Theoretische Grundlagen der Informatik

Vorlesung vom 23. Januar 2018

UT FÜR THEORETISCHE INFORMATIK

Übersicht Chomsky-2

Sei L eine Sprache mit $\varepsilon \notin L$.

$\exists$ PDA der L durch akzeptierenden Endzustand akzeptiert

Satz 5.25

Satz 5.26

$\exists$ PDA der L durch leeren Stack akzeptiert

Satz 5.27

$\exists$ Grammatik in Greibach-Normalform für L

nach Definition

Satz 5.23

$\exists$ Chomsky-2-Grammatik für L

Satz 5.28

- Heute beweisen wir Satz 5.27 und Satz 5.28.

Wdh.: Greibach-Normalform, Kellerautomat

Eine kontextfreie Grammatik ist in **Greibach-Normalform**, wenn alle Ableitungsregeln von der Form $A \to a\alpha$ mit $A \in V$, $a \in \Sigma$, $\alpha \in V^*$ sind.

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

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

Wdh.: Greibach-Normalform, Kellerautomat

Eine **Konfiguration eines PDA** 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.

Ein PDA **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.

Satz:
Für eine Grammatik G in Greibach-Normalform kann ein PDA konstruiert werden, der $L(G)$ durch leeren Stack akzeptiert.

Beweis

- Sei $G = (\Sigma, V, S, R)$ eine Grammatik in Greibach-Normalform
- Konstruiere gewünschten Kellerautomaten $\mathcal{A} = (Q, \Sigma, \Gamma, \delta, q_0, Z_0)$

$$
\begin{aligned}
Q &:= \{q_0\} \\
\Gamma &:= V \\
Z_0 &:= S \\
\delta(q_0, a, A) &:= \{(q_0, \alpha) \mid (A \to a\alpha) \in R\}
\end{aligned}
$$

Per Induktion über die Länge i einer Ableitung beweisen wir:

- $S \xrightarrow{*} w_1 \dots w_i A_1 \dots A_m \quad \Leftrightarrow \quad \mathcal{A}$ kann beim Lesen von $w_1 \dots w_i$ den STACK-Inhalt $A_1 \dots A_m$ erzeugen. Möglicherweise ist $A_1 \dots A_m = \varepsilon$.

Daraus folgt:

- $\mathcal{A}$ erkennt $w_1 \dots w_n$ durch leeren Stack $\Leftrightarrow S \xrightarrow{*} w_1 \dots w_n$ in G

Beweis

$$Q := \{q_0\} \quad \Gamma := V \quad Z_0 := S$$
$$\delta(q_0, a, A) := \{(q_0, \alpha) \mid (A \to a\alpha) \in R\}$$

Induktionsanfang ist mit $i = 0$ trivialerweise erfüllt.
Induktionsschritt:
Sei $i \geq 1$ und „$\xrightarrow{j}$“ stehe für eine Ableitung der Länge j. Dann gilt

$$S \xrightarrow{i} w_1 \dots w_i A_1 \dots A_m \iff \begin{array}{l} \exists A' \in V, r \in \{1, \dots, m\} \text{ mit} \\ S \xrightarrow{i-1} w_1 \dots w_{i-1} A' A_r \dots A_m \\ \to w_1 \dots w_i A_1 \dots A_m. \end{array}$$

Mit Induktionsvoraussetzung ist dies äquivalent zu

$\exists A' \in V, r \in \{1, \dots, m\}$ so, dass

- $\mathcal{A}$ das Wort $w_1 \dots w_{i-1}$ lesen und dabei STACK-Inhalt $A' A_r \dots A_m$ erzeugen kann, und
- $A' \to w_i A_1 \dots A_{r-1}$ Regel von G ist.

