2 月 16 日

1931026 大久保辰哉

システムソフトウェア特論 課題＃ 4

演習 4-1 の d を選択し解答した。

課題内容

例題をそのまま動かしてみよ。動いたら、次のような正規表現の認識器を作成してみよ (正規文法にまず変換し、それからオートマトンを描き、それをそのままプログラムにすること)。

$$(a + (b + |c) * d+)*$$

方針

まず、問題の正規表現を正規文法に変換、オートマトンの描画を行った。授業資料の Sam41.java をこれに対応するよう書き換えた。

成果物とその実行結果

4-1d の正規表現を正規文法に直したものは次の通りであると考えた。

$$\begin{aligned} P = \{&S \to \varepsilon, S \to aA, A \to aA, A \to bB, A \to cB, A \to dD, \\ &B \to bB, B \to cB, B \to dD, D \to dD, D \to aA, D \to \varepsilon\} \end{aligned}$$

これをオートマトンとして描画したものが図 1 の通りである。

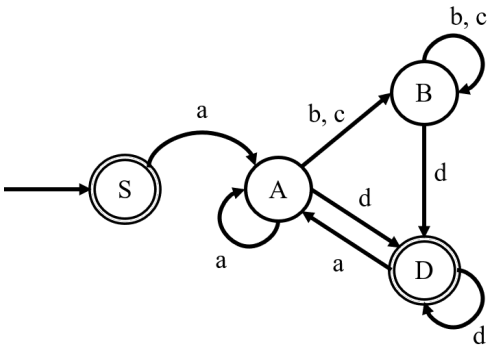

図 1: 4-1d の正規表現に対応するオートマトン

これを元に作成したプログラムは次の通りである。ただし、11 行目から 34 行目までを除く部分は授業資料の Sam41.java のものを使用した。

Listing 1: Sam41.java

```
1  import java.util.*;
2  import java.io.*;
3
4  public class Sam41 {
5    static CharToknizer tok;
6    public static void main(String[] args) throws Exception {
7      tok = new CharToknizer(args[0]);
8      int stat = 0;
9      while(true) {
10       switch(stat) {
11           //S
12       case 0:
13           if(tok.chkfwd("a")) { stat = 1; continue; }
14           if(tok.chkfwd("$")) { System.out.println("accept."); return; }
15           break;
16           //A
17       case 1:
18           if(tok.chkfwd("a")) { stat = 1; continue; }
19           if(tok.chkfwd("b")) { stat = 2; continue; }
20           if(tok.chkfwd("c")) { stat = 2; continue; }
21           if(tok.chkfwd("d")) { stat = 3; continue; }
22           break;
23           //B
24       case 2:
25           if(tok.chkfwd("b")) { stat = 2; continue; }
26           if(tok.chkfwd("c")) { stat = 2; continue; }
27           if(tok.chkfwd("d")) { stat = 3; continue; }
28           break;
29           //D
30       case 3:
31           if(tok.chkfwd("d")) {stat = 3; continue;}
32           if(tok.chkfwd("a")) {stat = 1; continue;}
33           if(tok.chkfwd("$")) { System.out.println("accept."); return; }
34           break;
35       }
36       System.out.println("error."); return;
37     }
38   }
39 }
40 class CharToknizer {
41   String str, tok;
42   int pos = -1;
43   boolean eof = false;
44   public CharToknizer(String s) { str = s; fwd(); }
45   public boolean isEof() { return eof; }
46   public String curTok() { return tok; }
47   public boolean chk(String s) { return tok.equals(s); }
48   public void fwd() {
49     if(eof) { return; }
50     pos += 1;
51     if(pos >= str.length()) { eof = true; tok = "$"; return; }
52     tok = str.substring(pos, pos+1);
53   }
54   public boolean chkfwd(String s) {
55     if(chk(s)) { fwd(); return true; } else { return false; }
56   }
57 }
```

いくつかのパターンで実行した。その結果は次の通りであった。

実行結果

```
java␣Sam41␣␣␣abcbbcbccd
accept.
java␣Sam41␣␣␣$
accept.
java␣Sam41␣␣␣addad
accept.
java␣Sam41␣␣␣addbad
error.
java␣Sam41␣␣␣bcd
error.
java␣Sam41␣␣␣acabcd
error.
java␣Sam41␣␣␣aaa
error.
java␣Sam41␣␣␣acb
error.
```

考察

実行結果より、図 1 の状態 S, D のどちらも最終状態として認識され受理されたことが確認できる。また、状態 D で b、S で b、B で a のように遷移できないパターンや、最終状態に到達しないパターンを入力すると拒否されたことが確認できた。

また、図 1 のオートマトンを見ると、文字 b と c の違いによる状態遷移の差が生じないと考えられる。つまり、元の正規表現以外にも $(a + (b|c) * d+)*$ や $(a + (b|c+) * d+)*$ などの正規表現に対応した認識機も全く同じプログラムで動作すると考えられる。

アンケート

Q1. 形式言語についてどのくらい知っていましたか。また、どの部分にとくに興味がありますか。

形式言語は学部の授業で少し触れ、正規文法からオートマトンを描くことは以前から出来ました。ただ、「正規表現」、「正規文法」といったような用語とその意味は忘れていました。

Q2. オートマトンによる認識器や CYK による認識器についてどう思いましたか。

図で表現できるため、オートマトンの認識器は分かりやすいという印象をうけました。しかし、CYK の方が CNF に変形が必要とはいえ、汎用性がありそうと思いました。

Q3. リフレクション (課題をやってみて気付いたこと)、感想、要望など。

今回は 4-1 の d を選択しましたが、もし c や b も選択した場合、c は今回のものを少し変えれば実現できそうであることに対して、b は状態数を 1 つ増やす必要がありそうと予想しました。正規表現が短く、より単純に見えても、状態や状態遷移の種類が増えることがありそうと感じました。