Inhaltsangabe

§ 10. Kellerautomaten

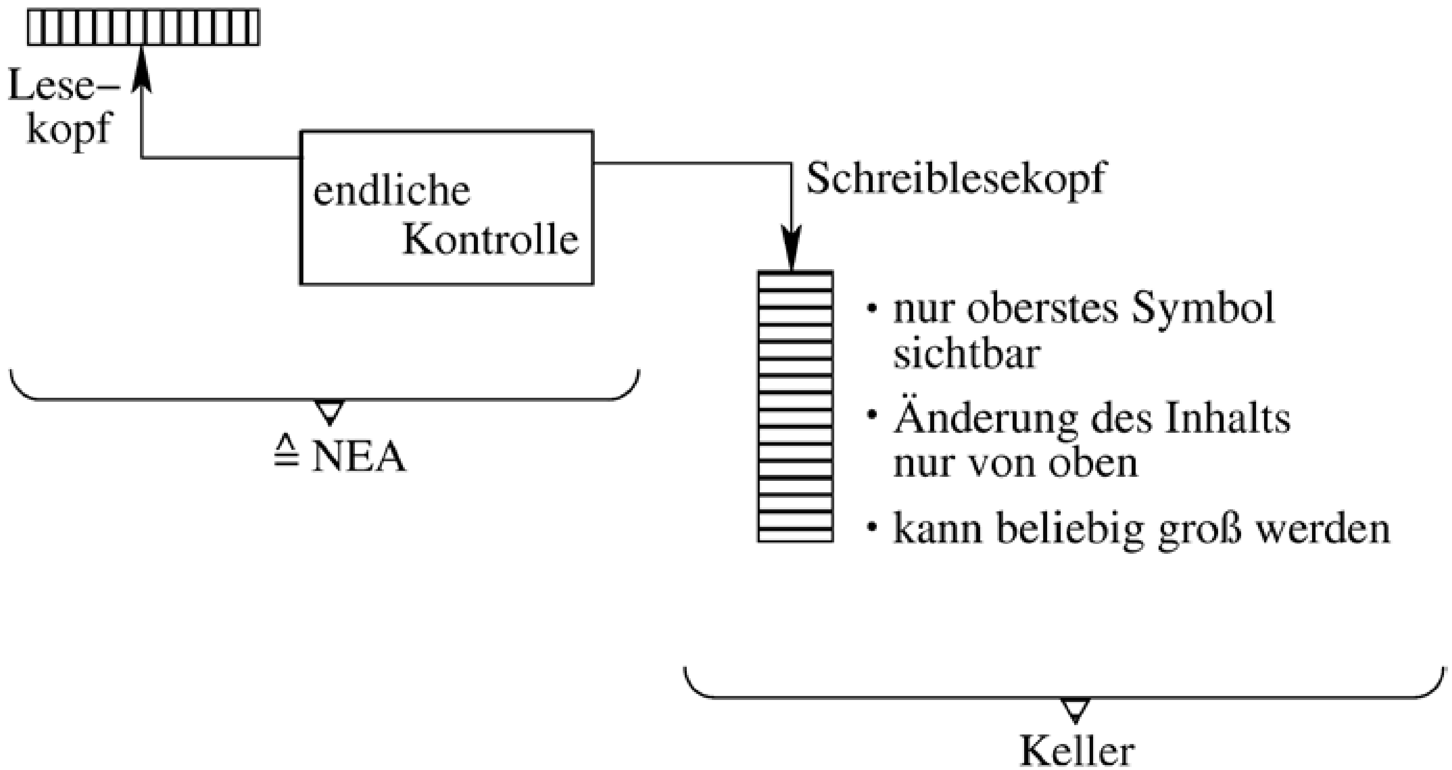
Charakterisierung kontextfreier Sprachen: bisher mit Hilfe von Grammatiken!

Automaten?

Wir haben gesehen, daß endliche Automaten nicht in der Lage sind, alle kontextfreien Sprachen zu akzeptieren.

Um die Beschreibung von kontextfreien Sprachen mit Hilfe von endlichen Automaten zu ermöglichen, muß man diese um eine (potentiell unendliche) Speicherkomponente, einen sogenannten Keller, erweitern.

