

# Formale Sprachen

K.J. Lange  
Sommersemester 2003

Ergänzungskapitel: "Das syntaktische Monoid"

# Inhaltsverzeichnis

<b>-1</b>	<b>Quellen</b>	<b>1</b>
<b>0</b>	<b>Das syntaktische Monoid</b>	<b>1</b>
0.1	Notation . . . . .	1
0.2	Automaten . . . . .	1
0.3	Monoide . . . . .	7
0.4	Beziehungen zwischen Familien formaler Sprachen und Mengen von Monoiden . . . . .	14
0.5	Endlich beschreibbare unendliche Automaten . . . . .	15
0.6	Konstruktion eines (i.A. unendlichen) Monoids aus einem Präfixautomaten . . . . .	17

## -1 Quellen

- A. Salomaa: *Formale Sprachen*. Springer, 1978.
- J. Hopcroft-Ullman, J. Ullman: *Formal Languages and Their Relation to Automata*. Addison-Wesley, 1979.
- M.A. Harrison: *An Introduction to Formal Language Theory*. Addison-Wesley, 1978.
- J. Berstel: *Transduction and context-free Languages*. Teubner.
- S. Eilenberg: *Automata, Languages and Machines*. A+B Academic Press 1976.
- H. Straubing: *Finite Automata, Formal Logic and, Circuit Complexity*. Springer Verlag, 1994.
- J. E. Pin: *Varieties of Formal Languages*. Plenum Press, 1986.

## 0 Das syntaktische Monoid

### 0.1 Notation

$$\mathbb{N} = \{1, 2, \dots\}$$

$$\mathbb{N}_0 = \{0, 1, 2, \dots\}$$

$$\mathbb{Z} = \{\dots, -1, 0, 1, \dots\}$$

$\Sigma$ : Endliches Alphabet

$\Sigma^*$ : Menge aller Wörter über  $\Sigma$  (genannt: das freie Monoid über  $\Sigma$ )

$\lambda$ : Das leere Wort Häufig  $\epsilon, I, \dots$

$|v|$ : Die Länge des Wortes  $v$   $|\lambda| := 0, |xa| := |x| + 1$

$\#_a(x)$ : Häufigkeit des Auftretens von  $a$  in  $x$

$$\#_a(x) := |\{(x_1, x_2) \in \Sigma^* \times \Sigma^* \mid x_1 a x_2 = x\}|$$

(ist Morphismus von  $\Sigma^*$  nach  $\mathbb{N}$ )

$$x \setminus L := \{y \mid xy \in L\} \quad L \setminus x := \{y \mid yx \in L\} \quad \lambda \setminus L = L = L / \lambda$$

### 0.2 Automaten

Wir wiederholen den Automatenbegriff, fordern aber nicht länger die Endlichkeit der Zustandsmenge.

#### Definition 1.

a.) Ein Automat ist ein Konstrukt  $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$  mit:

$Q$ : Die Menge der Zustände

$\Sigma$ : Das Eingabealphabet

$\delta: Q \times \Sigma \rightarrow Q$ : Die Übergangsfunktion

$q_0 \in Q$ : Der Startzustand und

$F \subseteq Q$ : Die Endzustände

b.)  $\delta$  wird in üblicher Weise fortgesetzt zu einer Abbildung zu  $\delta^* : Q \times \Sigma^* \rightarrow Q$   
 vermöge  $\delta^*(q, \lambda) := q$  und  $\delta^*(q, xa) := \delta(\delta^*(q, x), a)$

Die alternative Möglichkeit:  $\delta^*(q, ax) := \delta^*(\delta(q, a), x)$  ist dazu äquivalent:

dann gilt:  $\delta^*(q, x_1x_2) = \delta^*(\delta^*(q, x_1), x_2) \forall q, x_1, x_2$

c.) Die Akzeptierte Sprache ist  $L(\mathcal{A}) := \{x \in \Sigma^* \mid \delta^*(q_0, x) \in F\}$   
 Für  $q \in Q$  ist  $L_q(\mathcal{A}) := \{x \in \Sigma^* \mid \delta^*(q, x) \in F\}$  die Leistung von  $q$ .

Jedes  $L \subseteq \Sigma^*$  wird von einem Automaten erkannt:

**Konstruktion 2.**  $Z_L := (\Sigma^*, \Sigma, \delta, \lambda, L)$  mit  $\delta(x, a) := xa$

Grösstmöglicher Automat - kleinster interessiert mehr

**Beispiel 1.**

$L_1 := a^*b^*$  Der endliche Automat  $\mathcal{A}(L_1)$  ist eine "Abkürzung" des folgenden unendlichen Baums:

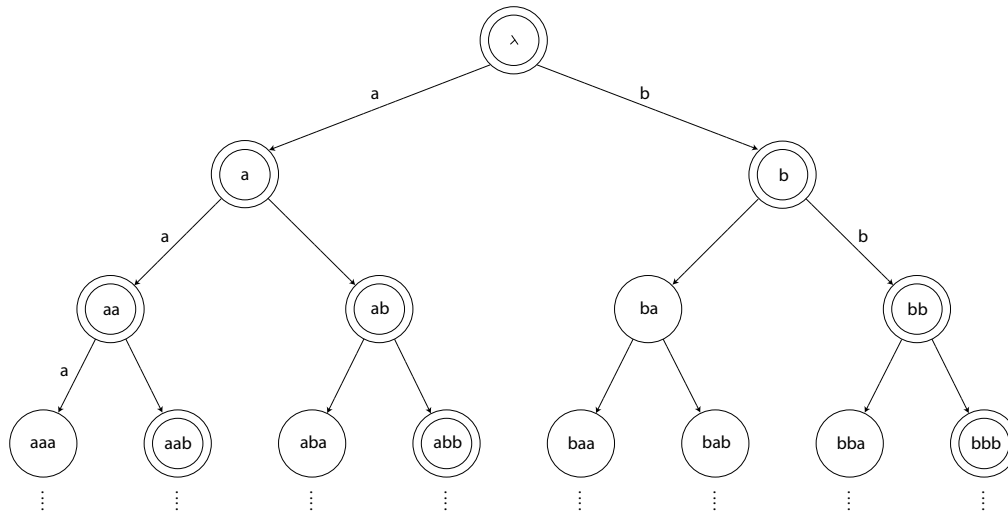


Abbildung 1:  $L_1$

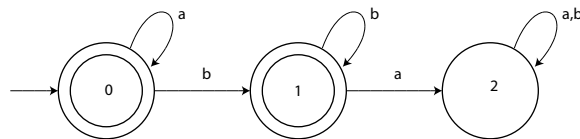


Abbildung 2:  $\approx$  Minimaler Automat  $L_1$

**Definition 3.**

Zu  $L \subseteq \Sigma^*$  und  $x, y \in \Sigma^*$  definiere  $x \stackrel{R}{\sim}_L y$  g.d.w.  $x \setminus L = y \setminus L$  gilt.  
Wenn also gilt:  $\forall z \in \Sigma^* \quad xz \in L \Leftrightarrow yz \in L$ .

