

コンパイラ (2012 年度) ・ 期末テスト問題用紙

(2012 年 08 月 02 日 (木) ・ 10:30 ~ 12:00)

(問題訂正適用済み)

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. 期末テストの配点は 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)

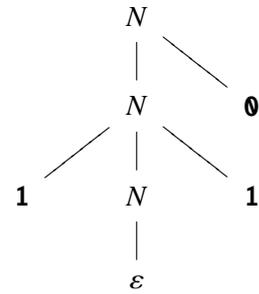
次のような BNF で表される文法を考える。

$$N \rightarrow N0 \mid 1N1 \mid \varepsilon$$

ただし、 N は非終端記号、 $0, 1$ は終端記号である。次の各記号列について、 N 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 X を記せ。(解析木は一通りとは限らないが、そのうち一つを書けば良い。)

- (1) 1 0 0 1 (2) 0 1 1 0 (3) 1 1 0 0 (4) 1 0 1 0

例: 110 に対する解析木



II. (正規表現)

以下の文字列について、「 $x(z|(yy)|(yx))^*x$ 」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 (L) 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 (R) を、両方の正規表現に全体がマッチする文字列には (B) を、どちらにも全体がマッチしない文字列には (N) をつけよ。

- (1) xyyyyzyx (2) xyzzzyyx (3) xyxzzxyyx (4) xyyyxzyyx

III. (コンパイラのフェーズ)

コンパイラは、字句(単語)を切り分ける字句解析フェーズ、プログラムの構造を木の形に表す構文解析フェーズ、変数の宣言や型のチェックを行なう意味解析(静的解析)フェーズ、目的のコードを生成するコード生成フェーズなどに概念的に分けることができる。

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。なお、(1)~(3)のいずれも単独でコンパイルされ、標準ライブラリとのみリンクされるものとする。(つまり、他のファイルに変数や関数が定義されていることはない。)

- (1) (main関数の最初のブレース「{」を忘れた。)

```
#include <stdio.h>

int main(void)
    printf("Hello World!\n");
    return 0;
}
```

- (2) (printf関数に浮動小数点数を第1引数として渡そうとした。)

```
#include <stdio.h>

int main(void) {
    printf(2.7182818);
    return 0;
}
```

- (3) (文字列の終わりを示す「\」を忘れた。)

```
#include <stdio.h>

int main(void) {
    printf("Hello! World\n");
    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、全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、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 で表された文法に対して、再帰的下降構文解析ルーチンを作成する。

$$\begin{array}{ll}
 S \rightarrow \mathbf{if} \ E \ \mathbf{then} \ S \ \mathbf{else} \ S \ \dots & \text{I} \\
 | \ \mathbf{begin} \ L \ \mathbf{end} & \dots \ \text{II} \\
 | \ \mathbf{write} \ E \ \mathbf{“;”} & \dots \ \text{III} \\
 L \rightarrow \varepsilon & \dots \ \text{IV} \\
 | \ S \ L & \dots \ \text{V} \\
 E \rightarrow F & \dots \ \text{VI} \\
 | \ E \ \mathbf{“+”} \ F & \dots \ \text{VII} \\
 F \rightarrow \mathbf{id} & \dots \ \text{VIII} \\
 | \ \mathbf{“("} \ E \ \mathbf{“)”} & \dots \ \text{IX}
 \end{array}$$

ただし、 S, L, E, F は非終端記号、**if, then, else, begin, end, write, “;”, “+”, id, “(”, “)”** は終端記号である。また、開始記号 (start symbol) は S である。… の後の I, II などは生成規則の番号である。

- (1) E の生成規則から左再帰を除去せよ。補助的に導入する非終端記号は E' とせよ。
以下の (2)~(4) は、(1) で E から左再帰を除去して得られた BNF について答えよ。
- (2) $\mathbf{First}(S)$ を求めよ。
- (3) $\mathbf{Follow}(E')$ を求めよ。
- (4) 下の構文解析表の E' の行を埋めよ。

	if	then	else	begin	end	write	;	+	id	()	\$
$S \rightarrow$												
$L \rightarrow$												
$E \rightarrow$												
$E' \rightarrow$												
$F \rightarrow$												

(4) の解答は、上記の BNF 中の生成規則の番号 (I ~ IX) (1) の解答欄中で、BNF の生成規則に自分で付けた番号、または、 \times (構文エラーの場合) から選んでもよい。(構文エラーの場合は、必ず \times を記入し、空欄のまま残さないこと。)

VI. (LR 構文解析)

次のような文法

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

に対して、演算子の優先順位、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
②	reduce I						
③		shift ②				shift ①	goto ④
④	shift ⑤	shift ②		shift ⑦	shift ⑥	shift ①	goto ⑧
⑤		shift ②	shift ⑨	shift ⑦	shift ⑥	shift ①	goto ⑧
⑥	accept						
⑦		shift ②				shift ①	goto ⑩
⑧	reduce III			shift ⑦	reduce III		goto ⑧
⑨	reduce V						
⑩	reduce II	shift ②	reduce II	shift ⑦	reduce II	shift ①	goto ⑧

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?

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

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) | <table border="1"><tr><td>① E ③ ⑥ E ⑩ * ⑦ a ①</td></tr></table> | ① E ③ ⑥ E ⑩ * ⑦ a ① | (B) | <table border="1"><tr><td>① E ③ a ①</td></tr></table> | ① E ③ a ① |
| ① E ③ ⑥ E ⑩ * ⑦ a ① | | | | | |
| ① E ③ a ① | | | | | |
| (C) | <table border="1"><tr><td>① E ③ ⑥ E ⑩ a ①</td></tr></table> | ① E ③ ⑥ E ⑩ a ① | (D) | <table border="1"><tr><td>① E ③ * ⑦ a ①</td></tr></table> | ① E ③ * ⑦ a ① |
| ① E ③ ⑥ E ⑩ a ① | | | | | |
| ① E ③ * ⑦ a ① | | | | | |
- (2) の選択肢
- | | | | | | |
|---------------------------|---|-----------------------|-----|---|---------------------------|
| (A) | <table border="1"><tr><td>① E ③ ⑥</td></tr></table> | ① E ③ ⑥ | (B) | <table border="1"><tr><td>① E ③ ⑥ E ⑩ ⑥</td></tr></table> | ① E ③ ⑥ E ⑩ ⑥ |
| ① E ③ ⑥ | | | | | |
| ① E ③ ⑥ E ⑩ ⑥ | | | | | |
| (C) | <table border="1"><tr><td>① E ③ ⑥ E ⑩ * ⑦ ⑥</td></tr></table> | ① E ③ ⑥ E ⑩ * ⑦ ⑥ | (D) | <table border="1"><tr><td>① E ③ * ⑦ ⑥ E ⑩ * ⑦ ⑥</td></tr></table> | ① E ③ * ⑦ ⑥ E ⑩ * ⑦ ⑥ |
| ① E ③ ⑥ E ⑩ * ⑦ ⑥ | | | | | |
| ① E ③ * ⑦ ⑥ E ⑩ * ⑦ ⑥ | | | | | |

コンパイラ (2012 年度) ・ 期末テスト 解答用紙 (2012 年 08 月 02 日)

学籍番号		氏名	
------	--	----	--

I. (Backus-Naur 記法)

(3×4)

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

II. (正規表現)

(3×4)

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

III. (コンパイラのフェーズ)

(4×3)

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

IV. (演算子順位法)

(4×4)

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

裏ページに続く。

