記号処理・中野・火 3 講時 2018 年 7 月 31 日

I　拡張 BNF 記法で書かれた次のような文法について以下の問いに答えなさい。

$\langle V \rangle$ ::= “x” | “y” | “z” .　　$\langle P \rangle$ ::= [$\langle V \rangle$ “:”] $\langle V \rangle$.
$\langle L \rangle$ ::= $\langle H \rangle$ “-” “>” $\langle V \rangle$.　　$\langle H \rangle$ ::= $\langle V \rangle$ | “(” [$\langle P \rangle$ { “,” $\langle P \rangle$ }] “)” .

(1) 次の表中の終端記号列が、非終端記号 P、L、H 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	P	L	H
x			
()			
(y)			
z:x			
(xy)			
y->z			
y:z->x			
(x:y,y,z)			

(2) 終端記号列 (x)->y の非終端記号 L 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)

この定期試験の採点結果と科目の総合成績は、8 月 7 日 (火) までに Email でお知らせします。
このお知らせが不要な受講者は右の「不要」を丸で囲ってください。

不要

(3) 非終端記号 P の構文図を書きなさい。(4 点)

(4) 非終端記号 P の構文図を決定的な構文図に書き換えなさい。(4 点)

(5) 非終端記号 H の構文図を書きなさい。(4 点)

(6) 次の 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 L 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`、`H` の定義をそれぞれ補って、このプログラムを完成しなさい。(24 点)

(次ページに続く)

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

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

int t;

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

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

```
void V(void) {
    switch (t) {
    case 'x': case 'y': case 'z':
        gettoken();
        break;
    default:
        error();
    }
}

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

```
void P(void) {









}

void L(void) {









}
```

学籍番号　　氏名　　(裏面に続く)

(問題 I (6) の解答欄の続き)

```
void H(void) {

}
```

(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

```
int b;

main()
{
    input b;
    if (                          ) {
        int n, s;

        input n;
        s = 0;
        while (n >= b) {
            s = s + log(b, n);
            n = n - 1;
        }
        print s;
    }
}

log(b, n)
{
    if (n < b)
            ;
    return log(b, n / b) + 1;
}
```

コンパイル結果

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

(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)

学籍番号　　氏名

(裏面に続く)

(2) この機械語プログラムを起動して、キーボードから、5 と 36 という数値を順に入力したとする。初めて 196 番地の機械語命令が実行された直後の MVM のスタックポインタ (`R14`) の値は 1048508 であった。この時点でのスタック中のデータ (整数値) を下図の空欄に補いなさい。ただし、下図の 1 つの欄は、それぞれ 32 bit (4 byte) のデータを表すものとする。(14 点)

MVM のメモリ (スタック) の内容

アドレス	
	⋮
1048508	
1048512	
1048516	1
1048520	5
1048524	1048540
1048528	236
1048532	
1048536	5
1048540	1048568
1048544	
1048548	
1048552	
1048556	
1048560	0
1048564	36
1048568	1048576
1048572	12

$2^{20} = 1048576$

(3) (2) の時点での `R11` と`R13` (フレームポインタ) の値を下の欄に書き込みなさい。(4 点)

`R11`	
`R13`	

(定期試験問題終り)