Theoretische Grundlagen der Informatik
Tutorium XII & XIII (SR 301)

Tut Nr. 3 – Üb2, Pumping-Lemma, Kontextfreie Sprachen

David Münch

Karlsruher Institut für Technologie
Institut für Informatik
IKS Müller-Quade

12. November 2009

Karlsruher Institut für Technologie

Inhaltsverzeichnis

Inhaltsverzeichnis

Inhaltsverzeichnis

Inhaltsverzeichnis

Organisatorisches

Email: muenchdavid@gmail.com

https://www.stud.uni-karlsruhe.de/~uhbro/

Tutorium 12: Donnerstags 8:00 Uhr - Raum 301
Tutorium 13: Donnerstags 9:45 Uhr - Raum 301

Übungsblattabgabe Mittwochs 12:00 Uhr.

Organisatorisches

Deckblatt benutzen: `http://www.stud.uni-karlsruhe.de/~unbdh/deckblatt/index.php?course=6`

Gruppenarbeit erwünscht, aber jeder muss handschriftliche Lösung mit Namen aller Gruppenteilnehmer abgeben.

50% der Punkte sind notwendig für den Schein.

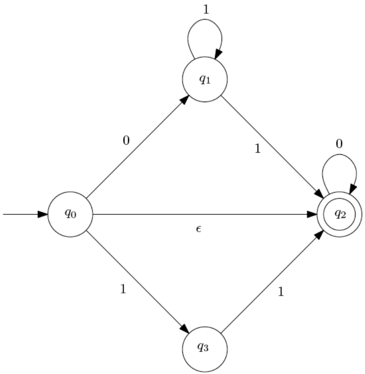
Raumänderung

Ab Donnerstag 12.11.2009 findet dieses Tutorium immer in SR 301 in Gebäude 50.34 statt.

Was wollen wir heute erreichen?

Aufgabe 1

Gegeben sei der folgende nichtdeterministische endliche Automat $\mathcal{M} = (\mathcal{Q}, \Sigma, \delta, q_0, \mathcal{F})$ mit $\mathcal{Q} = \{q_0, q_1, q_2\}$ und $\mathcal{F} = \{q_2\}$. Die Zustandsübergangsrelation δ sei gegeben durch



i) Überführen Sie $\mathcal{M}$ in einen deterministischen endlichen Akzeptor $\mathcal{M}'$.

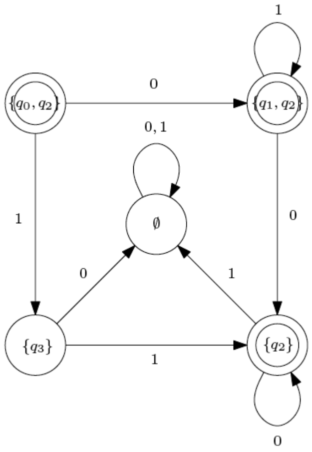
i) Überführen Sie $\mathcal{M}$ in einen deterministischen endlichen Akzeptor $\mathcal{M}'$.

Lösung:

Wir wenden die Potenzmengenkonstruktion nach Myhill-Büchi an. Dazu stellen wir die Zustandsübergangsfunktion iterativ auf, beginnend mit dem ϵ-Abschluss des Anfangszustandes $\{q_0, q_2\}$.

	0	1
$\{q_0, q_2\}$	$\{q_1, q_2\}$	$\{q_3\}$
$\{q_1, q_2\}$	$\{q_2\}$	$\{q_1, q_2\}$
$\{q_3\}$	$\emptyset$	$\{q_2\}$
$\{q_2\}$	$\{q_2\}$	$\emptyset$
$\emptyset$	$\emptyset$	$\emptyset$

Der resultierende DEA $\mathcal{M}'$ hat also die Form



ii) Welche Sprache erkennt $\mathcal{M}$? Geben Sie einen regulären Ausdruck an.

ii) Welche Sprache erkennt $\mathcal{M}$? Geben Sie einen regulären Ausdruck an.