**Behauptung 4.**

$\stackrel{R}{\sim}_L$  ist eine Rechtskongruenz, also eine Äquivalenzrelation, die zusätzlich erfüllt:  
 $\forall x, y, z \quad x \stackrel{R}{\sim}_L y \Rightarrow xz \stackrel{R}{\sim}_L yz$ .

Für  $x \in \Sigma^*$  sei  $[x]_L^R := \{y \mid x \stackrel{R}{\sim}_L y\}$  die Äquivalenzklasse von  $x$ .

**Beispiel 2.** Fortsetzung von  $L_1$  für

$L_1 = a^*b^*$ :  $\stackrel{R}{\sim}_{L_1}$  hat die drei Klassen

$$a^*, \text{ für } x \in a^* \text{ gilt } x \setminus L_1 = L_1,$$

$$a^*b^+, \text{ für } x \in a^*b^+ \text{ gilt } x \setminus L_1 = b^*,$$

$$a^*b^+a\{a, b\}^* \text{ für } x \in a^*b^+a\{a, b\}^* \text{ gilt } x \setminus L_1 = \emptyset$$

Es gilt  $L_1 = [a]_{L_1}^R \cup [b]_{L_1}^R$

Bemerkung:  $L_1^R$  ist keine Linkskongruenz:  $\lambda \stackrel{R}{\sim}_{L_1} a$  aber  $b \not\stackrel{R}{\sim}_{L_1} ba$

**Beispiel 3.**

$$L_2 = \{w \in \{a, b\}^* \mid \#_a(w) = \#_b(w)\}$$

$\stackrel{R}{\sim}_{L_2}$  hat folgende Klassenstruktur:

$$x \stackrel{R}{\sim}_{L_2} y \text{ g.d.w. } \#_a(x) - \#_b(x) = \#_a(y) - \#_b(y)$$

Es gibt also folgende Klassen für jedes  $i \in \mathbb{N}_0$ :

$$[a^i]_{L_2}^R = \{x \mid \#_a(x) - \#_b(x) = i\}.$$

Für  $x \in [a^i]_{L_2}^R$  ist dann  $x \setminus L_2 = \{y \in \{a, b\}^* \mid \#_a(y) - \#_b(y) = -i\}$

$$[b^i]_{L_2}^R = \{x \mid \#_a(x) - \#_b(x) = -i\}$$

Für  $x \in [b^i]_{L_2}^R$  gilt  $x \setminus L_2 = [a^i]_{L_2}^R$ .

Es gilt  $L_2 = [\lambda]_{L_2}^R$

$\stackrel{R}{\sim}_{L_2}$  ist auch eine Linkskongruenz.

**Beispiel 4.**

$$L_3 := \{a^n b^n \mid n \geq 0\} = L_1 \cap L_2$$

$\stackrel{R}{\sim}_{L_3}$  hat folgende Klassentstruktur:

Für  $i \geq 0$  gilt:  $[a^i]_{L_3}^R = \{a^i\}$  und  $a^i \setminus L_3 = \{a^j b^{i+j} \mid j \geq 0\}$ ,

$[a^{i+1}b]_{L_3}^R = \{a^{i+1}b^j \mid j \geq 1\}$  und für  $x \in [a^{i+1}b]_{L_3}^R$  gilt  $x \setminus L_3 = \{b^i\}$

und der 'Rest':

$$[b]_{L_3}^R = [a^{i+j}b^j a]_{L_3}^R = [a^i b^{i+1}]_{L_3}^R$$

Für  $x \in [b]_{L_3}^R$  gilt  $x \setminus L_3 = \emptyset$ .

$\stackrel{R}{\sim}_{L_3}$  ist also im Kreuzprodukt von  $\stackrel{R}{\sim}_{L_1}$  und  $\stackrel{R}{\sim}_{L_2}$  enthalten

$\Sigma^*$							
$a^*$	$\emptyset$	$\emptyset$	$\emptyset$	$\lambda$	$a$	$a^2$	$a^3$
$a^*b^*$	$\emptyset$	$\emptyset$	$\emptyset$	$ab$ $aabb$	$aab$ $aaabb$	$aaab$	
Rest		$bb$	$b$	$ba$	$aba$ $baa$	$baaa$ $\dots$	
	-3	-2	-1	0	+1	+2	+3

Abbildung 3: Rechtsklassen von  $L_3$

Es gilt  $L_3 = [\lambda]_{L_3}^R \cup [ab]_{L_3}^R$

**Feststellung: 5.**

Für eine Sprache  $L \subseteq \Sigma^*$  und ein Wort  $x \in \Sigma^*$  sind folgende Aussagen äquivalent:

- a.)  $[x]_L^R \subseteq L$
- b.)  $x \in L$
- c.)  $[x]_L^R \cap L \neq \emptyset$

**Beweis 6.**

"a  $\Rightarrow$  b  $\Rightarrow$  c": *trivial*

"c  $\Rightarrow$  a": Sei  $y \in [x]_L^R$ : zu Zeigen  $y \in L$

Nach Voraussetzung existiert ein  $y' \in [x]_L^R \cap L$

Wegen  $y' \in L$  gilt:  $\lambda \in y\lambda L$

Wegen  $y \sim_L^R y'$  gilt:  $y \setminus L = y' \setminus L$

Also  $\lambda \in y \setminus L$  mithin  $y \in L \square$

Die syntaktische Rechtskongruenz einer Sprache  $L$  beschreibt einen Automaten für  $L$  wie folgt.

**Konstruktion 7.** Definiere  $\mathcal{A}_L := (Q, \Sigma, \delta, q_0, F)$

Für  $L \subseteq \Sigma^*$

vermöge

$$Q := \{[x]_L^R \mid x \in \Sigma^*\} \approx \{x \setminus L \mid x \in \Sigma^*\}$$

$$\lambda([x]_L^R, a) := [xa]_L^R \approx \lambda(x \setminus L, a) := xa \setminus L$$

$$q_0 := [\lambda]_L^R \approx q_0 := \lambda \setminus L (= i)$$

$$F := \{[x]_L^R \mid x \in L\} \approx F := \{x \setminus L \mid x \in L\} = \{x \setminus L \mid \lambda \in x \setminus L\}$$

”Claims:” (ohne Beweis)

- $\delta$  ist wohldefiniert
- $L(\mathcal{A}_L) = L$
- $\langle L \text{ überstrapaziert} \rangle$

$\mathcal{A}_L$  ist der 'kleinste' Automat und  $Z_L$  der grösste, der  $L$  akzeptiert:

**Satz 8.**

Wird  $L \subseteq \Sigma^*$  akzeptiert von einem Automaten  $\mathcal{B} = (Q', \Sigma, \delta', q'_0, F')$  der nur erreichbare Zustände hat, so gilt:

- a.) Es gibt eine surjektive Abbildung  $g : Q' \rightarrow \{[x]_L^R \mid x \in \Sigma^*\}$ , die erfüllt:  
 $\forall_{q' \in Q'} L_{q'}(\mathcal{B}) = L_{g(q')}(\mathcal{A}_L)$
- b.) Es gibt eine surjektive Abbildung  $f : \Sigma^* \rightarrow Q'$ , die erfüllt:  
 $\forall_{x \in \Sigma^*} L_x(\mathcal{Z}_L) = L_{f(x)}(\mathcal{B})$

$\mathcal{A}_L$  hat also die kleinstmögliche Zustandsmenge und  $\mathcal{Z}_L$  die grösstmögliche

**Lemma 9.**

Für (jeden Automaten)  $\mathcal{B} = (Q', \Sigma, \delta', q_0, F')$  und für jedes  $x \in \Sigma^*$  gilt:  
 $L_{\delta'^*(q_0, x)}(\mathcal{B}) = x \setminus L(\mathcal{B})$

**Beweis 9.**

$$\begin{aligned} y \in L_{\delta'^*(q_0, x)}(\mathcal{B}) &\Leftrightarrow \\ \delta'^*(q_0, xy) \in F &\Leftrightarrow \\ xy \in L &\Leftrightarrow y \in x \setminus L. \end{aligned}$$

**Beweis 8.**

- a.) Ist  $q' \in Q'$  erreichbar durch  $x \in \Sigma^*$ , also  $q' = \delta'^*(q'_0, x)$ , so definiere  
 $g(q') := [x]_L^R$ 
  - $g$  ist wohldefiniert: Ist  $\delta(q'_0, x) = \delta^*(q'_0, y)$ , so gilt nach Lemma 9:  
 $x \setminus L = y \setminus L$ . Also  $[x]_L^R = [y]_L^R$ .
  - $g$  ist surjektiv. Ist  $x \in \Sigma^*$  so gilt:  $[x]_L^R = g(\delta'^*(q'_0, x))$ .
  - $L_{q'}(\mathcal{B}) = L_{g(q')}(\mathcal{A}_L)$  folgt aus Lemma 9:  
 $L_{q'}(\mathcal{B}) = L_{\delta'^*(q'_0, x)}(\mathcal{B}) = x \setminus L = L_{[x]_L^R}(\mathcal{A}_L)$
- b.) Für  $x \in \Sigma^*$  setze  $f(x) := \delta'^*(q'_0, x)$ .  
Gemäss Voraussetzung über  $\mathcal{B}$  ist  $f$  surjektiv. Aus Lemma 9 folgt:  
 $L_x(\mathcal{Z}_L) = x \setminus L = L_{\delta'^*(q'_0, x)}(\mathcal{B})$  wobei  $q'_0 = f(\lambda)$  zu beachten ist.  $\square$

**Korollar 10.**

$L$  ist genau dann regulär, wenn  $\overset{R}{\sim}_L$  endlichen Index hat, also endlich viele Klassen hat.

**Beweis 10.**

Ist  $L$  von einem endlichen Automaten  $\mathcal{B}$  akzeptiert, so ist nach Satz 8 a) der Index von  $\overset{R}{\sim}_L$  beschränkt durch die Anzahl der Zustände von  $\mathcal{B}$ . Ist umgekehrt  $\mathcal{B}$  endlich, so liefert Konstruktion 7 einen endlichen Automaten, der  $L$  akzeptiert.  $\square$

Die nicht vervollständigten minimalen Automaten für  $L_2$  und  $L_3$  haben folgende Gestalt:

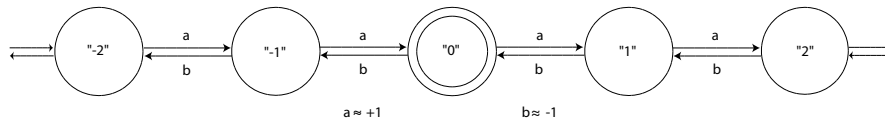


Abbildung 4: Minimaler Automat für  $L_2$

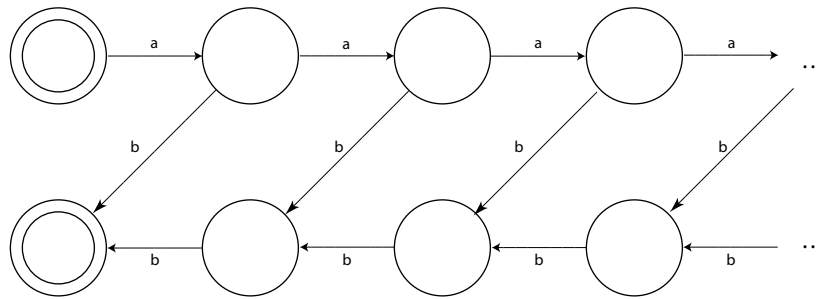


Abbildung 5: Minimaler Automat für  $L_3$

### 0.3 Monoide

Ein Monoid  $\mathcal{M} = (M, \cdot, e)$  besteht aus einer Trägermenge  $M$  (meist identifiziert mit  $\mathcal{M}$ ), einer assoziativen Verknüpfung  $\cdot$  und einem Neutralen Element  $e$ .

#### Beispiel 5.

- $(\mathbb{N}_0, +, 0)$  : die Natürlichen Zahlen (mit 0) mit Addition
- $(\mathbb{N}, \cdot, 1)$  : die Natürlichen Zahlen (mit 0) mit Multiplikation
- $(\Sigma^*, \cdot, \lambda)$  : das freie Monoid über  $\Sigma^*$

#### Definition 11.

Eine Sprache  $L \subseteq \Sigma^*$  wird von einem Monoid  $M = (M, \cdot, e)$  erkannt, wenn es einen Homomorphismus  $\varphi : \Sigma^* \rightarrow M$  und eine ("akzeptierende") Teilmenge  $E \subseteq M$  gibt mit  $L = \varphi^{-1}(E)$ .

