

Theoretische Informatik

Übungsstunde 5

Emil Vinu

Aufbau

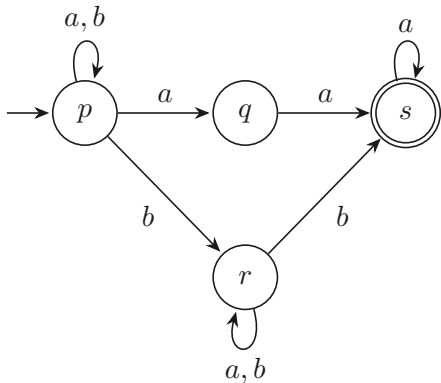
- Feedback Blatt 4
- Worked Example
- Recap
- Kahoot
- Aufgaben

Feedback Blatt 4

Feedback Blatt 4

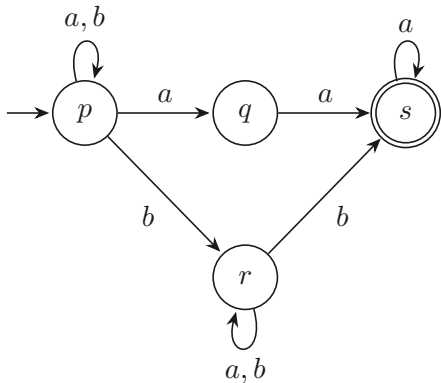
- Lief gut: 8a und die Wortwahl in 8b
- 7: $\lambda = a^0$ vergessen, 0 ist durch alles teilbar
- 7b: Zweig für *abaa*, *baaa* am Rest-0-Zustand
 - nach Rückkehr wieder erreichbar: *ababaa* wird akzeptiert
- 8b: Beweisrahmen
 - *jeder* EA mit ≤ 5 Zuständen, Schubfach, Lemma 3.3, Widerspruch

Aufgabe 8



- (a) Potenzmengenkonstruktion, nur erreichbare Zustände
- (b) Jeder äquivalente EA hat mindestens 6 Zustände.

Aufgabe 8



- (a) Potenzmengenkonstruktion, nur erreichbare Zustände
- (b) Jeder äquivalente EA hat mindestens 6 Zustände.

„ L : endet auf aa oder enthält mindestens zwei b “

Aufgabe 8 (a)

	a	b
$\langle \{p\} \rangle$	$\langle \{p, q\} \rangle$	$\langle \{p, r\} \rangle$

Aufgabe 8 (a)

	a	b
$\langle\{p\}\rangle$	$\langle\{p, q\}\rangle$	$\langle\{p, r\}\rangle$
$\langle\{p, q\}\rangle$	$\langle\{p, q, s\}\rangle$	$\langle\{p, r\}\rangle$
$\langle\{p, r\}\rangle$	$\langle\{p, q, r\}\rangle$	$\langle\{p, r, s\}\rangle$

Aufgabe 8 (a)

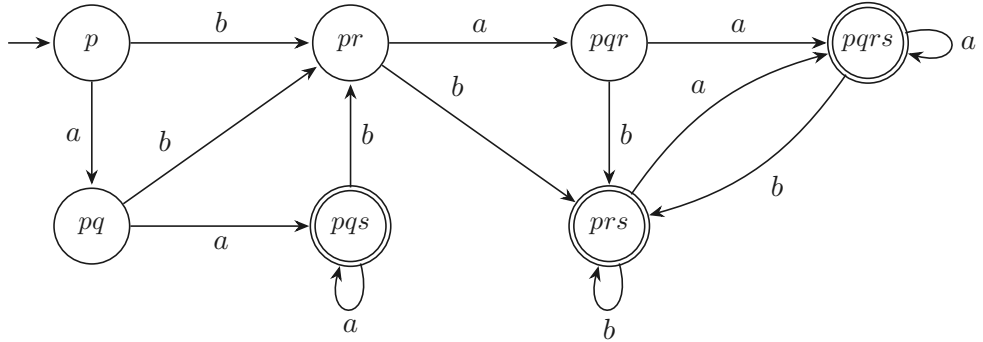
	a	b
$\langle \{p\} \rangle$	$\langle \{p, q\} \rangle$	$\langle \{p, r\} \rangle$
$\langle \{p, q\} \rangle$	$\langle \{p, q, s\} \rangle$	$\langle \{p, r\} \rangle$
$\langle \{p, r\} \rangle$	$\langle \{p, q, r\} \rangle$	$\langle \{p, r, s\} \rangle$
$\langle \{\mathbf{p}, \mathbf{q}, \mathbf{s}\} \rangle$	$\langle \{p, q, s\} \rangle$	$\langle \{p, r\} \rangle$
$\langle \{p, q, r\} \rangle$	$\langle \{p, q, r, s\} \rangle$	$\langle \{p, r, s\} \rangle$
$\langle \{\mathbf{p}, \mathbf{r}, \mathbf{s}\} \rangle$	$\langle \{p, q, r, s\} \rangle$	$\langle \{p, r, s\} \rangle$

