

## Hausaufgaben 13

**12./13.06.2023**

Abgabe der Lösung am 18.06.2023

### Aufgabe 1:

#### **Aufstellen eines nichtdeterministischen endlichen Automaten**

Schreiben Sie eine Klasse NFA (für „nondeterministic finite automaton“), die überprüft, ob ein String in der durch einen regulären Ausdruck beschriebenen Menge liegt. Die Klasse erhält den regulären Ausdruck als Kantenliste eines Graphen, der zur Auswertung mit NFA geeignet ist. Dieser Graph muss *nicht* selbst aus einem regulären Ausdruck ermittelt werden, sondern man erhält ihn im Konstruktor. Der Graph wird im Konstruktor als Kantenliste übergeben (siehe unten) und soll intern in Form einer Adjazenzliste (etwa wie im Skript SS 23 Folie 549) gespeichert werden. Anders als im Skript gibt es hier der Einfachheit halber keine  $\epsilon$ -Übergänge.

Die Details seien am folgenden Beispiel erläutert:

Im Konstruktor

```
public NFA(String x)
```

erhält die Klasse die Kantenliste

```
3,3,1,2,a,1,3,a,2,2,a,2,2,b,2,3,a
```

Die Liste hat mehrere Besonderheiten

- Der zweite Wert ist nicht die Anzahl der Kanten, sondern die Nummer des Zielknotens.
- Startknoten ist immer Knoten 1.
- Für jede Kante stehen in der Kantenliste außer dem Anfangs- und dem Endknoten der Kante noch das zugehörige Zeichen. Im Beispiel geht also die erste Kante von Knoten 1 zu Knoten 2 mit dem Zeichen a.
- Gehen Sie davon aus, dass der Graph keine  $\epsilon$ -Übergänge enthält.

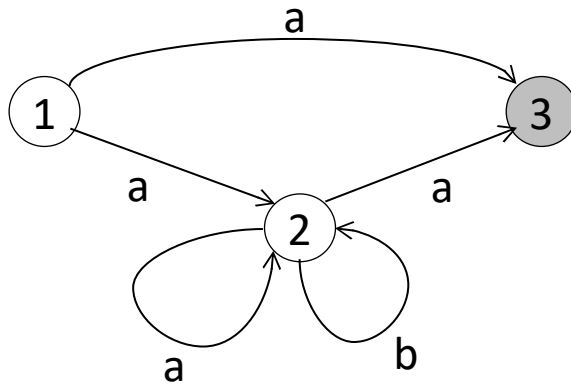
Schreiben Sie den Konstruktor sowie eine Methode

```
public boolean testString(String s)
```

die testet, ob der String s dem regulären Ausdruck entspricht, und das Ergebnis zurückgibt.

Beispiel:

Die obengenannte Kantenliste entspricht dem Graphen



bzw. dem regulären Ausdruck

$a|a(a|b)^*a$

Diesen Ausdruck erfüllen die Strings „abba“ und „a“, aber nicht „ab“.

```
NFA nfa_test = new NFA("3,3,1,2,a,1,3,a,2,2,a,2,2,b,2,3,a");
System.out.println(nfa_test.testString("abba"));           //true
System.out.println(nfa_test.testString("a"));             //true
System.out.println(nfa_test.testString("ab"));           //false
```

Hinweis:

Lassen Sie sich in der testString-Methode nach jedem Buchstaben des Teststrings den Status der Knoten (besetzt/unbesetzt) ausgeben und überprüfen Sie die Zwischenergebnisse.