

コンパイラ・期末テスト問題用紙
(2023年02月14日・10:30～12:00)

解答上、その他の注意事項

1. 問題は、問Ⅰ～Ⅲまでである。
2. 解答用紙の右上の欄に学籍番号・名前を記入すること。
3. 解答欄を間違えないよう注意すること。
4. 解答中の文字 (特に a と d) がはっきりと区別できるよう注意すること。
5. 持ち込みは不可である。筆記用具・時計・学生証以外のものは、かばんの中などにしまうこと。
6. テストの配点は 70 点である。合格は毎週の課題の得点を加点して、100 点満点中 60 点以上とする。

I. (演算子順位法)

次のBNFで表される文法を演算子順位法により構文解析する。

$$E \rightarrow \text{id} \mid E \wedge E \mid E * E \mid E < E \mid E , E \mid (E)$$

ただし、「E」は非終端記号、「id」,「^」,「*」,「<」,「,」,「(」,「)」は終端記号で、「id」はアルファベット1文字からなるトークンを表す。

この文法は曖昧なので、優先順位と結合性について次のように決めておく。

「^」は右結合、「*」は左結合、「<」は非結合、「,」は左結合、であり、「^」は「*」よりも優先順位が高く、「*」は「<」よりも優先順位が高く、「<」は「,」よりも優先順位が高いものとする。

つまり、下表中の左の欄の式は、右の欄の式として解釈される。

式	解釈	式	解釈
$a \wedge b \wedge c$	$a \wedge (b \wedge c)$	$a \wedge b , c$	$(a \wedge b) , c$
$a * b * c$	$(a * b) * c$	$a , b \wedge c$	$a , (b \wedge c)$
$a < b < c$	(構文エラー)	$a * b < c$	$(a * b) < c$
a , b , c	$(a , b) , c$	$a < b * c$	$a < (b * c)$
$a \wedge b * c$	$(a \wedge b) * c$	$a * b , c$	$(a * b) , c$
$a * b \wedge c$	$a * (b \wedge c)$	$a , b * c$	$a , (b * c)$
$a \wedge b < c$	$(a \wedge b) < c$	$a < b , c$	$(a < b) , c$
$a < b \wedge c$	$a < (b \wedge c)$	$a , b < c$	$a , (b < c)$

以下の演算子順位行列の空欄(1)～(15)を<, =, >, Xのうちもっとも適切なもので埋めよ。ただしXはエラーを表すものとする。(教科書などの記法では、エラーは空欄のままとしているが、このテストでは無回答と区別するために明示的にXを書くことにする。)

左 \ 右	,	<	*	^	()	id	終
始	<	<	<	<	<	X	<
,	(1)	(2)	<	<	<	>	<
<	(4)	(5)	(6)	<	<	>	<
*	>	>	(7)	(8)	<	>	<
^	>	>	(9)	(10)	<	(11)	<
(<	<	<	(12)	<	=	<
)	(13)	>	>	>	X	>	X
id	>	>	>	>	X	>	X

II. (再帰下降構文解析)

対のような BNF で定義された文法に対して再帰下降構文解析ルーチンを作成する。

$$\begin{array}{l}
 L \rightarrow A \mid L!A \\
 A \rightarrow N \mid A\&N \mid A\#N \\
 N \rightarrow \mathbf{id} \qquad \dots \textcircled{A} \\
 \quad \mid (L) \qquad \dots \textcircled{B}
 \end{array}$$

ただし、「 L 」, 「 A 」, 「 N 」は非終端記号、「 $!$ 」, 「 $\&$ 」, 「 $\#$ 」, 「 \mathbf{id} 」, 「 $($ 」, 「 $)$ 」は終端記号である。開始記号 (start symbol) は L である。

(1) L から導出される終端記号の列で、次の条件を満たすものの例を挙げよ。存在しなければ \mathbf{X} を記せ。

- (i) 「 \mathbf{id} 」の直後に「 $\#$ 」が続く。
- (ii) 「 $\#$ 」の直後に「 $!$ 」が続く。
- (iii) 「 $\&$ 」の直後に「 $($ 」が続く。
- (iv) 「 $)$ 」の直後に「 \mathbf{id} 」が続く。

