Homework 3 Morally Due Feb 25 at 11:00 AM
THIS HOMEWORK IS TWO PAGES LONG!!!!!!

1. (0 points, but if you actually miss the midterm without telling Dr. Gasarch ahead of time, you will lose 100 points on this homework) When will the midterm be (give date and time)? When will the final be (give date and time)? By when do you have to tell Dr. Gasarch that you cannot make the midterm?

2. (25 points) The alphabet is $\{a, b\}$. A *JOSH-NFA* is an NFA such that the underlying directed graph has no cycles at all. A language is JOSH-regular if it is accepted by a JOSH-NFA. (NFA acceptance is as usual: a string is accepted by a JOSH-NFA if there is a path that leads to acceptance.)

   (a) (10 points) Give an example of a language that is regular but not JOSH-regular. Prove your result.

   (b) (15 points) Fill in the following sentence and prove it (both directions):

$$L \text{ is JOSH-regular if and only if } L \text{ is } XXX.$$

SOLUTION

(a) Let $L = a^*$. Assume by way of contradiction that a JOSH-NFA M accepts a^*. Let n be the number of states in M. Consider running M on a^{n+1}. Since a^{n+1} is in L, there will be a path through M that processes the entire string and ends up in a final state. By the Pigeonhole Principle, there must be some state s that is visited more than once since M is n states and $n+1$ transitions, but this means that there is some path of transitions from s to itself. This forms a cycle, which is a contradiction. So a^* must NOT be JOSH-regular.

(b) Fill in the following statement and prove it (both directions):

$$L \text{ is JOSH-regular if and only if } L \text{ is } XXX.$$

XXX is finite.

If L is JOSH-regular, then L is finite: let M be the NFA that recognizes L. Assume M has n states. If M recognizes any string w of length $\geq n+1$, then when M is run on w, there will be a path that visits some state more than once, so there will be a cycle in the underlying directed graph. Thus, any accepted string w must be of length $\leq n$. Hence the number of strings M recognizes is finite.

If L is finite, then L is JOSH-regular: let $L = \{w_1, w_2, ..., w_\ell\}$. There is, for each $1 \leq i \leq \ell$, a JOSH-NFA $(Q_i, \Sigma, \Delta_i, s_i, F_i)$ that recognizes $\{w_i\}$. We create a new JOSH-NFA $(Q, \Sigma, \Delta, s, F)$ that recognizes L where

- $Q = \bigcup_i Q_i \cup \{s\}$
- $F = \bigcup_i F_i$
- $\Delta(q, \sigma) = \begin{cases} \Delta_i(q, \sigma) \text{ if } q \in Q_i \text{ and } \sigma \in \Sigma \\ \{s_i : 1 \leq i \leq \ell\} \text{ if } q = s \text{ and } \sigma = \varepsilon \end{cases}$

Thus L is JOSH-regular.

END OF SOLUTION

3. (25 points) Let

   - L_1 be accepted by DFA $M_1 = (Q_1, \Sigma, \delta_1, s_1, F_1)$, and
   - L_2 be accepted by DFA $M_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$.

   Formalize the construction of an NFA for $L_1 \cup L_2$ that Dr. Gasarch did in class. (The construction using ε-transitions). That is, define the NFA $(Q, \Sigma, \Delta, s, F)$ that recognizes $L_1 \cup L_2$ in terms of M_1 and M_2.

   SOLUTION

   $Q = Q_1 \cup Q_2 \cup \{s\}$

   Σ is the same.

   Δ is defined as follows.

$$\Delta(q, \sigma) = \begin{cases} \delta_1(q, \sigma) \text{ if } q \in Q_1 \\ \delta_2(q, \sigma) \text{ if } q \in Q_2 \\ \{s_1, s_2\} \text{ if } q = s \text{ and } \sigma = \varepsilon \end{cases}$$

   $F = F_1 \cup F_2$

   END OF SOLUTION

GOTO NEXT PAGE FOR MORE HOMEWORK

4. (25 points)

   (a) (25 points) Give an algorithm that does the following.
   Input: α a regular expression
   Output: β a regular expression for the complement of the language represented by α
   You may use any procedure described in class (e.g. procedures for intersection, union, complementation, powerset construction, equivalences between NFAs and DFAs, conversion from NFA to regular expression).

   (b) (0 points) Think about: if α is of length n, how long is β?

SOLUTION
1. Input α ($|\alpha|$ is of length n)
2. Convert to NFA N (N has $O(n)$ states)
3. Then convert to DFA M using powerset construction (M has $O(2^n)$ states)
4. Swap the final and nonfinal states of M to get M'. 5. Convert M' to a regular expression β via $R(i,j,k)$ method ($|\beta|$ is $O(2^{2^n})$)
END OF SOLUTION

5. (25 points) Let $L = \{w : \#_a(w) \equiv 0 \pmod 2\}$ and the alphabet is $\{a, b\}$.

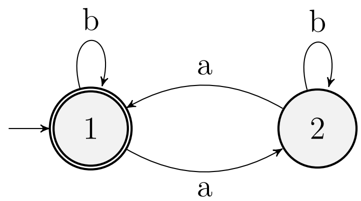
   (a) (9 points) Draw the DFA that accepts this language.

   (b) (8 points) Use the $R(i, j, k)$ method to find the regex for this language. YOU MUST USE THIS METHOD EXACTLY AS GIVEN.

   (c) (8 points) Determine the regex for L WITHOUT using the $R(i, j, k)$ method. It must be a SIMPLE regex. If we cannot understand your regex, YOU WILL GET NO CREDIT.

SOLUTION

(a) The following is the DFA.



(b) We begin with

$$\bigcup_{f\in F} R(1,f,n) = R(1,1,2) = R(1,1,1)\cup R(1,2,1)R(2,2,1)^*R(2,1,1)$$

so we ONLY want $R(1,1,2)$. We have

$$R(1,1,0) = \{\varepsilon\} \cup \{b\} = \{\varepsilon, b\}$$

$$R(1,2,0) = \{a\}$$

$$R(2,1,0) = \{a\}$$

$$R(2,2,0) = \{\varepsilon\} \cup \{b\} = \{\varepsilon, b\}$$

$R(1,1,1) = R(1,1,0)\cup R(1,1,0)R(1,1,0)^*R(1,1,0) = \{\varepsilon,b\}\cup\{\varepsilon,b\}\{\varepsilon,b\}^*\{\varepsilon,b\} = b^*$

$R(1,2,1) = R(1,2,0)\cup R(1,1,0)R(1,1,0)^*R(1,2,0) = \{a\}\cup\{\varepsilon,b\}\{\varepsilon,b\}^*a = b^*a$

$R(2,2,1) = R(2,2,0)\cup R(2,1,0)R(1,1,0)^*R(1,2,0) = \{\varepsilon,b\}\cup a\{\varepsilon,b\}^*a = \{\varepsilon,b\}\cup ab^*a$

$R(2,1,1) = R(2,1,0)\cup R(2,1,0)R(1,1,0)^*R(1,1,0) = \{a\}\cup a\{\varepsilon,b\}^*\{\varepsilon,b\} = ab^*$

All we need is $R(1,1,2)$. So our final regex is

$$b^* \cup b^*a(\{\varepsilon,b\} \cup ab^*a)^*ab^*$$

(c) The nicer regex is $(ab^*a \cup b)^*$.

END OF SOLUTION