CSE303 Q4 Practice Solutions Spring 2011

YES/NO questions Circle the correct answer. Write SHORT justification.

1. The set of terminals in a context free grammar G is a subset of alphabet of G
   Justify: $\Sigma \subseteq V$ **y**
2. The set of terminals and non- terminals in a context free grammar G form the alphabet of G
   Justify: $V = \Sigma \cup (V - \Sigma)$ **y**
3. The set of non-terminals is always non- empty
   Justify: $S \in V$ **y**
4. The set of terminals is always non- empty
   Justify: Finite set can be empty v **n**
5. $L(G) = \{w \in V : S \Rightarrow^*{}_G w\}$
   Justify: $w \in \Sigma^*$ **n**
6. $L \subseteq \Sigma^*$ is context-free if and only if $L = L(G)$
   Justify: only when G is a context -free grammar **n**
7. A language is context-free if and only if it is accepted by a context-free grammar.
   Justify: Generated, not accepted **n**
8. Any regular language is context-free
   Justify: 1. Any Finite Automata is a PDF automata
   2.Regular languages are generated by regular grammars, that are also CF. **y**
9. Language is regular if and only if is generated by a regular grammar (right- linear)
   Justify: proof in class **y**
10. $L = \{w \in \{a, b\}^* : w = w^R\}$ is context-free
    Justify: G with the rules: $S \rightarrow aSa|bSb|a|b||e$ **y**
11. The stack alphabet of a pushdown automaton is always non- empty
    Justify: finite set can be empty **n**
12. The input alphabet of a pushdown automaton is always non- empty
    Justify: finite set can be empty **n**
13. $\Delta \subseteq (K \times \Sigma^* \times \Gamma^*) \times (K \times \Gamma^*$ is a transition relation of a pushdown automaton (lecture definition)
    Justify: Δ must be finite **n**
14. $\Delta \subseteq (K \times (\Sigma \cup \{e\}) \times \Gamma^*) \times (K \times \Gamma^*$ is a transition relation of a pushdown automaton (book definition)
    Justify: Δ must be finite **n**

15. $L(M) = \{w \in \Sigma^* : (s, w, e) \models^*{}_M (f, e, e)\}$
Justify: $f \in F$ **n**

16. Any regular language is accepted by a pushdown automaton
Justify: Any finite automata is a pushdown automata operating on an empty stock. **y**

17. The class of languages accepted by pushdown automata is exactly the class of context-free languages
Justify: main theorem **y**

18. Context-free languages are not closed under union
Justify: we construct a CF grammar that is union of CF grammars **n**

19. Context-free languages are closed under intersection
Justify: Take $L_1 = a^n b^n c^m, L_2 = a^m b^n c^n$, both CF and we get that $L_1 \cap L_2 = a^n b^n c^n$ is not CF **n**

20. The intersection of a context-free language and regular language is a context-free language
Justify: basic theorem **y**

21. The union of a context-free language and regular language is a context-free language
Justify: regular language is also a context free language and context free languages are closed under union **y**

22. Every subset of a regular language is a language.
Justify: a subset of a set is a set. **y**

23. Any regular language is accepted by some PD automata.
Justify: Any regular language is accepted by a finite automata, and a finite automaton is a PD automaton (that never operates on the stock). **y**

24. A parse tree is always finite.
Justify: Any derivation of w in a CF grammar is finite. **y**

25. Parse trees are equivalence classes.
Justify: represent equivalence classes. **n**

26. For all languages, all grammars are ambiguous.
Justify: programming languages are never inherently ambiguous. **n**

27. A CF grammar G is called ambiguous if there is $w \in L(G)$ with at least two distinct parse trees.
Justify: definition **y**

28. A CF language L is inherently ambiguous iff all context-free grammars G, such that $L(G) = L$ are ambiguous.
Justify: definition **y**

29. Programming languages are sometimes inherently ambiguous.
Justify: never **n**

30. The largest number of symbols on the right-hand side of any rule of a CF grammar G is called called a fanout and denoted by $\phi(G)$.
Justify: definition **y**

31. The Pumping Lemma for CF languages uses the notion of the fanout.
Justify: condition on the length of $w \in L$ **y**

PROBLEMS

QUESTION 1 Consider a grammar $G = (V, \Sigma, R, S)$, where

$$V = \{a, b, S, A\}, \quad \Sigma = \{a, b\},$$

$$R = \{S \to AA, \quad A \to AAA \ |a \ |bA \ |Ab\}.$$

1. Which strings of $L(G)$ can be produced by derivations of 4 or fewer steps?

Solution

$S \Rightarrow AA \Rightarrow aA \Rightarrow aa$

$S \Rightarrow AA \Rightarrow bAA \Rightarrow baA \Rightarrow baa$

$S \Rightarrow AA \Rightarrow aA \Rightarrow abA \Rightarrow aba$

2. Give 2 derivations of a string $babbab$.

Solution

$S \Rightarrow AA \Rightarrow bAA \Rightarrow bAAb \Rightarrow bAbAb \Rightarrow bAbbAb \Rightarrow babbAb \Rightarrow babbab$

