

# Nichtdeterministischer Kellerautomat

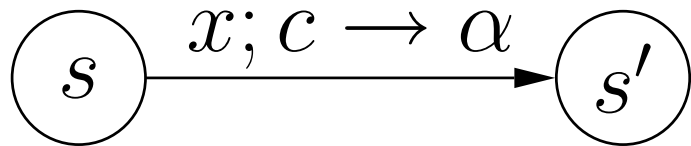
- ▶ endlicher Automat mit Zusatzspeicher in Form eines Kellers (**Stapel, Stack**) mit Speicheroperationen pro Übergang
- ▶ **Keller über  $X$** :  $w \in X^*$  mit den Operationen **push**, **head**, **pop** (d.h.  $push(x, u) = xu$ ,  $head(xu) = x$ ,  $pop(xu) = u$  für alle  $x \in X$ ,  $u \in X^*$ )

## Nichtdeterministischer Kellerautomat

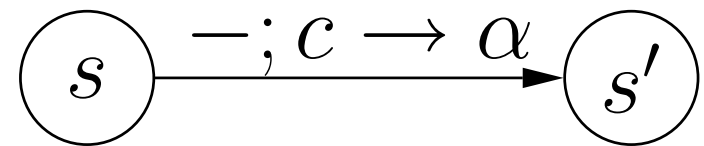
$K = (Z, I, C, d, s_0, F, c_0)$  mit

- $Z$ : endliche Menge von **Zuständen**
- $I$ : endliches **Eingabealphabet** mit  $- \notin I$
- $C$ : endliche Menge von **Kellersymbolen**
- $d: Z \times (I \cup \{-\}) \times C \leadsto Z \times C^*$ :  
**Zustandsüberführung**
- $s_0 \in Z$ : **Startzustand**
- $F \subseteq Z$ : **Endzustände**
- $c_0 \in C$ : **initiales Kellersymbol**

# Graphische Darstellung



für  $(s', \alpha) \in d(s, x, c)$



für  $(s', \alpha) \in d(s, -, c)$

# Konfiguration

Eine **Konfiguration**  $con = (s, w, \gamma)$  besteht aus einem Zustand  $s \in Z$ , einem Wort  $w \in I^*$  und einem Kellerwort  $\gamma \in C^*$ .

## Wichtige Konfigurationen

- **Anfangskonfiguration:**  $con_0 = (s_0, w, c_0)$
- **Endkonfiguration:**  $con_F = (s, \lambda, \gamma)$  mit  $s \in F$

## Folgekonfiguration

$(s, xv, c\gamma) \vdash (s', v, \alpha\gamma)$ , falls  $(s', \alpha) \in d(s, x, c)$

$(s, v, c\gamma) \vdash (s', v, \alpha\gamma)$ , falls  $(s', \alpha) \in d(s, -, c)$

### ► Schreibweise:

$$con = con_0 \vdash con_1 \vdash \dots \vdash con_n = con'$$

kann durch

$$\boxed{con \vdash^n con'} \quad \text{oder} \quad \boxed{con \vdash^* con'}$$

abgekürzt werden.

