University of Illinois at Urbana-Champaign
Department of Computer Science

Midterm 2

CS 373: Theory of Computation
Fall 2009

Name:	
Netid:	

- Print your name and netid, *neatly* in the space provided above; print your name at the upper right corner of *every* page. Please print legibly.
- This is a *closed book* exam. No notes, books, dictionaries, calculators, or laptops are permitted.
- You are free to cite and use any theorems from class or homeworks without having to prove them again.
- Write your answers in the space provided for the corresponding problem. Let us know if you need more paper.
- Suggestions: Read through the entire exam first before starting work. Do not spend too much time on any single problem. If you get stuck, move on to something else and come back later.
- If you run short on time, remember that partial credit will be given.
- If any question is unclear, ask us for clarification.

Question	Points	Score
Problem 1	20	
Problem 2	15	
Problem 3	10	
Problem 4	15	
Problem 5	20	
Problem 6	20	
Total	100	

1. Short Problems (20 points)

   Give answers to each of the following questions, including a short justification. For each, you will get two points for the correct answer and two points for the correct justification.

   (a) Consider a grammar $G = (V, \Sigma, R, S)$ in which $\Sigma = \{0, 1\}$ and the only rules are $S \to A$, $A \to AB$, and $B \to A$. What is $L(G)$? (4 points)
   Sol: All the rules just substitute variables with variables. Therefore, there is no way to derive a string in $\{0, 1\}^*$ from S. So $L(G) = \varnothing$.

   (b) Is $\{a^n b^m c^k d^n \in \Sigma^* \mid n, m, k \geq 0\}$ a context-free language? (4 points)
   Sol: Yes, it is the language of this grammar

$$S \to aSd \mid B \qquad B \to bB \mid Bc \mid \varepsilon$$

   (c) Let A and B be any two context-free languages. Must $A \setminus B$ be a context-free language (here, $\setminus$ denotes set difference)? (4 points)
   Sol: Note that for any set A, we have $\overline{A} = \Sigma^* - A$ and Σ^* is a CFL. Therefore if the claim is true, then CFLs would be closed under complementation. So the answer is No.

   (d) If a language A is Turing-decidable, then is its complement $\Sigma^* \setminus A$ Turing decidable? (4 points)
   Sol: Yes. A decider for $\overline{A}$ simulates a decider for A and toggles the result of it after simulation.

   (e) If we pick a language "at random", is it likely to be Turing-recognizable? To define "at random" more precisely, run through every string in Σ^*, and toss a fair coin to decide whether to include it in A. Is the probability greater or less than 1/2 that A is Turing recognizable? (4 points)
   Sol: Less than a half. Remember that the set of all languages is uncountable but the set of recognizable languages is countably infinite.

2. Turing Machine (15 points)

   Design a Turing machine that decides the following language:

$$\{ww \in \{0,1\}^* \mid w \in \{0,1\}^*\}$$

   Give an implementation-level description. Be sure to specify the tape alphabet Γ.

   Sol: We design a 2-tape TM M with $\Gamma = \{0, 1, 0', 1', B\}$, where B represents the blank symbol. First M finds the middle of the input string x: It repeatedly finds the leftmost character $c \in \{0,1\}$ on the first tape and replaces it with c' and then it finds the rightmost character $d \in \{0,1\}$ on the first tape and replaces it with d'. When all the characters are in the set $\{0', 1'\}$, then the head is on top of the middle of x. Now M copies the second half of x to the second tape and as it copies a character from the first tape to the second tape it overwrites that character with blank. Now M has the first half of x on the first tape and the second half of x on the second tape. It now starts to compare the content of the first and the second tape, if they match, then x is of the form ww and M accepts, otherwise M rejects.

3. Context Free (10 points)

Prove that

$$L = \{a^{n_1} b^{n_1} a^{n_2} b^{n_2} \cdots a^{n_k} b^{n_k} \in \{a,b\}^* \mid k, n_1, n_2, \ldots, n_k \geq 0\}$$

is a context-free language.

Sol: We show that $L(G) = L$ where G is the following grammar

$$S \to AS \mid \epsilon \qquad A \to aAb \mid \epsilon.$$

First we show that $L \subseteq L(G)$. We observe that A can derive any string of the form $a^j b^j$ since:

$$A \to aAb \to a^2Ab^2 \to \ldots \to a^j A b^j \to a^j \varepsilon b^j \to a^j b^j$$

Now consider this derivation in G:

$$S \to AS \to A^2S \to \ldots \to \overbrace{A \ldots A}^{k} S \to a^{n_1} b^{n_1} a^{n_2} b^{n_2} \cdots a^{n_k} b^{n_k}$$

where in the last step of the derivation above, the jth A will be replaced with $a^{n_j} b^{n_j}$. Therefore S derives every string of L and $L \subseteq L(G)$.

Second we show that $L(G) \subseteq L$. We use induction on the number of steps in derivations in G to prove that the outcome of every derivation of G is in L. For the base case, when the derivation has just one step, it will look like $S \to \varepsilon$ and clearly ε is in L. Now assume that every derivation of length at most k generates a string in L. Consider a derivation of length $k+1$. The first step of the derivation must be $S \to AS$. The rest of the steps replace A with a string of the form $a^j b^j$ (proved in the previous part) and replace S with some string w. So finally we have

$$S \to AS \overbrace{\to \ldots \to}^{k} a^j b^j w$$

Note that w is being derived from S in at most k steps and so is of the form $a^{n_1} b^{n_1} a^{n_2} b^{n_2} \cdots a^{n_m} b^{n_m}$. Therefore the outcome of the whole derivation is $a^j b^j a^{n_1} b^{n_1} a^{n_2} b^{n_2} \cdots a^{n_m} b^{n_m}$ which is obviously in L. So $L(G) \subseteq L$.

