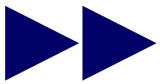
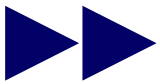
Theoretische Informatik
Kap 2: Berechnungstheorie



Gliederung der Vorlesung

Theoretische Informatik
Kap 2: Berechnungstheorie



Die Churchsche These

Qualitätskriterien (/* für Berechenbarkeitsmodelle */)

(1) jede im präzisierten Sinn berechenbare Funktion f sollte auch „intuitiv“ berechenbar sein

... man sollte sich davon überzeugen können, daß man einen Algorithmus A angeben kann, der f berechnet

(2) jede „intuitiv“ berechenbare Funktion f sollte auch im präzisierten Sinn berechenbar sein sein

... das ist der ungleich schwerere Teil, den man nicht so richtig in den Griff bekommt

Theoretische Informatik
Kap 2: Berechnungstheorie

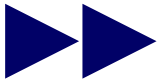
Die Churchsche These

Die Klasse der „intuitiv berechenbaren“ Funktionen über den natürlichen Zahlen stimmt mit der Klasse der Funktionen überein, die mit Turing-Maschinen berechnet werden können.

mit anderen Worten:

jeder Algorithmus läßt sich mit einer Turing-Maschine „realisieren“

Theoretische Informatik
Kap 2: Berechnungstheorie



Die Churchsche These

die Churchsche These

- kann nicht bewiesen werden
- es kann nur Evidenz für ihre Richtigkeit „gefunden“ werden

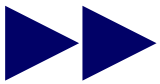
zugrunde liegende Problematik

- auf der einen Seite eine „unpräzise, intuitive“ Begriffsbildung
- auf der anderen Seite eine „mathematisch präzise“ Begriffsbildung

Evidenz

- alle bisher studierten „mathematisch präzisen“ Begriffsbildungen (/* d.h. alle Berechenbarkeitsmodelle */) sind „äquivalent“

Theoretische Informatik
Kap 2: Berechnungstheorie



Die Churchsche These

einige bekannte Berechenbarkeitsmodelle

- Turing-Maschinen
- partiell-rekursive Funktionen
- Markov-Algorithmen
- RAM
- Quantencomputer
- DNA-Computer
- ...

Für jede Funktion f über den natürlichen Zahlen gilt entweder (1) oder (2):

(1) f ist in jedem dieser Berechenbarkeitsmodelle berechenbar.

(2) f ist in keinem dieser Berechenbarkeitsmodelle berechenbar.

Theoretische Informatik
Kap 2: Berechnungstheorie

Die Churchsche These

zur Illustration

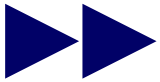
Jede Funktion über den natürlichen Zahlen, die man mit einer RAM berechnen kann, kann man auch mit einer TM berechnen.

sei R eine Random-Access-Maschine für eine Funktion f: $N^k \rightarrow N$

wir geben eine TM M an, die wie ein Interpreter für die RAM R funktioniert

... die zugrunde liegenden Ideen werden nur an einem Beispiel diskutiert

Theoretische Informatik
Kap 2: Berechnungstheorie



Die Churchsche These

Programm der RAM R

1. Load 1
2. Sub 2
3. If c(0) = 0 Goto 7
4. Load 2
5. Store 3
6. Goto 9
7. Load 1
8. Store 3
9. End

Zustände/Verhalten der TM M

z_0: Zustand zur Vorbereitung;
Folgezustand: z_1

z_1: Zustand zur Realisierung von 1;
Folgezustand: z_2

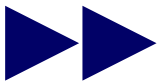
z_2: Zustand zur Realisierung von 2;
Folgezustand: z_3

z_3: Zustand zur Realisierung von 3;
Folgezustand: z_4 oder z_7

...

