Informatik III - Tutorium XIII & XV (SR -107)

Tut Nr. 13 – Üb12, Kellerautomaten, Greibachnormalform

David Münch

Universität Karlsruhe (TH)
Institut für Informatik
ITI Wagner

6. Februar 2008

Universität Karlsruhe (TH)
Forschungsuniversität • gegründet 1825

Inhaltsverzeichnis

Inhaltsverzeichnis

Inhaltsverzeichnis

Inhaltsverzeichnis

Organisatorisches

Email: muenchdavid@gmail.com

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

Tutorium 13: Mittwochs 8:00 Uhr - Raum -107
Tutorium 15: Mittwochs 9:45 Uhr - Raum -107

Übungsblattabgabe Donnerstag.

Was wollen wir heute erreichen?

Was wollen wir heute erreichen?

- Kellerautomaten verstehen und konstruieren können

Was wollen wir heute erreichen?

- Kellerautomaten verstehen und konstruieren können
- Algorithmus für Greibach Normalform anwenden können

Aufgabe 6

1. Zeige mithilfe des Pumping-Lemmas, dass die Sprache $L_1 = \{0^i 1^j 2^{i+j} 3^i \mid i \geq 0, j \geq 0\}$ nicht kontextfrei ist.

2. Zeige mithilfe von Ogden's Lemma, dass die Sprache $L_2 = \{w \in \{a, b, c\}^* : |w|_a = |w|_b = |w|_c\}$ nicht kontextfrei ist, wobei $|w|_x$ für die Häufigkeit des Buchstabens x im Wort w bezeichne.

Aufgabe 6

1. Wir nehmen an L_1 sei kontextfrei. Dann gibt es ein $n \in \mathbb{N}$ mit der Eigenschaft, dass jedes Wort w mit $|w| \geq n$ eine Zerlegung $w = uvwxy$ besitzt mit $|vx| \geq 1$ und $|vwx| \leq n$, so dass $uv^iwx^iy \in L_1$ für alle $i \geq 0$. Zu $n \in \mathbb{N}$ betrachten wir das Wort $w_n = 0^n2^n3^n$. Sei $w_n = uvwxy$ eine beliebige Zerlegung mit den gewünschten Eigenschaften. Nach Wahl der Zerlegung und w_n können nur zwei Fälle auftreten:

Aufgabe 6

1. Wir nehmen an L_1 sei kontextfrei. Dann gibt es ein $n \in \mathbb{N}$ mit der Eigenschaft, dass jedes Wort w mit $|w| \geq n$ eine Zerlegung $w = uvwxy$ besitzt mit $|vx| \geq 1$ und $|vwx| \leq n$, so dass $uv^iwx^iy \in L_1$ für alle $i \geq 0$. Zu $n \in \mathbb{N}$ betrachten wir das Wort $w_n = 0^n2^n3^n$. Sei $w_n = uvwxy$ eine beliebige Zerlegung mit den gewünschten Eigenschaften. Nach Wahl der Zerlegung und w_n können nur zwei Fälle auftreten:

Fall 1:
vwx enthält nur Nullen, Zweien oder Dreien. Falls vwx nur Nullen oder nur Dreien enthält, gilt $uv^0wx^0y = 0^i2^n3^j$ und entweder $i < n$ und $j = n$ oder $i = n$ und $j < n$, da vx mindestens eine Null oder Drei enthält. Also $uv^0wx^0y \notin L_1$. Falls vwx nur Zweien enthält, so gilt $uv^0wx^0y = 0^n2^i3^n$ mit $i < n$, da vx mindestens eine Zwei enthält. Und auch dann ist $uv^0wx^0y \notin L_1$.

