CMPS 130: HW 1

Kevin Wang

February 5, 2019

Solution to Problem 1

Definition 1.1. An all-NFA N is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ that accepts a string x if every possible state that N could be in after reading x is an accepting state.

Definition 1.2. A language $\mathcal{L}$ is regular if there exists some DFA A such that $L(A) = \mathcal{L}$.[1]

Lemma 1.1 (Hopcroft Theorem 2.11)**.** *If $D = (Q_D, \Sigma, \delta_D, q_{0_D}, F_D)$ is the DFA constructed from NFA $N = (Q_N, \Sigma, \delta_N, q_{0_N}, F_N)$ by the subset construction, then $L(D) = L(N)$*

Lemma 1.2 (Hopcroft Theorem 2.12)**.** *A language $\mathcal{L}$ is accepted by some DFA if and only if $\mathcal{L}$ is accepted by some NFA.*

Theorem 1. *All-NFAs recognize the class of regular languages.*

Proof. To prove that all-NFAs recognize the class of regular languages, we must show:

1. Every all-NFA recognizes some regular language.
2. Every regular language is recognized by some all-NFA.

Let $N = (Q_N, \Sigma, \delta_N, q_{0_N}, F_N)$ be an all-NFA that recognizes some language $L(N)$. If $L(N)$ is regular, then there is a DFA $M = (Q_M, \Sigma, \delta_M, q_{0_M}, F_M)$ that recognizes some language $L(N)$ such that $L(M) = L(N)$ (Definition 1.2). We construct M as follows:[2]

1. $Q_M = 2^{Q_N}$ (set of all subsets of Q_N)
2. Σ is the same
3. For $S \subseteq Q_N$,$S \in Q_M$ and $\exists a \in \Sigma$: $\delta_M(S, a) = \bigcup_{r \in R} \text{ECLOSE}(r)$, where $R = \bigcup_{s \in S} \delta_N(s, a)$
4. $q_{0_M} = \text{ECLOSE}(q_{0_N}) \in Q_M$, where $q_{0_N} \in Q_N$
5. $F_M = \{S | S \in Q_M, S \subseteq F_N\}$

Let $x \in \Sigma^*$. Let $\hat{\delta}_N$ be the set of states N could be in after reading x. DFA M then reads x, ending at state $\hat{\delta}_M$.

By definition of an all-NFA (Definition 1.1), $x \in L(N)$ if and only if $\hat{\delta}_N(q_{0_N}, x) \subseteq F_N \implies x \in \mathcal{L}$. Thus, every all-NFA recognizes some regular language.

By the construction of M, $x \in L(M) = \mathcal{L}$ if and only if $\hat{\delta}_N(q_{0_N}, x) \in F_M \implies \hat{\delta}_N(q_{0_N}, x) \subseteq F_N \implies x \in L(N)$.[3] Thus, every regular language is recognized by some all-NFA.

Therefore, all-NFAs recognize the class of regular languages. $\square$

[1] Referenced lec2.pdf
[2] Referenced lec3.pdf
[3] Referenced lec4.pdf

Solution to Problem 2

Definition 2.1. A language $\mathcal{L}$ is regular if there exists some DFA A such that $L(A) \iff \mathcal{L}$.[4]

Theorem 2. *Given a regular language $\mathcal{L}$ over the alphabet Σ, we can define a language:*

$$PREFIX(\mathcal{L}) = \{x | \exists w, xw \in \mathcal{L}\}$$

that is regular.