Eingabe: von links nach rechts; nur ein Symbol sichtbar



Definition 10.1 (Kellerautomaten, PDA)

Ein Kellerautomat (pushdown automaton, kurz PDA) hat die Form $\mathcal{A} = (Q, \Sigma, \Gamma, q_0, Z_0, \Delta, F)$, wobei

- Q eine endliche Menge von Zuständen ist,
- Σ das Eingabealphabet ist,
- Γ das Kelleralphabet ist,
- $q_0 \in Q$ der Anfangszustand ist,
- $Z_0 \in \Gamma$ das Kellerstartsymbol ist,
- $\Delta \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \times \Gamma^* \times Q$ eine endliche Übergangsrelation ist,
- $F \subseteq Q$ eine Menge von Endzuständen ist.

Bedeutung der Übergangsrelation

$$\Delta \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \times \Gamma^* \times Q$$

Anschaulich bedeutet:

(q, a, Z, γ, q'):

Im Zustand q mit aktuellem Eingabesymbol a und oberstem Kellersymbol Z darf der Automat Z durch γ ersetzen und in den Zustand q' und zum nächsten Eingabesymbol übergehen.

$(q, \varepsilon, Z, \gamma, q')$:

wie oben, nur daß das aktuelle Eingabesymbol nicht relevant ist und man nicht zum nächsten Eingabesymbol übergeht (der Lesekopf ändert seine Position nicht).

Für die formale Definition der Bedeutung benötigen wir den Begriff der Konfiguration eines Kellerautomaten.

Den aktuellen Zustand einer Kellerautomatenberechnung (die aktuelle Konfiguration) kann man beschreiben durch

- den noch zu lesenden Rest $w \in \Sigma^*$ der Eingabe
  (Lesekopf steht auf dem ersten Symbol von w)
- den Zustand $q \in Q$
- den Kellerinhalt $\gamma \in \Gamma^*$
  (Schreiblesekopf steht auf dem ersten Symbol von γ)

Dresden

Definition 10.2

Eine Konfiguration von $\mathcal{A}$ hat die Form

$$\mathcal{K} = (q, w, \gamma) \in Q \times \Sigma^* \times \Gamma^*.$$

Die Übergangsrelation ermöglicht die folgenden Konfigurationsübergänge:

- $(q, aw, Z\gamma) \vdash (q', w, \beta\gamma)$ falls $(q, a, Z, \beta, q') \in \Delta$
- $(q, w, Z\gamma) \vdash (q', w, \beta\gamma)$ falls $(q, \varepsilon, Z, \beta, q') \in \Delta$
- $\mathcal{K} \vdash^* \mathcal{K}'$ gdw $\exists n \geq 0\ \exists \mathcal{K}_0, \ldots, \mathcal{K}_n$ mit

$$\mathcal{K} = \mathcal{K}_0 \vdash \mathcal{K}_1 \vdash \cdots \vdash \mathcal{K}_n = \mathcal{K}'.$$

Der Automat $\mathcal{A}$ akzeptiert das Wort $w \in \Sigma^*$ gdw

$(q_0, w, Z_0) \vdash^* (q, \varepsilon, \gamma)$ mit $q \in F, \gamma \in \Gamma^*$.

Eingabewort ganz gelesen und Endzustand erreicht

Die von $\mathcal{A}$ akzeptierte Sprache ist

$$L(\mathcal{A}) = \{w \in \Sigma^* \mid \mathcal{A} \text{ akzeptiert } w\}.$$

Dresden

Beispiel 10.3 (ein PDA für $\{a^n b^n \mid n \geq 1\}$)

$Q = \{q_0, q_1, f\}$, $\Gamma = \{Z, Z_0\}$, $\Sigma = \{a, b\}$, $F = \{f\}$ und Δ:

q_0	a	Z_0	ZZ_0	q_0	(erstes a, speichere Z)
q_0	a	Z	ZZ	q_0	(weitere a's, speichere Z)
q_0	b	Z	ε	q_1	(erstes b, entnimm Z)
q_1	b	Z	ε	q_1	(weitere b's, entnimm Z)
q_1	ε	Z_0	ε	f	(lösche das Kellerstartsymbol und gehe in Endzustand)

Einige Konfigurationsübergänge:

