Turing Maschine

Thorsten Timmer

SS 2005

Proseminar Beschreibungskomplexität

bei

Prof. D. Wotschke

Turing Maschine

Inhalt

- Einführung
- Formale Definition
- Berechenbare Sprachen und Funktionen
- Berechnung ganzzahliger Funktionen
- Techniken zur Konstruktion von Turing Maschinen
- Modifikationen von Turing Maschinen (Optional)

Turing Maschine

Einführung

Die Turing Maschine ist:

- ein einfaches mathematisches Modell für einen Allzweckcomputer, wie wir ihn uns heute vorstellen
- ein Modell, daß die Berechenbarkeit eines Computers modelliert
- nur ein theoretisches (Hilfs-)Modell

Turing Maschine

Einführung

- Mit der Turing Maschine (TM) ist es möglich Probleme auf ihre Berechenbarkeit zu untersuchen
  (existiert ein Algorithmus für dieses Problem oder nicht).
- Existiert ein Algorithmus für ein gegebenes Problem, dann existiert auch eine TM um dieses Problem zu lösen und umgekehrt.
- Da alle endlichen berechenbaren Probleme mit einer TM gelöst werden können, ist es möglich mit ihr zu Beweisen, daß bestimmte Probleme nicht berechenbar sind.

Turing Maschine

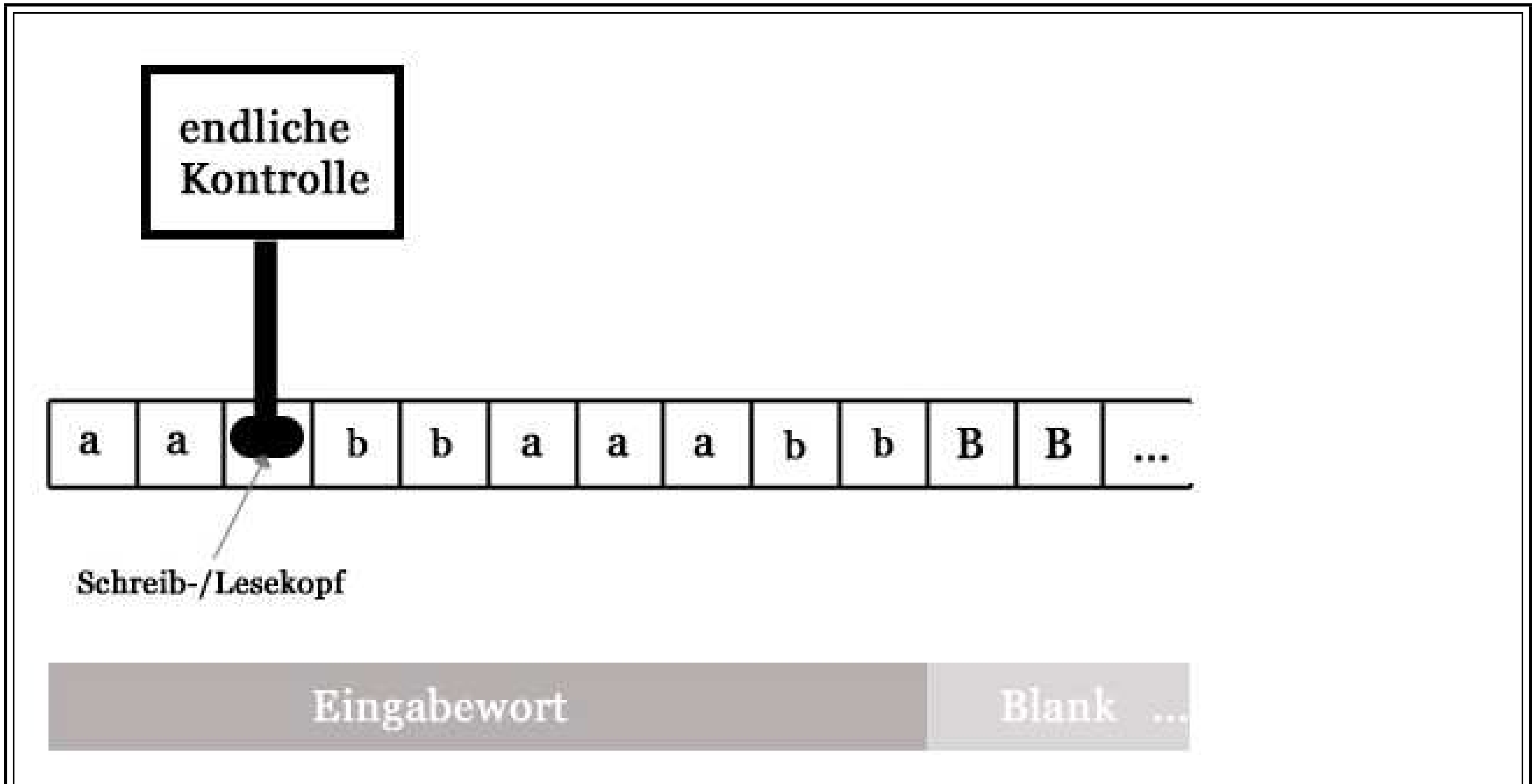
Einführung

Das Basismodell der TM besteht aus:

- einer `endlichen Kontrolle`
- einem nach links beschränkten und nach rechts unendlichen `Eingabeband`
- einem `Schreib-/Lesekopf`

Turing Maschine

Einführung



Basismodell der Turing Maschine

Turing Maschine

Formale Definition

Eine TM ist ein Tupel $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$,

wobei gilt:

- Q ist eine endliche Menge von Zuständen
- Σ ist eine endliche Menge von Eingabesymbolen.
- Γ ist eine endliche Menge von Bandsymbolen.
- δ ist die Übergangsfunktion, eine Abbildung von $Q \times \Gamma$ nach $Q \times \Gamma \times \{L, R\}$
