



Theoretische Grundlagen der Informatik

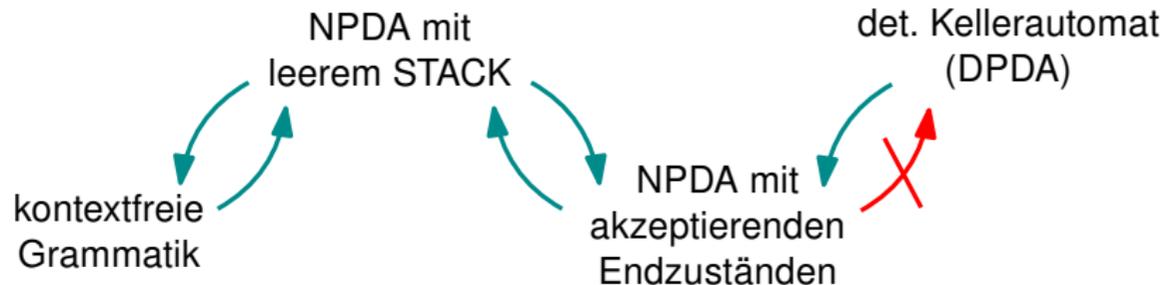
Vorlesung am 2.2.2022

Torsten Ueckerdt | 2. Februar 2022

Letzte Vorlesung: Ein Maschinenmodell für Chomsky-2

Satz.

Jede durch einen NPDA durch leeren STACK akzeptierte Sprache ist kontextfrei.



Wiederholung: Kellerautomaten

Definition.

Ein nichtdeterministischer **Kellerautomat** (NPDA) besteht aus $(Q, \Sigma, \Gamma, q_0, Z_0, \delta)$, wobei

- Q Zustandsmenge, $q_0 \in Q$ Anfangszustand, Σ Eingabealphabet
- Γ endliches STACK-Alphabet, $Z_0 \in \Gamma$ Initialisierung des STACK
- $\delta: Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \rightarrow 2^{Q \times \Gamma^*}$ Übergangsrelation, d.h.
 - $\delta(q, a, Z) \subseteq \{(q, \gamma) \mid q \in Q, \gamma \in \Gamma^*\}$
 - $\delta(q, \varepsilon, Z) \subseteq \{(q, \gamma) \mid q \in Q, \gamma \in \Gamma^*\}$

Wiederholung: Kellerautomaten

Definition.

Ein nichtdeterministischer **Kellerautomat** (NPDA) besteht aus $(Q, \Sigma, \Gamma, q_0, Z_0, \delta)$, wobei

- Q Zustandsmenge, $q_0 \in Q$ Anfangszustand, Σ Eingabealphabet
- Γ endliches STACK-Alphabet, $Z_0 \in \Gamma$ Initialisierung des STACK
- $\delta: Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \rightarrow 2^{Q \times \Gamma^*}$ Übergangsrelation, d.h.
 - $\delta(q, a, Z) \subseteq \{(q, \gamma) \mid q \in Q, \gamma \in \Gamma^*\}$
 - $\delta(q, \varepsilon, Z) \subseteq \{(q, \gamma) \mid q \in Q, \gamma \in \Gamma^*\}$

Eine **Konfiguration eines NPDA** ist ein Tripel (q, w, α) mit

- $q \in Q$,
- $w \in \Sigma^*$ der Teil der Eingabe, der noch nicht gelesen wurde,
- $\alpha \in \Gamma^*$ STACK-Inhalt.

Wiederholung: Kellerautomaten

Eine **Konfiguration eines NPDA** ist ein Tripel (q, w, α) mit

- $q \in Q$,
- $w \in \Sigma^*$ der Teil der Eingabe, der noch nicht gelesen wurde,
- $\alpha \in \Gamma^*$ STACK-Inhalt.

Definition.

Ein NPDA **akzeptiert** ein $w \in \Sigma^*$ **durch leeren STACK**, wenn es eine zulässige Folge von Konfigurationen aus der Anfangskonfiguration (q_0, w, Z_0) in eine Konfiguration $(q, \varepsilon, \varepsilon)$, $q \in Q$, gibt.

Wiederholung: Kellerautomaten

Eine **Konfiguration eines NPDA** ist ein Tripel (q, w, α) mit

- $q \in Q$,
- $w \in \Sigma^*$ der Teil der Eingabe, der noch nicht gelesen wurde,
- $\alpha \in \Gamma^*$ STACK-Inhalt.

Definition.

Ein NPDA **akzeptiert** ein $w \in \Sigma^*$ **durch leeren STACK**, wenn es eine zulässige Folge von Konfigurationen aus der Anfangskonfiguration (q_0, w, Z_0) in eine Konfiguration $(q, \varepsilon, \varepsilon)$, $q \in Q$, gibt.

