Kellerautomaten

Spracherkennung (Syntaxanalyse)

- Algorithmus gesucht, der für $L \subseteq T^*$ (möglichst schnell) entscheidet, ob $w \in L$ (Lösung des Wortproblems)

Grammatik	Automat	Aufwand
rechtslinear	endlich	linear
kontextfrei	?	?

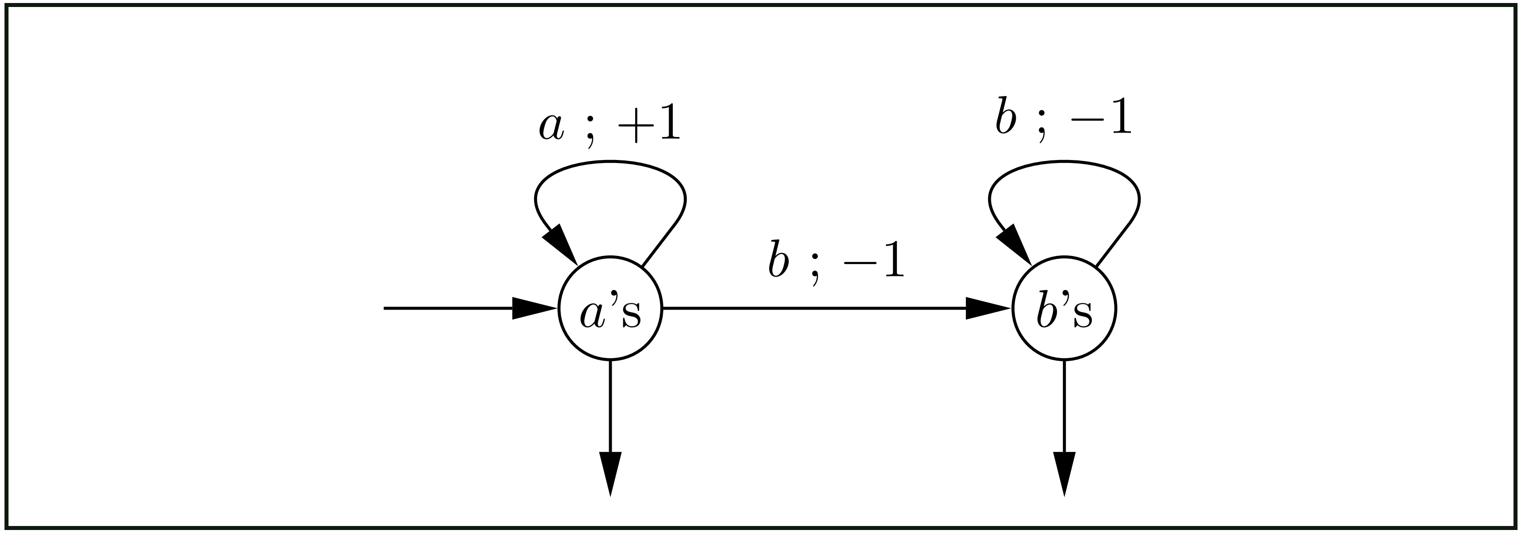
Problem: endliche Automaten können nicht unbeschränkt mitzählen und sich keine unbeschränkt langen Teilwörter merken

