

Komplexitätstheorie B

WiSe 19/20

Benedikt Lücken-Winkels

November 11, 2019

Contents

1	Wiederholung Berechenbarkeit und Komplexität	2
1.1	Probleme aus Formalen Sprachen (Nicht-leerheits-Problem)	2
1.2	Reduktion	2
1.3	Kodierung	2
1.4	Turingmaschinen	2
1.5	NP-Schwere	3
2	Parametrisierte Probleme	3
3	Allgemeines	3
3.1	Spezielles Halteproblem K	3
3.2	Allgemeines Halteproblem H	3
3.3	Grammatiken	3
3.4	Reduktion	4
3.5	Übersetzung von det TM auf nicht-det TM	4
3.6	Dynamisches Programmieren	4
4	Aufgaben	4

1 Wiederholung Berechenbarkeit und Komplexität

Ziel ist die eindeutige Zuweisung einer eines Problems zu einer Komplexitätsklasse anhand von geeigneten Maschinenmodellen. **Probleme** sind Entscheidungsprobleme bzw. Mengen von Ja-Instanzen.

1.1 Probleme aus Formalen Sprachen (Nicht-leerheits-Problem)

P_1 eine Phrasenstrukturgrammatik G . Gilt $L(G) \neq \emptyset$?

P_2 eine monotone Grammatik G . Gilt $L(G) \neq \emptyset$?

P_3 eine kontextfreie Grammatik G . Gilt $L(G) \neq \emptyset$? Ist entscheidbar, mit dem CYK-Algorithmus.

1.2 Reduktion

1.3 Kodierung

Abhängig von der Kodierung ändert sich die Problemgröße. Zulässige Kodierungen müssen berechenbar sein: Man kann einer Kodierung einer TM zB kein Halteproblembit einfügen, das bezeichnet, ob es eine Ja-Instanz ist.

Knapsack

- In: $a_1, \dots, a_n; p_1, \dots, p_n; a, c \in \mathbb{N}$
- $\exists I \subseteq \{1, \dots, n\}$ mit Summe von $p_i \geq p$ und Summe von $a_i \leq a$

1.4 Turingmaschinen

- Konfiguration: Randbegrenzer, beliebige Zeichen aus Alphabet, Zustandsübergänge
- Anfangskonfiguration: $\$s_0w$ *square* ist keine Konfiguration mit $w = \lambda$ (λ wird als blank definiert)
- Konfigurationsübergang: (beachte nach links nicht erweiterbar) Tupel aus
- Akzeptierte Sprache: Menge aller Eingabe Wörter für die wir beim Startsymbol mit beliebig vielen Schritten in einen Endzustand kommen.

Deterministische TM Für jedes Tupel aus Zustand und Alphabet gibt es höchstens ein Tripel aus Zustandübergängen

Mehrband TM Simultanes Arbeiten auf mehreren Bändern: Arbeitsband ist getrennt vom Eingabeband. Simulation von Mehrband TMs auf Einband TM durch **Spuren** und Aufblähung des Alphabets. Aufwand für eine k-Bandmaschine: $2k$

1.5 NP-Schwere

Generische Ausgangsproblem für NP-Schwere ist das Short NTM Acceptance.

SAT Formel in Aussagenlogik. Gibt es eine Belegung der Variablen, sodass die Formel erfüllt ist? Messung der Laufzeit eines det Algorithmus zB 2^n

2 Parametrisierte Probleme

Messen der Komplexität eines Problems nicht nur in Bitlänge, sondern Ressourcenbedarf. Parametrisiertes Problem (P, κ) mit κ als Parametrisierungsfunktion (Anzahl der Klauseln bei SAT). Für $(P, \kappa) \in FTP$ gibt es einen det Algorithmus (DTM) mit Zeit $O(f(\kappa(x))|x|^c)$.

Parametrisierte Reduktionen κ_2 der Eingabe x von P_1

3 Allgemeines

3.1 Spezielles Halteproblem K

- Eingabe: Kodierung einer TM
- Ja-Instanzen: Menge aller Kodierungen von TMs, die angewendet auf sich selbst halten.

