

# **Einführung in die Theoretische Informatik**

## **Tutorium – Woche 5**

**Esther Ney**

Sommersemester 2025

## Zulip

Für die diesjährige THEO-Vorlesung wurden **inoffizielle** Streams auf der ZULIP-Instanz der TUM INFORMATIK erstellt. Dieser wird von den meisten Tutor\*innen, aber nicht von PROF. ALBERS gelesen; wir versuchen natürlich trotzdem, organisatorische Fragen bei Bedarf weiterzuleiten. Man kann mich dort auch in Direktnachrichten anschreiben.

- Tutorin: ESTHER NEY
- Tutorien: Dienstag 16-18 (Di-16-2) und Donnerstag 16-18 (Do-16-1)
- E-Mail: [esther.ney@tum.de](mailto:esther.ney@tum.de)
- Aufbau einer Tutorstunde: kurze Wiederholung wichtiger Definitionen und Lemmata, dann selbständiges Bearbeiten der Aufgaben
- Folien: auf ZULIP oder <https://reflect.moe/uni/theo25/>

## Hinweis

Diese Folien sind keine offizielle Musterlösung und dienen lediglich als zusätzliches Material für die Nachbereitung der Tutorien.

Tutorienlösungen können fehlerhaft sein.

Im Zweifel gilt immer das, was in den offiziellen Vorlesungsfolien und in der Musterlösung steht.

## Urheberrechtlicher Hinweis

Die Folien sind nur für die Teilnehmer der entsprechenden Tutorien gedacht und dürfen nicht ohne die ausdrückliche Erlaubnis der Urheber vervielfältigt oder weitergegeben werden. Das Urheberrecht zu den Aufgaben liegt bei den Aufgabenerstellern.

## Definition 3.56 – Residualsprache

Die Residualsprache  $L^w$  zu einer Sprache  $L$  und einem Wort  $w \in \Sigma^*$  ist definiert als:

$$L^w := \{v \in \Sigma^* \mid wv \in L\} \quad (1)$$

Residualsprachen können gleich sein, etwa gilt für  $L' := L((a \mid b)a^*)$ :

$$L'^\varepsilon = L', \quad L'^a = L'^b = L(a^*), \quad L'^{bb} = \emptyset \quad (2)$$

Wir sagen, dass  $v \equiv_L w$ , wenn  $L^v = L^w$  gilt. Dies ist eine Äquivalenzrelation.

## Satz 3.60 – Regularitätskriterium / Satz von MYHILL und NERODE

Eine Sprache  $L$  ist genau dann regulär, wenn sie endlich viele Residualsprachen besitzt.

Intuitiv: Residualsprachen sind ein sehr mächtiges Werkzeug! Wir können es verwenden, um zu zeigen, dass eine Sprache regulär ist.

## Definition 3.53 – Äquivalenz von Zuständen eines DFAS

Wir sagen, dass Zustände  $q$  und  $p$  äquivalent sind (also  $q \equiv_M p$ ), wenn sie die gleiche Sprache erkennen. Die Sprache eines Zustandes  $q$  ist  $\{w \in \Sigma^* \mid \hat{\delta}(q, w) \in F\}$ .

## Satz 3.57 – Quotientenautomat ist ein minimaler DFA

Für  $M := (Q, \Sigma, \delta, \{q_0\}, F)$  ist  $M/\equiv$  mit  $M/\equiv := (Q/\equiv, \Sigma, \delta/\equiv, [q_0]_{\equiv}, F/\equiv)$  ein DFA mit  $L(M/\equiv) = L(M)$ .  $M/\equiv$  ist minimal, hat also weniger Zustände als alle anderen DFAS, die die gleiche Sprache erkennen.

Der Minimierungsalgorithmus aus der Vorlesung läuft in  $O(n^2)$ , der beste bekannte Algorithmus von HOPCROFT läuft in  $O(n \log n)$ .

