Formale Sprachen und Komplexitätstheorie

WS 2019/20

Robert Elsässer

Eingeschränkte Grammatiken

Definition

- Eine Grammatik heißt kontextsensitiv oder vom Typ Chomsky-1, falls für jede Regel $u \to v$ gilt: $|u| \leq |v|$.
- Eine Grammatik heißt kontextfrei oder vom Typ Chomsky-2, falls für jede Regel $u \to v$ gilt: $u \in V$.
- Eine Regel heißt regulär oder vom Typ Chomsky-3, falls alle Regeln der Art $u \to v$ mit $u \in V$ und:
  - $v = \varepsilon$
  - $v = a$, $a \in \Sigma$ oder
  - $v = aw$ mit $a \in \Sigma$ und $w \in V$

  sind.

Kontextfrei vs. regulär

- wir betrachten $L = \{0^n 1^n \mid n \geq 1\}$
- L ist nicht regulär (*Pumping Lemma* für reguläre Sprachen – siehe VO „Formale Systeme“)
- L ist kontextfrei, Grammatik gegeben z.B. durch

$$1.\ S \to 0S1$$

$$2.\ S \to 01$$

Nichtdeterministische Automaten

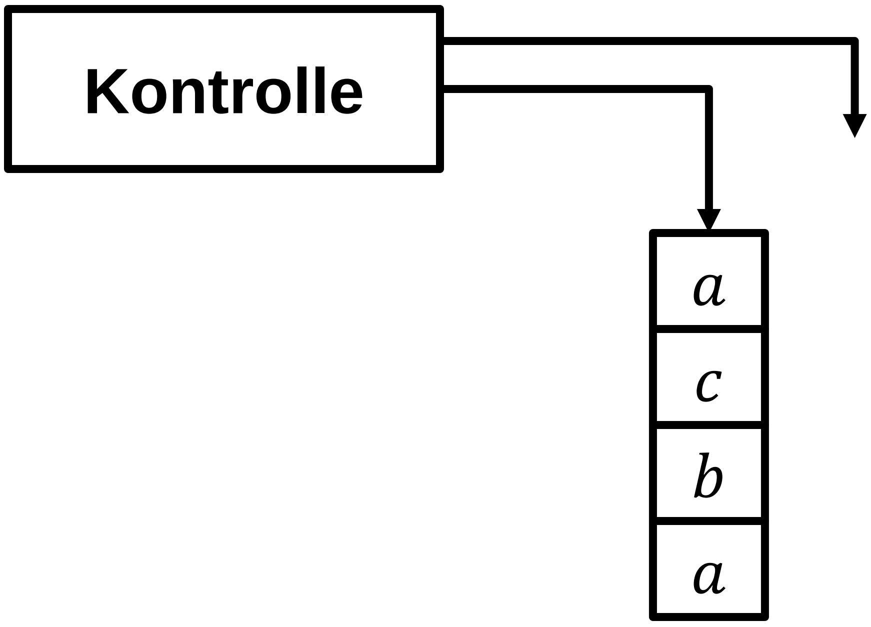
Definition

Ein nichtdeterministischer endlicher Automat $A = (Q, \Sigma, \delta, q_0, F)$ besteht aus:

- einer endlichen Menge von Eingabesymbolen Σ
- einer endlichen Menge von Zuständen Q
- einer Übergangsfunktion $\delta: Q \times \Sigma \to \mathcal{P}(Q)$
- einem Anfangszustand $q_0 \in Q$
- einer Menge $F \subseteq Q$ von Endzuständen

Kellerautomaten und Stapel

- Kellerautomaten sind nichtdeterministische Automaten mit zusätzlichem **Stapel** als Speicher
- Kellerautomaten durchlaufen einmal die Eingabe (fast)
- Stapel erlaubt Ablegen, Entfernen und Lesen von Elementen
- es kann nur das zuletzt abgelegte Element gelesen oder entfernt werden (last-in-first-out)



