"CS/ECE 374 A": Algorithms & Models of Computation, Spring 2025
Midterm 1 — Feb 24, 2025

Name:	Heng An Cheng
NetID:	hacheng2

- Please *clearly PRINT* your name and your NetID in the boxes above.
- This is a closed-book but you are allowed a 1 page (2 sides) hand written cheat sheet that you have to submit along with your exam. If you brought anything except your writing implements, put it away for the duration of the exam. In particular, you may not use *any* electronic devices.
- **Please read the entire exam before writing anything.** Please ask for clarification if any question is unclear. The exam has 6 problems, each worth 10 points.
- **You have 150 minutes (2.5 hours) for the exam.**
- If you run out of space for an answer, continue on the back of the page, or on the blank pages at the end of this booklet, **but please tell us where to look.**
- **Write everything inside the box around each page.** Anything written outside the box may be cut off by the scanner.
- **Proofs are required only if we specifically ask for them.** You may state and use (without proof or justification) any results proved in class or in the problem sets unless we explicitly ask you for one.
- You can do hard things!
- **Do not cheat.** You know the student code and all that jazz. Grades do matter, but not as much as you may think, and your values are more important.

---

1 Regular Expression

Assume $\Sigma = \{0,1\}$. Give regular expressions for the two languages below and **briefly explain how your expressions work.**

(a) (5 pts) All strings which contain the substring 01010 and which have an even number of blocks of 0's. Recall that a block (or a run) is a non-empty maximal substring of the same symbol.

even odd.

$$\boxed{\begin{aligned}&1^*(0^+1^+0^+1^+)^*0^*010100^*1^+0^+(1^+0^+1^+0^+)^*1^* \\ &+ 1^*(0^+1^+0^+1^+)^*0^+1^+0^*010100^*(1^+0^+1^+0^+)^*1^*\end{aligned}}$$

$1^*(0^+1^+0^+1^+)^*$ & $(1^+0^+1^+0^+)^*1^*$ will always create even number of block 0

The first expression represent there have odd number of blocks in the front of 01010 and even number after 01010 $\Rightarrow$ odd + 1 + even = even

reverse for the second expression $\Rightarrow$ even + 1 + odd = even #

(b) (5 pts) $\{1^{2n}w0^n \mid n \geq 2, w \in \{0,1\}^*\}$

$1111(0+1)^*00$

because w can be any string, so as long as the string has four 1 at the front and two 0 at the end, no matter what the strirs is I can view them as $n=2$

$\Rightarrow 1^4w0^2$, and will always in this Languages.

2 DFA Design

Give DFAs for the two languages below and **briefly describe the meaning of each state.**

(a) (5 pts) Strings in $\{0,1\}^*$ whose first character equals its second-to-last character. Examples of strings in this language include 00101, 11111, and 10. Examples of strings not in this language include ϵ, 1100, and 10101.

$M = (Q, \Sigma, \delta, s, A)$ is the DFA for this languages

use state (A,B,C) to represents the first, second-to-last, last letter

$\Rightarrow Q = \{(A,B,C) \mid A,B,C \in \Sigma^*\}$

$S = (\varepsilon, \varepsilon, \varepsilon)$

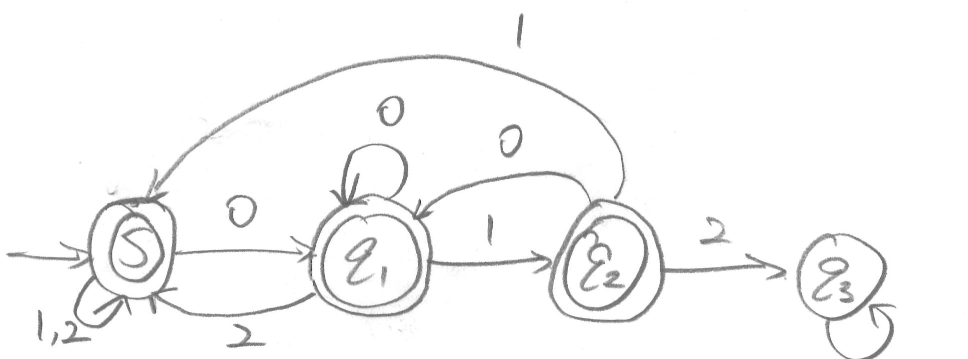
$\delta = Q \times \Sigma \to Q$, $\delta(s, a) = \begin{cases} (1, \varepsilon, 1) & \text{if } a = 1 \\ (0, \varepsilon, 0) & \text{if } a = 0 \end{cases}$

$\delta((A,B,C), a) = (A, C, a)$

$\Sigma = \{0, 1\}$

$A = \{(1,\varepsilon,1), (1,\varepsilon,0), (0,\varepsilon,0), (0,\varepsilon,1)\} \cup \{(A,B,C) \mid A=B\}$ #

(b) (5 pts) Strings in $\{0,1,2\}^*$ that do *not* contain the substring 012.



q_1 : detect substring 0

q_2 : detect substring 01

q_3 : detect substring 012

S : nothing detect

3 NFA Design

(10 pts) Let $k \geq 1$ be an integer and let

$L_k = \{w \in \{0,1\}^* : |w| \geq k$ and number of 0s and 1s differ by at most two in the last k bits of $w\}$

For example if $k = 5$ then the strings 0000011 and 11010 are in L_5 while 11110 and 100000 are not in L_5. Describe an NFA for L_k with $O(k^2)$ states and **briefly describe the meaning of each state.**

