Homework 2

CS304: Automata and Formal Languages

Due: Mon Sep 8, 2025

Submission Guidelines: Your submission must be a **single PDF file**. You are strongly encouraged to typeset your solutions using LaTeX, as it is the standard in computer science. (Overleaf is a great starting point for beginners.) However, solutions written in any other word processor (like MS Word or Google Docs) are acceptable, as long as you export the final document to PDF. **Note that LaTeX would be made compulsory for all submissions from Homework 3 onwards;** and you are encouraged to start using it from this homework itself. For NFA/DFA/Parse Tree diagrams you have two options:

- (**preferred**) Use a drawing tool (e.g., draw.io, TikZ, or any other vector-graphics editor) and embed the vector-image directly into your document.
- Draw the diagram neatly on paper and take a clear, well-lit photograph. Crop the resulting image and embed it legibly within your document.

Question 1 (20 points)**.** *For each of the following languages over the alphabet* $\Sigma = \{0, 1\}$*, provide the corresponding REs and diagram of NFAs (following the conventions used in class).*

- $L_a = \{01\}$
- $L_b = \{10\}$
- $L_c = L_a \cup L_b$
- $L_d = L_c^*$
- $L_e = \{\epsilon, 1\}$
- $L_f = L_e^*$
- $L_g = L_d L_f$

Question 2 (20 points). *Consider alphabet* $\Sigma = \{0, 1\}$ *and language* $L = \{0^n 1^m \mid n \neq m\}$, *for* $n, m \geq 0$.

- *Prove that* L *is irregular* [1]. *If you are using a language that was proven irregular in the class as a part of your solution, you need to reproduce that proof.*
- *(bonus) Apply the pumping lemma directly on* L *to show it is irregular.*

Question 3 (20 points). *Does the language* L *of Question 2 have a CFG?*

- *If yes, provide a CFG for it and then show that the strings* $\{0, 0111\} \subseteq L$ *by constructing the corresponding parse trees.*
- *If not, explain why in one or two paragraphs.*

Question 4 (20 points). *Use the state-elimination algorithm*[2] *to convert the following NFA* $N = (Q = \{q_0, q_1, q_2\}, \Sigma = \{0, 1\}, \delta, q_0, \{q_0, q_2\})$ *to a regular expression. Transitions* δ *of* N *are:*

- *From* q_0 *to* q_0 *on* 0
- *From* q_0 *to* q_1 *on* 1
- *From* q_1 *to* q_1 *on* 1
- *From* q_1 *to* q_2 *on* 1
- *From* q_2 *to* q_1 *on* 0

Question 5 (20 points). *Prove that the language of all valid C programs is irregular.* [3]

Question 6 (For practice; Do not submit). *Convert the NFA of Question 4 to a DFA using the power-set algorithm. Convert your RE solution of Question 4 back to an NFA using recursive construction. Did you get the same NFA? How'd you prove that both the NFAs(the one that you got and the original one) are equivalent?*

[1] **Hint:** Use Closure Properties

[2] Write solution step by step: draw NFA, say which state you will eliminate, re-draw NFA after eliminating it, and repeat until you reach your final solution.

[3] **Hint:** Prove that the language L_{balanced} of all balanced curly parentheses is irregular. Is this sufficient?