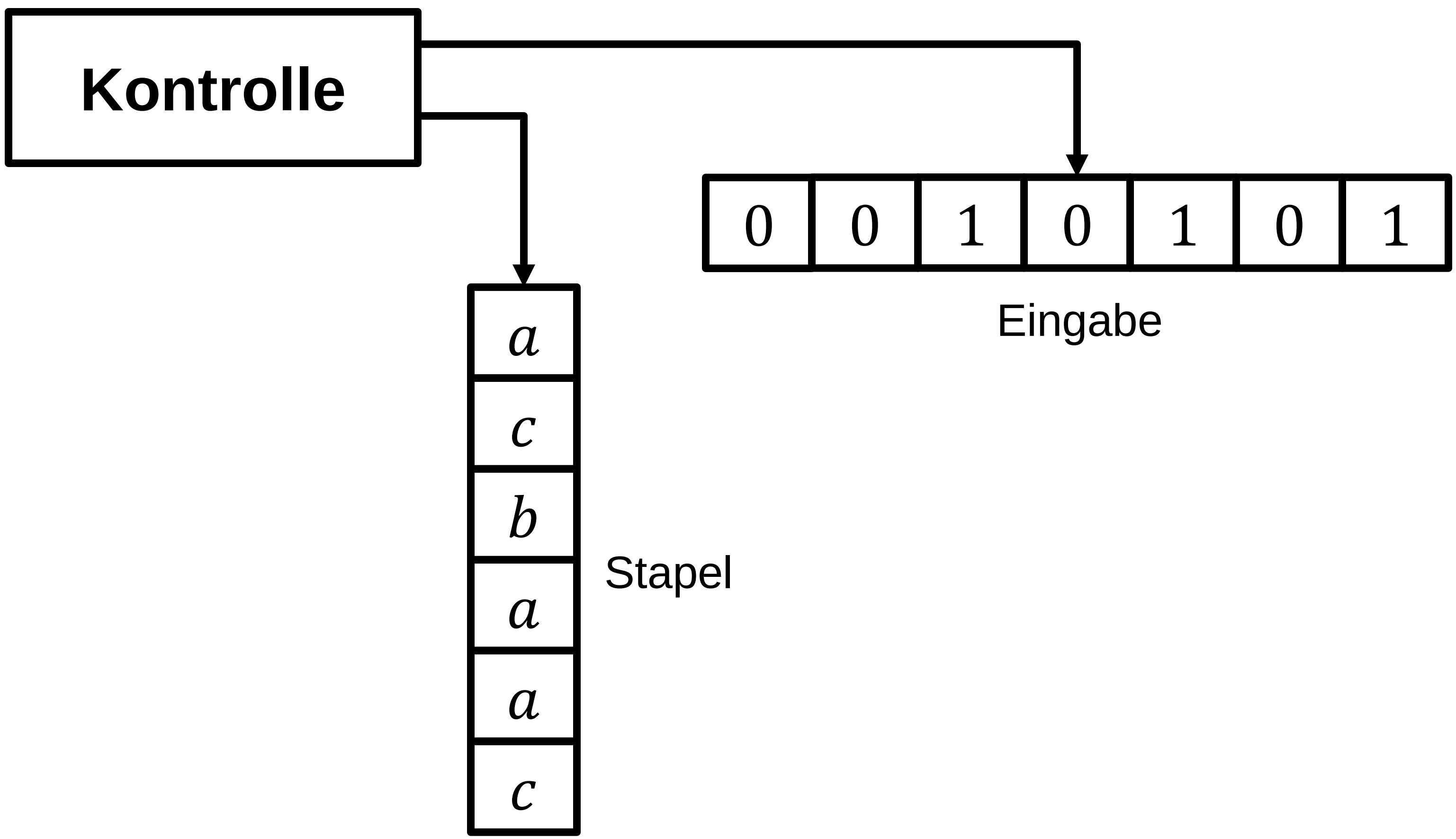
Kellerautomaten

Definition

Ein Kellerautomat (PDA) ist definiert durch ein 6-Tupel $(Q, \Sigma, \Gamma, \delta, q_0, F)$, wobei

1. Q eine endliche Menge von Zuständen ist,
2. Σ das endliche Eingabealphabet ist,
3. Γ das endliche Stapelalphabet ist,
4. $\delta: Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \to \mathcal{P}(Q \times \Gamma_\varepsilon)$ die Übergangsfunktion ist,
5. $q_0 \in Q$ der Startzustand und $F \subseteq Q$ die Menge der akzeptierenden Zustände ist.