Aufgabe 8 (a)

	a	b
$\langle \{p\} \rangle$	$\langle \{p, q\} \rangle$	$\langle \{p, r\} \rangle$
$\langle \{p, q\} \rangle$	$\langle \{p, q, s\} \rangle$	$\langle \{p, r\} \rangle$
$\langle \{p, r\} \rangle$	$\langle \{p, q, r\} \rangle$	$\langle \{p, r, s\} \rangle$
$\langle \{\mathbf{p}, \mathbf{q}, \mathbf{s}\} \rangle$	$\langle \{p, q, s\} \rangle$	$\langle \{p, r\} \rangle$
$\langle \{p, q, r\} \rangle$	$\langle \{p, q, r, s\} \rangle$	$\langle \{p, r, s\} \rangle$
$\langle \{\mathbf{p}, \mathbf{r}, \mathbf{s}\} \rangle$	$\langle \{p, q, r, s\} \rangle$	$\langle \{p, r, s\} \rangle$
$\langle \{\mathbf{p}, \mathbf{q}, \mathbf{r}, \mathbf{s}\} \rangle$	$\langle \{p, q, r, s\} \rangle$	$\langle \{p, r, s\} \rangle$

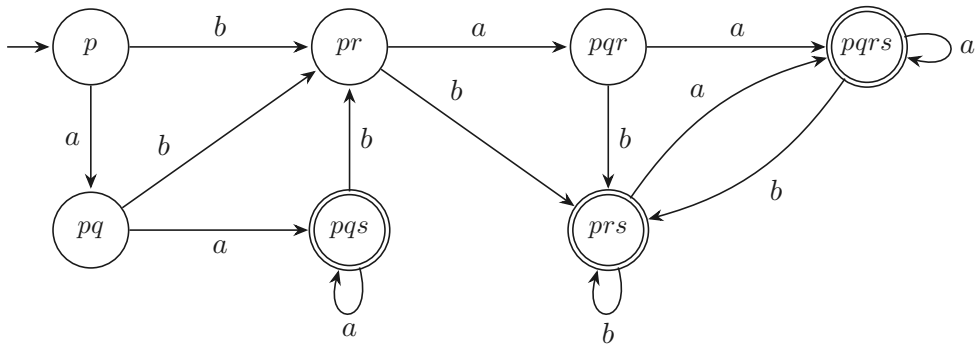
fett: akzeptierend

Der EA



pqr steht für $\langle \{p, q, r\} \rangle$

Der EA



pqr steht für $\langle \{p, q, r\} \rangle$

→ prs und $pqrs$: beide akzeptierend, man kommt nie mehr heraus.
Zusammen also 6 Zustände.

Aufgabe 8 (b): Rahmen

Angenommen, ein EA A für L hat höchstens 5 Zustände.

Dann landen zwei der 6 Wörter

$$\lambda, \quad a, \quad b, \quad aa, \quad ba, \quad bb$$

im selben Zustand. Nach Lemma 3.3 gilt dann für jedes z : $xz \in L \iff yz \in L$.

Widerspruch, sobald wir für jedes Paar ein trennendes z haben.

→ Wörter: je ein kürzestes Wort pro Zustand (pqr s weglassen)

Aufgabe 8 (b): Tabelle

	a	b	aa	ba	bb
λ	a	b	λ	b	λ
a	$-$	a	λ	b	λ
b	$-$	$-$	λ	a	λ
aa	$-$	$-$	$-$	λ	b
ba	$-$	$-$	$-$	$-$	λ

Zum Beispiel $b \cdot a = ba \notin L$, aber $ba \cdot a = baa \in L$.

Worked Example

Abgeschwächtes Pumping-Lemma

Sei L regulär über Σ . Dann existiert $n_0 \in \mathbb{N}$, so dass sich jedes Wort $w \in L$ mit $|w| \geq n_0$ als $w = yxz$ zerlegen lässt, so dass

- (i) $|yx| \leq n_0$, (ii) $|x| \geq 1$ und (iii) $\{yx^kz \mid k \in \mathbb{N}\} \subseteq L$.

Abgeschwächtes Pumping-Lemma

Sei L regulär über Σ . Dann existiert $n_0 \in \mathbb{N}$, so dass sich jedes Wort $w \in L$ mit $|w| \geq n_0$ als $w = yxz$ zerlegen lässt, so dass

(i) $|yx| \leq n_0$, (ii) $|x| \geq 1$ und (iii) $\{yx^kz \mid k \in \mathbb{N}\} \subseteq L$.

Lemma 3.4: für alle $w \in \Sigma^*$, und (iii) heisst $\subseteq L$ oder $\cap L = \emptyset$.

! hier wird nur in Wörtern aus L gepumpt

Die Aufgabe

$$L = \{a^m b^{n!+n} \mid m, n \in \mathbb{N} - \{0\}\} \cup \{b\}^*$$

-  (a) Zeigen Sie mit dem Pumping-Lemma aus der Vorlesung, dass L nicht regulär ist.
-  (b) Zeigen Sie, dass L das abgeschwächte Pumping-Lemma erfüllt.

