Formale Grundlagen der Informatik I
7. Übungsblatt

TECHNISCHE UNIVERSITÄT DARMSTADT

Fachbereich Mathematik
Prof. Dr. Martin Ziegler
Davorin Lešnik, Ph.D.
Carsten Rösnick

Sommersemester 2013
27. 05. 2013

Gruppenübung

Aufgabe G16 (Konstruktion von Turingmaschinen)
Beschreiben Sie formal (d. h. unter Angabe der Komponenten Σ, Q, δ) eine 1-Band Turingmaschine für die folgende Funktion: Bei Eingabe der Binärdarstellung einer Zahl $n \in \mathbb{N}$ soll die Zahl $2 \cdot n$ (wieder in Binärdarstellung) ausgegeben werden.

Tipp: Nehmen Sie zur Vereinfachung an, dass Sie die Zahl n in der Form $b_0 b_1 \ldots b_k$ (von links nach rechts geschrieben) auf dem Band vorliegen haben, wobei $n = \sum_{i=0}^{k} b_i \cdot 2^i$, $b_i \in \{0,1\}$.

Lösung:

(a) Komplizierte Lösung:
Definiere $\mathscr{M} = (\Sigma, Q, q_0, \delta, q^+, q^-)$ durch

$$\Sigma = \{0,1\},$$
$$Q = \{q_0, q^+, q^-, q_a, q_b\}$$

mit folgender Übergangsfunktion:

δ	$\square$	0	1
q_0	$(\square, >, q_0)$	$(0, >, q_a)$	$(0, >, q_b)$
q_a	$(\square, \circ, q^+)$	$(0, >, q_a)$	$(0, >, q_b)$
q_b	$(1, \circ, q^+)$	$(1, >, q_a)$	$(1, >, q_b)$

(b) Einfache Lösung: Schreibe an aktueller Position eine 0 und gehe in den akzeptierenden Zustand über.

Aufgabe G17 (Simulation von Turingmaschinen)
Beschreiben Sie informell, wie eine deterministische Turingmaschine $\mathscr{M}_0$ für jeden der nachfolgenden Punkte konstruiert werden kann. Geben Sie die dazu benötigten Zustände von $\mathscr{M}_0$ an und beschreiben Sie deren Funktion.

(a) Gegeben eine feste Zeichenkette $c = c_1 \ldots c_k \in \Sigma^*$. Die Maschine $\mathscr{M}_0$ lösche die Eingabe, schreibe c auf das Eingabeband, fahre zurück an die Startposition und halte.

(b) Gegeben eine feste Turingmaschine $\mathscr{M}$ und eine feste Zeichenkette $c \in \Sigma^*$. Die Maschine $\mathscr{M}_0$ lösche ihre Eingabe und rechne anschliessend weiter wie die Maschine $\mathscr{M}$ bei Eingabe der Konstanten c.

(c) Gegeben eine feste Turingmaschine $\mathcal{M}$. Die Maschine $\mathcal{M}_0$ simuliere $\mathcal{M}$ auf Eingabe $\langle \mathcal{M} \rangle$ so dass

$$\begin{aligned} w \xrightarrow{\mathcal{M}_0} \infty \quad &\Leftrightarrow \quad \langle \mathcal{M} \rangle \xrightarrow{\mathcal{M}} q^+, \\ w \xrightarrow{\mathcal{M}_0} q^- \quad &\Leftrightarrow \quad \langle \mathcal{M} \rangle \xrightarrow{\mathcal{M}} q^- \end{aligned}$$

für alle Eingaben $w \in \Sigma^*$ der Maschine $\mathcal{M}_0$. Sie dürfen hier verwenden, dass für alle Turingmaschinen $\mathcal{M}$ solch eine Kodierung $\langle \mathcal{M} \rangle$ existiert[1].

