Gradiance Online Accelerated Learning

Zayd

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

Help

Submission number:	84757
Submission certificate:	IF526365
Submission time:	2014-05-13 22:30:50 PST (GMT - 8:00)

Number of questions:	25
Positive points per question:	4.0
Negative points per question:	0.0
Your score:	88

1. Let L be the language of all strings of a's and b's such that no prefix (proper or not) has more b's than a's. Let G be the grammar with productions

```
S → aS | aSbS | ε
```

To prove that L = L(G), we need to show two things:

1. If S =>* w, then w is in L.
2. If w is in L, then S =>* w.

We shall consider only the proof of (1) here. The proof is an induction on n, the number of steps in the derivation S =>* w. Here is an outline of the proof, with reasons omitted. You need to supply the reasons.

Basis:

1) If n=1, then w is ε because _______.

2) w is in L because _______.

Induction:

3) Either (a) S => aS $=>^{n-1}$ w or (b) S => aSbS $=>^{n-1}$ w because _______.

4a) In case (a), w = ax, and S $=>^{n-1}$ x because _______.

5a) In case (a), x is in L because _______.

6a) In case (a), w is in L because _______.

4b) In case (b), w can be written w = aybz, where S $=>^{p}$ y and S $=>^{q}$ z for some p and q less than n because _______.

5b) In case (b), y is in L because _______.

6b) In case (b), z is in L because _______.

7b) In case (b), w is in L because _______.

For which of the steps above the appropriate reason is contained in the following argument:
"The following two statements are true
(i) if string x has no prefix with more b's than a's, then neither does string ax,
(ii) if strings y and z are such that no prefix has more b's than a's, then neither does string aybz."

a) 6b
b) 7b
c) 5a
d) 5b

You did not answer this question.

2. Use the construction from Theorem 10.15 (p. 457) to convert the following clauses:

1. (a+b)
2. (c+d+e+f)
3. (g+h+i+j+k+l+m)

to products of 3 literals per clause. In each case, the new clauses must be satisfiable if and only if the original clause is satisfiable. For the first clause, introduce variables x_1, x_2,... in that order from the left; for the second introduce y_1, y_2,... in that order from the left, and for the third introduce z_1, z_2,... in that order from the left. Use -w as shorthand for NOT w. Then identify, in the list below, the one clause that would appear among the clauses generated by the construction.

a) $(i+z_2+-z_3)$

b) $(m+z_4+-z_5)$

c) $(j+x_2+-z_3)$

d) $(h+z_1+-z_2)$

Answer submitted: **c)**

You have answered the question correctly.

3. Here is the transition table of a DFA that we shall call *M*:

	0	**1**
→A	B	G
B	C	H
*C	D	G
*D	A	H
E	F	C
F	G	I
*G	H	C
*H	A	D
I	E	I

Find the minimum-state DFA equivalent to the above. States in the minimum-state DFA are each the merger of some of the states of *M*. Find in the list below a set of states of *M* that forms one state of the minimum-state DFA.

a) {C,H}

b) {D,G}

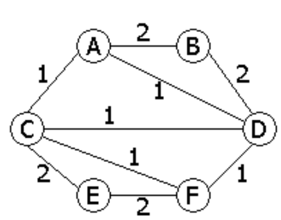
c) {C,G}

d) {B,E}

Answer submitted: **c)**

You have answered the question correctly.

4. Find all the minimum-weight Hamilton circuits in the graph below:



Then, identify in the list below the edge that is NOT on any minimum-weight Hamilton circuit.

a) (E,F)

b) (D,F)

c) (B,D)

d) (C,D)

Answer submitted: **d)**

You have answered the question correctly.

5. The operation DM(L) is defined as follows:

1. Throw away every even-length string from L.
2. For each odd-length string, remove the middle character.

For example, if L = {001, 1100, 10101}, then DM(L) = {01, 1001}. That is, even-length string 1100 is deleted, the middle character of 001 is removed to make 01, and the middle character of 10101 is removed to make 1001.

It turns out that if L is a regular language, DM(L) may or may not be regular. For each of the following languages L, determine what DM(L) is, and tell whether or not it is regular.

