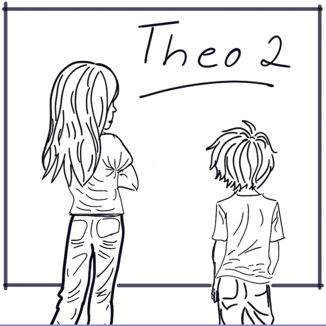
Theoretische Informatik 2

SoSe 2024

Berechenbarkeit und Komplexität

Prof. Dr. Sebastian Siebertz
AG Theoretische Informatik
MZH, Raum 3160
siebertz@uni-bremen.de

Universität Bremen



Die Chomsky-Hierarchy

Definition Chomsky-Hierarchy

Es sei $G = (N, \Sigma, P, S)$ eine Grammatik.

- G ist Grammatik vom Typ 3 (rechtslinear), falls alle Regeln die Form $A \longrightarrow uB$ oder $A \longrightarrow u$ haben mit $A, B \in N$, $u \in \Sigma^*$.
- G ist Grammatik vom Typ 2 (kontextfrei), falls alle Regeln die Form $A \longrightarrow w$ haben mit $A \in N, w \in (\Sigma \cup N)^*$.
- G ist Grammatik vom Typ 1 (nicht verkürzend, kontextsensitiv), falls alle Regeln nicht verkürzend sind, also die Form $w \longrightarrow u$ haben wobei $w, u \in (\Sigma \cup N)^+$ und $|u| \geq |w|$.

  Ausnahme: Die Regel $S \longrightarrow \varepsilon$ ist erlaubt, wenn S in keiner Produktion auf der rechten Seite vorkommt.
- Jede Grammatik G ist Grammatik vom Typ 0.

Automaten/Maschinenmodelle

- Jeder Grammatiktyp korrespondiert zu einem Maschinen/Automatenmodell.
  - Typ 3: endliche Automaten
  - Typ 2: Kellerautomaten
  - Typ 1: linear beschränkte NTM (diese Vorlesung)
  - Typ 0: Turingmaschine (letzte Vorlesung)

Grammatiken vs Turingmaschinen

Satz

Sei L eine Sprache. Dann ist L semi-entscheidbar genau dann, wenn $L \in \mathcal{L}_0$.

Satz

Eine Sprache $L \subseteq \Sigma^*$ ist entscheidbar genau dann, wenn L und $\bar{L}$ semi-entscheidbar sind.

Korollar

Eine Sprache $L \subseteq \Sigma^*$ ist entscheidbar genau dann, wenn es eine Grammatik G und eine Grammatik $\bar{G}$ gibt mit $L(G) = L$ und $L(\bar{G}) = \bar{L}$.

Kontextsensitive Grammatiken

- G ist Grammatik vom Typ 1 (nicht verkürzend, kontextsensitiv), falls alle Regeln nicht verkürzend sind, also die Form $w \longrightarrow u$ haben wobei $w, u \in (\Sigma \cup N)^+$ und $|u| \geq |w|$.
- Ableitung $S \vdash_G x_1 \vdash_G x_2 \vdash_G \ldots \vdash_G x_n = w$ in Typ 1 Grammatik:

$$|x_i| \leq |x_j| \leq |w| \text{ für alle } 1 \leq i \leq j \leq n.$$

- Nichtdeterministische Turingmaschine, die die Ableitung simuliert braucht nur Speicher $1, \ldots, |w|$.

Linear beschränkte Turingmaschine.

Outline

- In dieser Vorlesung:
  - Linear beschränkte deterministische Turingmaschinen.
  - Linear beschränkte nichtdeterministische Turingmaschinen.
  - Zusammenhang Typ 1 Grammatiken und linear beschränkte NTM.

