Übungsblatt01-Turingmaschinen - Musterlösung

TH Mittelhessen, FB MNI, Berechenbarkeit und Komplexität, Prof. Dr. B. Just

Aufgabe 1

Bitte definieren Sie eine Turingmaschine TM5, die entscheidet, ob der Input (über dem Alphabeth $\Sigma = \{0, 1\}$) genau so viele Nullen wie Einsen enthält (bzw enthielt - die TM kann den Input überschreiben). In diesem Fall soll sie im akzeptierenden Zustand enden, andernfalls im Reject.
Gehen Sie dazu wie folgt vor:

a.) Beschreiben Sie TM5 textlich;

b.) Beschreiben Sie TM5 formal, mit der Übergangsfunktion als Übergangsdiagramm;

c.) Dokumentieren Sie das Verhalten von TM5 auf dem Input 101100;

d.) Dokumentieren Sie das Verhalten von TM5 auf dem Input 101101.

Lösung Aufgabe 1:

a.) Idee: Lese jeweils ein Bit, kreuze es aus, suche das erste Komplementärbit, kreuze auch dieses aus und gehe zurück zum Bandanfang.
Wenn es irgendwann kein Komplementärbit gibt, reject.
Wenn irgendwann alle Bits ausgekreuzt sind, acceppt.

Textliche Beschreibung:

1. Falls Input leer, accept.

   sonst: Lies erstes Inputbit, kreuze es aus, merke es dir und markiere den Bandanfang.

2. Laufe so lange nach rechts, bis Komplementärbit gefunden wird.

   Falls Komplementärbit gefunden ist, kreuze es aus und gehe zurück zum Bandanfang (bzw. zu einer Position links von allen noch nicht ausgekreuzten Bits), .

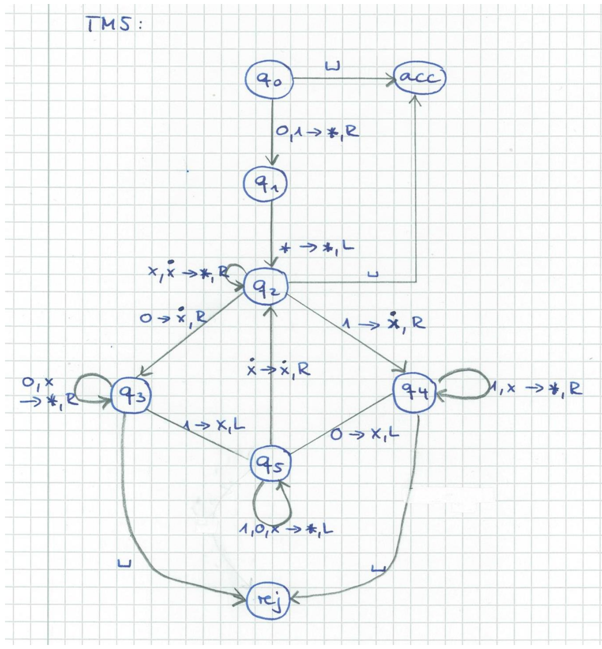
   sonst: reject.

3. Lies nach rechts bis erstes nicht ausgekreuztes Bit gefunden ist.

   Falls kein nicht ausgekreuztes Bit mehr exisitert: accept.

   Sonst: kreuze es aus und merke es dir. Gehe zu 2.)

b.) $\text{TM5} = (Q, \Sigma, \Gamma, \delta, q_0, \text{acc}, \text{rej})$ mit:
$Q = \{q_0, \text{acc}, \text{rej}, q_1, ...q_5\}$, dabei sind $q_0, \text{acc}, \text{rej}$ der Startzustand und die akzeptierenden bzw. verwerfenden Haltezustände
$\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, x, \dot{x}, \sqcup\}$, und δ wird durch das folgende Übergangsdiagramm gegeben:



Bemerkung: i.) Ursprünglich sollte $\dot{x}$ den Bandanfang markieren, nun markiert es aber das letzte Startbit von Schritt 2, das funktioniert auch.
ii.) Die TM könnte offenbar noch weiter vereinfacht werden: q_0 und q_1 werden nicht gebraucht, es könnte einfach q_2 der Startzustand sein. Dann müsste aber wiederum die textliche Beschreibung angepasst werden, dir Herleitung der TM wäre nicht mehr so offensichtlich.

