

コンパイラ・期末テスト問題用紙
(2022年02月08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 80 点である。合格は毎週の課題の得点を加算して、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。

$$E \rightarrow \text{id} \mid E + E \mid E > E \mid E \&\& E \mid E = E \mid (E)$$

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。

「+」は左結合、「>」は非結合、「&&」は左結合、「=」は右結合、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
$a + b + c$	$(a + b) + c$	$a + b = c$	$(a + b) = c$
$a > b > c$	(構文エラー)	$a = b + c$	$a = (b + c)$
$a \&\& b \&\& c$	$(a \&\& b) \&\& c$	$a > b \&\& c$	$(a > b) \&\& c$
$a = b = c$	$a = (b = c)$	$a \&\& b > c$	$a \&\& (b > c)$
$a + b > c$	$(a + b) > c$	$a > b = c$	$(a > b) = c$
$a > b + c$	$a > (b + c)$	$a = b > c$	$a = (b > c)$
$a + b \&\& c$	$(a + b) \&\& c$	$a \&\& b = c$	$(a \&\& b) = c$
$a \&\& b + c$	$a \&\& (b + c)$	$a = b \&\& c$	$a = (b \&\& c)$

以下の演算子順位行列の空欄(1)～(12)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)	<	(3)	<	>	<	>
&&	>	(4)	<	<	(5)	(6)	<	>
>	>	(7)	(8)	<	(9)	>	<	>
+	(10)	>	(11)	(12)	<	>	<	>
(<	<	<	(13)	<	≠	<	X
)	>	>	(14)	>	X	>	X	>
id	(15)	>	>	>	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。

$$\begin{aligned}
 S &\rightarrow \text{id} \{ E \} \mid S ; \text{id} \{ E \} \\
 E &\rightarrow F \mid E + F \mid E - F \\
 F &\rightarrow \text{id} \qquad \qquad \qquad \dots \textcircled{A} \\
 &\quad \mid \{ E ! S \} \qquad \qquad \dots \textcircled{B}
 \end{aligned}$$

ただし、「 S 」, 「 E 」, 「 F 」は非終端記号、「 id 」, 「 $\{$ 」, 「 $\}$ 」, 「 $;$ 」, 「 $+$ 」, 「 $-$ 」, 「 $!$ 」は終端記号である。開始記号 (start symbol) は S である。

(1) S 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) 「 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「 id 」が続く。

(2) S, E 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 S', E' 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 S, E から左再帰を除去して得られた BNF について答えよ。

(3) $First(F)$ を求めよ。

(4) $Follow(S')$ を求めよ。

(5) $Follow(E')$ を求めよ。

(6) 下の予測型構文解析表の F 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。(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 \mathbf{X} を書くことにする。)

(7) 下の予測型構文解析表の S, S', E, E' 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	{	}	;	+	-	!	\$
$S \rightarrow$								
$S' \rightarrow$								
$E \rightarrow$								
$E' \rightarrow$								
$F \rightarrow$								

(8) この文法に対して、入力が文法にしがっていれば「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 $S, S1, E, E1, F$ 関数のうち、 $E, E1$ 関数の定義を完

成させよ。ただし、S, S1, E, E1, F は、それぞれ非終端記号 S, S', E, E', F' に対応する関数である。予測型構文解析表の 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、マクロ `EOF` である。

`yylex` 関数は、入力を読んで、次の終端記号を返す関数である。`token` という大域変数に、現在処理中の終端記号を代入する。`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 int  yylex(void);
13 void eat(int t);
14
15 void S(void);
16 void S1(void);
17 void E(void);
18 void E1(void);
19 void F(void);
20
21 /* *****
22  * この部分に 関数 S, S1, E, E1, F の定義を挿入する。
23  * *****
24
25 /* ここ以降は解答に直接関係はない。 */
26 void reportError(void) {
27     printf("構文に誤りがあります。\\n"); exit(0); /* プログラムを終了
28 }
29
30 int main(void) { /* main関数 */
31     token = yylex(); /* 最初のトークンを読む */
32     S();
33     if (token == EOF) {
34         printf("正しい構文です!\\n");
35     } else {
36         reportError();
37     }
38 }
39
40 int yylex(void) { /* 簡易字句解析ルーチン */
41     int c;
42     char buf[256];
```