Beweis

$$\begin{aligned} Q &:= \{q_0\} \quad \Gamma := V \quad Z_0 := S \\ \delta(q_0, a, A) &:= \{(q_0, \alpha) \mid (A \to a\alpha) \in R\} \end{aligned}$$

Induktionsanfang ist mit $i = 0$ trivialerweise erfüllt.
Induktionsschritt:
Sei $i \geq 1$ und „$\xrightarrow{j}$“ stehe für eine Ableitung der Länge j.
Mit Induktionsvoraussetzung ist dies äquivalent zu

$\exists A' \in V, r \in \{1, \ldots, m\}$ so, dass

- $\mathcal{A}$ das Wort $w_1 \ldots w_{i-1}$ lesen und dabei STACK-Inhalt $A' A_r \ldots A_m$ erzeugen kann, und
- $A' \to w_i A_1 \ldots A_{r-1}$ Regel von G ist.

Dies ist genau dann erfüllt, wenn $\mathcal{A}$ das Wort $w_1 \ldots w_i$ lesen und dabei den STACK-Inhalt $A_1 \ldots A_m$ erzeugen kann.

Satz:
Jede durch einen PDA (durch leeren Stack oder durch akzeptierende Endzustände) akzeptierte Sprache ist kontextfrei.

Beweis

- Sei $\mathcal{A} = (Q, \Sigma, \Gamma, \delta, q_0, Z_0)$ PDA, 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

- Sei $\mathcal{A} = (Q, \Sigma, \Gamma, \delta, q_0, Z_0)$ PDA, 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

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

Beweis

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

$$(q, w, \alpha) \overset{*}{\vdash} (p, w', \beta)$$

beziehungsweise

$$(q, w, \alpha) \overset{k}{\vdash} (p, w', \beta)$$

für eine Folge von genau k Konfigurationen.

Beweis

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

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

Aus dieser Behauptung folgt dann

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

Beweis

Richtung

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

Beschreibung:

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

Beweis

- $V := \{[q, X, p] \mid p, q \in Q,\ X \in \Gamma\} \cup \{S\}$.
- $S \to [q_0, Z_0, q]$ für alle $q \in Q$
- $[q, X, q_{m+1}] \to a[q_1, Y_1, q_2][q_2, Y_2, q_3] \ldots [q_m, Y_m, q_{m+1}]$
  für alle Möglichkeiten $q_2, q_3, \ldots, q_{m+1} \in Q$,
  falls $(q_1, Y_1 Y_2 \ldots Y_m) \in \delta(q, a, X)$.

Induktionsanfang:

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

Beweis

- $V := \{[q, X, p] \mid p, q \in Q,\ X \in \Gamma\} \cup \{S\}$.
- $S \to [q_0, Z_0, q]$ für alle $q \in Q$
- $[q, X, q_{m+1}] \to a[q_1, Y_1, q_2][q_2, Y_2, q_3] \dots [q_m, Y_m, q_{m+1}]$
  für alle Möglichkeiten $q_2, q_3, \dots, q_{m+1} \in Q$,
  falls $(q_1, Y_1 Y_2 \dots Y_m) \in \delta(q, a, X)$.

Induktionsschritt:

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

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

wobei $q_{m+1} = p$ und $w = aw_1 \dots w_m$, mit $w_i \in \Sigma^*$, $a \in \Sigma$ und $[q_j, Y_j, q_{j+1}] \xrightarrow{k'} w_j$ mit $k' \leq k - 1$ für alle $1 \leq j \leq m$.

Beweis

- Betrachte eine Ableitung $[q, X, p] \overset{k}{\rightarrow} w$.
- Schreibe diese als
$$[q, X, p] \rightarrow a[q_1, Y_1, q_2][q_2, Y_2, q_3] \dots [q_m, Y_m, q_{m+1}] \overset{k-1}{\rightarrow} w,$$
wobei $q_{m+1} = p$ und $w = aw_1 \dots w_m$, mit $w_i \in \Sigma^*$, $a \in \Sigma$ und $[q_j, Y_j, q_{j+1}] \overset{k'}{\rightarrow} w_j$ mit $k' \leq k - 1$ für alle $1 \leq j \leq m$.

