



Formale Sprachen und Automaten

TINF20B1



Markus Eble

Kellerautomaten und Typ-2-Grammatiken

- erster Ansatz: Top-Down-Syntaxanalyse
 - modifizierte Kellerautomaten
- alternativer Ansatz: Bottom-Up-Syntaxanalyse

Syntaxanalyse mit Kellerautomaten

- ▶ Endliche Automaten sind äquivalent zu regulären Grammatiken
Endliche Automaten sind das Maschinenmodell das reguläre Grammatiken akzeptiert
- ▶ Nichtdeterministische Kellerautomaten sind äquivalent zu kontextfreien Grammatiken
Nichtdeterministische Kellerautomaten sind das Maschinenmodell das kontextfreie Grammatiken akzeptiert

Satz 4.11

Für jede formale Sprache L sind beiden folgenden Aussagen äquivalent:

- ▶ L kann von einem nichtdeterministischen Kellerautomaten erkannt werden.
- ▶ L kann von einer kontextfreien Grammatik erzeugt werden.

Beweisidee

Wir beschränken uns auf die folgende Richtung:

- ▶ Gegeben: kontextfreie Grammatik $G = (N, T, X_0, P)$
- ▶ Gesucht: Kellerautomat K mit $L(K) = L(G)$.

O.B.d.A. sei $* \notin N \cup T$.

Wähle

- ▶ Eingabealphabet: T
- ▶ Zustände: $Z = \{z_0, z, z_+\}$
- ▶ Startzustand: z_0
- ▶ Akzeptierende Zustände: $F = \{z_+\}$
- ▶ Kellularphabet: $Y = \{*\} \cup T \cup N$
- ▶ Kellernfangssymbol: $*$

Beweisidee

Die Arbeitsweise von K wird durch die folgenden Aktionen beschrieben:

- ▶ Am Anfang: $f(z_0, *, \varepsilon) = \{(z, X_0)\}$
- ▶ wenn oben auf dem Keller ein Nichtterminalsymbol $X \in N$:
„Produktionsschritt“ für Produktion $X \rightarrow w$:
 - ▶ $(z; w) \in f(z, X, \varepsilon)$
 - ▶ kein Eingabesymbol gelesen
- ▶ wenn oben auf dem Keller ein Terminalsymbol $x \in T$:
„Leseschritt“ für $x \in T$:
 - ▶ $(z, \varepsilon) \in f(z, x, x)$
 - ▶ Für $x \neq x'$ gibt es aber keine Aktion: $f(z, x, x') = \emptyset$.
- ▶ Am Ende: $f(z, *, \varepsilon) = \{(z_+, *)\}$

Beweisidee

- ▶ Die Entscheidung, ob ein Produktionsschritt oder ein Leseschritt stattfinden soll ist **deterministisch**.
 - ▶ Falls Nichtterminal auf Keller: Produktionsschritt
 - ▶ Falls Terminal auf Keller: Leseschritt
- ▶ Die Entscheidung, welcher Leseschritt stattfinden soll ist **deterministisch**.
 - ▶ Lese genau das Terminalzeichen, das auf dem Keller liegt
- ▶ Die Entscheidung, welcher Produktionsschritt stattfinden soll ist **nichtdeterministisch**.
 - ▶ Für das Nichtterminal auf dem Keller können mehrere Produktionen in der Grammatik existieren.

Beweisidee

Es gilt:

- ▶ Wenn es für ein Wort w eine Linksableitung in G gibt, dann gibt es eine Berechnung von K , nach der w akzeptiert wird.
- ▶ Kann umgekehrt K ein Wort w akzeptieren, dann muss es eine Linksableitung dafür in G geben.