1. $(q_0, aabb, Z_0) \vdash (q_0, abb, ZZ_0) \vdash (q_0, bb, ZZZ_0) \vdash (q_1, b, ZZ_0) \vdash$
   $(q_1, \varepsilon, Z_0) \vdash (f, \varepsilon, \varepsilon)$ akzeptiert

2. $(q_0, aab, Z_0) \vdash^* (q_0, b, ZZZ_0) \vdash (q_1, \varepsilon, ZZ_0)$ kein Übergang möglich

3. $(q_0, abb, Z_0) \vdash (q_0, bb, ZZ_0) \vdash (q_1, b, Z_0) \vdash (f, b, \varepsilon)$ akzeptiert nicht

Dresden

Der Kellerautomat aus Beispiel 10.3 ist deterministisch, da es zu jeder Konfiguration höchstens eine Folgekonfiguration gibt.

Definition 10.4 (deterministischer Kellerautomat)

Der PDA $\mathcal{A} = (Q, \Sigma, \Gamma, q_0, Z_0, \Delta)$ heißt deterministisch, falls die folgenden beiden Bedingungen erfüllt sind:

1. $\forall q \in Q \ \forall a \in \Sigma \cup \{\varepsilon\} \ \forall Z \in \Gamma$ existiert höchstens ein Übergang der Form $(q, a, Z, \ldots, \ldots) \in \Delta$.
2. Existiert ein Tupel $(q, \varepsilon, Z, \ldots, \ldots) \in \Delta$, so existiert kein Tupel der Form $(q, a, Z, \ldots, \ldots) \in \Delta$ mit $a \in \Sigma$.

Dresden

Beispiel 10.5

Für $w = a_1 \ldots a_n$ gilt
$\overleftarrow{w} = a_n \ldots a_1$

Die Sprache

$$L = \{w \overleftarrow{w} \mid w \in \{a, b\}^*\}$$

wird von einem nichtdeterministischen PDA akzeptiert.

Idee:

Der PDA legt die erste Worthälfte im Keller ab und greift darauf in umgekehrter Reihenfolge beim Lesen der zweiten Worthälfte zurück.

So kann Nichtübereinstimmung festgestellt werden (kein Übergang bei Nichtübereinstimmung).

Nichtdeterminismus ist intuitiv nötig, da man „raten“ muß, wann die Wortmitte erreicht ist.

Wir werden später zeigen, daß kein deterministischer PDA diese Sprache akzeptiert.

Beispiel 10.5: $L = \{w \overleftarrow{w} \mid w \in \{a,b\}^*\}$

Diese Sprache wird von dem folgenden Kellerautomaten akzeptiert:

$Q = \{q_0, q_1, q_2, f\}$, $\Gamma = \{a, b, Z_0\}$, $\Sigma = \{a, b\}$, $F = \{f\}$ und

$\Delta =$

q_0	ε	Z_0	ε	f	akzeptiert ε
q_0	a/b	Z_0	a/bZ_0	q_1	Lesen
q_1	a	b	ab	q_1	und
q_1	b	a	ba	q_1	Speichern
q_1	a	a	aa	q_1	der ersten
q_1	b	b	bb	q_1	Worthälfte
q_1	b	b	ε	q_2	Lesen der
q_1	a	a	ε	q_2	zweiten Hälfte
q_2	b	b	ε	q_2	und Vergleichen
q_2	a	a	ε	q_2	mit erster
q_2	ε	Z_0	ε	f	Übergang zu Endzustand

Für Kellerautomaten gibt es noch einen anderen (äquivalenten) Akzeptanzbegriff:

Definition 10.6 (Akzeptanz mit leerem Keller)

Es sei $\mathcal{A} = (Q, \Sigma, \Gamma, q_0, Z_0, \Delta)$ ein PDA ohne ausgezeichnete Endzustandsmenge. Wir sagen:

$\mathcal{A}$ akzeptiert w mit leerem Keller, falls

$$(q_0, w, Z_0) \vdash^* (q, \varepsilon, \varepsilon) \text{ für ein beliebiges } q \in Q.$$

Die durch $\mathcal{A}$ mit leerem Keller akzeptierte Sprache ist

$$N(\mathcal{A}) := \{w \in \Sigma^* \mid \mathcal{A} \text{ akzeptiert } w \text{ mit leerem Keller}\}.$$

Beispiel:

Die PDAs aus Beispiel 10.3 und 10.5 akzeptieren die entsprechende Sprache sowohl mit Endzustandsmenge $\{f\}$ als auch mit leerem Keller:

nur beim Übergang zu f wird das Kellerstartsymbol Z_0 entfernt und damit der Keller leer.

