Automaten und Sprachen

§ 11: Kellerautomaten

Überblick

Teil I: Endliche Automaten und reguläre Sprachen

0. Grundbegriffe
1. Endliche Automaten
2. Nachweis der Nichterkennbarkeit
3. Abschlusseigenschaften
4. Entscheidungsprobleme
5. Reguläre Ausdrücke und Sprachen
6. Minimale DEAs und die Nerode-Rechtskongruenz

Teil II: Grammatiken, kontextfreie Sprachen und Kellerautomaten

7. Die Chomsky-Hierarchie
8. Rechtslineare Grammatiken und reguläre Sprachen
9. Normalformen und Entscheidungsprobleme
10. Abschlusseigenschaften und Pumping-Lemma
11. Kellerautomaten
12. Die Struktur kontextfreier Sprachen

Intuitionen

Definition kontextfreier Sprachen: mittels Grammatiken

Natürliche Frage:

Gibt es auch ein passendes Automatenmodell?

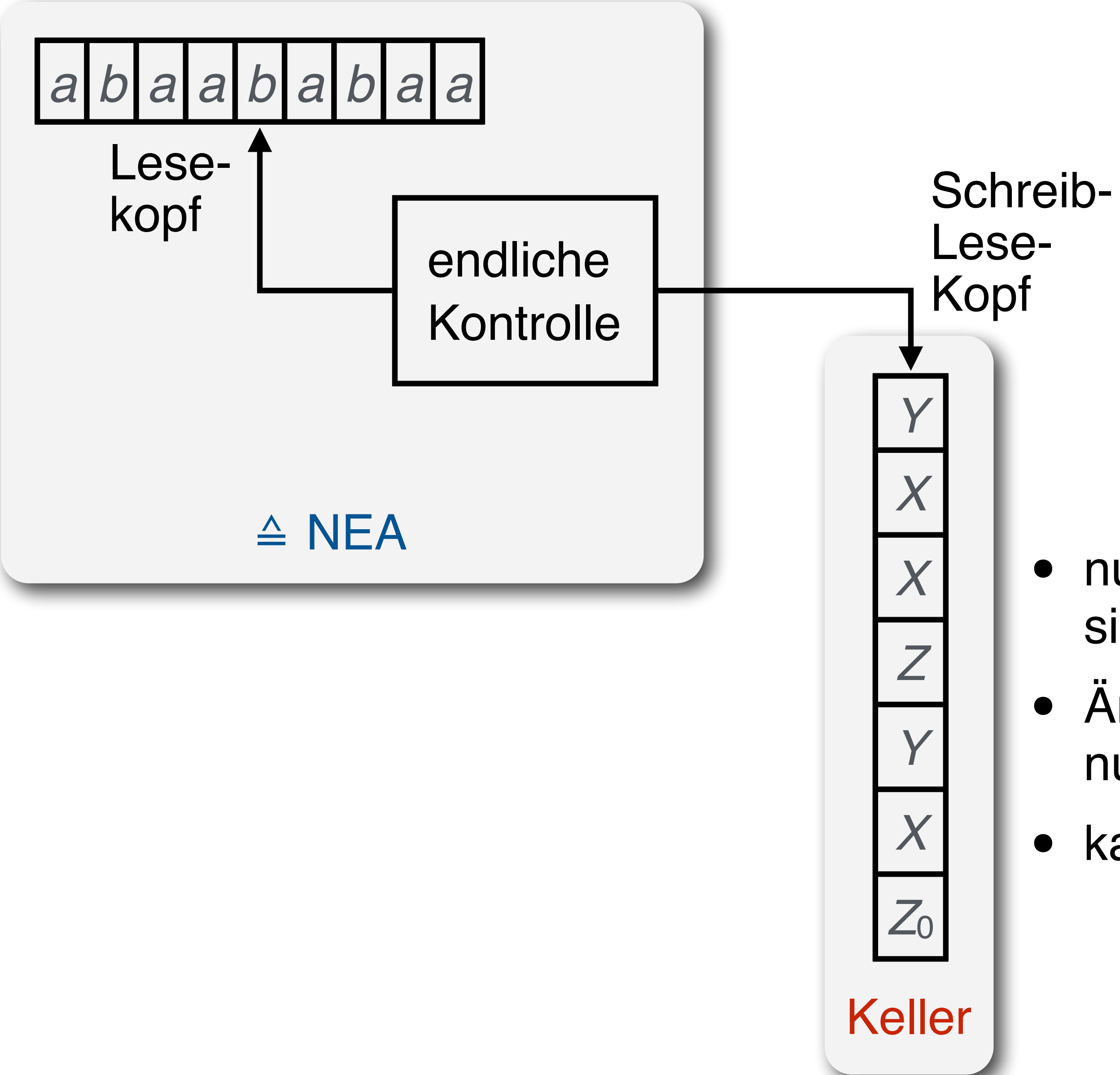
Endliche Automaten sind ungeeignet:

- sie können nicht alle kontextfreien Sprachen erkennen,
- denn sie können nicht „unbeschränkt zählen“

  (das muss man aber, um z.B. $\{a^n b^n \mid n \geq 0\}$ erkennen zu können).

Wir betrachten daher eine Erweiterung um eine unbeschränkte Speicherkomponente – einen Keller (engl.: Stack)

Aufbau



- nur oberstes Symbol sichtbar
- Änderung des Inhalts nur von oben
- kann beliebig groß werden

Definition

Definition 11.1 (Kellerautomat, PDA)

Ein Kellerautomat (auch PDA, für englisch pushdown automaton) 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,
- $\Delta \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \times \Gamma^* \times Q$
  die endliche Übergangsrelation ist und
- $F \subseteq Q$ eine Menge von akzeptierenden Zuständen ist.

Bedeutung der Übergangsrelation

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

$(q, a, Z, \gamma, q') \in \Delta$ bedeutet anschaulich:

- wenn der PDA im Zustand q ist,
- das aktuelle Eingabesymbol a und
- das oberste Kellersymbol Z liest,
- dann darf er Z durch γ ersetzen und
- in den Zustand q' und zum nächsten Eingabesymbol übergehen.

$(q, \varepsilon, Z, \gamma, q') \in \Delta$ bedeutet anschaulich:

dasselbe wie oben, nur dass kein Eingabesymbol gelesen wird (Lesekopf ändert Position nicht)

Beachte: dies ist ein nichtdeterministisches Automatenmodell.

Konfigurationen

Die Definition von **Akzeptanz** basiert auf dem Begriff einer Konfiguration

Die Konfiguration eines Kellerautomaten ist bestimmt durch

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

Konfigurationen und Akzeptanz

Definition 11.2

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

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

Die Übergangsrel. Δ ermöglicht die folgenden Konfigurationsübergänge:

- $(q, aw, Z\gamma) \;\vdash_{\mathcal{A}}\; (q', w, \beta\gamma)$ falls $(q, a, Z, \beta, q') \in \Delta$
- $(q, w, Z\gamma) \;\vdash_{\mathcal{A}}\; (q', w, \beta\gamma)$ falls $(q, \varepsilon, Z, \beta, q') \in \Delta$

Wir schreiben $\mathcal{K} \vdash_{\mathcal{A}}^* \mathcal{K}'$ wenn $\exists \mathcal{K}_0, \dots, \mathcal{K}_n$ mit:

$$\mathcal{K} = \mathcal{K}_0 \;\vdash_{\mathcal{A}}\; \mathcal{K}_1 \;\vdash_{\mathcal{A}}\; \cdots \;\vdash_{\mathcal{A}}\; \mathcal{K}_n = \mathcal{K}'$$

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

$(q_0, w, Z_0) \;\vdash_{\mathcal{A}}^*\; (q, \varepsilon, \gamma)$ für ein $q \in F$ und $\gamma \in \Gamma^*$

(akz. Zustand, Eingabe w zu Ende gelesen, Kellerinhalt egal)

