Grundzüge der Theoretischen Informatik

15. Dezember 2021

Markus Bläser

Universität des Saarlandes

Kapitel 18: Die Church-Turing-These

While-Berechenbarkeit und Turing-Berechenbarkeit

While-berechenbar = Turing-berechenbar

Identifizieren $\mathbb{N}$ mit $\{0,1\}^*$:

- $\mathrm{cod} : \{0,1\}^* \to \mathbb{N}$
- $\mathrm{cod}(x) = \mathrm{bin}^{-1}(1x) - 1$

Identifizieren $\mathbb{N} \to \mathbb{N}$ mit $\{0,1\}^* \to \{0,1\}^*$:

- Zu $f : \mathbb{N} \to \mathbb{N}$ definiere $\hat{f} : \{0,1\}^* \to \{0,1\}^*$ durch
$$\hat{f}(x) = \mathrm{cod}^{-1}(f(\mathrm{cod}(x))) \quad \text{für alle } x \in \{0,1\}^*.$$
- Zu $g : \{0,1\}^* \to \{0,1\}^*$, definiere $\hat{g} : \mathbb{N} \to \mathbb{N}$ durch
$$\hat{g}(n) = \mathrm{cod}(g(\mathrm{cod}^{-1}(n))) \quad \text{für alle } n \in \mathbb{N}.$$

$\hat{\hat{f}} = f$ und $\hat{\hat{g}} = g$

GOTO-Programme

Ein GOTO-Programm ist eine Folge

$$(1, s_1), (2, s_2), \ldots, (m, s_m)$$

wobei jedes s_μ eine Anweisung der Form

1. $x_i = x_j + x_k$ oder
2. $x_i = x_j - x_k$ oder
3. $x_i := c$ oder
4. **if** $x_i \neq 0$ **then goto** λ

ist.
Das Programm terminiert, wenn eine nicht vorhandene Zeile erreicht wird.

Von WHILE nach GOTO

Lemma (18.2)

Für jedes WHILE-Programm P *gibt es ein GOTO-Programm* Q *mit* $\varphi_P = \varphi_Q$.

1: **while** $x_i \neq 0$ **do**
2: $\quad P$
3: **od**

1: **if** $x_i \neq 0$ **then goto** 3
2: **goto** 5
3: P
4: **goto** 1
5: ...

Von GOTO zu Turingmaschinen

Lemma (18.3)

Sei $f : \mathbb{N} \to \mathbb{N}$. Falls f GOTO-berechenbar ist, dann ist $\hat{f} : \{0,1\}^* \to \{0,1\}^*$ *Turing-berechenbar.*

- einfache Anweisungen: $x_i{+}{+}$, $x_i{-}{-}$ und $x_i := 0$.
- Jede Variable wird durch ein Band dargestellt. $x_0, \ldots, x_\ell$
- Der Inhalt steht in binär von links nach rechts.
- Schrittweise Simulation
- Invariante: Zu Beginn der Simulation eines Schrittes stehen die Köpfe auf der Einerstelle.

Beispiel

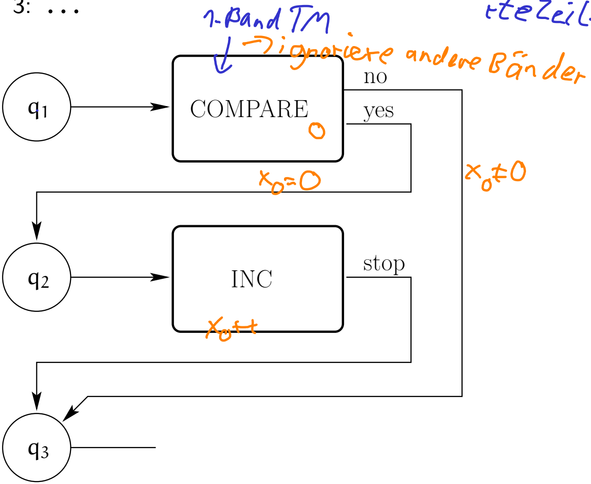
1: **if** $x_0 \neq 0$ **then goto** 3

2: $x_0{+}{+}$

3: ...

