Grundlagen der Theoretischen Informatik

Turingmaschinen und rekursiv aufzählbare Sprachen (IV)

3.07.2019

Viorica Sofronie-Stokkermans

e-mail: 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

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

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

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

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

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 $f : \Sigma_0^m \to \Sigma_0^n$ 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, Σ, s definiert wie bei determinierten Turing-Maschinen.

Übergangs**relation**:

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

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 Δ definierte) Nachfolgekonfiguration gibt.
- $\mathcal{M}$ **akzeptiert** ein Wort w, falls sie **von** $s, \#w\underline{\#}$ **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)

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 Nachfolgekonfiguration 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 azeptieren:

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 Σ_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:

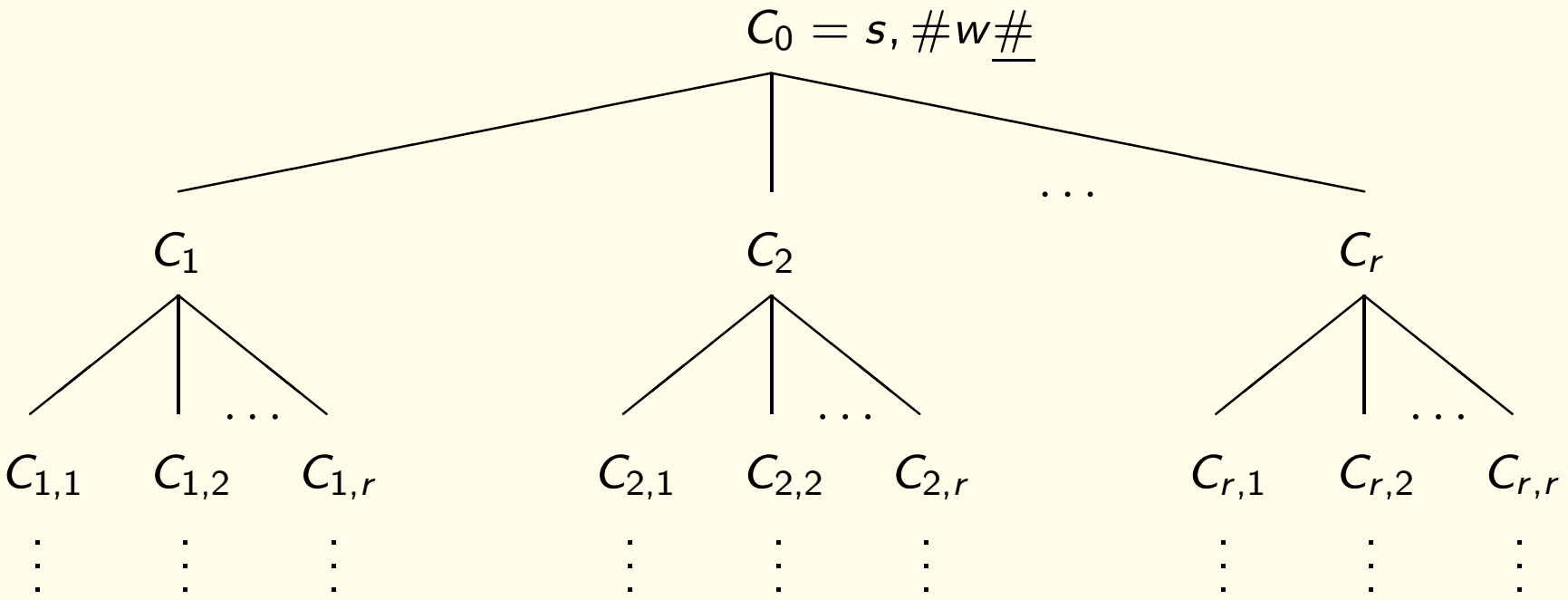
- $\mathcal{M}'$ durchläuft systematisch **alle** Rechnungen von $\mathcal{M}$,
  und sucht dabei nach einer Haltekonfiguration.
- $\mathcal{M}'$ hält genau dann, wenn sie eine Haltekonfiguration von $\mathcal{M}$ findet.

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**.

- Untersuche alle möglichen Rechnungen bis zum ersten Schritt.
- Untersuche alle möglichen Rechnungen bis zum zweiten Schritt.
- Untersuche 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

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

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 $s, \#w\underline{\#}$ 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 Nachfolgekonfigurationen 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 Nachfolgekonfigurationen gewählt worden, C_{d_1}.
- Von C_{d_1}, einem Knoten der Tiefe 1, aus wurde die d_2-te mögliche Nachfolgekonfiguration gewählt,
- usw.

Indeterminierte Turing-Maschinen (NTMs)

Beweis (Fortsetzung)

Ausführung des Iterative Deepening:

- Beginne mit 0 auf zweitem Band.
- Jeweils nächste zu betrachtende Rechnung durch 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 Rechnungsbaum 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 ein vorgegebenes „Programm“ (Regelmenge)
- „Normale“ Computer können beliebige Programme ausführen.

Tatsächlich geht das mit Turing-Maschinen auch!

Universelle determinierte Turing-Maschinen

Turing-Maschine, die andere TMs simuliert

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

Universelle determinierte Turing-Maschinen

Standardisierung von Alphabet, Zustandsmenge, Startzustand

- Unendliches Alphabet $\Sigma_\infty = \{a_0, a_1, \ldots\}$,
  so dass das Alphabet jeder DTM eine Teilmenge von Σ_∞ ist.
- Namen der Zustände einer DTM sind egal.
  Sie seien also $q_1, \ldots, q_n$
  (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 δ-Übergänge beschreiben.**

Universelle determinierte Turing-Maschinen

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

$$q_1, \# \mapsto q_2, L \qquad q_2, \# \mapsto h, \#$$
$$q_1, | \mapsto q_2, L \qquad q_2, | \mapsto q_2, L$$

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"
- λ für "links", ρ für "rechts"
- die Zahl n für den n-ten Zustand und für das n-te Zeichen von Σ_∞.

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.