Theoretische Informatik I, Wintersemester 2004/2005
Christoph Pflaum

Inhaltsverzeichnis

Dieses Skript entstand mit freundlicher Unterstützung von
Wolfang Degen und Helmut Meyn zur Vorlesung Theoretische Informatik I.

1 Grundbegriffe und Notationen

1.1 Logische Formeln und Konnektoren

Aussagen sind Sätze, die entweder wahr oder falsch sind. Beispiele:

- 7 ist eine Primzahl. (wahre Aussage)
- Herr Pflaum besitzt ein rotes Auto. (wahre Aussage)
- Alle Autos sind rot. (falsche Aussage)
- Es gibt unendlich viele "Primzahlzwillinge", d.h. Paare p_1, p_2 von Primzahlen, für die $p_2 - p_1 = 2$ ist. (unbekannt ob wahr oder falsch).

Einer Aussage kann man einen Bezeichnungen geben. Zum Beispiel kann man der Aussage

7 ist eine Primzahl.

den Bezeichnungen x geben und der Aussage

Herr Pflaum hat ein rotes Auto.

den Bezeichnungen y. Aussagen lassen sich durch Konnektoren verknüpfen.

- $F \wedge G$ steht für "F und G".
- $F \vee G$ steht für "F oder G".
- $\neg F$ steht für "nicht F".

Beispiel 1. *Seien x und y die Bezeichnungen der obigen Aussagen und z der Bezeichnung von*

Alle Autos sind rot.

Dann ist

$$x \wedge y$$

wahr. Aber

$$x \wedge z$$

falsch. Jedoch ist

$$x \vee z$$

wieder richtig.

Beispiel 2. *Es sei A irgendeine Aussage.*
Dann ist die Aussage

$$A \vee \neg A$$

immer wahr.
Die Aussage ist

$$A \wedge \neg A$$

immer falsch.

Gerne werden folgende Folgerungspfeile verwendet:

- $F \Rightarrow G$ steht für "aus F folgt G".
- $F \Leftarrow G$ steht für "aus G folgt F".
- $F \Leftrightarrow G$ steht für "F ist richtig genau dann wenn G richtig ist".

Beispiel 3. *Es sei A die Aussage*
33584320 ist durch 105280 teilbar,
B die Aussage
105280 ist durch 235 teilbar,
und C die Aussage
33584320 ist durch 235 teilbar.
Auf Grund der Teilbarkeitsregel in der Mathematik gilt dann:

$$A \wedge B \Rightarrow C.$$

Die Zeichen $\neg, \vee, \wedge, \Rightarrow$ sind Zeichen, die in der Aussagenlogik verwendet werden.

Bisher haben wir nur Aussagen über einzelne Objekte gemacht. Wichtig ist es, Aussagen über Mengen von Objekten zu machen.
Beispiele von Mengen sind:

Beispiel 4.
- *Die Menge* $\mathbb{N}$ *der natürlichen Zahlen.*
- *Die Menge* $\mathbb{R}$ *der reellen Zahlen.*
- *Die Menge* $\{a, b, c, ..., z\}$ *der Buchstaben das Alphabets.*

Die Aussage

8 ist eine gerade Zahl.

ist wahr. Ob jedoch

n ist eine gerade Zahl.

wahr ist, hängt von n ab. Damit ist

n ist eine gerade Zahl.

keine Aussage im obigem Sinne, sondern eine Eigenschaft von n, die abhängig von n wahr oder falsch sein kann.

Daher nennen wir

n ist eine gerade Zahl.

ein **Prädikat**.

Definition 1. *Ein Prädikat $P(n)$ ist eine Eigenschaft von Elementen n aus einer Menge. Für jedes n ist $P(n)$ eine Aussage.*

Es sei

P(n)

ein Prädikat. (Also z.B. *n ist eine gerade Zahl.*)
Dann steht

- $\forall x \;\; P(x)$ für " Für alle x gilt $P(x)$.".
- $\exists x \;\; P(x)$ für " Es gibt mindestens ein x, so dass $P(x)$ gilt.".
- $\exists! x \; P(x)$ für " Es existiert genau ein x, so dass $P(x)$ gilt.".

Man schreibt

- $\forall x \;\; (x \in A \Rightarrow P(x))$ für " Für alle $x \in A$ gilt $P(x)$.".
- $\exists x \;\; (x \in A \wedge P(x)$ für
  " Es gibt mindestens ein $x \in A$, so dass $P(x)$ gilt.".

Des weiteren schreibt man oft

- $\forall x \in A \;\; P(x) :\Leftrightarrow \forall x \;\; (x \in A \Rightarrow P(x))$ und

- $\exists x \in A \;\; P(x) :\Leftrightarrow \exists x \;\; (x \in A \wedge P(x))$.

Beispiel 5. *Definition der Stetigkeit einer Funktion an einem Punkt x_0:*

$$\forall \epsilon > 0 \quad \exists \delta > 0 \quad \forall x \quad (|x - x_0| < \delta \Rightarrow |f(x) - f(x_0)| < \epsilon)$$

Beispiel 6 (Formel des Beweises durch Induktion). *Es sei $P(n)$ ein Prädikat. Dann gilt*

$$P(1) \wedge (\forall n \in \mathbb{N} \quad (P(n) \Rightarrow P(n+1))) \quad \Rightarrow \quad \forall n \in \mathbb{N} \quad P(n)$$

Anwendung dieses Beispiels:

- $P(n)$ sei das Prädikat $\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$.
- $P(1)$: $\sum_{k=1}^{1} k = 1 = \frac{1(1+1)}{2}$ ist wahr.
- $\forall n \in \mathbb{N} \quad (P(n) \Rightarrow P(n+1))$: Es sei $n \in \mathbb{N}$ beliebig:
  Es sei $P(n)$ wahr. Dann folgt:

$$\begin{aligned} \sum_{k=1}^{n+1} k &= \sum_{k=1}^{n} k + (n+1) \\ &= \frac{n(n+1)}{2} + (n+1) \\ &= \frac{(n+2)(n+1)}{2} \\ &= \frac{(n+1)((n+1)+1)}{2}. \end{aligned}$$

  Daher ist $P(n+1)$ wahr.

- $\forall n \in \mathbb{N} \quad P(n)$:
  Für alle $n \in \mathbb{N}$ gilt die Formel

$$\sum_{k=1}^{n} k = \frac{n(n+1)}{2}.$$

1.2 Mengen und Funktionen

Beispiele von Mengen sind:

Beispiel:

- $\mathbb{N} = \{1, 2, ...\}$
- $\mathbb{N}_0 = \{0, 1, 2, ...\}$
- $\mathbb{Z} = \{..., -2, -1, 0, 1, 2, ...\}$
- $\mathbb{Q}$ Menge der Brüche (rationale Zahlen).
- $\mathbb{R}$ Menge der reellen Zahlen.
- Die Menge $\{a, b, c, ..., z\}$ der Buchstaben das Alphabets.
- $\emptyset$ leere Menge.
- $[a, b] = \{x \in \mathbb{R} | a \leq x \leq b\}$
- $]a, b] = \{x \in \mathbb{R} | a < x \leq b\}$
- $]a, b[= \{x \in \mathbb{R} | a < x < b\}$
- $(a, b) = \{x \in \mathbb{R} | a < x < b\}$
  (andere Schreibweise von $]a, b[$).

Hierbei bedeutet der Strich | "mit der Eigenschaft".
Also bedeutet $\{x \in \mathbb{R} | a \leq x \leq b\}$ die Menge aller x, die in $\mathbb{R}$ enthalten sind und die Eigenschaft $a \leq x \leq b$ besitzen. Eine Eigenschaft kann also durch eine ganze Formel konstruiert werden.

Definition 2. *Es seien M und N Mengen. Dann sei:*

- $M \cup N := \{x | x \in M \vee x \in N\}$ *die Vereinigung,*
- $M \cap N := \{x | x \in M \wedge x \in N\}$ *der Durchschnitt,*
- $M \setminus N := \{x \in M | x \notin N\}$, *die Menge M ohne N und*
- $M \subset N :\Leftrightarrow \forall x \in M \quad x \in N$ *die Teilmengen-Eigenschaft.*
- *M und N heißen disjunkt, falls $M \cap N = \emptyset$.*

Des weiteren schreiben wir

- $M \dot\cup N = A :\Leftrightarrow M \cup N = A \wedge M \cap N = \emptyset$.

$\{M, N\}$ ist eine Partition von A, falls $M \neq \emptyset$ und $N \neq \emptyset$.

Beispiel 7.

