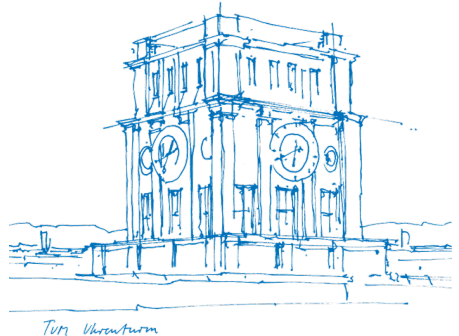


Einführung in die Theoretische Informatik

Tutorium – Woche 2

Esther Ney

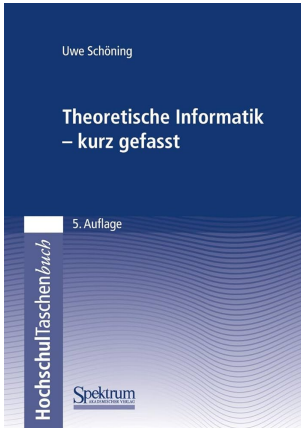
Sommersemester 2025



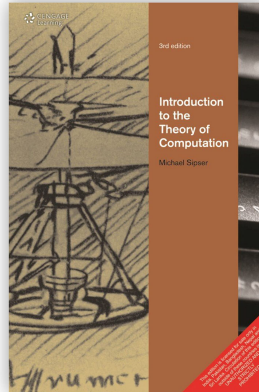
Zulip

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
- Aufbau einer Tutorstunde: kurze Wiederholung wichtiger Definitionen und Lemmata, dann selbständiges Bearbeiten der Aufgaben
- Folien: auf ZULIP oder <https://reflect.moe/uni/theo25/>



Uwe Schöningh – Theoretische Informatik kurz gefasst



Michael Sipser – Introduction to the Theory of Computation

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.

Die Folien sind nur für die Teilnehmer der entsprechenden Tutorien gedacht und dürfen nicht ohne die ausdrückliche Erlaubnis der Urheber vervielfältigt oder weitergegeben werden.

Definition 3.1 – Deterministischer endlicher Automat

Ein DETERMINISTISCHER ENDLICHER AUTOMAT (DFA) ist ein Tupel $M = (Q, \Sigma, \delta, q_0, F)$ mit ZUSTÄNDEN Q , EINGABEALPHABET Σ , ÜBERGANGSFUNKTION $\delta : Q \times \Sigma \rightarrow Q$, STARTZUSTAND q_0 und ENZUSTÄNDEN $F \subseteq Q$.

Q , Σ und F müssen endlich sein; δ muss TOTAL sein, d.h. auf jedem Zustand und Buchstaben definiert sein.

Definition 3.8 – Nichtdeterministischer endlicher Automat

Ein NICHTDETERMINISTISCHER ENDLICHER AUTOMAT (NFA) ist ein Tupel $N = (Q, \Sigma, \delta, q_0, F)$ mit ZUSTÄNDEN Q , EINGABEALPHABET Σ , ÜBERGANGSFUNKTION $\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q)$, STARTZUSTAND q_0 und ENZUSTÄNDEN $F \subseteq Q$.

Q , Σ und F müssen endlich sein; δ bildet auf die POTENZMENGE von Q ab, d.h. ein Übergang kann auch zu 0 oder MEHR ALS 1 Zuständen führen.

Intuitiv: Automaten lesen Wörter und nehmen wortabhängig verschiedene Zustände an. Zu jedem Zeitpunkt sind DFAs in genau einem und NFAs in beliebig vielen Zuständen.

Definition 3.2 – akzeptierte Sprache

Die von einem DFA M AKZEPTIERTE SPRACHE $L(M)$ ist definiert als

$$L(M) := \{w \in \Sigma^* \mid \hat{\delta}(q_0, w) \in F\}$$

Die von einem NFA N AKZEPTIERTE SPRACHE $L(N)$ ist definiert als

$$L(N) := \{w \in \Sigma^* \mid \hat{\hat{\delta}}(\{q_0\}, w) \cap F \neq \emptyset\}$$

Diese Definition motiviert, dass die Endzustände F eines Automaten auch als AKZEPTIERENDE ZUSTÄNDE bezeichnet werden.

