FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
TECHNISCHE FAKULTÄT

Technische Fakultät — Lehrstuhl für Informatik 8 · Theoretische Informatik

Skript der Veranstaltung

Algebraische und Logische Aspekte der Automatentheorie (ALoA)

gehalten im Wintersemester 2018/19 von
Dr. Henning Urbat

T.CS

Vorbermerkung

Dieses Skript enthält den Inhalt der Vorlesung „Algebraische und Logische Aspekte der Automatentheorie“ des Wintersemester 2018/19 bei Dr. Henning Urbat. Es wurde anhand von Mitschriften in LaTeX gesetzt und erhebt daher weder Anspruch auf Korrektheit noch Vollständigkeit und ist ***offensichtlich inoffiziell***. Bei Unstimmigkeiten und evtl. vorhandenen Fehlern bitte ich um eine Email an untenstehende Adresse. Dieses Skript stellt damit insbesondere **keine** offizielle Veröffentlichung des Lehrstuhl für Theoretische Informatik am Department Informatik der Friedrich-Alexander-Universität Erlangen-Nürnberg dar.

Florian Frank — florian.ff.frank@fau.de
Version vom 18. Februar 2019

Literaturvorschläge

[Egg63] L. C. Eggan. „Transition graphs and the star-height of regular events.“ In: *The Michigan Mathematical Journal* 10.4 (1963), S. 385–397. DOI: 10.1307/mmj/1028998975. URL: https://projecteuclid.org/DPubS/Repository/1.0/Disseminate?view=body&id=pdf_1&handle=euclid.mmj/1028998975.

[Has88] Kosaburo Hashiguchi. „Algorithms for determining relative star height and star height“. In: *Information and Computation* 78.2 (1988), S. 124–169. DOI: 10.1016/0890-5401(88)90033-8. URL: https://www.sciencedirect.com/science/article/pii/0890540188900338.

[PP04] Dominique Perrin und Jean Éric Pin. *Infinite words automata, semigroups, logic and games*. 1. Pure and applied mathematics. 2004, XI, 538 S. graph. Darst. ISBN: 0-12-532111-2.

[Pin18] Jean-Éric Pin. *Mathematical Foundations of Automata Theory*. 2018, 346 Seiten. URL: https://www.irif.fr/~jep/PDF/MPRI/MPRI.pdf.

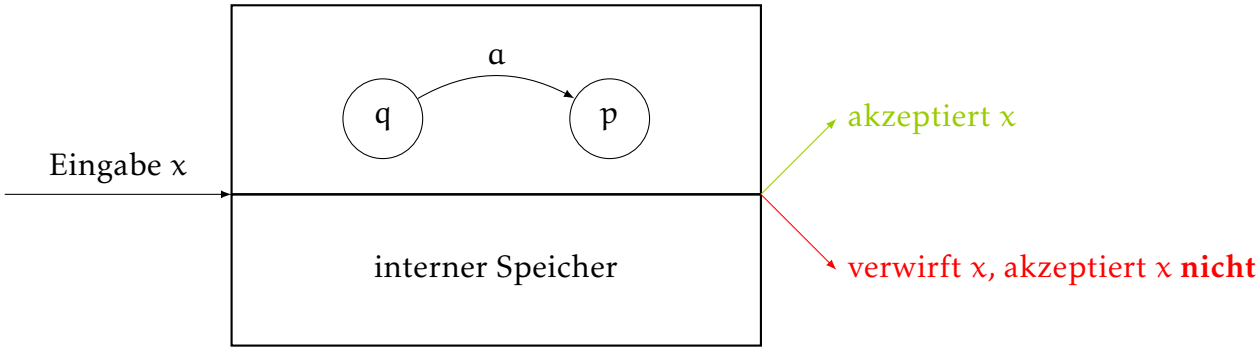
[Str94] Howard Straubing. *Finite automata, formal logic, and circuit complexity*. Progress in theoretical computer science. 1994, XII, 226 S. ISBN: 0-8176-3719-2 3-7643-3719-2.

Inhaltsverzeichnis

KAPITEL E

EINFÜHRUNG

Wir definieren zu Beginn den Begriff eines **Automaten** als ein *mathematisches Modell eines Algorithmus oder einer Maschine*.



Wir definieren die **Sprache** eines Automaten als die **Menge aller akzeptierten Eingaben**.
Es gibt hunderte Typen von Automaten in der Literatur. Sie unterscheiden sich ...

- ... in der Art der Eingaben: endliche oder unendliche Wörter, endliche oder unendiche Bäume, Graphen, ...
- ... in der Art des Speichers: keinen Speicher, einen Stack oder ein Turingband.
- ... in der Verarbeitung der Eingaben: von links nach rechts, von rechts nach links, top-down, bottom-up, bidirektional, ...
- ... in dem Freiheitsgrad der Zustandsübergänge: deterministisch, nicht deterministisch, probabilistisch, ...

endliche Automaten

B

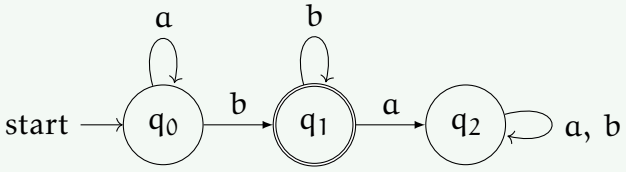
Endliche Automaten akzeptieren genau die **regulären Sprachen**.

→ endliche Wörter als Eingabe

→ keinen Speicher

→ Verarbeitung von links nach rechts

Anwendung: String Matching



Der Automat akzeptiert alle Wörter der Form

$a^m b^n$ mit $m \geqslant 0, n \geqslant 1$,

damit also die Sprache $a^* b^+$.

Büchi-Automaten

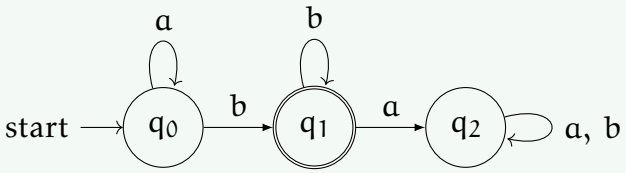
Büchi-Automaten akzeptieren genau die ω-**regulären Sprachen**.

→ unendliche Wörter als Eingabe

→ keinen Speicher

→ Verarbeitung von links nach rechts

Anwendung: Verifikation temporaler Systemeigenschaften



Der Automat akzeptiert die Sprache

a^*b^ω.

(*Akzeptanz wenn ein Finalzustand unendlich oft besucht wird.*)

Baumautomaten

Baumautomaten akzeptieren genau die **Baumsprachen**.

→ Bäume als Eingabe

→ keinen Speicher

→ Verarbeitung top-down oder bottom-up

Anwendung: XML-Parsing

Kellerautomaten

Kellerautomaten akzeptieren genau die **kontextfreien Sprachen**.

→ endliche Wörter als Eingabe

→ Stack

→ Verarbeitung von links nach rechts

Turingmaschinen

Turingmaschinen akzeptieren genau die **entscheidbaren Sprachen**.

→ endliche Wörter als Eingabe

→ Turingband

→ bidirektionale Verarbeitung

Zwei typische Fragen, welche wir uns hierbei stellen sind die der ...

... **Expressivität**: Welche Sprachen lassen sich mit Automaten eines gegebenen Typs akzeptieren und wo liegen die Grenzen dieses Typs? Wie unterscheiden sich Varianten eines Automatentyps? (bspw. Determinismus und Nichtdetereminismus)

... **Entscheidbarkeit**: Betrachte Entscheidungsprobleme der Form

Gegeben: Automat A

Frage: Hat A die Eigenschaft P?

Gesucht sind effiziente Algorithmen beziehungsweise Unentscheidbarkeitsergebnisse.

Diese Vorlesung soll eine logische, algebraische und topologische Sicht auf Automaten geben.

(1) **Automaten vs. Logik**

Idee: Beschreibe Eigenschaften von Wörtern (oder auch Bäumen und Graphen) durch logische Formen.

Beispiel: Die Wörter der Sprache a^*b^* sind durch die Eigenschaft „es steht kein b links von einem a“ charakterisiert. Als Formel in **monadischer Logik 2. Stufe (MSO)**:

$$\neg\,\exists\, x,\, y.\,[\, x < y \;\wedge\; P_b(x) \;\wedge\; P_a(y)\,]$$

Positionen in einem Wort — Pos. x ist links von y — An Pos. x steht ein b — An Pos. y steht ein a

Wir werden feststellen, dass **Reguläre Sprachen = durch MSO-Formeln beschreibbare Sprachen**.

Ziel: (a) Modelltheoretische Charakterisierung von Spracheigenschaften
„Ein Sprache hat Eigenschaft $P \Leftrightarrow$ sie lässt sich durch eine Formel vom Typ T beschreiben.“

(b) Verwende Automaten, um Gültigkeit logischer Formeln zu entscheiden:

$$\forall\, k,\, l,\, m,\, n \in \mathbb{N}.\,[\, k < l \;\wedge\; m < n \;\Rightarrow\; k + m \;<\; l + n\,]$$

Strategie: Übersetze Formel in einen „äquivalenten“ Automaten und teste, ob dieser Automat alle Eingaben akzeptiert.

(2) **Automaten vs. Algebra**

Idee: Assoziiere algebraische Strukturen zu Automaten/Sprachen.
Wir werden feststellen, dass **Reguläre Sprachen = durch endliche Monoide erkennbare Sprachen**.

Ziel: (a) Algebraische Charakterisierung von Spracheigenschaften
„Ein Sprache hat Eigenschaft $P \Leftrightarrow$ assoziierte Algebra hat Eigenschaft P'.“

(b) Konsequenz: Entscheidbarkeitsergebnisse — Wenn P' entscheidbar ist, dann auch P.

(3) **Automaten vs. Topologie**

Idee: Spracherkennung durch Automaten/Algebren als „stetiges“ Problem.
Wir werden feststellen, dass **Reguläre Sprachen = offen-abgeschlossene Teilmengen eines metrischen Raumes**.

Ziel: Charakterisierung von Spracheigenschaften durch **proendliche Gleichungen**.

KAPITEL 1

AUTOMATEN UND REGULÄRE SPRACHEN

Definition 1.1 (*Alphabet, Wörter, Konkatenation, Potenzen und Sprachen*)

- Ein **Alphabet** Σ ist eine *endliche* Menge von *Symbolen*.
  Beispiele: $\Sigma = \{0,1\}$ (BINÄR- oder INFORMATIKERALPHABET), $\Sigma = \{\text{ASCII-ZEICHEN}\}$.
- Ein **Wort über Σ** ist eine Zeichenfolge $w = a_1 \dots a_n$ mit $n \geqslant 0$ und $a_1, \dots, a_n \in \sigma$.
  Spezialfall $n = 0$: Wir sprechen dann vom leeren Wort ε.
  Sei n die Länge von w, so notieren wir $n =: |w|$.
- Σ^* := Menge *aller* Wörter über Σ
- Σ^+ := Menge *aller* ***nichtleeren*** Wörter über Σ
- Die **Konkatenation** zweier Wörter $v = a_1 \dots a_n$ und $w = b_1 \dots b_m$ mit $v, w \in \Sigma^*$ definieren wir als
  $$vw = a_1 \dots a_n b_1 \dots b_m.$$
  Insbesondere gilt
  $$v\varepsilon = \varepsilon v = v.$$
- Für $w \in \Sigma^*$ und $n \geqslant 0$ definieren wir die **n-te Potenz** als $w^n = \overbrace{w \dots w}^{n \text{ mal}}$. Formal:
  $$w^0 = \varepsilon \text{ und } w^n = w^{n-1}w (\text{für } n \geqslant 1)$$
- Wir definieren eine **Sprache über Σ** als eine Menge von Wörtern über Σ, also eine Teilmenge $L \subseteq \Sigma^*$.

Definition 1.2 (*nichtdeterministischer Automat und Läufe*)

Ein **nichtdeterministischer Automat (NA)** ist ein 5-Tupel

$$A = (Q, \Sigma, \to, I, F) \text{ mit}$$

- Q der Menge an Zuständen
- Σ das Alphabet
- $\to \subseteq Q \times \Sigma \times Q$ der Zustandsübergangsrelation (man schreibt $q \xrightarrow{a} q'$ für $(q, a, q') \in \to$)
- I der Menge an Initialzuständen und
- F der Menge an Finalzuständen.

Ein **Lauf** von A bei Eingabe $w = a_1 \dots a_n \in \Sigma^*$ ist eine Liste an Zuständen $q_1, \dots, q_n \in Q$ mit (i) $q_0 \in I$ und (ii) $q_0 \xrightarrow{a_1} \dots \xrightarrow{a_n} q_n$. Der Lauf ist **akzeptierend**, falls $q_n \in F$.

A **akzeptiert** $w \in \Sigma^*$, wenn es einen akzeptierenden Lauf für w gibt. Die von A akzeptierte Sprache ist

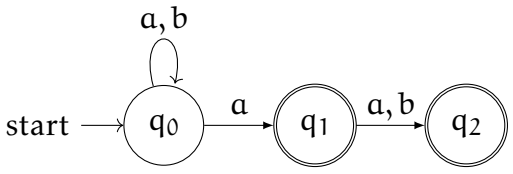
$$L(A) = \{w \in \Sigma^* \mid A \text{ akzeptiert } w\}.$$

Für $q, q' \in Q$ und $w = a_1 \dots a_n \in \Sigma^*$ schreibe $q \xrightarrow{w} q' \Leftrightarrow \exists q = q_0, q_1, \dots, q_n = q'$ mit $q = q_0 \xrightarrow{a_1} \dots \xrightarrow{a_n} q_n = q'$.
Damit ist es möglich die Definition der Sprache als

$$L(A) = \{w \in \Sigma^* \mid \exists\, q_i \in I, q_F \in F.\ q_i \xrightarrow{w} q_F\}$$

zu schreiben.

Beispiel 1.1:
Der nebenstehende Automat lässt sich durch $Q = \{q_0, q_1, q_2\}$, $\Sigma = \{a, b\}$, $I = \{q_0\}$, $F = \{q_1, q_2\}$ und $\to = \{(q_0, a, q_0), (q_0, b, q_0), (q_0, a, q_1), (q_1, a, q_2), (q_1, b, q_2)\}$ beschreiben.



Wir sehen, dass der nebenstehende Automat die Läufe q_0, q_0, q_0, q_0 und q_0, q_0, q_1, q_2 hat. Die Sprache des Automaten ist durch

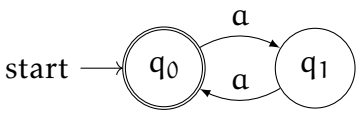
$$\begin{aligned} L(A) &= \text{alle Wörter auf } \{a, b\}^*, \text{ deren letztes oder vorletztes Symbol ein } a \text{ ist} \\ &= (a + b)^* a (b + \varepsilon). \end{aligned}$$

※

Definition 1.3 (*NEA*)

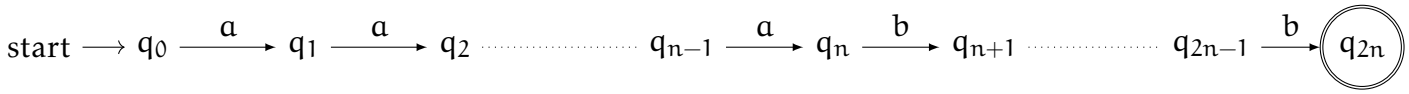
Ein **nichtdeterministischer *endlicher* Automat (NEA)** ist ein NA mit endlicher Zustandsmenge.
Eine Sprache $L \subseteq \Sigma^*$ ist **regulär**, wenn es einen NEA gibt mit $L(A) = L$.

Beispiel 1.2: Die Sprache $L = \{w \in \{a\}^* \ :\ |w| \text{ ist gerade}\}$ ist regulär. *Beweis der Tatsache:* Wir zeigen das durch Angabge des folgenden Automaten:

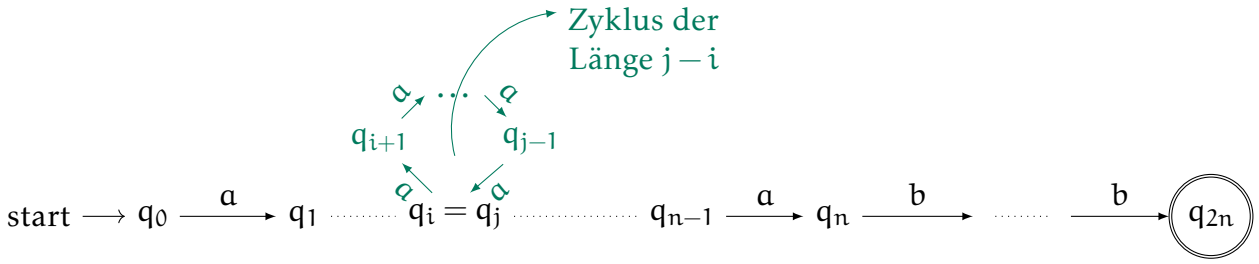


※

Beispiel 1.3: Die Sprache $L = \{a^n b^n \ :\ n \geqslant 0\}$ ist **nicht** regulär.
Intuition: Endliche Sprachen können sich wegen der Endlichkeit ihrer Zustandsmenge **keine beliebig großen** Zahlen merken.
Beweis durch Widerspruch: Angenommen L sei regulär. Sei $A = (Q, \{a, b\}, \to, I, F)$ NEA mit $L(A) = L$. Sei $n := |Q|$.
Wegen $a^n b^n \in L$ gibt es in A einen akzeptierenden Lauf für $a^n b^n$:



Wegen $n = |Q|$ sind die Zustände $q_0, \dots, q_n$ **nicht** paareweise verschieden. Es gibt also $0 \leqslant i < j \leqslant n$ mit $q_i = q_j$:



Das Entfernen des Zyklus liefert einen akzeptierenden Lauf für $a^{n-(j-i)}b^n \notin L$. Dies steht im Widerspruch zur Aussage, dass $L(A) = L$, und damit ist L **nicht regulär**. □ ※

Definition 1.4 (*D(E)A*)

Ein **deterministischer (endlicher) Automat** (D(E)A) ist ein N(E)A $A = (Q, \Sigma, \to, I, F)$ mit

- $|I| = 1$, dass heißt es gibt **genau einen** Initialzustand
- für alle $q \in Q$, $a \in \Sigma$ gibt es **genau** ein $q' \in Q$, so dass $q \xrightarrow{a} q'$ (*Notation:* $q' =: q.a$)

Für $q \in Q, w \in \Sigma^*$ ist $q.w$ der eindeutige Zustand, der von q aus bei Eingabe w erreicht wird.
Formal: $q.\varepsilon = q$ und $q.(va) = (q.v).a$ für alle $v \in \Sigma^*, a \in \Sigma$.

Fakt: Zu jedem N(E)A A gibt es einen D(E)A A_{det} mit $L(A_{det}) = L(A)$.
Idee: A_{det} simuliert parallel alle möglichen Läufe von A. Dazu merkt sich A_{det} über seinen Zustand, in welcehn Zuständen A sein könnte. Wir wollen die Äquivalenz

$$\text{Zustand von } A_{det} \mathrel{\widehat{=}} \text{ Menge von Zuständen von } A$$

Implementierung: Potenzmengenkonstruktion
Gegeben sei ein NA $A = (Q, \Sigma, \to, I, F)$, so definiere den DA A_{det} wie folgt:

- **Zustandsmenge:** $\mathfrak{P}(Q) = \{S \mid S \subseteq Q\}$ (Potenzmenge von Q)
- **Alphabet** Σ
- **Startzustand:** $I \in \mathfrak{P}(Q)$
- **Finalzustände:** $F_{det} = \{S \in \mathfrak{P}(Q) \mid S \cap F \neq \emptyset\}$
- **Übergänge:** $S.a = \{q \in Q : \exists\, p \in S : p \xrightarrow{a} q\}$ für $S \in \mathfrak{P}(Q)$, $a \in \Sigma$.

Wir behaupten nun:

Satz 1.1 (*Äquivalenz von D(E)A und N(E)A*)

Es gilt: $L(A_{det}) = L(A)$.

Beweisskizze: Der Fakt

$$\forall\, S \in \mathfrak{P}(Q), w \in \Sigma^*.\ S.w = \{q \in Q : \exists\, p \in S.\ p \xrightarrow{w} q\}$$

ist leicht zu sehen, man zeigt ihn per Induktion noch $|w|$.
Also gilt

$$\begin{aligned} w \in L(A) &\Leftrightarrow \exists q_i \in I, q_F \in F.\ q_i \xrightarrow{w} q_F \\ &\Leftrightarrow \exists q_F \in F.\ q_F \in I.w \\ &\Leftrightarrow I.w \text{ final in } A_{det} \Leftrightarrow w \in L(A_{det}) \end{aligned}$$ □

! **Beachte**: Wegen $|\mathfrak{P}(Q)| = 2^{|Q|}$ ist A_{det} **exponentiell** größer als A.
Die Potenzmengenkonstruktion ist **optimal**, das heißt es gibt im Allgemeinen keinen DEA A' mit $L(A') = L(A)$, der kleiner als A_{det} ist.

1.1 Operationen auf Sprachen $L, M \subseteq \Sigma^*$

(1) Boolsche Operationen:

- $L \cup M := \{w \in \Sigma^* : w \in L \vee w \in M\}$
- $L \cap M := \{w \in \Sigma^* : w \in L \wedge w \in M\}$
- $\overline{L} = L^C := \{w \in \Sigma^* : w \notin L\}$

(2) Konkatenation:

$$LM = \{vm : v \in L,\ w \in M\}$$

(3) Kleene-Stern:

Für $n \geqslant 0$ schreibe $L^n = \overbrace{L \dots L}^{n \text{ mal}}$.
Formal: $L^0 := \{\varepsilon\}$, $L^n := L^{n-1}L$ für $n > 0$. Dann ist der Kleene-Stern definiert als

$$L^* := \bigcup_{n \geqslant 0} L^n$$

Also $w \in L^* \Leftrightarrow$ es existieret eine Zerlegung $w = w_1 \dots w_n$ mit $n \geqslant 0$ und $w_1, \dots, w_n \in L$.

Es stellt sich nun natürlich die Frage nach Abgeschlossenheit:

Satz 1.2 (*Abgeschlossenheit regulärer Sprachen*)

Für alle regulären Sprachen $L, M \subseteq \Sigma^*$ sind auch $L \cup M$, $L \cap M$, $\overline{L}$, LM und L^* regulär.

Beweisidee: Gegeben seien NEAs A, B für L und M. Baue nun NEAs für $L \cup M$, $L \cap M$, $\overline{L}$, LM und L^*:

- $L \cup M, L \cap M$: Parallelschaltung der NEAs
- $\overline{L}$: vertausche Final- und Nichtfinalzustände
- LM: sequentielle Komposition
- L^*: Erweitere A um Feedback-Übergänge. □

Definition 1.5 (*Reguläre Ausdrücke*)

Die regulären Ausdrücke über Σ sind induktiv wie folgt definiert:

- $\emptyset$ ist RA
- ε ist RA
- a ist für $a \in \Sigma$ RA
- Sind r und s RA, so auch $r + s, rs$ und r^*.

Jeder RA r repräsentiert eine Sprache $L(r) \subseteq \Sigma^*$:

- $L(\emptyset) = \emptyset$
- $L(\varepsilon) = \{\varepsilon\}$
- $L(a) = \{a\}$
- $L(r + s) = L(r) \cup L(s), L(rs) = L(r) \cap L(s)$ und $L(r^*) = L(r)^*$.

Es gilt die Operatorpräzedenz: $()^* > (\cdot) > (+)$
Fakt: Eine Sprache $L \subseteq \Sigma^*$ ist genau dann **regulär**, wenn ein regulärer Ausdruck r existiert mit $L(r) = L$.
Normalerweise treffen wir keine Unterscheidung zwischen regulären Ausdrücken r und deren Sprachen $L(r)$, so steht beispielsweise $r = s$ für $L(r) = L(s)$ und nicht unbedingt für syntaktische Glechheit.

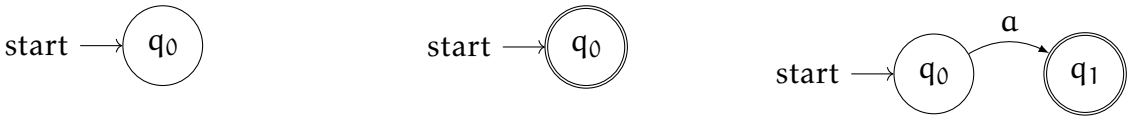
Satz 1.3

Eine Sprache $L \subseteq \Sigma^*$ ist genau dann regulär, wenn ein regulärer Ausdruck r mit $L(r) = L$ existiert.

Beweis:

„⇐“ Für jeden regulären Ausdruck r ist $L(r)$ regulär, denn:

- Die Sprachen $\emptyset, \{\varepsilon\}, \{a\}$ sind regulär. Wir zeigen dies durch Angabe von Automaten:



- Die regulären Sprachen sind unter Vereinigung, Konkatenation und Kleene-Stern abgeschlossen.

„⇒“ Sei $L \subseteq \Sigma^*$ regulär und $A = (Q, \Sigma, \to, I, F)$ ein NEA mit $L(A) = L$.

Ziel: Konstruiere RA r mit $L(r) = L$.
Idee: r ist die Lösung eines Gleichungssystems das A beschreibt.

Wir betrachten dazu folgendes Lemma, welches wir in als Übungsaufgabe zeigen wollen:

Lemma 1.4 (*Arden*)

Seien $K, L, M \subseteq \Sigma^*$ Sprachen mit $\varepsilon \notin K$. Dann gilt:

$$L = KL \cup M \Leftrightarrow L = K^*M.$$

Das heißt die Gleichung $L = KL \cup M$ (mit Unbekannter L und Konstanten K, M) hat die eindeutige Lösung $L = K^*M$.

Beweis: siehe Übungsblatt 1 Aufgabe 3 □
Eine *Konsequenz*, welche sich daraus ergibt ist, wenn r, s RA für K und M sind, r^*s ein RA für L ist.
Sei $Q = \{q_0, \ldots, q_n\}$. Definiere dann:

$$L_i := \left\{ w \in \Sigma^* \,:\, q_i \xrightarrow{w} q_F \quad \text{für ein } q_F \in F \right\}. \qquad (i \in \{0, \ldots, n\})$$

$$\text{Beachte:} \qquad L(A) = \bigcup_{q_i \in I} L_i \qquad (*)$$

Wir definieren dann noch die Sprache $K_{i,j}$, welche die Transitionen des Automaten durch

$$K_{i,j} := \left\{ a \in \Sigma \,:\, q_i \xrightarrow{a} q_j \right\} \qquad (i, j \in \{0, \ldots, n\})$$

kodiert, sowie die Sprache M_i, welche die Läufe im Finalzustand kodiert als

$$M_i := \begin{cases} \{\varepsilon\} & \text{, falls } q_i \in F \\ \emptyset & \text{, falls } q_i \notin F \end{cases} \qquad (i \in \{0,\dots,n\}).$$

Wir stellen nun die **Behauptung** auf, dass für alle $i \in \{0,\dots,n\}$ gilt, dass

$$L_i = \overbrace{\bigcup_{j=0}^{n} K_{i,j} L_j \cup M_i}^{=:N_i}.$$

Beweis der Behauptung: Zu zeigen ist, dass für alle $w \in \Sigma^*$ gilt, dass $w \in L_i \Leftrightarrow w \in N_i$:

Fall 1: $w = \varepsilon$

$$\begin{aligned} \varepsilon \in L_i &\Leftrightarrow q_i \in F && \text{(Def. } L_i\text{)} \\ &\Leftrightarrow \varepsilon \in M_i && \text{(Def. } M_i\text{)} \\ &\Leftrightarrow \varepsilon \in N_i && \text{(Def. } N_i\text{)} \\ &\quad \text{Beachte: } \varepsilon \notin K_{i,j} L_j \end{aligned}$$

Fall 2: $w \neq \varepsilon$, also $w = av$ mit $a \in \Sigma$ und $v \in \Sigma^*$

$$\begin{aligned} w \in L_i &\Leftrightarrow \exists\, q_f \in F.\ q_i \xrightarrow{w} q_F && \text{(Def. } L_i\text{)} \\ &\Leftrightarrow \exists q_F \in F. \exists j \in \{0,\dots,n\}.\ q_i \xrightarrow{a} q_j \xrightarrow{v} q_F \\ &\Leftrightarrow \exists j \in \{0,\dots,n\}.\ a \in K_{i,j} \wedge v \in L_j && \text{(Def. } K_{i,j}, L_j\text{)} \\ &\Leftrightarrow \exists j \in \{0,\dots,n\}.\ w \in K_{i,j} L_j \\ &\Leftrightarrow \exists j \in \{0,\dots,n\}.\ w \in N_i && \text{(Def. } N_i\text{)} \\ &\quad \text{Beachte: } \varepsilon \notin K_{i,j} L_j \end{aligned}$$

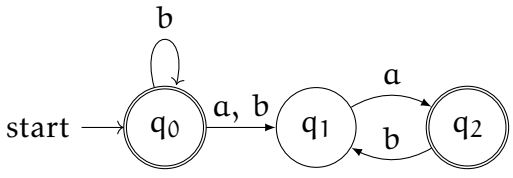
□

Iteriertes Einsetzen und Anwenden des Lemmas von Ardens liefert dann reguläre Ausdrücke $r_0,\dots,r_n$ für $L_0,\dots,L_n$. Wegen $(*)$ ist $r = \sum_{q_i \in I} r_i$, ein regulärer Ausdruck für $L(A) = L$.

□

Beispiel 1.4: Wir wollen einen regulären Ausdruck für nebenstehenden Automaten entwickeln:

$$L_0 = bL_0 + (a+b)L_1 + \emptyset L_2 + \varepsilon \qquad (0)$$
$$L_1 = \emptyset L_0 + \emptyset L_1 + aL_2 + \emptyset \qquad (1)$$
$$L_2 = \emptyset L_0 + bL_1 + \emptyset L_2 + \varepsilon \qquad (2)$$



Einsetzen von (2) in (1) liefert

$$L_1 = a(bL_1 + \varepsilon) = abL_1 + a \overset{\text{Arden}}{\Longrightarrow} L_1 = (ab)^* a. \qquad (3)$$

Einsetzen von (3) in (0) ergibt

$$L_0 = bL_0 + (a+b)(ab)^* a + \varepsilon.$$

Mit dem Lemma von Arden ergibt sich dann für L_0 (also als regulären Ausdruck für $L(A)$):

$$L(A) = L_0 = b^*[(a+b)(ab)^*a + \varepsilon]$$

※

Bemerkung

Ein regulärer Ausdruck kann als (nichtdeterministisches) **Programm** interpretiert werden:

- Der reguläre Ausdruck a, mit $a \in \Sigma$, ist ein **atomares Programm**.
- $\emptyset$ ist ein **fehlerhaftes Programm**.
- ε ist ein Programm, welches **nichts tut**.
- rs ist die **sequentielle Komposition** der Programme r und s.
- $r + s$ ist ein Programm, das (nichtdeterministisch) **entweder** r **oder** s ausführt.
- r^* ist ein Programm, das r **beliebig, aber endlich oft** iteriert.
- $L(r)$ st die **Menge der Ausführungssequenzen atomarer Programme**, die bei Ausführung von r auftreten können.
- Die Transformation von r in einen NEA entspricht der **Kompilierung in Maschinencode**.

Fazit: Die regulären Ausdrücke bilden eine höhere Programmiersprache für NEAs.

1.2 Morphismen von Automaten

Im Folgenden betrachten wir DAs über einem festen Alphabet Σ. Für die Daten eines DAs schreiben wir $A = (Q_A, \Sigma, \to_A, q_{0,A}, F_A)$.

Definition 1.6 (*Morphismus*)

Seien A und B DAs. Ein **(Homo-)Morphismus** von A nach B ist eine Funktion $h : Q_A \to Q_B$ mit:

(i) $h(q_{0,A}) = q_{0,B}$

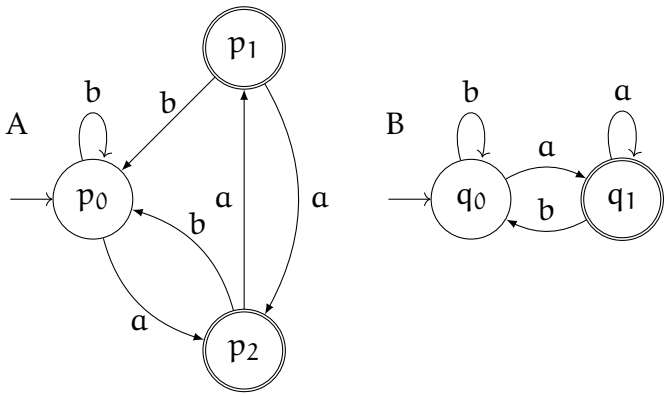
(ii) $\forall q \in Q_A.\ q \in F_A \Leftrightarrow h(q) \in F_B$

(iii) $\forall q \in Q_A.\ \forall a \in \Sigma.\ h(q.a) = h(q).a$

Notation: Schreibe $h : A \to B$ für einen Morphismus von A nach B.

Beispiel 1.5:

Wähle $h : A \to B$ mit $\left\{ \begin{array}{l} p_0 \mapsto q_0 \\ p_1 \mapsto q_1 \\ p_2 \mapsto q_1 \end{array} \right\}$. h ist ein Morphismus.



※

Satz 1.5

Sei $h : A \to B$ ein Morphismus von DAs $\Rightarrow L(A) = L(B)$.