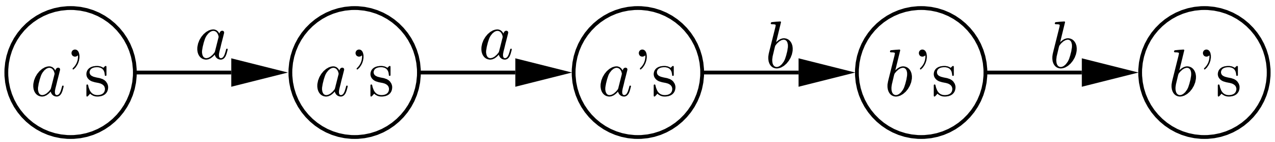
Beispiel: Klammerstrukturen

$$L = \{a^n b^n \mid n \in \mathbb{N}\}$$

- nicht regulär
- kontextfrei erzeugt durch $S ::= aSb \mid \lambda$

Idee: Endlicher Automat mit Zähler





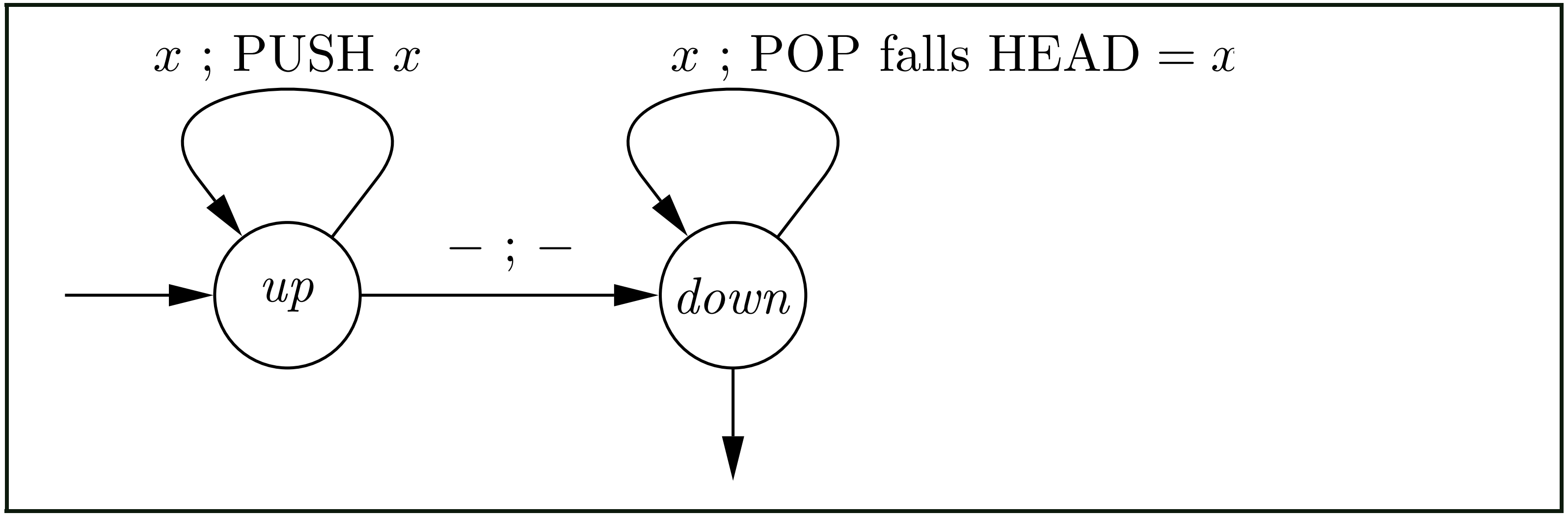
Zähler 0 1 2 1 0

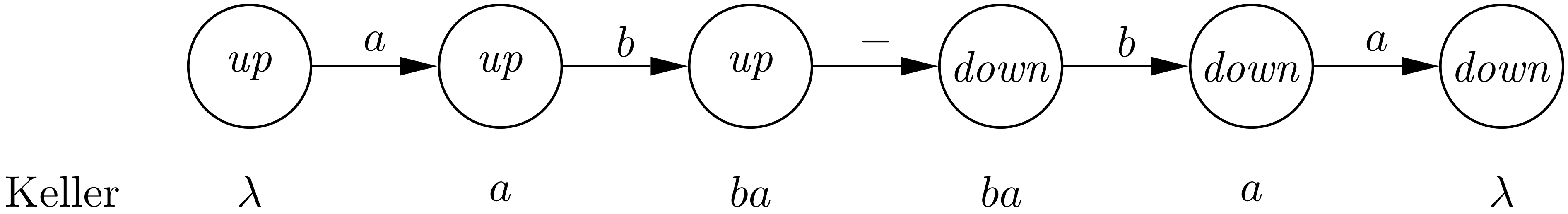
Beispiel: Palindrome

$$L_{pde} = \{w\,trans(w)\}$$

- nicht regulär
- kontextfrei erzeugt durch $S ::= xSx \mid \lambda$ f.a. $x \in T$

Endlicher Automat mit Keller/Speicher





Nichtdeterministischer Kellerautomat

Idee:

endlichen Automaten mit Zusatzspeicher in Form eines Kellers (Stapel, Stack) mit Speicheroperationen pro Übergang

Keller über X:

X^* mit den Operationen push, head, pop (d.h. $push(x,u) = xu$, $head(xu) = x$, $pop(xu) = u$ für alle $x \in X$, $u \in X^*$)