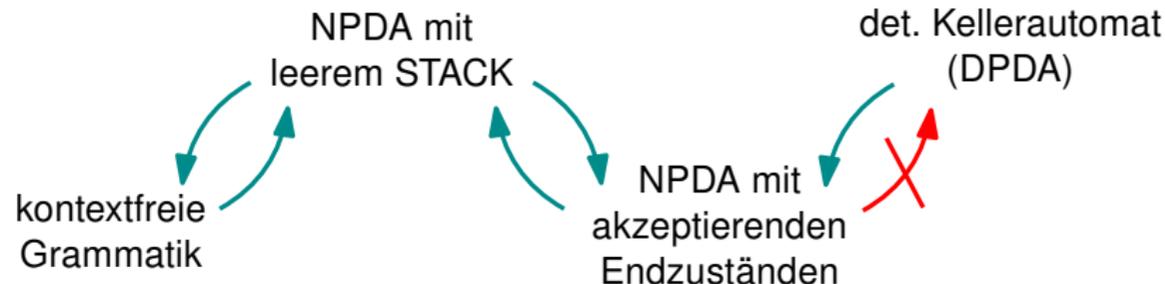
Ziel:

Für jeden beliebigen NPDA \mathcal{A} der eine Sprache L durch leeren STACK akzeptiert, konstruiere eine kontextfreie Grammatik G mit $L(G) = L$.

Ein Maschinenmodell für Chomsky-2

Satz.

Jede durch einen NPDA durch leeren STACK akzeptierte Sprache ist kontextfrei.



Beweis: NPDA \rightarrow kontextfreie Grammatik

- Sei $\mathcal{A} = (Q, \Sigma, \Gamma, \delta, q_0, Z_0)$ NPDA, der $L_{\mathcal{A}}$ durch leeren STACK akzeptiert.
- Wir geben eine kontextfreie Grammatik $G = (\Sigma, V, S, R)$ mit $L_{\mathcal{A}} = L(G)$ an.

Die Konstruktion von G heißt **Tripelkonstruktion**.

- Setze $V := \{[q, X, p] \mid p, q \in Q, X \in \Gamma\} \cup \{S\}$.
- Sei S Startsymbol.

Ziel: Aus $[q, X, p]$ sollen genau die $w \in \Sigma^*$ ableitbar sein, für die es eine Abarbeitung von \mathcal{A} gibt,

- die im Zustand q mit STACK-Inhalt X beginnt und
- nach Lesen von w im Zustand p mit leerem STACK endet.

Beweis: NPDA \rightarrow kontextfreie Grammatik

- Sei $\mathcal{A} = (Q, \Sigma, \Gamma, \delta, q_0, Z_0)$ NPDA, der $L_{\mathcal{A}}$ durch leeren STACK akzeptiert.
- Wir geben eine kontextfreie Grammatik $G = (\Sigma, V, S, R)$ mit $L_{\mathcal{A}} = L(G)$ an.

Die Konstruktion von G heißt **Tripelkonstruktion**.

- Setze $V := \{[q, X, p] \mid p, q \in Q, X \in \Gamma\} \cup \{S\}$.
- Sei S Startsymbol.

Die Regelmenge R ist gegeben durch

- $S \rightarrow [q_0, Z_0, q]$ für alle $q \in Q$
- $[q, X, q_{m+1}] \rightarrow a[q_1, Y_1, q_2][q_2, Y_2, q_3] \cdots [q_m, Y_m, q_{m+1}]$
für alle Möglichkeiten $q_2, q_3, \dots, q_{m+1} \in Q$,
falls $(q_1, Y_1 Y_2 \cdots Y_m) \in \delta(q, a, X)$,
- insbes. $[q, X, p] \rightarrow a$ falls $(p, \varepsilon) \in \delta(q, a, X)$.

Notation

- Eine Konfiguration eines NPDAs ist (q, w, α) mit q aktueller Zustand, w noch zu lesende Eingabe, α aktueller STACK-Inhalt.
- Für einen Schritt von Konfiguration (q, w, α) zu Konfiguration (p, w', β) schreiben wir

$$(q, w, \alpha) \vdash (p, w', \beta).$$

- Für eine Folge von Konfigurationen (q, w, α) nach (p, w', β) schreiben wir auch

$$(q, w, \alpha) \vdash^* (p, w', \beta)$$

- beziehungsweise

$$(q, w, \alpha) \vdash^k (p, w', \beta)$$

für eine Folge von genau k Konfigurationen.

Beweis: NPDA \rightarrow kontextfreie Grammatik

Wir werden per Induktion beweisen, dass für alle $p, q \in Q$, $X \in \Gamma$ und $w \in L$ gilt:

$$[q, X, p] \xrightarrow{*} w \text{ in } G \iff (q, w, X) \vdash^* (p, \varepsilon, \varepsilon)$$

Aus dieser Behauptung folgt dann

$$\begin{aligned}
 w \in L_{\mathcal{A}} &\iff \exists p \in Q \text{ mit } (q_0, w, Z_0) \vdash^* (p, \varepsilon, \varepsilon), \text{ wobei} \\
 &\quad (q_0, w, Z_0) \text{ Anfangskonfiguration von } \mathcal{A} \text{ ist} \\
 &\iff \exists p \in Q \text{ mit } [q_0, Z_0, p] \xrightarrow{*} w \\
 &\iff \exists p \in Q \text{ mit } S \rightarrow [q_0, Z_0, p] \xrightarrow{*} w \\
 &\iff w \in L(G)
 \end{aligned}$$

Beweis: NPDA \rightarrow kontextfreie Grammatik

Richtung:

- $[q, X, p] \xrightarrow{*} w \text{ in } G \implies (q, w, X) \vdash^* (p, \varepsilon, \varepsilon)$

Beschreibung:

- Induktion über die Länge k einer Ableitung $[q, X, p] \xrightarrow{k} w \text{ in } G$

Beweis: NPDA \rightarrow kontextfreie Grammatik

- $V := \{[q, X, p] \mid p, q \in Q, X \in \Gamma\} \cup \{S\}$.
- $[q, X, q_{m+1}] \rightarrow a[q_1, Y_1, q_2][q_2, Y_2, q_3] \cdots [q_m, Y_m, q_{m+1}]$
für alle $q_2, q_3, \dots, q_{m+1} \in Q$, mit $(q_1, Y_1 Y_2 \cdots Y_m) \in \delta(q, a, X)$.

