Gradiance Online Accelerated Learning

Zayd

- Home Page
- Assignments Due
- Progress Report
- Handouts
- Tutorials
- Homeworks
- Lab Projects
- Log Out

Help

Submission number:	73529
Submission certificate:	BC012386
Submission time:	2014-03-30 18:51:45 PST (GMT - 8:00)

Number of questions:	6
Positive points per question:	3.0
Negative points per question:	1.0
Your score:	14

Based on Sections 7.2, 7.3, and 7.4 of HMU.

1. The intersection of two CFL's need not be a CFL. Identify in the list below a pair of CFL's such that their intersection is not a CFL.

a) $L_1=\{a^nb^nc^ic \mid n>0, i>0\}$
$L_2=\{a^ja^nb^ic^i \mid n>0, i>0, j>0\}$

b) $L_1=\{aca^nb^jc^ic \mid n>0, i>0, j>0\}$
$L_2=\{aca^ja^nb^ic^i \mid n>0, i>0, j>0\}$

c) $L_1=\{aba^nb^nc^nba \mid n>0, i>0\}$
$L_2=\{aba^nb^ic^jba \mid n>0, i>0, j>0\}$

d) $L_1=\{a^nb^jc^ic \mid n>0, i>0, j>0\}$
$L_2=\{a^ja^nb^ic^i \mid n>0, i>0, j>0\}$

Answer submitted: **c)**

Your answer is incorrect.

L_1 is not a context free language. You can use the pumping lemma (Section 7.2, p. 279) to prove this fact.

Question Explanation:

The incorrect choices fall in two categories: either one of the two given languages is not a CFL, or one is a CFL and the other is a regular language, in which case we know that their intersection is a CFL (see Section 7.3.4 on p. 291).

For the four correct choices the intersections of the two given languages are:

1. $\{a^nb^nc^n \mid n> 0\}$.
2. $\{aba^nb^nc^n \mid n> 0\}$.
3. $\{aba^nb^nc^nba \mid n> 0\}$.

3. $\{a a b^n b^n a a \mid n > 0\}$.
4. $\{a^n b^n c^n \mid n > 0\}$.

In all cases we can prove the language not to be context-free by using the pumping lemma on a word $a^n b^n c^n$ for sufficiently large n. If we pump any pair of strings (of length much smaller than n) then the balance on the number of a's, b's and c's will be ruined.

The correct choice is: **a)**

2. The language L = {ss | s is a string of a's and b's} is not a context-free language. In order to prove that L is not context-free we need to show that for every integer n, there is some string z in L, of length at least n, such that no matter how we break z up as $z = uvwxy$, subject to the constraints $|vwx| \leq n$ and $|vx| > 0$, there is some $i \geq 0$ such that uv^iwx^iy is not in L.

Let us focus on a particular z = aabaaaba and n = 7. It turns out that this is the wrong choice of z for n = 7, since there are some ways to break z up for which we can find the desired i, and for others, we cannot. Identify from the list below the choice of u,v,w,x,y for which there is an i that makes uv^iwx^iy not be in L. We show the breakup of aabaaaba by placing four |'s among the a's and b's. The resulting five pieces (some of which may be empty), are the five strings. For instance, aa|b||aaaba| means u=aa, v=b, w=ε, x=aaaba, and y=ε.

a) aab|a|aab|a|
b) a|ab|aa|ab|a
c) |aa|b|aa|aba
d) a|ab|aaa|b|a

Answer submitted: **d)**

You have answered the question correctly.

Question Explanation:

In all cases for this problem, $i = 0$ works if anything works. For example one correct choice is aab|a|a|a|ba, where if we remove the second and 4th pieces, we get aababa, which is not of the form ss. That is, the first half, aab, is not the same as the second half, and therefore aababa is not in L.

For each of the incorrect choices, the choice explanation gives an argument as to why no "pumping" of v and x will lead to a string whose first half differs from the second half.

3. G_1 is a context-free grammar with start symbol S_1, and no other nonterminals whose name begins with "S." Similarly, G_2 is a context-free grammar with start symbol S_2, and no other nonterminals whose name begins with "S." S_1 and S_2 appear on the right side of no productions. Also, no nonterminal appears in both G_1 and G_2.

We wish to combine the symbols and productions of G_1 and G_2 to form a new grammar G, whose language is the concatenation of the languages of G_1 and G_2.

