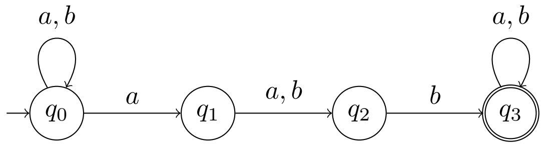
Homework 2
CS 770: Formal Language Theory

Problem 1. [Category: Design+Proof] Consider the language $A_2 \subseteq \{a,b\}^*$, from Homework 1, which was defined to be the collection of strings w where there is a position i in w such that the symbol at position i (in w) is a, and the symbol at position $i+2$ is b.

1. Design an NFA for language A_2 that has at most 4 states. You need not prove that your construction is correct, but the intuition behind your solution should be clear and understandable. **[5 points]**
2. Prove that any DFA recognizing A_2 has at least 5 states. **[5 points]**

Solution:

1. The NFA N_2 for the language A_2 as it reads the input, will "guess" some point when an a will be followed by a b two positions away. When it makes such a guess, it will check that the symbol two positions away is indeed a b and if so it will accept no matter what else follows. We could draw this NFA as follows. More generally, the same idea can be used to construct an NFA N_k with $k+2$ states that



$$\begin{tikzcd} \arrow[r] & q_0 \arrow[loop above, "{a,b}"] \arrow[r, "a"] & q_1 \arrow[r, "{a,b}"] & q_2 \arrow[r, "b"] & \boxed{q_3} \arrow[loop above, "{a,b}"] \end{tikzcd}$$

recognizes A_k. The formal definition of such an NFA would be as follows. $N_k = (Q_k^N, \{a,b\}, \delta_k^N, q_0, F_k^N)$ where

- $Q_k^N = \{q_0, q_1, \ldots q_{k+1}\}$
- $F_k^N = \{q_{k+1}\}$
- And the transition function is given as

$$\delta(q,c) = \begin{cases} \{q_0, q_1\} & \text{if } q = q_0 \text{ and } c = a \\ \{q_0\} & \text{if } q = q_0 \text{ and } c = b \\ \{q_{i+1}\} & \text{if } q = q_i,\ 0 < i < k \\ \{q_{k+1}\} & \text{if } q = q_k \text{ and } c = b \\ \{q_{k+1}\} & \text{if } q = q_{k+1} \end{cases}$$

and $\delta(q,c) = \emptyset$ in all other cases.

2. Let $M = (Q, \{a,b\}, \delta, q_0, F)$ be any DFA that recognizes A_2. We will argue that $|Q| \geq 5$. Let $u_A = aa$, $u_B = ab$, $u_C = ba$, $u_D = bb$, and $u_E = aab$. Let us denote by A, B, C, D, and E the states of M such that $q_0 \xrightarrow{u_A}_M A$, $q_0 \xrightarrow{u_B}_M B$, $q_0 \xrightarrow{u_C}_M C$, $q_0 \xrightarrow{u_D}_M D$, and $q_0 \xrightarrow{u_E}_M E$. Our main claim is that each of the states A, B, C, D, E are distinct. The proof of this fact is going to be based on the following observation. Suppose for some $p, q \in \{A, B, C, D, E\}$ we have $p = q$. Then since $q_0 \xrightarrow{u_p}_M p = q$ and $q_0 \xrightarrow{u_q}_M q = p$, for any string w, $\hat{\delta}_M(q_0, u_p w) = \hat{\delta}_M(q_0, u_q w)$. Thus, $u_p w \in A_2$ iff $u_q w \in A_2$. So to show that $p \neq q$, we will find a string w such that $u_p w \in A_2$ and $u_q w \notin A_2$ (or vice versa). Thus, in the various cases below we just list the "witness" string w in each case.

- **Case** $\{E\} \cap \{A, B, C, D\} = \emptyset$: Take $w = \epsilon$. Observe that $aab\epsilon = aab \in A_2$, but $aa\epsilon = aa$, $ab\epsilon = ab$, $ba\epsilon = ba$, and $bb\epsilon = bb$ are not in A_2. Thus, $E \neq A$, $E \neq B$, $E \neq C$, and $E \neq D$.
- **Case** $\{A, B\} \cap \{C, D\} = \emptyset$: Take $w = b$. Observe that $u_A w = aab$, and $u_B w = abb$ are members of A_2, but $u_C w = bab$ and $u_D w = bbb$ are not in A_2. Thus, $A \neq C$, $A \neq D$, $B \neq C$, and $B \neq D$.
- **Case** $A \neq B$: Consider $w = ab$. Now $u_A w = aaab \in A_2$ but $u_B w = abab \notin A_2$. Hence $A \neq B$.
- **Case** $C \neq D$: Consider $w = bb$. $u_C w = babb \in A_2$, but $u_D w = bbbb \notin A_2$. Hence, $C \neq D$.

Based on the above cases, and the argument above, we can conclude that all the states A, B, C, D, E must be distinct, and M has at least 5 states.

