

Automaten und Formale Sprachen

SoSe 2013 in Trier

Henning Fernau

Universität Trier

fernau@uni-trier.de

2. Juni 2013

Automaten und Formale Sprachen

Gesamtübersicht

- Organisatorisches
- Einführung
- **Endliche Automaten und reguläre Sprachen**
- Kontextfreie Grammatiken und kontextfreie Sprachen
- Chomsky-Hierarchie

Endliche Automaten und reguläre Sprachen

1. Deterministische endliche Automaten
2. Nichtdeterministische endliche Automaten
3. Reguläre Ausdrücke
4. **Nichtreguläre Sprachen**
5. Algorithmen mit / für endliche Automaten

Das Pumping-Lemma

Satz: Zu jeder regulären Sprache L gibt es eine Zahl $n > 0$, sodass jedes Wort $w \in L$ mit $\ell(w) \geq n$ als Konkatenation $w = xyz$ dargestellt werden kann mit geeigneten x, y, z mit folgenden Eigenschaften:

1. $\ell(y) > 0$;
2. $\ell(xy) \leq n$;
3. $\forall i \geq 0 : xy^i z \in L$.

Hinweis: Die Umkehrung gilt nicht !

Zur Anwendung des Pumping-Lemmas (schematisch)

1. Wir vermuten, eine vorgegebene Sprache L ist nicht regulär.
2. Im Widerspruch zu unserer Annahme nehmen wir an, L wäre doch regulär. Dann gibt es die im Pumping-Lemma genannte **Pumping-Konstante** n .
3. Wir wählen ein geeignetes, hinreichend langes Wort $w \in L$ (d.h., $\ell(w) \geq n$). Dies ist der Schritt, wo man leicht “gut” oder “schlecht” wählt!
Bemerkung: Da wir ja vermuten, L ist nicht regulär, ist L insbesondere unendlich, d.h., zu jedem n finden wir ein $w \in L$ mit $\ell(w) \geq n$.
4. Wir diskutieren alle möglichen Zerlegungen $w = xyz$ mit $\ell(y) > 0$ und $\ell(xy) \leq n$ und zeigen für jede solche Zerlegung, dass es ein $i \geq 0$ gibt, sodass $xy^iz \notin L$ gilt.

Die Komplexität dieser Diskussion hängt im Wesentlichen von der “geschickten” Wahl von w ab.

Das Anfangsstück von w der Länge n sollte “schön” sein.

Zur Anwendung des Pumping-Lemmas (ein geschickter Einsatz)

Betrachte $L = \{a^k b^k \mid k \in \mathbb{N}\}$.

Wäre L regulär, so gäbe es Pumping-Konstante n .

Betrachte $a^n b^n \in L$; denn: $\ell(a^n b^n) = 2n \geq n$ und Präfix der Länge n ist a^n (sehr schön).

Diskutiere $a^n b^n = xyz$ mit $\ell(xy) \leq n$ und $\ell(y) > 0$:

Offenbar ist $xy \in \{a\}^+$ und damit $y = a^m$ für ein $m > 0$.

\leadsto Nullpumpen liefert $a^{n-m} b^n \notin L$, \nexists zur Annahme, L wäre regulär.

Zur Anwendung des Pumping-Lemmas (ein ungeschickter Einsatz)

Betrachte $L = \{a^k b^k \mid k \in \mathbb{N}\}$.

Wäre L regulär, so gäbe es Pumping-Konstante n . Da L nur Wörter gerader Länge enthält, können wir annehmen, n ist gerade.

Betrachte $w = a^{n/2} b^{n/2} \in L$ mit $\ell(w) = n$.

Diskutiere $w = xyz$ mit $\ell(xy) \leq n$ und $\ell(y) > 0$:

Fall 1: $xy \in \{a\}^+$ (Nullpumpen ähnlich wie letzte Folie)

Fall 2: $xy = a^{n/2} b^m$ mit $m > 0$.