Mit folgendem Algorithmus bestimmen wir äquivalente Zustände:

- (a) Definiere  $\mathcal{W} := \{\{p, q\} \mid p \notin F, q \in F\}$ .
- (b) Solange möglich, füge  $\{p, q\}$  mit  $\{\delta(p, a), \delta(q, a)\} \in \mathcal{W}$  für ein  $a \in \Sigma$  zu  $\mathcal{W}$  hinzu.
- (c) Am Ende sind  $p$  und  $q$  äquivalent, wenn  $\{p, q\} \notin \mathcal{W}$  gilt.

## Kanonischer Minimalautomat

Wo ist der Zusammenhang zwischen den beiden Konzepten?

### Definition 3.58 – kanonischer Minimalautomat

Der DFA  $M_L := (\mathcal{R}_L, \Sigma, \delta_L, L, F_L)$  mit  $\mathcal{R}_L$  die (endliche, Satz 3.60) Menge der Residualsprachen von  $L$ ,  $\delta_L(L^w, a) := L^{wa}$  und  $F_L := \{L' \in \mathcal{R}_L \mid \epsilon \in L'\}$  ist der kanonische Minimalautomat zu  $L$ .

$M_L$  unterscheidet sich von anderen minimalen DFA zu  $L$  nur durch eine Umbenennung der Zustände.

Hinweis: NFAS sind ebenfalls mit dem Algorithmus aus der Vorlesung minimierbar. Allerdings können unterschiedliche NFAS zu der selben Sprache auch zu unterschiedlichen NFAS minimiert werden, die auch noch unterschiedlich viele Zustände haben können.

Häufig werden die Residualsprachen in  $\mathcal{R}_L$  durch einen regulären Ausdruck angegeben.

## Satz 3.35 – Abschlusseigenschaften regulärer Sprachen

Seien  $R, R_1$  und  $R_2$  reguläre Sprachen. Dann sind auch

$$R_1 R_2, R_1 \cup R_2, R_1 \cap R_2, R^*, R_1 \setminus R_2, \bar{R} \quad (3)$$

mit  $\bar{R} := \Sigma^* \setminus R$  reguläre Sprachen.

## Definition 3.42 – Entscheidungsprobleme

Folgende Probleme sind für einen DFA  $D$  oder NFA  $N$  entscheidbar:

- Wortproblem – Ist  $w \in L(D)$  bzw.  $w \in L(N)$ ?
- Leerheitsproblem – Ist  $L(D) = \emptyset$  bzw.  $L(N) = \emptyset$ ?
- Endlichkeitsproblem – Ist  $|L(D)| < \infty$  bzw.  $|L(N)| < \infty$ ?
- Äquivalenzproblem – Ist  $L(D_1) = L(D_2)$  bzw.  $L(N_1) = L(N_2)$ ?

Für RES und TYP 3-Grammatiken sind diese Probleme durch Konstruktion eines passenden NFA ebenfalls entscheidbar.

## Vorbereitung

Überprüfen Sie, dass Sie mit den folgenden Begriffen vertraut sind.

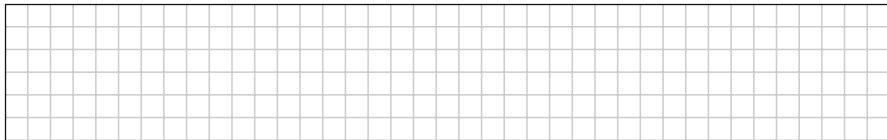
- Abschlusseigenschaften
- Entscheidungsprobleme
- Automatenminimierung
- Quotientenautomat
- Residualsprache
- $\equiv_M, \equiv_L$
- kanonischer Minimalautomat

