

Einführung in die Theoretische Informatik

Tutorium – Woche 10

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/>

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.

Kein Automat, sondern eine *Maschine*

Definition 5.10 – Turingmaschine (TM)

Wir sagen, dass $M := (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$ mit Zustandsmenge Q , Alphabet Σ , *Bandalphabet* Γ mit $\Sigma \subseteq \Gamma$, Anfangszustand q_0 , *Leerzeichen* $\square$, Übergangsfunktion $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, N\}$ und Endzuständen F eine *deterministische Turingmaschine* ist.

Annahme: für beliebige $q_f \in F$ und $a \in \Gamma$ ist $\delta(q_f, a)$ *undefiniert* – siehe Aufgabe 3.

Analog nennen wir M eine *nichtdeterministische Turingmaschine* für eine Übergangsfunktion $\delta : Q \times \Gamma \rightarrow \mathcal{P}(Q \times \Gamma \times \{L, R, N\})$.

Definition 5.14 – Akzeptanz und Berechenbarkeit

Wir sagen, dass eine Turingmaschine M die Sprache $L(M) := \{w \in \Sigma^* \mid \exists q_f \in F, \alpha, \beta \in \Gamma^* : (\varepsilon, q_0, w) \rightarrow_M^* (\alpha, q_f, \beta)\}$ akzeptiert.

Wir sagen, dass eine Funktion f (*Turing*-)berechenbar ist, falls es eine Turingmaschine M gibt, sodass $f(u) = v \iff (\varepsilon, q_0, u) \rightarrow_M^* (\square \dots \square, q_f, v \square \dots \square)$ für ein $q_f \in F$.

Definition 5.11 – Konfiguration einer Turingmaschine (TM)

Wir sagen, dass $(\alpha, q, \beta) \in \Gamma^* \times Q \times \Gamma^*$ eine *Konfiguration* einer Turingmaschine ist. Hierbei ist $\square\alpha\beta\square$ der Bandinhalt, und der Lese-/Schreibekopf beginnt links bei $\beta\square$.

Damit definieren wir die Relation $\rightarrow_M$:

Folie 236 – Die Relation $\rightarrow_M$

Wir definieren **first**(aw) = a und **first**(ε) = $\square$, **rest**(aw) = w und **rest**(ε) = ε , **last**(wa) = a und **last**(ε) = $\square$, **butlast**(w) = a und **butlast**(ε) = ε .

Dann definieren wir $\rightarrow_M$ abhängig von D in $\delta(q, \text{first}(\beta)) = (q', c, D)$:

$$(\alpha, q, \beta) \rightarrow_M (\alpha, q', c \text{rest}(\beta)) \quad (D = N)$$

$$(\alpha, q, \beta) \rightarrow_M (\alpha c, q', \text{rest}(\beta)) \quad (D = R)$$

$$(\alpha, q, \beta) \rightarrow_M (\text{butlast}(\alpha), q', \text{last}(\alpha) c \text{rest}(\beta)) \quad (D = L)$$

Folie 231 – Church-Turing-These

Intuitiv gilt: Alles, was sich durch eine Funktion oder Prozedur beschreiben lässt, lässt sich auch durch eine Turingmaschine berechnen.

Historisch gab es drei (als äquivalent bewiesene) Modelle für TYP 0-Sprachen:

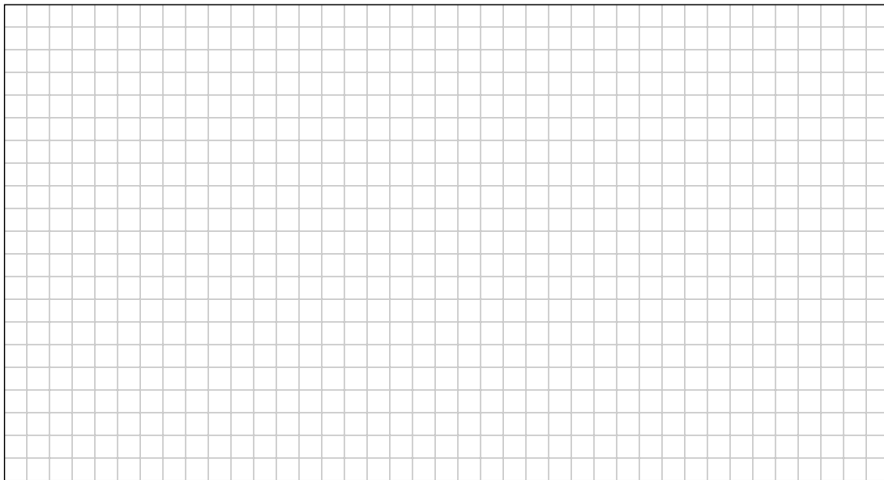
Turingmaschinen von ALAN TURING, ähnlich einer imperativen Sprache.

λ -Kalkül von ALONZO CHURCH, ähnlich einer funktionalen Sprache.

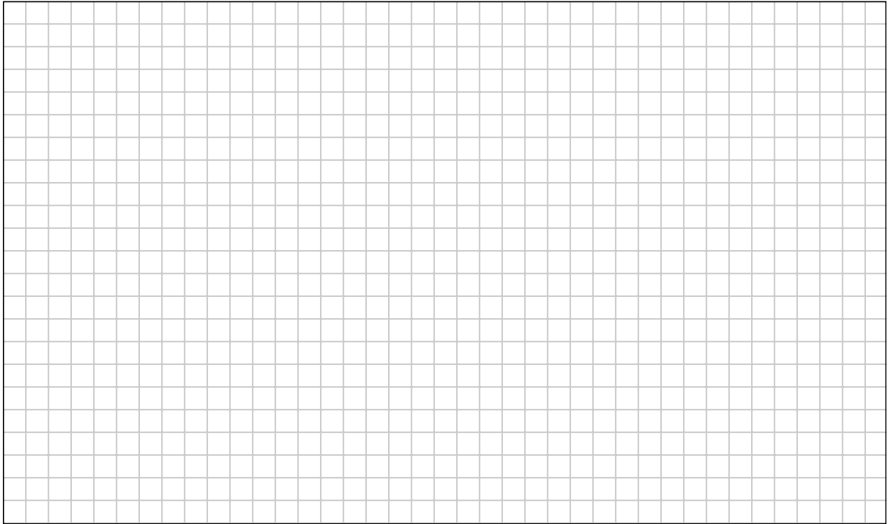
μ -rekursive Funktionen von KURT GÖDEL, als abgeschlossene Funktionsklasse.

Aufgabe T 10.1

Geben Sie eine Turingmaschine M mit $L(M) = \{a^n b^n c^n \mid n \in \mathbb{N}\}$ an.

