

# Grundlagen der Theoretischen Informatik

## Turingmaschinen und rekursiv aufzählbare Sprachen (IV)

6.07.2016

Viorica Sofronie-Stokkermans

e-mail: [sofronie@uni-koblenz.de](mailto:sofronie@uni-koblenz.de)

# Übersicht

---

1. Motivation
2. Terminologie
3. Endliche Automaten und reguläre Sprachen
4. Kellerautomaten und kontextfreie Sprachen
5. Turingmaschinen und rekursiv aufzählbare Sprachen
6. Berechenbarkeit, (Un-)Entscheidbarkeit
7. Komplexitätsklassen P und NP

# Übersicht

---

- Determinierte Turing-Maschinen (DTMs)
- Varianten von Turing-Maschinen
- Indeterminierte Turing-Maschinen (NTMs)
- Universelle determinierte Turing-Maschinen
- Entscheidbar/Aufzählbar
- Determinierte Turing-Maschinen entsprechen Typ 0
- Unentscheidbarkeit

# Variationen von Turing-Maschinen

---

## Bis jetzt:

- Standard-DTM
- Turing-Maschinen, die nie hängen
- DTM mit zweiseitig unbeschränktem Band (zw-DTM)

# DTM mit zweiseitig unbeschränktem Band

---

## Definition (DTM mit zweiseitig unbeschränktem Band, zw-DTM)

Eine **Turing-Maschine mit zweiseitig unbeschränktem Band (zw-DTM)** ist eine DTM, für die die Begriffe der Konfiguration und der Nachfolgekonfiguration wie folgt definiert sind:

Eine **Konfiguration**  $C$  einer zw-DTM  $\mathcal{M} = (K, \Sigma, \delta, s)$  ist von der Form

$$C = q, w\underline{a}u$$

Dabei ist

- $q \in K \cup \{h\}$  der aktuelle Zustand,
- $w \in (\Sigma - \{\#\})\Sigma^* \cup \{\epsilon\}$  der Bandinhalt links des Kopfes,
- $a \in \Sigma$  das Zeichen unter dem Kopf, und
- $u \in \Sigma^*(\Sigma - \{\#\}) \cup \{\epsilon\}$  der Bandinhalt rechts des Kopfes.

# DTM mit zweiseitig unbeschränktem Band

---

## Theorem (Simulation von zw-DTM durch DTM)

Zu jeder zw-DTM  $\mathcal{M}$ , die eine Funktion  $f$  berechnet oder eine Sprache  $L$  akzeptiert, existiert eine Standard-DTM  $\mathcal{M}'$ , die ebenfalls  $f$  berechnet bzw.  $L$  akzeptiert.

**Beweis** Sei  $w = a_1 \dots a_n$  die Eingabe für  $\mathcal{M} = (K, \Sigma, \delta, s)$ .

Dann sieht das beidseitig unendliche Band zu Beginn der Rechnung so aus:

$\dots \#\#\#a_1 \dots a_n\#\#\dots$

# DTM mit zweiseitig unbeschränktem Band

---

Beweis (Forts.) **Idee:**

- $\mathcal{M}$  hat quasi zwei unendlich lange Halbbänder.
- Ziel ist, den Inhalt beider Halbbänder von  $\mathcal{M}$  auf einem unterzubringen.
- Dazu: Den Teil des Bandes, der zwei Zeichen links vom Input  $w$  beginnt, umklappen:

$$\begin{array}{l} \text{Spur 1 } \# \# \dots \# \# \\ \text{Spur 2 } \# a_1 \dots a_n \underline{\#} \dots \end{array}$$

- Die DTM  $\mathcal{M}'$  hat zwei **Spuren**, d.h. zwei Reihen von Zeichen, die auf demselben Band untergebracht (kodiert) sind.
- Das Bandalphabet von  $\mathcal{M}'$  ist  $\Sigma' \supseteq \Sigma \times \Sigma$ .

# DTM mit zweiseitig unbeschränktem Band

---

Beweis (Forts.) Sei  $\mathcal{M}' = (K', \Sigma', \delta', s)$ .  $\mathcal{M}'$  rechnet so:

- $\mathcal{M}'$  legt zunächst eine zweite Spur an,
- simuliert dann die Arbeit von  $\mathcal{M}$ , und
- transformiert dann das Ergebnis wieder auf nur eine Spur herunter.