c.) Die Konfigurationsfolge von TM5 bei Input 101100 sieht wie folgt aus:

q_0	1	0	1	1	0	0	
1	q_1	0	1	1	0	0	
q_2	1	0	1	1	0	0	
$\dot{x}$	q_4	0	1	1	0	0	
q_5	$\dot{x}$	x	1	1	0	0	
$\dot{x}$	q_2	x	1	1	0	0	
$\dot{x}$	x	q_2	1	1	0	0	
$\dot{x}$	x	$\dot{x}$	q_4	1	0	0	
$\dot{x}$	x	$\dot{x}$	1	$\dot{x}$	0	0	
$\dot{x}$	x	$\dot{x}$	1	x	q_5	0	
$\dot{x}$	x	$\dot{x}$	1	q_5	x	0	
$\dot{x}$	x	$\dot{x}$	q_5	1	x	0	
$\dot{x}$	x	q_5	$\dot{x}$	1	x	0	
$\dot{x}$	x	$\dot{x}$	q_2	1	x	0	
$\dot{x}$	x	$\dot{x}$	$\dot{x}$	q_4	x	0	
$\dot{x}$	x	$\dot{x}$	$\dot{x}$	x	q_4	0	
$\dot{x}$	x	$\dot{x}$	$\dot{x}$	x	x	q_5	
$\dot{x}$	x	$\dot{x}$	$\dot{x}$	x	q_5	x	
$\dot{x}$	x	$\dot{x}$	$\dot{x}$	q_5	x	x	
$\dot{x}$	x	$\dot{x}$	q_5	$\dot{x}$	x	x	
$\dot{x}$	x	$\dot{x}$	$\dot{x}$	q_2	x	x	
$\dot{x}$	x	$\dot{x}$	$\dot{x}$	x	q_2	x	
$\dot{x}$	x	$\dot{x}$	$\dot{x}$	x	x	q_2	
$\dot{x}$	x	$\dot{x}$	$\dot{x}$	x	x	$\sqcup$	acc

d.) Die Konfigurationsfolge von TM5 bei Input 101101 sieht wie folgt aus:

q_0	1	0	1	1	0	1	
1	q_1	0	1	1	0	1	
q_2	1	0	1	1	0	1	
$\dot{x}$	q_4	0	1	1	0	1	
q_5	$\dot{x}$	x	1	1	0	1	
$\dot{x}$	q_2	x	1	1	0	1	
$\dot{x}$	x	q_2	1	1	0	1	
$\dot{x}$	x	$\dot{x}$	q_4	1	0	1	
$\dot{x}$	x	$\dot{x}$	1	$\dot{x}$	0	1	
$\dot{x}$	x	$\dot{x}$	1	x	q_5	1	
$\dot{x}$	x	$\dot{x}$	1	q_5	x	1	
$\dot{x}$	x	$\dot{x}$	q_5	1	x	1	
$\dot{x}$	x	q_5	$\dot{x}$	1	x	1	
$\dot{x}$	x	$\dot{x}$	q_2	1	x	1	
$\dot{x}$	x	$\dot{x}$	$\dot{x}$	q_4	x	1	
$\dot{x}$	x	$\dot{x}$	$\dot{x}$	x	q_4	1	
$\dot{x}$	x	$\dot{x}$	$\dot{x}$	x	1	q_4	
$\dot{x}$	x	$\dot{x}$	$\dot{x}$	x	1	$\sqcup$	rej

Aufgabe 2

Bitte definieren Sie eine Turingmaschine TM6, die entscheidet, ob der Input (über dem Alphabet $\Sigma = \{0, 1\}$) genau doppelt so viele Nullen wie Einsen enthält.
Gehen Sie dazu wie folgt vor:

a.) Beschreiben Sie TM6 textlich;

b.) Beschreiben Sie TM6 formal, mit der Übergangsfunktion als Übergangsdiagramm;

c.) Begründen Sie die Korrektheit von TM6 für alle möglichen Inputs.

d.) Wenn Sie möchten, können Sie TM6 auf der website

http://morphett.info/turing/turing.html

programmieren und ausprobieren.

Lösung Aufgabe 2:

a.) Idee: Man läuft auf dem Band ganz nach rechts und kreuzt dabei eine 1 aus. Dann läuft man ganz nach links und kreuzt dabei eine 0 aus. Dann läuft man wieder ganz nach rechts und kreuzt dabei eine zweite 0 aus. Dann läuft man nach links und beginnt von vorne.
Das wird so lange gemacht, bis man keine 1 mehr findet. In diesem Fall wird geprüft, ob noch Nullen übrig sind (dann wird verworfen), oder ob der gesamte Input ausgekreuzt ist (dann wird akzeptiert).
Wird auf der Suche nach einer 0 keine 0 gefunden, wird verworfen.
(Bemerkung: Andere Algorithmen sind möglich - vertrauen Sie Ihrem, programmieren Sie ihn im Zweifelsfall :)).