Beispiel

Grammatik $G = (\{S\}, \{ (,) \}, S, \{ S \rightarrow () \})$, Eingabe $()$.

gelesene Eingabe	neuer Zustand	neuer Kellerinhalt	verwendete Produktion
	z_0	$*$	
	z	$S *$	
	z	$()*$	$S \rightarrow ()$
$($	z	$) *$	
$()$	z	$*$	
$()$	z_+	$*$	

Beispiel

Grammatik $G = (\{S\}, \{ (,) \}, S, \{ S \rightarrow \varepsilon \mid (S) \})$, Eingabe $(())$.

gelesene Eingabe	neuer Zustand	neuer Kellerinhalt	verwendete Produktion
	z_0	$*$	
	z	$S *$	
	z	$(S) *$	$S \rightarrow (S)$
$($	z	$S) *$	
$($	z	$(S)) *$	$S \rightarrow (S)$
$(($	z	$S)) *$	
$(($	z	$)) *$	$S \rightarrow \varepsilon$
$(()$	z	$) *$	
$(())$	z	$*$	
$(())$	z_+	$*$	

Satz 4.12

Bei der Top-down-Syntaxanalyse „erzeugt“ der Kellerautomat eine Linksableitung.

Beispiel

Grammatik $G = (\{S\}, \{ (,) \}, S, \{ S \rightarrow \varepsilon \mid (S) \mid SS \})$, Eingabe $(()) ()$.

gelesene Eingabe	neuer Zustand	neuer Kellerinhalt	verwendete Produktion
	z_0	$*$	
	z	$S *$	
	z	$SS *$	$S \rightarrow SS$
	z	$(S)S *$	$S \rightarrow (S)$
$($	z	$S)S *$	
$($	z	$(S))S *$	$S \rightarrow (S)$
$(($	z	$S))S *$	
$(($	z	$))S *$	$S \rightarrow \varepsilon$
$(()$	z	$)S *$	
$((())$	z	$S *$	

Beispiel

Grammatik $G = (\{S\}, \{ (,) \}, S, \{ S \rightarrow \varepsilon \mid (S) \mid SS \})$, Eingabe $(()) ()$.

gelesene Eingabe	neuer Zustand	neuer Kellerinhalt	verwendete Produktion
(())	z	$S *$	
(())	z	$(S) *$	$S \rightarrow (S)$
(()) (z	$S) *$	
(()) (z	$) *$	$S \rightarrow \varepsilon$
(()) ()	z	$*$	
(()) ()	z_+	$*$	

Beispiel – alternativer Weg?

Grammatik $G = (\{S\}, \{ (,) \}, S, \{S \rightarrow \varepsilon | (S) | SS\})$, Eingabe $(()) ()$.

gelesene Eingabe	neuer Zustand	neuer Kellerinhalt	verwendete Produktion
	z_0	$*$	
	z	$S *$	
	z	$SS *$	$S \rightarrow SS$
	z	$(S)S *$	$S \rightarrow (S)$
$($	z	$S)S *$	
$($	z	$(S))S *$	$S \rightarrow (S)$
$(($	z	$S))S *$	
$(($	z	$(S)))S *$	$S \rightarrow (S)$

An dieser Stelle ist der Kellerautomat in einer **Sackgasse**, weil das nächste Eingabesymbol $)$ ist, aber auf dem Keller $($ liegt!

