# Problem Set 2

**Due:** Thursday, February 12 at 12:30 in class (i.e., SC 1105)

**Submission instructions:** Submit each problem on a **separate sheet of paper**, put your name on each sheet, and write your discussion section time and day (e.g. Tuesday 10am) in the upper righthand corner. These details may sound picky, but they make the huge pile of homeworks much easier to grade quickly and more importantly, since we return them in the discussion sections, easier for you to get them back.

Also, write on each exercise the name/netid of your group members.

1. Intersect 'em
   [**Category**: Category: Construction, **Points**: 10]

   You are given two NFAs $A_1 = (P, \Sigma, \delta_1, p_0, F_1)$ and $A_2 = (Q, \Sigma, \delta_2, q_0, F_2)$.

   Construct an NFA that will accept the language $L(A_1) \cap L(A_2)$ with no more than $|P| * |Q|$ states. Also, prove that it indeed accepts the language of the intersection as stated above.

   *Hint: You may want to think of a product construction. And you may want to look at the proof of the correctness of the product construction.*

2. Express yourself
   [**Category**: Category: Comprehension, **Points**: 10]

   (a) What is the language of each of the following regular expressions?
   *Note: A clear, crisp and one-level-interpretable English description is acceptable, like "This is the set of all binary strings with at least three zeros and at most a hundred ones", or like* $\{0^n(10)^m : n \text{ and } m \text{ are integers}\}$. *A vague, recursive or multi-level-interpretable description is not acceptable, like: "This is the set of binary strings such that they start and end by 1, and the rest of string starts and ends by 0, and the remainder of string is a smaller string of the same form!", or "This is the set of strings like* 010, 00100, 0001000 *and so on!!".*

   (1) $(0^* + 1 + 1^*)^*$
   (2) $(1 + \epsilon)(00^*1)^*0^*$
   (3) $0^*(1 + 000^*)^*0^*$
   (4) $(0^*1^*)000(0 + 1)^*$
   (5) $1^* + 0^* + (00^*11^*00^*11^*)^* + (11^*00^*11^*00^*)^*$

   (b) Write down a regular expression for each case below, that represents the desired language. (Binary strings are strings over the alphabet $\{0, 1\}$.)

   (1) All binary strings that their third character from the end is zero.
   (2) All binary strings which have 111 as a substring.
   (3) All binary strings that have 00 as a substring but do not contain 011 as a substring.
   (4) All binary strings such that in every prefix of the string, the number of one's and zero's differ by at most one.

(5) All binary strings such that every pair of consecutive zero's appears before any pair of consecutive one's.

3. Modifying DFAs
[**Category**: Comprehension, **Points**: 5+5]

Suppose that $M = (Q, \Sigma, \delta, q_0, F)$ and $N = (R, \Sigma, \gamma, r_0, G)$ are two DFAs sharing the common alphabet $\Sigma = \{a, b\}$.

(a) Define a new DFA $M' = ((Q \cup \{q_x\}) \times \{0, 1\}, \Sigma \cup \{\#\}, \delta', (q_0, 0), F')$ whose transition function is defined as follows

$$\delta'((q,i),t) = \begin{cases} (\delta(q,t), i) & q \in Q \text{ and } t \in \Sigma \\ (q_0, 1) & q \in F, i = 0, t = \# \\ (q_x, i) & \text{otherwise.} \end{cases}$$

and where $(q_j, i) \in F'$ iff $q_j \in F$ and $i = 1$

Describe the language accepted by $M'$ in terms of the language accepted by $M$.

(b) Show how to design a DFA $N'$ over $\Sigma \cup \{\#\}$ that accepts the language

$$L' = \left\{ x\#y\#z \;\middle|\; x \in \overline{L(M)}\ ,\ y \in L(N) \text{ and } z \in a^* \right\}$$

Define your DFA formally using notation similar to the definition of $M'$ in part (a).

4. Multiple destinations
[**Category**: Proof, **Points**: 7+3]

Let $L = aa^* + bb^*$.

(a) Prove that *any DFA* accepting $L$ must have more than one final state.

(b) Show that $L$ is acceptable by an NFA with only one final state.

5. Equality and more.
[**Category**: Construction, **Points**: 10]

Let $\Sigma = \{\mathtt{0}, \mathtt{1}, \$\}$. For any $n \in \mathbb{N}$, let the language $L_n$ be:

$$L_n = \left\{ w_1\$w_2 \;\middle|\; w_1, w_2 \in \{\mathtt{0}, \mathtt{1}\}^*,\ |w_1| = |w_2| = n,\ \text{and } w_1 = w_2 \right\}.$$

(a) Exhibit a DFA for $L_3$.

(b) For any fixed $k$, specify the DFA accepting $L_k$.

(c) Let $L$ be the language

$$L = \left\{ w_1\$w_2 \;\middle|\; w_1, w_2 \in \{\mathtt{0}, \mathtt{1}\}^*,\ \text{and } w_1 = w_2 \right\}.$$

Argue,as precisely as you can (a proof would be best), that $L$ is not regular.

(d) We can express the language $L$ as $\cup_{k=1}^{\infty} L_k$. It is tempting to reason as follows: The language $L$ is the union of regular languages, and hence is regular.

What is the flaw in this argument, and why is it *wrong* in our case?

