CS 373, Spring 2009. Midterm 1

INSTRUCTIONS (read carefully)

- Fill in your name, netid, and discussion section time below. Also write your netid on the other pages (in case they get separated).

NAME:

NETID: | **DISC:**

- There are 7 problems. Make sure you have a complete exam.
- The point value of each problem is indicated next to the problem, and in the table below.
- Points may be deducted for solutions which are correct but excessively complicated, hard to understand, or poorly explained. Please keep your solutions ***short*** and crisp.
- The exam is designed for one hour, but you have the full two hours to finish it.
- It is wise to skim all problems and point values first, to best plan your time.
- This is a closed book exam. No notes of any kind are allowed. Do all work in the space provided, using the backs of sheets if necessary. See the proctor if you need more paper.
- Please bring any apparent bugs to the attention of the proctors.
- After the midterm is over, discuss its contents with other CS 373 students **only** after verifying that they have also taken the exam (e.g. they aren't about to take the conflict exam).
- We indicate next to each problem how much time we suggest you spend on it. We also suggest you spend the last 25 minutes of the exam reviewing your answers.

Problem	Possible	Score
1	8	
2	6	
3	6	
4	6	
5	8	
6	8	
7	8	
Total	50	

Problem 1: Short Answers (8 points)

[10 minutes]

The answers to these problems should be short and not complicated.

(A) If a DFA M has k states then M must accept some word of length at most $k-1$. True or false?

☐ True ☐ False

And why? (At most 20 words.)

(B) Let L be a finite language. Is the complement language $\overline{L}$ regular?

☐ Yes ☐ No

And why? (At most 30 words.)

(C) Suppose that L is a regular language over the alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}\}$. And consider the language

$$L' = \left\{ 0^i \;\middle|\; i = |w|, w \in L \right\}.$$

Is the language L' regular?

☐ Yes ☐ No

And why? (At most 40 words.)

(D) $\mathtt{a}^*\emptyset = \mathtt{a}^*$. True or False?

☐ True ☐ False

(E) Let $L = \left\{ wx \;\middle|\; w \in \Sigma^*, x \in \Sigma^*, |w| = |x| \right\}$. Is L regular?

☐ Yes ☐ No

(F) Let L_i be a regular language, for $i = 1, \ldots, \infty$. Is the language $\bigcap_{i=1}^{\infty} L_i$ always regular? True or false?

☐ True ☐ False

(G) If L_1 and L_2 are two regular languages that are accepted by two DFAs D_1 and D_2, respectively, each with k_1 and k_2 states, then the language $L_1 \setminus L_2$ can always be recognized by a DFA with $k_1 * k_2$ states? True or false?

☐ True ☐ False

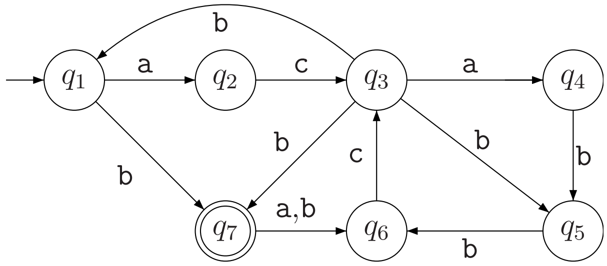
(H) The minimum size NFA for a regular language L, always has strictly fewer states then the minimum size DFA for the language L. True or False?

☐ True ☐ False

Problem 2: NFA transitions (6 points)

[5 minutes]

Suppose that the NFA $N = (Q, \{\texttt{a}, \texttt{b}\}, \delta, q_0, \mathcal{F})$ is defined by the following state diagram:



Fill in the following values:

(A) $\mathcal{F} =$

(B) $\delta(q_2, \texttt{a}) =$

(C) $\delta(q_3, \texttt{b}) =$

(D) $\delta(q_6, \texttt{c}) =$

(E) List the members of the set $\left\{ q \in Q \;\middle|\; q_5 \in \delta(q, \texttt{b}) \right\}$: ☐

