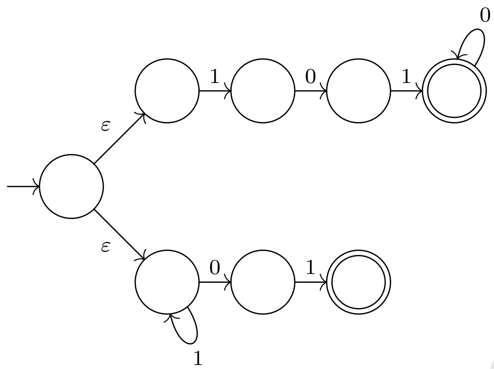
CS 360 MidTerm 1 Practice Problems
Solutions

1. Let $\Sigma = \{0, 1\}$. Write a regular expression for the language of the following ε-NFA.



Solution: From the top branch, we obtain 1010^*.

From the bottom branch, we obtain 1^*01.

The ε-transitions from the start state will lead to acceptance if we follow either branch to an accept state. Therefore the regular expression for the automaton is $1010^* + 1^*01$.

2. Prove or disprove each of the following statements about regular expressions.

   (a) $(R + S)^* = R^* + S^*$

   Solution: This identity is false. We present a counterexample in which the left hand side and right hand side are not equal.

   Let

$$\begin{aligned} \Sigma &= \{0, 1\}, \\ R &= 0, \text{ and} \\ S &= 1. \end{aligned}$$

   Then by definition, R and S are regular expressions over Σ. I claim that $01 \in L((0 + 1)^*)$ but $01 \notin L(0^* + 1^*)$.

   Proof that $01 \in L((0 + 1)^*)$:

$$\begin{aligned} 0 &\in L(0 + 1), \text{ and} \\ 1 &\in L(0 + 1), \text{ so} \\ 01 &\in L((0 + 1)^*). \end{aligned}$$

   Proof that $01 \notin L(0^* + 1^*)$:

$$\begin{aligned} 01 &\notin L(0^*), \text{ and} \\ 01 &\notin L(1^*), \text{ so} \\ 01 &\notin L(0^* + 1^*). \end{aligned}$$

As we have exhibited a word in the language of the left hand side and not in the language of the right hand side. Therefore these regular expressions are not equal, as we claimed.

(b) $(RS+R)^*R = R(SR+R)^*$

Solution: This identity is true.

Proof that $L((RS+R)^*R) \subseteq L(R(SR+R)^*)$: Let $w \in L((RS+R)^*R)$ be arbitrary. The proof is by induction on the number of words from $L(RS+R)$ included in the construction of w.

Base (0 words): Write $w = r$, for some $r \in L(R)$. Then it is clear that $w \in L(R(SR+R)^*)$, so the base case holds.

Induction ($n \geq 1$ words): The induction hypothesis is that any $x \in L(R(SR+R)^*)$ with fewer than n words from $L(RS+R)$ is in $L(R(SR+R)^*)$. Write

$$w = w_1 \underbrace{w_2 \cdots w_n r_1}_{\text{I. H. applies}}, \text{ for some } w_1, \ldots, w_n \in L(RS+R) \text{ and some } r_1 \in L(R).$$

We have the following possibilities for $w_1 \in L(RS+R)$.

- $w_1 = rs$, for some $r \in L(R)$ and $s \in L(S)$: Then we have

$$\begin{aligned} w_2 \cdots w_n r_1 &\in L(R(SR+R)^*), \text{ by the induction hypothesis} \\ s w_2 \cdots w_n r_1 &\in L(SR(SR+R)^*) \subseteq L((SR+R)^*) \\ r s w_2 \cdots w_n r_1 &\in L(R(SR+R)^*) \checkmark \end{aligned}$$

- $w_1 = r$, for some $r \in L(R)$: Then we have

$$\begin{aligned} w_2 \cdots w_n r_1 &\in L(R(SR+R)^*), \text{ by the induction hypothesis} \\ r s w_2 \cdots w_n r_1 &\in L(RR(SR+R)^*) \subseteq L(R(SR+R)^*) \checkmark \end{aligned}$$

So the containment is established.

Proof that $L((RS+R)^*R) \supseteq L(R(SR+R)^*)$: Let $w \in L(R(SR+R)^*)$ be arbitrary. The proof is by induction on the number of words from $L((SR+R)^*)$ included in the construction of w.

Base (0 words): Write $w = r$, for some $r \in L(R)$. Then it is clear that $w \in L((RS+R)^*R)$, so the base case holds.

