CSC 320 Midterm 2 Practice Questions

1) Select which of the following languages are context-free:

(a) $L = \{w \in \{0,1\}^* \mid w \text{ has exactly twice as many a's as b's}\}$
(b) $L = \{aaa, bbab, aaabbb, aa, c, ca\}$
(c) $L(R)^*$ where R is a regular expression
(d) $L = \{a^n b^{2n} c^{2n} \mid n \geq 0\}$

2) Select every true statement:

(a) If a language is context-free, then it is non-regular.
(b) If a language is context-free, then it is regular.
(c) If a language is regular, then it is context free.
(d) If a language is non-regular, then it is not context-free.
(e) If a language is not context free, then it is non-regular.

3) Let T_R denote the class of Turing-recognizable languages.

(a) For any language $L \in T_R$, there exists a nondeterministic Turing machine M with $L(M) = L$.
(b) For any language $L \in T_R$ there exists a nondeterministic finite automaton N with $L(N) = L$.
(c) Let R be a regular expression. Then $L(R) \in T_R$.
(d) Let P be a pushdown automaton. Then $L(P) \in T_R$.
(e) $\emptyset \in T_R$.

4) Consider the following CFG $G = (\{S, A, B\}, \{a, b\}, R, S)$ where the rules in R are given as follows.

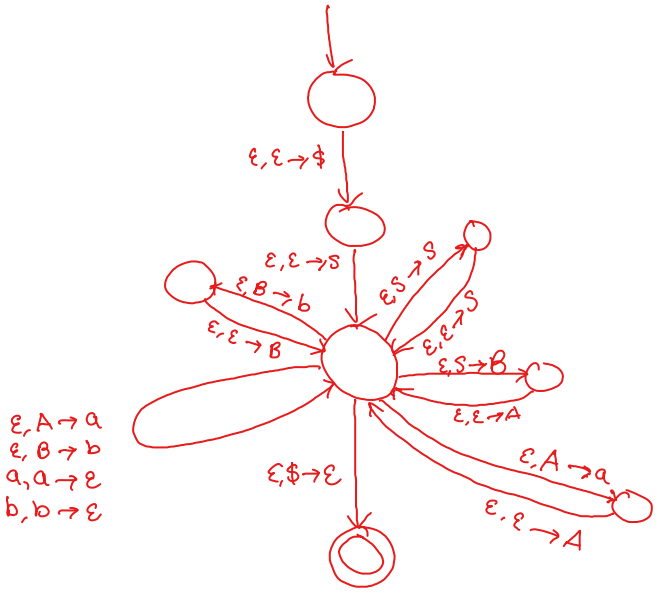
$$S \to SS \mid AB$$
$$A \to Aa \mid a$$
$$B \to Bb \mid b$$

(a) Show that G is ambiguous by giving two leftmost derivations of a string in $L(G)$.

1) $S \Rightarrow \underline{S}S \Rightarrow \underline{S}SS \Rightarrow \underline{A}BSS \Rightarrow a\underline{B}SS \Rightarrow ab\underline{S}S \Rightarrow ab\underline{A}BS \Rightarrow aba\underline{B}S \Rightarrow abab\underline{S}$
$\Rightarrow ababAB \Rightarrow ababa\underline{B} \Rightarrow ababab$

2) $S \Rightarrow \underline{S}S \Rightarrow ABS \Rightarrow a\underline{B}S\ ab\underline{S} \Rightarrow ab\underline{S}S \Rightarrow ab\underline{A}BS \Rightarrow aba\underline{B}S \Rightarrow abab\underline{S}$
$\Rightarrow ababAB \Rightarrow ababa\underline{B} \Rightarrow ababab$

(b) Convert G to an equivalent PDA following the steps of the CFG to PDA conversion.



(c) Convert G into Chomsky Normal Form. Show all your steps.

Step 1: $S_0 \to S$
$S \to SS \mid AB$
$A \to Aa \mid a$
$B \to Bb \mid b$

Step 2: (No ε-rules)

Step 3: $S_0 \to SS \mid AB$
$S \to SS \mid AB$
$A \to Aa \mid a$
$B \to Bb \mid b$

Step 4: $S_0 \to SS \mid AB$
$S \to SS \mid AB$
$A \to AX \mid a$
$B \to BY \mid b$
$X \to a$
$Y \to b$

5) Consider the language $L = \{0^i 1^j 2^k \mid i, j, k \geq 0 \text{ and } i + k = j\}$.

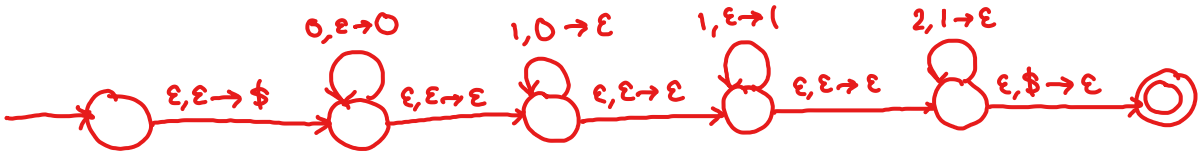
(a) Give a context free grammar G with $L(G) = L$.

$S \to AB$

$A \to 0A1 \mid \varepsilon$

$B \to 1B2 \mid \varepsilon$

(b) Give a state diagram for a PDA which recognizes L (without using the CFG to PDA conversion).



6) Prove that the language $L = \{0^n \mid n > 0,\ n \text{ is a prime number}\}$ is not context free using the pumping lemma for context free languages.

Assume for a contradiction that L is context free.

Let p be the pumping length given by the PL and let p' be the smallest prime number $\geq p$.

Choose $s = 0^{p'}$. $s \in L$ and $|s| \geq p$, so should be able to write $s = uvxyz$ such that all 3 properties of the PL are satisfied.

By property 1 of the PL, $|vy| > 0$. Let $m = |vy|$.

By property 3 of the PL, $uv^ixy^iz \in L$ for each i. uv^ixy^iz has form $0^{p' + (i-1)m}$

However, not all numbers $p' + (i-1)m$ are prime. Consider $i = p'+1$. Then uv^ixy^iz is the string $0^{p' + p'k} = 0^{p'(k+1)}$. The number of 0's in this string has factors p' and $(k+1)$, where $p' \geq 2$ since p' is prime and $k+1 > 1$ since $k > 0$. So, $uv^ixy^iz \notin L$ for $i = p' + 1$.

Thus, we cannot satisfy both properties 1 and 3 in any rewriting of s as $uvxyz$.

Therefore, L is not context free.

7) Give a high-level description of a Turing machine which recognizes the following language:

$$L = \{0^i 1^j 2^k \mid i \times j = k \text{ and } i, j, k \geq 1\}$$

1. Scan the tape from left to right and check if the input string has form some 0's followed by some 1's followed by some 2's. If not, reject. If so, return tape head to beginning.
2. Mark a 0 then move the tape head to the start of 1's. Alternate between 1's and 2's and mark each one until all 1's are marked. If any 1's remain after all 2's have been marked, reject.
3. Unmark all of the 1's and repeat step 2 if there are more 0's left unmarked. If all 0's are marked:
   - If all 2's are marked, accept
   - Otherwise, reject.