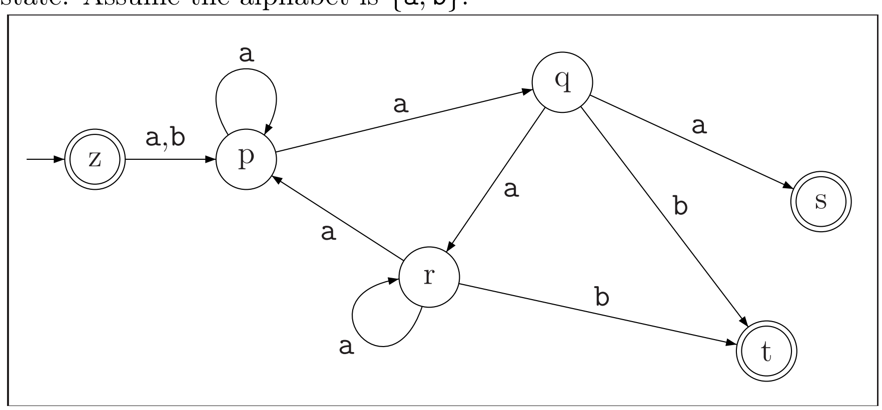
(F) Does the NFA accepts the word `acbacbb`? (Yes / No)

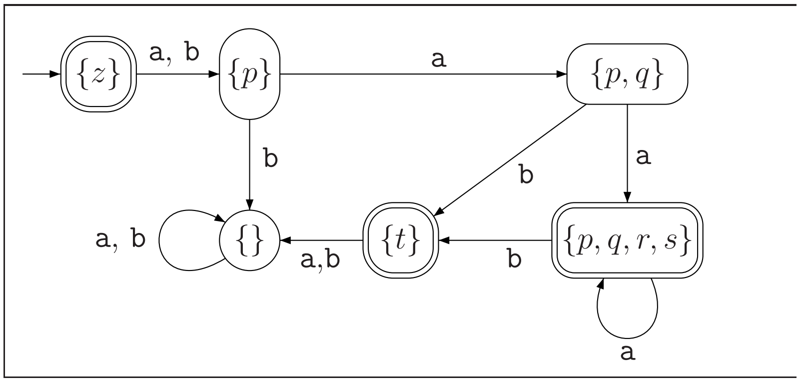
☐ Yes ☐ No

Problem 3: NFA to DFA (6 points)

[10 minutes]

Convert the following NFA to a DFA recognizing the same language, using the subset construction. Give a state diagram showing all states reachable from the start state, with an informative name on each state. Assume the alphabet is $\{\texttt{a}, \texttt{b}\}$.





Problem 4: Modifying DFAs (6 points)

[15 minutes]

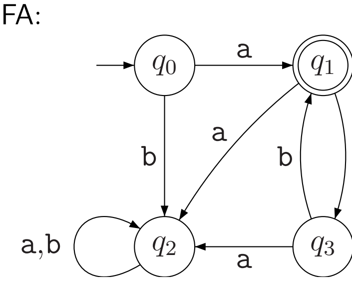
Suppose that $M = (Q, \Sigma, \delta, q_0, F)$ is a DFA, where $\Sigma = \{\mathtt{a}, \mathtt{b}\}$. Using M, we define a new DFA

$$M' = \big(Q \cup \{r, s\}\,,\, \Sigma \cup \{\#\}\,,\, \delta',\, q_0,\, F'\big)\,,$$

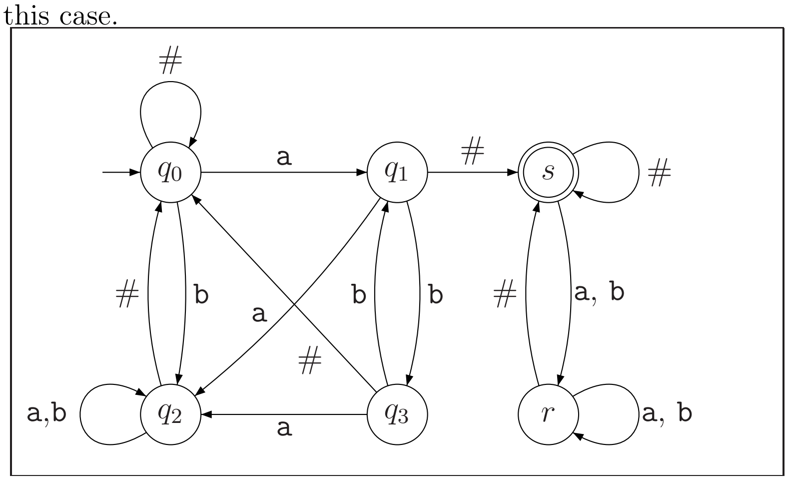
where $F' = \{s\}$ and the new transition function δ' is defined as follows

$$\delta'(q,t) = \begin{cases} \delta(q,t) & q \in Q \text{ and } t \in \Sigma \\ s & q \in F, t = \# \\ q_0 & q \in Q \setminus F, t = \# \\ r & q \in \{r,s\}\,, t \in \{\mathtt{a},\mathtt{b}\} \\ s & q \in \{r,s\}\,, t = \# \end{cases}$$

(A) Assume M is the following DFA:



Draw M' for this case.