Beweis

- Betrachte eine Ableitung $[q, X, p] \xrightarrow{k} w$.
- Schreibe diese als
$$[q, X, p] \to a[q_1, Y_1, q_2][q_2, Y_2, q_3] \dots [q_m, Y_m, q_{m+1}] \xrightarrow{k-1} w,$$
wobei $q_{m+1} = p$ und $w = aw_1 \dots w_m$, mit $w_i \in \Sigma^*$, $a \in \Sigma$ und $[q_j, Y_j, q_{j+1}] \xrightarrow{k'} w_j$ mit $k' \leq k - 1$ für alle $1 \leq j \leq m$.
- Induktionsvoraussetzung: $(q_j, w_j, Y_j) \overset{*}{\vdash} (q_{j+1}, \varepsilon, \varepsilon)$ für alle $1 \leq j \leq m$.
- Also $(q_j, w_j, Y_j \dots Y_m) \overset{*}{\vdash} (q_{j+1}, \varepsilon, Y_{j+1} \dots Y_m)$ für alle $1 \leq j \leq m$.
- Damit
$$\begin{aligned}(q, w, X) &\vdash (q_1, w_1 \dots w_m, Y_1 \dots Y_m)\\ &\overset{*}{\vdash} (q_2, w_2 \dots w_m, Y_2 \dots Y_m)\\ &\overset{*}{\vdash} (q_3, w_3 \dots w_m, Y_3 \dots Y_m)\\ &\overset{*}{\vdash} \dots \overset{*}{\vdash} (q_m, w_m, Y_m) \overset{*}{\vdash} (q_{m+1}, \varepsilon, \varepsilon) = (p, \varepsilon, \varepsilon)\end{aligned}$$

Beweis

Richtung

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

Beschreibung:

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

Beweis

- $V := \{[q, X, p] \mid p, q \in Q,\ X \in \Gamma\} \cup \{S\}$.
- $S \to [q_0, Z_0, q]$ für alle $q \in Q$
- $[q, X, q_{m+1}] \to a[q_1, Y_1, q_2][q_2, Y_2, q_3] \dots [q_m, Y_m, q_{m+1}]$
  für alle Möglichkeiten $q_2, q_3, \dots, q_{m+1} \in Q$,
  falls $(q_1, Y_1 Y_2 \dots Y_m) \in \delta(q, a, X)$.

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 $[q, X, p] \to w$ eine Regel von G.

Beweis

- $V := \{[q, X, p] \mid p, q \in Q,\ X \in \Gamma\} \cup \{S\}$.
- $S \to [q_0, Z_0, q]$ für alle $q \in Q$
- $[q, X, q_{m+1}] \to a[q_1, Y_1, q_2][q_2, Y_2, q_3] \ldots [q_m, Y_m, q_{m+1}]$
  für alle Möglichkeiten $q_2, q_3, \ldots, q_{m+1} \in Q$,
  falls $(q_1, Y_1 Y_2 \ldots Y_m) \in \delta(q, a, X)$.

Induktionsschritt:

- Betrachte eine Abarbeitung $(q, w, X) \overset{k}{\vdash} (p, \varepsilon, \varepsilon)$
- 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 $(q_1, w', Y_1 \ldots Y_m)$ die Konfiguration von $\mathcal{A}$ nach dem 1. Schritt.
- Dann gilt

$$(q, aw', X) \vdash (q_1, w', Y_1 \ldots Y_m) \overset{k'}{\vdash} (p, \varepsilon, \varepsilon)$$

mit $k' = k - 1$.

Beweis

- $V := \{[q, X, p] \mid p, q \in Q,\ X \in \Gamma\} \cup \{S\}$.
- $S \to [q_0, Z_0, q]$ für alle $q \in Q$
- $[q, X, q_{m+1}] \to a[q_1, Y_1, q_2][q_2, Y_2, q_3] \dots [q_m, Y_m, q_{m+1}]$
  für alle Möglichkeiten $q_2, q_3, \dots, q_{m+1} \in Q$,
  falls $(q_1, Y_1 Y_2 \dots Y_m) \in \delta(q, a, X)$.