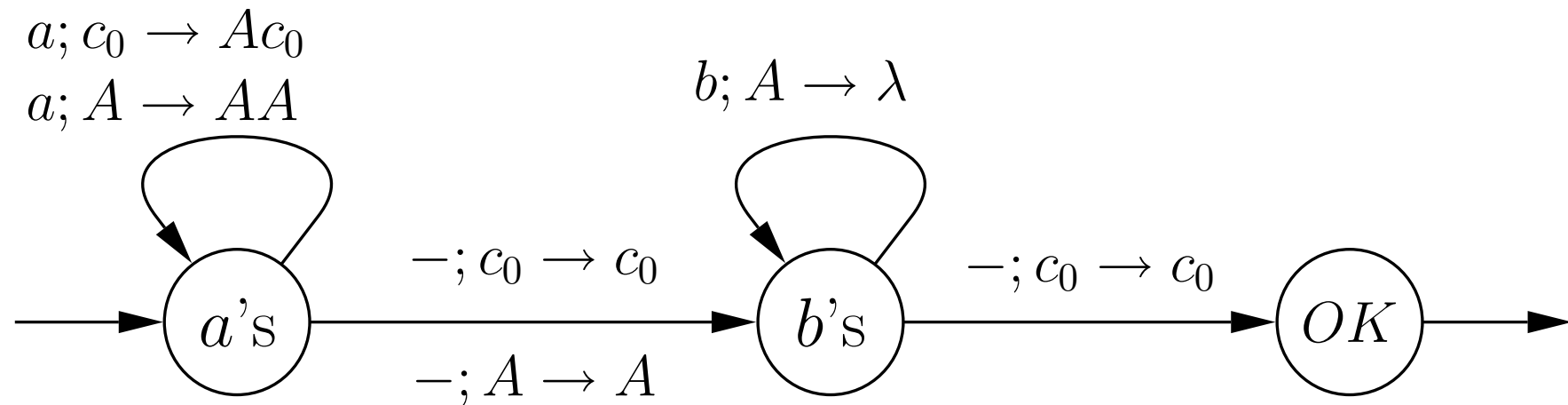
# Erkannte Sprache

Gegeben:  $K = (Z, I, C, d, s_0, F, c_0)$

## Erkannte Sprache

- $K$  **erkennt**  $w \in I^*$ , falls  $(s_0, w, c_0) \xrightarrow{*} (s'', \lambda, \gamma)$  für  $s'' \in F$ .
- Die Menge aller von  $K$  erkannten Wörter bildet die **erkannte Sprache**  $L(K)$ .

## Beispiel



Erkannte Sprache:  $\{a^n b^n \mid n \in \mathbb{N}\}$

Erkennen von  $a^2 b^2$ :

$(a's, aabb, c_0) \vdash (a's, abb, Ac_0) \vdash (a's, bb, AAc_0) \vdash$   
 $(b's, bb, AAc_0) \vdash (b's, b, Ac_0) \vdash (b's, \lambda, c_0) \vdash (OK, \lambda, c_0)$

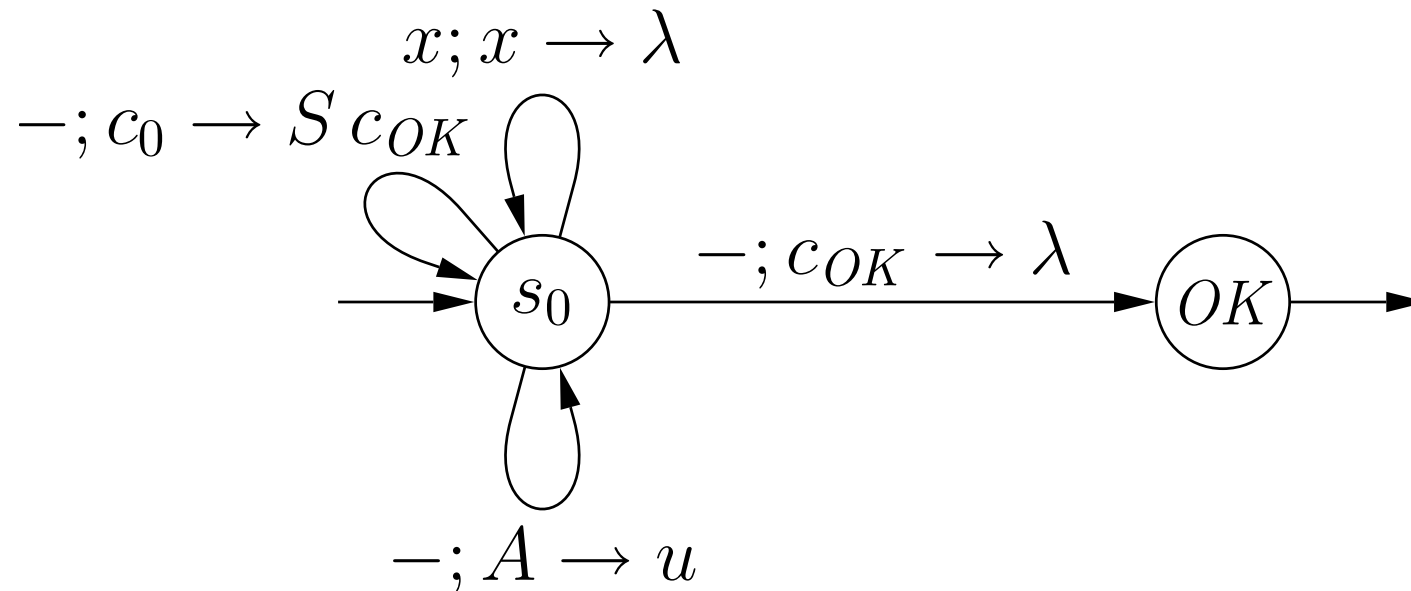
# Deterministischer Kellerautomat

- Jede Konfiguration hat **höchstens eine** Folgekonfiguration.
- Lineare Spracherkennung (wie endl. Aut.)
- Praktischer Einsatz bei Syntaxanalyse von Programmiersprachen
- Nicht jede kontextfreie Sprache wird von deterministischen Kellerautomaten erkannt.