Induktionsanfang:

- Für $k = 1$ gilt, dass $[q, X, p] \rightarrow w$ eine Regel in G ist.
- Also ist $(p, \varepsilon) \in \delta(q, w, X)$ und $|w| = 1$.
- Also gibt es die Abarbeitung $(q, w, X) \vdash^1 (p, \varepsilon, \varepsilon)$ in \mathcal{A} .

Wir zeigen:

$$\begin{array}{c}
 [q, X, p] \xrightarrow{k} w \text{ in } G \\
 \Downarrow \\
 (q, w, X) \vdash^* (p, \varepsilon, \varepsilon)
 \end{array}$$

Beweis: NPDA \rightarrow kontextfreie Grammatik

- $V := \{[q, X, p] \mid p, q \in Q, X \in \Gamma\} \cup \{S\}$.
- $[q, X, q_{m+1}] \rightarrow a[q_1, Y_1, q_2][q_2, Y_2, q_3] \cdots [q_m, Y_m, q_{m+1}]$
für alle $q_2, q_3, \dots, q_{m+1} \in Q$, mit $(q_1, Y_1 Y_2 \cdots Y_m) \in \delta(q, a, X)$.

Induktionsschritt:

- Betrachte eine Ableitung $[q, X, p] \xrightarrow{k} w$.
- Schreibe diese als

$$[q, X, p] \rightarrow a[q_1, Y_1, q_2][q_2, Y_2, q_3] \cdots [q_m, Y_m, q_{m+1}] \xrightarrow{k-1} w,$$

wobei $q_{m+1} = p$ und $w = aw_1 \cdots w_m$, mit $w_i \in \Sigma^*$, $a \in \Sigma$ und

$$[q_j, Y_j, q_{j+1}] \xrightarrow{k'} w_j \text{ mit } k' \leq k - 1 \text{ für alle } 1 \leq j \leq m.$$

Wir zeigen:

$$\begin{array}{c} [q, X, p] \xrightarrow{k} w \text{ in } G \\ \Downarrow \\ (q, w, X) \vdash^* (p, \varepsilon, \varepsilon) \end{array}$$

Beweis: NPDA \rightarrow kontextfreie Grammatik

- $V := \{[q, X, p] \mid p, q \in Q, X \in \Gamma\} \cup \{S\}$.
- $[q, X, q_{m+1}] \rightarrow a[q_1, Y_1, q_2][q_2, Y_2, q_3] \cdots [q_m, Y_m, q_{m+1}]$
für alle $q_2, q_3, \dots, q_{m+1} \in Q$, mit $(q_1, Y_1 Y_2 \cdots Y_m) \in \delta(q, a, X)$.

Induktionsschritt:

- Induktionsvoraussetzung: $(q_j, w_j, Y_j) \vdash^* (q_{j+1}, \varepsilon, \varepsilon)$ für alle $1 \leq j \leq m$.
- Also $(q_j, w_j, Y_j \cdots Y_m) \vdash^* (q_{j+1}, \varepsilon, Y_{j+1} \cdots Y_m)$ für alle $1 \leq j \leq m$.
- Damit $(q, w, X) \vdash (q_1, w_1 \cdots w_m, Y_1 \cdots Y_m)$
 $\vdash^* (q_2, w_2 \cdots w_m, Y_2 \cdots Y_m)$
 $\vdash^* (q_3, w_3 \cdots w_m, Y_3 \cdots Y_m)$
 $\vdash^* \cdots \vdash^* (q_m, w_m, Y_m) \vdash^* (q_{m+1}, \varepsilon, \varepsilon) = (p, \varepsilon, \varepsilon)$

Wir zeigen:

$$\begin{array}{c}
 [q, X, p] \xrightarrow{k} w \text{ in } G \\
 \Downarrow \\
 (q, w, X) \vdash^* (p, \varepsilon, \varepsilon)
 \end{array}$$

Beweis: NPDA \rightarrow kontextfreie Grammatik

Richtung:

- $[q, X, p] \xrightarrow{*} w \text{ in } G \iff (q, w, X) \vdash^* (p, \varepsilon, \varepsilon)$

Beschreibung:

- Induktion über die Länge k einer Abarbeitung $(q, w, X) \vdash^k (p, \varepsilon, \varepsilon)$

Beweis: NPDA \rightarrow kontextfreie Grammatik

- $V := \{[q, X, p] \mid p, q \in Q, X \in \Gamma\} \cup \{S\}$.
- $[q, X, q_{m+1}] \rightarrow a[q_1, Y_1, q_2][q_2, Y_2, q_3] \cdots [q_m, Y_m, q_{m+1}]$
für alle $q_2, q_3, \dots, q_{m+1} \in Q$, mit $(q_1, Y_1 Y_2 \cdots Y_m) \in \delta(q, a, X)$.