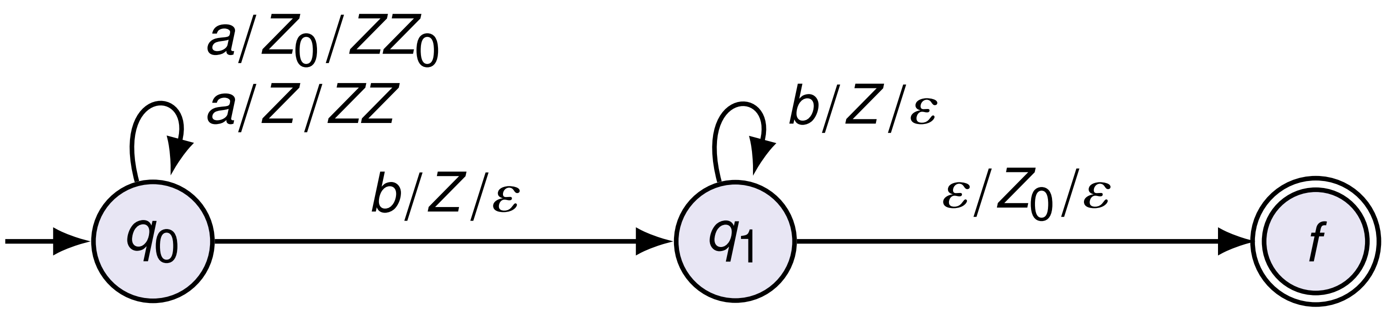
Die von $\mathcal{A}$ erkannte Sprache ist $L(\mathcal{A}) \;=\; \{w \in \Sigma^* \mid \mathcal{A} \text{ akzeptiert } w\}$.

Ein Beispiel

Beispiel 11.3 ein PDA für $\{a^n b^n \mid n \geq 1\}$

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

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	(wechsle in akzeptierenden Zustand)



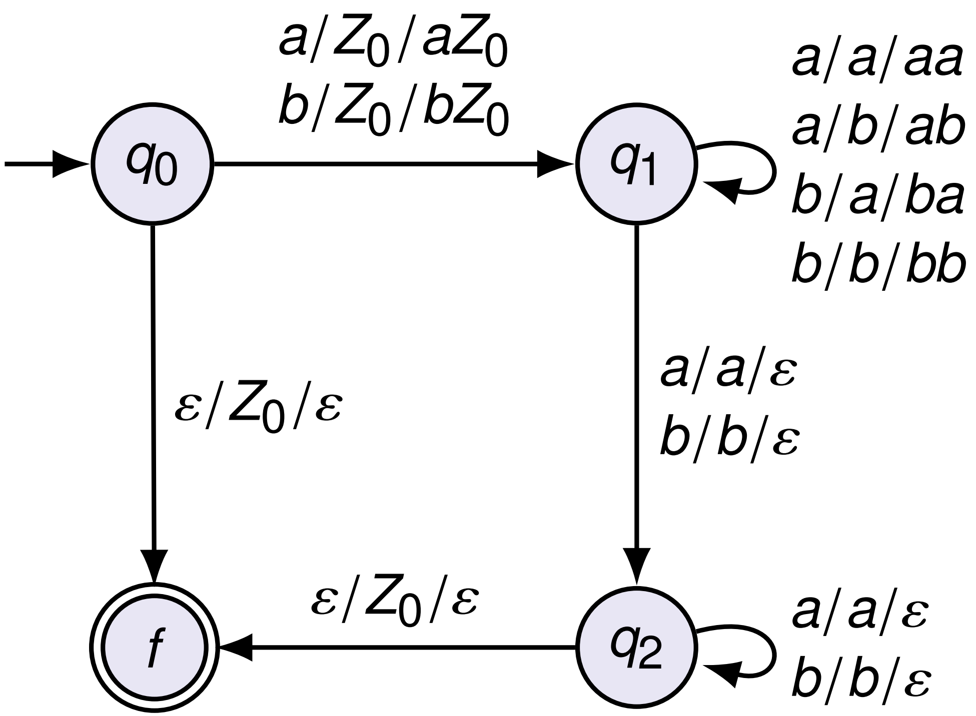
Betrachte Eingaben *aabb* *aab* *abb*

Ein weiteres Beispiel

Für Wort $w = a_1 \cdots a_n$ sei $w^R = a_n \cdots a_1$.

Beispiel 11.4 ein PDA für $\{w \cdot w^R \mid w \in \{a, b\}^*\}$

$Q = \{q_0, q_1, q_2, f\}$ $\Sigma = \{a, b\}$ $\Gamma = \{a, b, Z_0\}$ $F = \{f\}$ Δ wie im Bild:



Betrachte Eingabe *abba*

Varianten von PDAs

Auch von PDAs gibt es mehrere natürliche Varianten:

- deterministisch vs. nicht-deterministisch
- mit und ohne ε-Übergänge
- mit Akzeptanz per Zustand oder per leerem Keller (gleich mehr)

Es stellt sich die Frage:

Sind alle diese Varianten gleich ausdrucksstark, d.h. erkennen sie alle genau die kontextfreien Sprachen?

Akzeptanz per leerem Keller

Definition 11.5

Ein PDA mit Akzeptanz per leerem Keller ist ein Tupel

$$\mathcal{A} = (Q, \Sigma, \Gamma, q_0, Z_0, \Delta),$$

wobei alle Komponenten bis auf F wie in Definition 11.1 sind
(und es keine akzeptierenden Zustände gibt).

$\mathcal{A}$ akzeptiert w gdw. $(q_0, w, Z_0) \vdash_{\mathcal{A}}^{*} (q, \varepsilon, \varepsilon)$ für ein $q \in Q$.

(d. h. belieb. Zustand, Eingabe zu Ende gelesen, Keller leer)

Satz 11.6

Für jede formale Sprache L sind äquivalent:

1. L wird von einem PDA (mit akzeptierenden Zuständen) erkannt.
2. L wird von einem PDA mit Akzeptanz per leerem Keller erkannt.

Akzeptanz per leerem Keller

2 ⇒ 1 (leerer Keller ⇒ akzeptierender Zustand)

Sei $\mathcal{A} = (Q, \Sigma, \Gamma, q_0, Z_0, \Delta)$ PDA, der per leerem Keller akzeptiert.

Idee:

Füge akzeptierenden Zustand f hinzu, wechsle zu f sobald Keller leer

Problem: wenn der Keller leer ist, ist kein Übergang mehr möglich

Lösung:

Zusätzliches Kellerstartsymbol Z_0', das den leeren Keller repräsentiert

Definiere $\mathcal{A}' = (Q \uplus \{q_0', f\}, \Sigma, \Gamma \uplus \{Z_0'\}, q_0', Z_0', \Delta', \{f\})$, wobei

$\Delta' = \Delta \cup \{(q_0', \varepsilon, Z_0', Z_0 Z_0', q_0)\}$ Altes Kellerstartsymbol auf Keller

$\cup \{(q, \varepsilon, Z_0', \varepsilon, f) \mid q \in Q\}$ Bei leerem Keller zu f

Man kann nun zeigen, dass $L(\mathcal{A}) = L(\mathcal{A}')$.

Akzeptanz per leerem Keller

$1 \Rightarrow 2$ (akzeptierender Zustand $\Rightarrow$ leerer Keller)

Sei $\mathcal{A} = (Q, \Sigma, \Gamma, q_0, Z_0, \Delta, F)$ PDA, der per Zustand akzeptiert.

Idee: Füge ε-Transitionen hinzu, die von akz. Zustand aus den Keller leeren

Problem: der Keller könnte auch in nicht-akz. Zustand leer werden

Lösung:

Zusätzliches Kellerstartsymbol Z_0' liegt dauerhaft zuunterst im Keller

Definiere $\mathcal{A}' = (Q \uplus \{q_0', \ell\}, \Sigma, \Gamma \uplus \{Z_0'\}, q_0', Z_0', \Delta')$, wobei

$$\Delta' = \Delta \cup \{(q_0', \varepsilon, Z_0', Z_0 Z_0', q_0)\}$$

Altes Kellerstartsymbol auf Keller

$$\cup \{(q, \varepsilon, Z, \varepsilon, \ell) \mid q \in F \text{ und } Z \in \Gamma \cup \{Z_0'\}\}$$

$$\cup \{(\ell, \varepsilon, Z, \varepsilon, \ell) \mid Z \in \Gamma \cup \{Z_0'\}\}$$

Keller leeren

Man kann nun zeigen, dass $L(\mathcal{A}) = L(\mathcal{A}')$.

