Gradiance Online Accelerated Learning

Zayd

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

Help

Submission number:	68328
Submission certificate:	EG334705
Submission time:	2014-03-10 18:05:44 PST (GMT - 8:00)

Number of questions:	5
Positive points per question:	4.0
Negative points per question:	0.0
Your score:	20

Chapter 04: Minimization of DFA's Chapter 05: CFG's Chapter 05: CFG's --- Proofs Chapter 05: Parse Trees

1. The grammar G:

$S \rightarrow SS \mid a \mid b$

is ambiguous. That means at least some of the strings in its language have more than one leftmost derivation. However, it may be that some strings in the language have only one derivation. Identify from the list below a string that has exactly TWO leftmost derivations in G.

a) aa

b) bab

c) aaaa

d) ab

Answer submitted: **b)**

You have answered the question correctly.

Question Explanation:

A string of length 1 has only one leftmost derivation, e.g., $S \Rightarrow_{lm} a$. A string of length 2 has only one derivation, e.g., $S \Rightarrow_{lm} SS \Rightarrow_{lm} aS \Rightarrow_{lm} ab$.

However, a string of length 3 has exactly two derivations, e.g., $S \Rightarrow_{lm} SS \Rightarrow_{lm} SSS \Rightarrow_{lm} aSS \Rightarrow_{lm} abS \Rightarrow_{lm} aba$ and $S \Rightarrow_{lm} SS \Rightarrow_{lm} aS \Rightarrow_{lm} aSS \Rightarrow_{lm} abS \Rightarrow_{lm} aba$. In general, we can decide whether the first S generates a single terminal or two S's.

On the other hand, strings of length four or more have more than two generations. We can either start $S \Rightarrow_{lm} SS \Rightarrow_{lm} SSS \Rightarrow_{lm} SSSS$ or $S \Rightarrow_{lm} SS \Rightarrow_{lm} SSS \Rightarrow_{lm} aSS \Rightarrow_{lm} aSSS$ or $S \Rightarrow_{lm} SS \Rightarrow_{lm} aS \Rightarrow_{lm} aSS$, and there are other options as well.

2. Consider the grammars:

G_1:$S \rightarrow AB \mid a \mid abC$, $A \rightarrow b$, $C \rightarrow abC \mid c$

G_2:S → a | b | cC, C → cC | c

These grammars do not define the same language. To prove, we use a string that is generated by one but not by the other grammar. Which of the following strings can be used for this proof?

a) ababccc
b) cccc
c) ababababcccc
d) a

Answer submitted: **b)**

You have answered the question correctly.

Question Explanation:

Grammar G_1 does not use the first production for S because there is no production for B. It generates C to be

$(ab)^i c$, i=0,1,.... Then it uses the last production for S to generate all strings $(ab)^i c$, i=1,2,... Finally it uses the second production of S to generate string a.

Thus, G_1 produces the language {a, $(ab)^i c$, i=1,2,...}.

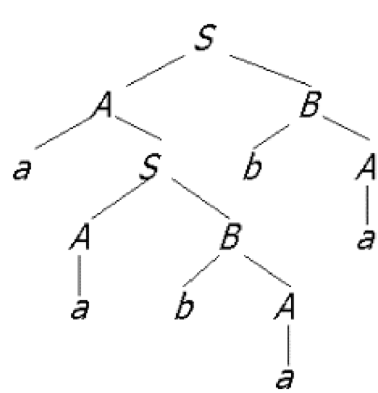
Grammar G_2 produces language {a,b,c^i, i=2,3,...}

Explanation of correct choices:

Strings b,ccccccc,cc,ccc,cccc: not generated by G_1.

Strings abc, ababc, abababc, ababababababc, ababababc: not generated by G_2.

3. The parse tree below represents a rightmost derivation according to the grammar `S → AB, A → aS|a, B → bA`.



Which of the following is a right-sentential form in this derivation?

a) aAbAba

a) aAbAba
b) aABbA
c) aabAba
d) aSbA

Answer submitted: **a)**

You have answered the question correctly.

Question Explanation:

To construct a rightmost derivation from a parse tree, we always replace the rightmost nonterminal in a right-sentential form. It helps to remember that each symbol of each right-sentential form corresponds to a node N of the parse tree. When a nonterminal is replaced by a production body, the introduced symbols correspond to the children of node N, in order from the left.

Thus, we start with right-sentential form S, where the symbol S corresponds to the root of the parse tree. At the first step, S is replaced by AB, the children of the root; A corresponds to the left child of the root, and B to the right child.

Since B is the rightmost nonterminal of AB, it is replaced by the string bA, formed by the labels of the two children of the right child of the root. The next right-sentential form is thus AbA. We proceed in this manner, next replacing the second of the two A's. The complete rightmost derivation is:

S => AB => AbA => Aba => aSba => aABba => aAbAba => aAbaba => aababa

4. Design the minimum-state DFA that accepts all and only the strings of 0's and 1's that have 110 as a substring. To verify that you have designed the correct automaton, we will ask you to identify the true statement in a list of choices. These choices will involve:

1. The number of *loops* (transitions from a state to itself).
2. The number of transitions into a state (including loops) on input 1.
3. The number of transitions into a state (including loops) on input 0.

Count the number of transitions into each of your states ("in-transitions") on input 1 and also on input 0. Count the number of loops on input 1 and on input 0. Then, find the true statement in the following list.

a) There is one loop on input 0 and two loops on input 1.
b) There are two states that have two in-transitions on input 0.
c) There is one state that has two in-transitions on input 0.
d) There are two states that have no in-transitions on input 1.

Answer submitted: **b)**

You have answered the question correctly.

Question Explanation:

Here is the transition table for the DFA. The intuitive meanings of the states are:

A = Nothing of 110 has been found, e.g., before any input has been read, or after you have just seen 00.
B = We've seen 1, but not 11.
C = We've seen 11 but never seen 110.
D = We've seen 110, either now or at some time in the past.

D — We've seen 110, either now or at some time in the past.

	0	1
→A	A	B
B	A	C
C	D	C
*D	D	D

5. Consider the grammars:

G_1: S → SaS | aa | a
G_2: S → SS | ε
G_3: S → SS | a
G_4: S → SS | aa
G_5: S → Sa | a
G_6: S → aSa | aa | a
G_7: S → SAS | ε

Describe the language of each of these grammars. Then, identify from the list below a pair of grammars that define the same language.

a) G_3, G_2

b) G_5, G_2

c) G_3, G_7

d) G_1, G_6

Answer submitted: **d)**

You have answered the question correctly.

Question Explanation:

G_1, G_3, G_5 and G_6 define L={a^i, i=1,2,...}
G_2 and G_7 define the language {ε}. G_4 defines L'={a^i, where i is even}.
Choice explanations provide a word that is produced by one grammar and not by the other.