Deterministische linear beschränkte Turingmaschinen

- Linear beschränkte deterministische Turingmaschine:
  Tupel $M = (Q, \Sigma, \Gamma, [,], \sqcup, \delta, q_s, q_a, q_r)$, wobei
  - Q eine endliche Zustandsmenge,
  - Σ das Eingabealphabet ist und Γ das Arbeitsalphabet ist mit $\Sigma \subseteq \Gamma$,
  - $[\in \Gamma \setminus \Sigma$ der linke Endmarker ist,
  - **$] \in \Gamma \setminus \Sigma$ der rechte Endmarker ist,**
  - $\sqcup \in \Gamma \setminus \Sigma$ das Blanksymbol ist,
  - $\delta : Q \times \Gamma \to Q \times \Gamma \times \{-1, 0, 1\}$ die Übergangsfunktion ist, so dass
    (1) für alle $p \in Q$ ein $q \in Q$ und $d \in \{0, 1\}$ existieren, so dass $\delta(p, [) = (q, [, d)$,
    (2) **für alle $p \in Q$ ein $q \in Q$ und $d \in \{-1, 0\}$ existieren, so dass $\delta(p,]) = (q,], d)$,**
    (2) für alle $a \in \Gamma$ gilt $\delta(q_a, a) = (q_a, a, 0)$ und $\delta(q_r, a) = (q_r, a, 0)$,
  - $q_s \in Q$ der Startzustand ist,
  - $q_a \in Q$ der akzeptierende Zustand ist und $q_r \in Q$ der verwerfende Zustand ist.

Konfigurationen und Berechnungen (mehr oder weniger Wiederholung)

- Sei $M = (Q, \Sigma, \Gamma, [,], \sqcup, \delta, q_s, q_a, q_r)$ eine deterministische linear beschränkte TuringMaschine.
  - Eine Konfiguration von m ist ein Tupel

$$\beta = (q, w, k) \in Q \times \Gamma^* \times \mathbb{N},$$

    so dass $w(1) = [$ und $w(|w| + 2) =]$.
    - M kann niemals den von $[$ und $]$ eingeschlossenen Bereich verlassen, also reicht $w \in \Gamma^*$ in den Konfigurationen.
  - $\alpha_1 = (q_s, [w], 1)$ ist die Startkonfiguration von M auf $w \in \Sigma^*$.

Konfigurationen und Berechnungen

- Eine Konfiguration $(q_a, w, k) \in \{q_a\} \times \Gamma^* \times \mathbb{N}$ heißt akzeptierend.
- Eine Konfiguration $(q_r, w, k) \in \{q_r\} \times \Gamma^* \times \mathbb{N}$ heißt verwerfend.
- Sei

  $$\alpha = (p, w, k) \in Q \times \Gamma^* \times \mathbb{N}$$

  eine Konfiguration. Sei

  $$\delta(p, w_k) = (q, b, d) \in Q \times \Gamma \times \{-1, 0, 1\}.$$

  Dann ist die Folgekonfiguration von α die Konfiguration

  $$\beta = (q, w[k \mapsto b], k + d)$$

  Wir schreiben $\alpha \vdash_M \beta$.

Konfigurationen und Berechnungen

- Berechnung von M auf w: (eindeutig bestimmte) unendliche Konfigurationsfolge

$$\alpha_1 \vdash_M \alpha_2 \vdash_M \dots,$$

wobei α_1 die Startkonfiguration von M auf w ist.

- Eine Berechnung $\alpha_1 \vdash_M \alpha_2 \vdash_M \dots$ ist akzeptierend, falls α_i akzeptierend ist für ein $i \geq 1$ und verwerfend, falls α_i verwerfend ist für ein $i \geq 1$.
- Die Maschine M hält oder terminiert auf w falls es eine akzeptierende oder verwerfende Berechnung von M auf w gibt. Sie akzeptiert w im ersten Fall und verwirft w im zweiten Fall.
- Die von M erkannte Sprache ist $L(M) = \{w \in \Sigma^* \ : \ m \text{ akzeptiert } w\}$.

Linear beschränkte nichtdeterministische Turingmaschinen

- Eine nichtdeterministische linear beschränkte Turingmaschine ist eine nichtdeterministische 1-Band Turingmaschine $M = (Q, \Sigma, \Gamma, \llcorner\!\lrcorner, \Delta, Q_s, F)$ mit der Eigenschaft, dass wenn $(q, \llcorner\!\lrcorner, q', b, d) \in \Delta$, so gilt $b = \llcorner\!\lrcorner$.
  - Das Symbol $\llcorner\!\lrcorner$ wird niemals überschrieben.
  - Es wird nur der Speicher $1, \dots, |w|$ wirklich benutzt.
  - Aber die Maschine kann den Kopf beliebig weit bewegen.
    - Wir zeigen: mit solchen Kopfbewegungen verschwendet die Maschine nur Zeit!

