CS 39: Theory of Computation **Winter '23**

Mid-Term Exam 1 — 02/13/2023

Prof. Chakrabarti ***Student: Amittai Siavava***

Credit Statement

All work on the mid-term is my own. I referred to class notes and the following books:

(i) **Introduction to the Theory of Computation** by **Michael Sipser**.

Problem 1.

Let $D = \{0,1,2,3,4,5,6,7,8,9\}$ be the alphabet of all decimal digits. A string $x \in D^*$ is said to be stable if, for each pair of adjacent digits in x, those two digits have a numerical difference of at most 1. For example:

$$\text{Stable:} \quad 433321001012,\ 556677654,\ 7,\ 0000,\ \varepsilon.$$

$$\text{Unstable:} \quad 6554667,\ 1213141516,\ 7890123.$$

Give a formal description of a DFA that recognizes $L_1 = \{x \in D^* : x \text{ is stable}\}$. Provide a high-level explanation of your design idea (no formal completeness and soundness proofs are required).

$$M = (Q, D, \delta, s, F), \text{ where}$$

$$Q = \{s, r\} \cup \{q_i : i \in D\}$$

$$\delta(q, x) = \begin{cases} q_x & \text{if } x \in D \text{ and } q \in \{s, q_{x-1}, q_x, q_{x+1}\}. \\ r & \text{otherwise.} \end{cases}$$

$$F = \{s\} \cup \{q_i : i \in D\} = Q \setminus \{r\}.$$

Design Idea:

We maintain a start state, a trapping reject state, and one state for each digit.

Here is how we handle transitions:

- If we are in the start state and read a digit, we transition to the corresponding digit state.
- If we are in a digit state q_x (for the digit x) and read a digit y, then:
  - if $y = x$, we loop to the same state q_x.
  - if $y = x \pm 1$, we transition to the corresponding digit state q_y.
  - if $y \neq x$ and $y \neq x \pm 1$, we transition to the reject state r.
- If we are in the reject state r and read any symbol, we stay in the reject state.

Finally, we accept if after processing a string we are still in the start state (meaning the string is empty) or in a digit state (meaning all the adjacent digits had a difference of at most 1).

Problem 2.

For each CFG G_i;

- Describe $\mathcal{L}(G_i)$ using set notation, as simply as possible.
- Either *draw* an NFA that recognizes $\mathcal{L}(G_i)$ or *prove* that $\mathcal{L}(G_i)$ is not regular.

(a) G_1

$$S \Rightarrow 0T0 \mid 1T1$$

$$T \Rightarrow 0T0 \mid 1T1 \mid X$$

$$X \Rightarrow AX \mid A$$

$$A \Rightarrow 0 \mid 1$$

(i) $\mathcal{L}(G_1) = \left\{xwx^R : x, w \in \{0,1\}^* \text{ and } |x| > 0, |w| > 0\right\}$

(ii) *Draw* an NFA that recognizes $\mathcal{L}(G_1)$.

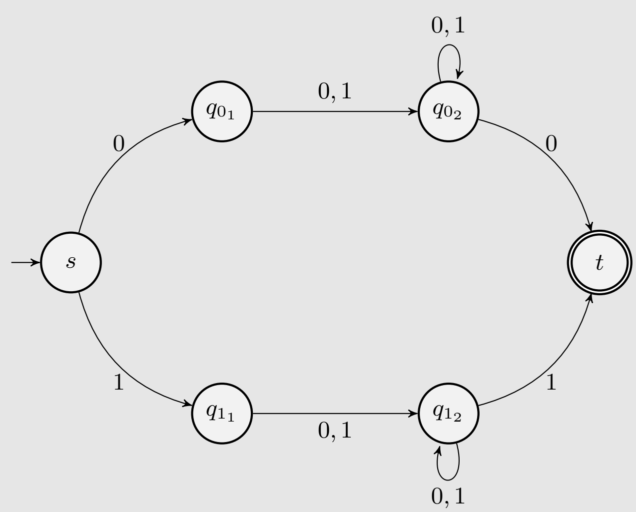

Figure 1. NFA for $\mathcal{L}(G_1)$

(b) G_2

$$S \Rightarrow 0X \mid 1Y$$
$$X \Rightarrow AXA \mid 0$$
$$Y \Rightarrow AYA \mid 1$$
$$A \Rightarrow 0 \mid 1$$

(i) $\mathcal{L}(G_2) = \left\{aw : a \in \{0,1\}, w \in \{0,1\}^* \text{ and the middle symbol of } w \text{ is } a\right\}$

(ii) *Prove* that $\mathcal{L}(G_2)$ is not regular or draw an NFA.

Claim 2.1. $L_2 = \mathcal{L}(G_2)$ *is not regular.*

Proof. Suppose L_2 is regular, and let p be the pumping length for L_2. Take $s = 10^p10^p$, then clearly $s \in L_2$. By the pumping lemma, there exists $u, v, w \in \{0,1\}^*$ such that $s = uvw$ and:

- $|uv| \le p$
- $|v| > 0$

  This gives us two possibilities:

  - If $u = \varepsilon$, then $v = 10^a$ for some $0 \le a \le p-1$, and $w = 0^{p-a}10^p$.
  - If $u \ne \varepsilon$, then $u = 10^a$, $v = 0^b$, and $w = 0^{p-a-b}10^p$, where $a + b \le p - 1$.
- $uv^kw \in L_2$ for all $k \ge 0$

  In the first case, pumping up the string tells us that $uv^2w = 10^b10^b0^{p-b}10^p \in L_2$.