# DTM mit zweiseitig unbeschränktem Band

---

Beweis (Forts.) Erste Phase der Rechnung:

$\mathcal{M}'$  rechnet

$$s, \#a_1 \dots a_n \underline{\#} \vdash_{\mathcal{M}'}^* q, \$ \begin{array}{c} \# \# \dots \# \# \\ \# a_1 \dots a_n \underline{\#} \end{array} \# \dots$$

Die zweite Spur wird nur so weit wie nötig angelegt.

Mit dem Symbol \$ markiert  $\mathcal{M}'$  das Ende des Halbbands, damit sie nicht hängenbleibt.

# DTM mit zweiseitig unbeschränktem Band

---

Beweis (Forts.) **Zweite Phase der Rechnung:**

$\mathcal{M}'$  simuliert  $\mathcal{M}$ .

Dabei muß sie sich immer merken, auf welcher der beiden Spuren sie gerade arbeitet.

Deshalb definieren wir  $K' \supseteq K \times \{1, 2\}$ .

$(q, i)$  bedeutet, dass die simulierte Maschine  $\mathcal{M}$  im Zustand  $q$  ist und  $\mathcal{M}'$  auf Spur  $i$  arbeitet.

# DTM mit zweiseitig unbeschränktem Band

---

Beweis (Forts.) Für die Simulation von  $\mathcal{M}$  durch  $\mathcal{M}'$  soll nun gelten:

$\mathcal{M}$  erreicht von  $s, \# \dot{\vdots} \# w \underline{\#}$  aus eine Konfiguration  $q, u_1 b \dot{\vdots} a u_2$

gdw

$\mathcal{M}'$  rechnet  $p, \$ \begin{matrix} \# & \dots & \# \\ \# & w & \underline{\#} \end{matrix} \vdash_{\mathcal{M}'}^* p', \$ \begin{matrix} b & u_1^R & \# \\ a & u_2 & \# \end{matrix} \dots \# \#$

(  $\dot{\vdots}$  steht in beiden Konf. zwischen denselben zwei Bandpositionen (an denen das Band “umklappt”) )

# DTM mit zweiseitig unbeschränktem Band

---

Beweis (Forts.)  $\mathcal{M}'$  simuliert  $\mathcal{M}$  wie folgt:

- Wenn  $\mathcal{M}'$  das Zeichen \$ erreicht, wechselt sie die Spur.
- Wenn die simulierte Maschine  $\mathcal{M}$  nach rechts (links) geht, geht  $\mathcal{M}'$  nach rechts (links) auf Spur 2 und nach links (rechts) auf Spur 1.
- Wenn  $\mathcal{M}'$  ein # erreicht (d.h. sie erreicht den Bandteil, wo noch nicht zwei Spuren angelegt sind), macht sie daraus  $\begin{matrix} \# \\ \# \end{matrix}$ .

Gilt etwa  $\delta_{\mathcal{M}}(q, a) = (q', L)$ , so muß in  $\mathcal{M}$  gelten:

- $\delta_{\mathcal{M}'}((q, 2), \begin{matrix} x \\ a \end{matrix}) = ((q', 2), L)$  für alle möglichen  $x$ ,
- $\delta_{\mathcal{M}'}((q, 1), \begin{matrix} a \\ x \end{matrix}) = ((q', 1), R)$  (auf der oberen Spur ist der Inhalt des “linken Halbbandes” revers notiert, deshalb muß hier die Laufrichtung entgegengesetzt sein).

# DTM mit zweiseitig unbeschränktem Band

---

Beweis (Forts.)

Außerdem gilt immer:

- $\delta_{\mathcal{M}'}((q, 1), \$) = ((q, 2), R)$
- $\delta_{\mathcal{M}'}((q, 2), \$) = ((q, 1), R)$   
Spurwechsel beim Überschreiten von \$
- $\delta_{\mathcal{M}'}((q, i), \#) = (q, i), \#)$   
Erzeugen eines neuen Doppelspurstücks
- etc.