Kellerautomaten versus kontextfreie Grammatiken

Satz 11.9

Für jede formale Sprache L sind äquivalent:

1. L wird von einer kontextfreien Grammatik erzeugt.
2. L wird von einem PDA erkannt.

Als Vorbereitung für den Beweis von „1 $\Rightarrow$ 2“
führen wir zunächst den Begriff der Linksableitung ein.

Linksableitungen

Definition 11.7

Sei G eine kfG. Eine Ableitung

$$S = w_0 \vdash_G w_1 \vdash_G w_2 \vdash_G \cdots \vdash_G w_n$$

heißt Linksableitung, wenn sich w_{i+1} aus w_i durch Anwenden einer Regel auf das am weitesten links stehende Nichtterminal ergibt, für alle $i < n$.

Lemma 11.8

Für jede kfG G gilt:

$$L(G) = \{w \in \Sigma^* \mid w \text{ kann in } G \text{ mit Linksableitung erzeugt werden}\}$$

Beweisidee:

- Wenn $w \in L(G)$, dann gibt es Ableitung von w in G.
- Diese kann als Ableitungsbaum dargestellt werden.
- Vom Ableitungsbaum kann man Linksableitung ablesen, die w erzeugt.

Von kfG zu PDA

Beweis von Satz 11.9.

„1 ⇒ 2“: Sei $G = (N, \Sigma, P, S)$ eine kfG.

Wir konstruieren PDA $\mathcal{A} = (Q, \Sigma, \Gamma, q_0, Z_0, \Delta)$ (Akz. per leerem Keller), mit $L(\mathcal{A}) = L(G)$.

Idee: $\mathcal{A}$ simuliert Linksableitungen auf dem Keller

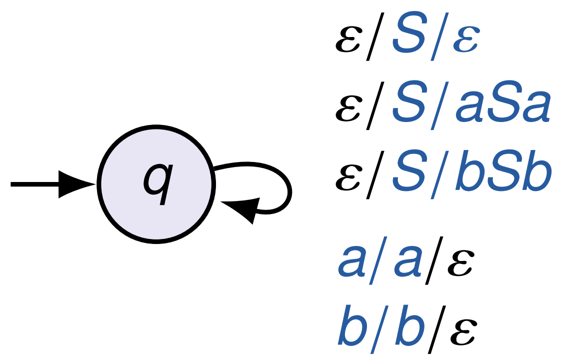
- $Q = \{q\}$ und $q_0 = q$
- $\Gamma = \Sigma \cup N$
- $Z_0 = S$
- Δ besteht aus folgenden Übergängen:
  1. Anwenden von Regeln auf Nichtterminale oben auf Keller:
     Übergang $(q, \varepsilon, A, \gamma, q)$ für jede Regel $A \to \gamma \in P$
  2. Entfernen von Terminalsymbolen oben auf Keller & aus Eingabe:
     Übergang $(q, a, a, \varepsilon, q)$ für jedes Terminalsymbol a

Von kfG zu PDA

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

… 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 Linksableitung $\quad S \vdash_G aSa \vdash_G abSba \vdash_G abba$
entspricht der Konfigurationsfolge

$$(q, abba, S) \vdash_{\mathcal{A}} (q, abba, aSa) \vdash_{\mathcal{A}} (q, bba, Sa) \vdash_{\mathcal{A}} (q, bba, bSba)$$
$$\vdash_{\mathcal{A}} (q, ba, Sba) \vdash_{\mathcal{A}} (q, ba, ba) \vdash_{\mathcal{A}} (q, a, a) \vdash_{\mathcal{A}} (q, \varepsilon, \varepsilon)$$

Von kfG zu PDA

Behauptung

Für alle $w \in \Sigma^*$ gilt:

$$S \vdash_G^* w \text{ mittels Linksableitung} \quad \text{gdw.} \quad (q, w, S) \vdash_{\mathcal{A}}^* (q, \varepsilon, \varepsilon)$$

Beweisdetails finden sich im Skript

Grundidee: Wandle Linksableitung in akzeptierende Konfigurationsfolge und umgekehrt.

Von kfG zu PDA ohne ε-Übergänge

Zur Erinnerung:

Jede kfG kann in Greibach Normalform gewandelt werden

Alle Produktionen haben dann folgende Form:

- $A \longrightarrow aB_1 \cdots B_n \quad (A \in N,\ a \in \Sigma,\ B_1, \ldots, B_n \in N)$ und
- eventuell $S \longrightarrow \varepsilon$, wobei S nirgends rechts vorkommt

Das führt zu einer leicht anderen Konstruktion von $\mathcal{A}$,

bei der keine ε-Übergänge verwendet werden.

Dies wird liefern:

PDAs mit und ohne ε-Übergänge erkennen dieselben Sprachen

Wir arbeiten mit Akzeptanz per Zustand,

konstruieren also PDA $\mathcal{A} = (Q, \Sigma, \Gamma, q_0, Z_0, \Delta, F)$

Von kfG zu PDA ohne ε-Übergänge

- $Q = \{q, f\} \cup \{q_A \mid A \in N\}$ wobei $q_0 = q$
- $\Gamma = N \cup \{Z_0\}$ (keine Terminalsymbole mehr auf Keller)
- $F = \{f, q\}$ falls $S \to \varepsilon \in P$ und $F = \{f\}$ sonst
- Δ besteht aus folgenden Übergängen:

1. Anwenden von Regeln auf Nichtterminale oben auf Keller:
   $(q, a, A, B_1 \cdots B_n, q)$ für jede Regel $A \to aB_1 \cdots B_n \in P$
2. Am Anfang: Z_0 wird erhalten:
   $(q, a, Z_0, B_1 \cdots B_n Z_0, q)$ für jede Regel $S \to aB_1 \cdots B_n \in P$
3. Wenn alle Nichtterminale abgearbeitet sind, akzeptiere:
   $(q, \varepsilon, Z_0, \varepsilon, f)$

Oh, Mist!

Idee: beim Übergang zu f lesen wir das letzte Terminal des Wortes!

Von kfG zu PDA ohne ε-Übergänge

Idee: beim Übergang zu f lesen wir das letzte Terminal des Wortes!

Bei einer Linksableitung und wegen Greibach-Normalform kann das letzte Terminal nur auf folgende Weise erzeugt werden:

$$\cdots \vdash_G a_1 \cdots a_{n-1} B \vdash_G a_1 \cdots a_{n-1} a_n$$

mittels $B \to a_n \in P$

Also darf B gar nicht erst auf den Keller, denn wir wollen den Übergang

$$(q, a_n, Z_0, \varepsilon, f)$$

und B würde das Z_0 “verbergen”.