Fall 2a: $y \in \{b\}^+$ (Nullpumpen ähnlich wie letzte Folie)

Fall 2b: $y = a^r b^m$ mit $r, m > 0$. Dann liegt auch $xy^2z = a^{n/2} b^m a^r b^{n/2} \in L \nleftrightarrow$ zur Struktur von L .

Die Spiegeloperation

Informell: w^R ergibt sich aus w durch “Rückwärtslesen” (Spiegeln).

Induktiv: $\lambda^R = \lambda$;

für $w = va$ mit $v \in \Sigma^*$, $a \in \Sigma$ definiere: $w^R := a(v^R)$.

Beispiel: $(abcd)^R = d(abc)^R = dc(ab)^R = dcba(a)^R = dcba(\lambda^R) = dcba\lambda = dcba$.

Erweiterung auf Wortmengen: $L^R = \{w^R \mid w \in L\}$.

Satz: Die regulären Sprachen sind unter Spiegelung abgeschlossen.

Beweis: Wichtig: Wahl des richtigen Modells!

Noch eine Anwendung des Pumping-Lemmas

Betrachte $L = \{w \in \{a, b\}^* \mid w = w^R\}$ (*Palindrome*)

Wäre L regulär, so gäbe es Pumping-Konstante n für L .

Betrachte $w = a^n b a^n \in L$ mit $\ell(w) \geq n$.

Widerspruch ergibt sich durch Nullpumpen.

... und noch eine ...

Betrachte $L = \{a^{k^2} \mid k \in \mathbb{N}\}$.

Wäre L regulär, so gäbe es Pumping-Konstante n für L .

Betrachte $w = a^{(n+1)^2} \in L$ mit $\ell(w) \geq n$.

Widerspruch ergibt sich durch Nullpumpen:

Genauer haben wir, dass für ein $0 < i \leq n$ stimmen muss, dass $a^{(n+1)^2-i} \in L$ gilt, im Widerspruch zu folgender Abschätzung, die $a^{(n+1)^2-i} = a^{r^2}$ annimmt:

$$r^2 \leq n^2 < n^2 + n + (n - i) + 1 = n^2 + 2n + 1 - i = (n + 1)^2 - i.$$

Der Beweis des Pumping-Lemmas

Ist L endlich, so ist die Aussage trivial mit $n := \max\{\ell(w) \mid w \in L\} + 1$.

Ist L unendlich aber regulär, so wird L von einem DEA A mit n Zuständen akzeptiert mit Anfangszustand q_0 .

Betrachte ein Wort $w = a_1 \dots a_m \in L$, $a_i \in \Sigma$, $m \geq n$.

Sei $(q_k, a_{k+1} \dots a_m)$ für $0 \leq k \leq n$ die Konfiguration nach k Schritten von A .

Da hierbei $n + 1$ Zustände durchlaufen werden, gibt es nach dem Schubfachprinzip einen Zustand, der zweimal erreicht wird, d.h., $\exists 0 \leq r < s \leq n : q_r = q_s$.

$\leadsto y = a_{r+1} \dots a_s$ erfüllt $(q_r, y) \vdash_A^* (q_r, \lambda)$.

$\leadsto \forall i \geq 0 : (q_r, y^i) \vdash_A^* (q_r, \lambda)$.

Mit $x = a_1 \dots a_r$ und $z = a_{s+1} \dots a_m$ folgt die Behauptung.

Nicht-Regularität durch Abschlusseigenschaften

Beispiel: Betrachte die Menge $L \subseteq \{a, b\}^*$ mit der Eigenschaft, dass $w \in L$ liegt gdw. w gleich viele a 's wie b 's besitzt.

Behauptung: L ist nicht regulär.

