

Satz: Wird eine Sprache L von einem NKA M akzeptiert, dann wird L auch von einem NKA M' durch leeren Keller akzeptiert, wobei M' nur **einen Zustand** besitzt.

Satz: Wird eine Sprache L von einem NKA M akzeptiert, dann wird L auch von einem NKA M' durch leeren Keller akzeptiert, wobei M' nur **einen Zustand** besitzt.

Beachte: Man kann M' auch als NKA **ohne Zustand** auffassen.

Satz: Wird eine Sprache L von einem NKA M akzeptiert, dann wird L auch von einem NKA M' durch leeren Keller akzeptiert, wobei M' nur **einen Zustand** besitzt.

Beweisidee: Lasse M' den NKA M nicht-deterministisch simulieren

Der Zustand von M wird auf dem Keller von M' gespeichert

Instanz von A auf Keller von M wird realisiert als

(q, A, q') auf Keller von M' mit der Bedeutung:

Wenn diese Instanz das nächste Mal oberstes Kellersymbol ist,
dann ist M in Zustand q ,
und wenn das Kellersymbol unter dieser Instanz betrachtet wird,
dann ist M im Zustand q' .

Beweisidee: Lasse M' den NKA M nicht-deterministisch simulieren

Der Zustand von M wird auf dem Keller von M' gespeichert

Instanz von A auf Keller von M wird realisiert als

(q, A, q') auf Keller von M' mit der Bedeutung:

Wenn diese Instanz das nächste Mal oberstes Kellersymbol ist,
dann ist M in Zustand q ,
und wenn das Kellersymbol unter dieser Instanz betrachtet wird,
dann ist M im Zustand q' .

Kellerinhalt von M im Zustand q $A_k A_{k-1} A_{k-2} \cdots A_1$

entspricht

Kellerinhalt von M' $(q, A_k, p_{k-1}) (p_{k-1}, A_{k-1}, p_{k-2}) (p_{k-2}, A_{k-2}, p_{k-3}) \cdots (p_1, A_1, p_0)$

für geeignete (nicht-deterministisch gewählte) Zustände $p_{k-1}, p_{k-2}, p_{k-3} \cdots p_1, p_0$

Grammatik alternativer Sprachspezifikationsmechanismus

$G = (\Sigma, V, S, P)$

Σ	Terminalalphabet
V	Variablen- (Nicht-terminal) –alphabet
$S \in V$	Startvariable
$P \subset FV F \times F$	“Produktionen” ($F = (\Sigma \cup V)^*$)

Σ, V, S, P müssen endlich sein
 $(\alpha, \beta) \in P$ wird geschrieben als $\alpha \rightarrow \beta$

Grammatik alternativer Sprachspezifikationsmechanismus

$G = (\Sigma, V, S, P)$

Σ	Terminalalphabet
V	Variablen- (Nicht-terminal) –alphabet
$S \in V$	Startvariable
$P \subset FV \times F$	“Produktionen” ($F = (\Sigma \cup V)^*$)

Σ, V, S, P müssen endlich sein
 $(\alpha, \beta) \in P$ wird geschrieben als $\alpha \rightarrow \beta$

G induziert Ableitungsschritt-Relation (Derivationsschritt-Relation)
 $\Rightarrow_G$ auf F durch $\gamma\alpha\gamma' \Rightarrow_G \gamma\beta\gamma'$ wenn $\alpha \rightarrow \beta$ Produktion in P

$\Rightarrow_G^*$ reflexive, transitive Hülle von $\Rightarrow_G$: Ableitungsrelation
(Derivationsrelation) auf F

G induziert Ableitungsschritt-Relation (Derivationsschritt-Relation) $\Rightarrow_G$ auf F durch $\gamma\alpha\gamma' \Rightarrow_G \gamma\beta\gamma'$ wenn $\alpha \rightarrow \beta$ Produktion in P

$\Rightarrow_G^*$ reflexive, transitive Hülle von $\Rightarrow_G$: Ableitungsrelation
(Derivationsrelation) auf F

Die von G generierte Sprache

$$L(G) = \{ w \in \Sigma^* \mid S \Rightarrow_G^* w \}$$

Beispiel: $G = (\Sigma, V, S, P)$ mit $\Sigma = \{a, b, c\}$, $V = \{C, D, S\}$ und

$$P = \{ S \rightarrow aDbc, S \rightarrow \varepsilon, \\ D \rightarrow aDbC, D \rightarrow \varepsilon, \\ Cb \rightarrow bC, \\ Cc \rightarrow cc \}$$

Beispielableitung:

$$\begin{aligned} S &\Rightarrow_G a\underline{D}bc \Rightarrow_G aa\underline{D}bCbcb \Rightarrow_G aaa\underline{D}bCbCbcb \Rightarrow_G aaab\underline{C}bCbcb \Rightarrow_G \\ &aaabb\underline{C}\underline{C}bcb \Rightarrow_G aaabb\underline{C}bCc \Rightarrow_G aaabbb\underline{C}\underline{C}c \Rightarrow_G \\ &aaabbb\underline{C}cc \Rightarrow_G aaabbbcccc \end{aligned}$$

Beispiel: $G = (\Sigma, V, S, P)$ mit $\Sigma = \{a, b\}$, $V = \{C, D, S\}$ und

$$P = \{ S \rightarrow aDbc, S \rightarrow \varepsilon, \\ D \rightarrow aDbC, D \rightarrow \varepsilon, \\ Cb \rightarrow bC, \\ Cc \rightarrow cc \}$$

Behauptung: $L(G) = \{ a^n b^n c^n \mid n \in \mathbb{N} \}$

Beweis: “ $\subseteq$ ”

- 1) Jeder abgeleitete String enthält gleich viele a 's wie b 's wie $\{c, C\}$'s.
- 2) Alle a 's kommen vor allen b 's und vor allen $\{c, C\}$'s vor.
- 3) Die c 's, die in einem abgeleiteten String vorkommen, bilden immer einen Suffix dieses Strings.