Lösung:

(a) Wir benötigen die Zustände $q_0, q^+, q_{c_1}, \dots, q_{c_k}, q^{c\rightarrow}, q^{c\leftarrow}, q^{\leftarrow c}$ mit folgender Bedeutung:

- q_0 Startzustand
- q^+ Akzeptierender Endzustand
- q_{c_i} Zustand, in dem der aktuelle Zelleninhalt mit c_i überschrieben und zum Zustand $q_{c_{i+1}}$ übergegangen wird, falls $i < k$, sonst wird zu $q^{c\rightarrow}$ übergegangen.
- $q^{c\rightarrow}$ Lösche den Rest des Eingabewortes vom Band indem so lange nach rechts gefahren und $\square$ geschrieben wird, bis das erste $\square$ erreicht ist. Gehe dann in den Zustand $q^{c\leftarrow}$ über.
- $q^{c\leftarrow}$ Fahre nach links, solange $\square$ gelesen wird, dann (also am Ende von c) gehe in den Zustand $q^{\leftarrow c}$ über.
- $q^{\leftarrow c}$ Fahre nach links, solange ein Zeichen aus Σ (d. h. nicht $\square$) gelesen wird; bei dem ersten gelesenen $\square$ (also genau ein Zeichen vor c) gehe in den Zustand q^+ über.

Das funktioniert im Fall $k > 0$; im Fall $k = 0$ (also $c = \varepsilon$) wähle beliebiges $x \in \Sigma$, mache alles wie oben für $c = x$ und am Ende lösche noch dieses x.

(b) Sei $\mathcal{M} = (\Sigma, Q, q'_0, \delta', q'^+, q'^-)$. Wir verwenden die (disjunkte) Vereinigung von Q mit den Zuständen aus (a). Die Zustände aus (a) sind nun mit folgender geänderter Bedeutung belegt:

- $q^{\leftarrow c}$ Fahre zur Anfangsposition des Bandes und (im Unterschied zur Maschine in (a)) gehe in den Startzustand q'_0 der Maschine $\mathcal{M}$ über.
- q'_0 Da der Lesekopf der Maschine $\mathcal{M}_0$ auf der Startposition, sowie c auf dem Band steht, verfahre $\mathcal{M}_0$ nun wie $\mathcal{M}$, indem es die Übergangsfunktion von $\mathcal{M}$ ausführt.
- q'^+ Gehe in den Zustand q^+ über.
- q'^- Gehe in den Zustand q^- über.

(c) Wie in (b) mit $c = \langle \mathcal{M} \rangle$, allerdings mit dem zusätzlichen Zustand q^∞, so dass q'^+ in den Zustand q^∞ übergehe. Im Zustand q^∞ gelte $\delta_{\mathcal{M}_0}(q^\infty, z) = (z, \circ, q^\infty)$ für alle $z \in (\Sigma_{\mathcal{M}_0} \cup \{\square\})$.

Aufgabe G18 (Entscheidbarkeit)
Beweisen oder widerlegen Sie:

(a) Die Sprache $L_k := \{ \langle M, w \rangle \mid M \text{ akzeptiert } w \text{ nicht innerhalb von } k \text{ Schritten} \}$ für eine Konstante $k \in \mathbb{N}$ ist entscheidbar.

(b) Die Sprache $L_A := \{ M \mid \forall\, w \in \Sigma^* .\, M \text{ hält bei Eingabe } w \}$ ist entscheidbar.

Lösung:

(a) Das Problem ist für jedes feste k entscheidbar. Konstruiere dazu eine Turingmaschine $\mathcal{M}'$ wie folgt: Bei Eingabe $\langle \mathcal{M}, w \rangle$, simuliere die Maschine $\mathcal{M}$ auf dem Eingabewort w für genau k Schritte. Akzeptiert $\mathcal{M}$ innerhalb dieser Simulationszeit, so verwerfe $\mathcal{M}'$. Akzeptiert $\mathcal{M}$ jedoch nicht innerhalb von k Schritten das Wort w, so akzeptiere $\mathcal{M}'$.

