

# Kapitel 7

## Kellerautomaten

## Abschnitt 7.1

# Definitionen und Beispiele

Wir erweitern endliche Automaten um einen unbeschränkt großen Speicher.

Wir erweitern endliche Automaten um einen unbeschränkt großen Speicher.

Allerdings schränken wir den Zugriff auf den Speicherinhalt ein:

- ▶ Es kann jeweils nur das Symbol aus dem Speicher gelesen werden, das zuletzt eingefügt wurde.

Wir erweitern endliche Automaten um einen unbeschränkt großen Speicher.

Allerdings schränken wir den Zugriff auf den Speicherinhalt ein:

- ▶ Es kann jeweils nur das Symbol aus dem Speicher gelesen werden, das zuletzt eingefügt wurde.
- ▶ Wir sprechen von einem **Kellerspeicher** oder **Stapel** (engl.: **stack** oder manchmal **pushdown stack**).

Wir erweitern endliche Automaten um einen unbeschränkt großen Speicher.

Allerdings schränken wir den Zugriff auf den Speicherinhalt ein:

- ▶ Es kann jeweils nur das Symbol aus dem Speicher gelesen werden, das zuletzt eingefügt wurde.
- ▶ Wir sprechen von einem **Kellerspeicher** oder **Stapel** (engl.: **stack** oder manchmal **pushdown stack**).
- ▶ Das zuletzt eingefügte Symbol liegt dann ganz oben auf dem Stapel. Um auf weiter unten liegende Symbole zugreifen zu können, muss man zunächst die darüber liegenden entfernen.

Wir erweitern endliche Automaten um einen unbeschränkt großen Speicher.

Allerdings schränken wir den Zugriff auf den Speicherinhalt ein:

- ▶ Es kann jeweils nur das Symbol aus dem Speicher gelesen werden, das zuletzt eingefügt wurde.
- ▶ Wir sprechen von einem **Kellerspeicher** oder **Stapel** (engl.: **stack** oder manchmal **pushdown stack**).
- ▶ Das zuletzt eingefügte Symbol liegt dann ganz oben auf dem Stapel. Um auf weiter unten liegende Symbole zugreifen zu können, muss man zunächst die darüber liegenden entfernen.

## Motivation

Man kann einen solchen Kellerspeicher zum Beispiel verwenden, um den Call-Stack eines rekursiven Programms abstrakt zu modellieren.

## Beispiel 7.1 (informell)

Wir betrachten die Sprache

$$\{a^n b^n \mid n \geq 0\}.$$



## Beispiel 7.1 (informell)

Wir betrachten die Sprache

$$\{a^n b^n \mid n \geq 0\}.$$

Ein Kellerautomat für diese Sprache arbeitet wie folgt.

- **Aufbau des Stapels:** Für jedes gelesene  $a$  lege ein Symbol auf dem Stapel ab. (Wir verwenden das Symbol  $\textcolor{brown}{Z}$  für „Zähler“.)

## Beispiel 7.1 (informell)

Wir betrachten die Sprache

$$\{a^n b^n \mid n \geq 0\}.$$

Ein Kellerautomat für diese Sprache arbeitet wie folgt.

- ▶ **Aufbau des Stapels:** Für jedes gelesene  $a$  lege ein Symbol auf dem Stapel ab. (Wir verwenden das Symbol  $Z$  für „Zähler“.)
- ▶ **Abbau des Stapels:** Für jedes gelesene  $b$  nehme ein Symbol  $Z$  vom Stapel herunter.

## Beispiel 7.1 (informell)

Wir betrachten die Sprache

$$\{a^n b^n \mid n \geq 0\}.$$

Ein Kellerautomat für diese Sprache arbeitet wie folgt.

- ▶ **Aufbau des Stapels:** Für jedes gelesene  $a$  lege ein Symbol auf dem Stapel ab. (Wir verwenden das Symbol  $Z$  für „Zähler“.)
- ▶ **Abbau des Stapels:** Für jedes gelesene  $b$  nehme ein Symbol  $Z$  vom Stapel herunter.
- ▶ Zwei Kontrollzustände sorgen dafür, dass Aufbau und Abbau des Stapels nur in dieser Reihenfolge möglich sind (Aufbau mit  $q_0$ , Abbau mit  $q_1$ , keine Rückkehr von  $q_1$  nach  $q_0$ ).

## Beispiel 7.1 (informell)

Wir betrachten die Sprache

$$\{a^n b^n \mid n \geq 0\}.$$

Ein Kellerautomat für diese Sprache arbeitet wie folgt.

- ▶ **Aufbau des Stapels:** Für jedes gelesene  $a$  lege ein Symbol  $Z$  auf dem Stapel ab. (Wir verwenden das Symbol  $Z$  für „Zähler“.)
- ▶ **Abbau des Stapels:** Für jedes gelesene  $b$  nehme ein Symbol  $Z$  vom Stapel herunter.
- ▶ Zwei Kontrollzustände sorgen dafür, dass Aufbau und Abbau des Stapels nur in dieser Reihenfolge möglich sind (Aufbau mit  $q_0$ , Abbau mit  $q_1$ , keine Rückkehr von  $q_1$  nach  $q_0$ ).
- ▶ Akzeptiere, wenn am Ende des Eingabewortes der Stapel vollständig abgebaut ist.

## Definition 7.2

Ein **Kellerautomat** (engl.: **pushdown automaton**, kurz: **PDA**) ist ein 7-Tupel

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

## Definition 7.2

Ein **Kellerautomat** (engl.: **pushdown automaton**, kurz: **PDA**) ist ein 7-Tupel

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

dabei ist

- ▶  $Q$  eine endliche Menge, die **Zustandsmenge**,

## Definition 7.2

Ein **Kellerautomat** (engl.: **pushdown automaton**, kurz: **PDA**) ist ein 7-Tupel

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

dabei ist

- ▶  $Q$  eine endliche Menge, die **Zustandsmenge**,
- ▶  $\Sigma$  ein Alphabet, das **Eingabealphabet**,

## Definition 7.2

Ein **Kellerautomat** (engl.: **pushdown automaton**, kurz: **PDA**) ist ein 7-Tupel

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

dabei ist

- ▶  $Q$  eine endliche Menge, die **Zustandsmenge**,
- ▶  $\Sigma$  ein Alphabet, das **Eingabealphabet**,
- ▶  $\Gamma$  ein Alphabet, das **Stapelalphabet** (oder **Kelleralphabet**),



## Definition 7.2

Ein **Kellerautomat** (engl.: **pushdown automaton**, kurz: **PDA**) ist ein 7-Tupel

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

dabei ist

- ▶  $Q$  eine endliche Menge, die **Zustandsmenge**,
- ▶  $\Sigma$  ein Alphabet, das **Eingabealphabet**,
- ▶  $\Gamma$  ein Alphabet, das **Stapelalphabet** (oder **Kelleralphabet**),
- ▶  $\Delta \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \times Q \times \Gamma^*$  endlich, die **Transitionsrelation**,

## Definition 7.2

Ein **Kellerautomat** (engl.: **pushdown automaton**, kurz: **PDA**) ist ein 7-Tupel

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

dabei ist

- ▶  $Q$  eine endliche Menge, die **Zustandsmenge**,
- ▶  $\Sigma$  ein Alphabet, das **Eingabealphabet**,
- ▶  $\Gamma$  ein Alphabet, das **Stapelalphabet** (oder **Kelleralphabet**),
- ▶  $\Delta \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \times Q \times \Gamma^*$  endlich, die **Transitionsrelation**,
- ▶  $q_0 \in Q$  der **Anfangszustand**,

## Definition 7.2

Ein **Kellerautomat** (engl.: **pushdown automaton**, kurz: **PDA**) ist ein 7-Tupel

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

dabei ist

- ▶  $Q$  eine endliche Menge, die **Zustandsmenge**,
- ▶  $\Sigma$  ein Alphabet, das **Eingabealphabet**,
- ▶  $\Gamma$  ein Alphabet, das **Stapelalphabet** (oder **Kelleralphabet**),
- ▶  $\Delta \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \times Q \times \Gamma^*$  endlich, die **Transitionsrelation**,
- ▶  $q_0 \in Q$  der **Anfangszustand**,
- ▶  $Z_0 \in \Gamma$  das **Stapelanfangssymbol** (oder **Kelleranfangssymbol**),

## Definition 7.2

Ein **Kellerautomat** (engl.: **pushdown automaton**, kurz: **PDA**) ist ein 7-Tupel

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

dabei ist

- ▶  $Q$  eine endliche Menge, die **Zustandsmenge**,
- ▶  $\Sigma$  ein Alphabet, das **Eingabealphabet**,
- ▶  $\Gamma$  ein Alphabet, das **Stapelalphabet** (oder **Kelleralphabet**),
- ▶  $\Delta \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \times Q \times \Gamma^*$  endlich, die **Transitionsrelation**,
- ▶  $q_0 \in Q$  der **Anfangszustand**,
- ▶  $Z_0 \in \Gamma$  das **Stapelanfangssymbol** (oder **Kelleranfangssymbol**),
- ▶  $F \subseteq Q$  die Menge der **Endzustände**.

## Definition 7.2

Ein **Kellerautomat** (engl.: **pushdown automaton**, kurz: **PDA**) ist ein 7-Tupel

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

dabei ist

- ▶  $Q$  eine endliche Menge, die **Zustandsmenge**,
- ▶  $\Sigma$  ein Alphabet, das **Eingabealphabet**,
- ▶  $\Gamma$  ein Alphabet, das **Stapelalphabet** (oder **Kelleralphabet**),
- ▶  $\Delta \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \times Q \times \Gamma^*$  endlich, die **Transitionsrelation**,
- ▶  $q_0 \in Q$  der **Anfangszustand**,
- ▶  $Z_0 \in \Gamma$  das **Stapelanfangssymbol** (oder **Kelleranfangssymbol**),
- ▶  $F \subseteq Q$  die Menge der **Endzustände**.

## Bemerkung 7.3

PDAs sind nichtdeterministisch mit  $\varepsilon$ -Transitionen. Wir werden später auch noch deterministische PDAs betrachten.

- ▶ PDAs bezeichnen wir mit  $\mathcal{A}, \mathcal{B}$  und Varianten wie  $\mathcal{A}', \mathcal{B}_1$ .

- ▶ PDAs bezeichnen wir mit  $\mathcal{A}, \mathcal{B}$  und Varianten wie  $\mathcal{A}', \mathcal{B}_1$ .
- ▶ Großbuchstaben  $X, Y, Z$  und Varianten wie  $Z_0$ , die wir zur besseren Lesbarkeit farbig darstellen, bezeichnen **Stapelsymbole**, also Elemente des Stapelalphabets  $\Gamma$ .

- ▶ PDAs bezeichnen wir mit  $\mathcal{A}, \mathcal{B}$  und Varianten wie  $\mathcal{A}', \mathcal{B}_1$ .
- ▶ Großbuchstaben  $X, Y, Z$  und Varianten wie  $Z_0$ , die wir zur besseren Lesbarkeit farbig darstellen, bezeichnen **Stapelsymbole**, also Elemente des Stapelalphabets  $\Gamma$ .
- ▶ Kleinbuchstaben  $a, b, \dots$  und Varianten bezeichnen **Eingabesymbole**, also Elemente des Eingabealphabets  $\Sigma$ .



- ▶ PDAs bezeichnen wir mit  $\mathcal{A}, \mathcal{B}$  und Varianten wie  $\mathcal{A}', \mathcal{B}_1$ .
- ▶ Großbuchstaben  $X, Y, Z$  und Varianten wie  $Z_0$ , die wir zur besseren Lesbarkeit farbig darstellen, bezeichnen **Stapelsymbole**, also Elemente des Stapelalphabets  $\Gamma$ .
- ▶ Kleinbuchstaben  $a, b, \dots$  und Varianten bezeichnen **Eingabesymbole**, also Elemente des Eingabealphabets  $\Sigma$ .
- ▶ Kleinbuchstaben  $v, w, \dots$  und Varianten bezeichnen Wörter in  $\Sigma^*$ .

- ▶ PDAs bezeichnen wir mit  $\mathcal{A}, \mathcal{B}$  und Varianten wie  $\mathcal{A}', \mathcal{B}_1$ .
- ▶ Großbuchstaben  $X, Y, Z$  und Varianten wie  $Z_0$ , die wir zur besseren Lesbarkeit farbig darstellen, bezeichnen **Stapelsymbole**, also Elemente des Stapelalphabets  $\Gamma$ .
- ▶ Kleinbuchstaben  $a, b, \dots$  und Varianten bezeichnen **Eingabesymbole**, also Elemente des Eingabealphabets  $\Sigma$ .
- ▶ Kleinbuchstaben  $v, w, \dots$  und Varianten bezeichnen Wörter in  $\Sigma^*$ .
- ▶ Griechische Buchstaben  $\gamma, \delta, \dots$  bezeichnen Wörter in  $\Gamma^*$ .