Von kfG zu PDA ohne ε-Übergänge

Wir ersetzen

3. Wenn alle Nichtterminale abgearbeitet sind, akzeptiere:
   $(q, \varepsilon, Z_0, \varepsilon, f)$

durch

3'. Nichtdeterministisches Raten des letzten Nichtterminals, Speichern in Zustand statt auf Keller:
   $(q, a, A, B_1 \cdots B_{n-1}, q_{B_n})$ für jede Regel $A \rightarrow aB_1 \cdots B_n \in P$

4. Ebenso am Anfang unter Erhalten von Z_0:
   $(q, a, Z_0, B_1 \cdots B_{n-1} Z_0, q_{B_n})$ für jede Regel $S \rightarrow aB_1 \cdots B_n \in P$

5. Verbliebene Nichtterminale vom Keller abarbeiten:
   $(q_B, a, A, B_1 \cdots B_n, q_B)$ für jede Regel $A \rightarrow aB_1 \cdots B_n \in P$

6. Verarbeiten des letzten Nichtterminals:
   $(q_B, a, Z_0, \varepsilon, f)$ für jede Regel $B \rightarrow a \in P$

Beispiel

$$P = \{S \longrightarrow aAB,\ \ A \longrightarrow a,\ \ B \longrightarrow bCD,\ \ B \longrightarrow b,\ \ C \longrightarrow c,\ \ D \longrightarrow d\}$$

Beispielhaft einige Übergänge:

q	a	S	AB	q	$S \longrightarrow aAB$	(Typ 1)
q	a	Z_0	ABZ_0	q	$S \longrightarrow aAB$	(Typ 2)
q	a	Z_0	AZ_0	q_B	$S \longrightarrow aAB$	(Typ 3')
q	a	A	ε	q	$A \longrightarrow a$	(Typ 1)
q	b	B	C	q_D	$B \longrightarrow bCD$	(Typ 3')
q_D	d	Z_0	ε	f	$D \longrightarrow d$	(Typ 6)

Die Linksableitung $S \vdash_G aAB \vdash_G aaB \vdash_G aabCD \vdash_G aabcD \vdash_G aabcd$ entspricht der Konfigurationsfolge

$$(q, aabcd, Z_0) \vdash_{\mathcal{A}} (q, abcd, ABZ_0) \vdash_{\mathcal{A}} (q, bcd, BZ_0) \vdash_{\mathcal{A}} (q_D, cd, CZ_0)$$
$$\vdash_{\mathcal{A}} (q_D, d, Z_0) \vdash_{\mathcal{A}} (f, \varepsilon, \varepsilon)$$

Von kfG zu PDA ohne ε-Übergänge

Ohne Beweis:

Behauptung

Für alle $w \in \Sigma^*$ gilt:

$S \vdash_G^* w$ mit Linksabl. gdw. $(q, w, S) \vdash_{\mathcal{A}}^* (p, \varepsilon, \gamma)$ für ein $p \in F$ und $\gamma \in \Gamma^*$.

Aus der Behauptung folgt unmittelbar: $L(G) = L(\mathcal{A})$

Überblick

Teil I: Endliche Automaten und reguläre Sprachen

0. Grundbegriffe
1. Endliche Automaten
2. Nachweis der Nichterkennbarkeit
3. Abschlusseigenschaften
4. Entscheidungsprobleme
5. Reguläre Ausdrücke und Sprachen
6. Minimale DEAs und die Nerode-Rechtskongruenz

Teil II: Grammatiken, kontextfreie Sprachen und Kellerautomaten

7. Die Chomsky-Hierarchie
8. Rechtslineare Grammatiken und reguläre Sprachen
9. Normalformen und Entscheidungsprobleme
10. Abschlusseigenschaften und Pumping-Lemma
11. Kellerautomaten
12. Die Struktur kontextfreier Sprachen

Kellerautomaten versus kontextfreie Grammatiken

Zur Erinnerung: Satz 11.9

Für jede formale Sprache L sind äquivalent:

1. L wird von einer kontextfreien Grammatik erzeugt.
2. L wird von einem PDA erkannt.

Wir beweisen jetzt „2 ⇒ 1“.

Dazu müssen wir zeigen:
Für jeden PDA $\mathcal{A}$ gibt es eine kfG G, so dass $L(G) = L(\mathcal{A})$.

Wir können o. B. d. A. annehmen,
dass $\mathcal{A}$ per leerem Keller akzeptiert (Satz 11.6).

Von PDA zu kfG

„2 ⇒ 1“:

Sei $\mathcal{A} = (Q, \Sigma, \Gamma, q_0, Z_0, \Delta)$ ein PDA mit Akz. per leerem Keller.

Die Nichtterminalsymbole der zu konstruierenden kfG 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

- $\mathcal{A}$ von Zustand p aus Zustand q erreichen kann (evtl. in mehreren Schritten),
- durch Lesen von u aus der Eingabe und
- Löschen von Z aus dem Keller, ohne dabei die Symbole unter Z anzutasten.

Es gilt also: $w \in L(\mathcal{A})$ gdw. $[q_0, Z_0, q] \vdash_G^* w$ für ein $q \in Q$

Von PDA zu kfG

$\mathcal{A}$ erreicht von p aus q unter Lesen von Wort u und Löschen von Z gdw.

es gibt Übergang $(p, a, Z, X_1 \cdots X_n, p_0)$ so dass man von p_0 aus

- den Zustand q erreichen kann und dabei
- den Rest von u (ohne Präfix a) lesen und (eventuell $a = \varepsilon$)
- $X_1 \cdots X_n$ vom Keller entfernen ohne die Symbole darunter anzutasten

Oder anders formuliert…

Von PDA zu kfG

$\mathcal{A}$ erreicht von p aus q unter Lesen von Wort u und Löschen von Z gdw.

es gibt Übergang $(p, a, Z, X_1 \cdots X_n, p_0)$ und Zustände $p_1, \ldots, p_{n-1}$, so dass

- u kann aufgeteilt werden in $u = a v_1 \cdots v_n$, so dass
- $[p_0, X_1, p_1]$ unter Lesen von v_1,
  $[p_1, X_2, p_2]$ unter Lesen von v_2,
  $\ldots$
  $[p_{n-1}, X_n, q]$ unter Lesen von v_n

(eventuell $a = \varepsilon$)

Unsere kfG enthält Produktion

$$[p, Z, q] \longrightarrow a\,[p_0, X_1, p_1] \cdots [p_{n-1}, X_n, q]$$

für alle möglichen Folgen von Zwischenzuständen $p_1, \ldots, p_{n-1}$

Beachte:

- die Länge n ist gegeben durch $X_1 \cdots X_n$ im ersten Übergang
- intuitiv rät die Grammmatik $p_0, \ldots, p_{n-1}$ nichtdeterministisch

Von PDA zu kfG

Wir konstruieren also kfG $G = (N, P, \Sigma, S)$ mit

$$N = \{S\} \cup \Big\{ [p, Z, q] \;\Big|\; p, q \in Q,\ Z \in \Gamma \Big\}$$

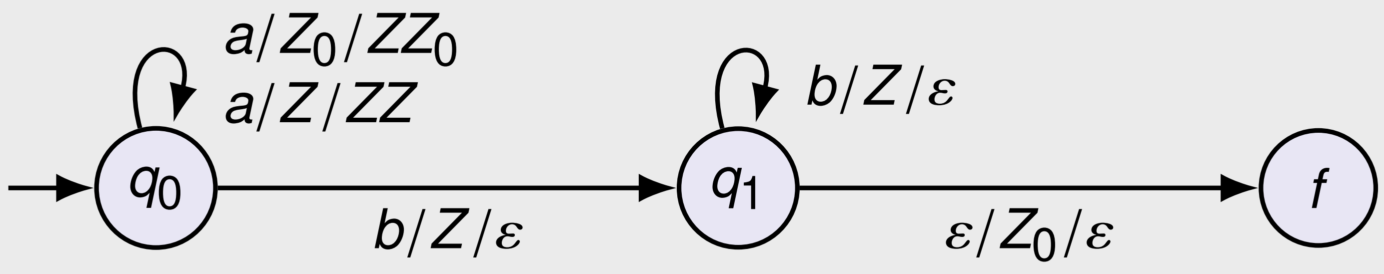
$$P = \Big\{ S \longrightarrow [q_0, Z_0, q] \;\Big|\; q \in Q \Big\} \cup$$

$$\cup \Big\{ [p, Z, q] \longrightarrow a \;\Big|\; (p, a, Z, \varepsilon, q) \in \Delta \text{ mit } a \in \Sigma \cup \{\varepsilon\} \Big\}$$

$$\cup \Big\{ [p, Z, q] \longrightarrow a\,[p_0, X_1, p_1]\,[p_1, X_2, p_2] \cdots [p_{n-1}, X_n, q] \;\Big|\; (p, a, Z, X_1 \cdots X_n, p_0) \in \Delta \text{ mit } n \geq 1 \text{ und } a \in \Sigma \cup \{\varepsilon\},\ p_1, \ldots, p_{n-1}, q \in Q \Big\}$$