Abgeschwächtes Pumping-Lemma. *Es gibt n_0 , so dass jedes $w \in L$ mit $|w| \geq n_0$ eine Zerlegung $w = yxz$ hat mit $|yx| \leq n_0$, $|x| \geq 1$ und $\{yx^k z \mid k \in \mathbb{N}\} \subseteq L$.*

Die Aufgabe

$$L = \{a^m b^{n!+n} \mid m, n \in \mathbb{N} - \{0\}\} \cup \{b\}^*$$

-  (a) Zeigen Sie mit dem Pumping-Lemma aus der Vorlesung, dass L nicht regulär ist.
-  (b) Zeigen Sie, dass L das abgeschwächte Pumping-Lemma erfüllt.

Abgeschwächtes Pumping-Lemma. *Es gibt n_0 , so dass jedes $w \in L$ mit $|w| \geq n_0$ eine Zerlegung $w = yxz$ hat mit $|yx| \leq n_0$, $|x| \geq 1$ und $\{yx^k z \mid k \in \mathbb{N}\} \subseteq L$.*

Mögliche Anzahl b hinter den a : $n! + n = 2, 4, 9, 28, 125, \dots$

Zwischenfrage



(a) Mit welchem Wort w fangen wir an? Warum klappt kein Wort aus L ?

(a) Die Idee

Für $w = a^m b^{n!+n} \in L$ wählt der Gegner $y = \lambda$, $x = a$. Alles Gepumpte bleibt in L .

(a) Die Idee

Für $w = a^m b^{n!+n} \in L$ wählt der Gegner $y = \lambda$, $x = a$. Alles Gepumpte bleibt in L .

Also: $w \notin L$, und dann so oft pumpen, bis wir in L landen.

(a) Die Idee

Für $w = a^m b^{n!+n} \in L$ wählt der Gegner $y = \lambda$, $x = a$. Alles Gepumpte bleibt in L .

Also: $w \notin L$, und dann so oft pumpen, bis wir in L landen.

$$w = ab^p, \quad p \geq n_0 \text{ Primzahl, } p > 2$$

(a) Die Idee

Für $w = a^m b^{n!+n} \in L$ wählt der Gegner $y = \lambda$, $x = a$. Alles Gepumpte bleibt in L .

Also: $w \notin L$, und dann so oft pumpen, bis wir in L landen.

$$w = ab^p, \quad p \geq n_0 \text{ Primzahl, } p > 2$$

$w \notin L$: aus $p = n! + n$ folgt $n \mid p$, also $n = 1$ ($p = 2$) oder $n = p$ ($p! + p > p$).

(a) Zwei Fälle

Nach (i) und (ii): $w = ab^p = yxz$ mit $|yx| \leq n_0$ und $|x| \geq 1$.

Fall $y = \lambda$: x enthält das a . $k = 0$: $yz \in \{b\}^* \subseteq L$, aber $yxz = w \notin L$.

(a) Zwei Fälle

Nach (i) und (ii): $w = ab^p = yxz$ mit $|yx| \leq n_0$ und $|x| \geq 1$.

Fall $y = \lambda$: x enthält das a . $k = 0$: $yz \in \{b\}^* \subseteq L$, aber $yxz = w \notin L$.

Fall $y \neq \lambda$: $y = ab^r$, $x = b^s$ mit $1 \leq s \leq n_0 \leq p$.

$$yx^k z = a b^{p+(k-1)s}$$

(a) Zwei Fälle

Nach (i) und (ii): $w = ab^p = yxz$ mit $|yx| \leq n_0$ und $|x| \geq 1$.

Fall $y = \lambda$: x enthält das a . $k = 0$: $yz \in \{b\}^* \subseteq L$, aber $yxz = w \notin L$.

Fall $y \neq \lambda$: $y = ab^r$, $x = b^s$ mit $1 \leq s \leq n_0 \leq p$.

$$yx^kz = a b^{p+(k-1)s}$$

Welches k ?

(a) Zwei Fälle

Nach (i) und (ii): $w = ab^p = yxz$ mit $|yx| \leq n_0$ und $|x| \geq 1$.

Fall $y = \lambda$: x enthält das a . $k = 0$: $yz \in \{b\}^* \subseteq L$, aber $yxz = w \notin L$.

Fall $y \neq \lambda$: $y = ab^r$, $x = b^s$ mit $1 \leq s \leq n_0 \leq p$.

$$yx^k z = a b^{p+(k-1)s}$$

Welches k ? $k = \frac{p!}{s} + 1$ (ganzzahlig, da $s \leq p$): $yx^k z = a b^{p!+p} \in L$.

In beiden Fällen Widerspruch zu (iii).



Zwischenfrage



(b) Wie zerlegen wir $w = a^m b^{n!+n}$, so dass alle yx^kz in L bleiben?