Beweis: Sei $w = a_1 \dots a_n \in \Sigma^*$. Betrachte den Lauf von w in A:

$$q_{0,A} \xrightarrow{a_1} q_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} q_n.$$

Wegen Eigenschaft (iii) hat w folgenden Lauf in B:

$$q_{0,B} \overset{(i)}{=} h(q_{0,A}) \xrightarrow{a_1} h(q_1) \xrightarrow{a_2} \cdots \xrightarrow{a_n} h(q_n).$$

Daraus folgt $L(A) = L(B)$. □

Proposition 1.6

(1) Für jeden DA A ist $id_A : A \to A, q \mapsto q$ ein Morphismus.

(2) Wenn $g : A \to B$ und $h : B \to C$ Morphismen von DAs sind, dann ist auch $h \circ g : A \to C, q \mapsto h(g(q))$ ein Morphismus.

Beweis:

ad (1) trivial

ad (2) Seien $g : A \to B$ und $h : B \to C$ Morphismen. Prüfe (i), (ii) und (iii) für $h \circ g$:

ad (i) $h(g(q_{0,A})) = h(q_{0,B}) = q_{0,C}$

ad (ii) $q \in F_A \Leftrightarrow g(q) \in F_B \Leftrightarrow h(g(q)) \in F_C$

ad (iii) $h(g(q.a)) = h(g(q).a) = h(g(q)).a$

Wir schlussfolgern, dass $h \circ g$ ein Morphismus ist.

□

Morphismen können auch durch **Kongruenzen** beschrieben werden!

Definition 1.7 (*Kongruenz*)

Sei A ein DA. Eine **Kongruenz** auf A ist eine Äquivalenzrelation $\sim \subseteq Q_A \times Q_A$ mit

(i) $q \sim q' \Rightarrow q.a \sim q'.a \qquad \forall a \in \Sigma$

(ii) $q \sim q' \Rightarrow [q \in F \Leftrightarrow q' \in F]$

Proposition 1.7

Sei $h : A \to B$ ein Morphismus von DAs. Betrachte $\sim_h \subseteq Q_A \times Q_A$:

$$q \sim_h q' :\Leftrightarrow h(q) = h(q')$$

Dann ist $\sim_h$ eine Kongruenz auf A, der sogenannte **Kern von** h.

Beweis: Klar ist, dass $\sim_h$ eine Äquivalenzrelation ist, denn dies folgt bereits aus der Definition. Überprüfe nun (i) und (ii):

ad (i) Sei $q, q' \in Q_A, a \in \Sigma$, so gilt

$$\begin{aligned} q \sim_h q' &\Rightarrow h(q) = h(q') && \text{(Definition von } \sim_h\text{)} \\ &\Rightarrow h(q).a = h(q').a \\ &\Rightarrow h(q.a) = h(q'.a) && (\sim_h \text{ ist Morphismus}) \\ &\Rightarrow q.a \sim_h q'.a && \text{(Definition von } \sim_h\text{)} \end{aligned}$$

ad (ii) Sei $q \sim_h q'$, so gilt

$$\begin{aligned} q \in F_A &\Leftrightarrow h(q) \in F_B && (\sim_h \text{ ist Morphismus}) \\ &\Rightarrow h(q') \in F_B && (h(q) = h(q')) \\ &\Rightarrow q' \in F_A && (\sim_h \text{ ist Morphismus}) \end{aligned}$$

□

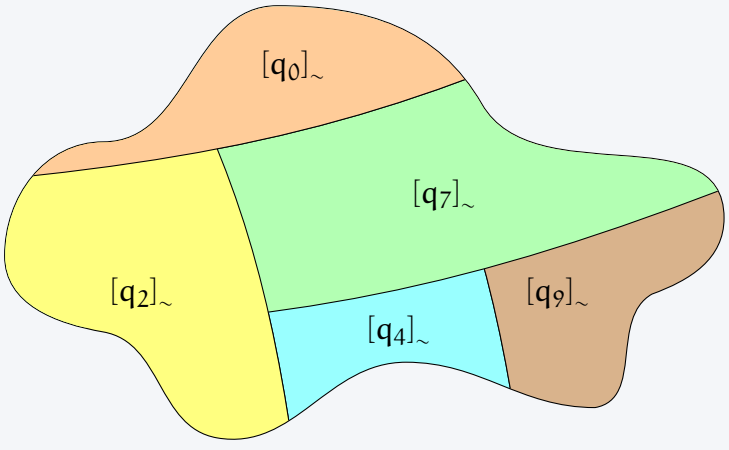
Für jede Kongruenz $\sim$ auf einem DA A kann man einen DA $A/\sim$ konstruieren, der durch Verschmelzen kongruenter Zustände entsteht.

Notation

Für $q \in Q_A$ sei

$$[q]_\sim = \left\{ q' \in Q_A \quad : \quad q \sim q' \right\} \qquad \text{(\textbf{Kongruenzklasse} von } q\text{)}$$

und $$Q_{A/\sim} = \left\{ [q]_\sim \quad : \quad q \in Q_A \right\} \qquad \text{(Menge \textbf{aller} Kongruenzklassen)}$$



Die Kongruenzklassen bilden eine Partition von Q_A

Definition 1.8

Sei A ein DA und $\sim$ eine Kongruenz auf A. Der **Quotientenautomat** $A/\sim$ ist wie folgt definiert:

- Zustandsmenge $Q_{A/\sim}$
- Alphabet Σ
- die Zustandsübergangsrelation $[q]_\sim.a = [q.a]_\sim$ für $q \in Q_A, a \in \Sigma$
  (das ist wohldefiniert, also unabhängig von der Wahl des Repräsentanten, durch Eigenschaft (i) einer Kongruenz)
- Initialzustand $[q_{0,A}]_\sim$ und
- Finalzustand: $[q]_\sim$ final in $A/\sim \Leftrightarrow q \in F_A$.
  (das ist wohldefiniert, also unabhängig von der Wahl des Repräsentanten, durch Eigenschaft (ii) einer Kongruenz)

Proposition 1.8

Sei A ein DA und $\sim$ Kongruenz auf A. Dann ist

$$h_\sim : A \to A/\sim,\ q \mapsto [q]_\sim$$

ein Morphismus mit Kern $\sim$. Insbesondere ist dann $L(A) = L(A/\sim)$.

Beweis: $h_\sim$ ist Morphismus:

ad (i) $h_\sim(q_{0,A}) = [q_{0,A}]_\sim$ und dieser Zustand ist initial in $A/\sim$.

ad (ii) $q \in F_A \Leftrightarrow \underbrace{[q]_\sim}_{=h_\sim(q)}$ final in $A/\sim$

ad (iii) Für $q \in Q_A$, $a \in \Sigma$. $h_\sim(q.a) = [q.a]_\sim = [q]_\sim.a = h_\sim(q).a$.

Daraus schließen wir direkt, dass $h_\sim$ ein Morphismus ist, zu zeigen bleibt, dass $\sim$ der Kern von $h_\sim$ ist.
Sei $\approx$ $(:=\sim_{h_\sim})$ der Kern von $h_\sim$. Dann gilt für $q, q' \in Q_A$:

$$q \approx q' \Leftrightarrow h_\sim(q) = h_\sim(q') \Leftrightarrow [q]_\sim = [q']_\sim \Leftrightarrow q \sim q'.$$

□

Satz 1.9 (*Homomorphiesatz*)

Seien $e : A \to B$, $h : A \to C$ Morphismen von DAs und e surjektiv. Dann gilt:

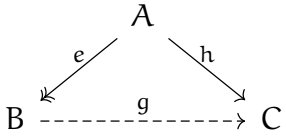
(1) Jede Funktion $g : Q_B \to Q_C$ mit $h = g \circ e$ ist ein Morphismus von B nach C.

(2) Folgende Aussagen sind äquivalent:

(a) Es gibt einen Morphismus $g : B \to C$ mit $h = g \circ e$

(b) $\sim_e \subseteq \sim_h$

(c) $\forall\, q, q' \in Q_A.\ e(q) = e(q') \Rightarrow h(q) = h(q')$



Beweis:

ad (1) Sei $g : Q_B \to Q_C$ mit $h = g \circ e$. Wir zeigen, dass g Morphismus ist:

ad (i)

$$\begin{aligned} g(q_{0,B}) &= g(e(q_{0,A})) && (\,e \text{ ist Morphismus})\\ &= h(q_{0,A}) && (g \circ e = h)\\ &= q_{0,C} && (\,\sim_h \text{ ist Morphismus}) \end{aligned}$$

ad (ii) Sei $q \in Q_B$. Wähle $p \in Q_A$ mit $e(p) = q$. (e ist surjektiv!)

$$\begin{aligned} q \in F_B &\Leftrightarrow e(p) \in F_B\\ &\Leftrightarrow p \in F_A && (\,e \text{ ist Morphismus})\\ &\Leftrightarrow h(p) \in F_C && (\,h \text{ ist Morphismus})\\ &\Leftrightarrow g(e(p)) \in F_C && (h = g \circ e)\\ &\Leftrightarrow g(q) \in F_C \end{aligned}$$

ad (iii) Sei $q \in Q_B$, $a \in \Sigma$. Wähle $p \in Q_A$ mit $e(p) = q$. (e ist surjektiv!)

$$\begin{aligned} g(q.a) &= g(e(p).a)\\ &= g(e(p.a)) && (\,e \text{ ist Morphismus})\\ &= h(p.a) && (g \circ e = h)\\ &= h(p).a && (\,h \text{ ist Morphismus})\\ &= g(e(p)).a && (h = g \circ e)\\ &= g(q).a \end{aligned}$$

ad (2) Es ist klar, dass (b) $\Leftrightarrow$ (c) nach Definition von $\sim_e$ und $\sim_h$. Wir zeigen damit noch (a) $\Leftrightarrow$ (c):

„(a) $\Rightarrow$ (c)“ Sei $g : B \to C$ ein Morphismus mit $h = g \circ e$, $q, q' \in Q_A$, so gilt

$$e(q) = e(q') \quad \Rightarrow \quad g(e(q)) = g(e(q')) \quad \Leftrightarrow \quad h(q) = h(q').$$

„(a) $\Leftarrow$ (c)“ Definiere $g : Q_B \to Q_C$ wie folgt: Sei $q \in Q_B$. Wähle $p \in Q_A$ mit $e(p) = q$ (e ist surjektiv) und definiere $g(q) := h(p)$. Das ist wegen (c) wohldefiniert, also *unabhängig* von der Wahl von p. Außerdem gilt $h = g \circ e$, weil $g(\underbrace{e(p)}_{=q}) = h(p)$. Nach (1) ist g ein Morphismus.

□

Isomorphismen *Idee:* Zwei DAs sind isomorph, wenn sie sich nur in der Benennung der Zustände unterscheiden.

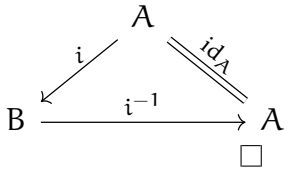
Definition 1.9 (*Isomorphismus*)

Ein Morphismus $i : A \to B$ heißt **Isomorphismus** genau dann, wenn er bijektiv ist.
A und B sind **isomorph**, wenn ein Isomorphismus $i : A \to B$ existiert. Schreibe dann: $A \cong B$.

Proposition 1.10

Wenn $i : A \to B$ ein Isomorphismus ist, dann auch die Umkehrfunktion $i^{-1} : B \to A$.

Beweis:
$i^{-1} \circ i = id_A$
Weil i, id_A Morphismen sind, ist auch i^{-1} ein Morphismus nach Satz 1.9 (Homomorphiesatz).

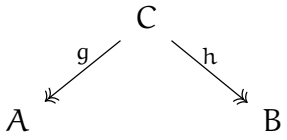


□

Proposition 1.11

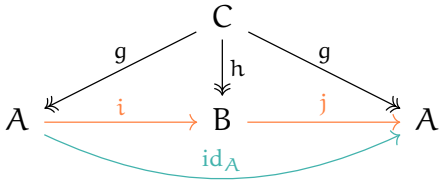
Seien $g : C \to A$ und $h : C \to B$ surjektive Morphismen.

$$\sim_g = \sim_h \quad \Rightarrow \quad A \cong B$$



Beweis:
Angenommen $\sim_g = \sim_h$.

$$\sim_g \subseteq \sim_h \xRightarrow{\text{Satz 1.9}} \exists\, i : A \to B.\ g \circ i = h$$
$$\sim_g \supseteq \sim_h \xRightarrow{\text{Satz 1.9}} \exists\, j : B \to A.\ h \circ j = g$$



Behauptung: $i \circ j = id_B$ und $j \circ i = id_A$
Beweis der Behauptung: Sei $q \in Q_A$. Wähle $p \in Q_C$ mit $g(p) = q$; dies ist erlaubt, da g surjektiv ist.

$$\rightsquigarrow \qquad j(i(q)) = j(i(g(p))) = j(h(p)) = g(p) = q \Rightarrow j \circ i = id_A\,.$$

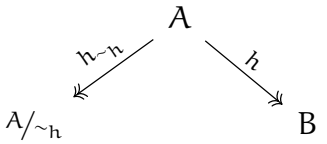
$i \circ j = id_B$ verläuft analog!
$\rightsquigarrow$ i ist Isomorphismus mit $i^{-1} = j$. □

Korrolar 1.11.1

Sei $h : A \twoheadrightarrow B$ surjektiver Morphismus von DAs. Dann gilt $B \cong A/\sim_h$.

Beweis:

Wende Proposition 1.11 auf die rechts dargestellten Morphismen an.
Beide Morphismen haben den Kern $\sim_h$.



□

1.3 Kanonische Automaten

Für jede reguläre Sprache $L \subseteq \Sigma^*$ gibt es unendlich viele DAen, die L akzeptieren. Drei davon sind durch universelle Eigenschaften charakterisiert:

- der **Initialautomat** Init(L)
- der **Finalautomat** Fin(L)
- der **Minimalautomat** Min(L)

Definition 1.10 (*Initialautomat*)

Sei $L \subseteq \Sigma^*$ eine beliebige Sprache. Der **Initialautomat** Init(L) ist der deterministische Automat mit

- Zustandsmenge Σ^*
- Alphabet Σ
- Übergänge $w.a = wa$ für $w \in \Sigma^*, a \in \Sigma$
- Initialzustand ε und
- Finalzustandsmenge L.

Satz 1.12

Sei $L \subseteq \Sigma^*$ eine beliebige Sprache, so gelten die folgenden Aussagen:

(1) $\mathsf{Init}(L)$ akzeptiert L

(2) Für jeden deterministischen Automaten A mit $L(A) = L$ gibt es genau einen Morphismus von $\mathsf{Init}(L)$ nach A, nämlich

$$i_A : \mathsf{Init}(L) \to A,\ i_A(w) = q_{0,A}.w.$$

Beweis: siehe Übungsblatt 2 Aufgabe 1. □

Definition 1.11 (*Finalautomat*)

Sei $L \subseteq \Sigma^*$ eine beliebige Sprache. Der **Finalautomat** $\mathsf{Fin}(L)$ ist der deterministische Automat mit

- Zustandsmenge $\mathfrak{P}(\Sigma^*)$ (Menge aller Sprachen über Σ)
- Alphabet Σ
- Übergänge $K \xrightarrow{a} a^{-1}K$ mit $K \in \mathfrak{P}(\Sigma^*)$, $a \in \Sigma$ und $a^{-1}K = \{v \in \Sigma^* \mid av \in K\}$ (**Links**ableitung von K bezüglich a)
- Initialzustand L und
- Finalzustände sind alle $K \in \mathfrak{P}(\Sigma^*)$ mit $\varepsilon \in K$.

Satz 1.13

Sei $L \subseteq \Sigma^*$ eine beliebige Sprache, so gelten die folgenden Aussagen:

(1) $\mathsf{Fin}(L)$ akzeptiert L

(2) Für jeden deterministischen Automaten A mit $L(A) = L$ gibt es genau einen Morphismus von A nach $\mathsf{Fin}(L)$, nämlich

$$f_A : A \to \mathsf{Fin}(L),\ f_A(q) = \left\{w \in \Sigma^* \;\middle|\; q.w \in F_A\right\}.$$

Insbesondere ist $f_A(q_{0,A}) = L(A)$.

Beweis: siehe Übungsblatt 2 Aufgabe 2. □

1.4 Minimierung von Automaten

Definition 1.12

Ein DEA A ist **minimal**, wenn es keinen DEA B gibt mit

$$L(B) = L(A) \quad \text{und} \quad |Q_B| < |Q_A|.$$

Wir wollen zeigen:

- Für jede reguläre Sprache $L \subseteq \Sigma^*$ gibt es einen – bis auf Isomorphie – eindeutigen Minimalautomaten $\mathsf{Min}(L)$, der L akzeptiert.
- $\mathsf{Min}(L)$ lässt sich aus jedem deterministischen Automaten A für L effektiv konstruieren.

Idee: Entferne zuerst die nicht erreichbaren Zustände und verschmelze anschließend verhaltensäquivalente Zustände.

Definition 1.13

Sei A ein deterministischer Automat. Dann heißt ein Zustand $q \in Q_A$ **erreichbar**, wenn es ein $w \in \Sigma^*$ gibt mit $q_{0,A}.w = q$.
Der Automat A heißt **erreichbar**, wenn alle Zustände von A erreichbar sind. Dies ist äquivalent zur Forderung, dass der kanonische Initialmorphismus i_A **surjektiv** ist.
Wir schreiben $\mathsf{reach}(A)$ für den erreichbaren Teil von A, der durch Entfernen aller nicht erreichbaren Zustände aus A entsteht.

Definition 1.14

Sei A ein deterministischer Automat. Dann heißen zwei Zustände $q, q' \in Q_A$ **verhaltensäquivalent** ($q \sim_A q'$), wenn

$$\forall\, w \in \Sigma^*.\ q.w \in F_A \Leftrightarrow q'.w \in F_A$$

A ist **einfach**, wenn für alle $q, q' \in Q_A$ gilt, dass $q \sim_A q' \Rightarrow q = q'$. Dies ist äquivalent zu der Forderung, dass der kanonische Finalmorphismus f_A **injektiv** ist.

Beachte

$\sim_A$ ist der Kern von f_A und damit eine Kongruenz auf A. Also können wir den Quotientenautomaten $A/\sim_A$ bilden. Seine Zustände sind genau die Kongruenzklassen $[q]_{\sim_A} = \{q' \in A \mid q \sim_A q'\}$.

Satz 1.14

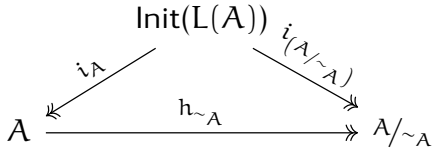
Sei A ein erreichbarer deterministischer Automat, so gilt:

(1) $A/\sim_A$ ist erreichbar und einfach.

(2) Falls $L(A)$ regulär ist (insbesondere also wenn A ein deterministischer endlicher Automat ist), ist $A/\sim_A$ minimal ist.

Beweis:

ad (1) Wir zeigen, dass $A/_{\sim_A}$ erreichbar und einfach ist. Überprüfen wir zuerst die Erreichbarkeit des Quotientenautomatens:
Aufgrund der Initialität von $\mathsf{Init}(\mathsf{L}(A))$ ist $i_{(A/_{\sim_A})} = h_{\sim_A} \circ i_A$. Weil A zudem noch erreichbar ist, ist i_A surjektiv. Die Surjektivität von $i_{(A/_{\sim_A})}$ und damit die Erreichbarkeit von $A/_{\sim_A}$ folgt aus der Surjektivität von $h_{\sim_A} : A \to A/_{\sim_A},\ q \mapsto [q]_{\sim_A}$ und aus der Tatsache, dass die Verkettung zweier surjektiver Abbildungen wieder surjektiv ist.



Nun müssen wir noch die Einfachheit des Quotientenautomatens nachweisen. Nach Definition von $A/_{\sim_A}$ gilt

$$[q]_{\sim_A}.a = [q.a]_{\sim_A} \quad \text{für } q \in Q_A, a \in \Sigma,$$

oder allgemeiner

$$[q]_{\sim_A}.w = [q.w]_{\sim_A} \quad \text{für } q \in Q_A, w \in \Sigma^*.$$

Sei $[q]_{\sim_A}$ und $[q']_{\sim_A}$ verhaltensäquivalente Zustände in $A/_{\sim_A}$. Wir müssen nun noch zeigen, dass die Kongruenzklassen übereinstimmen, dass also $[q]_{\sim_A} = [q']_{\sim_A}$ beziehungsweise $q \sim_A q'$ gilt. Wir stellen fest, dass für alle $w \in \Sigma^*$ gilt, dass

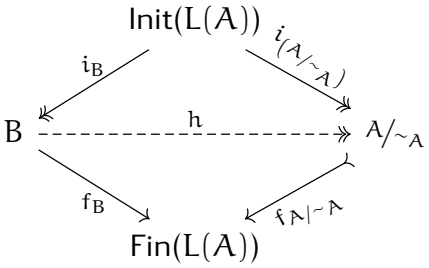
$$\begin{aligned} q.w \in F_A &\Leftrightarrow [q.w]_{\sim_A} \text{ final in } A/_{\sim_A} \\ &\Leftrightarrow [q]_{\sim_A}.w \text{ final in } A/_{\sim_A} \\ &\Leftrightarrow [q']_{\sim_A}.w \text{ final in } A/_{\sim_A} \\ &\Leftrightarrow [q'.w]_{\sim_A} \text{ final in } A/_{\sim_A} \\ &\Leftrightarrow q'.w \in F_A \end{aligned}$$

Damit folgt unmittelbar, dass $q \sim_A q'$.

ad (2) Sei B ein DEA mit $L(B) = L(A)$. Zu zeigen ist nun, dass $|Q_B| \geqslant \left|Q_{A/_{\sim_A}}\right|$ gilt. Œ dürfen wir annehmen, dass B erreichbar ist, sonst ersetze B durch $\mathsf{reach}(B)$. Wir zeigen die folgende Behauptung:
Behauptung: Es gibt einen surjektiven Morphismus $h : B \twoheadrightarrow A/_{\sim_A}$.
Beweis der Behauptung:

Nach dem Homomorphiesatz (Satz 1.9) genügt es zu zeigen

$$\forall\, u, w \in \Sigma^*.\ i_B(u) = i_B(w) \ \Rightarrow\ i_{A/_{\sim_A}}(u) = i_{A/_{\sim_A}}(w).$$



Sei also $i_B(u) = i_B(w)$. Dann folgt

$$f_B \circ i_B(u) = f_B \circ i_B(u).$$

Wegen der Initialität von $\mathsf{Init}(\mathsf{L}(A))$ ist $f_B \circ i_B = f_{A/_{\sim_A}} \circ i_{A/_{\sim_A}}$. Also gilt

$$f_{A/_{\sim_A}} \circ i_{A/_{\sim_A}}(u) = f_{A/_{\sim_A}} \circ i_{A/_{\sim_A}}(w).$$

Weil $A/_{\sim_A}$ einfach nach Aussage (1) ist, ist $f_{A/_{\sim_A}}$ injektiv. Deswegen gilt

$$i_{A/_{\sim_A}}(u) = i_{A/_{\sim_A}}(w).$$

Der Homomorphiesatz (Satz 1.9) liefert nun ein $h : B \to A/_{\sim_A}$ mit $i_{A/_{\sim_A}} = h \circ i_B$. Weil $i_{A/_{\sim_A}}$ surjektiv ist, was aus der Erreichbarkeit des Quotientenautomatens folgt, ist h surjektiv. □

Korrolar 1.14.1

Für jeden deterministischen endlichen Automaten A gilt:

$$A \text{ minimal} \Leftrightarrow A \text{ erreichbar und einfach}$$

Beweis:

„$\Rightarrow$“ Sei A minimal, dann ist A erreichbar, da sonst $\text{reach}(A)$ ein kleinerer DEA für $L(A)$ wäre, und einfach, da sonst $A/\sim_A$ ein kleinerer DEA für $L(A)$ wäre.

„$\Leftarrow$“ Sei A erreichbar und einfach, so folgt aus ersterem, dass $A/\sim_A$ minimal ist, und aus letzterem, dass A isomorph zu seinem Quotientenautomaten ist ($A \cong A/\sim_A$). Nimmt man beide Aussagen und kombiniert diese, so erhält man, dass A minimal ist.

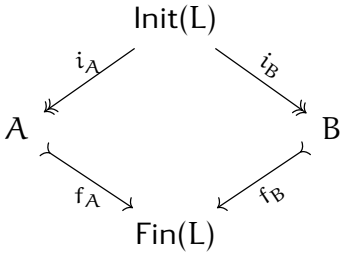
□

Korrolar 1.14.2

Für jede beliebige reguläre Sprache $L \subseteq \Sigma^*$ gibt es bis auf Isomorphie genau einen minimalen DEA, der L akzeptiert.

Beweis: Seien A und B minimale DEAs, welche beide die Sprache L akzeptieren ($L(A) = L(B) = L$).

$\not\exists$: $A \cong B$:
Weil A und B minimal – also erreichbar und einfach – sind, sind i_A und i_B sur- und f_A und f_B injektiv.
Nach Proposition 1.11 genügt es zu zeigen, dass $\sim_{i_A} = \sim_{i_B}$.
Für alle $v, w \in \Sigma^*$ gilt nun, dass



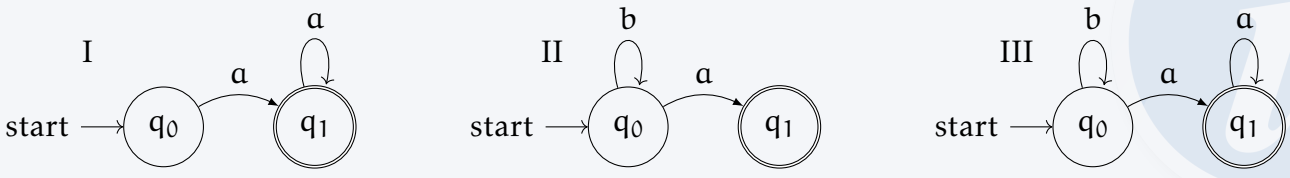
$$\begin{aligned}
v \sim_{i_A} w &\Leftrightarrow i_A(v) = i_A(w) && \text{(Def. von } \sim_{i_A}\text{)} \\
&\Leftrightarrow f_A \circ i_A(v) = f_A \circ i_A(w) && (f_A \text{ ist injektiv}) \\
&\Leftrightarrow f_B \circ i_B(v) = f_B \circ i_B(w) && \text{(Initialität von Init(L))} \\
&\Leftrightarrow i_B(v) = i_B(w) && (f_B \text{ ist inijektiv}) \\
&\Leftrightarrow v \sim_{i_B} w && \text{(Def. von } \sim_{i_B}\text{)}
\end{aligned}$$

Daraus folgt direkt, dass $\sim_{i_A} = \sim_{i_B}$. □

Notation

Schreibe Min(L) für den minimalen DEA der regulären Sprache L.

Eine reguläre Sprache hat im Allgemeinen **keinen** *eindeutigen* minimalen NEA!
Die Sprache a^+ hat beispielsweise **drei** minimale NEAs:



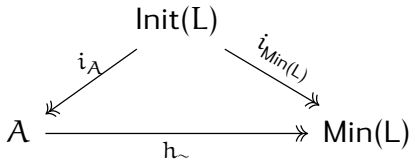
Dieser Umstand macht die Minimierung von NEAs $\mathcal{NP} \subseteq$ PSPACE-vollständig! Insbesondere also $\mathcal{NP}$-schwer.

Korrolar 1.14.3

Sei $L \subseteq \Sigma^*$ regulär. Für jeden erreichbaren DEA A mit $L(A) = L$ gibt es **genau einen** Morphismus von A nach $\mathsf{Min}(L)$.

Beweis:

Die Existenz eines solchen Morphismus folgt durch Angabe der kanonischen Surjektion $h_\sim : A \twoheadrightarrow {}^{A}/_{\sim_A} = \mathsf{Min}(L), q \mapsto [q]_{\sim_A}$.
Die Eindeutigkeit folgt direkt aus der $i_{\mathsf{Min}(L)} = h_\sim \circ i_A$ und der Surjektivität von i_A.



□

Definition 1.15 (*Nerode-Äquivalenz*)

Sei $L \subseteq \Sigma^*$ eine Sprache. Die **Nerode-Äquivalenz** von L ist die Äquivalenzrelation $\sim_L \subseteq \Sigma^* \times \Sigma^*$ mit

$$v \sim_L w \quad :\Leftrightarrow \quad (\forall x \in \Sigma^*.\, vx \in L \Leftrightarrow wx \in L)$$

Beachte

$\sim_L$ ist die Verhaltensäquivalenz des Initialautomaten $\mathsf{Init}(L) = (\Sigma^*, \Sigma, \to, \varepsilon, L)$, wobei $w \xrightarrow{x} wx$. Also können wir den Quotientenautomaten ${}^{\mathsf{Init}(L)}/_{\sim_L}$ bilden, mit Zustandsmenge ${}^{\Sigma^*}/_{\sim_L} = \{[w]_{\sim_L} : w \in \Sigma^*\}$.

Korrolar D1.15 (*Myhill-Nerode*)

Eine Sprache $L \subseteq \Sigma^*$ ist genau dann regulär, wenn ${}^{\Sigma^*}/_{\sim_L}$ endlich ist.

Beweis:

„⇒“ L regulär $\xRightarrow{\text{Satz 1.14}}$ ${}^{\mathsf{Init}(L)}/_{\sim_L}$ ist minimaler DEA für L mit Zustandsmenge ${}^{\Sigma^*}/_{\sim_L} \Rightarrow {}^{\Sigma^*}/_{\sim_L}$ ist also insbesondere **endlich**!

„⇐“ ${}^{\Sigma^*}/_{\sim_L}$ ist endlich $\Rightarrow$ ${}^{\mathsf{Init}(L)}/_{\sim_L}$ ist ein DEA für L, woraus die Regularität von L folgt.

□

Beispiel 1.6: $L = \{a^n b^n \mid n \geqslant 0\}$ ist **nicht regulär**.
Zeige, dass $\sim_L$ ∞ viele Äquivalenzklassen.
Für $i \neq j$ folgt, dass $a^i \not\sim_L a^j$, denn $a^i b^i \in L$, aber $a^j b^i \notin L$. Damit folgt direkt, dass es unendlich viele Äquivalenzklassen gibt, da die Äquivalenzklassen $[a^i]_{\sim_L}$ für alle $i \in \mathbb{N}_0$ paarweise verschieden sind. ※

B

Wir zeigen hier, dass das Pumpinglemma, welches vielleicht aus Grundlagenveranstaltungen bekannt ist, zu keinen schöneren und keinenfalls leichteren Beweisen führt. Dies geschieht mit **Beispiel 1.7:** Dazu wollen wir zeigen, dass die Sprache

$$L = \{a^i b^j \mid i \neq j\}$$

nicht regulär ist.
Für das Pumpinglemma zeigen wir nun, dass wir für jede beliebige Pumpingkonstante n

B ein Wort z finden können, so dass das Wort zwar in drei Teile uvw mit $|uv| \leqslant n$ und $|v| \geqslant 1$ aufteilen können, aber das „gepumpte“ Wort uv^iw für ein i nicht mehr in der Sprache liegt: Angenommen die Sprache L sei nun regulär, so sei mit n die Pumpingkonstante der Sprache L gegeben. Wir wählen dann das Wort $z = a^n b^{n+n!}$ mit $|z| \geqslant n$, womit wir eine Aufteilung finden können mit $z = uvw$, wobei gilt, dass $|uv| \leqslant n$, $|v| \geqslant 1$ und für alle $i \in \mathbb{N}_0$ gilt, dass $uv^iw \in L$.
Da $|uv| \leqslant n$ gilt, können wir u und v schreiben als $u = a^t$, $v = a^s$ und damit $w = a^{n-t-s}b^{n+n!}$ mit $1 \leqslant s \leqslant n$, da $|v| \geqslant 1$. Also wissen wir für alle $i \in \mathbb{N}_0$, dass $uv^iw = a^t a^{is} a^{n-t-s} b^{n+n!} = a^{n+s(i-1)} b^{n+n!} \in L$. Nehme nun den Index $i_0 = \frac{n!}{s} + 1$. Wir wissen, dass i_0 auf jeden Fall natürlich sein muss, da $1 \leqslant s \leqslant n$ ein Teiler von $n!$ ist. Also gilt per Pumpinglemma, dass

$$uv^{i_0}w = a^{n+s((\frac{n!}{s}+1)-1)} b^{n+n!} = a^{n+n!} b^{n+n!} \in L,$$

was direkt zu einem Widerspruch führt. ※

Minimierungsalgorithmen

Wir betrachten das Berechnungsproblem der **DEA-Minimierung**. Wir erhalten als **Eingabe** einen zu minimierenden DEA A und wollen den Minimalautomaten $\mathsf{Min}(L(A))$ ausgeben.

Algorithmus 1.1 (*Standardverfahren*)

Eingabe: DEA A

Schritt 1 Entferne nicht erreichbare Zustände von A. In anderen Worten: Ersetze A durch $\mathsf{reach}(A)$.