#### Konstruktion 12.

Aus einem Automaten  $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ , der eine Sprache  $L = L(\mathcal{A})$  akzeptiert, lässt sich ein erkennendes Monoid  $(\mathcal{M}, \varphi, E)$  konstruieren und umgekehrt:

" $\Rightarrow$ " Sei  $M := \{f : Q \rightarrow Q \mid f \text{ ist Abbildung}\}$ .  $M$  ist Monoid mit der Verknüpfung  $f \cdot g(q) := g(f(q))$  und der Identität  $e(q) := q$  als neutral Element.  
 Setze weiter  $E := \{f \in M \mid f(q_0) \in F\}$ .  
 Definiere nun  $\varphi : \Sigma^* \rightarrow M$  vermöge  $\varphi(x) := \delta^*(\cdot, x)$  also  $\varphi(x)(q) := \delta^*(q, x)$ .  
 $\varphi$  ist Homomorphismus:  $\varphi(v_1 v_2)(q) = \delta^*(q, v_1 v_2) = \delta^*(\delta^*(q, v_1), v_2) = \varphi(v_2)(\varphi(v_1)(q)) = (\varphi(v_1) \cdot \varphi(v_2))(q)$ .  
 Dann ist  $L(\mathcal{A}) = \varphi^{-1}(E)$  wegen  $x \in L(\mathcal{A}) \Leftrightarrow \delta^*(q_0, x) \in F \Leftrightarrow \varphi(x) \in E \Leftrightarrow x \in \varphi^{-1}(E)$

#### Bemerkung

Das so konstruierte Monoid heisst das Transformationsmonoid des Automaten  $\mathcal{A}$

" $\Leftarrow$ " Ist  $L = \varphi^{-1}(F)$ ,  $F \subseteq M$ , so konstruiere folgende Automaten:

$\mathcal{A} := (Q, \Sigma, \delta, q_0, F)$  vermöge

$Q := M$ ,  $\delta(m, a) := m \cdot \varphi(a)$ ,  $q_0 := \varphi(\lambda)$  und  $F = E$ .

Induktiv über  $|x|$  beweist man dann  $\forall m \in M \forall x \in \Sigma^* \delta^*(m, x) = m \cdot \varphi(x)$ .

Dann ist  $\varphi^{-1}(E) = L(\mathcal{A})$  wegen:

$x \in L(\mathcal{A}) \Leftrightarrow \delta^*(q_0, x) \in F \Leftrightarrow \varphi(x) = \varphi(\lambda) \cdot \varphi(x) \in E \Leftrightarrow x \in \varphi^{-1}(E) \square$

#### Bemerkung

Eine Sprache ist genau dann regulär, wenn sie von einem endlichen Monoid erkennbar ist.

Jedes  $L \subseteq \Sigma^*$  wird von dem freien Monoid  $\Sigma^*$  mit dem Homomorphismus  $\varphi(a) = a$  (Identität) und der akzeptierenden Mengen  $E := L$  erkannt. Dieses ist das feinste Monoid, das  $L$  erkennt. Im weiteren suchen wir nach dem größtmöglichen Monoid zur Erkennung von  $L$ .

**Definition 13.**

Für  $L \subseteq \Sigma^*$  wird die syntaktische Kongruenz zu  $L$  erklärt, vermöge  $x \sim_L y$  g.d.w.

$\forall z \in \Sigma^* z x \setminus L = z y \setminus L$ , also g.d.w.

$\forall z, z' \in \Sigma^* z x z' \in L \Leftrightarrow z y z' \in L$ .

**Behauptung 14.**

$\sim_L$  ist eine Kongruenzrelation, also eine Äquivalenzrelation, die zusätzlich erfüllt:

$\forall x, y, x', y' x \sim_L x' \wedge y \sim_L y' \Rightarrow xy \sim_L x'y'$ .

Für  $x \in \Sigma^*$  sei  $[x]_L := \{y \in \Sigma^* | x \sim_L y\}$  die Kongruenzklasse von  $x$ .

Es gilt immer  $x \sim_L y \Rightarrow x \sim_L^R y$  und daher  $[x]_L \subseteq [x]_L^R$ .

**Fortsetzung der Beispiele:**

**Beispiel 6.**  $L_1 = a^*b^* \sim_{L_1}$  besitzt folgende Klassen:

- $[\lambda]_{L_1} = \{\lambda\}$
- $[a]_{L_1} = a^+$
- $[b]_{L_1} = b^+$
- $[ab]_{L_1} = a^+b^+$
- $[ba]_{L_1} = \{a, b\}^*ba\{ab\}^*$

Es gilt  $L = [\lambda]_{L_1} \cup [a]_{L_1} \cup [b]_{L_1} \cup [ab]_{L_1}$

sowie  $[a]_{L_1}^R = [\lambda]_{L_1} \cup [a]_{L_1}$

und  $[b]_{L_1}^R = [b]_{L_1} \cup [ab]_{L_1}$

**Beispiel 7.**  $L_2 = \{x \in \{a, b\}^* | \#_a(x) = \#_b(x)\}$

Da  $L_2$  kommutativ ist, also  $(xy \in L \Leftrightarrow yx \in L)$  stimmt  $\sim_{L_2}^R$  mit  $\sim_{L_2}$  überein. Es ergibt sich also die gleiche Klassenstruktur.

**Beispiel 8.**  $L_3 = \{a^n b^n | n \geq 0\} = L_1 \cap L_2 \sim_{L_3}$  besitzt folgende Klassen:

- $[\lambda]_{L_3} = \{\lambda\}$  und für  $i \geq 1$ :
- $[a^i]_{L_3} = \{a^i\}$
- $[b^i]_{L_3} = \{b^i\}$
- $[a^{i+1}b]_{L_3} = \{a^{i+j}b^j | j \geq 1\}$
- $[ab^{i+1}]_{L_3} = \{a^j b^{i+j} | j \geq 1\}$  sowie
- $[ba] = \{a, b\}^*ba\{a, b\}^*$

$\sim_{L_3}$  lässt sich in dem Produkt ( $\approx$  Schnitt) von  $\sim_{L_1}$  und  $\sim_{L_2}$  einbetten.

	$\Sigma^*$				
$\lambda$	$\emptyset$	$\lambda$	$\emptyset$		
$a^+$	$\emptyset$	$\emptyset$	$a$	$a^2$	
$b^+$	$b^2$	$b$	$\emptyset$	$\emptyset$	$\emptyset$
$a^+b^+$	...	abb	ab	aab	...
$\Sigma^*ba\Sigma^*$	bba	ba	aba	baa	
	-2	-1	0	+1	+2

Abbildung 6: Syntaktische Klassen von  $L_3$   
Die untere Zeile bildet eine Klasse im Monoid  $L_3$

Es gilt also:

- $[a^i]_{L_3}^R = [a^i]_{L_3}$  für  $i \geq 0$  und
- $[a^{i+1}b] = [a^{i+1}b]_{L_3}$  sowie
- $[ba]_{L_3}^R = [ba]_{L_3} \cup \bigcup_{i \geq 1} [b^i]_{L_3} \cup \bigcup_{i \geq 1} [ab^{i+1}]_{L_3}$

**Feststellung 14.** Entsprechendes gilt auch für  $\sim_L$ .

**Feststellung 15.**

Folgende Aussagen sind äquivalent:

- a.)  $[x]_L \subseteq L$
- b.)  $x \in L$
- c.)  $[x]_L \cap L \neq \emptyset$

**Behauptung/ Definition 16.**

Die Menge der Kongruenzklassen  $M_L := \{[x]_L \mid x \in \Sigma^*\}$  bildet mit der Multiplikation  $[x]_L \cdot [y]_L := [xy]_L$  und dem neutralen Element  $[\lambda]_L$  ein Monoid, das sogenannte syntaktische Monoid von  $L$ :  $\mathcal{M}_L$

**Beweis 16.**

*Wohldefiniertheit:* Ist  $x \sim_L x'$  und  $y \sim_L y'$  so folgt  $xy \sim_L x'y'$  aus der Behauptung 16.