Von PDA zu kfG

Beispiel 11.3, Fortsetzung: ein PDA für $\{a^n b^n \mid n \geq 1\}$



Produktionen für (q_0, a, Z_0, ZZ_0, q_0):

$$[q_0, Z_0, q_0] \longrightarrow a\,[q_0, Z, q_0]\,[q_0, Z_0, q_0]$$
$$[q_0, Z_0, q_0] \longrightarrow a\,[q_0, Z, q_1]\,[q_1, Z_0, q_0]$$
$$[q_0, Z_0, q_0] \longrightarrow a\,[q_0, Z, f]\,[f, Z_0, q_0]$$
$$[q_0, Z_0, q_1] \longrightarrow a\,[q_0, Z, q_0]\,[q_0, Z_0, q_1]$$
$$[q_0, Z_0, q_1] \longrightarrow a\,[q_0, Z, q_1]\,[q_1, Z_0, q_1]$$
$$[q_0, Z_0, q_1] \longrightarrow a\,[q_0, Z, f]\,[f, Z_0, q_1]$$
$$[q_0, Z_0, f] \longrightarrow a\,[q_0, Z, q_0]\,[q_0, Z_0, f]$$
$$[q_0, Z_0, f] \longrightarrow a\,[q_0, Z, q_1]\,[q_1, Z_0, f]$$
$$[q_0, Z_0, f] \longrightarrow a\,[q_0, Z, f]\,[f, Z_0, f]$$

Produktionen für (q_0, a, Z, ZZ, q_0):

⟵ *analog*

Produktionen für $(q_0, b, Z, \varepsilon, q_1)$:

$$[q_0, Z, q_1] \longrightarrow b$$

Produktionen für $(q_1, b, Z, \varepsilon, q_1)$:

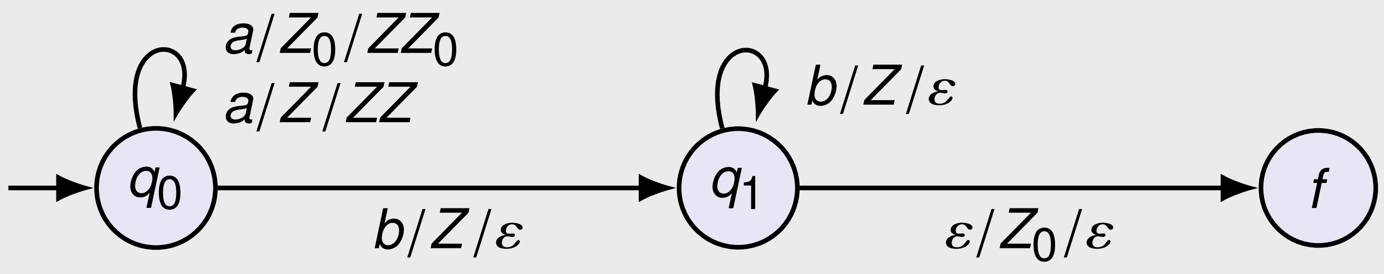
$$[q_1, Z, q_1] \longrightarrow b$$

Produktionen für $(q_1, \varepsilon, Z_0, \varepsilon, f)$:

$$[q_1, Z_0, f] \longrightarrow \varepsilon$$

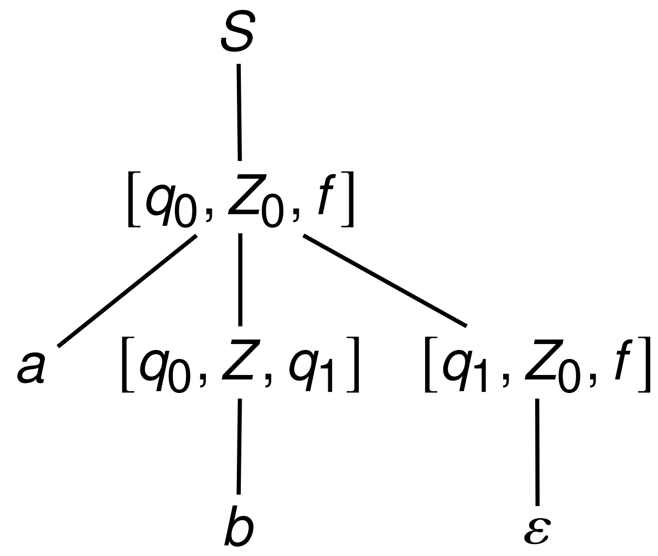
Von PDA zu kfG

Beispiel 11.3, Fortsetzung: ein PDA für $\{a^n b^n \mid n \geq 1\}$



Berechnung von $\mathcal{A}$ $\rightsquigarrow$ zugehörige Ableitung in G

$$
\begin{aligned}
& (q_0, ab, Z_0) \\
& \qquad (q_0, a, Z_0, ZZ_0, q_0) \\
\vdash_{\mathcal{A}}\ & (q_0, b, ZZ_0) \\
& \qquad (q_0, b, Z, \varepsilon, q_1) \\
\vdash_{\mathcal{A}}\ & (q_1, \varepsilon, Z_0) \\
& \qquad (q_1, \varepsilon, Z_0, \varepsilon, f) \\
\vdash_{\mathcal{A}}\ & (f, \varepsilon, \varepsilon)
\end{aligned}
$$



Von PDA zu kfG

Behauptung

Es gilt

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

für alle $p, q \in Q,\ u \in \Sigma^*,\ Z \in \Gamma,\ \gamma \in \Gamma^*$.

Beweis (Idee)

„⇒“ per Induktion über die Länge der Konfigurationsfolge

„⇐“ per Induktion über die Länge der Ableitung

Für $p = q_0,\ Z = Z_0$ und $\gamma = \varepsilon$ folgt daraus:

$$(q_0, u, Z_0) \vdash_{\mathcal{A}}^{*} (q, \varepsilon, \varepsilon) \quad \text{gdw.} \quad S \vdash_G [q_0, Z_0, q] \vdash_G^{*} u$$

und damit:

$$u \in L(\mathcal{A}) \qquad \text{gdw.} \qquad u \in L(G)$$

❑

Zwei Konsequenzen

Wir nennen zwei PDAs $\mathcal{A}$ und $\mathcal{A}'$ äquivalent, wenn $L(\mathcal{A}) = L(\mathcal{A}')$.

Korollar 11.10

1. Zu jedem PDA $\mathcal{A}$ existiert ein äquivalenter PDA $\mathcal{A}'$ mit Akzeptanz per leerem Keller, so dass $\mathcal{A}'$ nur einen Zustand hat.
2. Zu jedem PDA $\mathcal{A}$ existiert ein äquivalenter PDA $\mathcal{A}'$ mit Akzeptanz per Zustand, so dass $\mathcal{A}'$ keine ε-Übergänge hat.

Beweis

Wende erst die Konstruktion aus „2 ⇒ 1“ des vorigen Beweises an, dann die aus „1 ⇒ 2“ (in beiden Variationen).

Resultierender PDA ist äquivalent zum gegebenen und wie behauptet.

❑

Weitere Abschlusseigenschaft der Typ-2-Sprachen

Wir können nun PDAs verwenden,
um interessante Eigenschaften kontextfreier Sprachen zu zeigen.

Satz 11.11

Seien $L \subseteq \Sigma^*$ kontextfrei und $R \subseteq \Sigma^*$ regulär. Dann ist $L \cap R$ kontextfrei.

Beweis.

Seien $\mathcal{A} = (Q, \Sigma, \Gamma, q_0, Z_0, \Delta, F)$ PDA mit akz. Zuständen, $L(\mathcal{A}) = L$
und $\mathcal{A}' = (Q', \Sigma, q_0', \Delta', F')$ NEA mit $L(\mathcal{A}') = R$.