Ein nichtdeterministischer Kellerautomat ist ein System $K = (Z, I, C, d, s_0, F, c_0)$ mit

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

Graphische Darstellung

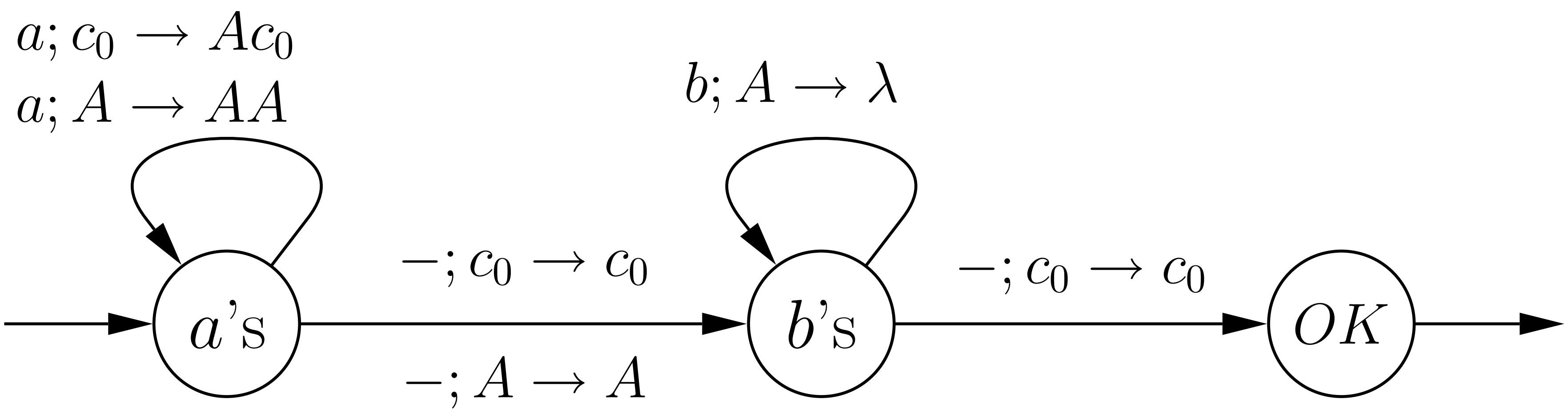
$$s \xrightarrow{x;\, c \,\longrightarrow\, \alpha} s'$$

$$\text{für } (s', \alpha) \in d(s, x, c)$$

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

$$\text{für } (s', \alpha) \in d(s, -, c)$$

Beispiel



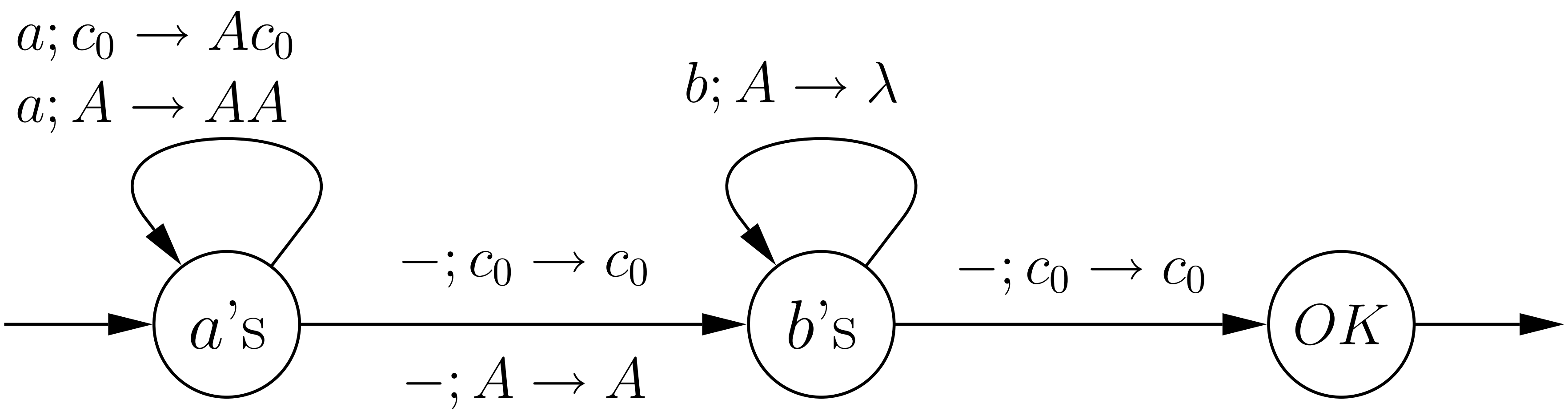
Konfiguration

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 $s \in F$

Beispiel



- Konfiguration: $(b's, bb, AAc_0)$
- Anfangskonfiguration: $(a's, aabb, c_0)$
- Endkonfiguration: (OK, λ, c_0)

