

□演習問題 A6 (締め切り=2009/05/25 演習中の回収時)

問題番号: A6

課題名：DFA の最小化と字句解析プログラム

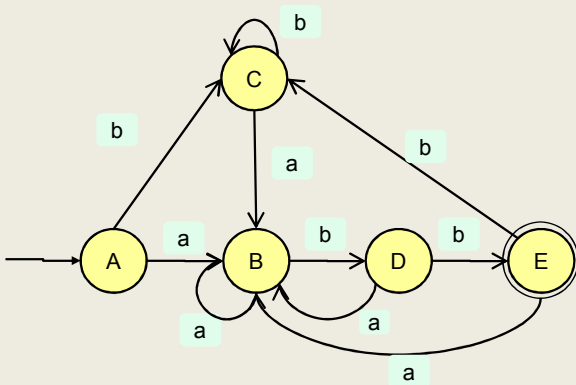
(1) 次の DFA の状態数を最小化せよ。(正規表現は $(ab)^*abb$)

(A) 最小化した決定性有限オートマトンの図を示せ。

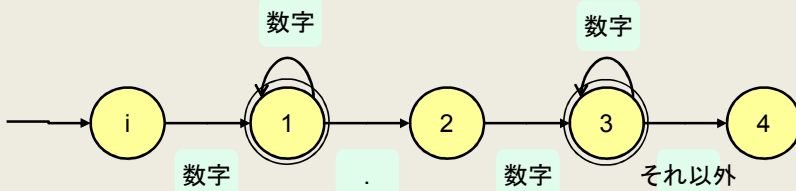
(1) 状態遷移表を示せ。

(2) 同一視できる状態を示せ。($\{\{i,1,4\}, \{f1,f2\}\}$ のように)

(B) いくつかの文字列がちゃんと受理されるか、あるいは受理されないかを確認せよ、示せ。



(2) 次の状態遷移機械を考える。



(A) これは「正の数」を受理するものであるが、より正確にどのような言語を受理するかを、例を用いながら説明せよ。

(B) この言語を受理する字句解析プログラムを作成せよ。(例題 1 を参考にせよ)

(3) 入力された文字列を字句解析して、次の種類のトークンに分割するプログラムを作成せよ。例題 2 を参考にせよ。

- 演算子+, 演算子-, 演算子*, 演算子/
- 名前 (識別子),
- 正の数 (問題 (2) のもの),

なお各トークンの間には空白（スペース）があってもよいこととする。ただしその空白はトークンとはみなさない。すなわち"200+abc"と " 200 + abc"はどちらも、[正数、演算子+、識別子]という3つのトークンの並びとなる。

問題（2）、（3）のための準備

[参考文献] 中田育男：コンパイラの構成と最適化、朝倉書店

□cis.k.hosei.ac.jp/~asasaki/lect/compiler/2008A/2008-0519.zip から例題プログラムをダウンロードすること。

□例題 0. 入力された文字列を一文字ずつとってくるメソッド `nextChar()` を作成せよ。ただし改行は、`'\n'` を返すこととし、EOF（入力の最後の場合は）0 を返すこととする。

→プログラム `CharReader.java` (参考文献 p37 図 4. 1 を改変)

□java のラベル付き制御構造：制御構造に名前をつけることができる。下記で `loop:` は外側の `while` につけたラベルである。`break loop` などで、指定した制御構造からの脱出を指定できる。（`break` のみだと、内側の `while` を抜けるだけ）

`loop:`

```
while(...){
    while(...){
        ...
        break loop
    }
}
```

□例題 1. 入力された文字列を字句解析して、名前（識別子）を読み取るプログラムを作成せよ。ここでは、文字列の先頭から識別子となっている場合は、それを表示し、文字列の先頭が識別子以外であればエラーとなるようにする。

入力 `a123.adf` → 表示：`a123`

入力 `a adf` → 表示：`a`

入力 `.adf` → エラー：文字列の先頭「`.`」は識別子には含まれない

入力 `123abc` → エラー：「`1`」は識別子には含まれない

→プログラム ReadIdentifierA.java, ReadIdentifierB.java(参考文献 p40,41 図 4.5, 4.7 を改変)

enum CharClassA → 文字の種類
 enum TokenClassA → トークンの種類
 ReadIdentifierA クラス
フィールド
CharReader reader → 例題0のもの。nextChar()によって入力から1文字とってくる。
StringBuilder stringBuffer → トークンの文字列(レクシム)の作成用
(char ch → 直前に入力から読み込んだ文字)
メソッド
CharClassA getCharClass(Char c) : 文字の種類を得る
String currentString() → 文字列を得る
TokenClassA TnextToken() : 次の1トークンを得る
main メソッド : 先頭から一つの識別子を得るテストルーチン

なお、ReadIdentifierB クラスは一文字先読みをしておく版

例題 2. 入力された文字列を字句解析して、次の種類のトークンに分割するプログラムを作成せよ。

- 区切り文字「;」 (TokenClass.DELIMITER)
- 正の整数 (TokenClass.NUM)
- 名前(識別子) (TokenClass.IDENT)

なお各トークンの間には空白(スペース)や改行があってもよいこととする。ただしそれらはトークンとはみなさない。すなわち abc;cde と abc ;cde はどちらも、[識別子、区切り文字「;」、識別子]という3つのトークンの並びとなる。

→プログラム TestTokenizer.java

参考文献 p44.45 図 4. 10 を改変