Pro Zeile im Goto-Progr. ein Zustand

i-te Zeile $\rightarrow q_i$



Von Turingmaschinen zu WHILE

Lemma (18.4)

Sei $g : \{0,1\}^* \to \{0,1\}^*$. *Falls* g *Turing-berechenbar ist, dann ist* $\hat{g}$ *WHILE-berechenbar.*

- Bänder werden als Arrays gespeichert
- Die Übergangsfunktion wird fest verdrahtet.

Die Church-Turing-These

Satz (18.5)

Sei $f : \mathbb{N} \to \mathbb{N}$. *Dann sind äquivalent:*

1. f *ist WHILE-berechenbar*
2. f *ist GOTO-berechenbar*
3. f *ist Turing-berechenbar*

Church-Turing-These

Jede Definition von "intuitiv berechenbar" ist äquivalent zu WHILE-berechenbar.

→ keine Aussage

→ nicht beweisbar

Kapitel 19: Komplexitätsklassen

Deterministische Zeitkomplexität

- Sei M eine DTM, M halte auf x.
- Es gibt eine (eindeutige) haltende Konfiguration C_t mit
  $SC(x) \vdash_M^* C_t$

  Start config. bei Eingabe x
- Sei $SC(x) \vdash_M C_1 \vdash_M \cdots \vdash_M C_t$ die Berechnung von M auf x.
- t ist die Anzahl der Schritte von M auf x, $\boxed{t =: \mathrm{Time}_M(x)}$
- Es sei $\mathrm{Time}_M(n) := \max\{\mathrm{Time}_M(x) \mid |x| = n\}$, $n \in \mathbb{N}$.

Definition

Sei $t : \mathbb{N} \to \mathbb{N}$.
M ist t-zeitbeschränkt, falls $\mathrm{Time}_M(n) \le t(n)$ für alle n.

z.B. n^2-zeitbeschränkt

"worst-case-Komplexität"

Deterministische Platzkomplexität

Kopfposition Bandinhalt

Zustand

- Zu einer Konfiguration $C = (q, \underbrace{(p_1, x_1)}_{\text{Band 1}}, \ldots (p_k, x_k))$ sei $\mathrm{Space}(C) = \max_{1 \leq \kappa \leq k} |x_\kappa|$.

„Länge des befüllten Bandes"

$L = \{x \in \{0,1\}^* \mid$ Anzahl der 1en in x ist gleich der Anzahl der 0en$\}$ kann mit Platz $O(\log n)$ erkannt werden.
Problem: Eingabe ist n lang.

- Extra-Eingabeband, read-only, → Eingabe zählt nicht zum Platz.
- zählt bei der Berechnung von $\mathrm{Space}(C)$ nicht mit.

Deterministische Platzkomplexität (2)

- Sei M eine DTM, M halte auf x.
- Sei $\mathrm{SC}(x) \vdash_M C_1 \vdash_M \cdots \vdash_M C_t$ die Berechnung von M auf x.
- $\mathrm{Space}_M(x) = \max\{\mathrm{Space}(C_\tau) \mid 1 \leq \tau \leq t\}$.
- Falls M nicht auf x hält, dann wird über das Maximum über unendlich viele Konfigurationen genommen.
- $\mathrm{Space}_M(x) = \infty$, falls das Maximum nicht existiert.
- $\mathrm{Space}_M(n) := \max\{\mathrm{Space}_M(x) \mid |x| = n\}$ für $n \in \mathbb{N}$.

Definition

Sei $s : \mathbb{N} \to \mathbb{N}$.
M ist s-platzbeschränkt, falls $\mathrm{Space}_M(n) \leq s(n)$ für alle n.

$s(n)$

"worst-case-Komplexität"

Deterministische Zeitklassen

Sprache $\subseteq \{0,1\}^*$

- L ist *deterministisch* t *zeit-entscheidbar*, falls es eine DTM M gibt, so dass $\underline{L = L(M)}$ und $\underline{\mathrm{Time}_M(n) \leq t(n)}$ für alle n.
- Zeitbeschränkte DTMs halten immer.

Definition (19.2)

Sei $t : \mathbb{N} \to \mathbb{N}$.

beliebige Anzahl Bänder

$$\mathrm{DTime}(t) = \{L \mid L \text{ ist deterministisch } t\text{-zeit-entscheidbar}\},$$

$$\mathrm{DTime}_k(t) = \{L \mid \text{es gibt eine } t\text{-zeitbeschränkte } k\text{-Band-DTM mit } L = L(M)\}.$$

z.B. Menge aller Polynome

- Zu einer Menge T von Funktionen $\mathbb{N} \to \mathbb{N}$ sei $\mathrm{DTime}(T) = \bigcup_{t \in T} \mathrm{DTime}(t)$.