(2) L, A から左再帰を除去せよ。補助的に導入する非終端記号はそれぞれ L', A' とせよ。後の解答で使用するために、生成規則に丸数字 (①, ②, ...) を付けておくこと。なお、採点の都合上、順番は次のようにせよ。

- 左再帰の除去で追加される ϵ は最後の選択肢とすること
- ϵ 以外の BNF の右辺の選択肢の順は、もとの BNF の選択肢の順と同じにすること
- 連続した番号をつけること

以下の問は (2). で L, A から左再帰を除去して得られた BNF について答えよ。

(3) $First(N)$ を求めよ。

(4) $Follow(L')$ を求めよ。

(5) $Follow(A')$ を求めよ。

(6) 下の予測型構文解析表の N の行を埋めよ。この問題の解答は \mathbf{X} , ①, ② の中から選べ。ただし、 \mathbf{X} は“エラー”を示す。無回答と区別するために、構文エラーの場合は、必ず \mathbf{X} を記入し、空欄のまま残さないこと。

(7) 下の予測型構文解析表の L, L', A, A' の行を埋めよ。この問題の解答は \mathbf{X} と ①, ②, ... (2) の解答で、BNF の生成規則に自分で付けた番号) から選べ。構文エラーの場合は、必ず \mathbf{X} を記入し、空欄のまま残さないこと。

	id	()	!	&	#	\$
$L \rightarrow$							
$L' \rightarrow$							
$A \rightarrow$							
$A' \rightarrow$							
$N \rightarrow$							

(8) この文法に対して、入力が文法にしたがっていれば「正しい構文です。」間違っていれば「構文に誤りがあります。」と表示する構文解析プログラムを作成する。プログラム (次ページ) 中の指定の部分に入る $L, L1, A, A1, N$ 関数のうち、 $A, A1$ 関数の定義を完成させよ。ただし、 $L, L1, A, A1, N$ は、それぞれ非終端記号 L, L', A, A', N に対応する関数である。終端記号を消費するときは eat 関数を、予測型構文解析表の \mathbf{X} に相当する入力には reportError 関数を呼び出すようにすること。

プログラムの補足説明:

プログラム中では、終端記号は「!」のような1文字のものは、その字そのもの（の文字コード）、`id`などのトークンは、C言語のマクロ（例えば`id`の場合は`ID`）として表現している。入力の終わり（`$`）に対応するのは、このプログラムの場合、改行文字 `'\n'`である。

`L, Ll, A, Al, N`関数が実行されるときは`token`という大域変数に、現在処理中（入力の先頭）の終端記号が代入されている。`eat`関数は、現在`token`に入っている値が、引数として与えられた終端記号と等しいかどうか確かめ、等しければ次の終端記号を`token`に読み込む。`reportError`関数は、「構文に誤りがあります。」と表示し、プログラムを終了する。

再帰下降構文解析プログラム

```
1 #include <stdio.h>      /* printf(), EOF など */
2 #include <stdlib.h>    /* exit()用      */
3 #include <ctype.h>     /* isalpha()用 */
4
5 /* 終端記号に対するマクロの定義 */
6 #define ID      257    /* トークン id */
7
8 int token;           /* 大域変数の宣言 */
9
10 /* 関数プロトタイプ宣言 */
11 void reportError(void);
12 void eat(int t);
13 int  yylex(void);
14
15
16 void L(void);
17 void Ll(void);
18 void A(void);
19 void Al(void);
20 void N(void);
21
22 /*
23 **** */
24 /* この部分に 関数 L, Ll, A, Al, N の定義を挿入する。
25 **** */
26
27 /* ここ以降は解答に直接関係はない。 */
28 void eat(int t) { /* token (終端記号) を消費して、次の token を読む */
29     if (token == t) {
30         /* 現在のトークンを捨てて、次のトークンを読む */
31         token = yylex();
32         return;
33     } else {
34         reportError();
35     }
36 }
37
38 void reportError(void) {
39     printf("構文に誤りがあります。 \n"); exit(0); /* プログラムを終了 */
40 }
41
42 int yylex(void) { /* 簡易字句解析ルーチン */
43     int c;
```

```

43 do { /* 空白は読み飛ばす。 */
44     c = getchar();
45 } while (c == ' ' || c == '\t');
46
47 if (c == EOF) { /* ファイルの終わり */
48     exit(0);
49 } else if (isalpha(c)) { /* アルファベットだったら... */
50     return ID;
51 } else {
52     /* 上のどの条件にも合わなければ、文字をそのまま返す。*/
53     return c; /* ';' など */
54 }
55 }
56
57 void processLine(void) { /* 各行の処理 */
58     L();
59     if (token == '\n') { /* 入力がブロックないように改行は特別扱い
*/
60         printf("Correct!\n"); /* eat('\n') の前に出力しておく */
61     }
62     eat('\n');
63 }
64
65 int main(void) { /* main関数 */
66     printf("Ctrl-c で終了します。 \n");
67     token = yylex(); /* 最初のトークンを読む */
68     while (1) { /* 無限ループ */
69         processLine(); /* 各行を処理する */
70     }
71
72     return 0;
73 }