Satz 10.7

Für PDAs ist Akzeptanz mit Endzuständen äquivalent zu Akzeptanz mit leerem Keller, d.h.

für eine formale Sprache L sind äquivalent:

1. $L = L(\mathcal{A})$ für einen PDA $\mathcal{A}$.

2. $L = N(\mathcal{B})$ für einen PDA $\mathcal{B}$.

Dresden

Beweis von Satz 10.7:

„1 $\longrightarrow$ 2“: $\mathcal{B}$ arbeitet im Prinzip wie $\mathcal{A}$.

Bei Erreichen eines Endzustands leert $\mathcal{B}$ noch den Keller.

Zusätzlich muß verhindert werden, daß der Keller leer wird, ohne daß ein Endzustand erreicht war:

$\mathcal{A}$ wird daher erweitert um

- einen neuen Anfangszustand ${q_0}'$,
- einen neuen Zustand q_1 zum Leeren des Kellers,
- ein neues Startsymbol X_0 (Z_0 ist nicht mehr Startsymbol),
- sowie die Übergänge

$$
\begin{array}{llllll}
({q_0}', & \varepsilon, & X_0, & Z_0X_0, & q_0), & \\
(q, & \varepsilon, & Z, & \varepsilon, & q_1) & \text{für } q \in F, Z \in \Gamma \cup \{X_0\}, \\
(q_1, & \varepsilon, & Z, & \varepsilon, & q_1) & \text{für } Z \in \Gamma \cup \{X_0\}.
\end{array}
$$

Beachte:

Da X_0 in den Übergängen von $\mathcal{A}$ nicht vorkommt, kann es nur entfernt werden, wenn vorher ein Endzustand erreicht war.

Beweis von Satz 10.7:

„2 $\longrightarrow$ 1": $\mathcal{A}$ arbeitet im Prinzip wie $\mathcal{B}$.

Zusätzlich muß $\mathcal{A}$ erkennen, wenn der Keller geleert ist und dann in einen Endzustand übergehen.

Erkennen des leeren Kellers (bei $\mathcal{B}$): verwende neues Kellerstartsymbol.

$\mathcal{B}$ wird daher ergänzt um

- einen neuen Anfangszustand ${q_0}'$,
- einen neuen Zustand f, der einziger Endzustand ist,
- ein neues Kellerstartsymbol X_0 (Z_0 ist nicht mehr Startsymbol),
- sowie die Übergänge

$$\begin{array}{llllll} ({q_0}', & \varepsilon, & X_0, & Z_0X_0, & q_0), & \\ (q, & \varepsilon, & X_0, & \varepsilon, & f) & \quad \text{für alle } q \in Q \end{array}$$

Wenn X_0 wieder auftaucht,
hat $\mathcal{B}$ seinen Keller geleert.

Satz 10.8

Für eine formale Sprache L sind äquivalent:

1. $L = L(G)$ für eine kontextfreie Grammatik G.
2. $L = N(\mathcal{A})$ für einen PDA $\mathcal{A}$.

Dresden

Beweis von Satz 10.8:

„1 $\longrightarrow$ 2": Es sei $G = (N, \Sigma, P, S)$ eine kontextfreie Grammatik.

Der zugehörige PDA simuliert Linksableitungen von G, d.h. Ableitungen, bei denen stets das am weitesten links stehende Nichtterminalsymbol zuerst ersetzt wird.

Beachte: Jedes Wort in $L(G)$ kann durch eine Linksableitung erzeugt werden (kontextfrei!).

