

プログラム言語論（'07年度）・テスト問題用紙

（'08年2月5日（火）・8:50～10:20）

解答上、その他の注意事項

- I. 問題は、問 I～VI までである。
- II. 解答用紙の右上の欄に学籍番号・名前を記入すること。
- III. 解答欄を間違えないよう注意すること。
- IV. 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。
- V. 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。
- VI. テストの配点は 80 点である。合格はレポートの得点を加えて、100 点満点中 60 点以上とする。

I. (Backus-Naur 記法)

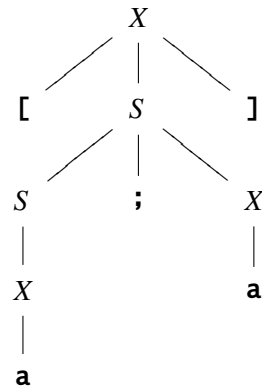
次のような BNF で表される文法を考える。

$$\begin{aligned} X &\rightarrow \text{“[” } S \text{ “]”} \\ &\quad | \quad \mathbf{a} \\ S &\rightarrow S \text{ “;” } X \\ &\quad | \quad X \end{aligned}$$

次の各文について、上の BNF の非終端記号 S から導出されるものには、その解析木 (parse tree) を右の例にならって書き、導出されないものには \times を記せ。(解析木は一通りとは限らないが、そのうちひとつを書けば良い。)

- (1) [a;a;a]
- (2) [[a]]
- (3) [[a;a];[]]

例: [a;a] に対する解析木



II. (正規表現)

「 $x(z|(yy)|(yx))^*x$ 」という正規表現に (一部でなく) 全体がマッチする文字列には (A) を、
「 $xy(x|(y*z)|z)^*y*yx$ 」という正規表現に全体がマッチする文字列には (B) を、両方に全体が
マッチする文字列には (C) を、どちらにも全体がマッチしない文字列には (D) をつけよ。

- (1) xyyzzyyx
- (2) xyyyyzyx
- (3) xyyxzzyyx
- (4) xyyxzxyyx

III. (コンパイラのフェーズ)

次の(1)~(3)のC言語のプログラムにはそれぞれ誤りがある。コンパイラのどのフェーズで誤りが検出されるか?(あるいはされないか?)もっとも適当なものを下の選択肢(A)~(E)から選べ。

- (1) (for文のセミコロン; が一つ多い)

```
#include <stdio.h>

int main(void) {
    int i;
    for (i=0; i<9; i++;) {
        printf("Hello World!¥n");
    }
    return 0;
}
```

- (2) (コメントの終わりを示す「*/」を忘れた。)

```
#include <stdio.h>

int main(void) {
    printf("Hello! World¥n"); /* 表示する *
    return 0;
}
```

- (3) (文字列リテラルに*演算子を適用しようとした。)

```
#include <stdio.h>

int main(void) {
    printf("Hello! World¥n" * 3);
    return 0;
}
```

(1)~(3)の選択肢

- (A) 字句解析フェーズでエラーが検出される。
- (B) 構文解析フェーズでエラーが検出される。
- (C) 意味解析フェーズでエラーが検出される。
- (D) コード生成フェーズでエラーが検出される。
- (E) 実行時にエラーとなるか、まったくエラーにならない(が作成者の意図と異なる動作をする)。

IV. (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。

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

ただし、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、「\$」は右結合で「+」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
$a + b + c$	$(a + b) + c$
$a + b > c$	$(a + b) > c$
$a > b + c$	$a > (b + c)$
$a > b > c$	構文エラー
$a > b \$ c$	$(a > b) \$ c$
$a \$ b > c$	$a \$ (b > c)$
$a \$ b \$ c$	$a \$ (b \$ c)$
$a + b \$ c$	$(a + b) \$ c$
$a \$ b + c$	$a \$ (b + c)$

以下の演算子順位行列の空欄(1)~(4)を <(シフト)、>(還元)、X(エラー)のうちもっとも適切なもので埋めよ。

左 \ 右	+	>	\$	()	id	終
始	<	<	<	<	X	<	=
+	>	(1)	>	<	>	<	>
>	(2)	(3)	>	<	>	<	>
\$	<	<	(4)	<	>	<	>
(<	<	<	<	=	<	X
)	>	>	>	X	>	X	>
id	>	>	>	X	>	X	>

V. (再帰下降構文解析)

次のような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。

$$\begin{aligned} E &\rightarrow \mathbf{id} \mid \{\} X \{\} \mid E \{\} X \{\} \mid E \cdot \mathbf{id} \\ X &\rightarrow \mathbf{id} \mathbf{=} E X' \\ X' &\rightarrow \varepsilon \mid \mathbf{,} \mathbf{id} \mathbf{=} E X' \end{aligned}$$

