CS 103: Mathematical Foundations of Computing
Problem Set #7

Quang Nguyen

May 31, 2022

Due Friday, November 12 at 2:30 pm Pacific

Do not put your answers to Problems 1, 4, and 5.ii. in this file.
Some symbols you may want to use here:

- The empty string is denoted ε.
- Alphabets are denoted Σ.
- The language of an automaton is denoted $\mathscr{L}(D)$.
- The Greek letter delta is denoted δ.
- You can say two strings are distinguishable relative to L by writing $x \not\equiv_L y$.
- The union symbol for regexes is denoted $\cup$.
- Kleene stars are denoted $L^\star$.
- You can type a brace character by writing { or }.
- The Optional Fun Problem uses the notation $\mathscr{F}$.

Problem Two: Regexes and Accessible Design

Neymar da Silva Santos Júnior is the name that does not match this regular expression. The name is made out of 4 words and the second one starts with a lowercase letter, not an uppercase.
Two more examples of names that aren't matched by this regular expression:
- Daniel Alves da Silva
- Ronaldo de Assis Moreira

Problem Three: Finite Languages

One simplest way is to list all the strings in L into the regular expression. This means that the regular expression covers all the strings in L.

Problem Five: Embracing the Braces

i.

Theorem: L_1 is not a regular language.
Proof: Let $S = \{\{^n | n \in \mathbb{N}\}$. We will prove that S is infinite and that S is a distinguishing set for L_1.
To see that S is infinite, note that S contains one string for each natural number.
To see that S is a distinguishing set for L_1, consider any strings $\{^m, \{^n \in S$ where $m \neq n$. Note that $\{^m\}^m \in L_1$ and that $\{^n\}^m \notin L_1$. Therefore, we see that $\{^m \not\equiv_{L_1} \{^n$, as required.
Since S is infinite and is a distinguishing set for L_1, by the Myhill-Nerode theorem we see that L_1 is not regular. ■

iii.

The error would be in the part building the distinguishing set S. Because the nesting depth is at most 4, the value of n in S can only be at most 4, which means that S is not an infinite set.

Problem Six: State Lower Bounds

i.

Theorem: If S is finite, then any DFA for L must have at least $|S|$ states.
Proof: Assume for the sake of contradiction that if S is finite, then any DFA for L has less than $|S|$ states. Pick a DFA D for L that D has less than $|S|$ states. Let m be a number of elements in S and n be the number of states in D such that $n < m$. By the pigeonhole principle, there are at least two elements $s_1 \in S$ and $s_2 \in S$ such that s_1 and s_2 end in the same state when running through D. Since S is the distinguishing set, we know that $s_1 \not\equiv_L s_2$. This means that when we run D on input s_1 and s_2, they must end up in different states, which is impossible. We have reached a contradiction, so our assumption must have been wrong. Therefore, if S is finite, then any DFA for L must have at least $|S|$ states. ■

ii.

Theorem: Any DFA for $TWEETS$ must have at least 282 states.
Proof: We want to show that any DFA for $TWEETS$ must have at least 282 states. Let $S = \{a^n \in \Sigma^\star | n \in \mathbb{N}, 0 \leq n \leq 281\}$. Notice that S has 282 elements in it. We will prove that S is a distinguishing set for $TWEETS$. Pick arbitrary $a^x \in S$ and $a^y \in S$ such that $x < y$. Consider $a^t \in \Sigma^\star$ such that $t = 280 - x$, we see that $a^x a^t = a^{x+t} = a^{x+280-x} = a^{280}$ and $a^y a^t = a^{y+t} > a^{280+0} = a^{280}$. This means that $a^x a^t \in TWEETS$ and $a^y a^t \notin TWEETS$. Therefore, S is the distinguishing set for $TWEETS$ where $|S| = 282$. By our earlier theorem, we know that any DFA for $TWEETS$ must have at least 282 states, as required. ■

iii.

$D = (Q, \Sigma, \delta, q_0, F)$ where

- $Q = \{q_n | n \in \mathbb{N}, 0 \leq n \leq 281\}$
- $\Sigma = \{\text{all Unicode characters}\}$
- δ is defined as follows:
$$\delta(q_k, a) = \begin{cases} q_{281}, & \text{if } k = 281 \\ q_{k+1}, & \text{otherwise} \end{cases}$$
- $F = \{q_n | n \in \mathbb{N}, 0 \leq n \leq 280\}$

Problem Seven: The Extended Transition Function

i.

Theorem: If $x, y \in \Sigma^\star$, then $\delta^\star(\delta^\star(q, x), y) = \delta^\star(q, xy)$.
Proof: Pick an arbitrary $x \in \Sigma^\star$. Let $P(y)$ be the statement "$\delta^\star(\delta^\star(q, x), y) = \delta^\star(q, xy)$". We will prove by induction that $P(y)$ holds for all $y \in \Sigma^\star$.
As our base case, we will show that $P(\varepsilon)$ is true, meaning that $\delta^\star(\delta^\star(q, x), \varepsilon) = \delta^\star(q, x\varepsilon)$. This is true because $\delta^\star(\delta^\star(q, x), \varepsilon) = \delta^\star(q, x)$ and $\delta^\star(q, x\varepsilon) = \delta^\star(q, x)$.
For our inductive step, assume for some arbitrary k that $P(k)$ is true, meaning that $\delta^\star(\delta^\star(q, x), k) = \delta^\star(q, xk)$. (1)
Pick an arbitrary $a \in \Sigma$. We will prove $P(ka)$, meaning that $\delta^\star(\delta^\star(q, x), ka) = \delta^\star(q, xka)$. Notice that

$$\begin{aligned}
\delta^\star(\delta^\star(q, x), ka) &= \delta(\delta^\star(\delta^\star(q, x), k), a) \\
&= \delta(\delta^\star(q, xk), a) && \text{(via (1))} \\
&= \delta^\star(q, xka). && \text{(by definition of } \delta^\star\text{)}
\end{aligned}$$

Therefore, $P(ka)$ holds, completing the induction. ■

ii. Fill in the blanks to Problem Seven, part ii. below.

- "The character a is in the alphabet of the DFA." $\underline{a \in \Sigma}$
- "The state that string w ends in when run through D." $\underline{\delta^\star(q_0, w)}$
- "D's start state is an accepting state." $\underline{q_0 \in F}$
- "Strings x and y end in the same state when run through D." $\underline{\delta^\star(q_0, x) = \delta^\star(q_0, y)}$
- "D accepts w." $\underline{\delta^\star(q_0, w) \in F}$

iii.

The language of D includes all strings w where w is a string composed of letters in Σ and the state outputted by inputting q_0 as a start state and w is an accepting state.