# DTM mit zweiseitig unbeschränktem Band

---

Beweis (Forts.)

Wenn dann  $\mathcal{M}$  mit  $h, \underline{u\#}$  hält, dann erreicht  $\mathcal{M}'$  eine Konfiguration, die eine der folgenden Formen hat:

$$(i) \quad (h, 1), \$ \begin{array}{c} \# \dots \overline{\#} \\ \# \dots \# \end{array} u^R \begin{array}{c} \# \dots \# \\ \# \dots \# \end{array} \text{ oder}$$

$$(ii) \quad (h, 2), \$ \begin{array}{c} \# \dots \# \\ \# \dots \# \end{array} \begin{array}{c} \# \dots \# \\ u \\ \# \end{array} \dots \# \text{ oder}$$

$$(iii) \quad (h, 2), \$ \begin{array}{c} u_1^R \\ u_2\# \end{array} \begin{array}{c} \# \dots \# \\ \# \end{array} \text{ mit } u_1 u_2 = u.$$

Bei Konfigurations-Form (iii) kann entweder das  $u_1^R$  über das  $u_2$  “hinausragen” oder umgekehrt.

# DTM mit zweiseitig unbeschränktem Band

---

Beweis (Forts.) **Dritte Phase der Rechnung:**

Die Simulation von  $\mathcal{M}$  ist abgeschlossen.

$\mathcal{M}'$  muß nun den Bandinhalt von zwei Spuren auf nur eine heruntertransformieren, um danach die Konfiguration  $h, \#u\#$  zu erreichen.

- $\mathcal{M}'$  macht zunächst alle  $\#$  rechts vom beschriebenen Bandteil zu  $\#$ . Für Fall (i) und (ii) löscht sie die  $\#$  links von  $u^R$  bzw.  $u$ .
- Für Fall (iii) schiebt  $\mathcal{M}'$  dann die untere Spur nach links, bis sie eine Konfiguration  $q, \# \overset{u_1^R}{\dots} \# u_2 \#$  erreicht.
- Für Fall (i) und (iii) muß  $\mathcal{M}'$  jetzt  $u_1^R$  bzw.  $u^R$  auf nur eine Spur transformieren und zugleich invertieren, sie muß also für den allgemeineren Fall (iii)  $q, \$ \# \overset{u_1^R}{\dots} \# u_2 \# \vdash_{\mathcal{M}'}^* q', \$ u_1 u_2 \#$  rechnen.
- Danach muß  $\mathcal{M}'$  nur noch das  $\$$  links löschen und nach rechts neben  $u$  laufen.

# DTM mit zweiseitig unbeschränktem Band

---

Damit hat die Standard-DTM  $\mathcal{M}'$  die Arbeitsweise der zw-DTM  $\mathcal{M}$  vollständig simuliert.

**Also kann man mit zw-Turing-Maschinen nicht mehr berechnen als mit Standard DTM'n.  $\square$**



# Variationen von Turing-Maschinen

---

- Standard-DTM
- Turing-Maschinen, die nie hängen
- DTM mit zweiseitig unbeschränktem Band (zw-DTM)
- DTM mit  $k$  Halbbändern

# DTM mit $k$ Halbbändern

## Definition (DTM mit $k$ Halbbändern, $k$ -DTM)

Eine **Turing-Maschine**  $\mathcal{M} = (K, \Sigma_1, \dots, \Sigma_k, \delta, s)$  mit  $k$  Halbbändern (mit je einem Kopf) ist eine Turing-Maschine mit einer Übergangsfunktion

$$\delta : K \times \Sigma_1 \times \dots \times \Sigma_k \rightarrow (K \cup \{h\}) \times (\Sigma_1 \cup \{L, R\}) \times \dots \times (\Sigma_k \cup \{L, R\})$$