  This implies that the middle symbol of $0^b10^p10^p$ is 1.

  For this to happen, either:

  - $b = 2p + 1$. This contradicts the condition that $b = |v| \le p$ (which follows from PL1).
  - $b + p + 1 = p$. This implies that $b = -1$, which contradicts the fact that we cannot have negative-length strings.

  Therefore, in the first case L_2 must not be regular.

  In the second case, pumping down the string tells us that $uw = 10^a0^{p-a-b}10^p \in L_2$.

  This implies that the middle symbol of $0^{p-b}10^p$ is 1.

  For this to happen, we must have that $p - b = p$, meaning $b = 0$, contradicting rule PL2 which says that $b = |v| > 0$.

  Therefore, in the second case L_2 must also not be regular.

□

(c) G_3

$$S \Rightarrow AAT \mid BBT$$

$$T \Rightarrow AAT \mid BBT \mid A \mid B$$

$$A \Rightarrow 0$$

$$B \Rightarrow 1$$

(i) $\mathcal{L}(G_3) = \{0^n 1^n : n \geq 0\}$

(ii) *Draw* an NFA that recognizes $\mathcal{L}(G_3)$.

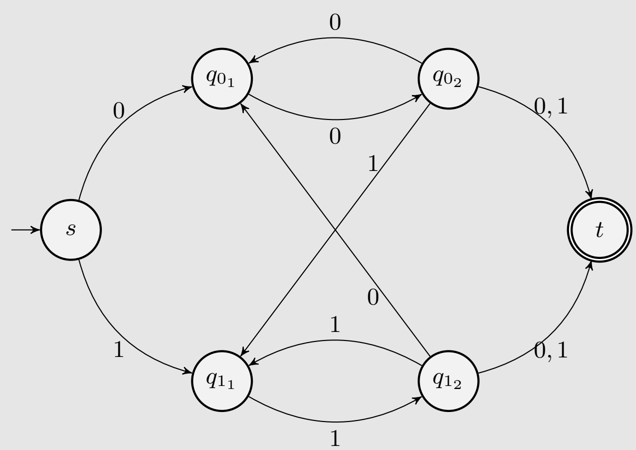

Figure 2. NFA for $\mathcal{L}(G_3)$

Problem 3.

Give a simple CFG that generates the language $L_3 = \left\{x \in \{0,1\}^* : x \neq x^R\right\}$.

Formally prove that your CFG is sound and complete.

$$S \to 0S0 \mid 0B1 \mid 1B0 \mid 1S1$$

$$B \Rightarrow 0B0 \mid 0B1 \mid 1B0 \mid 1B1 \mid \varepsilon$$

Problem 4.

For each $x \in \{0,1\}^*$, define grow(x) to be the string obtained by replacing every occurrence of '0' in x with '00'. For example:

$$\text{grow}(10110) = 1001100, \quad \text{grow}(000) = 000000, \quad \text{grow}(\varepsilon) = \varepsilon, \quad \text{grow}(11) = 11.$$

Let $P = (Q, \{0,1\}, \Gamma, \delta, q_0, F)$ be a PDA. Formally describe a PDA that recognizes $\{\text{grow}(x) : x \in \mathcal{L}(P)\}$. Do not assume anything about the design of P.

Give a high-level explanation of your construction (no formal completeness and soundness proofs are required).

Define a new PDA

$$P_2 = (Q_2, \{0,1\}, \Gamma, \delta_2, q_0, F)$$

where:

$$Q_2 = \bigcup_{q \in Q} \{q, q'\}$$

$$\delta_2(q, x, \gamma) = \begin{cases} \delta(q, x, \gamma) & \text{if } q \in Q \text{ and } x \neq 0 \\ r' & \text{where } r = \delta(q, x, \gamma)\text{, if } q \in Q \text{ and } x = 0 \\ r & \text{if } q = r' \text{ for some } r \in Q\text{, and } x = 0, \gamma = \varepsilon \end{cases}$$

Main Idea:

(i) We start at the same state. We also maintain the same accepting states.

(ii) We create a duplicate state q' for each state $q \in Q$.

(iii) Whenever there is an incoming transition $q \to r$ with a reading $x = 0$, we instead transition to r'. r' then only transitions to r if the next symbol is a 0. The transition from q to r' pushes to or pops from the stack as it would if transitioning directly from q to r in the original PDA. However, the transition from r' to r does not push or pop anything, making sure the stack remains consistent as it would if only the first transition had occurred.

(iv) All other transitions are handled the same way as in the original PDA.

Problem 5.

For a language L, define $\textsc{unique}(L) := \{x \in L : \nexists y \in L \text{ such that } |y| = |x|\}$.

In other words, $\textsc{unique}(L)$ is the set of all strings $x \in L$ such that x is the only string in L that has length $|x|$.

Is the class of regular languages closed under the operation $\textsc{unique}$? Prove your answer.

Problem 6.

Every language falls into into one of the following three categories:

(i) regular

(ii) context-free but not regular

(iii) not context-free

Which of these categories is the following language in?

$$L_6 = \left\{ x \in \{0,1\}^* : \exists y, z \in \{0,1\}^* \text{ such that } x = yz, |y| = |z|, \text{ and } \beta(y) \equiv 0 \pmod 3 \right\}.$$

Reminder: For a string $x \in \{0,1\}^*$, $\beta(x)$ is the string x interpreted as a binary number. For instance, $\beta(11001) = 25$ and $\beta(0011) = 3$. We also define $\beta(\varepsilon) = 0$.