Für die Definitionen von $\hat{\delta}$ und $\hat{\hat{\delta}}$ bitte in der Vorlesung nachschlagen.

Intuitiv: Ein Automat akzeptiert genau die Wörter, für der Automat einen Endzustand, d.h. einen Zustand in F , erreicht.

Notation 3.2 – Zustandsgraph

Endliche Automaten können durch einen ZUSTANDSGRAPHEN visualisiert werden, der als Knoten genau die ZUSTÄNDE Q und als Kanten genau die ÜBERGÄNGE aus der Übergangsfunktion δ besitzt. Kanten werden dabei mit den zugehörigen Buchstaben aus dem Alphabet Σ beschriftet.

Der Anfangszustand wird mit einem eingehenden Pfeil gekennzeichnet, und die Endzustände werden mit doppelten Kreisen gekennzeichnet.

Beispiele siehe Vorlesung.

In der Literatur ist für DFAS manchmal ein FALLZUSTAND $q_\emptyset \notin F$ (engl. TRAP STATE) implizit, der nicht gezeichnet wird, alle “fehlenden” Übergänge aufnimmt und für den gilt: $\delta(q_\emptyset, a) = q_\emptyset$ für alle $a \in \Sigma$. In THEO ist dies aber unerwünscht, bitte immer alle Zustände zeichnen.

Beispiel von letzter Woche

Sei $A = AA$. Zeige: $A^* \subseteq A$.

Antwort: Nach Vorlesung gilt $A^* = \bigcup_{n=0}^{\infty} A^n$. Nun zeigen wir per Induktion, dass $A^n \subseteq A$ für $n \in \mathbb{N}_0$:

Für $n = 0$ gilt, dass $\varepsilon \in A$: Sei w ein kürzestes Wort in A . Dann ist $w \in AA$, also gibt es $u, v \in A$ mit $uv = w$ und $|u| + |v| = |w|$. Aber da w kürzestes Wort ist, gilt schon $|w| \leq |u|$ und $|w| \leq |v|$ und somit $|w| = 0$, d.h. $w = \varepsilon$ und somit $\varepsilon \in A$.

Für den Induktionsschritt gilt $A^{n+1} = AA^n \subseteq AA = A$.

Es ist sehr wichtig, zu verstehen, WARUM dieses Argument funktioniert!

Im Werkzeugkasten der theoretischen Informatik kommen Argumente über die Länge eines Wortes sehr oft vor. Induktion kommt auch ziemlich oft vor.

Techniken wie “Sei w ein Wort, das kürzer als alle anderen Wörter in der Sprache ist”, aber auch Induktion: “Sei w ein beliebiges Wort. Dann gilt irgendwas für w per Induktion: Für $|w| = 0$ (oder $|w| = 1$) gilt das, und für $|w| = n + 1$ auch”, solltet ihr kennen.

Vorbereitung

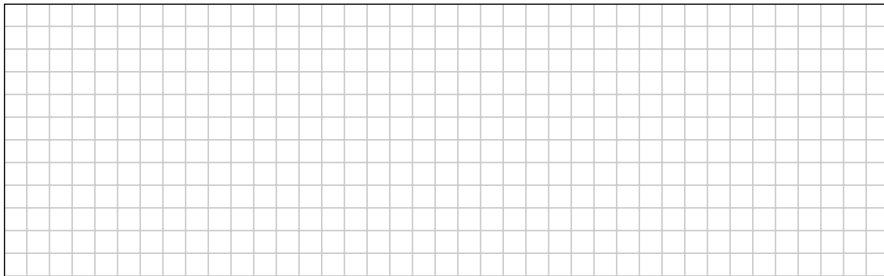
Überprüfen Sie, dass Sie die folgenden Begriffe und Notationen korrekt definieren können.