Sei

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

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

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

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

Beweis

- $V := \{[q, X, p] \mid p, q \in Q,\ X \in \Gamma\} \cup \{S\}$.
- $S \to [q_0, Z_0, q]$ für alle $q \in Q$
- $[q, X, q_{m+1}] \to a[q_1, Y_1, q_2][q_2, Y_2, q_3] \ldots [q_m, Y_m, q_{m+1}]$ für alle Möglichkeiten $q_2, q_3, \ldots, q_{m+1} \in Q$, falls $(q_1, Y_1 Y_2 \ldots Y_m) \in \delta(q, a, X)$.

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

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

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

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

Beweis

- $V := \{[q, X, p] \mid p, q \in Q,\ X \in \Gamma\} \cup \{S\}$.
- $S \to [q_0, Z_0, q]$ für alle $q \in Q$
- $[q, X, q_{m+1}] \to a[q_1, Y_1, q_2][q_2, Y_2, q_3] \ldots [q_m, Y_m, q_{m+1}]$
  für alle Möglichkeiten $q_2, q_3, \ldots, q_{m+1} \in Q$,
  falls $(q_1, Y_1 Y_2 \ldots Y_m) \in \delta(q, a, X)$.

Also gilt auch

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

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

$$[q, X, p] \to a[q_1, Y_1, q_2][q_2, Y_2, q_3] \ldots [q_m, Y_m, q_{m+1}] \overset{*}{\to} aw_1 \ldots w_m = w$$

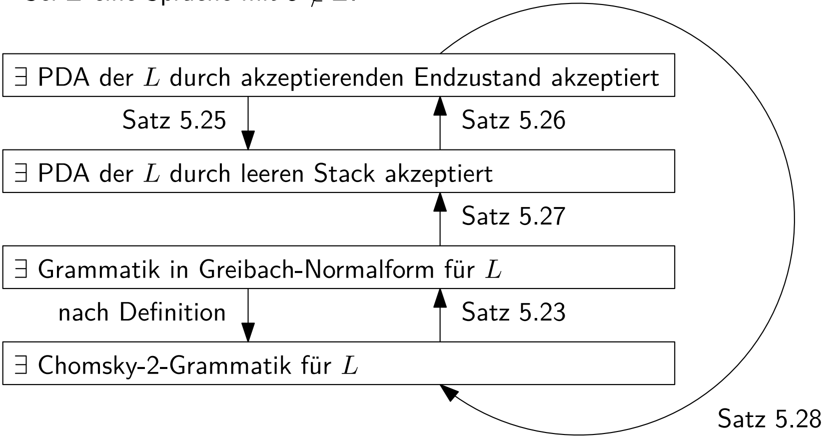
in G existiert.

Korollar

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

Übersicht Chomsky-2

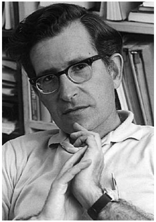
Sei L eine Sprache mit $\varepsilon \notin L$.



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

Karlsruher Institut für Technologie

- Es kann in polynomialer Laufzeit entschieden werden, ob zu einer kontextfreien Grammatik G die Sprache $L(G)$ leer bzw. endlich ist.

  $\Rightarrow$ Entfernung nutzloser Variablen (vgl. VL 15, Folie 23)

- Das Wortproblem für kontextfreie Grammatiken ist in polynomialer Laufzeit entscheidbar.

  $\Rightarrow$ Chomsky-Normalform und CYK-Algorithmus

- 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.

  $\Rightarrow$ (vgl. VL 15, Folie 25)

- Die kontextfreien Sprachen sind genau die Sprachen, die von nichtdeterministischen Kellerautomaten (NPDAs) akzeptiert werden.

  $\Rightarrow$ Greibach-Normalform (vgl. heute, Folie 17)

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.

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 $L(G_1) \cap L(G_2) = \emptyset$ 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

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

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 $i_1, \ldots, i_\ell \in \{1, \ldots, k\}$ gibt, so dass $x_{i_1} \ldots x_{i_\ell} = y_{i_1} \ldots y_{i_\ell}$ gilt.