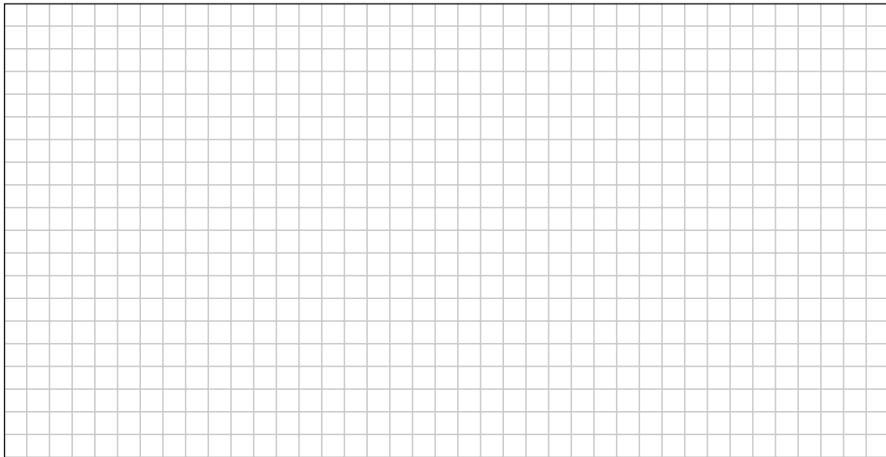


Aufgabe T 10.1

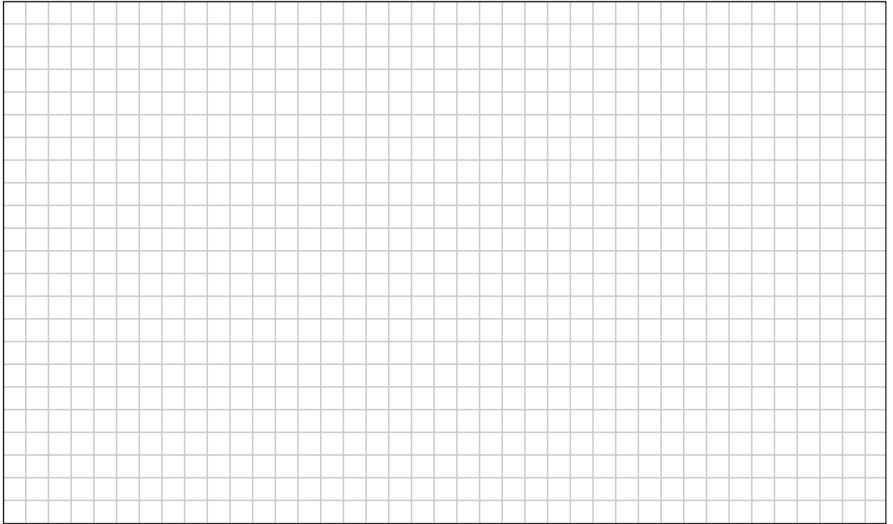


Aufgabe T 10.2

Geben Sie eine Turingmaschine M über dem Alphabet $\Sigma := \{\square\}$ an, die die Anzahl der Striche im Input *verdoppelt*, d.h. $(\varepsilon, q_0, |^n) \rightarrow (\square \dots \square, q_f, |^{2n} \square \dots \square)$ mit $q_f \in F$.



Aufgabe T 10.2



Hinweis – $\mathcal{M}$, $\mathcal{M}_A$, L_F , L_H und L_{FH}

Nicht alle Turingmaschinen erfüllen die Annahme, dass $\delta(q_f, a)$ undefiniert für $q_f \in F$, $a \in \Gamma$ ist. Daher definieren wir für eine beliebige Turingmaschine M :

$$L_F(M) := \{w \in \Sigma^* \mid \exists q_f \in F, \alpha, \beta \in \Gamma^* : (\varepsilon, q_0, w) \rightarrow_M^* (\alpha, q_f, \beta)\} \quad (L_F)$$

$$L_H(M) := \{w \in \Sigma^* \mid \exists q \in Q, \alpha, \beta \in \Gamma^* : (\varepsilon, q_0, w) \rightarrow_M^* (\alpha, q, \beta) \\ \text{und } \delta(q, \text{first}(\beta)) \text{ ist nicht definiert}\} \quad (L_H)$$

$$L_{HF}(M) := \{w \in \Sigma^* \mid \exists q_f \in F, \alpha, \beta \in \Gamma^* : (\varepsilon, q_0, w) \rightarrow_M^* (\alpha, q_f, \beta) \\ \text{und } \delta(q_f, \text{first}(\beta)) \text{ ist nicht definiert}\} \quad (L_{HF})$$

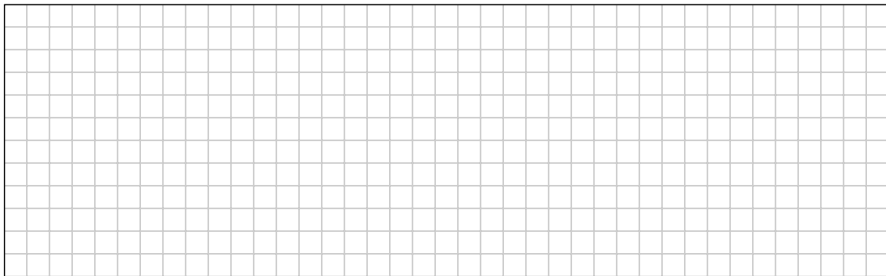
Weiter bezeichnen wir mit $\mathcal{M}_A$ die Klasse der Turingmaschinen, die obige Annahme erfüllen, und mit $\mathcal{M}$ die Klasse aller Turingmaschinen. Insbesondere gilt $\mathcal{M}_A \subsetneq \mathcal{M}$.