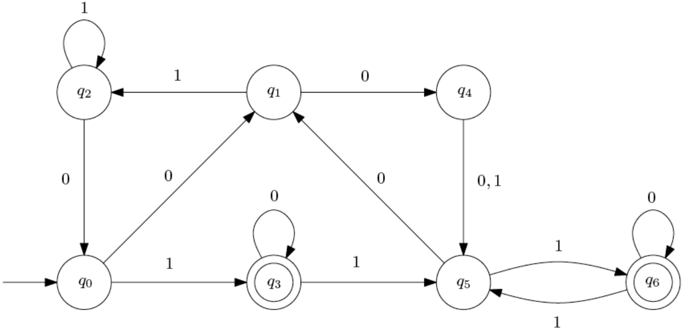
Lösung:

Ein regulärer Ausdruck für $\mathcal{L}(\mathcal{M})$ ist

$$R_{\mathcal{L}(\mathcal{M})} = \epsilon + 01^*(\epsilon + 00^*) + 110^*$$

Aufgabe 2

Gegeben sei der DEA $\mathcal{M} = (\mathcal{Q}, \Sigma, \delta, q_0, \mathcal{F})$ mit den Zuständen $\mathcal{Q} = \{q_0, \ldots, q_6\}$, dem Eingabealphabet $\Sigma = \{0, 1\}$ und der der Finalzustandsmenge $\mathcal{F} = \{q_3, q_6\}$. Die Zustandsübergangsfunktion δ sei gegeben durch



Minimierungsalgorithmus:

1. Entferne alle überflüssigen Zustände.

Minimierungsalgorithmus:

1. Entferne alle überflüssigen Zustände.
2. Stelle eine Tabelle aller Zustandspaare $\{p, q\}$ mit $p \neq q$ auf.

Minimierungsalgorithmus:

1. Entferne alle überflüssigen Zustände.
2. Stelle eine Tabelle aller Zustandspaare $\{p, q\}$ mit $p \neq q$ auf.
3. Markiere alle Paare $\{p, q\}$ mit entweder $p \in F$ oder $q \in F$.

Minimierungsalgorithmus:

1. Entferne alle überflüssigen Zustände.
2. Stelle eine Tabelle aller Zustandspaare $\{p, q\}$ mit $p \neq q$ auf.
3. Markiere alle Paare $\{p, q\}$ mit entweder $p \in F$ oder $q \in F$.
4. Für jedes noch unmarkierte Paar $\{p, q\}$ und jedes $a \in \Sigma$ teste, ob $\{\delta(p, a), \delta(q, a)\}$ bereits markiert ist. Wenn ja, dann markiere auch $\{p, q\}$.

Minimierungsalgorithmus:

1. Entferne alle überflüssigen Zustände.
2. Stelle eine Tabelle aller Zustandspaare $\{p, q\}$ mit $p \neq q$ auf.
3. Markiere alle Paare $\{p, q\}$ mit entweder $p \in F$ oder $q \in F$.
4. Für jedes noch unmarkierte Paar $\{p, q\}$ und jedes $a \in \Sigma$ teste, ob $\{\delta(p, a), \delta(q, a)\}$ bereits markiert ist. Wenn ja, dann markiere auch $\{p, q\}$.
5. Wiederhole Schritt 4 bis sich keine Änderung in der Tabelle mehr ergibt.

Minimierungsalgorithmus:

1. Entferne alle überflüssigen Zustände.
2. Stelle eine Tabelle aller Zustandspaare $\{p, q\}$ mit $p \neq q$ auf.
3. Markiere alle Paare $\{p, q\}$ mit entweder $p \in F$ oder $q \in F$.
4. Für jedes noch unmarkierte Paar $\{p, q\}$ und jedes $a \in \Sigma$ teste, ob $\{\delta(p, a), \delta(q, a)\}$ bereits markiert ist. Wenn ja, dann markiere auch $\{p, q\}$.
5. Wiederhole Schritt 4 bis sich keine Änderung in der Tabelle mehr ergibt.
6. Verschmelze alle jetzt noch unmarkierten Paare zu jeweils einem Zustand.

i) Minimieren Sie den Automaten $\mathcal{M}$ mit dem in der Vorlesung vorgestellten Verfahren.

i) Minimieren Sie den Automaten $\mathcal{M}$ mit dem in der Vorlesung vorgestellten Verfahren.