# Übersetzung kontextfreier Grammatiken in Kellerautomaten

Gegeben: kontextfreie Grammatik  $G = (N, T, P, S)$  mit  $c_0, c_{OK} \notin N \cup T$ .

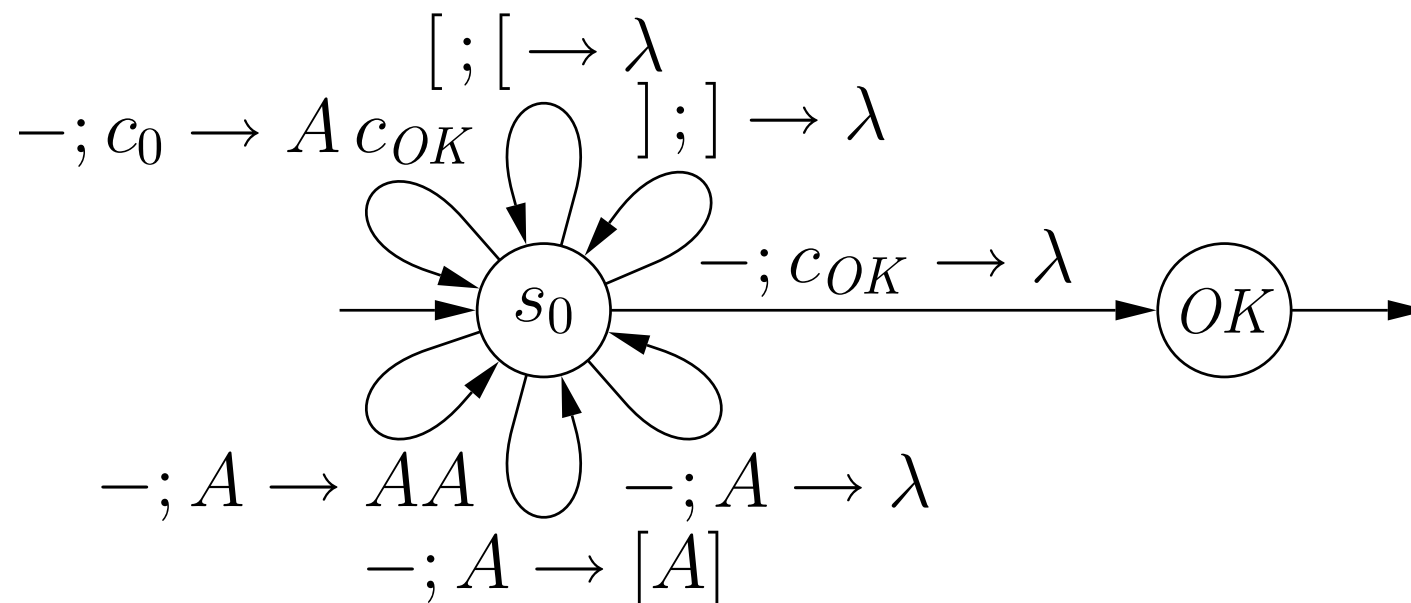


wobei  $x \in T$  und  $A ::= u \in P$

## Beispiel: Klammergebirge

$$G = (\{A\}, \{[, ]\}, \{A ::= AA \mid [A] \mid \lambda\}, A)$$

$L(G)$ : alle korrekten Klammerungen über dem Klammerpaar  $[$  und  $]$ .