$$\boxed{\begin{array}{l}\textbf{Folgekonfiguration}\\ (s, xv, c\gamma) \longmapsto (s', v, \alpha\gamma), \text{ falls } (s', \alpha) \in d(s, x, c)\\ (s, v, c\gamma) \longmapsto (s', v, \alpha\gamma), \text{ falls } (s', \alpha) \in d(s, -, c)\end{array}}$$

- **Schreibweise**:

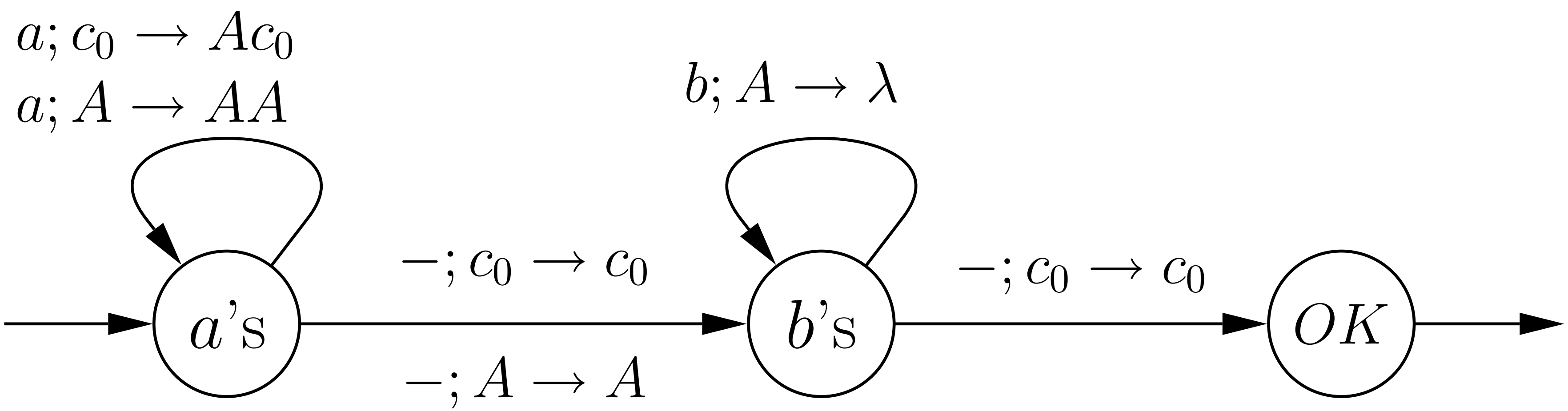
$$\boxed{con = con_0 \longmapsto con_1 \longmapsto \cdots \longmapsto con_n = con'}$$

kann durch

$$\boxed{con \overset{n}{\longmapsto} con'} \text{ oder } \boxed{con \overset{*}{\longmapsto} con'}$$

abgekürzt werden.

Beispiel



Konfigurationsfolge:

$$(a's, abb, Ac_0) \vdash\!\!\!- (a's, bb, AAc_0) \vdash\!\!\!- (b's, bb, AAc_0) \vdash\!\!\!- (b's, b, Ac_0)$$