3.2 Allgemeines Halteproblem H

- Einabe: Paar (Kodierung einer TM M die eine Grammatik realisiert, Wort)
- Ja-Instanzen: Das Wort gehört zu M

3.3 Grammatiken

- Phrasenstrukturgrammatik: Typ-0 Grammatik
- Kontextsensitiv: Typ-1 Grammatik
- Monoton
 - Rechte Seite darf beim Ersetzen nicht kürzer, als die linke Seite sein.
- Kontextfrei: Typ-2 Grammatik
 - Jede Regel hat genau ein NT-Symbol auf der linken Seite
 - Jede Regel hat eine nicht-leere Folge von NT- oder T-Symbolen auf der rechten Seite
- Rechtslinear: Äquivalent zu den Typ-3 Grammatiken

- Höchstens ein NT-Symbol auf der rechten Seite
- Das NT-Symbol darf nur am Ende der rechten Seite stehen
- Regulär: Typ-3 Grammatik
 - Rechte Seite nur das leere Wort, ein oder mehrere Terminalsymbole oder ein oder mehrere T-Symbole gefolgt von einem einzigen NT sein.

3.4 Reduktion

Karp-Reduktion

Many-One-Reduktion

3.5 Übersetzung von det TM auf nicht-det TM

3.6 Dynamisches Programmieren

Verwenden vorheriger Ergebnisse während der Laufzeit (\Rightarrow Rekursive Algorithmen)

4 Aufgaben

Aufgabe 1: Sind diese Sprachen entscheidbar?

P_4 eine zwei kontextfreie Grammatiken G_1, G_2 . Gilt $L(G_1) \cap L(G_2) \neq \emptyset$? Simulation von Berechnungen von TMs

P_5 eine zwei rechtslineare Grammatiken G_1, G_2 . Gilt $L(G_1) \cap L(G_2) \neq \emptyset$?

- Reguläre Sprachen sind Schnittabgeschlossen.
- Es gibt einen Algorithmus der aus dem Schnitt von 2 det TMs eine det TM liefert.
 \Rightarrow entscheidbar.

Aufgabe 2: Ist das Wortproblem (Folie 9) aufzählbar? $M \in K \Leftrightarrow f(M) = (G_K, c(M)) \in W$. G_K ist die Grammatik, die Kodierungen von TMs aus K beschreibt. Für $G \in W$ gilt, (G, w) mit $w \in L(G)$. Das heißt, eine TM M ist eine Ja-Instanz für K, sobald $L(G_K)$ eine Kodierung $(c(M))$ von M erzeugen kann.

Aufgabe 3: Wieso Polynomzeit für Unary Knapsack? Summe der Eingabe kodiert in Bits ergibt

- Verwende Datenstruktur Tabelle (Profit P/ Kosten A) und fülle sie mit 1 oder 0, wenn der Profit p_i mit Kosten a_j erreichbar ist

Aufgabe 4: Formalisierung der Kopfbewegungen einer TM (Folie 13) Vorsicht bei Linksbewegung, wegen \$

Aufgabe 5: Formalisierung einer DTM und Mehrband TM

Aufgabe 6: Ist die Zahlenkodierung/Einband-/Mehrband-TM bei Short NTM Acceptance wesentlich? Wenn die Schrittzahlbegrenzung k in unär kodiert ist. Besonderheit bei Mehrband:

Aufgabe 7: Wie schnell lässt sich SAT lösen? Bei KNF-SAT kann die Klauselanzahl eine Messgröße sein. Lässt sich das auf SAT übertragen?

Aufgabe 8: Lösen von VC mit Short NTM Acceptance (zwei many-one Polynomzeitreduktionen) Erstellen einer NTM und dann Reduktion mit Short NTM Acceptance (Rechentepich)

- 1. Reduktion:
- 2. Reduktion:

Aufgabe 9: Zeigen, dass $VC \leq \text{Short NTM Acceptance}$ gilt. und umgekehrt

Aufgabe 10

Aufgabe 11

Aufgabe 12

Aufgabe 13

Aufgabe 14

Aufgabe 15