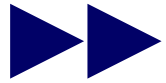


Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie

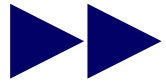


Gliederung der Vorlesung

- 0. Grundbegriffe
- 1. Formale Sprachen/Automatentheorie
 - 1.1. Grammatiken
 - 1.2. Reguläre Sprachen
 - 1.3. Kontextfreie Sprachen**
- 2. Berechnungstheorie
 - 2.1. Entscheidungsprobleme
 - 2.2. Berechenbarkeitsmodelle
 - 2.3. Die Churchsche These
 - 2.4. Unentscheidbarkeit
- 3. Komplexitätstheorie
 - 3.1. Nicht-deterministische Turing Maschinen
 - 3.1 Komplexitätsmaße
 - 3.2. Das P=NP? Problem

Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



Kontextfreie Sprachen

algorithmischer Aspekt

Beispiel: prüfen, ob der Quellcode eines Programm syntaktisch korrekt ist

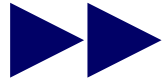
Annahme: unser Rechner schafft 10^9 Operationen pro Sekunde

Länge des Quellcodes	CYK-Algorithmus
100 Zeichen	0.001s
1000 Zeichen	1s
10000 Zeichen	≥ 16 min

... man sollte über Alternativen nachdenken !!!

Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



Kontextfreie Sprachen

Fakt

endliche Automaten können nicht alle kontextfreien Sprachen akzeptieren

Frage/Ansatz

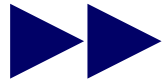
Wie viel muß ein endlicher Automat mehr können?

Randbedingung

... wir wollen nicht „mit Kanonen auf Spatzen schießen“

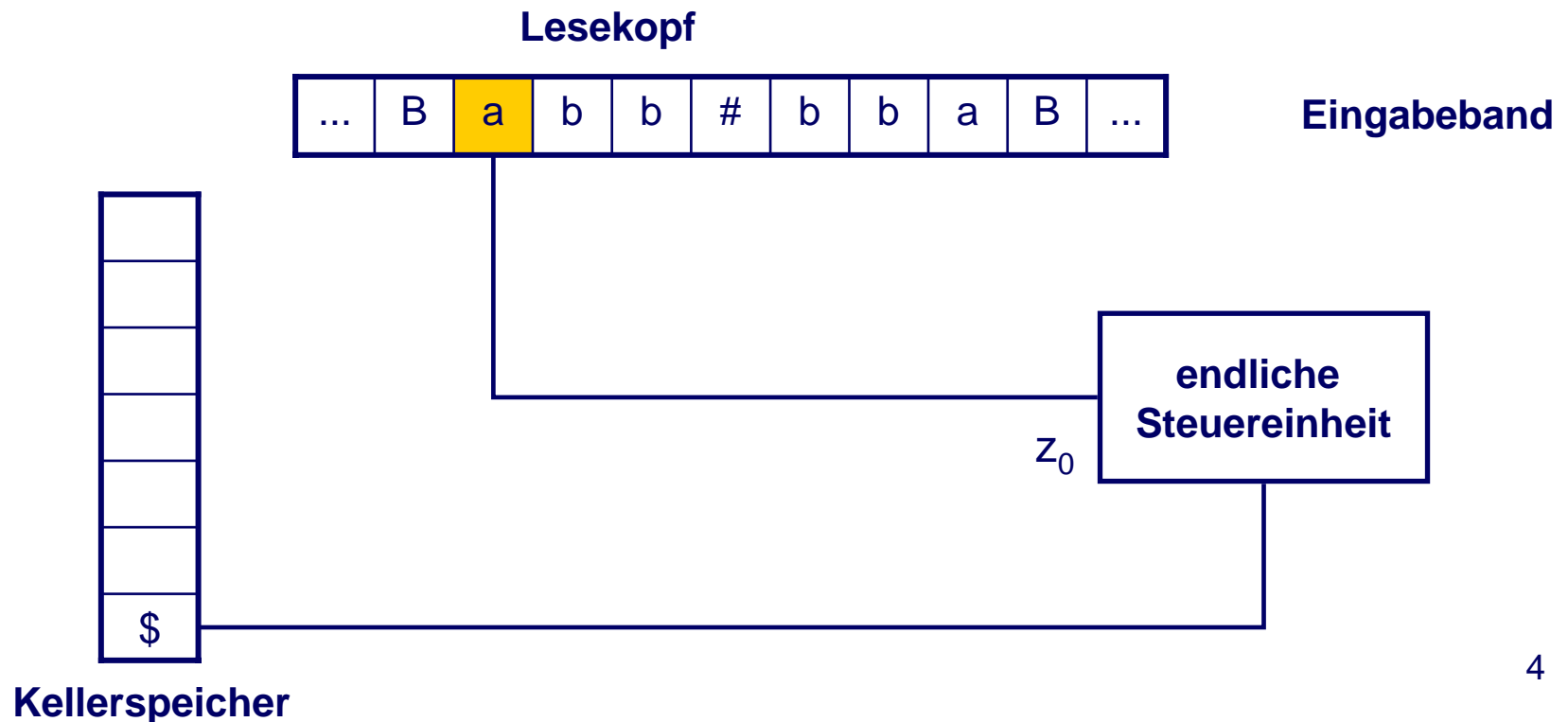
Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



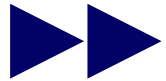
Kontextfreie Sprachen

Beispiel $L = \{ w\#w \mid w \in \{ a,b \}^* \setminus \{ \varepsilon \}; \underline{w}$ ist die gespiegelte Version von $w \}$



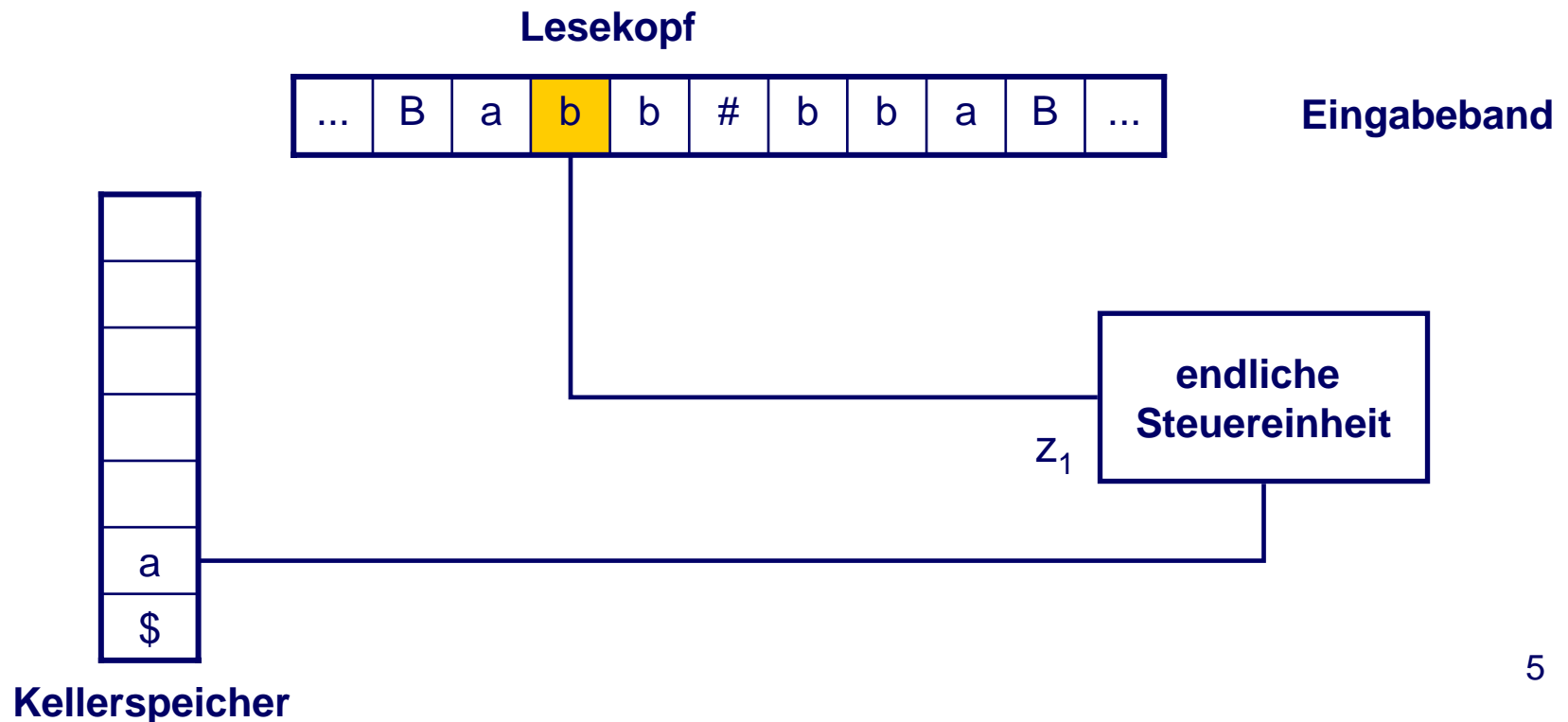
Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