- DFA und NFA
- Zustandsgraph
- akzeptierte Sprache
- Akzeptanzbedingung von DFA und NFA

Aufgabe T2.1

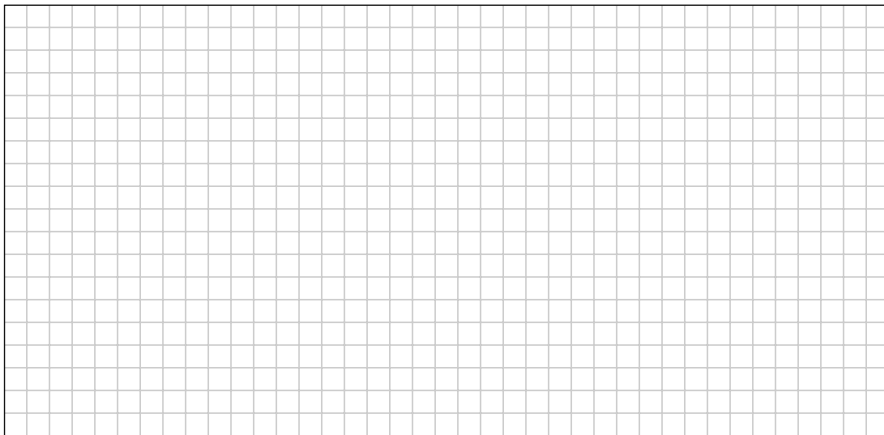
Wir betrachten die folgenden vier Sprachen über einem Alphabet Σ . Geben Sie jeweils einen DFA (graphisch) an, der genau die Sprache erkennt, die

- (a) am Anfang jedes Wortes eine bestimmte Sequenz von Buchstaben fordert,
- (b) alle Wörter gerader/ungerader Länge enthält,
- (c) alle Wörter einer fixierten Länge n enthält,
- (d) alle Wörter enthält, die an einer fixierten Position einen bestimmten Buchstaben enthalten.



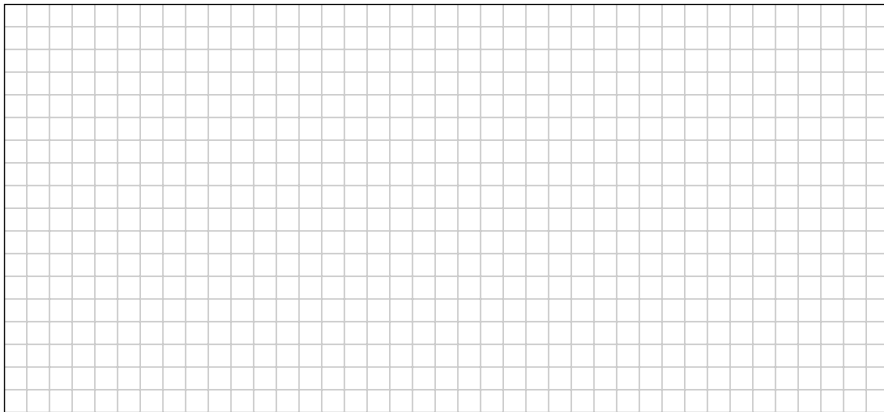
Aufgabe T2.1

- (a) am Anfang jedes Wortes eine bestimmte Sequenz von Buchstaben fordert,
- (b) alle Wörter gerader/ungerader Länge enthält.



Aufgabe T2.1

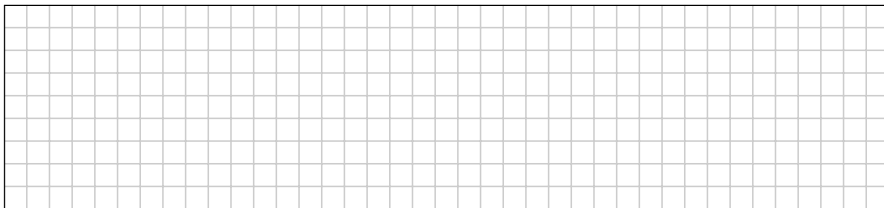
- (c) alle Wörter einer fixierten Länge n enthält,
- (d) alle Wörter enthält, die an einer fixierten Position einen bestimmten Buchstaben enthalten.



