

Grundlagen der Theoretischen Informatik

Turingmaschinen und rekursiv aufzählbare Sprachen

14.06.2018

Viorica Sofronie-Stokkermans

e-mail: sofronie@uni-koblenz.de

Bis jetzt

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

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

5. Turingmaschinen und rekursiv aufzählbare Sprachen

Inhalt von Teil 5

- Was ist eine **berechenbare** Funktion?
- Determinierte Turing-Maschinen (DTMs)?
- Modifikationen von DTMs:
(mehrere) Halbbänder, zweiseitig unbeschränkte Bänder
- Indeterminierte Turing-Maschinen (NTMs)
- **Gödelisieren**: Programme als Wörter in Σ^* .
- **Aufzählbar** vs. **entscheidbar**
- **Unentscheidbarkeit**, Reduktionen von Problemen aufeinander.

Determinierte Turing-Maschinen (DTMs)

Grundlegende Fragen

- **Frage: Berechenbarkeit?**

Betrachtet werden Abbildungen über den natürlichen Zahlen \mathbb{N} :

Welche davon sollen berechenbar genannt werden?

Determinierte Turing-Maschinen (DTMs)

Grundlegende Fragen

- **Frage: Berechenbarkeit?**

Betrachtet werden Abbildungen über den natürlichen Zahlen \mathbb{N} :

Welche davon sollen berechenbar genannt werden?

- **Frage: Komplexität?**

Um die Komplexität eines Algorithmus' zu messen braucht man ein Maschinenmodell zum Vergleich!

Determinierte Turing-Maschinen (DTMs)

Grundlegende Fragen

- **Frage: Berechenbarkeit?**

Betrachtet werden Abbildungen über den natürlichen Zahlen \mathbb{N} :

Welche davon sollen berechenbar genannt werden?

- **Frage: Komplexität?**

Um die Komplexität eines Algorithmus' zu messen braucht man ein Maschinenmodell zum Vergleich!

Welches Modell wird gewählt?

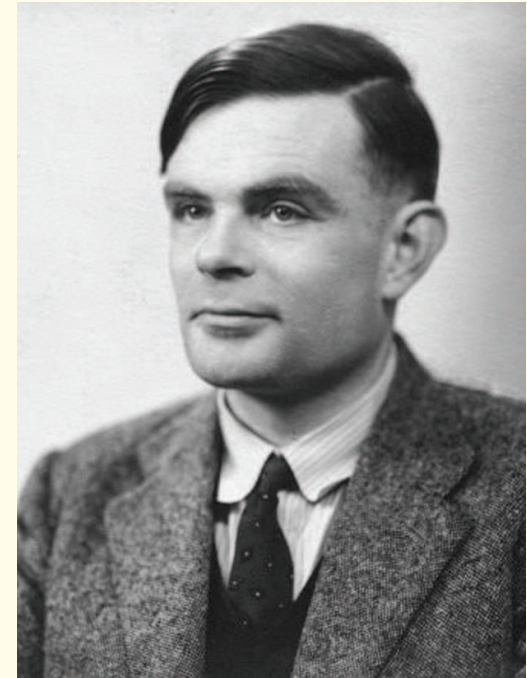
Robustheit:

Das Modell soll nicht von einfachen Modifikationen abhängig sein.

Antwort: Turing-Maschinen

Alan Turing 1912–1954

- Mathematiker und Logiker
- Einer der Begründer der Informatik
- 1936: Definition des Berechenbarkeitsmodells “Turing-Maschine”
- 1938: Promotion bei Church in Princeton
- Während des zweiten Weltkriegs: Kriegsentscheidender Beitrag zur Entschlüsselung deutscher Funkprüche
- Dozent an der Universität Manchester
- Beiträge zur KI (“Turing-Test”)
- Tragischer Tod: Strafverfolgung wegen Homosexualität; (vermutlich) Selbstmord
- Nach ihm benannt: Turing-Award



Turing machines

Alan Turing described a Turing machine (which he called “Logical Computing Machine”), as consisting of:

“ ... an unlimited memory capacity obtained in the form of an infinite tape marked out into squares, on each of which a symbol could be printed.

At any moment there is one symbol in the machine; it is called the scanned symbol.

The machine can alter the scanned symbol and its behavior is in part determined by that symbol, but the symbols on the tape elsewhere do not affect the behaviour of the machine.

However, the tape can be moved back and forth through the machine, this being one of the elementary operations of the machine. Any symbol on the tape may therefore eventually have an innings.”

Idee

Immer mächtigere Automaten

- Erinnerung: Pushdown-Automaten
 - akzeptieren kontextfreie Sprachen
 - Erster Speicher: der Zustand (endlich)
 - Zweiter Speicher: der **Keller**
(unbeschränkte Größe, beschränkte Zugriffsart)
 - Das Eingabewort wird nur einmal gelesen, von links nach rechts.

Idee

Immer mächtigere Automaten

- **Ausblick: Turing-Maschinen**
 - akzeptieren Sprachen **vom Typ 0**.
 - Erster Speicher: der Zustand (endlich)
 - Zweiter Speicher: Band
(unbeschränkte Größe, **Zugriff an beliebiger Stelle**)
 - Turing-Maschine hat einen Schreib-/Lesekopf, den sie über diesem Band in einem Rechenschritt um ein Feld nach rechts oder links bewegen kann.
 - Das Eingabewort steht (am Anfang) auf dem Band.
Die Maschine kann es **beliebig oft lesen**.

Turing-Maschine

Definition (Turing Machine (DTM))

Eine **determinierte Turing-Maschine (DTM)** \mathcal{M} ist ein Tupel

$$\mathcal{M} = (K, \Sigma, \delta, s)$$

Dabei ist

- K eine endliche Menge von Zuständen mit $h \notin K$,
(h ist der **Haltezustand**)
- Σ ein Alphabet mit $L, R \notin \Sigma$, $\# \in \Sigma$,
- $\delta : K \times \Sigma \rightarrow (K \cup \{h\}) \times (\Sigma \cup \{L, R\})$ eine Übergangsfunktion
- $s \in K$ ein Startzustand.

Anzahl der Zustände: $|K| - 1$
(Startzustand wird nicht mitgezählt).

Turing-Maschine

Arbeitsschritt einer Turing-Maschine

Übergang

$$\delta(q, a) = (q', x)$$

bedeutet:

In Abhängigkeit

- vom aktuellen Zustand $q \in K$
- von dem Zeichen $a \in \Sigma$, das unter dem Schreib-/Lesekopf steht

geschieht folgendes:

- entweder ein **Schritt nach links**, falls $x = L$ ist
- oder ein **Schritt nach rechts**, falls $x = R$ ist
- oder **das Zeichen a** , das momentan unter dem Schreib-/Lesekopf steht, wird **durch $b \in \Sigma$ überschreiben**, falls $x = b \in \Sigma$
- der **Zustand** wird zu $q' \in K \cup \{h\}$ **geändert**,

Turing-Maschine

Leerzeichen

Das spezielle Zeichen # (*blank*) ist das Leerzeichen.

Es ist nie Teil des Eingabeworts; man kann es u.a. dazu benutzen, Wörter voneinander abzugrenzen.

Begrenzung des Bandes

Das Band einer DTM ist **einseitig unbeschränkt**:

- Nach rechts ist es unendlich lang.
- Nach links hat es ein Ende.
- Wenn eine DTM versucht, das Ende zu überschreiten, bleibt sie „hängen“.

In diesem Fall **hält sie nicht**.