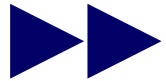
Kontextfreie Sprachen

Beispiel $L = \{ w\#w \mid w \in \{ a,b \}^* \setminus \{ \varepsilon \}; \underline{w}$ ist die gespiegelte Version von $w \}$



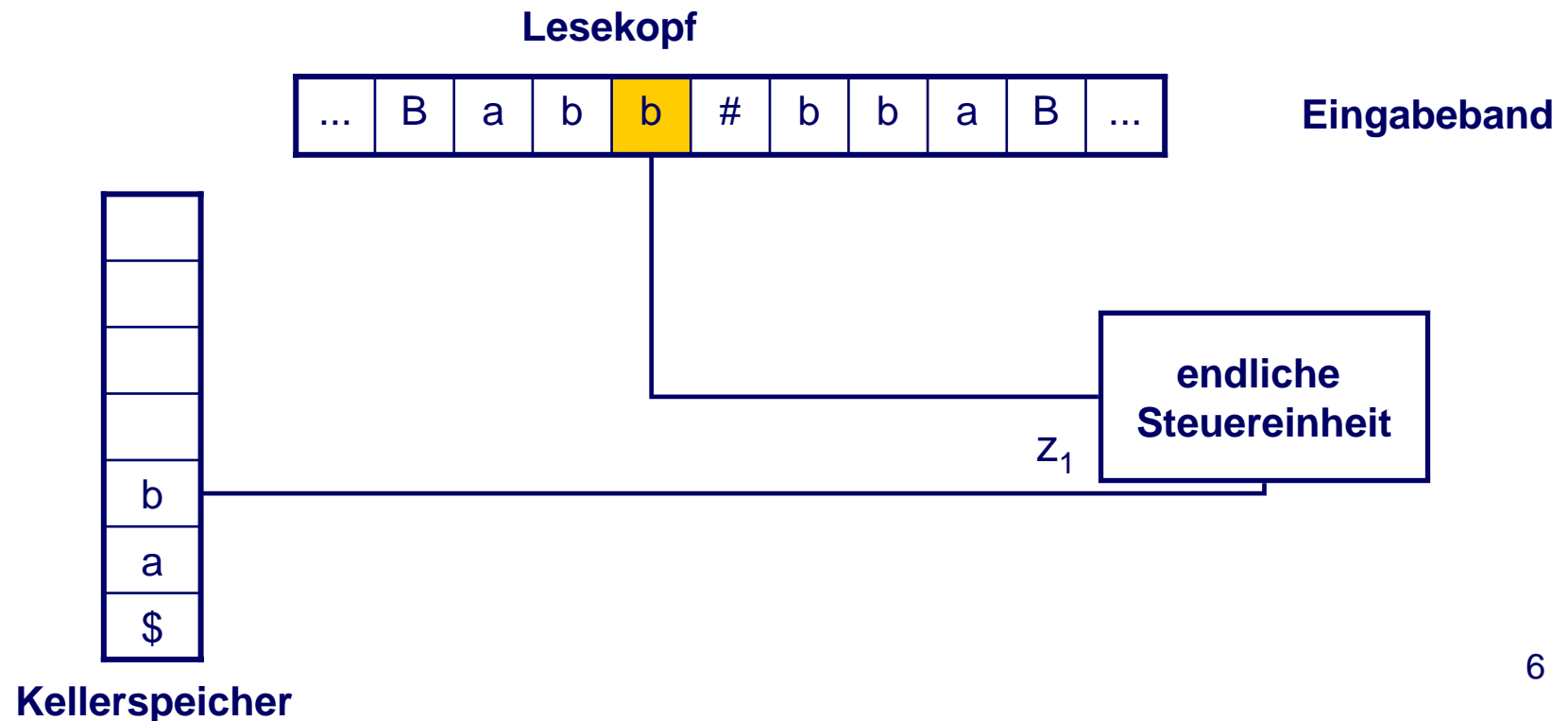
Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



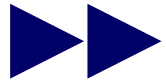
Kontextfreie Sprachen

Beispiel $L = \{ w\#w \mid w \in \{ a,b \}^* \setminus \{ \varepsilon \}; \underline{w}$ ist die gespiegelte Version von $w \}$



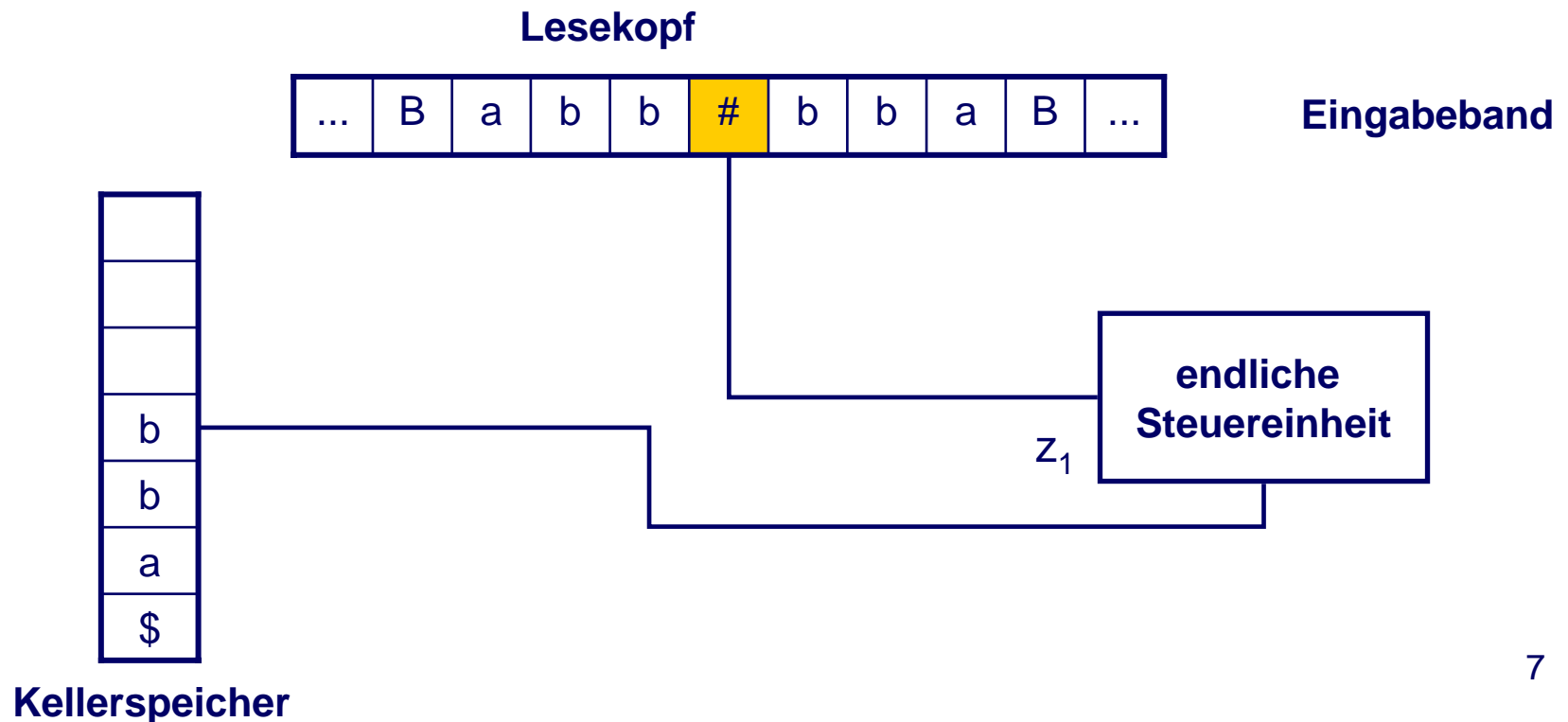
Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



