



Gradiane Online Accelerated Learning

Zayd

- [Home Page](#)
- [Assignments Due](#)
- [Progress Report](#)
- [Handouts](#)
- [Tutorials](#)
- [Homeworks](#)
- [Lab Projects](#)
- [Log Out](#)

Submission number: 71192
Submission certificate: FF943426
Submission time: 2014-03-30 18:17:18 PST (GMT - 8:00)

Number of questions: 6
Positive points per question: 3.0
Negative points per question: 1.0
Your score: 10

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

Help

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

$S \rightarrow AB \mid BC$
 $A \rightarrow BA \mid a$
 $B \rightarrow CC \mid b$
 $C \rightarrow AB \mid a$

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) $X_{16} = \{A\}$
- b) $X_{23} = \{A\}$
- c) $X_{15} = \{S, C\}$
- d) $X_{16} = \{B\}$

Answer submitted: **c)**

Your answer is incorrect.

Here are some suggestions:

1. Remember that in the given input ababaa, positions 1, 3, 5, and 6 hold a , and positions 2 and 4 hold b .
2. To compute X_{ij} , ask which variables have a body consisting of only the i th input symbol; those are the variables in X_{ii} .
3. To compute X_{ij} for $j > i$, you need (among other things) to compare $X_{i,j-1}$ ("the first set") with X_{ij} ("the second set"). Try to find a variable Y in the first set and a variable Z in the second set, such that YZ is a production body. For each such production body, put the head in X_{ij} . Then, proceed to let $X_{i,j-2}$ be the first set and $X_{j-1,j}$ be the second set, and repeat. March down the indexes, considering all pairs of a first set X_{ik} and a second set $X_{k+1,j}$.

The complete CYK algorithm is described in Section 7.4.4 (p. 303).

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	

The correct choice is: **d)**

2. If h is the homomorphism defined by $h(a) = 0$ and $h(b) = \epsilon$, which of the following strings is in $h^{-1}(000)$?
- a) babab
 - b) aabbaa
 - c) aababb
 - d) abbbabaab

Look for the string with three a's.

Answer submitted: **c)**

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 $h(b) = \epsilon$. Of the choices, only aababb has exactly three a 's.

3. The language $L = \{ss \mid s \text{ is a string of } a\text{'s and } b\text{'s}\}$ 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, $a|b|aa|ba|a$ means $u = a, v = b, w = aa, x = ba, y = a$, and uv^iwx^iy

instance, $u=aa$, $v=b$, $w=\epsilon$, $x=aaaba$, and $y=\epsilon$.

- a) $a|ab|aaa|b|a$
- b) $|a|ab|a|aaba$
- c) $a|aba|a|aba|$
- d) $|aa|b|aa|aba$

Answer submitted: **c)**

Your answer is incorrect.

When we pump this v and x , we get strings of the form $a(aba)^i a(aba)^i$. Such strings are always of the form ss , with $s = a(aba)^i$. You should examine the statement of the Pumping Lemma in Section 7.2.2 (p. 280) and the examples in Section 7.2.3.

Question Explanation:

In all cases for this problem, $i = 0$ works if anything works. For example one correct choice is $aab|a|a|ba$, 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.

The correct choice is: **a)**

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 $L(G_1) \cup L(G_2)$?

- a) $S \rightarrow S_1 S_2, S_1 \rightarrow \epsilon, S_2 \rightarrow \epsilon$
- b) $S \rightarrow S_1 S_3 \mid S_2 S_3, S_3 \rightarrow \epsilon$
- c) $S \rightarrow S_1 S_2 \mid S_2 S_1$
- d) $S \rightarrow S_1 S_2$

Answer submitted: **b)**

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 $L(G_1) \cup L(G_2)$, 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 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

$$\begin{aligned} S &\rightarrow S_1 \\ S_1 &\rightarrow S_2 \end{aligned}$$

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 suddenly appear in the middle of derivations that should be in G_1 only.

$$\begin{aligned} S &\rightarrow S_1 \mid S_3 \\ S_3 &\rightarrow S_2 \end{aligned}$$

5. 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) $S \rightarrow S_1S_3, S_3 \rightarrow S_2$
- b) $S \rightarrow S_1S \mid S_2S \mid \epsilon$
- c) $S \rightarrow S_1S_2 \mid \epsilon$
- d) $S \rightarrow S_1 \mid S_2$

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

$$S \rightarrow S_1S_3S_2$$

$$S_3 \rightarrow e$$

6. 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) $L_1 = \{aba^n b^n c^i \mid n > 0, i > 0\}$
 $L_2 = \{aba^n b^j c^i \mid n > 0, i > 0, j > 0\}$
- b) $L_1 = \{a^n b^n c^i \mid n > 0, i > 0\}$
 $L_2 = \{a^i a^n b^j c^i \mid n > 0, i > 0, j > 0\}$
- c) $L_1 = \{a^n b^j c^i \mid n > 0, i > 0, j > 0\}$
 $L_2 = \{a^i a^n b^j c^i \mid n > 0, i > 0, j > 0\}$
- d) $L_1 = \{aca^n b^j c^i \mid n > 0, i > 0, j > 0\}$
 $L_2 = \{aca^i a^n b^j c^i \mid n > 0, i > 0, j > 0\}$

Answer submitted: **b)**

You have answered the question correctly.

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. $\{a^n b^n c^n \mid n > 0\}$.
2. $\{aba^n b^n c^n \mid n > 0\}$.
3. $\{aba^n b^n c^n ba \mid n > 0\}$.
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.