- *Es sei M_n eine endliche Menge mit n Elementen. Dann setzen wir $|M_n| := n$. Für zwei endliche Mengen gilt:*

$$A = M \dot\cup N \;\Rightarrow\; |A| = |M| + |N|.$$

- *Es sei $T_{k,n}$ die Menge aller k-elementigen Teilmengen der Menge $M_n = \{1, 2, ..., n\}$ mit n Elementen. Dann gilt*

$$|T_{k,n}| = \binom{n}{k}$$

für $0 \leq k \leq n$. <u>Beweis durch Induktion:</u>
Es sei $t_{k,n} := |T_{k,n}|$. Wir zeigen durch Induktion $t_{k,n} = \binom{n}{k}$.
<u>$n = 1$:</u>
$|T_{0,1}| = |\emptyset| = 1 = \binom{1}{0}$.
$|T_{1,1}| = |M_1| = 1 = \binom{1}{1}$.
<u>$n \to n+1$:</u>
1. Fall: $k = 0 \vee k = n$:
$|T_{0,n}| = |\emptyset| = 1 = \binom{n}{0}$.
$|T_{n,n}| = |M_n| = 1 = \binom{n}{n}$.
2. Fall: $0 < k < n$:

$$\begin{aligned}
T_{k,n+1} &= \{A \mid A \subset M_{n+1}, |A| = k\} \\
&= \{A \mid A \subset M_{n+1}, |A| = k, n+1 \in A\} \\
&\quad \dot\cup \quad \{A \mid A \subset M_{n+1}, |A| = k, n+1 \notin A\} \\
&= \{A \mid A \subset M_n, |A| = k\} \\
&\quad \dot\cup \quad \{B \cup \{n+1\} \mid B \subset M_n, |B| = k-1\} \\
&\Downarrow \\
|T_{k,n+1}| &= |T_{k,n}| + |T_{k-1,n}| \\
&= \binom{n}{k} + \binom{n}{k-1} = \binom{n+1}{k}.
\end{aligned}$$

□

Es sei A eine Grundmenge und $M, N \subset A$ Teilmengen von A.

Definition 3. *Das Komplement relativ zu A ist definiert durch:*

- $M^C := A \setminus M$.

Formel 1 (Komplementenregel von de Morgan). *Es gilt:*

- $(M \cup N)^C = M^C \cap N^C$.
- $(M \cap N)^C = M^C \cup N^C$.

Beweis von $(M \cup N)^C := M^C \cap N^C$:

$$\begin{aligned} x \in (M \cup N)^C &\Leftrightarrow \\ x \notin M \cup N &\Leftrightarrow \\ \neg(x \in M \vee x \in N) &\Leftrightarrow \\ (\neg x \in M) \wedge (\neg x \in N) &\Leftrightarrow \\ (x \in M^C) \wedge (x \in N^C) &\Leftrightarrow \\ x \in M^C \cap N^C & \quad . \end{aligned}$$

Hierbei haben wir die Regel von de Morgan aus der Aussagenlogik verwendet:

$$\neg(A \vee B) \Leftrightarrow (\neg A) \wedge (\neg B)$$

welche wir später mit einer Wahrheitstafel beweisen werden. □

Definition 4. *Es sei A eine Menge und es sei M_a eine Menge für jedes $a \in A$. Dann sei:*

- $\bigcup_{a \in A} M_a := \{x | \exists a \in A \;\; x \in M_a\}$,
- $\bigcap_{a \in A} M_a := \{x | \forall a \in A \;\; x \in M_a\}$.

Definition 5. *Für Mengen $M_1, ..., M_n$ ist das Kreuzprodukt definiert durch*

$$M_1 \times ... \times M_n := \{(a_1, ..., a_n) | a_1 \in M_1, ..., a_n \in M_n\}.$$

Die Elemente des Kreuzprodukts $(a_1, ..., a_n)$ sind endliche Folgen.
Die Kurzschreibweise des Kreuzprodukts ist

$$\bigtimes_{i=1}^{n} M_i := M_1 \times ... \times M_n$$

und

$$\bigtimes_{i=m}^{n} M_i := M_m \times ... \times M_n$$

für $m \leq n$.

Wichtig ist die Frage: Was ist eine Menge? Um diese Frage zu untersuchen betrachten wir folgendes Problem:

Problem: *Es sei $\mathcal{U}$ die Menge aller Mengen. Dann sei M die Menge aller Mengen, die sich selbst nicht als Element enthält. Es sei also:*

$$M = \{A \in \mathcal{U} | A \notin A\}, \quad M \in \mathcal{U}.$$

Ist nun $M \in M$ wahr oder falsch?

Wir beweisen nun, dass die Aussage "$M \notin M$" wahr ist und dass die Aussage "$M \in M$" wahr ist. Dies ist dann ein Widerspruch.

"$M \notin M$ ist wahr."
Nehmen wir an "$M \notin M$ ist falsch." Dann ist "$M \in M$" wahr.
Da $M \in \mathcal{U}$ und M nicht die Eigenschaft "$M \notin M$" besitzt, muss gelten "$M \notin M$ ist wahr.".
"$M \in M$ ist wahr."
Nehmen wir an "$M \in M$ ist falsch." Dann ist "$M \notin M$" wahr.
Da $M \in \mathcal{U}$ und M die Eigenschaft "$M \notin M$" besitzt, muss gelten "$M \in M$ ist wahr."

Man kann den obigen Widerspruch auch kürzer wie folgt formulieren:

$$(M \in M = \{A \in \mathcal{U} | A \notin A\} \wedge M \in \mathcal{U}) \Leftrightarrow (M \notin M \wedge M \in \mathcal{U})$$

Aus letztem Problem sieht man, dass die Konstruktion einer Menge aller Mengen und einer wie oben konstruierten Menge zu Widersprüchen führt.

Wir gehen daher immer wie folgt vor:

- Wir beginnen mit einer der uns bekannten Mengen $\mathbb{N} = \{1, 2, ...\}$ oder $\{a, b, c, ..., z\}$ oder ... .
- Mit Hilfe des Element-Zeichens $\in$, des Strichs $|$ und des Kreuzprodukts zwischen Mengen konstruieren wir aus dieser Grundmenge weitere Mengen.

Geht man bei der Konstruktion von Menge wie beschrieben vor, dann können keine Widersprüche entstehen. Insbesondere ist $a \in M$ wahr oder falsch aber nicht beides.

Beispiel 8. *a) Die Menge der geraden Zahlen ist*

$$\{2n | n \in \mathbb{N}\}.$$

b) Es sei M eine beliebige Menge. Dann ist die Potenzmenge von M

$$\mathcal{P}(M) := \{N | N \subset M\}.$$

c) Für Mengen $M_1, ..., M_n$ ist das Kreuzprodukt

$$M_1 \times ... \times M_n := \{(a_1, ..., a_n) | a_1 \in M_1, ..., a_n \in M_n\}.$$

Formel 2. *Es gilt*

$$\neg(\forall x \in A \;\; P(x)) \;\Leftrightarrow\; \exists x \in A \;\; \neg P(x) \;\; \textit{und}$$
$$\neg(\exists x \in A \;\; P(x)) \;\Leftrightarrow\; \forall x \in A \;\; \neg P(x).$$

Beweis von $\neg(\forall x \in A \;\; P(x)) \Leftrightarrow \exists x \in A \;\; \neg P(x)$:
Es sei

$$A_P = \{x \in A | P(x)\}.$$

Dann sieht man

$$\forall x \in A \;\; P(x) \Leftrightarrow A_P = A.$$

Also

$$\neg(\forall x \in A \;\; P(x)) \Leftrightarrow A_P \neq A.$$

Natürlich ist

$$A_P \neq A \Leftrightarrow \exists x \in A \;\; \neg P(x).$$

Damit haben wir

$$\neg(\forall x \in A \;\; P(x)) \Leftrightarrow \exists x \in A \;\; \neg P(x)$$

bewiesen. □

Definition 6. *Es seien A, B Mengen.*
Eine Funktion (oder Abbildung) $f : A \to B$ ist
eine Teilmenge $f \subset A \times B$, für die gilt:

- $\forall a \in A \;\; \forall b_1, b_2 \in B \;\; ((a, b_1) \in f \wedge (a, b_2) \in f \Rightarrow b_1 = b_2)$
  d.h. zu jedem Wert aus A gibt es höchstens einen Funktionswert aus B.

- $\forall a \in A \;\; \exists b \in B \;\; (a, b) \in f$
  d.h. jedem Wert aus A muß ein Funktionswert aus B zugeordnet werden.