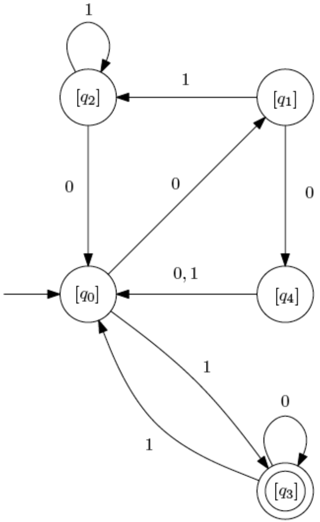
Lösung:

Wir erhalten folgende Tabelle:

	q_0	q_1	q_2	q_3	q_4	q_5
q_1	2	-				
q_2	2	3	-			
q_3	1	1	1	-		
q_4	2	3	3	1	-	
q_5		2	2	1	2	-
q_6	1	1	1		1	1

Der minimierte Automat hat folgenden Zustandsübergangsgraphen:

Übungsblatt 2



ii) Geben Sie einen regulären Ausdruck für die von $\mathcal{M}$ erkannte Sprache $\mathcal{L}$ an.

ii) Geben Sie einen regulären Ausdruck für die von $\mathcal{M}$ erkannte Sprache $\mathcal{L}$ an.

Lösung: Wir betrachten den minimierten Automaten. Liest man von q_0 eine 0, so ist $0(11^*0 + 00 + 01)$ ein regulärer Ausdruck für die Menge der Worte, die zu q_0 zurückführen. Liest man von $[q_0]$ eine 1, so ist 10^*1 die Menge der Worte die zu q_0 zurückführen. Um von $[q_0]$ in den Endzustand $[q_3]$ zu gelangen sind 10^* zulässige Worte. Man erhält also folgenden regulären Ausdruck für $\mathcal{L}$:

$$R_{\mathcal{L}} = (0(11^*0 + 00 + 01) + 10^*1)^*10^*$$

Pumping-Lemma für reg. Sprachen

Satz: Pumping-Lemma

Sei L eine reguläre Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, sodass für jedes Wort $w \in L$ mit $|w| > n$ eine Darstellung

$$w = uvx \text{ mit } |uv| \leq n, v \neq \epsilon,$$

existiert, bei der auch $uv^ix \in L$ ist für alle $i \in \mathbb{N}_0$.

Satz: Pumping-Lemma

Sei L eine reguläre Sprache. Dann existiert eine Zahl $n \in \mathbb{N}$, sodass für jedes Wort $w \in L$ mit $|w| > n$ eine Darstellung

$$w = uvx \text{ mit } |uv| \leq n, v \neq \epsilon,$$

existiert, bei der auch $uv^ix \in L$ ist für alle $i \in \mathbb{N}_0$.

Hinweis:
Man verwendet das Pumping-Lemma um zu zeigen, dass eine Sprache nicht regulär ist.

Beispiel:
$L = \{a^n b^n | n > 0, n \in \mathbb{N}\}$

Um zu zeigen, dass L nicht regulär ist, verwenden wir das Pumping-Lemma.

Beispiel:
$L = \{a^n b^n | n > 0, n \in \mathbb{N}\}$

Um zu zeigen, dass L nicht regulär ist, verwenden wir das Pumping-Lemma.