## Aufgabe T5.1

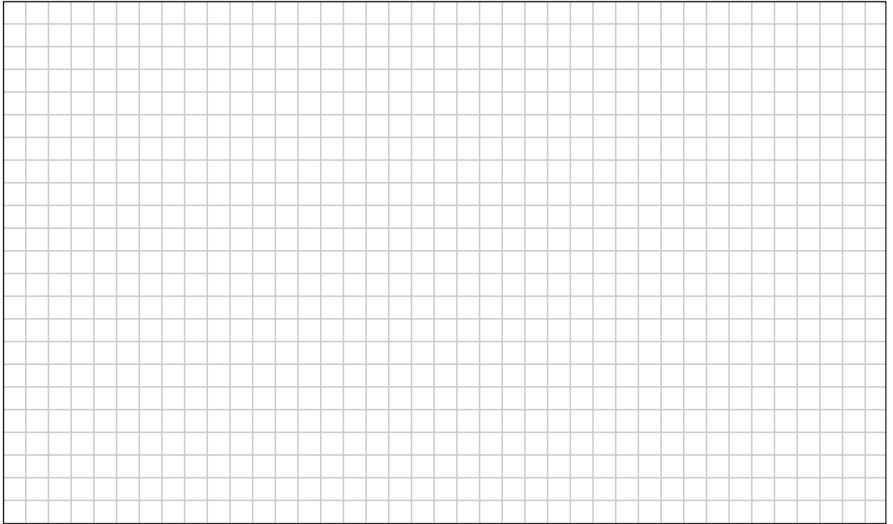
Bestimmen Sie für jede der folgenden Aussagen, ob sie wahr oder falsch ist.

Falls die Aussage wahr ist, geben Sie eine kurze Begründung an. Falls die Aussage falsch ist, geben Sie ein Gegenbeispiel an und erklären Sie kurz, warum das Gegenbeispiel die Aussage widerlegt.

- (a) Seien  $L_1, L_2 \subseteq \Sigma^*$  beliebige nichtleere Sprachen. Zeigen oder widerlegen Sie: Wenn  $L_1 L_2$  regulär ist, dann sind auch  $L_1$  und  $L_2$  regulär.
- (b) Sei  $L \subseteq \Sigma^*$  regulär und  $a \in \Sigma$ . Zeigen oder widerlegen Sie: Die Sprache aller Wörter aus  $L$ , die nicht mit  $a$  enden, ist regulär.
- (c) Zeigen oder widerlegen Sie: Es gibt einen minimalen DFA  $M$  mit maximal 4 Zuständen, sodass die minimale Pumping-Lemma-Zahl für  $L(M)$  kleiner als die Anzahl der Zustände von  $M$  ist.