■

Problem 2. [Category: Design+Proof] For a string $w \in \Sigma^*$, let w^R denote the reverse of w, i.e., if $w = w_1 w_2 \cdots w_n$, where $w_i \in \Sigma$ then $w^R = w_n w_{n-1} \cdots w_1$. For a language L, let $L^R = \{w^R \mid w \in L\}$. Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA.

1. Design a *DFA* M^R that recognizes $\mathbf{L}(M)^R$, i.e., $\mathbf{L}(M) = (\mathbf{L}(M))^R$. **[5 points]**
2. Prove that your DFA M^R in the previous part is correct. **[5 points]**

Solution:

1. One way to construct the DFA M^R is by first constructing an NFA N^R that recognizes $(\mathbf{L}(M))^R$ and then converting N^R into a DFA using the algorithm described in class. The NFA N^R is constructed by "reversing" the direction of transitions of M, has a new initial state that has ϵ-transitions to the final states of M.

   The other way is to directly construct a DFA M^R that runs M in "reverse" on the input. This machine is in some sense combining both the steps in the previous approach directly. This is the one that we will present here. The DFA M^R will have as states subsets of states of M; after reading an input u, M^R will be in a set of states A iff M can reach a final state on input u^R from each state in A. We now define this precisely.

   Let $M = (Q, \Sigma, \delta, q_0, F)$. The DFA $M^R = (Q^R, \Sigma, \delta^R, q_0^R, F^R)$ where

   - $Q^R = 2^Q$
   - $q_0^R = F$
   - $F^R = \{A \subseteq Q \mid q_0 \in A\}$
   - And the transition function δ^R is described as

   $$\delta^R(A, a) = \{q \in Q \mid \delta(q, a) \in A\}$$

2. The correctness can be established by capturing the relationship between computations of M and computation of M^R which was stated informally before. The statement to be proved by induction is

   $$\forall w \in \Sigma^*.\ q_0^R \xrightarrow{w}_{M^R} A \text{ iff } \forall q \in A.\ \hat{\delta}_M(q, w^R) \subseteq F$$

   That is M^R reaches a state A (from its initial state) on w iff A is the collection of all states from which w^R is accepted. Observe that proving this statement allows us to establish the correctness of M^R as follows: M^R accepts w iff (by definition of F^R) $q_0^R \xrightarrow{w}_{M^R} A$ where $q_0 \in A$ iff (by the statement) for every $q \in A$, $\hat{\delta}_M(q, w^R) \subseteq F$ iff (since $q_0 \in A$) $\hat{\delta}_M(q_0, w^R) \subseteq F$ iff M accepts w^R.

   We are now ready to prove the statement relating M and M^R by induction on $|w|$.

- **Base Case:** Let w be such that $|w| = 0$. Then $w = \epsilon$. Since M^R and M are DFAs, $q_0^R = F \xrightarrow{\epsilon}_{M^R} q_0^R$ and for any $q \in Q$, $q \xrightarrow{\epsilon}_M q$. Thus, $\hat{\delta}_M(q,\epsilon) \subseteq F$ iff $q \in F = q_0^R$, which establishes the base case.
- **Induction Hypothesis:** For all w such that $|w| < n$, $q_0^R \xrightarrow{w}_{M^R} A$ iff $\forall q \in A.\ \hat{\delta}_M(q, w^R) \subseteq F$.
- **Induction Step:** Let $w = ua$, where $u \in \Sigma^{n-1}$ and $a \in \Sigma$. The induction step can be established by the following reasoning.

$$\begin{aligned}
q_0^R \xrightarrow{ua}_{M^R} A \quad & \text{iff } \exists B \in Q^R.\ q_0^R \xrightarrow{u}_{M^R} B \xrightarrow{a}_{M^R} A \text{ where } \delta^R(B,a) = A \\
& \text{iff } A = \{q \in Q \mid \delta(q,a) \in B\} \text{ and } \forall q' \in B.\ \hat{\delta}_M(q, u^R) \subseteq F \text{ because of ind. hyp. and defn. of } \delta^R \\
& \text{iff } \forall q \in A.\ \hat{\delta}_M(q, au^R) \subseteq F
\end{aligned}$$

Observe that since $(ua)^R = au^R$, the last line above establishes the induction step.

■

Problem 3. [Category: Comprehension+Design] An ***all***-NFA M is a 5 tuple $(Q, \Sigma, \delta, q_0, F)$ like an NFA, where Q is a finite set of states, Σ is the input alphabet, $\delta : Q \times (\Sigma \cup \{\epsilon\}) \to 2^Q$ is the transition function, $q_0 \in Q$ is the initial state, and $F \subseteq Q$ is the set of final states. The only difference between an ***all***-NFA and an NFA is that M accepts $u \in \Sigma^*$ iff every possible state that M could be in after reading u is in F (and at least one state is in F, i.e., all threads cannot die).