Schritt 2 Berechne $\sim_A$

Schritt 3 RETURN $A/\sim_A$

Meistens implementiert man Schritt 1 durch eine Breitensuche und Schritt 2 durch das sogenannte *partition refinement*, sprich die Tabelle, in der man die gefundene Partition immer weiter verfeinert (bekannt aus Grundlagenveranstaltung ThProg).
Wir wollen uns jetzt einen alternativen Algorithmus anschauen, welcher nur auf zwei Operationen auf NEAen basiert. Dazu definieren wir:

Definition 1.16

Sei B ein NEA mit $B = (Q, \Sigma, \to, I, F)$.
Sei dann definiert:

1. $\mathsf{rdet}(B) = \mathsf{reach}(B_{\mathsf{det}})$ als der **erreichbare Teil des Potenzmengenautomaten** von B und
2. $\mathsf{rev}(B)$ als der **Spiegelautomat von** B, welcher durch das Umdrehen aller Übergänge und dem Vertauschen der Initial- und Finalzustände entsteht. Also
$\mathsf{rev}(B) = (Q, \Sigma, \to_{\mathsf{rev}}, F, I)$ mit $\forall\, q, q' \in Q.\, \forall\, a \in \Sigma.\, q \xrightarrow{a}_{\mathsf{rev}} q'$ in $\mathsf{rev}(B) :\Leftrightarrow\ q' \xrightarrow{a} q$ in B.

i Beachte

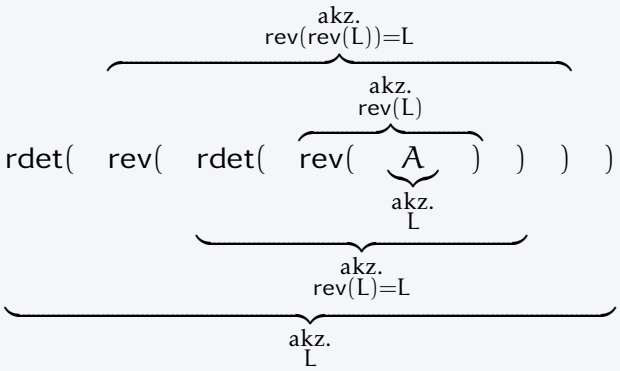
Akzeptiert ein NEA B die Sprache L, so akzeptiert $\mathsf{rev}(B)$ die Sprache $\mathsf{rev}(L) = \{a_1 \dots a_n \in \Sigma^* \mid a_n \dots a_1 \in L\}$.

Algorithmus 1.2 (*Algorithmus von Brzozowski*)

Eingabe: DEA A

Schritt 1 **Return** $\mathrm{rdet}(\mathrm{rev}(\mathrm{rdet}(\mathrm{rev}(A))))$

Beachte



Beweis: Die Korrektheit basiert auf zwei Konzepten:

Definition 1.17

Sei $B = (Q, \Sigma, \to, I, F)$ ein NEA.

(1) B ist **co-erreichbar**, wenn

$$\forall\, q \in Q.\ \exists\, q_F \in F.\ \exists\, w \in \Sigma^*.\ q \xrightarrow{w} q_F.$$

Mit anderen Worten genau dann, wenn $\mathrm{rev}(B)$ erreichbar ist.

(2) B ist **co-deterministisch**, wenn

(a) B hat **genau einen** Finalzustand q_F

(b) Für alle $q \in Q$ und $a \in \Sigma$ gibt es genau ein $p \in Q$ mit $p \xrightarrow{a} q$.

Mit anderen Worten genau dann, wenn $\mathrm{rev}(B)$ deterministisch ist.

Beachte

Für jeden DEA A ist $B = \mathrm{rev}(\mathrm{rdet}(\mathrm{rev}(A)))$ co-erreichbar und co-deterministisch, denn $\mathrm{rdet}(\mathrm{rev}(A))$ ist erreichbar und determinstisch.

Damit folgt die Korrektheit des Algorithmus direkt aus

Lemma 1.15

Sei B ein co-erreichbarer und co-deterministischer NEA, so ist $\mathrm{rdet}(B)$ minimal.

Beweis: Wir zeigen, dass $\mathrm{rdet}(B)$ erreichbar und einfach ist. Die Erreichbarkeit folgt direkt aus der Definition von rdet, wir müssen damit nur noch die Einfachheit zeigen.
Sei nun also $B = (Q, \Sigma, \to, I, F = \{q_F\})$ und $S, T \in \mathfrak{P}(Q)$ zwei beliebige verhaltensäquivalente Zustände von $\mathrm{rdet}(B)$. Um die Einfachheit zu inferrieren, müssen wir $S = T$ zeigen, Œ reicht aufgrund der hier vorliegenden Symmetrie allerdings $S \subseteq T$ aus.
Sei nun $q \in S$. Weil B co-erreichbar ist, gibt es ein $w \in \Sigma^*$ mit $q \xrightarrow{w} q_F$. Dies impliziert $q \in S.w$. □

Aus dem Lemma und der Bemerkung oben folgt unmittelbar die Korrektheit des Verfahrens. □

Beachte

Der Algorithmus von Brzozowski (Algorithmus 1.2) bleibt auch dann korrekt, wenn A ein NFA ist.

1.5 Algorithmisches Lernen regulärer Sprachen

Wir betrachten folgendes Szenario:

- Der Lehrer denkt sich eine reguläre Sprache $L \subseteq \Sigma^*$ aus, wobei Σ bekannt ist.
- Der Schüler (S) möchte L lernen, indem er einen DEA für L findet. Dazu kann S zwei Arten von Fragen an den Lehrer stellen:

  I Gegeben sei das Wort $w \in \Sigma^*$, ist $w \in L$?

  II Gegeben sei ein DEA A, ist $L(A) = L$? Falls **ja**, hat S die Sprache L erfolgreich gelernt, falls **nein**, gibt der Lehrer ein *Gegenbeispiel*, also ein Wort $w \in L(A) \oplus L$[1] zurück.

Brute-Force — Die einfachste Lernstrategie für S

S zählt alle DEAs A in aufsteigender Größe auf und fragt jeweils, ob $L(A) = L$ ist.
Falls $n = |\mathsf{Min}(L)|$ ist, benötigt S im **schlimmsten Fall** $\Omega(n^n)$ Fragen, bis ein korrekter DEA gefunden ist, da n^n ungefähr der Anzahl DEAs mit n Zuständen entspricht.

! Keine intelligente Strategie. Sie mag zwar in endlicher Zeit vorbei sein, ist allerdings suboptimal gegenüber der anderen Strategie, welche wir jetzt kennenlernen werden.

Intelligente Fragestrategie

Wir suchen nun eine intelligentere Strategie, welche nur mit polynomiell vielen Fragen bezüglich n auskommt. Dazu soll S zu jedem Zeitpunkt ein Paar (Q, T) betrachten, wobei $Q, T \subseteq \Sigma^*$ endliche Mengen von Wörtern sind.
Idee: Q ist die Zustandsmenge eines DEAs, welcher $\mathsf{Min}(L)$ von unten approximiert. Die Wörter aus Q sind paarweise **nicht** Nerode-äquivalent und die Wörter aus T bilden Zeugen für ebendiese Nichtäquivalenz. Am Anfang starten wir mit $(Q, T) = (\{\varepsilon\}, \{\varepsilon\})$.

Definition 1.18 (T-*Äquivalenz*)

Für $T \subseteq \Sigma^*$ definieren wir die folgende Äquivalenzrelation auf Σ^*:

$$v \equiv_T w \Leftrightarrow (\forall x \in T.\ vx \in L \Leftrightarrow wx \in L)$$

[1] $A \oplus B := (A \setminus B) \cup (B \setminus A)$

Beachte

(1) Für $T \in \Sigma^*$ ist $\equiv_T = \sim_L$.

(2) Für $T \subseteq \Sigma^*$ ist $\sim_L \subseteq \equiv_T$.

(3) Falls T endlich ist, kann S mit maximal $2|T|$ Fragen vom Typ I entscheiden, ob $v \equiv_T w$ ist.

Definition 1.19

Sei $Q, T \subseteq \Sigma^*$. Das Paar (Q, T) ist ...

... **korrekt**, wenn für alle $q, q' \in Q$ gilt, dass $q \equiv_T q' \Rightarrow q = q'$.

... **vollständig**, wenn für ale $q \in Q$, $a \in \Sigma$ ein $q' \in Q$ existiert mit $qa \equiv_T q'$. (a *ist bis auf* T*-Äquivalenz abgeschlossen unter Transition*)

Jedes korrekte und vollständige Paar (Q, T) induziert einen DEA H („Hypothese“) mit

- Zustandsmenge Q
- Alphabet Σ
- Übergänge $q \xrightarrow{a} q'$, wobei q' der eindeutige Zustand mit $qa \equiv_T q'$ ist. (Wegen der Vollständigkeit existiert dieses q', wegen der Korrektheit von (Q, T) ist dieses eindeutig)
- Initialzustand ε und
- Finalzustände $Q \cap L$.

S kann H mit $\mathcal{O}\left(|Q|^2 \cdot |T| \cdot |\Sigma|\right)$ Fragen vom Typ I konstruieren.

Lemma 1.16

Wenn (Q, T) korrekt ist, gilt $|Q| \leqslant |\mathsf{Min}(L)|$.

Beweis:

$$\begin{aligned} |Q| &\leqslant |\Sigma^*/\equiv_T| && ((Q,T) \text{ ist korrekt}) \\ &\Leftrightarrow |\Sigma^*/\sim_L| = |\mathsf{Min}(L)| && (\sim_L \subseteq \equiv_T). \end{aligned}$$

□

Lemma 1.17

Sei (Q, T) korrekt, aber **nicht** vollständig. Dann existiert ein $q \in \Sigma^* \setminus Q$, so dass $\left(Q \cup \{q\}, T\right)$ korrekt ist.
Solch ein q kann S mit $\mathcal{O}\left(|Q|^2 \cdot |T| \cdot |\Sigma|\right)$ Fragen vom Typ I finden.

Beweis: Sei (Q, T) nicht vollständig. Dann gibt es $p \in Q$ und $a \in \Sigma$ mit $pa \not\equiv_T p'$ für alle $p' \in Q$. Dann ist $\left(Q \cup \{qa\}, T\right)$ korrekt. Das Finden von p und a erfordert – über das Iterieren der Tripel (p, a, p') – $\mathcal{O}\left(|Q|^2 \cdot |T| \cdot |\Sigma|\right)$ Fragen. □

Lemma 1.18

Sei (Q, T) korrekt **und** vollständig. Sei H der Hypothesen-DEA und $w \in \Sigma^*$ ein Gegenbeispiel

($w \in L(H) \oplus L$). Dann gibt es $q \in \Sigma^* \setminus Q$ und $t \in \Sigma^*$, so dass $\big(Q \cup \{q\}, T \cup \{t\}\big)$ korrekt ist. So ein q und t kann S mit $\mathcal{O}(\log|w|)$ Fragen finden.

Beweis: Sei $w = a_1 \dots a_m$. Betrachte Lauf von H mit Eingabe w.

$$\varepsilon = q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \dots \xrightarrow{a_{m-1}} q_{m-1} \xrightarrow{a_m} q_m.$$

Wir nennen nun den Zustand q_i **korrekt**, wenn

$$q_i a_{i+1} \dots a_m \in L \Leftrightarrow w \in L.$$

Beachte

q_0 ist **korrekt**, da $q_0 a_1 \dots a_m = w$.

q_m ist **nicht** korrekt. Dies folgt der Tatsache, dass w ein Gegenbeispiel ist. Andernfalls wäre $q_m \in L \Leftrightarrow w \in L$. Daraus folgt

$$\begin{aligned} w \in L(H) &\Leftrightarrow q_m \text{ final in } H \\ &\Leftrightarrow q_m \in L && \text{(Definition H)} \\ &\Leftrightarrow w \in L. \end{aligned}$$

Dies steht aber im Widerspruch zur Annahme zu $w \in L(H) \oplus L$.
Es gibt also $i \in \{1, \dots, m\}$, für das q_{i-1} korrekt und q_i nicht korrekt ist. So ein i kann der Schüler mit $\mathcal{O}(\log|w|)$ Fragen finden (Binäre Suche!)

Definiere dann $Q' := Q \cup \{q_{i-1}a_i\}$ und $T' = T \cup \{a_{i+1} \dots a_m\}$ und zeige, dass (Q', T') korrekt und $q_{i-1}a_i \notin Q$:
Nach Definition vopn H ist q_i der einzige Zustand aus Q mit $q_i \equiv_T q_{i-1}a_i$. Wir zeigen, dass $q \not\equiv_{T'} q_{i-1}a_i$ für alle $q \in Q$:

- Für $q \neq q_i$ ist $q \not\equiv_T q_{i-1}a_i$, insbesondere also auch $q \not\equiv_{T'} q_{i-1}a_i$
- Für $q = q_i$ ist

$$(q_{i-1}a_i)\underbrace{a_{i+1} \dots a_m}_{T'} \in L \overset{q_{i-1} \text{ korrekt}}{\Longleftrightarrow} w \in L \overset{q_i \textbf{ nicht korrekt}}{\Longleftrightarrow} q_i \underbrace{a_{i+1} \dots a_m}_{T'} \notin L.$$

  Daraus folgt $q_{i-1}a_i \not\equiv_{T'} q_i = q$.

Also gilt $q_{i-1}a_i \notin Q$ und (Q', T') ist korrekt. □

Algorithmus 1.3 (*Algorithmus von Angluin*)

Eingabe: DEA A

Schritt 1 $(Q, T) = \big(\{\varepsilon\}, \{\varepsilon\}\big)$

Schritt 2 Vergrößere Q durch wiederholte Anwendung von Lemma 1.17 bis (Q, T) vollständig ist.

Schritt 3 Konstruiere den DEA H für (Q, T) und frage, ob $L(H) = L$.

Schritt 3.1 Falls ja: **Return** H

Schritt 3.2 Falls nein: Vergrößere (Q, T) mit Lemma 1.18

Schritt 4 Goto Schritt 2

Beweis:

Terminierung Nach Lemma 1.16 gilt immer $|Q| \leqslant |\mathsf{Min}(L)|$. Weil Q mit jeder Anwendung von Lemma 1.17/1.18 wächst, muss der Algorithmus terminieren.

Korrektheit Wenn der Algorithmus terminiert, so ist notwendigerweise $L(H) = L$.

Komplexität Sei $n := |\mathsf{Min}(L)|$ und m die Länge des *längsten* Gegenbeispiels. Uns ist bekannt, dass zu jedem Zeitpunkt die Ungleichungskette

$$|T| \leqslant |Q| \leqslant |\mathsf{Min}(L)|$$

gilt. Daher werden die Lemmatas 1.17 und 1.18 höchstens n-mal angewandt. Es ergeben sich folgende Komplexitäten:

Anwendung von Lemma 1.17	$\mathcal{O}\left(n^2 \cdot |\Sigma| \cdot m\right)$ Fragen
Anwendung von Lemma 1.18	$\mathcal{O}\left(\log m\right)$ Fragen
Konstruktion von H	$\mathcal{O}\left(n^2 \cdot |\Sigma| \cdot m\right)$ Fragen
Insgesamt	$\mathcal{O}\left(n^3 \cdot |\Sigma| \cdot m + n \log m\right)$ Fragen

□

Beachte

Falls der Lehrer nun **immer** das kürzeste Gegenbeispiel liefert gilt ständig $m \leqslant n$. Es ergibt sich somit eine Fragenkomlexität von $\mathcal{O}\left(n^4 \cdot |\Sigma|\right)$.

KAPITEL 2

Reguläre Sprachen und Logik

Wir wollen nun die Wörter einer regulären Sprache durch logische Formeln beschreiben. Wir erinnern dazu an **Beispiel 2.1:** Die Wörter der Sprache a^*b^* sind durch die Eigenschaft „es steht kein b links von einem a“ charakterisiert. Als Formel in **monadischer Logik 2. Stufe (MSO)**:

$$\neg \exists x, y. [\, x < y \wedge P_b(x) \wedge P_a(y) \,]$$

Positionen in einem Wort — Pos. x ist links von y — An Pos. x steht ein b — An Pos. y steht ein a

※

Wir wollen zeigen, dass die **reguläre Sprachen genau die durch MSO-Formeln beschreibbare Sprachen** sind.

2.1 Grundlegendes

Definition 2.1 (*MSO-Syntax*)

Fixiere zwei **abzählbar unendliche Mengen**

- $V_1 = \{x, y, z, \dots\}$ (Variablen *erster* Stufe, Kleinbuchstaben) und
- $V_2 = \{X, Y, Z, \dots\}$ (Variablen **zweiter** Stufe, Großbuchstaben)

Dabei sollen Variablen *erster* Stufe *Positionen in einem Wort* und die Variablen **zweiter** Stufe **Mengen von Positionen in einem Wort**.
Die Menge der MSO-Formeln ist induktiv definiert.

- Es gibt eine Reihe an atomaren Formeln:

  $x < y$ (Pos. x steht links von y) — $P_a(x)$ (für $a \in \Sigma$) (An Pos. x steht ein a) — $X(x)$ ($x \in X$)

- Sind φ und ψ MSO-Formeln, dann auch

  $$\neg\varphi \qquad \varphi \vee \psi \qquad \exists x.\, \varphi \qquad \exists X.\, \varphi.$$

Aus diesen Formeln lassen sich dann weitere ableiten:

$$\begin{aligned}
\varphi \wedge \psi &:= \neg(\neg\varphi \vee \neg\psi) \\
\varphi \to \psi &:= \neg\varphi \vee \psi \\
\varphi \leftrightarrow \psi &:= (\varphi \to \psi) \wedge (\psi \to \varphi) \\
\forall x.\, \varphi &:= \neg\exists x.\, \neg\varphi \\
\forall X.\, \varphi &:= \neg\exists X.\, \neg\varphi
\end{aligned}$$

Definition 2.2 (*Variableninterpretation*)

Sei $w = a_1 \dots a_n \in \Sigma^*$. Eine **Variableninterpretation** I ist ein Paar von Funktionen

$$I = \left(I_1 : V_1 \to \{1,\dots,n\} \quad , \quad I_2 \,:\, V_2 \to \mathfrak{P}(\{1,\dots,n\})\right)$$

Definition 2.3 (*MSO-Semantik*)

Sei $w = a_1 \dots a_n \in \Sigma^*$ und I eine Variableninterpretation. Dann ist die Relation $w, I \vDash \varphi$ induktiv wie folgt definiert:

- $w, I \vDash x < y$ genau dann, wenn $I_1(x) < I_1(y)$.
- $w, I \vDash P_a(x)$ genau dann, wenn $a_i = a$ für $i = I_1(x)$.
- $w, I \vDash X(x)$ genau dann, wenn $I_1(x) \in I_2(X)$.
- $w, I \vDash \neg\varphi$ genau dann, wenn $w, I \nvDash \varphi$.
- $w, I \vDash \varphi \vee \psi$ genau dann, wenn $w, I \vDash \varphi$ oder $w, I \vDash \psi$.
- $w, I \vDash \exists\, x.\ \varphi$ genau dann, wenn es ein $i \in \{1,\dots,n\}$ gibt mit $w, I[i/x] \vDash \varphi$.
- $w, I \vDash \exists\, X.\ \varphi$ genau dann, wenn es ein $S \subseteq \{1,\dots,n\}$ gibt mit $w, I[S/x] \vDash \varphi$.

Dabei ist $I[i/x]$ die Interpretation, die x auf i abbildet, und alle anderen Variablen wie I interpretiert. Analog definiert man $I[S/x]$.
Man sagt dann, wenn die Relation erfüllt ist, dass w φ bezüglich I erfüllt.

Definition 2.4 (*Erfüllbarkeit und Allgemeingültigkeit*)

Eine MSO-Formel φ ist ...

... **erfüllbar** genau dann, wenn ein $w \in \Sigma^*$ und eine Variableninterpretation I existiert mit $w, I \vDash \varphi$.

... **allgemeingültig** genau dann, wenn **für alle** $w \in \Sigma^*$ und **für alle** Variableninterpretation I existiert mit $w, I \vDash \varphi$.

Definition 2.5 (*Äquivalenz*)

Zwei MSO-Formeln φ, ψ sind **äquivalent** (wir schreiben dann $\varphi \equiv \psi$), wenn gilt

$$\forall\, w \in \Sigma^*.\ \forall\, I.\ (w, I \vDash \varphi) \leftrightarrow (w, I \vDash \psi).$$

Definition 2.6 (*Freie Variablen und geschlossene Formeln*)

Eine Variable $v \in V_1 \cup V_2$ heißt **frei** in φ, wenn sie außerhalb eines Quantors $\forall\, v$ oder $\exists\, v$ in φ vorkommt.
Eine Formel φ ist **geschlossen**, wenn sie **keine** freien Variablen hat. In diesem Fall ist der Wahrheitswert *unabhängig* von der Interpretation I. Wir schreiben dann auch

$$w \vDash \varphi \quad \text{statt} \quad w, I \vDash \varphi,$$

wobei I beliebig ist.

Beispiel 2.2: Die Variable x ist frei in $x < y \wedge \forall x.\ P_a(x)$, X ist frei in $\forall x.\ X(x)$. ※

Definition 2.7 (*Sprache einer Formel*)

Für eine geschlossene MSO-Formel φ sei

$$L(\varphi) = \left\{ w \in \Sigma^* \;\middle|\; w \vDash \varphi \right\}$$

die **von φ definierte Sprache**.

2.2 Äquivalenz zwischen regulären Sprachen und MSO-Formeln

Wir wollen nun zeigen, dass reguläre Sprachen den MSO-definierbaren Sprachen entsprechen. Dazu definieren wir die folgenden Hilfsformeln:

$$\begin{array}{lrcl}
(1) & x \leqslant y & := & \neg(y < x) \\
(2) & x = y & := & x \leqslant y \wedge y \leqslant x \\
(3) & \mathsf{first}(x) & := & \forall\, y.\; x \leqslant y \\
(4) & \mathsf{last}(x) & := & \forall\, y.\; y \leqslant x \\
(5) & \mathsf{suc}(x,y) & := & x < y \wedge \neg\exists\, z.\; (x < z \wedge z < y)
\end{array}$$

Beispiel 2.3: Sei hier $\Sigma = \{a, b\}$. Wir wollen nun Sprachen durch MSO-Formeln darstellen:

(1) $(a+b)^* ab (a+b)^*$
$\varphi_1 = \exists\, x.\; \exists\, y.\; (\mathsf{suc}(x,y) \wedge P_a(x) \wedge P_b(y))$

(2) $(a+b)^*$
$\varphi_2 = \forall\, x.\; x = x.$

(3) a^*
$\varphi_3 = \forall\, x.\; P_a(x).$

(4) $(aa)^*$
$\varphi_4 = \exists\, X.\; (\forall\, x.\; (\mathsf{first}(x) \to X(x)) \wedge \forall\, y.\; (\mathsf{last}(y) \to \neg X(y)) \wedge \forall\, u, v.\; (\mathsf{suc}(u,v) \to (X(u) \leftrightarrow \neg X(v))))$.

Wir werden später sehen, dass $(aa)^*$ nicht durch eine Formel erster Stufe definierbar ist. ※

Von regulären Sprachen zu MSO-Formeln

Proposition 2.1

Für jede reguläre Sprache $L \subseteq \Sigma^*$ gibt es eine MSO-Formel φ mit $L = L(\varphi)$.

Beweis: Sei $A = (Q, \Sigma, \to, I, F)$ ein NEA mit $L(A) = L$. Wir suchen nun eine Formel φ, so dass für alle $w \in \Sigma^*$ gilt

$$w \vDash \varphi \leftrightarrow A \text{ hat akzeptierenden Lauf für } w.$$

Sei $Q = \{q_0, \ldots, q_m\}$ und $w = a_1 \ldots a_n$. Betrachte einen Lauf von A für w:

$$q_{j_0} \xrightarrow{a_1} q_{j_1} \xrightarrow{a_2} \ldots \xrightarrow{a_n} q_{j_n} \text{ mit } j_0, \ldots, j_n \in \{0, \ldots, m\}.$$

Für $j = 0, \ldots, m$ definiere

$$X_j := \left\{ i \in \{1, \ldots, n\} \;:\; j_i = j \right\}$$

als die Menge aller Positionen von w, bei denen der Lauf den Zustand q_j annimmt.
Wir betrachten dazu **Beispiel 2.4:** Für den Lauf $q_0 \xrightarrow{a} q_2 \xrightarrow{b} q_1 \xrightarrow{a} q_2 \xrightarrow{c} q_0$ ergeben sich die Mengen

$$X_0 = \{4\} \qquad X_1 = \{2\} \qquad X_2 = \{1,3\}.$$

※

Beachte

Ist q_{j_0} bekannt, lässt sich der Lauf aus $X_0, \ldots, X_m$ rekonstruieren.

Die Mengen $X_0, \ldots, X_m$ haben folgende Eigenschaften:

(1) $\bigcup_{j=0}^{m} X_j = \{1, \ldots, n\}$
(2) $X_j \cap X_k = \emptyset$ für $j \neq k$

} X_j bilden eine Partition

(3) $i \in X_j$ und $(i+1) \in X_k$, dann gilt $q_j \xrightarrow{a_{i+1}} q_k$ in A.

(4) $i \in X_j$, so folgt, dass es $q \in I$ mit $q \xrightarrow{a_1} q_j$ in A gibt.

(5) Der Lauf ist akzeptierend, wenn $n \in X_j \Rightarrow q_j \in F$.

Umgekehrt gilt: Sind $X_0, \ldots, X_m \subseteq \{1, \ldots, n\}$ beliebige Mengen mit den Eigenschaften (1) – (5), kann man einen akzeptierenden Lauf für w konstruieren. Dann wegen (1) und (2) gibt es für jedes $i \in \{1, \ldots, n\}$ genau ein $j_i \in \{0, \ldots, m\}$ mit $i \in X_{j_i}$. Wegen Eigenschaft (4) gilt dann $q \xrightarrow{a_1} q_{j_1}$ für $q \in I$, denn $1 \in X_{j_1}$. Wegen Eigenschaft (3) gibt es dann Übergänge

$$q_{j_1} \xrightarrow{a_2} \ldots \xrightarrow{a_n} q_{j_n}$$

und wegen Eigenschaft (5) ist q_{j_n} final, da $n \in X_{j_n}$.
Damit ist $q \xrightarrow{a_1} q_{j_1} \xrightarrow{a_2} \ldots \xrightarrow{a_n} q_{j_n}$ ein akzeptierender Lauf für w. Zusammengefasst gilt nun also die Äquivalenz, dass A einen akzeptierenden Lauf für w hat genau dann, wenn es Mengen $X_0, \ldots, X_m \subseteq \{1, \ldots, n\}$ mit den Eigenschaften (1) – (5) gibt.
Diese Eigenschaften lassen sich aber auch als MSO-Formel ausdrücken, wir erhalten als Formel

$$\varphi = \exists X_0, \ldots, X_m.\ \varphi_1 \wedge \varphi_2 \wedge \varphi_3 \wedge \varphi_4 \wedge \varphi_5 \qquad \text{mit}$$

$$\varphi_1 = \forall x.\ \left(\bigvee_{i=0}^{m} X_i(x) \right)$$

$$\varphi_2 = \bigwedge_{j \neq k} \neg \exists x.\ \left(X_j(x) \wedge X_k(x) \right)$$

$$\varphi_3 = \bigwedge_{j,k} \forall x, y.\ \left(\left(\mathsf{suc}(x,y) \wedge X_j(x) \wedge X_k(x) \right) \rightarrow \bigvee_{q_j \xrightarrow{a} q_k} P_a(y) \right)$$

$$\varphi_4 = \bigwedge_{j} \forall x.\ \left(\left(\mathsf{first}(x) \wedge X_j(x) \right) \rightarrow \bigvee_{\substack{q \xrightarrow{a} q_k \\ q \in I}} P_a(x) \right)$$

$$\varphi_5 \quad = \bigvee_{\substack{j\in\{0,\dots,m\}\\ q_j\in F}} \exists x.\ (\mathsf{last}(x) \wedge X_j(x))$$

Nach Konstruktion gilt somit $L(\varphi) = L(A) = L$. □

Von MSO-Formeln zu regulären Sprachen

Wir wollen nun zeigen, dass für jede MSO-Formel φ $L(\varphi)$ regulär ist. Wir stossen relativ schnell auf eine technische Hürde; wollen wir nämlich per Induktion argumentieren, werden wir schnell feststellen, dass wir auch Formeln φ **mit freien Variablen** eine Sprache zuordnen müssen. Wir lösen dies über eine Kodierung der Interpretation von freien Variablen von φ in das Alphabet hinein. Betrachte dazu **Beispiel 2.5:** Sei φ eine Formel mit den freien Variablen x, y, z der ersten Stufe und X der zweiten Stufe. Sei $w = aabab$. Betrachte dann die Interpretation $I_1(x) = 2, I_1(y) = I_1(z) = 4, I_2(x) = \{1,2,5\}$. Wir können dies auch tabellarisch darstellen:

a	a	b	a	b
—	x	—	y,z	—
X	X	—	—	X

Kodiert als Wort über dem Alphabet

$$\{a,b\} \times \mathfrak{P}(\{x,y,z\}) \times \mathfrak{P}(\{X\})$$

ergibt sich

$$w = (a,\emptyset,\{X\})\,(a,\{x\},\{X\})\,(b,\emptyset,\emptyset)\,(a,\{y,z\},\emptyset)\,(b,\emptyset,\{X\}).$$

※

Wir definieren damit allgemein:

Definition 2.8

Seien $W_1 \subseteq V_1$ und $W_2 \subseteq V_2$ **endliche** Mengen von Variablen erster beziehungsweise zweiter Stufe. Dann ist ein (W_1, W_2)-Wort ein Wort über dem Alphabet $\Sigma \times \mathfrak{P}(W_1) \times \mathfrak{P}(W_2)$, also von der Form $(a_1, S_1, T_1)\dots(a_n, S_n, T_n)$ mit $S_i \subseteq W_1$, $T_i \subseteq W_2$ und

$$W_1 = \bigcup_{i=1}^{n} S_i \quad \text{und} \quad S_i \cap S_j = \emptyset \text{ für } i \neq j.$$

Mit anderen Worten kommt jede Variable aus W_1 genau in einem S_i vor.

Beachte

Es gibt einen DEA A_{W_1,W_2}, der die Sprache aller (W_1, W_2)-Wörter akzeptiert.

Sei $w = a_1 \dots a_n \in \Sigma^*$ und $I = (I_1 : V_1 \to \{1,\dots,n\} \quad , \quad I_2 : V_2 \to \mathfrak{P}(\{1,\dots,n\}))$ eine Interpretation von W_1, W_2. Definiere dann das W_1, W_2-Wort

$$w_I = (a_1, S_1, T_1)\dots(a_n, S_n, T_n)$$

mit

$$S_i = \{x \in W_1 \mid I_1(x) = i\} \text{ und } T_i = \{X \in W_2 \mid i \in I_2(X)\}.$$

Definition 2.9

Sei φ eine MSO-Formel mit $\text{free}_1(\varphi) \subseteq W_1$ und $\text{free}_2(\varphi) \subseteq W_2$, wobei $\text{free}_i(\varphi)$ die Menge der freien Variablen i-ter Stufe in φ ist. Die von φ definierte Sprache ist dann

$$L_{W_1,W_2}(\varphi) = \left\{ w_I \;\middle|\; w \in \Sigma^*, I \text{ ist Interpretation von } W_1, W_2 \text{ und } w, I \vDash \varphi \right\}.$$

Bemerkung

Falls φ eine geschlossene Formel ist, kann man $W_1 = W_2 = \emptyset$ wählen. Dann ist

$$(\Sigma \times \mathfrak{P}(\emptyset) \times \mathfrak{P}(\emptyset))^* \supseteq L_{\emptyset,\emptyset}(\varphi) \cong L(\varphi) \subseteq \Sigma^*.$$

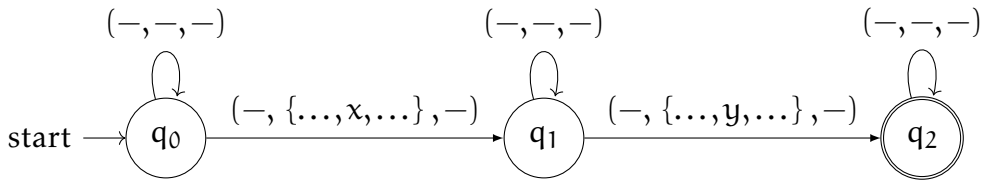
Dazu identifiziere $(a, \emptyset, \emptyset)$ mit a.

Proposition 2.2

Für jede geschlossene MSO-Formel φ ist $L(\varphi)$ regulär.

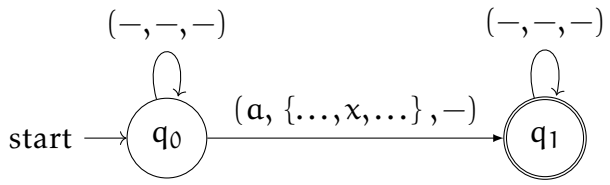
Beweis: Wir zeigen nun allgemeiner: Für alle endliche Mengen $W_1 \subseteq V_1$, $W_2 \subseteq V_2$ und alle MSO-Formeln φ mit $\text{free}_1(\varphi) \subseteq W_1$ und $\text{free}_2(\varphi) \subseteq W_2$ ist die Sprache $L_{W_1,W_2}(\varphi)$ regulär. Konkret wollen wir zeigen, dass es einen NEA A_φ für diese Sprache gibt. Wir zeigen per struktureller Induktion:

$\varphi = x < y$ Wir bauen den NEA $A_{x<y}$ als eine Parallelschaltung aus A_{W_1,W_2} mit

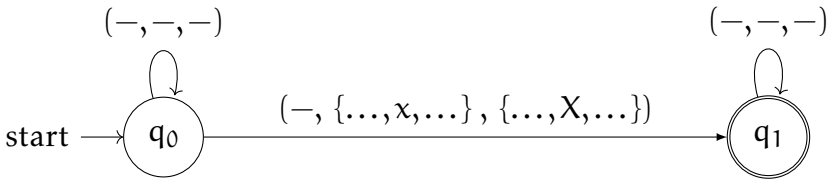


Die Parallelschaltung stellt sicher, dass nur (W_1, W_2)-Wörter akzeptiert werden.