Eine **Konfiguration** einer  $k$ -Turing-Maschine hat die Form

$$C = q, w_1 \underline{a_1} u_1, \dots, w_k \underline{a_k} u_k.$$

# DTM mit $k$ Halbbändern

---

## DTM mit $k$ Halbbändern

- Die Köpfe einer  $k$ -DTM können sich **unabhängig** bewegen (sonst hätten wir nur eine DTM mit  $k$  Spuren).
- Die Definition der Nachfolgekonfiguration verläuft analog zu der Definition bei Standard-DTM.
- Für eine  $k$ -DTM, die eine Funktion  $f : \Sigma_0^m \rightarrow \Sigma_0^n$  berechnet, legen wir fest, dass sowohl die  $m$  Eingabewerte als auch – nach der Rechnung – die  $n$  Ergebniswerte auf dem ersten Band stehen sollen.
- Es übertragen sich alle Begriffe wie *berechenbar*, *entscheidbar* etc. kanonisch auf  $k$ -DTM.

# DTM mit $k$ Halbbändern

---

## **Theorem [Simulation von $k$ -DTM durch DTM]**

Zu jeder  $k$ -DTM  $\mathcal{M}$ , die eine Funktion  $f$  berechnet (resp. eine Sprache  $L$  akzeptiert), existiert eine DTM  $\mathcal{M}'$ , die ebenfalls  $f$  berechnet (resp.  $L$  akzeptiert).

# DTM mit $k$ Halbbändern

---

## Beweis (Skizze)

Wir arbeiten mit einer Turing-Maschine mit mehreren Spuren.

Um eine  $k$ -DTM zu simulieren, verwenden wir  $2k$  Spuren, also Bandzeichen, die aus  $2k$  übereinander angeordneten Buchstaben bestehen.

- In den Spuren mit ungerader Nummer stehen die Inhalte der  $k$  Bänder von  $\mathcal{M}$ .
- Die Spuren mit gerader Nummer verwenden wir, um die Positionen der Köpfe von  $\mathcal{M}$  zu simulieren:  
Die  $2i$ -te Spur enthält an genau einer Stelle ein  $\wedge$ , nämlich da, wo  $\mathcal{M}$  gerade seinen  $i$ -ten Kopf positioniert hätte, und ansonsten nur Blanks.

$\mathcal{M}'$  kodiert zunächst die Eingabe von  $\mathcal{M}$ . Dann simuliert  $\mathcal{M}'$  die Maschine  $\mathcal{M}$ . Am Ende der Rechnung wird noch die Ausgabe dekodiert.

# Übersicht

---

- Determinierte Turing-Maschinen (DTMs)
- Varianten von Turing-Maschinen
- **Indeterminierte Turing-Maschinen (NTMs)**
- Universelle determinierte Turing-Maschinen
- Entscheidbar/Aufzählbar
- Determinierte Turing-Maschinen entsprechen Typ 0
- Unentscheidbarkeit

# Indeterminierte Turing-Maschinen (NTMs)

---

## Definition (Indeterminierte Turing-Maschine, NTM)

Eine **indeterminierte Turing-Maschine**  $\mathcal{M}$  ist ein Tupel

$$M = (K, \Sigma, \Delta, s)$$

Dabei sind  $K$ ,  $\Sigma$ ,  $s$  definiert wie bei determinierten Turing-Maschinen.

**Übergangsrelation:**

$$\Delta \subseteq (K \times \Sigma) \times ((K \cup \{h\}) \times (\Sigma \cup \{L, R\}))$$

# Indeterminierte Turing-Maschinen (NTMs)

---

## Mehrere Nachfolgekonfigurationen

Konfigurationen sind definiert wie bei DTMs.

**Nun kann eine Konfiguration aber mehrere mögliche Nachfolgekonfigurationen haben.**

**Definition (NTM: Halten, Hängen, Akzeptieren)** Sei  $\mathcal{M} = (K, \Sigma, \Delta, s_0)$  eine indeterminierte Turing-Maschine.

- $\mathcal{M}$  **hält** bei Input  $w$ , falls es **unter den möglichen Rechnungen**, die  $\mathcal{M}$  wählen kann, **eine gibt**, so dass  $\mathcal{M}$  eine Haltekonfiguration erreicht.
- $\mathcal{M}$  **hängt** in einer Konfiguration, wenn es keine (durch  $\Delta$  definierte) Nachfolgekonfiguration gibt.
- $\mathcal{M}$  **akzeptiert** ein Wort  $w$ , falls sie **von  $s, \#w\#$  aus einen Haltezustand erreichen kann**, und  $\mathcal{M}$  akzeptiert eine Sprache  $L$ , wenn sie genau alle Wörter  $w \in L$  akzeptiert.