Angenommen L sei regulär. Dann $\exists n$, sodass sich alle Wörter $w \in L$ der Länge $\geq n$ wie im Pumping Lemma beschrieben zerlegen lassen. Betrachten wir speziell das Wort $a^n b^n$ der Länge $2n$. Die entsprechende Zerlegung uvx dieses Wortes ist aufgrund $v \neq \epsilon$ so, dass v nicht leer ist, und aufgrund $|uv| \leq n$ kann v nur aus a's bestehen. Aufgrund von $uv^i x \in L$ wäre dann das Wort $ux = a^{n-|v|} b^n$ in der Sprache, was im Widerspruch zur Definition von L steht. Deshalb ist L nicht regulär.

Aufgabe 4

$L = \{0^m | m \text{ ist Quadratzahl}\}$ Zeige, dass L nicht regulär ist.

Aufgabe 4

$L = \{0^m | m \text{ ist Quadratzahl}\}$ Zeige, dass L nicht regulär ist.

Aufgabe 5

$L = \{0^p | p \text{ ist Primzahl}\}$ Zeige, dass L nicht regulär ist.

Aufgabe 6

$L = \{a^n b a^m b a^{n+m} | n, m \geq 1\}$ Zeige, dass L nicht regulär ist.

Aufgabe 6

$L = \{a^n b a^m b a^{n+m} | n, m \geq 1\}$ Zeige, dass L nicht regulär ist.

Aufgabe 7

Zeige für die Sprache $L = \{01^k 0^l 1 | k, l \geq 0\}$ explizit, dass die im Pumping-Lemma formulierte notwendige Bedingung für Regularität erfüllt ist. (Gib also eine konkrete Belegung für die existenzquantifizierten Ausdrücke an und begründe.)

Aufgabe 8

$L = \emptyset$

Ist für L das Pumping-Lemma anwendbar?

Aufgabe 8

$L = \emptyset$

Ist für L das Pumping-Lemma anwendbar?

Aufgabe 9

$L = \{00, 11\}$

Ist für L das Pumping-Lemma anwendbar?

Definition: Chomsky-Hierarchie

- Grammatiken ohne weitere Einschränkungen heissen Grammatiken vom **Typ 0**.

Definition: Chomsky-Hierarchie

- Grammatiken ohne weitere Einschränkungen heissen Grammatiken vom **Typ 0**.
- Grammatiken, bei denen alle Ableitungsregeln die Form

  $u \rightarrow v$ mit $u \in V^+, v \in ((V \cup \Sigma)\backslash\{S\})^+$ und $|u| \leq |v|$, oder
  $S \rightarrow \varepsilon$

  haben, heissen **kontextsensitiv** od. Grammatiken vom **Typ 1**.

Definition: Chomsky-Hierarchie

- Grammatiken ohne weitere Einschränkungen heissen Grammatiken vom **Typ 0**.
- Grammatiken, bei denen alle Ableitungsregeln die Form

  $u \to v$ mit $u \in V^+, v \in ((V \cup \Sigma)\backslash\{S\})^+$ und $|u| \leq |v|$, oder
  $S \to \varepsilon$

  haben, heissen **kontextsensitiv** od. Grammatiken vom **Typ 1**.
- Grammatiken, bei denen alle Ableitungsregeln die Form
  $A \to v$ mit $A \in V$ und $v \in (V \cup \Sigma)^*$
  haben, heissen **kontextfrei** oder Grammatiken vom **Typ 2**.

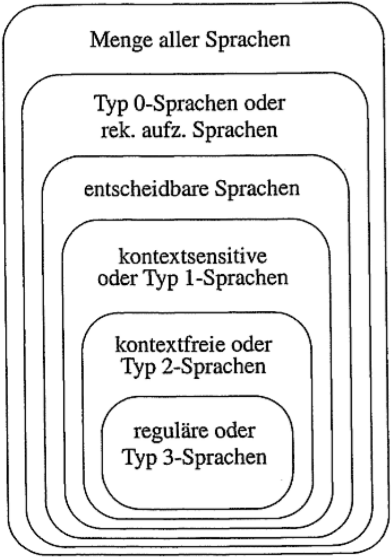
Definition: Chomsky-Hierarchie

- Grammatiken ohne weitere Einschränkungen heissen Grammatiken vom **Typ 0**.
- Grammatiken, bei denen alle Ableitungsregeln die Form