Aufgabe 6

Fall 2:
vwx enthält nur Nullen und Zweien (aber je mindestens eine) oder nur Zweien und Dreien (aber je mindestens eine). Falls vwx nur Nullen und Zweien enthält, so gilt $uv^0wx^0y = 0^i2^j3^n$ mit $i < n$ oder $j < n$, da vx mindestens eine Null oder eine Zwei enthält und somit $uv^0wx^0y \notin L_1$. Die Argumentation ist analog, wenn vwx nur Zweien und Dreien enthält.

Aufgabe 6

Fall 2:
vwx enthält nur Nullen und Zweien (aber je mindestens eine) oder nur Zweien und Dreien (aber je mindestens eine). Falls vwx nur Nullen und Zweien enthält, so gilt $uv^0wx^0y = 0^i2^j3^n$ mit $i < n$ oder $j < n$, da vx mindestens eine Null oder eine Zwei enthält und somit $uv^0wx^0y \notin L_1$. Die Argumentation ist analog, wenn vwx nur Zweien und Dreien enthält.

Beide Fälle führen in einen Widerspruch zum Pumping Lemma. Also ist L_1 nicht kontextfrei.

Aufgabe 6

2. Wir nehmen an L_2 sei kontextfrei. Zu $n \in \mathbb{N}$ betrachten wir das Wort $w_n = a^{n+1}b^{n+1}c^{n+1} \in L_2$ und markieren in diesem Wort die $n+1$ bs. Sei nun $w_n = uvwxy$ eine Zerlegung mit der Eigenschaft, dass von den n markierten Zeichen mindestens eines zu vx gehört und höchstens n zu vwx. Nach Wahl der markierten Buchstaben in w_n enthält vwx entweder nur as oder bs oder nur bs oder cs, da vwx ansonsten $n+1$ markierte Zeichen enthalten würde. Wir nehmen o.B.d.A. an, vwx enthalte nur as und bs. Damit gilt $uv^0wx^0y = a^ib^jc^{n+1}$ mit $i < n+1$ oder $j < n+1$, da vx mindestens ein a oder ein b enthält, und somit $uv^0wx^0y \notin L_2$. Also erfüllt L_2 Ogden's Lemma nicht, was im Widerspruch zur Annahme steht. L_2 ist also nicht kontextfrei.

Aufgabe 7

Zeige oder widerlege, dass die kontextfreien Sprachen unter Spiegelung abgeschlossen sind. Die Spiegelung einer Sprache entsteht durch Spiegelung aller Wörter der Sprache, d.h. die Spiegelsprache L^R ist gegeben durch $L^R = \{w^R \mid w \in L\}$. $w^R = w_k \cdots w_1$ bezeichne dabei das Spiegelwort von $w = w_1 \cdots w_k$.

Aufgabe 7

Die kontextfreien Grammatiken sind unter Spiegelung abgeschlossen. Sei $G = (V, \Sigma, S, R)$ eine kontextfreie Grammatik. Eine kontextfreie Grammatik $G^R = (V, \Sigma, S, R^R)$ für die Spiegelsprache lässt sich konstruieren, indem man für jede Regel $A \to \beta$ aus R eine gespiegelte Regel $A \to \beta^R$ zu R^R hinzunimmt. Begründung: Zu zeigen ist $L(G)^R = L(G^R)$: Wir zeigen zunächst die folgende Beobachtung per Induktion über k: Sei $R_1, R_2, \ldots, R_k$ eine Folge von Regeln einer Ableitung $S \xrightarrow{*} \alpha$ mit $\alpha \in (V \cup \Sigma)^*$ mit $R_i \in R$. Dann ist $R_1^R, R_2^R, \ldots, R_k^R$ eine Folge von Regeln einer Ableitung $S \xrightarrow{*} \alpha^R$ mit $R_i^R \in R^R$.