*Die Assoziativität folgt aus der Assoziativität der Konkatenation in  $\Sigma^*$ , genauso die Neutralität von  $[\lambda]_L$ .  $\square$*

**Fortsetzung der Beispiele****Beispiel 9.**

$\mathcal{M}_{L_1}$  hat folgende Multiplikationstafel:

$\lambda$	$a$	$b$	$ab$	$\emptyset$
$a$	$a$	$ab$	$ab$	$\emptyset$
$b$	$\emptyset$	$b$	$\emptyset$	$\emptyset$
$ab$	$\emptyset$	$ab$	$\emptyset$	$\emptyset$
$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

**Beispiel 10.**

$\mathcal{M}_{L_2}$  ist isomorph zur Gruppe der ganzen Zahlen ( $a \sim +1, b \sim -1, \lambda \sim 0$ )

**Beispiel 11.**

$\mathcal{M}_{L_3}$  hat folgende Multiplikationstafel:

$\lambda$	$a^i$	$b^i$	$a^{i+1}b$	$ab^{i+1}$	$\emptyset$
$a^i$	$a^{i+i'}$	$\begin{cases} a^{i'-i+1}b : i' \geq i \\ ab^{i-i'+1} : i' \leq i \end{cases}$	$a^{i+i'+1}b$	$\begin{cases} a^{i'-i+1}b : i' \geq i \\ ab^{i-i'+1} : i' \leq i \end{cases}$	$\emptyset$
$b^i$	$\emptyset$	$b^{i+i'}$	$\emptyset$	$\emptyset$	$\emptyset$
$a^{i+1}b$	$\emptyset$	$\begin{cases} a^{i'-i+1}b : i' \geq i \\ ab^{i-i'+1} : i' \leq i \end{cases}$	$\emptyset$	$\emptyset$	$\emptyset$
$ab^{i+1}$	$\emptyset$	$ab^{i'+i+1}$	$\emptyset$	$\emptyset$	$\emptyset$
$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

**Bemerkung 17.**

Die Abbildung  $\eta_L : \begin{cases} \Sigma^* \rightarrow \mathcal{M}_L \\ x \rightarrow [x]_L \end{cases}$  heisst der *syntaktische Homomorphismus* von  $L$ . Mit  $E_L = \eta_L(L) = \{[x]_L | x \in L\}$  gilt dann  $L = \eta_L^{-1}(E_L)$ ;  $L$  wird also von seinem syntaktischen Monoid erkannt. Ohne Beweis.

**Satz 18.**

$L \subseteq \Sigma^*$  ist genau dann von einem Monid  $\mathcal{M}$  erkennbar, wenn es ein Untermonoid  $\mathcal{M}' \subseteq \mathcal{M}$  und einen surjektiven Homomorphismus  $\psi : \mathcal{M}' \rightarrow \mathcal{M}_L$  gibt, der  $\eta_L$  "zerlegt", dass also ein Homomorphismus  $\varphi : \Sigma^* \rightarrow \mathcal{M}'$  existiert mit  $\eta_L = \varphi \circ \psi$

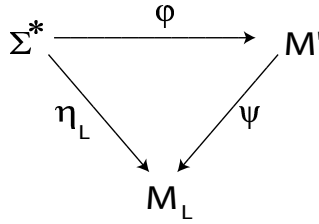


Abbildung 7:

**Beweis 18.**

"⇒"

Gilt  $L = \varphi^{-1}(E)$  für einen Homomorphismus  $\varphi : \Sigma^* \rightarrow \mathcal{M}$  und ein  $E \subseteq \mathcal{M}$ , so setzt  $\mathcal{M}' := \varphi(\Sigma^*)$ .

Dann ist  $\varphi$  surjektiver Morphismus  $\varphi : \Sigma^* \rightarrow \mathcal{M}'$ .

Ist  $m \in \mathcal{M}'$ , also  $m = \varphi(x)$  für ein  $x \in \Sigma^*$ , so definiere  $\psi(m) := \eta_L(x)$

**Behauptung:**

$\psi$  ist wohldefiniert:

Ist  $m = \varphi(x) = \varphi(y)$  so gilt für alle  $z, z' \in \Sigma^*$   $\varphi(zxz') = \varphi(zyz')$

wg  $\varphi$  homomorph

Also  $zxx' \in L \Leftrightarrow \varphi(zxz') \in E \Leftrightarrow \varphi(zyz') \in E \Leftrightarrow zyz' \in L$

Also  $\eta_L(x) = \eta_L(y)$ .

**Behauptung:**

$\psi$  ist Homomorphismus:

Ist  $m = \varphi(x)$  und  $m' = \varphi(y)$  so gilt:

$\psi(m \cdot m') = \psi(\varphi(x) \cdot \varphi(y)) = \psi(\varphi(xy)) = \eta_L(xy) = \eta_L(x) \cdot \eta_L(y) = \psi(\varphi(x)) \cdot$

$\psi(\varphi(y)) = \psi(m) \cdot \psi(m')$

**Behauptung:**

$\psi$  ist surjektiv:

Für  $[x]_L \in (M_L)$  gilt  $\psi(\varphi(x)) = [x]_L$ .

**Behauptung:**

$\eta = \varphi \circ \psi$ :

$\eta_L(x) = \psi(\varphi(x))$  nach Definition von  $\psi$ .

"⇐"

Ist  $\eta_L = \varphi \circ \psi$  für ein  $\psi : \mathcal{M}' \rightarrow \mathcal{M}_L$  und ein  $M' \subseteq L$ , so setze  $E := \psi^{-1}(E_L)$   $[= \varphi(L)!]$  dann gilt  $\varphi^{-1}(E) = \varphi^{-1}(\psi^{-1}(E_L)) = \eta_L^{-1}(E_L) = L \square$

**Bemerkung 19.**

- a.) Motiviert durch Satz 18 sagt man ein Monoid  $\mathcal{N}$  teile ein Monoid  $\mathcal{M}$  (i.Z.  $N \prec M$ ), wenn es ein  $M' \subseteq M$  und einen surjektiven Homomorphismus  $\psi : M' \rightarrow N$  gibt.

$\prec$  als Abschlusseigenschaft

Wie in Satz 8:  $L = \varphi^{-1}(E)$  und sind "alle Zustände" erreichbar  $\Leftrightarrow M = \varphi(\Sigma^*)$ , dann gibt es einen surjektiven Homomorphismus von  $\Sigma^*$  auf  $\mathcal{M}$ , das aber ist schon  $\varphi$ .

- b.) Genau wie in Korollar 10 folgt auch hier:  
 $L \subseteq \Sigma^*$  ist genau dann regulär, wenn  $\sim_L$  endlichen Index hat.

