Theoretische Informatik

FSS 2020 - Tutorium #2

Alexander Moch

2020-03-11

Aufgabe 2.1

Aufgabe 2.1

Zeige, dass das Halteproblem, das spezielle Halteproblem und die Diagonalsprache rekursiv aufzählbar sind.

Aufgabe 2.1

Zeige, dass das Halteproblem, das spezielle Halteproblem und die Diagonalsprache rekursiv aufzählbar sind.

Eine Sprache L heißt rekursiv aufzählbar, falls es eine TM gibt, welche genau die Eingaben x akzeptiert, für die $x \in L$ gilt.

Aufgabe 2.1

Zeige, dass das Halteproblem, das spezielle Halteproblem und die Diagonalsprache rekursiv aufzählbar sind.

Eine Sprache L heißt rekursiv aufzählbar, falls es eine TM gibt, welche genau die Eingaben x akzeptiert, für die $x \in L$ gilt.

- **Halteproblem**: $H = \{([M], x) \mid M \text{ hält auf } x\}$

Aufgabe 2.1

Zeige, dass das Halteproblem, das spezielle Halteproblem und die Diagonalsprache rekursiv aufzählbar sind.

Eine Sprache L heißt rekursiv aufzählbar, falls es eine TM gibt, welche genau die Eingaben x akzeptiert, für die $x \in L$ gilt.

- **Halteproblem**: $H = \{([M], x) \mid M \text{ hält auf } x\}$

  Starte M auf x und akzeptiere genau dann wenn M hält.

Aufgabe 2.1

Zeige, dass das Halteproblem, das spezielle Halteproblem und die Diagonalsprache rekursiv aufzählbar sind.

Eine Sprache L heißt rekursiv aufzählbar, falls es eine TM gibt, welche genau die Eingaben x akzeptiert, für die $x \in L$ gilt.

- **Halteproblem**: $H = \{([M], x) \mid M \text{ hält auf } x\}$

  Starte M auf x und akzeptiere genau dann wenn M hält.
- **Spez. Halteproblem**: $H_\epsilon = \{[M] \mid M \text{ hält auf leerem Band}\}$

Aufgabe 2.1

Zeige, dass das Halteproblem, das spezielle Halteproblem und die Diagonalsprache rekursiv aufzählbar sind.

Eine Sprache L heißt rekursiv aufzählbar, falls es eine TM gibt, welche genau die Eingaben x akzeptiert, für die $x \in L$ gilt.

- **Halteproblem**: $H = \{([M], x) \mid M \text{ hält auf } x\}$

  Starte M auf x und akzeptiere genau dann wenn M hält.
- **Spez. Halteproblem**: $H_\epsilon = \{[M] \mid M \text{ hält auf leerem Band}\}$

  Starte M auf ϵ und akzeptiere genau dann wenn M hält.

Aufgabe 2.1

Zeige, dass das Halteproblem, das spezielle Halteproblem und die Diagonalsprache rekursiv aufzählbar sind.

Eine Sprache L heißt rekursiv aufzählbar, falls es eine TM gibt, welche genau die Eingaben x akzeptiert, für die $x \in L$ gilt.

- **Halteproblem**: $H = \{([M], x) \mid M \text{ hält auf } x\}$

  Starte M auf x und akzeptiere genau dann wenn M hält.
- **Spez. Halteproblem**: $H_\epsilon = \{[M] \mid M \text{ hält auf leerem Band}\}$

  Starte M auf ϵ und akzeptiere genau dann wenn M hält.
- **Diagonalsp.**: $D = \{[M] \mid M \text{ hält auf } [M] \text{ und verwirft } [M]\}$

Aufgabe 2.1

Zeige, dass das Halteproblem, das spezielle Halteproblem und die Diagonalsprache rekursiv aufzählbar sind.

Eine Sprache L heißt rekursiv aufzählbar, falls es eine TM gibt, welche genau die Eingaben x akzeptiert, für die $x \in L$ gilt.