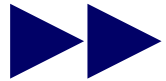
Kontextfreie Sprachen

Beispiel $L = \{ w\#w \mid w \in \{ a,b \}^* \setminus \{ \varepsilon \}; \underline{w}$ ist die gespiegelte Version von $w \}$



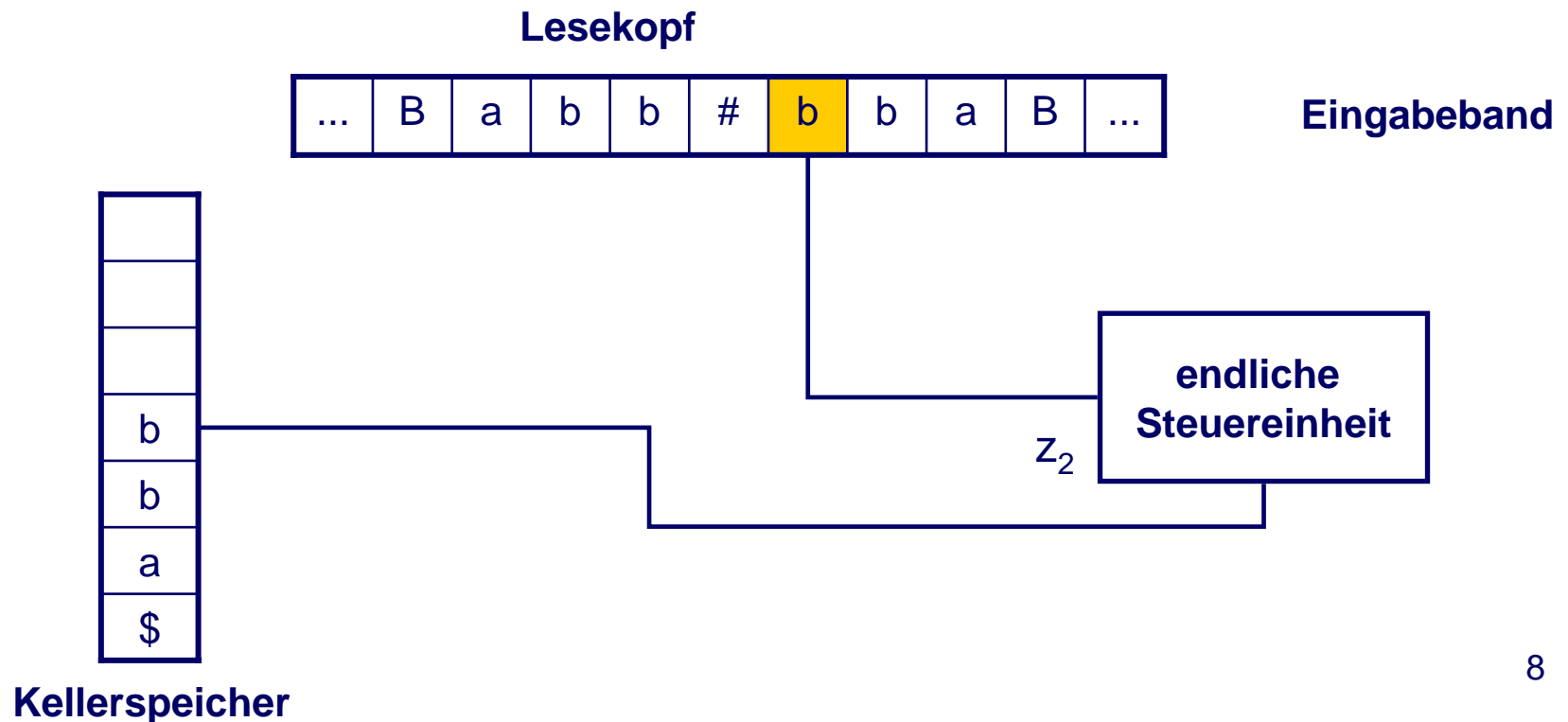
Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



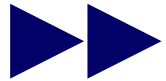
Kontextfreie Sprachen

Beispiel $L = \{ w\#w \mid w \in \{ a,b \}^* \setminus \{ \varepsilon \}; \underline{w}$ ist die gespiegelte Version von $w \}$



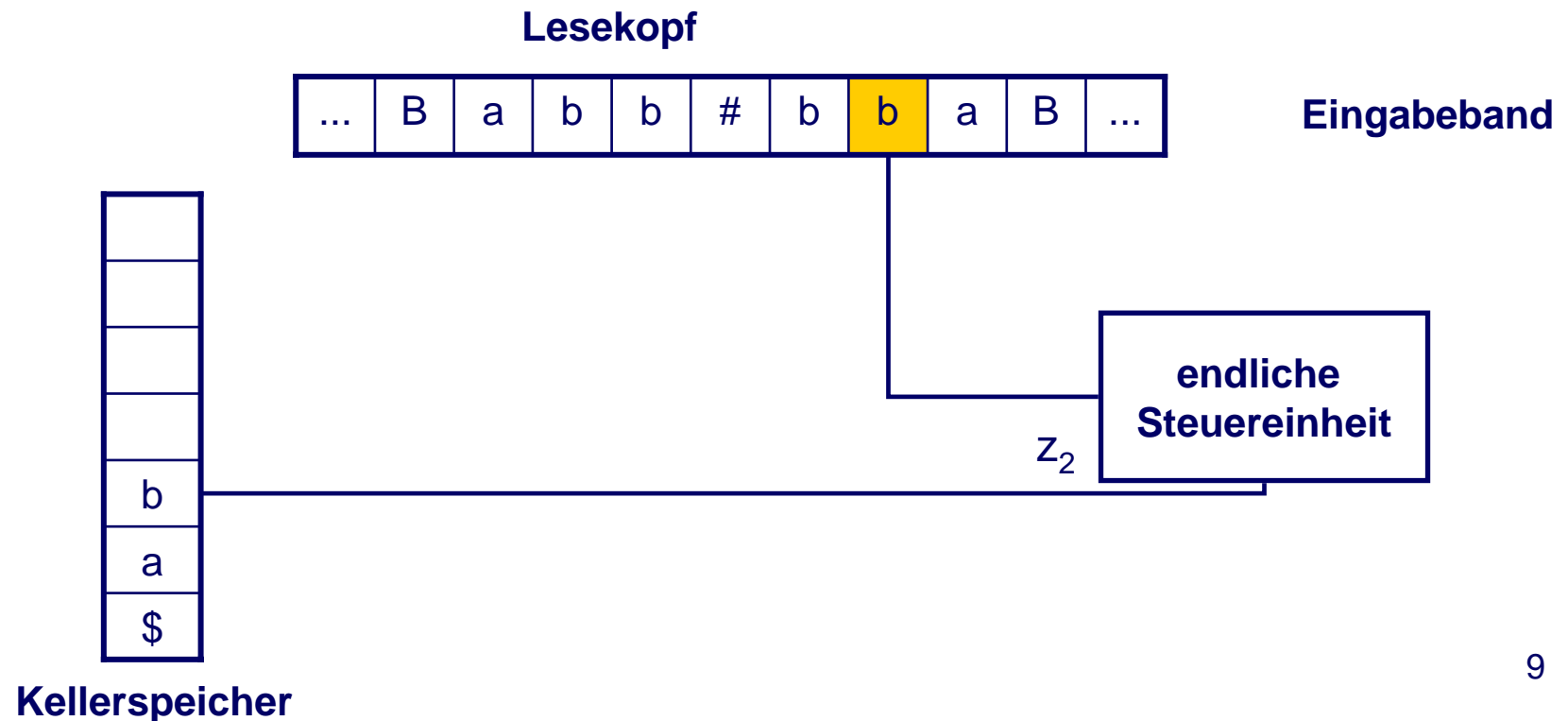
Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



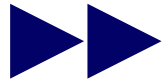
Kontextfreie Sprachen

Beispiel $L = \{ w\#w \mid w \in \{ a,b \}^* \setminus \{ \varepsilon \}; \underline{w}$ ist die gespiegelte Version von $w \}$