Beobachtung

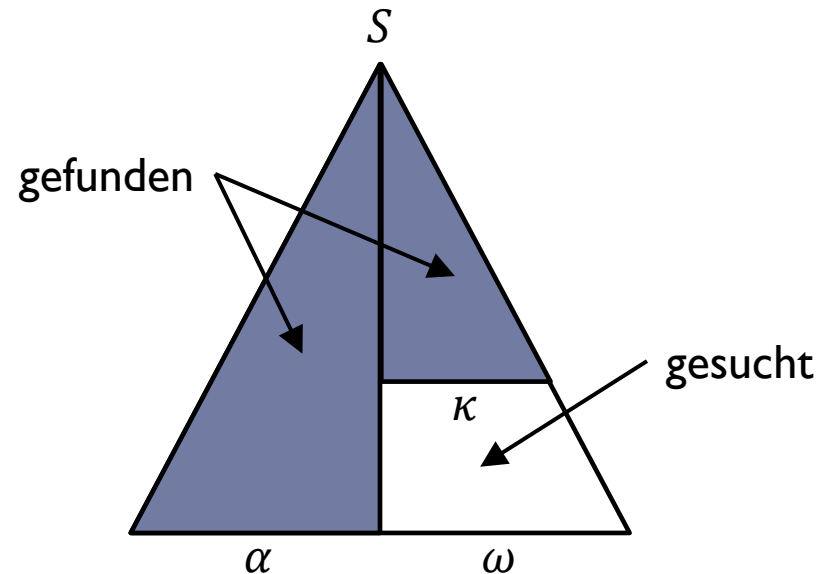
- ▶ Grammatik $G = (\{S\}, \{ (,) \}, S, \{S \rightarrow \varepsilon \mid (S) \mid SS\})$, Eingabe $(()) ()$.
- ▶ Verwendete Produktionen in der Reihenfolge der Tabelle:
 $S \rightarrow SS, S \rightarrow (S), S \rightarrow (S), S \rightarrow \varepsilon, S \rightarrow (S), S \rightarrow \varepsilon$
- ▶ Es wird immer das linke Nichtterminal im Keller ersetzt.
- ▶ Der Kellerautomat erzeugt eine Linksableitung:
 $S \Rightarrow \underline{S}S \Rightarrow (\underline{S})S \Rightarrow ((\underline{S}))S \Rightarrow (())\underline{S} \Rightarrow (())(\underline{S}) \Rightarrow (()) ()$

Top-Down-Syntaxanalyse

- ▶ Kellerautomat „konstruiert“ Ableitungsbaum von oben nach unten.
- ▶ α : bereits gelesener Teil der Eingabe
- ▶ ω : noch ausstehender Teil der Eingabe
- ▶ κ : Kellerinhalt
- ▶ Situation unmittelbar vor Produktionsschritt:

schon klar: $S \xRightarrow{l}^* \alpha\kappa$

noch zu prüfen: $\kappa \xRightarrow{l}^* \omega ?$



Satz 4.12

Bei der Top-down-Syntaxanalyse „erzeugt“ der Kellerautomat eine Linksableitung.

Beobachtung

Nichtdeterministische Kellerautomaten sind unpraktisch.

Mögliche Auswege:

- ▶ **simuliere den Kellerautomaten deterministisch**
 - ▶ systematische Suche nach einer akzeptierenden Berechnung
 - ▶ Vorsicht: unendlichen Rekursion droht.
 - ▶ Außerdem: Zeitaufwand evtl. zu groß
 - ▶ Backtracking erschwert Aktualisierung von Datenstrukturen
 - ▶ Besser nicht ...
- ▶ **schränke Kellerautomaten ein**
 - ▶ Extremfall: nur deterministische Variante
 - ▶ Günstiger: erlaube Vorausschau im Eingabestrom
- ▶ **vergiss Kellerautomaten und mache die Syntaxanalyse anders**
 - ▶ auch erzwungen, wenn die Syntax nicht kontextfrei

Bottom-Up Syntaxanalyse

- ▶ Die Bottom-Up Syntaxanalyse ist ein alternativer Ansatz der ebenfalls mit 2 Schritten arbeitet
- ▶ **Leseschritt:**
Das nächste Eingabesymbol wird gekellert.
- ▶ **Reduktionsschritt:**
Oberste Kellersymbole bilden die rechte Seite einer Produktion, die durch zugehörige linke Seite ersetzt wird.

Modifizierte Kellerautomaten

- ▶ Kellerautomat kann in einem Schritt mehrere oberste Kellersymbole lesen.
- Mit „normalem“ Kellerautomaten simulierbar, erleichtert aber die Beschreibung wesentlich.
- ▶ Wir benutzen diese Erweiterung um die ganze rechte Seite einer Produktion zu lesen
- ▶ Beim „Auslesen“ eines Wortes, dessen Symbole oben auf dem Keller liegen, ist das oberste Kellersymbol das letzte Wortsymbol, das zweitoberste Kellersymbol das vorletzte Wortsymbol, usw.