$\varphi = P_a(x)$ Wir bauen den NEA $A_{P_a(x)}$ als eine Parallelschaltung aus A_{W_1,W_2} mit



$\varphi = X(x)$ Wir bauen den NEA $A_{X(x)}$ als eine Parallelschaltung aus A_{W_1,W_2} mit



$\varphi = \neg\psi$ Uns ist bekannt, dass in $L_{W_1,W_2}(\varphi)$ all die (W_1, W_2)-Wörter liegen, die **nicht** in $L_{W_1,W_2}(\psi)$ liegen. Damit erhalte den NEA $A_{\neg\psi}$ wie folgt:

Schritt 1 Determinisiere A_ψ

Schritt 2 Vertausche Final- und Nichtfinalzustände

Schritt 3 Schalte A_{W_1,W_2} parallel

$\varphi = \varphi_1 \vee \varphi_2$ Uns ist bekannt, dass in $L_{W_1,W_2}(\varphi)$ die Vereinigung der Sprachen $L_{W_1,W_2}(\varphi_1)$ und $L_{W_1,W_2}(\varphi_2)$ ist. Damit erhalte einen NEA für φ durch Vereinigung der NEAs A_{φ_1} und A_{φ_2}.

$\varphi = \exists\, x.\, \psi$ Es gilt $\mathrm{free}_1(\psi) \subseteq \mathrm{free}_1(\varphi) \cup \{x\} \subseteq W_1 \cup \{x\}$ und $\mathrm{free}_2(\psi) \subseteq \mathrm{free}_2(\varphi) \subseteq W_2$. Wir sehen nun, dass ein (W_1, W_2)-Wort $(a_1, S_1, T_1)\dots(a_n, S_n, T_n)$ genau dann in $L_{W_1,W_2}(\varphi)$ liegt, wenn es ein $i \in \{1,\dots,n\}$ gibt, so dass das $(W_1 \cup \{x\}, W_2)$-Wort $(a_1, S_1, T_1)\dots(a_i, S_i \cup \{x\}, T_i)\dots(a_n, S_n, T_n)$ in $L_{W_1\cup\{x\},W_2}(\psi)$ liegt.
Mit anderen Worten entsteht $L_{W_1,W_2}(\varphi)$ aus $L_{W_1\cup\{x\},W_2}(\psi)$, indem man jedes Symbol

$$(a, S, T) \in \Sigma \times \mathfrak{P}(W_1 \cup \{x\}) \times \mathfrak{P}(W_2)$$

durch $(a, S\backslash\{x\}, T) \in \Sigma \times \mathfrak{P}(W_1) \times \mathfrak{P}(W_2)$ ersetzt.

Erhalte den NEA $A_{\exists\, x.\, \psi}$ aus A_ψ, indem man jeden Übergang $p \xrightarrow{(a,S,T)} q$ in A_ψ durch $p \xrightarrow{(a,S\backslash\{x\},T)} q$ ersetzt.

$\varphi = \exists\, X.\, \psi$ volkommen analog zum letzten Fall, ersetze hier (a, S, T) durch $(a, S, T\backslash\{X\})$

□

Mit den Propositionen 2.1 und 2.2 haben wir folgenden Satz gezeigt:

Satz 2.3 (*Satz von Büchi, Elgot, Trakhtenbrot*)

Eine Sprache $L \subseteq \Sigma^*$ ist genau dann regulär, wenn es eine MSO-Formel φ mit $L = L(\varphi)$ gibt.

Korrolar 2.3.1

Das Erfüllbarkeitsproblem für MSO^1 ist entscheidbar.

Beweis:
Algorithmus 2.1

Schritt 1 Konstruiere A_φ mit $L(A_\varphi) = L(\varphi)$.

Schritt 2 Prüfe, ob $L(A_\varphi) \neq \emptyset$, das heißt, ob ein Finalzustand von A_φ erreichbar ist.

Die Korrektheit ist offensichtlich, damit folgt unmittelbar die Entscheidbarkeit des Erfüllbarkeitsproblems. □

Einschub – Existenzielle Logik zweiter Stufe

Allgemeine Logik zweiter Stufe (SO) hat Variablen zweiter Stufe

$$V_2^m = \{R, S, T, \dots\} \qquad (m \geqslant 1),$$

die m-stellige Prädiakte $R \subseteq \{1,\dots,n\}^m$ repräsentieren. Bei der MSO gilt $m = 1$. Damit lassen sich aber auch nicht-reguläre Sprachen beschreiben, wir betrachten dazu **Beispiel 2.6:**

$$L = \left\{ w \in \{a, b\}^* \;\middle|\; |w|_a = |w|_b \right\}$$

Da die Anzahl an a und b identisch ist, können wir eine bijektive Paarung zwischen as und bs erstellen. Wähle also eine Variable $R \subseteq V_2^2$ und baue eine SO-Formel für L:

$$\varphi = \exists\, R.\ \Bigg(\underbrace{\forall\, x.\, \forall\, y.\, \big(R(x,y) \to P_a(x) \wedge P_b(y)\big)}_{\text{definiert Relation}} \wedge \underbrace{\forall\, x.\, \exists!\, y.\, R(x,y) \wedge \forall\, y.\, \exists!\, x.\, R(x,y)}_{\text{garantiert Bijektivität}} \Bigg)$$

※

Wir wollen an dieser Stelle auf die Besonderheit der obigen Sprache eingehen. Wir erkennen, dass die Formel hinter dem ersten Existenzquantor eine Formel erster Stufe ist und wir nur mit einem Existenzquator über Variablen zweiter Stufe quantifizieren. Man nennt diese Formeln dann auch ESO-Formeln und erkennt einen Zusammenhang zur Komplexitätsklasse $\mathcal{NP}$.

Definition 2.10 (*ESO*)

Eine ESO-Formel (existenzielle Logik zweiter Stufe) ist eine SO-Formel der Form

$$\exists\, R_1.\ \ldots \exists\, R_k.\ \varphi,$$

wobei $R_1, \ldots, R_k$ Variablen zweiter Stufe sind und φ eine Formel erster Stufe ist.

Mit dieser Definition und ähnlichen Überlegungen wie oben ergibt sich der folgende Satz:

Satz 2.4 (*Satz von Fagin*)

Die ESO-definierbaren Sprachen sind genau die Sprachen in $\mathcal{NP}$.

Aus dem Beweis des Satzes lässt sich dann schließen, dass die Formeln der allgemeinen Logik zweiter Stufe (SO) genau die Polynomialzeithierarchie beschreiben. Die Aussagen sind im Teilbereich der deskriptiven Komplexitätstheorie wiederzufinden.

2.3 Logik erster Stufe

Wir betrachten nun das MSO-Fragment FO (**first-order**) aller Formeln φ, in denen nur Variablen erster Stufe vorkommen. In BNF definieren wir also:

$$\varphi ::= x < y \quad | \quad P_a(x) \quad | \quad \neg\varphi \quad | \quad \varphi \vee \varphi \quad | \quad \exists x.\, \varphi,$$

wobei $x \in V_1$ erfüllt sein muss.
Wir definieren dann damit äquivalent zu einem gegebenen Wort $u = a_1 \ldots a_n \in \Sigma^*$ und einer FO-Formel φ eine **– zu u und φ passende – Interpretation** als eine Funktion

$$I : V \to \{1, \ldots, n\},$$

wobei $V \subseteq V_1$ und $\mathrm{free}_1(\varphi) \subseteq V$.

Definition 2.11

Sei $V \subseteq V_1$ eine endliche Menge, so definieren wir ...

(a) ... ein **V-Wort** als ein Wort über dem Alphabet $\Sigma \times \mathfrak{P}(V)$, also als ein Wort der Form

$w = (a_1, S_1) \ldots (a_n, S_n)$ mit $a_i \in \Sigma, S_i \subseteq V$, wobei die S_i eine Partition von V bilden.

Dabei ist die Partitionseigenschaft sichergestellt durch

$$\bigcup_{i=1}^{n} S_i = V \text{ und } S_i \cap S_j = \emptyset \text{ für } i \neq j.$$

V-Wörter kodieren FO-Interpretationen, was wir auch sehen an **Beispiel 2.7:** Gegeben sei das V-Wort

$$w_1 = (a_1, \{x\})(b, \emptyset)(a, \{y, z\}).$$

Damit ist eine Interpretation $\mathcal{I}$ mit

$$\mathcal{I}(x) = 1 \text{ und } \mathcal{I}(y) = \mathcal{I}(z) = 3$$

kodiert.

(b) Sei φ eine FO-Formel mit $\text{free}_1(\varphi) \subseteq V_1$. Das V-Wort w **erfüllt** φ, wenn $a_1 \dots a_n, I \vDash \varphi$ für die Interpretation I, die von w kodiert wird. Wir schreiben dann auch $w \vDash \varphi$. Am obigen **Beispiel 2.7** (*Fortsetzung*)**:** Wir sehen, dass

$$w_1 \vDash x < y.$$

※

Ziel der nächsten Vorlesungen: Wir wollen zeigen, dass die Sprache $(aa)^*$ nicht FO-definierbar ist.
Methode: Wir wollen dazu Ehrenfeucht-Fraïssé-Spiele, eine Standardtechnik der Modelltheorie für das Zeigen der Grenzen Logik erster Stufe, verwenden. Dazu definieren wir weiter:

Definition 2.12

Der **Quantorenrang** $\text{qr}(\varphi)$ einer FO-Formel φ ist induktiv definiert als:

- $\text{qr}(x < y) = \text{qr}(P_a(x)) = 0$
- $\text{qr}(\neg\varphi) = \text{qr}(\varphi)$
- $\text{qr}(\varphi \vee \psi) = \max\{\text{qr}(\varphi), \text{qr}(\psi)\}$ und
- $\text{qr}(\exists x.\ \varphi) = \text{qr}(\varphi) + 1$

Beispiel 2.8:

(1) Formeln mit Quantorenrang 0 sind genau die quantorenfreien Formeln, das heißt boolsche Kombinationen von atomaren Formeln, wie zum Beispiel

$$\neg((x < y) \wedge P_a(y)) \to (z < x).$$

(2) Die Formel

$$(\exists x.\ ((x < y) \wedge \exists z.\ P_a(z))) \to \exists y.\ P_a(y)$$

hat Quantorenrang 2.

※

Definition 2.13

Seien w, w' zwei V-Wörter, dann definieren wir:

(a) Eine FO-Formel φ mit $\text{free}_1(\varphi) \subseteq V$ **unterscheidet** w und w', wenn genau eines der beiden Wörter die Formel φ erfüllt.

(b) w und w' sind k-**unterscheidbar**, wenn es eine Formel φ mit einem Quantorenrang $r \leqslant k$ gibt, welche w und w' unterscheidet.

Beachte

Wenn w und w' durch $\neg\varphi$ unterschieden werden, dann auch durch φ. Wenn w und w' durch $\varphi \vee \psi$ unterschieden werden, dann auch durch φ oder durch ψ.
Das heißt, wenn durch eine boolsche Kombination von Formeln w und w' unterscheiden werden, dann gibt es eine Unterformel $\exists\, x.\ \varphi$, $P_a(x)$ oder $x < y$, welche w und w' unterscheidet.

Als **Konsequenz** stellen wir fest, dass w und w' genau dann 0-unterscheidbar sind, wenn es eine atomare Formel gibt, die w und w' unterscheidet. Das heißt aber, dass einer der folgenden Fälle eintritt:

Fall i: Es gibt ein $x \in V$, so dass die x-markierten Positionen in w und w' unterschiedliche Symbole a und a' aus Σ tragen. Dann wäre $P_a(x)$ eine „trennende Formel“, also eine Formel, die w und w' unterscheidet.

Fall ii: Es gibt $x, y \in V$, so dass in genau einem der Wörter w und w' die x-Position links von der y-Position steht. Dann wäre $x < y$ eine „trennende Formel“.

Interpretation von k-Unterscheidbarkeit durch Ehrenfeucht-Fraïssé-Spiele. Fixiere V-Wörter w und w' und $k \geqslant 0$. Wir betrachten dann folgendes 2-Personen-Spiel:

Verfahren 2.2 (*Ehrenfeucht-Fraïssé-Spiel*)

Spieler: Spieler 0 („**spoiler**“, kurz S) möchte zeigen, dass w und w' k-unterscheidbar sind. Spieler 1 („**duplicator**“, kurz D) möchte es wiederlegen.

Es werden k Runden gespielt. In jeder Runde $i \in \{1, \ldots, k\}$ wählt **spoiler** eines der Wörter w oder w' und markiert eine Position in diesem Wort mit einer neuen Variable x_i.

$$\text{In anderen Worten:} \qquad x_i \notin V \cup \{x_1, \ldots, x_{i-1}\}$$

Duplicator antwortet, indem er in dem anderen Wort eine Position mit x_i markiert.
Resultat: Nach k Runden entstehen somit **zwei** $V \cup \{x_1, \ldots, x_k\}$-Wörter.
Spoiler gewinnt, wenn diese beiden Wörter 0-unterscheidbar sind. Sonst gewinnt **duplicator**.
Wir sagen, dass **Spoiler** eine **Gewinnstrategie** hat, wenn er seine Züge (in Abhängigkeit von den vorherigen Zügen von **duplicator**) so wählen kann, dass er garantiert nach k Runden gewinnt.

Wir betrachten **Beispiel 2.9:** Gegeben seien die $\emptyset$-Wörter

$$w = abbabbab \text{ und } w' = ababbabb.$$

Spoiler hat folgende Gewinnstrategie für das 2-Runden-Spiel auf w, w':

Runde 1: **Spoiler** markiert Position 7 von w' mit x_1. **Duplicator muss** nun eine b-Position von w mit x_i markieren.

Runde 2: **Spoiler** betrachtet zwei Fälle:

Fall a: Falls das **letzte** b von w mit x_1 markiert wurde, dann markiert **spoiler** Position 8 von w.
Duplicator muss eine Position von w mit x_2 markieren, die rechts von x_1 steht, aber das ist UNMÖGLICH!

Fall b: Falls ein **anderes** b von w mit x_1 markiert wurde, dann markiert **spoiler** eine a-Position von b, die rechts von x_1 steht.
Duplicator muss eine a Position rechts von x_1 in w' mit x_2 markieren, aber das ist UNMÖGLICH!

※

Satz 2.5 (*Satz von Ehrenfeucht-Fraïssé*)

Sei $k \geqslant 0$. Zwei V-Wörter w und w' sind genau dann k-unterscheidbar, wenn **spoiler** eine Gewinnstrategie für das k-Runden-Spiel auf den Wörtern w und w' hat.

Beweis:

„⇐“ Angenommen **spoiler** hat eine Gewinnstrategie für das k-Runden-Spiel auf w und w'. Wir zeigen, dass w und w' dann k-unterscheidbar sind, per Induktion nach k:
Induktionsanfang ($k = 0$):

Wenn **spoiler** nach 0 Runden gewinnt, sind w und w' 0-unterscheidbar.

Induktionsschritt ($k > 0$):

Sei nun

$$w = (a_1, S_1)\dots(a_n, S_n) \qquad \text{und} \qquad w' = (a'_1, S'_1)\dots(a'_{n'}, S'_{n'}).$$

Betrachte dann die Gewinnstrategie von **spoiler**. In Runde 1 wählt **spoiler** Œ das Wort w und markiert Position $i \in \{1,\dots,n\}$ mit x_1. **Duplicator** markiert daraufhin eine Position $j \in \{1,\dots,n'\}$ in w' mit x_1.
Nach Runde 1 ergeben sich also die $V \cup \{x_1\}$-Wörter

$$w_i = (a_1, S_1)\dots(a_i, S_i \cup \{x_1\})\dots(a_n, S_n) \quad \text{und} \quad w'_j = (a'_1, S'_1)\dots(a'_j, S'_j \cup \{x_1\})\dots(a'_{n'}, S'_{n'}).$$

Weil **spoiler** das k-Runden-Spiel auf w und w' notwendigerweise gewinnt (per Voraussetzung), hat er ebenso notwendig für jedes $j \in \{1,\dots,n'\}$ eine Gewinnstrategie für das $(k-1)$-Runden-Spiel auf w_i und w'_j.
Per Induktion gibt es also für **jedes** $j \in \{1,\dots,n'\}$ eine Formel φ_j mit $qr(\varphi_j) \leqslant k-1$, welche w_i und w'_j unterscheidet.
Das heißt dann aber auch, dass Œ gilt, dass

$$w_i \vDash \varphi_j \qquad \text{und} \qquad w'_j \nvDash \varphi_j.$$

Dann gilt aber auch

$$w \vDash \exists\, x_1.\ \bigwedge_{j=1}^{n'} \varphi_j \qquad \text{, aber} \qquad w' \nvDash \exists\, x_1.\ \bigwedge_{j=1}^{n'} \varphi_j,$$

wobei wir x_1 als Position i interpretieren.
Damit ist $\exists\, x_1.\ \bigwedge_{j=1}^{n'} \varphi_j$ eine Formel vom Quantorenrang kleiner als k, die w und w' unterscheidet.

„⇒“ Sei φ eine FO-Formel mit Quantorenrang $\leqslant k$, die w und w' unterscheidet. Wir zeigen, dass **spoiler** eine Gewinnstrategie für das k-Runden-Spiel auf w und w' hat, per Induktion nach k:

Induktionsanfang ($k = 0$):

Weil φ eine quantorenfreie Formel ist, sind w und w' 0-unterscheidbar. Also gewinnt – per definitionem – **spoiler** das 0-Runden-Spiel auf w und w'.

Induktionsschritt ($k > 0$):

Œ dürfen wir annehmen, dass

$$\varphi = \exists\, x_1.\, \psi$$

gilt, denn wenn φ eine boolsche Kombination solcher Formeln ist, ersetze φ einfach durch eine geeignete Teilformel.
Dann gibt es in **genau einem** der Wörter w oder w' eine Position i, so dass φ mit der Interpretation $x_1 \mapsto i$ erfüllt ist.
Spoiler wählt in Runde 1 ebendieses Wort und markiert Position i mit x_1. In dem anderen Wort ist ψ für **keine** Interpretation von x_i erfüllt. Das heißt egal wie **duplicator** in Runde 1 antwortet, werden die beiden resultierenden $V \cup \{x_1\}$-Wörter v und v' durch die Formel ψ unterschieden. Weil nun $\mathrm{qr}(\psi) = \mathrm{qr}(\varphi) - 1 \leqslant k - 1$, hat **spoiler** per Induktion eine Gewinnstrategie für das $(k-1)$-Runden-Spiel auf v und v'.
Spielt **spoiler** die Runden 2 bis k gemäß dieser Strategie, gewinnt er das k-Runden-Spiel auf w und w'. □

Proposition 2.6

Sei $k \geqslant 0$ und $m \geqslant 2^k$. Dann sind die Wörter $w = a^m$ und $w' = a^{m+1}$ **nicht** k-unterscheidbar.

Beweis: Wir zeigen, dass **duplicator** eine Gewinnstrategie für das k-Rundenspiel auf w und w' hat. Daraus folge nach dem Satz von Ehrenfeucht-Fraïssé (Satz 2.5), dass w und w' **nicht** k-unterscheidbar sind. Wir beweisen per Induktion nach k:

Induktionsanfang ($k = 0$):

Weil w und w' $\emptyset$-Wörter sind, gibt es keine freie QF-Formel, die w und w' unterscheidet. Also gewinnt **duplicator** das 0-Runden-Spiel auf w und w'.

Induktionsschritt ($k > 0$):

Betrachte das k-Runden-Spiel auf $w == a^m$ und $w' = a^{m+1}$ mit $m \geqslant 2^k$. In Runde 1 wählt **spoiler** eines der Wörter w oder w' und markiert eine Position $s+1$ mit x_1. Das heißt, wir erhalten eine Zerlegung

$$w = a^s\, a\, a^t \qquad \text{mit } s + t + 1 = m$$

oder

$$w' = a^s\, a\, a^{t+1} \qquad \text{mit } s + t + 1 = m.$$

Aus Symmetriegründen dürfen wir $s \leqslant t$ annehmen. **Duplicator** antwortet, indem er dieselbe Position $s+1$ in dem anderen Wort mit x_1 markiert.
Nach Runde 1 erhalten wir damit zswei $\{x_1\}$-Wörter

$$w = a^s\, (a, \{x_1\})\, a^t \qquad \text{und} \qquad a^s\, (a, \{x_1\})\, a^{t+1}.$$

Dann gilt offensichtlich, dass $t \geqslant 2^{k-1}$.[2] Per Induktion hat **duplicator** nun eine Gewinnstrategie für das $k-1$-Runden-Spiel auf a^t und a^{t+1}. In der folgenden Runden $i \in \{2,\ldots,k\}$ spielt **duplicator** dann wie folgt:

1. Falls **spoiler** eine der ersten $s+1$ Positionen von w oder w' mit x_i markiert, dann markiert **duplicator** dieselbe Position des anderen Wortes mit x_i.
2. Falls **spoiler** eine Position größer als $s+1$ eines Wortes mit x_i markiert (das heißt im Suffix a^t von w oder a^{t+1} von w'), antwortet **duplicator** gemäß seiner Gewinnstrategie für a^t und a^{t+1}.

Damit gibt es eine Gewinnstrategie für **duplicator** für das k-Rundenspiel auf w und w'. □

Satz 2.7

Es gibt keine FO-Formel φ mit

$$L(\varphi) = (aa)^*$$

Beweis: Angenommen eine solche Formel φ existiert. Sei k der Quantorenrang von φ, so gilt:

$$(aa)^* \ni a^{2k} \vDash \varphi$$

und

$$(aa)^* \not\ni a^{2k+1} \nvDash \varphi,$$

womit die Wörter a^{2k} und a^{2k+1} k unterscheidbar wären. Dies steht allerdings im Widerspruch zur Proposition 2.6. □

Eine der nun noch offenen Fragen ist die Entscheidbarkeit der FO-definierbarkeit. Auch diese Frage ist geklärt, es ist nämlich möglich diese algorithmisch zu entscheiden. Man benötigt hierfür aber algebraische Methoden.

[2] Denn angenommen t wäre kleiner oder gleich $2^{k-1}-1$, dann gölte aber auch, dass $2^k \leqslant m = |w| = s+t+1 \leqslant 2t+1 \leqslant 2(2^{k-1}-1)+1 = 2^k-1$, was unmittelbar zum Widerspruch führt.

KAPITEL 3

Reguläre Sprachen und Monoide

Unser Ziel in diesem Kapitel ist die algebraische Charakterisierung von regulären Sprachen und ihren Eigenschaften.

3.1 Monoide

Wir wollen zu Beginn definieren:

Definition 3.1 (*Monoid*)

Ein **Monoid** ist eine Menge M mit einer binären Operation

$$\begin{aligned} \cdot \quad : \quad & M \times M \to M, \\ & (m, m') \mapsto m \cdot m' \end{aligned}$$

und einem (Eins-)element $\mathbf{1} \in M$ mit folgenden Eigenschaften:

(1) Assoziativität von $\cdot$, sprich

$$\forall\, p,\ m,\ n \in M.\ (m \cdot n) \cdot p = m \cdot (n \cdot p)$$

(2) $\mathbf{1}$ ist neutrales Element von $\cdot$, sprich

$$\forall\, m \in M.\ \mathbf{1} \cdot m = m \cdot \mathbf{1} = m.$$

Wir wollen das Monoid allgemein als Tripel

$$(M, \cdot, \mathbf{1})$$

auffassen, bei ersichtlicher Verknüpfung und neutralem Element schreiben wir manchmal aber auch nur M.

Beispiel 3.1:

(1) $(\mathbb{N}, +, 0)$ ist ein Monoid

(2) $(\mathbb{N}, \cdot, 1)$ ist ein Monoid

(3) Σ^* ist ein Monoid bezüglich der Operation der *Konkatenation* zweier Wörter ($v \cdot w = vw$) und mit neutralem Element ε

(4) Für $n \geqslant 1$ sei $\mathbb{Z}_n = \{0, \ldots, n-1\}$. Für $a, b \in \mathbb{Z}_n$ definiere dann die Operation $\oplus$ durch

$$a \oplus b = (a + b) \mod n$$

als die „Addition modulo n“. Dann ist $(\mathbb{Z}_n, \oplus, 0)$ ein Monoid.

(5) Für $n \geqslant 1$ sei $U_n = \{1, u_1, \ldots, u_n\}$. Definiere eine binäre Multiplikation auf U_n durch

$$u_i \cdot u_j = u_j \quad \text{und} \quad 1 \cdot u_i = u_i \cdot 1 = u_i$$

für alle $i, j = 1, \ldots, n$. Dann ist $(U_n, \cdot, 1)$ ein Monoid.

(6) Für jede Menge X ist die Menge $\mathfrak{P}(X \times X) = \{R : R \subseteq X \times X\}$ **aller** binären Relationen auf X ein Monoid bezüglich der Komposition von Relationen, welche durch

$$RS = \{(x, y) \in X \times X : \exists z \in X.\ (x, z) \in R \wedge (z, y) \in S\}$$

definiert ist. Das neutrale Element ist die Identität $\mathrm{id}_X = \{(x, x) : x \in X\}$.

(7) Das **triviale Monoid** ist das Monoid $\{\mathbf{1}\}$, welches nur aus dem neutralen Element besteht. Es ist gleichzeitig auch das einzige Monoid mit nur einem Element (bis auf Isomorphie). Wir schreiben sehr oft auch $\mathbf{1}$ statt $\{\mathbf{1}\}$.

※

Wir wollen nun die Sätze und Definitionen aus Kapitel 1 auf Monoide verallgemeinern um danach einen Zusammenhang zu regulären Sprachen herstellen zu können.

Definition 3.2 (*Morphismen*)

Ein **Morphismus** zwischen zwei Monoiden M und N ist eine Funktion $h : M \to N$ mit den Eigenschaften

(i) $h(m \cdot m') = h(m) \cdot h(m')$ für alle $m, m' \in M$ und

(ii) $h(\mathbf{1}) = \mathbf{1}$.

Beispiel 3.2:

(1) $h : (\mathbb{N}, +, 0) \to (\mathbb{Z}_n, \oplus, 0), \quad k \mapsto k \mod n$ ist ein Morphismus.

(2) $h : \Sigma^* \to (\mathbb{N}, +, 0), \quad w \mapsto |w|$ ist ein Morphismus.

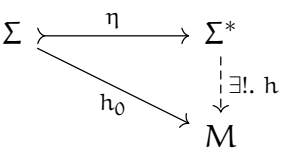
(3) $h : \{a, b\}^* \to U_2 = \{1, u_1, u_2\}$, wobei für alle $w \in \{a, b\}^*$ gelte, dass $h(\epsilon) = 1$, $h(wa) = u_1$ und $h(wb) = u_2$, ist ein Morphismus.

※

Wir beschäftigen uns damit nun wieder mit universellen Eigenschaften von Monoiden:

Proposition 3.1 (*universelle Eigenschaften von Σ^**)

Σ^* ist das **freie Monoid** auf Σ, das heißt, dass es für jedes Monoid M und jede Funktion $h_0 : \Sigma \to M$ *genau einen* Monoidmorphismus $h : \Sigma^* \to M$ mit $h(a) = h_0(a)$ für alle $a \in \Sigma$ gibt.



Beweis: Wir zeigen Existenz und Eindeutigkeit:

zur Existenz Für $w = a_1 \ldots a_n \in \Sigma^*$ definere $h(w) = h_0(a_1) \ldots h_0(a_n)$. Dies liefert einen Morphismus $h : \Sigma^* \to M$ mit $h(a) = h_0(a)$ für alle $a \in \Sigma$.

zur Eindeutigkeit Sei h' ein weiterer Morphismus, für welchen $h'(a) = h_0(a)$ für alle $a \in \Sigma$ gelte, dann gilt allerdings für $w = a_1 \dots a_n \in \Sigma^*$, dass

$$\begin{aligned} h'(w) &= h'(a_1 \dots a_n) \\ &= h'(a_1) \cdot \dots h'(a_n) && (h' \text{ ist Morphismus}) \\ &= h_0(a_1) \cdot \dots \cdot h_0(a_n) && (h'(a_i) = h_0(a_i)) \\ &= h(w) && (\text{Def. von } h(w)) \end{aligned}$$

Damit sind h und h' identisch.

□

Definition 3.3 (*Kongruenz*)

Eine **Kongruenz** auf einem Monoid M ist eine Äquivalenzrelation $\equiv \subseteq M \times M$, so dass für alle $m, m', n, n' \in M$ gilt:

$$m \equiv m' \wedge n \equiv n' \to m \cdot n \equiv m' \cdot n'.$$

Proposition 3.2 (*Kongruenzen vs. Morphismen*)

(1) Für jeden Monoidmorpismus $h : M \to N$ ist der **Kern** von h definiert als

$$m \equiv_h m' \leftrightarrow h(m) = h(m')$$

und damit eine Kongruenz auf M.

(2) Eine jede Kongruenz $\equiv$ auf M induziert das **Quotientenmonoid**

$$M/_{\equiv} = \left\{ [m]_{\equiv} \;\middle|\; m \in M \right\} \qquad (\text{Menge an Kongruenzklassen})$$

mit der Multiplikation $\cdot$ definiert durch

$$[m]_{\equiv} \cdot [m']_{\equiv} = [m \cdot_M m']_{\equiv}$$

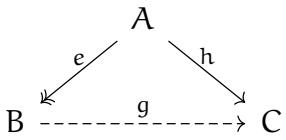
für beliebiges $m, m' \in M$ und neutralem Element $[\mathbf{1}_M]_{\equiv}$, wobei $\cdot_M$ die Multiplikation auf M und $\mathbf{1}_M$ das neutrale Element von M ist.
Die Funktion $h_{\equiv} : M \to M/_{\equiv}$, $m \mapsto [m]_{\equiv}$ ist ein Monoidmorphismus mit Kern $\equiv$.

Beweis: Auch hier beweist man die Aussage analog wie bei den Automaten (Propositionen 1.7 und 1.8). □

Wir kommen nun erneut zum Homomorphiesatz:

Satz 3.3 (*Homomorphiesatz*)

Seien $e : A \to B$, $h : A \to C$ Morphismen von DAs und e surjektiv. Dann gilt:



(1) Jede Funktion $g : Q_B \to Q_C$ mit $h = g \circ e$ ist ein Morphismus von B nach C.

(2) Folgende Aussagen sind äquivalent:

(a) Es gibt einen Morphismus $g : B \to C$ mit $h = g \circ e$

(b) $\sim_e \subseteq \sim_h$

(c) $\forall\, q, q' \in Q_A.\ e(q) = e(q') \Rightarrow h(q) = h(q')$

Beweis: Der Beweis verläuft analog zu Satz 1.9. □

Definition 3.4 (*Isomorphismen*)

Ein Monoidmorphismus $\iota : M \to N$ heißt **Isomorphismus**, wenn er bijektiv ist.
M und N sind **isomorph**, wenn ein Isomorphismus $\iota : M \to N$ existeiert, wir schreiben dann $M \cong N$.

Korrolar 3.3.1

Wenn $\iota : M \to N$ ein Isomorphismus ist, dann auch $\iota^{-1} : N \to M$.

Definition 3.5 (*Untermonoide*)

Sei M ein Monoid. Eine Teilmenge $N \subseteq M$ heißt **Untermonoid** von M, wenn

(i) $\mathbf{1} \in N$

(ii) für alle $n, n' \in N$ ist $n \cdot n' \in N$

Insbesondere ist dann $(N, \cdot_N, \mathbf{1})$ selbst ein Monoid, wobei $\cdot_N : N \times N \to N$ die Einschränkung von $\cdot : M \times M \to M$ ist.

Proposition 3.4

Für jeden Monoidmorphismus $h : M \to N$ ist das Bild

$$h[M] := \left\{ h(m) \;\middle|\; m \in M \right\}$$

ein Untermonoid von N und

$$M/_{\equiv_h} \cong h[M], [m]_{\equiv_h} \mapsto h(m).$$

Wir nennen diesen Morphismus den kanonischen Morphismus.

Beweis: Der Beweis verläuft analog wie bei Automaten. □

Definition 3.6 (*Produktmonoid*)

Seien M und N Monoide. Das **Produktmonoid** von M und N ist gegeben durch

$$M \times N = \left\{ (m, n) \;\middle|\; m \in M, n \in N \right\}$$

mit Multiplikation $(m,n)\cdot(m',n') = (m \cdot_M m', n \cdot_N n')$ und neutralem Element $(\mathbf{1}_M, \mathbf{1}_N)$.

3.2 Spracherkennung durch Monoide

Definition 3.7

Sei $L \subseteq \Sigma^*$ und M ein Monoid.

(1) Ein Monoidmorphismus $h : \Sigma^* \to M$ **erkennt** die Sprache L, wenn es eine Teilmenge $S \subseteq M$ gibt mit

$$L = h^{-1}[S] = \left\{ w \in \Sigma^* \;\middle|\; h(w) \in S \right\}.$$

(2) Das Monoid M **erkennt die Sprache L**, wenn es einen Monoidmorphismus $h : \Sigma^* \to M$ gibt, der L erkennt.

Wir werden zeigen, dass die regulären Sprachen genau den durch endliche Monoide erkennbare Sprachen entsprechen. Davor betrachten wir noch **Beispiel 3.3:**

1. Sei $L = (aa)^*$ und $h : \{a\}^* \to \mathbb{Z}_2$ der eindeutige Morphismus $h(a) = 1$. Dann gilt:

   $$h(a^n) = n \mod 2$$

   und damit $L = h^{-1}[0]$. Somit erkennt h die Sprache L.

