RWTH Aachen **SS 2017**
Lehrgebiet Theoretische Informatik Blatt 4
Rossmanith–Dreier–Hark–Kuinke 22.5.2017

Übung zur Vorlesung Formale Sprachen, Automaten und Prozesse

Aufgabe T11

1. Betrachten Sie $L = \{ a^n b^n \mid n \geq 0 \}$. Gelten $a \equiv_L aa$, $b \equiv_L bb$ oder $ab \equiv_L ba$?
2. Es sei nun $L = (ab)^*$. Gelten $a \equiv_L b$, $aa \equiv_L a$ oder $\epsilon \equiv_L ab$? Wie lauten in diesem Fall die Äquivalenzklassen von $\equiv_L$?

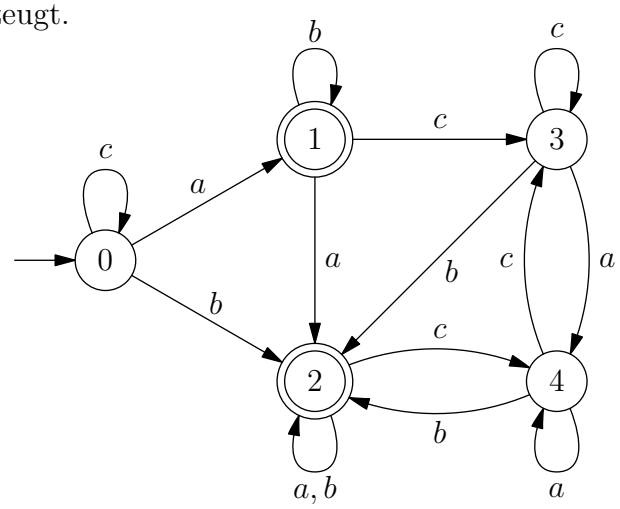
Aufgabe T12

Zeigen Sie mit dem Satz von Myhill–Nerode, daß folgende Sprachen nicht regulär sind:

1. $L = \{ ww \mid w \in \{a, b\}^* \}$
2. $L = \{ a^n b^m \mid |n - m| < 5 \}$

Aufgabe T13

Minimieren Sie folgenden DFA und finden Sie anschließend einen regulären Ausdruck, der seine Sprache erzeugt.



Aufgabe T14

Richtig oder falsch?

a) Es gibt eine reguläre Sprache, die von einem NFA mit 10 Zuständen akzeptiert wird, aber von keinem DFA mit 100 Zuständen.

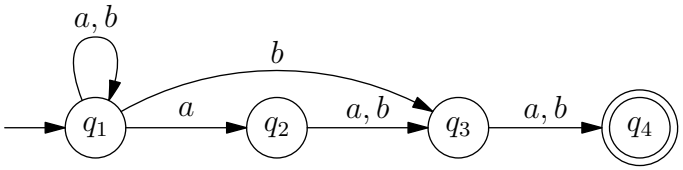
b) Es gibt eine reguläre Sprache, die von einem NFA mit 10 Zuständen akzeptiert wird, aber von keinem DFA mit 1217 Zuständen.

c) Zwei minimale DFAs, welche jeweils komplementäre Sprachen akzeptieren, haben gleich viele Zustände.

d) Zwei minimale NFAs, welche jeweils komplementäre Sprachen akzeptieren, haben gleich viele Zustände.

Aufgabe H7 (5+5+5 Punkte)

a) Bilden Sie den Potenzautomaten von nebenstehendem NFA M.



b) Minimieren Sie ihn mit dem Markierungsalgorithmus.

c) Geben Sie möglichst viele Wörter $w_1, \ldots, w_n$ an, so daß $w_i \not\equiv_L w_j$ für $i \neq j$. Wie haben Sie die Wörter gefunden und warum erfüllen Sie diese Bedingung?

Aufgabe H8 (15 Punkte)

Programmieren Sie einfache NFAs in einer objektorientierten, gängigen Programmiersprache, am besten in Java. Erstellen Sie hierzu eine generische Klasse $NFA\langle S, A\rangle$, welche einen geeigneten Konstruktor und folgende Methoden besitzt:

1. $addTransition(S\ q, A\ a, S\ p)$ fügt eine Transition vom Zustand q zum Zustand p mit dem Symbol a hinzu.

2. $Set\langle S\rangle\ simulate(S\ q, List\langle A\rangle\ w)$ simuliert den NFA vom Zustand q beginnend und das Wort w lesend. Sie berechnet dabei $\hat{\delta}(q, w)$.

Die Parameter A und S stehen hierbei für den Typ des Alphabets und der Zustände. Natürlich sollte Ihr Programm gut lesbar und einfach zu verstehen sein, so daß es auch von anderen Personen gewartet und angepaßt werden könnte.
Folgendes Programmstück würde einen NFA konstruieren und dann simulieren:

```
import java.util.*;
public class MyNFA {
  static public void main(String args[]) {
    Set<Integer> states = new HashSet<Integer>();
    states.add(1); states.add(2); states.add(3); states.add(4);
    NFA<Integer, String> M = new NFA<Integer, String>(states);
    M.addTransition(1, "rechts", 2); M.addTransition(1, "runter", 3);
    M.addTransition(2, "links", 1); M.addTransition(2, "runter", 4);
    M.addTransition(3, "rechts", 4); M.addTransition(3, "hoch", 1);
    M.addTransition(4, "links", 3); M.addTransition(4, "hoch", 2);
    List<String> eingabe = new LinkedList<String>();
    eingabe.add("rechts"); eingabe.add("runter"); eingabe.add("links");
    Set<Integer> reachable = M.simulate(1, eingabe);
    System.out.println(reachable);
  }
}
```

Verwenden Sie Ihr Programm, um $\hat{\delta}(7, ababbbaa)$ für den NFA zu berechnen, dessen Transitionen Sie auf der Webseite neben dem Aufgabenblatt finden. Er hat die Zustandsmenge $\{1, \ldots, 34\}$ und das Eingabealphabet $\{a, b\}$. Was ist das Ergebnis?
Ihr Programm sollte halbwegs effizient sein, insbesondere im Spezialfall eines deterministischen Automatens. Achten Sie darauf, den Code möglichst modular zu gestalten.

Aufgabe H9 (5 Punkte)

Sei L die Sprache aller Palindrome über dem Alphabet $\{😀, 🤡, 🐺, 💩\}$. Zeigen Sie mithilfe des Satzes von Myhill–Nerode, daß diese Sprache nicht regulär ist.