- L_1: the language of regular expression (01)*0.
- L_2: the language of regular expression (0+1)*1(0+1)*.
- L_3: the language of regular expression (101)*.
- L_4: the language of regular expression 00*11*.

Now, identify the true statement below.

a) $DM(L_4)$ is not regular; it consists of all strings of the form 0^n1^n.

b) $DM(L_1)$ is not regular; it consists of all strings of the form $(01)^n(10)^n$ for $n \geq 1$.

c) $DM(L_3)$ is not regular; it consists of all strings of the form $(101)^n11(101)^n$.

d) $DM(L_4)$ is regular; it is the language of regular expression 0*1*.

Answer submitted: **c)**

You have answered the question correctly.

6. Here is an instance of the Modified Post's Correspondence Problem:

	List A	List B
1	01	010
2	11	110
3	0	01

If we apply the reduction of MPCP to PCP described in Section 9.4.2 (p. 404), which of the following would be a pair in the resulting PCP instance.

a) (*$, $)

b) ($, *$)

c) (0*1*, *0*1*0*)

d) ($, $*)

Answer submitted: **b)**

You have answered the question correctly.

7. 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) $S \rightarrow S_1S_1$, $S_1 \rightarrow S_2$

c) $S \rightarrow S_1S \mid S_2S \mid \varepsilon$

d) $S \rightarrow S_1.S_2$

Answer submitted: **a)**

You have answered the question correctly.

8. Suppose we execute the Chomsky-normal-form conversion algorithm of Section 7.1.5 (p. 272). Let $A \rightarrow BC0DE$ be one of the productions of the given grammar, which has already been freed of ε-productions and unit productions. Suppose that in our construction, we introduce new variable X_a to derive a terminal *a*, and when we need to split the right side of a production, we use new variables Y_1, Y_2,...

What productions would replace $A \rightarrow BC0DE$? Identify one of these replacing productions from the list below.

a) $Y_3 \rightarrow DE$

b) $Y_3 \rightarrow DY_4$

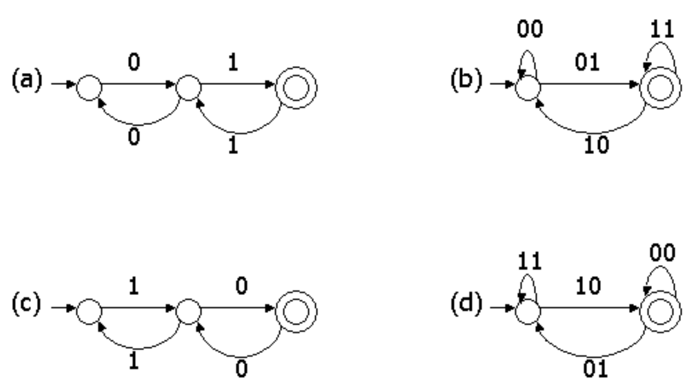
c) $Y_3 \rightarrow X_0Y_4$

d) $Y_1 \rightarrow Y_2C$

Answer submitted: **a)**

You have answered the question correctly.

9. Which automata define the same language?



Note: (b) and (d) use transitions on strings. You may assume that there are nonaccepting intermediate states, not shown, that are in the middle of these transitions, or just accept the extension to the conventional finite automaton that allows strings on transitions and, like the conventional FA accepts strings that are the concatenation of labels along any path from the start state to an accepting state.

a) b and c

b) a and d

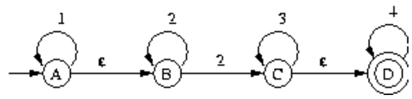
c) a and b

d) a and c

Answer submitted: **c)**

You have answered the question correctly.

10. Here is a nondeterministic finite automaton with epsilon-transitions:



Suppose we use the extended subset construction from Section 2.5.5 (p. 77) to convert this epsilon-NFA to a deterministic finite automaton with a dead state, with all transitions defined, and with no state that is inaccessible from the start state. Which of the following would be a transition of the DFA?

Note: we use S-x->T to say that the DFA has a transition on input x from state S to state T.

a) {B,C,D}-2->{C,D}

b) {A,B}-3->{C,D}

c) {C,D}-4->{D}

d) {A,B}-ε->{B}

Answer submitted: **c)**

You have answered the question correctly.