Kellerautomaten – schematische Darstellung

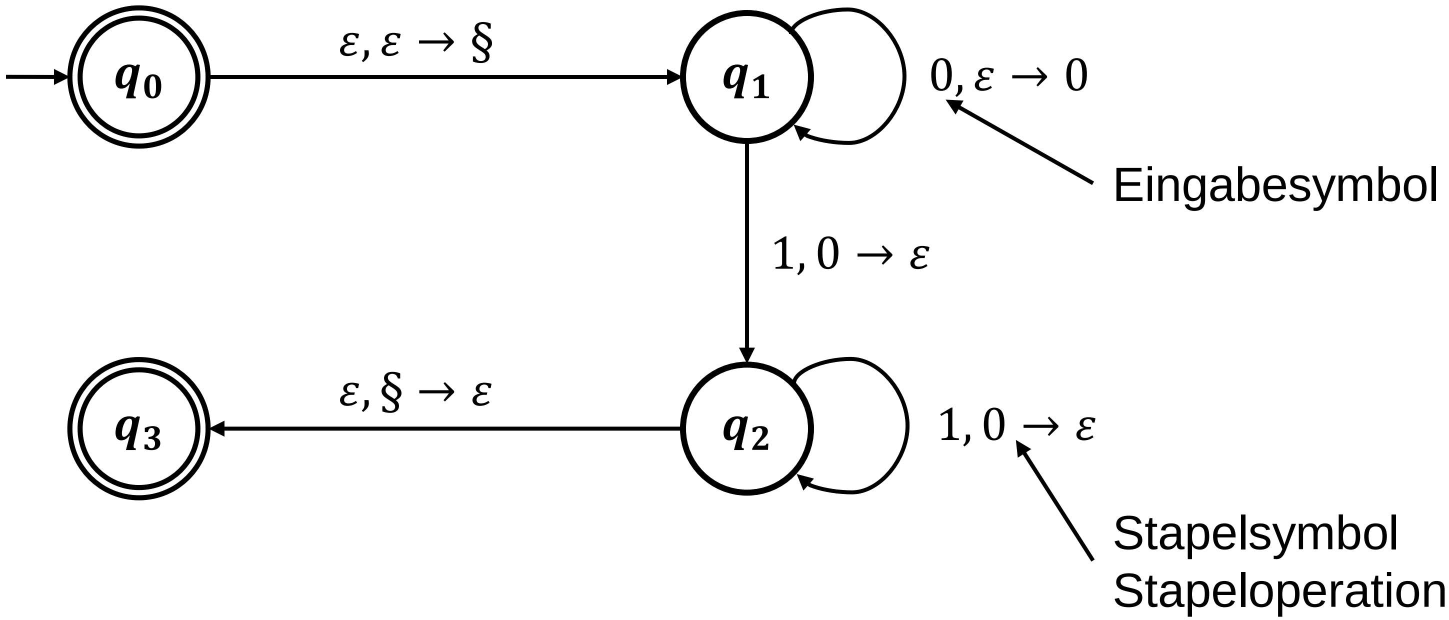


Kellerautomaten

Definition

- Kellerautomaten sind nichtdeterministisch
- $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}, \Gamma_\varepsilon = \Gamma \cup \{\varepsilon\}$
- Übergangsfunktion $(q_2, c) \in \delta(q_1, a, b), a, b, c = \varepsilon$ zugelassen
- $a = \varepsilon$: Lesekopf wird nicht bewegt, also Kellerautomaten durchlaufen einmal die Eingabe, dürfen aber dabei anhalten
- $b = \varepsilon$: c wird zusätzlich auf Stapel gelegt (push-Operation)
- $c = \varepsilon$: b wird vom Stapel entfernt (pop-Operation)

Kellerautomaten – graphische Darstellung



§ markiert Boden des Stapels

PDA – Berechnung

$PDA = (Q, \Sigma, \Gamma, \delta, q_0, F)$. Berechnung bei Eingabe $w \in \Sigma^*$:

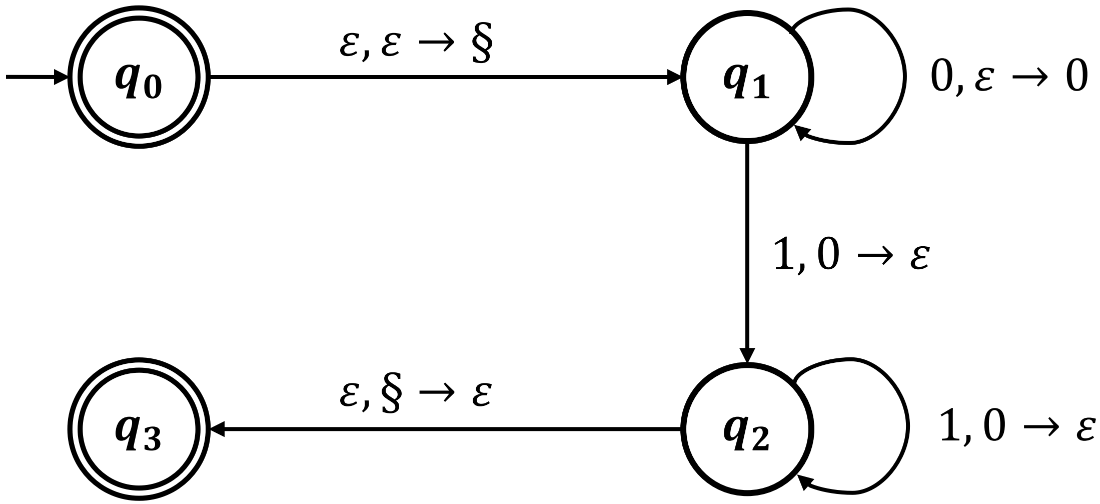
- wir schreiben $w = w_1 w_2 \dots w_m, w_i \in \Sigma_\varepsilon$, um Anhalten zu modellieren
- wendet in jedem Rechenschritt Übergangsfunktion δ an, dabei wird stets das nächste w_i gelesen
- Berechnung endet, falls Ende der Eingabe erreicht wurde oder für gelesenes Tripel (Zustand, Eingabesymbol, Stapelsymbol) Wert der Übergangsfunktion $= \emptyset$ ist.
- **Konfiguration** ist gegeben durch aktuelle Position i in der Eingabe, den Zustand und den Stapelinhalt

PDA – Berechnung

$PDA\ \ K = (Q, \Sigma, \Gamma, \delta, q_0, F)$. Berechnung bei Eingabe $w \in \Sigma^*$:

- Berechnung startet in den Zustand q_0, mit leerem Stapel und Lesekopf auf w_1
- durchläuft Folge von Konfigurationen
- Berechnung heißt **akzeptierend**, wenn Ende der Eingabe erreicht wird und letzter Zustand in F liegt
- abbrechende Berechnungen ($\delta(q, a, v) = \emptyset$) sind **ablehnend**
- w wird von K akzeptiert, falls es eine akzeptierende Berechnung von K bei Eingabe w gibt

PDA – Berechnung



Stapel:

ε
$\S$
$0\S$
$00\S$
$0\S$
$\S$
ε

- $w = 0011$, schreiben $w = \varepsilon 0011 \varepsilon$
- Folge von Zuständen ist $q_0, q_1, q_1, q_1, q_2, q_2, q_3$

PDA – akzeptierte Sprache

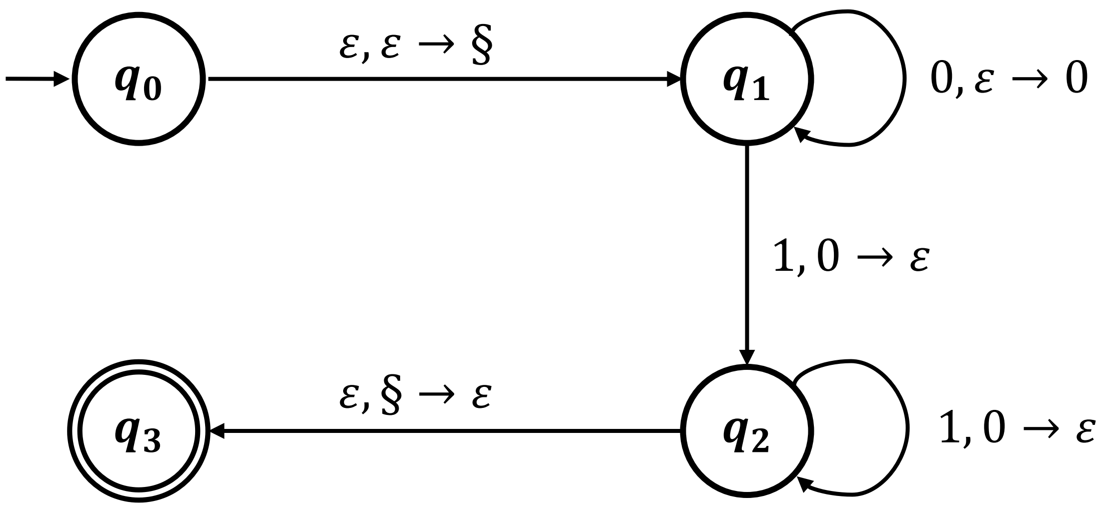
$PDA\ \ K = (Q, \Sigma, \Gamma, \delta, q_0, F)$