Induktionsanfang:

- Für $k = 1$ folgt aus $(q, w, X) \vdash (p, \varepsilon, \varepsilon)$, dass
 - $|w| = 1$ und
 - $(p, \varepsilon) \in \delta(q, w, X)$.
- Dann ist $[q, X, p] \rightarrow w$ eine Regel von G .

Wir zeigen:

$$\begin{array}{c}
 [q, X, p] \xrightarrow{*} w \text{ in } G \\
 \uparrow \\
 (q, w, X) \vdash^k (p, \varepsilon, \varepsilon)
 \end{array}$$

Beweis: NPDA \rightarrow kontextfreie Grammatik

- $V := \{[q, X, p] \mid p, q \in Q, X \in \Gamma\} \cup \{S\}$.
- $[q, X, q_{m+1}] \rightarrow a[q_1, Y_1, q_2][q_2, Y_2, q_3] \cdots [q_m, Y_m, q_{m+1}]$
für alle $q_2, q_3, \dots, q_{m+1} \in Q$, mit $(q_1, Y_1 Y_2 \cdots Y_m) \in \delta(q, a, X)$.

Induktionsschritt:

- Betrachte eine Abarbeitung $(q, w, X) \stackrel{k}{\vdash} (p, \varepsilon, \varepsilon)$
- Zerlege $w = aw'$ wobei
 - $a = \varepsilon$, falls der erste Schritt von \mathcal{A} ein ε -Übergang ist
 - $a \in \Sigma$, also der erste Buchstabe von w , sonst.
- Sei $(q_1, w', Y_1 \cdots Y_m)$ die Konfiguration von \mathcal{A} nach dem 1. Schritt.
- Dann gilt $(q, aw', X) \vdash (q_1, w', Y_1 \cdots Y_m) \stackrel{k'}{\vdash} (p, \varepsilon, \varepsilon)$
mit $k' = k - 1$.

Wir zeigen:

$$\begin{array}{c}
 [q, X, p] \xrightarrow{*} w \text{ in } G \\
 \uparrow \\
 (q, w, X) \stackrel{k}{\vdash} (p, \varepsilon, \varepsilon)
 \end{array}$$

Beweis: NPDA \rightarrow kontextfreie Grammatik

- $V := \{[q, X, p] \mid p, q \in Q, X \in \Gamma\} \cup \{S\}$.
- $[q, X, q_{m+1}] \rightarrow a[q_1, Y_1, q_2][q_2, Y_2, q_3] \cdots [q_m, Y_m, q_{m+1}]$
für alle $q_2, q_3, \dots, q_{m+1} \in Q$, mit $(q_1, Y_1 Y_2 \cdots Y_m) \in \delta(q, a, X)$.

Sei

$$w' = w_1 \cdots w_m \text{ Zerlegung von } w \text{ mit } w_j \in \Sigma^*$$

so, dass \mathcal{A} startend mit der Konfiguration

$$(q_1, w', Y_1 \cdots Y_m)$$

bei der betrachteten Abarbeitung gerade nach dem Lesen von $w_1 \cdots w_j$ zum ersten Mal den STACK-Inhalt $Y_{j+1} \cdots Y_m$ erzeugt. Sei q_{j+1} der zu diesem Zeitpunkt erreichte Zustand.

Wir zeigen:

$$\begin{array}{c}
 [q, X, p] \xrightarrow{*} w \text{ in } G \\
 \uparrow \\
 (q, w, X) \stackrel{k}{\vdash} (p, \varepsilon, \varepsilon)
 \end{array}$$

Beweis: NPDA \rightarrow kontextfreie Grammatik

- $V := \{[q, X, p] \mid p, q \in Q, X \in \Gamma\} \cup \{S\}$.
- $[q, X, q_{m+1}] \rightarrow a[q_1, Y_1, q_2][q_2, Y_2, q_3] \cdots [q_m, Y_m, q_{m+1}]$
für alle $q_2, q_3, \dots, q_{m+1} \in Q$, mit $(q_1, Y_1 Y_2 \cdots Y_m) \in \delta(q, a, X)$.

Dann gilt: $q_{m+1} = p$ und

$$(q_j, w_j \cdots w_m, Y_j \cdots Y_m) \stackrel{k'}{\vdash} (q_{j+1}, w_{j+1} \cdots w_m, Y_{j+1} \cdots Y_m),$$

$k' \leq k - 1$, und während der gesamten Abarbeitung liegt $Y_{j+1} \cdots Y_m$ ungelesen auf dem STACK.