Aufgabe 7

Für $k = 1$ ist die Behauptung offensichtlich nach Konstruktion von R^R richtig.
Sei $R_1, R_2, \ldots, R_k$ eine Folge von Regeln einer Ableitung $S \xrightarrow{*} \chi$ mit $\chi \in (V \cup \Sigma)^*$. Dann gibt es eine Zerlegung $\chi = \alpha\beta\gamma$ mit $\alpha, \beta, \gamma \in (V \cup \Sigma)^*$ und $R_i \in R$ mit $S \xrightarrow{*} \alpha A \gamma \xrightarrow{R_k} \alpha\beta\gamma$ mit $R_k : A \to \beta$. Nach Induktionsannahme liefern die Regeln $R_1^R, \ldots R_{k-1}^R$ die Ableitung $S \xrightarrow{*} (\alpha A \gamma)^R = \gamma^R A \alpha^R$. Wegen $R_k^R : S \to \beta^R$ liefert die Anwendung von R_k^R auf $\gamma^R A \alpha^R$ dann die Ableitung $S \xrightarrow{*} \gamma^R \beta^R \alpha^R = (\alpha\beta\gamma)^R$.
Sei w also aus $L(G)$, dann gibt es eine Ableitung von w^R über der Regelmenge R^R, also $L(G)^R \subseteq L(G^R)$. Wegen der Dualität $(G^R)^R = G$ und $(L^R)^R = L$ gilt obige Induktion auch für die gespiegelte Grammatik, woraus die Umkehrung analog folgt.

Aufgabe 8

Sei $G = (V, \Sigma, S, R)$ mit $V = \{A, B, C, D, E\}$ und $\Sigma = \{a, b\}$ die durch folgende Regelmenge gegebene Grammatik:

$$\begin{aligned} S &\rightarrow AB \mid AC \mid DD \\ A &\rightarrow a \\ B &\rightarrow b \\ C &\rightarrow SB \mid SC \\ D &\rightarrow AE \mid EB \mid E \\ E &\rightarrow DA \mid DB \end{aligned}$$

1. Enthält die Grammatik nutzlose Symbole? Wenn ja, welche?
2. Ist die von der Grammatik erzeugte Sprache endlich? Begründe Deine Antwort.

Aufgabe 8

1. Berechne zunächst erzeugende Variablen: A, B, S, C.
   Berechne anschliessend nützliche Variablen ausgehend von den erzeugenden Variablen: Hier sind alle erzeugenden Variablen auch nützlich. Nutzlose Variablen sind also D, E. (Nützliche Variablen können z.B. durch Tiefensuche mit Wurzel S aus dem durch die erzeugenden Variablen und erzeugenden Regeln induzierten Abhängigkeitsgraphen ermittelt werden. Dieser Graph enthält als Knoten die erzeugenden Variablen und gerichtete Kanten (A, B), wenn es eine Regel mit linker Seite A und rechter Seite r gibt, so dass r die Variable B und ansonsten nur erzeugende Variablen und Terminale enthält.)

2. Betrachte Abhängigkeitsgraphen der durch die nützlichen Variablen und Regeln induziert wird (Dieser Graph enthält als Knoten die nützlichen Variablen und gerichtete Kanten (A, B), wenn es eine Regel mit linker Seite A und rechter Seite r gibt, so dass r die Variable B und ansonsten nur nützliche Variablen und Terminale enthält.). Dieser enthält zwei Zyklen $S \to C \to S$ und $C \to C$. Da die Zyklen nicht ausschliesslich durch Kettenregeln induziert werden, ist die von der Grammatik erzeugte Sprache unendlich.
Wir geben zusätzlich noch eine Konstruktion zum Aufpumpen eines Wortes an. Da C nützlich ist, kommt es in einer Ableitung $S \xrightarrow{*} w$ mit $w \in \Sigma^*$ vor. Ausserdem gelten $C \xrightarrow{*} abC$ und $C \xrightarrow{*} abb$. Damit sind alle Voraussetzungen zum Pumpen gemäss Pumping Lemma erfüllt und wir können mit dieser Konstruktion beliebig lange Wörter erzeugen.