$L(K) = \{w \in \Sigma^* \mid \text{es gibt akzeptierende Berechnung von } K \text{ bei Eingabe } w\}$

PDA für $L_1 = \{0^n 1^n \mid n \geq 1\}$

1. Solange das Eingabesymbol eine 0 ist, lege eine 0 auf den Stapel.
2. Wird eine 1 gelesen, entferne eine 0 vom Stapel.
3. Entferne bei jeder weiteren gelesenen 1 eine 0 vom Stapel, bis entweder das Ende der Eingabe erreicht worden ist, oder eine 0 gelesen wird, oder der Stapel leer ist.
4. Akzeptiere, wenn am Ende der Rechnung der Stapel leer ist.

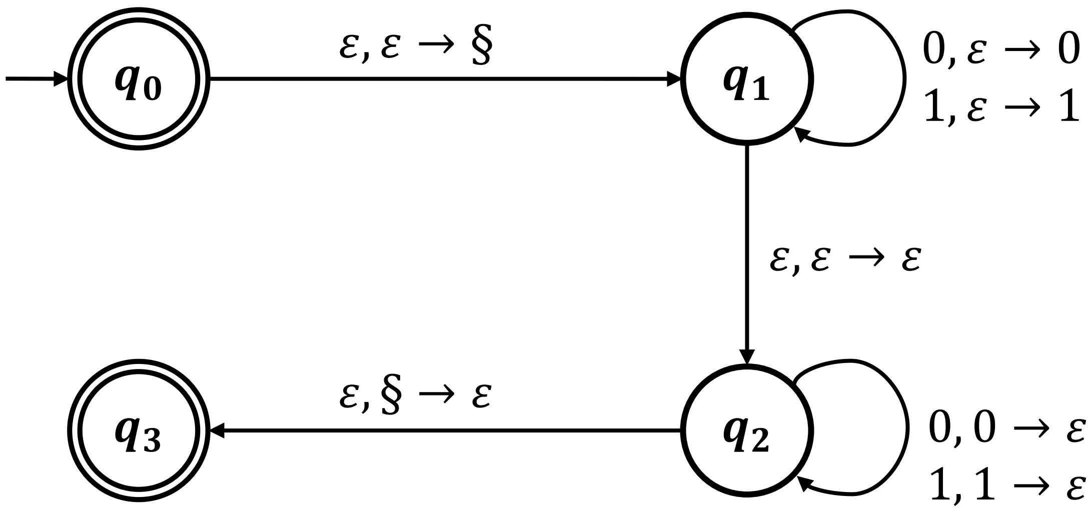
PDA für $L_1 = \{0^n 1^n \mid n \geq 1\}$



PDA für $L_2 = \{ww^R \mid w \in \{0,1\}^*\}$

1. Lege nichtdeterministisch einen Präfix der Eingabe auf den Stapel.
2. Überprüfe, ob Rest der Eingabe mit dem Präfix (in der Reihenfolge der Symbole auf dem Stapel) übereinstimmt.
3. Akzeptiere, falls dies der Fall ist, sonst lehne ab.

PDA für $L_2 = \{ww^R \mid w \in \{0,1\}^*\}$



PDAs und kontextfreie Sprachen

Satz

Sei L eine kontextfreie Sprache.

Dann gibt es einen PDA K, der L akzeptiert.

Satz

Sei K ein PDA und $L = L(K)$.

Dann ist L eine kontextfreie Sprache.