Also gilt auch

$$(q_j, w_j, Y_j) \stackrel{k'}{\vdash} (q_{j+1}, \varepsilon, \varepsilon).$$

Wir zeigen:

$$\begin{array}{c}
 [q, X, p] \xrightarrow{*} w \text{ in } G \\
 \uparrow \\
 (q, w, X) \stackrel{k}{\vdash} (p, \varepsilon, \varepsilon)
 \end{array}$$

Beweis: NPDA \rightarrow kontextfreie Grammatik

- $V := \{[q, X, p] \mid p, q \in Q, X \in \Gamma\} \cup \{S\}$.
- $[q, X, q_{m+1}] \rightarrow a[q_1, Y_1, q_2][q_2, Y_2, q_3] \cdots [q_m, Y_m, q_{m+1}]$
für alle $q_2, q_3, \dots, q_{m+1} \in Q$, mit $(q_1, Y_1 Y_2 \cdots Y_m) \in \delta(q, a, X)$.

Also gilt auch

$$(q_j, w_j, Y_j) \stackrel{k'}{\vdash} (q_{j+1}, \varepsilon, \varepsilon).$$

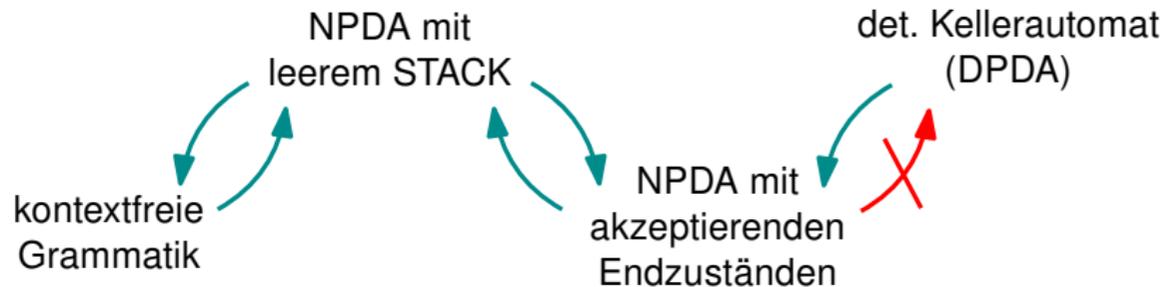
Nach Induktionsvoraussetzung folgt daraus, dass $[q_j, Y_j, q_{j+1}] \xrightarrow{*} w_j$ in G existiert. Damit erhalten wir, dass auch

$[q, X, p] \rightarrow a[q_1, Y_1, q_2][q_2, Y_2, q_3] \cdots [q_m, Y_m, q_{m+1}] \xrightarrow{*} aw_1 \cdots w_m = w$
in G existiert.

Wir zeigen:

$$\begin{array}{c}
 [q, X, p] \xrightarrow{*} w \text{ in } G \\
 \uparrow \\
 (q, w, X) \stackrel{k}{\vdash} (p, \varepsilon, \varepsilon)
 \end{array}$$

Übersicht



Korollar.

Die Klasse der von nichtdeterministischen Kellerautomaten akzeptierten Sprachen ist gleich der Klasse der kontextfreien Sprachen.

Ein Maschinenmodell für Chomsky-2

Typ-0	semi-entscheidbar	\iff	DTM / NTM akzeptiert
Typ-1	kontextsensitiv	\iff	$NTAP\mathcal{E}(n)$ (offen, ob $NTAP\mathcal{E}(n) = DTAP\mathcal{E}(n)$)
Typ-2	kontextfrei	\iff	nichtdet. Kellerautomat (NPDA)
Typ-3	regulär	\iff	DEA / NEA

Korollar.

Die Klasse der von nichtdeterministischen Kellerautomaten akzeptierten Sprachen ist gleich der Klasse der kontextfreien Sprachen.

Exkurs

Wofür braucht man eigentlich Grammatiken und Berechnungsmodelle wie endliche Automaten oder Turingmaschinen?

- Die Chomsky-Hierarchie wurde 1956 von dem Linguisten Noam Chomsky entworfen. Ursprünglich war sie als Mittel zur Beschreibung natürlicher Sprachen gedacht (hat sich nicht erfüllt).
- Grammatiken und Automaten sind fundamental für die Beschreibung von Programmiersprachen.
- XML basiert auf sogenannten Dokumenttypdefinitionen (DTD). Diese sind kontextfreie Grammatiken.



Noam Chomsky
(geb. 1928, hier um 1960)

Zwischenfazit zu kontextfreien Grammatiken

- Es kann in polynomialer Laufzeit entschieden werden, ob zu einer kontextfreien Grammatik G die Sprache $L(G)$ leer bzw. endlich ist.
⇒ Entfernung nutzloser Variablen (vgl. VL 15)
- Das Wortproblem für kontextfreie Grammatiken ist in polynomialer Laufzeit entscheidbar.
⇒ Chomsky-Normalform und CYK-Algorithmus (vgl. VL 14)
- Für kontextfreie Grammatiken G , G_1 und G_2 sind auch die Sprachen $L(G)^*$, $L(G_1) \cup L(G_2)$ und $L(G_1) \cdot L(G_2)$ kontextfrei.
⇒ (vgl. VL 15)
- Kontextfreie Sprachen sind genau die Sprachen, die von nichtdet. Kellerautomaten (NPDAs) akzeptiert werden.
⇒ Tripelkonstruktion (vgl. VL 16 & heute)

Satz.

Das Problem für kontextfreie Grammatiken G_1 und G_2 zu entscheiden, ob $L(G_1) \cap L(G_2) = \emptyset$ ist, ist nicht entscheidbar.