  $u \to v$ mit $u \in V^+, v \in ((V \cup \Sigma)\backslash\{S\})^+$ und $|u| \leq |v|$, oder
  $S \to \varepsilon$

  haben, heissen **kontextsensitiv** od. Grammatiken vom **Typ 1**.
- Grammatiken, bei denen alle Ableitungsregeln die Form
  $A \to v$ mit $A \in V$ und $v \in (V \cup \Sigma)^*$
  haben, heissen **kontextfrei** oder Grammatiken vom **Typ 2**.
- Grammatiken, bei denen alle Ableitungsregeln die Form
  $A \to v$ mit $A \in V$ und $v = \varepsilon$ oder $v = aB$ mit $a \in \Sigma, B \in V$
  haben, heissen **rechtslinear** oder Grammatiken vom **Typ 3**.

Kontextfreie Sprachen



Definition: Chomsky-Normalform

Eine kontextfreie Grammatik G heißt in CNF, falls alle Regeln eine der folgenden Formen haben:

- $A \rightarrow BC$
- $A \rightarrow a$
- $S \rightarrow \varepsilon$.

Dabei sind A, B, C, S Variablen, S Startsymbol und a Terminal.

Kellerautomat

Definition: Kellerautomat

Ein (nichtdeterministischer) Kellerautomat (NPDA bzw. PDA) $\mathcal{K}$ besteht aus $(Q, \Sigma, \Gamma, \delta, q_0, F)$, wobei

- Q endliche Zustandsmenge

Definition: Kellerautomat

Ein (nichtdeterministischer) Kellerautomat (NPDA bzw. PDA) $\mathcal{K}$ besteht aus $(Q, \Sigma, \Gamma, \delta, q_0, F)$, wobei

- Q endliche Zustandsmenge
- Σ endliches Eingabealphabet

Definition: Kellerautomat

Ein (nichtdeterministischer) Kellerautomat (NPDA bzw. PDA) $\mathcal{K}$ besteht aus $(Q, \Sigma, \Gamma, \delta, q_0, F)$, wobei

- Q endliche Zustandsmenge
- Σ endliches Eingabealphabet
- Γ endliches STACK-Alphabet

Definition: Kellerautomat

Ein (nichtdeterministischer) Kellerautomat (NPDA bzw. PDA) $\mathcal{K}$ besteht aus $(Q, \Sigma, \Gamma, \delta, q_0, F)$, wobei

- Q endliche Zustandsmenge
- Σ endliches Eingabealphabet
- Γ endliches STACK-Alphabet
- $q_0 \in Q$ Anfangszustand

Definition: Kellerautomat

Ein (nichtdeterministischer) Kellerautomat (NPDA bzw. PDA) $\mathcal{K}$ besteht aus $(Q, \Sigma, \Gamma, \delta, q_0, F)$, wobei

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

Definition: Kellerautomat

Ein (nichtdeterministischer) Kellerautomat (NPDA bzw. PDA) $\mathcal{K}$ besteht aus $(Q, \Sigma, \Gamma, \delta, q_0, F)$, wobei

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

Kellerautomat

Ein Beispiel für einen Zustandsübergang:

$$\delta(q_0, a, S) = (q_1, AS)$$

Bedeutet:

Kellerautomat

Ein Beispiel für einen Zustandsübergang:

$$\delta(q_0, a, S) = (q_1, AS)$$

Bedeutet:

- PDA ist im Zustand q_0,

Kellerautomat

Ein Beispiel für einen Zustandsübergang:

$$\delta(q_0, a, S) = (q_1, AS)$$

Bedeutet:

- PDA ist im Zustand q_0,
- liest das Eingabezeichen a,

Kellerautomat

Ein Beispiel für einen Zustandsübergang:

$$\delta(q_0, a, S) = (q_1, AS)$$

Bedeutet:

- PDA ist im Zustand q_0,
- liest das Eingabezeichen a,
- liest das Stack-Element S,

Kellerautomat

Ein Beispiel für einen Zustandsübergang:

$$\delta(q_0, a, S) = (q_1, AS)$$

Bedeutet:

- PDA ist im Zustand q_0,
- liest das Eingabezeichen a,
- liest das Stack-Element S,
- wechselt in den Zustand q_1 und

Kellerautomat

Ein Beispiel für einen Zustandsübergang:

$$\delta(q_0, a, S) = (q_1, AS)$$

Bedeutet:

- PDA ist im Zustand q_0,
- liest das Eingabezeichen a,
- liest das Stack-Element S,
- wechselt in den Zustand q_1 und
- legt AS auf den Stack (A steht ganz oben)

Arbeitsweise

Wie arbeitet ein (nichtdeterministischer) Kellerautomat?

- Eingabewort $w \rightarrow$ Eingabeband. Lesekopf auf dem 1. Zeichen von links

Arbeitsweise

Wie arbeitet ein (nichtdeterministischer) Kellerautomat?

- Eingabewort $w \rightarrow$ Eingabeband. Lesekopf auf dem 1. Zeichen von links
- Im Zustand $q_i \in Q$ liest der [N]PDA ein Zeichen $a \in \Sigma$ vom Eingabeband und das oberste Zeichen $A \in \Gamma$ vom Stack

Arbeitsweise

Wie arbeitet ein (nichtdeterministischer) Kellerautomat?

- Eingabewort $w \rightarrow$ Eingabeband. Lesekopf auf dem 1. Zeichen von links
- Im Zustand $q_i \in Q$ liest der [N]PDA ein Zeichen $a \in \Sigma$ vom Eingabeband und das oberste Zeichen $A \in \Gamma$ vom Stack
- Lesekopf wird eine Position nach rechts gefahren

Arbeitsweise

Wie arbeitet ein (nichtdeterministischer) Kellerautomat?

- Eingabewort $w \rightarrow$ Eingabeband. Lesekopf auf dem 1. Zeichen von links
- Im Zustand $q_i \in Q$ liest der [N]PDA ein Zeichen $a \in \Sigma$ vom Eingabeband und das oberste Zeichen $A \in \Gamma$ vom Stack
- Lesekopf wird eine Position nach rechts gefahren
- oberstes Zeichen von Stack gelöscht

Arbeitsweise

Wie arbeitet ein (nichtdeterministischer) Kellerautomat?

- Eingabewort $w \rightarrow$ Eingabeband. Lesekopf auf dem 1. Zeichen von links
- Im Zustand $q_i \in Q$ liest der [N]PDA ein Zeichen $a \in \Sigma$ vom Eingabeband und das oberste Zeichen $A \in \Gamma$ vom Stack
- Lesekopf wird eine Position nach rechts gefahren
- oberstes Zeichen von Stack gelöscht
- abhängig von q_i, a und A wechselt der [N]PDA in den Zustand $q_j \in Q$ und schreibt $B \in \Gamma^*$ auf den Stack

Arbeitsweise

Wie arbeitet ein (nichtdeterministischer) Kellerautomat?