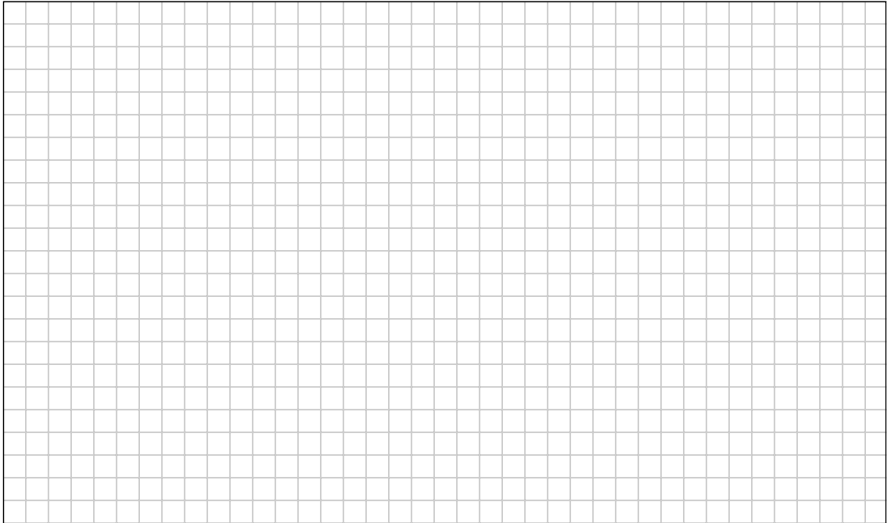
Aufgabe T2.2

Sei $\Sigma = \{a, b\}$ und $B_n := \{w \in \Sigma^* \mid \exists i : w_i = w_{i+n}\}$ die Sprache aller Wörter über Σ , in denen an irgendeiner Stelle der selbe Buchstabe im Abstand n vorkommt. Insbesondere ist B_0 die Menge aller nicht leeren Wörter, und B_1 die Menge aller Wörter in denen ein Buchstabe zweimal hintereinander vorkommt. VERSUCHEN SIE, in beiden Aufgabenteilen NFAS und DFAS mit möglichst wenigen Zuständen anzugeben.

- (a) Geben Sie jeweils einen NFA für B_0 , B_1 und B_2 an.
- (b) Geben Sie jeweils einen DFA für B_0 , B_1 und B_2 an.
- (c) Beschreiben Sie kurz, wie der NFA für B_n und der DFA für B_n für beliebige $n \in \mathbb{N}$ aussehen.