# Wirkung der Transitionen

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

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

Transitionen haben also die Gestalt

$$(p, a, Z, q, \gamma) \quad \text{oder} \quad (p, \varepsilon, Z, q, \gamma)$$

# Wirkung der Transitionen

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

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

Transitionen haben also die Gestalt

$$(p, a, Z, q, \gamma) \quad \text{oder} \quad (p, \varepsilon, Z, q, \gamma)$$

Wirkung der Transition  $(p, a, Z, q, \gamma)$ :

In Zustand  $p$ , bei Eingabesymbol  $a$  und Symbol  $Z$  oben auf dem Stapel,

- nimm  $Z$  vom Stapel, gehe in den Zustand  $q$  über, lege die Symbole aus  $\gamma$  auf den Stapel (in richtiger Reihenfolge, so dass das erste Symbol von  $\gamma$  anschließend oben auf dem Stapel liegt) und gehe zum nächsten Eingabesymbol.

# Wirkung der Transitionen

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

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

Transitionen haben also die Gestalt

$$(p, a, Z, q, \gamma) \quad \text{oder} \quad (p, \varepsilon, Z, q, \gamma)$$

Wirkung der Transition  $(p, a, Z, q, \gamma)$ :

In Zustand  $p$ , bei Eingabesymbol  $a$  und Symbol  $Z$  oben auf dem Stapel,

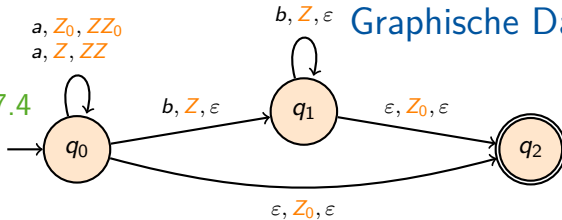
- nimm  $Z$  vom Stapel, gehe in den Zustand  $q$  über, lege die Symbole aus  $\gamma$  auf den Stapel (in richtiger Reihenfolge, so dass das erste Symbol von  $\gamma$  anschließend oben auf dem Stapel liegt) und gehe zum nächsten Eingabesymbol.

Wirkung der Transition  $(p, \varepsilon, Z, q, \gamma)$ :

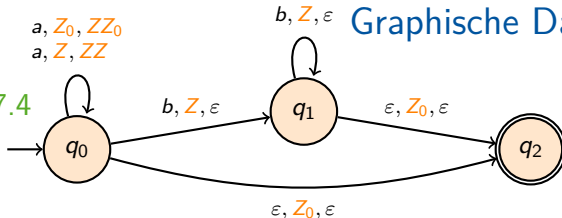
In Zustand  $p$  und bei Symbol  $Z$  oben auf dem Stapel,

- nimm  $Z$  vom Stapel, gehe in den Zustand  $q$  über, lege die Symbole aus  $\gamma$  auf den Stapel.

## Beispiel 7.4

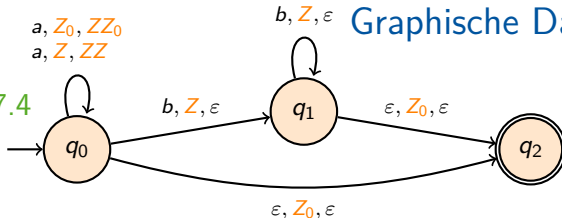


## Beispiel 7.4



Wie endliche Automaten stellen wir PDAs graphisch als Transitionsgraphen dar, deren Knoten die Zustände sind.

## Beispiel 7.4



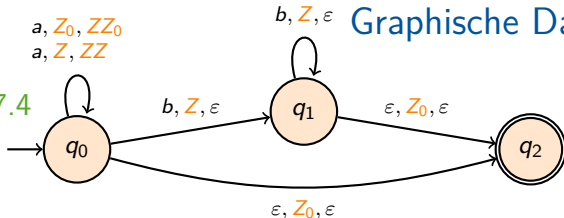
Wie endliche Automaten stellen wir PDAs graphisch als Transitionsgraphen dar, deren Knoten die Zustände sind.

Allerdings sind jetzt Kanten im Graphen nicht mehr mit einzelnen Buchstaben des Eingabealphabets oder „ $\epsilon$ “ beschriftet, sondern mit Tripeln

$$\sigma, Z, \gamma \in (\Sigma \cup \{\epsilon\}) \times \Gamma \times \Gamma^*.$$



## Beispiel 7.4



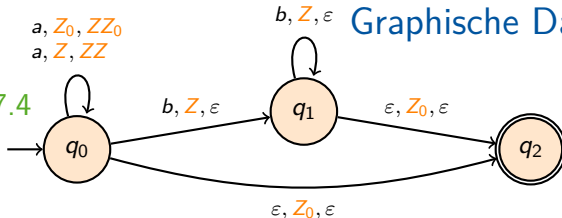
Wie endliche Automaten stellen wir PDAs graphisch als Transitionsgraphen dar, deren Knoten die Zustände sind.

Allerdings sind jetzt Kanten im Graphen nicht mehr mit einzelnen Buchstaben des Eingabealphabets oder „ $\epsilon$ “ beschriftet, sondern mit Tripeln

$$\sigma, Z, \gamma \in (\Sigma \cup \{\epsilon\}) \times \Gamma \times \Gamma^*.$$

Eine mit „ $\sigma, Z, \gamma$ “ beschriftete Kante vom Zustand  $p$  zum Zustand  $q$  zeigt an, dass  $(p, \sigma, Z, q, \gamma) \in \Delta$ .

## Beispiel 7.4



Wie endliche Automaten stellen wir PDAs graphisch als Transitionsgraphen dar, deren Knoten die Zustände sind.

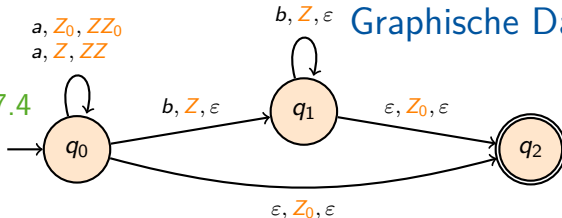
Allerdings sind jetzt Kanten im Graphen nicht mehr mit einzelnen Buchstaben des Eingabealphabets oder „ $\epsilon$ “ beschriftet, sondern mit Tripeln

$$\sigma, Z, \gamma \in (\Sigma \cup \{\epsilon\}) \times \Gamma \times \Gamma^*.$$

Eine mit „ $\sigma, Z, \gamma$ “ beschriftete Kante vom Zustand  $p$  zum Zustand  $q$  zeigt an, dass  $(p, \sigma, Z, q, \gamma) \in \Delta$ .

Mehrere Transitionen zwischen zwei Zuständen fassen wir zusammen; die zugehörigen Tripel  $(\sigma, Z, \gamma)$  schreiben wir übereinander.

## Beispiel 7.4



Wie endliche Automaten stellen wir PDAs graphisch als Transitionsgraphen dar, deren Knoten die Zustände sind.

Allerdings sind jetzt Kanten im Graphen nicht mehr mit einzelnen Buchstaben des Eingabealphabets oder „ $\epsilon$ “ beschriftet, sondern mit Tripeln

$$\sigma, Z, \gamma \in (\Sigma \cup \{\epsilon\}) \times \Gamma \times \Gamma^*.$$

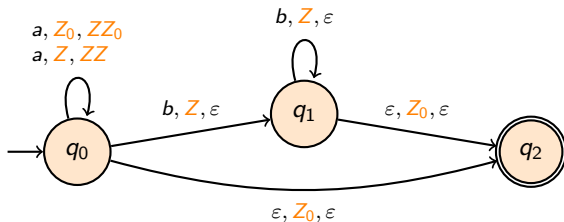
Eine mit „ $\sigma, Z, \gamma$ “ beschriftete Kante vom Zustand  $p$  zum Zustand  $q$  zeigt an, dass  $(p, \sigma, Z, q, \gamma) \in \Delta$ .

Mehrere Transitionen zwischen zwei Zuständen fassen wir zusammen; die zugehörigen Tripel  $(\sigma, Z, \gamma)$  schreiben wir übereinander.

Das Stapelanfangssymbol ist aus dieser Darstellung nicht erkennbar. Wenn nichts anderes gesagt wird, nehmen wir an, es ist  $Z_0$ .

# Graphische Darstellung (Forts.)

## Beispiel 7.4 (Forts.)



beschreibt den PDA

$$(\{q_0, q_1, q_2\}, \{a, b\}, \{Z_0, Z\}, \Delta, q_0, Z_0, \{q_2\})$$

mit

$$\Delta = \{(q_0, a, Z_0, q_0, ZZ_0), (q_0, a, Z, q_0, ZZ), (q_0, b, Z, q_1, \epsilon), \\ (q_0, \epsilon, Z_0, q_2, \epsilon), (q_1, b, Z, q_1, \epsilon), (q_1, \epsilon, Z_0, q_2, \epsilon)\}.$$

## Idee

Eine **Konfiguration** eines Systems soll den „Gesamtzustand“ des Systems zu einem Zeitpunkt einer Berechnung beschreiben. Die in der Konfiguration zusammengefasste Information muss ausreichen, um den weiteren Verlauf der Berechnung festzulegen.

## Idee

Eine **Konfiguration** eines Systems soll den „Gesamtzustand“ des Systems zu einem Zeitpunkt einer Berechnung beschreiben. Die in der Konfiguration zusammengefasste Information muss ausreichen, um den weiteren Verlauf der Berechnung festzulegen.

Für einen PDA besteht dieser „Gesamtzustand“ aus dem Zustand, dem Inhalt des Stapels, und dem verbleibenden Rest des Eingabeworts:

$$\kappa = \left( \underbrace{q}_{\text{Zustand}}, \underbrace{\gamma}_{\text{Stapelinhalt}}, \underbrace{w}_{\text{Rest des Eingabeworts}} \right).$$

## Idee

Eine **Konfiguration** eines Systems soll den „Gesamtzustand“ des Systems zu einem Zeitpunkt einer Berechnung beschreiben. Die in der Konfiguration zusammengefasste Information muss ausreichen, um den weiteren Verlauf der Berechnung festzulegen.

Für einen PDA besteht dieser „Gesamtzustand“ aus dem Zustand, dem Inhalt des Stapels, und dem verbleibenden Rest des Eingabeworts:

$$\kappa = \left( \underbrace{q}_{\text{Zustand}}, \underbrace{\gamma}_{\text{Stapelinhalt}}, \underbrace{w}_{\text{Rest des Eingabeworts}} \right).$$

## Definition 7.5

Eine **Konfiguration** eines PDA  $\mathcal{A} = (Q, \Sigma, \Gamma, \Delta, q_0, Z_0, F)$  ist ein Tripel

$$(q, \gamma, w) \in Q \times \Gamma^* \times \Sigma^*$$

## Definition 7.6

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

1. Die **Übergangsrelation**  $\rightarrow_{\mathcal{A}}$  von  $\mathcal{A}$  ist die folgendermaßen definierte zweistellige Relation auf den Konfigurationen:

$(q, \gamma, w) \rightarrow_{\mathcal{A}} (q', \gamma', w') : \Longleftrightarrow$  es gibt eine Transition

$$(q, \sigma, Z, q', \delta') \in \Delta,$$

und ein  $\delta \in \Gamma^*$ , so dass  $\gamma = Z\delta$  und  $\gamma' = \delta'\delta$  und  $w = \sigma w'$ .



## Definition 7.6

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

1. Die **Übergangsrelation**  $\rightarrow_{\mathcal{A}}$  von  $\mathcal{A}$  ist die folgendermaßen definierte zweistellige Relation auf den Konfigurationen:

$(q, \gamma, w) \rightarrow_{\mathcal{A}} (q', \gamma', w') : \Longleftrightarrow$  es gibt eine Transition

$$(q, \sigma, Z, q', \delta') \in \Delta,$$

und ein  $\delta \in \Gamma^*$ , so dass  $\gamma = Z\delta$  und  $\gamma' = \delta'\delta$  und  $w = \sigma w'$ .

2. Seien  $\kappa, \lambda$  Konfigurationen von  $\mathcal{A}$ . Ein **Lauf von  $\mathcal{A}$  von  $\kappa$  nach  $\lambda$**  ist eine Folge

$$(\kappa_0 = \kappa, \kappa_1, \dots, \kappa_n = \lambda)$$

von Konfigurationen von  $\mathcal{A}$ , so dass  $\kappa_{i-1} \rightarrow_{\mathcal{A}} \kappa_i$  für  $1 \leq i \leq n$ .

# Akzeptierende Läufe und die Sprache eines PDA

## Definition 7.7

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

1. Sei  $w \in \Sigma^*$ . Ein **Lauf von  $\mathcal{A}$  auf  $w$**  ist ein Lauf von  $(q_0, Z_0, w)$  nach  $(q, \gamma, \varepsilon)$ , für ein  $q \in Q$  und  $\gamma \in \Gamma^*$ .

# Akzeptierende Läufe und die Sprache eines PDA

## Definition 7.7

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

1. Sei  $w \in \Sigma^*$ . Ein **Lauf von  $\mathcal{A}$  auf  $w$**  ist ein Lauf von  $(q_0, Z_0, w)$  nach  $(q, \gamma, \varepsilon)$ , für ein  $q \in Q$  und  $\gamma \in \Gamma^*$ .