Textliche Beschreibung:

1. Wenn der Input leer ist, akzeptiere. Sonst markiere den Bandanfang.
2. Laufe ans rechte Ende des Inputs.

   Wenn dabei eine 1 gefunden wird, kreuze sie aus, merke dir das und gehe zu 3.)

   sonst: gehe zu 5.).
3. Laufe nach links zur ersten Markierung (Bandanfang, oder spätere Stelle, auf deren linker Seite keine Nullen oder Einsen mehr stehen).

   Wenn unterwegs eine Null gefunden wird, kreuze sie aus, merke dir das und gehe zu Schritt 4.)

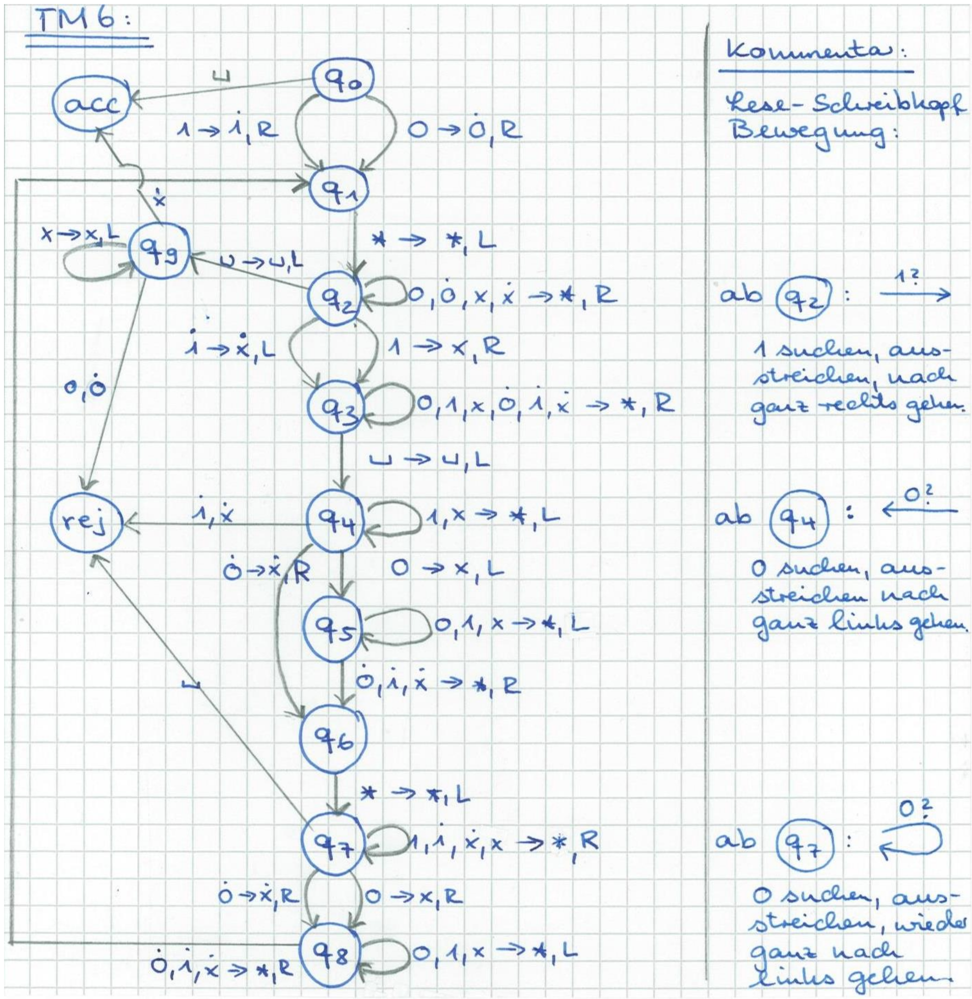
   sonst: reject.
4. Laufe nach rechts, bis eine Null gefunden wird. Kreuze sie aus, merke dir das, und laufe nach links zurück zur ersten Markierung und gehe zu Schritt 2.)

   Falls keine Null gefunden wird: reject.
5. (wird nur aus Schritt 2. erreicht) Laufe nach links zur ersten Markierung.