Beweis durch Widerspruch: Wäre L regulär, so auch $L' = L \cap \{a\}^*\{b\}^*$, denn

- $\{a\}^*\{b\}^*$ ist regulär, und
- der Schnitt zweier regulärer Sprachen ist wiederum regulär.

Offenbar gilt: $L' = \{a^k b^k \mid k \in \mathbb{N}\}$, und

von dieser Sprache wissen wir bereits, dass sie nicht-regulär ist.

↯ zu unserer Annahme, L wäre regulär.

Auch hier **Schwierigkeit:** “Geschickte” Wahl der Operation...

Äquivalenzrelationen (hoffentlich noch bekannt ?!)

Eine Relation $R \subseteq X \times X$ heißt *Äquivalenzrelation* gdw.

(1) $R^0 = \Delta_X \subseteq R$ (Reflexivität)

(2) $R^2 = R \circ R \subseteq R$ (Transitivität)

(3) Mit $R^{-1} = \{(y, x) \mid (x, y) \in R\}$ gilt $R^{-1} \subseteq R$ (Symmetrie)

Eine ÄR auf X induziert eine *Partition* von X
in *Äquivalenzklassen* $[x]_R = \{y \in X \mid xRy\}$.

Eine Äquivalenzrelation auf Σ^*

Es sei $h : (\Sigma^*, \cdot, \lambda) \rightarrow (M, \circ, e)$ ein Monoidmorphismus.
Dann ist Definiere $x \equiv_h y$ gdw. $h(x) = h(y)$.

Satz: $x \equiv_h y$ ist eine Äquivalenzrelation auf Σ^* .

Erinnerung: Der Kern eines Homomorphismus ist (sogar) eine Kongruenzrelation, also eine Äquivalenzrelation, die mit den Monoid-Operationen verträglich ist.

Noch eine Äquivalenzrelation auf Σ^*

Es sei $A = (Q, \Sigma, \delta, q_0, F)$ ein DEA.

Definiere $u \equiv_A v$ gdw. $\exists q \in Q : ((q_0, u) \vdash_A^* (q, \lambda)) \wedge ((q_0, v) \vdash_A^* (q, \lambda))$.

Satz: $u \equiv_A v$ ist eine Äquivalenzrelation auf Σ^* .

Ein direkter Beweis ist eine gute Übung.

(Es ist klar, dass diese Relation reflexiv, symmetrisch und transitiv ist ?!)

Alternativ: Dies folgt mit der Beziehung über Transformationsmonoide auch unmittelbar aus dem vorigen Satz.

... und noch eine Äquivalenzrelation auf Σ^*

Es sei $L \subseteq \Sigma^*$. L *trennt* zwei Wörter $x, y \in \Sigma^*$ gdw. $|\{x, y\} \cap L| = 1$.

Zwei Wörter u und v heißen *kongruent modulo L* (i.Z.: $u \equiv_L v$), wenn für jedes beliebige Wort w aus Σ^* die Sprache L die Wörter uw und vw *nicht* trennt, d.h. wenn gilt:

$$(\forall w \in \Sigma^*) (uw \in L \Leftrightarrow vw \in L)$$

Satz: Für jede Sprache $L \subseteq \Sigma^*$ ist \equiv_L eine Äquivalenzrelation.

Def.: \equiv_L heißt auch *Myhill-Nerode Äquivalenz*.

Beweis: Reflexivität: $\forall u \in \Sigma^* : u \equiv_L u \checkmark$

Symmetrie: $\forall u, v \in \Sigma^* : u \equiv_L v \Rightarrow v \equiv_L u \checkmark$

Transitivität: $\forall u, v, x \in \Sigma^* : (u \equiv_L v \wedge v \equiv_L x) \Rightarrow u \equiv_L x$

Betrachte $u, v, x, w \in \Sigma^*$ mit $u \equiv_L v$ und $v \equiv_L x$ sowie w beliebig.

(a) Falls $uw \in L$, so auch $vw \in L$, denn $u \equiv_L v$; wegen $v \equiv_L x$ gilt daher $xw \in L$.