- $q_0 \in Q$ ist der Anfangszustand
- B ist das Blanksymbol
- $F \subseteq Q$ ist die Menge der Endzustände

Turing Maschine

Formale Definition

- Die Übergangsfunktion δ beschreibt den Übergang von einer Zustandsbeschreibung in die nächste.
- Eine Zustandsbeschreibung ist folgendermaßen definiert:

$$\alpha_1 q \alpha_2 \qquad , \alpha_1 \alpha_2 \in \Gamma^*, q \in Q$$

  - $\alpha_1 \alpha_2$ ist der momentane Bandinhalt, gefolgt von unendlich vielen Blank Symbolen
  - q ist der Zustand, in der sich die endliche Kontrolle befindet.
  - Der Schreib-/Lesekopf befindet sich bei einer solchen, auch Konfiguration genannten Zustandsbeschreibung auf dem ersten Symbol von α_2. Ist $\alpha_2 = \epsilon$, also leer, befindet sich der Kopf rechts neben der Eingabe auf einem Blank.

Turing Maschine

Formale Definition

- $\delta(q, X_i) = (p, Y, L)$ beschreibt den Übergang von Zustand q mit gelesenem X_i in den Zustand p, wobei X_i durch Y ersetzt wird und der Bandkopf sich danach um eine Stelle nach links bewegt.
- Die dazugehörigen Zustandsbeschreibungen sehen folgendermaßen aus:

$$X_1 X_2 \ldots X_{i-1} q X_i \ldots X_n \vdash_M X_1 X_2 \ldots X_{i-2} p X_{i-1} Y X_{i+1} \ldots X_n$$

- $\vdash_M$besagt, dass die rechts stehende Zustandsbeschreibung aus der links stehenden Zustandsbeschreibung folgt. Das M kann weggelassen werden, wenn klar ist auf welche TM sich $\vdash$ bezieht. Folgt eine Zustandsbeschreibung durch eine endliche Anzahl von Zuständen aus einer anderen, dann kann man auch $\vdash^*$ schreiben.
- Eine Konfiguration ohne Folgekonfiguration ist eine Endkonfiguration. Die TM hält in diesem Fall an. Ist der Zustand ein akzeptierender, so akzeptiert auch die TM. Anderfalls verwirft sie.

