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

$$K = (Z, I, C, d, s_0, F, c_0) \text{ mit}$$

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

Graphische Darstellung

$$s \xrightarrow{x;\,c\,\longrightarrow\,\alpha} s'$$

für $(s', \alpha) \in d(s, x, c)$

$$s \xrightarrow{-;\,c\,\longrightarrow\,\alpha} s'$$

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

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

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

- Schreibweise:

$$\boxed{con = con_0 \vdash\!\!\!\!\longrightarrow con_1 \vdash\!\!\!\!\longrightarrow \cdots \vdash\!\!\!\!\longrightarrow con_n = con'}$$

kann durch

$$\boxed{con \overset{n}{\vdash\!\!\!\!\longrightarrow} con'} \text{ oder } \boxed{con \overset{*}{\vdash\!\!\!\!\longrightarrow} con'}$$

abgekürzt werden.

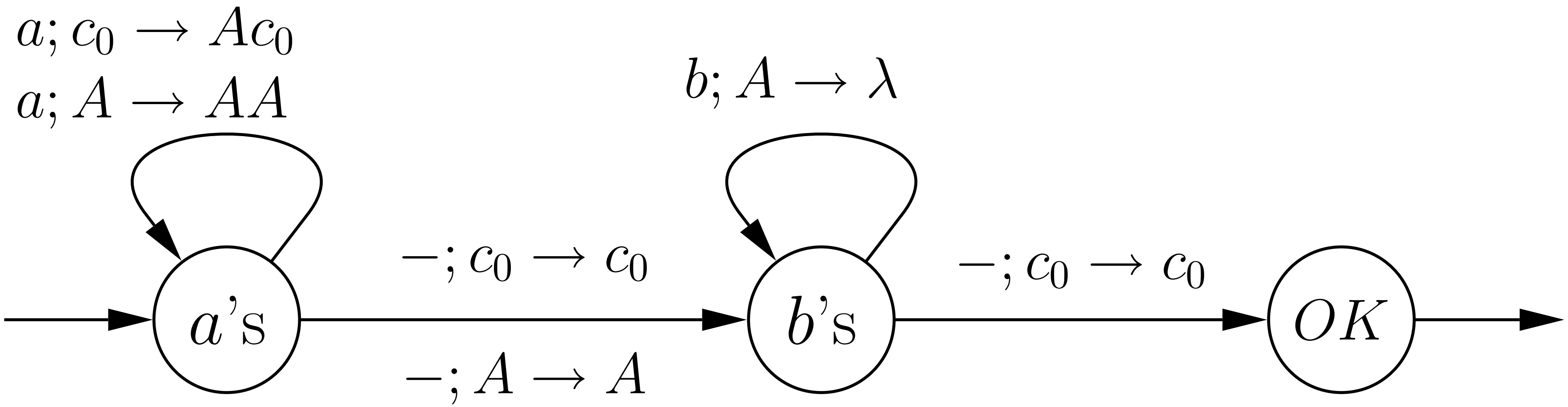
Erkannte Sprache

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

Erkannte Sprache

- K erkannt $w \in I^*$, falls $(s_0, w, c_0) \vdash^* (s'', \lambda, \gamma)$ 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^2b^2:

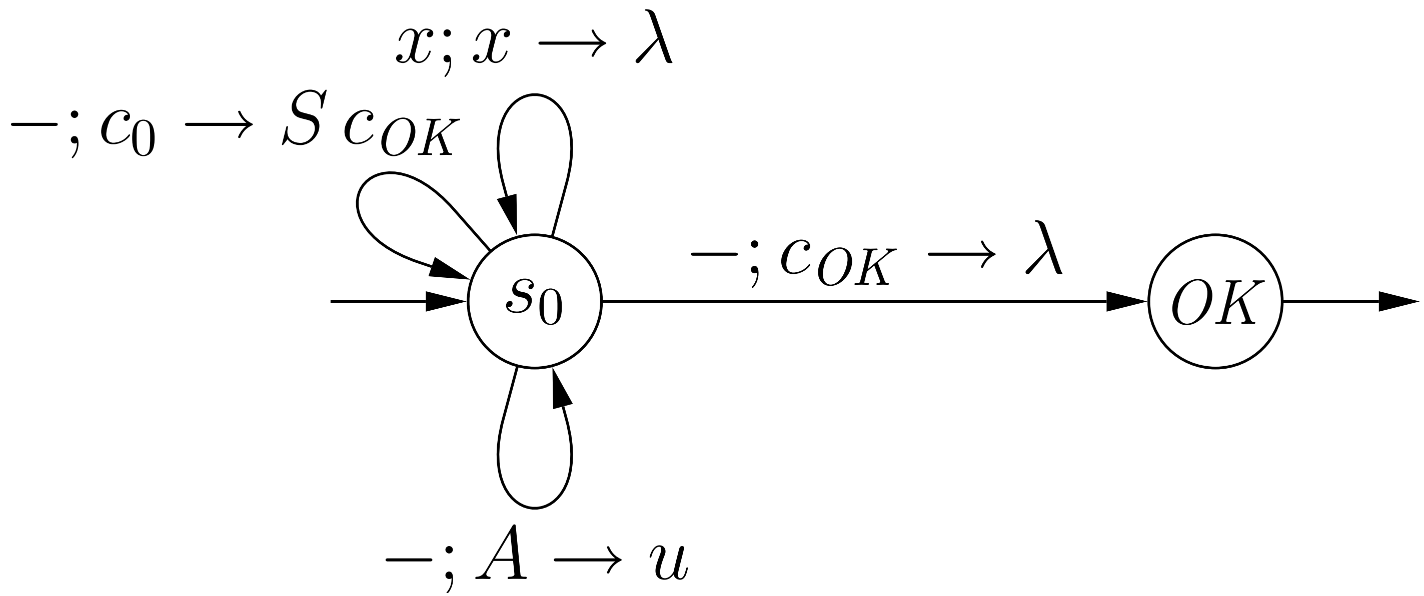
$$(a's, aabb, c_0) \longmapsto (a's, abb, Ac_0) \longmapsto (a's, bb, AAc_0) \longmapsto$$
$$(b's, bb, AAc_0) \longmapsto (b's, b, Ac_0) \longmapsto (b's, \lambda, c_0) \longmapsto (OK, \lambda, c_0)$$

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 $c_0, c_{OK} \notin N \cup T$.