Der Lauf ist **akzeptierend**, wenn  $q \in F$ .

# Akzeptierende Läufe und die Sprache eines PDA

## Definition 7.7

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

1. Sei  $w \in \Sigma^*$ . Ein **Lauf von  $\mathcal{A}$  auf  $w$**  ist ein Lauf von  $(q_0, Z_0, w)$  nach  $(q, \gamma, \varepsilon)$ , für ein  $q \in Q$  und  $\gamma \in \Gamma^*$ .

Der Lauf ist **akzeptierend**, wenn  $q \in F$ .

2.  $\mathcal{A}$  **akzeptiert**  $w \in \Sigma^*$ , wenn es einen akzeptierenden Lauf von  $\mathcal{A}$  über  $w$  gibt.

# Akzeptierende Läufe und die Sprache eines PDA

## Definition 7.7

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

1. Sei  $w \in \Sigma^*$ . Ein **Lauf von  $\mathcal{A}$  auf  $w$**  ist ein Lauf von  $(q_0, Z_0, w)$  nach  $(q, \gamma, \varepsilon)$ , für ein  $q \in Q$  und  $\gamma \in \Gamma^*$ .

Der Lauf ist **akzeptierend**, wenn  $q \in F$ .

2.  $\mathcal{A}$  **akzeptiert**  $w \in \Sigma^*$ , wenn es einen akzeptierenden Lauf von  $\mathcal{A}$  über  $w$  gibt.

3. Die von  $\mathcal{A}$  **erkannte Sprache** ist

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

# Akzeptierende Läufe und die Sprache eines PDA

## Definition 7.7

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

1. Sei  $w \in \Sigma^*$ . Ein **Lauf von  $\mathcal{A}$  auf  $w$**  ist ein Lauf von  $(q_0, Z_0, w)$  nach  $(q, \gamma, \varepsilon)$ , für ein  $q \in Q$  und  $\gamma \in \Gamma^*$ .

Der Lauf ist **akzeptierend**, wenn  $q \in F$ .

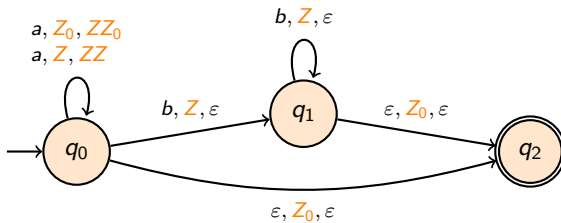
2.  $\mathcal{A}$  **akzeptiert**  $w \in \Sigma^*$ , wenn es einen akzeptierenden Lauf von  $\mathcal{A}$  über  $w$  gibt.

3. Die von  $\mathcal{A}$  **erkannte Sprache** ist

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

4. Eine Sprache  $L$  ist **PDA-erkennbar**, wenn es einen PDA  $\mathcal{A}$  mit  $L(\mathcal{A}) = L$  gibt.

## PDA $\mathcal{A}$

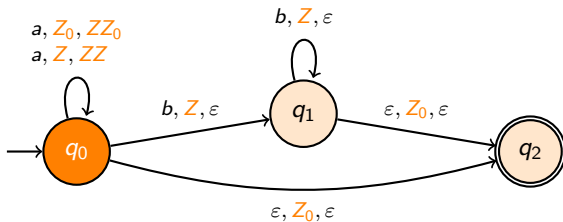


## Eingabewort $w$

$a \ a \ a \ b \ b \ b$

## Beispiel 7.8

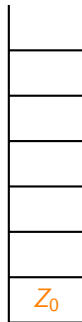
PDA  $\mathcal{A}$



Eingabewort  $w$

$a \ a \ a \ b \ b \ b$   
▲

Stapel



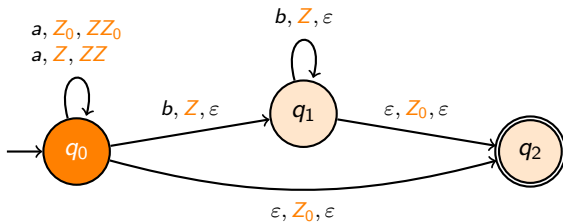
Lauf von  $\mathcal{A}$  auf  $w$

$(q_0, Z_0, aaabbb)$




## Beispiel 7.8

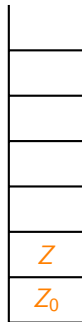
PDA  $\mathcal{A}$



Eingabewort  $w$

$a \quad a \quad a \quad b \quad b \quad b$   


Stapel

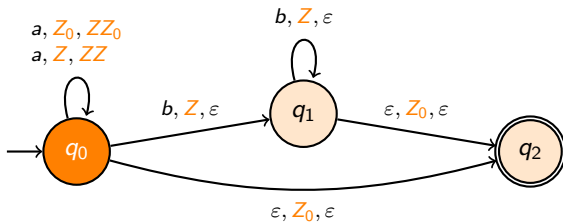


Lauf von  $\mathcal{A}$  auf  $w$

$(q_0, Z_0, aaabbb) \rightarrow (q_0, ZZ_0, aabbb)$

## Beispiel 7.8

PDA  $\mathcal{A}$



Eingabewort  $w$

a a a b b b  
 ▲

Stapel

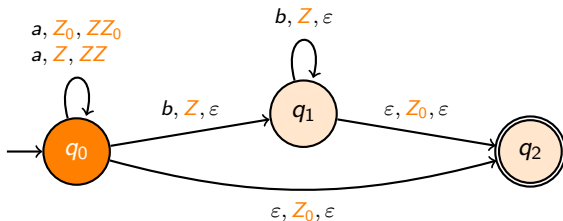


Lauf von  $\mathcal{A}$  auf  $w$


$(q_0, Z_0, aaabbb) \rightarrow (q_0, ZZ_0, aabbb) \rightarrow (q_0, ZZZ_0, abbb)$

## Beispiel 7.8

PDA  $\mathcal{A}$



Eingabewort  $w$

a a a b b b  


Stapel

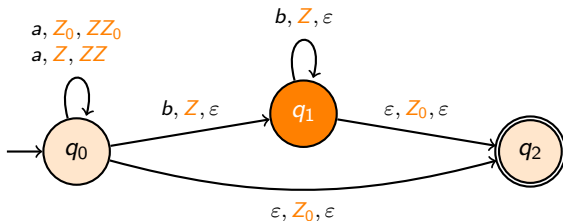


Lauf von  $\mathcal{A}$  auf  $w$

$(q_0, Z_0, aaabbb) \rightarrow (q_0, ZZ_0, aabbb) \rightarrow (q_0, ZZZ_0, abbb) \rightarrow (q_0, ZZZZ_0, bbb)$

## Beispiel 7.8

PDA  $\mathcal{A}$



Eingabewort  $w$

a a a b b b  
          ▲

Stapel

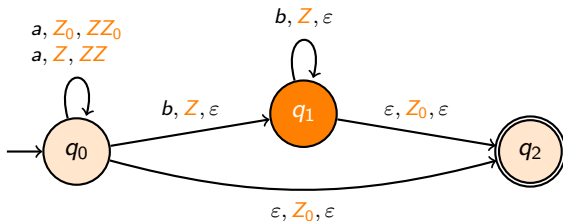


Lauf von  $\mathcal{A}$  auf  $w$

$(q_0, Z_0, aaabbb) \rightarrow (q_0, ZZ_0, aabbb) \rightarrow (q_0, ZZZ_0, abbb) \rightarrow (q_0, ZZZZ_0, bbb)$   
 $\rightarrow (q_1, ZZZ_0, bb)$

## Beispiel 7.8

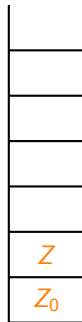
PDA  $\mathcal{A}$



Eingabewort  $w$

$a \ a \ a \ b \ b \ b$   
                  ▲

Stapel

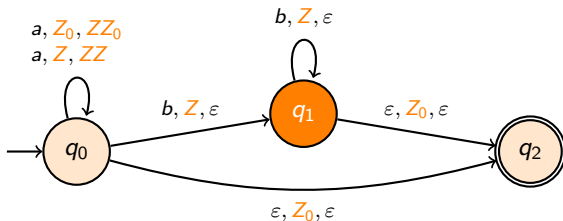


Lauf von  $\mathcal{A}$  auf  $w$

$(q_0, Z_0, aaabbb) \rightarrow (q_0, ZZ_0, aabbb) \rightarrow (q_0, ZZZ_0, abbb) \rightarrow (q_0, ZZZZ_0, bbb)$   
 $\rightarrow (q_1, ZZZ_0, bb) \rightarrow (q_1, ZZ_0, b)$

## Beispiel 7.8

PDA  $\mathcal{A}$

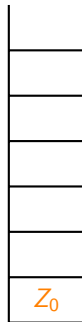


Eingabewort  $w$

$a \ a \ a \ b \ b \ b$



Stapel

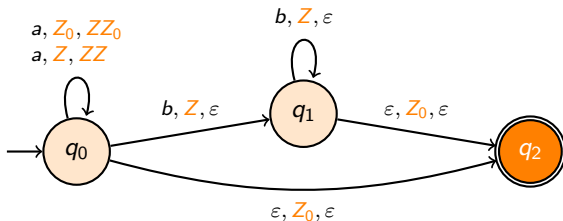


Lauf von  $\mathcal{A}$  auf  $w$

$(q_0, Z_0, aaabbb) \rightarrow (q_0, ZZ_0, aabbb) \rightarrow (q_0, ZZZ_0, abbb) \rightarrow (q_0, ZZZZ_0, bbb)$   
 $\rightarrow (q_1, ZZZ_0, bb) \rightarrow (q_1, ZZ_0, b) \rightarrow (q_1, Z_0, \epsilon)$

## Beispiel 7.8

PDA  $\mathcal{A}$

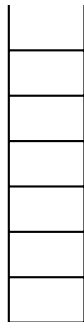


Eingabewort  $w$

$a \ a \ a \ b \ b \ b$



Stapel

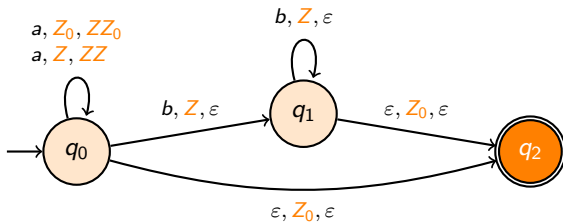


Lauf von  $\mathcal{A}$  auf  $w$

$(q_0, Z_0, aaabbb) \rightarrow (q_0, ZZ_0, aabbb) \rightarrow (q_0, ZZZ_0, abbb) \rightarrow (q_0, ZZZZ_0, bbb)$   
 $\rightarrow (q_1, ZZZ_0, bb) \rightarrow (q_1, ZZ_0, b) \rightarrow (q_1, Z_0, \varepsilon) \rightarrow (q_2, \varepsilon, \varepsilon)$

## Beispiel 7.8

PDA  $\mathcal{A}$

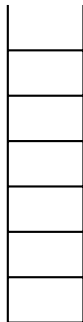


Eingabewort  $w$

$a \ a \ a \ b \ b \ b$



Stapel



$$L(\mathcal{A}) = \{a^n b^n \mid n \geq 0\}.$$



# Terminologie und Notation

Sei  $\mathcal{A}$  ein PDA.

- ▶ Wir schreiben  $\kappa \xrightarrow{*}_{\mathcal{A}} \lambda$ , wenn es einen Lauf von  $\mathcal{A}$  von  $\kappa$  nach  $\lambda$  gibt.

Sei  $\mathcal{A}$  ein PDA.

- ▶ Wir schreiben  $\kappa \xrightarrow{*}_{\mathcal{A}} \lambda$ , wenn es einen Lauf von  $\mathcal{A}$  von  $\kappa$  nach  $\lambda$  gibt.
- ▶ Die **Länge** eines Laufes  $(\kappa_0, \dots, \kappa_n)$  von  $\mathcal{A}$  ist  $n$ .

Wir schreiben  $\kappa \xrightarrow{n}_{\mathcal{A}} \lambda$ , wenn es einen Lauf der Länge  $n$  von  $\kappa$  nach  $\lambda$  gibt.

Weiterhin verwenden wir Notationen wie  $\kappa \xrightarrow{\leq n}_{\mathcal{A}} \lambda$  und  $\kappa \xrightarrow{> n}_{\mathcal{A}} \lambda$ , mit der offensichtlichen Bedeutung.

Sei  $\mathcal{A}$  ein PDA.

- ▶ Wir schreiben  $\kappa \xrightarrow{*}_{\mathcal{A}} \lambda$ , wenn es einen Lauf von  $\mathcal{A}$  von  $\kappa$  nach  $\lambda$  gibt.
- ▶ Die **Länge** eines Laufes  $(\kappa_0, \dots, \kappa_n)$  von  $\mathcal{A}$  ist  $n$ .