[1] siehe auch: Gödelnummer (http://de.wikipedia.org/wiki/Gödelnummer)

(b) Die Sprache L_A ist *nicht entscheidbar*, was wir durch die Reduktion auf das (semientscheidbare, aber *nicht entscheidbare*) Halteproblem beweisen.

Angenommen, die Sprache L_A ist entscheidbar. Dann lässt sich aus einer Maschine $\mathscr{M}_A$ wie folgt eine Turingmaschine $\mathscr{M}_H$ konstruieren, die das Halteproblem entscheidet (das entspricht Beispiel 4.3.6 im Skript):

- Die Eingabe sei die Kodierung einer Turingmaschine $\mathscr{M}$: $x = \langle M \rangle$.
- Berechne die Kodierung $\langle \mathscr{M}' \rangle$ einer Maschine $\mathscr{M}'$, die
  - ihre Eingabe vom Band löscht,
  - dafür das Wort $\langle \mathscr{M} \rangle$ auf das Band schreibt
  - und anschließend die Berechnung gemäß der Übergangsfunktion von $\mathscr{M}$ fortsetzt.
- Simuliere nun die Maschine $\mathscr{M}_A$ auf der Eingabe $\langle \mathscr{M}' \rangle$. Akzeptiere genau dann, wenn die Simulation akzeptiert; verwerfe genau dann, wenn die Simulation verwirft.

Die Maschine $\mathscr{M}'$ dient dazu, die Eingabe $\langle \mathscr{M} \rangle$ in eine Instanz des Halteproblems umzubauen.

Hausübung

Aufgabe H16 (Entscheidbarkeit) (5 Punkte)
Wir stellen uns die Frage nach den Konsequenzen der Unentscheidbarkeit des in der Vorlesung vorgestellten Halteproblems. Dazu betrachten wir das folgende Problem:

> *Sei $\mathscr{P}$ ein beliebiges* Java*-Programm ohne Eingabe, wobei $\mathscr{P}$ mindestens $\ell \in \mathbb{N}$ Codezeilen enthalte. Wird Zeile ℓ jemals ausgeführt?*

Beweisen Sie, dass es keinen Algorithmus gibt, der dieses Problem für beliebige $\mathscr{P}$ und ℓ entscheidet.

Lösung: Angenommen es gäbe einen solchen Algorithmus. Betrachte das folgende Programm (im Pseudocode):

$$\begin{array}{rl} (*) & \texttt{ruf } \mathscr{P} \texttt{ auf} \\ (**) & \texttt{drucke } \text{"}\mathscr{P} \text{ hat terminiert!"} \end{array}$$

Durch Entscheiden, ob die Zeile $(**)$ ausgeführt wird, können wir für ein beliebiges Java-Programm (ohne Eingabe) $\mathscr{P}$ entscheiden, ob es terminiert. Denn Java ist Turing-vollständig, würden wir dadurch das Halteproblem entscheiden. Widerspruch.

Aufgabe H17 (Simulation von Turingmaschinen) (5 Punkte)
Mithilfe des Turingmaschinensimulators

`http://db.ing.puc.cl/turingmachine/`

implementieren Sie die folgenden Turingmaschinen:

(a) Die Turingmaschine aus **Aufgabe G16**.

(b) Die Turingmaschine, die bei Eingabe der Binärdarstellung einer Zahl $n \in \mathbb{N}$ die Zahl $n+1$ (wieder in Binärdarstellung) ausgibt.

(c) Die Turingmaschine, die Ihre Matrikelnummer in Binärform auf das Band schreibt (Sie können annehmen, dass das Band am Anfang leer ist). Versuchen Sie die Maschine mit möglichst kleiner Anzahl der Zustände zu finden.

Sie können annehmen, dass alle Binärzahlen umgekehrt (wie in **Aufgabe G16**) angegeben sind.

Bemerkung: In diesem Simulator fängt der Lesekopf auf dem ersten Zeichen der Eingabe an, statt links vor der Eingabe, wie im Skript!