Schreibweise: $f(a) = b :\Leftrightarrow a \in A \wedge b \in B \wedge (a, b) \in f$.

$$\begin{array}{rrcl} \text{Oft schreibt man} & f : A & \to & B \\ & a & \mapsto & T(a), \end{array}$$

wobei T ein Term ist (sein kann), also zum Beispiel $T(a) = a * a + 2 * a$.

Definition 7. *Die Funktion $f : A \to B$ heißt*

- *injektiv, falls $f(x) = f(y) \Rightarrow x = y$.*
- *surjektiv, falls $\forall y \in B \;\; \exists x \in A \;\; f(x) = y$.*
- *bijektiv, falls f injektiv und surjektiv.*

Beispiel 9. *Es sei $f(x) = x^2$.*

- *$f : \mathbb{R} \to \mathbb{R}$ ist weder injektiv noch surjektiv.*
- *$f : \mathbb{R} \to \mathbb{R}_0^+ := \{x \in \mathbb{R} | x \geq 0\}$ ist surjektiv.*
- *$f : \mathbb{R}_0^+ \to \mathbb{R}$ ist injektiv.*
- *$f : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ ist bijektiv.*

Definition 8. *Es seien $f : A \to B$ und $g : B \to C$ Funktionen.*
Dann ist

$$\begin{array}{rcl} g \circ f : A & \to & C \\ g \circ f : z & \mapsto & g(f(z)) \end{array}$$

die Komposition von f und g.

Lemma 1.

- g, f *injektiv* $\Rightarrow g \circ f$ *injektiv.*
- g, f *surjektiv* $\Rightarrow g \circ f$ *surjektiv.*

Lemma 2. *Es sei* $f : A \to B$ *eine bijektive Funktion.*
Dann gibt es eine Funktion

$$f^{-1} : B \to A,$$

so dass

$$\forall z \in A \quad f^{-1} \circ f(z) = z$$

und

$$\forall z \in B \quad f \circ f^{-1}(z) = z.$$

Definition 9. *Es sei* M *eine Menge.*

- M *heißt endlich, falls es ein* $n \in \mathbb{N}_0$ *gibt, so dass* M *bijektiv zu*

  $$M_n := \{1, 2, 3, \ldots, n\}.$$

  $|M| := n$ *ist dann die Anzahl der Elemente von* M*.*

- *Falls* M *nicht endlich ist, dann setzen wir* $|M| = \infty$*.*

Definition 10. *Es seien* M, N *Mengen.*

- M *heißt mindestens genauso mächtig wie* N *genau dann, wenn es eine injektive Abbildung* $g : N \to M$ *gibt. Wir schreiben dann*

  $$M \geq N.$$

- M *heißt mächtiger als* N*, falls*

  $$N \leq M \quad \textit{und} \quad \neg(M \leq N).$$

- M, N *heißen gleichmächtig, falls es eine bijektive Abbildung*

  $$f : M \to N$$

  gibt.

Satz 1. *Man ordne n Objekte in m Schubfächer. Das bedeutet es sei*

$$f : N \to M$$

eine Abbildung mit $|N| = n$ und $|M| = m$. Falls $m < n$, dann gibt es mindestens zwei unterschiedliche Objekte im selben Schubfach. Das heißt es gibt zwei unterschiedliche Objekte $a, b \in N$ mit $f(a) = f(b)$.

Beweis: Falls der Satz falsch ist, dann gibt es eine injektive Abbildung

$$\tilde{f} : \{1, ..., n\} \to \{1, ..., m\}$$

mit $m < n$. Dann gibt es auch eine injektive Abbildung

$$\begin{aligned} \hat{f} &:= \tilde{f}|_{\{1,...,m+1\}}, \\ \hat{f} &: \{1, ..., m+1\} \to \{1, ..., m\}. \end{aligned}$$

Wir beweisen durch Induktion, dass das nicht möglich ist.
$m = 1$:

$$\hat{f} : \{1, 2\} \to \{1\}.$$

Dann ist $\hat{f}(1) = 1 = \hat{f}(2)$. Also ist $\hat{f}$ nicht injektiv.
$m \to m + 1$:
Es sei

$$\hat{f} : \{1, ..., m+2\} \to \{1, ..., m+1\}.$$

Wir zeigen nun, dass $\hat{f}$ nicht injektiv ist.
1.Fall:

$$\hat{f}(\{1, ..., m+1\}) \subset \{1, ..., m\}$$

Dann ist

$$\hat{f}|_{\{1,...,m+1\}} \to \{1, ..., m\}$$

nicht injektiv und somit $\hat{f}$ auch nicht injektiv.
2.Fall:

$$\hat{f}(\{1, ..., m+1\}) \not\subset \{1, ..., m\}.$$

Dann gibt es ein x mit $1 \leq x \leq m + 1$ mit

$$\hat{f}(x) = m + 1.$$

Falls es ein $y \neq x$ gibt mit $\hat{f}(y) = m + 1$, dann ist $\hat{f}$ nicht injektiv. Ansonsten ist

$$\hat{f} = \hat{f}|_{\{1,...,m+2\}\setminus\{x\}} \to \{1, ..., m\}$$

eine Abbildung. Diese Abbildung ist nach Induktionsvoraussetzung nicht injektiv.
□.

Beispiel 10.

- *Unter 15 Kindern gibt es mindestens zwei, die im selben Monat Geburtstag haben.*
- *Unter sechs Personen gibt es mindestens drei die sich paarweise kennen oder paarweise nicht kennen. Hierbei wird angenommen, dass " sich kennen" eine symmetrische Eigenschaft ist.*

Definition 11.

- *Eine Menge M heißt abzählbar, wenn $\mathbb{N}$ mindestens so mächtig ist wie M. Das heißt M ist die leere Menge oder es gibt eine surjektive Abbildung $f : \mathbb{N} \to M$.*
- *Eine nicht abzählbare Menge heißt überabzählbar.*

Beispiel 11. $\mathbb{N} \times \mathbb{N}$ *ist abzählbar.*

Beweis:
Die Abbildung

$$\begin{aligned} f : \mathbb{N} &\to \mathbb{N} \times \mathbb{N} \\ f : n &\mapsto (a, b) \end{aligned}$$

$$f(n) = \begin{cases} (a,b) & \text{falls} \quad n = 2^a * 3^b \\ (1,1) & \text{sonst} \end{cases}$$

ist wohl definiert, da $2, 3$ Primzahlen und die Primfaktorzerlegung eindeutig ist.
$f : \mathbb{N} \mapsto \mathbb{N} \times \mathbb{N}$ ist surjektiv. □

Beispiel 12. *(Menge der Wörter einer Sprache)*
Es sei $\Sigma = \{a, b, ..., \rho\}$ eine endliche Menge.
Dann ist

$$\Sigma^+ := \bigcup_{n \in \mathbb{N}} \bigtimes_{i=1}^{n} \Sigma$$

abzählbar.

Beweis:
Es gilt:

$$|\bigtimes_{i=1}^{n} \Sigma| = |\Sigma|^n =: a_n.$$

Man kann also $\bigtimes_{i=1}^{n} \Sigma$ von 1 bis a_n durchnummerieren. Es sei $s(w)$ diese Nummer für $w \in \bigtimes_{i=1}^{n} \Sigma$. Dann setzen wir für $w \in \bigcup_{n \in \mathbb{N}} \bigtimes_{i=1}^{n} \Sigma$:

$$f(w) := s(w) + \sum_{i=1}^{n-1} a_i.$$

$f : \Sigma^+ \to \mathbb{N}$ ist eine Bijektion. Also ist f^{-1} surjektiv und somit Σ^+ abzählbar. □

Beispiel 13. *Die Menge M der Funktionen $\varphi : \mathbb{N} \to \mathbb{N}$ ist überabzählbar.*

Beweis:(Cantorsches Diagonalverfahren)
Nehmen wir an M ist abzählbar. Dann gibt es eine surjektive Abbildung

$$f : \mathbb{N} \to M.$$

Wir setzen nun

$$\begin{array}{rcl} d : \mathbb{N} & \to & \mathbb{N} \\ i & \mapsto & d(i) := f(i)(i) + 1. \end{array}$$

Also ist $d \in M$. Da f surjektiv ist, gibt es $j \in \mathbb{N}$ mit

$$f(j) = d$$

Dann folgt

$$f(j)(j) = d(j) = f(j)(j) + 1.$$

Dies ist ein Widerspruch. Damit ist M überabzählbar. □

1.3 Algebraische Grundbegriffe

Definition 12. *Es sei M eine Menge und*

$$\circ : M \times M \to M$$

eine Verknüpfung (Funktion).