Turing Maschine

Formale Definition

- Die von einer TM erkannte Sprache kann man damit als die Menge

$$\{\, w \mid w \in \Sigma^* \wedge\, q_0 w \vdash^* \alpha_1 p \alpha_2,\, p \in F \wedge\, \alpha_1, \alpha 2 \in \Gamma^* \}$$

  schreiben.

Turing Maschine

Formale Definition

- Beispiel:
  - $T(M) = \{0^n 1^n \mid n \geq 1\}$
  - Anfangseingabe$= 0^n 1^n$, gefolgt von unendlich vielen Blank Symbolen
  - Ersetze am weitesten links stehende 0 durch X
  - Bewegung nach rechts zur am weitesten links stehenden 1 und ersetze diese durch Y.
  - Bewegung nach links zum am weitesten rechts stehenden X. Ein Schritt nach rechts zur nächsten 0.
  - Wiederhole die Schleife bis:
  - 1. Wird auf der Suche nach einer 1 ein B gefunden, hält M ohne zu erkennen.
  - 2. M findet keine 0 mehr. Dann Suche nach einer 1. Gibt es keine mehr, dann akzeptiere w. Verwerfe andernfalls.

Turing Maschine

Formale Definition

Zustand	0	1	X	Y	B
q_0	(q_1, X, R)	–	–	(q_3, Y, R)	–
q_1	$(q_1, 0, R)$	(q_2, Y, L)	–	(q_1, Y, R)	–
q_2	$(q_2, 0, L)$	–	(q_0, X, R)	(q_2, Y, L)	–
q_3	–	–	–	(q_3, Y, R)	(q_4, B, R)
q_4	–	–	–	–	–

Tabelle 1: Die Übergangsfunktion

Turing Maschine

Berechenbare Sprachen und Funktionen

- Die von der TM erkannten Sprachen sind die `rekursiv aufzählbaren` Sprachen.
- Die `regulären Sprachen` und die `kontextfreien Sprachen` sind Teilmengen der `rekursiv aufzählbaren Sprachen`.
- $\exists L \in RA$ und eine TM M mit:
  L wird von M erkannt (aber nicht entschieden),
  $w \in L \Rightarrow M$ hält mit Eingabe w
  $w \notin L \Rightarrow M$ hält nicht mit Eingabe w
  - Diese Sprachen sind semi-entscheidbar.

Turing Maschine

`Berechenbare Sprachen und Funktionen`

- Die `rekursiven Mengen` sind eine Teilmenge der rekursiv aufzählbaren Mengen und entscheidbar.
  Die TM terminiert für jedes Wort $w \in L$ mit $L \in \{$rekursiven Mengen$\}$

Turing Maschine

Berechnung ganzzahliger Funktionen

- Darstellung einer ganzen Zahl i in unärer Darstellung (0^i).
- Hat eine Funktion k Argumente, so werden diese durch das Zeichen 1 getrennt ($0^{i_1}10^{i_2}1\ldots10^{i_k}1$).
- Das Ergebnis der Funktion f liegt nach der Berechnung in Form von der Zeichenkette 0^m auf dem Band vor.
- Also ist $f(i_1,\ldots,i_k) = m$.

Turing Maschine

Berechnung ganzzahliger Funktionen

- Beispiel:
  - Die echte Subtraktion $m \dot{-} n$
  - $m \dot{-} n = m - n$,für $m \geq n$
    $m \dot{-} n = 0$,für $m < n$
  - Sei M die TM, die diese Berechnung durchführt, mit
    $M = (\{q_0, \ldots, q_6\}, \{0, 1\}, \{0, 1, B\}, \delta, q_6, B, \emptyset)$
  - M startet mit der Eingabe $0^m 1 0^n$ auf dem Band und stoppt mit $0^{m \dot{-} n}$ auf dem Band.

Turing Maschine

Berechnung ganzzahliger Funktionen

Die Übergangsfunktion δ aufgelistet und beschrieben:

1. $\delta(q_0, 0) = (q_1, B, R)$
   Die führende Null wird durch ein Blank ersetzt.
2. $\delta(q_1, 0) = (q_1, 0, R)$
   $\delta(q_1, 1) = (q_2, 1, R)$
   Die erste Eins wird gesucht.
3. $\delta(q_2, 1) = (q_2, 1, R)$
   $\delta(q_2, 0) = (q_3, 1, L)$
   Ist die erste Eins schon gefunden, werden alle weiteren übersprungen und die nächste folgende Null durch eine Eins ersetzt.