Linear beschränkte nichtdeterministische Turingmaschinen

Lemma

Sei $M = (Q, \Sigma, \Gamma, \llcorner\!\lrcorner, \Delta, Q_s, F)$ *eine nichtdeterministische linear beschränkte Turingmaschine.*

Dann existiert eine nichtdeterministische linear beschränkte Turingmaschine $M' = (Q', \Sigma, \Gamma, \llcorner\!\lrcorner, \Delta', Q_s', F')$ *mit*

- ▷ $L(M) = L(M')$ *und*
- ▷ M' *besucht auf Eingabe* $w \in \Sigma^*$ *nur die Positionen* $0, 1, \ldots, |w| + 1$.

- Beweis im Skript.
- Idee: kürze Berechnung im negativen Bereich bzw. Bereich $|w| + 2, |w| + 3, \ldots$ durch eine direkte Transition ab.

Offene Frage der Informatik

Offene Frage der Informatik

Kann jede von einer nichtdeterministischen linear beschränkten Turingmaschine M erkannte Sprache auch von einer deterministischen linear beschränkten Turingmaschine M' erkannt werden?

Turingmaschinen vs Grammatiken

Satz

Sei L eine Sprache. Dann gilt $L \in \mathcal{L}_1$ genau dann, wenn L von einer nichtdeterministischen linear beschränkten Turingmaschine erkannt wird.

Beweis.

- Sei $L \in \mathcal{L}_1$ und sei G eine kontextsensitive Grammatik mit $L(G) = L$.
- Ableitung $S \vdash_G x_1 \vdash_G x_2 \vdash_G \ldots \vdash_G x_n = w$ in Typ 1 Grammatik:

$$|x_i| \leq |x_j| \leq |w| \text{ für alle } 1 \leq i \leq j \leq n.$$

- Konstruiere linear beschränkte Turingmaschine $M = (Q, \Sigma, \Gamma, \sqcup, \Delta, Q_s, F)$ mit $L(M) = L(G)$.
- Konstruktion genau wie für $L \in \mathcal{L}_0$ mit einer 2-Spur Maschine, bei der eine Ableitung auf der zweiten Spur nur ausgeführt werden kann, wenn die erste Spur an dieser Stelle beschrieben ist.

Turingmaschinen vs Grammatiken

- Umgekehrt sei $L = L(M)$ für eine linear beschränkte Turingmaschine $M = (Q, \Sigma, \Gamma, \sqcup, \Delta, Q_s, F)$.
  - Wir nehmen an, dass $\varepsilon \notin L(M)$, sonst füge die Regel $S \to \varepsilon$ hinzu.
- Nach Lemma können wir annehmen, dass M auf einer Eingabe w nur die Positionen $0, 1, \ldots, |w| + 1$ besucht.
- Außerdem nehmen wir an, dass M auf Positionen $1, \ldots, |w|$ niemals ein $\sqcup$ schreibt.
- Konfigurationen kodiert als $\sqcup v_1 \ldots v_{k-1}\, q\, v_k \ldots v_{|w|} \sqcup \in \Gamma^*$.
- Konstruiere $G = (N, \Sigma, P, S)$ mit $L(G) = L(M)$.