- *Die Verknüpfung $\circ$ heißt assoziativ, falls* $(a \circ b) \circ c = a \circ (b \circ c)$,
- *und kommutativ, falls* $a \circ b = b \circ a$.
- $0_M \in M$ *heißt Nullelement, falls* $\forall a \in M \quad 0_M \circ a = a \circ 0_M = 0_M$.
- $1_M \in M$ *heißt Einselement, falls* $\forall a \in M \quad 1_M \circ a = a \circ 1_M = a$.
- $(M, \circ)$ *heißt* Halbgruppe*, falls* $\circ$ *assoziativ ist.*
- *Die Halbgruppe $(M, \circ)$ heißt* Monoid*, falls es in M ein Einselement 1_M gibt.*

Beispiel 14.
- $(\mathbb{N}, +)$ *ist kommutative Halbgruppe.*

- $(\mathbb{N}_0, +)$ *ist kommutatives Monoid mit Einselement* 0.
- $(\mathbb{N}, *)$ *ist kommutatives Monoid mit Einselement* 1.
- $(\mathbb{N} \setminus \{1\}, *)$ *ist kommutative Halbgruppe.*
- $(\mathbb{N}_0, *)$ *ist kommutative Halbgruppe mit Nullelement* 0.
- $\mathcal{F}_A = \{f : A \to A\}$ *die Menge der Funktionen von* A *nach* A *ist Monoid mit Einselement die Identität*

  $$Id_A : a \mapsto a$$

  und der Komposition $\circ$ *als Verknüpfung.*

Definition 13. *Es sei* $(M, \circ)$ *ein* Monoid.

- $a_i \in M$ *heißt inverses Element von* $a \in M$*, falls* $a \circ a_i = a_i \circ a = 1_M$*. Man schreibt dann* a^{-1} *für dieses Inverse* a_i*.*
- *Das Monoid* $(M, \circ)$ *heißt* Gruppe*, falls es zu jedem Element* $a \in M$ *ein inverses Element gibt.*
- *Die Gruppe* $(M, \circ)$ *heißt* kommutative Gruppe*, falls* $\circ$ *kommutativ.*

Beispiel 15.
- $(\mathbb{Z}, +)$ *ist kommutative Gruppe.*
- $(\mathbb{Q} - \{0\}, *)$ *ist kommutative Gruppe.*
- $(\mathbb{N}_0, +)$ *ist kommutatives Monoid mit Einselement* 0.
- $\mathcal{F}_{A,bijektiv} = \{f : A \to A | f \text{ ist bijektiv}\}$ *die Menge der Bijektionen von* A *nach* A *ist eine Gruppe mit Einselement die Identität und der Komposition* $\circ$ *als Verknüpfung.*
  Beachte: Dies ist i.A. keine kommutative Gruppe!

Satz 2. *a) Ein Monoid besitzt genau ein Einselement.*

b) Eine Gruppe besitzt für jedes Element genau ein inverses Element.

Beweis:

zu a) Es seien 1_G und $\tilde{1}_G$ Einselemente. Dann folgt

$$1_G = 1_G \circ \tilde{1}_G = \tilde{1}_G.$$

zu b) Es seien a und $\tilde{a}$ inverse Elemente zu b. Dann folgt

$$a = a \circ 1_G = a \circ b \circ \tilde{a} = 1_G \circ \tilde{a} = \tilde{a}.$$

Es sei Σ eine endliche Menge *(Alphabet)* .

Beispiel 16. *1.* $\Sigma = \{a, b\}$.

2. $\Sigma = \{a, b, ..., z\}$.

3. $\Sigma = \{\nabla, \oplus\}$.

4. $\Sigma = \{\alpha, \beta, \gamma\}$.

Dann sei

$$\Sigma^+ := \bigcup_{n\in\mathbb{N}} \bigtimes_{i=1}^{n} \Sigma.$$

Die Elemente von Σ^+ sind n-Tupel $(a_1, a_2, ..., a_n)$ mit $a_1, a_2, ..., a_n \in \Sigma$.

- Statt dieser n-Tupel schreiben wir kurz

$$a_1a_2...a_n$$

- $w = a_1a_2...a_n$ heißt ein Wort über Σ.
- $|w| = n$ ist die Länge des Wortes.

Beispiel 17. $\Sigma = \{a, b\}$. *Dann ist*

$$\Sigma^+ = \{a, b, aa, ab, ba, bb, aaa, ...\}.$$

Auch wenn Σ endlich ist, ist Σ^+ abzählbar unendlich (siehe Beispiel 12).

Definition 14. *Es sei* Σ *eine endliche Menge.*
Dann wird auf

$$\Sigma^+ = \{a_1a_2...a_n \mid a_1, a_2, ..., a_n \in \Sigma, n \in \mathbb{N}\}$$

die folgende Verknüpfung, genannt Konkatenation, von Wörtern definiert

$$\begin{aligned} \circ : \Sigma^+ \times \Sigma^+ &\to \Sigma^+ \\ (a_1a_2...a_n, b_1b_2...b_m) &\mapsto a_1a_2...a_nb_1b_2...b_m \end{aligned}$$

Wir schreiben kurz:

$$\begin{aligned} a^1 &:= a \\ a^{n+1} &:= a^n \circ a. \end{aligned}$$

Lemma 3. $(\Sigma^+, \circ)$ *ist eine Halbgruppe und heißt die von der Menge* Σ *frei erzeugte Halbgruppe.*

Beispiel 18.

- $\Sigma = \{a\}$. *Dann ist*

$$\Sigma^+ = \{a, aa, aaa, aaaa, ...\}.$$

- *Es sei* $x = kuh$ *und* $y = stall$. *Dann ist*

$$x \circ y = kuhstall$$

Definition 15.

- *Mit* ϵ *bezeichnen wir das leere Wort.*
- $|\epsilon| = 0$.
- $\Sigma^* = \Sigma^+ \cup \{\epsilon\}$.
- *Wir schreiben:* $a^0 := \epsilon$.
- *Die Konkatenation von Wörtern erweitern wir wie folgt*

$$\begin{array}{rcl} \circ : \Sigma^* \times \Sigma^* & \rightarrow & \Sigma^* \\ (a_1a_2...a_n, b_1b_2...b_m) & \mapsto & a_1a_2...a_nb_1b_2...b_m \\ (\epsilon, b_1b_2...b_m) & \mapsto & b_1b_2...b_m \\ (a_1a_2...a_n, \epsilon) & \mapsto & a_1a_2...a_n \end{array}$$

Lemma 4. $(\Sigma^*, \circ)$ *ist ein Monoid mit Einselement* ϵ.
$(\Sigma^*, \circ)$ *heißt das von der Menge* Σ *frei erzeugte Monoid.*

Beispiel 19. $\Sigma = \{a\}$. *Dann ist*

$$\begin{array}{rcl} \Sigma^* & = & \{\epsilon, a, aa, aaa, aaaa, ...\} \\ & = & \{a^n \mid n \in \mathbb{N}_0\}. \end{array}$$

Die Abbildung

$$\begin{array}{rcl} \varphi : \mathbb{N}_0 & \rightarrow & \Sigma^* \\ n & \mapsto & a^n \end{array}$$

ist eine Bijektion und hat die Eigenschaft

$$\varphi(n + m) = \varphi(n) \circ \varphi(m).$$

$\rightarrow$ $(\Sigma^*, \circ)$ *hat die selbe Struktur wie* $(\mathbb{N}_0, +)$*!*

Definition 16.
- *u heißt* Präfix *eines Wortes w genau dann wenn* $\exists r \in \Sigma^* \; ur = w$.
- *u heißt* Infix *eines Wortes w genau dann wenn* $\exists r, s \in \Sigma^* \; rus = w$.
- *u heißt* Suffix *eines Wortes w genau dann wenn* $\exists r \in \Sigma^* \; ru = w$.

Beispiel 20.
- `++` *ist Präfix des Wortes* `++i` .
- `++` *ist Suffix des Wortes* `i++` .
- `versit` *ist Infix des Wortes* `Universität`.

Definition 17. *Es seien $(G, \otimes)$ und $(H, \circ)$ Monoide und es sei*

$$\begin{aligned} \varphi : G &\to H \\ x &\mapsto \varphi(x) \end{aligned}$$

eine Abbildung mir der Eigenschaft

$$\forall x, y \in G \;\; \varphi(x \otimes y) = \varphi(x) \circ \varphi(y).$$

Dann heißt φ

- *Homomorphismus,*
- *Isomorphismus, falls φ bijektiv und*
- *Epimorphismus, falls φ surjektiv.*

Beispiel 21.
- *Die Abbildung*

$$\begin{aligned} \varphi : \mathbb{N}_0 &\to \{a\}^* \\ n &\to a^n \end{aligned}$$

  im Beispiel 19 ist ein Isomorphismus.