Wir definieren dazu $\mathcal{A} = (\{q\}, \Sigma, \Sigma \cup N, q, S, \Delta)$ mit

$\Delta := \{(q, \varepsilon, A, \gamma, q) \mid A \longrightarrow \gamma \in P\} \cup$	Anwenden einer Produktion auf oberstes Kellersymbol
$\{(q, a, a, \varepsilon, q) \mid a \in \Sigma\}$	Entfernen bereits erzeugter Terminalsymbole von der Kellerspitze, wenn sie in der Eingabe vorhanden sind

Dresden

Beispiel:

$$P = \{S \longrightarrow \varepsilon, S \longrightarrow aSa, S \longrightarrow bSb\}$$

liefert die Übergänge

q	ε	S	ε	q	$S \longrightarrow \varepsilon$
q	ε	S	aSa	q	$S \longrightarrow aSa$
q	ε	S	bSb	q	$S \longrightarrow bSb$
q	a	a	ε	q	a entfernen
q	b	b	ε	q	b entfernen

Die Ableitung

$$S \vdash aSa \vdash abSba \vdash abba$$

entspricht der Konfigurationsfolge

$$(q, abba, S) \vdash (q, abba, aSa) \vdash (q, bba, Sa) \vdash (q, bba, bSba) \vdash$$

$$(q, ba, Sba) \vdash (q, ba, ba) \vdash (q, a, a) \vdash (q, \varepsilon, \varepsilon).$$

Behauptung:

Für $u, v \in \Sigma^*$ und $\alpha \in \{\varepsilon\} \cup N \cdot (\Sigma \cup N)^*$ gilt:

$S \vdash_G^* u\alpha$ mit Linksableitung gdw $(q, uv, S) \vdash^* (q, v, \alpha)$.

Beachte:

Für $\alpha = \varepsilon = v$ folgt:

$$S \vdash_G^* u \;\text{ gdw }\; (q, u, S) \vdash^* (q, \varepsilon, \varepsilon),$$

d.h. $L(G) = N(\mathcal{A})$.

Dresden

„2 $\longrightarrow$ 1“: Es sei $\mathcal{A} = (Q, \Sigma, \Gamma, q_0, Z_0, \Delta)$ ein PDA.

Die Nichtterminalsymbole der zugehörigen Grammatik G sind alle Tripel

$$[p, Z, q] \in Q \times \Gamma \times Q.$$

Idee: Es soll

$$[p, Z, q] \vdash_G^* u \in \Sigma^*$$

genau dann gelten, wenn

- $\mathcal{A}$ kommt vom Zustand p in den Zustand q
- durch Lesen von u auf dem Eingabeband und
- Löschen von Z aus dem Keller

ohne dabei die Symbole unter Z anzutasten.

Wie realisiert man dies durch geeignete Produktionen?

Betrachte den PDA-Übergang

$$(p, a, Z, X_1 \ldots X_n, p') \in \Delta.$$

Hier wird a auf dem Eingabeband verbraucht (falls nicht $a = \varepsilon$)

und das Kellersymbol Z wird durch $X_1 \ldots X_n$ ersetzt.

Um den Kellerinhalt zu erreichen, den man durch einfaches Wegstreichen von Z erhalten würde, muß man also nun noch $X_1 \ldots X_n$ löschen.

Löschen von X_i kann durch das Nichtterminalsymbol $[p_{i-1}, X_i, p_i]$ beschrieben werden (für geeignete Zwischenzustände p_{i-1}, p_i).

Formale Definition:

$$
\begin{aligned}
G &:= (N, \Sigma, P, S) \text{ mit} \\
N &:= \{S\} \cup \{[p, Z, q] \mid p, q \in Q, Z \in \Gamma\} \\
P &:= \{\, S \longrightarrow [q_0, Z_0, q] \mid q \in Q \,\} \cup \\
&\quad\ \{\, [p, Z, q] \longrightarrow a[p_0, X_1, p_1][p_1, X_2, p_2] \ldots [p_{n-1}, X_n, q] \mid \\
&\qquad\quad (p, a, Z, X_1 \ldots X_n, p_0) \in \Delta \\
&\qquad\quad a \in \Sigma \cup \{\varepsilon\} \\
&\qquad\quad p, q, p_0, p_1, \ldots, p_{n-1} \in Q, \\
&\qquad\quad \text{wobei } q = p_0 \text{ falls } n = 0 \quad \}
\end{aligned}
$$

Beachte:

Für $n = 0$ hat man die Produktion $[p, Z, q] \longrightarrow a$,
welche dem Übergang $(p, a, Z, \varepsilon, q) \in \Delta$ entspricht.