Greibach-Normalform

Definition: Greibach-Normalform

Eine kontextfreie Grammatik ist in Greibach-Normalform, wenn alle Ableitungsregeln von der Form

$$A \to a\alpha \text{ mit } A \in V, a \in \Sigma \text{ und } \alpha \in V^*$$

sind.

Überführung in Greibach-Normalform

Die Herstellung der Greibach Normalform geschieht nun so, dass wir zunächst von einer kontextfreien Grammatik in Chomsky Normalform ausgehen.
Die vorkommenden Variablen werden durchnummeriert, sagen wir $A_1, ..., A_m$. Das erste Ziel ist es nun, die Regeln so zu modifizieren, dass für jede Regel der Form $A_i \rightarrow A_j\alpha$ gilt $i < j$. Dies wird durch das folgende Programm erreicht:

Algorithm 1 GNF

input : Regelmenge von kfG in CNF
output: Regelmenge von kfG mit $A_i \to A_j\alpha$ wobei $i < j$

for *i := 1* **to** *m* **do**
  for *j := 1* **to** *i-1* **do**
    forall $A_i \to A_j\alpha$ **do**
      Seien $A_j \to \beta_1|...|\beta_n$ alle A_j-Regeln
      Füge hinzu: $A_i \to \beta_1\alpha|...|\beta_n\alpha$
      Streiche: $A_i \to A_j\alpha$
    end
  end
  if *es gibt Regeln der Form* $A_i \to A_i\alpha$ **then**
    Wende die Umformung in der Vorüberlegung an, wobei eine neue Variable B_i eingeführt wird.
  end
end

Überführung in Greibach-Normalform

Nun gilt für alle Regeln der Form $A_i \rightarrow A_j\alpha$, dass $i < j$. Ferner ist die Beobachtung wichtig, dass jetzt alle A_m-Regeln auf der rechten Seite mit einem Terminalzeichen beginnen. Indem wir bei A_{m-1} beginend, von hinten nach vorne, entsprechende Ersetzungen durchführen, können wir erreichen, dass alle A_i-Regeln auf der rechten Seite mit einem Terminalzeichen beginnen. Dies führt der folgende Algorithmus aus.

Algorithm 2 GNF'

input : Regelmenge von kfG mit $A_i \rightarrow A_j\alpha$ wobei $i < j$
output: Regelmenge von kfG in fast GNF
for *i := m-1* **downto** *1* **do**
  forall $A_i \rightarrow A_j\alpha$ *mit* $i < j$ **do**
    Seien $A_j \rightarrow \beta_1|...|\beta_n$ alle A_j-Regeln
    Füge hinzu: $A_i \rightarrow \beta_1\alpha|...|\beta_n\alpha$
    Streiche: $A_i \rightarrow A_j\alpha$
  end
end

Überführung in Greibach-Normalform

Nun sind alle A_i-Regeln in Greibach Normalform. Lediglich die im ersten Algorithmus evt. eingeführten neuen Regeln für die B_i-Variablen sind noch nicht in der richtigen Form. Angenommen $B_i \rightarrow A_j\alpha$ ist eine solche Regel. Seien $A_j \rightarrow \beta_1|...|\beta_k$ alle A_j-Regeln, welche ja bereits in GNF sind. Wir ersetzen nun die Regel $B_i \rightarrow A_j\alpha$ einfach durch $B_i \rightarrow \beta_1\alpha|...|\beta_k\alpha$ und erhalten so schliesslich eine äquivalente Grammatik in Greibach Normalform.

Beispiel 1

Überführe die Grammatik $G = (V, \Sigma, A_1, R)$ mit $V = \{A_1, A_2, A_3\}, \Sigma = \{0, 1\}$ und $R = \{$

$$
\begin{aligned}
A_1 &\rightarrow A_2 A_3, \\
A_2 &\rightarrow A_3 A_1, \\
A_2 &\rightarrow 1, \\
A_3 &\rightarrow A_1 A_2, \\
A_3 &\rightarrow 0\}
\end{aligned}
$$

in Greibach-Normalform.