- **Halteproblem**: $H = \{([M], x) \mid M \text{ hält auf } x\}$

  Starte M auf x und akzeptiere genau dann wenn M hält.
- **Spez. Halteproblem**: $H_\epsilon = \{[M] \mid M \text{ hält auf leerem Band}\}$

  Starte M auf ϵ und akzeptiere genau dann wenn M hält.
- **Diagonalsp.**: $D = \{[M] \mid M \text{ hält auf } [M] \text{ und verwirft } [M]\}$

  Starte M auf $[M]$ und akzeptiere genau dann wenn M anhält und $[M]$ verwirft.

Aufgabe 2.2

Aufgabe 2.2

Zeige: Ist für eine rekursiv aufzählbare Sprache L auch ihr Komplement $\overline{L}$ rekursiv aufzählbar, so ist L rekursiv.

Aufgabe 2.2

Zeige: Ist für eine rekursiv aufzählbare Sprache L auch ihr Komplement $\overline{L}$ rekursiv aufzählbar, so ist L rekursiv.

- Seien M_L^0 und M_L^1 TMs, welche L bzw. $\overline{L}$ aufzählen.

Aufgabe 2.2

Zeige: Ist für eine rekursiv aufzählbare Sprache L auch ihr Komplement $\overline{L}$ rekursiv aufzählbar, so ist L rekursiv.

- Seien M_L^0 und M_L^1 TMs, welche L bzw. $\overline{L}$ aufzählen.
- Turingmaschine M betreibt M_L^0 und M_L^1 parallel.

Aufgabe 2.2

Zeige: Ist für eine rekursiv aufzählbare Sprache L auch ihr Komplement $\overline{L}$ rekursiv aufzählbar, so ist L rekursiv.

- Seien M_L^0 und M_L^1 TMs, welche L bzw. $\overline{L}$ aufzählen.
- Turingmaschine M betreibt M_L^0 und M_L^1 parallel.
- M akzeptiert, wenn M_L^0 akzeptiert...

Aufgabe 2.2

Zeige: Ist für eine rekursiv aufzählbare Sprache L auch ihr Komplement $\overline{L}$ rekursiv aufzählbar, so ist L rekursiv.

- Seien M_L^0 und M_L^1 TMs, welche L bzw. $\overline{L}$ aufzählen.
- Turingmaschine M betreibt M_L^0 und M_L^1 parallel.
- M akzeptiert, wenn M_L^0 akzeptiert...
- ... und verwirft wenn M_L^1 akzeptiert.

Aufgabe 2.2

Zeige: Ist für eine rekursiv aufzählbare Sprache L auch ihr Komplement $\overline{L}$ rekursiv aufzählbar, so ist L rekursiv.

- Seien M_L^0 und M_L^1 TMs, welche L bzw. $\overline{L}$ aufzählen.
- Turingmaschine M betreibt M_L^0 und M_L^1 parallel.
- M akzeptiert, wenn M_L^0 akzeptiert...
- ... und verwirft wenn M_L^1 akzeptiert.

M entscheidet L.

Aufgabe 2.3

Aufgabe 2.3

$H^{\tilde{M}}$ bezeichnet die Menge all jener Eingaben auf denen $\tilde{M}$ hält.

Zeige, dass $H^{\tilde{M}}$ nicht entscheidbar ist.

Aufgabe 2.3

$H^{\tilde{M}}$ bezeichnet die Menge all jener Eingaben auf denen $\tilde{M}$ hält.

Zeige, dass $H^{\tilde{M}}$ nicht entscheidbar ist.

Wähle $\tilde{M}$ als universelle Turingmaschine M_U.

Aufgabe 2.3

$H^{\tilde{M}}$ bezeichnet die Menge all jener Eingaben auf denen $\tilde{M}$ hält.

Zeige, dass $H^{\tilde{M}}$ nicht entscheidbar ist.

Wähle $\tilde{M}$ als universelle Turingmaschine M_U.

Wenn H^{M_U} entscheidbar ist, so auch das Halteproblem H.

Aufgabe 2.3

$H^{\tilde{M}}$ bezeichnet die Menge all jener Eingaben auf denen $\tilde{M}$ hält.

Zeige, dass $H^{\tilde{M}}$ nicht entscheidbar ist.

Wähle $\tilde{M}$ als universelle Turingmaschine M_U.

Wenn H^{M_U} entscheidbar ist, so auch das Halteproblem H.