11. A Turing machine *M* with start state q_0 and accepting state q_f has the following transition function:

δ(q,a)	**0**	**1**	**B**
q_0	$(q_0,1,R)$	$(q_1,1,R)$	(q_f,B,R)
q_1	$(q_2,0,L)$	$(q_2,1,L)$	(q_2,B,L)
q_2	-	$(q_0,0,R)$	-
q_f	-	-	-

Deduce what *M* does on any input of 0's and 1's. Hint: consider what happens when *M* is started in state q_0 at the left end of a sequence of any number of 0's (including zero of them) and a 1. Demonstrate your understanding by identifying the true transition of *M* from the list below.

a) $q_0$1100 |-* 1111Bq_f

b) $q_0$0011 |-* 1100q_f

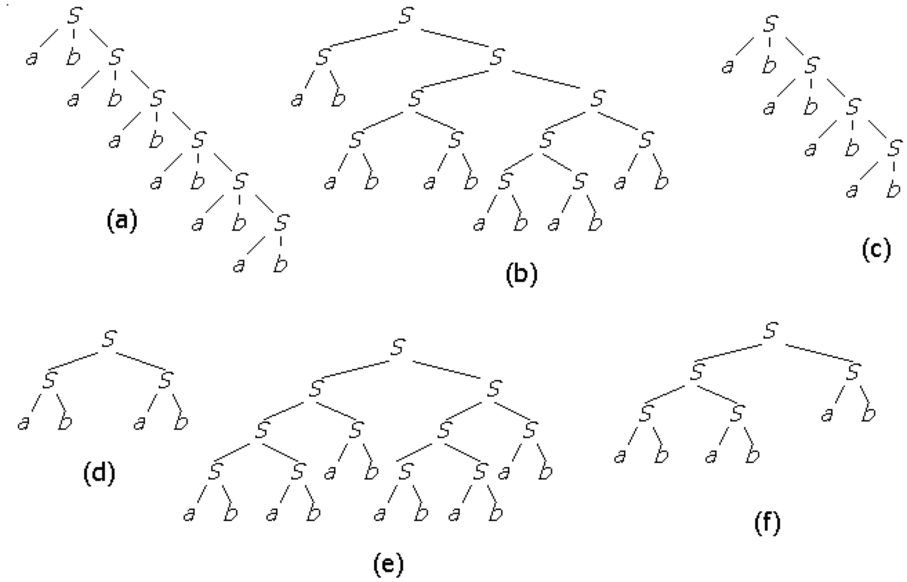
c) $q_0$0101 |-* 1010Bq_f

d) $q_0$1010 |-* 1001Bq_f

Answer submitted: **c)**

You have answered the question correctly.

12. Which of the following is a parse tree for the grammar S → abS, S → ab?



a) (d)
b) (c)
c) (b)
d) (e)

Answer submitted: **b)**

You have answered the question correctly.

13. The grammar G:

S → SS | a | 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) aaaa
b) aba
c) b
d) abbab

Answer submitted: **b)**

You have answered the question correctly.

14. A *unit pair* (X,Y) for a context-free grammar is a pair where:

1. X and Y are variables (nonterminals) of the grammar.
2. There is a derivation X =>* Y that uses only unit productions (productions with a body that consists of exactly one occurrence of some variable, and nothing else).

For the following grammar:

```
S → A | B | 2
A → C0 | D
B → C1 | E
C → D | E | 3
D → E0 | S
E → D1 | S
```

Identify all the unit pairs. Then, select from the list below the pair that is NOT a unit pair.

a) (E,D)
b) (B,A)
c) (C,E)
d) (D,C)

Answer submitted: **d)**

You have answered the question correctly.

15. Let L be the language of all strings of a's and b's such that no prefix (proper or not) has more b's than a's. Let G be the grammar with productions

```
S → aS | aSbS | ε
```

To prove that L = L(G), we need to show two things:

1. If S =>* w, then w is in L.
2. If w is in L, then S =>* w.

We shall consider only the proof of (1) here. The proof is an induction on n, the number of steps in the derivation S =>* w. Here is an outline of the proof, with reasons omitted. You need to supply the reasons.

