ETH zürich



Theoretische Informatik

Olivér Kovács

22. Okt 2026

Midterm

Altklausuren: https://exams.vis.ethz.ch/

Stoff hat sich geändert:

- Kolmogorovkomplexität, Satz 3.1 nicht relevant.
- Relevant bis einschliesslich
  - Vorlesung am 03. Novemeber,
  - Blatt 6,
  - Quiz 7.
- Manches davon war früher nicht auf den Midterms!

Nebenbemerkung: Minimalität von DEAs

- Zwei Zustände $p, q \in Q$ in einem DEA können nicht zusammengefasst werden gdw ein Wort w existiert, so dass gilt

$$\hat{\delta}(p, w) \in F \iff \hat{\delta}(q, w) \notin F.$$

- Ein DEA ist minimal gdw für alle Zustände paarweise gilt, dass sie nicht zusammengefasst werden können, und **jeder Zustand erreichbar** ist.

- Bitte einfach Beweis schreiben wie in der Vorlesung hergezeigt!

Bemerkungen zur Serie

- In unserem Formalismus sind in NEAs **keine** λ-Transitionen erlaubt!
- Startzustand und akzeptierenden Zustände nicht vergessen.
  - Insbesondere λ beachten!
- Bei Minimalitätsbeweis:
  - Bitte Tabelle wie in der Vorlesung.
  - Lemma 3.3 erwähnen.
  - Landen die Wörter tatsächlich in verschiedenen Zustände?
  - Man darf λ als Suffix wählen.

Quiz

Gibt es Fragen zum Quiz?

Turingmaschine

Definition 4.1. Eine *Turingmaschine (TM)* ist ein 7-Tupel $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$. Dabei ist

- Q die *Zustandsmenge* von M,
- Σ das *Eingabealphabet*,
- Γ das *Arbeitsalphabet*,
- $\delta : (Q - \{q_{\text{accept}}, q_{\text{reject}}\}) \times \Gamma \to Q \times \Gamma \times \{L, R, N\}$ die *Übergangsfunktion*,
- $q_0 \in Q$ der *Anfangszustand*,
- $q_{\text{accept}} \in Q$ der *akzeptierende Zustand*,
- $q_{\text{reject}} \in Q$ der *verwerfende Zustand*.

Rekursiv aufzählbar

Definition 4.1. [...] Eine Sprache $L \subseteq \Sigma^*$ heist *rekursiv aufzählbar* falls eine TM M existiert, so dass $L = L(M)$.

$$\mathcal{L}_{\mathsf{RE}} = \{L(M) \mid M \text{ ist eine TM}\}$$

ist die *Klasse aller rekursiv aufzählbaren Sprachen*.

Mehrband-Turingmaschine (MTM)

Abbildung 4.7. Eine k-*Band-Turingmaschine* besteht aus

- einer endlichen Kontrolle,
- einem endlichen Eingabeband mit einem Lesekopf,
- k Arbeitsbänder, jeweils mit einem Lese-/Schreibkopf.

Insbesondere ist eine 1-Band-TM nicht das gleiche wie eine TM!

Äquivalenz

Lemma 4.1: Für jede TM existiert eine äquivalente 1-Band-TM.

Lemma 4.2: Für jede MTM existiert eine äquivalente TM.

Beweis?

Nebenbemerkung

`mov` is Turing-complete

Stephen Dolan
Computer Laboratory, University of Cambridge
stephen.dolan@cl.cam.ac.uk

Abstract

It is well-known that the x86 instruction set is baroque, overcomplicated, and redundantly redundant. We show just how much fluff it has by demonstrating that it remains Turing-complete when reduced to just one instruction.

The instruction we choose is `mov`, which can do both loads and stores. We use no unusual addressing modes, self-modifying code, or runtime code generation. Using just this instruction (and a single unconditional branch at the end of the program to make nontermination possible), we demonstrate how an arbitrary Turing machine can be simulated.

registers for now, but later we show how their number can be reduced without losing expressiveness.

We have the following instructions (if you like RISC) or addressing modes (if you like CISC). We use Intel x86 syntax, where the `mov` instructions have the destination first and the source second, and square brackets indicate memory operands.