- *Die Abbildung*

$$\begin{aligned} \mathbb{N}_0 \times \mathbb{N}_0 &\to \{a, b\}^* \\ (n, m) &\mapsto a^n \circ b^m \end{aligned}$$

  im Beispiel 19 ist kein Homomorphismus, da a und b nicht vertauschen.

Satz 3. *Es sei $\varphi : G \to H$ ein Homomorphismus zwischen zwei Gruppen $(G, \otimes)$ und $(H, \circ)$. Dann gilt*

a) $\varphi(1_G) = 1_H$.

b) $\forall a \in G \quad \varphi(a^{-1}) = \varphi(a)^{-1}$.

Beweis:

zu a)

$$\begin{aligned} \varphi(1_G) &= 1_H \circ \varphi(1_G) = \\ (\varphi(1_G))^{-1} \circ \varphi(1_G) \circ \varphi(1_G) &= (\varphi(1_G))^{-1} \circ \varphi(1_G \otimes 1_G) = (\varphi(1_G))^{-1} \circ \varphi(1_G) \\ &= 1_H \end{aligned}$$

zu b) Wegen Satz 2 gilt:

$$\begin{aligned} 1_H &= \varphi(1_G) = \varphi(a \otimes a^{-1}) \\ &= \varphi(a) \circ \varphi(a^{-1}) \\ &\Rightarrow (\varphi(a))^{-1} = \varphi(a^{-1}) \end{aligned}$$

Satz 4. *Es sei $(H, \cdot)$ ein Monoid und $E = \{h_1, ..., h_n\} \subset H$ ein Erzeugendensystem. Dies bedeutet,*

$$\forall h \in H \quad \exists h_{i_1}, h_{i_2}, ..., h_{i_m} \in E \quad (h = h_{i_1} \cdot h_{i_2} \cdot ... \cdot h_{i_m}).$$

Dann ist für $\Sigma = \{h_1, ..., h_n\}$

$$\begin{aligned} \varphi : \Sigma^* &\to H \\ h_{i_1} \circ h_{i_2} \circ ... \circ h_{i_m} &\mapsto h_{i_1} \cdot h_{i_2} \cdot ... \cdot h_{i_m} \end{aligned}$$

ein Epimorphismus.

Beweis: φ ist surjektiv, da $h_1, ..., h_n$ ein Erzeugendensystem ist.

$$\begin{aligned} &\varphi((h_{i_1} \circ h_{i_2} \circ ... \circ h_{i_m}) \circ (h_{j_1} \circ h_{j_2} \circ ... \circ h_{j_l})) = \\ &= \varphi(h_{i_1} \circ h_{i_2} \circ ... \circ h_{i_m} \circ h_{j_1} \circ h_{j_2} \circ ... \circ h_{j_l}) \\ &= h_{i_1} \cdot h_{i_2} \cdot ... \cdot h_{i_m} \cdot h_{j_1} \cdot h_{j_2} \cdot ... \cdot h_{j_l} \\ &= (h_{i_1} \cdot h_{i_2} \cdot ... \cdot h_{i_m}) \cdot (h_{j_1} \cdot h_{j_2} \cdot ... \cdot h_{j_l}) \\ &= \varphi(h_{i_1} \circ h_{i_2} \circ ... \circ h_{i_m}) \cdot \varphi(h_{j_1} \circ h_{j_2} \circ ... \circ h_{j_l}). \end{aligned}$$

Beispiel 22. *Es sei* $\Sigma = \{a, b, a^{-1}, b^{-1}\}$. *Dann sei*

$$\begin{aligned} H = \{z \in \Sigma^* \quad | \quad & \neg\exists p, q \in \Sigma^* \quad z = paa^{-1}q \wedge \\ & \neg\exists p, q \in \Sigma^* \quad z = pa^{-1}aq \wedge \\ & \neg\exists p, q \in \Sigma^* \quad z = pbb^{-1}q \wedge \\ & \neg\exists p, q \in \Sigma^* \quad z = pb^{-1}bq \ \}. \end{aligned}$$

Es sei nun

$$\begin{aligned} \rho : \Sigma^* &\rightarrow \Sigma^* \\ \rho(q_1 q_2 ... q_t) &:= q_1 q_2 ... q_s q_{s+3} ... q_t \quad \textit{für } q_1 q_2 ... q_t \in \Sigma, \end{aligned}$$

wobei $q_{s+1}q_{s+2}$ *von links das erste Infix in* $q_1q_2...q_t$ *von der Form* aa^{-1} *oder* $a^{-1}a$ *oder* bb^{-1} *oder* $b^{-1}b$ *ist. Wir definieren nun rekursiv über die Länge von Wörtern folgende Funktion*

$$\begin{aligned} \varphi : \Sigma^* &\rightarrow H \\ \varphi(q) &:= q \quad \textit{falls } q \in \Sigma. \\ \varphi(q) &:= q \quad \textit{falls } \rho(q) = q \in \Sigma^*. \\ \varphi(q) &:= \varphi(\rho(q)) \quad \textit{falls } \rho(q) \neq q \in \Sigma^*. \end{aligned}$$

H mit der Verknüpfung

$$p \cdot q := \varphi(p \circ q)$$

ist eine Gruppe und $\varphi : \Sigma^* \rightarrow H$ *ein Homomorphismus.*

Definition 18. • *Eine Sprache L ist eine Menge von Wörtern über einem Alphabet* Σ. *Das bedeutet* $L \subset \Sigma^*$.

Beispiel 23. • Σ^* *ist eine Sprache über* Σ.

- `C` *ist eine Sprache über* $\{a, b, ..., z, A, ..., Z, ., :, ...\}$. *Dies bedeutet: Jeder korrekte* `C`*-Code ist ein Wort der Sprache* `C`.
- $\{a^{n^2} | n \in \mathbb{N}\}$ *ist eine Sprache über* $\{a\}$.
- *Es sei* $\Sigma = \{a, b, a^{-1}, b^{-1}\}$. *Dann ist*

$$\begin{aligned} \{z \in \Sigma^* \quad | \quad & \neg\exists p, q \in \Sigma^* \quad z = paa^{-1}q \wedge \\ & \neg\exists p, q \in \Sigma^* \quad z = pa^{-1}aq \wedge \\ & \neg\exists p, q \in \Sigma^* \quad z = pbb^{-1}q \wedge \\ & \neg\exists p, q \in \Sigma^* \quad z = pb^{-1}bq \ \} \end{aligned}$$

  eine Sprache über Σ.

Definition 19.

- *Die Konkatenation zweier Sprachen über Σ ist*

$$L_1 \circ L_2 := \{uv \mid u \in L_1, v \in L_2\}.$$

Wir schreiben kurz $L_1 L_2 := L_1 \circ L_2$.

- *Die Iteration einer Sprache ist wie folgt rekursiv definiert*

$$L^0 := \{\epsilon\}, \quad L^{n+1} := LL^n.$$

- *Der Kleene-*-Abschluß einer Sprache L ist $L^* := \bigcup_{i \geq 0} L^i$.*
- *Der Kleene-+-Abschluß einer Sprache L ist $L^+ := \bigcup_{i \geq 1} L^i$.*

Beispiel 24.

$$\{a\}^* \circ \{b\}^* = \{a^n \circ b^m \mid n, m \in \mathbb{N}_0\} \neq \{a, b\}^*.$$

Aber es ist

$$(\{a\}^* \circ \{b\}^*)^* = \{a, b\}^*.$$

1.4 Relationen und Graphen

Definition 20. *Eine n-stellige* Relation *über den Mengen $M_1, \cdots, M_n$ ist eine Menge $R \subset \bigtimes_{i=1}^{n} M_i$.*

Beispiel 25. *Mitarbeiter einer Firma, deren Ausbildungsgrad und Geschlecht.*

$$\begin{aligned} M_1 &= \{\textit{Müller}, \textit{Heinz}, \textit{Kraft}, \textit{Ruben}, \textit{Huber}\}, \\ M_2 &= \{\textit{Ing.}, \textit{Dr.}, \textit{Meister}, \textit{Geselle}\}, \\ M_3 &= \{\textit{weiblich}, \textit{männlich}\}. \end{aligned}$$

$$\begin{aligned} R = \quad & \{(\textit{Heinz}, \textit{Dr.}, \textit{weiblich}), \quad (\textit{Kraft}, \textit{Meister}, \textit{männlich}), \\ & (\textit{Müller}, \textit{Ing.}, \textit{männlich}), \quad (\textit{Heinz}, \textit{Ing.}, \textit{weiblich}), \\ & (\textit{Ruben}, \textit{Geselle}, \textit{weiblich}), \quad (\textit{Huber}, \textit{Geselle}, \textit{männlich})\}. \end{aligned}$$

Schreibweise:
Sei $R \subset M \times M$ eine binäre Relation.
Dann schreibt man oft aRb anstelle von $(a, b) \in R$.