Intuitiv ist $L_F(M)$ die Menge der Wörter, für die M irgendwann einen Endzustand besucht; $L_H(M)$ die Menge der Wörter, für die M hält; $L_{HF}(M)$ die Menge der Wörter, für die M in einem Endzustand hält.

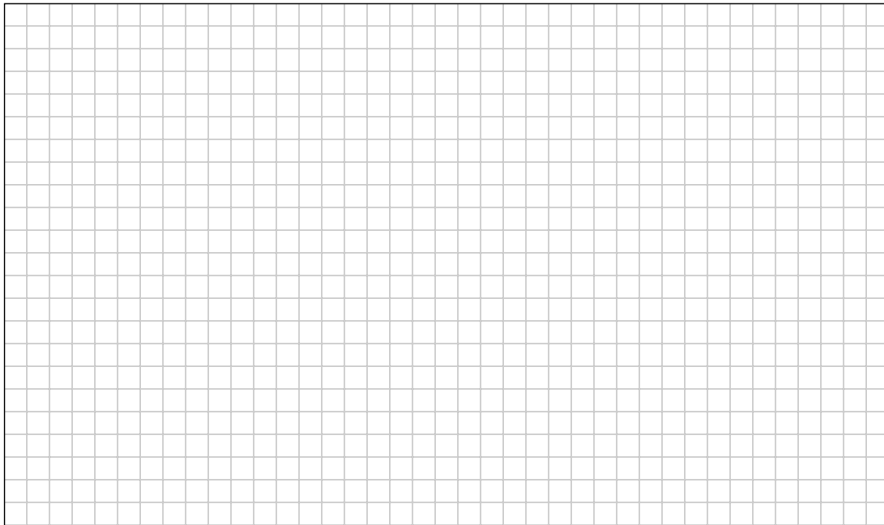
Aufgabe T 10.3

Zeigen Sie folgende Aussagen, indem Sie eine Konstruktion angeben.

- (a) Sei $M \in \mathcal{M}_A$. Dann gibt es ein $M' \in \mathcal{M}$ mit $L_F(M) = L_H(M')$.
- (b) Sei $M \in \mathcal{M}$. Dann gibt es ein $M' \in \mathcal{M}_A$ mit $L_H(M) = L_F(M')$.
- (c) Sei $M \in \mathcal{M}_A$. Dann gibt es ein $M' \in \mathcal{M}$ mit $L_F(M) = L_F(M')$.
- (d) Sei $M \in \mathcal{M}$. Dann gibt es ein $M' \in \mathcal{M}_A$ mit $L_F(M) = L_F(M')$.
- (e) Sei $M \in \mathcal{M}$. Dann gibt es ein $M' \in \mathcal{M}$ mit $L_F(M) = L_H(M')$.