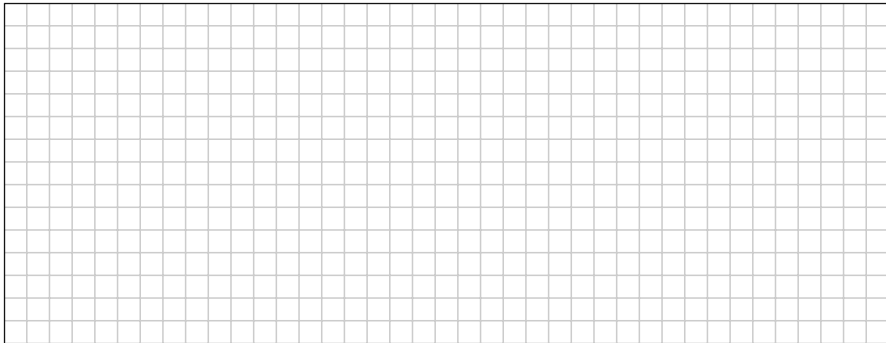
Aufgabe T2.2



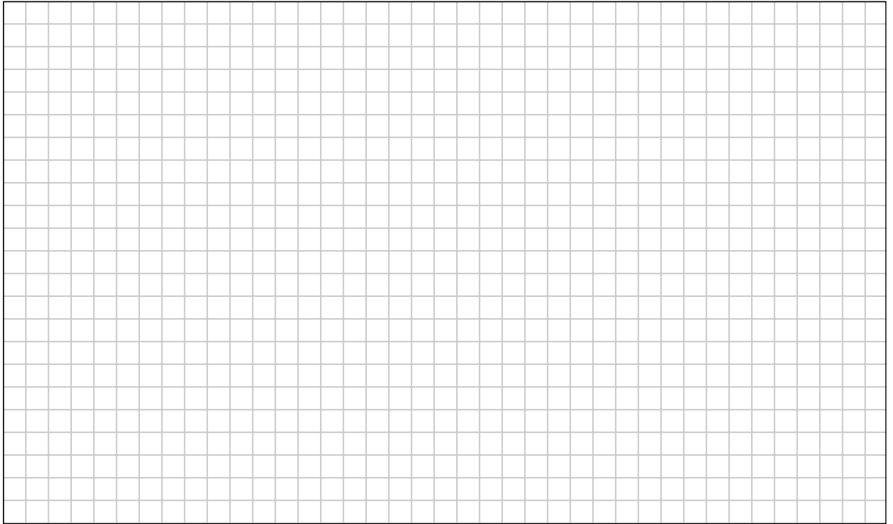
Aufgabe T2.3

Sei $M = (Q, \Sigma, \delta, q_0, F)$ ein deterministischer endlicher Automat (DFA). Wir betrachten Zustände $q_1, q_2, \dots, q_{n+1} \in Q$ und Zeichen $a_1, a_2, \dots, a_n \in \Sigma$. Nach Vorlesung, Folie 30, ist die relationale Notation $p \xrightarrow{a} q$ gleichbedeutend mit $\delta(p, a) = q$. Zeigen Sie:

$$q_1 \xrightarrow{a_1} q_2 \xrightarrow{a_2} \dots \xrightarrow{a_n} q_{n+1} \implies \hat{\delta}(q_1, a_1 a_2 \dots a_n) = q_{n+1}.$$



Aufgabe T2.3

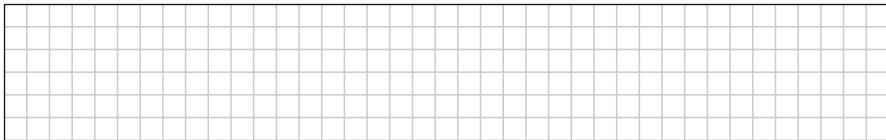


Aufgabe T2.4

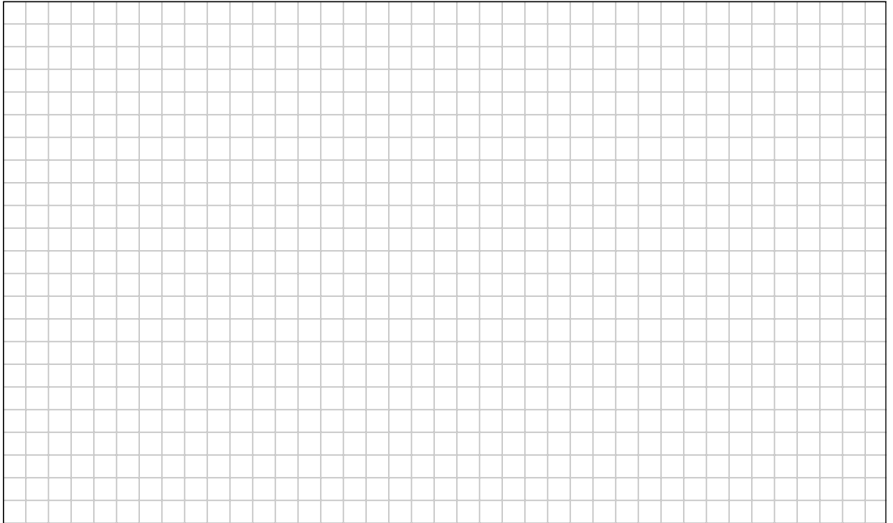
Entscheiden Sie, ob die folgenden Aussagen korrekt sind, und begründen Sie Ihre Behauptung, indem Sie entweder ein Gegenbeispiel oder eine passende Konstruktion angeben.

