Vertiefung Theoretische Informatik, SS 2025 — 4. Übungsblatt

Abgabe bis Freitag, 6. Juni 2025, 12:10 Uhr, online im Whiteboard oder zu Beginn der Vorlesung.

---

21. Eingebaute universelle Turingmaschine, 10 Punkte

    In PYTHON gibt es eine eingebaute Funktion *exec*, die die umgekehrte Rolle von *repr* spielt: *exec*(s) führt in der Zeichenkette s enthaltenen PYTHON-Anweisungen aus.

    Wie kann man die selbstdruckenden Programme und das Program für den Rekursionssatz (`make-H-from-h.py`) mit Hilfe der *exec*-Funktion kürzer machen?

22. Fixpunktsatz, 10 Punkte

    Bestimmen Sie für jeder der folgenden Funktionen $f\colon \mathcal{T} \to \mathcal{T}$ die Beschreibung einer Turingmaschine s, die ein Fixpunkt von f im Sinne des Fixpunktsatzes ist.

    (a) Die Funktion f vertauscht den akzeptierenden Zustand q_{ja} mit dem nicht akzeptierenden Zustand q_{nein}.

    (b) $f(s)$ ist (eine Beschreibung der) folgenden Turingmaschine M': M' simuliert die Turingmaschine $M = \text{TM}(s)$ und zählt die Übergänge mit. Wenn M nach 100 Schritten noch nicht hält, wird die Simulation abgebrochen. In jedem Fall (nach dem normalen Halten von M innerhalb der ersten 100 Schritte oder nach dem Abbruch) wird das Band am Ende aufgeräumt, sodass nur der aktuelle Bandinhalt der simulierten Maschine M' auf dem Band stehenbleibt.

    (c) Wie (b), aber wenn M innerhalb der ersten 99 Schritte hält, geht die Maschine M' in eine unendliche Schleife.

23. Zählermaschine, 0 Punkte

    Stellen Sie die nebenstehende Übergangsfunktion $\delta\colon Q \times \{0,+\}^2 \to (Q \times \{0,+1,-1\}^2) \cup \{\text{Halt}\}$ für eine Zählermaschine mit zwei Zählern `X` und `Y` durch ein `GOTO`-Programm dar.

q_0	0	0	q_0	+1	0
q_0	0	+	q_1	+1	0
q_0	+	0	q_0	+1	0
q_0	+	+	q_1	+1	−1
q_1	0	0	q_0	0	0
q_1	0	+	q_1	0	−1
q_1	+	0	Halt		
q_1	+	+	q_0	+1	+1

24. FRACTRAN, 0 Punkte

    Ein Programm in der von John Conway erfundenen Programmiersprache FRACTRAN ist durch eine Folge $\alpha_1, \alpha_2, \ldots, \alpha_k$ von positiven rationalen Zahlen gegeben. Die Eingabe ist eine positive natürliche Zahl x.

    In jedem Schritt werden der Reihe nach die Produkte $x\alpha_1, x\alpha_2, \ldots$ gebildet, bis das Ergebnis zum ersten Mal eine ganze Zahl ist. Dann wird x durch diese Zahl ersetzt und von neuem begonnen. Das Programm hält, wenn keines der Produkte eine ganze Zahl liefert.

    (a) Simulieren Sie das Programm $5/3, 2/5$ mit der Eingabe $x = 72$.

    (b) Was ist das Ergebnis für $x = 2^a 3^b$?

    (c) Was ist das Ergebnis des Programms $9/2$ (mit nur einem Bruch, $k = 1$) für eine Eingabe der Form $x = 2^a 3^b$?

    (d) Was ist das Ergebnis des Programms $9/2, 2/3$ für die Eingabe $x = 666$?

    (e) Was ist das Ergebnis des Programms $5/6, 5/2, 5/3$ für die Eingabe $x = 2^a 3^b$?

    (f) Wie ist es für ein Eingabe der Form $x = 2^a 3^b 35^2$?

25. Halteproblem für FRACTRAN, 10 Punkte

    Zeigen Sie, dass das Halteproblem für FRACTRAN-Programme unentscheidbar ist.

    Hinweis: Sie können eine Zählermaschine in Primzahldarstellung simulieren. Verwenden Sie für jeden Zustand eine zusätzliche eigene Primzahl.

    Beachten Sie, dass Zähler und Nenner gekürzt wird, wenn Sie gleichzeitig durch einen Faktor p dividieren und mit p multiplizieren wollen. Wie kann man dieses Problem umgehen?

26. Satz von Rice, Anwendung, 0 Punkte

    Zeigen Sie, dass es unentscheidbar ist, ob zwei Beschreibungen $s, t \in \mathcal{T}$ äquivalente Turingmaschinen beschreiben, in dem Sinn, dass $\mathrm{TM}(s)$ und $\mathrm{TM}(t)$ für die gleichen Eingaben halten, und in diesem Fall auch die gleichen Ergebnisse auf das Band geschrieben haben.

27. Satz von Rice, alternativer Beweisanzatz, 0 Punkte

    Es sei M_0 eine feste Turingmaschine.

    (a) Für $s \in \mathcal{T}$ betrachten wir die Turingmaschine $M_1 := M_1(s)$, die bei Eingabe von $x \in \Sigma^*$ folgendes durchführt.

    - Simuliere $\mathrm{TM}(s)$ mit leerer Eingabe.
    - Wenn $\mathrm{TM}(s)$ hält, simuliere M_0 mit Eingabe x.
    - Wenn M_0 die Eingabe x akzeptiert, akzeptiere x.

    Was ist die von M_1 akzeptierte Sprache? (Die Antwort hängt von s ab.)

    (b) Nehmen wir an, wir könnten mit einem Algorithmus A für jedes $t \in \mathcal{T}$ entscheiden, ob die von $\mathrm{TM}(t)$ akzeptierte Sprache zu einer gegebenen Klasse $\mathbb{L}$ von Sprachen gehört. Nehmen wir zusätzlich an, dass $L(M_0) \in \mathbb{L}$ und $\emptyset \notin \mathbb{L}$ ist.

    Dann könnte man auf folgende Art das Halteproblem für Turingmaschinen mit leerer Eingabe lösen:

    - Für gegebenes $s \in \mathcal{T}$ konstruiere die Turingmaschine $M_1 := M_1(s)$ von Aufgabe (a) und wende Algorithmus A auf ihre Beschreibung $t = \langle M_1 \rangle$ an. Wir wissen dann: $\mathrm{TM}(s)$ hält genau dann, wenn $L(M_1) \in \mathbb{L}$ ist.

    (c) Wie kann man vorgehen, wenn die beiden zusätzlichen Annahmen in Aufgabe (b) nicht gelten?