Beweis

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

$$S_1 \to a_i x_i \text{ und } S_1 \to a_i S_1 x_i \text{ für alle } 1 \leq i \leq k;$$

- Es sei $G_2 = (\Sigma', V_2 = \{S_2\}, S_2, R_2)$ mit Regeln

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

Dann gilt

$$\begin{aligned} L(G_1) &= \{a_{i_n} \cdots a_{i_1} x_{i_1} \cdots x_{i_n} \mid n \in \mathbb{N},\ 1 \leq i_j \leq k\} \\ L(G_2) &= \{a_{i_n} \cdots a_{i_1} y_{i_1} \cdots y_{i_n} \mid n \in \mathbb{N},\ 1 \leq i_j \leq k\}\ . \end{aligned}$$

Beweis

$$
\begin{aligned}
K &= ((x_1, y_1), \dots, (x_k, y_k)) \\
L(G_1) &= \{a_{i_n} \cdots a_{i_1} x_{i_1} \cdots x_{i_n} \mid n \in \mathbb{N},\ 1 \leq i_j \leq k\} \\
L(G_2) &= \{a_{i_n} \cdots a_{i_1} y_{i_1} \cdots y_{i_n} \mid n \in \mathbb{N},\ 1 \leq i_j \leq k\}\ .
\end{aligned}
$$

Es folgt

$$
\begin{aligned}
K \text{ hat Lösung} &\Leftrightarrow \exists i_1, \dots, i_n \text{ mit } x_{i_1} \cdots x_{i_n} = y_{i_1} \cdots y_{i_n} \\
&\Leftrightarrow \exists i_1, \dots, i_n \text{ mit } a_{i_n} \cdots a_{i_1} x_{i_1} \cdots x_{i_n} = a_{i_n} \cdots a_{i_1} y_{i_1} \cdots y_{i_n} \\
&\Leftrightarrow \exists w \in L(G_1) \cap L(G_2) \\
&\Leftrightarrow L(G_1) \cap L(G_2) \neq \emptyset
\end{aligned}
$$

Eindeutigkeit von kontextfreien Grammatiken

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.

Beweisskizze

- Annahme: Es sei entscheidbar, ob eine kontextfreie Grammatik 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 mehrdeutig ist, wenn $L(G_1) \cap L(G_2) \neq \emptyset$:

$$\begin{aligned} V &= V_1 \cup V_2 \cup \{S\} \text{ wobei } S \text{ neues Startsymbol} \\ R &= R_1 \cup R_2 \cup \{S \to S_1 | S_2\} \end{aligned}$$

- 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 $S \to S_1 \xrightarrow{*} w$ und $S \to S_2 \xrightarrow{*} w$ 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 $w_1, w_2, \ldots, w_n$ die Abfolge der Konfigurationen einer Berechnung von $\mathcal{M}$ ist, so kann dieser Rechenweg durch das Wort $w_1 \# w_2 \# \ldots \# w_n \#$, mit $\# \notin \Gamma$ Trennsymbol, kodiert werden.

Sprache der korrekten Rechenwege

- 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

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

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

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

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.

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

Wir konstruieren L_1 und L_2 aus den Sprachen

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

Falls L und L' kontextfrei sind, so sind auch

$$\begin{aligned} L_1 &:= (L\{\#\})^*(\{\varepsilon\} \cup \Gamma^* F \Gamma^* \{\#\}) \\ L_2 &:= \{q_0\}\Sigma^*\{\#\}(L'\{\#\})^*(\{\varepsilon\} \cup \Gamma^* F \Gamma^* \{\#\}) \end{aligned}$$

kontextfrei, wobei

- Γ Bandalphabet,
- Σ Eingabealphabet,
- q_0 Anfangszustand und
- F Endzustandsmenge

von $\mathcal{M}$.

Beweis

Offensichtlich haben alle Wörter aus L_1 die Form

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

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

mit

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

für alle $1 \leq j \leq i$ und w_{2i+1} akzeptierende Konfiguration, falls vorhanden.

