COMP 3721: Theory of Computation
Spring 2012

1. This is an open-book, open-notes, open-brain exam. But you are not allowed you use any device with a wireless connection.
2. Time limit: 120 minutes.
3. When asked to construct an automaton, you can use either a state diagram (recommended) or the formal mathematical definition.
4. You can write on the back of the paper if you run out of space. Please let us know if you need more scratch paper.
5. Relax and breathe, it's just a midterm.

Student name: ________________ **Student ID:** ________________

1. (20 pts) Fix the alphabet $\Sigma = \{a, b\}$.

   (a) (5 pts) Construct a DFA that accepts the language $\{w \in \Sigma^* \mid w \text{ has } aba \text{ as a substring}\}$.

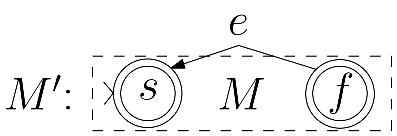
   (b) (5 pts) Construct a DFA that accepts the language $\{w \in \Sigma^* \mid w \text{ does not have } aba \text{ as a substring}\}$.

(c) (5 pts) Construct an NFA that accepts the language $\{w \in \Sigma^* \mid w \text{ ends with } aba\}$.

(d) (5 pts) Convert the NFA you constructed in (c) to an equivalent DFA that accepts the same language.

2. (15 pts) Suppose that we want to prove that the class of regular languages are closed under Kleene star. Let L be a regular language and let M be an NFA shown below that accepts L; we can assume that it has only one "yes" state.

We then build the following NFA M' that tries to accept L^*. As mentioned in class, this does not work.



(a) Can you give a counter example, i.e., a concrete M such that $L(M') \neq L^*$?

(b) Which one of the following is true? Justify your answer.

i. $L(M') \subset L^*$

ii. $L(M') \supset L^*$

iii. Neither of above.

3. (20 pts) Finite languages

(a) (10 pts) Show that any finite language over Σ is a regular language.

(b) (10 pts) Fix the alphabet $\Sigma = \{0, 1\}$. Let FL be the class of all finite languages over Σ. You have just proved that FL is a subset of the class of regular languages. However, the following "proof" shows that FL is uncountable. Where is the flaw in the proof?

Proof: We will use the diagonalization principle. Suppose FL is countable, so we can enumerate all members of FL as

$$\mathsf{FL} = \{L_0, L_1, L_2, \ldots\}.$$

Meanwhile, we also enumerate all strings over the alphabet Σ as

$$\Sigma^* = \{w_0, w_1, w_2, \ldots\}.$$

Now consider the diagonal language

$$D = \{w_i : w_i \notin L_i, i = 0, 1, 2, \ldots\},$$

which should be a member of FL. However, for any i, $D \neq L_i$, since either $w_i \in L_i$ and $w_i \notin D$ or $w_i \notin L_i$ and $w_i \in D$. Thus, $D \notin \mathsf{FL}$ and we have reached a contradiction. So FL is not countable. $\square$

4. (25 pts) Let $\Sigma = \{a, b\}$ and $L = \{w \in \Sigma^* \mid w \text{ has more } a's \text{ than } b's\}$.

(a) (5 pts) Give a context-free grammar that generates L.

(b) (5 pts) Construct a pushdown automaton that accepts L.

(c) (15 pts) Prove that L is not regular.

5. (20 + 10 pts) The pushdown automaton is equipped with a stack. Now we will consider the case where the stack is replaced by a queue.

   (a) (10 pts) Please complete the following mathematical definition of such a *queue automaton.*

   A (nondeterministic) *queue automaton* is a 6-tuple $M = (K, \Sigma, \Psi, \Delta, s, F)$, where K is the set of states, Σ is the alphabet of the input symbols, Ψ is the alphabet of the queue symbols, $s \in K$ is the initial state, F is the set of final states, and $\Delta \in (K \times (\Sigma \cup \{e\}) \times \Psi^*) \times (K \times \Psi^*)$ is the transition relation.

   A configuration of M is (q, w, α) where $q \in K$ is the current state, $w \in \Sigma^*$ is the unread portion of the input, and $\alpha \in \Psi^*$ is the current queue content. The machine starts with an empty queue. If the machine is currently in state q, the symbol it is reading from the input is a, and there is a transition $((p, a, \beta), (q, \gamma)) \in \Delta$, then the machine can go to state q after reading a, while popping β from the front of the queue and pushing γ into the back of the queue. We say that M accepts an input string, if starting from the initial state, M can nondeterministically reach a final state with an empty queue after having read the string completely,

   Please give the formal definition of "(p, x, α) yields (q, y, ζ) in one step", denoted $(p, x, \alpha) \vdash_M (q, y, \zeta)$:

   Please give the formal definition of "M accepts a string w":

   The language accepted by M, denoted $L(M)$, is the set of all strings accepted by M.

(b) (10 pts) Let $\Sigma = \{a, b\}$. Construct a queue automaton that accepts the language $\{a^n b^m a^n b^m : n, m \geq 0\}$.

(c) (extra credits[1]: 10 pts) Show that any context-free language can be accepted by a queue automaton. You can give the proof on an intuitive level. Combined with (b), you will have proved that queue automata are strictly stronger than pushdown automata in terms of computability.

[1] Your final score will be min{100, normal points + extra points}