Hinweis: Als Lösung jeder Teilaufgabe zählt die Abgabe des Programmcodes in Moodle in **Übungsblatt 7**. Dort finden Sie (nach dem Klick auf „Abgabe hinzufügen“) einen Editor, in dem Sie Ihre Lösungen einfügen können.

Lösung:

Teilaufgabe (a) $\boxed{\text{1 P.}}$	Teilaufgabe (b) $\boxed{\text{2 P.}}$	Teilaufgabe (c) $\boxed{\text{2 P.}}$

`q0,0`
`q1,0,<`
 	
`q0,1`
`q1,1,<`
 	
`q1,_`
`q+,0,-`
	`q0,0`
	`q+,1,-`
	
	`q0,1`
	`q1,1,-`
	
	`q1,_`
	`q+,1,-`
	
	`q1,0`
	`q+,1,-`
	
	`q1,1`
	`q1,0,>`
		Die naive Lösung ist klar. ☺

Aufgabe H18 (Verallgemeinerung von Turingmaschinen) (5 Punkte)

In dieser Aufgabe schwächen wir die Definition der Turingmaschine so ab, dass wir nach rechts *unendlich viele* Symbole aus Σ als Eingabe erlauben.

Sei $\Sigma = \{0, 1\}$. Das gegebene Problem ist, ob es irgendwo in der Eingabe eine 0 gibt.

(a) Ist dieses Problem *semientscheidbar*, im Sinne, dass eine solch verallgemeinerte Turingmaschine existiert, die genau die Zeichenfolgen mit mindestens einer 0 akzeptiert?

(b) Ist dieses Problem *entscheidbar*, im Sinne, dass eine solch verallgemeinerte Turingmaschine existiert, die auf jeder Eingabe terminiert und genau die Zeichenfolgen mit mindestens einer 0 akzeptiert?

Beweisen Sie Ihre Antworten.

Lösung:

(a) $\boxed{\text{2 P.}}$ Das Problem ist semientscheidbar. Die Idee ist, man geht einfach durch die Zeichenfolge und sobald man eine 0 beobachtet, terminiert man im akzeptierenden Zustand. Also sieht die Übergangsfunktion der entsprechenden Turingmaschine so aus:

δ	$\square$	0	1
q_0	$(\square, >, q_0)$	$(0, \circ, q^+)$	$(1, >, q_0)$

(b) $\boxed{\text{3 P.}}$ Das Problem ist nicht entscheidbar; wir zeigen das durch Überführen auf das Halteproblem. Um zu entscheiden, ob eine (übliche) Turingmaschine $\mathscr{M}$ auf $\langle \mathscr{M} \rangle$ terminiert, produziere algorithmisch die folgende unendliche Zeichenfolge: der Term auf k-ter Stelle ist 1, wenn $\mathscr{M}$ auf $\langle \mathscr{M} \rangle$ in k Schritten noch nicht terminiert hat (vgl. **Aufgabe G18**), sonst 0. Zu entscheiden, ob diese Folge eine 0 hat, ist äquivalent zum Entscheiden, ob $\mathscr{M}$ auf $\langle \mathscr{M} \rangle$ terminiert.

Minitest

Aufgabe M18 ((Semi-)Entscheidbarkeit)

Entscheiden Sie für die folgenden Sprachen, ob sie (semi-)entscheidbar sind:

(a) beliebige kontextfreie Sprache
☐ entscheidbar ☐ semientscheidbar, aber nicht entscheidbar ☐ nicht semientscheidbar

(b) die Sprache $\{a^n b^n c^n \mid n \in \mathbb{N}\}$
☐ entscheidbar ☐ semientscheidbar, aber nicht entscheidbar ☐ nicht semientscheidbar

(c) das Halteproblem $H \subseteq \Sigma^*$
☐ entscheidbar ☐ semientscheidbar, aber nicht entscheidbar ☐ nicht semientscheidbar

(d) das Komplement des Halteproblems $\overline{H} = \Sigma^* \setminus H$
☐ entscheidbar ☐ semientscheidbar, aber nicht entscheidbar ☐ nicht semientscheidbar

Lösung:

(a) beliebige kontextfreie Sprache
☒ entscheidbar ☐ semientscheidbar, aber nicht entscheidbar ☐ nicht semientscheidbar
Begründung: Per Satz 4.4.2 aus dem Skript ist jede kontextsensitive und somit auch jede kontextfreie Sprache entscheidbar.

