平成20年度　**後期定期試験問題**
科目名　コンパイラ
担当教員　石原真紀夫
実施日付　1月 28日(水)　1時限
持ち込み　許 可 ・ 禁 止

情報工学科　年　組　学籍番号
氏名

問１　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。〔３５点〕
※文法の記述においてＢＮＦ形式の記号は網掛けで示す
※LR(0)項の括弧[ ]、集合の括弧｛｝、closure や goto の引数の括弧（）を明確に区別すること
設問１　文法Ａは０と１が交互にでる語を生成する。次の５通りの LR(0)項の集合において以下に示す closure()と goto()を求めよ。〔各３点、計１５点〕
（文法A）
非終端記号　E　T　　終端記号　0　1
生成規則
E ::= 0 T
T ::= 1 E
T ::= 1
出発記号　E
（LR(0)項の集合）
I_1 = {[E→0・T]}
I_2 = {[T→1・E],[T→1・]}
I_3 = {[T→1E・]}
I_4 = {[E→0・T],[T→・1E]}
I_5 = {[T→1・E],[T→1・]}
【解答欄】

(1) closure(I_1)=
(2) closure(I_2)=
(3) closure(I_3)=
(4) goto(I_4, T)=
(5) goto(I_5, E)=

設問２　文法Ｂはセミコロン；で終わる語を生成する。文法Ｂの生成規則に　E'→E を加えた文法 B'における LR(0)項の正規集合Ｃを求めよ。〔１０点〕
（文法B）
非終端記号　E　　終端記号　s　;
生成規則
E ::= s E
E ::= ;
出発記号　E

【解答欄】

設問３　文法Cは引数をもつ関数呼び出しを表す語を生成する。文法Cの LR 解析表は以下のように得られる。解答欄に示す２つの入力記号列それぞれの解析過程を示せ。解析過程は１行に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。解析中にエラーが発生した場合（解析表の該当欄が空の場合）は動作欄に「エラー」と記して処理を中止すること。〔各５点、計１０点〕
（文法Ｃ）
非終端記号　E　T　　終端記号　f　(　)　,　a
生成規則
E ::= f (T)　(1)
T ::= a , T　(2)
T ::= a　(3)
出発記号　E
（LR 解析表）※表中、r の後の数値は生成規則の番号に対応

	f	(	)	,	a	$	E	T
0	s2						1	
1						acc		
2		s3						
3					s5			4
4			s6					
5			r3	s7				
6						r1		
7					s5			8
8			r2					

【解答欄】

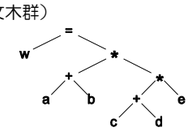
	入力記号列	スタック	動作
1	f (a , a) $	0	
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			

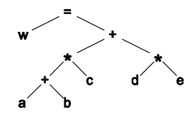
	入力記号列	スタック	動作
1	f ( ) $	0	
2			
3			
4			
5			

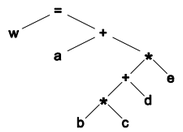
問２　次に示す算術式の中間コードに関する設問に答えなさい。〔４５点〕
(1)　w = (a + b * c + d) * e　　(2)　w = a + b * (c + d) * e
(3)　w = (a + b) * (c + d) * e　　(4)　w = a + (b * c + d) * e
(5)　w = (a + b) * c + d * e
設問１　各算術式(1)～(5)において同等の演算を行う３番地コードを（３番地コード群）より１つずつ記号で選びなさい。〔各３点、計１５点〕
（３番地コード群）
A. (+, a, b, A)　(*, A, c, A)　(*, d, e, B)　(+, A, B, w)
B. (+, c, d, A)　(*, A, b, A)　(*, A, e, A)　(+, a, A, w)
C. (*, b, c, A)　(+, A, d, A)　(*, A, e, A)　(+, a, A, w)
D. (+, a, b, A)　(+, c, d, B)　(*, A, B, A)　(*, A, e, w)
E. (*, b, c, A)　(+, a, A, A)　(+, A, d, A)　(*, A, e, w)
【解答欄】

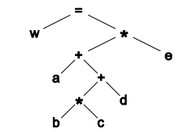
算術式	３番地コード	算術式	３番地コード
(1)		(4)	
(2)		(5)	
(3)			

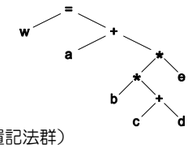
設問２　各算術式(1)～(5)において同等の演算を行う構文木を（構文木群）より１つずつ記号で選びなさい。さらに、各構文木の後置記法表現を（後置記法群）より１つ選びなさい。〔各３点、計３０点〕

（構文木群）
A.


B.


C.


D.


E.


（後置記法群）
A.　w a b c * d + e * + =
B.　w a b + c d + e * * =
C.　w a b + c * d e * + =
D.　w a b c d + * e * + =
E.　w a b c * d + + e * =
【解答欄】

構文木	構文木の後置記法
A.	
B.	
C.	
D.	
E	

問３　ある算術演算を実行するスタック機械用の機械語コードを生成したい。スタック機械は表（問題用紙裏面）に示す命令をもつものとする。以下に示す４つの算術演算の後置記法表現より機械語コードを生成しなさい。演算の実行前、スタックは空とし、実行後は演算の結果のみが残るものとする。結果が真または偽で与えられる場合は、0以外を真、0を偽とする。〔各5点、計20点〕

（後置記法）	（対応する算術演算）
(1) a b c * +	a + b * c
(2) a b + c *	(a + b) * c
(3) a b c * <	a < (b * c)
(4) a b c + !=	a != (b + c)

【解答欄】

(1)	(3)
(2)	(4)

お疲れ様です。

（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む。
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ずる。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。