Complexity Theory

Henrik Tscherny

19. November 2021

Inhaltsverzeichnis

1 Definitionen

1.1 Entscheidbarkeit und Erkennbarkeit

- Sei M eine TM mit dem Eingabealphabet Σ
- Sei $L(M) := \{w \in \Sigma^* | M akzeptiert w\}$

- Ein Sprache L ist **erkennbar** gdw. es eine TM gibt welche diese Sprache erkennt ($L = L(M)$)
- Außerdem ist eine Sprache **erkennbar** gdw. es einen Enumerator E mit $G(E) = L$
- Ist eine Sprache **erkennbar**, dann existiert ein Enumerator E für L welcher jedes Wort in L nur genau einmal ausgibt
- Eine Sprache L ist **entscheidbar** gdw. sie erkennbar ist und die TM auf jedem Input hält
- Eine Sprache L ist **erkennbar** gdw. es einen Enumerator E gibt welcher die Wörter in aufsteigender Länge aufzählt
- Eine Sprache L ist **co-semi-entscheidbar** gdw. $\bar{L}$ semi-entscheidbar ist
- Ist L semi-entscheidbar und co-semi-entscheidbar, so ist L entscheidbar Beweis:
  - Sei TM_L eine TM welche L erkennt
  - Sei $TM_{\bar{L}}$ eine TM welche $\bar{L}$ erkennt
  - Simuliere TM_L und $TM_{\bar{L}}$
  - eine von beiden muss halten, da L semi-entscheidbar und co-semi-entscheidbar ist
  - somit kann $w \in L$ entschieden werden
- Es gibt Probleme Welche weder semi- npch co-semi-entscheidbar sind z.B. TM Äquivalenz (Beweis siehe Many-One-Reduktionen)

Note: Erkennbar = semi-entscheidbar

1.2 Enumerator (Aufzähler)

Eine Multi-Band TM M mit:

- M hat ein **write-only output Band**, auf dem der Head nur nach Links laufen kann
- M hat ein # Symbol welches die Wörter auf dem Output Band trennt
- Die von **M erzeugte Sprache** sei die Sprache aller Wörter welche irgendwann auf dem Output Band zwischen Zwei # auftauchen
- M **startet** auf einem **leeren Band**

2 Arten von TM's

2.1 einseitig begrenzte einband TM

Definition

- $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$
- Q: Zustandsmenge
- Σ: Eingabealphabet (ohne ␣)
- : Γ: Bandalphabet mit $\Sigma \cup \{␣\} \subseteq \Gamma$
- δ: Übergangsfunktion mit $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$
- q_0: Startzustand
- q_{accept}: akzeptierender Endzustand mit $q_{accept} \in Q$
- q_{reject}: ablehnender Endzustand mit $q_{reject} \in Q, q_{accept} \neq q_{reject}$

2.2 k-Band TM

Definition

- $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$
- $\delta : Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{L, R, N\}^k$

Äquivalenzbeweis

- Sei M eine k-Band TM und S eine TM welche M simuliert
- Schreibe den Inhalt der k Bänder von M hintereinander jeweils getrennt durch ein Trennzeichen (#)
- Die Head-Positionen von M werden durch spezielle Zeichen in S dargestellt ($\dot{x}$)

2.3 Oracle TM

Definition

- besitzt ein spezielles **Oracle Band**
- hat spezielle Zustände $q_?, q_{yes}, q_{no}$
- Wenn die OTM $q_?$ erreicht, springt diese zu q_{yes} oder q_{no}, je nachdem ob der Inhalt des Oracle Bandes in O ist
- O kann ein beliebig schweres Problem sein, sogar unentscheidbar
- Das Oracle zu benutzen benötigt lediglich einen Schritt
- Ein Oracle zu Komplementieren hat keine Auswirkung

3 Reduktionen

3.1 Turing-Reduktionen

Definition

ein Problem P ist **Turing reduzierbar** auf ein Problem Q (P $\leq_T$ Q), wenn P durch eine OTM M^Q mit Oracle Q entschieden werden kann
Turing-Reduktion kann genutzt werden, um Unentscheidbarkeit zu beweisen
Ist P unentscheidbar und P $\leq_T$ Q dann ist Q unentscheidbar
Beweis mittels Kontrapositiv:

- Nehme an P $\leq_T$ Q mit Q ist entscheidbar
- Man kann die benötigte OTM als normale TM konstruieren, so dass gilt P $\leq_T$ Q
- daraus folgt das P entscheidbar ist

Note: Ist Q semi-entscheidbar und P $\leq_T$ Q, dann kann oder kann P nicht semi-entscheidbar sein

Beispiel-Reduktion

Reduzieren des ϵ-Halteproblems auf das Halteproblem
(Hält eine TM M auf einem leeren Inputband ?)

