Vorlesung: Komplexitätstheorie

Wintersemester 2012/2013

Steffen Reith
Steffen.Reith@hs-rm.de

Hochschule RheinMain

12. Oktober 2012

Über den Dozenten

- Prof. Dr. Steffen Reith, geboren 1968, verheiratet, ein Kind
- Seit Sommersemester 2006 an der Hochschule RheinMain
- Vorher tätig als Softwareentwickler für kryptographische und mathematische Algorithmen für tief eingebettete System in KFZs.
- Spezialgebiete: Theoretische Informatik (Komplexität von verallgemeinerten Erfüllbarkeitsproblemen), Komplexitätstheorie, Logik in der Informatik und Kryptographie / diskrete Mathematik

EMail:
Steffen.Reith@hs-rm.de

IM (Skype):
Steffen.Reith

Büro:
Raum 202 (C Gebäude)

Weitere Informationen

Webseite:
`http://www.cs.hs-rm.de/~reith`

Auf der Webseite kann auch ein RSS-Feed abonniert werden, auf dem alle relevanten Ankündigungen mitgeteilt werden.

Grundlegende Begriffe

Definition

Sei Σ ein Alphabet, d.h. eine endliche Menge von Buchstaben, dann ist

- Σ^* die Menge alle Wörter über Σ (einschließlich dem leeren Wort)
- Und $L \subseteq \Sigma^*$ heißt Sprache (über Σ)

Definition

Ein Problem $\mathcal{P}$ ist eine Relation $\mathcal{P} \subseteq \mathcal{I} \times \mathcal{S}$, wobei

- $\mathcal{I}$ die Menge der Probleminstanzen und
- $\mathcal{S}$ ist Menge der Problemlösungen ist.

Funktions- und Entscheidungsprobleme

Definition

Ist das Problem $\mathcal{P}$

- rechtseindeutig, d.h. eine (partielle) Funktion, dann heißt $\mathcal{P}$ Funktionsproblem,
- rechtseindeutig und $\mathcal{S} = \{0, 1\}$, dann heißt $\mathcal{P}$ Entscheidungsproblem

Eine Probleminstanz $x \in \mathcal{I}$ kann auch ein m Tupel sein, d.h. die Definition deckt auch m-stellige Funktionsprobleme ab.

Beispiel

Sei $\mathcal{P}$ ein Entscheidungsproblem, dann ist $L =_{\text{def}} \{x \in \mathcal{I} \mid \mathcal{P}(x) = 1\}$ eine Sprache und $\mathcal{P}$ heißt Wortproblem der Sprache L.

Das Erfüllbarkeitsproblem der Aussagenlogik

Beispiel

Sei $V = \{x_1, x_2, x_3, \dots\}$ eine Menge von aussagenlogischen Variablen und $\mathcal{I}$ die aussagenlogischen Formeln über (Teilmengen von) V.

- Sei $\mathcal{S}$ die Menge aller Funktionen von endlichen Teilmengen von V nach $\{0, 1\}$, $\text{sat} \subseteq \mathcal{I} \times \mathcal{S}$, wobei $(H, f) \in \text{sat}$ genau dann, wenn
  - H ist eine aussagenlogische Formel über $V' \subseteq V$,
  - $f\colon V' \to \{0, 1\}$ und
  - f ist eine erfüllende Belegung von H.
- Eine Ordnung auf $\mathcal{S}$ kann wie folgt definiert werden:
  $f_1 < f_2$ gdw. $\exists x_i (f_1(x_i) < f_2(x_i) \wedge \forall x_j (j < i \rightarrow f_1(x_j) = f_2(x_j))$

$$\text{LEXMINSAT}(H) =_{\text{def}} \begin{cases} \text{kleinste Fkt. } f \text{ so, dass} \\ (H, f) \in \text{sat} & \text{, falls } f \text{ existiert} \\ f_0 \in \mathcal{S} \text{ mit } f_0(x) = 0 \\ \text{für alle } x \in V' & \text{, sonst} \end{cases}$$

Einige Bemerkungen

Die Komplexitätstheorie beschäftigt sich mit der Komplexität von Berechnungen wie z.B.

- Anzahl der Schritte/Laufzeit,
- Speicherbedarf oder
- Anzahl der Befehle des kürzesten Lösungsalgorithmus

auf unterschiedlichen Berechnungsmodellen (Turingmaschinen (TM), RandomAccessMaschine (RAM) Quantencomputern, Parallelrechnern).

Wie formalisiert man diese Vielfalt?

Komplexitätsmaße und Komplexitätsklassen

Definition

Ein Algorithmus A (TM, RAM, C) berechnet eine Funktion $f_A : \mathcal{I} \to \mathcal{S}$, wenn

$$f_A(x) = \begin{cases} \text{Ergebnis von } A \text{ bei Eingabe } x & \text{, falls } A \text{ stoppt} \\ \text{undef,} & \text{, sonst} \end{cases}$$

Definition

Der Algorithmus A hat ϕ-Komplexität (Laufzeit, Speicherplatz), wobei $\phi_A : \mathcal{I} \to \mathbb{N}$, und

$$\phi_A(x) = \begin{cases} \phi\text{-Komplexität von } A \text{ bei Eingabe } x & \text{, falls } A \text{ stoppt} \\ \text{undef,} & \text{, sonst} \end{cases}$$

Komplexitätsmaße und Komplexitätsklassen (II)

Definition

Sei $|x|$ die Länge von x und $\phi_A \colon \mathbb{N} \to \mathbb{N}$ vermöge

$$\phi_A(x) =_{\text{def}} \max_{|x|=n} \phi_A(x),$$

dann heißt ϕ_A worst-case Komplexität (von A).

Die Komplexität wird (fast) immer über die Eingabelänge gemessen!

Definition

Gegeben sei ϕ-Komplexität, τ-Algorithmentyp (z.B. $\mathrm{TM}, \mathrm{RAM}, \mathrm{C}, \ldots$) und eine Schrankenfunktion, dann

$$F_\tau\phi(t) =_{\text{def}} \{f \mid f \text{ total und es ex. Algorithmus } A \text{ vom Typ } \tau \text{ der die Funktion } f \text{ berechnet, wobei } \phi_A \leq_{\text{ae}} t\}$$

Komplexitätsklassen

Definition

Gegeben sei ϕ-Komplexität, τ-Algorithmentyp (z.B. $\mathrm{TM}, \mathrm{RAM}, \mathrm{C}, \ldots$) und eine Schrankenfunktion, dann

$$\tau\phi(t) =_{\text{def}} \{L \mid L \text{ Sprache und es ex. Algorithmus } A \text{ vom Typ } \tau \text{ der das Wortproblem } L \text{ entscheidet, wobei } \phi_A \leq_{\text{ae}} t\}$$

Dabei ist
$t_1 \leq_{\text{ae}} t_2$ gdw. es ex. ein $n_0 \geq 0$ und $t_1(n) \leq t_2(n)$ für alle $n \geq n_0$
(t_1 almost everywhere less or equal than t_2).

Definition (Komplexitätsklassen)

$F_\tau\phi(O(t)) =_{\text{def}} \bigcup_{k \geq 0} F_\tau\phi(k \cdot t)$ (Funktionsklasse)
$\tau\phi(O(t)) =_{\text{def}} \bigcup_{k \geq 0} \tau\phi(k \cdot t)$ (Sprachklasse)