z_{10}: Zustand zur Nachbereitung;
Folgezustand: z_a

Theoretische Informatik
Kap 2: Berechnungstheorie



Die Churchsche These

Programm der RAM R

1. Load 1
2. Sub 2
3. If c(0) = 0 Goto 7
4. Load 2
5. Store 3
6. Goto 9
7. Load 1
8. Store 3
9. End

Zustände/Verhalten der TM M

Anfangsbandbelegung

$Bbin(x_1)Bbin(x_2)B$

im Zustand z_0 erzeugt M folgende Bandbelegung und geht in z_1:

$\#_0 0 \#_1 bin(x_1) \#_2 bin(x_2) \#_3 0 \#$

Hinweis: R berechnet eine zweistellige Funktion

Hinweis: R benutzt nur die Register c(0),c(1), c(2) und c(3)

Die Churchsche These

Programm der RAM R

1. Load 1
2. Sub 2
3. If c(0) = 0 Goto 7
4. Load 2
5. Store 3
6. Goto 9
7. Load 1
8. Store 3
9. End

Zustände/Verhalten der TM M

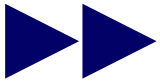
aktuelle Bandbelegung

$\#_0 0 \#_1 bin(x_1) \#_2 bin(x_2) \#_3 0 \#$

im Zustand z_1 erzeugt M folgende Bandbelegung und geht in z_2:

$\#_0 bin(x_1) \#_1 bin(x_1) \#_2 bin(x_2) \#_3 0 \#$

Theoretische Informatik
Kap 2: Berechnungstheorie



Die Churchsche These

Programm der RAM R

1. Load 1
2. Sub 2
3. If c(0) = 0 Goto 7
4. Load 2
5. Store 3
6. Goto 9
7. Load 1
8. Store 3
9. End

Zustände/Verhalten der TM M

aktuelle Bandbelegung

$\#_0 bin(x_1) \#_1 bin(x_1) \#_2 bin(x_2) \#_3 0 \#$

im Zustand z_2 erzeugt M folgende Bandbelegung und geht in z_3:

$\#_0 bin(x_1 - x_2) \#_1 bin(x_1) \#_2 bin(x_2) \#_3 0 \#$

Die Churchsche These

Programm der RAM R

1. Load 1
2. Sub 2
3. If c(0) = 0 Goto 7
4. Load 2
5. Store 3
6. Goto 9
7. Load 1
8. Store 3
9. End

Zustände/Verhalten der TM M

aktuelle Bandbelegung

$\#_0 bin(x_1\text{-}x_2)\#_1 bin(x_1)\#_2 bin(x_2)\#_3 0\#$

im Zustand z_3 prüft M, ob zwischen $\#_0$ und $\#_1$ genau eine 0 steht; falls ja, so geht M in z_4, sonst in z_7

Theoretische Informatik
Kap 2: Berechnungstheorie

Die Churchsche These

Programm der RAM R

1. Load 1
2. Sub 2
3. If c(0) = 0 Goto 7
4. Load 2
5. Store 3
6. Goto 9
7. Load 1
8. Store 3
9. End

Zustände/Verhalten der TM M

aktuelle Bandbelegung

$\#_0 bin(x_1\text{-}x_2)\#_1 bin(x_1)\#_2 bin(x_2)\#_3 0\#$

im Zustand z_4 erzeugt M folgende Bandbelegung und geht in z_5:

$\#_0 bin(x_2)\#_1 bin(x_1)\#_2 bin(x_2)\#_3 0\#$

Die Churchsche These

Programm der RAM R

1. Load 1
2. Sub 2
3. If c(0) = 0 Goto 7
4. Load 2
5. Store 3
6. Goto 9
7. Load 1
8. Store 3
9. End

Zustände/Verhalten der TM M

aktuelle Bandbelegung:

$\#_0 bin(x_1) \#_1 bin(x_1) \#_2 bin(x_2) \#_3 0 \#$

im Zustand z_5 erzeugt M folgende Bandbelegung und geht in z_6:

$\#_0 bin(x_2) \#_1 bin(x_1) \#_2 bin(x_2) \#_3 bin(x_2) \#$

Die Churchsche These

Programm der RAM R

1. Load 1
2. Sub 2
3. If c(0) = 0 Goto 7
4. Load 2
5. Store 3
6. Goto 9
7. Load 1
8. Store 3
9. End

Zustände/Verhalten der TM M

aktuelle Bandbelegung:

$\#_0 bin(x_2) \#_1 bin(x_1) \#_2 bin(x_2) \#_3 bin(x_2) \#$

im Zustand z_6 geht M in z_9

Die Churchsche These

Programm der RAM R

1. Load 1
2. Sub 2
3. If c(0) = 0 Goto 7
4. Load 2
5. Store 3
6. Goto 9
7. Load 1
8. Store 3
9. End

Zustände/Verhalten der TM M

aktuelle Bandbelegung:

$\#_0 bin(x_1\text{-}x_2)\#_1 bin(x_1)\#_2 bin(x_2)\#_3 0\#$

im Zustand z_7 erzeugt M folgende Bandbelegung und geht in z_8

$\#_0 bin(x_1)\#_1 bin(x_1)\#_2 bin(x_2)\#_3 0\#$

Die Churchsche These

Programm der RAM R

1. Load 1
2. Sub 2
3. If c(0) = 0 Goto 7
4. Load 2
5. Store 3
6. Goto 9
7. Load 1
8. Store 3
9. End

Zustände/Verhalten der TM M

aktuelle Bandbelegung:

$\#_0 bin(x_1) \#_1 bin(x_1) \#_2 bin(x_2) \#_3 0 \#$

im Zustand z_8 erzeugt M folgende Bandbelegung und geht in z_9

$\#_0 bin(x_1) \#_1 bin(x_1) \#_2 bin(x_2) \#_3 bin(x_1) \#$

Die Churchsche These

Programm der RAM R

1. Load 1
2. Sub 2
3. If c(0) = 0 Goto 7
4. Load 2
5. Store 3
6. Goto 9
7. Load 1
8. Store 3
9. End

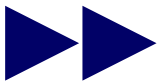
Zustände/Verhalten der TM M

aktuelle Bandbelegung:

$\#_0 bin(y) \#_1 bin(x_1) \#_2 bin(x_2) \#_3 bin(y) \#$

im Zustand z_9 erzeugt M folgende Bandbelegung und geht in z_a

$B bin(y) B$



Die Churchsche These

Zwischenfazit

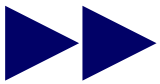
- die Idee, zu einer RAM R eine TM M zu konstruieren, die wie ein Interpreter für R arbeitet, funktioniert immer
- entscheidend ist, daß sich jeder einzelne Befehl einer RAM auf einer TM realisieren läßt

... es gibt nur endlich viele verschiedene Befehlstypen für eine RAM

- Transportbefehle
- bedingte bzw. unbedingte Sprüng
- arithmetische Operationen bzw. Vergleichsoperationen

... jeder Befehlstyp kann auf einer „passenden“ Basis-TM realisiert werden

... die zugehörige TM M entsteht einfach durch Kombination dieser Basis-TM



Die Churchsche These

eine wichtige Beobachtung

- jede Basis-TM benötigt – in Abhängigkeit von der Länge der Eingaben – nur polynomiell viele Rechenschritte, um das relevante Zwischenergebnis zu bestimmen

Konsequenz

- wenn die gegebene RAM R – in Abhängigkeit von der Länge der Eingaben – nur polynomiell viele Rechenschritte benötigt, um das Ergebnis zu berechnen, so kommt die zugehörige TM M auch mit polynomiell vielen Rechenschritten aus

Theoretische Informatik
Kap 2: Berechnungstheorie

Die Churchsche These

zentrale Folgerung

Es sei f: N^k → N eine Funktion.

Wenn f nicht auf einer TM in polynomieller Zeit berechnet werden kann, so kann f auch nicht auf einer RAM in polynomieller Zeit berechnet werden.

... wichtig für Untersuchungen in der Komplexitätstheorie

Anmerkung: dieser Zusammenhang gilt für alle Berechenbarkeitsmodelle; Ausnahme: Quantencomputer und DNA-Computer