

1. オートマトン

正規表現において、 xy は x の後ろに y のくる記号の列を表し、 $xy|yz$ は xy または yz を表し、 $\{z\}$ は z のゼロ回以上の繰り返しを表すとする。

(a) 正規表現

$\{ab|ba\}ab$

を認識する非決定性有限オートマトン NFA を書きなさい。

(b) それを決定性有限オートマトン DFA に変換しなさい。

2. 下向き構文解析

ある言語の出力文は、

```
printf("%d %d %d %n", a, b+c, aa[i+2]);
```

のように、printf に続く () の中に、まず文字列定数がきて、次に任意個数の式がコンマで区切られて並ぶ形であり、最後にセミコロンがくるとする。最初の文字列定数が省略されることはないが、それに続く式はコンマも含めて省略できる。式は変数、整定数、括弧でくくった式、およびそれらのいくつかを乗除演算子(*, /)で結合した式、ならびに上記の式をいくつか加減演算子(+, -)で結合した式とする。変数は識別子または識別子に[]の中に式を書いた添字のついたものであるとする。

(a) 上記の出力文の構文を BNF で表現しなさい。ここで、printf と識別子、整定数、文字列定数、(), [], コンマ(,), セミコロン(;), +, -, *, / は終端記号とする。

(b) その構文規則を再帰的下向き構文解析に適するように(左再帰がない、生成規則右辺の先頭記号で一意に識別できる)書き換えるとともに、生成規則に番号付けをしなさい。はじめから再帰的下向き構文解析に適する形に書かれているなら、その旨言いなさい。

(c) 上記(b)の各非終端記号に対する First 集合と Follow 集合を求めなさい。

入力終了記号は \$ で表すとする。

(d) 上記に基づいて、再帰的下向き構文解析用の構文解析表を作りなさい。

3. コード生成

```
int a, b, x, y;
```

```
....
```

```
if (a < x) {
```

```
    if (x < b)
```

```
        y = x;
```

```
    else
```

```
        y = b;
```

```
}else
```

```
    y = a;
```

に対して、どのようなコード(機械命令列)を生成すればよいか、教科書 123 ページ(配布資料 5)の抽象マシン命令、あるいは何らかのマシン命令を使って表現しなさい。