(b) Falls $uw \notin L$, so auch $vw \notin L$, denn $u \equiv_L v$; wegen $v \equiv_L x$ gilt daher $xw \notin L$.

(a) und (b) zusammen liefert: $uw \in L \iff xw \in L$, also $u \equiv_L x$, da w beliebig. □

Eigenschaften

Beobachtung. Gilt $u \in L$ und $v \equiv_L u$, so auch $v \in L$.

Beweis: Aus $(\forall w \in \Sigma^*) (uw \in L \Leftrightarrow vw \in L)$ folgt für $w = \lambda$: $u \in L \Leftrightarrow v \in L$, also die Beh. \square

Lemma: \equiv_L ist sogar eine *Rechtskongruenz*,

d.h., aus $u \equiv_L v$ folgt für bel. Wörter $x \in \Sigma^*$: $ux \equiv_L vx$.

Zu zeigen bliebe: $(\forall w \in \Sigma^*) (uw \in L \Leftrightarrow vw \in L)$

impliziert: $(\forall x, w' \in \Sigma^*) (uxw' \in L \Leftrightarrow vxw' \in L)$. (leichte Übung)

Bsp.: $L = \{w \in \{a\}^* \mid |w| \equiv 1 \pmod{3}\}$ hat drei Myhill-Nerode Äquivalenzklassen.

Beispiel: Betrachte

$$L = \{a^k b^k \mid k > 0\}$$

$a^i b \not\equiv_L a^j b$ für $i \neq j$:

Verwende $w = b^{i-1}$ mit $a^i b w \in L$ und $a^j b w \notin L$.

Damit hat man für $i = 1, 2, 3, \dots$ bereits

unendlich viele verschiedene Äquivalenzklassen $[a^i b]$ gefunden.

Genauer gilt: $[a^i b] = \{a^i b, a^{i+1} b^2, a^{i+2} b^3, \dots\}$.

Ferner gilt: $[ab] = L$.

Lemma: Es sei $L \subseteq \Sigma^*$ regulär, d.h., L ist durch ein endliches Monoid (M, \circ, e) , einen Monoidmorphismus $h : \Sigma^* \rightarrow M$ und eine endliche Menge $F \subseteq M$ beschrieben. Dann gilt: Falls $u \equiv_h v$, so $u \equiv_L v$.

Beweis: Betrachte zwei Wörter $u, v \in \Sigma^*$ mit $u \equiv_h v$, also $h(u) = h(v)$.

Da h Morphismus, ist für $w \in \Sigma^*$: $h(uw) = h(u) \circ h(w) = h(v) \circ h(w) = h(vw)$.

Also liegen entweder sowohl uw als auch vw in L oder beide nicht.

Daher gilt $u \equiv_L v$. □

Hinweis: Ähnlicher Beweis über DEA-Äquivalenz \equiv_A !

Folgerung: Ist L regulär, so hat \equiv_L nur endlich viele Äquivalenzklassen.

Noch mehr Folgerungen aus dem letzten Beweis:

Betrachte reguläre Sprache $L \subseteq \Sigma^*$ und sie beschreibende Homomorphismen h bzw. Automaten A :

Ist $\mathcal{L} := \{L_1, \dots, L_n\}$ die durch \equiv_L induzierte Zerlegung von Σ^* ,
so gilt für die durch \equiv_h induzierte Zerlegung $\mathcal{H} := \{H_1, \dots, H_m\}$ von Σ^*
(bzw. für die durch \equiv_A induzierte Zerlegung $\mathcal{A} := \{A_1, \dots, A_\ell\}$ von Σ^*):

Für jedes H_i (bzw. A_i) gibt es ein L_j mit $H_i \subseteq L_j$ (bzw. $A_i \subseteq L_j$).