2. Sei $L = \left\{ w \in \{a, b\}^* \;\middle|\; w \text{ endet nicht mit } a \right\}$. Betrachte das Monoid

   $$U_2 = \{1, u_1, u_2\} \qquad \text{mit } u_i \cdot u_j = u_j \quad i,j \in \{1,2\}.$$

   Wir betrachten nun den Morphismus $h : \{a,b\}^* \to U_2,\ a \mapsto u_1,\ b \mapsto u_2$. Dann gilt: $h(\varepsilon) = 1,\ h(wa) = u_1,\ h(wb) = u_2$ für alle $w \in \{a,b\}^*$. Also ist

   $$L = h^{-1}\left[\{1, u_2\}\right].$$

   Also erkennt h die Sprache L.

3. Jeder Morphismus $h : \Sigma^* \to M$ erkennt die Sprachen $\emptyset$ und Σ^* durch

   $$\emptyset = h^{-1}[\emptyset] \qquad \text{und} \qquad \Sigma^* = h^{-1}[M].$$

3.2.1 Von endlichen Monoiden zu endlichen Automaten

Proposition 3.5

Jede von einem endlichen Monoid erkannte Sprache ist regulär.

Beweis: Sei $L \subseteq \Sigma^*$ und M ein endlicher Monoid, der L erkennt. Dann gibt es einen Morphismus $h : \Sigma^* \to M$ und eine Teilmenge $S \subseteq M$ mit

$$L = h^{-1}[S].$$

Wir können dann einen DEA A für L wie folgt konstruieren:

- Zustandsmenge M
- Alphabet Σ
- Übergänge $m.a = m \cdot h(a)$ für alle $m \in M, a \in \Sigma$.
- Initialzustand $\mathbf{1}$ und
- Finalzustände $S \subseteq M$.

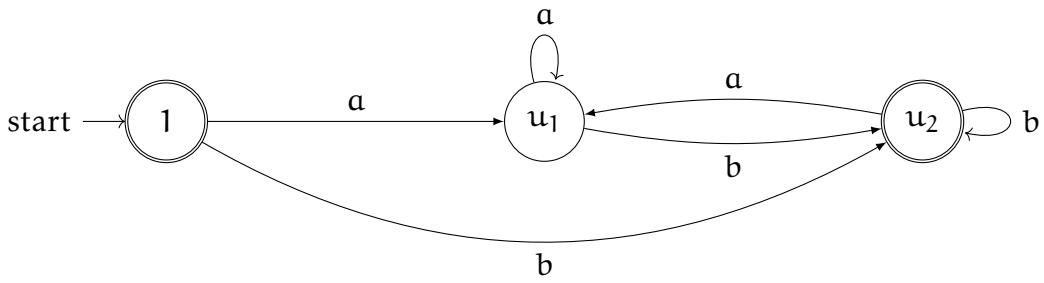
Es gilt $L(A) = L$, denn für alle Wörter $w = a_1 \dots a_n \in \Sigma^*$ ist

$$\begin{aligned} w \in L(A) &\Leftrightarrow \mathbf{1}.w \in S \\ &\Leftrightarrow \mathbf{1}.h(a_1 \dots a_n) = \mathbf{1} \cdot h(a_1) \cdot \dots \cdot h(a_n) \in S \\ &\Leftrightarrow h(w) \in S \\ &\Leftrightarrow w \in L = h^{-1}[S]. \end{aligned}$$

□

Betrachte nun weiter **Beispiel 3.3** (*Fortsetzung*)**:**

2. Wir können nun einfach einen endlichen Automaten konstruieren:



※

3.2.2 Von endlichen Automaten zu endlichen Monoiden

Sei $A = (Q, \Sigma, \to, I, F)$ ein NEA. Für jedes $w \in \Sigma^*$ betrachte die Relation $R_w \subseteq Q \times Q$ mit

$$(q, q') \in R_2 \Leftrightarrow q \xrightarrow{w} q'.$$

Dann gilt offensichtlich

$$R_\varepsilon = \mathrm{id}_Q \qquad \text{und} \qquad R_{vw} = R_v \cdot R_w. \tag{3.1}$$

Also ist $\mathrm{Tr}(A) = \{R_w \ : \ w \in \Sigma^*\}$ ein Untermonoid von $\mathfrak{P}(Q \times Q)$, dem Monoid aller binären Relationen auf Q. Wir definieren:

Definition 3.8

$\mathrm{Tr}(A)$ ist das **Transitionsmonoid** von A.

Proposition 3.6

Jede reguläre Sprache wird von einem endlichen Monoid erkannt.

Beweis: Sei $L \subseteq \Sigma^*$ regulär und A ein NEA mit $L(A) = L$. Wir zeigen, dass $\mathrm{Tr}(A)$ die Sprache L erkennt. Betrachte dazu den Morphismus

$$h : \Sigma^* \to \mathrm{Tr}(A),\ w \mapsto R_w.$$

h ist ein Morphismus wegen der Eigenschaften (3.1) und es existiert eine Menge $S \subseteq \mathrm{Tr}(A)$ mit

$$R_w \in S \Leftrightarrow w \in L.$$

Dies ist wohldefiniert, denn wenn $R_w = R_v$ gilt und $w \in L$, so gibt es auch ein $q_0 \in I$und ein $q \in F$ mit $(q_0, q) \in R_w = R_v$. Damit ist auch $v \in L(A) = L$. Daraus folgt dann

$$L = h^{-1}[S],$$

denn für alle $w \in \Sigma^*$ ist

$$w \in L \Leftrightarrow R_w \in S \Leftrightarrow h(w) \in S \Leftrightarrow w \in h^{-1}[S].$$

Also erkennt h und damit auch $\mathrm{Tr}(A)$ die Sprache L. □
Zusammengefasst ergibt sich der Satz

Satz 3.7

Eine Sprache $L \subseteq \Sigma^*$ ist genau dann regulär, wenn es ein endliches Monoid gibt, welches L erkennt.

Beweis: Folgt aus den Propositionen 3.5 und 3.6. □

Proposition 3.8

Sei $L \subseteq \Sigma^*$ und $h : \Sigma^* \to M$ ein Monoidmorphismus, welcher L erkennt. Dann erkennt der surjektive Morphismus

$$\widetilde{h} : \Sigma^* \twoheadrightarrow h[\Sigma^*], w \mapsto h(w)$$

die Sprache L.

Beweis: Sei $S \subseteq M$ mit $L = h^{-1}[S]$. Dann ist

$$L = h^{-1}\Big[\underbrace{S \cap h[\Sigma^*]}_{\subseteq h[\Sigma^*]}\Big] = \widetilde{h}^{-1}[S \cap h[\Sigma^*]].$$

□

Proposition 3.9

Jeder Monoidmorphismus $h : \Sigma^* \to M$, der die Sprache $L \subseteq \Sigma^*$ erkennt, erkennt auch $\overline{L} = \Sigma^* \backslash L$.

Beweis: Sei $S \subseteq M$ mit $L = h^{-1}[S]$, dann ist $\overline{L} = h^{-1}[M \backslash S]$. □

Proposition 3.10

Seien $K, L \subseteq \Sigma^*$ Sprachen, die von den Monoiden M beziehungsweise N erkannt werden. Dann erkennt das Produktmonoid $M \times N$ die Sprachen

$$K \cap L \qquad \text{und} \qquad K \cup L.$$

Beweis: Wähle die Morphismen $g : \Sigma^* \to M$ und $h : \Sigma^* \to N$ und die Teilmengen $S \subseteq M$ und $T \subseteq N$ mit

$$K = g^{-1}[S] \qquad \text{und} \qquad L = h^{-1}[T].$$

Betrachte dann den Morphismus

$$\langle g, h \rangle : \Sigma^* \to M \times N, w \mapsto (g(w), h(w)).$$

Dann gilt

$$K \cap L = \langle g, h \rangle^{-1}[S \times T] \qquad \text{und} \qquad K \cup L = \langle g, h \rangle^{-1}[S \times N \cup M \times T].$$

Also erkennt $\langle g, h \rangle$ und damit $M \times N$ die Sprachen $K \cup L$ und $K \cap L$. □

3.3 Das syntaktische Monoid einer Sprache

Das syntaktische Monoid einer Sprache ist das algebraische Gegenstück zum Minimalautomaten. Wir wollen uns wieder an

Definition 1.15 (*Nerode-Äquivalenz*)

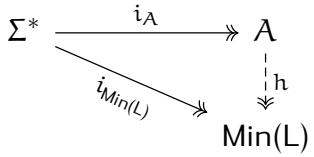
Sei $L \subseteq \Sigma^*$ eine Sprache. Die **Nerode-Äquivalenz** von L ist die Äquivalenzrelation $\sim_L \subseteq \Sigma^* \times \Sigma^*$ mit

$$v \sim_L w \quad :\Leftrightarrow \quad (\forall x \in \Sigma^*. vx \in L \Leftrightarrow wx \in L)$$

sowie die charakteristischen Eigenschaften des Minimalautomatens wie

$$\mathsf{Min}(L) = {}^{\Sigma^*}/_{\sim_L} \qquad \text{, wobei } \Sigma^* \text{ der Initialautomat von } L \text{ ist}$$

und der universellen Eigenschaft:



erinnern, um damit ein algebraisches Äquivalent zu finden. Wir definieren

Definition 3.9 (*syntaktische Kongruenz*)

Sei $L \subseteq \Sigma^*$. Die **syntaktische Kongruenz** für L ist die Äquivalenzrelation $\equiv_L \subseteq \Sigma^* \times \Sigma^*$ mit

$$v \equiv_L w \Leftrightarrow (\forall x, y \in \Sigma^*. xvy \in L \Leftrightarrow xwy \in L).$$

Lemma 3.11

$\equiv_L$ ist eine Monoidkongruenz auf Σ^*.

Beweis: Sei $v \equiv_L v'$ und $W \equiv_L w'$. Dann ist $vw \equiv v'w'$, denn für alle $x, y \in \Sigma^*$ gilt

$$xvwy \in L \overset{v \equiv_L v'}{\Longleftrightarrow} xv'wy \in L \overset{w \equiv_L w'}{\Longleftrightarrow} xv'w'y \in L.$$

□

Definition 3.10

Das **syntaktische Monoid** einer Sprache $L \subseteq \Sigma^*$ ist das Quotientenmonoid

$$\mathsf{Syn}(L) = {}^{\Sigma^*}/_{\equiv_L}.$$

Der Morphismus

$$h_L : \Sigma^* \twoheadrightarrow \mathsf{Syn}(L), w \mapsto [w]_{\equiv_L}$$

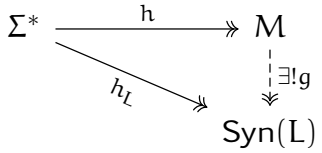
ist der **syntaktische Morphismus** von L.

Satz 3.12 (*Universelle Eigenschaft von* $\mathsf{Syn}(L)$)

Sei $L \subseteq \Sigma^*$ eine Sprache, so gilt:

(1) Der syntaktische Morphismus $h_L : \Sigma^* \twoheadrightarrow \mathsf{Syn}(L)$ erkennt L.

(2) Für jeden surjektiven Morphismus $h \ \Sigma^* \twoheadrightarrow M$, der L erkennt, gibt es genau einen Morphismus $g : M \to \mathsf{Syn}(L)$ mit $h_L = g \circ h$.



Beweis:

ad (1) Sei $S_L := \left\{ [w]_{\equiv_L} : w \in L \right\} \subseteq \mathsf{Syn}(L)$. Dann ist $L = h^{-1}[S_L]$, denn für alle $w \in \Sigma^*$ gilt

$$w \in h^{-1}[S_L] \Leftrightarrow h_L(w) \in S_L \Leftrightarrow [w]_{\equiv_L} \in S_L \Leftrightarrow w \in L.$$

Also erkennt h_L die Sprache L.

ad (2) Sei $h : \Sigma^* \twoheadrightarrow M$ ein surjektiver Morphismus, der L erkennt, also $L = h^{-1}[S]$ für ein $S \subseteq M$. Nach dem Homomorphiesatz ist nun für alle $v, w \in \Sigma^*$ zu zeigen, dass $h(v) = h(w) \Rightarrow h_L(v) = h_L(w)$ oder in anderen Worten $v \equiv_L w$.
Sei nun also $h(v) = h(w)$ und $x, y \in \Sigma^*$. Dann gilt $xvy \in L \Leftrightarrow xwy \in L$, denn

$$\begin{aligned} xvy \in L &\Leftrightarrow h(xvy) \in S \\ &\Leftrightarrow h(x)h(v)h(y) = h(x)h(w)h(y) \in S \\ &\Leftrightarrow h(xwy) \in S \\ &\Leftrightarrow xwy \in L, \end{aligned}$$

womit $v \equiv_L w$ gilt. □

Korrolar 3.12.1

Eine Sprache L ist genau dann regulär, wenn $\mathsf{Syn}(L)$ endlich ist, also wenn $\equiv_L$ nur endlich viele Kongruenzklassen hat.

Satz 3.13

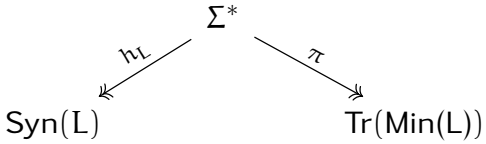
Das syntaktische Monoid einer Sprache $L \subseteq \Sigma^*$ ist der Transitionsmonoid des Minimalautomaten für L:

$$\mathsf{Syn}(L) \cong \mathsf{Tr}(\mathsf{Min}(L))$$

Beweis: $\mathsf{Min}(L) = \Sigma^*/_{\sim_L}$ hat die Transitionsfunktion

$$R_w : \Sigma^*/_{\sim_L} \to \Sigma^*/_{\sim_L}, [v]_{\sim_L} \mapsto [vw]_{\sim_L}.$$

Betrachte dann die surjektiven Morphismen



Nach dem Homomorphiesatz genügt es nun zu zeigen, dass h_L und π denselben Kern haben, also dass für alle $v, w \in \Sigma^*$ gilt

$$\underbrace{h_L(v) = h_L(w)}_{v \equiv_L w} \Leftrightarrow \underbrace{\pi(v) = \pi(w)}_{R_v = R_w}.$$

Das folgt aus

$$\begin{aligned} v \equiv_L w &\Leftrightarrow (\forall x, y \in \Sigma^*.\ xvy \in L \Leftrightarrow xwy \in L) \\ &\Leftrightarrow \forall x \in \Sigma^*.\ xv \sim_L xw \\ &\Leftrightarrow \forall x \in \Sigma^*.\ R_v([x]_{\sim_L}) = R_w([x]_{\sim_L}) \\ &\Leftrightarrow R_v = R_w. \end{aligned}$$

□

3.4 Eigenschaften von Monoiden

Unser Ziel ist es nun eine algebraische Charakterisierung von Spracheeigenschaften der Form

„Eine reguläre Sprache L hat die Eigenschaft P $\Longleftrightarrow$ Syn(L) hat die Eigenschaft Q“

zu finden. Aus einer solchen Äquivalenz folgt dann, dass wenn Q eine entscheidbare Eigenschaft von endlichen Monoiden ist, dann auch entscheidbar ist, ob ein gegebener DEA A mit Sprache $L(A)$ die Eigenschaft P hat. Dazu verwendet man den Algorithmus:

Algorithmus 3.1

Schritt 1 Berechne Min(L(A)).

Schritt 2 Berechne das Transitionsmonoid des Minimalautomaten ($\cong$ Syn(L))

Schritt 3 Prüfe, ob dieses Monoid die Eigenschaft Q hat.

Wir beschäftigen uns nun also mit verscheidenen Eigenschaften von Monoiden und ihren zugehörigen Spracheigenschaften. Zur **Notation**: Für $x \in M$ und $n \geqslant 1$ schreibe

$$x^n = \underbrace{x \cdot \cdots \cdot x}_{n-\text{mal}} = \begin{cases} x \cdot x^{n-1} & \text{, falls } n > 1 \\ x & \text{sonst} \end{cases}$$

für die **Potenzen von x**.

Definition 3.11

Ein Element x eines Monoids M ist **idempotent**, wenn $x^2 = x$.

Proposition 3.14

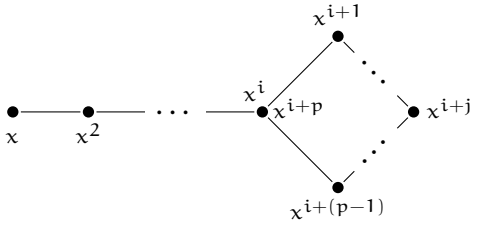
Jedes Element x eines **endlichen** Monoids M hat eine eindeutige idempotente Potenz x^ω.

Beweis: Wir zeigen zunächst Existenz, danach die Eindeutigkeit einer solchen Potenz:
Existenz:

Betrachte die Liste aller Potenzen von x:

$$x = x^1 \, , \, x^2 \, , \, x^3 \, , \, \ldots$$

Weil M endlich ist, gibt es $i, p \geqslant 1$ mit $x^i = x^{i+p}$.



Daraus folgt dann unmittelbar

$$x^k = x^{k+p} \quad \text{für alle } k \geqslant i. \tag{3.2}$$

Wähle dann $k := i \cdot p$, dann ist x^k idempotent, denn

$$\left(x^k\right)^2 = x^{k+k} = x^{k+i\cdot p} \overset{(3.2)}{=} x^k,$$

wobei der Zusammenhang aus Gleichung (3.2) im letzten Schritt i-mal angewendet wurde.

Eindeutigkeit:

Seien x^k und x^l mit $k, l \geqslant 1$ zwei idempotente Potenzen von x, so gilt

$$x^k = \left(x^k\right)^l = \left(x^l\right)^k = x^l.$$

□

Definition 3.12

Ein Monoid M ist ...

- ... **kommutativ** genau dann, wenn $x \cdot y = y \cdot x$ für alle $x, y \in M$ gilt.
- ... **idempotent** genau dann, wenn $x^2 = x$ für alle $x \in M$ gilt.
- ... **aperiodisch** genau dann, wenn es für alle $x \in M$ ein $n \geqslant 1$ gibt mit $x^{n+1} = x^n$.

$$x - x^2 - x^3 - \cdots - x^n \supset$$

- ... eine **Gruppe**, wenn es für alle $x \in M$ ein $x^{-1} \in M$ gibt mit

$$x \cdot x^{-1} = \mathbf{1} \qquad \text{und} \qquad x^{-1} \cdot x = \mathbf{1}.$$

Anmerkungen

Wenn $x^{n+1} = x^n$ für ein $x \geqslant 1$, dann ist auch $x^{m+1} = x^m$ für alle $m \geqslant n$.
Folgerung: Für endliche Monoide gilt:

$$\left(\forall\, x.\ \exists\, n.\ x^{n+1} = x^n\right) \Leftrightarrow \left(\exists\, n.\ \forall\, x.\ x^{n+1} = x^n\right)$$

Man kann also die Zahl n in der Definition eines endlichen aperiodischen Monoids für alle x gleich wählen.

Jedes idempotente Monoid ist aperiodisch mit $n = 1$.

Jede aperiodische Gruppe ist trivial, denn aus $x^{n+1} = x^n$ folgt $x = \mathbf{1}$.

Anmerkungen (*fort.*)

Bei Gruppen reicht es aus nur Linksinverse oder nur Rechtsinverse zu fordern, da die Existenz und Gleichheit des jeweilig anderen direkt folgt. Am Beispiel Rechtsinverse implizieren Linksinverse:
Für jedes $x \in M$ gilt, dass

$$x^{-1} = x^{-1} \cdot e = x^{-1}\left(xx^{-1}\right) = \left(x^{-1}x\right)x^{-1},$$

womit direkt folgt

$$e = x^{-1}\left(x^{-1}\right)^{-1} = \left(\left(x^{-1}x\right)x^{-1}\right)\left(x^{-1}\right)^{-1} = \left(x^{-1}x\right)\left(x^{-1}\left(x^{-1}\right)^{-1}\right) = \left(x^{-1}x\right)e = x^{-1}x.$$

Damit ist x^{-1} sowohl Rechts- wie auch Linksinverses. Die andere Richtung ist analog zu zeigen.

Für **endliche** Monoide wird jede der vier Eigenschaften aus Definition 3.12 auf **Untermonoide**, **homomorphe Bilder** und **Produkte** vererbt.

Mit dieser Erkenntnis definieren wir genauer:

Definition 3.13 (*Varietät*)

Eine **Varietät** von endlichen Monoiden ist eine *nichtleere* Klasse $\mathbb{V}$ von endlichen Monoiden, die unter Untermonoiden, homomorphen Bildern und Produkten abgeschlossen ist. Mit anderen Worten gilt:

(i) Für $M \in \mathbb{V}$ und $N \subseteq M$ Untermonoid folgt automatisch $N \in \mathbb{V}$.

(ii) Für alle $M \in \mathbb{V}$ und alle surjektiven Morphismen $e : M \to N$ gilt $N \in \mathbb{V}$.

(iii) Gilt $M, N \in \mathbb{V}$, so auch $M \times N \in \mathbb{V}$.

Lemma 3.15

Die folgenden Klassen von *endlichen* Monoiden bilden Varietäten:

- $\mathbb{V}$ = kommutative Monoide
- $\mathbb{V}$ = aperiodische Monoide
- $\mathbb{V}$ = idempotente Monoide
- $\mathbb{V}$ = (*endliche*) Gruppen
- alle Klassen von endlichen Monoiden, die durch Kombination dieser Varietäten entstehen (*Beispiel*: $\mathbb{V}$ = kommutative und aperiodische Monoide)

Beweis: Wir zeigen exemplarisch die Aussage nur für $\mathbb{V}$ = aperiodische Monoide, die restlichen Aussagen sind analog zu zeigen:

ad (i) Sei M aperiodisch und $N \subseteq M$ ein Untermonoid, sowie $x \in N \subseteq M$. Dann gibt es ein $n \geqslant 1$ mit $x^{n+1} = x^n$ in M und damit auch in N.

ad (ii) Sei M aperiodisch und $e : M \twoheadrightarrow N$ ein surjektiver Morphismus, sowie $x \in N$. Wähle dann $y \in M$ mit $e(y) = x$. Dann gibt es ein $n \geqslant 1$ mit $y^{n+1} = y^n$. Daraus schließen wir

$$x^{n+1} = (e(y))^{n+1} = e\left(y^{n+1}\right) = e(y^n) = (e(y))^n = x^n.$$

ad (iii) Seien M, N aperiodisch und $(x, y) \in M \times N$. Wähle ein $n \geqslant 1$ mit

$$x^{n+1} = x^n \quad \text{und } y^{n+1} = y^n.$$

Dann gilt allerdings auch

$$(x, y)^{n+1} = \left(x^{n+1}, y^{n+1}\right) = (x^n, y^n) = (x, y)^n.$$

Damit bilden die aperiodischen endlichen Monoide eine Varietät. □

Proposition 3.16

Sei $\mathbb{V}$ eine Varietät und $L \subseteq \Sigma^*$. Wenn es ein Monoid in $\mathbb{V}$ gibt, das L erkennt, dann liegt $\mathsf{Syn}(L)$ in $\mathbb{V}$.

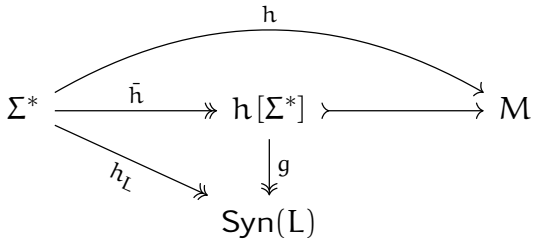
Beweis: Sei $M \in \mathbb{V}$ und $h : \Sigma^* \to M$ ein Morphismus, der L erkennt. Dann erkennt auch der surjektive Morphismus

$$\bar{h} : \Sigma^* \twoheadrightarrow h[\Sigma^*], w \mapsto h(w)$$

die Sprache L. Nach der universellen Eigenschaft von $\mathsf{Syn}(L)$ (Satz 3.12) gibt es dann einen surjektiven Morphismus

$$g : h[\Sigma^*] \twoheadrightarrow \mathsf{Syn}(L)$$

mit $g \circ \bar{h} = h_L$. Weil $\mathbb{V}$ unter Untermonoiden und homomorphen Bildern abgeschlossen ist folgt dann $\mathsf{Syn}(L) \in \mathbb{V}$.
In Diagrammform:



□

Proposition 3.17

Seien M, N endliche Monoide.

(1) Wenn $N \subseteq M$ ein Untermonoid ist, dann wird jede von N erkannte Sprache auch von M erkannt.

(2) Wenn $e : M \twoheadrightarrow N$ ein surjektiver Morphismus ist, dann wird jede von N erkannte Sprache auch von M erkannt.

(3) Jede von $M \times N$ erkannte Sprache ist eine bool'sche Kombination von Sprachen, die von M oder N erkannt werden.

Beweis:

ad (1) Sei $N \subseteq M$ und $L \subseteq \Sigma^*$ sei eine von N erkannte Sprache. Es gibt also einen Morphismus $h : \Sigma^* \to N$ und $S \subseteq N$ mit $L = h^{-1}[S]$. Betrachte dann den Einbettungsmorphismus $i : N \rightarrowtail M, n \mapsto n$. Dann ist $g = i \circ h : \Sigma^* \to M$ ein Morphismus und $S \subseteq M$, sowie $L = g^{-1}[S]$. Daraus folgt unmittelbar die Erkennbarkeit von L durch M.

ad (2) siehe Übungsblatt 5, Aufgabe 3

ad (3) Sei $L \subseteq \Sigma^*$ eine Sprache, die von $M \times N$ erkannt wird, das heißt es gibt ein $h : \Sigma^* \to M \times N$ und $S \subseteq M \times N$ mit $L = h^{-1}[S]$. Betrachte die Projektionsmorphismen

$$\pi_M : M \times N \to M, (m,n) \mapsto m \quad \text{und} \quad \pi_N : M \times N \to N, (m,n) \mapsto n.$$

Definiere dann

$$h_M = \pi_M \circ h : \Sigma^* \to M \quad \text{und} \quad h_n = \pi_N \circ h : \Sigma^* \to N.$$

Dann gilt $h(w) = (h_M(w), h_N(w))$ für $w \in \Sigma^*$ und damit

$$L = h^{-1}[S] = \bigcup_{(m,n)\in S} h^{-1}[(m,n)] = \bigcup_{(m,n)\in S} \left(h_M^{-1}[m] \cap h_N^{-1}[n] \right).$$

□

3.4.1 Kommutative Gruppen

Für jedes $n \geqslant 2$ ist $\mathbb{Z}_n = \{0, \ldots, n-1\}$ eine kommutative Gruppe.

Satz 3.18 (*Hauptsatz endlicher abelscher Gruppen*)

Für jede endliche kommutative Gruppe G gibt es Zahlen $n_1, \ldots, n_k \geqslant 2$ mit

$$G \cong \mathbb{Z}_{n_1} \times \cdots \times \mathbb{Z}_{n_k}.$$

Fixiere nun Σ. Für $a \in \Sigma, n \geqslant 2$ und $0 \geqslant k < n$ definiere

$$L_{a,k,n} = \left\{ w \in \Sigma^* \;\middle|\; |w|_a = k \mod n \right\}.$$

Satz 3.19

Für $L \subseteq \Sigma^*$ sind äquivalent:

(1) L ist eine boolesche Kombination von Sprachen der Form $L_{a,k,n}$.

(2) Es gibt eine endliche kommutative Gruppe, die L erkennt.

(3) Syn(L) ist eine endliche kommutative Gruppe.

Beweis:

„(2) $\Leftrightarrow$ (3)“ gilt, weil die endlichen kommutativen Gruppen eine Varietät bilden.

„(1) $\Rightarrow$ (2)“ Es genügt zu zeigen, dass jede Sprache $L = L_{a,k,n}$ durch ein endliches kommutatives Monoid erkannt wird, die allgemeine Aussage folgt dann durch Bildung von Produkten.
$L_{a,k,n}$ wird durch $\mathbb{Z}_n$ erkannt. Für den Morphismus

$$h : \Sigma^* \to \mathbb{Z}_n \qquad \text{mit } h(a) = 1 \text{ und } h(b) = 0 \text{ für } b \neq a$$

gilt $h(w) = |w|_a \mod n$, womit automatisch folgt, dass $L_{a,k,n} = h^{-1}[k]$.

„(1) $\Leftarrow$ (2)“ Sei G eine endliche kommutative Gruppe, die L erkennt, das heißt es gibt $h : \Sigma^* \to G$ und $S \subseteq G$ mit $L = h^{-1}[S]$. Nach dem Hauptsatz (Satz 3.18) können wir $G = \mathbb{Z}_n$ annehmen. Sei dann $\Sigma = \{a_1, \ldots, a_m\}$ und $k_i := h(a_i) \in \mathbb{Z}_n$ für $i = 1, \ldots, m$. Definiere dann

$$E = \Big\{(r_1, \ldots, r_m) \;\Big|\; r_i \in \mathbb{Z}_n \text{ und } \sum_{i=1}^{m} r_i \cdot k_i \in S\Big\}.$$

Für $(r_1, \ldots, r_m) \in E$ definiere weiter

$$L_{r_1, \ldots, r_m, n} = \bigcap_{i=1}^{m} L_{a_i, r_i, n} = \big\{w \in \Sigma^* \;\big|\; |w|_{a_i} = r_i \mod n, i = 1, \ldots, m\big\}.$$

Dann gilt:

$$L = \bigcup_{(r_1, \ldots, r_m) \in E} L_{r_1, \ldots, r_m, n} \tag{3.3}$$

Zeige nun (3.3): Für $w \in \Sigma^*$ gilt

$$h(w) = \sum_{i=1}^{m} |w|_{a_i} \cdot \underbrace{h(a_i)}_{=k_i}.$$

Also für beliebiges $w \in L$ gilt:

$$\begin{aligned} w \in L &\Leftrightarrow h(w) \in S \\ &\Leftrightarrow \sum_{i=1}^{m} |w|_{a_i} \cdot k_i \in S \\ &\Leftrightarrow \left(|w|_{a_1}, \ldots, |w|_{a_n}\right) \in E \\ &\Leftrightarrow w \in \bigcup_{(r_1, \ldots, r_m) \in E} L_{r_1, \ldots, r_m, n} \end{aligned}$$

□

3.4.2 Kommutative idempotente Monoide

Definition 3.14

Eine Sprache $L \subseteq \Sigma^*$ heißt **alphabetisch**, wenn sie eine boolesche Kombination von Sprachen der Form Δ^* mit $\Delta^* \subseteq \Sigma^*$ ist.

Beispiel 3.4: Sei $\Sigma = \{a, b, c\}$. Die Sprache

$$L = \{w \in \Sigma^* \mid w \text{ enthält ein } a \text{ oder kein } c\}$$

ist alphabetisch, denn

$$L = \overline{\{b, c\}^*} \cup \{a, b\}^*.$$

//

Definition 3.15

Eine **FO^1**-Formel ist eine FO-Formel, in der nur eine Variable (eventuell mehrfach) vorkommt.

Beispiel 3.4 (*Fortsetzung*)**:** L wird durch folgende FO^1-Formel definiert:

$$\exists x.\, P_a(x) \vee \neg \exists x.\, P_c(x)$$

※

Satz 3.20

Für jede reguläre Sprache $L \subseteq \Sigma^*$ sind folgende Aussagen äquivalent:

1. L ist alphabetisch.
2. Es gibt eine FO^1-Formel φ mit $L = L(\varphi)$.
3. Es gibt ein endliches kommutatives idempotentes Monoid, das L erkennt.
4. Syn(L) ist kommutativ und idempotent.

Beweis:

„3 ⇔ 4“ gilt, weil die endlichen kommutativen und idempotenten Monoide eine Varietät bilden.

„1 ⇒ 2“ Jede Sprache Δ^* mit $\Delta \subseteq \Sigma$ ist durch eine FO^1-Formel definierbar:

$$\varphi_\Delta = \forall x.\, \bigvee_{a \in \Delta} P_a(x).$$

Eine alphabetische Sprache ist eine boolesche Kombination von Δ^*s und wird damit durch eine boolesche Kombination von φ_Δs beschrieben.

„1 ⇐ 2“ Sei φ eine geschlossene FO^1-Formel mit der einzigen Variable x. Jede atomare Unterformel von φ hat die Form

$$P_a(x) \qquad \text{oder} \qquad x < x \equiv \bot.$$

Also ist φ bis auf semantische Äquivalenz eine boolesche Kombination von Formeln der Form $\exists.\, P_a(x)$ und

$$L(\exists x.\, P_a(x)) = \overline{(\Sigma \setminus \{a\})^*}.$$

Also ist $L(\varphi)$ alphabetisch.

„(1) ⇒ (3)“ Sei L alphabetisch. Zu zeigen ist, das es ein endliches kommutatives idempotentes Monoid gibt, das L erkennt. Es genügt den Fall $L = \Delta^*$ mit $\Delta \subseteq \Sigma$ zu betrachten; der allgemeine Fall folgt durch Bildung von Produktonoiden.
Betrachte das kommutative idempotente Monoid $M = \{0, 1\}$ mit

$$\mathbf{1} \cdot \mathbf{1} = \mathbf{1} \quad \text{und } 0 \cdot \mathbf{1} = \mathbf{1} \cdot 0 = 0 \cdot 0 = 0,$$

und dem Morphismus

$$h : \Sigma^* \to M \quad \text{mit } h(a) = \begin{cases} 1 & \text{, falls } a \in \Delta \\ 0 & \text{, falls } a \in \Sigma \backslash \Delta \end{cases}.$$

Dann gilt für $w \in \Sigma^*$

$$h(w) = \begin{cases} 1 & \text{, falls } w \in \Delta^* \\ 0 & \text{, falls } w \in \Sigma^* \backslash \Delta^* \end{cases}.$$

Also gilt $L = h^{-1}[\mathbf{1}]$, das heißt M erkennt L.