(b) die Sprache $\{a^n b^n c^n \mid n \in \mathbb{N}\}$
☒ entscheidbar ☐ semientscheidbar, aber nicht entscheidbar ☐ nicht semientscheidbar
Begründung: Wir wissen bereits, dass diese Sprache kontextsensitiv ist, also ist sie wieder mithilfe des Satzes 4.4.2 aus dem Skript entscheidbar.

(c) das Halteproblem $H \subseteq \Sigma^*$
☐ entscheidbar ☒ semientscheidbar, aber nicht entscheidbar ☐ nicht semientscheidbar
Begründung: Das Halteproblem wurde in der Vorlesung bereits als nicht entscheidbar bewiesen. Die Semientscheidbarkeit ist sehr leicht einzusehen: Eine Turingmaschine für H muss lediglich alle Turingmaschinen $\mathscr{M}$ *akzeptieren*, die gestartet mit ihrem eigenen Code $\langle \mathscr{M} \rangle$ halten und akzeptieren. H kann wie folgt akzeptiert werden: Gegeben $\langle \mathscr{M} \rangle$, simuliere $\mathscr{M}$ auf der Eingabe $\langle \mathscr{M} \rangle$ und akzeptiere genau dann, wenn die Simulation hält und akzeptiert.

(d) das Komplement des Halteproblems $\overline{H} = \Sigma^* \setminus H$
☐ entscheidbar ☐ semientscheidbar, aber nicht entscheidbar ☒ nicht semientscheidbar
Begründung: Wäre $\overline{H}$ semientscheidbar, dann würde zusammen mit der Semientscheidbarkeit von H die Entscheidbarkeit des *unentscheidbaren* Halteproblems folgen.

Aufgabe M19 (Turingmaschinen)
Gegeben sei die Turingmaschine $\mathscr{M}$ über dem Alphabet $\{a, b\}$ mit Startzustand q_0 und folgender Übergangsfunktion:

δ	$\Box$	a	b
q_0	$(\Box, >, q_a)$		
q_a	$(\Box, \circ, q^+)$	$(a, >, q_b)$	$(b, \circ, q^-)$
q_b	$(\Box, \circ, q^-)$	$(a, \circ, q^-)$	$(b, >, q_a)$

Kreuzen Sie alle richtigen Aussagen an.

- ☐ $\mathscr{M}$ erkennt die Sprache $\mathtt{L}((\mathtt{a}+\mathtt{b})^*)$.
- ☐ $\mathscr{M}$ erkennt die Sprache $\mathtt{L}((\mathtt{ab})^*)$.
- ☐ Die von $\mathscr{M}$ erkannte Sprache kann nicht durch einen regulären Ausdruck beschrieben werden.
- ☐ $\mathscr{M}$ hält immer.
- ☐ $\mathscr{M}$ hält nicht bei jeder Eingabe.
- ☐ $\mathscr{M}$ hält bei keiner Eingabe.

Lösung:

- ☐ $\mathscr{M}$ erkennt die Sprache $\mathtt{L}((\mathtt{a}+\mathtt{b})^*)$.
- ☒ $\mathscr{M}$ erkennt die Sprache $\mathtt{L}((\mathtt{ab})^*)$.
- ☐ Die von $\mathscr{M}$ erkannte Sprache kann nicht durch einen regulären Ausdruck beschrieben werden.
- ☒ $\mathscr{M}$ hält immer.
- ☐ $\mathscr{M}$ hält nicht bei jeder Eingabe.
- ☐ $\mathscr{M}$ hält bei keiner Eingabe.