Beispiel 26. *Mathematische Beispiele sind*

- *Gleichheits-Zeichen für die Elemente einer Menge A:*

$$\begin{aligned} \Delta &= \{(a,a) \in A \times A \mid a \in A\} \\ a = b &:\Leftrightarrow (a,b) \in \Delta. \end{aligned}$$

- *Größer-Zeichen für die Elemente einer bezüglich $>$ geordneten Menge A:*

$$>:= \{(a,b) \in A \times A \mid a \text{ ist größer als } B.\}.$$

Beispiel 27. *Ein weiteres mathematische Beispiel ist*

- *die Konkruenz modulo k. Es sei $k \in \mathbb{N}$ eine natürliche Zahl. Wir sagen $a, b \in \mathbb{Z}$ sind kongruent modulo k, falls a, b beim Teilen durch k den gleichen Rest ergeben.*

$$\begin{aligned} K &:= \{(a,b) \in \mathbb{Z} \times \mathbb{Z} \mid (a-b)/k \in \mathbb{Z}\}, \\ a \equiv b \text{ modulo } k &:\Leftrightarrow (a,b) \in K. \end{aligned}$$

Definition 21. *Eine zweistellige Relation $\tau \subset A \times A$ heißt*

- **reflexiv** *falls $\forall a \in A\ a\tau a$.*
- **irreflexiv** *falls $\forall a \in A\ \neg(a\tau a)$.*
- **symmetrisch** *falls $\forall a, b \in A\ (a\tau b \Rightarrow b\tau a)$.*
- **antisymmetrisch** *falls $\forall a, b \in A\ (a\tau b\ \wedge\ b\tau a\ \Rightarrow\ a = b)$.*
- **asymmetrisch** *falls $\forall a, b \in A\ (a\tau b \Rightarrow \neg(b\tau a))$.*
- **transitiv** *falls $\forall a, b, c \in A\ (a\tau b\ \wedge\ b\tau c\ \Rightarrow\ a\tau c)$.*
- *Äquivalenzrelation falls sie reflexiv, symmetrisch und transitiv ist.*
- *Ordnungsrelation, falls sie reflexiv, antisymmetrisch und transitiv ist.*
- *strikte Ordnungsrelation, falls sie irreflexiv und transitiv ist ($\Rightarrow$ asymmetrisch).*

Weitere Eigenschaften von Restklassen modulo k

- $a \equiv b$ modulo k ist eine Äquivalenzrelation.

- Zu jeder Zahl $a \in \mathbb{Z}$ gibt es genau eine Zahl $\phi(a)$ aus

$$\mathbb{Z}_k = \{0, 1, 2, ..., k-1\},$$

  so dass $a \equiv \phi(a)$ modulo k.

- Rechnen auf $\mathbb{Z}_k$: $\quad a + b \equiv \phi(a + b)$ modulo k.

Definition 22. *Es sei $\tau \subset A \times A$ eine Äquivalenzrelation. Dann heißen die Mengen*

$$[a]_\tau = \{b \in A \mid a \tau b\}$$

für $a \in A$ Äquivalenzklassen. *Die Menge aller Äquivalenzklassen heißt* Partitionierung *von A und wird mit A/τ bezeichnet:*

$$A/\tau := \{[a]_\tau \mid a \in A\}.$$

Jedes $b \in [a]_\tau$ heißt Repräsentant von $[a]_\tau$.

Beispiel 28.

- *Restklassen modulo k*

$$[a]_{\equiv \text{ modulo } k}$$

- *Restklassen modulo* 2

$$\begin{aligned} [0]_{\equiv \text{ modulo } 2} &= \{..., -4, -2, 0, 2, 4, ...\} \\ [1]_{\equiv \text{ modulo } 2} &= \{..., -3, -1, 1, 3, 5, ...\} \end{aligned}$$

- *Restklassen modulo* 4

$$\begin{aligned} [0]_{\equiv \text{ modulo } 4} &= \{..., -8, -4, 0, 4, 8, ...\} \\ [1]_{\equiv \text{ modulo } 4} &= \{..., -7, -3, 1, 5, 9, ...\} \\ [2]_{\equiv \text{ modulo } 4} &= \{..., -6, -2, 2, 6, 10, ...\} \\ [3]_{\equiv \text{ modulo } 4} &= \{..., -5, -1, 3, 7, 11, ...\}. \end{aligned}$$

Die Menge der Restklassen modulo k ($\mathbb{Z}/\equiv$ modulo k) bildet mit folgender Verknüpfung

$$[a]_{\equiv} \text{ modulo } k \oplus [b]_{\equiv} \text{ modulo } k := [a + b]_{\equiv} \text{ modulo } k \cdot$$

eine Gruppe. Man muß zeigen, dass diese Verknüpfung $\oplus$ wohldefiniert ist.

Wir schreiben kurz

$$a \oplus b = c$$

für $0 \leq a, b, c \leq p - 1$ und c Repräsentant von $[a + b]_{\equiv}$ modulo $k \cdot$

Beispiel 29. *Die Addition $\oplus$ der Restklassen modulo 4 ergibt folgende Tabelle*

$\oplus$	*0*	*1*	*2*	*3*
0	*0*	*1*	*2*	*3*
1	*1*	*2*	*3*	*0*
2	*2*	*3*	*0*	*1*
3	*3*	*0*	*1*	*2*

Betrachte folgende Drehungen:

Φ_0: Drehung um 0°

Φ_2: Drehung um 180°

Φ_1: Drehung um 90°

Φ_3: Drehung um 270°

Die Menge der Drehungen $\mathcal{D} = \{\Phi_0, \Phi_1, \Phi_2, \Phi_3\}$ mit der Hintereinanderausführung $\circ$ ist eine Gruppe.

Man sieht, dass die Abbildung

$$\begin{aligned} \varphi : \mathbb{Z}/\equiv \text{ modulo } 4 &\rightarrow \mathcal{D} \\ [a]_{\equiv} \text{ modulo } 4 &\mapsto \Phi_a, \end{aligned}$$

wobei $a = 0, 1, 2, 3$, ein Isomorphismus ist.

Definition 23. *Es sei V eine Menge, die Menge der Knoten (vertices) genannt wird und $E \subset V \times V$ eine Relation, die Kanten (edges) genannt wird.*

- *Dann heißt $G = (V, E)$ Graph.*
- *G heißt ungerichtet, falls $\forall a, b \in V \ ((a, b) \in E \Rightarrow (b, a) \in E)$.*
- *Ein nicht ungerichteter Graph heißt gerichteter Graph.*

Beispiel 30.

$$\begin{aligned} V &= \{V1, V2, V3, V4, V5\} \\ E &= \{e1, e2, e3, e4, e5\} \end{aligned}$$

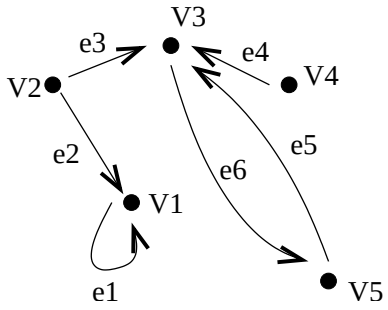


Abbildung 1: Ein Graph.

$$\begin{aligned} e1 &= (V1, V1) \\ e2 &= (V2, V1) \\ e3 &= (V2, V3) \\ e4 &= (V4, V3) \\ e5 &= (V5, V3) \\ e6 &= (V3, V5) \end{aligned}$$

Definition 24. *Es sei (V, E) ein Graph.*

- *Falls $v, v' \in V$, dann schreibt man dafür $v \to v'$. v heißt Vater von v' und v' Sohn von v.*
- *Ein nicht-leeres Wort $v_1 v_2 ... v_n$ heißt Weg (der Länge $n-1$) von v_1 nach v_n falls $\forall 1 \leq i < n$ $v_i \to v_{i+1}$.*
- *v heißt Vorfahre von v' falls es einen Weg von v nach v' gibt. v' heißt dann Nachkomme von v.*
- *v_1 und v_2 heißen Brüder falls es eine Knoten v gibt, der Vater von v_1 und v_2 ist.*
- *Ein Weg $v_1 v_2 ... v_n v_1$ heißt Kreis.*

Definition 25. *Ein Baum ist ein gerichteter Graph (V, E) mit einem ausgezeichneten Knoten $v_0 \in V$, der Wurzel des Baumes genannt wird,*
so dass folgende Eigenschaften gelten

- *v_0 ist Vorfahre aller $v \in V$.*
- *Alle Knoten $v \neq v_0$ haben genau einen Vater.*

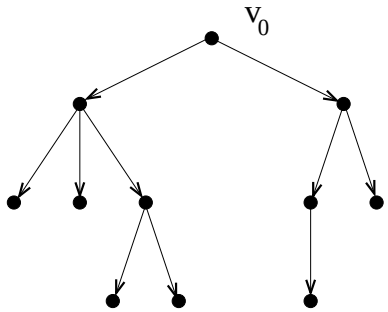


Abbildung 2: Ein Baum.