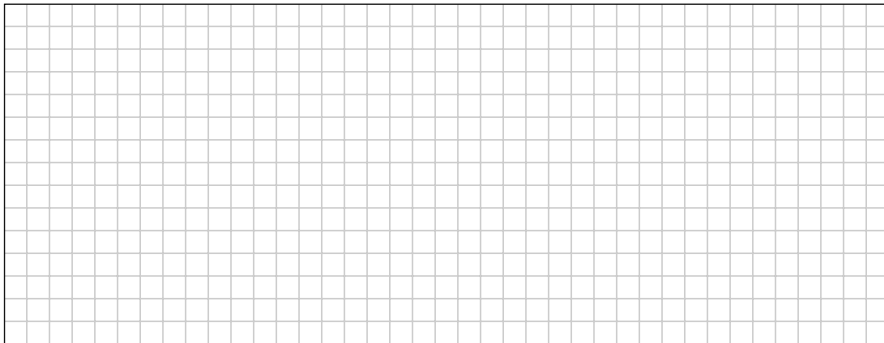
## Aufgabe T5.1



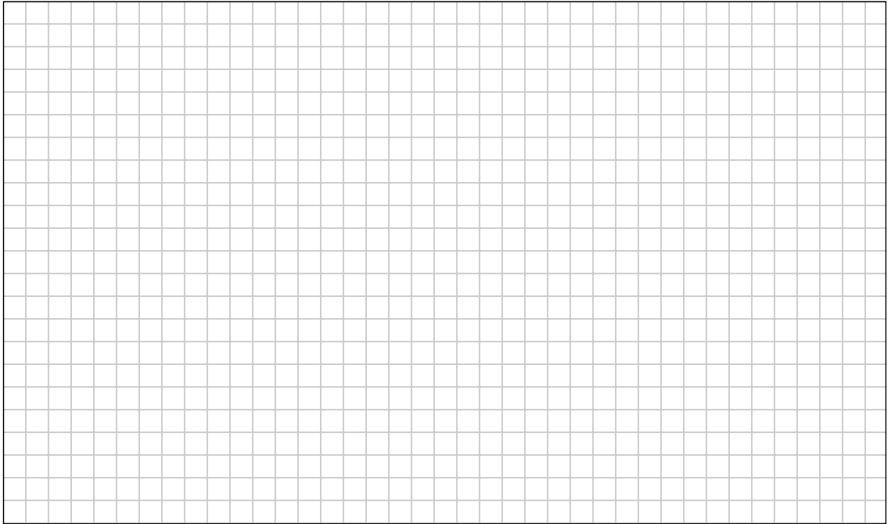
## Aufgabe T5.2

Skizzieren Sie ein Entscheidungsverfahren, welches für einen gegebenen DFA  $M := (Q, \Sigma, \delta, q_0, F)$  feststellt, ob  $|L(M)| \leq 100$  gilt, also ob in der Sprache von  $M$  höchstens 100 Wörter sind.

Hinweis: Der Algorithmus darf auch exponentielle Laufzeit oder noch schlechtere Laufzeit haben. Er muss nur für jeden beliebigen DFA terminieren.

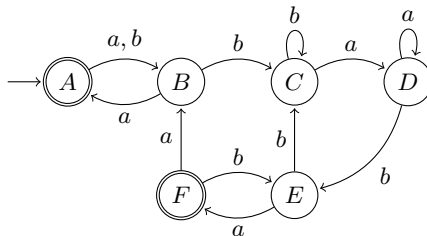


## Aufgabe T5.2



## Aufgabe T5.3

- (a) Minimieren Sie den DFA  $D$  mit dem Algorithmus aus der Vorlesung (Folie 116).

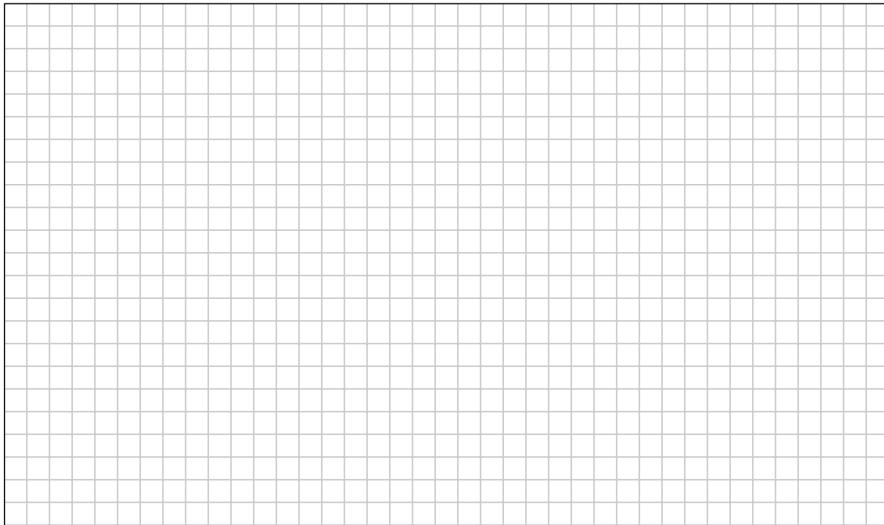


**Abbildung 1** Der Automat  $D$ .

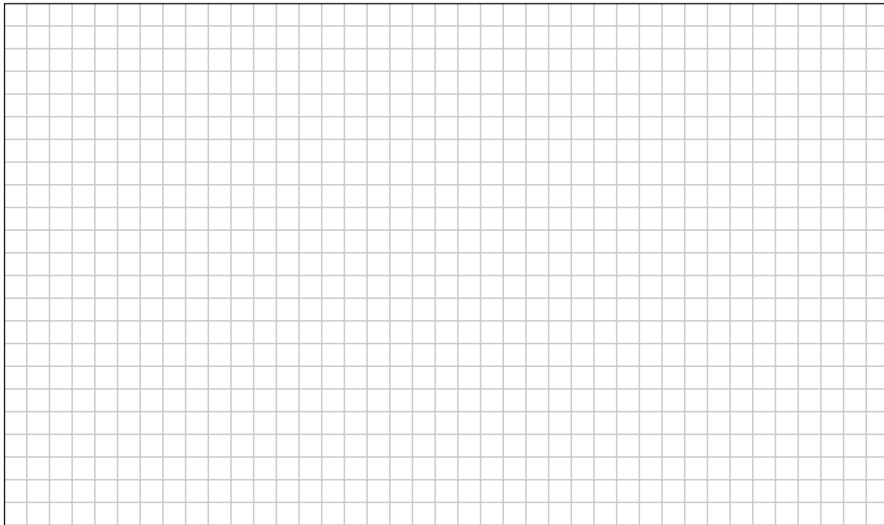
- (b) Ändern Sie den Minimierungsalgorithmus aus der Vorlesung so ab, dass er neben einem minimalen DFA auch noch für alle Zustände  $(q_1, q_2)$ , die nicht äquivalent sind, ein möglichst kurzes Wort  $w$  findet, das zeigt, dass  $q_1$  und  $q_2$  nicht äquivalent sind.