# Indeterminierte Turing-Maschinen (NTMs)

---

## Bemerkung

Wenn es nicht nur darauf ankommt, ob die Maschine hält, sondern auch mit welchem Bandinhalt:

**Welche der vielen Haltekonfigurationen sollte dann gelten?**

Um dies Problem zu umgehen, übertragen wir die Begriffe des *Entscheidens* und *Aufzählens* **nicht** auf NTM. Im Allgemeinen verwendet man NTM auch nicht dazu, Funktionen zu berechnen.

# Indeterminierte Turing-Maschinen (NTMs)

---

## Wie rechnet eine indeterminierte Turing-Maschine?

- Die Regeln einer determinierten DTM kann man sich als Programm (aus sehr einfachen Schritten) vorstellen.
- **Bei NTM ist das anders!**
- **Eine NTM ist nicht einfach eine Maschine, die immer richtig rät!**

Dieselbe Diskussion hatten wir bei indeterminierten endlichen Automaten.

# Indeterminierte Turing-Maschinen (NTMs)

---

## Vorstellung von einer NTM

- Korrekte Vorstellung:
  - Übergänge von Konfiguration zu Nachfolgekongfiguration entspr. Regelmenge
  - **plus Suchverfahren!**

oder: Eine NTM beschreitet alle möglichen Rechenwege parallel.

- Eine NTM akzeptiert ein Wort, wenn es mindestens einen Berechnungsweg gibt, der in einer Haltekonfiguration endet.
- Sprechweise **“Die NTM rät”**: Wir verwenden diese Sprechweise, sie ist aber mit Vorsicht zu genießen!

# Indeterminierte Turing-Maschinen (NTMs)

---

## Beispiel.

Sei

$$L = \{|^n \mid n \text{ ist nicht prim und } n \geq 2\}$$

Eine NTM kann diese Sprache wie folgt akzeptieren:

1. Eine Zahl „raten“ und (nach rechts) aufs Band schreiben.
2. Noch eine Zahl „raten“ und daneben schreiben.
3. Die beiden Zahlen miteinander multiplizieren.
4. Das Ergebnis mit der Eingabe vergleichen.
5. Genau dann, wenn beide gleich sind, anhalten

# Indeterminierte Turing-Maschinen (NTMs)

---

## Theorem [Simulation von NTM durch DTM]

Jede Sprache, die von einer indeterminierten Turing-Maschine akzeptiert wird, wird auch von einer Standard-DTM akzeptiert.

Beweis (Anfang) Sei

- $L$  eine Sprache über  $\Sigma_0^*$  mit  $\# \notin \Sigma_0$ ;
- $\mathcal{M} = (K, \Sigma, \Delta, s)$  eine indeterminierte Turing-Maschine, die  $L$  akzeptiert.

# Indeterminierte Turing-Maschinen (NTMs)

---

Beweis (Fortsetzung) Wir konstruieren zu  $\mathcal{M}$  eine Standard-DTM  $\mathcal{M}'$ , die so rechnet:

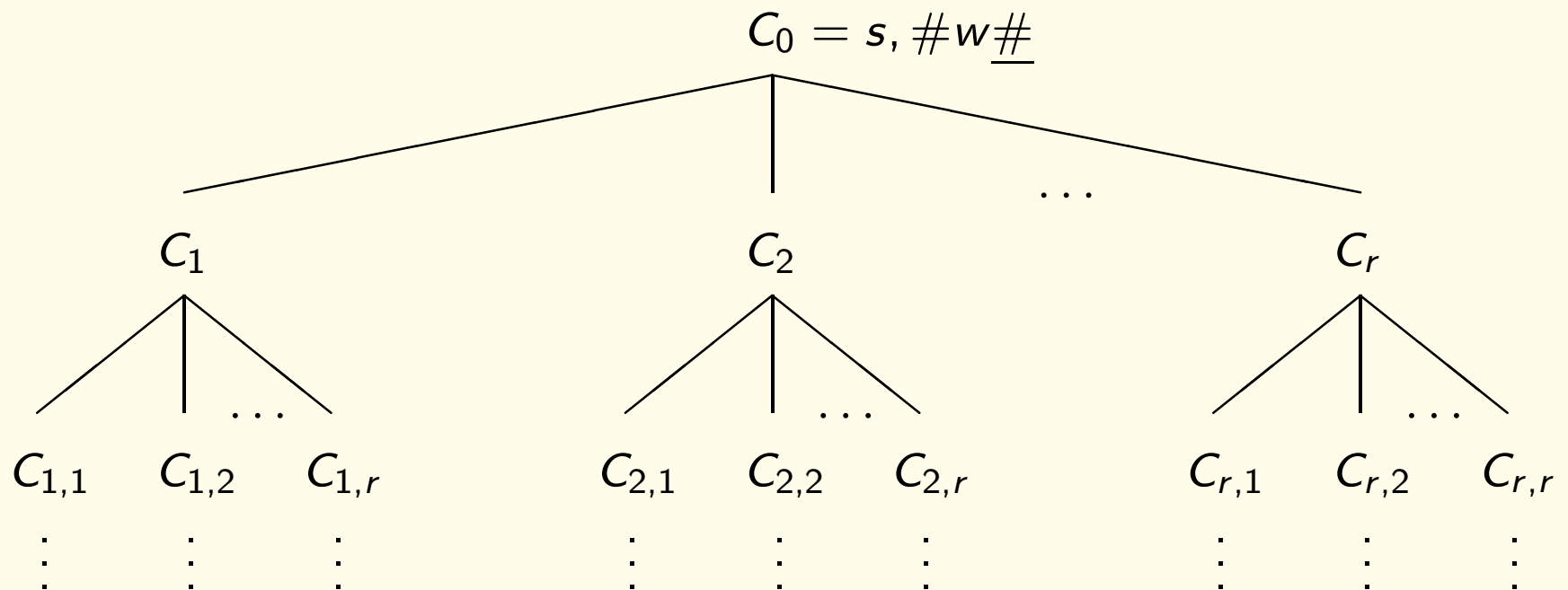
- $\mathcal{M}'$  durchläuft systematisch **alle** Rechnungen von  $\mathcal{M}$ , und sucht dabei nach einer Haltekonfiguration.
- $\mathcal{M}'$  hält genau dann, wenn sie eine Haltekonfiguration von  $\mathcal{M}$  findet.

# Indeterminierte Turing-Maschinen (NTMs)

Beweis (Fortsetzung) Suchbaum, Rechnungsbaum:

Stelle alle Rechnungen von  $\mathcal{M}$  von einer Startkonfiguration  $C_0$  dar als einen Baum mit Wurzel  $C_0$ .

Ein Ast ist eine mögliche Rechnung von  $\mathcal{M}$ .



# Indeterminierte Turing-Maschinen (NTMs)

---

Beweis (Fortsetzung)

**Problem:**

Es kann zur Startkonfiguration

$$C_0 = s, \#w\underline{\#}$$

unendlich viele Rechnungen von  $\mathcal{M}$  geben,  
und jede einzelne von ihnen kann unendlich lang sein.

Wir können also nicht erst einen Ast ganz durchlaufen und dann den nächsten Ast durchsuchen.



# Indeterminierte Turing-Maschinen (NTMs)

---

Beweis (Fortsetzung)

**Die Lösung:**

Breitensuche

Durchlaufe den Rechnungsbaum nicht depth-first, sondern per **iterative deepening**.

- Untersuchen alle möglichen Rechnungen bis zum ersten Schritt.
- Untersuchen alle möglichen Rechnungen bis zum zweiten Schritt.
- Untersuchen alle möglichen Rechnungen bis zum dritten Schritt.
- usw.

# Indeterminierte Turing-Maschinen (NTMs)

---

Beweis (Fortsetzung)

Können wir damit denn in endlicher Zeit eine Haltekonfiguration finden, falls es eine gibt?