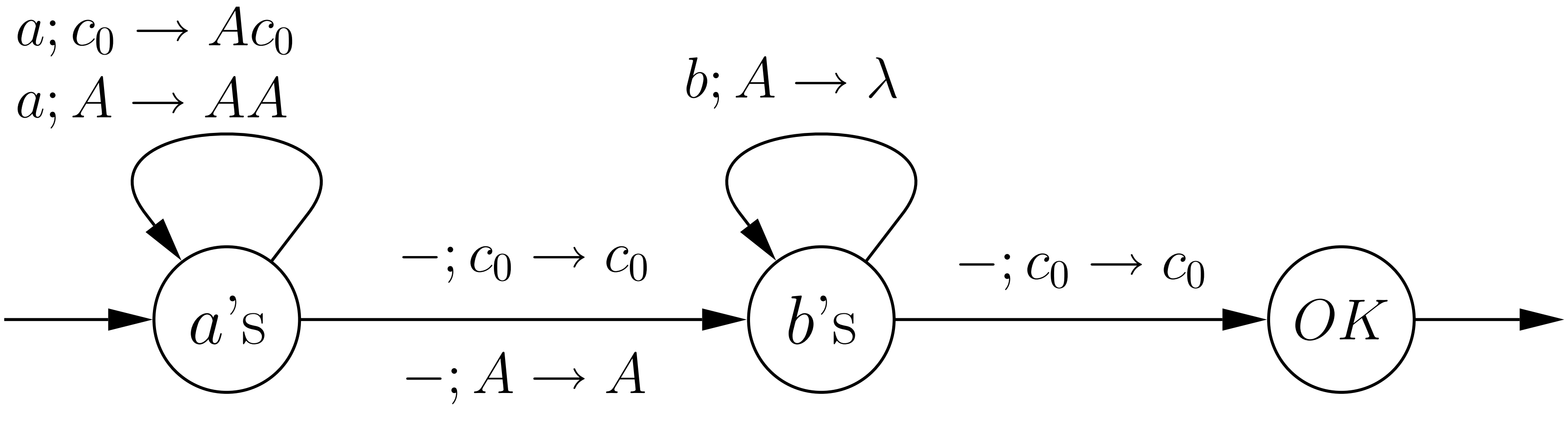
Erkannte Sprache

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

Erkannte Sprache

- K erkennt $w \in I^*$, falls $(s_0, w, c_0) \vdash^* (s'', \lambda, \gamma)$ mit $s'' \in F$.
- 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:

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

Fragestellungen

- Wie hängen endliche Automaten und kfG's zusammen?
- Sind kfG's kompositional?
- Ist für kfG's das Wortproblem schnell lösbar?
- Was können kfG's nicht?