(b) Lösung

$n_0 = 1$, also jedes $w \in L$ mit $w \neq \lambda$.

$w \in \{b\}^*$: $y = \lambda, x = b$. Alle $yx^kz \in \{b\}^* \subseteq L$.

(b) Lösung

$n_0 = 1$, also jedes $w \in L$ mit $w \neq \lambda$.

$w \in \{b\}^*$: $y = \lambda, x = b$. Alle $yx^kz \in \{b\}^* \subseteq L$.

$w = a^m b^{n!+n}$: $y = \lambda, x = a, z = a^{m-1} b^{n!+n}$

$k \geq 1$: $a^{m+k-1} b^{n!+n} \in L$

(b) Lösung

$n_0 = 1$, also jedes $w \in L$ mit $w \neq \lambda$.

$w \in \{b\}^*$: $y = \lambda, x = b$. Alle $yx^kz \in \{b\}^* \subseteq L$.

$w = a^m b^{n!+n}$: $y = \lambda, x = a, z = a^{m-1} b^{n!+n}$

$k \geq 1$: $a^{m+k-1} b^{n!+n} \in L$

$k = 0, m \geq 2$: $a^{m-1} b^{n!+n} \in L$

$k = 0, m = 1$: $b^{n!+n} \in \{b\}^* \subseteq L$

(b) Lösung

$n_0 = 1$, also jedes $w \in L$ mit $w \neq \lambda$.

$w \in \{b\}^*$: $y = \lambda, x = b$. Alle $yx^kz \in \{b\}^* \subseteq L$.

$w = a^m b^{n!+n}$: $y = \lambda, x = a, z = a^{m-1} b^{n!+n}$

$k \geq 1$: $a^{m+k-1} b^{n!+n} \in L$

$k = 0, m \geq 2$: $a^{m-1} b^{n!+n} \in L$

$k = 0, m = 1$: $b^{n!+n} \in \{b\}^* \subseteq L$

$\rightarrow$ das abgeschwächte Lemma sieht nur Wörter aus L , und dort rettet $\{b\}^*$ alles

Recap

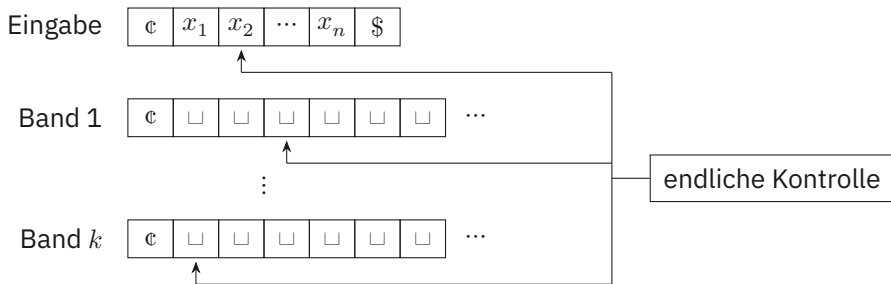
Funktionen berechnen

Definition. Wir sagen, dass M eine Funktion $F: \Sigma^* \rightarrow \Gamma^*$ berechnet, falls für alle $x \in \Sigma^*$: $q_0 \mathbb{C} x \vdash_M^* q_{\text{accept}} \mathbb{C} F(x)$.

Definition. Eine Funktion $F: \Sigma_1^* \rightarrow \Sigma_2^*$ für zwei Alphabete Σ_1, Σ_2 heißt **total berechenbar**, falls eine TM M existiert, die F berechnet.

→ M hält auf jeder Eingabe, der Kopf steht am Ende auf $\mathbb{C}$

Mehrband-Turingmaschine



- Eingabeband $\mathbb{C}w\$$: nur lesen
- k Arbeitsbänder, jedes mit eigenem Lese-/Schreibkopf

! eine k -Band-TM hat $k + 1$ Köpfe

Mehrband-TM formal

Übergänge:

$$\delta: Q \times (\Sigma \cup \{\mathbb{C}, \$\}) \times \Gamma^k \rightarrow Q \times \{L, R, N\} \times (\Gamma \times \{L, R, N\})^k$$

Konfiguration:

$$(q, w, i, u_1, i_1, \dots, u_k, i_k) \in Q \times \Sigma^* \times \mathbb{N} \times (\Gamma^* \times \mathbb{N})^k$$

w, i Eingabe $\mathbb{C}w\$$, Lesekopf auf Feld i

u_j, i_j Band j enthält $\mathbb{C}u_j \sqcup \dots$, Kopf auf Feld i_j

Zwischenfrage



Wie erkennt eine Mehrband-TM mit einem Arbeitsband die Palindrome über $\{a, b\}$?

Palindrome mit einem Arbeitsband

1. Eingabe auf das Arbeitsband kopieren.
2. Lesekopf zurück auf ϵ , Arbeitskopf bleibt auf dem letzten Zeichen.
3. Lesekopf nach rechts, Arbeitskopf nach links: Zeichen vergleichen.
4. Ungleich: verwerfen. Lesekopf auf $\$$: akzeptieren.