Proof. If $\mathcal{L}$ is regular, then there is a DFA $D = (Q, \Sigma, \delta, q_0, F)$ such that $L(D) = \mathcal{L}$ (Definition 2.1). We will prove PREFIX($\mathcal{L}$) is regular by constructing a DFA D' that accepts PREFIX($\mathcal{L}$). DFA $D' = (Q', \Sigma, \delta', q_0', F')$ is constructed as follows:

1. $Q' = Q$
2. Σ
3. $\delta' = \delta$
4. $q_0' = q_0$
5. For $S \subseteq Q$ and $\exists x \in \Sigma^*$, $F' = \{S | \delta(s \in S, x) = f \in F\}$

We end up with a DFA D' that is very similar to DFA D with the exception of the set of accepting states F'. By construction of D', we determine based on F' that $x \in$ PREFIX($\mathcal{L}$) if $x \in \mathcal{L}$.

As the set of accepting states F' of D' is the set of prerequisite states for reaching $f \in F$ of D, we can determine that by definition of the language PREFIX($\mathcal{L}$), containing all prefixes of strings in $\mathcal{L}$, $x \in \mathcal{L}$ if $x \in$ PREFIX($\mathcal{L}$).

Thus DFA D recognizes language PREFIX($\mathcal{L}$), proving that it is regular (Definition 2.1) $\square$

[4] Referenced lec2.pdf

Solution to Problem 3

Theorem 3. *All languages recognized by PFSAs are regular.*

Proof. Let PFSA $P_k = (Q, \Sigma, \delta, q_0, F)$ where k is the number of pebbles initially at q_0. Let DFA $D = (Q, \Sigma, \delta, q_0, F)$ accept some regular language $L(D)$. We first observe that P_1, a PFSA with 1 pebble, is equivalent to DFA D, as there is no non-deterministic choosing of a pebble to move. Thus, $L(P_1) = L(D)$ and thus $L(P_1)$ is regular.

Assume $L(P_k)$ is regular and accepted by some DFA $M = (Q, \Sigma, \delta, q_0, F)$. Observe that any input x accepted by a PFSA with k pebbles is also accepted by a PFSA with $n > k$ pebbles (by keeping $n - k$ pebbles at the start, q_0). Thus, we can conclude that $L(P_k) \subsetneq L(P_{k+1})$.
We then construct a DFA M' by adding a path from each state in DFA M to a copy of DFA D, such that DFA M' becomes a fractal of D of scale k. Thus $L(P_{k+1})$ is regular.

By induction, all languages accepted by PFSAs are regular. $\square$

Solution to Problem 4

Lemma 4.1 (Closure Under Complement)**.** *The complement of a regular language is regular.*

Lemma 4.2 (Pumping Lemma for Regular Languages)**.** *Let $\mathcal{L}$ be a regular language. There exists a constant n (which depends on $\mathcal{L}$) such that $\forall w \in \mathcal{L}$ such that $|w| \geq n$, we can break w into three strings, $w = xyz$ such that:*

1. *$y \neq \epsilon$.*
2. *$|xy| \leq n$.*
3. *For all $k \geq 0$, the string $xy^k z$ is also in $\mathcal{L}$.*

Theorem 4.1. *Let the alphabet $\Sigma = \{0, 1\}$. The language $\mathcal{L}$ – defined as the set of strings that are not of the form ss – is not regular.*

Proof. Assume $\mathcal{L}$ is regular. Let $\overline{\mathcal{L}}$ be the complement of language $\mathcal{L}$. We define this language as the set of strings that are exactly of the form ss. By the closure properties of regular languages – specifically Closure under Complement (Lemma 4.1) – $\overline{\mathcal{L}}$ must also be regular.

Let n be the number of states in $\overline{\mathcal{L}}$. Since $\overline{\mathcal{L}}$ is regular, the Pumping Lemma (Lemma 4.2) should hold. Let $w = 0^n 1^n 0^n 1^n$, satisfying $|w| = 4n \geq n$. Let $w = xyz$, where by the Pumping Lemma $|xy| \leq n$. Therefore, y consists of only 0's where $y \neq \epsilon$.