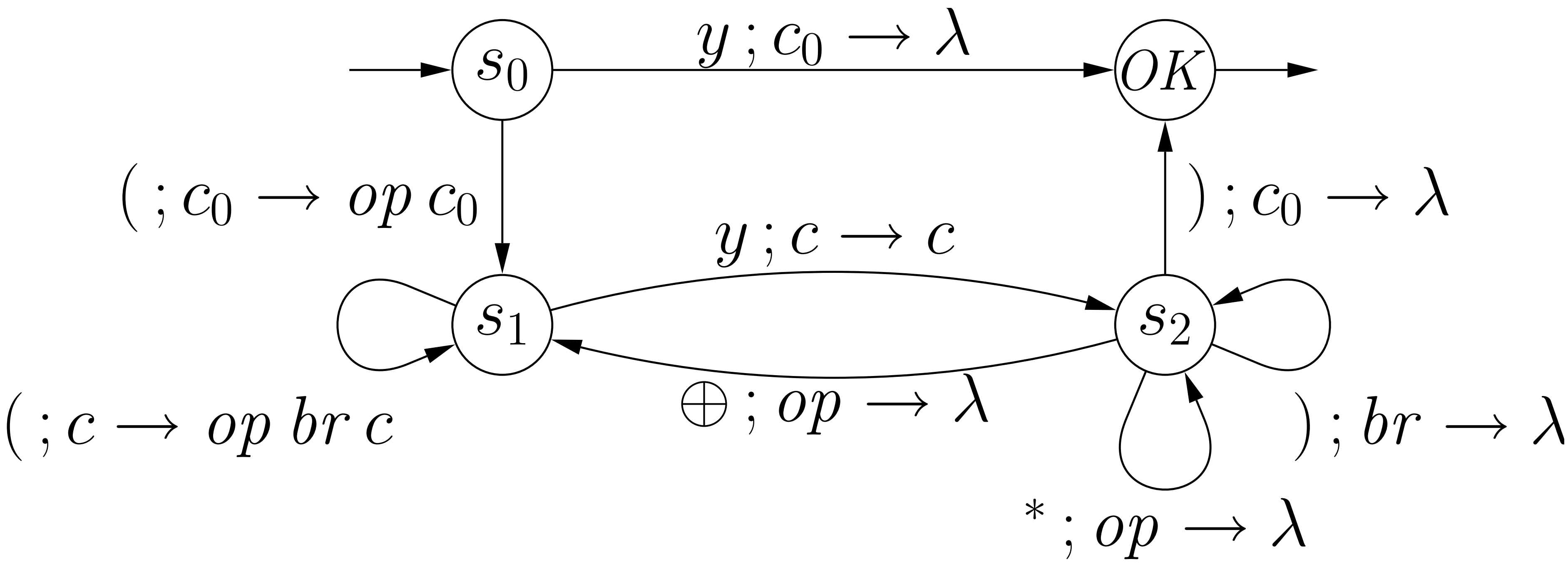
Deterministischer Kellerautomat

- 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.

Ein deterministischer Kellerautomat ist ein System
$K = (Z, I, C, d, s_0, F, c_0)$ mit

- Z: endliche Menge von Zuständen
- I: endliches Eingabealphabet mit $- \notin I$
- C: endliche Menge von Kellersymbolen
- $d\colon Z \times (I \cup \{-\}) \times C \rightsquigarrow Z \times C^*$: Zustandsüberführung mit $|d(s, x, c)| + |d(s, -, c)| \leq 1$ für alle $s \in Z$, $x \in I$, $c \in C$,
- $s_0 \in Z$: Startzustand
- $F \subseteq Z$: Endzustände
- $c_0 \in C$: initiales Kellersymbol

Beispiel: Reguläre Ausdrücke



mit $y \in I \cup \{empty, lambda\}$, $c \in \{op, br, c_0\}$ und $\bigoplus \in \{+, \circ\}$

Erkennen von $((x^*) + empty)$

$$
\begin{array}{lllr}
