Grundbegriffe der Informatik — Aufgabenblatt 11
Lösungsvorschläge

Matr.nr.:	
Nachname:	
Vorname:	
Tutorium Nr.:	Tutor*in:

Ausgabe: 21. Januar 2021, 12:00 Uhr

Abgabe: 28. Februar 2021, 12:30 Uhr
in dem Holzkasten neben Raum -119
im UG des Info-Gebäudes (50.34)

Lösungen werden nur korrigiert, wenn sie
- handschriftlich erstellt sind (Tablet-Ausdruck erlaubt) und
- mit dieser Seite als Deckblatt
- in der oberen **linken** Ecke zusammengeheftet

rechtzeitig abgegeben werden.

Abgaberegeln für Teilnehmer der Online-Tutorien:
- handschriftlich erstellt (lesbare Fotos akzeptiert)
- **rechtzeitig**, mit diesem Deckblatt in **genau einer** PDF-Datei
- direkt an den entsprechenden Tutor abgeben.

---

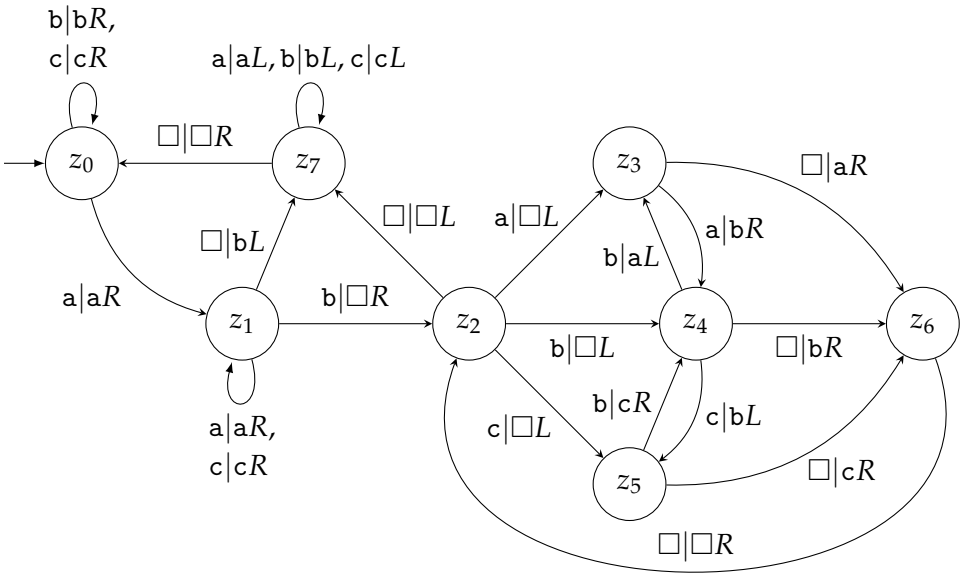
*Von Tutor*in auszufüllen:*

erreichte Punkte

Blatt 11:	/ 16
Blätter 7 – 11:	/ 102

Hinweis: Für alle Aufgaben gilt: Halten Sie ihre Lösungen übersichtlich.
Kommentieren Sie im Zweifelsfall die Funktionsweise ihrer Turingmaschinen.
Die Korrektur unübersichtlicher Lösungen kann abgelehnt werden!

Aufgabe 11.1 (0,5 + 1 + 2,5 + 1 = 5 Punkte)
Gegeben folgende Turingmaschine T, mit $Z = \{z_i \mid i \in \mathbb{Z}_8\}$ und $X = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \square\}$:



a) Geben Sie die ersten 7 Schritte der Berechnung von T bei Eingabe „abca“ an. Nutzen Sie hierfür das Format der Beispielrechnungen aus den Vorlesungsfolien (Kapitel 16, Folie 24). Sie brauchen den Berechnungszustand nur nach Schritten angeben, in denen sich mindestens ein Bandsymbol geändert hat.

b) Terminiert die Berechnung von T für jede Eingabe $w \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}^*$? Begründen Sie.

c) Eigentlich sollte die Turingmaschine für jede Eingabe $w \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}^*$ folgende Berechnung durchführen:
   - Jedes „a“ in w wird gelöscht.
   - Für jedes gelöschte „a“ in w wird das linkeste „b“ in w gelöscht. Sofern kein „b“ in w vorhanden, wird w stattdessen hinten um ein „c“ ergänzt.
   - Die Reihenfolge nicht gelöschter Zeichen bleibt unverändert und die Zeichen des Ergebniswortes sind am Ende der Berechnung nicht durch „□“ getrennt.