→ *Session 4 mit einem Band: für jedes Zeichen einmal hin und her*

Äquivalenz von Maschinen

Definition. Seien A und B zwei Maschinen (TM, MTM), die mit dem gleichen Eingabealphabet Σ arbeiten. A ist äquivalent zu B , falls für jede Eingabe $x \in \Sigma^*$ gilt:

- (i) A akzeptiert $x \iff B$ akzeptiert x ,
- (ii) A verwirft $x \iff B$ verwirft x ,
- (iii) A arbeitet unendlich lange auf $x \iff B$ arbeitet unendlich lange auf x .

! $L(A) = L(B)$ reicht nicht: A kann verwerfen, wo B unendlich läuft

TM und MTM

Lemma 4.1. *Zu jeder TM A existiert eine zu A äquivalente 1-Band-TM B .*

„Eingabe aufs Arbeitsband kopieren und dort weitermachen.“

Lemma 4.2. *Für jede Mehrband-TM A existiert eine zu A äquivalente TM B .*

Beweisidee Lemma 4.2: Spuren

Eingabe	⊞	a	b	b	\$	□	
Kopf			↑				
Band 1	⊞	0	1	□	□	□	
Kopf		↑					...
Band 2	⊞	1	1	0	□	□	
Kopf					↑		

Ein Schritt von A :

1. links $\rightarrow$ rechts: alle $\uparrow$ suchen, die Symbole darüber im Zustand merken
2. rechts $\rightarrow$ links: Symbole ersetzen, $\uparrow$ verschieben

$\rightarrow$ ein Symbol von B ist eine ganze Spalte

Maschinenmodelle

Definition 4.2. *Zwei Maschinenmodelle (Maschinenklassen) $\mathcal{A}$ und $\mathcal{B}$ für Entscheidungsprobleme sind äquivalent, falls*

- (i) für jede Maschine $A \in \mathcal{A}$ eine zu A äquivalente Maschine $B \in \mathcal{B}$ existiert, und*
- (ii) für jede Maschine $C \in \mathcal{B}$ eine zu C äquivalente Maschine $D \in \mathcal{A}$ existiert.*

Satz 4.1. *Die Maschinenmodelle von Turingmaschinen und Mehrband-Turingmaschinen sind äquivalent.*

→ für „ L ist rekursiv“ reicht eine MTM

Church'sche These

Church'sche These. *Die Turingmaschinen sind die Formalisierung des Begriffes „Algorithmus“, d. h., die Klasse der rekursiven Sprachen (der entscheidbaren Entscheidungsprobleme) stimmt mit der Klasse der algorithmisch (automatisch) erkennbaren Sprachen überein.*

→ *Algorithmus = TM, die immer hält*

Nichtdeterministische TM

Definition 4.3. Eine *nichtdeterministische Turingmaschine (NTM)* ist ein 7-Tupel $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$, wobei

- (i) $Q, \Sigma, \Gamma, q_0, q_{\text{accept}}, q_{\text{reject}}$ die gleiche Bedeutung wie bei einer TM haben, und
- (ii) $\delta: (Q - \{q_{\text{accept}}, q_{\text{reject}}\}) \times \Gamma \rightarrow \mathcal{P}(Q \times \Gamma \times \{\text{L}, \text{R}, \text{N}\})$ die Übergangsfunktion von M ist und die folgende Eigenschaft hat:

$$\delta(p, c) \subseteq \{(q, c, X) \mid q \in Q, X \in \{\text{R}, \text{N}\}\} \quad \text{für alle } p \in Q.$$

Akzeptierte Sprache

Definition 4.3. *Die von der NTM M akzeptierte Sprache ist*

$$L(M) = \{w \in \Sigma^* \mid q_0 \mathfrak{C} w \vdash_M^* y q_{\text{accept}} z \text{ für irgendwelche } y, z \in \Gamma^*\}.$$

→ mindestens eine Berechnung akzeptiert

! die anderen dürfen verwerfen oder unendlich laufen

Raten und Verifizieren

NTM mit einem Arbeitsband für $L = \{uu \mid u \in \{a, b\}^*\}$:

1. Eingabe aufs Arbeitsband kopieren, bis sie rät: „jetzt kommt die Mitte“.
2. Arbeitskopf zurück auf ϵ .
3. Rest der Eingabe mit dem Arbeitsband vergleichen.
4. Gleich, und beide gleichzeitig am Ende: akzeptieren.

→ falsch geraten: diese Berechnung verwirft, eine richtige reicht

Berechnungsbaum

Definition 4.4. Sei $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$ eine NTM und sei x ein Wort über dem Eingabealphabet Σ von M . Ein **Berechnungsbaum** $T_{M,x}$ von M auf x ist ein (potentiell unendlicher) gerichteter Baum mit einer Wurzel, der wie folgt definiert wird.