# Erzeugen und Erkennen von $[]$

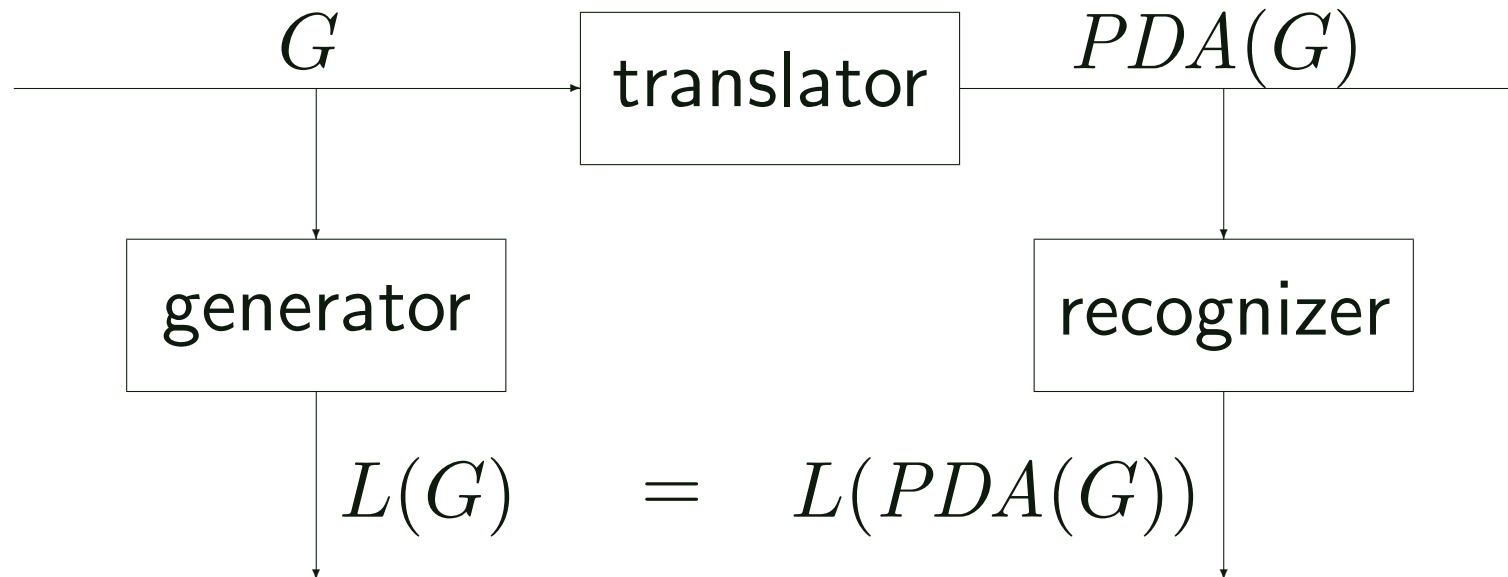
$$\triangleright A \xrightarrow{A ::= [A]} A \xrightarrow{A ::= \lambda} []$$



$$(s_0, [], c_0) \vdash (s_0, [], Ac_{OK}) \vdash (s_0, [], [A]c_{OK}) \vdash$$

$$(s_0, ], A]c_{OK}) \vdash (s_0, ], ]c_{OK}) \vdash (s_0, \lambda, c_{OK}) \vdash (OK, \lambda, \lambda)$$

# Korrektheit der Übersetzung



## Linksableitungen

Sei  $G = (N, T, P, S)$  eine kontextfreie Grammatik. Eine **Linksableitung** ist eine Ableitung  $u_1 \xrightarrow{P} u_2 \xrightarrow{P} \cdots \xrightarrow{P} u_n$ , bei der in jedem Schritt  $u_i \xrightarrow{A_i ::= v_i} u_{i+1}$  ( $1 \leq i < n$ ) das am weitesten links stehende Nichtterminal ersetzt wird, d.h.  $u_i = x_i A_i y_i$  und  $u_{i+1} = x_i v_i y_i$  mit  $x_i \in T^*$ .

**Schreibweise:**  $-\ell \rightarrow$

## Beispiel

$G = (\{E\}, \{+, *, (, ), id\}, P, E)$  mit den Regeln

$$E ::= E + E \mid E * E \mid (E) \mid id$$

Es gibt zwei verschiedene Linksableitungen für  $id + id * id$ :

$$\begin{aligned} 1. \quad & E \longrightarrow E + E \longrightarrow id + E \longrightarrow id + E * E \\ & \longrightarrow id + id * E \longrightarrow id + id * id \end{aligned}$$

$$\begin{aligned} 2. \quad & E \longrightarrow E * E \longrightarrow E + E * E \longrightarrow id + E * E \\ & \longrightarrow id + id * E \longrightarrow id + id * id \end{aligned}$$

## Lemma

Sei  $G = (N, T, P, S)$  eine kontextfreie Grammatik.  
Dann lässt sich jede Ableitung  $A \xrightarrow[P]{*} v$  mit  $A \in N$ ,  
 $v \in T^*$  in eine Linksableitung  $A \xrightarrow[P]{\ell^*} v$  umformen.

# Pumping-Lemma für kontextfreie Sprachen

## Satz

Zu jeder kontextfreien Sprache  $L$  existiert eine natürliche Zahl  $p \in \mathbb{N}$ , so dass gilt:

Ist  $z \in L$  mit  $\text{length}(z) \geq p$ , dann lässt sich  $z$  schreiben als  $z = uvwxy$ , wobei  $\text{length}(vwx) \leq p$  ist und  $vx \neq \lambda$ , und für alle  $i \in \mathbb{N}$  gilt:

$$uv^iwx^iy \in L.$$