Unentscheidbare Probleme für kontextfreie Grammatiken

Satz.

Das Problem für kontextfreie Grammatiken G_1 und G_2 zu entscheiden, ob $L(G_1) \cap L(G_2) = \emptyset$ ist, ist nicht entscheidbar.

Beweisskizze:

- Wir beweisen, dass aus der Entscheidbarkeit von $L(G_1) \cap L(G_2) \neq \emptyset$ die Entscheidbarkeit des Post'schen Korrespondenzproblems (PKP) folgt.
- Dies ist ein Widerspruch zur Nichtentscheidbarkeit des PKP.
- Wir geben für jede PKP-Instanz K kontextfreie Grammatiken G_1 und G_2 an, so dass es ein Wort $w \in L(G_1) \cap L(G_2)$ genau dann gibt, wenn es eine Lösung für K gibt.

Das Post'sche Korrespondenzproblem

Post'sches Korrespondenzproblem.

Gegeben ist endliche Folge von Wortpaaren

$$K = ((x_1, y_1), \dots, (x_k, y_k))$$

über einem endlichen Alphabet Σ . Es gilt $x_i \in \Sigma^+$ und $y_i \in \Sigma^+$. Gefragt ist, ob es eine endliche Folge von Indizes $i_1, \dots, i_\ell \in \{1, \dots, k\}$ gibt, so dass $x_{i_1} \cdots x_{i_\ell} = y_{i_1} \cdots y_{i_\ell}$ gilt.

Beweis

- Gegeben sei PKP-Instanz $K = ((x_1, y_1), \dots, (x_k, y_k))$ über Alphabet Σ .
- Es sei $\Sigma' = \Sigma \cup \{a_1, \dots, a_k\}$ für neue Symbole a_1, \dots, a_k .
- Es sei $G_1 = (\Sigma', V_1 = \{S_1\}, S_1, R_1)$ mit Regeln

$$S_1 \rightarrow a_i x_i \text{ und } S_1 \rightarrow a_i S_1 x_i \text{ für alle } 1 \leq i \leq k;$$
- Es sei $G_2 = (\Sigma', V_2 = \{S_2\}, S_2, R_2)$ mit Regeln

$$S_2 \rightarrow a_i y_i \text{ und } S_2 \rightarrow a_i S_2 y_i \text{ für alle } 1 \leq i \leq k.$$

Dann gilt

$$L(G_1) = \{a_{i_1} \cdots a_{i_1} x_{i_1} \cdots x_{i_1} \mid n \in \mathbb{N}, 1 \leq i_j \leq k\}$$

$$L(G_2) = \{a_{i_1} \cdots a_{i_1} y_{i_1} \cdots y_{i_1} \mid n \in \mathbb{N}, 1 \leq i_j \leq k\}.$$

Beweis

$$K = ((x_1, y_1), \dots, (x_k, y_k))$$

$$L(G_1) = \{a_{i_1} \cdots a_{i_n} x_{i_1} \cdots x_{i_n} \mid n \in \mathbb{N}, 1 \leq i_j \leq k\}$$

$$L(G_2) = \{a_{i_1} \cdots a_{i_n} y_{i_1} \cdots y_{i_n} \mid n \in \mathbb{N}, 1 \leq i_j \leq k\}.$$

Es folgt

$$K \text{ hat Lösung} \Leftrightarrow \exists i_1, \dots, i_n \text{ mit } x_{i_1} \cdots x_{i_n} = y_{i_1} \cdots y_{i_n}$$

$$\Leftrightarrow \exists i_1, \dots, i_n \text{ mit } a_{i_1} \cdots a_{i_n} x_{i_1} \cdots x_{i_n} = a_{i_1} \cdots a_{i_n} y_{i_1} \cdots y_{i_n}$$

$$\Leftrightarrow \exists w \in L(G_1) \cap L(G_2)$$

$$\Leftrightarrow L(G_1) \cap L(G_2) \neq \emptyset$$

Eindeutigkeit von kontextfreien Grammatiken

Definition.

Eine Grammatik G ist **eindeutig**, wenn es für jedes $w \in L(G)$ genau einen Syntaxbaum gibt.

Satz.

Das Problem, für eine kontextfreie Grammatik G zu entscheiden, ob sie eindeutig ist, ist nicht entscheidbar.

G ist nicht eindeutig \iff es gibt $w \in L(G)$ mit mindestens zwei verschiedenen Syntaxbäumen

Eindeutigkeit von kontextfreien Grammatiken

Definition.

Eine Grammatik G ist **eindeutig**, wenn es für jedes $w \in L(G)$ genau einen Syntaxbaum gibt.

Satz.

Das Problem, für eine kontextfreie Grammatik G zu entscheiden, ob sie eindeutig ist, ist nicht entscheidbar.

G ist nicht eindeutig \iff es gibt $w \in L(G)$ mit mindestens zwei verschiedenen Syntaxbäumen

Beweisskizze.

- Annahme: Es sei entscheidbar, ob eine kontextfreie Grammatik nicht eindeutig ist.
- Dann könnten wir das PKP entscheiden.
- Dies ist ein Widerspruch.