Basis:

1) If n=1, then w is ε because _______.

2) w is in L because _______.

Induction:

3) Either (a) S => aS $=>^{n-1}$ w or (b) S => aSbS $=>^{n-1}$ w because _______.

4a) In case (a), w = ax, and S $=>^{n-1}$ x because _______.

5a) In case (a), x is in L because _______.

6a) In case (a), w is in L because _______.

4b) In case (b), w can be written w = aybz, where S $=>^{p}$ y and S $=>^{q}$ z for some p and q less than n because _______.

5b) In case (b), y is in L because _______.

6b) In case (b), z is in L because _______.

7b) In case (b), w is in L because _______.

Some of the steps above have one of the following reasons:

I) "The following two statements are true:
(i) if string x has no prefix with more b's than a's, then neither does string ax,
(ii) if strings y and z are such that no prefix has more b's than a's, then neither does string aybz."

II) "All n-step derivations of w produce either ε (for n=1) or use one of the productions with at least one nonterminal in the body (for n > 1). In case the production S → aS is used, then w=ax with x being produced by a (n-1)-step derivation. In case the production S → aSbS is used then w=aybz with y and z being produced by derivations with number of steps less than n."

III) "by the inductive hypothesis"

Choose as correct a (STEP, REASON) pair. (I.e., a correct pair means that step STEP is true because of reason REASON.)

a) (6a,II)
b) (4b,III)
c) (4a,II)
d) (1,III)

Answer submitted: **a)**

Your answer is incorrect.

The correct reason for 6a is:
If x has no prefix with more a's than b's, w = ax surely has no such prefix.

The "Only If" part of Theorem 5.7 (p. 180) is a useful example, as is the proof of Theorem 5.18 (p. 193). Also, see Section 1.4.2 (p. 22) on the general form of inductions on integers (which includes an induction on the lengths of derivations).

16. Consider the grammar G1:

$S \rightarrow \varepsilon \mid aS \mid aSbS$

Which of the following is correct (for a choice to be correct, all propositions must be correct)?

a) a) G1 generates all and only the strings of a's and b's such that every string has at least as many a's as b's. b) The inductive hypothesis to prove it is: For n < k, it holds: Any word in G1 of length n, is such that all its prefixes contain more a's than b's or as many a's as b's.

b) For any word w with every prefix having at least as many a's as b's, there is a unique b in w such that w can be written as aw'bw" --- hence w can be generated from shorter words using the production $S \rightarrow aSbS$.

c) The string aaabbbababaabba is not generated by the grammar.

d) For any word w with every prefix having at least as many a's as b's, either there is a unique b in w such that w can be written as aw"bw'" with w" and w'" being such that every prefix has as many a's as b's--hence can be generated from shorter words of the grammar using the rule $S \rightarrow aSbS$-- or w can be written as w=aw' with w' having every prefix with as many a's as b's -- hence can be generated by the rule $S \rightarrow aS$.

Answer submitted: **d)**

You have answered the question correctly.

17. Here are the transitions of a determininstic pushdown automaton. The start state is q_0, and *f* is the accepting state.

State-Symbol	a	b	ε
q_0-Z_0	(q_1,AAZ_0)	(q_2,BZ_0)	(f,ε)
q_1-A	(q_1,AAA)	(q_1,ε)	-
q_1-Z_0	-	-	(q_0,Z_0)
q_2-B	(q_3,ε)	(q_2,BB)	-
q_2-Z_0	-	-	(q_0,Z_0)
q_3-B	-	-	(q_2,ε)
q_3-Z_0	-	-	(q_1,AZ_0)

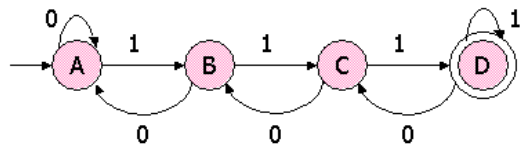
Describe informally what this PDA does. Then, identify below the one input string that the PDA accepts.

a) babbabbba
b) bababbbb
c) baabbba
d) bbbabaa

Answer submitted: **a)**

You have answered the question correctly.