Wir schreiben  $\kappa \xrightarrow{n}_{\mathcal{A}} \lambda$ , wenn es einen Lauf der Länge  $n$  von  $\kappa$  nach  $\lambda$  gibt.

Weiterhin verwenden wir Notationen wie  $\kappa \xrightarrow{\leq n}_{\mathcal{A}} \lambda$  und  $\kappa \xrightarrow{> n}_{\mathcal{A}} \lambda$ , mit der offensichtlichen Bedeutung.

- ▶ In all diesen Notationen ( $\rightarrow_{\mathcal{A}}$ ,  $\xrightarrow{*}_{\mathcal{A}}$ , etc.) lassen wir den Index  $\mathcal{A}$  weg, wenn der PDA  $\mathcal{A}$  aus dem Kontext hervorgeht.

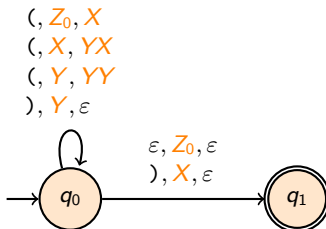
## Beispiel 7.9

Wir betrachten wieder die Sprache  $L_K \subseteq \{ (, ) \}^*$  der korrekten Klammerausdrücke.

## Beispiel 7.9

Wir betrachten wieder die Sprache  $L_K \subseteq \{ (, ) \}^*$  der korrekten Klammerausdrücke.

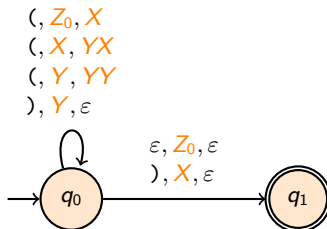
Sei  $\mathcal{A}_K$  folgender PDA:



## Beispiel 7.9

Wir betrachten wieder die Sprache  $L_K \subseteq \{ (, ) \}^*$  der korrekten Klammerausdrücke.

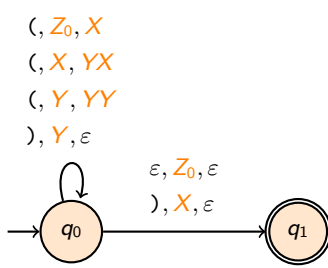
Sei  $\mathcal{A}_K$  folgender PDA:



Akzeptierender Lauf von  $\mathcal{A}$  über dem Wort  $((() ()))$

$$\begin{aligned} & (q_0, Z_0, ((() ())) \rightarrow (q_0, X, () ())) \rightarrow (q_0, YX, ) ()) \rightarrow (q_0, X, ()) \\ & \rightarrow (q_0, YX, )) \rightarrow (q_0, X, ) \rightarrow (q_1, \varepsilon, \varepsilon) \end{aligned}$$

$\mathcal{A}_K$



### Behauptung 1

$$L(\mathcal{A}_K) = L_K.$$

Um die Behauptung zu beweisen, verwenden wir die Charakterisierung von  $L_K$  aus Satz 3.22:

$$L_K = \{w \in \Sigma^* \mid \text{auf}(w) = \text{zu}(w) \text{ und } \text{auf}(v) > \text{zu}(v) \text{ f\"ur alle } v \sqsubseteq w \text{ mit } v \neq \varepsilon\}. \quad (\star)$$

Dabei ist  $\Sigma := \{ (, ) \}$ , und f\"ur alle  $w \in \Sigma^*$

$$\text{auf}(w) := |w|_{(} \quad \text{und} \quad \text{zu}(w) := |w|_{)}.$$

### Behauptung 2

Sei  $w = xy \in \Sigma^*$ .

1. Wenn  $(q_0, X, w) \xrightarrow{*} (q_0, \gamma, y)$  f\"ur ein  $\gamma \in \Gamma^*$ , dann  $\text{auf}(x) \geq \text{zu}(x)$  und  $\gamma = Y^k X$  f\"ur  $k = \text{auf}(x) - \text{zu}(x)$ .
2. Wenn  $\text{auf}(x') \geq \text{zu}(x')$  f\"ur alle  $x' \sqsubseteq x$ , dann  $(q_0, X, w) \xrightarrow{*} (q_0, \gamma, y)$  f\"ur ein  $\gamma \in \Gamma^*$ .

### Beweis von Behauptung 2.

1. Wir zeigen die Behauptung per Induktion \"uber die L\"ange  $n$  eines Laufes von  $\mathcal{A}$  von  $(q_0, X, w)$  nach  $(q_0, \gamma, y)$ .

**Induktionsanfang  $n = 0$ :** Gelte  $(q_0, X, w) \xrightarrow{0} (q_0, \gamma, y)$ . Dann ist  $\gamma = X$  und  $y = w$  und damit  $x = \varepsilon$ . Also  $\text{auf}(x) - \text{zu}(x) = 0$  und  $\gamma = Y^0 X$ .

**Induktionsschritt  $n \rightarrow n + 1$ :** Sei

$$(\kappa_0 = (q_0, X, w), \kappa_1, \dots, \kappa_{n+1} = (q_0, \gamma, y))$$

ein Lauf von  $\mathcal{A}$  von  $(q_0, X, w)$  zu  $(q_0, \gamma, y)$ .

Dann ist  $\kappa_n = (q_0, \gamma', y')$  f\"ur ein  $\gamma' \in \Gamma^*$  und ein Suffix  $y'$  von  $w$ , denn es gibt keine Transition im Zustand  $q_1$ .

Sei  $x' \in \Sigma^*$  so, dass  $x'y' = w$ . Nach Induktionsannahme ist  $\gamma' = Y^{k'} X$ , wobei  $k' = \text{auf}(x') - \text{zu}(x')$ .

**Fall 1:**  $k' = 0$ .

Dann muss die Transition von  $\kappa_n = (q_0, X, y')$  nach  $\kappa_{n+1} = (q_0, \gamma, y)$

$$(q_0, (, X, q_0, YX)$$

sein. Also gilt  $x = x' ($  und  $\gamma = YX$ . Au\"aerdem

$$k = \text{auf}(x) - \text{zu}(x) = \text{auf}(x') - \text{zu}(x') + 1 = k' + 1 = 1$$

und damit  $\gamma = Y^k X$ .

**Fall 2:**  $k' > 0$ .

Dann muss die Transition von  $\kappa_n = (q_0, Y^{k'} X, y')$  nach

$$\kappa_{n+1} = (q_0, \gamma, y)$$

$$(q_0, (, Y, q_0, YY) \quad \text{oder} \quad (q_0, ), Y, q_0, \varepsilon)$$

sein.

**Fall 2a:** Die Transition von  $\kappa_n$  nach  $\kappa_{n+1}$  ist  $(q_0, (, Y, q_0, YY)$ .

Dann gilt  $x = x' ($  und  $\gamma = Y^{k'+1} X$ . Au\"aerdem

$$k = \text{auf}(x) - \text{zu}(x) = \text{auf}(x') - \text{zu}(x') + 1 = k' + 1$$

und damit  $\gamma = Y^k X$ .

**Fall 2b:** Die Transition von  $\kappa_n$  nach  $\kappa_{n+1}$  ist  $(q_0, ), Y, q_0, \varepsilon)$ .

Dann gilt  $x = x' )$  und  $\gamma = Y^{k'-1} X$ . Au\"aerdem

$$k = \text{auf}(x) - \text{zu}(x) = \text{auf}(x') - (\text{zu}(x') + 1) = k' - 1$$

und damit  $\gamma = Y^k X$ .

2. Wir zeigen die Behauptung per Induktion \"uber  $n := |x|$ .

**Induktionsanfang  $n = 0$ :** Dann gilt  $y = w$  und  $(q_0, X, w) \xrightarrow{0} (q_0, X, w)$ .

**Induktionsschritt  $n \rightarrow n + 1$ :**

Sei  $x'$  das Pr\"afix von  $x$  der L\"ange  $|x'| = n$  und  $y'$  das Suffix, so dass  $w = x'y'$ . Nach Induktionsannahme gilt

$$(q_0, X, w) \xrightarrow{*} (q_0, \gamma', y')$$

f\"ur ein  $\gamma' \in \Gamma^*$ .

**Fall 1:**  $x = x' ($ .

Dann gilt  $(q_0, X, w) \xrightarrow{*} (q_0, \gamma', y') \rightarrow (q_0, Y\gamma', y)$ .

**Fall 2:**  $x = x' )$ .

Dann gilt  $k' := \text{auf}(x') - \text{zu}(x') > \text{auf}(x) - \text{zu}(x) \geq 0$ . Nach Teil 1 der Behauptung gilt  $\gamma' = Y^{k'} X$ .

Dann  $(q_0, X, w) \xrightarrow{*} (q_0, \gamma', y') \rightarrow (q_0, Y^{k'-1} X, y)$ .

□

### Beweis von Behauptung 1.

„ $\subseteq$ “: Sei  $w \in L(\mathcal{A}_K)$ . Falls  $w = \varepsilon$ , so ist  $w \in L_K$ .

Im Folgenden nehmen wir an, dass  $|w| \geq 1$ .

Sei  $(\kappa_0, \dots, \kappa_m)$  ein akzeptierender Lauf von  $\mathcal{A}$  auf  $w$ .

Dann gilt:

- (i)  $\kappa_0 = (q_0, Z_0, w)$ .
- (ii)  $\kappa_m = (q_1, \gamma_m, \varepsilon)$  f\"ur ein  $\gamma_m \in \Gamma^*$ .
- (iii)  $m \geq 2$ , denn  $|w| \geq 1$ , und von  $q_0$  nach  $q_1$  gibt es mit Stapelsymbol  $Z_0$  nur eine  $\varepsilon$ -Transition.
- (iv) F\"ur  $1 \leq i \leq m - 1$  ist  $\kappa_i = (q_0, \gamma_i, y_i)$  f\"ur ein  $\gamma_i \in \Gamma^*$  und ein Suffix  $y_i$  von  $w$ . Der Zustand in  $\kappa_i$  muss  $q_0$  sein, weil in  $q_1$  keine Transitionen m\"oglich sind.  
Sei  $x_i \in \Sigma^*$  das Pr\"afix von  $w$ , so dass  $w = x_i y_i$ .
- (v)  $\kappa_{m-1} = (q_0, X, )$ , weil  $(q_0, ), X, q_1, \varepsilon)$  die einzige Transition von  $q_0$  nach  $q_1$  ist.  
Also gilt  $\gamma_{m-1} = X$  und  $y_{m-1} = )$ .
- (vi)  $\kappa_1 = (q_0, X, y_1)$  und  $x_1 = ($ , weil  $m \geq 2$  und weil  $(q_0, (, Z_0, q_0, X)$  die einzige Transition von  $q_0$  nach  $q_0$  mit Stapelsymbol  $Z_0$  ist.  
Das erste Symbol von  $w$  und von allen  $x_i$ s ist also  $($ .
- (vii) F\"ur  $1 \leq i \leq m - 1$  sei  $x'_i \in \Sigma^*$ , so dass  $x_i = (x'_i$ .  
Dann gilt  $(q_0, X, x'_i y_i) \xrightarrow{*} (q_0, \gamma_i, y_i)$ .  
Also  $k_i := \text{auf}(x'_i) - \text{zu}(x'_i) \geq 0$  und  $\gamma_i = Y^{k_i} X$  wegen Behauptung 2(1).
- (viii) Wegen (v) ist  $\gamma_{m-1} = X$ , also gilt  $k_{m-1} = 0$ .

Es gilt  $w = x_{m-1} y_{m-1} = (x'_{m-1})$  nach (v) und damit nach (viii)

$$\text{auf}(w) - \text{zu}(w) = \text{auf}(x'_{m-1}) - \text{zu}(x'_{m-1}) = k_{m-1} = 0,$$

also  $\text{auf}(w) = \text{zu}(w)$ .

Die Pr\"afixe  $v$  von  $w$  mit  $1 \leq |v| < |w|$  sind  $x_i = (x'_i$  f\"ur  $1 \leq i \leq m - 1$ , und nach (vii) gilt

$$\text{auf}(x_i) - \text{zu}(x_i) = 1 + \text{auf}(x'_i) - \text{zu}(x'_i) > 0$$

und damit  $\text{auf}(x_i) > \text{zu}(x_i)$ .

Also gilt  $w \in L_K$  wegen  $(\star)$ .

„ $\supseteq$ “: Sei  $w \in L_K$ . Falls  $w = \varepsilon$  ist  $(q_0, Z_0, \varepsilon) \rightarrow (q_1, \varepsilon, \varepsilon)$  ein akzeptierender Lauf von  $\mathcal{A}$  auf  $w$ .

Nehmen wir an,  $w \neq \varepsilon$ . Dann folgt aus  $(\star)$ , dass  $w = (w')$  f\"ur ein  $w' \in \Sigma^*$ , so dass  $\text{auf}(w') - \text{zu}(w') = 0$  und  $\text{auf}(v') - \text{zu}(v') \geq 0$  f\"ur alle  $v' \sqsubseteq w'$ .