   Wenn unterwegs eine Null gefunden wird, reject.

   sonst: accept.

b.) TM6 $= (Q, \Sigma, \Gamma, \delta, q_0, \mathrm{acc}, \mathrm{rej})$ mit:
$Q = \{q_0, \mathrm{acc}, \mathrm{rej}, q_1, ...q_9\}$, dabei sind $q_0, \mathrm{acc}, \mathrm{rej}$ der Startzustand und die akzeptierenden bzw. verwerfenden Haltezustände
$\Sigma = \{0, 1\}$, $\quad \Gamma = \{0, 1, x, \dot{0}, \dot{1}, \dot{x}, \sqcup\}$, und δ wird durch das folgende Übergangsdiagramm gegeben:



c.) Man überzeugt sich, dass TM6 folgende Eigenschaften hat:

i.) Der leere Input wird sofort akzeptiert.

ii.) Bei nichtleerem Input ist nach dem ersten Verarbeitungsschritt zu jedem Zeitpunkt genau das erste Eingabefeld mit einem Punkt markiert. Denn es wird im ersten Verarbeitungsschritt markiert, und danach werden nur noch markierte mit markierten Symbolen überschrieben, und nicht markierte mit nicht markierten Symbolen.

iii.) Beim Eintritt in Zustand q_2 steht der Schreib-Lesekopf immer auf dem linkesten Symbol der Bandinschrift, welches markiert ist. Denn dies gilt sowohl, wenn q_2 aus q_1 erreicht wird, als auch, wenn q_2 aus q_9 erreicht wird.

iv.) Bei Eintritt in q_4 steht der Schreib-Lesekopf auf dem rechtesten Symbol der Bandinschrift.

v.)Bei Eintritt in q_7 steht der Schreib-Lesekopf auf dem linkesten Symbol der Bandinschrift.

vi.) Beim Eintritt in q_9 steht der Schreib-Lesekopf auf dem zweitlinkesten Feld des Bandes.

vii.) Seien $K0$ und $K1$ zu jedem Zeitpunkt die Anzahlen der seit Beginn der Verarbeitung ausgekreuzten Nullen bzw. Einsen. Ist $2 \cdot K1 = K0$, so erfüllte der Input die Bedingung, dass es doppelt soviele Nullen wie Einsen gibt, genau dann, wenn das Wort auf dem Band diese Bedingung erfüllt. Die Situation $2 \cdot K1 = K0$ nenne wir „pari“. Es gilt (Beweis durch Induktion nach der Anzahl der Schritte von q_9 nach q_2):

- Beim Eintritt in q_2 ist $K0 = 2 \cdot K1$, die Situation ist also pari;

- Beim Eintritt in q_4 ist $2 \cdot K1 - 2 = K0$, es müssen also noch zwei Nullen ausgekreuzt werden, um wieder pari zu sein ;

- Beim Eintritt in q_7 ist $2 \cdot K1 - 1 = K0$, es muss also noch eine Null ausgekreuzt werden, um wieder pari zu sein.

viii.) Der akzeptierende Zustand wird nur erreicht, wenn entweder der Input das leere Band ist, oder wenn die Verarbeitung aus q_9 kommt. q_9 wird nur nach einer Pari-Situation (Eintritt in q_2) erreicht, wenn keine 1 mehr auf dem Band steht (sonst ginge die Verarbeitung mit q_3 weiter). Der akzeptierende Zustand wird dann nur erreicht, wenn auch keine 0 mehr auf dem Band steht.
Also enthält der Input tatsächlich doppelt soviele Nullen wie Einsen, wenn der akzeptierende Zustand erreicht wird.

ix.) Enthält der Input umgekehrt doppelt soviele Nullen wie Einsen, wird der akzeptierende Zustand erreicht (Induktion nach der Anzahl der Einsen).

x.) TM6 hält immer an. Denn die Bandinschrift wird nie länger als der Input.
Das Übergangsdiagramm ist vollständig definiert (in den Zuständen q_4 und q_7 ist es nicht möglich, ein blank zu finden, da sie in einer Linksbewegung stattfinden. Alle anderen Übergangsvorschriften finden sich im Diagramm).
Sofern keine Schleife unendlich oft durchlaufen wird, erreicht die Verarbeitung bei jedem Input also den akzeptierenden oder verwerfenden Zustand. Wir zeigen also, dass jede Schleife nur endlich oft durchlaufen wird.
i.) Mit jedem Durchlauf der Schleife $q_8 \longrightarrow q_1$ werden drei Symbole ausgekreuzt, es kann also nur endlich viele Durchläufe dieser Schleife geben.
ii.) In den Schleifen in q_2, q_3, q_4, q_5, q_7 und q_8 bewegt sich der Lese-Schreibkopf nur in eine Richtung, sie enden jeweils mit dem linken oder rechten Ende der Bandinschrift. Sie werden also bei jedem Aufruf nur endlich oft durchlaufen. Dass sie nur endlich oft aufgerufen werden, folgt aus i.).