- v_0 *hat keinen Vater.*

Satz 5. *Ein Baum enthält keinen Kreis.*

Beweis: Es sei (V, E) ein Baum. Wir nehmen an es gibt einen Kreis $w_0 w_1 ... w_n w_0$. Da v_0 Vorfahre jeder Ecke, gibt es einen Weg $v_0 v_1 ... v_s$, wobei $v_s = w_0$. Es ist nicht möglich, dass $v_0 = w_j$ für ein j, da ansonsten v_0 einen Vater hätte. Daher gibt es j, i, so dass $v_i = w_j$ und $v_{i-1} \neq w_{j-1}$. Dann sind w_{j-1} und v_{i-1} Vater von v_i. Dies ist ein Widerspruch dazu, dass (V, E) ein Baum ist.

Definition 26. *Es sei $B = (V, E, v_0)$ ein Baum.*

- *Ein innerer Knoten ist ein Knoten mit mindestens einem Sohn.*
- *Ein Blatt ist ein Knoten ohne Söhne.*
- *Ein Ast ist ein Weg von einem Knoten zu einem Blatt.*
- *Die Tiefe von B ist die maximale Länge eines Weges von v_0 zu einem Blatt.*

Beispiel 31. *Zu jedem logischen Ausdruck gibt es einen Syntaxbaum, der angibt, wie der Ausdruck syntaktisch aufgebaut ist. Zum Beispiele ist der Syntax Baum zu*

$$(A \vee (B \wedge C)) \vee (\neg B \vee C)$$

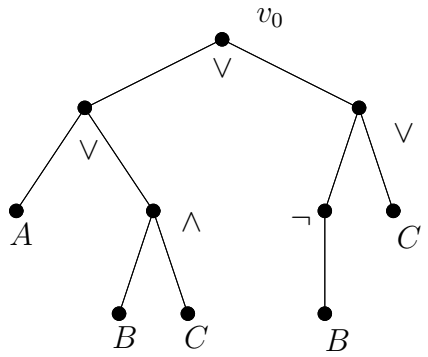


Abbildung 3: Ein Syntaxbaum

2 Aussagenlogik

2.1 Syntax der Aussagenlogik

Es gibt Aussagen ohne Quantoren und Aussagen mit Quantoren. In der Aussagenlogik geht es um Aussagen ohne Quantoren.

Definition 27. *Es sei VAR eine nicht leere abzählbare Menge. Diese Menge nennen wir atomare Formeln oder Atome oder Variablen (z.B. x, y, z, x_i, y_i, z_k, ...). Eine Formel in der Aussagenlogik (über VAR) ist induktiv wie folgt definiert:*

- *Jede atomare Formel aus VAR ist eine Formel.*
- *Falls F, G Formeln der Aussagenlogik, dann sind auch*

  $$\neg F, \quad (F \wedge G) \quad \textit{und} \quad (F \vee G).$$

  Formeln der Aussagenlogik.
- *$\mathcal{F}_{AL}$ sei die Menge der Formeln der Aussagenlogik (über VAR).*
- *$VAR(F)$ sei die Menge der atomaren Formeln die in der Formel F vorkommen.*

Wir verwenden folgende Abkürzungen in der Aussagenlogik:

- $(F \Rightarrow G)$ für $(\neg F \vee G)$.
- $(F \Leftrightarrow G)$ für $((F \Rightarrow G) \wedge (G \Rightarrow F))$.
- $F_1 \wedge F_2 \wedge ... \wedge F_n$ für $\Big((...(F_1 \wedge F_2) \wedge ...) \wedge F_n\Big)$.

- $F_1 \vee F_2 \vee ... \vee F_n$ für $\Big((...(F_1 \vee F_2) \vee ...) \vee F_n\Big)$.

Den Namen $\neg$, $\wedge$ und $\vee$ werden folgende Namen gegeben:

- $\neg F$ heißt Negation.
- $F \wedge G$ heißt Konjunktion.
- $F \vee G$ heißt Disjunktion.
- Eine atomare Formel oder Negation einer atomaren Formel heißt Literal.

2.2 Semantik der Aussagenlogik

Aus der Definition 27 ist nicht klar, was $\neg$, $\wedge$ und $\vee$ bedeuten. Um diesen Zeichen eine Bedeutung zu geben, benötigt man die Semantik der Aussagenlogik.

Ein Wahrheitswert ist ein Element der Menge $\{\texttt{true}, \texttt{false}\}$.

Eine Belegung ist eine Abbildung

$$\mathcal{A} : VAR \to \{\texttt{true}, \texttt{false}\}.$$

Wir erweitern $\mathcal{A}$ wie folgt:

$$\begin{array}{rcl} \mathcal{A} : \mathcal{F}_{AL} & \to & \{\texttt{true}, \texttt{false}\} \\ \mathcal{A}(\neg F) = \texttt{true} & :\Leftrightarrow & \mathcal{A}(F) = \texttt{false} \\ \mathcal{A}(F \wedge G) = \texttt{true} & :\Leftrightarrow & \mathcal{A}(F) = \texttt{true} \quad \text{und} \quad \mathcal{A}(G) = \texttt{true} \\ \mathcal{A}(F \vee G) = \texttt{true} & :\Leftrightarrow & \mathcal{A}(F) = \texttt{true} \quad \text{oder} \quad \mathcal{A}(G) = \texttt{true} \end{array}$$

Beispiel 32. *Es sei* $VAR = \{a, b, c\}$ *und*

$$\mathcal{A}(a) = \texttt{true}, \quad \mathcal{A}(b) = \texttt{false}, \quad \mathcal{A}(c) = \texttt{true}.$$

Dann ist

$$\begin{array}{rcl} \mathcal{A}(a \vee (b \wedge \neg c)) & = & \texttt{true} \\ \mathcal{A}(a \wedge (b \vee \neg c)) & = & \texttt{false}. \end{array}$$

Dieses Beispiel zeigt, dass eine Formel nicht immer wahr sein muß. Ob eine Formel wahr ist hängt von der Belegung ab. Dies führt zu folgender Definition.

Definition 28.
- *Ein Modell für eine Formel F ist eine Belegung $\mathcal{A}$, so dass diese Formel wahr ist:*

$$\mathcal{A}(F) = \texttt{true}.$$

- *Eine Formel F heißt erfüllbar, wenn sie mindestens ein Modell besitzt.*
- *Wenn F von jeder Belegung erfüllt wird, dann heißt F Tautologie.*

Satz 6. *Eine Formel F ist eine Tautologie genau dann wenn $\neg F$ nicht erfüllbar.*

Beispiel 33. *Es sei wieder $VAR = \{a, b, c\}$ und*

$$\mathcal{A}(a) = \texttt{true}, \quad \mathcal{A}(b) = \texttt{false}, \quad \mathcal{A}(c) = \texttt{true}.$$

Dann ist

- *$\mathcal{A}$ ein Modell für $a \vee (b \wedge \neg c)$.*
- *$\mathcal{A}$ kein Modell für $a \wedge (b \vee \neg c)$.*

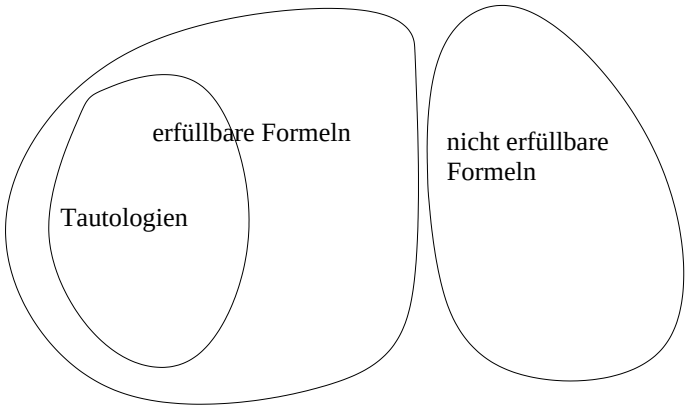


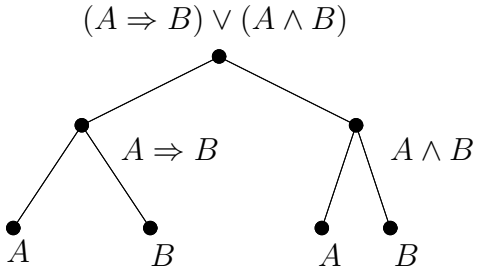
Abbildung 4: Tautologien und (nicht) erfüllbare Formeln.