Konstruiere PDA mittels Produktkonstruktion:

$$\mathcal{B} := \Big(Q \times Q',\ \Sigma,\ \Gamma,\ (q_0, q_0'),\ Z_0,\ \Delta'',\ F \times F'\Big) \quad \text{mit}$$

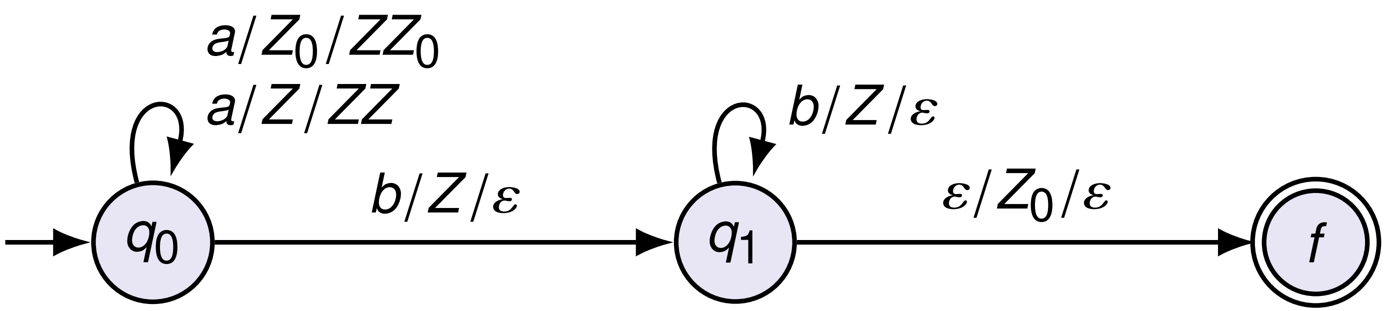
$$\begin{aligned}\Delta'' := &\Big\{\Big((p, p'), a, Z, \gamma, (q, q')\Big) \;\Big|\; (p, a, Z, \gamma, q) \in \Delta \text{ und } (p', a, q') \in \Delta'\Big\} \cup \\ &\Big\{\Big((p, p'), \varepsilon, Z, \gamma, (q, p')\Big) \;\Big|\; (p, \varepsilon, Z, \gamma, q) \in \Delta\Big\}\end{aligned}$$

Man zeigt nun leicht, dass $L(\mathcal{B}) = L \cap R$. ❑

Deterministische Kellerautomaten

Deterministische Kellerautomaten

Betrachte nochmals den PDA für Sprache $\{a^n b^n \mid n \geq 1\}$ aus Beispiel 11.3:

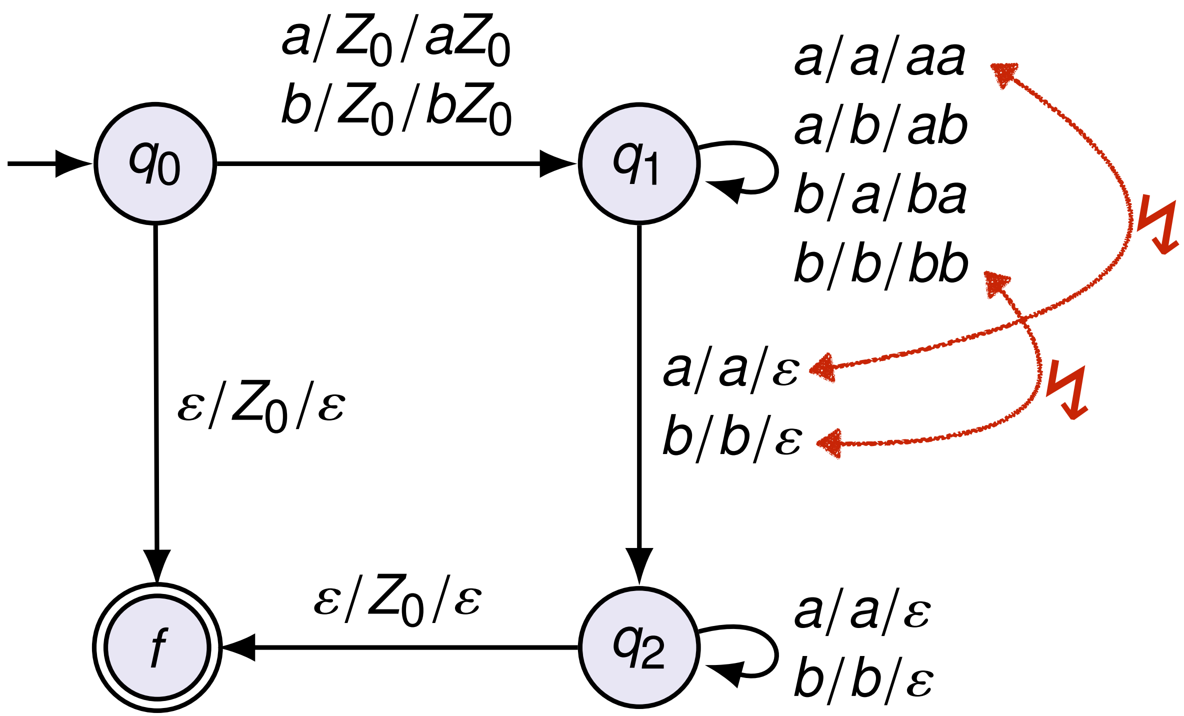


Intuitiv ist dieser Automat deterministisch:

pro Zustand, Eingabesymbol und oberstem Kellersymbol höchstens ein Folgezustand

Deterministische Kellerautomaten

Der PDA für $L = \{ww^R \mid w \in \{a, b\}^*\}$ (Bsp. 11.4) ist nichtdeterministisch:



Zur Erinnerung: Dieser Automat „rät“ nichtdeterministisch die Wortmitte.

Definition dPDA

Definition 11.12 (deterministischer Kellerautomat)

Ein deterministischer Kellerautomat (dPDA) ist ein PDA

$$\mathcal{A} = (Q, \Sigma, \Gamma, q_0, Z_0, \Delta, F),$$

der folgende Eigenschaften erfüllt.

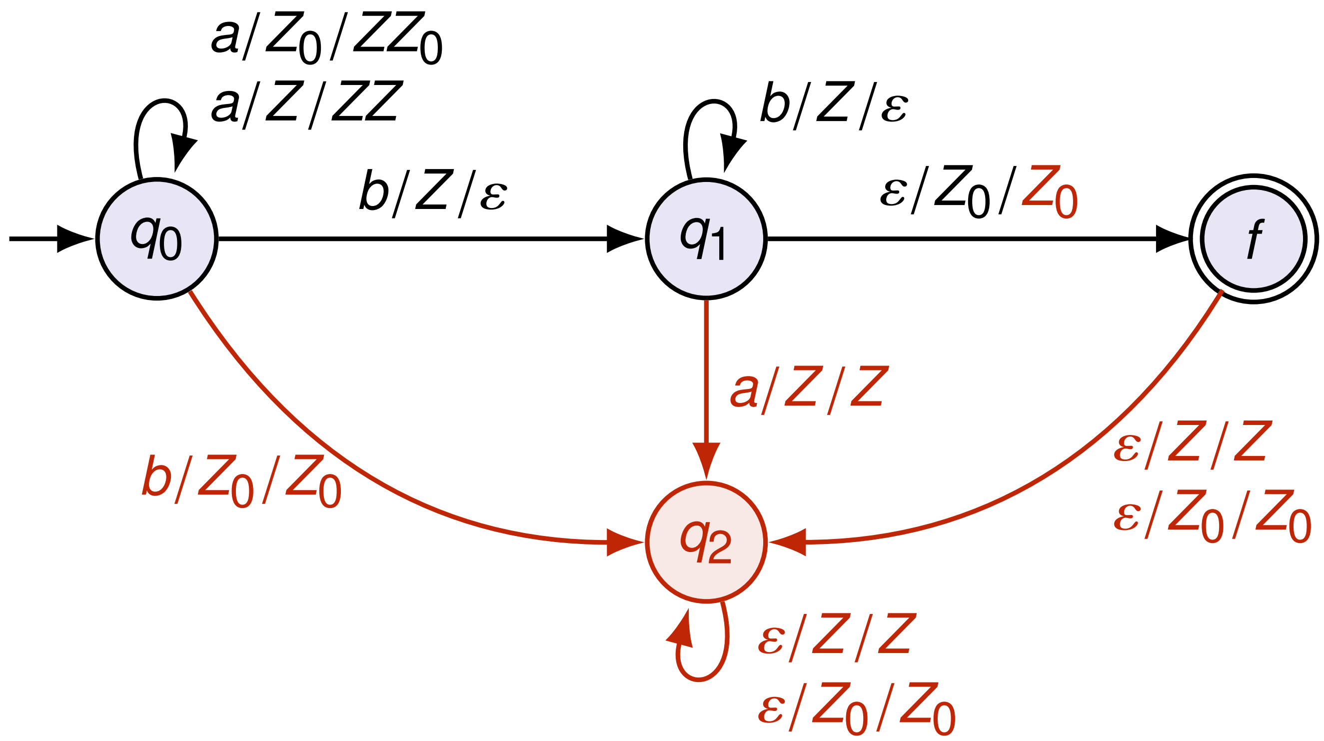
- Für alle $q \in Q$, $a \in \Sigma$ und $Z \in \Gamma$ gibt es genau einen Übergang der Form (q, a, Z, γ, q') oder $(q, \varepsilon, Z, \gamma, q')$.
- Wenn ein Übergang das Kellerstartsymbol Z_0 entfernt, so muss er es direkt wieder zurückschreiben: $(q, a, Z_0, \gamma Z_0, q')$ mit $\gamma \in \Gamma^*$