Behauptung:

Für alle $p, q \in Q, u \in \Sigma^*, Z \in \Gamma$ gilt

$$(p, u, Z) \vdash_{\mathcal{A}}^* (q, \varepsilon, \varepsilon) \text{ gdw } [p, Z, q] \vdash_G^* u$$

Beweis durch Induktion über die Länge der Konfigurationsfolge („⇒“) bzw die Länge der Ableitung („⇐“).

Für $p = q_0$ und $Z = Z_0$ folgt daraus:

$$(q_0, u, Z_0) \vdash^* (q, \varepsilon, \varepsilon) \text{ gdw } S \vdash_G [q_0, Z_0, q] \vdash_G^* u,$$

d.h. $u \in N(\mathcal{A})$ gdw $u \in L(G)$.

Dresden

Beispiel: PDA für $\{a^n b^n \mid n \geq 1\}$ (vgl. Beispiel 10.3)

q_0	a	Z_0	ZZ_0	q_0
q_0	a	Z	ZZ	q_0
q_0	b	Z	ε	q_1
q_1	b	Z	ε	q_1
q_1	ε	Z_0	ε	f

$$(q_0, ab, Z_0) \overset{(q_0,a,Z_0,ZZ_0,q_0)}{\vdash} (q_0, b, ZZ_0) \overset{(q_0,b,Z,\varepsilon,q_1)}{\vdash} (q_1, \varepsilon, Z_0) \overset{(q_1,\varepsilon,Z_0,\varepsilon,f)}{\vdash} (f, \varepsilon, \varepsilon)$$

entspricht der Ableitung

$$S \;\vdash_G\; [q_0, Z_0, f] \;\vdash_G\; a[q_0, Z, q_1][q_1, Z_0, f] \;\vdash_G\; ab[q_1, Z_0, f] \;\vdash_G\; ab$$

Dresden

Wegen der gerade gezeigten Äquivalenz zwischen kontextfreien Sprachen und PDA akzeptierbaren Sprachen, kann man Eigenschaften von kontextfreien Sprachen mit Hilfe von Eigenschaften von Kellerautomaten zeigen.

Als Beispiel betrachten wir den Durchschnitt von kontextfreien Sprachen mit regulären Sprachen.

Wir wissen:

der Durchschnitt zweier kontextfreier Sprachen muß nicht kontextfrei sein.

Dahingegen gilt:

Korollar 10.9

Es sei $L \subseteq \Sigma^*$ kontextfrei und $R \subseteq \Sigma^*$ regulär.

Dann ist $L \cap R$ kontextfrei.

Beweis von Korollar 4.9

Es sei

- $L = L(\mathcal{A})$ für einen PDA $\mathcal{A} = (Q, \Sigma, \Gamma, q_0, Z_0, \Delta, F)$ und
- $R = L(\mathcal{A}')$ für einen DEA $\mathcal{A}' = (Q', \Sigma, q_0', \delta', F')$.

Wir konstruieren den „Produktautomaten" von $\mathcal{A}$ und $\mathcal{A}'$,
der ein PDA für $L \cap R$ ist:

$$\mathcal{B} \;=\; (Q \times Q', \Sigma, \Gamma, (q_0, q_0{}'), Z_0, \Delta', F \times F')$$

mit

$$\begin{aligned}\Delta' = &\{((p,p'), a, Z, \gamma, (q,q')) \mid a \in \Sigma, (p,a,Z,\gamma,q) \in \Delta \text{ und } \delta(p',a) = q'\} \cup \\ &\{((p,p'), \epsilon, Z, \gamma, (q,p')) \mid (p,\epsilon,Z,\gamma,q) \in \Delta\}\end{aligned}$$

Man zeigt nun leicht (durch Induktion über k):

$$((p,p'), uv, \gamma) \vdash_{\mathcal{B}}^{k} ((q,q'), v, \beta) \quad gdw. \quad (p, uv, \gamma) \vdash_{\mathcal{A}}^{k} (q, v, \beta) \text{ und } p' \xrightarrow{u}_{\mathcal{A}'} q'.$$

Dresden

Bemerkung 10.10

Ist $\mathcal{A}$ in dem Beweis ein deterministischer PDA,
so ist der konstruierte Produktautomat $\mathcal{B}$ auch deterministisch,
wenn man $\mathcal{A}'$ o.E. deterministisch wählt.

