記号処理・中野・火 2 講時

2019 年 7 月 30 日

I　拡張 BNF 記法で書かれた次の文法について考える。

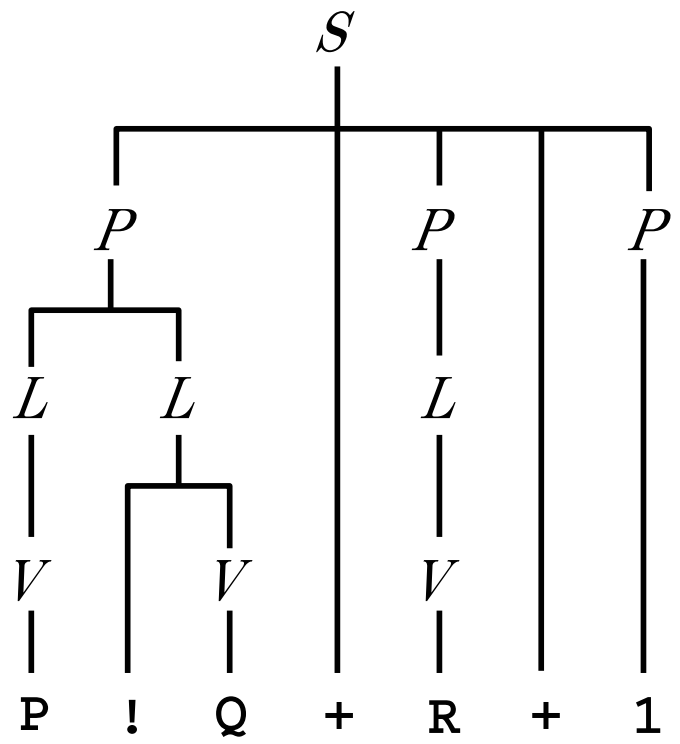
$\langle V \rangle$ ::= “P” | “Q” | “R” .　　$\langle L \rangle$ ::= [“!”] $\langle V \rangle$.

$\langle P \rangle$ ::= “1” | $\langle L \rangle$ { $\langle L \rangle$ }.　　$\langle S \rangle$ ::= “0” | { $\langle P \rangle$ “+” } $\langle P \rangle$.

(1) 次の表中の終端記号列が、非終端記号 L、P、S から導出できるかどうかについて、次の表の空欄に、導出できる場合は○を、できない場合は×をそれぞれ書き込みなさい。(12 点)

終端記号列	L	P	S
0	×	×	○
1	×	○	○
P	○	○	○
P1	×	×	×
0+1	×	×	×
P!Q!R	×	○	○
!PP+!QQ+!RR	×	×	○

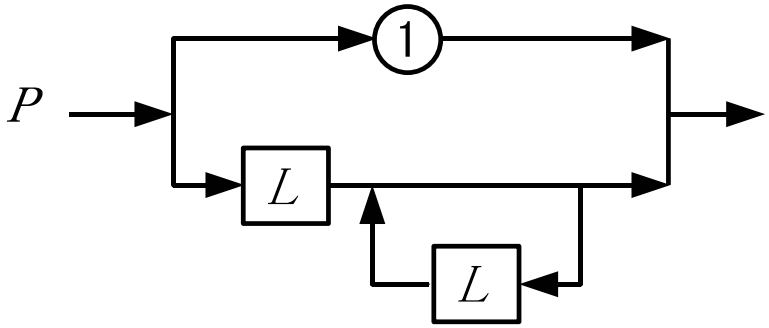
(2) 終端記号列 P!Q+R+1 の非終端記号 S に対する構文木を書きなさい。そのような構文木が存在しない場合は「構文木なし」と書きなさい。(4 点)



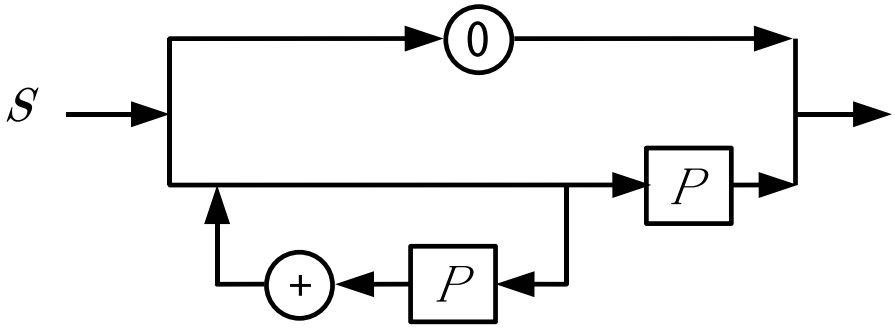
学籍番号　　氏名

(裏面に続く)

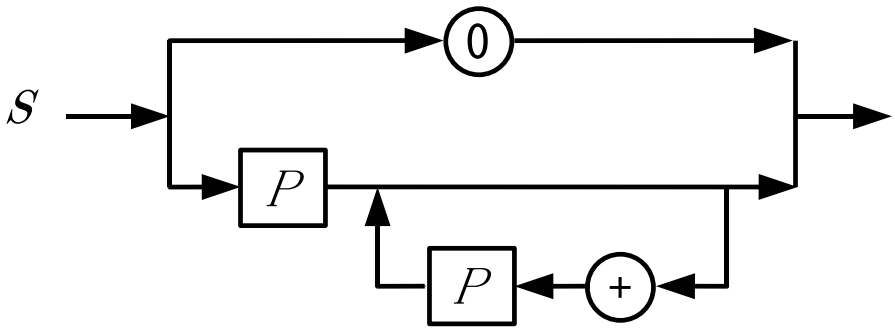
(3) 非終端記号 P の導出規則を構文図で書き表しなさい。(4 点)