„(1) ⇐ (3)“ Sei M ein endliches, kommutatives und idempotentes Monoid und $h : \Sigma^* \to M$ ein Morphismus, der L erkennt, also $L = h^{-1}[S]$ mit $S \subseteq M$.
Definiere für $w \in \Sigma^*$ $\alpha(w) = \{a \in \Sigma \mid a \text{ kommt in } w \text{ vor.}\}$. Dann gilt $h(w) = \prod_{a \in \alpha(w)} h(a)$.
Beispiel 3.5: $h(abbab) = h(a)h(b)h(b)h(a)h(b) = h(a)^2 h(b)^3 = h(a)h(b)$. ※
Definiere für $\Delta \subseteq \Sigma$:

$$L_\Delta = \{w \in \Sigma^* \mid \alpha(w) = \Delta\} = \Delta^* \cap \bigcap_{a \in \Delta} \overline{(\Delta \backslash \{a\})^*}.$$

Dann ist

$$L = \bigcup_{w \in L} L_{\alpha(w)}. \tag{3.4}$$

und damit alphabetisch. Zeige nun noch (3.4):

„⊆“ $V \in L$, woraus folgt, dass $v \in L_{\alpha(v)} \subseteq \bigcup_{w \in L} L_{\alpha(w)}$.
„⊇“ Sei $v \in L_{\alpha(w)}$ für ein $w \in L$. Dann gilt $\alpha(v) = \alpha(w)$ und damit

$$h(v) = \prod_{a \in \alpha(v)} h(a) = \prod_{a \in \alpha(w)} h(a) = h(w) \in S.$$

Also $v \in h^{-1}[S] = L$. □

3.4.3 Aperiodische Monoide, FO-Logik und sternfreie Sprachen

Definition 3.16 (*verallgemeinerte regulären Ausdrücke*)

Die **verallgemeinerten regulären Ausdrücke** über Σ entstehen aus den regulären Ausdrücken durch Hinzufügen des Komplementoperators:
Syntax: $r ::= \emptyset \mid \varepsilon \mid a \mid r + r \mid rr \mid \overline{r} \mid r^*$
Semantik: $L(\overline{r}) = \overline{L(r)}$ und alle anderen Operatoren werden wie zuvor interpretiert!

Definition 3.17

Eine Sprache ist **sternfrei**, wenn sie durch einen verallgemeinerten regulären Ausdruck darstellbar ist, in denen **kein** $(\dots)^*$ vorkommt.
Mit anderen Worten: Die sternfreien Sprachen über Σ bilden den Abschluss der Menge der endlichen Sprachen unter Vereinigung, Konkatenation und Komplement.

Beispiel 3.6: Sei $\Sigma = \{a, b\}$, dann gilt:

1. $(a+b)^*$ ist **sternfrei**, denn $(a+b)^* = \overline{\emptyset}$.
2. a^*b^* ist **sternfrei**, denn $a^*b^* = \overline{(a+b)^*ba(a+b)^*} = \overline{\overline{\emptyset}ba\overline{\emptyset}}$.
3. $(aa)^*$ ist **nicht sternfrei**, was aus Satz 3.21 folgt.

※

Satz 3.21 (*Satz von McNaughton, Papert und Schützenberger*)

Sei $L \subseteq \Sigma^*$ regulär, dann sind die folgenden Aussagen äquivalent:

(1) L ist sternfrei.

(2) Es gibt eine FO-definierbar.

(3) Es gibt ein endliches aperiodisches Monoid, das L erkennt.

(4) $\mathsf{Syn}(L)$ ist aperiodisch.

Beweis: Folgt unmittelbar aus Propositionen 3.22, 3.26, 3.27. □

Korrolar 3.21.1

Es ist entscheidbar, ob eine gegebene reguläre Sprache FO-definierbar oder sternfrei ist.

Einschub – Sternhierarchie

Definition 3.18

Die **Sternhierarchie** eines verallgemeinerten regulären Ausdrucks r ist die maximale Verschachtelungstiefe von Sternen in r. Formal definiert ist sie durch:

- $\mathrm{sh}(\emptyset) = \mathrm{sh}(\varepsilon) = \mathrm{sh}(a) = 0$
- $\mathrm{sh}(r^*) = \mathrm{sh}(r) + 1$
- $\mathrm{sh}(\overline{r}) = \mathrm{sh}(r)$
- $\mathrm{sh}(r+s) = \mathrm{sh}(rs) = \max\{\mathrm{sh}(r), \mathrm{sh}(s)\}$

Die Sternhöhe einer regulären Sprache L ist die minimale Sternhöhe eines verallgemeinerten regulären Ausdrucks für L.

Beispiel 3.7:

1. Die Sprachen der Sternhöhe 0 sind genau die sternfreien Sprachen.

2. Für $L = \{w \in \{a\}^* \mid |w| \text{ gerade}\}$ gilt $sh(L) = 1$.

   - Die Sternhöhe muss mindestens eins sein, denn $\{aa\}^*$ ist ein regulärer Ausdruck für L der Sternhöhe 1.
   - Die Sternhöhe kann andererseits auch nicht null sein, denn $Syn(L) = \mathbb{Z}_2$ ist eine nicht triviale Gruppe und damit nicht aperiodisch. Dementsprechend folgt mit Satz 3.21 die nicht Sternfreiheit von L.

Offenes Problem

Gibt es eine Sprache der Sternhöhe $\geqslant 2$?

※

Definition 3.19

Die **modifizierte Sternhöhe** $sh'(L)$ einer regulären Sprache L ist die minimale Sternhöhe eines regulären Ausdruckes für L.

Es gilt:

(1) $sh(L) \leqslant sh'(L)$

(2) Es gibt für jedes $n \geqslant 0$ eine Sprache der modifizierten Sternhöhe n. [Egg63]

(3) Die modifizierte Sternhöhe ist algorithmisch berechenbar. [Has88]

3.4.3.1 Von sternfreien zu FO-definierbaren Sprachen

Proposition 3.22

Jede sternfreie Sprache ist FO-definierbar.

Beweis: Für jeden sternfreien regulären Ausdruck r konstruieren wir induktiv eine FO-Formel φ_r mit $L(r) = L(\varphi_r)$:

- $\varphi_\emptyset = \exists x.\, x \wedge \neg \exists x.\, x = x$
- $\varphi_\varepsilon = \neg \exists x.\, x = x$
- $\varphi_a = \exists x.\, (\mathsf{first}(x) \wedge \mathsf{last}(x) \wedge P_a(X))$
- $\varphi_{r+s} = \varphi_r \vee \varphi_s$
- $\varphi_{\overline{r}} = \overline{\varphi_r}$
- φ_{rs} — Wegen

$$KL = (K \setminus \{\varepsilon\})(L \setminus \{\varepsilon\}) [\cup K] [\cup L]$$

(wobei der violette resp. grüne Teil nur in den Fällen von $\varepsilon \in L$ oder $\varepsilon \in K$ hinzukommen) dürfen wir annehmen, dass $L(r), L(s) \subseteq \Sigma^+$. Ein Wort $w = a_1 \dots a_n$ liegt genau dann in $L(rs) = L(r)L(s)$, wenn es eine Position $i \in \{1, \dots, n-1\}$ gibt mit

$$a_1, \dots, a_i \in L(r) \quad \text{und} \quad a_{i+1}, \dots, a_n \in L(s),$$

in anderen Worten, wenn es ein i gibt mit

$$a_1, \ldots, a_i \vDash \varphi_r \quad \text{und} \quad a_{i+1}, \ldots, a_n \vDash \varphi_s.$$

Konstruieren wir nun eine Formel für φ_{rs}: Sei φ eine FO-Formel und x eine Variable, die nicht in φ vorkommt. Sei $\varphi^{\leqslant x}$ die Formel, die aus φ entsteht, indem man jede Teilformel $\exists\, y.\ \psi$ durch

$$.\exists\, y.\ (y \leqslant x \wedge \psi)$$

ersetzt. Dann gilt für alle $w = a_1 \ldots a_n$ und alle Interpretationen I mit $I(x) = i$:

$$w,\ I \vDash \varphi^{\leqslant x} \quad \text{gdw.} \quad a_1, \ldots, a_i \vDash \varphi.$$

Analog definiert man nun $\varphi^{>x}$. Damit erhalten wir folgende Formel für $L(rs)$:

$$\varphi_{rs} = \exists\, x.\ \left[\neg \mathsf{last}(x) \wedge \varphi_r^{\leqslant x} \wedge \varphi_s^{>x}\right],$$

wobei x eine frische Variable ist.

□

3.4.3.2 Von aperiodischen Monoiden zu sternfreien Sprachen

Fixiere nun ein endliches aperiodisches Monoid M und einen Morphismus $h : \Sigma^* \to M$. Die von h erkannten Sprachen haben die Form

$$L = h^{-1}[S] = \bigcup_{m \in S} h^{-1}[m] \text{ mit } S \subseteq M.$$

Es genügt nun zu zeigen, dass für alle $m \in M$ $h^{-1}[m]$ sternfrei.

Lemma 3.23 (*Kürzungslemma*)

Für $m, p, q \in M$ gilt

$$m = pmq \Rightarrow m = pm \wedge m = mq.$$

Insbesondere gilt für $m = \mathbf{1}$: $pq = 1 \Rightarrow p = q = 1$.

Beweis: Sei $m = pmq$. Wähle dann $n \geqslant 1$ mit $p^{n+1} = p^n$. Dann gilt aber

$$m = pmq = p^2mq^2 = \cdots = p^nmq^n = p^{n+1}mq^n = pm.$$

Analog $m = m \cdot q$. □

Definition 3.20

Für $m \in M$ definiere ...

... das von m erzeugte **Rechtsideal**

$$mM = \{mn \mid n \in M\}$$

... das von m erzeugte **Linksideal**

$$Mm = \{nm \mid n \in M\}$$

... das von m erzeugte **Ideal**

$$MmM = \{nmp \mid n, p \in M\}$$

... die **Menge der Faktoren** von m

$$F(m) = \{n \in M \mid m \in MnM\}$$

Lemma 3.24 (*Ideallemma*)

Für alle $m \in M$ gilt

$$mM \cap Mm \cap F(m) = \{m\}.$$

Beweis:

„$\supseteq$“ Gilt wegen

$$m = m \cdot \mathbf{1} = \mathbf{1} \cdot m = \mathbf{1} \cdot m \cdot \mathbf{1}.$$

„$\subseteq$“ Sei $n \in (mM \cap Mm \cap F(m))$. Zeige dann, dass $n = m$. Wähle dazu p, q, r und $s \in M$ mit

$$n = m \cdot p = q \cdot m \qquad \text{und} \qquad m = rns.$$

Daraus folgt dann, dass $m = rqms$, womit mit Lemma 3.23 folgt, dass

$$m = r \cdot qm = r \cdot n = r \cdot m \cdot p.$$

Erneutes Anwenden von Lemma 3.23 liefert dann

$$m = m \cdot p = n.$$

□

Lemma 3.25 (*Darstellungslemma*)

Für $m \in M \backslash \{1\}$ gilt

$$h^{-1}[m] = (\mathfrak{U}\Sigma^* \cap \Sigma^*\mathfrak{V}) \backslash \Sigma^*\mathfrak{W}\Sigma^*$$

mit

$$\mathfrak{U} = \{w \in \Sigma^* \mid h(w)M = mM \text{ und } h(v)M \neq mM \text{ f.a. } \textbf{echten} \text{ Präfixe } v \text{ von } w\}$$

Beweis:

„$\supseteq$“ Sei $w \in h^{-1}[m]$. Zeige nun die Eigenschaften

$$\text{(i) } w \in \mathfrak{U}\Sigma^*, \qquad \text{(ii) } w \in \Sigma^*\mathfrak{V} \qquad \text{und} \qquad \text{(iii) } w \notin \Sigma^*\mathfrak{W}\Sigma^*$$

ad (i) Wähle das kürzeste Prefix uv von w mit $h(u)M = mM$. Dann ist $u \neq \varepsilon$, sonst wäre $M = mM$, also $\mathbf{1} = mn$ mit $n \in M$, womit mit dem Kürzungslemma (3.23) gölte, dass $m = \mathbf{1}$, was dann direkt zum Widerspruch zur Annahme, dass $m \neq \mathbf{1}$ sei, stünde. Also ist $u \in \mathfrak{U}$ und damit gilt $w \in \mathfrak{U}\Sigma^*$.

ad (ii) analog zum Beweis von (i)

ad (iii) Angenommen es gölte $w \in \Sigma^*\mathfrak{W}\Sigma^*$. Damit gilt dann aber auch $w = xw'y$ mit x, $y \in \Sigma^*$ und $w' \in \mathfrak{W}$. Dann ist aber $h(w') \notin F(m)$ nach Definition von w'. Andererseits gilt aber auch

$$m = h(w) = h(x)h(w')h(y),$$

also insbesondere $h(w') \in F(m)$, was zum Widerspruch führt.

„$\subseteq$“ Sei $w \in (\mathfrak{U}\Sigma^* \cap \Sigma^*\mathfrak{V})\setminus\Sigma^*\mathfrak{W}\Sigma^*$. Nach Ideallemma (Lemma 3.24) genügt es nun die Eigenschaften

$$\text{(i) } h(w) \in mM\,, \qquad \text{(ii) } h(w) \in Mm \qquad \text{und} \qquad \text{(iii) } h(w) \in F(m)$$

zu zeigen:

ad (i) Wegen der Tatsache, dass $w \in \mathfrak{U}\Sigma^*$ ist $w = ux$ mit $u \in \mathfrak{U}$ und $x \in \Sigma^*$. Daraus folgt dann unmittelbar

$$h(w) = h(u)h(x) \in h(u)M = mM.$$

ad (ii) analog zum Beweis von (i)

ad (iii) Angenommen es gölte $h(w) \notin F(m)$, das heißt $m \notin Mh(w)M$. Wähle dann einen kürzesten Faktor x von w mit

$$m \notin Mh(x)M.$$

Schreibe $w = uxv$ mit $u,\ v \in \Sigma^*$. Wir unterscheiden dann drei Fälle:

- Falls $x = \varepsilon$ gilt, ist $m \notin M\mathbf{1}M = MM = M$ „.
- Falls $x \in \Sigma$ gilt, ist $x \in \mathfrak{W}$, denn $h(x) \notin F(m)$ und für den einzigen echten Faktor ε von x gilt $\mathbf{1} = h(\varepsilon) \in F(m)$, was trivialerweise zum Widerspruch führt.
- Falls $|x| \geqslant 2$, dann schreibe $x = arb$ mit $a,\ b \in \Sigma^*$ und $r \in \Sigma^*$. Dann gilt aber $m \notin Mh(r)M$ und $m \in Mh(a)h(r)M$, sowie $m \in Mh(r)h(b)M$, da x minimal ist. Damit ist $x \in \mathfrak{W}$. Daraus folgte dann aber, dass $w \in \Sigma^*\mathfrak{W}\Sigma^*$, was aber im Widerspruch zur Annahme stünde.

□

Proposition 3.26

Für alle $m \in M$ ist $h^{-1}[m]$ sternfrei.

Beweis: Wir beweisen per Induktion nach $|M| - |MmM|$.

Induktionsanfang ($M = MmM$):

Gemäß dem Kürzungslemma (Lemma 3.23) gilt in diesem Fall $m = 1$. Wir stellen nun die Behauptung auf, dass $h^{-1}[\mathbf{1}] = A^*$, wobei $A = \{a \in \Sigma \mid h(a) = \mathbf{1}\}$ gilt. (Wohlgemerkt ist $h^{-1}[\mathbf{1}]$ sternfrei)

Dass dies gilt ist schnell einzusehen. Die Richtung „$\supseteq$“ ist klar, weil h ein Morphismus ist. Für die Gegenrichtung („$\subseteq$“) gilt für ein $w = a_1 \dots a_n \in \Sigma^*$ mit $h(w) = \mathbf{1}$, dass $h(a_1)\dots h(a_n) = \mathbf{1}$ und somit per Kürzungslemma (Lemma 3.23) $h(a_1) = \dots = h(a_n) = \mathbf{1}$.

Induktionsschritt:

Sei nun $MmM \subseteq M$ und damit insbesondere $m \neq \mathbf{1}$. Wegen $h^{-1}[m] = (\mathfrak{U}\Sigma^* \cap \Sigma^*\mathfrak{V})\setminus\Sigma^*\mathfrak{W}\Sigma^*$ (aus dem Darstellungslemma 3.25) genügt es zu zeigen, dass $\mathfrak{U}$, $\mathfrak{V}$ und $\mathfrak{W}$ sternfrei sind.

ad $\mathfrak{U}$: Es gilt

$$\mathfrak{U} = \bigcup_{(n,a)\in I} h^{-1}[n]a \qquad \text{mit } I = \left\{ (n,a) \in M \times \Sigma \;\middle|\; nh(a)M = mM \text{ und } n \notin mM \right\}.$$

Man stellt nun fest, dass für alle $(n, a) \in M \times \Sigma$ gilt, dass $MmM \subsetneqq MnM$, woraus die Sternfreiheit von $h^{-1}[n]$ für alle $(n, a) \in I$, und somit auch von $\mathfrak{U}$ folgt. Dies macht man sich klar, da einerseits mit $(n, a) \in I$ beliebig

$$mM = nh(a)M \subseteq nM$$

gilt und somit auch $MmM \subseteq MnM$. Andererseits nehme man an, dass $MmM = MnM$, womit $n \in MmM$, oder in anderen Worten, dass $n = umv$ mit $u, v \in M$. Wegen $mM \subseteq nM$ ist $m = np$ mit $p \in M$. Also gilt $n = unpv$, womit nach Kürzungslemma (Lemma 3.23) folgt, dass

$$n = npv = mv,$$

was im Widerspruch zu $n \notin mM$ steht.

ad $\mathfrak{V}$: Man beweist diesen Fall analog wie bei $\mathfrak{U}$.

ad $\mathfrak{W}$: Es gilt

$$\mathfrak{W} = \bigcup_{(a,n,b)\in J} ah^{-1}[n]b \cup (\mathfrak{W} \cap \Sigma)$$

$$\text{mit } J = \left\{(a, n, b) \in \Sigma \times M \times \Sigma \;\middle|\; h(a) \cap h(b) \notin F(m) \text{ und } h(a)n \in F(m) \text{ und } nh(b) \in F(m)\right\}.$$

Wir stellen auch hier wieder fest, dass für alle $(a, n, b) \in J$ $MmM \subsetneqq MnM$, womit $h^{-1}[n]$ wieder sternfrei per Induktion für alle $(a, n, b) \in J$, womit $\mathfrak{W}$ sternfrei ist. Denn sei einerseits $(a, n, b) \in J$, so gilt wegen $n \in F(m)$ $MmM \subseteq MnM$. Andererseits nehme man wieder an, dass $MmM = MnM$, womit $n \in MmM$, also $n = umv$ für $u, v \in M$, gelte. Wegen $h(a)n \in F(m) \ni nh(b)$ gilt dann

$$m = rh(a)ns \qquad m - pnh(b)t \qquad \text{mit } r, s, p, t \in M.$$

Daraus folgt: $n = umv = urh(a)nsv$, mit dem Kürzungslemma (Lemma 3.23), dass $n = urh(a)n$, und somit dann auch

$$m = pnh(b)t = purh(a)nh(b)t,$$

was aber im Widerspruch zu $h(a)nh(b) \notin F(m)$ steht. □

3.4.3.3 Von FO-definierbaren Sprachen zu aperiodischen Monoiden

Proposition 3.27

Jede FO-definierbare Sprache wird von einem aperiodischen endlichen Monoid erkannt.

Beweis: Sei V eine endliche Menge von FO-Variablen und φ eine FO-Formel mit $\text{free}_1(\varphi) \subseteq V$. Wir wollen zeigen, dass die Sprache

$$L_V(\varphi) = \left\{w \in (\Sigma \times \mathfrak{P}(V))^* \;\middle|\; w \text{ ist } V\text{-Wort und } w \vdash \varphi\right\}$$

von einem endlichen aperiodischen Monoid erkannt wird. Wir beweisen per struktureller Induktion:

$\varphi = P_a(x)$: Rechne nach, dass $\text{Syn}(L_V(\varphi))$ aperiodisch ist (Blatt 06, Aufgabe 2).

$\varphi = x < y$: Rechne nach, dass $\text{Syn}(L_V(\varphi))$ aperiodisch ist (Blatt 06, Aufgabe 2).

$\varphi = \neg\psi$: Es gilt $L_V(\varphi) = \overline{L_V(\psi)} \cap L_V$, wobei $L_V \subseteq (\Sigma \times \mathfrak{P}(V))^*$ die Sprache aller V-Wörter ist. Per Induktion ist $\mathsf{Syn}(L_V(\psi)) = \mathsf{Syn}(\overline{L_V(\psi)})$ aperiodisch. Ebenso kann man nachrechnen, dass $\mathsf{Syn}(L_V)$ aperiodisch ist. Also ist $\mathsf{Syn}(L_V(\psi)) \times \mathsf{Syn}(L_V)$ ein aperiodisches endliches Monoid, das $L_V(\varphi)$ erkennt.

$\varphi = \varphi_1 \vee \varphi_2$: Es gilt

$$L_V(\varphi) = L_V(\varphi_1) \cup L_V(\varphi_2).$$

Per Induktion ist $\mathsf{Syn}(L_V(\varphi_i))$ für $i = 1,2$ aperiodisch. Also ist $\mathsf{Syn}(L_V(\varphi_1)) \times \mathsf{Syn}(L_V(\varphi_2))$ ein endliches aperiodisches Monoid, welches $L_V(\varphi)$ erkennt.

$\varphi = \exists x.\, \psi$: Per Induktion ist $\mathsf{Syn}\left(L_{V\cup\{x\}}(\psi)\right)$ aperiodisch, das heißt es gibt $n \geqslant 1$, so dass für alle $r, w \in \left(\Sigma \times \mathfrak{P}(V \cup \{x\})\right)^*$

$$rw^n s \in L_{V\cup\{x\}}(\psi) \Leftrightarrow rw^{n+1}s \in L_{V\cup\{x\}}(\psi) \qquad (*)$$

Wir wollen zeigen, dass für $u, v, w \in (\Sigma \times \mathfrak{P}(\Sigma))^*$ gilt, dass

$$uw^{2n+1}v \in L_V(\exists x_i.\, \psi) \Leftrightarrow uw^{2n+2}v \in L_V(\exists x_i.\, \psi),$$

woraus die Aperiodizität von $\mathsf{Syn}\left(L_V(\exists x_i.\, \psi)\right)$ unmittelbar folgt.

„$\Rightarrow$“ Sei $uw^{2n+1}v \in L_V(\exists x_i.\, \psi)$. Dann kann man eine Position von $uw^{2n+1}v$ mit x markieren, so dass das resultierende $V \cup \{x\}$-Wort die Formel ψ erfüllt. Weil in einem der beiden Teilwörter w^n **keine** Positiion mit x markiert ist, hat dieses $V \cup \{x\}$-Wort die Form $rw^n s$ mit $r, s \in \left(\Sigma \times \mathfrak{P}(V \cup \{x\})\right)^*$. Wegen $rw^n s \in L_{V\cup\{x\}}(\psi)$ ist gemäß Gleichung $(*)$ auch $rw^{n+1}s \in L_{V\cup\{x\}}(\psi)$. Das Entfernen der Markierungen x liefert das Wort $rw^{2n+2}v$, womit $uw^{2n+2}v \in L_V(\exists x_i.\, \psi)$ gilt.

„$\Leftarrow$“ Wir verwenden hier ein symmetrisches Argument wie eben.

□

Wir sehen nun, dass aus den Propositionen 3.22, 3.26 und 3.27 direkt Satz 3.21 folgt.

3.5 Varietäten und Gleichungen

Eigenschaften von Monoiden werden häufig durch Gleichungen beschrieben, beispielsweise

- ein Monoid ist **kommutativ**, wenn die Gleichung $xy = yx$ gilt.
- ein endliches Monoid ist **aperiodisch**, wenn die Gleichung $x^{n+1} = x^n$ für hinreichend großes n gilt.

Wir werden zeigen, dass **gleichungsdefinerbare Monoideigenschaften** genau diejenigen sind, die Varietäten bestimmen. Wir führen nun noch Notation ein:

Notation

Fixiere eine *abzählbar unendliche* Menge $X = \{x_0, x_1, \ldots\}$ von Variablen und schreibe für $n \geqslant 0$

$$X_n = \{x_0, \ldots, x_n\}.$$

Definition 3.21

Eine **Gleichung** ist ein Paar $(s,t) \in X^* \times X^*$. Wir notieren dafür $s = t$. Ein Monoid M erfüllt $s = t$, wenn für *alle* Morphismen $h : X^* \to M$ gilt, dass $h(s) = h(t)$.

Erläuterung

Jeder Morphismus $h : X^* \to M$ ist durch seine Einschränkung $h_0 : X \to M$ *eindeutig* bestimmt, das heißt er entspricht einer Interpretation der Variablen durch Elemente von M. Die Gleichung $s = t$ gilt in M, wenn die linke und rechte Seite für jede Interpretation denselben Wert haben.

Definition 3.22

Sei $E = (s_n = t_n)_{n \geqslant 0}$ eine Folge von Gleichungen. Ein Monoid M erfüllt **fast alle Gleichungen** in E, wenn es ein $n_0 \geqslant 0$ gibt, so das M alle Gleichungen $s_n = t_n$ mit $n \geqslant n_0$ erfüllt. Schreibe dann

$$\mathbb{V}(E) = \{M \mid M \text{ ist ein endl. Monoid, welches fast alle Gleichungen in } E \text{ erfüllt.}\}$$

Beispiel 3.8: Für $E = (x^{n+1} = x^n)_{n \geqslant 1} = (x^2 = x,\ x^3 = x^2,\ \ldots)$ ist $\mathbb{V}(E)$ die Varietät der aperiodischen endlichen Monoide.
Für $E = (xy = yx)_{n \geqslant 0}$ ist $\mathbb{V}(E)$ die Varietät der kommutativen endlichen Monoide. ※
Wir kommen nun zu dem Hauptsatz in diesem Kapitel, dem Satz von Eilenberg-Schützenberger:

Satz 3.28 (*Satz von Eilenberg-Schützenberger*)

Eine Klasse $\mathbb{V}$ von endlichen Mengen ist genau dann eine Varietät, wenn es eine Folge E von Gleichungen gibt mit $\mathbb{V}(E) = \mathbb{V}$.

Wir werden den Beweis nach hinten verschieben, da uns noch ein wichtiges Lemma fehlt, welches wir zuerst aufstellen und beweisen wollen.

Definition 3.23

Sei $\equiv \subseteq M \times M$ eine Kongruenz auf einem Monoid M. Wir definieren dann:

(1) $\equiv$ ist **endlich** genau dann, wenn $M/\equiv_M$ endlich ist.

(2) Eine Teilmenge $\mathcal{W} \subseteq \equiv$ **erzeugt** $\equiv$, wenn für alle Kongruenzen $\equiv'$ gilt

$$\mathcal{W} \subseteq \equiv' \to \equiv \subseteq \equiv',$$

das heißt $\equiv$ ist die **kleinste Relation**, welche $\mathcal{W}$ enthält oder in anderen Worten ist $\equiv$ der Durchschnitt aller Kongruenzen, die $\mathcal{W}$ enthalten.

(3) $\equiv$ ist **endlich erzeugt**, wenn $\equiv$ eine endliche erzeugende Teilmenge hat.

Lemma 3.29

Eine jede endliche Kongruenz $\equiv$ auf X_n^* ist endlich erzeugt.

Beweis: Sei $\equiv$ eine endliche Kongruenz auf X_n^*. Dann gibt es ein $k > 0$, so dass jede Kongruenzklasse ein Wort der Länge $\ell < k$ beinhaltet. Betrachte nun die Menge

$$\mathcal{W} = \{\ (u,v) \in X^* \times X^* \mid u \equiv v \text{ und } |u| \leqslant k, |v| \leqslant k\ \}.$$

Sei $\equiv_{\mathcal{W}}$ die von $\mathcal{W}$ erzeugte Kongruenz, sprich der Durchschnitt aller Kongruenzen, welche $\mathcal{W}$ enthalten, so wollen wir nun noch zeigen, dass $\equiv_{\mathcal{W}} = \equiv$ gilt.

„$\subseteq$“ Diese Richtung ist klar per definitionem, weil $\mathcal{W} \subseteq \equiv$.

„$\supseteq$“ Wir wollen zuerst zeigen, dass es für alle Wörter $w \in X_n^*$ ein $w' \in X_n^*$ mit $|w'| < k$ und $w' \equiv_{\mathcal{W}} w$ existiert. Dies zeigen wir per Induktion nach $|w|$. Für $|w| < k$ ist die Aussage klar; wir wählen dazu einfach $w' = w$. Sei also $|w| \geqslant k$. Schreibe nun w als $w = vx$ mit $v \in X_n^*$ und $x \in X_n$. Per Induktion gibt es $v' \in X_n^*$ mit $|v'| < k$ und $v' \equiv_{\mathcal{W}} w$. Nach der Wahl von k gibt es dann ein $w' \in X_n^*$ mit $|w'| < k$ und $w' \equiv v'x$. Wegen $|v'x| \leqslant k$ und $|w'| < k$ gilt $(v'x, w') \in \mathcal{W}$, also insbesondere $v'x \equiv_{\mathcal{W}} w'$. Also gilt ebenso $w = vx \equiv v'x \equiv_{\mathcal{W}} w'$.
Sei dann $u \equiv v$. Dann gilt mit vorheriger Feststellung, dass $u \equiv_{\mathcal{W}} u'$ für ein u' mit $|u'| < k$ und $v \equiv_{\mathcal{W}} v'$ für ein v' mit $|v'| < k$. Weil nun $\equiv_{\mathcal{W}} \subseteq \equiv$ gilt dann $u' \equiv u \equiv v \equiv v'$, insbesondere also auch $(u', v') \in \mathcal{W}$, woraus direkt folgt, dass $u \equiv_{\mathcal{W}} u' \equiv_{\mathcal{W}} v' \equiv_{\mathcal{W}} v$, was zu zeigen war. □

Wir zeigen nun noch den Satz von Eilenberg-Schützenberger.
Beweis des Satzes 3.28: Wir zeigen die Äquivalenz in zwei Richtungen, wovon die eine als Übungsaufgabe gestellt wird.

„$\Leftarrow$“ siehe Übungsblatt 6, Aufgabe 3

„$\Rightarrow$“ Sei $\mathbb{V}$ eine Varietät, so gilt einerseits:

(1) Es gibt eine Folge von Monoiden $M_0, M_1, M_2, \cdots \in \mathbb{V}$, so dass jedes Monoid $M \in \mathbb{V}$ ein homomorphes Bild fast aller M_ns ist.
Dies ist leicht und schnell zu sehen, denn seien $S_0, S_1, S_2, \ldots$ die Elemente von $\mathbb{V}$ (bis auf Isomorphie). Definiere dann M_n als

$$M_n := \underbrace{\prod_{i=0}^{n} S_i}_{\in \mathbb{V}} \qquad (n \geqslant 0).$$

Für **alle** Monoide $M \in \mathbb{V}$ ist $M \cong S_m$ für ein $m \geqslant 0$. Für $n \geqslant m$ erhalten wir dann folgenden surjektiven Morphismus

$$M_n = \textstyle\prod_{i=0}^{n} S_i \xrightarrow{\pi_{m,n}} S_m \xrightarrow{\cong} M,$$

wobei $\pi_{m,n}$ die Projektion $(s_0, \ldots, s_n) \mapsto (s_m)$ beschreibt. Dementsprechend ist M homomorphes Bild von M_n mit $n \geqslant m$.

Wir betrachten nun für $n \geqslant 0$ folgende Kongruenz auf X_n^*:

$$u \equiv_n v \iff M_n \text{ erfüllt } u = v \iff \text{ für alle Morphismen } h : X_n^* \to M_n \text{ gilt } h(u) = h(v)$$

Dann gilt andererseits:

(2) $X_n^*/\equiv_n \in \mathbb{V}$ für $n \geqslant 0$.
Dieser Fakt ist ebenso ersichtlich, denn seien $h_1, \ldots, h_p : X_n^* \to M_n$ die **endlich** vielen Morphismen von X_n^* nach M_n. Betrachte dann den Morphismus

$$\begin{aligned} g : X_n^* &\longrightarrow \overbrace{\prod_{i=1}^{p}}^{p \text{ Faktoren}} M_n, \\ w &\longmapsto (h_1(w), \ldots, h_p(w)). \end{aligned}$$

Dann ist $\equiv_n$ der Kern von g. Mit dem Homomorphiesatz folgt dann direkt

$$X_n^*/\equiv_n \cong g[X_n^*] \subseteq \prod_{i=1}^{p} M_n.$$

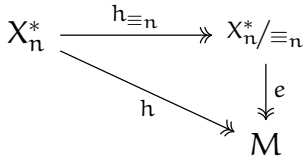
Weil $M_n \in \mathbb{V}$ und $\mathbb{V}$ unter endlichen Produkten und Untermonoiden abgeschlossen ist, folgt direkt $X_n^*/\equiv_n \in \mathbb{V}$.

Nach (2) ist $\equiv_n$ eine endliche Kongruenz und damit nach Lemma 3.29 endlich erzeugt, womit es eine endliche Menge $W_n \subseteq \equiv_n$ gibt, welche $\equiv_n$ erzeugt.
Sei $E = (s_i = t_i)_{i\geqslant 0}$ eine Folge, die alle Elemente von $\bigcup_{n\geqslant 0} W_n$ auflistet. Wir wollen dann zeigen, dass $\mathbb{V} = \mathbb{V}(E)$ gilt.

„$\subseteq$“ Sei $M \in \mathbb{V}$. Nach (1) gibt es $m \geqslant 0$, so dass M ein homomorphes Bild aller M_n mit $n \geqslant m$ ist. Weil M_n alle Gleichungen in W_n erfüllt, gilt das auch für das homomorphe Bild M von M_n. Also erfüllt M alle Gleichungen in $\bigcup_{n\geqslant m} W_n$, sprich fast alle Gleichungen in E.

