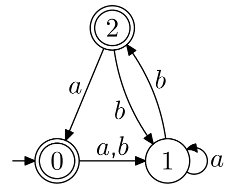
CS 4384	**Automata Theory**	Sample
Fall 2012		Final

Do not open the exam until instructed.

This exam is closed book and closed notes, except that you may bring two sheets of paper with anything written on them front and back. In your answers, any results proved in class or on homeworks may be applied without re-proving or re-deriving them.

Write your answers on the loose sheets provided, and be sure to write your name atop ALL sheets that you turn in. Turn in this exam paper with your solutions. Scratch work will be considered for partial credit so is worth including in your submitted answers, but *clearly mark the solutions you want me to grade as your final answers.* Otherwise you will receive the lowest score of your various attempts.

(1) **(6 pts)** Convert the following NFA to a regular CFG:



(2) Consider the following CFG:

$$
\begin{aligned}
S &\to aB \mid bA \mid D \mid E \\
A &\to a \mid aS \mid bAA \mid c \\
B &\to b \mid bS \mid aBB \mid c \\
D &\to dD \mid Dd \mid d \\
E &\to EE \mid EdE
\end{aligned}
$$

- **(a)** **(4 pts)** Give a left-most derivation of the string $aabbcc$.
- **(b)** **(2 pts)** Give a right-most derivation of the same string.
- **(c)** **(4 pts)** Is the grammar ambiguous? Why or why not?

(3) **(12 pts)** *Mathematically* define a DFA that accepts the language of all strings over alphabet $\{a, b\}$ in which each b is separated from the next by at least 100 a's.

(4) **(10 pts)** Write a CFG that generates the language of ALL palindromes over alphabet $\{a, b\}$ that do not contain the substring aa.

(5) **(7 pts)** For all strings $s, s' \in \Sigma^*$, we write $s \rightsquigarrow s'$ if there is a way to remove zero or more (possibly non-adjacent) symbols from s to get s'. For example, $final \rightsquigarrow fnl$ and $final \rightsquigarrow fin$, but $final \not\rightsquigarrow fan$. Define $p(L) = \{s' \mid s \in L,\ s \rightsquigarrow s'\}$. Complete the following proof that p is a closure property for the context-free languages:

> *Proof.* Let a CFL L be given. Since L is a CFL, there exists a PDA $A = (Q, \Sigma, \Gamma, \Delta, q_0, F)$ that generates L. Define a new PDA $A' = (Q', \Sigma, \Gamma, \Delta', q_0', F')$ where
>
> $$
> \begin{aligned}
> Q' &= \textit{(you write this part)} \\
> \Delta' &= \textit{(you write this part)} \\
> q_0' &= \textit{(you write this part)} \\
> F' &= \textit{(you write this part)}
> \end{aligned}
> $$
>
> PDA A' accepts language $p(L)$, so $p(L)$ is context-free. We conclude that p is a closure property of the CFL's. □

(6) **(10 pts)** Prove that the language $L_1 = \{a^i b^j c^k \mid j<i,\ j<k\}$ is NOT context-free.

(7) **(9 pts)** Prove that the following decision problem is NOT Turing-decidable by reducing the universal language (L_U) to it: *Given two TM's A_1 and A_2, does $|L(A_1) \cap L(A_2)| = 2$? That is, are there exactly 2 strings that are accepted by both A_1 and A_2?*

Solutions

(1) $S \to aA \mid bA \mid \varepsilon$
$A \to aA \mid bB$
$B \to aS \mid bA \mid \varepsilon$

(2) **(a)** $S \Rightarrow aB \Rightarrow aaBB \Rightarrow aabSB \Rightarrow aabbAB \Rightarrow aabbcB \Rightarrow aabbcc$

(b) $S \Rightarrow aB \Rightarrow aaBB \Rightarrow aaBc \Rightarrow aabSc \Rightarrow aabbAc \Rightarrow aabbcc$

(c) Yes, the grammar is ambiguous because there are two different leftmost derivations of the string dd: (1) $S \Rightarrow D \Rightarrow dD \Rightarrow dd$ and (2) $S \Rightarrow D \Rightarrow Dd \Rightarrow dd$.

(3) Define $A = (Q, \{a, b\}, \delta, q_0, F)$ with $Q = [0, 100] \cap \mathbb{Z}$, $q_0 = 100$, $F = Q$, and

$$\delta = \{((100, a), 100),\ ((100, b), 0)\} \cup \{((q, a), q + 1) \mid q \in Q - \{100\}\}$$

(4) $S \to abSba \mid bSb \mid aba \mid a \mid b \mid \varepsilon$

(5) Define $Q' = Q$, $q_0' = q_0$, $F' = F$, and

$$\Delta' = \Delta \cup \{((q, \varepsilon, \gamma), (q', \gamma')) \mid ((q, \sigma, \gamma), (q', \gamma')) \in \Delta\}$$

(6) *Proof.* Expecting a contradiction, assume L_1 is a CFL and let $p \in \mathbb{N}_1$ be its pumping length. Define $s = a^{p+1}b^pc^{p+1}$. Let $uvxyz = s$ be a partitioning such that $|vxy| \leq p$ and $|vy| \geq 1$. Since $|vxy| \leq p$, vy cannot have a's, b's, and c's. Since $|vy| \geq 1$, it follows that vy does not have equal numbers of a's, b's, and c's. With this ruled out, there are only two other possibilities:

Case 1: If vy has more a's or more c's than b's, then choose $s' = uxz$ and observe that s' has at most as many a's or at most as many c's as b's (because more a's or more c's than b's were removed).

Case 2: If vy has more b's than a's or than c's, then choose $s' = uv^2xy^2z$ and observe that s' has at least as many b's as a's or c's (because more b's were added than a's or c's).

In both cases $s' \notin L_1$, contradicting the pumping lemma. We conclude that L_1 is not context-free. □

(7) $L_D = \{\langle A_1\rangle\#\langle A_2\rangle \mid |L(A_1) \cap L(A_2)| = 2\}$ is undecidable.

Proof. Expecting a contradiction, assume there exists an always-halting TM D such that $L(D) = L_D$. To decide whether $\langle A\rangle\#\langle s\rangle \in L_U$, construct a new TM A' that first compares its input to s. If they match, A' simulates $A(s)$. Otherwise, A' next compares its input to $s\sigma$ where σ is some arbitrarily chosen symbol from A's alphabet. If they match, A' immediately accepts; otherwise it immediately rejects. We then simulate $D(\langle A'\rangle\#\langle A'\rangle)$, accepting if it accepts and rejecting if it rejects.

Observe that (1) A' always accepts $s\sigma$, (2) it accepts s if and only if A accepts s, and (3) it always rejects everything else. Thus, $|L(A') \cap L(A')| = |L(A')| = 2$ if and only if $A(s)$ accepts (and it equals 1 otherwise). Hence, $D(\langle A'\rangle\#\langle A'\rangle)$ accepts if and only if $A(s)$ accepts.

This contradicts the undecidability of L_U, so we conclude that no such TM D actually exists; i.e., L_D is undecidable. □