```

43
44 do { /* 空白は読み飛ばす。 */
45     c = getchar();
46 } while (c == ' ' || c == '\t' || c == '\n');
47
48 if (isalpha(c)) { /* アルファベットだったら... */
49     char* ptr = buf;
50     ungetc(c, stdin);
51     while (1) {
52         c = getchar();
53         if (!isalpha(c) && !isdigit(c)) break;
54         *ptr++ = c;
55     }
56     *ptr = '\0';
57     ungetc(c, stdin);
58
59     return ID;
60     reportError();
61 } else {
62     /* 上のどの条件にも合わなければ、文字をそのまま返す。*/
63     return c; /* ';' など */
64 }
65 }
66
67 void eat(int t) { /* token (終端記号) を消費して、次の token を読む
68     if (token == t) {
69         /* 現在のトークンを捨てて、次のトークンを読む */
70         token = yylex();
71         return;
72     } else {
73         reportError();
74     }
75 }

```

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。

$$\begin{array}{l}
 E \rightarrow \mathbf{while} E \mathbf{do} E \dots \text{I} \\
 | \mathbf{id} = E \dots \text{II} \\
 | E + E \dots \text{III} \\
 | \mathbf{id} \dots \text{IV}
 \end{array}$$

に対して、LR 構文解析表を作成する。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。「while」、「do」、「id」、「=」、「+」は終端記号である。このうち、「id」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。)

	while	do	id	=	+	\$	E
①	s ①		s ②				g ③
②	s ①		s ②				g ④
③	r IV		s ⑤	r IV			
④				s ⑦	s ⑥		
⑤		s ⑧		s ⑦			
⑥	s ①		s ②				g ⑨
⑦	accept						
⑧	s ①		s ②				g ⑩
⑨	s ①		s ②				g ⑪
⑩	r II			s ⑦	r II		
⑪	r III						
⑫	r I			s ⑦	r I		

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 = b ± c (2) a = b = c ± d (3) a = b + c ≡ while d do e
 (4) a = b + while c do d ± e

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). ①E③+⑦ (B). ①E③=⑤ (C). ①id②=⑤id②+⑦ (D). ①id②=⑤id②=⑤
 (E). ①id②=⑤E⑨+⑦ (F). ①id②=⑤E⑨=⑤ (G). ①E③=⑤id⑨=⑤ (H). ①E③=⑤E⑨+⑦
 (I). ①id②=⑤id②=⑤id②+⑦ (J). ①id②=⑤id②=⑤E⑨+⑦ (K). ①id②=⑤id②+⑦id②=⑤
 (L). ①id②=⑤E⑨+⑦id②=⑤ (M). ①id②=⑤E⑨+⑦E⑩+⑦
 (N). ①id②=⑤id②+⑦while①E④do⑧E⑩+⑦ (O). ①id②=⑤id②+⑦while①E④do⑧id②+⑦
 (P). ①id②=⑤E⑨+⑦while①E④do⑧id②+⑦ (Q). ①id②=⑤E⑨+⑦while①E④do⑧E⑩+⑦

計算用紙

計算用紙

コンパイラ・期末テスト解答用紙（2022年02月08日）

学籍番号		氏名	
------	--	----	--

I. (1 × 15)

(1)		(2)		(3)		(4)		(5)	
(6)		(7)		(8)		(9)		(10)	
(11)		(12)		(13)		(14)		(15)	

II. (2 × 4, 6, 3, 4, 4, 4, 8, 8)

(1- i)		(1- ii)																																																			
(1- iii)		(1- iv)																																																			
(2)	$S \rightarrow$ $S' \rightarrow$ $E \rightarrow$ $E' \rightarrow$																																																				
(3)																																																					
(4)																																																					
(5)																																																					
(6)	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <td style="width: 15%;"></td> <td style="width: 5%;">id</td> <td style="width: 5%;">{</td> <td style="width: 5%;">}</td> <td style="width: 5%;">;</td> <td style="width: 5%;">+</td> <td style="width: 5%;">-</td> <td style="width: 5%;">!</td> <td style="width: 5%;">\$</td> <td style="width: 20%;"></td> </tr> <tr> <td style="padding: 5px;">$F \rightarrow$</td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> </table>		id	{	}	;	+	-	!	\$		$F \rightarrow$																																									
	id	{	}	;	+	-	!	\$																																													
$F \rightarrow$																																																					
(7)	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <td style="width: 15%;"></td> <td style="width: 5%;">id</td> <td style="width: 5%;">{</td> <td style="width: 5%;">}</td> <td style="width: 5%;">;</td> <td style="width: 5%;">+</td> <td style="width: 5%;">-</td> <td style="width: 5%;">!</td> <td style="width: 5%;">\$</td> <td style="width: 20%;"></td> </tr> <tr> <td style="padding: 5px;">$S \rightarrow$</td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td style="padding: 5px;">$S' \rightarrow$</td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td style="padding: 5px;">$E \rightarrow$</td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td style="padding: 5px;">$E' \rightarrow$</td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> </table>		id	{	}	;	+	-	!	\$		$S \rightarrow$										$S' \rightarrow$										$E \rightarrow$										$E' \rightarrow$											
	id	{	}	;	+	-	!	\$																																													
$S \rightarrow$																																																					
$S' \rightarrow$																																																					
$E \rightarrow$																																																					
$E' \rightarrow$																																																					