**Satz 20.**

Für jede Sprache  $L \subseteq \Sigma^*$  ist das syntaktische Monoid  $\mathcal{M}_L$  isomorph zum Transformationsmonoid (vgl. Konstruktion 12) des minimalen Automaten  $A_L$ .

**Formalsprachliche Operationen und das syntaktische Monoid**

**Satz 21.** Für beliebige Sprachen  $L_1, L_2 \subseteq \Sigma^*$  gilt:  $\mathcal{M}_{L_1 \cap L_2} \prec \mathcal{M}_{L_1} \times \mathcal{M}_{L_2}$

**Beweis 21.** Setze  $\mathcal{M}' := \{[x]_{L_1}, [x]_{L_2} \mid x \in \Sigma^*\} \subseteq \mathcal{M}_{L_1} \times \mathcal{M}_{L_2}$   
 Definiere:  $\varphi : \mathcal{M}' \rightarrow \mathcal{M}_{L_1 \cap L_2}$  vermöge  $\varphi([x]_{L_1}, [x]_{L_2}) := [x]_{L_1 \cap L_2}$

**Bemerkung 22.**

- $\varphi$  ist wohldefiniert:  
 Sei  $x \sim_{L_1} y$  und  $x \sim_{L_2} y$  zu zeigen  $x \sim_{L_1 \cap L_2} y$ :  
 $\forall z \ zx \setminus L_1 \cap L_2 = zx \setminus L_1 \cap zx \setminus L_2 = zy \setminus L_1 \cap zy \setminus L_2 = zy \setminus L_1 \cap L_2$ .  
 Also  $x \sim_{L_1 \cap L_2} y$ .
- $\varphi$  ist Morphismus:  $\varphi([xy]_{L_1}, [xy]_{L_2}) = [xy]_{L_1 \cap L_2} = [x]_{L_1 \cap L_2} \cdot [y]_{L_1 \cap L_2} = \varphi([x]_{L_1}, [x]_{L_2}) \cdot \varphi([y]_{L_1}, [y]_{L_2})$ .
- $\varphi$  ist offenbar surjektiv.  $\square$
- $\varphi$  ist im Allgemeinen nicht injektiv:  
 Bsp.:  $L_1 = a^*b^*$  und  $L_2 = \{x \in \{a, b\}^* \mid \#_a(x) = \#_b(x)\}$ .  
 Dann ist  $([ba]_{L_1}, [ba]_{L_2}) \neq ([baa]_{L_1}, [ba]_{L_2})$  aber  $[ba]_{L_1 \cap L_2} = [baa]_{L_1 \cap L_2}$ .  
 Also erkennt  $\mathcal{M}_1 \times \mathcal{M}_2$  die Sprache  $L_1 \cap L_2$ :  
 Definiere  $\psi : \Sigma^* \rightarrow \mathcal{M}_1 \times \mathcal{M}_2$  durch  $\psi(x) := ([x]_{L_1}, [x]_{L_2})$   
 (damit  $L_1 \cap L_2 = \psi \circ \psi$ ) und  $E := \{([x]_{L_1}, [x]_{L_2}) \mid x \in L_1 \cap L_2\}$   
 Dann ist  $L_1 \cap L_2 = \psi^{-1}(E)$ .

**Sätzchen 23.**

Ist  $h : \Sigma^* \rightarrow \Pi^*$  ein Homomorphismus und  $L \subseteq \Pi^*$ , so gilt  $\mathcal{M}_{h^{-1}(L)} \prec \mathcal{M}_L$

**Beweis 23.** Mit  $E := \eta_L(L)$  gilt  $\eta_L^{-1}(E) = L$   
 also  $h^{-1}(L) = h^{-1}(\eta_L^{-1}(E)) = (h \circ \eta_L)(E)$ .  
 Also erkennt  $\mathcal{M}_{h^{-1}(L)}$  und daher  $\mathcal{M}_{\eta^{-1}(L)} \prec \mathcal{M}_L$ .  $\square$

**Sätzchen 24.**

Ist  $L \subseteq \Sigma^*$  und  $x \in \Sigma$ , so gelten  $\mathcal{M}_{a \setminus L} \prec \mathcal{M}_L$  und  $\mathcal{M}_{L/a} \prec \mathcal{M}_L$ .

**Beweis 24.**

Setze  $E' := \{[x]_L \in \mathcal{M}_L \mid ax \in L\}$  und  $E'' := \{[x]_L \in \mathcal{M}_L \mid xa \in L\}$ .

Dann gelten  $a \setminus L = h_L^{-1}(E')$  und  $L/a = h_L^{-1}(E'')$ .  $\square$

Denn  $a \setminus L = \{hy \mid ay \in L\} = \{y \mid [a][y] \in E := h_L(L)\} = \{y \mid [y] \in E'\} = h_L^{-1}(E')$ .

$\square$

**Eine Charakteristik des syntaktischen Monoids****Beweis 24.**

Definiere  $\varphi : \mathcal{M}_L \rightarrow T(\mathcal{A}_L)$  ( $T$  das Transformationsmonoid)

vermöge:  $\varphi([x]_L) := \delta^*(\cdot, x)$ .

Behauptung:  $\varphi$  ist ein Isomorphismus:

-  $\varphi$  ist wohldefiniert:

Ist  $x \sim_L y$ , so gilt  $\forall_{z_1, z_2 \in \Sigma^*}$ :

$z_1 x z_2 \in L \Leftrightarrow z_1 y z_2 \in L$  also

$\forall_{z_1} z_1 x \sim_L^R z_1 y$  also

$\forall_{z_1} \delta^*([z_1]_L^R, x) = [z_1 x]_L^R = [z_1 y]_L^R = \delta^*([z_1]_L^R, y)$  also

$\delta^*(\cdot, x) = \delta^*(\cdot, y)$ .

-  $\varphi$  ist injektiv:

Ist  $x \not\sim_L y$  so gilt:

$\exists_{z_1, z_2} z_1 x z_2 \in L \Leftrightarrow z_1 y z_2 \notin L$  also

$\exists_{z_1} [z_1 x]_L^R \neq [z_1 y]_L^R$  und damit  $\exists_{z_1} \delta^*([z_1]_L^R, x) \neq \delta^*([z_1]_L^R, y)$  mithin

$\varphi([x]_L^R) \neq \varphi([y]_L^R)$

-  $\varphi$  ist offenbar surjektiv.

-  $\varphi$  ist Homomorphismus:

$\varphi([x]_2 [y]_2) = \varphi([xy]_2) = \delta^*(\cdot, xy) = \delta^*(\delta^*(\cdot, x), y) =$

$\varphi([x]_2) \cdot \varphi([y]_2)$

## 0.4 Beziehungen zwischen Familien formaler Sprachen und Mengen von Monoiden

Eine Menge  $\mathcal{M}$  von Monoiden heisst eine Varietät, wenn sie abgeschlossen ist unter endlich direkten Produkten und unter Teilerbildung, d.h. ist  $M \in \mathcal{M}$  und  $\mathcal{M}' \prec \mathcal{M}$ , so gilt  $M' \in \mathcal{M}$ .

