

Softwaretechnik und Programmiersprachen I - Sommersemester 2011

Übungsblatt 10

Besprechung der Aufgaben in 25.12.02.55 am Mittwoch 14:30-16:00 Uhr oder Freitag 10:30-12:00 Uhr
Bei Fragen wenden Sie sich bitte an John Witulski: John.Witulski@uni-duesseldorf.de.

Aufgabe 10.1

1. Geben Sie jeweils einen regulären Ausdruck an und konstruieren Sie aus diesen einen nichtdeterministischen endlichen Automaten (NFA) über das Alphabet $\{a, b, c\}$ mit folgenden Eigenschaften:
 - (a) beginnt und endet mit a (mindestens Länge 2).
 - (b) es gibt maximal ein a .
 - (c) es gibt mindestens zwei a 's.
 - (d) die Anzahl von den a 's und b 's sind gleich.
2. Erzeugen Sie aus den NFA in (a) einen DFA.
3. Gibt es reguläre Ausdrücke, für die das nicht möglich ist?

Aufgabe 10.2

Konstruieren Sie einen NFA, der alle Binärzahlen akzeptiert, die durch 3 teilbar sind.

Aufgabe 10.3

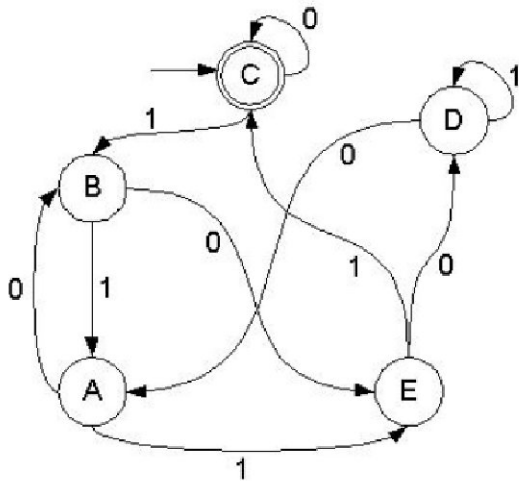
Beschreiben Sie die Sprachen, die mit den folgenden regulären Ausdrücke bezeichnet werden:

- (a) $a(a | b)^*a$
- (b) $((\epsilon | a)b^*)^*$
- (c) $(a | b)^*a(a | b)(a | b)$
- (d) $a^*ba^*ba^*ba^*$

Aufgabe 10.4

Lexen Sie von Hand folgendes Programm, die entsprechenden Tokenklassen können Sie selber festlegen:

```
int a(n) {return (n+1);}
int x=2;
void b(){x=a; printf("%d\n",x);}
void c(){int x=1;printf("%d\n",a);}
int main(){b();c();}
```



Aufgabe 10.5

Implementieren Sie in Java den oben angegebenen Automaten, der eine Binärzahl akzeptiert, die durch 5 teilbar ist.

1. Implementieren Sie dazu eine Klasse Node, die einen Zustand des Automaten repräsentiert und das folgende Interface implementiert:

```

public interface INode {
    public abstract void addTransition(Node dest, char c);
    public abstract INode getNext(char c);
    public abstract String getName();
    public abstract boolean isFinal();
}

```

Beachten Sie, dass die Methode getNext(char c) null zurückliefern soll genau dann, wenn es keine Transition für c gibt.

2. Schreiben Sie eine Hauptklasse DFA1, in der der Automat kodiert ist. Die Kodierung des Automaten soll der Einfachheit halber in dem Konstruktor der Klasse erfolgen. Der Konstruktor der Klasse wird ohne Parameter aufgerufen. Die Klasse soll eine Methode accept(String word) haben. In dieser Methode werden der Reihe nach die besuchten Knoten in ausgegeben und anschließend entweder ablehnen oder akzeptieren. Die gesamte Ausgabe muss in einer Zeile stehen und zwischen der Knotenliste und dem Ergebnis muss ein Leerzeichen sein. Auf die Eingabe „10010“ soll die gleiche Ausgabe erfolgen wie durch das Kommando:

```
System.out.println("CBEDDA ablehnen")
```

Auf die Eingabe „11001“ soll die gleiche Ausgabe erfolgen wie durch das Kommando:

```
System.out.println("CBABEC akzeptieren")
```