Damit ist gezeigt, dass TM6 stets anhält, und für jeden Input den richtigen Output liefert. Sie ist somit korrekt.

Aufgabe 3

a.) Bitte beschreiben Sie textlich eine TM7, die einen hexadezimalen Input in die entsprechende Binärfolge umwandelt.

b.) Bitte beschreiben Sie textlich eine TM8, die als Input eine Buchstabenfolge erhält, und als Output die Morse-Alphabet-Übersetzung dieser Buchstabenfolge liefert. Dabei wird der Einfachheit halber davon ausgegangen, dass das Input-Alphabet nur die 26 Großbuchstaben von A bis Z und keine Trennung zwischen den einzelnen Worten enthält. Das Morsealphabet enthält die 3-elementige Zeichenmenge $\{. \, , \, - \, , \, \_\}$.
Die genaue Übersetzung findet man in Wikipedia. Das „_“ steht für das Blank des normalen Morsealphabets, das aber zu Widersprüchen zur Definition einer TM führen würde.

Bemerkung: Nach dieser Aufgabe sollte klar sein, dass es für die Berechnungen nicht wichtig ist, in welcher Codierung der Input vorliegt. Denn eine TM kann umcodieren :).

Lösung Aufgabe 3:

a.) Idee: Die Idee besteht darin, zunächst das Ende des Inputs durch ein Sonderzeichen zu markieren.
Dann wird vom Bandanfang an jeweils ein Symbol ausgekreuzt, und die entsprechende Binärfolge in den 4 Bits am Ende der Bandinschrift eingefügt.
Zuletzt wird der Input gelöscht, und die Bandinschrift bis zum Bandanfang nach links verschoben.

Textliche Beschreibung:

1.) Markiere das Ende des Inputs durch ein Sonderzeichen #, und gehe zurück zum Bandanfang.

2.) Solange nicht alle Zeichen vor dem # ausgekreuzt sind:

i.) Merke dir das gelesene Zeichen X im Zustand Q_X. Aus diesem Zustand wird ans Ende der Bandinschrift gelaufen. Ist dieses erreicht, werden die 4 Bits der Binärdarstellung von X aufs Band geschrieben. Dafür werden die Zustände $Q_{X,1}, Q_{X,2}, Q_{X,3}, Q_{X,4}$ durchlaufen.

ii.) Gehe zurück nach links bis zum letzten nicht ausgekreuzten Symbol.

Bemerkung: Die Turingmaschine hat also neben den Zuständen, die zum normalen Navigieren, Auskreuzen und Verschieben braucht, noch (mindestens) $16 \cdot 5 = 80$ weitere Zustände: $\{Q_X, Q_{X,i} : X \in \{0, ...9, A, B, C, D, E, F\}, i \in \{1, 2, 3, 4\}\}$.

3.) Gehe zurück an den Bandanfang und überschreibe alle ausgekreuzten Felder und das # mit $\sqcup$.

4.) Solange zu Beginn des Bandes noch Blanks stehen:

Laufe zum Beginn der Bandinschrift. Laufe sie von links nach rechts ab und verschiebe jedes Symbol, um ein Feld nach links. Hierfür werden nur wenige Zustände gebraucht, die Symbole sind ja jetzt 0 oder 1.

b.) Die TM funktioniert im Grunde genauso wie die TM aus Teil a.).
Der Unterschied besteht nur darin, dass nicht jedes Input-Zeichen aus den Buchstaben von A bis Z zu genau einem Output-Zeichen führt. So wird z.B. das E kodiert zu „.“ und benötigt daher nur einen zusätzlichen Zustand $Q_{E,1}$ zum Schreiben. Das Y wird codiert zu „-.–“, und benötigt daher vier Zustände $Q_{Y,1}, Q_{Y,2}, Q_{Y,3}, Q_{Y,4}$ zum Schreiben des Morsezeichens.

Aufgabe 4

Bitte lesen Sie sich im Sipser die Seiten 165 bis 175 über Turingmaschinen durch. Das meiste müssten Sie verstehen, und so sehen Sie alles noch einmal aus einem anderen Blickwinkel, und in Englisch :).

Lösung Aufgabe 4:

Viel Spaß beim Lesen :).