18. Converting a DFA such as the following:



to a regular expression requires us to develop regular expressions for limited sets of paths --- those that take the automaton from one particular state to another particular state, without passing through some set of states. For the automaton above, determine the languages for the following limitations:

1. L_{AA} = the set of path labels that go from A to A without passing through C or D.
2. L_{AB} = the set of path labels that go from A to B without passing through C or D.
3. L_{BA} = the set of path labels that go from B to A without passing through C or D.
4. L_{BB} = the set of path labels that go from B to B without passing through C or D.

Then, identify a correct regular expression from the list below. Note: there are several different regular expressions possible for each of these languages. However, each of the correct answers can be thought of as built from more limited components. For example, the regular expression **1** is the set of path labels that go from A to B without passing through any of the four states.

a) L_{AA} = (0*10)*

b) L_{BA} = 0(00*10)*

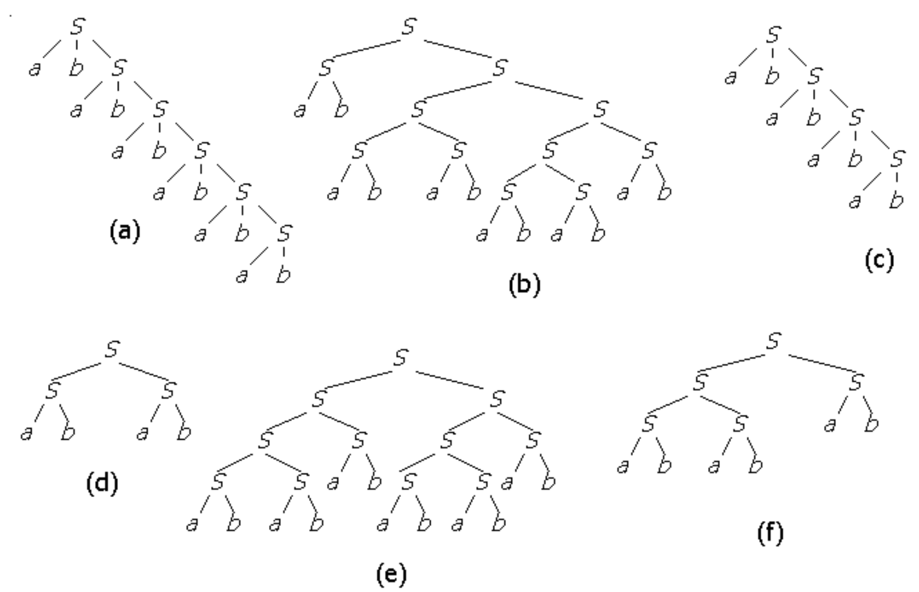
c) L_{BA} = 0(0+10)*

d) L_{BA} = (00*1+10)*0

Answer submitted: **c)**

You have answered the question correctly.

19. Consider the grammar G: S → abS, S → ab. Which of the following strings is a word of L(G) AND is the yield of one of the parse trees for grammar G in the figure below?



a) abab

b) ababababababab

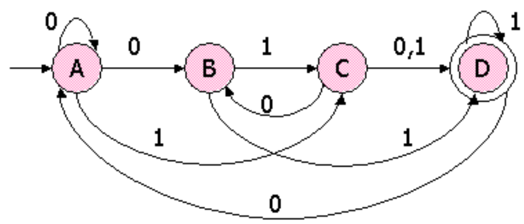
c) abababab

d) ababab

Answer submitted: **c)**

You have answered the question correctly.

20. Here is a nondeterministic finite automaton:



Some input strings lead to more than one state. Find, in the list below, a string that leads from the start state A to three different states (possibly including A).

a) 1110

b) 101000

c) 110110

d) 100010

Answer submitted: **d)**

You have answered the question correctly.