Aufgabe T 10.3

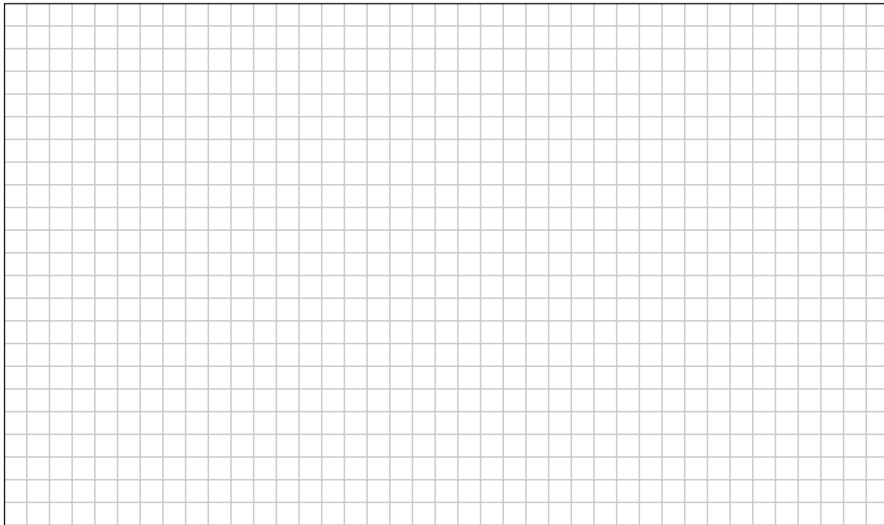


Aufgabe T 10.4

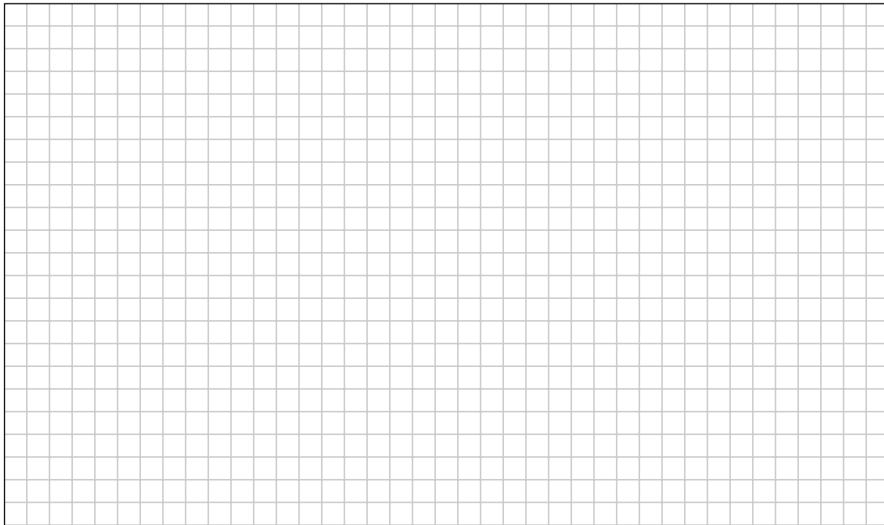
Entscheiden Sie, ob die folgenden Aussagen wahr oder falsch sind. Ein Beweis ist nicht notwendig; es reicht eine intuitive Begründung.

- (a) Seien f und g berechenbar. Dann ist die Komposition $f \circ g$ berechenbar.
- (b) Sei die Komposition $f \circ g$ berechenbar. Dann sind f und g berechenbar.
- (c) In einem unzugänglichen Safe befindet sich ein Zettel, auf dem eine unbekannte natürliche Zahl z steht. Dann ist die Funktion $1_z(n) := 1$ wenn $n = z, 0$ sonst berechenbar.
- (d) Die Klasse der “ruhelosen” Turingmaschinen, die ihren Kopf in jedem Schritt bewegen müssen (d.h. für alle Transitionen $\delta(q, a) = (q, b, D)$ muss $D \neq N$ sein), akzeptiert genau die TYP 0-Sprachen.
- (e) Sei $k \in \mathbb{N}$ eine Konstante. Die Klasse der Turingmaschinen, die ihren Kopf maximal k -mal bewegen können, akzeptiert genau die regulären Sprachen. (Transitionen der Form $\delta(q, a) = (q, b, N)$ zählen dabei nicht als Bewegung.)
- (f) Die Klasse der Turingmaschinen, die ihren Kopf nur nach rechts bewegen können (d.h. jede Transition ist von der Form $\delta(q, a) = (q, b, R)$), akzeptiert genau die regulären Sprachen.

Aufgabe T 10.4



Aufgabe T 10.4



Heute keine Altklausuraufgabe!

Wegen des Stoffumfangs **entfällt** heute die Altklausuraufgabe.