$S \Rightarrow AA \Rightarrow bAA \Rightarrow bAAb \Rightarrow bAbAb \Rightarrow bAbbAb \Rightarrow bAbbab \Rightarrow babbab$

3. For any $m, n, p > 0$, describe a derivation in G of the string $b^m ab^n ab^p$.

Solution

$S \Rightarrow AA \Rightarrow^m b^m AA \Rightarrow^n b^m Ab^n A \Rightarrow^p b^m Ab^n Ab^p \Rightarrow b^m Ab^n ab^p \Rightarrow b^m ab^n ab^p$

QUESTION 2 Construct a context-free grammar G such that

$$L(G) = \{w \in \{a, b\}^* : \ w = w^R\}.$$

Justify your answer.

Solution $G = (V, \Sigma, R, S)$, where

$$V = \{a, b, S\}, \quad \Sigma = \{a, b\},$$

$$R = \{S \to aSa \ |bSb \ | \ a \ | \ b \ | \ e\}.$$

Derivation example: $S \Rightarrow aSa \Rightarrow abSba \Rightarrow ababa$
$ababa^R = ((ab)a(ba))^R = (ba)^R a^R (ab)^R = ababa.$

Observation 1 We proved in class that for any $x, y \in \Sigma^*$, $(xy)^R = y^R x^R$. From this we have that

$$(xyz)^R = ((xy)z)^R = z^R(xy)^R = z^R y^R x^R$$

Grammar correctness justification: observe that the rules $S \to aSa \mid bSb \mid e$ generate the language $L_1 = \{ww^R : \ w \in \Sigma^*\}$. With additional rules $S \to a \mid b$ we get hence the language $L = L_1 \cup \{waw^R : \ w \in \Sigma^*\} \cup \{wbw^R : w \in \Sigma^*\}$. Now we are ready to prove that

$$L = L(G) = \{w \in \{a,b\}^* : \ w = w^R\}.$$

Proof Let $w \in L$, i.e. $w = xx^R$ or $w = xax^R$ or $w = xbx^R$. We show that in each case $w = w^R$ as follows.

c1: $w^R = (xx^R)^R = (x^R)^R x^R = xx^R = w$ (used property: $(x^R)^R = x$).

c2: $w^R = (xax^R)^R = (x^R)^R a^R x^R = xax^R = w$ (used Observation 1 and properties: $(x^R)^R = x$ and $a^R = a$).

c3: $w^R = (xbx^R)^R = (x^R)^R b^R x^R = xbx^R = w$ (used Observation 1 and properties: $(x^R)^R = x$ and $b^R = b$).

DEFINITION A context-free grammar $G = (V, \Sigma, R, S)$ is called **regular**, or **right-linear** iff

$$R \subseteq (V - \Sigma) \times \Sigma^*((V - \Sigma) \cup \{e\}).$$

QUESTION 3 Given a **Regular grammar** $G = (V, \Sigma, R, S)$, where

$$V = \{a, b, S, A\}, \quad \Sigma = \{a, b\},$$

$$R = \{S \to aS \mid A \mid e, \quad A \to abA \mid a \mid b\}.$$

1. Construct a finite automaton M, such that $L(G) = L(M)$.

Solution We construct a non-deterministic finite automata

$$M = (K, \Sigma, \Delta, s, F)$$

as follows:

$$K = (V - \Sigma) \cup \{f\}, \ \Sigma = \Sigma, s = S, \quad F = \{f\},$$

$$\Delta = \{(S, a, S), (S, e, A), (S, e, f), (A, ab, A), (A, a, f), (A, b, f)\}$$

2. Trace a transitions of M that lead to the acceptance of the string $aaaababa$, and compare with a derivation of the same string in G.

