



Aufgabe 3 (Ausführung einer Turing-Maschine [klausurtypisch], 5+2 Punkte)

- a) Gegeben sei die Turing-Maschine auf der rechten Seite. Geben Sie für die Startworte abc und ε die zugehörigen Berechnungen als Folge von Konfigurationen an, bis keine Folgekonfiguration mehr möglich ist.

% Zustaende
Z: Start z1 z2 z3
% Alphabet, Leerzeichen # automatisch dabei
A: a b c
% Start
S: Start

- b) Was vermuten Sie, was die Turing-Maschine generell macht?

% Ueberfuehrungsfunktion

% alt	lesen	neu	schreiben	Richtung
Start	#	z1	#	L

Aufgabe 4 (Erstellung einer Turing-Maschine [klausurtypisch], 10+4 Punkte)

- a) Zeigen Sie durch die Angabe einer Turing-Maschine, dass die Sprache $\{cca^n b^n \mid n \geq 0\}$ von einer Turing-Maschine akzeptiert wird. Beschreiben Sie vorher Ihren Ansatz mit mindestens 2 Sätzen.
- b) Geben Sie für die Startworte ccab, ε , und cb die zugehörigen Berechnungen als Folge von Konfigurationen an, bis keine Folgekonfiguration mehr möglich ist.

z1	a	z2	b	L
z1	b	z2	b	L
z1	c	z2	c	L
z1	#	z3	#	R
z2	a	z1	b	L
z2	b	z1	b	L
z2	c	z1	c	L
z2	#	z3	#	R
z3	a	z3	b	R
z3	b	z3	b	R
z3	c	z3	b	R
z3	#	z3	#	S

Aufgabe 5 (Ausführung einer Turing-Maschine [klausurtypisch], 5+2 Punkte)

- a) Gegeben sei die Turing-Maschine auf der rechten Seite. Geben Sie für die Startworte aaaa und aaa die zugehörigen Berechnungen als Folge von Konfigurationen an, bis keine Folgekonfiguration mehr möglich ist.
- b) Was vermuten Sie, was die Turing-Maschine generell macht?

% Zustaende
Z: Start z1 z2 z3 z4 S
% Alphabet, Leerzeichen # automatisch dabei
A: a b
% Start
S: Start

% Ueberfuehrungsfunktion

% alt lesen neu schreiben Richtung

% alt	lesen	neu	schreiben	Richtung
Start	#	z1	#	L
z1	a	z2	b	L
z1	b	z2	b	L
z1	#	z3	#	R
z2	a	z1	a	L
z2	b	z1	a	L
z2	#	z3	#	R
z3	a	z4	a	R
z3	b	z4	#	R
z4	a	z4	a	R
z4	b	z4	b	R
z4	#	S	#	S

Aufgabe 6 (Erstellung einer Turing-Maschine [klausurtypisch], 10+4 Punkte)

- a) Zeigen Sie durch die Angabe einer Turing-Maschine, dass die Sprache $\{a^n ccb^n \mid n \geq 0\}$ von einer Turing-Maschine akzeptiert wird. Beschreiben Sie vorher Ihren Ansatz mit mindestens 2 Sätzen.
- b) Geben Sie für die Startworte accb, ε , und ab die zugehörigen Berechnungen als Folge von Konfigurationen an, bis keine Folgekonfiguration mehr möglich ist.



Aufgabe 7 (Weitere Experimente mit Turing-Maschinen)

Von der Veranstaltungsseite ist ein Eclipse-Projekt erhältlich, dass u. a. die Simulation von Turing-Maschinen erlaubt. Lesen Sie das Kapitel 3 der theoriesammlung-Doku unter <http://kleuker.iui.hs-osnabrueck.de/querschnittlich/Theoriesammlung.pdf>.

- a) Übertragen Sie Ihre Turing-Maschine aus 4a) in das Format des Simulators in die vorhandene Datei `beispiele/turingmaschinen/TMcca_nb_n.tm`. Prüfen Sie mit den JUnit-Tests aus `test.turingMaschine.TMcca_nb_nTest.java` ob Ihre Maschine die enthaltenen Tests besteht. Korrigieren Sie gegebenenfalls Ihre Turing-Maschine.
- b) Übertragen Sie Ihre Turing-Maschine aus 6a) in das Format des Simulators in die vorhandene Datei `beispiele/turingmaschinen/TMa_nccb_n.tm`. Prüfen Sie mit den JUnit-Tests aus `test.turingMaschine.TMa_nccb_nTest.java` ob Ihre Maschine die enthaltenen Tests besteht. Korrigieren Sie gegebenenfalls Ihre Turing-Maschine.
- c) Zeigen Sie mit Hilfe einer Turing-Maschine, dass die Funktion $f(l^n) = l^n \text{ div } 2$ (div für die ganzzahlige Division) turing-berechenbar ist. Beschreiben Sie Ihren Ansatz mit mindestens 2 Sätzen. Schreiben Sie Ihre Maschine in die Datei `beispiele/turingmaschinen/TMHalbieren.tm`. Prüfen Sie Ihr Ergebnis mit `test.turingMaschine.TMHalbierenTest.java`.
- d) In der Vorlesung wurde versucht zu zeigen, dass die Funktion $f(l^n) = l^{2n}$ turing-berechenbar ist. Dies ist bis auf unnötige Leerzeichen am Anfang auch gelungen. Korrigieren Sie die Turing-Maschine `TM2malN.tm` (oder schreiben Sie sie neu) in der Datei `TM2malNVersion2.tm`, so dass das geforderte Ergebnisformat mit nur einem Leerzeichen am Anfang eingehalten wird. Sie dürfen das Alphabet natürlich erweitern. Beschreiben Sie Ihren Ansatz mit mindestens 2 Sätzen. Prüfen Sie Ihr Ergebnis mit `test.turingMaschine.TM2malNVersion2Test.java`.
- e) Die JUnit-Tests aus den vorherigen Teilaufgaben lassen vermuten, dass Turing-Maschinen problemlos mit Tests geprüft werden können. Dies ist nicht der Fall. Machen Sie das deutlich, indem Sie eine Turing-Maschine zu c) in der Datei `Tm2malNFalsch.tm` entwickeln, die die Tests aus c) alle erfüllen, aber trotzdem fehlerhaft ist. Zur Vereinfachung sind die Tests aus c) auch in `test.turingMaschine.Tm2malNFakeTest.java` enthalten. Ergänzen Sie dann einen Test, der zeigt, dass Ihre Maschine fehlerhaft ist.
- f) 🍷 Konvertieren Sie Ihre Turing-Maschine aus c) und bringen Sie diese auf <https://turingmachinesimulator.com/> zum Laufen.
- g) 🍷 Konvertieren Sie Ihre Turing-Maschine aus c) und bringen Sie diese mit Tursi_{zum} zum Laufen.