

Übungsaufgaben Informatik 4 Stünder KW15 Di, 7 April

Turing Maschinen, Teil 1!

Grundaufbau im Vergleich:

- Turing-Maschinen und PDAs haben gemeinsam: Beide besitzen einen Speicher!
- Die Turing-Maschinen unterscheiden sich von PDAs durch die Art des Speichers: PDAs besitzen einen Stack, Turing-Maschinen ein „Tape“.
- Ein ‚Tape‘ ist ein endlos langes Band mit unendlich vielen Speicherzellen.
- Ein PDA besitzt ein eigenes Kelleralphabet,

Eine TM besitzt ein Bandalphabet, und das Eingabealphabet ist eine Teilmenge des Bandalphabets!

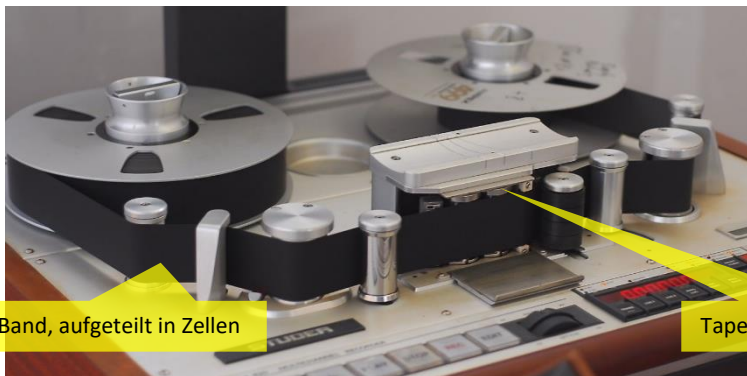


Abbildung 1: Analogie Band der Turing-Maschine mit Tape-Recorder (Tatsächlich wurde das ‚Band‘ in Turings theoretischem Konstrukt oft mit dem Magnetophon verglichen!)

Endlos langes Band, aufgeteilt in Zellen

Tape-Head / Lese-Schreibkopf

Analogie Tape – Array:

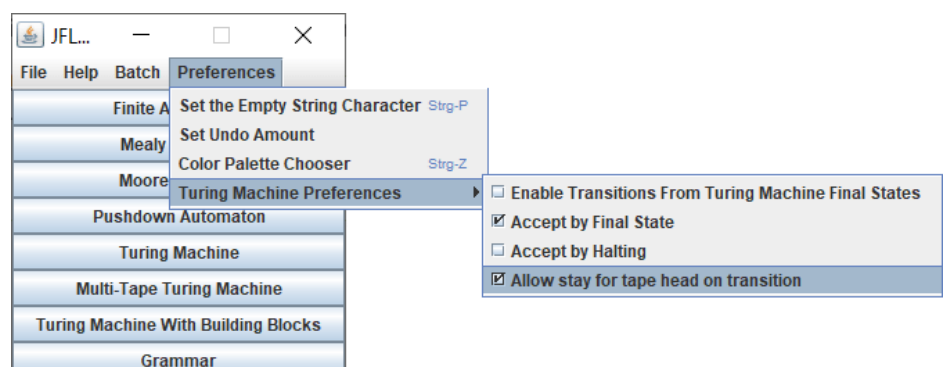
- Aus den Speicherzellen des TuringMachine-Tapes kann man den Wert des Inhalt herauslesen und einen neuen Wert hineinschreiben. Man kann sich das wie ein Array vorstellen. Es passt immer nur ein Wert in eine Speicherzelle – beim Array wäre das ein bestimmter Wert, der nur in jeweils ein Element passt.
- Wie kommt man an eine entsprechende Speicherzelle? Indem man einen gedachten „Tape-Head“ dorthin bewegt – das ist so, als würde man den aktuellen Index eines Arrays setzen. Allerdings kann sich der Tape-Head nur entweder eins nach links, eins nach rechts oder garnicht bewegen. Größere Distanzen springen kann er nicht.
- Befindet sich der ‚Tape-Head‘ an einem bestimmten Ort des Bandes (beim Array wäre das ein bestimmter Wert der Indexvariablen), so kann genau dort – und nur dort – der Wert ausgelesen und verändert werden.

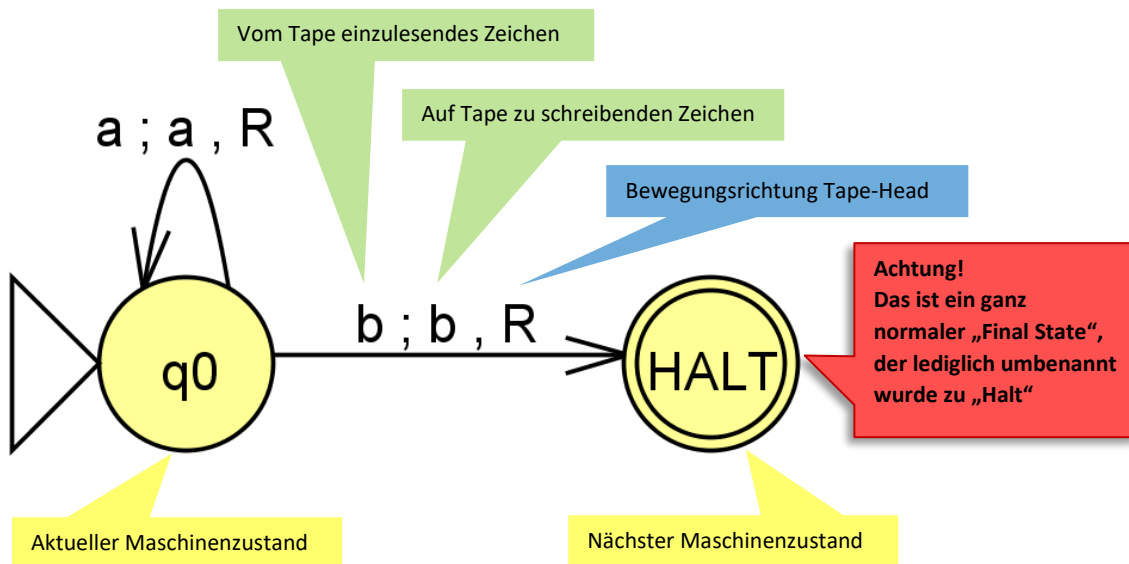
Endzustand:

Die Turing-Maschine beendet ihr Programm, wenn sie in einen Endzustand kommt. Diesen Endzustand nennt man auch ‚Halt‘-State.

Grundeinstellung JFLAP

Bitte wie in nebenstehender Abbildung die Präferenzen setzen.





Die Bewegungsrichtung des Tape-Heads kann sein R („right“), L („left“), S („stay“)

Aufgabe 1: Einstieg in Turing-Maschinen

a) Baue in JFLAP die obige Turing-Maschine, gib die folgenden Eingabewörter ein und simuliere mit „Input – Step“:

ab aaab aabaa aa

b) Erläutere in Stichworten deine Beobachtung

c) Trage in die Lücken ein:

Die TM akzeptiert die Wörter: _____ / Die TM akzeptiert NICHT: _____

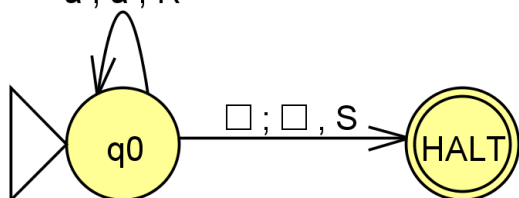
Die TM akzeptiert dann, wenn _____

Aufgabe 2: Was macht diese Turing-Maschine?

a) Werden die folgenden Eingabewörter akzeptiert? Wenn ja, erkläre warum. Teste mit den folgenden Eingabewörtern:

b ; b , R
 B ; b , R
 A ; a , R
 a ; a , R

ab baabb AB BAaBb



b) Vergleiche die Wörter auf dem Band vor und nach dem Programmablauf. Was fällt dir auf, bzw. wozu dient diese TM?

Aufgabe 3: Eine Turing-Maschine ist ein Septupel:

Definition Turing-Maschine: $TM = (Q, \Sigma, \Gamma, \delta, q_0, F, \square)$, $\Sigma \subset \Gamma$, „Ist Teilmenge von“ „Blank“: $\square \in \Gamma \setminus \Sigma$ „Ist Element von“

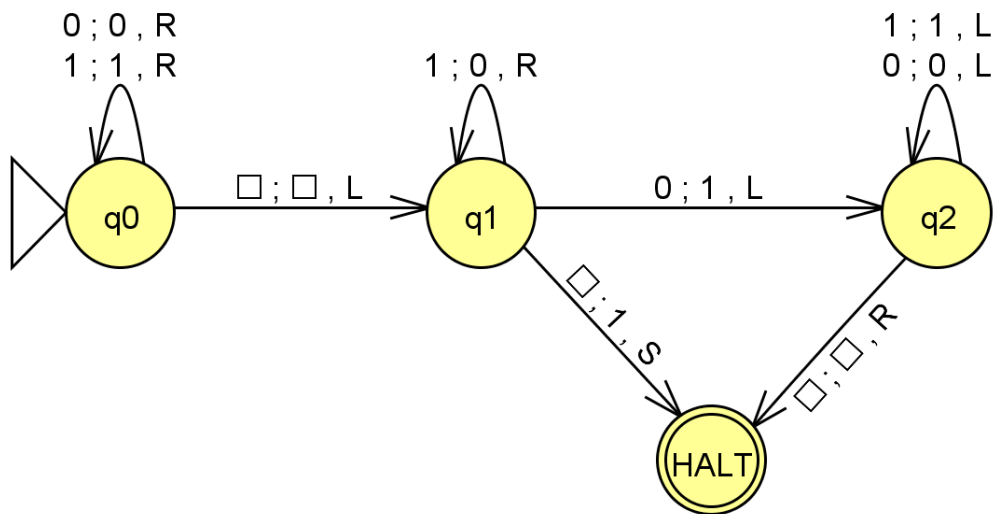
Definition PDA (Wdh): $PDA = (Q, \Sigma, \Gamma, \delta, q_0, F, Z)$ „ohne“

Beschreibe anhand der Definitionen die Übereinstimmungen und Unterschiede zum PDA

Aufgabe 4: Was macht diese Turing-Maschine?

Unsere erste TM, die einem ‚sinnvollen‘ Computerprogramm am nächsten kommt.

a) Baue sie nach (oder überlege mit Zettel und Stift), welche Ausgaben die TM auf dem Tape erzeugt:



Eingabewerte:	Ausgabewerte:
1	
10	
11	
100	
1011	
1111	

b) Beschreibe in Stichworten, „wie“ die TM arbeitet!