Turing Maschine

Berechnung ganzzahliger Funktionen

4. $\delta(q_3, 0) = (q_3, 0, L)$
   $\delta(q_3, 1) = (q_3, 1, L)$
   $\delta(q_3, B) = (q_0, B, R)$
   Alle Nullen und Einsen nach links werden übersprungen, bis ein Blank gefunden ist. Ist es gefunden, bewegt sich die TM um ein Feld nach rechts und geht in q_0 über, um die Schleife zu wiederholen.

5. $\delta(q_2, B) = (q_4, B, L)$
   $\delta(q_4, 1) = (q_4, B, L)$
   $\delta(q_4, 0) = (q_4, 0, L)$
   $\delta(q_4, B) = (q_6, 0, R)$
   Sollte in Zustand q_2 keine Null sondern ein Blank gefunden werden, so bewegt sich die TM in Zustand q_4 und bei einer Bewegung nach links werden alle Einsen durch B ersetzt, bis ein B gefunden wird. Dieses Blank wird durch eine Null ersetzt, die TM geht in Zustand q_6 über und stoppt.

Turing Maschine

Berechnung ganzzahliger Funktionen

6. $\delta(q_0, 1) = (q_5, B, R)$
   $\delta(q_5, 0) = (q_5, B, R)$
   $\delta(q_5, 1) = (q_5, B, R)$
   $\delta(q_5, B) = (q_6, B, R)$
   Findet die TM in Zustand q_0 eine Eins statt einer Null, so sind die ersten m Nullen erschöpft. M geht dann in Zustand q_5 über, um das Band von dem überflüssigen Rest zu bereinigen und stoppt dann in Zustand q_6.

Turing Maschine

Techniken zur Konstruktion von Turing Maschinen

- Es ist nicht gerade einfach für komlexere Probleme eine TM zu entwerfen
- Dafür gibt es einige Tricks, die nun vorgestellt werden:
  - Speichern in der endlichen Kontrolle
  - Mehrspurige Turing Maschinen
  - Markierungssymbole
  - Verschiebungen
  - Unterprogramme

Turing Maschine

`Speichern in der endlichen Kontrolle`

- Die endliche Kontrolle kann erweitert werden, um Information zu speichern.
- Die Bezeichner für die Zustände der endlichen Kontrolle werden einfach als Tupel geschrieben.
- Beispiel: Als Beispiel betrachten wir eine TM M, die sich das erste Symbol des Eingabewortes w anschaut, es in der endlichen Kontrolle speichert und mit den folgenden Symbolen von w vergleicht. Kommt das entsprechende Symbol nicht mehr in w vor, akzeptiert M, andernfalls verwirft M.

Turing Maschine

Speichern in der endlichen Kontrolle

- Sei

$$M = (Q, \{0, 1\}, \{0, 1, B\}, \delta, [q_0, B], B, F)$$

mit $Q \in \{q_0, q_1\} \times \{0, 1, B\}$. Q ist also die Menge der Paare

$$\{[q_0, 0], [q_0, 1], [q_0, B], [q_1, 0], [q_1, 1], [q_1, B]\}.$$

Die Menge F ist $\{[q_0, B]\}$. Die erste Komponente der Zustände kontrolliert also die nächste Aktion und die zweite Komponente läßt die TM sich an das erste gelesene Eingabesymbol „erinnern“.

Turing Maschine

Speichern in der endlichen Kontrolle

- δ ist nun folgendermaßen definiert:
  - - $\delta([q_0, B], 0) = ([q_1, 0], 0, R)$
    - $\delta([q_0, B], 1) = ([q_1, 1], 1, R)$

    Zu Beginn befindet sich M in Zustand q_0 und liest entweder eine 0, oder eine 1 und bewegt sich dann nach rechts. Die erste Komponente wird nun zu q_1, während in der zweiten das gelesene Symbol gespeichert wird.