Beweis

Analog haben alle Wörter aus L_2 die Form

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

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

mit

- w_j Konfiguration von $\mathcal{M}$
- w_1 Anfangskonfiguration
- w_{2j+1} direkte Nachfolgekonfiguration von w_{2j}

für alle $1 \leq j \leq i-1$ und w_{2i} 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 $S \to aSa$, $a \in \Gamma \backslash \{\sqcup\}$;

(ii) für alle Übergänge $\delta(q, a) = (q', b, R)$ von $\mathcal{M}$ die Regeln $S \to qaAq'b$;

(iii) für alle Übergänge $\delta(q, a) = (q', b, L)$ von $\mathcal{M}$ die Regeln $S \to xqaAbxq'$, wobei x Symbol links von a beim Lesen von a im Zustand q;

(iv) für alle Übergänge $\delta(q, a) = (q', b, N)$ von $\mathcal{M}$ die Regeln $S \to qaAbq'$;

(v) für alle $a \in \Gamma$ die Regeln $A \to aAa$;

(vi) die Regel $A \to \#$.

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.

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.

Beweis

$B_{\mathcal{M}}$ ist kontextfrei $\Leftarrow L(\mathcal{M})$ endlich ist

- Sei $L(\mathcal{M})$ endlich
- Zu jeder Eingabe aus $L(\mathcal{M})$ gibt es genau eine akzeptierende Berechnung.
- Damit ist $B_{\mathcal{M}}$ auch endlich.
- Jede endliche Sprache ist regulär, also auch kontextfrei.

Beweis

$B_{\mathcal{M}}$ ist kontextfrei $\Rightarrow L(\mathcal{M})$ endlich ist

- Angenommen $L(\mathcal{M})$ sei unendlich und $B_{\mathcal{M}}$ wäre kontextfrei.
- Da $L(\mathcal{M})$ unendlich ist, gibt es zu der Konstanten n aus Ogden's Lemma ein $\tilde{w} \in B_{\mathcal{M}}$ mit $\tilde{w} = w_1 \# w_2^R \# \dots$ und $|w_2^R| \geq n$.
- Wenn alle Symbole aus $\# w_2^R \#$ markiert werden, muss es eine Zerlegung $uvwxy$ von $\tilde{w}$ geben, sodass vx mindestens einen und vwx höchstens n markierte Buchstaben enthält und $uv^i wx^i y \in B_{\mathcal{M}}$ für alle $i \geq 0$.
- Da $\mathcal{M}$ mindestens zwei Berechnungsschritte ausführt, existieren die Konfigurationen w_1, w_2 und w_3.
- Entsprechend der Zerlegung von $\tilde{w}$ enthalten $\# w_2^R \#$ und vx mindestens einen gemeinsamen Buchstaben, und nur eines der Worte w_1 und w_3 hat ebenfalls gemeinsame Buchstaben mit vx.

Beweis

$B_{\mathcal{M}}$ ist kontextfrei $\Rightarrow L(\mathcal{M})$ endlich ist

- Da $\mathcal{M}$ mindestens zwei Berechnungsschritte ausführt, existieren die Konfigurationen w_1, w_2 und w_3.
- Entsprechend der Zerlegung von $\tilde{w}$ enthalten $\#w_2^R\#$ und vx mindestens einen gemeinsamen Buchstaben, und nur eines der Worte w_1 und w_3 hat ebenfalls gemeinsame Buchstaben mit vx.
- Wenn w_1 keinen gemeinsamen Buchstaben mit vx hat, ist $uv^2wx^2y \notin B_{\mathcal{M}}$, da die Berechnung für die Anfangskonfiguration w_1 eindeutig ist.
- Aus demselben Grund ist $uv^2wx^2y \notin B_{\mathcal{M}}$, falls $w_1\#$ Präfix von uv ist.
- Falls v ein Teilwort von w_1 wäre, müsste x ein Teilwort von w_2^R sein, damit für großes i das Wort $uv^iwx^iy \in B_{\mathcal{M}}$ ist, da zwei aufeinander folgende Konfigurationen etwa gleich lang sind.
- Dann wäre aber w_3 als Nachfolgekonfiguration zu kurz, uv^iwx^iy also keine Kodierung eines korrekten Rechenweges von $\mathcal{M}$.
- Dies ist ein Widerspruch.