$([M], x) \in H^{M_U}$

Aufgabe 2.3

$H^{\tilde{M}}$ bezeichnet die Menge all jener Eingaben auf denen $\tilde{M}$ hält.

Zeige, dass $H^{\tilde{M}}$ nicht entscheidbar ist.

Wähle $\tilde{M}$ als universelle Turingmaschine M_U.

Wenn H^{M_U} entscheidbar ist, so auch das Halteproblem H.

$([M], x) \in H^{M_U} \iff M_U$ hält auf $([M], x)$

Aufgabe 2.3

$H^{\tilde{M}}$ bezeichnet die Menge all jener Eingaben auf denen $\tilde{M}$ hält.

Zeige, dass $H^{\tilde{M}}$ nicht entscheidbar ist.

Wähle $\tilde{M}$ als universelle Turingmaschine M_U.

Wenn H^{M_U} entscheidbar ist, so auch das Halteproblem H.

$([M], x) \in H^{M_U} \iff M_U$ hält auf $([M], x) \iff M$ hält auf x.

Aufgabe 2.4

Aufgabe 2.4

Zeige: Eine Sprache ist genau dann rekursiv, wenn es eine Turingmaschine gibt, welche die Wörter $w \in L$ in kanonischer Reihenfolge auf ein Ausgabeband schreibt.

Aufgabe 2.4

Zeige: Eine Sprache ist genau dann rekursiv, wenn es eine Turingmaschine gibt, welche die Wörter $w \in L$ in kanonischer Reihenfolge auf ein Ausgabeband schreibt.

- Richtung "$\Rightarrow$":

Aufgabe 2.4

Zeige: Eine Sprache ist genau dann rekursiv, wenn es eine Turingmaschine gibt, welche die Wörter $w \in L$ in kanonischer Reihenfolge auf ein Ausgabeband schreibt.

- Richtung "$\Rightarrow$":
  - Wir testen Wörter in kanonischer Reihenfolge.

Aufgabe 2.4

Zeige: Eine Sprache ist genau dann rekursiv, wenn es eine Turingmaschine gibt, welche die Wörter $w \in L$ in kanonischer Reihenfolge auf ein Ausgabeband schreibt.

- Richtung "$\Rightarrow$":
  - Wir testen Wörter in kanonischer Reihenfolge.
  - Wenn $w \in L$ dann geben wir das Wort aus.

Aufgabe 2.4

Zeige: Eine Sprache ist genau dann rekursiv, wenn es eine Turingmaschine gibt, welche die Wörter $w \in L$ in kanonischer Reihenfolge auf ein Ausgabeband schreibt.

- Richtung "$\Rightarrow$":
  - Wir testen Wörter in kanonischer Reihenfolge.
  - Wenn $w \in L$ dann geben wir das Wort aus.
- Richtung "$\Leftarrow$":

Aufgabe 2.4

Zeige: Eine Sprache ist genau dann rekursiv, wenn es eine Turingmaschine gibt, welche die Wörter $w \in L$ in kanonischer Reihenfolge auf ein Ausgabeband schreibt.

- Richtung "$\Rightarrow$":
  - Wir testen Wörter in kanonischer Reihenfolge.
  - Wenn $w \in L$ dann geben wir das Wort aus.
- Richtung "$\Leftarrow$":
  - Sei ein Wort w gegeben.

Aufgabe 2.4

Zeige: Eine Sprache ist genau dann rekursiv, wenn es eine Turingmaschine gibt, welche die Wörter $w \in L$ in kanonischer Reihenfolge auf ein Ausgabeband schreibt.

- Richtung "$\Rightarrow$":
  - Wir testen Wörter in kanonischer Reihenfolge.
  - Wenn $w \in L$ dann geben wir das Wort aus.
- Richtung "$\Leftarrow$":
  - Sei ein Wort w gegeben.
  - Schreibe Wörter aus L in kanonischer Reihenfolge.

Aufgabe 2.4

Zeige: Eine Sprache ist genau dann rekursiv, wenn es eine Turingmaschine gibt, welche die Wörter $w \in L$ in kanonischer Reihenfolge auf ein Ausgabeband schreibt.

