《编译原理》课程期中考试试卷

2017 年 5 月

姓名：________________学号：______________专业：______________________

一、Single Choice (12 cents)

1、 which of the following language is generated by the context free grammar G: S→xSx | y

[A] xyx [B] (xyx)* [C] $x^n y x^n$ (n>=0) [D] x*yx*

2、 If the context-free grammar G is not ambiguous, for any sentence generated by G，which of the following description is correct? _________

[A] the parse tree corresponding to the left-most derivation must be the same to that of the right-most derivation.

[B] the parse tree corresponding to the left-most derivation may not be the same to that of the right-most derivation.

[C] the left-most derivation must be the same to the right-most derivation.

[D] there will be more than one left-most derivation, but the parse tree is the same.

3、If one CFG grammar is LL(1) and contains the rules: $A\rightarrow\alpha_1|\alpha_2$; $B\rightarrow\beta_1|\beta_2$, then the following condition ( ) must be satisfied.

[A] First (A) ∩First (B) is empty [B] First(α_1) ∩First(α_2) is empty

[C] First (A) ∩Follow (A) is empty [D] First (B) ∩Follow (A) is empty

4、LR(1) item [A → α . B γ , a], follow(B)= { }.

[A] a [B] γ [C] { γ , a } [D] { γa }

5、the parsing method of YACC is ( )

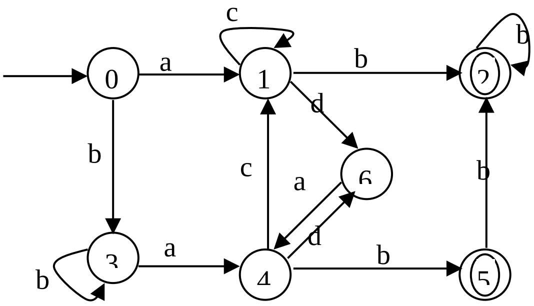
[A] LALR(1) [B]LR(1) [C] SLR(1) [D] LL(1)

6、In the Bottom-Up Parsing, the action ( ) will never be used.

[A] Reduce [B] Match [C] Shift [D] Accept

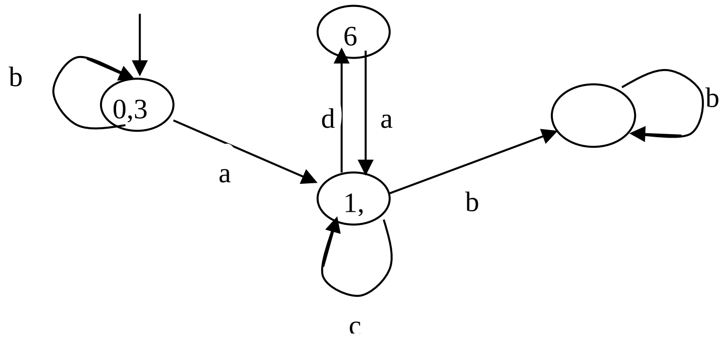
二、question (48 cents)

1、Apply the state minimization algorithm to the following DFA. Then give a regular expression to describe this language. (10 cents)



Answer:

b*a (da|c)*b+ （**4 分**）



2. Show that the following grammar is ambiguous.(8 cents)

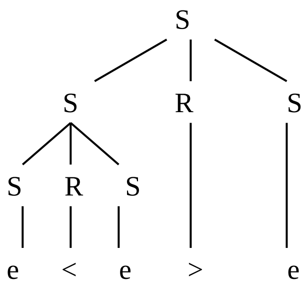
G: $S \rightarrow S\ R\ S \mid e$
$R \rightarrow < \mid >$

[Answer]

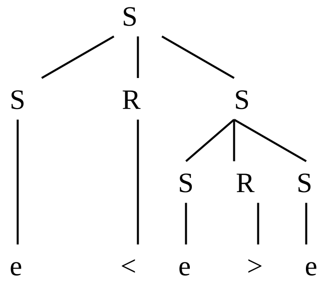
Given the string e<e>e, which belongs to the language L(G)

There are two distinct parsing trees illustrated as follows:

(1) The first parsing tree



(2) The second parsing tree



Therefore, according to the definition of ambiguous grammar, the grammar G is ambiguous.

2. Consider the following grammar (30 cents)

P → A | L
L → (S)
S→ S P| P
A → n|i

(1) Remove the left recursion

(2) Construct First and Follow sets from the non-terminals of the resulting grammar

(3) Construct the LL(1) parsing table for the resulting grammar.

(4) Show the actions of the corresponding LL(1) parser, given the input string (i (i (n))(i))

[Answer]

(1) The revised version of the above grammar after removing the left recursion is below: (5 cents)

P → A | L
L → (S)
S→ PS’
S’→ PS’| ε
A → n|i

(2) The First and Follow sets for the non-terminals in the revised grammar (7 cents)

First(P)={n, i, (}	Follow(P)={$, n, i, (,) }（不一定有 $）
First(L)={ (}	Follow(L)={ $, n, i, (,) } (不一定有$)
First(S)={ n, i, (}	Follow(S)={) }
First(S’)={ n, i, (,ε}	Follow(S’)={)}
First(A)={ n, i}	Follow(A)={ $, n, i, (,)} (不一定有$)

(3) The LL(1) parsing table of the revised grammar (10 cents)

	n	i	(	)	$
P	P → A	P → A	P → L		
L			L → (S)		
S	S→ PS’	S→ PS’	S→ PS’		
S’	S’→ PS’	S’→ PS’	S’→ PS’	S’→ ε	
A	A → n	A → i			

(4) Given the input string (i (i (n))(i)), the actions of the corresponding LL(1) parser are as follows: (8 cents)

Stack	Input string	Actions
$P	(i(i(n))(i))$	Generation
$L	(i(i(n))(i))$	Generation
$)S(	(i(i(n))(i))$	Match
$)S	i(i(n))(i))$	Generation
$)S’P	i(i(n))(i))$	Generation

$)S’A	i(i(n))(i))$	Generation
$)S’i	i(i(n))(i))$	Match
$) S’	(i(n))(i))$	Generation
$)S’ P	(i(n))(i))$	Generation
$)S’ L	(i(n))(i))$	Generation
$)S’)S(	(i(n))(i))$	Match
$)S’)S	i(n))(i))$	Generation
$)S’) S’P	i(n))(i))$	Generation
$)S’) S’A	i(n))(i))$	Generation
$)S’) S’i	i(n))(i))$	Match
$)S’) S’	(n))(i))$	Generation
$)S’) S’ P	(n))(i))$	Match
….	…	…