Daher heißen \mathcal{H} und \mathcal{A} auch **Verfeinerungen** von \mathcal{L} .

Bsp.: Der Homomorphismus $h : \{a\}^* \rightarrow \mathbb{Z}_6, w \mapsto |w| \bmod 6$ beschreibt mit $F = \{1, 4\}$ die Sprache $L = \{w \in \{a\}^* \mid |w| \equiv 1 \pmod{3}\}$.

Jede Äquivalenzklasse von \equiv_L enthält / besteht aus genau zwei Äquivalenzklassen von \equiv_h .

Satz: [Myhill und Nerode] Eine Sprache $L \subseteq \Sigma^*$ ist genau dann regulär, wenn es nur endlich viele Äquivalenzklassen bezüglich \equiv_L gibt.

Beweis: 1. L regulär $\Rightarrow L$ hat endlich viele Äquivalenzklassen (siehe Folgerung).

2. Umkehrung: Sei k Zahl der Klassen von \equiv_L , d.h. $\Sigma^* = [x_1] \cup \dots \cup [x_k]$.

Definiere den *Minimalautomaten* $A(L) = (S, \Sigma, \delta, s_0, F)$ durch

$Q = \{[x_1], \dots, [x_k]\}$

$q_0 := [\lambda]$

F bestehe aus allen Äquivalenzklassen $[x_i]$ mit $x_i \in L$

$\delta([x], a) := [xa]$

Wichtig: Mit $[x] = [y]$ ist $xaw \in L \Leftrightarrow yaw \in L$,

also auch $[xa] = [ya]$, \rightsquigarrow

$$\delta([x], a) = [xa] = [ya] = \delta([y], a)$$

$\rightsquigarrow \delta$ ist wohldefiniert! (Rechtskongruenz!)

Offensichtlich gilt (Beweis ist eine einfache Induktion): $([\lambda], x) \vdash_A^* ([x], \lambda) \rightsquigarrow$

$$x \in L(A) \iff \exists q \in F : ([\lambda], x) \vdash_A^* (q, \lambda) \iff [x] \in F \iff x \in L$$

□

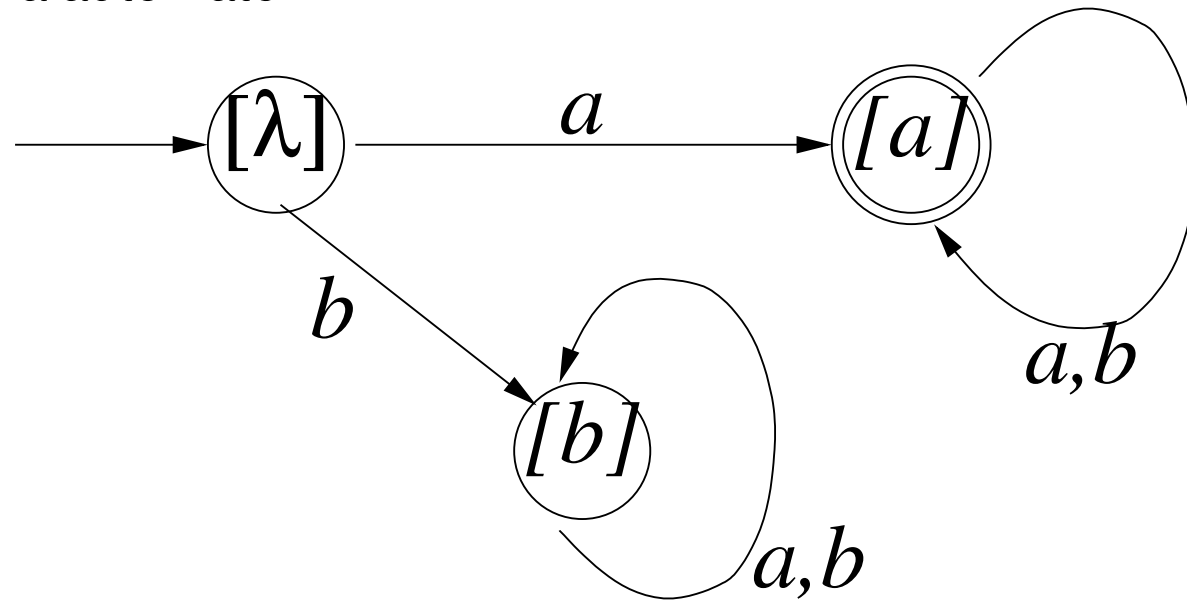
Beispiel: Betrachte $L = \{a\}\{b, a\}^*$