   Korrigieren Sie T, sodass T wie beschrieben arbeitet. Hierfür dürfen Sie ausschließlich neue Zustände und Übergänge hinzufügen und oder Übergänge (keine Zustände) entfernen. Fügen Sie nicht mehr als 7 Zustände hinzu und entfernen Sie alle *überflüssigen* Übergänge. Ein Übergang U ist *überflüssig*, wenn keine Eingabe $w' \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}^*$ existiert, sodass U bei Berechnung mit w' verwendet wird.
   Anmerkung: Teilpunkte für Lösungen mit > 7 neuen Zuständen.

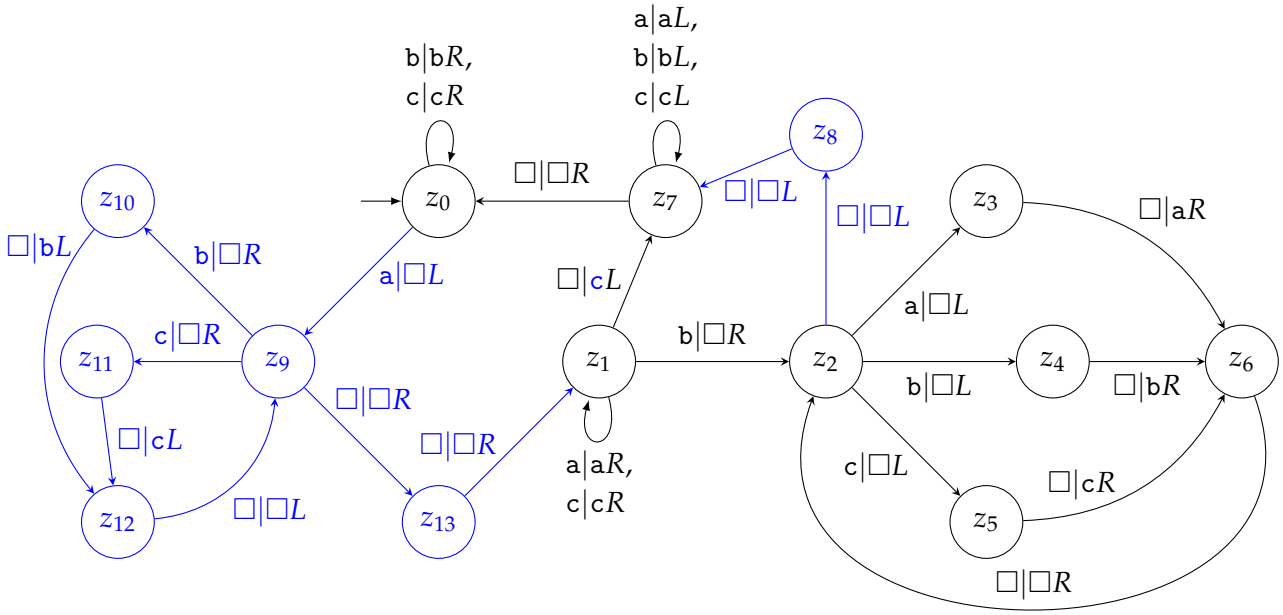
d) Geben Sie die Berechnung ihrer korrigierten Turingmaschine aus Teilaufgabe c) bei Eingabe „cbaaba“ an. Verwenden Sie das Format aus a). Geben Sie alle Berechnungszustände an, bei denen sich die Bandbeschriftung verändert.

Lösung 11.1

a) (0) $\square\overset{z_0}{\mathtt{a}}\mathtt{bca}\square$
(2) $\square\mathtt{a}\square\overset{z_2}{\mathtt{c}}\mathtt{a}\square$
(3) $\square\mathtt{a}\overset{z_5}{\square}\square\mathtt{a}\square$
(4) $\square\mathtt{ac}\overset{z_6}{\square}\mathtt{a}\square$
(6) $\square\mathtt{ac}\overset{z_3}{\square}\square\square$
(7) $\square\mathtt{aca}\overset{z_6}{\square}\square$

b) Ja. Wenn sich T in z_6 befindet, befindet sich der Lesekopf stets auf einem „□“ hinter dem letzten Zeichen des Eingabewortes. Somit passieren dann stets die folgenden Übergänge: $z_6 \overset{R}{\to} z_2 \overset{L}{\to} z_7 \overset{R}{\to} z_0$, bei denen stets „□“ gelesen wird und schlussendlich landet der Lesekopf auf einem „□“ und T in Zustand z_0. Für diese Konfiguration ist kein Übergang definiert.

c)



d) Lösung stark abhängig von der Lösung von Teilaufgabe c):
Berechnung nach Musterlösung von c). Spaltenweise zu lesen.