Aufgabe 1

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

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

in Chomsky-Normalform. Überführe G in eine äquivalente Grammatik in Greibach-Normalform.

Kellerautomat

Definition: Kellerautomat

Ein (nichtdeterministischer) Kellerautomat (NPDA bzw. PDA) $\mathcal{K}$ besteht aus $(Q, \Sigma, \Gamma, q_0, Z_0, \delta, F)$, wobei

- Q endliche Zustandsmenge

Definition: Kellerautomat

Ein (nichtdeterministischer) Kellerautomat (NPDA bzw. PDA) $\mathcal{K}$ besteht aus $(Q, \Sigma, \Gamma, q_0, Z_0, \delta, F)$, wobei

- Q endliche Zustandsmenge
- Σ endliches Eingabealphabet

Definition: Kellerautomat

Ein (nichtdeterministischer) Kellerautomat (NPDA bzw. PDA) $\mathcal{K}$ besteht aus $(Q, \Sigma, \Gamma, q_0, Z_0, \delta, F)$, wobei

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

Definition: Kellerautomat

Ein (nichtdeterministischer) Kellerautomat (NPDA bzw. PDA) $\mathcal{K}$ besteht aus $(Q, \Sigma, \Gamma, q_0, Z_0, \delta, F)$, 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 $(Q, \Sigma, \Gamma, q_0, Z_0, \delta, F)$, wobei

- Q endliche Zustandsmenge
- Σ endliches Eingabealphabet
- Γ endliches STACK-Alphabet
- $q_0 \in Q$ Anfangszustand
- $Z_0 \subseteq \Gamma$ Initialisierung des STACK

Definition: Kellerautomat

Ein (nichtdeterministischer) Kellerautomat (NPDA bzw. PDA) $\mathcal{K}$ besteht aus $(Q, \Sigma, \Gamma, q_0, Z_0, \delta, F)$, wobei

- Q endliche Zustandsmenge
- Σ endliches Eingabealphabet
- Γ endliches STACK-Alphabet
- $q_0 \in Q$ Anfangszustand
- $Z_0 \subseteq \Gamma$ Initialisierung des STACK
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to 2^{Q \times \Gamma^*}$, 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 $(Q, \Sigma, \Gamma, q_0, Z_0, \delta, F)$, wobei

- Q endliche Zustandsmenge
- Σ endliches Eingabealphabet
- Γ endliches STACK-Alphabet
- $q_0 \in Q$ Anfangszustand
- $Z_0 \subseteq \Gamma$ Initialisierung des STACK
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to 2^{Q \times \Gamma^*}$, d.h.
  $\delta(q, a, Z) \subseteq \{(q, \gamma) : q \in Q, \gamma \in \Gamma^*\}$ und
  $\delta(q, \varepsilon, Z) \subseteq \{(q, \gamma) : q \in Q, \gamma \in \Gamma^*\}$.
- $F \subseteq Q$ 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
  - Stack ist leer oder 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:

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

Satz: Chomsky-2 $\rightarrow$ Kellerautomat

Für eine Grammatik G in Greibach-Normalform kann ein PDA konstruiert werden, der $L(G)$ mit leerem STACK akzeptiert.

Chomsky-2 $\longrightarrow$ Kellerautomat

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

Chomsky-2 $\longrightarrow$ Kellerautomat

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

- $Q := \{q_0\}$

Chomsky-2 $\longrightarrow$ Kellerautomat

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

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

Chomsky-2 $\longrightarrow$ 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 $\longrightarrow$ 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?

- Greibach-Normalform erstellen

Reflexion

Was haben wir heute gelernt?

- Greibach-Normalform erstellen
- Kellerautomaten konstruieren

Noch Fragen?

Vorschau

Vorschau

- Klausurwiederholung, daher bitte konkrete Wünsche per Email bis Freitag!

Bis zum nächsten Mal