Sol: We know that $A = \{a^n b^n \mid n \geq 0\}$ is a context free language. By definition of Kleene star, we observe that $A^* = L$. Since we know that context free languages are closed under Kleene star, L is context free.

4. Pumping Lemma (15 points)

   Use the pumping lemma to prove that

$$L = \{a^n b^m c^n d^m \in \{a, b, c, d\}^* \mid n, m \geq 1\}$$

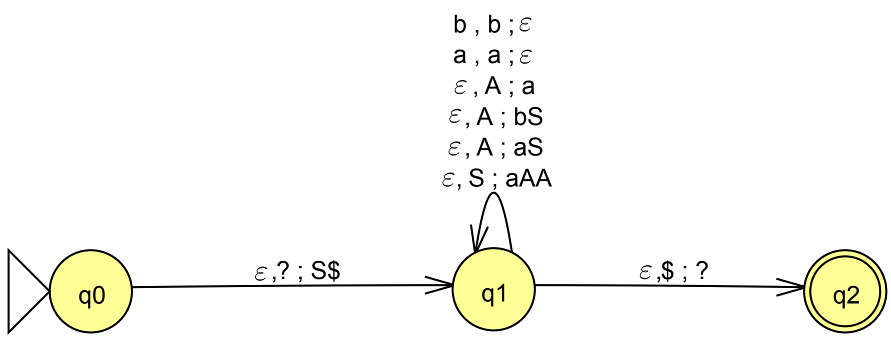
   is not a context-free language.
   Sol: Assume to the contrary that L is context free. Therefore according to pumping lemma, it has a pumping length $p > 0$ such that for any string s where $|s| > p$, we can write s as $s = xuyvz$ such that $|uyv| \leq p$ and $|uv| \neq 0$ and for any $i \geq 0$ we have $s_i = xu^i yv^i z \in L$.

   Consider the string $s = a^p b^p c^p d^p$. Since $|s| = 4p > p$, we have the described break down $s = xuyvz$. Since $|uyv| \leq p$, the substring uyv is formed of either just one kind of letters or two kind of adjacent letters. Moreover both u and v contain just one kind of letter (since otherwise in s_2 the letters will be out of order). Now if uv includes some a (some c) it can not include any c (any a) and in s_2 the number of as and cs will not be balanced. If uv includes some b (some d) it can not include any d (any b) and in s_2 the number of bs and ds will not be balanced. Therefore $|uv| = 0$ which is a contradiction.

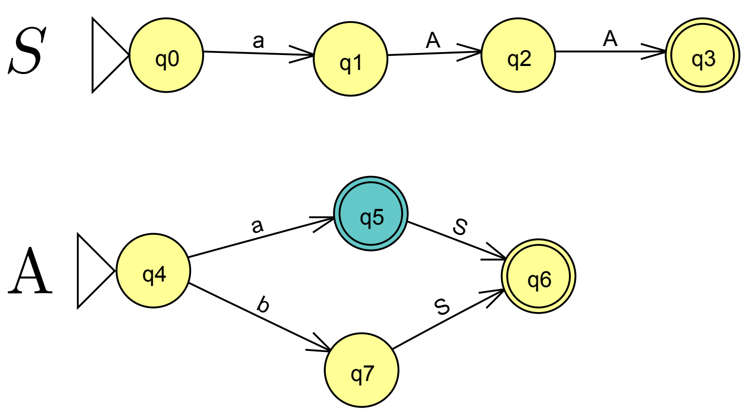
5. PDA and RA Design (20 points)

Consider the following grammar: $G = (V, \Sigma, R, S)$ in which $V = \{S, A\}$, $\Sigma = \{a, b\}$, and the rules are $S \to aAA$ and $A \to aS \mid bS \mid a$.

(a) Construct a PDA that recognizes the language described by G. It is sufficient to show the state diagram with edges appropriately labeled. (14 points)
Sol:



(b) Construct an RA (recursive automaton) that recognizes the language described by G. It is sufficient to show the state diagrams with edges appropriately labeled. (6 points)
Sol:



6. Context Free Grammars (20 points)

   Consider the following grammar: $G = (V, \Sigma, R, S)$ in which $V = \{S\}$, $\Sigma = \{a, b\}$, and the rules are $S \to aS \mid Sb \mid a \mid b$.

   (a) Use induction on the length of strings to show that no string in $L(G)$ has ba as a substring. (12 points)
   Sol: As the basecase we consider the strings of length 1. Since there is no nullable in this grammar, the only possible derivations that generate strings of length 1 are $S \to a$ and $S \to b$ and a and b do not contain ba.

   Now assume that no generated string of length at most k contains ba. Consider a generated string w of length $k + 1$. The first step of the derivation must be $S \to aS$ or $S \to Sb$. If it is $S \to aS$, then the rest of this derivation replaces the second S by some string w', so $w = aw'$. Length of w' is k (since $|w| = k + 1$) therefore by induction hypothesis w' contains no ba. The only way for $w = aw'$ to have a ba substring is that ba sits inside w'. Therefore w in this case contains no ba. The other case is very similar. This proves that no generated string contains ba as a substring.

   (b) Convert G into Chomsky normal form. (8 points)
   Sol: S is the start symbol.

$$S \to T_a S_0 \mid S_0 T_b \mid a \mid b$$

$$S_0 \to T_a S_0 \mid S_0 T_b \mid a \mid b$$

$$T_a \to a \qquad T_b \to b$$