- Richtung "$\Rightarrow$":
  - Wir testen Wörter in kanonischer Reihenfolge.
  - Wenn $w \in L$ dann geben wir das Wort aus.
- Richtung "$\Leftarrow$":
  - Sei ein Wort w gegeben.
  - Schreibe Wörter aus L in kanonischer Reihenfolge.
  - Wenn w geschrieben wird, akzeptieren wir.

Aufgabe 2.4

Zeige: Eine Sprache ist genau dann rekursiv, wenn es eine Turingmaschine gibt, welche die Wörter $w \in L$ in kanonischer Reihenfolge auf ein Ausgabeband schreibt.

- Richtung "$\Rightarrow$":
  - Wir testen Wörter in kanonischer Reihenfolge.
  - Wenn $w \in L$ dann geben wir das Wort aus.
- Richtung "$\Leftarrow$":
  - Sei ein Wort w gegeben.
  - Schreibe Wörter aus L in kanonischer Reihenfolge.
  - Wenn w geschrieben wird, akzeptieren wir.
  - Wenn ein Wort hinter w geschrieben wir, verwerfen wir.

Aufgabe 2.5

Aufgabe 2.5

Zeige: Eine Sprache L ist genau dann rekursiv aufzählbar, wenn es eine Turingmaschine gibt, welche die Wörter $w \in L$ auf ein Ausgabeband schreibt.

Aufgabe 2.5

Zeige: Eine Sprache L ist genau dann rekursiv aufzählbar, wenn es eine Turingmaschine gibt, welche die Wörter $w \in L$ auf ein Ausgabeband schreibt.

- Richtung "$\Rightarrow$":

Aufgabe 2.5

Zeige: Eine Sprache L ist genau dann rekursiv aufzählbar, wenn es eine Turingmaschine gibt, welche die Wörter $w \in L$ auf ein Ausgabeband schreibt.

- Richtung "$\Rightarrow$":
  - Arbeiten in Runden $i \in \mathbb{N}$.

Aufgabe 2.5

Zeige: Eine Sprache L ist genau dann rekursiv aufzählbar, wenn es eine Turingmaschine gibt, welche die Wörter $w \in L$ auf ein Ausgabeband schreibt.

- Richtung "$\Rightarrow$":
  - Arbeiten in Runden $i \in \mathbb{N}$.
  - In Runde i lassen wir M auf jedem Wort mit kanonischer Nummer $j \in \{1, ..., i\}$ jeweils $i - j + 1$ Schritte laufen.

Aufgabe 2.5

Zeige: Eine Sprache L ist genau dann rekursiv aufzählbar, wenn es eine Turingmaschine gibt, welche die Wörter $w \in L$ auf ein Ausgabeband schreibt.

- Richtung "$\Rightarrow$":
  - Arbeiten in Runden $i \in \mathbb{N}$.
  - In Runde i lassen wir M auf jedem Wort mit kanonischer Nummer $j \in \{1, ..., i\}$ jeweils $i - j + 1$ Schritte laufen.
  - Falls t Schritte zum Akzeptieren benötigt werden, wird w erstmals in Runde $j + t - 1$ akzeptiert.

Aufgabe 2.5

Zeige: Eine Sprache L ist genau dann rekursiv aufzählbar, wenn es eine Turingmaschine gibt, welche die Wörter $w \in L$ auf ein Ausgabeband schreibt.

- Richtung "$\Rightarrow$":
  - Arbeiten in Runden $i \in \mathbb{N}$.
  - In Runde i lassen wir M auf jedem Wort mit kanonischer Nummer $j \in \{1, ..., i\}$ jeweils $i - j + 1$ Schritte laufen.
  - Falls t Schritte zum Akzeptieren benötigt werden, wird w erstmals in Runde $j + t - 1$ akzeptiert.
  - Alle akzeptierten Wörter werden auf das Ausgabeband geschrieben.

Aufgabe 2.5

Zeige: Eine Sprache L ist genau dann rekursiv aufzählbar, wenn es eine Turingmaschine gibt, welche die Wörter $w \in L$ auf ein Ausgabeband schreibt.

