

平成21年度	後期定期試験問題
科目名	コンバイラ
担当教員	石原真紀夫
実施日付	2月1日(月) 4限目
持ち込み	許可・禁止

情報工学科 年 組 学籍番号

\_\_\_\_\_点／102点

氏名

問1 次の文法1の下降型構文解析LL(1)に関する以下の設問に答えなさい。BNFの記号として ::=〔生成規則〕を用います。

(文法1)

非終端記号 A B C D

終端記号 「 」 a 、 (←加マ)

生成規則

- A ::= 「B」C ①  
B ::= aD ②  
C ::= 、 「B」C ③  
C ::= ε ④ ※ε (イ・ヨウ)は長さ0の語です  
D ::= aD ⑤  
D ::= ε ⑥

出発記号 A

設問1 文法1における次のFirst()とFollow()を答えなさい。ここで、Follow(A) = {\$}です。 【各2計14】

- 〔1〕First(A) 〔2〕First(B) 〔3〕First(C)  
〔4〕First(D) 〔5〕Follow(B) 〔6〕Follow(C)  
〔7〕Follow(D)

設問2 文法1のLL解析表を以下に示す。入力記号列を「a」、「aa」として構文解析を行いなさい。解析過程は1ステップずつ示すこととし、複数のステップを一度に行わないこと。 【10】

	「	」	a	、	\$
A	A→「B」C				
B			B→aD		
C				C→、 「B」C	C→ε
D		D→ε	D→aD		

問2 次の文法2のLL解析表を作りたい。各生成規則①～④が入る解析表内の場所に番号①～④をそれぞれ答えなさい。同じ生成規則が複数の場所に入る場合もあります。BNFの記号として ::=〔生成規則〕を用います。 【各4計16】

(文法2)

非終端記号 A B C

終端記号 { } ; s

生成規則

- A ::= {B} ①  
B ::= C ; B ②  
B ::= ε ③  
C ::= s ④

出発記号 A

問3 問1の文法1の上昇型構文解析Simple LR(1)に関する以下の設問に答えなさい。

※LR(O)項の括弧[]，集合の括弧{}，closure()やgoto()の引数の括弧()を明確に区別して記述してください。

設問1 文法1における次のclosure()とgoto()を答えなさい。

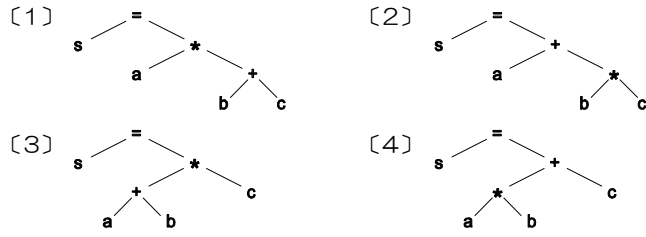
- 〔1〕closure({[A→「・B」C]}) 【各2計10】  
〔2〕closure({[C→、 「B」・C]})  
〔3〕goto({[A→「・B」C], [B→・aD]}, a)  
〔4〕goto({[A→「B・」C],、})  
〔5〕goto({[A→「B」・C], [C→・、 「B」C], [C→・、 ], C)

設問2 文法1のLR解析表を以下に示す。入力記号列を「a」、「aa」として構文解析を行いなさい。解答欄に示した4ステップまでを参考に続きを完成させなさい。解析過程は1ステップずつ示すこととし、複数のステップを一度に行わないこと。 【10】

(LR解析表)

	「	」	a	、	\$	A	B	C	D
0	s 2				1				
1				Acc					
2			s 4				3		
3		s 5							
4		r 6	s 7						6
5				s 9	r 4			8	
6		r 2							
7		r 6	s 7						10
8					r 1				
9	s 1 1								
10		r 5							
11			s 4				12		
12		s 1 3							
13				s 9	r 4			14	
14					r 3				

問4 次の算術式の構文木に関する設問に答えなさい。



設問1 それぞれの構文木の逆ポーランド記法表現を答えなさい。 【各3計12】

設問2 次は算術式の3番地コード表現です。それぞれの構文木と同等の演算を行う3番地コード表現を記号で答えなさい。 【各3計12】

- 〔a〕 (\*, a, b, X) 〔b〕 (+, a, b, X)  
(+, X, c, s) (\*, X, c, s)  
〔c〕 (\*, b, c, X) 〔d〕 (+, b, c, X)  
(+, a, X, s) (\*, a, X, s)

問5 次の分岐文と繰り返し文を処理するスタック機械用のコードを生成した。それぞれを処理するスタック機械用のコードをコード群から記号で選びなさい。スタック機械のそれぞれの命令コードの機能と意味を下の表に示す。 【各3計18】

- 〔1〕 if (a<b) a=2 ;  
〔2〕 if (a==b) a=2 ;  
〔3〕 if (a==b) a=2 ; else b=2 ;  
〔4〕 while (a!=b) a+=2 ;  
〔5〕 while (a<b) a+=2 ;  
〔6〕 do a+=2 ; while (a!=b) ;  
(コード群)