Let $k = 0$. The Pumping Lemma states that $xy^k z$ is in $\overline{\mathcal{L}}$, if $\overline{\mathcal{L}}$ is regular.[5] However, xz has $2n$ 1's since all the 1's of w are in z. But xz also has fewer than $2n$ 0's, because we lost the 0's of y. Since $y \neq \epsilon$ we know that there can be no more than $2n - 1$ 0's among both x and z. Thus xz is not a part of $\overline{\mathcal{L}}$ contradicting our prior assumption of $\overline{\mathcal{L}}$'s regularity.

Having proven by contradiction that $\overline{\mathcal{L}}$ is not regular, this contradicts the Closure under Complement properties (Lemma 4.1) thus proving that the language $\mathcal{L}$, is not regular. $\square$

Theorem 4.2. *Let the alphabet $\Sigma = \{0, 1\}$. The language $\mathcal{C} = \{1^i a | a$ has at most i 1's, $i \geq 0\}$ is not regular.*

Proof. Assume $\mathcal{C}$ is regular. Let $i = n$ be the number of states in $\mathcal{C}$. Since $\mathcal{C}$ is regular, the Pumping Lemma (Lemma 4.2) should hold. Let $w = 1^n 0^n 1^n$, satisfying $|w| = 3n \geq n$. Let $w = xyz$, where by the Pumping Lemma $|xy| \leq n$. Therefore, y consists of only 1's where $y \neq \epsilon$.

Let $k = 0$. The Pumping Lemma states that $xy^k z$ is in $\mathcal{C}$, if $\mathcal{C}$ is regular. As such, $xy \to x$ has at most $n - 1$ 1's since $y \neq \epsilon$. However, z still has n 1's since all the latter 1's of the string are in z. Hence, $a = xz$ does not have at most $i = n$ 1's. Thus xz is not a part of $\mathcal{C}$ contradicting our prior assumption, proving that $\mathcal{C}$ is not regular. $\square$

[5] Referenced Hopcroft 4.1

Solution to Problem 5

Lemma 5.1. *A language $\mathcal{L}$ is regular if and only if $\mathcal{L}$ can be described by a regular expression.*[6]

Theorem 5. *Let $\Sigma = \{0, 1\}$. Prove that language $\mathcal{B} = \{1^k y | y$ has at least k 1's, $k \geq 1\}$ is regular.*

Proof. We make the observation that any string in $\mathcal{B}$ start with at least one 1 (when $k = 1$). By definition of the substring y, the string also includes at least one other 1. As such, let language $\mathcal{B}_1 = \{1^1 y | y$ has at least 1 1's, $\}$. By construction, we note that $\mathcal{B}_1 \subset \mathcal{B}$ and thus any string w accepted by the language $\mathcal{B}_1$ is also accepted by $\mathcal{B}$. Thereby, we can construct a regular expression:

$$1^k(0^*1)^k\Sigma^* = 10^*1\Sigma^*$$

that defines this language. By Lemma 5.1, language $\mathcal{B}_1$ is regular, and thus the base case is satisfied.

Let $\mathcal{B}_k$ be a regular language where $\mathcal{B}_k \subset \mathcal{B}$.

$$\begin{aligned}\mathcal{B}_{k+1} &= 1^{k+1}(0^*1)^{k+1}\Sigma^* \\ &= 11^k0^*1(0^*1)^k\Sigma^*\end{aligned}$$

As such, we have:

$$\begin{aligned}\mathcal{B}_1 &= 10^*1\Sigma^* \\ \mathcal{B}_2 &= 110^*10^*1\Sigma^* \\ \mathcal{B}_3 &= 1110^*10^*10^*1\Sigma^* \\ \mathcal{B}_4 &= 11110^*10^*10^*10^*1\Sigma^* \\ &\vdots \\ \mathcal{B}_k &= 1^k(0^*1)^k\Sigma^* \\ &\vdots\end{aligned}$$

Thus, because there is a regular expression defining $\mathcal{B}$ ($\forall k \geq 1$) and $\mathcal{B} = \bigcup_{k\geq 1} \mathcal{B}_k$, $\mathcal{B}$ must be a regular language. $\square$