Nach Behauptung 2 gilt dann  $(q_0, X, w') \xrightarrow{*} (q_0, X, \varepsilon)$ . Also

$$(q_0, Z_0, w) \rightarrow (q_0, X, w') \xrightarrow{*} (q_0, X, ) \rightarrow (q_1, \varepsilon, \varepsilon).$$

Also akzeptiert  $\mathcal{A}$  das Wort  $w$ . □

## Beobachtung 7.10

*In einer Konfiguration mit leerem Stapel kann ein PDA keine Transition mehr durchführen.*



## Beobachtung 7.10

*In einer Konfiguration mit leerem Stapel kann ein PDA keine Transition mehr durchführen.*

*Eine solche Konfiguration kann also nur als letzte Konfiguration in einem Lauf auftreten.*

## Beobachtung 7.10

*In einer Konfiguration mit leerem Stapel kann ein PDA keine Transition mehr durchführen.*

*Eine solche Konfiguration kann also nur als letzte Konfiguration in einem Lauf auftreten.*

Wir können es als Akzeptanzkriterium verwenden, ob der Stapel in der letzten Konfiguration eines Laufes leer ist. Wir brauchen dann keine Endzustände.

## Beobachtung 7.10

*In einer Konfiguration mit leerem Stapel kann ein PDA keine Transition mehr durchführen.*

*Eine solche Konfiguration kann also nur als letzte Konfiguration in einem Lauf auftreten.*

Wir können es als Akzeptanzkriterium verwenden, ob der Stapel in der letzten Konfiguration eines Laufes leer ist. Wir brauchen dann keine Endzustände.

## Definition 7.11

Ein **PDA, der mit leerem Stapel akzeptiert**, ist ein Tupel  $(Q, \Sigma, \Gamma, \Delta, q_0, Z_0)$ , wobei die Zustandsmenge  $Q$ , das Eingabealphabet  $\Sigma$ , das Stapelalphabet  $\Gamma$ , der Anfangszustand  $q_0$  und das Stapelanfangssymbol  $Z_0$  wie bei einem PDA definiert sind.

## Beobachtung 7.10

*In einer Konfiguration mit leerem Stapel kann ein PDA keine Transition mehr durchführen.*

*Eine solche Konfiguration kann also nur als letzte Konfiguration in einem Lauf auftreten.*

Wir können es als Akzeptanzkriterium verwenden, ob der Stapel in der letzten Konfiguration eines Laufes leer ist. Wir brauchen dann keine Endzustände.

## Definition 7.11

Ein **PDA**, der mit **leerem Stapel akzeptiert**, ist ein Tupel  $(Q, \Sigma, \Gamma, \Delta, q_0, Z_0)$ , wobei die Zustandsmenge  $Q$ , das Eingabealphabet  $\Sigma$ , das Stapelalphabet  $\Gamma$ , der Anfangszustand  $q_0$  und das Stapelanfangssymbol  $Z_0$  wie bei einem PDA definiert sind.

Für einen PDA, der mit leerem Stapel akzeptiert, definieren wir **Konfigurationen** und **Läufe** wie für einen PDA. Ein Lauf  $(\kappa_0, \dots, \kappa_n)$  ist **akzeptierend**, wenn  $\kappa_n = (q, \varepsilon, \varepsilon)$ , für ein beliebiges  $q \in Q$ .

## Beobachtung 7.10

*In einer Konfiguration mit leerem Stapel kann ein PDA keine Transition mehr durchführen.*

*Eine solche Konfiguration kann also nur als letzte Konfiguration in einem Lauf auftreten.*

Wir können es als Akzeptanzkriterium verwenden, ob der Stapel in der letzten Konfiguration eines Laufes leer ist. Wir brauchen dann keine Endzustände.

## Definition 7.11

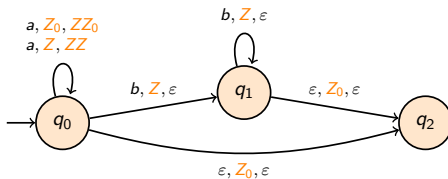
Ein **PDA, der mit leerem Stapel akzeptiert**, ist ein Tupel  $(Q, \Sigma, \Gamma, \Delta, q_0, Z_0)$ , wobei die Zustandsmenge  $Q$ , das Eingabealphabet  $\Sigma$ , das Stapelalphabet  $\Gamma$ , der Anfangszustand  $q_0$  und das Stapelanfangssymbol  $Z_0$  wie bei einem PDA definiert sind.

Für einen PDA, der mit leerem Stapel akzeptiert, definieren wir **Konfigurationen** und **Läufe** wie für einen PDA. Ein Lauf  $(\kappa_0, \dots, \kappa_n)$  ist **akzeptierend**, wenn  $\kappa_n = (q, \varepsilon, \varepsilon)$ , für ein beliebiges  $q \in Q$ .

Jetzt können wir die **erkannte Sprache** wie bei einem PDA definieren.

## Beispiel 7.12

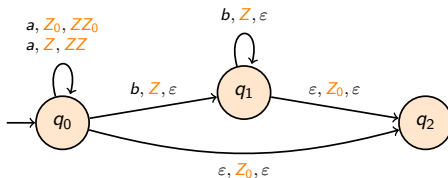
Der PDA (vgl. Beispiel 7.8)



erkennt die Sprache  $\{a^n b^n \mid n \geq 0\}$  mit leerem Stapel.

## Beispiel 7.12

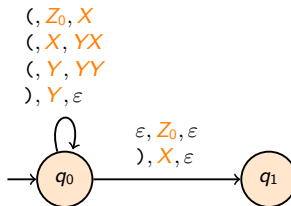
Der PDA (vgl. Beispiel 7.8)



erkennt die Sprache  $\{a^n b^n \mid n \geq 0\}$  mit leerem Stapel.

## Beispiel 7.13

Der PDA (vgl. Beispiel 7.9)



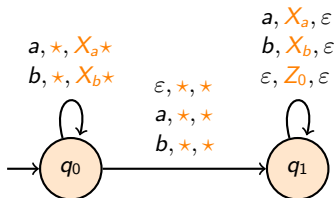
erkennt die Sprache  $L_K$  mit leerem Stapel.

## Beispiel 7.14

Wir betrachten die Sprache der Palindrome über dem Alphabet  $\{a, b\}$ :

$$L_P = \{a_1 \dots a_n \in \{a, b\}^* \mid i \in \mathbb{N}, a_i = a_{n+1-i} \text{ für } 1 \leq i \leq n\}.$$

Folgender PDA  $\mathcal{A}_P$  erkennt  $L_P$  mit leerem Stapel:



Dabei steht in jeder Transition  $\star$  für ein beliebiges Symbol im Stapelalphabet  $\{Z_0, X_a, X_b\}$ .



# Äquivalenz der Akzeptanzbedingungen

## Satz 7.15

*Eine Sprache ist genau dann PDA-erkennbar, wenn sie erkennbar ist durch einen PDA, der mit leerem Stapel akzeptiert.*

(Beweis Übung)

## Queue-Systeme

Wie PDAs, der Speicher ist jedoch als Warteschlange (Queue) organisiert: neue Speicherinhalte werden hinten angefügt; jeweils das erste Symbol kann gelesen werden.

## Queue-Systeme

Wie PDAs, der Speicher ist jedoch als Warteschlange (Queue) organisiert: neue Speicherinhalte werden hinten angefügt; jeweils das erste Symbol kann gelesen werden.

## Turingmaschinen

Speicherzugriff kann an jeder beliebigen Stelle erfolgen.

## Queue-Systeme

Wie PDAs, der Speicher ist jedoch als Warteschlange (Queue) organisiert: neue Speicherinhalte werden hinten angefügt; jeweils das erste Symbol kann gelesen werden.

## Turingmaschinen

Speicherzugriff kann an jeder beliebigen Stelle erfolgen.

Mehr dazu in der Vorlesung [Berechenbarkeit und Komplexität](#).

## Abschnitt 7.2

# Kontextfreie Sprachen und Kellerautomaten

## Satz 7.16

*Eine Sprache ist genau dann PDA-erkennbar, wenn sie kontextfrei ist.*

# Erinnerung: Chomsky-Normalform

Eine kontextfreie Grammatik ist in **Chomsky-Normalform (CNF)**, wenn sie nur Regeln der Gestalt  $A \rightarrow BC$  und  $A \rightarrow a$  für Nichtterminalsymbole  $A, B, C$  und Terminalsymbole  $a$  enthält.

# Erinnerung: Chomsky-Normalform

Eine kontextfreie Grammatik ist in **Chomsky-Normalform (CNF)**, wenn sie nur Regeln der Gestalt  $A \rightarrow BC$  und  $A \rightarrow a$  für Nichtterminalsymbole  $A, B, C$  und Terminalsymbole  $a$  enthält.

## Beispiel 7.17

Folgende Grammatik  $\mathcal{G}$  für die Sprache  $\{a^n b^n \mid n \geq 1\}$  ist in CNF:

$$\begin{aligned} S &\rightarrow AC \mid AB & C &\rightarrow SB \\ A &\rightarrow a \\ B &\rightarrow b \end{aligned}$$



# Konstruktion eines PDA aus einer Grammatik

## Definition 7.18

Sei  $\mathcal{G} = (N, \Sigma, P, S)$  eine Grammatik in CNF.

Wir definieren einen PDA  $\mathcal{A}_{\mathcal{G}} = (Q, \Sigma, \Gamma, \Delta, q_0, Z_0)$ , der mit leerem Stapel akzeptiert, durch:

- ▶  $Q := \{q_0\}$ ,
- ▶  $\Gamma := N$ ,
- ▶  $\Delta := \{(q_0, \varepsilon, A, q_0, BC) \mid A \rightarrow BC \in P\} \cup \{(q_0, a, A, q_0, \varepsilon) \mid A \rightarrow a \in P\}$ ,
- ▶  $Z_0 := S$ .

## Beispiel 7.17 (Forts.)

Wir betrachten wieder die Grammatik  $\mathcal{G}$ :

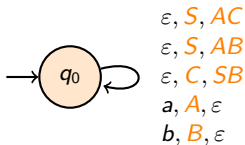
$$S \rightarrow AC \mid AB$$

$$A \rightarrow a$$

$$B \rightarrow b$$

$$C \rightarrow SB$$

$\mathcal{A}_{\mathcal{G}}$  ist folgender PDA mit Stapelanfangssymbol  $S$ :



## Beispiel 7.17 (Forts.)

$\mathcal{G}$

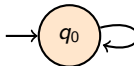
$S \rightarrow AC \mid AB$

$C \rightarrow SB$

$A \rightarrow a$

$B \rightarrow b$

$\mathcal{A}_{\mathcal{G}}$



$\varepsilon, S, AC$

$\varepsilon, S, AB$

$\varepsilon, C, SB$

$a, A, \varepsilon$

$b, B, \varepsilon$

## Beispiel 7.17 (Forts.)

$\mathcal{G}$

$S \rightarrow AC \mid AB$

$C \rightarrow SB$

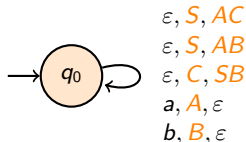
$A \rightarrow a$

$B \rightarrow b$

Ableitung von  $aabb$  in  $\mathcal{G}$

$\rightarrow \underline{S}$   
 $\rightarrow \underline{AC}$   
 $\rightarrow a\underline{C}$   
 $\rightarrow a\underline{SB}$   
 $\rightarrow a\underline{ABB}$   
 $\rightarrow aa\underline{BB}$   
 $\rightarrow aab\underline{B}$   
 $\rightarrow aabb$

$\mathcal{A}_{\mathcal{G}}$



Akz. Lauf von  $\mathcal{A}_{\mathcal{G}}$  auf  $aabb$

$(q_0, S, aabb)$   
 $\rightarrow (q_0, AC, aabb)$   
 $\rightarrow (q_0, C, abb)$   
 $\rightarrow (q_0, SB, abb)$   
 $\rightarrow (q_0, ABB, abb)$   
 $\rightarrow (q_0, BB, bb)$   
 $\rightarrow (q_0, B, b)$   
 $\rightarrow (q_0, \epsilon, \epsilon)$

## Definition 7.19

Ein **Linksableitung** in einer Grammatik ist eine Ableitung, in der in jedem Schritt das am weitesten links stehende Nichtterminalsymbol ersetzt wird.

## Definition 7.19

Ein **Linksableitung** in einer Grammatik ist eine Ableitung, in der in jedem Schritt das am weitesten links stehende Nichtterminalsymbol ersetzt wird.

## Beispiel 7.20

Linksableitung von  $aabb$  in der Grammatik  $\mathcal{G}$  aus Beispiel 7.17:

$$\underline{S} \rightarrow \underline{A}C \rightarrow a\underline{C} \rightarrow a\underline{S}B \rightarrow a\underline{A}BB \rightarrow aa\underline{B}B \rightarrow aab\underline{B} \rightarrow aabb$$

## Definition 7.19

Ein **Linksableitung** in einer Grammatik ist eine Ableitung, in der in jedem Schritt das am weitesten links stehende Nichtterminalsymbol ersetzt wird.