Beispiel

Grammatik $G = (\{S\}, \{ (,) \}, S, \{S \rightarrow ()\})$, Eingabe $()$.

neuer Kellerinhalt	neuer Zustand	noch nicht gelesen	verwendete Produktion
*	z_0	$()$	
* (z)	
* ()	z		
* S	z		$S \rightarrow ()$
* S	z_+		

Beispiel

Grammatik $G = (\{S\}, \{ (,) \}, S, \{ S \rightarrow \varepsilon \mid (S) \})$, Eingabe $(())$.

neuer Kellerinhalt	neuer Zustand	noch nicht gelesen	verwendete Produktion
*	z_0	$(())$	
* (z	$()$	
* ((z	$)$	
* ((S	z	$)$	$S \rightarrow \varepsilon$
* ((S)	z	$)$	
* (S	z	$)$	$S \rightarrow (S)$
* (S)	z		
* S	z		$S \rightarrow (S)$
* S	z_+		

Beispiel

Grammatik $G = (\{S\}, \{ (,) \}, S, \{ S \rightarrow \varepsilon | (S) | SS \})$, Eingabe $(()) ()$.

neuer Kellerinhalt	neuer Zustand	noch nicht gelesen	verwendete Produktion
*	z_0	$(()) ()$	
* (z	$() ()$	
* ((z	$) ()$	
* ((S	z	$) ()$	$S \rightarrow \varepsilon$
* ((S)	z	$) ()$	
* (S	z	$) ()$	$S \rightarrow (S)$
* (S)	z	$)$	
* S	z	$)$	$S \rightarrow (S)$

Beispiel

Grammatik $G = (\{S\}, \{ (,) \}, S, \{S \rightarrow \varepsilon | (S) | SS\})$, Eingabe $(()) ()$.

neuer Kellerinhalt	neuer Zustand	noch nicht gelesen	verwendete Produktion
* S	z	$()$	$S \rightarrow (S)$
* $S($	z	$)$	
* $S(S$	z	$)$	$S \rightarrow \varepsilon$
* $S(S)$	z		
* SS	z		$S \rightarrow (S)$
* S	z		$S \rightarrow SS$
* S	z_+		

Beobachtung

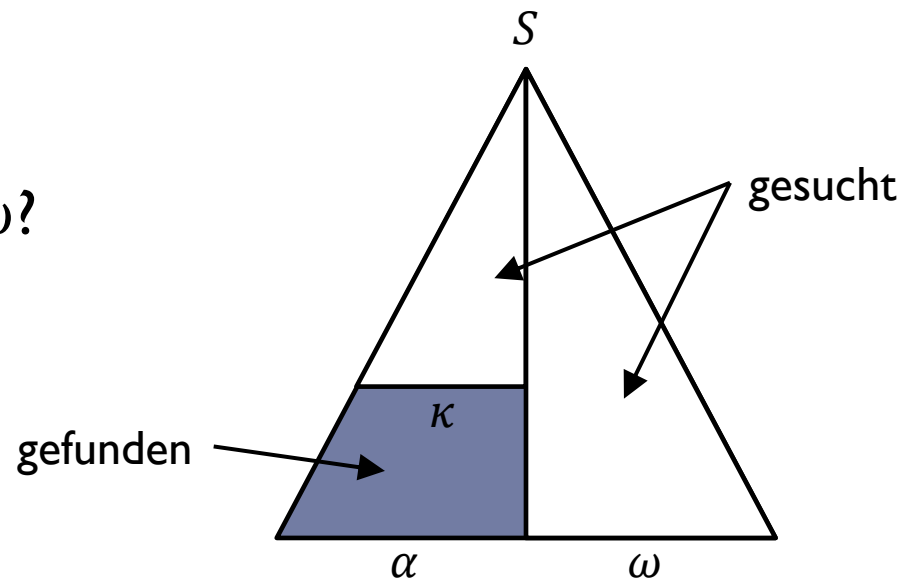
- ▶ Grammatik $G = (\{S\}, \{ (,) \}, S, \{ S \rightarrow \varepsilon \mid (S) \mid SS \})$, Eingabe $(()) ()$.
- ▶ Verwendete Produktionen in der Reihenfolge der Tabelle:
 $S \rightarrow \varepsilon, S \rightarrow (S), S \rightarrow (S), S \rightarrow \varepsilon, S \rightarrow (S), S \rightarrow SS$
- ▶ Es wird immer das rechteste Nichtterminal im Keller ersetzt.
- ▶ Der Kellerautomat erzeugt rückwärts eine Rechtsableitung:
 $S \Rightarrow S\underline{S} \Rightarrow S(\underline{S}) \Rightarrow \underline{S}() \Rightarrow (\underline{S})() \Rightarrow ((\underline{S}))() \Rightarrow (())()$