---

[6] Referenced lec5.pdf

Solution to Problem 6

Lemma 6.1 (Closure Under Concatenation)**.** *Given two regular languages $\mathcal{L}$ and $\mathcal{M}$, the language $\mathcal{L} \cdot \mathcal{M}$ is also regular.*

Theorem 6. *Regular languages are closed under rotational closure, where the rotational closure of a regular language $\mathcal{L}$ is defined as RC($\mathcal{L}$)= $\{xy|yx \in \mathcal{L}\}$.*

Proof. Let language $\mathcal{L} = \mathcal{X} \cdot \mathcal{Y}$ such that $x \in \mathcal{X}$, $y \in \mathcal{Y}$, and $xy \in \mathcal{L}$. As $\mathcal{L}$ is known to be regular, then by the converse of Lemma 6.1, languages $\mathcal{X}$ and $\mathcal{Y}$ must also be regular.

The rotational closure of $\mathcal{L}$ can thus be formed by the concatenation of $\mathcal{Y}$ and $\mathcal{X}$:

$$\begin{aligned}\mathrm{RC}(\mathcal{L}) &= \{xy|yx \in \mathcal{L}\} \\ &= \{xy|yx,\ y \in \mathcal{Y} \text{ and } x \in \mathcal{X}\} \\ &= \mathcal{Y} \cdot \mathcal{X}\end{aligned}$$

Thus by closure under concatenation (Lemma 6.1), RC($\mathcal{L}$) is regular – proving that regular languages are closed under rotational closure. $\square$

Solution to Problem 7

Lemma 7.1 (Closure Under Concatenation)**.** *Given two regular languages $\mathcal{L}$ and $\mathcal{M}$, the language $\mathcal{L} \cdot \mathcal{M}$ is also regular.*

Lemma 7.2 (Closure Under Intersection)**.** *Given two regular languages $\mathcal{L}$ and $\mathcal{M}$, the language $\mathcal{L} \cap \mathcal{M}$ is also regular.*

Lemma 7.3 (Closure Under Difference)**.** *Given two regular languages $\mathcal{L}$ and $\mathcal{M}$, the language $\mathcal{L} - \mathcal{M}$ is also regular.*

Theorem 7. *Given two regular languages $\mathcal{L}$ and $\mathcal{M}$, the language $\mathcal{L} \ominus \mathcal{M}$ is also regular, where:*

$$\mathcal{L} \ominus \mathcal{M} = \{x | x \in \mathcal{L} \textit{ and } x \textit{ does not contain any string of } \mathcal{M} \textit{ as substring}\}$$

Proof. Let $\mathcal{L} \oplus \mathcal{M} = \{x | x \in \mathcal{L}$ and x contains some string of $\mathcal{M}$ as substring$\}$. Observe this is equal to the expression $\mathcal{L} \cap (\Sigma^* \cdot \mathcal{M} \cdot \Sigma^*)$. [7]

By definition,

$$\begin{aligned} \mathcal{L} \ominus \mathcal{M} &= \mathcal{L} - (\mathcal{L} \oplus \mathcal{M}) \\ &= \mathcal{L} - (\mathcal{L} \cap (\Sigma^* \cdot \mathcal{M} \cdot \Sigma^*)) \\ &= \mathcal{L} - (\Sigma^* \cdot \mathcal{M} \cdot \Sigma^*) \end{aligned}$$

Therefore by closure under concatenation, intersection, and difference (Lemmas 7.1, 7.2, and 7.3), $\mathcal{L} \ominus \mathcal{M}$ is regular – proving closure under substring difference. □

---

[7] Wasn't sure if $\mathcal{L} \oplus \mathcal{M} = \overline{\mathcal{L} \ominus \mathcal{M}}$. Otherwise would've used Closure under Complement.