- 〔a〕 PUSH a 〔b〕 L1: PUSH a  
PUSH b PUSH 2  
EQOP ADD  
FJUMP L1 ASSIGN a  
PUSH 2 PUSH a  
ASSIGN a PUSH b  
JUMP L2 NEOP  
L1: PUSH 2 TJUMP L1  
ASSIGN b L2:

- 〔c〕 PUSH a 〔d〕 PUSH a  
PUSH b PUSH b  
NEOP LTOP  
TJUMP L1 FJUMP L1  
PUSH 2 PUSH 2  
ASSIGN a ASSIGN a

- 〔e〕 L1: PUSH a 〔f〕 L1: PUSH a  
PUSH b PUSH b  
LTOP NEOP  
FJUMP L2 FJUMP L2  
PUSH a PUSH a  
PUSH 2 PUSH 2  
ADD ADD  
ASSIGN a ASSIGN a  
JUMP L1 JUMP L1  
L2: L2:

(スタック機械命令コード)

命令(オペコード)	パラメータ(オペランド)	機能	意味
PUSH	para	プッシュ	paraが変数であればその中の値をスタックへ積み、paraが数値であればその値をスタックへ積み。
POP		ポップ	スタックのトップの値を取り出す。
ASSIGN	var	アサイン	スタックのトップをポップし、その値を変数varへ書き込む。
JUMP	label	分岐	ラベルlabelへ飛ぶ。
FJUMP	label	分岐	スタックのトップをポップし、0であればラベルlabelへ飛ぶ。
TJUMP	label	分岐	スタックのトップをポップし、0でなければラベルlabelへ飛ぶ。
INV		符号反転	スタックのトップをポップし、その値の符号を反転する。結果をスタックのトップへプッシュする。
ADD		加算	スタックのトップと2番目をポップし、それらを加算する。結果をスタックのトップへプッシュする。
SUB		減算	スタックのトップと2番目をポップし、2番目からトップを減する。結果をスタックのトップへプッシュする。
MULT		乗算	スタックのトップと2番目をポップし、それらを乗算する。結果をスタックのトップへプッシュする。
DIV		除算	スタックのトップと2番目をポップし、2番目をトップで割る。結果をスタックのトップへプッシュする。
MOD		剰余	スタックのトップと2番目をポップし、2番目をトップで割ったときの余りを計算する。結果をスタックのトップへプッシュする。
GTOP		>	スタックのトップと2番目をポップし、2番目がトップより大きければ1、そうでなければ0をスタックのトップへプッシュする。
GEOP		>=	スタックのトップと2番目をポップし、2番目がトップ以上であれば1、そうでなければ0をスタックのトップへプッシュする。
LTOP		<	スタックのトップと2番目をポップし、2番目がトップより小さければ1、そうでなければ0をスタックのトップへプッシュする。
LEOP		<=	スタックのトップと2番目をポップし、2番目がトップ以下であれば1、そうでなければ0をスタックのトップへプッシュする。
EQOP		==	スタックのトップと2番目をポップし、2番目とトップが等しいならば1、そうでなければ0をスタックのトップへプッシュする。
NEOP		!=	スタックのトップと2番目をポップし、2番目とトップが等しくないならば1、そうでなければ0をスタックのトップへプッシュする。
ANDOP		&&	スタックのトップと2番目をポップし、2番目とトップで一方または両方が0ならば0、そうでなければ1をスタックのトップへプッシュする。
OROP			スタックのトップと2番目をポップし、2番目とトップで両方が0ならば0、そうでなければ1をスタックのトップへプッシュする。

【解答欄】

【問1】設問1

- 〔1〕First(A) =  
〔2〕First(B) =  
〔3〕First(C) =  
〔4〕First(D) =  
〔5〕Follow(B) =  
〔6〕Follow(C) =  
〔7〕Follow(D) =

【問1】設問2

	入力記号列	スタック	動作
1	「a」、「aa」\$	A\$	
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			

【問2】

	{	}	;	s	\$
A					
B					
C					

【問3】設問1

- 〔1〕closure({[A→「・B」C]}) =  
〔2〕closure({[C→、 「B」・C]}) =  
〔3〕goto({[A→「・B」C], [B→・aD]}, a) =  
〔4〕goto({[A→「B・」C],、}) =  
〔5〕goto({[A→「B」・C], [C→・、 「B」C], [C→・、 ], C) =

【問3】設問2

	入力記号列	生成規則⑥の右辺はεなので0個スタックから消し、状態4で左辺のDを読み状態6を加えます	スタック	動作
1	「a」、「aa」\$		0	s 2
2	a」、「aa」\$		2, 0	s 4
3	」、「aa」\$		4, 2, 0	r 6
4	」、「aa」\$		6, 4, 2, 0	r 2
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				

【問4】設問1

構文木	逆ポーランド記法表現
〔1〕	
〔2〕	
〔3〕	
〔4〕	

【問4】設問2

構文木	対応する3番地コード表現の記号
〔1〕	
〔2〕	
〔3〕	
〔4〕	

【問5】

構文木	対応するコードの番号	構文木	対応するコードの番号
〔1〕		〔4〕	
〔2〕		〔5〕	
〔3〕		〔6〕	

お疲れ様でした