Instruction	x86 **syntax**
Load Immediate	`mov` $R_{\texttt{dest}}$`, c`
Load Indexed	`mov` $R_{\texttt{dest}}$`, [`$R_{\texttt{src}}$ `+` $R_{\texttt{offset}}$`]`
Store Indexed	`mov [`$R_{\texttt{dest}}$ `+` $R_{\texttt{offset}}$`],` $R_{\texttt{src}}$

On x86, these are all addressing modes of the same `mov` in-

Church'sche These

Die Turingmaschinen sind die Formalisierung des Begriffes „Algorithmus“, d.h., die Klasse der rekursiven Sprachen (der entscheidbaren Entscheidungs- probleme) stimmt mit der Klasse der algorithmisch (automatisch) erkennbaren Sprachen überein.

Nichtdeterminismus

Eine *nichtdeterministische Turingmaschine (NTM)* ist ein 7-Tupel $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$. Dabei

- sind $Q, \Sigma, \Gamma, q_0, q_{\text{accept}}, q_{\text{reject}}$ gleich wie bei einer TM,
- ist $\delta\colon (Q - \{q_{\text{accept}}, q_{\text{reject}}\}) \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{\text{L}, \text{R}, \text{N}\})$ die *Übergangsfunktion*.

Äquivalenz

Satz 4.2. Für jede NTM existiert eine äquivalente TM.

Beweis?

Kodierung

Wir können Turingmaschinen als Wörter über Σ_{bool} darstellen. (Wie?)

Definition 4.5. $\mathsf{Kod}(M)$ ist die *Kodierung* der TM M. Wir definieren

$$\mathsf{KodTM} = \{\mathsf{Kod}(M) \mid M \text{ ist eine TM}\}.$$

Abzählbarkeit

Definition 5.1. Seine A, B zwei Mengen. Wir sagen, dass

1. $|A| \leq |B|$ falls eine injektive Funktion $f : A \to B$ existiert,
2. $|A| = |B|$ falls $|A| \leq |B|$ und $|B| \leq |A|$ gilt,
3. $|A| < |B|$ falls $|A| \leq |B|$ und nicht $|B| \leq |A|$ gilt.

Definition 5.2. Eine Menge A heisst *abzählbar* gdw A endlich ist oder $|A| = |\mathbb{N}|$.

Abzählbarkeit (2)

Lemma 5.1. Sei Σ ein beliebiges Alphabet. Dann ist Σ^* abzählbar.

Satz 5.1. Die Menge KodTM der Turingmaschinenkodierung ist abzählbar.

Lemma 5.2. Die Menge $(\mathbb{N} - \{0\}) \times (\mathbb{N} - \{0\})$ ist abzählbar.

Satz 5.2. $\mathbb{Q}^+$ ist abzählbar.

Satz 5.3. $[0, 1]$ ist nicht abzählbar.

Satz 5.4. $\mathcal{P}((\Sigma_{\text{bool}})^*)$ ist nicht abzählbar.

Korollar 5.1. $|\text{KodTM}| < |\mathcal{P}((\Sigma_{\text{bool}})^*)|$ und somit existieren unendlich viele nicht rekursiv aufzählbare Sprachen.

Diagonalsprache

Definition. Seien $w_1, w_2, w_3, \ldots$ die Wörter über Σ_{bool} in kanonischer Ordnung und $M_1, M_2, M_3, \ldots$ die Folge aller Turingmaschinen. Wir definieren die *Diagonalsprache* L_{diag} als

$$L_{\text{diag}} = \{w \in (\Sigma_{\text{bool}})^* \mid w = w_i \text{ und } M_i \text{ akzeptiert } w_i \text{ nicht}\}.$$

Satz 5.5. $L_{\text{diag}} \notin \mathcal{L}_{\text{RE}}$.

Aufgaben

1. **Aufgabe 1.** Worked Example 1.
2. **Aufgabe 2.** Beschreibe informell eine NTM, welche die Sprache

   $$L_{\text{composite}} = \{w = 0^n \mid n \in \mathbb{N} \text{ ist nicht prim}\}$$

   akzeptiert.
3. **Aufgabe 3.** Wir betrachten die Sprache

   $L_{\lambda+1} = \{\text{Kod}(M) \mid \text{TM } M \text{ akzeptiert } \lambda \text{ und mindestens ein weiteres Wort}\}$.

   Zeige, dass $L_{\lambda+1} \in \mathcal{L}_{\text{RE}}$.