- Richtung "$\Rightarrow$":
  - Arbeiten in Runden $i \in \mathbb{N}$.
  - In Runde i lassen wir M auf jedem Wort mit kanonischer Nummer $j \in \{1, ..., i\}$ jeweils $i - j + 1$ Schritte laufen.
  - Falls t Schritte zum Akzeptieren benötigt werden, wird w erstmals in Runde $j + t - 1$ akzeptiert.
  - Alle akzeptierten Wörter werden auf das Ausgabeband geschrieben.
- Richtung "$\Leftarrow$":

Aufgabe 2.5

Zeige: Eine Sprache L ist genau dann rekursiv aufzählbar, wenn es eine Turingmaschine gibt, welche die Wörter $w \in L$ auf ein Ausgabeband schreibt.

- Richtung "$\Rightarrow$":
  - Arbeiten in Runden $i \in \mathbb{N}$.
  - In Runde i lassen wir M auf jedem Wort mit kanonischer Nummer $j \in \{1, ..., i\}$ jeweils $i - j + 1$ Schritte laufen.
  - Falls t Schritte zum Akzeptieren benötigt werden, wird w erstmals in Runde $j + t - 1$ akzeptiert.
  - Alle akzeptierten Wörter werden auf das Ausgabeband geschrieben.
- Richtung "$\Leftarrow$":
  - Lassen Maschine Wörter auf das Ausgabeband schreiben.

Aufgabe 2.5

Zeige: Eine Sprache L ist genau dann rekursiv aufzählbar, wenn es eine Turingmaschine gibt, welche die Wörter $w \in L$ auf ein Ausgabeband schreibt.

- Richtung "$\Rightarrow$":
  - Arbeiten in Runden $i \in \mathbb{N}$.
  - In Runde i lassen wir M auf jedem Wort mit kanonischer Nummer $j \in \{1, ..., i\}$ jeweils $i - j + 1$ Schritte laufen.
  - Falls t Schritte zum Akzeptieren benötigt werden, wird w erstmals in Runde $j + t - 1$ akzeptiert.
  - Alle akzeptierten Wörter werden auf das Ausgabeband geschrieben.
- Richtung "$\Leftarrow$":
  - Lassen Maschine Wörter auf das Ausgabeband schreiben.
  - Wird w geschrieben, dann akzeptieren wir.

Aufgabe 2.5

Zeige: Eine Sprache L ist genau dann rekursiv aufzählbar, wenn es eine Turingmaschine gibt, welche die Wörter $w \in L$ auf ein Ausgabeband schreibt.

- Richtung "$\Rightarrow$":
  - Arbeiten in Runden $i \in \mathbb{N}$.
  - In Runde i lassen wir M auf jedem Wort mit kanonischer Nummer $j \in \{1, ..., i\}$ jeweils $i - j + 1$ Schritte laufen.
  - Falls t Schritte zum Akzeptieren benötigt werden, wird w erstmals in Runde $j + t - 1$ akzeptiert.
  - Alle akzeptierten Wörter werden auf das Ausgabeband geschrieben.
- Richtung "$\Leftarrow$":
  - Lassen Maschine Wörter auf das Ausgabeband schreiben.
  - Wird w geschrieben, dann akzeptieren wir.
  - Wenn $w \notin L$ wird w nie geschrieben.

Aufgabe 2.6

Aufgabe 2.6

Gib eine TM an, welche das Wort $x = (x_1, ..., x_n)$ auf das Band schreibt und mit dem Kopf auf x_1 stoppt.

Aufgabe 2.6

Gib eine TM an, welche das Wort $x = (x_1, ..., x_n)$ auf das Band schreibt und mit dem Kopf auf x_1 stoppt.

Turingmaschine M

δ	0	1	#
q_0	—	—	(q_1,x_n,L)
...	—	—	...
q_i	—	—	(q_{i+1},x_{n-i},L)
...	—	—	...
q_{n-1}	—	—	(q_{n-1},x_1,N)

Aufgabe 2.6

Gib eine TM an, welche das Wort $x = (x_1, ..., x_n)$ auf das Band schreibt und mit dem Kopf auf x_1 stoppt.