Deterministische Platzklassen

- L ist *deterministisch s-platz-erkennbar*, falls es eine DTM M gibt, so dass $L = L(M)$ und $\mathrm{Space}_M(n) \le s(n)$ für alle n.
- Eine platz-beschränkte DTM muss nicht auf $x \notin L(M)$ halten.

Definition (19.3)

Sei $s : \mathbb{N} \to \mathbb{N}$.

DSpace(n^2)

$$\mathrm{DSpace}(s) = \{L \mid L \text{ ist deterministisch } s\text{-platz-erkennbar}\},$$
$$\mathrm{DSpace}_k(s) = \{L \mid \text{es gibt eine } s\text{-platz-beschränkte } k\text{-Band-DTM } M \text{ mit } L = L(M)\}.$$

In der Definition von DSpace(s) haben die TMs ein Extra-Eingabeband.

- Zu einer Menge S von Funktionen $\mathbb{N} \to \mathbb{N}$ sei $\mathrm{DSpace}(S) = \bigcup_{s \in S} \mathrm{D}\cancel{\mathrm{Time}}(s)$.
  Space

$COPY = \{w\#w \mid w \in \{0,1\}^*\}$

$\{0,1\} \times \{\text{fertig}, \text{todo}\}$ $\in \Sigma$ $\leftarrow | \rightarrow$

$\rightarrow O(\log n)$ Platz

$k \rightarrow$ Binärzähler auf anderem Band

$O(n)$ Schritte pro Iteration

n Iterationen

$\rightarrow O(n^2)$-zeitbeschränkt

$COPY \in DTime(O(n^2))$

$\in DTime_2(O(n^2))$

$COPY \in DSpace(O(\log n))$

Kopiere w auf anderes Band $O(n)$ Schritte

$\rightarrow$ Direkter Vergleich mit $O(1)$ Schritte pro Iteration

$COPY \notin DTime_1(O(n))$ $\rightarrow COPY \in DTime(O(n))$

Nichtdeterministische Turingmaschinen

Zustand aktuelle Zeichen auf Bändern

Statt

Ein Nachfolger

$$\delta : Q \times \Gamma^k \to Q \times \Gamma^k \times \{L, S, R\}^k,$$

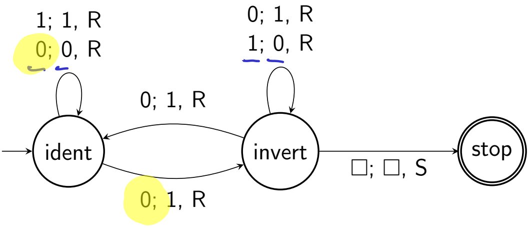
nun

$$\delta : Q \times \Gamma^k \to \mathcal{P}(Q \times \Gamma^k \times \{L, S, R\}^k)$$

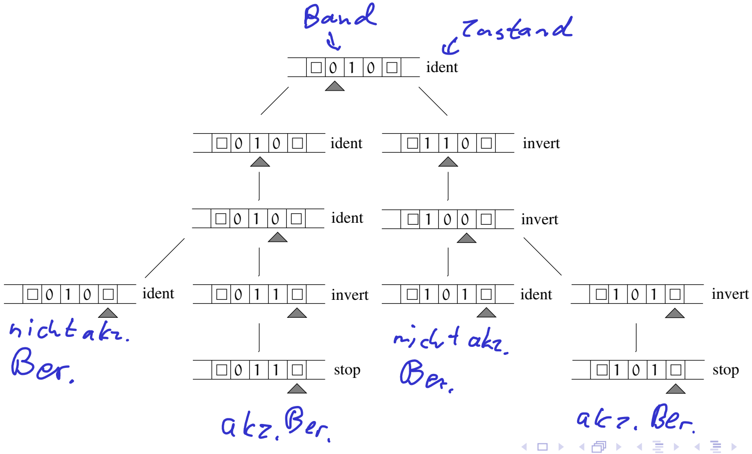
mehrere Nachfolger

- Eine Konfiguration hat nun mehrere Nachfolgekonfigurationen.
- Berechnungsbaum auf Eingabe x:
  - Wurzel ist mit $\mathrm{SC}(\overset{x}{\cancel{w}})$ beschriftet.
  - Ist ein Knoten mit C beschriftet und sind $C_1, \dots, C_s$ die Nachfolge-Konfigurationen von C, so sind die Kinder mit $C_1, \dots, C_s$ beschriftet.