1. Taking $q_1 \xrightarrow{w}_M q_2$ to be the same as definition on page 17 in lecture 3, define formally when an ***all***-NFA M accepts u, and the language recognized by M (definitions similar to definitions on page 21 in lecture 3). **[4 points]**
2. Give a formal definition of a DFA dfa(M) such that $\mathbf{L}(\text{dfa}(M)) = \mathbf{L}(M)$. You need not prove your construction to be correct. **[6 points]**

Solution:

1. An ***all***-NFA M accepts w iff there is $q \in F$ such that $q_0 \xrightarrow{w}_M q$ and for every q' if $q_0 \xrightarrow{w}_M q'$ then $q' \in F$. We could also define it using $\hat{\delta}$ as follows. Let $\hat{\delta}_M(q_1, w) = \{q_2 \in Q \mid q_1 \xrightarrow{w}_M q_2\}$. Then M accepts w iff $\hat{\delta}_M(q_0, w) \neq \emptyset$ and $\hat{\delta}_M(q_0, w) \subseteq F$.

   The language accepted/recognized by M is given by

$$\mathbf{L}(M) = \{w \in \Sigma^* \mid M \text{ accepts } w\}$$

2. The DFA that recognizes the same language as M is going to have the same states and transitions as the DFA that we constructed for NFAs. The only difference will be in terms of the final states. Formally the DFA dfa(M) $= (2^Q, \Sigma, \delta', q_0', F')$ where

   - $q_0' = \hat{\delta}_M(q_0, \epsilon)$
   - $F' = 2^F \setminus \{\emptyset\}$, i.e., it is the set of all non-empty subsets of F.
   - $\delta'(A, a) = \cup_{q \in A} \hat{\delta}_M(q, a)$

   The correctness proof (which you were not required to do) is identical to the correctness proof the DFA constructed which is equivalent to NFAs. Recall from lecture 5, for an NFA N and DFA dfa(N) we can prove

- $\forall w \in \Sigma^*,\ \hat{\delta}_{\det(N)}(q_0', w) = \{A\}$ iff $\hat{\delta}_N(q_0, w) = A$.

Since for an all-NFA M, the DFA construction is the same in terms of states and transtions, this property holds here as well (and we don't need to reprove it). That is,

$$\forall w \in \Sigma^*,\ \hat{\delta}_{\det(M)}(q_0', w) = \{A\} \text{ iff } \hat{\delta}_M(q_0, w) = A$$

Using this observation correctness can be established as follows:

$$\begin{array}{llll} w \in \mathbf{L}(M) & \text{iff} & \hat{\delta}_M(q_0, w) = A \text{ and } A \neq \emptyset \text{ and } A \subseteq F & \text{defn. of acceptance} \\ & \text{iff} & \hat{\delta}_{\det(M)}(q_0', w) = \{A\} \text{ and } A \neq \emptyset \text{ and } A \subseteq F & \text{correctness property above} \\ & \text{iff} & \hat{\delta}_{\det(M)}(q_0', w) = \{A\} \text{ and } A \in F' & \text{defn. of } F' \\ & \text{iff} & w \in \mathbf{L}(\det(M)) & \text{defn. of language} \end{array}$$

■

Problem 4. [Category: Comprehension+Design]

1. Describe the language of the following regular expressions. A clear, crisp one-level interpretable English description is acceptable, like "This is the set of all binary strings with at least three 0s and at most hundred 1s", or like "$\{0^n(10)^m \mid n$ and m are integers$\}$". A vague, recursive or multi-level-interpretable description is not, like "This is a set of binary strings that starts and ends in 1, and the rest of the string starts and ends in 0, and the remainder of the string is a smaller string of the same form!" or "This is a set of strings like 010, 00100, 0001000, and so on!". You need not prove the correctness of your answer.
   (a) $(0^* \cup 0 \cup 1^*)^*$ **[1 points]**
   (b) $0(10)^*1$ **[2 points]**
   (c) $1^*(0 \cup 111^*)^*1^*$ **[2 points]**
2. Give regular expressions that accurately describe the following languages. You need not prove the correctness of your answer.
   (a) All binary strings with no more than three 0s. **[1 points]**
   (b) All binary strings such that in every prefix, the number of 0s and 1s differ by at most 1. **[2 points]**
   (c) All binary strings with exactly one occurrence of the substring 000. **[2 points]**

Solution: 1(a) The set of all binary strings.

1(b) The set of all binary strings with alternating 0s and 1s, which begin with a 0 and end with a 1.

1(c) The set of all binary strings that don't have 010 as a substring.

2(a) $1^*(0 \cup \epsilon)1^*(0 \cup \epsilon)1^*(0 \cup \epsilon)1^*$

2(b) $(01 \cup 10)^*(0 \cup 1 \cup \epsilon)$

2(c) $(1 \cup 01 \cup 001)^*000(1 \cup 10 \cup 100)^*$ ■