6. How big?
[**Category**: Proof, **Points**: 10]

**(Extra credit.)**

Let $\Sigma = \{\mathtt{0}, \mathtt{1}, \$\}$, and consider the language

$$L_n = \left\{ w_1 \$ w_2 \;\middle|\; w_1, w_2 \in \{\mathtt{0}, \mathtt{1}\}^*,\ |w_1| = |w_2| = n,\ \text{and } w_1 = w_2 \right\}.$$

(i) Prove that any DFA accepting $L_n$ must have at least $2(2^{n+1} - 1)$ states.

**Solution:**

For a string $w \in \Sigma^*$ that is a prefix of a word in $L_n$, let $C(w)$ be a string in $\Sigma^*$ that completes $w$ to be a word in $L_n$. That is $wC(w) \in L_n$.

Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA accepting $L_n$, and consider the set $B_n$ of all strings over $\{\mathtt{0}, \mathtt{1}\}$ of length at most $n$. Observe that

$$|B_n| = \sum_{i=0}^{n} 2^i = 2^{n+1} - 1.$$

**Lemma 0.1** *For any two different strings $x, y \in B_n$, we have that $\delta(q_0, x) \neq \delta(q_0, y)$ (i.e., $M$ is in different states after reading $x$ or reading $y$).*

*Proof:* If this was false and there were two strings $x, y \in B_n$, $x \neq y$ such that $\delta(q_0, x) = \delta(q_0, y)$, then $x$ can be complete into a string in $L(n)$, and this implies that $\delta(q_0, xC(x)) \in F$. As such, we have that

$$\delta(q_0, yC(x)) = \delta(\delta(q_0, y), C(x)) = \delta(\delta(q_0, x), C(x)) = \delta(q_0, xC(x)) \in F.$$

Namely, $yC(x) \in L(M)$. But that is a contradiction, since $yC(x)$ is not in $L_n$, since $y \neq x$. ■

**Lemma 0.2** *Let $x$ and $y$ be two distinct strings, such that $|x| > n$ and $|y| > n$. Furthermore, assume that $x$ and $y$ can be completed into strings in $L_n$ (that is, there exists $C(x)$ and $C(y)$ such that $xC(x) \in L_n$ and $yC(y) \in L_n$). Then, if $C(x) \neq C(y)$ then $\delta(q_0, x) \neq \delta(q_0, y)$.*

*Proof:* Reasoning as above, if $\delta(q_0, x) = \delta(q_0, y)$ then $M$ would accept the strings $xC(y)$ and $yC(x)$ which are both illegal. A contradiction. ■

Clearly $M$ after reading a \$ must be in a state unreachable from any input that does not contain a \$. As such, $M$ by Lemma 0.1 must have $2^{n+1} - 1$ distinct states that are for inputs before seeing the \$ character. Now, by Lemma 0.2, after seeing the \$ character in the input, the number of different states, is the number of possible suffixes to such strings. But clearly, for every string in $w \in B_n$ there is a valid suffix to some input $x \in \{\mathtt{0}, \mathtt{1}\}^n \, \$ \, \{\mathtt{0}, \mathtt{1}\}^*$ such that $xw \in L_n$. Namely, there are $|B_n|$ different states in $M$ after seeing the \$ in the input. We conclude, that the number of states of $M$ is at least $2|B_n|$. In fact, one needs also a reject state, so the right lower bound is $2|B_n| + 1$, and it is not hard to build a DFA with exactly this number of states for $L_n$.

(ii) Prove that any NFA accepting $L_n$ must have at least $2(2^{n+1} - 1)$ states.

**Solution:**

The proof is similar, but somewhat trickier than above. It is usually harder to argue about the size of NFAs, but in this case it is doable.

So, let $N = (Q, \Sigma, \delta, q_0, F)$ be a NFA accepting $L_n$. For an input string $w \in \Sigma^*$, the NFA $N$ might be in several states after reading $w$; that is, all the states in the set $\delta(q_0, w)$. We are interested in all the states in this set that are "useful", that is, we can travel from them to reach an accepting state with the string $C(w)$. Formally, we define

$$\Delta(w) = \left\{ q \in \delta(q_0, w) \;\middle|\; \delta(q, C(w)) \cap F \neq \emptyset \right\}.$$

The key observation is that if $\Delta(w) \cap \Delta(v) \neq \emptyset$, for two distinct strings $w, v$, then $wC(v) \in L(N)$ and $vC(w) \in L_n$.

Using this argument in Lemma 0.1, yield the following lemma.

**Lemma 0.3** *For any two different strings $x, y \in B_n$, we have that $\Delta(x) \cap \delta(y) = \emptyset$, $|\Delta(x)| > 0$, and $|\Delta(y)| > 0$.*

And Lemma 0.4 becomes:

**Lemma 0.4** *Let $x$ and $y$ be two distinct strings, such that $|x| > n$ and $|y| > n$. Furthermore, assume that $x$ and $y$ can be completed into strings in $L_n$ (that is, there exists $C(x)$ and $C(y)$ such that $xC(x) \in L_n$ and $yC(y) \in L_n$). Then, if $C(x) \neq C(y)$ then $\Delta(x) \cap \delta(y) = \emptyset$.*

*(Observe, that in this case we must have $|\Delta(x)| > 0$, and $|\Delta(y)| > 0$.)*

Now, we can argue as above that since all these subsets of sets are disjoint, and each is of size at least one, it follows that $N$ must have at least $2\,|B_n|$ states.