Für jeden NFA $N = (Q, \Sigma, \delta, q_0, F)$ gibt es einen NFA $N' = (Q', \Sigma, \delta', q'_0, F')$ mit $L(N) = L(N')$ und ...

- (a) der Startzustand hat keine eingehenden Kanten.
- (b) kein Endzustand hat eine ausgehende Kante.
- (c) für jeden Zustand q gilt: alle eingehenden Kanten von q sind mit demselben Zeichen beschriftet.
- (d) für jeden Zustand q gilt: alle ausgehenden Kanten von q sind mit demselben Zeichen beschriftet.



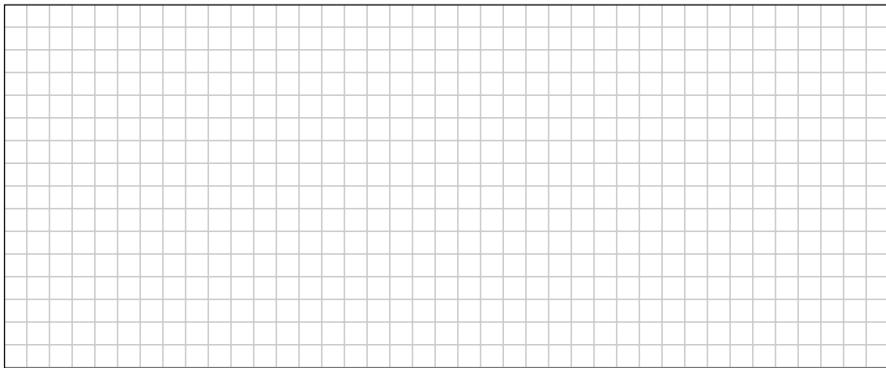
Aufgabe T2.4



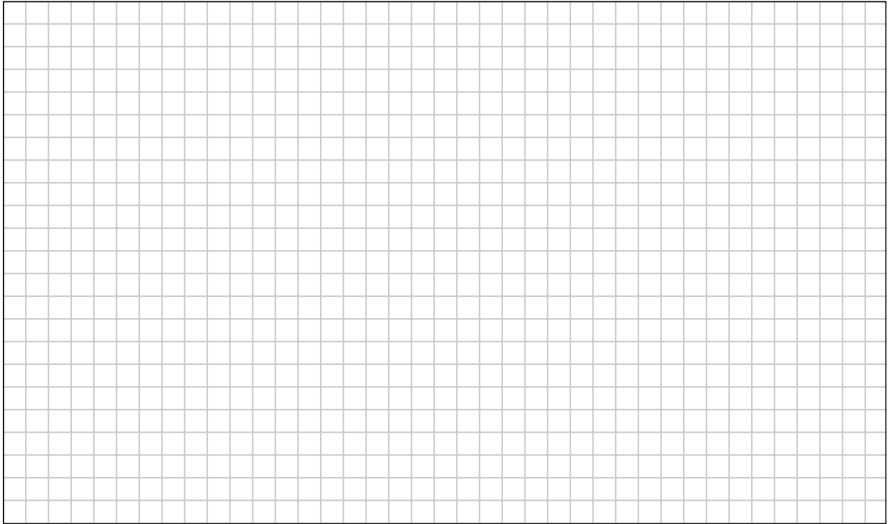
Aufgabe T2.5

Sei $\Sigma = \{a, b\}$ und $L = \{ba^n\}$ für ein beliebiges aber festes $n \in \mathbb{N}_0$.

- (a) Sei $n = 3$. Zeichnen Sie einen NFA, der L erkennt.
- (b) Zeigen sie, dass jeder NFA, der die Sprache L erkennt, mindestens $n + 2$ Zustände hat.



Aufgabe T2.5



Heute keine Altklausuraufgabe!

Dafür war zu wenig Stoff. Nächste Woche gibts aber wieder eine.