Akzeptanz definiert wie für PDAs, im Fall von dPDAs ergibt sich:

- zu jeder Konf. (q, w, γ) mit $\gamma \neq \varepsilon$ gibt es genau eine Folgekonfiguration
- insbesondere wird nach Lesen der Eingabe eindeutiger Zustand erreicht
- danach sind eventuell noch beliebig viele ε-Übergange möglich
- dPDA akzeptiert, wenn mit diesen akz. Zustand erreicht werden kann

Ein Beispiel-dPDA

Beispiel 11.13: dPDA für $\{a^n b^n \mid n \geq 1\}$ (Variante des PDA aus 11.3)



Betrachte Eingaben *aabb* *aaba* *ba*

ε-Übergänge

Eine kurze Anmerkung zur Rolle von ε-Übergängen:

- wir haben gesehen: nicht-deterministische PDAs haben auch ohne ε-Übergänge dieselbe Ausdrucksstärke (bei Akzeptanz per Zustand)
- für deterministische PDAs ist das nicht der Fall: dPDAs ohne ε-Übergänge erkennen weniger Sprachen (ohne Beweis)

Zur Akzeptanzbedingung von dPDAs

Nachbemerkung.

Für dPDAs stellt sich Akzeptanz per leerem Keller als sehr schwach heraus:

Lemma 11.14

Es gibt keinen dPDA, der die endliche (also reguläre) Sprache $L = \{a, aa\}$ per leerem Keller erkennt.

(Man würde das Löschen von Z_0 vom Keller dann natürlich erlauben)

Beweis:

- Angenommen, es gäbe doch so einen dPDA
- Da er a akzeptiert, ist nach dem Lesen von a der Keller leer
- Da er deterministisch ist, ist auch nach Lesen der Präfixes a von Eingabe aa der Keller leer

  $\leadsto$ kein weiterer Übergang möglich $\leadsto$ aa wird verworfen ↯

Deterministisch kontextfrei versus kontextfrei

Die interessanteste Frage ist aber natürlich:

Definieren dPDAs und PDAs dieselben Sprachen?

Wir haben am Anfang der VL ja gesehen:

Für DEAs und NEAs ist das der Fall (Potenzmengenkonstruktion)

Es stellt sich heraus, dass das nicht der Fall ist:

dPDAs sind echt schwächer als PDAs

Deterministisch kontextfreie Sprachen

Definition 11.15

Eine formale Sprache L heißt deterministisch kontextfrei,
wenn es einen dPDA $\mathcal{A}$ gibt mit $L(\mathcal{A}) = L$.

Die Menge aller deterministisch kontextfreien Sprachen ist $\mathcal{L}_2^d$.

Satz 11.16

$\mathcal{L}_3 \subset \mathcal{L}_2^d \subseteq \mathcal{L}_2$

Beweis. $\mathcal{L}_3 \subseteq \mathcal{L}_2^d$ Ein DEA kann als dPDA mit nur einem Kellersymbol Z_0 angesehen werden, der seinen Keller nie modifiziert:
aus $\delta(q, a) = q'$ wird Übergang (q, a, Z_0, Z_0, q')

$\mathcal{L}_3 \neq \mathcal{L}_2^d$ $\{a^n b^n \mid n \geq 1\}$ ist nicht regulär, aber in $\mathcal{L}_2^d$

$\mathcal{L}_2^d \subseteq \mathcal{L}_2$ Jeder dPDA $\mathcal{A}$ ist auch PDA, also $L(\mathcal{A}) \in \mathcal{L}_2$

❑

Deterministisch kontextfrei versus kontextfrei

Wie schon gesagt gilt $\mathcal{L}_2^d \subseteq \mathcal{L}_2$

Man kann das ...

1. entweder direkt zeigen durch Angeben einer Sprache in $\mathcal{L}_2 \setminus \mathcal{L}_2^d$
2. oder den Abschluss unter Komplement betrachten.

Beide Wege sind recht technisch, wir wählen Weg 2.

(Beachte: Für Weg 1 müssten wir zeigen, dass eine Sprache nicht in $\mathcal{L}_2^d$ ist)

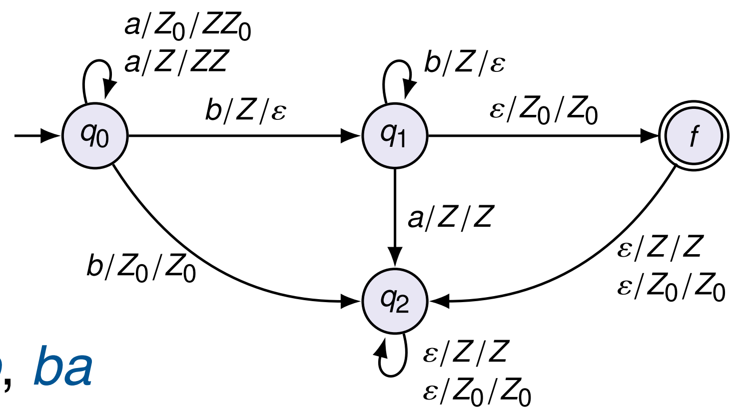
Folgendes Resultat beweisen wir hier nicht, Details im Kozen-Buch

Satz 11.17

$\mathcal{L}_2^d$ ist unter Komplement abgeschlossen.

Einige Ideen:

- Einfaches Vertauschen von akzeptierenden und nicht-akz. Zuständen funktioniert nicht, weil
  1. nach Lesen der Eingabe noch ε-Übergänge möglich sind
  2. Eingabe nicht vollständig gelesen werden muss