## Beispiel 7.20

Linksableitung von  $aabb$  in der Grammatik  $\mathcal{G}$  aus Beispiel 7.17:

$$\underline{S} \rightarrow \underline{A}C \rightarrow a\underline{C} \rightarrow a\underline{S}B \rightarrow a\underline{A}BB \rightarrow aa\underline{B}B \rightarrow aab\underline{B} \rightarrow aabb$$

## Notation

$$\alpha \xrightarrow{L} \beta, \text{ auch } \alpha \xrightarrow{L}^* \beta, \alpha \xrightarrow{L}^n \beta.$$

## Definition 7.19

Ein **Linksableitung** in einer Grammatik ist eine Ableitung, in der in jedem Schritt das am weitesten links stehende Nichtterminalsymbol ersetzt wird.

## Beispiel 7.20

Linksableitung von  $aabb$  in der Grammatik  $\mathcal{G}$  aus Beispiel 7.17:

$$\underline{S} \rightarrow \underline{A}C \rightarrow a\underline{C} \rightarrow a\underline{S}B \rightarrow a\underline{A}BB \rightarrow aa\underline{B}B \rightarrow aab\underline{B} \rightarrow aabb$$

## Notation

$$\alpha \xrightarrow{L} \beta, \text{ auch } \alpha \xrightarrow{*} \beta, \alpha \xrightarrow[n]{L} \beta.$$

## Beobachtung 7.21

Für alle kontextfreien Grammatiken  $\mathcal{G} = (N, \Sigma, P, S)$ , alle Satzformen  $\alpha, \beta \in (N \cup \Sigma)^*$  und alle  $n \geq 0$  gilt:

$$\alpha \xrightarrow{n} \beta \iff \alpha \xrightarrow[n]{L} \beta.$$



**Satz 7.22**

Sei  $\mathcal{G}$  eine kontextfreie Grammatik in CNF. Dann gilt

$$L(\mathcal{A}_G) = L(\mathcal{G}).$$

**Beweis.**

Sei  $\mathcal{G} = (N, \Sigma, P, S)$ .

Wir zeigen per Induktion über  $n \in \mathbb{N}$ , dass für alle  $x, y \in \Sigma^*$  und  $\beta \in N^*$  gilt:

$$S \xrightarrow[L]{n} x\beta \iff (q_0, S, xy) \xrightarrow{\mathcal{A}_G} (q_0, \beta, y). \quad (*)$$

Daraus folgt sofort die Behauptung des Lemmas.

**Induktionsanfang  $n = 0$ .**

$$\begin{aligned} S &\xrightarrow[L]{0} x\beta \iff x\beta = S \\ &\iff x = \varepsilon \text{ und } \beta = S \\ &\iff (q_0, S, xy) = (q_0, \beta, y) \\ &\iff (q_0, S, xy) \xrightarrow{0} (q_0, \beta, y). \end{aligned}$$

**Induktionsschritt  $n \rightarrow n + 1$ .**

„ $\implies$ “: Gelte  $S \xrightarrow[L]{n+1} x\beta$ . Sei  $\alpha \in (N \cup \Sigma)^*$ , so dass

$$S \xrightarrow[L]{n} \alpha \xrightarrow[L]{} x\beta.$$

**Fall 1:** Die im letzten Ableitungsschritt verwendete Regel ist  $B \rightarrow CD$ .

Weil es sich um eine Linksableitung handelt, ist dann  $\alpha = xB\gamma$  und  $\beta = CD\gamma$  für ein  $\gamma \in N^*$ .

Nach Induktionsannahme gilt

$$(q_0, S, xy) \xrightarrow{n} (q_0, B\gamma, y).$$

Weil  $(q_0, \varepsilon, B, q_0, CD) \in \Delta$  gilt dann

$$(q_0, B\gamma, y) \rightarrow (q_0, CD\gamma, y),$$

also  $(q_0, S, xy) \xrightarrow{n+1} (q_0, \beta, y)$ .

**Fall 2:** Die im letzten Ableitungsschritt verwendete Regel ist  $B \rightarrow b$ .

Weil es sich um eine Linksableitung handelt, ist dann  $\alpha = x'B\beta$  und  $x = x'b$  für ein  $x' \in \Sigma^*$ .

Nach Induktionsannahme gilt

$$(q_0, S, x'by) \xrightarrow{n} (q_0, B\beta, by).$$

Weil  $(q_0, b, B, q_0, \varepsilon) \in \Delta$  gilt dann

$$(q_0, B\beta, by) \rightarrow (q_0, \beta, y),$$

also  $(q_0, S, xy) \xrightarrow{n+1} (q_0, \beta, y)$ .

„ $\impliedby$ “: Gelte  $(q_0, S, xy) \xrightarrow{n+1} (q_0, \beta, y)$ . Seien  $\beta' \in \Gamma^* = N^*$  und  $y' \in \Sigma^*$ , so dass

$$(q_0, S, xy) \xrightarrow{n} (q_0, \beta', y') \rightarrow (q_0, \beta, y).$$

**Fall 1:** Die im letzten Schritt des Laufes verwendete Transition ist

$(q_0, \varepsilon, B, q_0, CD)$ .

Dann ist  $y' = y$  und  $\beta' = B\gamma$  und  $\beta = CD\gamma$  für ein  $\gamma \in \Gamma^*$ .

Nach Induktionsannahme gilt

$$S \xrightarrow[L]{n} xB\gamma,$$

und weil  $xB\gamma \xrightarrow{L} xCD\gamma$  gilt  $S \xrightarrow[L]{n+1} x\beta$ .

**Fall 2:** Die im letzten Schritt des Laufes verwendete Transition ist

$(q_0, b, B, q_0, \varepsilon)$ .

Dann ist  $y' = by$  und  $\beta' = B\beta$ .

Wähle  $x'$  so, dass  $x = x'b$ . Nach Induktionsannahme gilt

$$S \xrightarrow[L]{n} x'B\beta,$$

und weil  $x'B\beta \xrightarrow{L} x\beta$  gilt  $S \xrightarrow[L]{n+1} x\beta$ .

□

## Korollar 7.23

*Jede kontextfreie Sprache ist PDA-erkennbar.*

## Beweis.

Sei  $L$  eine kontextfreie Sprache und  $L' = L \setminus \{\varepsilon\}$ . Sei  $\mathcal{G}$  eine kontextfreie Grammatik in CNF für  $L'$ .

## Korollar 7.23

*Jede kontextfreie Sprache ist PDA-erkennbar.*

### Beweis.

Sei  $L$  eine kontextfreie Sprache und  $L' = L \setminus \{\varepsilon\}$ . Sei  $\mathcal{G}$  eine kontextfreie Grammatik in CNF für  $L'$ .

Satz 7.22  $\implies L' = L(\mathcal{A}_{\mathcal{G}})$ .

## Korollar 7.23

*Jede kontextfreie Sprache ist PDA-erkennbar.*

### Beweis.

Sei  $L$  eine kontextfreie Sprache und  $L' = L \setminus \{\varepsilon\}$ . Sei  $\mathcal{G}$  eine kontextfreie Grammatik in CNF für  $L'$ .

Satz 7.22  $\implies L' = L(\mathcal{A}_{\mathcal{G}})$ .

Falls  $L' = L$  ist damit  $L$  PDA-erkennbar.

## Korollar 7.23

*Jede kontextfreie Sprache ist PDA-erkennbar.*

### Beweis.

Sei  $L$  eine kontextfreie Sprache und  $L' = L \setminus \{\varepsilon\}$ . Sei  $\mathcal{G}$  eine kontextfreie Grammatik in CNF für  $L'$ .

Satz 7.22  $\implies L' = L(\mathcal{A}_{\mathcal{G}})$ .

Falls  $L' = L$  ist damit  $L$  PDA-erkennbar.

Sonst können wir aus  $\mathcal{A}_{\mathcal{G}}$  leicht einen PDA konstruieren, der  $L = L(\mathcal{A}_{\mathcal{G}}) \cup \{\varepsilon\}$  erkennt (Übung). □

# Von PDAs zu Grammatiken: Idee

Zu einem PDA

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

wollen wir eine Grammatik  $\mathcal{G}_{\mathcal{A}}$  konstruieren.

# Von PDAs zu Grammatiken: Idee

Zu einem PDA

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

wollen wir eine Grammatik  $\mathcal{G}_{\mathcal{A}}$  konstruieren.

- ▶ Als Nichtterminalsymbole in  $\mathcal{G}_{\mathcal{A}}$  benutzen wir ein Startsymbol  $S$  und Tripel der Form  $[pZq]$  mit  $p, q \in Q, Z \in \Gamma$ .

# Von PDAs zu Grammatiken: Idee

Zu einem PDA

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

wollen wir eine Grammatik  $\mathcal{G}_{\mathcal{A}}$  konstruieren.

- ▶ Als Nichtterminalsymbole in  $\mathcal{G}_{\mathcal{A}}$  benutzen wir ein Startsymbol  $S$  und Tripel der Form  $[pZq]$  mit  $p, q \in Q, Z \in \Gamma$ .
- ▶ Die Regeln von  $\mathcal{G}$  sollen folgendes leisten

$$[pZq] \xrightarrow{*}_{\mathcal{G}_{\mathcal{A}}} w \iff (p, Z, w) \xrightarrow{*}_{\mathcal{A}} (q, \varepsilon, \varepsilon)$$



# Von PDAs zu Grammatiken: Idee

Zu einem PDA

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

wollen wir eine Grammatik  $\mathcal{G}_{\mathcal{A}}$  konstruieren.

- ▶ Als Nichtterminalsymbole in  $\mathcal{G}_{\mathcal{A}}$  benutzen wir ein Startsymbol  $S$  und Tripel der Form  $[pZq]$  mit  $p, q \in Q, Z \in \Gamma$ .
- ▶ Die Regeln von  $\mathcal{G}$  sollen folgendes leisten

$$[pZq] \xrightarrow{*}_{\mathcal{G}_{\mathcal{A}}} w \iff (p, Z, w) \xrightarrow{*}_{\mathcal{A}} (q, \varepsilon, \varepsilon)$$

# Von PDAs zu Grammatiken: Idee

Zu einem PDA

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

wollen wir eine Grammatik  $\mathcal{G}_{\mathcal{A}}$  konstruieren.

- ▶ Als Nichtterminalsymbole in  $\mathcal{G}_{\mathcal{A}}$  benutzen wir ein Startsymbol  $S$  und Tripel der Form  $[pZq]$  mit  $p, q \in Q, Z \in \Gamma$ .
- ▶ Die Regeln von  $\mathcal{G}$  sollen folgendes leisten

$$[pZq] \xrightarrow{*}_{\mathcal{G}_{\mathcal{A}}} w \iff (p, Z, w) \xrightarrow{*}_{\mathcal{A}} (q, \varepsilon, \varepsilon)$$

Insbesondere:

$$[q_0 Z_0 q] \xrightarrow{*}_{\mathcal{G}_{\mathcal{A}}} w \text{ für ein } q \in Q$$

$$\iff$$

$\mathcal{A}$  akzeptiert  $w$  mit leerem Stapel.

# Von PDAs zu Grammatiken: Idee

Zu einem PDA

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

wollen wir eine Grammatik  $\mathcal{G}_{\mathcal{A}}$  konstruieren.

- ▶ Als Nichtterminalsymbole in  $\mathcal{G}_{\mathcal{A}}$  benutzen wir ein Startsymbol  $S$  und Tripel der Form  $[pZq]$  mit  $p, q \in Q, Z \in \Gamma$ .
- ▶ Die Regeln von  $\mathcal{G}$  sollen folgendes leisten

$$[pZq] \xrightarrow{*}_{\mathcal{G}_{\mathcal{A}}} w \iff (p, Z, w) \xrightarrow{*}_{\mathcal{A}} (q, \varepsilon, \varepsilon)$$

Insbesondere:

$$[q_0 Z_0 q] \xrightarrow{*}_{\mathcal{G}_{\mathcal{A}}} w \text{ f\"ur ein } q \in Q$$

$$\iff$$

$\mathcal{A}$  akzeptiert  $w$  mit leerem Stapel.

- ▶ Wir nehmen also die Regel  $S \rightarrow [q_0 Z_0 q]$  f\"ur jeden Zustand  $q$  hinzu.

$\mathcal{A}$  habe eine Transition  $(p, a, Z, XY, p')$ .

$\mathcal{A}$  habe eine Transition  $(p, a, Z, XY, p')$ .

Dann kann  $\mathcal{A}$  wie folgt von  $p$  nach  $q$  gelangen und dabei  $Z$  vom Stapel entfernen:

$\mathcal{A}$  habe eine Transition  $(p, a, Z, XY, p')$ .

Dann kann  $\mathcal{A}$  wie folgt von  $p$  nach  $q$  gelangen und dabei  $Z$  vom Stapel entfernen:

1.  $\mathcal{A}$  liest das Eingabesymbol  $a$  und das Stapelsymbol  $Z$ , geht in den Zustand  $p'$  über und legt  $XY$  auf den Stapel.

$\mathcal{A}$  habe eine Transition  $(p, a, Z, XY, p')$ .