- (i) Jeder Knoten von $T_{M,x}$ ist mit einer Konfiguration beschriftet.
- (ii) Die Wurzel ist der einzige Knoten von $T_{M,x}$ mit dem Eingangsgrad 0 und ist mit der Startkonfiguration $q_0 \mathbb{C} x$ beschriftet.
- (iii) Jeder Knoten des Baumes, der mit einer Konfiguration C beschriftet ist, hat genauso viele Kinder wie C Nachfolgekonfigurationen hat, und diese Kinder sind mit diesen Nachfolgekonfigurationen von C markiert.

NTM $\rightarrow$ TM

Satz 4.2. *Sei M eine NTM. Dann existiert eine TM A , so dass*

- (i) $L(M) = L(A)$ und*
- (ii) falls M keine unendlichen Berechnungen auf Wörtern aus $(L(M))^c$ hat, dann hält A immer.*

Kodierung: Symbole

$$Q = \{q_0, q_1, \dots, q_m, q_{\text{accept}}, q_{\text{reject}}\}, \Gamma = \{A_1, A_2, \dots, A_r\}$$

$$\text{Code}(q_i) = 10^{i+1}1$$

$$\text{Code}(A_j) = 110^j11$$

$$\text{Code}(q_{\text{accept}}) = 10^{m+2}1$$

$$\text{Code}(N) = 1110111$$

$$\text{Code}(q_{\text{reject}}) = 10^{m+3}1$$

$$\text{Code}(R) = 11100111$$

$$\text{Code}(L) = 111000111$$

→ Zustände beginnen mit 10, Symbole mit 110, Richtungen mit 1110

Kodierung: Transitionen

$$\text{Code}(\delta(p, A_l) = (q, A_j, \alpha)) = \# \text{Code}(p) \text{Code}(A_l) \text{Code}(q) \text{Code}(A_j) \text{Code}(\alpha) \#$$

$$\text{Code}(M) = \#0^{m+3}\#0^r\#\# \text{Code}(\text{Transition}_1)\# \text{Code}(\text{Transition}_2)\# \dots$$

→ zuerst $|Q|$ und $|\Gamma|$, dann alle Transitionen

Kodierung über Σ_{bool}

$$h(\#) = 01, \quad h(0) = 00, \quad h(1) = 11$$

Definition 4.5. Für jede Turingmaschine M wird

$$\text{Kod}(M) = h(\text{Code}(M))$$

die Kodierung der TM M genannt. $\text{Kod}_{\text{TM}} = \{\text{Kod}(M) \mid M \text{ ist eine TM}\}$ bezeichnet die Menge der Kodierungen aller Turingmaschinen.

$\rightarrow M \mapsto \text{Kod}(M)$ ist injektiv

Die i -te Turingmaschine

Definition 4.6. Sei $x \in (\Sigma_{\text{bool}})^*$. Für jedes $i \in \mathbb{N} - \{0\}$ sagen wir, dass x die Kodierung der i -ten TM ist, falls

- (i) $x = \text{Kod}(M)$ für eine TM M und
- (ii) die Menge $\{y \in (\Sigma_{\text{bool}})^* \mid y \text{ ist vor } x \text{ in kanonischer Ordnung}\}$ enthält genau $i - 1$ Wörter, die Kodierungen von Turingmaschinen sind.

Dann ist M die i -te Turingmaschine M_i .

Lemma 4.3

$\text{Gen}: \mathbb{N} - \{0\} \rightarrow \Sigma_{\text{bool}}^*$, $\text{Gen}(i) = \text{Kod}(M_i)$

Lemma 4.3. *Die Funktion Gen ist total rekursiv, d. h., es existiert eine Turingmaschine (Programm) A_{Gen} , die für eine gegebene Zahl i die Kodierung $\text{Kod}(M_i)$ berechnet.*

Idee: Wörter in kanonischer Ordnung durchgehen, mit A_{ver} testen, mitzählen.

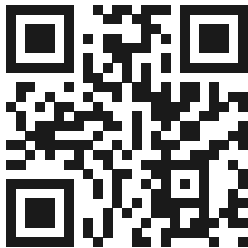
$\rightarrow M_1, M_2, M_3, \dots$ ist eine Liste aller TMs

Unterscheidung

rekursiv aufzählbar ($\mathcal{L}_{\text{RE}}$)	TM antwortet bei „ja“, bei „nein“ evtl. nie
rekursiv ($\mathcal{L}_{\text{R}}$)	TM antwortet immer: ja oder nein
total rekursiv	TM antwortet immer mit einem Wort $F(x)$

Kahoot

Kahoot!



<https://kahoot.it>

Aufgaben

Aufgaben



Aufgabe 1:

Beschreiben Sie in Worten eine Mehrband-TM mit einem Arbeitsband, die immer hält und $L = \{w \in \{a, b\}^* \mid |w|_a = 2 \cdot |w|_b\}$ akzeptiert.