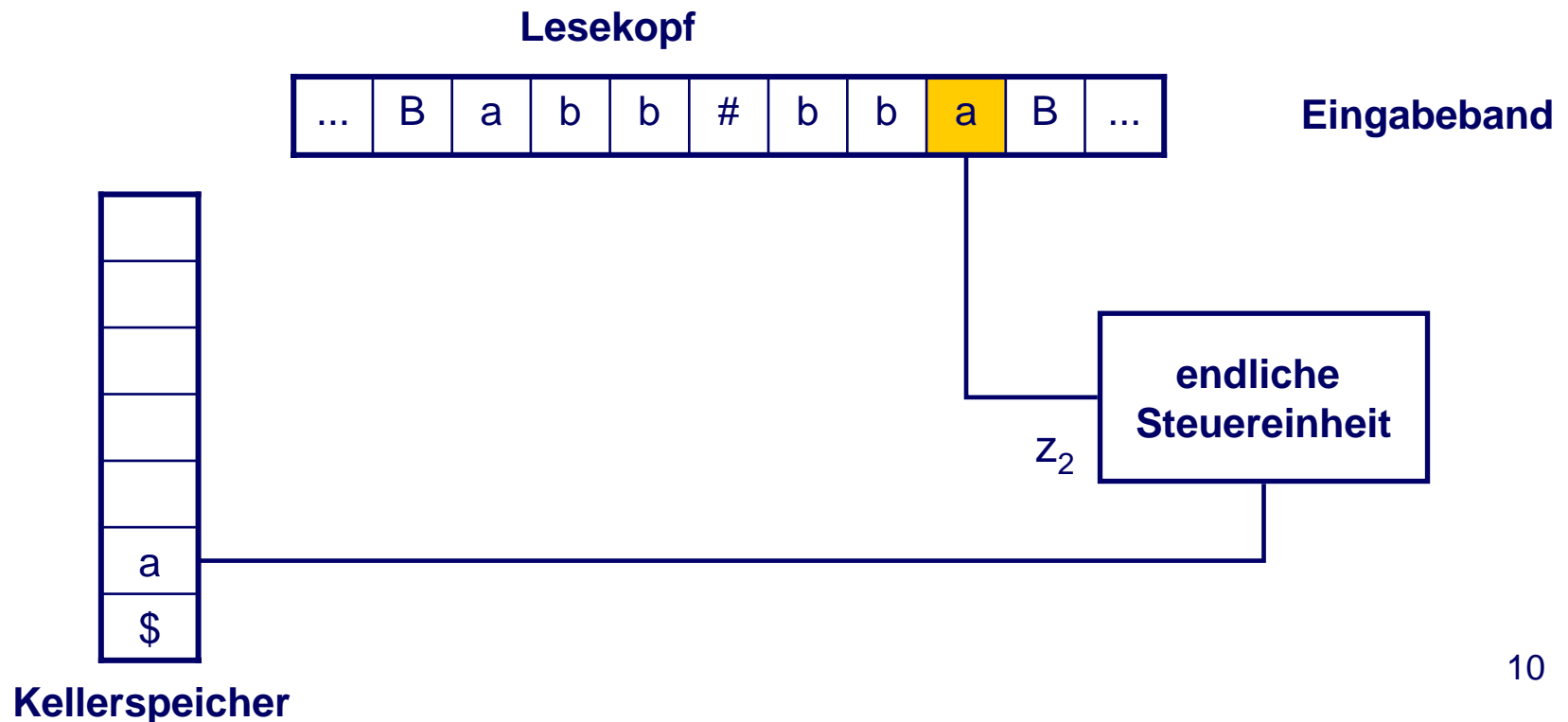
Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



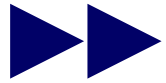
Kontextfreie Sprachen

Beispiel $L = \{ w\#w \mid w \in \{ a,b \}^* \setminus \{ \varepsilon \}; \underline{w}$ ist die gespiegelte Version von $w \}$



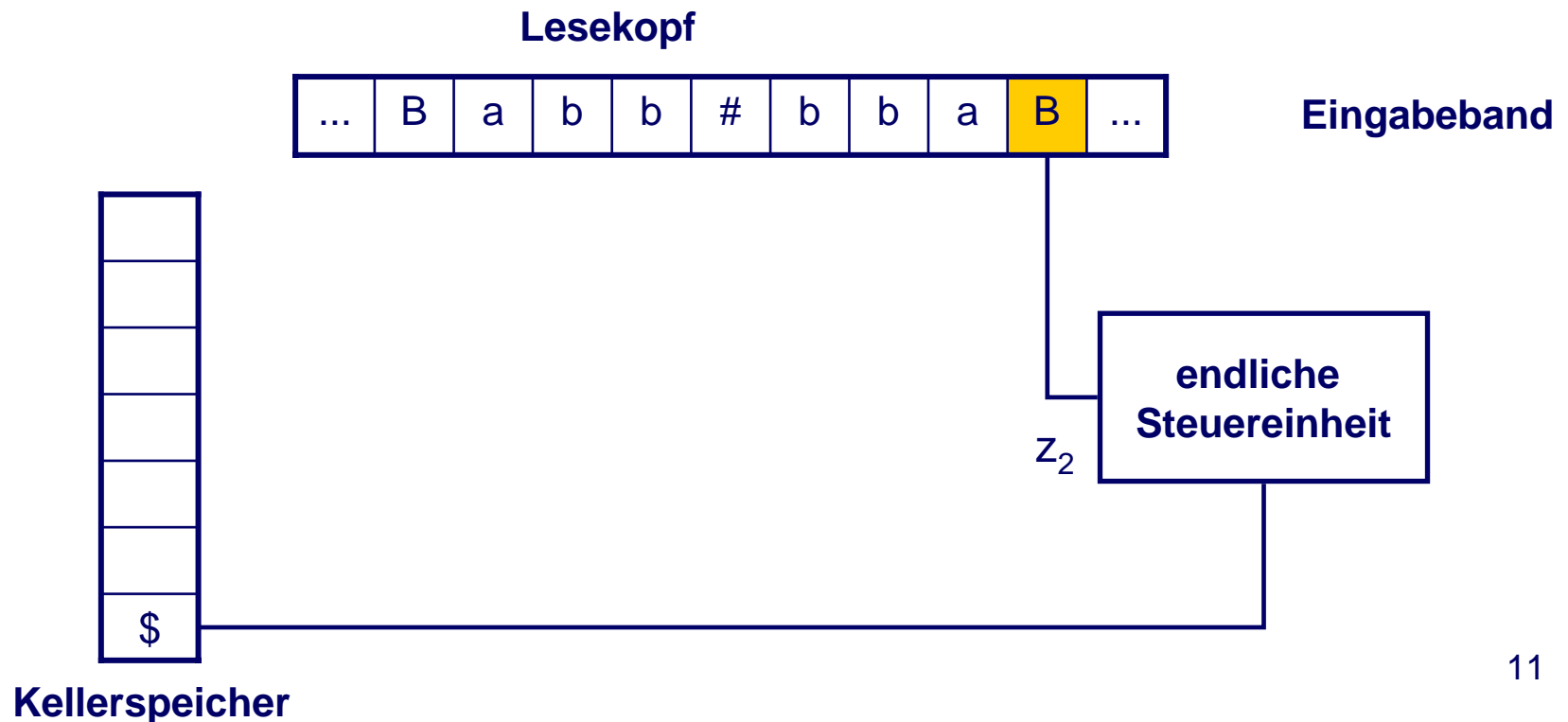
Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



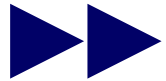
Kontextfreie Sprachen

Beispiel $L = \{ w\#w \mid w \in \{ a,b \}^* \setminus \{ \varepsilon \}; \underline{w}$ ist die gespiegelte Version von $w \}$