(B) In general, define the language of M' in terms of the language of an arbitrary M. (Hint: Make sure that your answer works for the above example!)

Problem 5: NFA construction (8 points)

[15 minutes]

A string y is a ***subsequence*** of string $x = x_1 x_2 \cdots x_n \in \Sigma^*$, if there exists indices $i_1 < i_2 < \cdots < i_m$ such that $y = x_{i_1} x_{i_2} \cdots x_{i_m}$. Note, that the empty word ϵ is a subsequence of every string. For example, aaba is a subsequence of c<u>a</u>dcd<u>a</u>c<u>ba</u>, but abc is not a subsequence of cbacba.

Let $N = (Q, \Sigma, \delta, q_0, F)$ be an NFA with language L. Describe a construction for an NFA N' such that:

$$L(N') = \left\{ x \;\middle|\; \text{there is a } y \in L, \text{ such that } x \text{ is a subsequence of } y \right\}.$$

(A) Explain the idea of your construction.

(B) Write down your construction formally (in tuple notation, as we did for example in constructing the product of two DFA's in lecture). Namely, you are given the NFA $N = (Q, \Sigma, \delta, q_0, F)$, and you need to specify the new NFA N'. (Use formal notations and few words.)

Problem 6: Proof (8 points)

[20 minutes]

Consider ***effective regular expressions***. These are regular expressions that are not allowed to use the empty-set notation. Formally, they are defined as follows:

regex	conditions	set represented
a	$\mathtt{a} \in \Sigma$	$\{\mathtt{a}\}$
ϵ		$\{\epsilon\}$
$R + S$	R, S regexps	$L(R) \cup L(S)$
RS	R, S regexps	$L(R)L(S)$
R^*	R a regex	$L(R)^*$

The ***complexity*** $\Delta(R)$ of a regular expression R is the number of operators appearing in R. Thus, $\Delta(\mathtt{ab}) = \Delta(\mathtt{a} \circ \mathtt{b}) = 1$ (we concatenate a with b), $\Delta(\ (\mathtt{a}+\mathtt{b})^*) = 2$, and $\Delta\Big(((\mathtt{a}+\mathtt{b})^*(\mathtt{b}+\mathtt{c}+\epsilon))^* \ \Big) = 6$ (note, that parenthesis are not counted).

Prove formally the following claim.

Claim. *If R is an effective regular expression, then $L(R)$ contains at least one word of length at most $\Delta(R) + 1$.*

Proof:

Problem 7: NFA modification with a proof (8 points)

[20 minutes]

Fix a finite alphabet Σ. We want to transform an NFA A over Σ into an NFA B over Σ such that B accepts all suffixes of words accepted by A. The suffix language of $L(A)$ is

$$S = \left\{ y \in \Sigma^* \;\middle|\; \exists x \in \Sigma^*, xy \in L(A) \right\}$$

We want to build a NFA B such that $L(B) = S$.

Let $A = (Q, \Sigma, \delta, q_0, F)$ be an NFA where all states in Q are *reachable* from the initial state (i.e. for any state $q \in Q$, there is a set of transitions that connect the initial state q_0 to q). More formally, for any $q \in Q$, there is a word $w \in \Sigma^*$ such that $q \in \delta(q_0, w)$ (where δ is the transition function extended to words).

Let us define $B = (Q \cup \{q_0'\}, \Sigma, \delta', q_0', F)$ where we define

$$\begin{aligned}
\delta'(q, a) = &\quad \delta(q, a) \quad \textit{for all } q \in Q, a \in \Sigma \cup \{\epsilon\} \\
\delta'(q_0', \epsilon) = &\quad Q \\
\delta'(q_0', a) = &\quad \emptyset \quad \textit{for all } a \in \Sigma
\end{aligned}$$

Prove *formally* that $L(B) = S$.

Hint: The proof goes by showing that $L(B)$ is contained in S, and that $L(B)$ is contained in S. So, you must show both inclusions using precise arguments. You do not need to use complete mathematical notation, but your proof should be precise, correct, short and convincing. (Make sure you are not writing unnecessary text [like copying the hint, or writing text unrelated to the proof].)