- $\lambda \not\equiv_L a$ mit $\lambda\lambda = \lambda \notin L$, $a\lambda = a \in L$.
- $b \not\equiv_L a$ mit $b\lambda = b \notin L$, $a\lambda = a \in L$
- $\lambda \not\equiv_L b$ mit $\lambda a = a \in L$, $b a \notin L$.
- $b w \equiv_L b$
- $a w \equiv_L a$

Also gilt

$$\Sigma^* = [\lambda] \cup [b] \cup [a]$$

mit dem Minimalautomaten



Warum heißt der Minimalautomat so?

Lemma: Ist L regulär, so ist $A(L)$ der L akzeptierende DEA mit der kleinsten Anzahl von Zuständen.

Beweis: Zunächst sieht man: $\equiv_L = \equiv_{A(L)} \rightsquigarrow$

Zustände von $A(L)$ ist gleich # Äquivalenzklassen von \equiv_L .

Aus dem Beweis von obigem Lemma lesen wir ab:

Ist A ein DEA mit $L = L(A)$, so ist:

Zustände von A ist gleich

Äquivalenzklassen von \equiv_A ist größer gleich

Äquivalenzklassen von \equiv_L .

Automatenmorphismen

Es seien $A_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$ und $A_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$ DEAs.

Eine Funktion $f : Q_1 \rightarrow Q_2$ heißt *Automaten(homo)morphismus* von A_1 nach A_2 gdw.:

- Für alle $a \in \Sigma$ und für alle $q \in Q_1$ gilt $f(\delta_1(q, a)) = \delta_2(f(q), a)$.
- $f(q_{01}) = q_{02}$.
- Für alle $q \in Q_1$ gilt: $q \in F_1 \iff f(q) \in F_2$.

Ist f bijektiv, ist f ein *Automatenisomorphismus*.

Vgl. die Begriffsbildungen aus DS!

Exkurs Automatenmorphismus

Satz: Es seien $A_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1)$ und $A_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2)$ DEAs und $f : Q_1 \rightarrow Q_2$ ein Automatenmorphismus. Dann gilt: $L(A_1) = L(A_2)$.

Beweis: Betrachte $w \in \Sigma^n$ mit $w \in L(A_1)$, $w = a_1 \cdots a_n$ mit $a_i \in \Sigma$ für $1 \leq i \leq n$.

Es gibt also eine Folge von Zuständen $q_{01}, q_{11}, \dots, q_{n1}$ aus Q_1 , $q_{n1} \in F_1$ mit $\delta_1(q_{i-1,1}, a_i) = q_{i,1}$ für $i = 0, \dots, n-1$.

Da $f : Q_1 \rightarrow Q_2$ Automatenmorphismus, gibt es eine Folge von Zuständen $q_{02}, q_{12}, \dots, q_{n2}$ aus Q_2 mit $f(q_{i1}) = q_{i2}$. Hierfür gilt: $f(q_{01}) = q_{02}$ ist der Anfangszustand von A_2 , $q_{n2} \in F_2$, und $\delta_2(q_{i-1,2}, a_i) = \delta_2(f(q_{i-1,1}), a_i) = f(\delta_1(q_{i-1,1}, a_i)) = f(q_{i,1}) = q_{i,2}$ für $i = 0, \dots, n-1$.