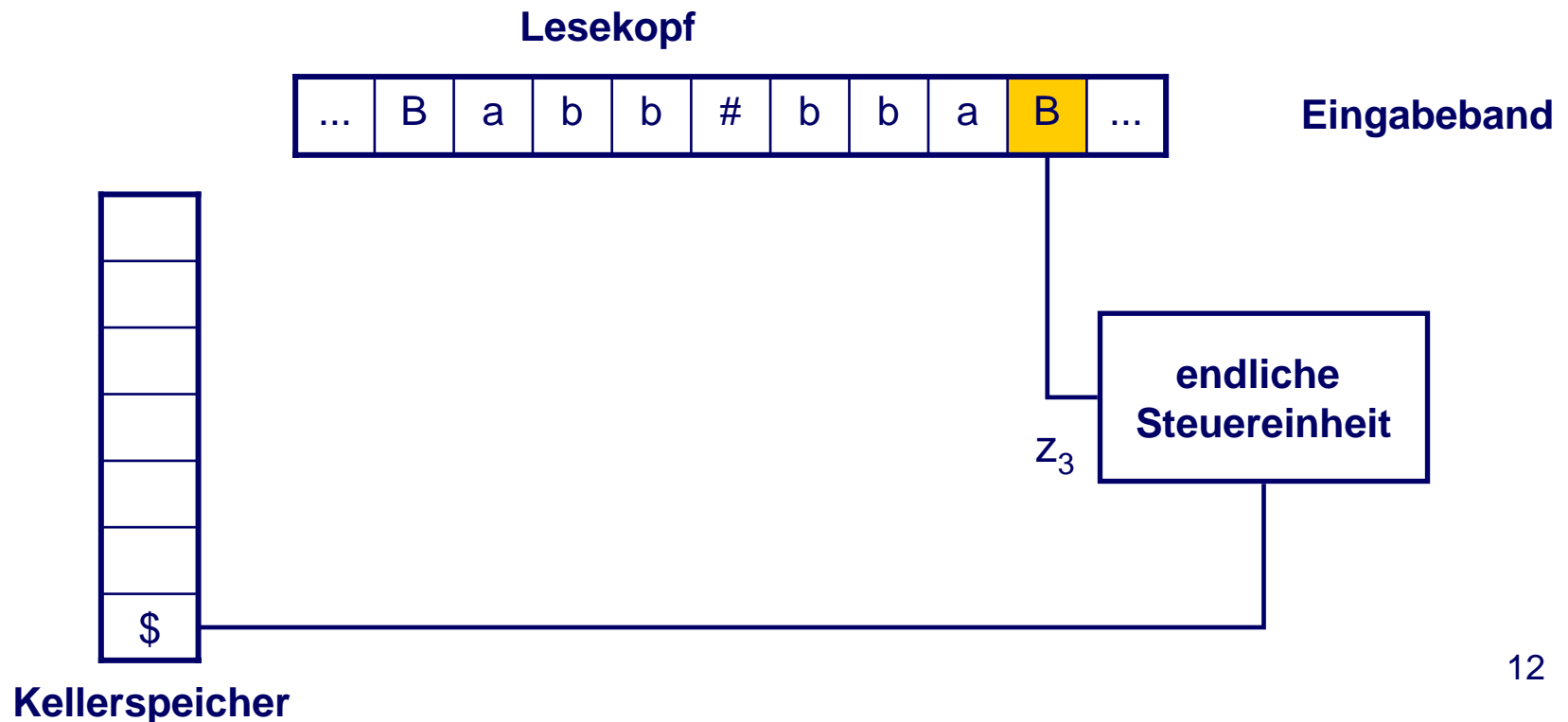
Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



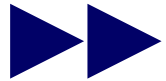
Kontextfreie Sprachen

Beispiel $L = \{ w\#w \mid w \in \{ a,b \}^* \setminus \{ \varepsilon \}; \underline{w}$ ist die gespiegelte Version von $w \}$



Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



Kontextfreie Sprachen

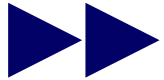
algorithmischer Aspekt

solcherart Automaten können benutzt werden, um das Membership-Problem für kontextfreie Sprachen (/ * in vielen Fällen auch effizient *) zu lösen

... notwendig, Wissen über Kellerspeicher „auffrischen“

Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



Kontextfreie Sprachen

Kellerspeicher (/* angepaßt an unsere Bedürfnisse */)

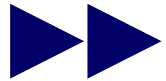
- implementiert das LIFO-Prinzip
- erlaubt Datenelemente zu speichern
- eingeschränkter Zugriff: es kann nur auf das zuletzt gespeicherte Datenelement zugegriffen werden

- Keller K ; initial steht ein $\$$ im Keller K
- endliches Kellularphabet Γ mit $\$ \notin \Gamma$
- Operationen: pop, push

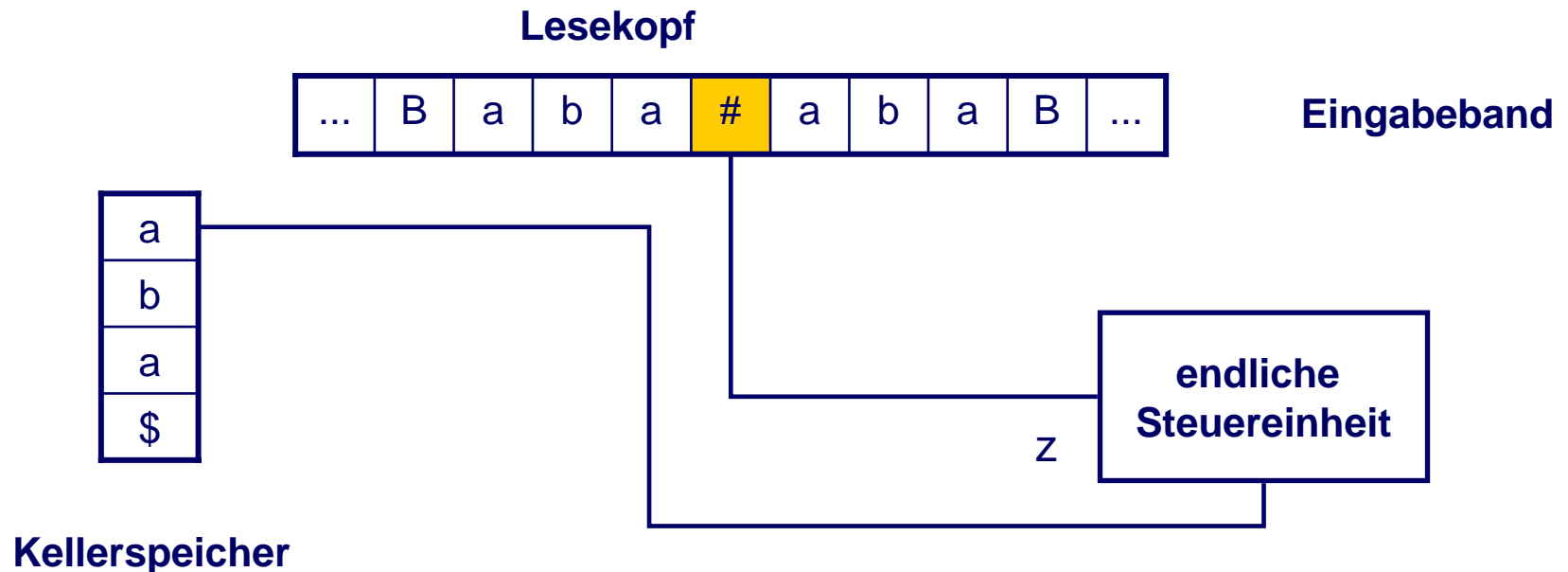
Eigenschaften: $\text{pop}(\text{push}(x, K)) = K$
 $\text{pop}(\$) = \text{Error}$

Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



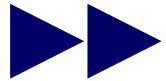
Kontextfreie Sprachen



- auf dem Eingabeband steht das Wort w , für welches zu entscheiden ist, ob es zur Sprache L gehört oder nicht
- das Wort w wird zeichenweise von links nach rechts gelesen
- falls sich der Kellerautomat nach vollständiger Verarbeitung des Wortes w in einem Endzustand befindet, so gehört w zu L ; andernfalls gehört w nicht zu L

Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie

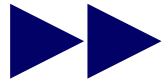


Kontextfreie Sprachen

- das Eingabeband ist in Felder unterteilt; jedes Feld enthält ein Zeichen
- endliche Steuereinheit; mit einem Lesekopf und einem Kellerspeicher verbunden
 - bekommt Information über den Inhalt eines Feldes und mittels pop das zuletzt im Kellerspeicher gespeicherte Zeichen
 - befindet sich in einem von endlich vielen Zuständen
 - legt in Abhängigkeit vom aktuellen Zustand, vom Inhalt des aktuell gesehenen Feldes und vom zuletzt im Kellerspeicher gespeicherte Zeichen fest, was im nächsten Arbeitsschritt passiert
- ein Arbeitsschritt besteht aus:
 - Bewegung des Lesekopfs um ein Feld nach rechts bzw. Lesekopf nicht bewegen
 - Speichern von endlich vielen Zeichen im Kellerspeicher
 - Festlegung des nächsten Zustands

Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



Kontextfreie Sprachen

es sei G eine kontextfreie Grammatik

... man verwendet einen Kellerautomaten, um für ein gegebenes Wort w zu überprüfen, ob es eine Linksableitung für w gibt