Induction ($n \geq 1$ words): The induction hypothesis is that any $x \in L(R(SR+R)^*)$ with fewer than n words from $L((SR+R)^*)$ is in $L((RS+R)^*R)$. Write

$$w = \underbrace{r_1 w_1 \cdots w_{n-1}}_{\text{I. H. applies}} w_n, \text{ for some } w_1, \ldots, w_n \in L(SR+R) \text{ and some } r_1 \in L(R).$$

We have the following possibilities for $w_n \in L(SR+R)$.

- $\underline{w_n = sr, \text{ for some } s \in L(S) \text{ and } r \in L(R)}$: Then we have

$$\begin{aligned} r_1 w_1 \cdots w_{n-1} &\in L((RS+R)^*R), \text{ by the induction hypothesis} \\ r_1 w_1 \cdots w_{n-1} s &\in L((RS+R)^*RS) \subseteq L((RS+R)^*) \\ r_1 w_1 \cdots w_{n-1} sr &\in L((RS+R)^*R) \surd \end{aligned}$$

- $\underline{w_n = r, \text{ for some } r \in L(R)}$: Then we have

$$\begin{aligned} r_1 w_1 \cdots w_{n-1} &\in L((RS+R)^*R), \text{ by the induction hypothesis} \\ r_1 w_1 \cdots w_{n-1} r &\in L((RS+R)^*RR) \subseteq L((RS+R)^*R) \surd \end{aligned}$$

So the containment is established.

(c) $(RS+R)^*RS = (RR^*S)^*$

Solution: This identity is false. We present a counterexample in which the left hand side and right hand side are not equal.

Let

$$\begin{aligned} \Sigma &= \{0, 1\}, \\ R &= 0, \text{ and} \\ S &= 1. \end{aligned}$$

Then by definition, R and S are regular expressions over Σ. It is clear that $\varepsilon \in L((00^*1)^*)$ but $\varepsilon \notin L((01+0)^*01)$.

As we have exhibited a word in the language of the right hand side and not in the language of the right hand side. Therefore these regular expressions are not equal, as we claimed.

3. Prove that each of the following languages is <u>not</u> regular.

(a) Let $\Sigma = \{(,)\}$. L is all strings of well-balanced parentheses. Examples of words in L are (), (()), (()())(()).

Solution: The proof is by the Pumping Lemma. Let n be a positive integer. Let $x = \underbrace{(\cdots(}_{n \text{ copies}}\underbrace{)\cdots)}_{n \text{ copies}}$. It is clear that $x \in L$ and satisfies our definition of a long word. For any decomposition $x = uvw$ with $|uv| \leq n$ and $|v| \geq 1$, we have that uv is composed of all (s, and therefore so is v. But then $uw \notin L$, as it contains fewer (s than)s. Thus x cannot be pumped. Thus, by the Pumping Lemma, L is not regular.

(b) Let $\Sigma = \{0, 1\}$. $L = \{0^n \mid n \text{ is a perfect square }\}$.

Solution: The proof is by the Pumping Lemma. Let n be a positive integer. Let $x = 0^{(n+1)^2}$. It is clear that $x \in L$ and satisfies our definition of a long word. For any decomposition $x = uvw$ with $|uv| \leq n$ and $|v| \geq 1$, we have that uv is composed of all 0s, and therefore so is v. Write $v = 0^i$, for some $1 \leq i \leq n$.

We need to record one key fact about the difference between consecutive perfect squares before we proceed. Let k be any non-negative integer. Then we note that the difference between the consecutive perfect squares $(k+1)^2$ and k^2 is

$$\begin{aligned} &= (k+1)^2 - k^2 \\ &= k^2 + 2k + 1 - k^2 \\ &= 2k + 1 \end{aligned} \tag{1}$$

Now we must show that all possible choices $1 \leq i \leq n$ lead to a word $uw \notin L$.

- If $i = 1$, then $uw = 0^{(n+1)^2-1}$, and by the fact on line (1), $(n+1)^2 - 1$ cannot be a perfect square since $(n+1)^2$ is a perfect square.
- If $i = 2$, then $uw = 0^{(n+1)^2-2}$, and by the fact on line (1), $(n+1)^2 - 2$ cannot be a perfect square since $(n+1)^2$ is a perfect square.
- $\vdots$
- If $i = n$, then $uw = 0^{(n+1)^2-n}$, and by the fact on line (1), $(n+1)^2 - n$ cannot be a perfect square since $(n+1)^2$ is a perfect square.

But then, for all possible choices for i, we see that $uw \notin L$. Thus x cannot be pumped. Thus, by the Pumping Lemma, L is not regular.

(c) Let $\Sigma = \{0, 1\}$. $L = \{0^i 1^j \mid \gcd(i, j) = 1\}$.

Solution: The proof is by the Pumping Lemma. Let n be a positive integer. Let $i = p$, for some prime p satisfying $p > n + 1$. There is always a large enough prime available. Let $j = (1)(2) \cdots (p-1)$. Then by construction, we have that $\gcd(i, j) = 1$.