„$\supseteq$“ Sei $M \in \mathbb{V}(E)$. Wähle $n \geqslant |M|$, so dass M alle Gleichungen in W_n erfüllt. Wähle einen surjektiven Morphismus

$$h : X_n^* \twoheadrightarrow M.$$



Dann gilt $h(u) = h(v)$ für $(u,v) \in W_n$, das heißt $W_n \subseteq \equiv_n$ und $\equiv_n \subseteq \equiv_h$, da W_n der Kern von h. Nach dem Homomorphiesatz gibt es einen Morphismus $e : X_n^*/\equiv_n \to M$ mit $e \circ h_{\equiv_n} = h$.

Weil $X_n^*/\equiv_n \in \mathbb{V}$ (nach (2)) und $\mathbb{V}$ unter homomorphen Bildern abgeschlossen ist, ist $M \in \mathbb{V}$.

3.6 Varietäten von Sprachen

Unser Ziel wird in diesem Kapitel sein eine generische Korrespondenz zwischen den Eigenschaften von Sprachen und Monoiden, das heißt ein gemeinsames Dach über Ergebnissen der Form

$$L \text{ hat die Eigenschaft } P \Leftrightarrow \mathrm{Syn}(L) \text{ hat die Eigenschaft } Q,$$

zu finden.
Typischerweise ist die Klasse $\mathbb{V}$ aller endlicher Monoide mit Eigenschaft Q eine Varietät von endlichen Monoiden, das heißt abgeschlossen unter endlichen Produkte, Untermonoiden und homomorphen Bildern. Man kommt schnell auf die Idee Sprachen mit einer Eigenschaft P ebenso durch Abschlusseigenschaften zu charakterisieren.

Notation

Für eine Klasse $\mathcal{V}$ von regulären Sprachen bezeichnet $\mathcal{V}(\Sigma)$ die Menge aller Sprachen über Σ in $\mathcal{V}$.

Damit lässt sich nun definieren:

Definition 3.24

Eine **Varietät von Sprachen** ist eine Klasse $\mathcal{V}$ von regulären Sprachen, welche unter booleschen Operationen, Ableitungen und homomorphen Urbildern abgeschlossen ist.
Genauer: Für alle Alphabete Σ und Δ gilt ...

(a) … $K, L \in \mathcal{V}(\Sigma) \Rightarrow K \cup L, K \cap L, \Sigma^* \backslash L, \emptyset, \Sigma^* \in \mathcal{V}(\Sigma)$.

(b) … $L \in \mathcal{V}(\Sigma)$ und $v, w \in \Sigma^* \Rightarrow v^{-1} L w^{-1} \in \mathcal{V}(\Sigma)$, wobei wir definieren

$$v^{-1} L w^{-1} := \left\{ u \in \Sigma^* \;\middle|\; vuw \in L \right\}.$$

(c) … $L \in \mathcal{V}(\Sigma)$ und $h : \Delta^* \to \Sigma^*$ ein Morphismus, dann gilt

$$h^{-1}[L] = \left\{ w \in \Delta^* \;\middle|\; h(w) \in L \right\} \in \mathcal{V}(\Delta).$$

Lemma 3.30

Für jede Varietät $\mathbb{V}$ von endlichen Monoiden ist die Klasse $\mathcal{V}$ mit

$$\mathcal{V}(\Sigma) = \left\{ L \subseteq \Sigma^* \;\middle|\; \mathsf{Syn}(L) \in \mathbb{V} \right\}$$

eine Varietät von Sprachen.

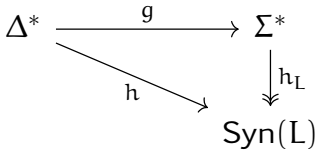
Beweis: siehe Übungsblatt 7, Aufgabe 1. □

Beispiel 3.9: Für $\mathbb{V}$ = aperiodische endliche Monoide ist $\mathcal{V}$ = alle sternfreien Sprachen (nach Satz 3.21). Insbesondere sind die sternfreien Sprachen unter Ableitungen und homomorphen Urbildern abgeschlossen (was direkt schwer zu beweisen wäre). ※

Lemma 3.31

Sei $\mathcal{V}$ eine Varietät von Sprachen und $L \in \mathcal{V}(\Sigma)$.
Dann liegt jede von $\mathsf{Syn}(L)$ erkannte Sprache in $\mathcal{V}$.

Beweis: Sei $h : \Delta^* \to \mathsf{Syn}(L)$ ein Morphismus.
Wir wollen zeigen, dass $h^{-1}[x] \in \mathcal{V}(\Delta)$ für jedes Element $x \in \mathsf{Syn}(L)$.
Sei dazu h_L der syntaktische Morphismus von L. Wähle dann den Morphismus $g : \Delta^* \to \Sigma^*$ mit $h_L \circ g = h$. Dieser muss nach Homomorphiesatz existieren und eindeutig sein. Dann ist



$$h^{-1}[x] = g^{-1}\left[h^{-1}[x]\right].$$

Weil $\mathcal{V}$ nun per definitionem unter Urbildern abgeschlossen ist, genügt es zu zeigen, dass für beliebiges $x \in \mathsf{Syn}(L)$ $h^{-1}[x] \in \mathcal{V}(\Sigma)$ gilt. Weil L von h_L erkannt wird ist $L = h^{-1}[S]$ für $S \subseteq \mathsf{Syn}(L)$. Definiere dann

$$E = \left\{ (u, w) \in \Sigma^* \times \Sigma^* \;\middle|\; h_L(u) x h_L(w) \in S \right\}.$$

Gölte nun

$$h_L^{-1}[x] = \left(\bigcap_{(v,w) \in E} v^{-1} L w^{-1} \right) \setminus \left(\bigcup_{(v,w) \notin E} v^{-1} L w^{-1} \right),$$

so folgte direkt, dass $h_L^{-1}[x] \in \mathcal{V}(\Sigma)$, da $\mathcal{V}$ unter booleschen Operationen und Ableitungen abgeschlossen ist. Zeigen wir nun also die Gleichung:

„$\subseteq$“ Sei $u \in h_L^{-1}[x]$ und $(v, w) \in E$. Dann ist

$$h_L(vuw) = h_L(v) x h_L(w) \in S,$$

also $vuw \in h_L^{-1}[S] = L$ und somit auch $u \in v^{-1} L w^{-1}$. Analog gilt für $(v, w) \notin E$, dass $u \notin v^{-1} L w^{-1}$. Dementsprechend ist u ein Element der rechten Seite.

„$\supseteq$“ Sei u nun ein Wort der rechten Seite. Dann gilt für alle Wörter $v, w \in \Sigma^*$, dass

$$h_L(v)h_L(u)h_L(w) \in S \quad \Leftrightarrow \quad h_L(v)xh_L(w) \in S.$$

Wähle nun $y \in h^{-1}[x]$ beliebig. Dann gilt

$$\begin{aligned} vuw \in L &\Leftrightarrow h_L(v)h_L(u)h_L(w) \in S \\ &\Leftrightarrow h_L(v)xh_L(w) \in S \\ &\Leftrightarrow h_L(v)h_L(y)h_L(w) \in S \\ &\Leftrightarrow vyw \in L. \end{aligned}$$

Also gilt $u \equiv_L y$, das heißt $h_L(u) = h_L(y) = x$, woraus folgt, dass $u \in h_L^{-1}[x]$ gelten muss.

□

Lemma 3.32

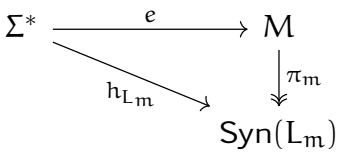
Sei $\mathbb{V}$ eine Varietät von endlichen Monoiden und $M \in \mathbb{V}$. Dann gibt es ein Alphabet Σ und von M erkannte Sprachen $L_1, \ldots, L_k \subseteq \Sigma^*$, so dass M ein Untermonoid von

$$\prod_{i=1}^{k} \mathsf{Syn}(L_i)$$

ist.

Beweis:
Wähle Σ so, dass ein surjektiver Morphismus $e : \Sigma^* \twoheadrightarrow M$ existiert. Sei $L_m = e^{-1}[m]$ für $m \in M$ und π_m der eindeutige Morphismus mit

$$h_{L_m} = \pi_m \circ e.$$



Betrachte nun den Morphismus

$$\pi : M \to \prod_{m \in M} \mathsf{Syn}(L_m), \qquad n \mapsto \Big(\pi_m(n)\Big)_{m \in M}.$$

Wir behaupten nun, und beweisen gleich, dass π injektiv ist. Denn seien $m, n \in M$ mit $\pi(m) = \pi(n)$, so gilt insbesondere $\pi_m(m) = \pi_m(n)$. Wähle dann $v, w \in \Sigma^*$ mit $e(v) = m$ und $e(w) = n$. Dann gilt

$$h_{L_m}(v) = (\pi_m \circ e)(v) = \pi_m(m) = \pi_m(n) = (\pi_m \circ e)(w) = h_{L_m}(w),$$

womit $v \equiv_{L_m} w$. Wegen $v \in L_m$ gilt dann auch $w \in L_m$ und somit $n = e(w) = m$.
Aus der Injektivität von π folgt dann

$$M \cong \pi[M] \overset{\text{U.M.}}{\subseteq} \prod_{m \in M} \mathsf{Syn}(L_m),$$

was zu zeigen war. □

Lemma 3.33

Für jede Varietät $\mathcal{V}$ von Sprachen ist

$$\mathbb{V} = \left\{ M \;\middle|\; M \text{ ist endliches Monoid und jede von } M \text{ erkannte Sprache liegt in } \mathcal{V} \right\}$$

eine Varietät von endlichen Monoiden.

Beweis: Wir sehen dieses Lemma als eine Kosequenz der folgenden Beobachtungen für beliebige endliche Monoide M, N an: Einerseits ist gemäß Proposition 3.17.(1) und 3.17.(2) eine jede von einem Untermonoid oder homomorphen Bild von M erkannte Sprache auch von M erkennbar, andererseits ist gemäß Proposition 3.17.(3) jede von $M \times N$ erkannte Sprache eine boolesche Kombination von Sprachen, welche von M oder N erkannt werden. □

Lemma 3.34

Für jede Varietät $\mathcal{V}$ von Sprachen wird $\mathbb{V}$ von den Monoiden $\mathsf{Syn}(L)$ mit $L \in \mathcal{V}$ erzeugt, das heißt $\mathbb{V}$ ist der Abschluss der Klasse $\left\{ \mathsf{Syn}(L) \;\middle|\; L \in \mathcal{V} \right\}$ unter endlichen Produkten, Untermonoiden und homomorphen Bildern.

Beweis: Sei $\mathbb{W}$ die von den $\mathsf{Syn}(L)$ erzeugte Varietät von Monoiden, wobei $L \in \mathcal{V}$ gilt. Wir zeigen dann, dass $\mathbb{V} = \mathbb{W}$.

„$\subseteq$“ Für jedes $L \in \mathcal{V}$ ist $\mathsf{Syn}(L) \in \mathbb{V}$ nach Lemma 3.31. Weil $\mathbb{W}$ nun von den Monoiden $\mathsf{Syn}(L)$ mit $L \in \mathcal{V}$ erzeugt wird, folgt automatisch $\mathbb{W} \subseteq \mathbb{V}$.

„$\supseteq$“ Sei nun $M \in \mathbb{V}$. Nach Lemma 3.32 gibt es dann von M erkannte Sprachen $L_1, \ldots, L_k$ mit

$$M \rightarrowtail \prod_{i=1}^{k} \mathsf{Syn}(L_i).$$

Dann ist $L_i \in \mathcal{V}$ nach der Definition von $\mathbb{V}$ und damit $\mathsf{Syn}(L_i) \in \mathbb{W}$ nach der Definition von $\mathbb{W}$. Also ist $M \in \mathbb{W}$, weil $\mathbb{W}$ unter endlichen Produkten und Untermonoiden abgeschlossen ist. □

Satz 3.35 (*Eilenberg-Korrespondenz*)

Die Abbildungen

$$\mathcal{V} \mapsto \mathbb{V} \qquad \text{und} \qquad \mathbb{V} \mapsto \mathcal{V}$$

sind zueinander invers, das heißt sie definieren eine **bijektive Korrespondenz** zwischen Varietäten von Sprachen und Varietäten von endlichen Monoiden.

Beweis: Für alle Varietäten $\mathcal{V}$ von Sprachen und $\mathbb{V}$ von Monoiden schreibe …

- … $\mathbb{V} \to \mathcal{V}$, falls $\mathcal{V}(\Sigma) = \left\{ L \subseteq \Sigma^* \;\middle|\; \mathsf{Syn}(L) \in \mathbb{V} \right\}$
- … $\mathcal{V} \to \mathbb{V}$, falls $\mathbb{V} = \left\{ M \text{ endliches Monoid} \;\middle|\; \text{jede von } M \text{ erkannte Sprache liegt in } \mathcal{V} \right\}$

Wir zeigen nun die Korrespondenz in zwei Richtungen:

(1) Sei $\mathbb{V} \to \mathbb{V}$ und $\mathbb{V} \to \mathcal{W}$, so wollen wir zeigen, dass $\mathcal{V} = \mathcal{W}$.

„$\subseteq$“ Sei $L \in \mathcal{V}$. Dann ist nach Lemma 3.34 $\mathsf{Syn}(L) \in \mathbb{V}$ und somit per definitionem $L \in \mathcal{W}$.

„$\supseteq$“ Sei $L \in \mathcal{W}$. Dann gibt es per definitionem ein $M \in \mathbb{V}$, das L erkennt. Nach Definition von $\mathbb{V}$ liegt **jede** von M erkannte Sprache in $\mathcal{V}$, insbesondere also $L \in \mathcal{V}$.

(2) Sei $\mathbb{V} \to \mathcal{V}$ und $\mathcal{V} \to \mathbb{W}$, so wollen wir zeigen, dass $\mathbb{V} = \mathbb{W}$.

„$\subseteq$“ Sei $M \in \mathbb{V}$. Nach Lemma 3.32 gibt es von M erkannte Sprachen $L_1, \ldots, L_k$ mit

$$M \rightarrowtail \prod_{i=1}^{k} \mathsf{Syn}(L_i),$$

insbesondere gilt also $L_1, \ldots, L_k \in \mathcal{V}$. Nach Lemma 3.34 folgt nun aber $\mathsf{Syn}(L_1), \ldots, \mathsf{Syn}(L_k) \in \mathbb{W}$ und somit auch $M \in \mathbb{W}$, da $\mathbb{W}$ eine Varietät ist.

„$\supseteq$“ Für jedes $L \in \mathcal{V}$ gilt per definitionem $\mathsf{Syn}(L) \in \mathbb{V}$. Weil $\mathbb{W}$ nach Lemma 3.34 von den $\mathsf{Syn}(L)$ mit $L \in \mathcal{V}$ erzeugt wird folgt $\mathbb{W} \subseteq \mathbb{V}$.

□

KAPITEL 4

Reguläre Sprachen und Topologie

Ziel dieses Kapitels ist eine topologische Charakterisierung der regulären Sprachen als offen-abgeschlossene Mengen im Raum der proendlichen Wörter.

4.1 Grundlegendes

Wir wollen zu Beginn ersteinmal den Begriff des metrischen Raumes prägen:

Definition 4.1 (*Metrik*)

Eine **Metrik** auf einer Menge X ist eine Funktion

$$d : X \times X \to \mathbb{R}^+ = [0, \infty),$$

so dass für $x, y, z \in X$ gilt:

(1) $d(x,x) = 0$ und $d(x,y) > 0$ für $x \neq y$ (**Positivität**)

(2) $d(x,y) = d(y,x)$ (**Symmetrie**)

(3) $d(x,y) \leqslant d(x,z) + d(z,y)$ (**Dreiecksungleichung**)

Definition 4.2 (*Metrischer Raum*)

Ein Raum (X, d), wobei X eine Menge und d eine Metrik auf X ist, heißt **metrischer Raum**. Ist die zugehörige Metrik klar, so wird der Raum auch mit der Menge identifiziert.

Beispiel 4.1:

a. $\mathbb{R}$ ist ein metrischer Raum bezüglich der **euklidischen Metrik**

$$d(x,y) = |x - y|.$$

b. Die Menge $\{0,1\}^n$ der n-stelligen Binärwörter ist ein metrischer Raum bezüglich der **Hamming-Metrik**

$$d(v,w) = \left| \left\{ i \in \{1,\dots,n\} \;\middle|\; v_i \neq w_i \right\} \right|.$$

※

Definition 4.3

Sei Σ Alphabet, M ein Monoid und $v, w \in \Sigma^*$.

(1) Ein Morphismus $h : \Sigma^* \to M$ **trennt** v und w, falls $h(v) \neq h(w)$.

(2) M **trennt** v und w, wenn ein solches h existiert, welches v und w trennt.

Intuition

Sind v und w strukturell ähnliche Wörter benötigt man ein großes Monoid, um sie zu trennen.

Definition 4.4

Für $v, w \in \Sigma^*$ definiere

$$r(v,w) = \min\left\{ |M| \;\middle|\; M \text{ ist endliches Monoid, welches } v \text{ und } w \text{ trennt} \right\}$$

und

$$d(v,w) = 2^{-r(v,w)}.$$

Wir verwenden dabei die Konvention, dass $\min(\emptyset) = \infty$ und $2^{-\infty} = 0$.
Es gilt damit also:

$$d(v,w) < 2^{-n} \iff \text{Kein Monoid der Größe} \leqslant n \text{ trennt } v \text{ und } w$$

Beispiel 4.2:

a. Alle Wörter $u, v \in \Sigma^*$ mit $|u| \leqslant |v|$ lassen sich durch ein Monoid der Größe $|u|+2$ trennen, nämlich durch

$$M_{|u|+1} = \left\{0, 1, \ldots, |u|+1\right\} \quad \text{mit} \quad i \oplus j = \min\{i+j, |u|+1\}.$$

Für den Morphismus

$$h : \Sigma^* \to M_{|u|+1}, a \in \Sigma \mapsto 1$$

gilt $h(u) = |u| \neq |u|+1 = h(v)$. Dementsprechend ist $d(u,v) \geqslant 2^{-(|u|+2)}$.

b. Alle Wörter der Form va und wb mit $v, w \in \{a,b\}^*$ lassen sich durch ein Monoid der Größe 3 trennen, nämlich

$$U_2 = \{1, a_1, a_2\}, \quad \text{mit } a_i \cdot a_j = a_j \text{ für } i,j = 1,2.$$

Für den Morphismus

$$h : \{a,b\}^* \to U_2, \begin{cases} a \mapsto a_1 \\ b \mapsto a_2 \end{cases}$$

gilt $h(va) = a_1$ und $h(wb) = a_2$. Das heißt für die Metrik $d(va, wb) \geqslant 2^{-3} = 1/8$.

※

Lemma 4.1

d ist die Metrik auf Σ^*. Für alle $u, v, w \in \Sigma^*$ gilt zudem die **verschärfte Dreiecksungleichung**

$$d(u,v) \leqslant \max\{d(u,w), d(w,v)\} \leqslant d(u,w) + d(w,v),$$

sowie die **Kürzungsregel**

$$d(uv, uw) \leqslant d(v,w) \quad \text{und} \quad d(vu, wu) \leqslant d(v,w).$$

Beweis: Seien $u, v, w \in \Sigma^*$ beliebig. Wir zeigen die Metrikeigenschaften, die verschärfte Dreiecksungleichung, sowie die Kürzungsregel:

ad i (**Positivität**) Es ist klar, dass $d(w,w) = 0$. Für $v \neq w$ ist $d(v,w) \neq 0$, das heißt es gibt ein endliches Monoid M, welches v und w trennt. In der Tat: Betrachte $L = \{w\}$ und $M = \mathrm{Syn}(L)$. Weil der syntaktische Morphismus $h_L : \Sigma^* \twoheadrightarrow M$ die Sprache L erkennt, trennt er v und w.

ad ii (**Symmetrie**) Klar

ad iii (**Dreiecksungleichung**) wird durch die verschärfte Dreiecksungleichung gezeigt.

ad iv (**verschärfte Dreiecksungleichung)** Sei M ein endliches Monoid, welches v und w trennt. Dann trennt M entweder v und u oder u und w. Hieraus folgt direkt

$$r(v,w) \geqslant \min\{r(v,u), r(u,w)\}$$

und somit auch

$$d(v,w) \leqslant \max\{d(v,u), d(u,w)\}.$$

ad v (**Kürzungsregel**) Sei M ein endliches Monoid, welches vu und wu trennt. Dann trennt M auch v und w. Dementsprechend ist $r(vu,wu) \geqslant r(v,w)$ und somit auch $d(vu,wu) \leqslant d(v,w)$.

□

Notation

Für $n > 1$ und $v, w \in \Sigma^*$ definere

$$\begin{aligned} v \equiv_n w \quad &:\Longleftrightarrow \quad d(v,w) < 2^{-n} \\ &\Longleftrightarrow \quad \text{Kein Monoid der Größe } \ell \leqslant n \text{ trennt } v \text{ und } w \end{aligned}$$

Korrolar 4.1.1

$\equiv_n$ ist eine endliche Kongruenz auf Σ^*.

Beweis: Die Kongruenzeigenschaft folgt direkt aus Lemma 4.1. Betrachte – zum Nachweis der Endlichkeit – **alle** Morphismen von Σ^* in Monoide der Größe $\ell \leqslant n$ (bis auf Isomorphie):

$$h_i : \Sigma^* \to M_i \quad \text{für } i \in I.$$

Dann ist I endlich und für alle $v, w \in \Sigma^*$ gilt

$$v \equiv_n w \quad \Longleftrightarrow \quad h_i(v) = h_i(w) \forall i \in I.$$

Also ist folgende Abbildung wohldefiniert und injektiv:

$$\begin{aligned} h : {}^{\Sigma^*}/_{\equiv_n} &\longrightarrow \prod_{i \in I} M_i, \\ [w]_{\equiv_n} &\longmapsto (h_i(w))_{i \in I}. \end{aligned}$$

Weil $\prod_{i \in I} M_i$ endlich ist, ist auch ${}^{\Sigma^*}/_{\equiv_n}$. □

Definition 4.5 (*Offenheit, Abgeschlossenheit*)

Sei (X, d) ein metrischer Raum, $x \in X$ und $\varepsilon > 0$. Dann ist

$$B(x,\varepsilon) = \left\{ y \in X \;\middle|\; d(x,y) < \varepsilon \right\}$$

die **offene Kugel** mit Mittelpunkt x und Radius ε.
Eine Teilmenge $U \subseteq X$ heißt **offen**, wenn für jedes $x \in U$ ein $\varepsilon > 0$ existiert mit

$$B(x, \varepsilon) \subseteq U.$$

Insbesondere ist $B(x, \varepsilon)$ selbst eine offene Menge.
Eine Menge $C \subseteq X$ heißt **abgeschlossen**, wenn $X \backslash C$ offen ist. Eine Menge $A \subseteq X$ heißt **abgeschloffen**, wenn $X \backslash A$ *sowohl offen als auch abgeschlossen* ist.

Definition 4.6 (*Topologie*)

Sei (X, d) ein metrischer Raum. Dann bilden die offenen Mengen von X eine **Topologie**, das heißt ...

- $\emptyset$ und X sind offene Mengen.
- Sind U_i $(i \in I)$ offene Mengen, so ist auch $\bigcup_{i \in I} U_i$ offen.
- Sind $U_1, \ldots, U_n$ offene Mengen, so ist auch $\bigcap_{i=1}^{n} U_i$ offen.

Lemma 4.2

Σ^* trägt die diskrete Topologie, das heißt jede Menge ist offen.

Beweis: Für $U \subseteq \Sigma^*$ gilt $U = \bigcup_{u \in U} \{u\}$. Damit genügt es zu zeigen, dass für $u \in \Sigma^*$ $\{u\}$ offen ist. Sei $M = \mathrm{Syn}(\{u\})$ und $n = |M|$. Wir zeigen nun, dass wenn $B(u, 2^{-n}) = \{u\}$ gilt $\{u\}$ offen ist. Sei dazu $v \in B(u, 2^{-n})$, also $d(u, v) < 2^{-n}$ und somit $r(u, v) > n$. Damit werden u und v vom kleinem Monoid der Größe $\leqslant n$ getrennt. Insbesondere ist M ein solches MOnoid, das heißt der syntaktische Morphismus $h_{\{u\}} : \Sigma^* \to M$ identifiziert u und v, womit $u = v$ gelten muss. □

Proposition 4.3

Jede offene Kugel in Σ^* ist eine reguläre Sprache. Umgekehrt ist jede reguläre Sprache $L \subseteq \Sigma^*$ eine endliche Vereinigung von offenen Kugeln.

Beweis:

„⇒“ Sei $w \in \Sigma^*$ und $\varepsilon > 0$. Zu zeigen $B(w, \varepsilon) \subseteq \Sigma^*$ ist regulär. Wähle $n \geqslant 0$ mit $2^{-(n+1)} < \varepsilon \leqslant 2^{-n}$. Betrachte den Morphismus

$$\pi_n : \Sigma^* \twoheadrightarrow {}^{\Sigma^*}/_{\equiv_n},\ u \mapsto [u]_{\equiv_n}.$$

Dann erkennt π_n die Sprache $B(w, \varepsilon)$, denn

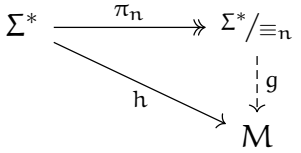
$$\begin{aligned} B(w, \varepsilon) &= B(w, 2^{-n}) \\ &= \left\{ v \in \Sigma^* \;\middle|\; v \equiv_n w \right\} \\ &= [w]_{\equiv_n} \\ &= \pi_n^{-1}[\pi_n(w)] \end{aligned}$$

„⇐“ Sei $L \subseteq \Sigma^*$ regulär. Wähle das endliche Monoid M und einen Morphismus $h : \Sigma^* \to M$, der L erkennt, das heißt

$$L = h^{-1}[S] \quad \text{mit } S \subseteq M.$$

Sei $n = |M|$. Für alle $v, w \in \Sigma^*$ gilt

$$v \equiv_n w \Rightarrow h(v) = h(w).$$



Gemäß Homomorphiesatz gibt es ein

$$g : \Sigma^*/_{\equiv_n} \to M \quad \text{mit} \quad g \circ \pi_n = h.$$

Es folgt

$$\begin{aligned} L &= h^{-1}[S] = \pi_n^{-1}\left[g^{-1}[S]\right] = \bigcup_{[w]_{\equiv_n} \in g^{-1}[S]} \pi_n^{-1}[[w]_{\equiv_n}] \\ &= \bigcup_{[w]_{\equiv_n} \in g^{-1}[S]} [w]_{\equiv_n} \\ &= \bigcup_{[w]_{\equiv_n} \in g^{-1}[S]} B(w, 2^{-n}). \end{aligned}$$

□

Definition 4.7 (*Stetigkeit*)

Seien (X, d_X) und (Y, d_Y) metrische Räume. Eine Funktion $f : X \to Y$ ist **stetig**, wenn gilt, dass

$$\forall x \in X.\, \forall \varepsilon > 0.\, \exists \delta > 0.\, \forall x' \in X.\, \left(d_X(x, x') < \delta\right) \Rightarrow d_Y(f(x), f(x')) < \varepsilon.$$

Man kann dann folgende Äquivalenzen zeigen, wir wollen dies an dieser Stelle überspringen:

Satz 4.4 (*Äquivalenzen der Stetigkeit*)

Die folgenden Aussagen sind äquivalent:

(1) f ist stetig.

(2) Für jede *offene* Menge $U \subseteq Y$ ist $f^{-1}[U] \subseteq X$ *offen*. (In anderen Worten: Offene Mengen haben offene Urbilder)

(3) Für jede *abgeschlossene* Menge $C \subseteq Y$ ist $f^{-1}[C] \subseteq X$ *abgeschlossen*. (In anderen Worten: Abgeschlossene Mengen haben abgeschlossene Urbilder)

Definition 4.7 (*Stetigkeit – fortgesetzt*)

Eine Funktion $f : X \to Y$ ist **gleichmäßig stetig**, wenn gilt, dass

$$\forall \varepsilon > 0.\, \exists \delta > 0.\, \forall x, x' \in X.\, \left(d_X(x, x') < \delta\right) \Rightarrow d_Y(f(x), f(x')) < \varepsilon.$$

Man kann zeigen:

Lemma 4.5 (*Aussagen zur (gleichmäßigen) Stetigkeit*)

Jede gleichmäßig stetige Funktion ist stetig. Sind $f : X \to Y$ und $g : Y \to Z$ (gleichmäßig) stetige Funktionen, so ist auch die Verkettung dieser $g \circ f : X \to Z$ (gleichmäßig) stetig.

Proposition 4.6

Eine Funktion $f : \Delta^* \to \Sigma^*$ ist genau dann gleichmäßig stetig, wenn für jede reguläre Sprache

$L \subseteq \Sigma^*$ auch $f^{-1}[L] \subseteq \Delta^*$ regulär ist.

Beweis:

„⇒“ Sei $f : \Delta^* \to \Sigma^*$ gleichmäßig stetig und $L \subseteq \Sigma^*$ regulär. Zeige nun, dass $f^{-1}[L]$ regulär ist. Weil ein endliche Vereinigung von offenen Kugeln in Σ^* existiert, dürfen wir $L = B(w, \varepsilon)$ mit $w \in \Sigma^*$ und $\varepsilon > 0$ annehmen.

$$f \text{ ist gleichmäßig stetig } \implies \left(\exists\, n \geqslant 0.\ \forall\, u, v \in \Delta^*.\ d(u,v) < 2^{-n} \Rightarrow d(f(u), f(v)) < \varepsilon\right). \quad (*)$$

Wir behaupten nun

$$f^{-1}[L] = \bigcup_{u \in f^{-1}[L]} B(u, 2^{-n}),$$

denn

„⊆“ Klar.

„⊇“ Sei $v \in B(u, 2^{-n})$ für ein $u \in f^{-1}[L]$. Dann ist $f(u) \in L = B(w, \varepsilon)$, das heißt $d(w, f(u)) < \varepsilon$. Wegen $d(u,v) < 2^{-n}$ gilt dann gemäß $(*)$ $d(f(u), f(v)) < \varepsilon$. Daraus folgt aber unmittelbar

$$d(f(v), w) \leqslant \max\left\{ \overbrace{d(f(v), f(u))}^{<\varepsilon}, \overbrace{d(f(u), w)}^{<\varepsilon} \right\} < \varepsilon.$$

Dann folgt aber $f(v) \in B(w, \varepsilon) = L$, woraus folgt, dass $v \in f^{-1}[L]$.

Betrachte dann den Morphismus

$$\pi_n : \Delta^* \twoheadrightarrow {}^{\Delta^*}\!/_{\equiv_n},\ w \mapsto [w]_{\equiv_n} = B(w, 2^{-n})$$

und erkenne, dass $f^{-1}[L]$ von π_n erkannt wird, denn

$$f^{-1}[L] = \bigcup_{u \in f^{-1}[L]} B(u, 2^{-n}) = \bigcup_{u \in f^{-1}[L]} \pi_n^{-1}[\pi_n(u)] = \pi_n^{-1}\left[\bigcup_{u \in f^{-1}[L]} \pi_n(u) \right],$$

womit dann folgt, dass $f^{-1}[L]$ regulär ist.

„⇐“ Sei $f^{-1}[L]$ regulär für jede reguläre Sprache $L \subseteq \Sigma^*$. Fixiere dann $n \geqslant 0$. Sei $\mathcal{B}_n$ die endliche Menge aller offenen Kugeln in Σ^* mit Radius 2^{-n}. Für jedes $L \in \mathcal{B}_n$ ist $f^{-1}[L]$ regulär. Definiere dann

$$k = \max\left\{ \left| \mathsf{Syn}\left(f^{-1}[L]\right) \right| \;\middle|\; L \in \mathcal{B}_n \right\}.$$

Wir behaupten nun, dass für alle $u, v \in \Delta^*$ gilt, dass

$$d(u,v) < 2^{-k} \Rightarrow d(f(u), f(v)) < 2^{-n},$$

woraus die gleichmäßige Stetigkeit von f folgt.
Man sieht, dass die Behauptung gilt, denn sei $d(u,v) < 2^{-k}$. Weil die Kugeln in $\mathcal{B}_n$ eine Partition von Σ^* bilden, gilt

$$\Delta^* = \bigcup_{L \in \mathcal{B}_n} f^{-1}[L].$$

Also gibt es ein $L \in \mathcal{B}_n$ mit $u \in f^{-1}[L]$. Wegen $d(u,v) < 2^{-k}$ werden u und v von keinem Monoid der Größe $\ell \leqslant k$ getrennt. Insbesondere werden sie **nicht** von dem syntaktischen Monoid $\mathsf{Syn}\left(f^{-1}[L]\right)$ getrennt. Wegen $u \in f^{-1}[L]$ folgt dann auch $v \in f^{-1}[L]$ und somit folgt, da $f(u), f(v) \in L$, $d(f(u), f(v)) < 2^{-n}$.

□

Korrolar 4.6.1

Jeder Morphismus $h : \Delta^* \to \Sigma^*$ ist gleichmäßig stetig.

Wir beschäftigen uns nun mit einem weiteren zentralen Thema, der Konvergenz von Folgen.