Beispiel $L = \{ w\#\underline{w} \mid w \in \{ a,b \}^* \setminus \{ \varepsilon \}; \underline{w} \text{ ist die gespiegelte Version von } w \}$

$$\Sigma = \{ a, b, \# \}$$

$$V = \{ S \}$$

S

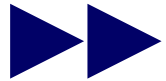
$$S \rightarrow_G aSa \rightarrow_G ab\#ba$$

$$S \rightarrow a\#a \mid b\#b$$

$$S \rightarrow aSa \mid bSb$$

Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



Kontextfreie Sprachen

Beispiel $L = \{ w\# \underline{w} \mid w \in \{ a,b \}^* \setminus \{ \varepsilon \}, \underline{w} \text{ ist die gespiegelte Version von } w \}$

$\Sigma = \{ a, b \}$

$V = \{ S \}$

S

$S \rightarrow a\#a \mid b\#b$

$S \rightarrow aSa \mid bSb$

$S \rightarrow_G aSa \rightarrow_G ab\#ba$

Keller

Eingabewort

\$

ab#ba

S\$

ab#ba

aSa\$

ab#ba

Sa\$

b#ba

b#ba\$

b#ba

#ba\$

#ba

ba\$

ba

a\$

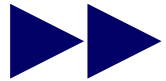
a

\$

Hinweis: zuletzt im Kellerspeicher gespeichertes Zeichen ist unterstrichen

Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



Kontextfreie Sprachen

Beispiel $L = \{ a^n b^n \mid n \geq 1 \}$

$\Sigma = \{ a, b \}$

$V = \{ S \}$

S

$S \rightarrow ab \mid aSb$

$S \rightarrow_G aSb \rightarrow_G aabb$

Keller

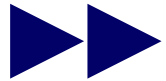
\$
S\$
aSb\$
Sb\$
ab\$
bb\$
b\$
 \$

Eingabewort

aabb
 aabb
 aabb
 abb
 abb
 bb
 b

Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



Kontextfreie Sprachen

Beispiel $L = \{ a^n b^n \mid n \geq 1 \}$

$\Sigma = \{ a, b \}$

$V = \{ S \}$

S

$S \rightarrow ab \mid aSb$

$S \rightarrow_G aSb \rightarrow_G aabb$

Keller

\$

S\$

aSb\$

Sb\$

aSbb\$

Sbb\$

aSbb\$

...

Eingabewort

aabb

aabb

aabb

abb

abb

bb

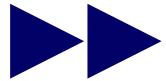
bb

...

... der Automat muß zwangsläufig nicht-deterministisch arbeiten 20

Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



Kontextfreie Sprachen

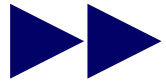
Bestimmungsstücke eines nicht-deterministischen Kellerautomaten

- eine Menge Z von Zuständen
- ein endliches Eingabealphabet Σ von Symbolen
- ein endliches Kelleralphabet Γ
- ein ausgezeichnete Startzustand $z_0 \in Z$
- ein ausgezeichnetes Zeichen $\$ \notin \Gamma$
- eine ausgezeichnete Menge $E \subseteq Z$ (* die Endzustände *)
- eine Überföhrungsfunktion $\delta: Z \times (\Sigma \cup \{\varepsilon\}) \times (\Gamma \cup \{\$\}) \rightarrow 2^{(Z \times \Gamma^*)}$

... man benötigt die nicht-deterministische Variante eines Kellerautomaten,
um alle kontextfreien Sprachen in den Griff zu bekommen !!!

Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



Kontextfreie Sprachen

Beschreibung von nicht-deterministischen Kellerautomaten

Beispiel $L = \{ w\# \underline{w} \mid w \in \{ a,b \}^* \setminus \{ \varepsilon \}, \underline{w} \text{ ist die gespiegelte Version von } w \}$

$$Z = \{ z_0, z_1, z_2, z_3 \}$$

$$\Sigma = \{ a,b,\# \}$$

$$\Gamma = \{ a,b \}$$

$$z_0$$

$$\$$$

$$E = \{ z_3 \}$$

$$\delta(z_0, a, \$) = \{ (z_1, a) \}$$

$$\delta(z_0, b, \$) = \{ (z_1, b) \}$$

$$\delta(z_1, a, a) = \{ (z_1, aa) \}$$

$$\delta(z_1, a, b) = \{ (z_1, ab) \}$$

$$\delta(z_1, b, a) = \{ (z_1, ba) \}$$

$$\delta(z_1, b, b) = \{ (z_1, bb) \}$$

$$\delta(z_1, \#, a) = \{ (z_2, a) \}$$

$$\delta(z_1, \#, b) = \{ (z_2, b) \}$$

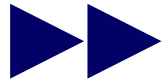
$$\delta(z_2, a, a) = \{ (z_2, \varepsilon) \}$$

$$\delta(z_2, b, b) = \{ (z_2, \varepsilon) \}$$

$$\delta(z_2, \varepsilon, \$) = \{ (z_3, \varepsilon) \}$$

Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



Kontextfreie Sprachen

Beschreibung von nicht-deterministischen Kellerautomaten

Beispiel $L = \{ w\# \underline{w} \mid w \in \{ a,b \}^* \setminus \{ \varepsilon \}, \underline{w} \text{ ist die gespiegelte Version von } w \}$

$$Z = \{ z_0, z_1, z_2, z_3 \}$$

$$\Sigma = \{ a,b,\# \}$$

$$\Gamma = \{ a,b \}$$

$$z_0$$

$$\$$$

$$E = \{ z_3 \}$$

$$\delta(z_0, a, \$) = \{ (z_1, a) \}$$

$$\delta(z_0, b, \$) = \{ (z_1, b) \}$$

$$\delta(z_1, a, a) = \{ (z_1, aa) \}$$

$$\delta(z_1, a, b) = \{ (z_1, ab) \}$$

$$\delta(z_1, b, a) = \{ (z_1, ba) \}$$

$$\delta(z_1, b, b) = \{ (z_1, bb) \}$$

$$\delta(z_1, \#, a) = \{ (z_2, a) \}$$

$$\delta(z_1, \#, b) = \{ (z_2, b) \}$$

$$\delta(z_2, a, a) = \{ (z_2, \varepsilon) \}$$

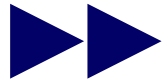
$$\delta(z_2, b, b) = \{ (z_2, \varepsilon) \}$$

$$\delta(z_2, \varepsilon, \$) = \{ (z_3, \varepsilon) \}$$

gehe auf dem Eingabeband weiter;
schreibe erst das a zurück in den Keller
und danach das b in den Keller

Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



Kontextfreie Sprachen

Beschreibung von nicht-deterministischen Kellerautomaten

Beispiel $L = \{ w\# \underline{w} \mid w \in \{ a,b \}^* \setminus \{ \varepsilon \}, \underline{w} \text{ ist die gespiegelte Version von } w \}$

$$Z = \{ z_0, z_1, z_2, z_3 \}$$

$$\Sigma = \{ a,b,\# \}$$

$$\Gamma = \{ a,b \}$$

$$z_0$$

$$\$$$

$$E = \{ z_3 \}$$

$$\delta(z_0, a, \$) = \{ (z_1, a) \}$$

$$\delta(z_0, b, \$) = \{ (z_1, b) \}$$

$$\delta(z_1, a, a) = \{ (z_1, aa) \}$$

$$\delta(z_1, a, b) = \{ (z_1, ab) \}$$

$$\delta(z_1, b, a) = \{ (z_1, ba) \}$$

$$\delta(z_1, b, b) = \{ (z_1, bb) \}$$

$$\delta(z_1, \#, a) = \{ (z_2, a) \}$$

$$\delta(z_1, \#, b) = \{ (z_2, b) \}$$

$$\delta(z_2, a, a) = \{ (z_2, \varepsilon) \}$$

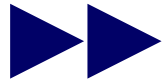
$$\delta(z_2, b, b) = \{ (z_2, \varepsilon) \}$$

$$\delta(z_2, \varepsilon, \$) = \{ (z_3, \varepsilon) \}$$

gehe auf dem Eingabeband weiter;
schreibe nichts in den Keller

Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



Kontextfreie Sprachen

Beschreibung von nicht-deterministischen Kellerautomaten

Beispiel $L = \{ w\# \underline{w} \mid w \in \{ a,b \}^* \setminus \{ \varepsilon \}, \underline{w} \text{ ist die gespiegelte Version von } w \}$