Aufgabe 2:

Eine TM hat $Q = \{q_0, q_1, q_{\text{accept}}, q_{\text{reject}}\}$. (a) Welche Transition hat den Code #100111001110000111000011111000111#? (b) Warum beginnt kein Wort aus Kod_{TM} mit 0110?



Aufgabe 3:

Zeigen Sie: $L \in \mathcal{L}_R \iff L \in \mathcal{L}_{RE} \text{ und } \Sigma^* - L \in \mathcal{L}_{RE}$.

Lösung Aufgabe 1

1. Eingabe lesen: für jedes a ein X aufs Arbeitsband schreiben.
2. Lesekopf zurück auf ϵ . Der Arbeitskopf steht auf dem letzten X .
3. Eingabe nochmals lesen: für jedes b zwei X löschen (zweimal nach links).
Steht der Arbeitskopf dabei schon auf ϵ : verwerfen.
4. Lesekopf auf $\$$: akzeptieren genau dann, wenn der Arbeitskopf auf ϵ steht.

→ jede Phase ist ein Durchlauf, also hält die MTM immer

Lösung Aufgabe 2

(a) $m = 1$:

$$\# \underbrace{1001}_{q_1} \underbrace{110011}_{A_2} \underbrace{100001}_{q_{\text{reject}}} \underbrace{11000011}_{A_4} \underbrace{111000111}_{L} \#$$

Also $\delta(q_1, A_2) = (q_{\text{reject}}, A_4, L)$. Vier Nullen: $10^{m+3}1 = \text{Code}(q_{\text{reject}})$.

Lösung Aufgabe 2

(a) $m = 1$:

$$\# \underbrace{1001}_{q_1} \underbrace{110011}_{A_2} \underbrace{100001}_{q_{\text{reject}}} \underbrace{11000011}_{A_4} \underbrace{111000111}_{L} \#$$

Also $\delta(q_1, A_2) = (q_{\text{reject}}, A_4, L)$. Vier Nullen: $10^{m+3}1 = \text{Code}(q_{\text{reject}})$.

(b) $\text{Code}(M)$ beginnt mit $\#0$, also $\text{Kod}(M)$ mit $h(\#)h(0) = 0100$.

Lösung Aufgabe 3: $\Rightarrow$

M hält immer und $L(M) = L$. Insbesondere $L \in \mathcal{L}_{\text{RE}}$.

Lösung Aufgabe 3: $\Rightarrow$

M hält immer und $L(M) = L$. Insbesondere $L \in \mathcal{L}_{\text{RE}}$.

M' : q_{accept} und q_{reject} vertauschen. M' hält immer und akzeptiert genau, was M verwirft:

$$L(M') = \Sigma^* - L \in \mathcal{L}_{\text{RE}}$$

$\rightarrow$ sogar $\Sigma^* - L \in \mathcal{L}_{\text{R}}$

Lösung Aufgabe 3: $\Leftarrow$

M_1 akzeptiert L , M_2 akzeptiert $\Sigma^* - L$. Beide dürfen unendlich laufen.

Lösung Aufgabe 3: $\Leftarrow$

M_1 akzeptiert L , M_2 akzeptiert $\Sigma^* - L$. Beide dürfen unendlich laufen.

2-Band-TM A : w auf beide Bänder kopieren, dann abwechselnd

- einen Schritt von M_1 auf Band 1,
- einen Schritt von M_2 auf Band 2.

M_1 akzeptiert: akzeptieren. M_2 akzeptiert: verwerfen.

Lösung Aufgabe 3: $\Leftarrow$

M_1 akzeptiert L , M_2 akzeptiert $\Sigma^* - L$. Beide dürfen unendlich laufen.

2-Band-TM A : w auf beide Bänder kopieren, dann abwechselnd

- einen Schritt von M_1 auf Band 1,
- einen Schritt von M_2 auf Band 2.

M_1 akzeptiert: akzeptieren. M_2 akzeptiert: verwerfen.

Jedes w liegt in L oder in $\Sigma^* - L$, also akzeptiert einer der beiden nach endlich vielen Schritten. A hält immer, mit Lemma 4.2 gibt es eine TM. $\square$

! nacheinander geht nicht: M_1 kann unendlich laufen

Noch drei Aufgaben



Aufgabe 4:

Seien $L_1, L_2 \in \mathcal{L}_{\text{RE}}$. Zeigen Sie, dass auch $L_1 \cup L_2$ und $L_1 L_2$ in $\mathcal{L}_{\text{RE}}$ liegen.



Aufgabe 5:

Eine TM M verlässt auf keiner Eingabe w die Felder $0, 1, \dots, |w| + 1$. Zeigen Sie, dass $L(M) \in \mathcal{L}_{\text{R}}$.



Aufgabe 6:

Sei $L \in \mathcal{L}_{\text{RE}}$ unendlich. Zeigen Sie: Es gibt eine unendliche Sprache $L' \subseteq L$ mit $L' \in \mathcal{L}_{\text{R}}$.

