

Diskrete Modellierung

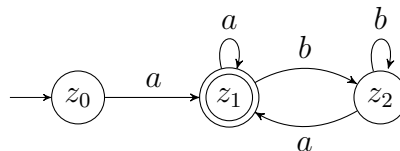
Wintersemester 2008/2009

Übungsblatt 12

Abgabe: bis 4. Februar 2009, 8.15 Uhr vor der Vorlesung
(oder bis 4. Februar 2009, 8.15 Uhr in Raum 113, Robert-Mayer-Str. 11-15)

Aufgabe 1: (5 + 6 + 9 = 20 Punkte)

Sei A der folgende endliche Automat über dem Alphabet $\Sigma = \{a, b\}$:



- (a) Geben Sie die Menge der Zustände, den Startzustand, die Menge der akzeptierenden Zustände und die Übergangsfunktion von A an. Ist A deterministisch oder nicht-deterministisch?
- (b) Welche der folgenden Wörter werden von A akzeptiert, welche nicht? Begründen Sie Ihre Antworten.
- babaa
 - abbaababa
 - aabbab
- (c) Beschreiben Sie umgangssprachlich, welche Sprache $L(A)$ von A akzeptiert wird.

Aufgabe 2: (6 + 8 + 8 + 8 = 30 Punkte)

Geben Sie für jede der folgenden Sprachen einen möglichst einfachen NFA und einen DFA an (beide in graphischer Darstellung), die die Sprache akzeptieren.

- (a) $L_1 := \{w \in \{0, 1, \dots, 9\}^+ : w \text{ beginnt nicht mit } 0\}$
- (b) $L_2 := \{w \in \{a, b\}^* : |w| \text{ ist gerade}\}$
- (c) $L_3 := \{w \in \{a, b\}^* : w \text{ enthält mindestens drei } a\}$
- (d) $L_4 := \{w \in \{a, b\}^* : w \text{ enthält das Teilwort } aba, \text{ d.h. ex. } x, y \in \{a, b\}^* \text{ s.d. } w = xabay\}$

Aufgabe 3: (25 Punkte)

Beweisen Sie, dass die Sprache

$$L := \{a^n b^m : n, m \in \mathbb{N}, n \leq m\}$$

nicht regulär ist.

Aufgabe 4:**(25 Punkte)**

Wir betrachten ein System bestehend aus zwei Prozessen P_1, P_2 , die mit einer gemeinsamen Datenbank arbeiten. Die Prozesse dürfen die Datenbank gleichzeitig lesen. Damit Inkonsistenzen vermieden werden, dürfen die beiden Prozesse aber nicht gleichzeitig in die Datenbank *schreiben*. Bevor einer der Prozesse in die Datenbank schreiben kann, muss er sich die Schreibrechte für die Datenbank reservieren. Solange ein Prozess die Schreibrechte besitzt, darf der jeweils andere Prozess nicht in die Datenbank schreiben. Nachdem alle Schreibvorgänge abgeschlossen sind, kann der Prozess die Schreibrechte wieder abgeben und gegebenenfalls später wieder beantragen. Zu Beginn besitzt keiner der beiden Prozesse die Schreibrechte.

Modellieren Sie das System durch einen DFA, der als Eingabe ein Wort über dem Alphabet

$$\Sigma := \{lock_1, lock_2, release_1, release_2, write_1, write_2, other_1, other_2\}$$

erhält. Die Symbole bedeuten hierbei Folgendes:

- $lock_i$: P_i erhält die Schreibrechte.
- $release_i$: P_i gibt die Schreibrechte wieder ab.
- $write_i$: P_i führt eine Schreiboperation aus.
- $other_i$: P_i führt eine Nicht-Schreiboperation aus.

Zum Beispiel besagt das Wort

$$w := lock_1 write_1 other_1 other_2 write_1 release_1 other_1 lock_2 write_2 release_2 lock_1 write_1 release_1,$$

dass

- (1) P_1 die Schreibrechte erhält und bis zur Abgabe der Schreibrechte nacheinander Folgendes passiert: P_1 schreibt in die Datenbank, P_1 führt eine Nicht-Schreiboperation aus, P_2 führt eine Nicht-Schreiboperation aus, P_1 schreibt in die Datenbank.
- (2) P_1 eine Nicht-Schreiboperation ausführt.
- (3) P_2 die Schreibrechte erhält, in die Datenbank schreibt und die Schreibrechte wieder abgibt.
- (4) P_1 die Schreibrechte erhält, in die Datenbank schreibt und die Schreibrechte wieder abgibt.

Der DFA soll alle Wörter über dem Alphabet Σ akzeptieren, die gültige Aktionsfolgen in dem oben beschriebenen System sind. Z.B. sollen das oben definierte Wort w sowie die Wörter

$$lock_1 write_1 release_1 lock_2 release_2 \quad \text{und} \quad lock_1 write_1 other_2$$

akzeptiert werden; die Wörter

$$lock_1 write_1 write_2 release_1 \quad \text{und} \quad lock_1 write_1 lock_2 release_1$$

sollen aber nicht akzeptiert werden.