Dann kann  $\mathcal{A}$  wie folgt von  $p$  nach  $q$  gelangen und dabei  $Z$  vom Stapel entfernen:

1.  $\mathcal{A}$  liest das Eingabesymbol  $a$  und das Stapelsymbol  $Z$ , geht in den Zustand  $p'$  über und legt  $XY$  auf den Stapel.
2.  $\mathcal{A}$  gelangt von  $p'$  zu einem Zwischenzustand  $p''$  und entfernt dabei  $X$  vom Stapel.

$\mathcal{A}$  habe eine Transition  $(p, a, Z, XY, p')$ .

Dann kann  $\mathcal{A}$  wie folgt von  $p$  nach  $q$  gelangen und dabei  $Z$  vom Stapel entfernen:

1.  $\mathcal{A}$  liest das Eingabesymbol  $a$  und das Stapelsymbol  $Z$ , geht in den Zustand  $p'$  über und legt  $XY$  auf den Stapel.
2.  $\mathcal{A}$  gelangt von  $p'$  zu einem Zwischenzustand  $p''$  und entfernt dabei  $X$  vom Stapel.
3.  $\mathcal{A}$  gelangt von  $p''$  nach  $q$  und entfernt dabei  $Y$  vom Stapel.



$\mathcal{A}$  habe eine Transition  $(p, a, Z, XY, p')$ .

Dann kann  $\mathcal{A}$  wie folgt von  $p$  nach  $q$  gelangen und dabei  $Z$  vom Stapel entfernen:

1.  $\mathcal{A}$  liest das Eingabesymbol  $a$  und das Stapelsymbol  $Z$ , geht in den Zustand  $p'$  über und legt  $XY$  auf den Stapel.
2.  $\mathcal{A}$  gelangt von  $p'$  zu einem Zwischenzustand  $p''$  und entfernt dabei  $X$  vom Stapel.
3.  $\mathcal{A}$  gelangt von  $p''$  nach  $q$  und entfernt dabei  $Y$  vom Stapel.

Dieses Verhalten lässt sich in folgenden Regeln zusammenfassen:

$$[pZq] \rightarrow a[p'Xp''] [p''Yq],$$

für alle  $p'' \in Q$ .

$\mathcal{A}$  habe eine Transition  $(p, a, Z, XY, p')$ .

Dann kann  $\mathcal{A}$  wie folgt von  $p$  nach  $q$  gelangen und dabei  $Z$  vom Stapel entfernen:

1.  $\mathcal{A}$  liest das Eingabesymbol  $a$  und das Stapelsymbol  $Z$ , geht in den Zustand  $p'$  über und legt  $XY$  auf den Stapel.
2.  $\mathcal{A}$  gelangt von  $p'$  zu einem Zwischenzustand  $p''$  und entfernt dabei  $X$  vom Stapel.
3.  $\mathcal{A}$  gelangt von  $p''$  nach  $q$  und entfernt dabei  $Y$  vom Stapel.

Dieses Verhalten lässt sich in folgenden Regeln zusammenfassen:

$$[pZq] \rightarrow a[p'Xp''] [p''Yq],$$

für alle  $p'' \in Q$ .

Dies geht analog auch für  $\varepsilon$  statt  $a$ .

## Definition 7.24

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

Die kontextfreie Grammatik  $\mathcal{G}_{\mathcal{A}} = (N, \Sigma, P, S)$  sei wie folgt definiert:

## Definition 7.24

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

Die kontextfreie Grammatik  $\mathcal{G}_{\mathcal{A}} = (N, \Sigma, P, S)$  sei wie folgt definiert:

- ▶  $S$  ist ein neues Symbol.

## Definition 7.24

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

Die kontextfreie Grammatik  $\mathcal{G}_{\mathcal{A}} = (N, \Sigma, P, S)$  sei wie folgt definiert:

- ▶  $S$  ist ein neues Symbol.
- ▶  $N := \{S\} \cup \{[pZq] \mid p, q \in Q, Z \in \Gamma\}$ .

## Definition 7.24

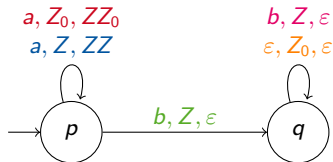
Sei  $\mathcal{A} = (Q, \Sigma, \Gamma, \Delta, q_0, Z_0)$  ein PDA, der mit leerem Stapel akzeptiert.

Die kontextfreie Grammatik  $\mathcal{G}_{\mathcal{A}} = (N, \Sigma, P, S)$  sei wie folgt definiert:

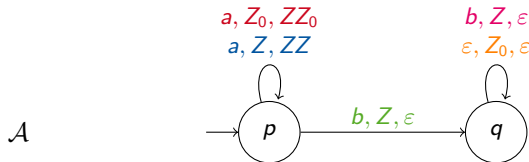
- ▶  $S$  ist ein neues Symbol.
- ▶  $N := \{S\} \cup \{[pZq] \mid p, q \in Q, Z \in \Gamma\}$ .
- ▶  $P := \left\{ [pZp_m] \rightarrow \sigma[p_0Z_1p_1][p_1Z_2p_2] \dots [p_{m-1}Z_mp_m] \mid \right.$   
 $(p, \sigma, Z, p_0, Z_1 \dots Z_m) \in \Delta, m \geq 1, p_1, \dots, p_m \in Q \}$   
 $\cup \left\{ [pZp_0] \rightarrow \sigma \mid (p, \sigma, Z, p_0, \varepsilon) \in \Delta \right\}$ .  
 $\cup \left\{ S \rightarrow [q_0Z_0q] \mid q \in Q \right\}$ .

## Beispiel 7.25

$\mathcal{A}$



## Beispiel 7.25



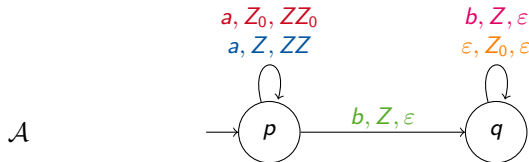
$G_{\mathcal{A}} = (N, \{a, b\}, P, S)$  mit

$$N := \{S, [pZ_0p], [pZ_0q], [pZp], [pZq], [qZ_0p], [qZ_0q], [qZp], [qZq]\}$$

$$P := \left\{ \begin{array}{ll} S & \rightarrow [pZ_0p], \\ S & \rightarrow [pZ_0q], \\ [pZ_0p] & \rightarrow a[pZp][pZ_0p], \\ [pZ_0p] & \rightarrow a[pZq][qZ_0p], \\ [pZ_0q] & \rightarrow a[pZp][pZ_0q], \\ [pZ_0q] & \rightarrow a[pZq][qZ_0q], \\ [pZp] & \rightarrow a[pZp][pZp], \\ [pZp] & \rightarrow a[pZq][qZp], \\ [pZq] & \rightarrow a[pZp][pZq], \\ [pZq] & \rightarrow a[pZq][qZq], \\ [pZq] & \rightarrow b, \\ [qZq] & \rightarrow b, \\ [qZ_0q] & \rightarrow \varepsilon. \end{array} \right\}$$



## Beispiel 7.25



$G_{\mathcal{A}} = (N, \{a, b\}, P, S)$  mit

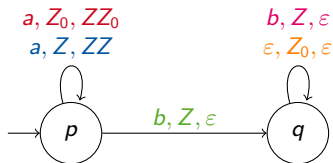
$$N := \{S, [pZ_0p], [pZ_0q], [pZp], [pZq], [qZ_0p], [qZ_0q], [qZp], [qZq]\}$$

$$P := \left\{ \begin{array}{ll} S & \rightarrow [pZ_0p], \\ S & \rightarrow [pZ_0q], \\ [pZ_0p] & \rightarrow a[pZp][pZ_0p], \\ [pZ_0p] & \rightarrow a[pZq][qZ_0p], \\ [pZ_0q] & \rightarrow a[pZp][pZ_0q], \\ [pZ_0q] & \rightarrow a[pZq][qZ_0q], \\ [pZp] & \rightarrow a[pZp][pZp], \\ [pZp] & \rightarrow a[pZq][qZp], \\ [pZq] & \rightarrow a[pZp][pZq], \\ [pZq] & \rightarrow a[pZq][qZq], \\ [pZq] & \rightarrow b, \\ [qZq] & \rightarrow b, \\ [qZ_0q] & \rightarrow \varepsilon. \end{array} \right\}$$

Die grauen Nichtterminale tauchen nie auf der linken Seite einer Regel auf und werden deswegen nicht benötigt. Dadurch fallen auch einige Regeln weg.

## Beispiel 7.25 (Forts.)

$\mathcal{A}$

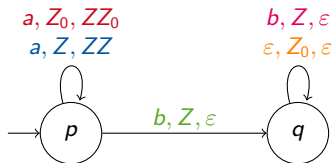


$G_{\mathcal{A}}$  (vereinfacht)

$S \rightarrow [pZ_0p] \mid [pZ_0q],$   
 $[pZ_0p] \rightarrow a[pZp][pZ_0p],$   
 $[pZ_0q] \rightarrow a[pZp][pZ_0q] \mid a[pZq][qZ_0q],$   
 $[pZp] \rightarrow a[pZp][pZp],$   
 $[pZq] \rightarrow a[pZp][pZq] \mid a[pZq][qZq],$   
 $[pZq] \rightarrow b,$   
 $[qZq] \rightarrow b,$   
 $[qZ_0q] \rightarrow \varepsilon.$

## Beispiel 7.25 (Forts.)

$\mathcal{A}$



$G_{\mathcal{A}}$  (vereinfacht)

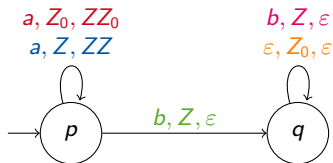
$S \rightarrow [pZ_0p] \mid [pZ_0q],$   
 $[pZ_0p] \rightarrow a[pZp][pZ_0p],$   
 $[pZ_0q] \rightarrow a[pZp][pZ_0q] \mid a[pZq][qZ_0q],$   
 $[pZp] \rightarrow a[pZp][pZp],$   
 $[pZq] \rightarrow a[pZp][pZq] \mid a[pZq][qZq],$   
 $[pZq] \rightarrow b,$   
 $[qZq] \rightarrow b,$   
 $[qZ_0q] \rightarrow \varepsilon.$

Beispiellauf

$(p, Z_0, aabb) \rightarrow (p, ZZ_0, abb)$   
 $\rightarrow (p, ZZZ_0, bb)$   
 $\rightarrow (q, ZZ_0, b)$   
 $\rightarrow (q, Z_0, \varepsilon)$   
 $\rightarrow (q, \varepsilon, \varepsilon)$

## Beispiel 7.25 (Forts.)

$\mathcal{A}$



### Beispiellauf

$(p, Z_0, aabb) \rightarrow (p, ZZ_0, abb)$   
 $\rightarrow (p, ZZZ_0, bb)$   
 $\rightarrow (q, ZZ_0, b)$   
 $\rightarrow (q, Z_0, \varepsilon)$   
 $\rightarrow (q, \varepsilon, \varepsilon)$

### $G_{\mathcal{A}}$ (vereinfacht)

$S \rightarrow [pZ_0p] \mid [pZ_0q],$   
 $[pZ_0p] \rightarrow a[pZp][pZ_0p],$   
 $[pZ_0q] \rightarrow a[pZp][pZ_0q] \mid a[pZq][qZ_0q],$   
 $[pZp] \rightarrow a[pZp][pZp],$   
 $[pZq] \rightarrow a[pZp][pZq] \mid a[pZq][qZq],$   
 $[pZq] \rightarrow b,$   
 $[qZq] \rightarrow b,$   
 $[qZ_0q] \rightarrow \varepsilon.$

### Beispielableitung

$S \rightarrow [pZ_0q] \rightarrow a[pZq][qZ_0q]$   
 $\rightarrow aa[pZq][qZq][qZ_0q]$   
 $\rightarrow aab[qZq][qZ_0q]$   
 $\rightarrow aabb[qZ_0q]$   
 $\rightarrow aabb$

### Satz 7.26

*Sei  $\mathcal{A}$  ein PDA, der mit leerem Stapel akzeptiert. Dann gilt*

$$L(\mathcal{G}_{\mathcal{A}}) = L(\mathcal{A}).$$

### Satz 7.26

*Sei  $\mathcal{A}$  ein PDA, der mit leerem Stapel akzeptiert. Dann gilt*

$$L(\mathcal{G}_A) = L(\mathcal{A}).$$

### Korollar 7.27

*Jede PDA-erkennbare Sprache ist kontextfrei.*

### Satz 7.26

*Sei  $\mathcal{A}$  ein PDA, der mit leerem Stapel akzeptiert. Dann gilt*

$$L(\mathcal{G}_A) = L(\mathcal{A}).$$

### Korollar 7.27

*Jede PDA-erkennbare Sprache ist kontextfrei.*

Damit ist der Äquivalenzsatz 7.16 bewiesen.