The start symbol of G will be S. All productions and symbols of G_1 and G_2 will be symbols and productions of G. Which of the following sets of productions, added to those of G, is guaranteed to make L(G) be $L(G_1)L(G_2)$?

a) $S \rightarrow S_3S_2$, $S_3 \rightarrow S_4S_1$, $S_4 \rightarrow \varepsilon$

b) $S \rightarrow S_1S_3$, $S_3 \rightarrow S_1S_2$

c) $S \rightarrow S_1S_2 \mid \varepsilon$

d) $S \rightarrow S_1S_3$, $S_3 \rightarrow S_2 \mid \varepsilon$

Answer submitted: **a)**

You have answered the question correctly.

Question Explanation:

Each of the choices involves only S, S_1, S_2, and in some cases other nonterminals whose names begin with "S" and that therefore are known not to appear in G_1 or G_2. As a result, we need only to look at the strings involving S_1 and S_2 only, that are derivable from S, using the new productions. In order for L(G) to be $L(G_1)L(G_2)$, it is necessary and sufficient that the strings involving only symbols S_1 and S_2 that are derived from S using only the additional productions be exactly the one string $\{S_1S_2\}$.

For example, adding $S \rightarrow S_1S_2$ obviously has this property. So does the set of productions

$S \rightarrow S_1S_3S_2$
$S_3 \rightarrow e$

4. G_1 is a context-free grammar with start symbol S_1, and no other nonterminals whose name begins with "S." Similarly, G_2 is a context-free grammar with start symbol S_2, and no other nonterminals whose name begins with "S." S_1 and S_2 appear on the right side of no productions. Also, no nonterminal appears in both G_1 and G_2.

We wish to combine the symbols and productions of G_1 and G_2 to form a new grammar G, whose language is the union of the languages of G_1 and G_2. The start symbol of G will be S. All productions and symbols of G_1 and G_2 will be symbols and productions of G. Which of the following sets of productions, added to those of G, is guaranteed to make L(G) be $L(G_1)$ [union] $L(G_2)$?

a) $S \rightarrow S_1S_2$

b) $S \rightarrow S_1$, $S_1 \rightarrow S_2$, $S_2 \rightarrow \varepsilon$

c) $S \rightarrow S_1 \mid S_3$, $S_3 \rightarrow S_2$

d) $S \rightarrow S_1S_3$, $S_3 \rightarrow S_2$

Answer submitted: **c)**

You have answered the question correctly.

Question Explanation:

Each of the choices involves only S, S_1, S_2, and in some cases other nonterminals whose names begin with "S" and that therefore are known not to appear in G_1 or G_2. As a result, we need only to look at the strings involving S_1 and S_2 only, that are derivable from S, using the new productions. In order for L(G) to be $L(G_1)$ [union]

$L(G_2)$, it is necessary and sufficient that the strings involving only symbola S_1 and S_2 that are derived from S using only the additional productions be exactly the two strings $\{S_1, S_2\}$.

For example, adding $S \rightarrow S_1 \mid S_2$ obviously has this property. So does the set of productions

$S \rightarrow S_1$
$S_1 \rightarrow S_2$

Note that if S_1 could appear on the right side of productions of G_1, then this choice would not work --- derivations of G_2 could suddently appear in the middle of derivations that should be in G_1 only.

$S \rightarrow S_1 \mid S_3$
$S_3 \rightarrow S_2$

5. If *h* is the homomorphism defined by h(a) = 0 and h(b) = ε, which of the following strings is in $h^{-1}(000)$?

a) abbba

b) baabbbabb

c) babab

d) abbbabaab

Answer submitted: **b)**

You have answered the question correctly.

Question Explanation:

Since there are three 0's in h(w), *w* must have exactly three *a*'s. It can have any number of *b*'s, since h(b) = ε. Of the choices, only baabbbabb has exactly three *a*'s.

6. Apply the CYK algorithm to the input ababaa and the grammar:

```
S → AB | BC
A → BA | a
B → CC | b
C → AB | a
```

Compute the table of entries X_{ij} = the set of nonterminals that derive positions *i* through *j*, inclusive, of the string ababaa. Then, identify a true assertion about one of the X_{ij}'s in the list below.

a) $X_{23} = \{A\}$

b) $X_{26} = \{S,A\}$

c) $X_{26} = \{B\}$

d) $X_{36} = \{S,A,C\}$

Answer submitted: **b)**

You have answered the question correctly.

Question Explanation:

Here is the table:

B					
SAC	SA				
B	B	SA			
B	SC	B	-		
SC	SA	SC	SA	B	
AC	B	AC	B	AC	AC
a	**b**	**a**	**b**	**a**	**a**