Let $x = 0^i 1^j$. It is clear that $x \in L$ and satisfies our definition of a long word. For any decomposition $x = uvw$ with $|uv| \leq n$ and $|v| \geq 1$, we have that uv is composed of all 0s, and therefore so is v. Write $v = 0^i$, for some $1 \leq i \leq n$.

Now we must show that all possible choices $1 \leq i \leq n$ lead to a word $uw \notin L$.

- If $i = 1$, then $uw = 0^{p-1}1^j$, and $\gcd(p-1, j) = p - 1 \neq 1$ if $p > 2$.
- If $i = 2$, then $uw = 0^{p-2}1^j$, and $\gcd(p-2, j) = p - 2 \neq 1$ if $p > 3$.
- $\vdots$
- If $i = n$, then $uw = 0^{p-n}1^j$, and $\gcd(p-n, j) = p - n \neq 1$ if $p > n + 1$.

All the lower bounds on p are satisfied, by the choice of p.

But then $uw \notin L$, by the definition of L. Thus x cannot be pumped. Thus, by the Pumping Lemma, L is not regular.

4. (a) If L is a language over Σ and $a \in \Sigma$ is a symbol, then we define the quotient of L and a, denoted L/a by

$$L/a = \{w \mid wa \in L\}.$$

For example, if $\Sigma = \{a, b\}$ and $L = \{a, aab, baa\}$, then $L/a = \{\varepsilon, ba\}$. Prove that if L is regular, then so is L/a.

Solution: Let L be a regular language. Let D be a DFA for language L, with accepting states $F \subset Q$. Construct a new DFA D' for L/a, as follows:

- Take the same states and transitions as in D.
- For each accept state f of D, find all states q in D with a transition into f for the alphabet symbol a. Declare each such state q to be an accept state of D'.

Then, by construction,

$$\begin{array}{rcl} D' & \text{accepts} & w \\ \Leftrightarrow D & \text{accepts} & wa \\ \Leftrightarrow wa & \in & L \\ \Leftrightarrow w & \in & L/a. \end{array}$$

Now D' is a DFA, because D is. D' accepts L/a, and therefore L/a is regular.

(b) If L is a language over Σ and $a \in \Sigma$ is a symbol, then we define $a \backslash L$ by

$$a \backslash L = \{w \mid aw \in L\}.$$

For example, if $\Sigma = \{a, b\}$ and $L = \{a, aab, baa\}$, then $a \backslash L = \{\varepsilon, ab\}$. Prove that if L is regular, then so is $a \backslash L$.

Solution: Let L be a regular language. Recall from class that the reversal of a regular language is regular. Then we have

$$\begin{array}{rcl} a \backslash L & = & \{w \mid aw \in L\} \\ & = & \{w \mid (aw)^R \in L^R\} \\ & = & \{w \mid w^R a^R \in L^R\} \\ & = & \{w \mid w^R a \in L^R\}, \text{ as } a \text{ is a single character} \\ & = & \{(w^R)^R \mid w^R a \in L^R\} \\ & = & \{(w^R) \mid w^R a \in L^R\}^R \\ & = & \{v \mid va \in L^R\}^R, \text{ letting } v = w^R \\ & = & [(L^R)/a]^R. \end{array}$$

Now note that

- L is regular, therefore L^R is regular.
- L^R is regular, therefore $(L^R)/a$ is regular, by part 4a.

- $(L^R)/a$ is regular, therefore $[(L^R)/a]^R$ is regular.

And so we are done.

5. (a) Suppose L is a regular language over an alphabet Σ. Give an algorithm to tell whether $L = \Sigma^*$, i.e. all possible strings over the given alphabet.

Solution:

- As L is regular, so is the complement L' (as proved in class).
- So obtain a DFA D' for L' from Kleene's Theorem.
- Determine whether $L(D') = \emptyset$, by our test from class.
  - If $L(D') = \emptyset$, then $L = \Sigma^*$.
  - Otherwise, i.e. if $L(D') \neq \emptyset$, then $L \neq \Sigma^*$.

(b) Suppose L_1, L_2 are regular languages over an alphabet Σ. Give an algorithm to tell whether L_1 and L_2 have at least one word in common.

Solution:

- Obtain DFAs M_1 and M_2 for L_1 and L_2, respectively.
- Construct an ε-NFAs M for $L_1 \cap L_2$, as in class.
- Test whether $L(M) = \emptyset$, by our test from class.
  - If $L(M) = \emptyset$, then L_1 and L_2 have no words in common.
  - Otherwise, i.e. if $L(M) \neq \emptyset$, then L_1 and L_2 have at least one word in common.