$\square\overset{z_0}{\mathtt{c}}\mathtt{baaba}\square$
$\square\mathtt{c}\overset{z_9}{\mathtt{b}}\square\mathtt{aba}\square$
$\square\mathtt{c}\overset{z_{10}}{\square}\square\mathtt{aba}\square$
$\square\mathtt{c}\overset{z_{12}}{\square}\mathtt{baba}\square$
$\square\square\overset{z_{11}}{\square}\mathtt{baba}\square$
$\square\square\overset{z_{12}}{\mathtt{c}}\mathtt{baba}\square$
$\square\square\mathtt{c}\overset{z_2}{\square}\mathtt{aba}\square$

$\square\square\mathtt{c}\overset{z_3}{\square}\square\mathtt{ba}\square$
$\square\square\mathtt{ca}\overset{z_6}{\square}\mathtt{ba}\square$
$\square\square\mathtt{ca}\overset{z_4}{\square}\square\mathtt{a}\square$
$\square\square\mathtt{cab}\overset{z_6}{\square}\mathtt{a}\square$
$\square\square\mathtt{cab}\overset{z_3}{\square}\square\square$
$\square\square\mathtt{caba}\overset{z_6}{\square}\square$

$\square\square\overset{z_9}{\mathtt{c}}\square\mathtt{ba}\square\square$
$\square\square\square\overset{z_{11}}{\square}\mathtt{ba}\square\square$
$\square\square\square\overset{z_{12}}{\mathtt{c}}\mathtt{ba}\square\square$
$\square\square\square\mathtt{c}\overset{z_2}{\square}\mathtt{a}\square\square$
$\square\square\square\mathtt{c}\overset{z_3}{\square}\square\square\square$
$\square\square\square\mathtt{ca}\overset{z_6}{\square}\square\square$
$\square\square\square\overset{z_9}{\mathtt{c}}\square\square\square\square$

$\square\square\square\square\overset{z_{11}}{\square}\square\square\square$
$\square\square\square\square\mathtt{c}\overset{z_{12}}{\square}\square\square$
$\square\square\square\square\overset{z_7}{\mathtt{c}}\mathtt{c}\square\square$
$\square\square\square\square\mathtt{cc}\overset{z_0}{\square}\square$

Aufgabe 11.2 (2 + 2 = 4 Punkte)

a) Definieren Sie (formal vollständig) einen Turingmaschinenakzeptor T mit $L(T) = \{w \in \{\mathtt{a},\mathtt{b},\mathtt{c}\}^* \mid \mathcal{N}_{\mathtt{a}}(w) + \mathcal{N}_{\mathtt{b}}(w) \geq \mathcal{N}_{\mathtt{c}}(w)\}$.
Stellen Sie f, g und m grafisch dar. T soll höchstens 4 Zustände haben. Beschreiben Sie die Funktion (also die Aufgabe) jedes Zustands knapp.
Anmerkung: Turingmaschinen mit mehr Zuständen erhalten Teilpunkte

b) Ihnen ist sicher aufgefallen, dass die „korrigierte" Turingmaschine T aus Aufgabe 11.1c) unübersichtlich viele Zustände hat. Wofür wird ein Großteil dieser Zustände benötigt? Wie könnte man eine Turingmaschine T' konstruieren, welche die selben Ausgaben wie T berechnet, jedoch weniger Zustände benötigt?
Beschreiben Sie präzise das Vorgehen einer solchen Turingmaschine.
generisches Beispiel einer Vorgehensbeschreibung:
Wenn T' ein [x] liest, macht sie statt [Verhalten von T] [alternatives Verhalten von T'] ...

Lösung 11.2

$T = (Z, z_c, X, f, g, m, F)$ mit:

- $Z = \{z_c, z_{a/b}, z_{l1}, z_{l2}\}$
- Startzustand z_c
- $X = \{\mathtt{a},\mathtt{b},\mathtt{c}, -, \square\}$
- $F = \{z_c\}$

und f, g, m siehe Grafik: ($X_{\setminus\square} = X \setminus \{\square\}$)

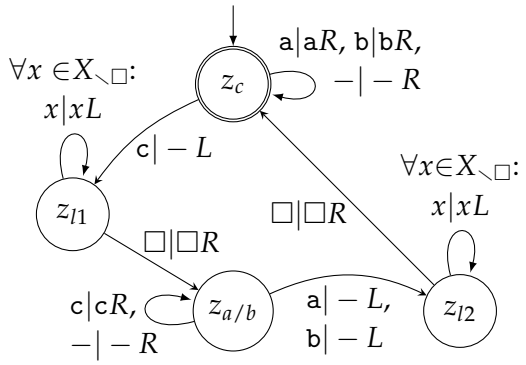
Zustandsbeschreibung:

z_c: Spult nach rechts über das Band, bis zum Auftreten eines „c". Löscht das erste „c" und wechselt zu $z_{a/b}$.