Beweis

- Gegeben sei PKP-Instanz $K = ((x_1, y_1), \dots, (x_k, y_k))$ über Alphabet Σ .
- Seien $G_1 = (\Sigma', V_1, S_1, R_1)$ und $G_2 = (\Sigma', V_2, S_2, R_2)$ wie im letzten Beweis.
- Wir konstruieren eine neue Grammatik $G = (\Sigma', V, S, R)$, die genau dann nicht eindeutig ist, wenn $L(G_1) \cap L(G_2) \neq \emptyset$:

$$V = V_1 \cup V_2 \cup \{S\} \text{ wobei } S \text{ neues Startsymbol}$$

$$R = R_1 \cup R_2 \cup \{S \rightarrow S_1 | S_2\}$$
- Da G_1 und G_2 eindeutig sind, existiert $w \in L(G_1) \cap L(G_2)$ genau dann, wenn es in G Ableitungen $S \rightarrow S_1 \xrightarrow{*} w$ und $S \rightarrow S_2 \xrightarrow{*} w$ gibt, also G mehrdeutig ist.

Sprache der korrekten Rechenwege

- Sei $\mathcal{M} = (Q, \Sigma, \Gamma, \sqcup, q_0, \delta, F)$ eine TM.
- Eine Berechnung von \mathcal{M} kann durch die Folge der durchlaufenen **Konfigurationen** $\alpha(q)\beta$ mit $\alpha, \beta \in \Gamma^*$ und $q \in Q$ beschrieben werden.
- $\alpha(q)\beta$ bedeutet, dass
 - auf dem Band das Wort $\alpha\beta$, umgeben von Blanksymbolen, steht,
 - die Turingmaschine im Zustand q ist
 - und der Lese-/Schreibkopf auf die Stelle des Bandes, an der das erste Symbol von β steht, zeigt.
- Wenn w_1, w_2, \dots, w_n die Abfolge der Konfigurationen einer Berechnung von \mathcal{M} ist, so kann dieser Rechenweg durch das Wort $w_1\#w_2\#\dots\#w_n\#$, mit $\# \notin \Gamma$ Trennsymbol, kodiert werden.
- Hierbei sei (q) für $q \in Q$ ein einziges Zeichen.

Sprache der korrekten Rechenwege

- Sei $\mathcal{M} = (Q, \Sigma, \Gamma, \sqcup, q_0, \delta, F)$ eine TM.
- Wenn w_1, w_2, \dots, w_n die Abfolge der Konfigurationen einer Berechnung von \mathcal{M} ist, so kann dieser Rechenweg durch das Wort $w_1\#w_2\#\dots\#w_n\#$, mit $\# \notin \Gamma$ Trennsymbol, kodiert werden.
- Hierbei sei (q) für $q \in Q$ ein einziges Zeichen.
- Allerdings lässt sich die Sprache aller Wörter, die in dieser Weise die korrekten Rechenwege einer TM kodieren, nicht unbedingt durch kontextfreie Grammatiken beschreiben.
- Daher wird ein “Trick” angewendet und jede zweite Konfiguration gespiegelt kodiert.

Sprache der korrekten Rechenwege

Definition.

Die **Sprache $B_{\mathcal{M}}$ der korrekten Rechenwege einer TM \mathcal{M}** besteht aus allen Worten

$$w_1 \# w_2^R \# w_3 \# w_4^R \cdots w_n^R \#, \text{ falls } n \text{ gerade und}$$

$$w_1 \# w_2^R \# w_3 \# w_4^R \cdots w_n \#, \text{ falls } n \text{ ungerade,}$$

wobei

- die w_i , $1 \leq i \leq n$, Konfigurationen von \mathcal{M} sind,
- w_1 eine Anfangskonfiguration,
- w_n eine akzeptierende Konfiguration und
- für alle $1 \leq i \leq n - 1$ die Konfiguration w_{i+1} die direkte Nachfolgekonfiguration von w_i bei einer korrekten Berechnung von \mathcal{M}

ist.

Sprache der korrekten Rechenwege

Lemma.

Für alle Turingmaschinen \mathcal{M} ist $B_{\mathcal{M}}$ der Durchschnitt zweier Sprachen

- $L_1 = L(G_1)$
- $L_2 = L(G_2)$,

wobei G_1 und G_2 kontextfreie Grammatiken sind.

Beweis

Sei $\mathcal{M} = (Q, \Sigma, \Gamma, \sqcup, q_0, \delta, F)$ eine Turing-Maschine.

Wir konstruieren L_1 und L_2 aus den Sprachen

$$L := \{u\#v^R \mid v \text{ ist direkte Nachfolgekonfiguration von } u \text{ für } \mathcal{M}\}$$

$$L' := \{v^R\#u \mid u \text{ ist direkte Nachfolgekonfiguration von } v \text{ für } \mathcal{M}\}$$

$$E := \{\varepsilon\} \cup (\Gamma^* \cdot \{(q) \mid q \in F\} \cdot \Gamma^* \cdot \{\#\})$$

Falls L und L' kontextfrei sind, so sind auch

$$L_1 := (L \cdot \{\#\})^* \cdot E$$

$$L_2 := \{(q_0)\} \cdot \Sigma^* \cdot \{\#\} \cdot (L' \cdot \{\#\})^* \cdot E$$

kontextfrei.