- Eingabewort $w \rightarrow$ Eingabeband. Lesekopf auf dem 1. Zeichen von links
- Im Zustand $q_i \in Q$ liest der [N]PDA ein Zeichen $a \in \Sigma$ vom Eingabeband und das oberste Zeichen $A \in \Gamma$ vom Stack
- Lesekopf wird eine Position nach rechts gefahren
- oberstes Zeichen von Stack gelöscht
- abhängig von q_i, a und A wechselt der [N]PDA in den Zustand $q_j \in Q$ und schreibt $B \in \Gamma^*$ auf den Stack
- Terminierung, falls
  - Finalzustand wird erreicht $\Rightarrow$ bisher gelesenes Wort wird akzeptiert
  - Kein Eingabesymbol mehr da $\Rightarrow$ Wort wird nicht akzeptiert

Beispiel 2

Gib für die Sprache L einen PDA an:

$$L = \{0^n 1^m \mid n \leq m\}$$

Aufgabe 2

Gib für die Sprache L einen PDA an:

$$L = \{0^n 1^m 0^n \mid n, m > 0\}$$

Chomsky-2 $\rightarrow$ Kellerautomat

Wie konstruiere ich einen Kellerautomaten aus einer kontextfreien Grammatik $G = (V, \Sigma, S, R)$?
Setze:

Chomsky-2 $\rightarrow$ Kellerautomat

Wie konstruiere ich einen Kellerautomaten aus einer kontextfreien Grammatik $G = (V, \Sigma, S, R)$?
Setze:

- $Q := \{q_0\}$

Chomsky-2 → Kellerautomat

Wie konstruiere ich einen Kellerautomaten aus einer kontextfreien Grammatik $G = (V, \Sigma, S, R)$?
Setze:

- $Q := \{q_0\}$
- $\Sigma := \Sigma$

Chomsky-2 $\rightarrow$ Kellerautomat

Wie konstruiere ich einen Kellerautomaten aus einer kontextfreien Grammatik $G = (V, \Sigma, S, R)$?
Setze:

- $Q := \{q_0\}$
- $\Sigma := \Sigma$
- $\Gamma := V$

Chomsky-2 → Kellerautomat

Wie konstruiere ich einen Kellerautomaten aus einer kontextfreien Grammatik $G = (V, \Sigma, S, R)$?
Setze:

- $Q := \{q_0\}$
- $\Sigma := \Sigma$
- $\Gamma := V$
- $Z_0 := S$

Chomsky-2 $\rightarrow$ Kellerautomat

Wie konstruiere ich einen Kellerautomaten aus einer kontextfreien Grammatik $G = (V, \Sigma, S, R)$?
Setze:

- $Q := \{q_0\}$
- $\Sigma := \Sigma$
- $\Gamma := V$
- $Z_0 := S$
- $\delta(q_0, a, A) := \{(q_0, \alpha) | (A \rightarrow a\alpha) \in R\}$

Aufgabe 3

Gegeben ist die kontextfreie Grammatik $G = (V, \Sigma, S, R)$ mit $V = \{S, A, B, C\}, \Sigma = \{a, b, c\}$ und der Regelmenge $R = \{$

$$\begin{aligned} S &\rightarrow aBC|bBBCC|aACCA \\ B &\rightarrow b|aAB \\ A &\rightarrow a \\ C &\rightarrow c\} \end{aligned}$$

1. Gebe einen Kellerautomat $\mathcal{K}$ an, der gerade die von G erzeugte Sprache akzeptiert.
2. Skizziere die Herleitung der Wortes $w = baabbcc$ in G und gebe eine akzeptierende Berechnung von $\mathcal{K}$ für w an.

Reflexion

Was haben wir heute gelernt?

Reflexion

Was haben wir heute gelernt?

- Pumping-Lemma

Reflexion

Was haben wir heute gelernt?

- Pumping-Lemma
- Kontextfreie Sprachen

Reflexion

Was haben wir heute gelernt?

- Pumping-Lemma
- Kontextfreie Sprachen
- CNF und CYK

Reflexion

Was haben wir heute gelernt?

- Pumping-Lemma
- Kontextfreie Sprachen
- CNF und CYK
- Kellerautomaten

Noch Fragen?

Vorschau

Vorschau

- Chomsky 1

Vorschau

- Chomsky 1
- Touringmaschinen

Vorschau

- Chomsky 1
- Touringmaschinen
- Pumping-Lemma für kf Sprachen

Bis zum nächsten Mal