Bottom-up-Syntaxanalyse

- ▶ Der Kellerautomat versucht, rückwärts eine Rechtsableitung des Eingabewortes gemäß der zu Grunde gelegten Grammatik zu finden.
- ▶ Situation unmittelbar vor Reduktionsschritt:

schon klar: $\kappa \xRightarrow{*r} \alpha$

noch zu prüfen: $S \xRightarrow{*r} \kappa\omega$?



Bottom-up-Syntaxanalyse

▶ Leseschritt:

Das nächste Eingabesymbol w_r wird gelesen und gekellert.
Das erhält die Eigenschaft $\kappa \Rightarrow^* \alpha$

▶ Reduktionsschritt:

Kellerautomat ersetzt rechte Seite einer Produktion oben auf dem Keller durch linke Seite.

Ist $\kappa = \kappa' \gamma$ und $X_r \rightarrow \gamma$ benutzte Produktion, gilt:

$$\kappa' X \Rightarrow \kappa' \gamma = \kappa \Rightarrow^* \alpha.$$

▶ Ziel:

- ▶ $\kappa = S$ und
- ▶ die gesamte Eingabe w gelesen, also $\omega = \varepsilon$.
- ▶ Dann $S = \kappa \xRightarrow[r]{*} \alpha = w$.

Satz 4.13

Der Kellerautomat findet bei Bottom-up-Syntaxanalyse die Produktionen einer Rechtsableitung „von hinten nach vorne“.

Beobachtung

- ▶ **Nichtdeterministische Entscheidungen bei der Bottom-up-Syntaxanalyse:**
 - ▶ Es kann sowohl ein Lese- als auch ein Reduktionsschritt möglich sein.
 - ▶ Bei einem Reduktionsschritt können unterschiedlich lange Kellerenden reduziert werden.
 - ▶ Bei einem Reduktionsschritt kann unter mehreren Produktionen mit gleicher rechter Seite ausgewählt werden.

Beobachtung

- ▶ Will man eine Grammatik so gestalten, dass die Syntaxanalyse deterministisch ist,
 - ▶ muss man bei der **Bottom-Up-Syntaxanalyse** auf die **rechten** Seiten der Produktionen achten und
 - ▶ muss man bei der **Top-Down-Syntaxanalyse** auf die **linken** Seiten der Produktionen achten.
- ▶ Die rechten Seiten einer Typ-2-Grammatik sind variantenreicher ...

Zusammenfassung

- ▶ Nichtdeterministische Kellerautomaten und kontextfreie Grammatiken sind äquivalente Beschreibungsmittel.
- ▶ Bei Top-Down-Syntaxanalyse wird eine Linksableitung von vorne nach hinten konstruiert.
- ▶ Bei Bottom-Up-Syntaxanalyse wird eine Rechtsableitung von hinten nach vorne konstruiert.

Ausblick

Wie kommt man von den nichtdeterministischen Kellerautomaten weg?

Grammatikkonstrukte die Probleme machen:

- ▶ Linksrekursion bei der Top-Down Analyse
- ▶ ε -Produktionen
- ▶ Lookahead in der Eingabe

Ausblick

- ▶ Top-Down und Bottom-Up Analyse sind die Basis für Compiler
- ▶ Das einfachste Rezept:
Der rekursiv absteigende Parser
[Rekursiver Abstieg – Wikipedia](#)
[Recursive descent parser – Wikipedia](#)
ist eine Umsetzung der Top-Down Syntax-Analyse