Beweis

$$L := \{u\#v^R \mid v \text{ ist direkte Nachfolgekonfiguration von } u \text{ für } \mathcal{M}\}$$

$$L' := \{v^R\#u \mid u \text{ ist direkte Nachfolgekonfiguration von } v \text{ für } \mathcal{M}\}$$

$$E := \{\varepsilon\} \cup (\Gamma^* \cdot \{(q) \mid q \in F\} \cdot \Gamma^* \cdot \{\#\})$$

$$L_1 := (L \cdot \{\#\})^* \cdot E$$

$$L_2 := \{(q_0)\} \cdot \Sigma^* \cdot \{\#\} \cdot (L' \cdot \{\#\})^* \cdot E$$

Offensichtlich haben alle Wörter aus L_1 die Form

$$w_1\#w_2^R\#\cdots\#w_{2i-1}\#w_{2i}^R\# \text{ oder}$$

$$w_1\#w_2^R\#\cdots\#w_{2i-1}\#w_{2i}^R\#w_{2i+1}\#$$

wobei für alle $1 \leq j \leq i$ gilt

- w_j ist Konfiguration von \mathcal{M}
- w_{2j} ist direkte Nachfolgekonfiguration von w_{2j-1}

und w_{2i+1} ist akzeptierende Konfiguration, falls vorhanden.

Beweis

Analog haben alle Wörter aus L_2 die Form

$$w_1 \# w_2^R \# \cdots w_{2i-1} \# w_{2i}^R \# \text{ oder} \\ w_1 \# w_2^R \# \cdots w_{2i-2}^R \# w_{2i-1} \#$$

wobei für alle $1 \leq j \leq i - 1$ gilt

- w_j ist Konfiguration von \mathcal{M} ,
- w_1 ist Anfangskonfiguration,
- w_{2j+1} ist direkte Nachfolgekongfiguration von w_{2j}

und w_{2i} ist akzeptierende Konfiguration, falls vorhanden.

Dann ist $B_{\mathcal{M}} = L_1 \cap L_2$.

Beweis

$$L := \{u\#v^R \mid v \text{ ist direkte Nachfolgekonfiguration von } u \text{ für } \mathcal{M}\}$$

Wir geben nun eine kontextfreie Grammatik G für L an mit Startvariable S und zusätzlicher Variable A .

G enthalte folgende Regeln:

- (i) alle Regeln $S \rightarrow aSa$, $a \in \Gamma \setminus \{\sqcup\}$;
- (ii) für alle Übergänge $\delta(q, a) = (q', b, R)$ von \mathcal{M} die Regeln $S \rightarrow (q)aA(q')b$;
- (iii) für alle Übergänge $\delta(q, a) = (q', b, L)$ von \mathcal{M} und alle $x \in \Gamma \cup \{\varepsilon\}$ die Regeln $S \rightarrow x(q)aAbx(q')$;
- (iv) für alle Übergänge $\delta(q, a) = (q', b, N)$ von \mathcal{M} die Regeln $S \rightarrow (q)aAb(q')$;
- (v) für alle $a \in \Gamma$ die Regeln $A \rightarrow aAa$;
- (vi) die Regel $A \rightarrow \#$.

Beweis

- Analog kann eine kontextfreie Grammatik G' für L' angegeben werden.
- Es ist leicht zu zeigen, dass $L(G) = L$ und $L(G') = L'$ ist.
- Damit ist die Behauptung bewiesen.

Beweis

- Analog kann eine kontextfreie Grammatik G' für L' angegeben werden.
- Es ist leicht zu zeigen, dass $L(G) = L$ und $L(G') = L'$ ist.
- Damit ist die Behauptung bewiesen.

Lemma.

Für alle Turingmaschinen \mathcal{M} ist $B_{\mathcal{M}}$ der Durchschnitt zweier kontextfreier Sprachen.

Bemerkung.

Falls \mathcal{M} in jeder Berechnung nur höchstens einen Rechenschritt ausführt, ist $B_{\mathcal{M}}$ sogar selbst kontextfrei.

Weiterhin gilt ...

Lemma.

Für alle Turingmaschinen \mathcal{M} ist $B_{\mathcal{M}}$ der Durchschnitt zweier kontextfreier Sprachen.

Bemerkung.

Falls \mathcal{M} in jeder Berechnung nur höchstens einen Rechenschritt ausführt, ist $B_{\mathcal{M}}$ sogar selbst kontextfrei.

Lemma.

Sei \mathcal{M} eine TM, die auf jeder Eingabe mindestens zwei Rechenschritte ausführt. Dann ist die Sprache $B_{\mathcal{M}}$ genau dann kontextfrei, wenn $L(\mathcal{M})$ endlich ist.

Ohne Beweis.

Zusammenfassung Chomsky-Hierarchie

Testen Sie sich:

- ▶ Können Sie folgende Tabelle ausfüllen?
- ▶ Welche Ergebnisse sind aus der heutigen Vorlesung?

Chomsky- Typ	Regeln in der Grammatik	Komplexität des Wort- problems	zugehöriges Maschinen- modell	Beispiel- sprache
Typ 0				
Typ 1				
Typ 2				
Typ 3				