Definition 4.8 (*Konvergenz*)

Sei (X, d) ein metrischer Raum. Eine Folge $(x_n)_{n\geqslant 0}$ von Elementen **konvergiert** gegen ein $x \in X$, wenn

$$\forall\, \varepsilon > 0.\ \exists\, n_0 \geqslant 0.\ \forall\, n \geqslant n_0.\ d(x, x_n) < \varepsilon.$$

Notation

Wenn die Folge $(x_n)_{n\geqslant 0} \subseteq X$ gegen $x \in X$ konvergiert, so notieren wir

$$x_n \longrightarrow x \qquad \text{für } n \to \infty.$$

Eine andere Notation ist es die Bedingung $n \to \infty$ auf den Konvergenzpfeil zu schreiben.

Ein weiterer Satz über stetige Funktionen, welcher bereits aus grundlegenden Analysisveranstaltungen bekannt sein sollte ist:

Satz 4.7 (*Erhaltung von Grenzwerten*)

Eine Funktion $f : X \to Y$ ist genau dann stetig, wenn sie Grenzwerte von Folgen erhält, das heißt für $(x_n)_{n\geqslant 0} \subseteq X$ und $x \in X$ gilt

$$x_n \xrightarrow{n\to\infty} x \Rightarrow f(x_n) \xrightarrow{n\to\infty} f(x).$$

Definition 4.9 (*Cauchy-Folge*)

Eine Folge $(x_n)_{n\geqslant 0} \subseteq X$ heißt **Cauchy-Folge**, wenn gilt:

$$\forall\, \varepsilon > 0. \exists\, n_0 \geqslant 0.\ \forall\, m, n \geqslant n_0.\ d(x_m, x_n) < \varepsilon.$$

Es gilt, dass eine **jede konvergente** Folge auch eine Cauchy-Folge ist, nicht aber umgekehrt. Diese „Lücke“ führt uns zum Begriff der Vollständigkeit:

Definition 4.10 (*Vollständigkeit*)

Ein metrischer Raum (X, d) heißt **vollständig**, wenn umgekehrt jede Cauchy-Folge auch gegen **ein Element aus der Menge X** konvergiert.

Beispiel 4.3: $\mathbb{Q}$ ist nicht vollständig, $\mathbb{R}$ hingegen schon. Man betrachte für ersteres dazu die Folge beschrieben durch

$$E = \left\{ x \in \mathbb{Q} \;\middle|\; x^2 < 2 \right\}$$

und sehe, dass sie keine obere Schranke in $\mathbb{Q}$ hat und damit auch nicht in $\mathbb{Q}$ konvergiert. $\mathbb{R}$ ist über die Vervollständigung (siehe unten) von $\mathbb{Q}$ definiert.[1] ※

[1] Die Definition über die Menge aller Dedekind’schen Schnitte ist dazu selbstverständlich äquivalent hierzu.

Definition 4.11 (*Vervollständigung*)

Eine **Vervollständigung** von X ist ein Raum $\left(\widehat{X}, \widehat{d}\right)$ zusammen mit einer Funktion

$$\iota : X \to \widehat{X},$$

so dass ...

(1) ... $\widehat{X}$ vollständig ist.

(2) ... ι ist eine **isometrische Einbettung**, das heißt

$$\forall\, x, y.\ \widehat{d}(\iota(x), \iota(y)) = d(x, y).$$

Man kann also X als Unterraum von $\widehat{X}$ auffassen.

(3) ... X liegt **dicht** in $\widehat{X}$, das heißt für alle $x \in \widehat{X}$ gibt es eine Folge $(x_n) \subseteq X$ mit $x_n \to x$ für $n \to \infty$.

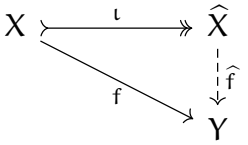
Beispiel 4.4: $\mathbb{R}$ ist definiert als eine Vervollständigung von $\mathbb{Q}$. ⁂

Wir kommen nun zur universellen Eigenschaft der Vervollständigung. Auch diesen Satz werden wir an dieser Stelle nicht beweisen, da er analog zum Beweis von Satz 3.12 abläuft.

Satz 4.8

(1) Jeder metrische Raum X hat eine Vervollständigung $\widehat{X}$ und diese ist bis auf isometrische Isomorphie eindeutig bestimmt.

(2) Für jeden vollständigen Raum Y und jede gleichmäßig stetige Funktion $f : X \to Y$ gibt es genau eine gleichmäßige stetige Funktion $\widehat{f} : \widehat{X} \to Y$ mit $\widehat{f} \circ \iota = f$. Wir nennen dies die **universelle Eigenschaft von** $\widehat{X}$.



Konstruktion von $\widehat{X}$ Zwei Cauchy-Folgen $(x_n), (y_n) \subseteq X$ sind **äquivalent**, wenn

$$d(x_n, y_n) \xrightarrow{n\to\infty} 0.$$

Definiere $\widehat{X}$ dann als die Menge **aller** Äquivalenzklassen von Cauchy-Folgen in X mit Metrik

$$d_{\widehat{X}}([(x_n)],\ [(y_n)]) = \lim_{n\to\infty} d(x_n, y_n).$$

Die Einbettung von X in $\widehat{X}$ ist dann gegeben durch

$$\iota : X \rightarrowtail \widehat{X}, x \mapsto [(x, x, x, \dots)].$$

4.2 Proendliche Wörter

Definition 4.12 (*Proendliche Wörter*)

Sei $\widehat{\Sigma}^*$ die Vervollständigung von Σ^*. Die Elemente von $\widehat{\Sigma}^*$ heißen **proendliche Wörter** über Σ.

Definition 4.13 (*Produktmetrik*)

Seien X, Y metrische Räume. Definiere dann den Produktraum

$$X \times Y := \{ (x,y) \mid x \in X, y \in Y \}$$

mit der Metrik

$$d_{X\times Y}((x,y),(x',y')) = \max\left\{ d_X(x,x'), d_Y(y,y') \right\}.$$

Man kann auch hier zeigen:

Lemma 4.9 (*Aussagen über Produkträume*)

Die Projektionen

$$\pi_X : X \times Y \to X \qquad \text{und} \qquad \pi_Y : X \times Y \to Y$$

sind dann gleichmäßig stetig.
Eine Funktion $f : Z \to X \times Y$ ist genau dann gleichmäßig stetig, wenn beide Komponenten $\pi_X \circ f$ und $\pi_Y \circ f$ gleichmäßig stetig sind.
Sind $X_0 \subseteq X$ und $Y_0 \subseteq Y$ dichte Unterräume, so ist $X_0 \times Y_0 \subseteq X \times Y$ ebenso dicht. Sind X und Y vollständig, so auch der Produktraum $X \times Y$.

Wir werden dieses Lemma nun nicht beweisen; ein Beweis ist in jedem Standardlehrbuch der Analysis/Topologie zu finden.

Proposition 4.10

Die Monoidoperation auf Σ^*

$$\cdot : \Sigma^* \times \Sigma^* \to \Sigma^*, (u,v) \mapsto uv$$

ist gleichmäßig stetig und kann insbesondere eindeutig zu einer gleichmäßig stetigen binären Operation auf $\widehat{\Sigma}^*$ fortgesetzt werden. Bezüglich dieser Operation ist $\widehat{\Sigma}^*$ ein Monoid mit neutralem Element ε.

Beweis: Wir zeigen zuerst die gleichmäßige Stetigkeit: Für $u', u, v', v \in \Sigma^*$ gilt

$$\begin{aligned} d(uv, u'v') &\leqslant \max\left\{ d(uv, uv'), d(uv', u'v') \right\} \\ &\leqslant \max\left\{ d(v,v'), d(u',u') \right\} \\ &\leqslant d_{\Sigma^*\times\Sigma^*}((u,v),(u',v')). \end{aligned}$$

Damit folgt die gleichmäßige Stetigkeit der Operation auf Σ^*.
Für den zweiten Teil betrachten wir folgendes Kommutativitätsdiagramm:

$$\begin{array}{ccc} \Sigma^* \times \Sigma^* & \xrightarrow{\iota\times\iota} & \widehat{\Sigma^*} \times \widehat{\Sigma^*} = \widehat{\Sigma^* \times \Sigma^*} \\ \downarrow{\scriptstyle \cdot} & & \downarrow{\scriptstyle \widehat{\cdot}} \\ \Sigma^* & \xrightarrow{\iota} & \widehat{\Sigma^*} \end{array}$$

Sei $\iota : \Sigma^* \rightarrowtail \widehat{\Sigma^*}$ die isometrische Einbettung. Dann ist

$$\iota \times \iota : \Sigma^* \times \Sigma^* \rightarrowtail \widehat{\Sigma^*} \times \widehat{\Sigma^*}$$

die Vervollständigung von $\Sigma^* \times \Sigma^*$. Nach der universellen Eigenschaft der Vervollständigung gibt es nun eine eindeutige gleichmäßig stetige Funktion

$$\widehat{\cdot} : \widehat{\Sigma^*} \times \widehat{\Sigma^*} \to \widehat{\Sigma^*}$$

für die das Quadrat kommutiert.
Wir schreiben dann $(x,y) \mapsto xy$ für diese Operation und zeigen ihre Assoziativität und die Existenz eines neutralen Elements ε. Seien $x,y,z \in \widehat{\Sigma^*}$. Weil $\Sigma^* \subseteq \widehat{\Sigma^*}$ dicht ist, gibt es Folgen $(x_n),(y_n),(z_n) \subseteq \Sigma^*$, die gegen x, y und z konvergieren.
Aufgrund der Stetigkeit der Operation gilt nun

$$x(yz) \xleftarrow{n\to\infty} x_n(y_nz_n) = (x_ny_n)z_n \xrightarrow{n\to\infty} (xy)z.$$

Für das neutrale Element zeigt man die Eigenschaft analog:

$$x\varepsilon = \varepsilon x = x.$$

□

Bemerkung

Die **diskrete Metrik** auf einer Menge X ist gegeben durch

$$d(x,y) = \begin{cases} 0 & \text{, wenn } x = y \\ 1 & \text{, sonst} \end{cases}.$$

Bezüglich dieser Metrik ist die Topologie von X diskret. **Konvention**: Jedes endliche Monoid wird als diskreter metrischer Raum aufgefasst.

Proposition 4.11

Jeder Morphismus $h : \Sigma^* \to M$ in ein endliches Monoid M ist gleichmäßig stetig und kann eindeutig zu einem gleichmäßig stetigen Morphismus $\widehat{h} : \widehat{\Sigma^*} \to M$ fortgesetzt werden.

Beweis: Sei M endlich und $h : \Sigma^* \to M$ ein Morphismus. Wir stellen fest, dass für alle $u,v \in \Sigma^*$ der Zusammenhang

$$d(u,v) < 2^{-|M|} \Rightarrow h(u) = h(v)$$

gilt. Also ist h gleichmäßig stetig. Wegen der universellen Eigenschaft von $\widehat{\Sigma^*}$ lässt sich h eindeutig zu einer gleichmäßig stetigen Funktion $\widehat{h} : \widehat{\Sigma^*} \to M$ fortsetzen, da M als diskreter Raum vollständig ist. Was noch zu zeigen ist, ist die Morphismuseigenschaft für $\widehat{h}$. Es gilt offensichtlich $\widehat{h}(\varepsilon) = h(\varepsilon) = \mathbf{1}$. Betrachte nun also

$$D = \left\{ (u,v) \in \widehat{\Sigma^*} \times \widehat{\Sigma^*} \;\middle|\; \widehat{h}(uv) = \widehat{h}(u) \cdot \widehat{h}(v) \right\}$$

die Menge aller Elemente, welche die Morphismuseigenschaft erfüllen. Zu zeigen bleibt nur noch die Gleichheit von D mit $\widehat{\Sigma^*} \times \widehat{\Sigma^*}$. Da h Morphismus ist gilt $\Sigma^* \times \Sigma^* \subseteq D$. Weil nun aber $\Sigma^* \times \Sigma^*$ dicht in $\widehat{\Sigma^*} \times \widehat{\Sigma^*}$ liegt, ist auch D dicht. Es genügt also zu zeigen, dass $D \subseteq \widehat{\Sigma^*} \times \widehat{\Sigma^*}$ abgeschlossen ist. Denn allgemein gilt, dass wenn X ein metrischer Raum ist und $D \subseteq X$ ein dichter Unterraum ist, welcher ebenso **abgeschlossen** ist, ebenso $D = X$ gelten muss.

Sei nun $\widehat{\pi} : \widehat{\Sigma^*} \times \widehat{\Sigma^*} \to \widehat{\Sigma^*}$ die Monoidoperation von $\widehat{\Sigma^*}$ und $\sigma : M \times M \to M$ die Monoidoperation auf M. Dann gilt allerdings

$$D = \underbrace{\bigcup_{m \in M} \left(\overbrace{\left(\widehat{h} \circ \widehat{\pi}\right)^{-1}[m]}^{\text{abgeschlossen}} \cap \overbrace{\left(\sigma \circ \left(\widehat{h} \times \widehat{h}\right)\right)^{-1}[m]}^{\text{abgeschlossen}} \right)}_{\text{abgeschlossen}},$$

denn weil $\widehat{h}$, $\widehat{\pi}$ und σ stetig sind, sind auch alle diese Urbilder abgeschlossen, womit die Abgeschlossenheit von D folgt. □

Korrolar 4.11.1

$\widehat{\Sigma^*}$ ist das **freie proendliche Monoid** über Σ: **Jede** Funktion $h_0 : \Sigma \to M$ in ein endliches Monoid kann **eindeutig** zu einem gleichmäßig stetigen Morphismus $\widehat{h} : \widehat{\Sigma^*} \to M$ fortgesetzt werden.

Definition 4.14

Ein endliches Monoid M **trennt** $u, v \in \widehat{\Sigma^*}$, wenn ein stetiger Morphismus $\widehat{h} : \widehat{\Sigma^*} \to M$ mit $\widehat{h}(u) = \widehat{h}(v)$ existiert. Definiere dann

$$r(u,v) = \min \left\{ \; |M| \;\middle|\; M \text{ trennt } u \text{ und } v \; \right\}.$$

Korrolar 4.11.2

Die Metrik von $\widehat{\Sigma^*}$ ist gegeben durch

$$d(u,v) = 2^{-r(u,v)} \qquad \text{für } u, v \in \widehat{\Sigma^*}.$$

Beweis: siehe Übungsblatt 7, Aufgabe 4 □

Korrolar 4.11.3

Sei $(u_n) \subseteq \widehat{\Sigma^*}$ und $u \in \widehat{\Sigma^*}$. Dann sind die folgenden Aussagen äquivalent:

(1) $u_n \xrightarrow{n \to \infty} u$

(2) Für jeden Morphismus $\widehat{h} : \widehat{\Sigma^*} \to M$ in ein endliches Monoid gilt

$$\widehat{h}(u_n) = \widehat{h}(u) \qquad \text{für hinreichend großes } n$$

Beweis:

ad „(1) ⇒ (2)“ Sei $u_n \to u$ für $n \to \infty$. Weil $\widehat{u}$ stetig ist, gilt

$$\widehat{h}(u_n) \xrightarrow{n \to \infty} \widehat{h}(u);$$

weil M diskret ist, folgt (2).

ad „(2) ⇒ (1)“ Sei (2) erfüllt und $k > 0$. Weil es nur endlich viele Monoide der Größe $\leqslant k$ gibt, gibt es $n_0 \geqslant 0$, so dass für alle Morphismen

$$h : \Sigma^* \to M \qquad (|M| \leqslant k)$$

gilt, dass $\widehat{h}(u_n) = \widehat{h}(u)$ für $n \geqslant n_0$. Also gilt $d(u_n, u) < 2^{-k}$ für $n \geqslant n_0$, woraus folgt, dass $u_n \xrightarrow{n\to\infty} u$.

□

Wir werden nun ein explizites Beispiel für ein proendliches Wort kennenlernen in **Beispiel 4.5:** Das proendliche Wort x^ω ist für $x \in \widehat{\Sigma^*}$ definiert als

$$x^\omega := \lim_{n\to\infty} x^{n!}.$$

※

Wir wollen nun das Beispiel durch folgendes „Sätzchen“ rechtfertigen:

„Sätzchen“ 4.12 (*Rechtfertigung von Beispiel 4.5*)

Sei $x \in \widehat{\Sigma^*}$.

(1) Die Folge $\left(x^{n!}\right)_{n\geqslant 0}$ ist eine Cauchy-Folge in $\widehat{\Sigma^*}$, das heißt, dass der Limes x^ω existiert.

(2) x^ω ist idempotent.

(3) Für jeden Morphismus $\widehat{h} : \widehat{\Sigma^*} \to M$ in ein endliches Monoid gilt

$$\widehat{h}(x^\omega) = \underbrace{h(x)^\omega}_{\in M}$$

Beweis:

ad (1) Es genügt zu zeigen, dass für alle $n \geqslant 0$ und $p, q \geqslant n$

$$d\left(x^{p!}, x^{q!}\right) < 2^{-n}$$

gilt. Sei dazu M ein Monoid der Größe $\ell \leqslant n$ und $\widehat{h} : \widehat{\Sigma^*} \to M$ ein stetiger Morphismus. Für $y \in M$ ist

$$y^\omega = y^r \qquad \text{für } r \leqslant |M| \leqslant n.$$

Weil r ein Teiler von $p!$ ist, folgt

$$y^\omega = y^r = y^{p!}.$$

Man zeigt dann analog $y^\omega = y^{q!}$. Daraus folgt dann mit $y = \widehat{h}(x)$, dass

$$\widehat{h}\left(x^{p!}\right) = \widehat{h}(x)^{p!} = \widehat{h}(x)^\omega = \widehat{h}(x)^{q!} = \widehat{h}\left(x^{q!}\right).$$

ad (3) Sei $\widehat{h} : \widehat{\Sigma^*} \to M$ stetiger Morphismus in ein endliches Monoid M. Wegen $x^{n!} \xrightarrow{n\to\infty} x^\omega$ gilt für hinreichend großes n, dass

$$\widehat{h}(x^\omega) = \widehat{h}\left(x^{n!}\right) = \widehat{h}(x)^{n!} = \widehat{h}(x)^\omega.$$

ad (2) Wegen der Stetigkeit der Monoidoperation auf $\widehat{\Sigma^*}$ gilt

$$x^{n!} \cdot x^{n!} \xrightarrow{n\to\infty} x^\omega \cdot x^\omega.$$

Andererseits gilt aber auch

$$x^{n!} \cdot x^{n!} \xrightarrow{n\to\infty} x^\omega,$$

denn für jeden stetigen Morphismus

$$\widehat{h} : \widehat{\Sigma^*} \to M$$

in ein endliches Monoid gilt für hinreichend großes n, dass

$$\begin{aligned} \widehat{h}\left(x^{n!} \cdot x^{n!}\right) &= \widehat{h}(x)^\omega \cdot \widehat{h}(x)^\omega \\ &= \widehat{h}(x)^\omega = \widehat{h}(x^\omega). \end{aligned}$$

Somit gilt dann $x^\omega \cdot x^\omega = x^\omega$.

□

Wir kommen nun zu dem ebenso wichtigem Thema der **Kompaktheit**:

Definition 4.15 (*offene Überdeckung, Kompaktheit und totale Beschränktheit*)

Sei (X, d) ein metrischer Raum. Eine **offene Überdeckung** von X ist eine Familie U_i mit $i \in I$ von offenen Teilmengen von X mit

$$X = \bigcup_{i\in I} U_i.$$

X heißt **kompakt**, wenn jede offene Überdeckung U_i mit $i \in I$ eine **endliche** Teilüberdeckung hat, das heißt es gibt eine endliche Teilmenge $I_0 \subseteq I$, für die gilt, dass

$$X = \bigcup_{i\in I_0} U_i.$$

X heißt **total beschränkt**, wenn X für jedes $\varepsilon > 0$ durch endlich viele offene Kugeln vom Radius ε überdeckt werden kann.

Auch hier kann man zeigen:

Lemma 4.13 (*Äquivalente Definitionen der Kompaktheit in metrischen Räumen*)

Äquivalent zur Defintion der Kompaktheit ist es zu sagen, dass jede Folge $(x_n) \subseteq X$ eine konvergente Teilfolge besitzt, oder X vollständig und total beschränkt ist.

Anmerkungen

Es gilt:

- Ist X total beschränkt, so ist auch $\widehat{X}$ total beschränkt. ($\Rightarrow$)
- $\widehat{X}$ ist kompakt genau dann wenn X total beschränkt ist. ($\Leftrightarrow$)
- Sind X und Y kompakt, so ist auch der Produktraum $X \times Y$ kompakt. ($\Rightarrow$)

Proposition 4.14

$\widehat{\Sigma^*}$ ist kompakt.

Beweis: Es genügt zu zeigen, dass Σ^* total beschränkt ist. Das ist aber klar, weil Σ^* für jedes $n > 0$ nur endlich viele offene 2^{-n} Kugeln hat, weil $\equiv_n$ eine endliche Kongruenz ist (gemäß Korrolar 4.1.1) □

Definition 4.16

Sei $P \subseteq \widehat{\Sigma^*}$ gegeben.

(1) P ist **erkennbar**, wenn es einen Morphismus

$$h : \widehat{\Sigma^*} \to M$$

in ein endliches Monoid M und ein $S \subseteq M$ gibt mit

$$P = h^{-1}[S].$$

(2) Die **syntaktische Kongruenz** von P ist die folgende Kongruenz auf $\widehat{\Sigma^*}$:

$$s \equiv_P t \iff \Big[\forall\, u, v \in \widehat{\Sigma^*}.\ usv \in P \Leftrightarrow utv \in P \Big].$$

Das Monoid $\widehat{\Sigma^*}/\equiv_P$ ist das **syntaktische Monoid** von P.

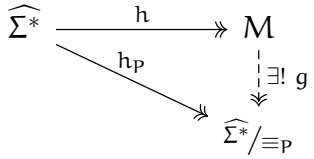
Es gilt:

Lemma 4.15

Der syntaktische Morphismus

$$h_p : \widehat{\Sigma^*} \twoheadrightarrow \widehat{\Sigma^*}/\equiv_P,\ u \mapsto [u]_{\equiv_P}$$

erkennt P und faktorisiert durch jeden surjektiven Morphismus $h : \widehat{\Sigma^*} \twoheadrightarrow M$, der P erkennt:



Beweis: Analog zu Satz 3.12. □

Proposition 4.16

Sei $P \subseteq \widehat{\Sigma^*}$ und M das syntaktische Monoid von P. Dann sind äquivalent:

(1) P ist abgeschloffen.

(2) $\equiv_P \subseteq \widehat{\Sigma^*} \times \widehat{\Sigma^*}$ ist abgeschloffen.

(3) P ist erkennbar und der syntaktische Morphismus $h_p : \widehat{\Sigma^*} \twoheadrightarrow M$ ist stetig.

Beweis:

ad „(1) ⇒ (2)“ Für $u, v \in \widehat{\Sigma^*}$ und $Q \subseteq \widehat{\Sigma^*}$ definiere

$$u^{-1}Qv^{-1} := \left\{ x \in \widehat{\Sigma^*} \;\middle|\; uxv \in Q \right\}.$$

Ist Q abgeschloffen, so auch $u^{-1}Qv^{-1}$, denn diese Menge ist das Urbild von Q bezüglich der stetigen Funktion

$$f : \widehat{\Sigma^*} \times \widehat{\Sigma^*}, x \mapsto uxv.$$

Sei $P \subseteq \widehat{\Sigma^*}$ dann abgeschloffen. Dann gilt

$$\equiv_P = \bigcap_{u,v \in \widehat{\Sigma^*}} \left(u^{-1}Pv^{-1} \times u^{-1}Pv^{-1} \right) \cup \left(u^{-1}P^{\complement}v^{-1} \times u^{-1}P^{\complement}v^{-1} \right),$$

womit $\equiv_P$ abgeschlossen ist. Zu zeigen bleibt noch, dass $\equiv_P^{\complement} = \left(\widehat{\Sigma^*} \times \widehat{\Sigma^*} \right) \setminus \equiv_P$ ebenfalls abgeschlossen ist.
Sei $(s_n, t_n)_{n \geqslant 0}$ eine Folge in $\equiv_P^{\complement}$, die gegen $(s,t) \in \widehat{\Sigma^*} \times \widehat{\Sigma^*}$ konvergiert. Wegen $s_n \not\equiv_P t_n$ gibt es Œ $u_n, v_n \in \widehat{\Sigma^*}$ mit

$$u_n s_n v_n \in P \qquad \text{und} \qquad u_n t_n v_n \notin P.$$

Weil $\widehat{\Sigma^*} \times \widehat{\Sigma^*}$ kompakt ist, hat die Folge (u_n, v_n) eine konvergente Teilfolge. Sei (u,v) deren Grenzwert. Weil die Monoidoperation von $\widehat{\Sigma^*}$ stetig ist und P und $P^{\complement}$ abgeschloffen sind, gilt

$$usv \in P \qquad \text{und} \qquad utv \notin P.$$

Daraus folgt, dass $s \not\equiv_P t$, womit $(s,t) \subseteq \equiv_P^{\complement}$. Also ist $\equiv_P^{\complement}$ abgeschlossen.

ad „(2) ⇒ (3)“ Sei $\equiv_P$ abgeschloffen. Sei dann $u \in P$. Dann gibt es $\varepsilon > 0$ mit

$$\equiv_P \supseteq \underbrace{\mathcal{B}_{\widehat{\Sigma^*} \times \widehat{\Sigma^*}}((u,u), \varepsilon)}_{\mathcal{B}_{\widehat{\Sigma^*}}(u,\varepsilon) \times \mathcal{B}_{\widehat{\Sigma^*}}(u,\varepsilon)}.$$

Also liegen **alle** Elemente von $\mathcal{B}_{\widehat{\Sigma^*}}(u,\varepsilon)$ in derselben Klasse von $\equiv_P$. Folglich ist jede Kongruenzklasse offen, das heißt die Kongruenzklassen bilden eine offene Partition von $\widehat{\Sigma^*}$. Weil $\widehat{\Sigma^*}$ aber kompakt ist, gibt es nur endlich viele Kongruenzklassen, das heißt $M = \widehat{\Sigma^*}/\equiv_P$ ist endlich, womit die erkennbarkeit von P folgt. Für $[u]_{\equiv_P}$ ist

$$h_P^{-1}\left[[u]_{\equiv_P}\right] = [u]_{\equiv_P}$$

offen, woraus die Stetigkeit von h_P folgt.

ad „(3) ⇒ (1)“ Sei nun P erkennbar und h_P ist stetig. Dann ist M endlich und

$$P = h_P^{-1}[S] \qquad \text{mit } S \subseteq M.$$

Weil S abgeschloffen ist folgt auch, dass P abgeschloffen ist.

□

Notation

Sei X ein metrischer Raum und $Y \subseteq X$, dann ist

$$\begin{aligned} \overline{Y} &= \bigcap \left\{ C \;\middle|\; Y \subseteq C \subseteq X \text{ und } C \text{ ist abgeschlossen} \right\} \\ &= \left\{ x \in X \;\middle|\; \exists\, (y_n) \subseteq Y \text{ , welche gegen } x \text{ konvergiert} \right\} \end{aligned}$$

der **Abschluss** von Y.

Lemma 4.17

(1) $L \subseteq \Sigma^* \Rightarrow L = \overline{L} \cap \Sigma^* \subseteq \widehat{\Sigma^*} \cap \Sigma^*$

(2) $P \subseteq \widehat{\Sigma^*}$ ist abgeschloffen, dann ist $P = \overline{(P \cap \Sigma^*)}$

Beweis:

ad (1) „$\subseteq$“ klar

„$\supseteq$“ Sei $u \in \overline{L} \cap \Sigma^*$. Sei M das syntaktische Monoid von $\{u\}$. Dann trennt M jedes $v \neq u$ von u, das heißt

$$r(u,v) \leqslant |M| \qquad \text{für } u \neq v.$$

Wähle dann $(u_n) \subseteq L$ mit $u_n \xrightarrow{n\to\infty} u$. Für hinreichend großes n gilt dann aber

$$d(u_n, u) < 2^{-|M|},$$

womit $u_n = u$ gelten muss, also ist $u \in L$.

ad (2) Sei $P \subseteq \widehat{\Sigma^*}$ abgeschloffen. Weil P offen und Σ^* dicht in $\widehat{\Sigma^*}$ liegt, ist $P \cap \Sigma^*$ dicht in P. Weil P aber abgeschlossen ist, folgt $\overline{P \cap \Sigma^*} = P$.

□

Anmerkung

Die eben verwendete Tatsache lässt sich auch allgemein formulieren: Sei X ein metrischer Raum, $U \subseteq X$ offen und $D \subseteq X$ dicht, so ist $U \cap D$ dicht in U.

Wir wollen nun die topoliogische Charakterisierung der regulären Sprachen formulieren und beweisen. Doch zuerst werden wir eine leicht weiter gefasste Proposition beweisen:

Proposition 4.18

Für $L \subseteq \Sigma^*$ sind äquivalent:

(1) L ist regulär.

(2) $L = P \cap \Sigma^*$ für eine abgeschloffene Menge $P \subseteq \widehat{\Sigma^*}$

(3) $\overline{L} \subseteq \widehat{\Sigma^*}$ ist abgeschloffen.

④ $\overline{L}$ ist erkennbar.

Beweis:

ad „①⇒②“ Sei L regulär, so gibt es einen Morphismus

$$h : \Sigma^* \to M$$

mit M endlich und $L = h^{-1}[S]$ mit $S \subseteq M$. Sei $\widehat{h} : \widehat{\Sigma^*} \to M$ eine stetige Fortsetzung. Definiere dann

$$P := \widehat{h}^{-1}[S].$$

Weil S abgeschloffen, als Teilmenge eines diskreten Monoids, gilt dies auch für P. Weil h und $\widehat{h}$ auf Σ^* übereinstimmen gilt dann:

$$L = h^{-1}[S] = \widehat{h}^{-1}[S] \cap \Sigma^* = P \cap \Sigma^*.$$

ad „②⇒③“ Sei $L = P \cap \Sigma^*$ mit P abgeschloffen, dann ist $\overline{L} = \overline{P \cap \Sigma^*} = P$ gemäß Lemma 4.17, ②.

ad „③⇒④“ Dies folgt aus Proposition 4.16.

ad „④⇒①“ Sei $\overline{L}$ erkennbar, so betrachte $h : \widehat{\Sigma^*} \to M$ den syntaktischen Morphismus von $\overline{L}$. Damit ist $\overline{L} = h^{-1}[S]$ mit $S \subseteq M$. Sei $g : \Sigma^* \to M$ die Einschränkung von h. Dann gilt gemäß Lemma 4.17, dass $L = \overline{L} \cap \Sigma^*$ und somit $L = \overline{L} \cap \Sigma^* = h^{-1}[S] \cap \Sigma^* = g^{-1}[S]$, woraus die Regularität von L folgt.

□

Damit kommen wir nun zur Charakterisierung:

Satz 4.19 (*Topologische Charakterisierung der regulären Sprachen*)

Es besteht eine **bijektive Korrespondenz**

$$\left\{ \text{reguläre Sprachen über } \Sigma \right\} \quad \longleftrightarrow \quad \left\{ \text{abgeschloffene Mengen in } \widehat{\Sigma^*} \right\}.$$
$$L \subseteq \Sigma^* \quad \longmapsto \quad \overline{L} \subseteq \widehat{\Sigma^*}$$
$$P \cap \Sigma^* \quad \longmapsfrom \quad P \subseteq \widehat{\Sigma^*}.$$

Beweis: Dies folgt unmittelbar aus Proposition 4.18. □

4.3 Topologische Charakterisierung von Varietäten

Dieses Kapitel gibt nur einzelne Aussagen wieder, diese allerdings ohne Beweis, was aus Zeitmangel geschah. Sei aber im Folgenden $X := \{x_0, \dots\}$ und $X_n := \{x_0, \dots, x_n\}$. Wir definieren dann:

Definition 4.17 (*proendliche Gleichungen und proendliche Varietäten*)

Eine **proendliche Gleichung** ist ein Paar $(u, v) \in \widehat{X} \times \widehat{X}$ für ein $n \geqslant 0$, notiert als $u = v$. Ein endliches Monoid M **erfüllt** $u = v$, wenn für jeden stetigen Morphismus

$$\widehat{h} : \widehat{X}_n^* \to M \text{ gilt, dass } \widehat{h}(u) = \widehat{h}(v).$$

Sei E eine Menge von proendliche Gleichungen, definiere dann

$$\mathbb{V}(E) = \left\{ M \;\middle|\; M \text{ endliches Monoid, welches alle Gleichungen in } E \text{ erfüllt} \right\}.$$

Auch hier gibt es eine zentrale Korrespondenz, die von Reitermann:

Satz 4.20 (*Reitermann-Korrespondenz*)

Eine Klasse $\mathbb{V}$ von endlichen Monoiden ist genau dann eine Varietät, wenn es eine Menge E von proendlichen Gleichungen gibt mit

$$\mathbb{V} = \mathbb{V}(E).$$

Beispiel 4.6: Die Varietät der aperiodischen proendlichen Monoide wird durch die proendlichen Gleichungen

$$x^\omega = x^\omega \cdot x$$

definiert. ※

KAPITEL A

AUSBLICK

Diese Vorlesung erkannten wir reguläre Sprachen und ihre zugehöhrigen Korrespondenzen:

$$\text{endliche Automaten} \iff \text{MSO} \iff \text{Monoide} \iff \text{proendliche Topologien}$$

Es gibt aber noch ähnliche Ergebnisse für beispielsweise *Sprachen von unendlichen Wörtern*, *Baumsprachen*, *gewichtete Sprachen* und weitere spezifische Sprachtypen. Mittels **Monaden und Dualität** erhält man dann eine kategorientheoretische Verallgemeinerung der in der Vorlesung vorgestellten Konzepte. Dies ist aber Stoff einer (beziehungsweise mehrerer) Folgeveranstaltungen.