Turing Maschine

Speichern in der endlichen Kontrolle

- $\delta([q_1, 0], 1) = ([q_1, 0], 1, R)$
- $\delta([q_1, 1], 0) = ([q_1, 1], 0, R)$

In diesen Zuständen wird geprüft, ob das erste gelesene und in der zweiten Komponente gespeicherte Symbol ein zweites mal gelesen wird oder nicht. Wird jeweils ein anderes Symbol gelesen, bewegt sich die TM um ein Feld nach rechts, ohne etwas zu verändern.

Turing Maschine

`Speichern in der endlichen Kontrolle`

- - $\delta([q_1, 0], B) = ([q_1, B], 0, L)$
  - $\delta([q_1, 1], B) = ([q_1, B], 1, L)$

  Wird bis zum ersten Blank Symbol das erste Symbol nicht mehr gelesen, so geht M in den akzeptierenden Zustand über.
- Die Übergänge von den Zuständen $[q_1, 0]$ und einer gelesenen 0 und $[q_1, 1]$ und einer gelesenen 1 sind nicht definiert, so dass die TM in diesen Fällen ohne zu akzeptieren anhält.
- Die endliche Kontrolle kann aus k Komponenten bestehen, wobei eine Komponente immer die Aktion angibt und die restlichen zum Speichern von Information verwendet werden können.

Turing Maschine

Mehrspurige Turing Maschinen

- Man kann sich das Band der TM in k Spuren aufgeteilt vorstellen, wobei die Bandsymbole dann als k-Tupel aufgefasst werden.

§	1	0	1	1	1	1	$	B	...
B	B	B	B	1	0	1	B	B	...
B	1	0	0	1	0	1	B	B	...

Mehrspurige Turing Maschine

Turing Maschine

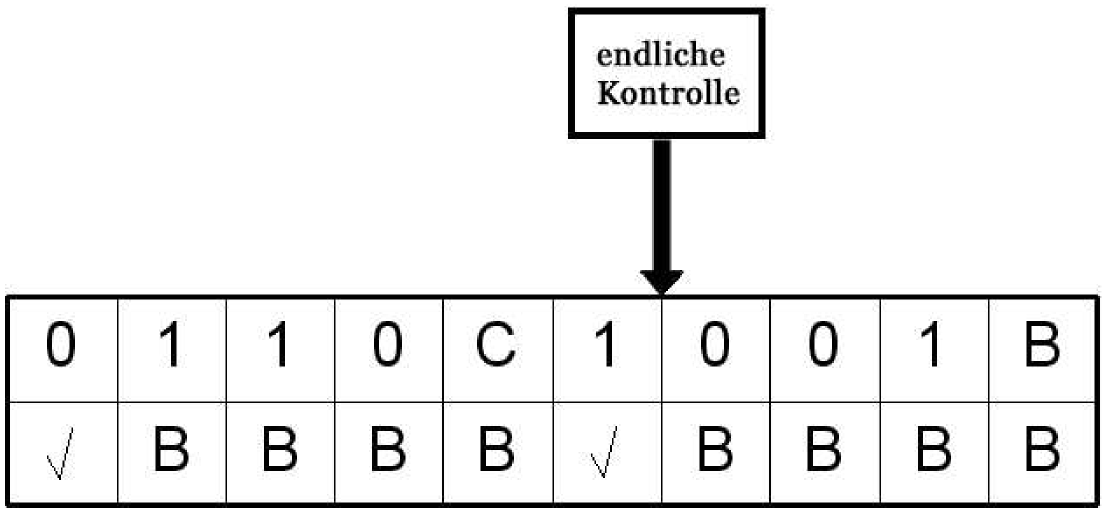
Mehrspurige Turing Maschine

- Es ist klar, wie die Tupel beispielsweise vor der ersten Bewegung der TM aussehen. Es handelt sich dabei um die Eingabesymbole, die $[\S, B, B], [0, B, B], [1, B, B]$ oder $[\$, B, B]$ sein können. Das Blanksymbol ist $[B, B, B]$.