Lösung Aufgabe 4

M_1, M_2 akzeptieren L_1, L_2 . Wir bauen NTMs und benutzen Satz 4.2 (i).

$L_1 \cup L_2$: rate $i \in \{1, 2\}$, simuliere M_i auf w .

Lösung Aufgabe 4

M_1, M_2 akzeptieren L_1, L_2 . Wir bauen NTMs und benutzen Satz 4.2 (i).

$L_1 \cup L_2$: rate $i \in \{1, 2\}$, simuliere M_i auf w .

$L_1 L_2$: rate eine Zerlegung $w = uv$, dann

1. simuliere M_1 auf u ,
2. akzeptiert M_1 : simuliere M_2 auf v , akzeptiere, wenn M_2 akzeptiert.

Akzeptierende Berechnung $\iff$ es gibt $u \in L_1, v \in L_2$ mit $w = uv$.

! deterministisch alle Zerlegungen nacheinander geht nicht: M_1 kann unendlich laufen

Lösung Aufgabe 5

Sei $n = |w|$. Wie bei Kahoot-Frage 15 hat M auf w höchstens

$$N = |Q| \cdot (n + 2) \cdot |\Gamma|^{n+1}$$

Konfigurationen.

Läuft M mehr als N Schritte, kommt eine Konfiguration doppelt vor: M läuft unendlich.

Lösung Aufgabe 5: die MTM

MTM A :

1. N binär auf Band 2 schreiben.
2. M auf Band 1 simulieren, nach jedem Schritt N um 1 verringern.
3. M hält: dasselbe tun. Zähler bei 0: verwerfen.

A hält immer und $L(A) = L(M)$.



Lösung Aufgabe 6: Idee

M akzeptiert L . Auf $x \notin L$ kann M unendlich laufen: Wir wissen nie, ob noch etwas kommt.

Lösung Aufgabe 6: Idee

M akzeptiert L . Auf $x \notin L$ kann M unendlich laufen: Wir wissen nie, ob noch etwas kommt.

Idee: L' so wählen, dass wir wissen, wann wir aufhören dürfen zu warten.

Lösung Aufgabe 6: Idee

M akzeptiert L . Auf $x \notin L$ kann M unendlich laufen: Wir wissen nie, ob noch etwas kommt.

Idee: L' so wählen, dass wir wissen, wann wir aufhören dürfen zu warten.

1. L aufzählen: alle Wörter aus L nacheinander ausgeben.
2. Nur Wörter behalten, die länger sind als alle bisher behaltenen.

„Kam schon ein Wort, das länger ist als x , dann kommt x nicht mehr.“

Lösung Aufgabe 6: L aufzählen

Sei $w_1, w_2, w_3, \dots$ die kanonische Ordnung von Σ^* .

Runde $t = 1, 2, 3, \dots$: simuliere M je t Schritte auf $w_1, \dots, w_t$. Jedes neu akzeptierte Wort wird ausgegeben.

Lösung Aufgabe 6: L aufzählen

Sei $w_1, w_2, w_3, \dots$ die kanonische Ordnung von Σ^* .

Runde $t = 1, 2, 3, \dots$: simuliere M je t Schritte auf $w_1, \dots, w_t$. Jedes neu akzeptierte Wort wird ausgegeben.

- Es kommen nur Wörter aus L .
- Akzeptiert M das Wort w_i nach s Schritten, kommt w_i spätestens in Runde $\max(i, s)$.

! nicht erst w_1 fertig simulieren: Darauf kann M unendlich laufen.

Lösung Aufgabe 6: immer länger

Die Aufzählung gibt $u_1, u_2, u_3, \dots$ aus.

L' : u_1 und jedes u_j , das länger ist als alle davor behaltenen.

Ausgabe	0110	1	00000	11	010101
in L'	ja		ja		ja

Lösung Aufgabe 6: immer länger

Die Aufzählung gibt $u_1, u_2, u_3, \dots$ aus.

L' : u_1 und jedes u_j , das länger ist als alle davor behaltenen.

Ausgabe	0110	1	00000	11	010101
in L'	ja		ja		ja

- $L' \subseteq L$
- L' ist unendlich: Jede Länge hat nur endlich viele Wörter, L ist unendlich, also kommt immer noch ein längeres.

Lösung Aufgabe 6: L' entscheiden

Eingabe x : die Wörter von L' erzeugen, bis eines länger als x ist.

x ist vorher gekommen: akzeptieren. Sonst: verwerfen.

Lösung Aufgabe 6: L' entscheiden

Eingabe x : die Wörter von L' erzeugen, bis eines länger als x ist.

x ist vorher gekommen: akzeptieren. Sonst: verwerfen.

- hält immer: L' ist unendlich, und die Längen wachsen
- richtig: Die Längen wachsen, also kommt x danach nie mehr.

Also $L' \in \mathcal{L}_R$.



→ die Aufzählung ist deterministisch, also liegt L' fest

Bis nächste Woche!