Wiederholung/Anpassung der allgemeinen Konstruktion

- Für eine NTMT definieren wir $G = (N, \Sigma, P, S)$ mit $N = Q \cup \Gamma \setminus \Sigma \cup \{S, W, B, A\}$ und den folgenden Regeln.
- Regeln für die Ableitung von α_1 für die Startkonfiguration von M auf w.
  - $S \longrightarrow \sqcup q_s W$,
  - $W \longrightarrow aW$ für $a \in \Sigma$,
  - $W \longrightarrow B$,
  - $B \longrightarrow \sqcup$.
- Das sind nicht-verkürzende Regeln $\checkmark\checkmark\checkmark$

Wiederholung/Anpassung der allgemeinen Konstruktion

- Regeln zur Simulation der Berechnung.
  - Für alle $q, q' \in Q, a, b, c \in \Gamma$ mit $\delta(q, a) = (q', b, -1)$ die Regel
    $$cqa \longrightarrow q'cb.$$
  - Für alle $q, q' \in Q, a, b \in \Gamma$ mit $\delta(q, a) = (q', b, 0)$ die Regel
    $$qa \longrightarrow q'b.$$
  - Für alle $q, q' \in Q, a, b \in \Gamma$ mit $\delta(q, a) = (q', b, +1)$ die Regel
    $$qa \longrightarrow bq'.$$
- Das sind nicht-verkürzende Regeln ✓✓✓

Wiederholung/Anpassung der allgemeinen Konstruktion

- Letzte Regeln: Aufräumphase (angepasst)
- Wir können wieder annehmen, dass wenn M das Wort w akzeptiert, so terminiert sie in Konfiguration $(q_a, \sqcup w \sqcup, 1)$ für ein eindeutiges $q_a \in F$.
- Hieraus muss G noch w ableiten.
  - $\sqcup \longrightarrow \varepsilon$,
  - $q_a \longrightarrow \varepsilon$.
- Problem: Das sind verkürzende Regeln!

Anpassung der allgemeinen Konstruktion

- Lösung: Benutze Kombinationen von Symbolen als Nichtterminale
  - $S \longrightarrow (\sqcup q_s)(aW)$ für $a \in \Sigma$,
    wobei $(\sqcup q_s)$ und (aW) einzelne Symbole sind.
  - $(aW) \longrightarrow a(bW)$ für $a, b \in \Sigma$.
  - $(aW) \longrightarrow (a\sqcup)$.

Anpassung der allgemeinen Konstruktion

- Weitere Nichtterminale als Kombinationen:
  - $(qa), (aq), (\sqcup a), (a\sqcup), (\sqcup qa)$ und $(aq\sqcup)$ für $q \in Q, a \in \Gamma$.
- Regeln zur Simulation der Berechnung:
  - z.B. für alle $q, q' \in Q, a, b, c \in \Gamma$ mit $(q, a, q', b, -1) \in \Delta$
  - vorher
$$cqa \longrightarrow q'cb.$$
  - jetzt
$$c(qa) \longrightarrow (q'c)b.$$
- Regeln um Nichtterminale verschwinden zu lassen.
  - Für $a \in \Gamma$
    - $(q_a a) \longrightarrow a$
    - $(\sqcup a) \longrightarrow a$
    - $(a\sqcup) \longrightarrow a$

Beweis fortgesetzt

- G hat also keine verkürzenden Regeln.
- Jetzt zeigen wir wie für den allgemeinen Fall, dass $w \in L(G) \Leftrightarrow w \in L(M)$.

Satz

Sei L eine Sprache. Dann gilt $L \in \mathcal{L}_1$ genau dann, wenn L von einer nichtdeterministischen linear beschränkten Turingmaschine erkannt wird.

Kuroda-Normalform

- Eine Grammatik $G = (N, \Sigma, P, S)$ ist in *Kuroda-Normalform*, wenn alle Produktionen eine der folgenden Formen haben:
  - $A \longrightarrow a$ für $A \in N, a \in \Sigma$,
  - $A \longrightarrow B$ für $A, B \in N$,
  - $A \longrightarrow BC$ für $A, B, C \in N$,
  - $AB \longrightarrow CD$ für $A, B, C, D \in N$.
- Grammatik in Kuroda-Normalform: kontextsensitiv
- Wir haben mit unserer Konstruktion gezeigt, dass jede kontextsensitive Grammatik G äquivalent ist zu einer Grammatik in Kuroda-Normalform (falls $\varepsilon \in L$, so ist $S \to \varepsilon$ erlaubt, aber dann S darf nicht auf einer rechten Seite vorkommen).
- Allgemein brauchen wir auch $A \to \varepsilon$ (siehe Übung).