**Problem:**

Kann der Rechnungsbaum nicht nur **unendlich tief**, sondern auch **unendlich breit** werden?

**Nein**, denn:

Maximale Anzahl von Nachfolgekonfigurationen

$$r = \max\{|\Delta(q, a)| \mid q \in K, a \in \Sigma\}$$

# Indeterminierte Turing-Maschinen (NTMs)

---

Beweis (Fortsetzung)

$\mathcal{M}'$  kann (z.B.) als eine 3-DTM gewählt werden:

- **Auf dem ersten Band steht immer das Eingabewort  $w$ .**

Da die Rechnung immer wieder neu mit  $s, \#w\#$  von  $\mathcal{M}$  beginnt, wird das Eingabewort immer wieder gebraucht.

- **Auf dem zweiten Band steht, welcher Weg durch den Rechnungsbaum gerade verfolgt wird.**

Der Einfachheit halber: Wenn eine Konfiguration weniger als  $r$  Nachfolgekongfigurationen hat, soll der zugehörige Knoten trotzdem  $r$  Söhne haben, und die überzähligen Konfigurationen sind leer.

# Indeterminierte Turing-Maschinen (NTMs)

---

Beweis (Fortsetzung)

Darstellung des aktuellen Pfades im Rechnungsbaum als Zahl im  $r$ -adischen System.

Eine Zahl  $d_1 \dots d_n$  bedeutet:

- Von der Startkonfiguration  $C_0$  aus ist die  $d_1$ -te der  $r$  möglichen Nachfolgekongfigurationen gewählt worden,  $C_{d_1}$ .
- Von  $C_{d_1}$ , einem Knoten der Tiefe 1, aus wurde die  $d_2$ -te mögliche Nachfolgekongfiguration gewählt,
- usw.

# Indeterminierte Turing-Maschinen (NTMs)

---

Beweis (Fortsetzung)

Ausführung des Iterative Deepening:

- Beginne mit 0 auf zweitem Band.
- Jeweils nächste zu betrachtende Rechnung erhöhen der Zahl auf Band 2 um 1
- Auf Band 3 wird eine Rechnung von  $\mathcal{M}$  **determiniert** simuliert.  
Und zwar entsprechend der Zahl  $d_1 \dots d_n$  auf Band 2.  
Die Endkonfiguration  $C_{d_1 \dots d_n}$  dieser Rechnung steht im Rechenbaum an dem Knoten, der das Ende des Pfades  $d_1 \dots d_n$  bildet.
- Ist die Konfiguration  $C_{d_1 \dots d_n}$  eine Haltekonfiguration, so hält  $\mathcal{M}'$ .
- Sonst Zahl auf Band 2 erhöhen und die nächste Rechnungssimulation beginnen

# Indeterminierte Turing-Maschinen (NTMs)

---

Beweis (Ende)

Damit gilt:

$\mathcal{M}'$  hält bei Input  $w$

gdw

es gibt in  $R_{C_0}$  eine Haltekonfiguration.

Das ist genau dann der Fall, wenn

- $\mathcal{M}$  bei Input  $w$  hält,
- $w$  in  $L$  liegt.

□

# Übersicht

---

- Determinierte Turing-Maschinen (DTMs)
- Varianten von Turing-Maschinen
- Indeterminierte Turing-Maschinen (NTMs)
- **Universelle determinierte Turing-Maschinen**
- Entscheidbar/Aufzählbar
- Determinierte Turing-Maschinen entsprechen Typ 0
- Unentscheidbarkeit

# Universelle determinierte Turing-Maschinen

---

Vergleich Turing-Maschine / „normaler“ Computer Turing-Maschinen sind sehr mächtig.

**Wie mächtig sind sie wirklich?**

- Eine Turing-Maschine hat eine vorgegebenes „Programm“ (Regelmenge)
- „Normale“ Computer können beliebige Programme ausführen.