Solution

The accepting computation is:

$$(S, aaaababa) \vdash_M (S, aaababa) \vdash_M (S, aababa) \vdash_M (S, ababa) \vdash_M (A, ababa)$$

$$\vdash_M (A, aba) \vdash_M (A, a) \vdash_M (f, e)$$

G derivation is:

$$S \Rightarrow aS \Rightarrow aaS \Rightarrow aaaS \Rightarrow aaaA \Rightarrow aaaabA \Rightarrow aaaababA \Rightarrow aaaababa$$

QUESTION 4 Construct a **pushdown** automaton M such that

$$L(M) = \{a^m b^n : \ m \leq n \leq 2m\}$$

Solution $M = \{K, \Sigma, \Gamma, \Delta, s, F\}$ for

$$K = \{s, f\}, \Sigma = \{a, b\}, \Gamma = \{a\}, s = s, F = \{f\},$$

$$\Delta = \{((s, a, e), (s, aa)), ((s, a, e), (s, a)), ((s, e, e), (f, e)), ((f, b, a), (f, e))\}$$

Trace a transitions of M that lead to the acceptance of the string *aaabbbb*.

Solution The accepting computation is:

$$(s, aaabbbb, e) \vdash_M (s, aabbbb, a) \vdash_M (s, abbbb, aa) \vdash_M (s, bbbb, aaaa)$$

$$\vdash_M (f, bbbb, aaaa) \vdash_M (f, bbb, aaa) \vdash_M (f, bb, aa) \vdash_M (f, b, a) \vdash_M (f, e, e)$$

QUESTION 5 Use closure under union to show that $L = \{a^n b^n : \ n \neq m\}$ is CONTEXT-Free.

Solution 1 We know that $L_1 = \{a^m b^n : m > n\}$ and $L_2 = \{a^m b^n : m < n\}$ are context-free languages (we constructed proper grammars for both of them). $L = L_1 \cup L_2$, hence L is context free as the class of context free languages is closed under union.

Solution 2 Observe that $L_1 = \{a^m b^n : m > n\} = \{a\}^+\{a^n b^n : n \in N\}$ We proved (Pumping Lemma)that $\{a^n b^n : n \in N\}$ is context free and the class of context free languages is closed under concatenation, hence L_1 is also context free.

Similarly, $L_2 = \{a^m b^n : m < n\} = \{a^n b^n : n \in N\}\{b\}^+$, so L_2 is context free. $L = L_1 \cup L_2$, hence L is context free as the class of context free languages is closed under union.

QUESTION 6 Prove that a language is regular iff there is a regular grammar that generates it.

Solution

Implication $\Rightarrow$: Assume that L is a regular language. Let $M = \{K, \Sigma, \delta, s, F\}$ be a deterministic finite state automaton such that $L = L(M)$. We construct a context free grammar G as follows.

$$G = (K \cup \Sigma, \Sigma, R, S = s)$$

for R defined below.

$$R = \{p \to \sigma q : \ \delta(p, \sigma) = q\} \cup \{q \to e : \ q \in F\}$$

We prove that $L(G) = L(M) = L$ by a straightforward induction on the length of derivation/computation.

Implication $\Leftarrow$: Assume the a grammar $G = (V, \Sigma, R, S)$ is regular, or right-linear. I.e. that

$$R \subseteq (V - \Sigma) \times \Sigma^*((V - \Sigma) \cup \{e\}).$$

Observe that the rules of R are, by definition, of the form:

$$A \to wB, \ A \to w, \ A \to e$$

for $A, B \in V - \Sigma$, $w \in \Sigma^*$. We define a non-deterministic finite automaton M as follows:

$$M = \{K = (V - \Sigma) \cup \{f\}, \Sigma, \Delta, s = S, F = \{f\}\}$$

for $\Delta = \{(A, w, B) : \ A \to wB \in R\} \cup \{(A, w, f) : \ A \to w \in R\}$.

QUESTION 7 Use Pumping Lemma to prove that

$$L = \{a^n b^n c^n : \quad n \in N\}$$

is NOT CF language.

Solution Assume that L is CF. Let $G = (V, \Sigma, R, S)$ be a CF grammar such that $L = L(G)$. Take $n > \frac{(\phi G)^{|V - \Sigma|}}{3}$ and $w = a^n b^n c^n$. Of course $|w| = 3n > (\phi G)^{|V - \Sigma|}$ and the Pumping Lemma holds. It means that we can split w in such a way that $w = uvxyz$ for $xynot = e$ and $uv^n xy^n z \in L$, for all $n \in N$.

Let's now look at the form of vy, i.e analyze which occurrences of a, b, c it contains. We have to consider two cases. Both of them lead to contradiction.

1. vy contains all occurrences of a, b, c. Then at least one of v, y must contain at least two of them; but in this case uv^2xy^2z contains two occurrences out of order as $(ab)^2 = abab$ and $uv^2xy^2z \notin L$. Contradiction.

2. If vy contains occurrences of some, but not all of a, b, c, then uv^2xy^2z has unequal number of a's, b's and c's and $uv^2xy^2z \notin L$. Contradiction.