- Definiere eine OTM wie folgt:
  - Input: Eine TM M und ein Wort w
  - Konstruiere eine TM M_w wie folgt:
    1. Lösche alles vom Inputtape und schreibe w darauf
    2. Bearbeite den Input wie M
  - Löse das ϵ-Halteproblem für M_w mit einem Oracle
  - Output: Ergebnis des ϵ-Halteproblem
- $\Rightarrow$ **Unentscheidbar**

3.2 Many-one-Reduktion (mapping Reduction)

Definition

Eine Sprach P ist **many-one-reduzierbar** auf eine Sprache Q (P $\leq_m$ Q), wenn es eine **totale berechenbare Funktion** f gibt, so dass $f : \Sigma^* \to \Sigma^*$, so dass $\forall w \in \Sigma^* : w \in P \Leftrightarrow f(w) \in Q$

- Many-one-Reduktion **erhält (co-)semi-entscheidbarkeit** (anders als die Turing-Reduktion)
- P $\leq_m$ Q $\Rightarrow$ P $\leq_T$ Q
  Beweis:
  - Man erhält eine OTM mit Oracle Q, welches P erkennt wie folgt:
    * Input w $\to$ berechne f(w)
    * Frage das Oracle und gebe das Ergebnis zurück (ja = akzeptieren, nein = verwerfen)
- gilt P $\leq_m$ Q und Q ist entscheidbar, dann ist P entscheidbar
- gilt P $\leq_m$ Q und Q ist semi-entscheidbar, dann ist P semi-entscheidbar
  Beweis:
  - gegeben eine TM welche Q erkennt, man erhält eine TM welche P erkennt wie folgt:
    * Auf dem Input w, berechne f(w)
    * Simuliere die TM für Q und gebe das Ergebnis zurück

Beispiel-Reduktion Zwei TM's M und N sind äquivalent wenn sie die gleiche Sprache erkennen (L(M) = L(N))
TM Äquivalenz ist unentscheidbar:
Beweis:

- definiere f, so dass $w \in \epsilon - \text{Halting}$ gdw. $f(w) \in$ Äquivalenz
- Sei M_a eine TM welche alle Inputs akzeptiert (hält auf jedem beliebigen w)
- Definiere für eine TM M eine TM M^* wie folgt:
  - Simuliere M auf ϵ
  - wenn M hält akzeptiere
- M^* akzeptiert also alle Inputs sofern, M auf ϵ hält
- M^* ist äquivalent zu M_a gdw. M auf ϵ hält
- $f(w) = \begin{cases} \langle M^*, M_a \rangle, & w = \langle M \rangle \quad \text{(valide Kodierung)} \\ \epsilon, & \text{ungültige Kodierung} \end{cases}$
  $\rightarrow$ M^* auf M_a akzeptiert immer, da M_a für für jeden beliebigen Input hält, somit auch auf ϵ

Äquivalenz von TMs ist weder semi- noch co-semi-entscheidbar
Beweis:

- Wie gezeigt gilt ϵ-Halteproblem $\leq_m$ Äquivalenz
- Da das ϵ-Halteproblem nicht co-semi-entscheidbar ist, ist Äquivalenz es auch nicht
- Jedoch kann man zeigen, dass nicht-ϵ-Halteproblem $\leq_m$ Äquivalenz
  - Sei $M_\emptyset$ eine TM welche alle Inputs ablehnt (Gegenteil zu M_a)
  - Äquivalenz zu M_a korrespondiert zu ϵ-halten
  - Äquivalenz zu $M_\emptyset$ korrespondiert zu ϵ-nicht-halten
  - $f(w) = \begin{cases} \langle M^*, M_\emptyset \rangle, & w = \langle M \rangle \quad \text{(valide Kodierung)} \\ \langle M^*, M_\emptyset \rangle, & \text{ungültige Kodierung} \end{cases}$

4 Satz von Rice

Trivialität

- Sei P eine Menge von Sprachen
- Eine Sprache L hat die Eigenschaft P wenn $L \in P$
- P ist **nicht-trivial**, wenn es erkennbare Sprachen gibt welche P haben und solche die P nicht haben
  → es muss Entscheidungsmöglichkeiten geben