Let $N = (Q, \{0,1\}, \delta, s, A)$ be the NFA for L_k

define differ = #0's − #1's in current input

$\begin{cases} \text{len} = k, \text{ when input length} > k \\ \text{otherwise len} = (\text{input length}) \end{cases}$ $\Rightarrow q = (\text{differ}, \text{len})$

$\Rightarrow Q = \{ (\text{differ}, \text{len}) \mid |\text{differ}| \leq k,\ 0 \leq \text{len} \leq k \}$

$s = (0, 0)$

$\delta((d, l), 0) = \begin{cases} \{(d+1, l+1)\}, & l < k \\ \{(d+2, l), (d, l)\}, & l = k \end{cases}$

$\delta((d, l), 1) = \begin{cases} \{(d-1, l+1)\}, & l < k \\ \{(d-2, l), (d, l)\}, & l = k \end{cases}$

$A = \{ (d, l) \mid |d| \leq 2,\ l = k \}$

$\Rightarrow$ ∵ Q has $O(k^2)$ states, and N accept all strings in L_k #

$\Rightarrow N$ is an NFA for L_k #

4 Context-Free Grammars

Give a context-free grammar for the two languages below, where the terminal set T is $\{0,1\}$. In order to get full credit you need to **briefly explain how your grammar works, and the role of each non-terminal.**

(a) (5pts) $L = \{0^i 1 0^j 1 0^k \mid i+k \geq 3j\}$

$i \geq 2j$ or

6

$S \to N \mid 0S \mid S0 \mid AB \mid CE \mid BA \mid EC$ $\{0^i 1 0^j 1 0^k \mid i+k \geq 3j\}$

$A \to 0000A0 \mid E$ $\{0^i 1 0^j \mid i =$

$B \to 0B00 \mid E$

$C \to 000C0 \mid E$

$E \to 1$

$N \to 001010 \mid 010100$ $\{0^i 1 0^j 1 0^k \mid i+k=3\}$

(b) (5 pts) All strings that have an odd number of 0's and end in 01. $G = (\{S, A, E\}, \{0,1\}, R, S)$

R: $S \to 1S \mid 0A \mid E$ $\{(1+01^*0)^* 01\}$

$A \to 1A \mid 0S$ $\{1^*0\}$

$E \to 01$ $\{01\}$

Because there has 0 in 01 $\Rightarrow$ the prefix before 01 should have even number of 0's.

$\Rightarrow$ A can help you generate string 01^*0 to make sure the number of 0's is even

E $\to$ generate the end 01

5 Fooling Sets and Non-Regularity

(10 pts) Prove that the language

$$L = \{w \in \{0,1\}^* \mid w \text{ is a palindrome and } w \text{ has at least three 0s}\}$$

is not regular by providing a fooling set for it and showing that every pair of strings in your fooling set can be distinguished.

Let $F = \{0^n 1 \mid n \geq 3\}$

Consider $x, y \in F$, $x \neq y$

$\Rightarrow x = 0^i 1$, $y = 0^j 1$, where $i \neq j$ and $i, j \geq 3$.

we claim that $z = 0^i$ distinguishes x any y with respect to L.

Because $xz = 0^i 1 0^i$ is a palindrome and $2i > 3$

$\Rightarrow xz \in L$

But $yz = 0^j 1 0^i$ is not a palindrome, ($\because i \neq j$).

$\Rightarrow yz \notin L$

Hence F is an infinate fooling set for L,

which proves that L is not regular #

6 Language Transformation

(10 pts) Let $\Sigma = \{0, 1\}$. For a language $L \subseteq \Sigma^*$ we define an operation **deletepropermid** as follows:

$$\textbf{deletepropermid}(L) = \{uw \mid uvw \in L \text{ and } u, v, w \in \Sigma^*, |v| \geq 2\}.$$

Prove that if L is regular then **deletepropermid**(L) is also regular.

Let $M = (Q, s, A, \delta)$ be a DFA for L

we construct a NFA $N' = (Q', s', A', \delta')$ for deletepropermid(L), by nondetermisticálly adding v in the mid of u, w, then running uvw on M.

M' will guess the missing v, and make sure $|v| \geq 2$, by definirg 4 state before, add1, add2, after

before $\Rightarrow$ we don't guess anything after $\Rightarrow$ finish guessing.

add1 $\Rightarrow$ we guess 1 element add2 $\rightarrow$ we guess at least 2 element

Formally we have

$Q' = Q \times \{\text{before}, \text{add1}, \text{add2}, \text{after}\}$

$s' = (s, \text{before})$, $A' = A \times \{\text{after}\}$

δ': $\delta((q, \text{before}), a) = \{(\delta(q, a), \text{before})\}$

$\delta'((q, \text{after}), a) = \{(\delta(q, a), \text{after})\}$

$\delta((q, \text{before}), \varepsilon) = \{(\delta(q, \Sigma), \text{add1})\}$

$\delta'((q, \text{add1}), \varepsilon) = \{(\delta(q, \Sigma), \text{add2})\}$

$\delta'((q, \text{add2}), \varepsilon) = \{(q, \text{after}), (\delta(q, \Sigma), \text{add2})\}$ #

This page is for extra work.

Thus, we create a NFA for deletepropermid(L), by creating some new states and simulating with M. So, If L is regular, then deletepropermid(L) is also regular.

This page is for extra work.

This page is for extra work.