 & (s_0, & ((x^*) + empty), & c_0) \\
\vdash & (s_1, & (x^*) + empty), & op\, c_0) \\
\vdash & (s_1, & x^*) + empty), & op\, br\, op\, c_0) \\
\vdash & (s_2, & {}^*) + empty), & op\, br\, op\, c_0) \\
\vdash & (s_2, &) + empty), & br\, op\, c_0) \\
\vdash & (s_2, & +empty), & op\, c_0) \\
\vdash & (s_1, & empty), & c_0) \\
\vdash & (s_2, &), & c_0) \\
\vdash & (OK, & \lambda, & \lambda)
\end{array}
$$

Mächtigkeit

- $\mathcal{L}_{DKA}$: Klasse aller von deterministischen Kellerautomaten erkannten Sprachen
- $\mathcal{L}_{NKA}$: Klasse aller von nichtdeterministischen Kellerautomaten erkannten Sprachen

Es gibt keinen deterministischen Kellerautomaten, der

$$\{wtrans(w) \mid w \in \{a, b\}^*\}$$

erkennt.

Also gilt

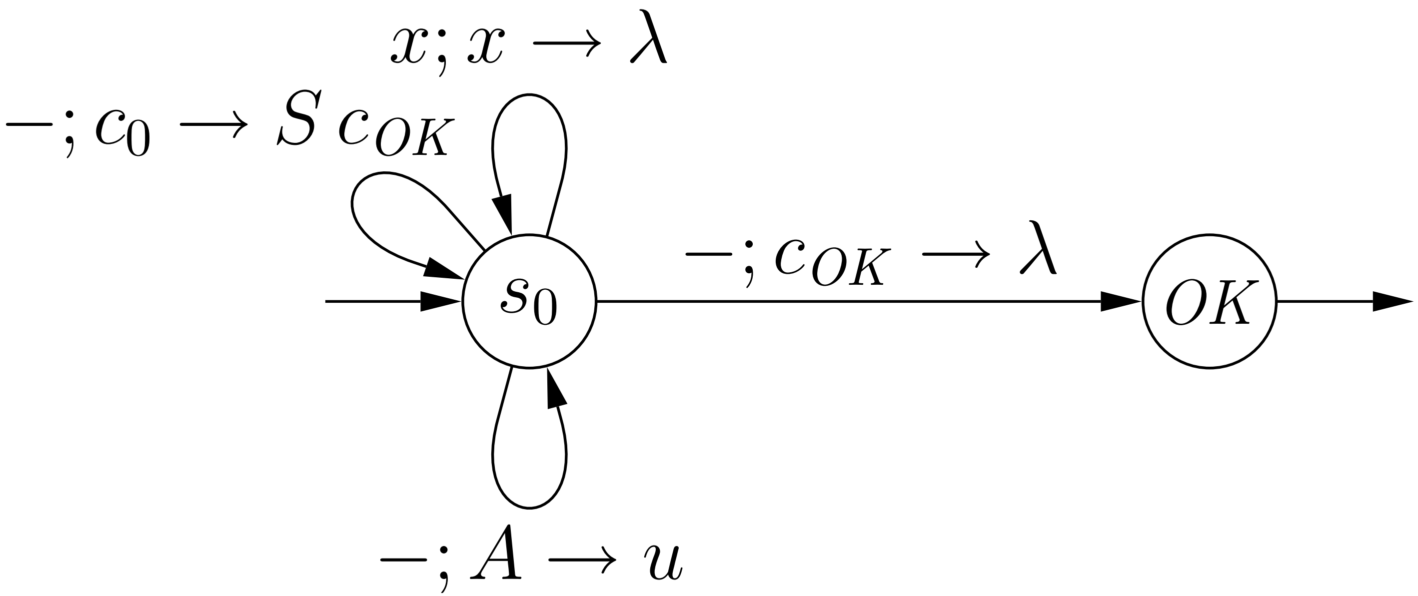
$$\mathcal{L}_{DKA} \subset \mathcal{L}_{NKA}$$