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

$\mathcal{G}_{\mathcal{A}}$  hat folgende Regeln:

- ▶  $S \rightarrow [q_0 Z_0 q]$  für alle  $q \in Q$ ,
- ▶  $[pZp_m] \rightarrow \sigma[p_0 Z_1 p_1][p_1 Z_2 p_2] \dots [p_{m-1} Z_m p_m]$  für alle  $(p, \sigma, Z, p_0, Z_1 \dots Z_m) \in \Delta$ , wobei  $m \geq 1$ , und  $p_1, \dots, p_m \in Q$ .
- ▶  $[pZq] \rightarrow \sigma$  für alle  $(p, \sigma, Z, q, \varepsilon) \in \Delta$ .

## Behauptung 1

Für alle  $n \geq 1$ ,  $p, q \in Q$ ,  $Z \in \Gamma$  und  $w \in \Sigma^*$  gilt

$$(p, Z, w) \xrightarrow{n}_{\mathcal{A}} (q, \varepsilon, \varepsilon) \implies [pZq] \xrightarrow{*}_{\mathcal{G}_{\mathcal{A}}} w.$$

**Beweis.**

Induktion über  $n$ .

**Induktionsanfang  $n = 1$ :**

Gelte  $(p, Z, w) \rightarrow (q, \varepsilon, \varepsilon)$ .

Dann ist  $w \in \Sigma \cup \{\varepsilon\}$  und  $(p, w, Z, q, \varepsilon) \in \Delta$ .

Also  $[pZq] \rightarrow w \in P$ , und es gilt  $[pZq] \xrightarrow{1} w$ .

**Induktionsschritt  $1, \dots, n \rightarrow n + 1$ :**

Gelte  $(p, Z, w) \xrightarrow{n+1}_{\mathcal{A}} (q, \varepsilon, \varepsilon)$ .

Sei  $(p, \sigma, Z, Z_1 \dots Z_m, p_0)$  die im ersten Schritt verwendete Transition. Es gilt  $m \geq 1$ , denn sonst wäre der Stapel nach dem ersten Schritt leer und der Lauf würde abbrechen.

Sei  $w' \in \Sigma^*$ , so dass  $w = \sigma w'$ .

Dann gilt

$$(p, Z, w) \rightarrow (p_0, Z_1 \dots Z_m, w') \xrightarrow{n}_{\mathcal{A}} (q, \varepsilon, \varepsilon).$$

Also gibt es  $w_1, \dots, w_m \in \Sigma^*$  und  $p_1, \dots, p_m \in Q$ , so dass  $w' = w_1 \dots w_m$  und  $p_m = q$  und

$$(p_{i-1}, Z_i, w_i) \xrightarrow{n_i}_{\mathcal{A}} (p_i, \varepsilon, \varepsilon)$$

für geeignete  $n_i \leq n$ .

Nach Induktionsannahme gilt  $[p_{i-1} Z_i p_i] \xrightarrow{*}_{\mathcal{G}_{\mathcal{A}}} w_i$ .

Also nach dem Kombinationslemma 6.9 gilt

$$[pZq] \rightarrow \sigma[p_0 Z_1 p_1] \dots [p_{m-1} Z_m p_m] \xrightarrow{*}_{\mathcal{G}_{\mathcal{A}}} \sigma w_1 \dots w_m = w.$$

Damit ist die Behauptung bewiesen. □

## Behauptung 2

Für alle  $n \geq 1$ ,  $p, q \in Q$ ,  $Z \in \Gamma$  und  $w \in \Sigma^*$  gilt

$$[pZq] \xrightarrow{n}_{\mathcal{G}_{\mathcal{A}}} w \implies (p, Z, w) \xrightarrow{*}_{\mathcal{A}} (q, \varepsilon, \varepsilon).$$

**Beweis.**

Induktion über  $n$ .

Ähnlich wie der Beweis von Behauptung 1. □

## Beweis des Satzes.

“ $L(\mathcal{A}) \subseteq L(\mathcal{G}_{\mathcal{A}})$ ”: Sei  $w \in \Sigma^*$ .

$$\begin{aligned} \mathcal{A} \text{ akzeptiert } w &\implies \text{ es gibt ein } q \text{ mit } (q_0, Z_0, w) \xrightarrow{*}_{\mathcal{A}} (q, \varepsilon, \varepsilon) \\ &\implies S \rightarrow [q_0 Z_0 q] \xrightarrow{*}_{\mathcal{G}_{\mathcal{A}}} w \quad (\text{nach Behauptung 1}) \\ &\implies w \in L(\mathcal{G}_{\mathcal{A}}). \end{aligned}$$

“ $L(\mathcal{A}) \supseteq L(\mathcal{G}_{\mathcal{A}})$ ”: Sei  $w \in \Sigma^*$ .

$$\begin{aligned} S \xrightarrow{*}_{\mathcal{G}_{\mathcal{A}}} w &\implies \text{ es gibt ein } q \text{ mit } S \rightarrow_{\mathcal{G}_{\mathcal{A}}} [q_0 Z_0 q] \xrightarrow{*}_{\mathcal{G}_{\mathcal{A}}} w \\ &\implies (q_0, Z_0, w) \xrightarrow{*}_{\mathcal{A}} (q, \varepsilon, \varepsilon) \quad (\text{nach Behauptung 2}) \\ &\implies \mathcal{A} \text{ akzeptiert } w. \end{aligned}$$

□



## Abschnitt 7.3

# Deterministische Kellerautomaten

## Definition 7.28

1. Ein PDA  $\mathcal{A} = (Q, \Sigma, \Gamma, \Delta, q_0, Z_0, F)$  ist **deterministisch** (kurz: ein **DPDA**), wenn in jeder Konfiguration höchstens eine Transition anwendbar ist.

## Definition 7.28

1. Ein PDA  $\mathcal{A} = (Q, \Sigma, \Gamma, \Delta, q_0, Z_0, F)$  ist **deterministisch** (kurz: ein **DPDA**), wenn in jeder Konfiguration höchstens eine Transition anwendbar ist.

Das heißt, für alle  $q \in Q$ ,  $a \in \Sigma$  und  $Z \in \Gamma$  trifft höchstens eine der folgenden beiden Möglichkeiten zu:

## Definition 7.28

1. Ein PDA  $\mathcal{A} = (Q, \Sigma, \Gamma, \Delta, q_0, Z_0, F)$  ist **deterministisch** (kurz: ein **DPDA**), wenn in jeder Konfiguration höchstens eine Transition anwendbar ist.

Das heißt, für alle  $q \in Q$ ,  $a \in \Sigma$  und  $Z \in \Gamma$  trifft höchstens eine der folgenden beiden Möglichkeiten zu:

- es gibt genau ein  $r \in Q$  und genau ein  $\gamma \in \Gamma^*$ , so dass  $(q, a, Z, r, \gamma) \in \Delta$ ;

## Definition 7.28

1. Ein PDA  $\mathcal{A} = (Q, \Sigma, \Gamma, \Delta, q_0, Z_0, F)$  ist **deterministisch** (kurz: ein **DPDA**), wenn in jeder Konfiguration höchstens eine Transition anwendbar ist.

Das heißt, für alle  $q \in Q$ ,  $a \in \Sigma$  und  $Z \in \Gamma$  trifft höchstens eine der folgenden beiden Möglichkeiten zu:

- ▶ es gibt genau ein  $r \in Q$  und genau ein  $\gamma \in \Gamma^*$ , so dass  $(q, a, Z, r, \gamma) \in \Delta$ ;
- ▶ es gibt genau ein  $r \in Q$  und genau ein  $\gamma \in \Gamma^*$ , so dass  $(q, \varepsilon, Z, r, \gamma) \in \Delta$ .

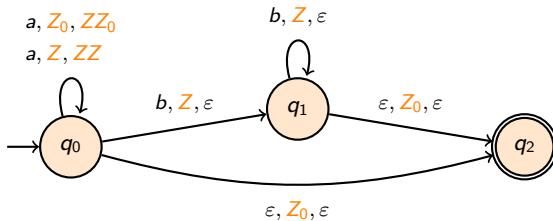
## Definition 7.28

1. Ein PDA  $\mathcal{A} = (Q, \Sigma, \Gamma, \Delta, q_0, Z_0, F)$  ist **deterministisch** (kurz: ein **DPDA**), wenn in jeder Konfiguration höchstens eine Transition anwendbar ist.

Das heißt, für alle  $q \in Q$ ,  $a \in \Sigma$  und  $Z \in \Gamma$  trifft höchstens eine der folgenden beiden Möglichkeiten zu:

- ▶ es gibt genau ein  $r \in Q$  und genau ein  $\gamma \in \Gamma^*$ , so dass  $(q, a, Z, r, \gamma) \in \Delta$ ;
  - ▶ es gibt genau ein  $r \in Q$  und genau ein  $\gamma \in \Gamma^*$ , so dass  $(q, \varepsilon, Z, r, \gamma) \in \Delta$ .
2. Eine Sprache  $L$  ist **DPDA-erkennbar**, wenn es einen DPDA gibt, der  $L$  erkennt.

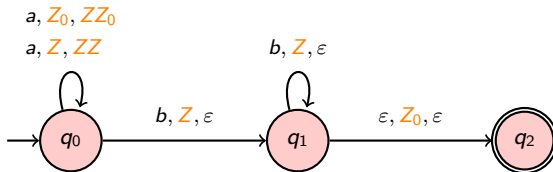
Wir betrachten noch einmal den PDA



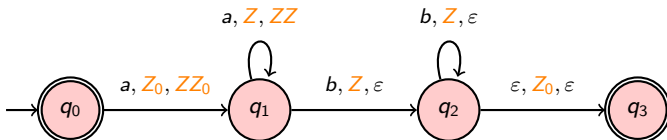
aus Beispiel 7.4, der die Sprache  $\{a^n b^n \mid n \geq 0\}$  erkennt.

Dieser PDA ist kein DPDA, weil in der Konfiguration  $(q_0, Z_0, a)$  zwei Transitionen möglich sind.

Folgende Variante des PDA ist ein DPDA; er erkennt die Sprache  $\{a^n b^n \mid n \geq 1\}$ :



Ein DPDA für die Sprache  $\{a^n b^n \mid n \geq 0\}$  erhalten wir durch Hinzufügen eines neuen Zustands:



## Satz 7.30

1. *Jede DPDA-erkennbare Sprache ist kontextfrei.*



## Satz 7.30

1. *Jede DPDA-erkennbare Sprache ist kontextfrei.*
2. *Jede reguläre Sprache ist DPDA-erkennbar.*

## Satz 7.30

1. *Jede DPDA-erkennbare Sprache ist kontextfrei.*
2. *Jede reguläre Sprache ist DPDA-erkennbar.*
3. *Es gibt DPDA-erkennbare Sprachen, die nicht regulär sind.*

## Satz 7.30

1. *Jede DPDA-erkennbare Sprache ist kontextfrei.*
2. *Jede reguläre Sprache ist DPDA-erkennbar.*
3. *Es gibt DPDA-erkennbare Sprachen, die nicht regulär sind.*
4. *Es gibt kontextfreie Sprachen, die nicht DPDA-erkennbar sind.*

1. *Jede DPDA-erkennbare Sprache ist kontextfrei.*  
Jeder DPDA ist ein PDA, also ist auch jede DPDA-erkennbare Sprache PDA-erkennbar und damit kontextfrei.

1. *Jede DPDA-erkennbare Sprache ist kontextfrei.*

Jeder DPDA ist ein PDA, also ist auch jede DPDA-erkennbare Sprache PDA-erkennbar und damit kontextfrei.

2. *Jede reguläre Sprache ist DPDA-erkennbar.*

Jeder DFA lässt sich durch eine DPDA simulieren, der den Stapel nicht verwendet.

1. *Jede DPDA-erkennbare Sprache ist kontextfrei.*  
Jeder DPDA ist ein PDA, also ist auch jede DPDA-erkennbare Sprache PDA-erkennbar und damit kontextfrei.
2. *Jede reguläre Sprache ist DPDA-erkennbar.*  
Jeder DFA lässt sich durch eine DPDA simulieren, der den Stapel nicht verwendet.
3. *Es gibt DPDA-erkennbare Sprachen, die nicht regulär sind.*  
Die Sprache  $\{a^n b^n \mid n \geq 0\}$  ist ein Beispiel.

1. *Jede DPDA-erkennbare Sprache ist kontextfrei.*  
Jeder DPDA ist ein PDA, also ist auch jede DPDA-erkennbare Sprache PDA-erkennbar und damit kontextfrei.
2. *Jede reguläre Sprache ist DPDA-erkennbar.*  
Jeder DFA lässt sich durch eine DPDA simulieren, der den Stapel nicht verwendet.
3. *Es gibt DPDA-erkennbare Sprachen, die nicht regulär sind.*  
Die Sprache  $\{a^n b^n \mid n \geq 0\}$  ist ein Beispiel.
4. *Es gibt kontextfreie Sprachen, die nicht DPDA-erkennbar sind.*  
Die Sprache  $L_P$  der Palindrome über  $\{a, b\}$  ist ein Beispiel (ohne Beweis).