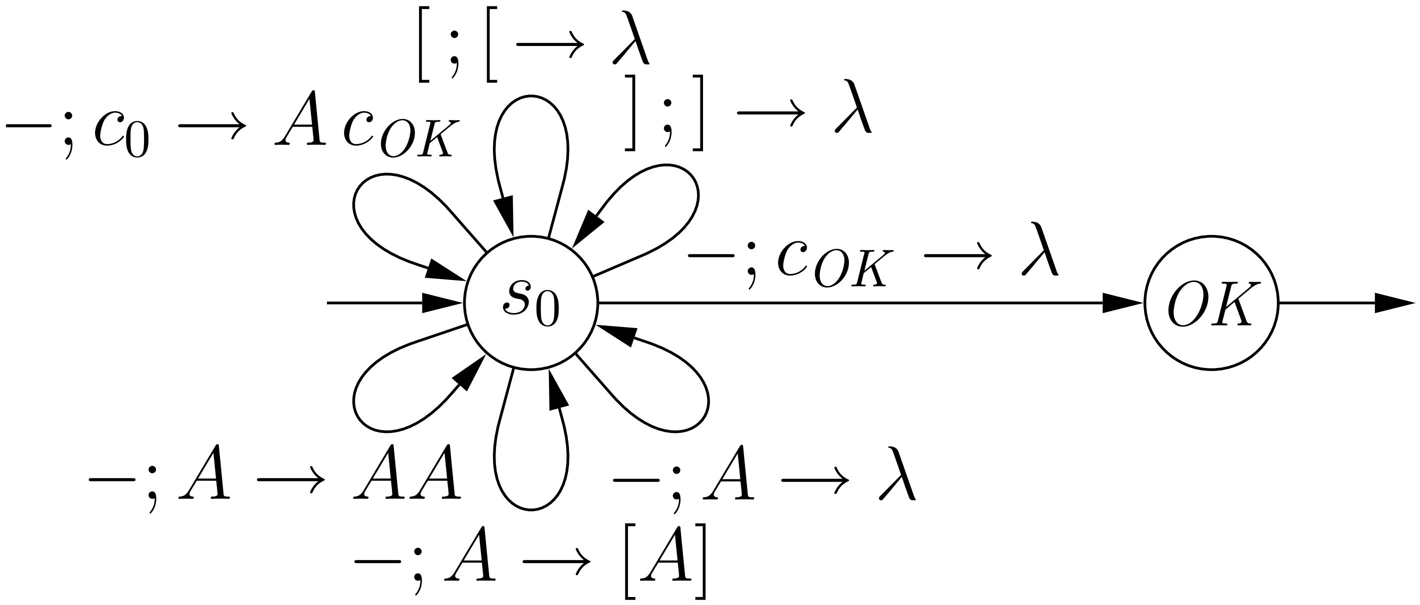


wobei $x \in T$ und $A ::= u \in P$

Beispiel: Klammergebirge

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

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

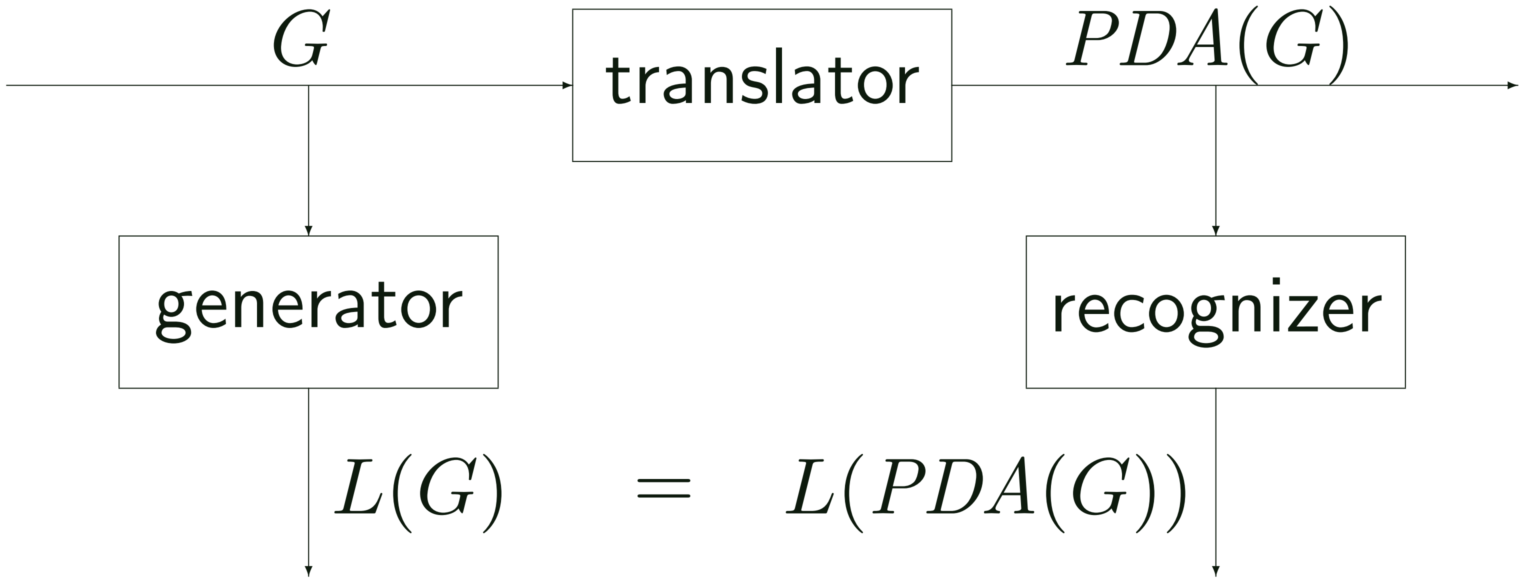


Erzeugen und Erkennen von $[]$

- $A \underset{A::=[A]}{\longrightarrow} A \underset{A::=\lambda}{\longrightarrow} []$

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

Korrektheit der Übersetzung



Linksableitungen

Sei $G = (N, T, P, S)$ eine kontextfreie Grammatik. Eine Linksableitung ist eine Ableitung $u_1 \underset{P}{\longrightarrow} u_2 \underset{P}{\longrightarrow} \cdots \underset{P}{\longrightarrow} u_n$, bei der in jedem Schritt $u_i \underset{A_i ::= v_i}{\longrightarrow} u_{i+1}$ $(1 \leq i < n)$ 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: $-\ell\!\!\longrightarrow$

Beispiel

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

$$E ::= E + E | E * E | (E) | id$$

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

1. $E \longrightarrow E{+}E \longrightarrow id{+}E \longrightarrow id{+}E{*}E$
   $\longrightarrow id + id{*}E \longrightarrow id + id * id$

2. $E \longrightarrow E{*}E \longrightarrow E{+}E{*}E \longrightarrow id{+}E{*}E$
   $\longrightarrow id + id{*}E \longrightarrow id + id * id$

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

$$uv^i wx^i y \in L.$$