must*** cite ***every*** source that you use, and you ***must*** write everything yourself in your own words. See the academic integrity policies on the course web site for more details including the policy on using AI tools.
- **Avoid the Two Deadly Sins!** Any homework or exam solution that breaks any of the following rules will be given an ***automatic zero***, unless the solution is otherwise perfect. Yes, we really mean it. We're not trying to be scary or petty (Honest!), but we do want to break a few common bad habits that seriously impede mastery of the course material.
  - Always give complete solutions, not just examples.
  - Always declare all your variables, in English. In particular, always describe the specific problem your algorithm is supposed to solve.

See the course web site for more information.

If you have any questions about these policies,
please don't hesitate to ask in class, in office hours, or on Ed.

---

1. (a) (6 pts) Prove that the following languages are not regular by providing a fooling set. You need to provide an infinite set and also prove that it is a valid fooling set for the given language. Alternatively, you can describe a fooling set F_n of size n for every $n > 0$ and prove its validity.

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

   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 $\{\mathtt{0}, \mathtt{1}, \mathtt{2}^*\}^*$ that have unequal number of **0** blocks and **1** blocks. For example, L contains the strings **00**, **00121** but does not contain the strings **000110001100011** and **00000000111**.

   iii. $L = \{\mathtt{0}^{\lceil n\sqrt{n}\rceil} \mid n \geq 1\}$.

   (b) (2.5 pts) Let $L_k = \{w \in \{\mathtt{0}, \mathtt{1}\}^* : |w| \geq 2k$ and last $2k$ characters of w have equal number of **0**s and **1**s$\}$ If $k = 3$ then **0001011** and **01000111000** are in L_3 while **000110** and **000111100** are not. Describe a fooling set for L_k of size at least 2^k and prove that it is valid. In fact 2^k is not the best one can do.
   Not to submit for grading: Design an NFA for L_k with $O(k^2)$ states.

   (c) (1.5 pts) Suppose L is not regular and L' is a finite language. Prove that $L \cup L'$ is not regular. Give a simple example of a non-regular language L and a regular language L' such that $L \cup L'$ is regular.

2. 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.

   (a) (3 pts) Let CFG G be the following grammar with $T = \{a, b\}$.

$$S \to aSb \mid bY \mid Ya$$
$$Y \to bY \mid aY \mid \varepsilon$$

   Explain in English what $L(G)$ is and give a CFG for $\overline{L(G)}$, the complement of the language generated by G.

   (b) (3.5 pts) $L = \{a^i b^j c^k d^\ell \mid i + j = k + \ell\}$

   (c) (3.5 pts) $L = \{\mathtt{0}^i \mathtt{1} \mathtt{0}^j \mathtt{1} \mathtt{0}^k \mid i + k = 2j\}$

   (d) **Not to submit:** $L = \{w \in \{0, 1, 2\}^* \mid \#(0, w) = \#(1, w) + \#(2, w)\}$

   (e) **Not to submit:** $L = \{x_1 \# x_2 \# \ldots \# x_k \mid k \geq 1$, each $x_i \in \{0, 1\}^*$, and for some i and j, $x_i = x_j^R\}$. Note that i can be equal to j in the definition and there can be multiple pairs that satisfy the condition. Here the terminal set T is $\{0, 1, \#\}$.

3. **Not to submit:** 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.

   (a) Prove that R_Σ is not regular.

   (b) Describe a context free grammar (CFG) for R_Σ which will prove that it is a CFL.

This shows that we need more expressive languages than regular languages to describe regular expressions.

Solved problem

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

   (a) Describe a CFG for the language L.
   [Hint: For any string u define $\Delta(u) = \#(\texttt{0},u) - 2\#(\texttt{1},u)$. 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: $S \to \varepsilon \mid SS \mid \texttt{00}S\texttt{1} \mid \texttt{0}S\texttt{1}S\texttt{0} \mid \texttt{1}S\texttt{00}$ ■

   (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: 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 `0`s as `1`s.*

   Proof: As suggested by the hint, for any string u, let $\Delta(u) = \#(\texttt{0},u) - 2\#(\texttt{1},u)$. 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 $\#(\texttt{0},w) = \#(\texttt{1},w) = 0$ 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]

[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.

- Suppose the derivation begins $S \rightsquigarrow \texttt{00}S\texttt{1} \rightsquigarrow^* w$. Then $w = \texttt{00}x\texttt{1}$ 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 $S \rightsquigarrow \texttt{1}S\texttt{00} \rightsquigarrow^* w$. Then $w = \texttt{1}x\texttt{00}$ 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 $S \rightsquigarrow \texttt{0}S\texttt{1}S\texttt{1} \rightsquigarrow^* w$. Then $w = \texttt{0}x\texttt{1}y\texttt{0}$ 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 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 $\Delta(u) = \#(\texttt{0}, u) - 2\#(\texttt{1}, u)$. 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 $w = \texttt{00}x\texttt{1}$ 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 $S \to \texttt{00}S\texttt{1}$ implies $w \in L(G)$.
  - Suppose $\Delta_i < 0$ for all $0 < i < |w|$. A symmetric argument to the previous case implies $w = \texttt{1}x\texttt{00}$ for some binary string x with $\Delta(x) = 0$. The induction hypothesis implies $x \in L(G)$, and thus the production rule $S \to \texttt{1}S\texttt{00}$ 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 $w = \texttt{0}x\texttt{1}y\texttt{0}$ for some binary strings x and y, where $|\texttt{0}x\texttt{1}| = i$.

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