```

III. (LR 構文解析)

次のような BNF で与えられる文法は曖昧であるが、優先順位・結合性を適切に指定することにより、LR 構文解析表を作成することができる。

$$\begin{array}{l}
 E \rightarrow a \quad \dots I \\
 | E ! E \quad \dots II \\
 | E E \quad \dots III \\
 | E * \quad \dots IV \\
 | (E) \quad \dots V
 \end{array}$$

ただし、

- …の後の I, IIなどは生成規則の番号である。
- 「E」は非終端記号である。「a」, 「!」, 「*」, 「(」, 「)」は終端記号である。このうち、「a」はアルファベット 1文字からなるトークンを表す。
- 開始記号 (start symbol) は (当然) E である。

bison の出力する LR 構文解析表は次のようになる。(注: bison に -v オプションを指定することによって、LR 構文解析表をファイルに出力させることができる。)

	a	!	*	()	\$		E
①	s ①			s ②				g ③
②	r I							
③	s ①	s ⑥	s ⑦	s ②		s ⑤		g ⑧
④	s ①	s ⑥	s ⑦	s ②	s ⑨			g ⑧
⑤	accept							
⑥	s ①			s ②				g ⑩
⑦	r IV							
⑧	r III		s ⑦		r III			g ⑧
⑨	r V							
⑩	s ①	r II	s ⑦	s ②	r II			g ⑧

注: ここで、s ② は、「シフト (shift) して状態 ② へ遷移」、g ③ は、「状態 ③ へ遷移 (go)」、r XII は、「生成規則 XII を使って還元 (reduce)」を表す。

オートマトンの開始状態は ① である。

次の (1) ~ (4) の入力列に対して、下線の記号をシフトした直後の (つまりシフトしたあと、還元がまだ起こっていないときの) スタックの状態はどのようになっているか?

(1) a a * a (2) a a * ! a (3) a ! a * a (4) (a ! a) * a

下の選択肢 ((1) ~ (4) 共通) から選べ。(左がスタックの底とする。)

- (A). ①a① (B). ①E③a① (C). ①E③E⑧!⑥a① (D). ①E③E⑧*⑦a① (E). ①E③E⑧!⑥E⑩a①
 (F). ①E③!⑥E⑧*⑦a① (G). ①E③E⑧a① (H). ①(②E④)⑨a① (I). ①(②E④)⑨*⑦a①
 (J). ①E③!⑥a① (K). ①E③E⑧*⑦E⑧!⑥a① (L). ①E③!⑥E⑩a① (M). ①E③*⑦E⑧!⑥a①

計算用紙

計算用紙

コンパイラ・期末テスト解答用紙（2023年02月14日）

学籍番号		氏名	
------	--	----	--

I. (1 × 13)

(1)		(2)		(3)		(4)		(5)	
(6)		(7)		(8)		(9)		(10)	
(11)		(12)		(13)					

II. (2 × 4, 6, 3, 3, 3, 2, 8, 4)

(1)	(i)							(ii)		
	(iii)							(iv)		
(2)	$L \rightarrow$									
	$L' \rightarrow$									
	$A \rightarrow$									
	$A' \rightarrow$									
(3)	{ }									
(4)	{ }									
(5)	{ }									
(6)			id	()	!	&	#	\$	
	$N \rightarrow$									
(7)			id	()	!	&	#	\$	
	$L \rightarrow$									
	$L' \rightarrow$									
	$A \rightarrow$									
$A' \rightarrow$										