ただし、「 E 」、「 X 」、「 X' 」は非終端記号で、「 \mathbf{id} 」、「 $\{\}$ 」、「 $\}$ 」、「 \cdot 」、「 $\mathbf{=}$ 」、「 $\mathbf{,}$ 」は終端記号とする。開始記号 (start symbol) は E である。

- (1) E の左再帰を除去する。新しく補助的な非終端記号 E' を導入して、 E の書換え規則を

$$E \rightarrow \mathbf{id} E' \mid \{\} X \{\} E'$$

のようにする時、 E' の書換え規則を書け。

- (2) $First(X')$ を求めよ。
 (3) $Follow(X')$ を求めよ。
 (4) $Follow(E')$ を求めよ。

この文法に対する構文解析表を作成する。

	\mathbf{id}	$\{\}$	$\}$	\cdot	$\mathbf{,}$	$\mathbf{=}$	$\mathbf{\$}$
$E \rightarrow$	$\mathbf{id} E'$	$\{\} X \{\} E'$	\times	\times	\times	\times	\times
$E' \rightarrow$	(5)						
$X \rightarrow$	(6)						
$X' \rightarrow$	(7)						

以下の問に答えよ。なお、解答中でエラーとなる欄には明示的に \times と書き、空欄のままにしないこと。

- (5) E' の行を埋めよ。
 (6) X の行を埋めよ。
 (7) X' の行を埋めよ。

VI. (LR 構文解析)

次のような文法

$$\begin{aligned}
 E &\rightarrow a \quad \dots \text{ I} \\
 &| E \text{ “|” } E \quad \dots \text{ II} \\
 &| E E \quad \dots \text{ III} \\
 &| E \text{ “*”} \quad \dots \text{ IV} \\
 &| \text{ “(” } E \text{ “)”} \quad \dots \text{ V}
 \end{aligned}$$

に対して、演算子の優先度、結合性を通常の正規表現と同じになるように指定して、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、「|」、「*」、「(」、「)」は終端記号である。
- 「a」は終端記号で、アルファベット1文字からなるトークンを表す。

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		shift ②				shift ①	goto ③
①	reduce I						
②		shift ②				shift ①	goto ④
③	accept	shift ②		shift ⑥	shift ⑤	shift ①	goto ⑦
④		shift ②	shift ⑧	shift ⑥	shift ⑤	shift ①	goto ⑦
⑤		shift ②				shift ①	goto ⑨
⑥	reduce IV						
⑦	reduce III			shift ⑥	reduce III		goto ⑦
⑧	reduce V						
⑨	reduce II	shift ②	reduce II	shift ⑥	reduce II	shift ①	goto ⑦

ここで、shift ⑤は、「シフトして状態⑤へ遷移」、goto ⑤は、「状態⑤へ遷移」、reduce X は、「生成規則 X 番を使って還元」を表す。

次の入力に対して、↑の次(右)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?

$$\begin{array}{cc}
 (1) a*b*|cd* & (2) a*b*|cd* \\
 \uparrow & \uparrow
 \end{array}$$

下の選択肢から選べ。(左がスタックの底とする)

(1) の選択肢

(A).

①	a	①	a	①	*	⑥
---	---	---	---	---	---	---

(B).

①	E	③	*	⑥
---	---	---	---	---

(C).

①	E	③	a	①	*	⑥
---	---	---	---	---	---	---

(D).

①	E	③	E	⑦	*	⑥
---	---	---	---	---	---	---

(2) の選択肢

(A).

①	E	③		⑤
---	---	---	--	---

(B).

①	E	③	E	⑦		⑤
---	---	---	---	---	--	---

(C).

①	E	③	E	⑦	*	⑥		⑤
---	---	---	---	---	---	---	--	---

(D).

①	E	③	*	⑥	E	③	*	⑥		⑤
---	---	---	---	---	---	---	---	---	--	---

プログラム言語論（'07年度）・テスト解答用紙（'08年2月5日）

学籍番号		氏名	
------	--	----	--

I. (Backus-Naur 記法) (4×3)

(1).	(2).	(3).

II. (正規表現) (4×4)

(1).		(2).		(3).		(4).	
------	--	------	--	------	--	------	--

III. (コンパイラのフェーズ) (3×3)

(1).		(2).		(3).	
------	--	------	--	------	--

IV. (演算子順位法) (3,3,4,4)

(1).		(2).		(3).		(4).	
------	--	------	--	------	--	------	--

裏ページに続く。