Satz

Ist P eine **nicht-triviale** Eigenschaft einer erkennbaren Sprache, dann ist folgendes Problem unentscheidbar:
$P - ness = \{\langle M \rangle | L(M) \in P\}$
d.h. Die Frage ob eine TM eine nicht-triviale Eigenschaft, hat ist unentscheidbar

Beweis

Reduzieren auf ϵ-Halteproblem

- Sei $\emptyset \notin P$
- Sei M_L eine TM welche die Sprache L erkennt und L die Eigenschaft P hat (L ∈ P)
- Konstruiere für eine beliebige TM M eine TM M^* wie folgt für einen Input $w \in \Sigma^*$:
  1. Simuliere M mit leerem Input (ϵ)
  2. Hält M, dann simuliere M_L auf w
- Dadurch gilt $L(M^*) = L \in P$ wenn M auf ϵ hält, da (2) lediglich M_L auf w simuliert
- Sonnst gilt $L(M^*) = \emptyset \notin P$, da M^* in (1) festhängt
- **Die Überprüfung ob $\langle M^* \rangle \in P$ würde das ϵ-Halteproblem entscheiden ⇒ Unentscheidbar**

Anwendungen

Unter anderem kann für folgende Eigenschaften für eine beliebige TM M mit Rice gezeigt werden, das die Sprachen unentscheidbar sind:

- Leerheit
- Endlichkeit
- Entscheidbarkeit
- Regularität
- Kontextfreiheit
- Wortproblem

5 Rekursion

Eine **Quine** ist ein Programm, welches wenn gestartet, ohne zusätzlichen Input seinen eigenen Sourcecode ausgibt und dann hält
Ziel: Konstruieren einer TM SELF welche sich selbst als Encoding zurück gibt

- Es gibt eine berechenbare Funktion $q : \Sigma^* \to \Sigma^*$, so dass
  $\forall w \in \Sigma^* : q(w) = print(\langle M \rangle)$
  Der Teil mit 'für jedes Wort' heißt einfach: ignoriere den Input (Input-unabhängig)
- Man kann eine TM P_w konstruieren, welche für jedes Wort den Tapeinhalt mit w ersetzt
- q kann man jetzt berechnen, indem man eine TM nimmt, welche mit w als input P_w erzeugt und anschließend $\langle P_w \rangle$ ausgibt

Eine Quine kann in zwei Teile unterteilt werden:

- A: berechne den Sourcecode $\langle B \rangle$ eines Programms B
- B: nutze $\langle B \rangle$ um folgendes zu printen:
  Sourcecode $\langle A \rangle$ welcher $\langle B \rangle$ berechnet und den Sourcecode $\langle B \rangle$ selbst

A kann mit der zuvor konstruierten TM $P_{\langle B \rangle}$ implementiert werden
B funktioniert wie folgt auf einem Input $\langle M \rangle$:

- berechne $q(\langle M \rangle)$

- konkatoniere die TM's gegeben durch $q(\langle M \rangle)$ und $\langle M \rangle$
- gebe die Kodierung dieser Konkatenation zurück

die TM SELF istnun eine TM konstruiert durch B auf dem Input $\langle B \rangle$

6 Zeitkomplexität

Definition

Sei M eine TM und f eine Funktion mit $f : \mathbb{N} \to \mathbb{R}^+$
M ist **f-time beschränkt**, wenn M auf jedem input $w \in \Sigma^*$ nach maximal $f(|w|)$ Schritten hält

6.1 Landau-Symbole

Big-O

- Klassifiziert eine Funktion mit einer asymptotischen oberen Schranke
- $f(n) = O(g(n)) \Leftrightarrow \exists c > 0 \exists n_0 \in \mathbb{N} \forall n > n_0 : f(n) \leq c \cdot g(n)$
- es gibt also ein Punkt (n_0) ab dem die Funktion f dauerhaft kleiner gleich der Funktion g ist, selbst wenn g mit einer beliebigen Konstante (c) multipliziert wird

Note: für **small-o** sagt man auch f wird asymtotisch von g dominiert
Symboltabelle:

Notation	$C = \lim_{n\to\infty} \frac{f(n)}{g(n)}$	Bedeutung
$f \in O(g)$	$c < 0$	$f \leq g$
$f \in \Omega(g)$	$c > 0$	$f \geq g$
$f \in \Theta(g)$	$0 < c < \infty$	$f = g$
$f \in o(g)$	$c = 0$	$f < g$
$f \in \omega(g)$	$c = \infty$	$f > g$