Beispiele von Varietäten:

- Die endlichen Monoide
- Gruppen
- Kommutative Monoide ( $ab = ba$ )
- Idempotente Monoide ( $a^2 = a$ )

Eine nichtleere Familie von Sprachen heisst Varietät, wenn sie abgeschlossen ist unter folgenden Operatoren auf Sprachen:

- $\cup, \cap, \neg$  (bool'sche Abgeschlossenheit)
- $L \rightarrow h^{-1}(L)$
- $L \rightarrow a \setminus L, L \rightarrow L/a$  (Schreibweise häufig  $a^{-1}L, La^{-1}$ )

### Mitteilung: 25.

- 1.) Varietäten formaler Sprachen und Varietäten von Monoiden stehen in einer Eins-Zu-Eins-Beziehung.
- 2.) Monoidvarietäten lassen sich durch Gleichungen charakterisieren.

Bsp.:  $ab = ba$ :      Kommutativität  
 $a^2 = 1$             Idempotenz  
 $a^w = a^{w+1}$       Aperiodik

### Definition 26.

Die Menge der Zählerfreien (regulären) Sprachen ist die kleinste aller Familien von Sprachen, die alle endlichen Sprachen enthält und abgeschlossen ist unter bool'schen Operationen und Konkatenation.

### Mitteilung: 27.

Folgende Aussagen sind für eine Sprache  $L \subseteq \Sigma^*$  äquivalent:

- a.)  $L$  ist zählerfrei.
- b.)  $\mathcal{M}_L$  enthält keine Gruppe als Teilmenge
- c.)  $L$  wird von einer prädikaten logischen Formel  $\phi \in FO[<]$  erkannt.

Gbsp:  $Parity := \{x \in \{a, b\}^* \mid \#_a(x) = 0 \text{ mod } 2\}$

Bsp.:  $\forall x \forall y \ x < y \Rightarrow Q_a(x) \vee Q_b(y) = a^*b^*$   
 $a^* = \overline{\emptyset b \emptyset}$  und entsprechend  $b^*$   
 $\mathcal{M}_L$  gruppenfrei:  $\forall x, \forall k, k' \ [x^k] = [x^{k+m}] \Rightarrow [x^{k+m}] = [x^{k+m+1}]$

### 0.5 Endlich beschreibbare unendliche Automaten

Ansatz kontextfreie Grammatik  $G = (N, T, P, S)$  in Greibach-Normalform:  $P \subset N \times (T \cup TN \cup TN^2)$

Betrachte nichtdeterministischen Automaten  $\mathcal{A} = (N^*, T, \delta, S, \lambda)$  mit  $\delta(A\alpha, a) = \beta\alpha$  wenn  $A \rightarrow a\beta \in P$ .

**Beispiel 12.**

$G = (N, \{a, b\}, P, S)$   
 $N = \{S\}, P = \{S \rightarrow aSSa, S \rightarrow b\} \ L(G) = \text{Lukasiewicz-Sprache}$

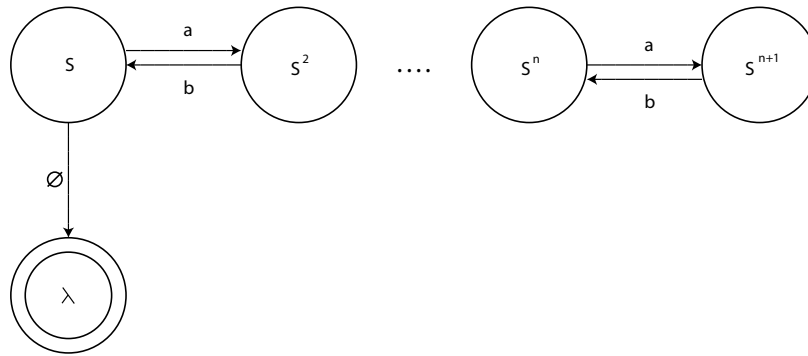


Abbildung 8: Lukasiewicz

**Problem:**

Leerer Keller impliziert Stop; deterministisch nur präfixfreie Sprachen erkennbar.

**Definition 28.**

Ein Präfixautomat ist ein Automat / 5-Tupel  $\mathcal{P} = (N^+, \Sigma, \delta, S, E)$  mit  $S \in N^+$  und  $(N^2 \cup N) \times \Sigma \rightarrow (N^2 \cup N)$ , einer (deterministischen) Übergangsfunktion.

$\delta : (N^2 \cup N) \times \Sigma \rightarrow (N^2 \cup N)$

$\forall A \forall a \forall B \ [\delta(A, a) \text{ undefiniert oder } \delta(AB, a) \text{ undefiniert}]$

$\mathcal{P}$  induziert einen (deterministischen) Automaten:

$\mathcal{A}(\mathcal{P}) = (Q := N^+ \cup \{F\}, \Sigma, \delta, S, F := E \circ N^*)$  mit  $\delta : N^+ \times E \rightarrow N^*$ , definiert vermöge:

$\delta(A\alpha, a) := \eta(A, a)\alpha$  wenn  $(A, a) \in DFg$  bzw.  $\delta(AB\alpha, a) := \eta(AB, a)\alpha$  wenn  $(AB, a) \in DFg$  für  $\alpha \in N^*$ .

Wir betrachten also Kellerautomaten ohne Zustände, die keine  $\lambda$ -Übergänge aufweisen. Sehen wir die unendliche Menge der Kellerinhalte als Zustandsmenge an, so haben wir es mit einem Spezialfall der unendlichen Automaten gemäss Definition 1 zu tun.

**Beispiel 13.**

- 1.) Der obige Automat wird dann zu  $(S, a) \rightarrow SS, (SS, b) \rightarrow S, (S\#, b) \rightarrow SS$
- 2.)  $L_2 = \{x \in \{a, b\}^* \mid \#_a(x) = \#_b(x)\}$  wird von folgendem Präfixautomaten erkannt:  
 $\mathcal{P} = (\{A, B, \#\}, \{a, b\}, \eta, \#)$  mit  $\eta$ :

- $(\#, a) \rightarrow A\#, \quad (\#, b) \rightarrow B\#$   
 $(A, a) \rightarrow AA, \quad (B, b) \rightarrow BB$   
 $(BB, a) \rightarrow B \quad (AA, b) \rightarrow A$   
 $(B\#, a) \rightarrow \# \quad (A\#, b) \rightarrow \#$

Der sich ergebende "unendliche" Automat stimmt mit  $\mathcal{A}_{L_2}$  überein

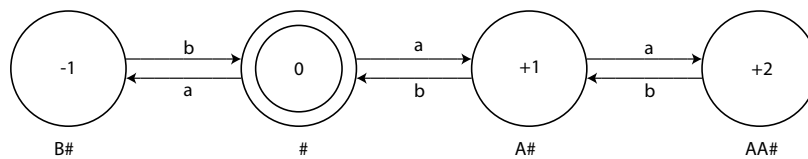


Abbildung 9: entsp. Präfixautomat

Frage:

Wann stimmt  $\mathcal{A}_{\mathcal{P}}$  mit  $\mathcal{A}_{L(\mathcal{P})}$  überein?  
 Hinreichende Bedingungen an  $M_L$ , die Existenz eines Präfix-Automaten implizieren.