Wenden Sie den neuen Algorithmus auf den DFA  $D$  aus Teilaufgabe (a) an.

## Aufgabe T5.3



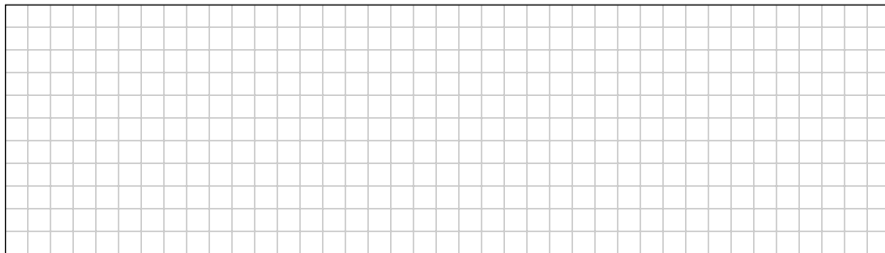
## Aufgabe T5.3



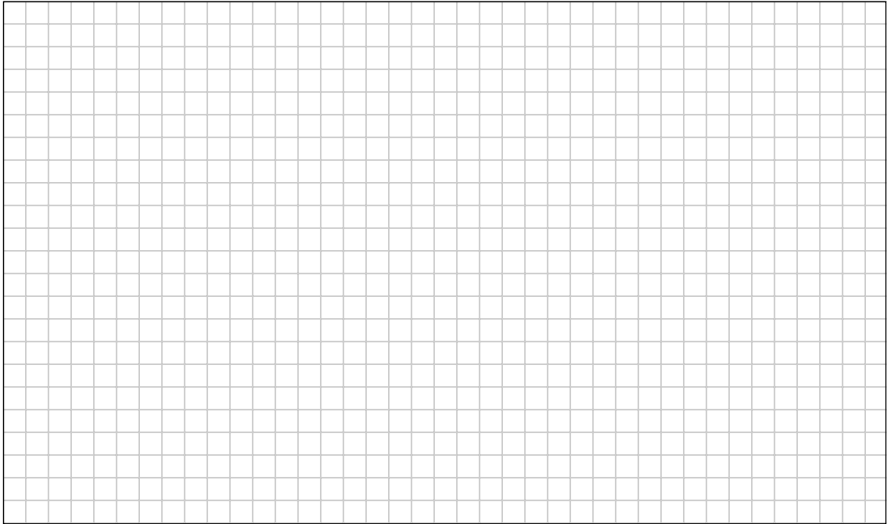
## Aufgabe T5.4

Sei  $L := L((a^*b \mid c)^*a)$  eine Sprache über dem Alphabet  $\Sigma := \{a, b, c, d\}$ .

- (a) Zeigen oder widerlegen Sie die folgenden Äquivalenzen:  
(i)  $b \equiv_L c$ , (ii)  $\varepsilon \equiv_L a$ , (iii)  $abc \equiv_L cba$
- (b) Sei  $v := aababc$ . Geben Sie ein Wort  $w \neq v$  an, sodass  $v \equiv_L w$  gilt.
- (c) Geben Sie die Mengen  $L^{ab}$ ,  $L^{ac}$  und  $L^{ca}$  an.
- (d) Finden Sie eine Sprache  $L' \subseteq \Sigma^*$  mit  $\varepsilon, aba \in L'$ , sodass  $c \equiv_{L'} ba$ ,  $c \not\equiv_{L'} ab$  und  $aba \equiv_L bab$  gelten.