Man kann diese Abschlusseigenschaften von kontextfreien Sprachen unter anderem dazu verwenden, für gewisse Sprachen nachzuweisen, daß sie nicht kontextfrei sind.

Beispiel 10.11

Die Sprache $L = \{ww \mid w \in \{a, b\}^*\}$ ist nicht kontextfrei.

Für endliche Automaten hatten wir gesehen, daß bereits die deterministischen endlichen Automaten alle erkennbaren Sprachen akzeptieren. Dazu haben wir gezeigt, daß man zu jedem NEA einen äquivalenten DEA konstruieren kann.

Für Kellerautomaten ist dies nicht möglich.

Definition 10.12

Eine Sprache heißt deterministisch kontextfrei (dkf), falls sie von einem deterministischen Kellerautomaten (DPDA) akzeptiert wird.

Wir werden zeigen, daß die Sprache

$$L = \{w\overleftarrow{w} \mid w \in \{a,b\}^*\}$$

nicht dkf ist.

Wir haben bereits gesehen (Beispiel 10.5), daß L kontextfrei ist.

Dresden

Definition 10.13

Zu einer Sprache $L \subseteq \Sigma^*$ sei

$$Min(L) := \{w \in L \mid \text{kein echtes Präfix von } w \text{ liegt in } L\}.$$

Lemma 10.14

Ist L dkf, so auch $Min(L)$.

Beweis:

Es sei $\mathcal{A} = (Q, \Sigma, \Gamma, q_0, Z_0, \Delta, F)$ ein DPDA für L,
d.h. $L = L(\mathcal{A})$ (Akzeptanz mit Endzustand).

Wir ändern $\mathcal{A}$ so ab, daß gilt: wurde das erste Mal ein Endzustand erreicht, so kann danach keiner mehr erreicht werden.

Dresden

Definition 10.13

Zu einer Sprache $L \subseteq \Sigma^*$ sei

$$Min(L) := \{w \in L \mid \text{kein echtes Präfix von } w \text{ liegt in } L\}.$$

Lemma 10.14

Ist L dkf, so auch $Min(L)$.

Beachte:

Bei einem nicht-deterministischen PDA funktioniert die Konstruktion nicht, da es für Eingabe u verschiedene Konfigurationsfolgen geben kann.

Satz 10.15

Die Sprache $L = \{w\overleftarrow{w} \mid w \in \{a,b\}^*\}$ ist nicht deterministisch kontextfrei.

Beweis:

Wäre L dkf, so mit Bemerkung 10.10 und Lemma 10.14 auch

$$L' := Min(L) \cap (ab)^+(ba)^+(ab)^+(ba)^+.$$

Es ist

$$L' = \{(ab)^i(ba)^j(ab)^j(ba)^i \mid i > j > 0\}.$$

Für $i \leq j$ wäre $(ab)^i(ba)^i \in L$ echtes Präfix.

Mit Hilfe des Pumping Lemmas zeigt man, daß L' nicht kontextfrei ist.

Deterministisch kontextfreie Sprachen sind für die Syntaxanalyse von Programmen interessant, da für sie das Wortproblem linear entscheidbar ist (im Gegensatz zu kubisch beim CYK-Algorithmus für allgemeine kontextfreie Sprachen).

Dies ist wichtig im Compilerbau

(siehe z.B. Ingo Wegner: Theoretische Informatik, S. 199–216).

Im Gegensatz zu den kf Sprachen ist die Klasse der dkf Sprachen unter Komplement abgeschlossen.

Satz 10.16

Ist L deterministisch kontextfrei, so auch $\overline{L}$.

(ohne Beweis)

Beachte:

Für einen DPDA, der L mit Endzuständen aus F akzeptiert, genügt es nicht, als neue Endzustandsmenge einfach $Q \setminus F$ zu nehmen, um einen Automaten für $\overline{L}$ zu erhalten.

Der Grund ist, dass es für den DPDA zwei verschiedene Gründe geben kann, warum er ein Wort w nicht akzeptiert:

i) wenn w ganz gelesen ist, so ist der Automat nicht in einem Endzustand

ii) das Wort w wird nicht ganz gelesen

Man muß den DPDA zunächst in einen DPDA umwandeln, bei dem der zweite Grund nie auftritt (siehe z.B. Ingo Wegner: Theoretische Informatik, S. 193–195, für Details).