**0.6 Konstruktion eines (i.A. unendlichen) Monoids aus einem Präfixautomaten**

Zu  $\mathcal{P} = (N, \Sigma, \delta, S, E)$  betrachte als Grundmenge  $\overline{N}^+ N^+$ , wobei  $\overline{N} := \{\overline{A} \mid A \in N\}$  eine Kopie von  $N$  sei. Elemente des zu konstruierenden Monoids  $\mathcal{M}_{\mathcal{P}}$  werden gewisse "erreichbare" endliche Teilmengen von  $\overline{N}^+ N^+$  sein.

Ziel ist weiterhin die Konstruktion eines Epimorphismus  $\varphi : \Sigma^* \rightarrow \mathcal{M}_{\mathcal{P}}$ . Anschauliche Bedeutung von  $\overline{\alpha} \overline{\beta} \in \varphi(x)$  wird sein, dass  $\mathcal{P}$  aus dem Kellerinhalt  $\alpha$  (+ evtl. der Rest, der nicht gelesen wird) unter Lesen des Eingabewortes  $x$  der Kellerinhalt  $\beta$  entsteht.

Also  $\delta^*(\alpha, x) = \beta$ .

Auf Mengen  $X, Y \subseteq \overline{N}^+ N^+$  definieren wir dazu folgende assoziative Verknüpfung:

$$X \square Y := \{ \overline{\alpha\alpha'} \overline{\eta} \mid \exists \beta \in N^+ : \overline{\alpha} \overline{\beta} \in X, \overline{\beta\alpha'} \overline{\eta} \in Y \} \cup \{ \overline{\alpha} \overline{\eta} \overline{\eta} \mid \exists \beta \in N^+ : \overline{\alpha} \overline{\eta} \overline{\beta} \in X, \overline{\beta} \overline{\eta} \in Y \}$$

Definiere die Abbildung  $\varphi$  zunächst nur auf  $\Sigma$  :  $\varphi(a) := \{ \overline{\alpha} \overline{\beta} \mid \delta(\alpha, a) = \beta \}$

Sei nun  $\mathcal{M}_{\mathcal{P}}$  die Menge aller  $X \subseteq \overline{N}^+ N^+$ , die von den  $\varphi(a), a \in \Sigma$  mit  $\square$  erzeugt werden. ( $\mathcal{M}_{\mathcal{P}}$  ist das gesuchte Monoid.) Dann lässt sich  $\varphi$  eindeutig fortsetzen (da  $\Sigma^*$

frei) zu einem Epimorphismus von  $\Sigma^*$  auf  $\mathcal{M}_{\mathcal{P}}$

Setzt man  $E := \{x \in \mathcal{M}_{\mathcal{P}} \mid \exists \alpha \in FN^* : \overline{S} \overline{\alpha} \in X\}$  so gilt:

$$L(\mathcal{P}) = \varphi^{-1}(E).$$

Denn  $\overline{S} \overline{\alpha} \in X \iff \varphi(x)$  bedeutet  $\delta^*(S, x) = \alpha \in FN^*$ , also  $x \in L(\mathcal{P})$ .

Daher  $\mathcal{M}_{L(\mathcal{P})} \prec \mathcal{M}_{\mathcal{P}}$

**Beispiel 14.**

- Ist  $\mathcal{P}$  ein endlicher Automat, also  $\eta : N \times \Sigma \rightarrow N$ , so ist  $\mathcal{M}_{\mathcal{P}}$  isomorph zum Transformationsmonoid von  $\mathcal{A}_{\mathcal{P}}$
- Das Monoid  $\mathcal{M}_{\mathcal{P}}$  zur Lukasiewicz-Grammatik stimmt mit dem syntaktischen Monoid überein.

Genauer: ist isomorph:  $\varphi$  ist injektiv.

- Das Monoid  $\mathcal{M}_{\mathcal{P}}$  zur Sprache  $L_2 = \{x \in \{a, b\}^* \mid \#_a(x) = \#_b(x)\}$  stimmt nicht mit dem syntaktischen Monoid  $\mathcal{M}_{L_2} = \mathbb{Z}$  überein.

**Bemerkung: 29.**

Schneidet man in der Definition von  $\mathcal{M}_{\mathcal{P}}$  und  $\square$  alles mit  $(\overline{N} \cup \overline{N}^2)(N \cup N^2)$  (dann alles endlich), so ist  $\square$  nicht mehr assoziativ. Statt eines unendlichen Monoids erhält man ein endliches Gruppoid.  $L$  ist kontextfrei, g.d.w.  $L$  "erkennbar" ist durch ein endliches Gruppoid!

Kein Homomorphismus! Was ist  $\varphi(x \circ y)$ ?  $\rightarrow$  nicht definierte Menge!

Fazit: Viele, viele offene Fragen; z.B. Tool (wie Amore) für  $LL(1)$  Sprachen.

**Anmerkung:**

Für kontextfreie Sprachen ist i.A. weder die syntaktische Kongruenz noch die syntaktische Rechtskongruenz entscheidbar. Folgt direkt aus der Unentscheidbarkeit des Äquivalenzproblems: Mit kontextfreien Sprachen  $L_1$  und  $L_2$  ist auch  $L := \$L_1 \cup \dagger L_2$  kontextfrei. Hierbei seien  $\$$  und  $\dagger$  zwei "neue" Symbole, dann gilt offenbar  $[\$]_L^R = [\dagger]_L^R$  g.d.w.  $L_1 = L_2$  ist.

**Problem 'Stetigkeit'**

$$\forall_p \mathbb{Z}/p.\mathbb{Z} \prec \mathbb{Z} \text{ über } \varphi(n) := m \pmod p.$$

Dann ist  $Ke\varphi = p.\mathbb{Z}$ : unendlich, nicht zusammenhängend.

Bei unendlichen Monoiden scheint eine Beschränkung der akzeptierten Mengen nötig zu sein.  $\approx$  Stetigkeit.

**Beispiel 15.**

$L_2 \approx (\mathbb{Z}; \{0\})$  hat syntaktisches Monoid  $\mathbb{Z}$  mit der akzeptierenden Menge  $\mathbb{N}$

$$MAJ := \{x \in \{a, b\}^+ \mid \#_a(x) - \#_b(x) > 0\} \approx (\mathbb{Z}, \mathbb{N})$$

Evenlength  $\#_a(x) - \#_b(x) = 0 \pmod 2 = (\{a, b\}^*)$  hat syntaktisches Monoid  $\mathbb{Z}/2\mathbb{Z}$