Turingmaschine M

δ	0	1	#
q_0	—	—	(q_1, x_n, L)
...	—	—	...
q_i	—	—	$(q_{i+1}, x_{n-i}, \mathrm{L})$
...	—	—	...
q_{n-1}	—	—	$(q_{n-1}, x_1, \mathrm{N})$

$\leftarrow$	$\leftarrow$		$\leftarrow$		$\leftarrow$	$\leftarrow$
x_1	x_2	$\cdots$	x_{n-i}	$\cdots$	x_{n-1}	x_n
q_{n-1}	q_{n-2}		q_i		q_1	q_0

Aufgabe 2.7

Aufgabe 2.7

Gib eine TM an, welche nur einen Zustand hat und nie stoppt.

Aufgabe 2.7

Gib eine TM an, welche nur einen Zustand hat und nie stoppt.

Turingmaschine *M*

δ	0	1	#
q_0	$(q_0,\#,R)$	$(q_0,\#,R)$	$(q_0,\#,R)$

Aufgabe 2.8

Aufgabe 2.8

Die Funktion $p : \{0, 1\}^* \longrightarrow \{0, 1\}$ sei wie folgt definiert:

$$p(x) = 1 \;\Leftrightarrow\; x \text{ ist eine Primzahl in Binärdarstellung.}$$

Ist für Programme (allgemein) entscheidbar, ob diese p berechnen?

Aufgabe 2.8

Theorem 1 (Satz von Rice)

- *Sei R die Menge der partiell berechenbaren Funktionen.*
- *Für alle $S \subseteq R$ sei $L_S = \{[M], M$ berechnet partiell ein $f \in S\}$.*
- *Für alle nichtleeren $S \subseteq R - \{u\}$ gilt $L_S \notin REK$.*

Aufgabe 2.8

Die Funktion $p : \{0,1\}^* \longrightarrow \{0,1\}$ sei wie folgt definiert:

$$p(x) = 1 \;\Leftrightarrow\; x \text{ ist eine Primzahl in Binärdarstellung.}$$

Ist für Programme (allgemein) entscheidbar, ob diese p berechnen?

Aufgabe 2.8

Die Funktion $p : \{0,1\}^* \longrightarrow \{0,1\}$ sei wie folgt definiert:

$$p(x) = 1 \;\Leftrightarrow\; x \text{ ist eine Primzahl in Binärdarstellung.}$$

Ist für Programme (allgemein) entscheidbar, ob diese p berechnen?

- Sei $S = \{p\}$.

Aufgabe 2.8

Die Funktion $p : \{0,1\}^* \longrightarrow \{0,1\}$ sei wie folgt definiert:

$$p(x) = 1 \;\Leftrightarrow\; x \text{ ist eine Primzahl in Binärdarstellung.}$$

Ist für Programme (allgemein) entscheidbar, ob diese p berechnen?

- Sei $S = \{p\}$.
- L_S ist nicht berechenbar.

Aufgabe 2.8

Die Funktion $p : \{0,1\}^* \longrightarrow \{0,1\}$ sei wie folgt definiert:

$$p(x) = 1 \iff x \text{ ist eine Primzahl in Binärdarstellung.}$$

Ist für Programme (allgemein) entscheidbar, ob diese p berechnen?

- Sei $S = \{p\}$.
- L_S ist nicht berechenbar.
- Es ist nicht berechenbar, ob ein gegebener Algorithmus korrekt das Primzahlproblem berechnet.

Aufgabe 2.8

Die Funktion $p : \{0,1\}^* \longrightarrow \{0,1\}$ sei wie folgt definiert:

$$p(x) = 1 \;\Leftrightarrow\; x \text{ ist eine Primzahl in Binärdarstellung.}$$

Ist für Programme (allgemein) entscheidbar, ob diese p berechnen?

- Sei $S = \{p\}$.
- L_S ist nicht berechenbar.
- Es ist nicht berechenbar, ob ein gegebener Algorithmus korrekt das Primzahlproblem berechnet.

Es existiert keine Turingmaschine mit Eingaben der Form $[M]$, die stets hält und genau dann akzeptiert, wenn die Eingabe $[M]$ das Programm einer Turingmaschine M ist, die korrekt das Primzahlproblem berechnet.

Vielen Dank für die Aufmerksamkeit.