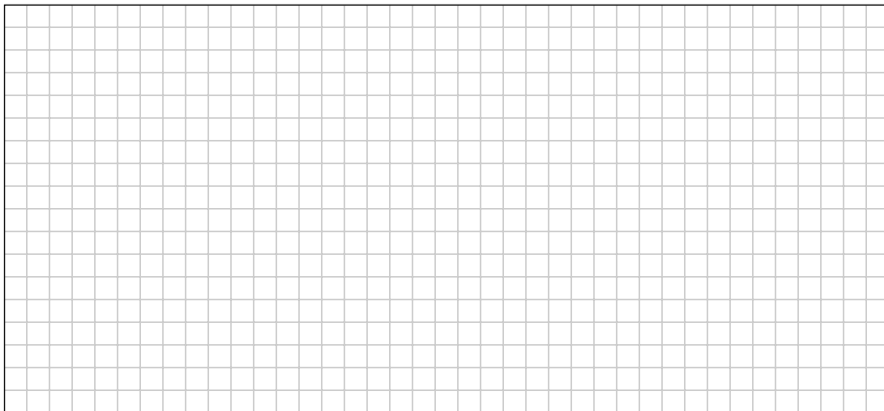


## Aufgabe T5.4

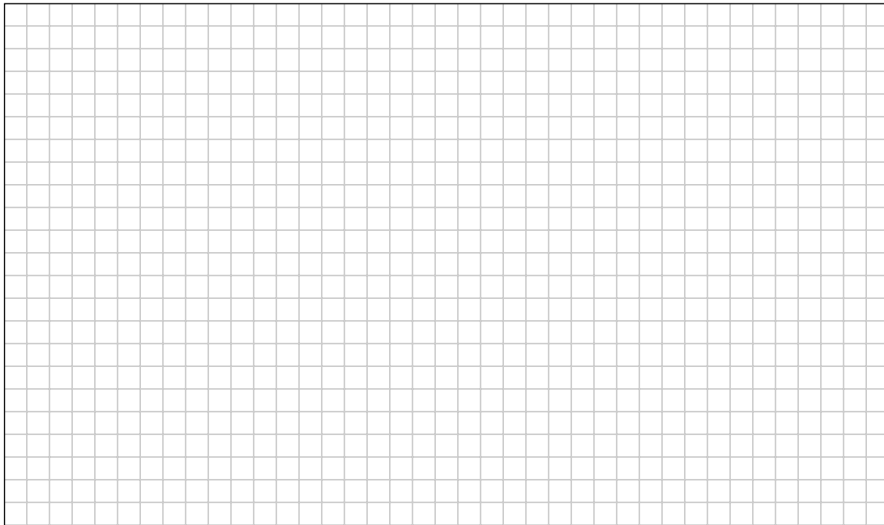


## Aufgabe T5.5

Konstruieren Sie den kanonischen Minimalautomaten (nach Definition 3.58) zu dem regulären Ausdruck  $r := ab \mid ba^*$ . Benennen Sie dabei jeden Zustand mit einem regulären Ausdruck für die entsprechende Residualsprache.

A large empty grid for drawing the automaton. The grid is 20 columns wide and 20 rows high, providing a workspace for the student to draw the canonical minimal automaton for the regular expression  $r := ab \mid ba^*$ .

## Aufgabe T5.5



## Endterm 2022 – Aufgabe 4b), 3 Punkte

Sei  $N$  ein DFA mit Zuständen  $\{p_1, p_2, p_3, p_4\}$ . Für  $i \in \{1, 2, 3, 4\}$  gilt  $L_N(p_i) = L(r_i)$ , mit den regulären Ausdrücken  $r_1 := a(ba)^*$ ,  $r_2 := (ba)^*$ ,  $r_3 := (ab)^*a$  und  $r_4 := \emptyset$ .

Unter welchen Bedingungen ist  $N$  minimal?