Beispiel



1; 1, R
0; 0, R
0; 1, R
1; 0, R
0; 1, R
0; 1, R
ident
invert
stop
□; □, S



Band
Zustand
□ 0 1 0 □ ident
□ 0 1 0 □ ident
□ 1 1 0 □ invert
□ 0 1 0 □ ident
□ 1 0 0 □ invert
□ 0 1 0 □ ident
□ 0 1 1 □ invert
□ 1 0 1 □ ident
□ 1 0 1 □ invert
nicht akz. Ber.
□ 0 1 1 □ stop
nicht akz. Ber.
□ 1 0 1 □ stop
akz. Ber.
akz. Ber.

Nichtdeterministische Komplexität

- Ein Pfad zu der Wurzel zu einem Blatt heißt *Berechnungspfad*.
- Der Pfad heißt akzeptierend, falls die Konfiguration im Blatt akzeptierend ist.
- Ist die Konfiguration verwerfend, so heißt der Pfad verwerfend.
- Unendliche Pfade sind verwerfend.
- Eine NTM M akzeptiert eine Eingabe x, falls der Berechnungsbaum auf x einen akzeptierenden Pfad hat.
- $L(M) = \{x \in \Sigma^* \mid M \text{ akzeptiert } x\}$.
- Bei DTMs sind Berechnungsbäume Pfade.

$$
\begin{array}{c}
SC(x) \\
| \\
C_1 \\
| \\
C_2 \\
\vdots \\
C_t
\end{array}
$$

Nichtdeterministische Zeitklassen

- $\mathrm{Time}_M(x)$ ist die Länge einer kürzesten Berechnung von M auf x.
- $\mathrm{Time}_M(n) = \max\{\mathrm{Time}_M(x) \mid |x| = n, x \in L(M)\}$
- Sei $t : \mathbb{N} \to \mathbb{N}$. M ist *schwach t-zeit-beschränkt*, falls $\mathrm{Time}_M(n) \leq t(n)$ für alle n.

M muss nicht halten für $x \notin L(M)$

Definition (19.5)

Sei $t : \mathbb{N} \to \mathbb{N}$.

$$\mathrm{NTime}(t) = \{L \mid \text{es gibt eine schwach } t\text{-zeit-beschränkte NTM } M \text{ mit } L = L(M)\},$$

$$\mathrm{NTime}_k(t) = \{L \mid \text{es gibt eine schwach } t\text{-zeit-beschränkte } k\text{-Band-NTM } M \text{ mit } L = L(M)\}.$$

Nichtdeterministische Platzklassen

- $\mathrm{Space}_M(x)$ ist das Minimum über alle akzeptierenden Pfade des maximalen Platzverbrauchs auf diesem Pfads.
- $\mathrm{Space}_M(n) = \max\{\mathrm{Space}_M(x) \mid |x| = n,\ x \in L(M)\}$.
- Sei $s : \mathbb{N} \to \mathbb{N}$. M ist *schwach s-platz-beschränkt*, falls $\mathrm{Space}_M(n) \leq s(n)$ für alle n.

$\mathrm{Space}_M(n) = 0$ falls kein Wort mit Länge n in $L(M)$ ist.

Definition (19.6)

Sei $s : \mathbb{N} \to \mathbb{N}$.

$$\begin{aligned}
\mathrm{NSpace}(s) = \{L \mid & \text{ es gibt eine schwach } s\text{-platz-beschränkte} \\
& \text{NTM } M \text{ mit } L = L(M)\}, \\
\mathrm{NSpace}_k(s) = \{L \mid & \text{ es gibt eine schwach } s\text{-platz-beschränkte} \\
& k\text{-Band-NTM } M \text{ mit } L = L(M)\}.
\end{aligned}$$

t(n) s(n)	DTMs	NTMs
Zeit	DTime(t) (T) $DTime_k(t)$ (T)	NTime(t) (T) $NTime_k(t)$ (T)
Platz	DSpace(s) (S) $DSpace_k(s)$ (S)	NSpace(s) (S) $NSpace_k(s)$ (S)