$z_{a/b}$: Spult nach rechts über das Band bis zum Auftreten eines „a" oder „b". Löscht das erste „a" oder „b" und wechselt zu z_{l2}.

z_{l1}, z_{l2}: Spult nach Links über das Band bis zum Wortanfang. Wechselt dann zurück zu z_c.

a)



b) Die meisten Zustände der Turingmaschine aus 11.1c) *implementieren* eine Funktionalität, die es erlaubt, Lücken innerhalb des Wortes auf dem Band zu schließen, indem das Wort entweder von rechts oder links „nachrückt". Da dies zu verschiedenen Zeitpunkten in der Berechnung passiert (nach dem Löschen eines „a", sowie nach dem löschen eines „b") ist diese Funktionalität „doppelt" implementiert. Sofortiges Schließen von Lücken im Wort ist essenziell, damit das Blanksymbol („□") ein eindeutiges Zeichen für Wortanfang /- Ende auf dem Band darstellt. Dies kann allerdings auch erreicht werden, indem das Löschen eines „a" oder „b" jeweils durch überschreiben mit einem eindeutigen Bandsymbol, (z.b. „−") passiert, welches erst nachträglich aus dem Wort via „Nachrücken" gelöscht wird. Hierdurch kann man im Vergleich zur Musterlösung 3 Zustände sparen.
Anmerkung: Die Berechnung wird dadurch auch insgesamt tendenziell beschleunigt

Aufgabe 11.3 (2 Punkte)

Eine spezielle Form von Turingmaschinenakzeptoren sind die sogenannten „Ja/Nein-Akzeptoren". Diese Turingmaschinen besitzen zwei sogenannte *Endzustände* z_{JA}, z_{NEIN},

von denen keine Übergänge definiert sind. Jede endliche Berechnung terminiert stets in einem dieser Zustände, wobei für einen Ja/Nein-Akzeptor T gilt: $L(T) = \{w|\text{Berechnung von } w \text{ endet in } z_{JA}\}$.
Sei $T = (Z, z_0, X, f, g, m, F)$ ein Turingmaschinenakzeptor. Definieren Sie eine Konstruktionsvorschrift, mit der sich abhängig von T ein Ja/Nein-Akzeptor $T_{J/N}$ konstruieren lässt, sodass $L(T) = L(T_{J/N})$ gilt. Nehmen sie o.B.d.A. an, dass gilt: $z_{JA}, z_{NEIN} \notin Z$.

Lösung 11.3
Definiere $T_{J/N} = (Z', z_0, X, f', g', m', F')$ abhängig von T wie folgt:

- $Z' = Z \cup \{z_{JA}, z_{NEIN}\}$
- Startzustand z_0 wie bei T
- Bandalphabet X wie bei T
- $f'(z,x) = \begin{cases} f(z,x), & \text{falls definiert} \\ z_{NEIN}, & \text{falls: } f(z,x) \text{ undefiniert} \wedge z \notin F \\ z_{JA}, & \text{falls: } f(z,x) \text{ undefiniert} \wedge z \in F \end{cases} \quad \forall z \in Z, x \in X$
- $g'(z,x) = \begin{cases} g(z,x), & \text{falls } g(z,x) \text{ definiert} \\ x, & \text{sonst} \end{cases} \quad \forall z \in Z, x \in X$
- $m'(z,x) = \begin{cases} m(z,x), & \text{falls } m(z,x) \text{ definiert} \\ 0, & \text{sonst} \end{cases} \quad \forall z \in Z, x \in X$
- $F' = \{z_{JA}\}$