21. Consider the following languages and grammars. G_1: S → aA|aS, A → ab
G_2: S → abS|aA, A → a
G_3: S → Sa|AB, A → aA|a, B → b
G_4: S → aS|b
L_1: $\{a^ib \mid i=1,2,...\}$
L_2: $\{(ab)^iaa \mid i=0,1,...\}$
L_3: $\{a^ib \mid i=2,3,...\}$
L_4: $\{a^iba^j \mid i=1,2,..., j=0,1,...\}$
L_5: $\{a^ib \mid i=0,1,...\}$

Match each grammar with the language it defines. Then, identify a correct match from the list below.

a) G_1 defines L_2.

b) G_2 defines L_2.

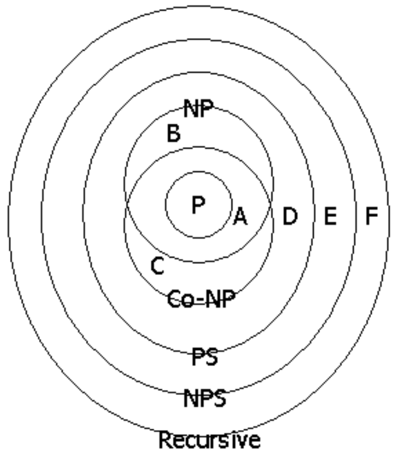
c) G_1 defines L_1.

d) G_3 defines L_5.

Answer submitted: **b)**

You have answered the question correctly.

22. In the diagram below we see certain complexity classes (represented as circles or ovals) and certain regions labeled A through F that represent the differences of some of these complexity classes.



The state of our knowledge regarding the existence of problems in the regions A-F is imperfect. In some cases, we know that a region is nonempty, and in other cases we know that it is empty. Moreover, if P=NP, then we would know more about the emptiness or nonemptiness of some of these regions, but still would not know everything.

Decide what we know about the regions A-F currently, and also what we would know if P=NP. Then, identify the true statement from the list below.

a) If P=NP, it would still not be known whether region F is empty.

b) Region A is definitely not empty.

c) It is not known whether region A is empty.

d) Region B is definitely not empty.

Answer submitted: **a)**

Your answer is incorrect.

Remember that a recursive language that can be recognized by a Turing machine that always halts. Is there a limit on the time or space used by such a Turing machine? See Section 9.2.1 (p. 383).

23. Find, in the list below, a regular expression whose language is the reversal of the language of this regular expression: 10*(0+1)*. Recall that the reversal of a language is formed by reversing all its strings, and the reversal of a string $a_1a_2...a_n$ is $a_n...a_2a_1$.

a) 1(0+1)*0*

b) 0*1(0+1)*

c) (0+1)*0*1

d) (0+1)*10*

Answer submitted: **c)**

You have answered the question correctly.

24. In this question, L_1, L_2, L_3, L_4 refer to languages and M, M_1, M_2 refer to Turing machines. Let
L_1 = {(M_1,M_2) | L(M_1) is a subset of L(M_2)},
L_2 = {M | There exists an input on which TM M halts within 100 steps},
L_3 = {M | There exists an input *w* of size less than 100, such that M accepts *w*},
L_4 = {M | L(M) contains at least 2 strings}.

Decide whether each of L_1, L_2, L_3 and L_4 are recursive, RE or neither. Then identify the true statement below.

a) The complement of L_2 is recursive.

b) The complement of L_3 is recursively enumerable.

c) The intersection of L_2 and L_3 is not recursively enumerable.

d) L_2 is not recursive.

Answer submitted: **a)**

You have answered the question correctly.

25. Consider the languages.
(a) $\{0^{2n}1^n \mid n > 0\}$
(b) $\{0^{5n}1^n \mid n > 0\}$
(c) {w | w a string of 0's and 1's such that when interpreted in reverse as a binary integer it is a multiple of 5}
(d) $\{0^n1^n|n>0\}$
(e) {w | w a string of 0's and 1's such that its length is a perfect square}
(f) {w | w string of 0's and 1's such that when interpreted as a binary integer it is not a multiple of 5}
(g) {w | w a string of 0's and 1's such that its length is not a perfect cube}
(h) {w | w a string of 0's and 1's such that the number of 0's is not equal to twice the number of 1's}

Which is a regular language?

a) (h)

b) (c)

c) (b)

d) (g)

Answer submitted: **b)**

You have answered the question correctly.