Turing Maschine

Markierungssymbole

- Markierungssymbole sind ein gutes Hilfsmittel, um aufzuzeigen, wie eine TM Wörter erkennt, in denen bestimmte Zeichenketten sich wiederholen. Beispielsweise



TM mit Markierungssymbolen

Turing Maschine

Verschiebungen und Unterprogramme

- Um auf dem Band Platz zu schaffen, ist es der TM möglich alle Symbole des Bandes um eine bestimmte Anzahl von Plätzen nach rechts zu verschieben.
- In einigen Programmen ist es sinnvoll, teile die immer wieder benötigt werden als Unterprogramme zu konstruieren. Dazu schreiben wir das Unterprogramm so, dass es einen Anfangszustand und einen Rückkehrzustand hat. Der Rückkehrzustand beinhaltet im allgemeinen Fall keine Bewegung, sie wird erst beim Einfügen in ein anderes Programm bestimmt, da wir vorher nicht wissen, wo das entsprechende Programm nach Beendigung des Unterprogramms weiterarbeiten wird.

Turing Maschine

Modifikationen von Turing Maschinen

- Beidseitig unendliches Band
- Mehrbändige Turing Maschinen
- Nichtdeterministische Turing Maschinen
- Mehrdimensionale Turing Maschinen
- Turing Maschine mit mehreren Köpfen

Turing Maschine

Beidseitig unendliches Band

- Die formale Definition bleibt größtenteils zur oben genannten gleich. Die Ausnahmen sind folgende:

  Für $\delta(q, X) = (p, Y, L)$ gilt $qX\alpha \vdash_M pBY\alpha$ und für $\delta(q, X) = (p, B, R)$ gilt $qX\alpha \vdash_M p\alpha$.

- Im ersten Fall konnte ursprünglich keine Bewegung nach links gemacht werden, im zweiten Fall tauchte B links von p auf.

- Es gilt folgender Satz:

Satz: L wird von einer TM mit beidseitig unendlichem Band erkannt $\Leftrightarrow$ L wird von einer TM mit einem nach einer Seite unendlichen Band erkannt.

Turing Maschine

Beidseitig unendliches Band

- Beweis:
  Es gilt beide Richtungen zu beweisen:
  Seien M_1 die TM mit einseitig unendlichem Band und M_2 die TM mit beidseitig unendlichem Band.
  1. L wird von einer TM M_2 mit beidseitig unendlichem Band erkannt $\Leftarrow$ L wird von einer TM M_1 mit einem nach einer Seite unendlichen Band erkannt.
  Um M_1 durch M_2 zu simulieren, fügen wir einfach ein Zeichen auf dem Band von M_2 ein, das das Ende des Bandes in die linke Richtung bezeichnet. Wird dieses Symbol erreicht, so hält M_2 ohne zu akzeptieren, wie es auch M_1 tun würde.

Turing Maschine

Beidseitig unendliches Band

- 2. L wird von einer TM M_2 mit beidseitg unendlichem Band erkannt $\Rightarrow$ L wird von einer TM M_1 mit einem nach einer Seite unendlichen Band erkannt.
  Als Beweis konstruieren wir eine TM M_1, die M_2 simuliert. In dem M_1 einfach mit zwei Spuren ausgestattet wird, wobei auf der ersten Spur der Bandinhalt rechts und inklusive des Startsymbols von M_2 steht und auf der zweiten Spur im ersten Feld das Symbol $\S$ gefolgt vom Bandinhalt links des Startsymbols von M_2.

Turing Maschine

Beidseitig unendliches Band

- Sollte sich der Kopf, während er auf Spur 2 arbeitet, nach links über das Symbol § bewegen, so arbeitet die TM auf Spur 1 in die entgegengesetzte Richtung , also rechts, weiter.
  Dadurch, dass wir wissen, dass einzelne Spuren separat bearbeitet werden können, ist es klar, das damit M_2 simuliert werden kann.

Turing Maschine

Modifikationen von Turing Maschinen

- Ausblick:
  - Mehrbändige Turing Maschinen
  - Nichtdeterministische Turing Maschinen
  - Mehrdimensionale Turing Maschinen
  - Turing Maschine mit mehreren Köpfen
- Keine dieser Modifikationen erweitern die Mächtigkeit der TM. Die erkannte Sprachklasse bleibt die gleiche.