$$Z = \{ z_0, z_1, z_2, z_3 \}$$

$$\Sigma = \{ a,b,\# \}$$

$$\Gamma = \{ a,b \}$$

$$z_0$$

$$\$$$

$$E = \{ z_3 \}$$

$$\delta(z_0, a, \$) = \{ (z_1, a\$) \}$$

$$\delta(z_0, b, \$) = \{ (z_1, b\$) \}$$

$$\delta(z_1, a, a) = \{ (z_1, aa) \}$$

$$\delta(z_1, a, b) = \{ (z_1, ab) \}$$

$$\delta(z_1, b, a) = \{ (z_1, ba) \}$$

$$\delta(z_1, b, b) = \{ (z_1, bb) \}$$

$$\delta(z_1, \#, a) = \{ (z_2, a) \}$$

$$\delta(z_1, \#, b) = \{ (z_2, b) \}$$

$$\delta(z_2, a, a) = \{ (z_2, \varepsilon) \}$$

$$\delta(z_2, b, b) = \{ (z_2, \varepsilon) \}$$

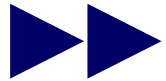
$$\delta(z_2, \varepsilon, \$) = \{ (z_3, \varepsilon) \}$$



gehe auf dem Eingabeband nicht weiter;
schreibe nichts in den Keller

Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



Kontextfreie Sprachen

allgemein

die Zustandsüberföhrungsfunktion δ wird mit Gleichungen der Art $\delta(z,x,y) = \{ (z',w'), \dots, (z'',w') \}$ beschrieben

es sei $(z',w') \in \delta(z,x,y)$

Fall 1: $x \in \Sigma, y \in \Gamma, w' = a_1, \dots, a_{n-1}, a_n$

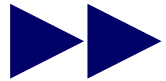
ein Übergang von z nach z' ist möglich, falls x das Zeichen ist, daß der Lesekopf sieht und y das oberste Zeichen im Keller ist

Effekt:

- der Lesekopf geht ein Zeichen nach rechts
- aus dem Keller wird y gelöscht und es werden nacheinander die Zeichen a_n, a_{n-1}, \dots, a_1 in den Keller geschrieben

Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



Kontextfreie Sprachen

allgemein

die Zustandsüberföhrungsfunktion δ wird mit Gleichungen der Art $\delta(z,x,y) = \{ (z',w'), \dots, (z'',w') \}$ beschrieben

es sei $(z',w') \in \delta(z,x,y)$

Fall 2: $x \in \Sigma, y \in \Gamma, w' = \varepsilon$

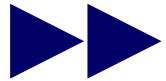
ein Übergang von z nach z' ist möglich, falls x das Zeichen ist, daß der Lesekopf sieht und y das oberste Zeichen im Keller ist

Effekt:

- der Lesekopf geht ein Zeichen nach rechts
- aus dem Keller wird y gelöscht

Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



Kontextfreie Sprachen

allgemein

die Zustandsüberföhrungsfunktion δ wird mit Gleichungen der Art $\delta(z,x,y) = \{ (z',w'), \dots, (z'',w') \}$ beschrieben

es sei $(z',w') \in \delta(z,x,y)$

Fall 3: $x = \varepsilon, y \in \Gamma, w' = a_1, \dots, a_{n-1}, a_n$

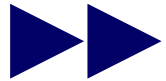
ein Übergang von z nach z' ist möglich, falls y das oberste Zeichen im Keller ist

Effekt:

- der Lesekopf bewegt sich nicht
- aus dem Keller wird y gelöscht und es werden nacheinander die Zeichen a_n, a_{n-1}, \dots, a_1 in den Keller geschrieben

Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



Kontextfreie Sprachen

allgemein

die Zustandsüberföhrungsfunktion δ wird mit Gleichungen der Art $\delta(z,x,y) = \{ (z',w'), \dots, (z'',w') \}$ beschrieben

es sei $(z',w') \in \delta(z,x,y)$

Fall 4: $x = \varepsilon, y \in \Gamma, w' = \varepsilon$

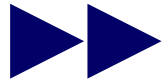
ein Übergang von z nach z' ist möglich, falls x das Zeichen ist, daß der Lesekopf sieht und y das oberste Zeichen im Keller ist

Effekt:

- der Lesekopf bewegt sich nicht
- aus dem Keller wird y gelöscht

Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



Kontextfreie Sprachen

allgemein

die Zustandsüberföhrungsfunktion δ wird mit Gleichungen der Art $\delta(z,x,y) = \{ (z',w'), \dots, (z'',w') \}$ beschrieben

es sei $(z',w') \in \delta(z,x,y)$

Fall 5+i: $\dots, y = \$, \dots$

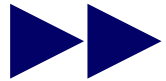
ein Übergang von z nach z' ist möglich, falls der Keller leer ist

Effekt:

...

Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



Kontextfreie Sprachen

es sei $C = [Z, \Sigma, \Gamma, z_0, \$, E, \delta]$ ein Kellerautomat

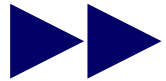
Ein Wort w gehört zu der von C akzeptierten Sprache $L(C)$, falls es eine Möglichkeit gibt, daß C das Wort w vollständig verarbeitet und sich anschließend in einem Endzustand befindet. Falls es keine solche Möglichkeit gibt, gehört w nicht zu $L(C)$.

Hinweis: wenn in einer Situation kein Zustandsübergang möglich ist, so meldet der Kellerautomat einen Fehler

... man könnte auch explizit wie bei endl. Automaten ein Fehlerzustand einführen³¹

Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



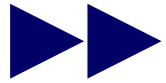
Kontextfreie Sprachen

Zusammenhang zwischen nicht-deterministischen Kellerautomaten und kontextfreien Grammatiken

Zu jeder kontextfreien Grammatik G gibt es einen nicht-deterministischen Kellerautomaten C mit $L(C) = L(G)$.

Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



Kontextfreie Sprachen

Beispiel $L = \{ a^n b^n \mid n \geq 1 \}$

$\Sigma = \{ a, b \}$

$V = \{ S \}$

S

$S \rightarrow ab \mid aSb$

$Z = \{ z_0, z_1, z_2 \}$

$\Sigma = \{ a, b \}$

$\Gamma = \{ a, b, S \}$

z_0

$\$$

$E = \{ z_2 \}$

$\delta(z_0, \varepsilon, \$) = \{ (z_1, S) \}$

$\delta(z_1, \varepsilon, S) = \{ (z_1, ab), (z_1, aSb) \}$

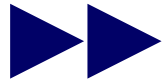
$\delta(z_1, a, a) = \{ (z_1, \varepsilon) \}$

$\delta(z_1, b, b) = \{ (z_1, \varepsilon) \}$

$\delta(z_1, \varepsilon, \$) = \{ (z_2, \varepsilon) \}$

Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



Kontextfreie Sprachen

es sei $G = [\Sigma, V, S, R]$ eine kontextfreie Grammatik

bilde den gesuchten nicht-deterministischen Kellerautomat C wie folgt:

$$Z = \{ z_0, z_1, z_2 \}$$

Σ

$$\Gamma = \Sigma \cup V$$

z_0

$\$$

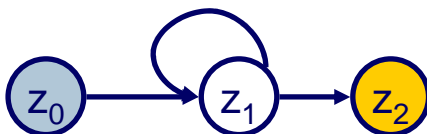
$$E = \{ z_2 \}$$

$$\delta(z_0, \varepsilon, \$) = \{ (z_1, S) \}$$

$$\delta(z_1, x, x) = \{ (z_1, \varepsilon) \} \text{ f\u00fcr alle } x \in \Sigma$$

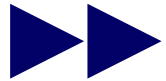
$$\delta(z_1, \varepsilon, \$) = \{ (z_2, \varepsilon) \}$$

$$\delta(z_1, \varepsilon, l) = \{ (z_1, r) \} \text{ f\u00fcr alle Regeln } l \rightarrow r \text{ in } R$$



Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



Kontextfreie Sprachen

Beispiel $L = \{ w\#w \mid w \in \{ a,b \}^* \setminus \{ \varepsilon \}, \underline{w}$ ist die gespiegelte Version von $w \}$

$\Sigma = \{ a, b, \# \}$	$S \rightarrow aHa \mid bHb$
$V = \{ S \}$	$S \rightarrow aSa \mid bSb$
S	$H \rightarrow \#$