Fragestellungen

- Sind kontextfreie Sprachen kompositional? (Die Klasse der kontextfreien Sprachen ist abgeschlossen unter Vereinigung, Konkatenation und Kleene-Stern.)
- Ist für kontextfreie Sprachen das Wortproblem schnell lösbar? (Die von determinitischen Kellerautomaten erkannten Sprachen lassen sich in linearer Zeit erkennen.)
- Wie hängen Kellerautomaten und kontextfreie Sprachen zusammen?
- Was können Kellerautomaten nicht?

Übersetzung kontextfreier Grammatiken in Kellerautomaten

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

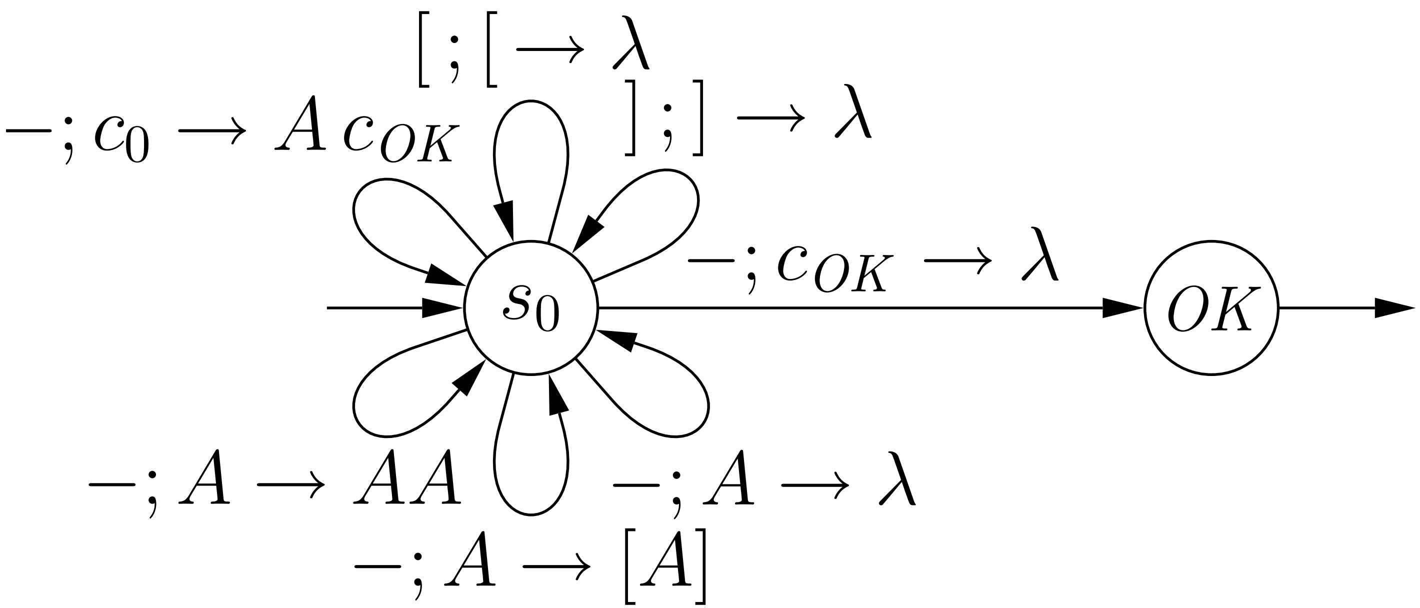


wobei $x \in T$ und $A ::= u \in P$

Beispiel: Klammergebirge

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

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



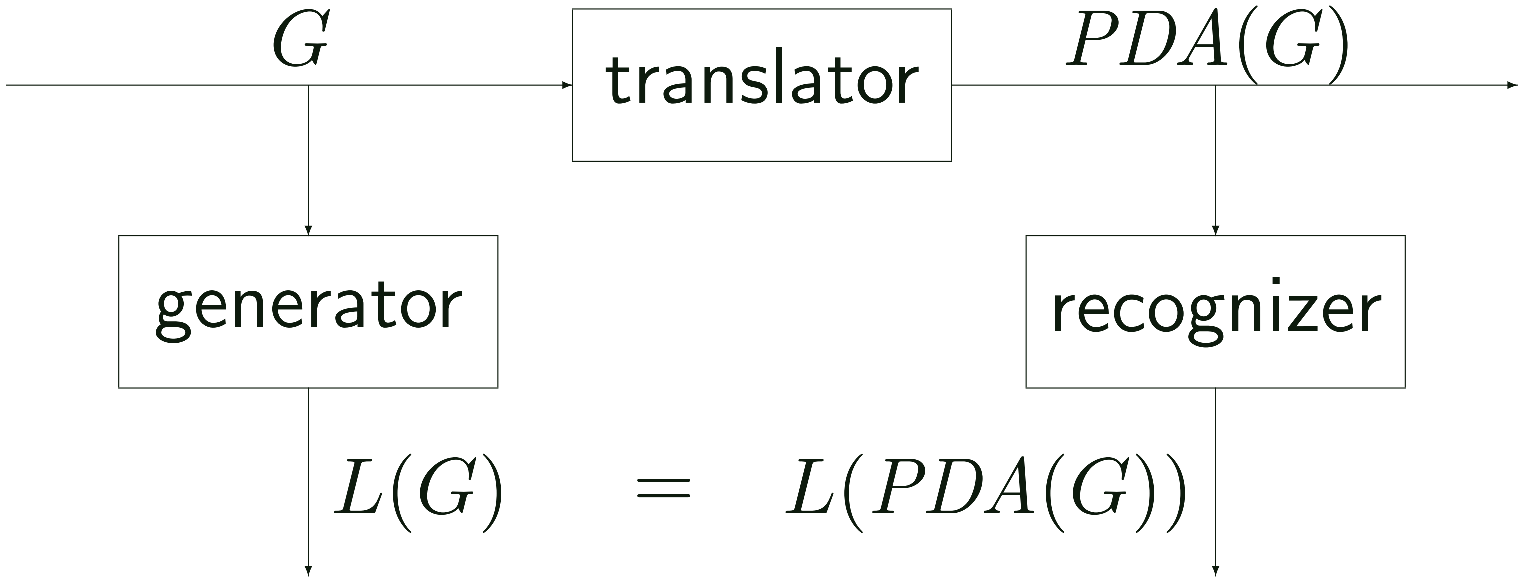
Erzeugen und Erkennen von []

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

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

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

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: $\overset{}{-\ell\!\!\longrightarrow}$

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.

Zusammenhang zwischen Kellerautomaten und kontextfreien Sprachen

Für jede kontextfreie Grammatik G gibt es einen Kellerautomaten K, so dass $L(G) = L(K)$.

Für jeden Kellerautomaten K gibt es eine kontextfreie Grammatik G, so dass $L(K) = L(G)$. (Ohne Beweis)

Folgerung

$$\boxed{\mathcal{L}_{NKA} = \mathcal{L}_{KFS}.}$$

- $\mathcal{L}_{NKA}$: Klasse aller von nichtdeterministischen Kellerautomaten erkannten Sprachen
- $\mathcal{L}_{KFS}$: Klasse der kontextfreien Sprachen