Also gilt: $w \in L(A_2)$.

Gilt $w \in \Sigma^n$ mit $w \notin L(A_1)$, so überführt w den Automaten A_1 in einen Nicht-Endzustand q_{n1} .

Mit derselben Überlegung wie soeben überführt w den Automaten A_2 ebenfalls in einen Nicht-Endzustand q_{n2} . □

Es gibt nur einen Minimalautomaten

Lemma: Der Minimalautomat ist “bis auf Isomorphie” (Umbenennen der Zustände) eindeutig bestimmt.

Beweis: Es sei L regulär und n die Zustandsanzahl von $A(L)$ sowie die eines evtl. anderen DEA $A = (Q, \Sigma, \delta, q_0, F)$ mit $L(A) = L$.

(Erinnerung: Allgemein gilt $|Q(A)| \geq n$ für DEAs A mit $L(A) = L$, denn \equiv_A ist eine Verfeinerung von \equiv_L .)

Gilt nun sogar $|Q| = n$, so ist $\equiv_L = \equiv_A$.

$\rightsquigarrow x \equiv_L y \iff x \equiv_A y \iff \exists q \in Q : ((q_0, x) \vdash_A^* (q, \lambda)) \wedge ((q_0, y) \vdash_A^* (q, \lambda))$ für alle $x, y \in \Sigma^*$.

Definiere $\phi : Q \rightarrow 2^{\Sigma^*}, q \mapsto \{w \in \Sigma^* \mid (q_0, w) \vdash_A^* (q, \lambda)\}$.

ϕ identifiziert die Zustände von A mit den Äquivalenzklassen von \equiv_A und somit mit denen von \equiv_L . Anfangs- und Endzustände werden erhalten.

Für irgendein Wort $w_q \in \phi(q)$ gilt:

(1) $([w_q]_L, a) \vdash_{A(L)} ([w_q a]_L, \lambda)$ sowie (2) $(q, a) \vdash_A (q', \lambda)$ mit $\phi(q') = [w_q a]_L$.

Daher wird auch die Übergangsfunktion mit ϕ erhalten $\rightsquigarrow \phi$ ist Automatenisomorphismus. \square

Eine Anwendung des Satzes von Myhill und Nerode

Folgerung: Hat \equiv_L unendlich viele Äquivalenzklassen, so ist L nicht regulär.

Beispiel: Zu

$$L = \{a^k b^k \mid k > 0\}$$

hat \equiv_L unendlich viele Äquivalenzklassen, L ist also nicht regulär.

Noch eine Äquivalenzrelation

Def.: Es sei $L \subseteq \Sigma^*$. $u, v \in \Sigma^*$ heißen *syntaktisch kongruent modulo L*, i.Z. $u \equiv_L^{\text{synt}} v$ gdw. $\forall x, y \in \Sigma^*: (xuy \in L \iff xvy \in L)$.

Beobachte: $u \equiv_L^{\text{synt}} v$ impliziert $u \equiv_L v$.

Satz: Für jede Sprache L ist $u \equiv_L^{\text{synt}} v$ eine Kongruenzrelation.

Auf der Menge der Kongruenzklassen ist also eine “Konkatenation” wohldefiniert und macht diese zu einem Monoid, dem *syntaktischen Monoid* von L .

Folgerung: Eine Sprache ist regulär gdw. sie besitzt ein endliches syntaktisches Monoid.

Satz: Ist L regulär, so ist das syntaktische Monoid von L isomorph zum Transformationsmonoid des Minimalautomaten $A(L)$ von L .

Folgerung: Ist $L \subseteq \Sigma^*$ regulär und (M, \circ, e) das Transformationsmonoid von $A(L)$ mit zugehörigem Morphismus $h : \Sigma^* \rightarrow M$, so ist \equiv_h die syntaktische Kongruenz von L .