z. B. Eingabe *aabb*, *ba*

Abgeschlossenheit unter Komplement

Folgendes Resultat beweisen wir hier nicht, Details im Kozen-Buch

Satz 11.17

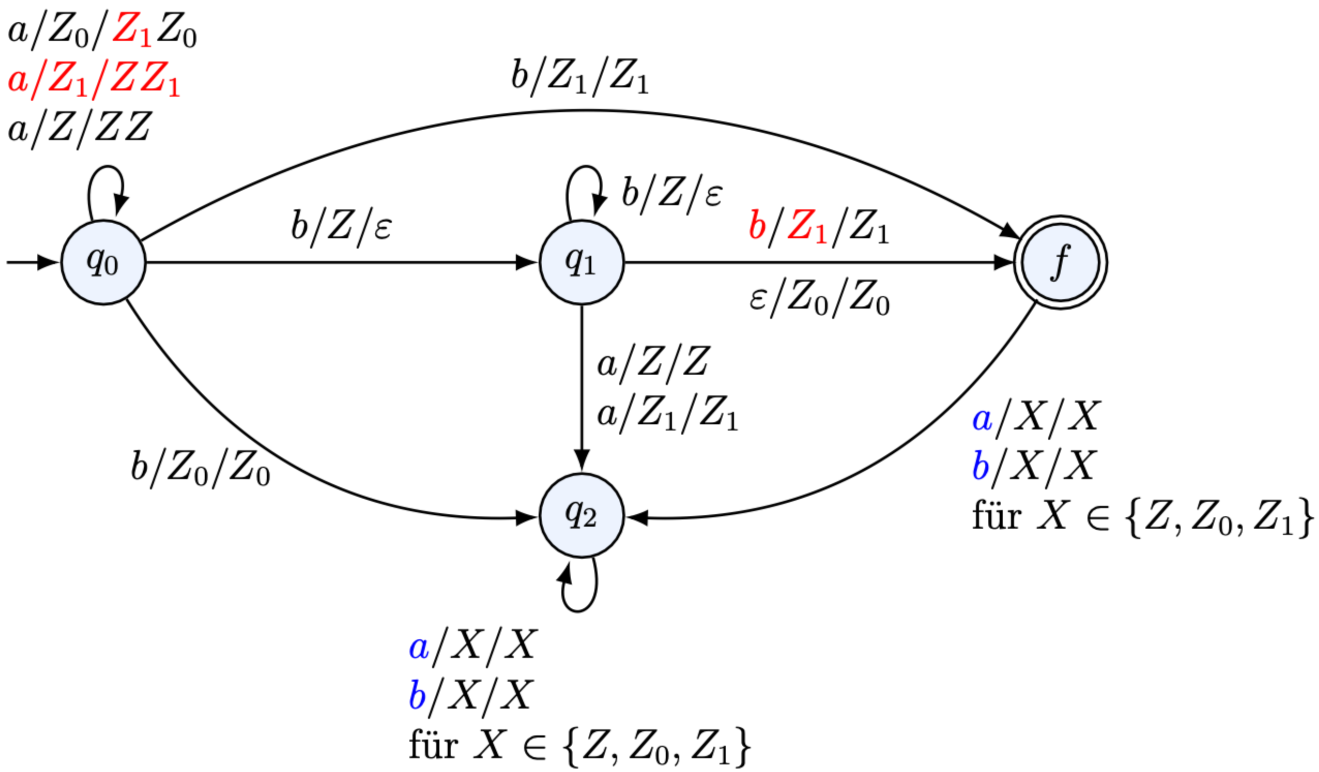
$\mathcal{L}_2^d$ ist unter Komplement abgeschlossen.

Einige Ideen:

- Einfaches Vertauschen von akzeptierenden und nicht-akz. Zuständen funktioniert nicht
  1. nach Lesen der Eingabe noch ε-Übergänge möglich sind
  2. Eingabe nicht vollständig gelesen werden muss
- Daher muss man den dPDA zunächst geeignet modifizieren
- Dabei müssen insbesondere gewisse ε-Übergänge vermieden werden die den Problemen 1+2 zugrunde liegen
- Im Detail ist das recht subtil (zusätzliche Zustände, etc)

Abgeschlossenheit unter Komplement

Exemplarisch: Modifikation des dPDA aus dem vorigen Beispiel



wegen *a*, *b* statt ε wird *ba* nun komplett gelesen (andere Eingaben auch)

wegen Kellersymbol Z_1 bei erstem *a* wird *aabb* ohne zusätzlichen ε-Übergang am Ende akzeptiert

Übersicht Abschlusseigenschaften

	$\cap$	$\cup$	$^{-}$	$\bullet$	$*$
Typ 0					
kontext-sensitiv					
kontextfrei	✗	✓	✗	✓	✓
determ. kontextfrei	✗	✗	✓	✗	✗
regulär	✓	✓	✓	✓	✓

Zurück zu deterministisch k-frei versus k-frei

Satz 11.17

$\mathcal{L}_2^d \subset \mathcal{L}_2$

Beweis, nicht-konstruktiv.

Angenommen $\mathcal{L}_2^d = \mathcal{L}_2$.

Dann ist mit $\mathcal{L}_2^d$ auch $\mathcal{L}_2$ unter Komplement abgeschlossen. $\lightning$

Alternativer Beweis, konstruktiv.

In der Übung wird gezeigt (ÜS 6, Übungsaufgabe 2):

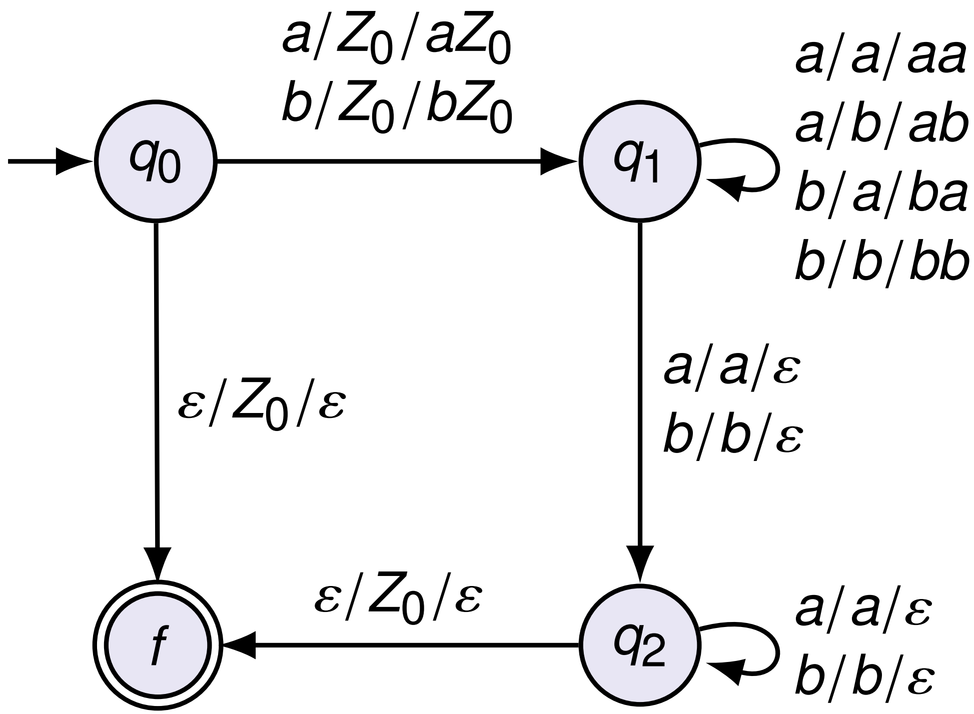
- $L = \{w \mid \forall v \in \{a,b\}^* : w \neq vv\} \in \mathcal{L}_2$,
- $\overline{L} = \{w \mid \exists v \in \{a,b\}^* : w = vv\} \notin \mathcal{L}_2$

Also ist $L \notin \mathcal{L}_2^d$, denn sonst wäre $\overline{L} \in \mathcal{L}_2^d \subseteq \mathcal{L}_2$. ❑

Deterministisch kontextfreie Sprachen sind u. a. im Compilerbau wichtig, da für sie das Wortproblem in Linearzeit gelöst werden kann.

Eine weitere *nicht* deterministisch k-freie Sprache

Zurück zum PDA für $\{ww^R \mid w \in \{a, b\}^*\}$ (Bsp. 11.4):



Man kann zeigen: auch diese Sprache ist nicht deterministisch kontextfrei.

Intuitiv: auf deterministische Weise kann die Wortmitte nicht bestimmt werden.

Die Sprache $\{wcw^R \mid w \in \{a, b\}^*\}$ ist hingegen in $\mathcal{L}_2^d$.

Überblick

Teil I: Endliche Automaten und reguläre Sprachen

0. Grundbegriffe
1. Endliche Automaten
2. Nachweis der Nichterkennbarkeit
3. Abschlusseigenschaften
4. Entscheidungsprobleme
5. Reguläre Ausdrücke und Sprachen
6. Minimale DEAs und die Nerode-Rechtskongruenz

Teil II: Grammatiken, kontextfreie Sprachen und Kellerautomaten

7. Die Chomsky-Hierarchie
8. Rechtslineare Grammatiken und reguläre Sprachen
9. Normalformen und Entscheidungsprobleme
10. Abschlusseigenschaften und Pumping-Lemma
11. Kellerautomaten
12. Die Struktur kontextfreier Sprachen