**Tatsächlich geht das mit Turing-Maschinen auch!**



# Universelle determinierte Turing-Maschinen

---

## Turing-Maschine, die andere TMen simuliert

- Universelle TM  $\mathcal{U}$  bekommt als Eingabe:
  - die Regelmenge einer beliebigen Turing-Maschine  $\mathcal{M}$  und
  - ein Wort  $w$ , auf dem  $\mathcal{M}$  rechnen soll.
- $\mathcal{U}$  simuliert  $\mathcal{M}$ , indem sie jeweils nachschlägt, welchen  $\delta$ -Übergang  $\mathcal{M}$  machen würde.

# Universelle determinierte Turing-Maschinen

---

## TM als Eingabe für eine andere TM

### Frage:

In welches Format fasst man die Regeln einer DTM  $\mathcal{M}$  am besten, um sie einer universellen DTM als Eingabe zu geben?

### Was muss man angeben, um eine DTM komplett zu beschreiben?

- das Alphabet,
- die Zustände,
- die  $\delta$ -Übergänge
- den Startzustand.

# Universelle determinierte Turing-Maschinen

---

## Standardisierung von Alphabet, Zustandsmenge, Startzustand

- Unendliches Alphabet  $\Sigma_\infty = \{a_0, a_1, \dots\}$ ,  
so dass das Alphabet jeder DTM eine Teilmenge von  $\Sigma_\infty$  ist.
- Namen der Zustände einer DTM sind egal.  
Sie seien also  $q_1, \dots, q_n$   
( $n$  kann dabei von DTM zu DTM verschieden sein).
- Sei  $q_1$  immer der Startzustand, und  
bezeichne  $q_0$  den Haltezustand

Damit:

**Wir können eine DTM komplett beschreiben,  
indem wir nur ihre  $\delta$ -Übergänge beschreiben.**

# Universelle determinierte Turing-Maschinen

---

**(Mögliche) Kodierung der Übergangsrelation** Die DTM  $\mathcal{L}_\#$  habe die Regeln

$$q_1, \# \mapsto q_2, L \quad q_2, \# \mapsto h, \#$$

$$q_1, | \mapsto q_2, L \quad q_2, | \mapsto q_2, L$$

Dabei sei:  $\# = a_0$  und  $| = a_1$

Dann kann die DTM  $\mathcal{L}_\#$  so beschrieben werden:

	$a_0$	$a_1$
$q_1$	$q_2, L$	$q_2, L$
$q_2$	$h, a_0$	$q_2, L$

oder kürzer:

$Z S 2\lambda S 2\lambda$

$Z S 00 S 2\lambda$

# Universelle determinierte Turing-Maschinen

---

**(Mögliche) Kodierung der Übergangsrelation** Dabei steht:

- $Z$  für “nächste Zeile”
- $S$  für “nächste Spalte”
- $\lambda$  für “links”,  $\rho$  für “rechts”
- die Zahl  $n$  für den  $n$ -ten Zustand und für das  $n$ -te Zeichen von  $\Sigma_\infty$ .

Damit ist die DTM insgesamt durch ein einziges Wort beschrieben:

$ZS2\lambda S2\lambda ZS00S2\lambda$

# Universelle determinierte Turing-Maschinen

---

## Gödelisierung

Ein Verfahren, jeder Turing-Maschine eine Zahl oder ein Wort (**Gödelzahl** bzw. **Gödelwort**) so zuzuordnen, dass man aus der Zahl bzw. dem Wort die Turing-Maschine effektiv rekonstruieren kann.

# Kurt Gödel

---

## Kurt Gödel 1906-1978

- Bedeutendster Logiker des 20. Jahrhunderts
- Vollständigkeitssatz (1929)  
Promotion in Wien
- Unvollständigkeitssatz (1931)  
Idee der Gödelisierung
- Beweis der Unabhängigkeit der  
Kontinuumshypothese
- Dozent in Princeton,  
befreundet mit Albert Einstein
- Tragischer Tod:  
Verfolgungswahn, Depressionen,  
Tod durch Unterernährung.



# Übersicht

---

- Determinierte Turing-Maschinen (DTMs)
- Varianten von Turing-Maschinen
- Indeterminierte Turing-Maschinen (NTMs)
- Universelle determinierte Turing-Maschinen
- **Entscheidbar/Aufzählbar**
- Determinierte Turing-Maschinen entsprechen Typ 0
- Unentscheidbarkeit