$Z = \{ z_0, z_1, z_2 \}$	$\delta(z_0, \varepsilon, \$) = \{ (z_1, S) \}$
---------------------------	---

Σ	$\delta(z_1, a, a) = \{ (z_1, \varepsilon) \}$
----------	--

$\Gamma = \{ a, b, \# \} \cup \{ S, H \}$	$\delta(z_1, b, b) = \{ (z_1, \varepsilon) \}$
---	--

z_0	$\delta(z_1, \#, \#) = \{ (z_1, \varepsilon) \}$
-------	--

$\$$	$\delta(z_1, \varepsilon, \$) = \{ (z_2, \varepsilon) \}$
------	---

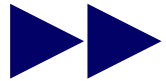
$E = \{ z_2 \}$	
-----------------	--

	$\delta(z_1, \varepsilon, S) = \{ (z_1, aHa), (z_1, bHb), (z_1, aSa), (z_1, bSb) \}$
--	--

	$\delta(z_1, \varepsilon, H) = \{ (z_1, \#) \}$
--	---

Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



Kontextfreie Sprachen

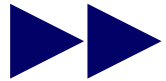
Zwischenfazit

- zu jeder gegebenen kontextfreien Grammatik G kann man einen nicht-deterministischen Kellerautomat C mit $L(C) = L(G)$ konstruieren
- umgekehrt kann man zu jedem gegebenen nicht-deterministischen Kellerautomat C eine kontextfreie Grammatik G mit $L(G) = L(C)$ konstruieren
- aber ...

nicht-deterministische Kellerautomaten kann man „nicht“ sinnvoll implementieren (/* Laufzeit */)

Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



Kontextfreie Sprachen

Spezialfall: Deterministische Kellerautomaten

$$Z = \{ z_0, z_1, z_2, z_3 \}$$

$$\Sigma = \{ a, b, \# \}$$

$$\Gamma = \{ a, b \}$$

$$z_0$$

$$\$$$

$$E = \{ z_3 \}$$

$$\delta(z_0, a, \$) = \{ (z_1, a) \}$$

$$\delta(z_0, b, \$) = \{ (z_1, b) \}$$

$$\delta(z_1, a, a) = \{ (z_1, aa) \}$$

$$\delta(z_1, a, b) = \{ (z_1, ab) \}$$

$$\delta(z_1, b, a) = \{ (z_1, ba) \}$$

$$\delta(z_1, b, b) = \{ (z_1, bb) \}$$

$$\delta(z_1, \#, a) = \{ (z_2, a) \}$$

$$\delta(z_1, \#, b) = \{ (z_2, b) \}$$

$$\delta(z_2, a, a) = \{ (z_2, \varepsilon) \}$$

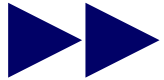
$$\delta(z_2, b, b) = \{ (z_2, \varepsilon) \}$$

$$\delta(z_2, \varepsilon, \$) = \{ (z_3, \varepsilon) \}$$

... mit diesem Kellerautomaten kann man zu einem Wort w in genau w Schritten entscheiden, ob w zu $L(C)$ gehört oder nicht

Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



Kontextfreie Sprachen

Spezialfall: Deterministische Kellerautomaten

Idee: ... in jeder Situation ist nur eine „Aktion“ möglich

mögliche Situationen sind definiert durch

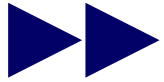
- den aktuellen Zustand, das Zeichen auf dem Eingabeband, das der Lesekopf sieht, und das zuletzt in den Kellerspeicher geschriebene Zeichen

mögliche Aktionen haben zur Folge, daß sich

- der Zustand und der Inhalt des Kellerspeichers ändert und der Lesekopf ein Zeichen nach rechts geht
- der Zustand und der Inhalt des Kellerspeichers ändert und der Lesekopf auf demselben Zeichen steht

Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



Kontextfreie Sprachen

Spezialfall: Deterministische Kellerautomaten

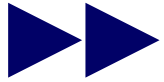
es sei $D = [Z, \Sigma, \Gamma, z_0, \$, E, \delta]$ ein Kellerautomat

D ist ein deterministischer Kellerautomat, falls für alle $z \in Z$, alle $a \in \Sigma$ und alle $y \in \Gamma \cup \{ \$ \}$ gilt:

- wenn $\delta(z, \varepsilon, y) \neq \emptyset$,
so ist $\text{card}(\delta(z, \varepsilon, y)) = 1$ und $\delta(z, x, y) = \emptyset$ für alle $x \in \Sigma$
- wenn $\delta(z, a, y) \neq \emptyset$,
so ist $\text{card}(\delta(z, a, y)) = 1$ und $\delta(z, \varepsilon, y) = \emptyset$

Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



Kontextfreie Sprachen

der „Nutzen“ von deterministische Kellerautomaten

es sei L eine kontextfreie Sprache, für die es einen deterministischen Kellerautomaten D mit $L(D) = L$ gibt

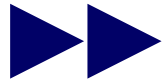
Dann kann man einen Algorithmus angeben, mit dem man zu jedem Wort $w \in \Sigma^*$ in Zeit $O(|w|)$ entscheiden kann, ob $w \in L$ oder $w \notin L$ gilt.

wie das im Detail geht: siehe Vorlesung „Compilerbau“

Hinweis: man kann auch den Kellerautomaten D „direkt“ verwenden; dann muß man mit den Übergängen vom Typ „ $\delta(z, \varepsilon, y)$ “ aufpassen (/ * zusätzliche Steuerung */)

Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



Kontextfreie Sprachen

Beispiel: prüfen, ob der Quellcode eines Programm syntaktisch korrekt ist *)

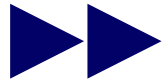
Annahme: unser Rechner schafft 10^9 Operationen pro Sekunde

Länge des Quellcodes	es gibt einen determ. Kellerautomat für L	es gibt keinen determ. Kellerautomat für L
100 Zeichen	0.0000001 s	0.001 s
1000 Zeichen	0.000001 s	1 s
10000 Zeichen	0.00001 s	≥ 16 min

*) ... alle syntaktisch korrekten Programme bilden eine kontextfreie Sprache

Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



Kontextfreie Sprachen

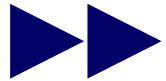
Zwischenfazit

- die Syntax von Programmiersprachen beschreibt man derart, daß sicher gestellt ist, daß es für die beschriebene Sprache einen deterministischen Kellerautomaten gibt

... das erfordert sowohl Verständnis für die zugrunde liegende Problematik als auch Kreativität !!!

Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



Kontextfreie Sprachen

Begründung (/ * Teil 1 * /)

Es gibt kontextfreie Sprachen L mit folgender Eigenschaft:

Es gibt keinen deterministischen Kellerautomaten D mit $L(D) = L$.

Beispiel: $L = \{ w\underline{w} \mid w \in \{ a,b \}^* \setminus \{ \varepsilon \}, \underline{w}$ ist die gespiegelte Version von w }

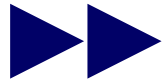
Einsicht:

man muß beim „Design“ aufpassen

... das gilt nicht nur für die Syntax von Programmiersprachen

Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



Kontextfreie Sprachen

Begründung (/ * Teil 2 *)

Es gibt keinen Algorithmus, der folgendes leistet:

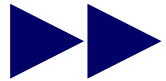
- der Algorithmus akzeptiert als Eingabe eine kontextfreie Grammatik G
- falls es einen deterministischen Kellerautomat D mit $L(D) = L(G)$ gibt, so liefert der Algorithmus das Resultat „ja“
- andernfalls liefert der Algorithmus das Resultat „nein“

Einsicht:

man kann nicht damit rechnen, daß einem die Arbeit durch „Computer-Tools“ entscheidend vereinfacht wird

Theoretische Informatik

Kap 1: Formale Sprachen/Automatentheorie



Kontextfreie Sprachen

Begriffe:

- kontextfreie Grammatik, Linksableitung, Chomsky-Normalform
- deterministischer/nicht-deterministischer Kellerautomat

Sätze / Zusammenhänge:

- kf. Grammatik und Chomsky-Normalform
- kf. Grammatik und nicht-det. KA
- det. KA können echt „weniger“
- Pumping-Lemma

Methoden / Techniken / Algorithmen:

- Übersetzung (/* kf. Grammatik \rightarrow Chomsky-Normalform, kf. Grammatik \rightarrow nicht-det. KA */)
- CYK-Algorithmus
- Anwendung des Pumping-Lemma