Aufgabe 11.4 (2 + 1 + 1 + 1 = 5 Punkte)
Eine „Sture-Steinbock-Turingmaschine“ (SSTM) ist eine spezielle Turingmaschine mit zwei (äußerst unnachgiebigen) Lese-Schreib-Köpfen K_L, K_R. K_L startet die Berechnung auf dem ersten Zeichen eines Eingabewortes w, K_R auf dem Letzten. Jeder Kopf hat seine eigene (endliche) Zustandsmenge Z_L, Z_R. Ein Berechnungsschritt einer SSTM besteht aus zwei Phasen: In Phase 1 liest jeder Kopf zunächst das Bandsymbol seines aktuellen Feldes. In Phase 2 führt daraufhin jeder Kopf eine Bewegung und einen Zustandsübergang aus, sowie gibt eine Ausgabe aus. Bewegung, Zustandsübergang und Ausgabe hängen jedoch sowohl vom eigenen Zustand, dem selbst gelesenen Bandsymbol, sowie dem Zustand des anderen Kopfes und dem von ihm gelesenen Bandsymbol ab. Beide Köpfe machen diese Berechnung *gleichzeitig*, abhängig von ihren Zuständen, bzw. gelesenen Bandsymbolen aus Phase 1. Berechnungszustände, bei denen K_L und K_R verschiedene Ausgaben auf das selbe Bandfeld schreiben wollen, sind unzulässig. Während Sie zusammen an der Berechnung des Eingabewortes arbeiten, stehen K_L und K_R jedoch gleichzeitig im Wettbewerb darum, möglichst großen Anteil am Ergebnis zu haben. Deshalb ist ihre Bewegungsmöglichkeit eingeschränkt: K_L kann sich ausschließlich nach rechts (oder nicht), und K_R ausschließlich nach links (oder nicht) bewegen. Die Berechnung einer SSTM endet entweder, falls kein Übergang definiert ist oder nach dem Berechnungsschritt, in dem K_L auf einem beliebigen Feld weiter *rechts* von K_R auf dem Band steht.

a) Definieren Sie eine SSTM formal korrekt (als Tupel). Geben Sie für f, g und m jeweils Definitions- und Bildbereich an.
b) Betrachten Sie das *Palindromerkennungsbeispiel* aus (Kapitel 16, Folie 30ff.) der Vorlesung. Entwerfen Sie (grafisch) eine SSTM, die genau alle *Palindrome* aus $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}^+$ akzeptiert.
c) Evaluieren Sie die Berechnungsgeschwindigkeit ihrer SSTM aus Teilaufgabe b) im Vergleich zur Beispiel-TM aus der Vorlesung. Vergleichen Sie hierzu die benötigten

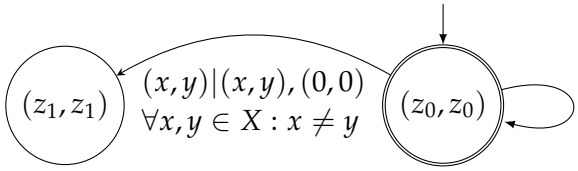
Berechnungsschritte der SSTM, bzw. Bsp-TM bis zum Ende der Berechnung. Hierbei soll ein Berechnungsschritt der SSTM wie 2 Berechnungsschritte der Bsp-TM gelten, da die SSTM 2 Bandsymbole in einem Berechnungsschritt schreiben kann. Welche der beiden Turingmaschinen ist in der Regel schneller? Begründen Sie.

d) Terminiert die Berechnung einer beliebigen SSTM immer? Begründen Sie.

Lösung 11.4

a) Eine SSTM SST lässt sich definieren als $SST = \{Z, (z_0^l, z_0^r), X, f, g, m\}$, mit:
- **endliche** Zustandsmenge $Z = Z_L \times Z_R$
- Startzustand (z_0^l, z_0^r)
- **endliches** Bandalphabet X (meist mit Blanksymbol $\times$)
- (partielle) Übergangsfunktion $f : (Z_L \times Z_R \times X \times X) \dashrightarrow (Z_L \times Z_R)$
- (partielle) Ausgabefunktion $g : (Z_L \times Z_R \times X \times X) \dashrightarrow (X \times X)$
- (partielle) Bewegungsfunktion $m : (Z_L \times Z_R \times X \times X) \dashrightarrow (\{0,1\}, \{-1,0\})$ oder $(\{0,R\}, \{L,0\})$

b)



c) Die $SSTM$ ist in der Regel *deutlich* schneller als die Beispiel-TM, da sie nur einmal über das Wort iterieren muss, während die Beispiel-TM immer wieder von links nach rechts (und umgekehrt) über das Wort spulen muss.

d) Nein. Auch SSTMs können unendliche Berechnungen haben, bei denen sich ab einem bestimmten Punkt, keiner der Beiden Köpfe aufeinander zubewegt, aber sie dennoch das Bandsymbol und oder den Zustand ändern.
Anmerkung: Dies führt jedoch zu einer periodischen Wiederholung der Konfiguration, weshalb jede von einer SSTM akzeptierbare Sprache eine entscheidbare Sprache ist, da man diese Wiederholung erkennen und daraufhin terminieren könnte.
Anmerkung2: ***Viel*** *mehr zu Entscheidbarkeit und Terminierung von Turingmaschinen erwartet sie in „Theoretische Grundlagen der Informatik“*