(4) 非終端記号 S の導出規則を構文図で書き表しなさい。(4 点)



(5) (4) の構文図を決定的なものに書き換えなさい。(4 点)



(次ページに続く)

(6) 以下の C プログラムは、標準入力 (キーボード) から、文字列と改行文字を入力すると、入力された文字列が非終端記号 S から導出できるかどうかを判定するプログラムの一部である。関数 P、L、S の定義を補って、このプログラムを完成しなさい。(25 点)

```
#include <stdio.h>
#include <stdlib.h>

extern void V(void);
extern void P(void);
extern void L(void);
extern void S(void);

int t;

void error(void) {
    printf("Error!\n");
    exit(1);
}

void gettoken(void) {
    t = getchar();
}
```

```
void V(void) {
    switch (t) {
    case 'P': case 'Q': case 'R':
        gettoken();
        break;
    default:
        error();
    }
}

int main(void) {
    gettoken();
    S();
    if (t != '\n')
        error();
    printf("OK!\n");
    return 0;
}
```

```
void P(void)
{

    switch (t) {
    case '1':
        gettoken();
        break;
    case '!': case 'P': case 'Q': case 'R':
        L();
        while (t == '!' || t == 'P' || t == 'Q' || t == 'R')
            L();
        break;
    default:
        error();
    }









}
```

学籍番号　　氏名　　**(裏面に続く)**

(問題 I (6) の解答欄の続き)

```
void L(void)
{

     if (t == '!')
         gettoken();
     V();









}

void S(void)
{

     switch (t) {
     case '0':
         gettoken();
         break;
     case '1': case '!': case 'P': case 'Q': case 'R':
         P();
         while (t == '+') {
             gettoken();
             P();
         }
         break;
     default:
         error();
     }









}
```

(次ページに問題 II)

Ⅱ　次は Minimum C のソースプログラムと、それをコンパイルして得られた MVM の機械語プログラムである。

ソースプログラム

```
sqrt(n)
{
    int a, b;

    a = 0;
    b = n + 1;
    while ( a + 1 < b ) {
        int t;

        t = (a + b) / 2;
        if ( t * t > n )
            b = t;
        else
            a = t;
    }
    return a;
}

int x, y;

main()
{
    input x;
    input y;

    while (x <= y) {
        print sqrt(x);
        x = x + 1;
    }
}
```

コンパイル結果

```
  0  ADD R13,R14,R0
  4  LDI R11,272
  8  CALL R0, 184
 12  EXIT R1
 16  PUSH R13
 20  ADD R13,R14,R0
 24  SUBI R14,R14,8
 28  LDI R1,0
 32  ST R1,R13,-4
 36  LD R1,R13,8
 40  LDI R2,1
 44  ADD R1,R1,R2
 48  ST R1,R13,-8
 52  LD R1,R13,-4
 56  LDI R2,1
 60  ADD R1,R1,R2
 64  LD R2,R13,-8
 68  SUB R0,R1,R2
 72  JPGE R0,156
 76  SUBI R14,R14,4
 80  LD R1,R13,-4
 84  LD R2,R13,-8
 88  ADD R1,R1,R2
 92  LDI R2,2
 96  DIV R1,R1,R2
100  ST R1,R13,-12
104  LD R1,R13,-12
108  LD R2,R13,-12
112  MUL R1,R1,R2
116  LD R2,R13,8
120  SUB R0,R1,R2
124  JPLE R0,140
128  LD R1,R13,-12
132  ST R1,R13,-8
136  JP R0,148
140  LD R1,R13,-12
144  ST R1,R13,-4
148  ADDI R14,R14,4
152  JP R0,52
156  LD R1,R13,-4
160  ADD R14,R13,R0
164  POP R13
168  POP R15
172  ADDI R14,R14,8
176  POP R13
180  POP R15
184  PUSH R13
188  ADD R13,R14,R0
192  IN R1
196  ST R1,R11,0
200  IN R1
204  ST R1,R11,4
208  LD R1,R11,0
212  LD R2,R11,4
216  SUB R0,R1,R2
220  JPGT R0,264
224  LD R1,R11,0
228  PUSH R1
232  CALL R0,16
236  ADDI R14,R14,4
240  OUT R1
244  LD R1,R11,0
248  LDI R2,1
252  ADD R1,R1,R2
256  ST R1,R11,0
260  JP R0,208
264  POP R13
268  POP R15
```

学籍番号　　氏名

（裏面に続く）

(1) ソースプログラムとコンパイル結果が対応するように、空欄部分を補いなさい。(30 点)

(2) この機械語プログラムを起動して、キーボードから整数 5 と 10 を順に入力したとする。120 番地の機械語命令が初めて実行される時 (直前) の MVM のスタック中のデータ (整数値) を下図の空欄に書き込みなさい。ただし、下図の 1 つの欄は、それぞれ 32bit (4byte) のデータを表すものとし、プログラムの実行開始時には、スタックポインタ (R14) は 2^{20} ($=$ 1048576) で初期化されるものとする。スタックとして使用されていない領域は空欄のままにしておくこと。(12 点)

MVM のメモリ (スタック) の内容

(アドレス)	
	⋮
1048532	
1048536	
1048540	
1048544	3
1048548	6
1048552	0
1048556	1048568
1048560	236
1048564	5
1048568	1048576
1048572	12
1048576	

(3) (2) の時点での R1、R2、R11、R13 (フレームポインタ)、R14 (スタックポインタ) の値を下の欄に書き込みなさい。(5 点)

R1	9
R2	5
R11	272
R13	1048556
R14	1048544

(定期試験問題終り)