2.3 Auswertung von Formeln

Der Syntaxbaum von

$$F := (A \Rightarrow B) \vee (A \wedge B)$$

ist folgender Baum



Um diese Formel auszuwerten lege man eine Tabelle an, die zu jedem Knoten eine Spalte enthält. Links schreibt man die Spalten zu den Blättern des Syntaxbaums und rechts die Spalte der Wurzel.

A	B	$A \Rightarrow B$	$A \wedge B$	F
true	true	true	true	true
true	false	false	false	false
false	true	true	false	true
false	false	true	false	true

Satz 7. *Es gibt einen Algorithmus, der entscheidet ob eine Formel $F \in \mathcal{F}_{AL}$ eine Tautologie ist.*

2.4 Äquivalenz von Formeln

Definition 29. *Zwei Formeln F, G aus der Aussagenlogik heißen äquivalent, wenn für alle Belegungen $\mathcal{A}$ gilt:*

$$\mathcal{A}(F) = \texttt{true} \Leftrightarrow \mathcal{A}(G) = \texttt{true}$$

In Zeichen: $F \hat{=} G$.

Beispiel 34.

Idempotenzen:	$F \wedge F$	$\hat{=}$	F
Kommutativität:	$F \wedge G$	$\hat{=}$	$G \wedge F$
de Morgan:	$\neg(F \wedge G)$	$\hat{=}$	$\neg G \vee \neg F$
	$\neg(F \vee G)$	$\hat{=}$	$\neg G \wedge \neg F$
Distributivität:	$F \vee (G \wedge H)$	$\hat{=}$	$(F \vee G) \wedge (F \vee H)$

Satz 8.

$$\begin{aligned} \neg(F_1 \vee ... \vee F_n) &\hat{=} \neg F_1 \wedge ... \wedge \neg F_n \\ \neg(F_1 \wedge ... \wedge F_n) &\hat{=} \neg F_1 \vee ... \vee \neg F_n. \end{aligned}$$

Mit Hilfe von Wahrheitstafeln kann man feststellen ob zwei Formeln äquivalent sind. Wichtig ist es auch zu wissen, ob es zu einer gegebenen Formel eine äquivalente kürzere Formel gibt. Um diese Frage zu beantworten helfen die Normalformen KNF und DNF.

Definition 30.

- *Eine Formel ist in konjunktiver Normalform (KNF) genau dann wenn F ein Konjunktion von Disjunktionen von Literalen ist.*
- *Eine Formel ist in disjunktiver Normalform (DNF) genau dann wenn F ein Disjunktion von Konjunktionen von Literalen ist.*

Das heißt:

- Eine Formel in KNF hat die Form $\bigwedge_{i=1}^{m} \bigvee_{j=1}^{n} L_{ij}$
- Eine Formel in DNF hat die Form $\bigvee_{i=1}^{m} \bigwedge_{j=1}^{n} L_{ij}$

hierbei sind L_{ij} Literale.

Beispiel 35. *Die Formel* $F = \neg(x \vee (y \wedge (x \vee \neg z)))$ *ist in keiner Normalform.*

- *Die KNF von* F *ist*
$$(\neg x \vee \neg x) \wedge (\neg y \vee z)$$
- *Die DNF von* F *ist*
$$(\neg x \wedge \neg y) \vee (\neg x \wedge z)$$

Satz 9. *Zu jeder Formel* F *in der Aussagenlogik gibt es eine äquivalente Formel in KNF und eine äquivalente in DNF.*

Beweis: Die Formel F besitze die Variablen $\{x_1, ..., x_n\} = VAR$. Es sei $\mathcal{B}$ die Menge aller Belegungen:

$$\mathcal{B} := \{\mathcal{A} : VAR \to \{\texttt{true}, \texttt{false}\}\}.$$

Es sei nun

$$\begin{array}{rcl} \Psi : \mathcal{B} & \to & \{\texttt{true}, \texttt{false}\} \\ \mathcal{A} & \mapsto & \mathcal{A}(F). \end{array}$$

Die Urbildmenge

$$\Psi^{-1}(\texttt{true}) := \{\mathcal{A} | \Psi(\mathcal{A}) = \texttt{true}\}$$

ist endlich. Des weiteren seien folgende Literale definiert:

$$l_{\mathcal{A},j} := \begin{cases} x_j & \text{falls} \quad \mathcal{A}(x_j) = \texttt{true} \\ \neg x_j & \text{falls} \quad \mathcal{A}(x_j) = \texttt{false} \end{cases}$$

Die KNF von F ist nun:

$$F_{KNF} := \bigvee_{\mathcal{A} \in \Psi^{-1}(\texttt{true})} \bigwedge_{j=1}^{n} l_{\mathcal{A},j}.$$

Dies sieht man wie folgt.

Es sei $\mathcal{A}$ eine Belegung mit $\mathcal{A}(F) = \texttt{true}$. Dann ist $\mathcal{A}(l_{\mathcal{A},j}) = \texttt{true}$ für jedes $j = 1, ..., n$. Also $\mathcal{A}(F_{KNF}) = \texttt{true}$.

Es sei nun $\mathcal{A}(F) = \texttt{false}$. Dann gibt es für jede Belegung $\mathcal{A}' \in \Psi^{-1}(\texttt{true})$ mit $\mathcal{A}' \neq \mathcal{A}$ ein j mit $\mathcal{A}'(l_{\mathcal{A},j}) = \texttt{false}$. Also $\mathcal{A}(F_{KNF}) = \texttt{false}$.

□

3 Einführung in die Graphentheorie

3.1 Ein paar Beispiele zur Motivation

- Es sei eine Landkarte mit m Ländern gegeben. Jedes Land wird mit einer Farbe markiert, so dass zwei benachbarte Länder unterschiedliche Farben haben. Wie viele Farben benötigt man hierzu?
- Gegeben sei eine Landkarte mit m Orten. Diese Orte seien durch Straßen verbunden. Was ist der kürzeste Weg von einem Ort A zu einem Ort B?
- Gegeben seien m Orte die durch Flüsse getrennt seien. Die Flüsse kann man mit Hilfe von bestimmten Brücken überqueren. Ist es möglich von einem Ort zum anderen zu gehen und dabei jede Brücke genau einmal zu überqueren?

3.2 Reduktion eines Gleichungssystems

Problem 1. *Es sei A eine invertierbare $n \times n$ Matrix und $b \in \mathbb{R}^n$ ein n-Vektor. Gesucht ist ein Vektor $x \in \mathbb{R}^n$, so dass*

$$Ax = b.$$

Dieses Problem kann man durch Invertierung der Matrix A lösen:

$$x = A^{-1}b.$$

A^{-1} ist jedoch oft schwer zu berechnen!

A^{-1} ist in folgender Situation etwas leichter zu berechnen:

A habe folgende Blockstruktur $A = \begin{pmatrix} A_{oo} & A_{ou} \\ 0 & A_{uu} \end{pmatrix}$.

Hierbei ist A_{oo} eine $s_o \times s_o$-Matrix, A_{uu} eine $s_u \times s_u$-Matrix, A_{ou} eine $s_o \times s_u$-Matrix und $s_o + s_u = n$.
Da A invertierbar, sind auch A_{oo} und A_{uu} invertierbar.
Wir schreiben nun x und b in der Form:

$$x = \begin{pmatrix} x_o \\ x_u \end{pmatrix}, \quad b = \begin{pmatrix} b_o \\ b_u \end{pmatrix}.$$

Hierbei sind $x_o, b_o \in \mathbb{R}^{s_o}$ und $x_u, b_u \in \mathbb{R}^{s_u}$.
Das Gleichungssystem $\begin{pmatrix} A_{oo} & A_{ou} \\ 0 & A_{uu} \end{pmatrix} \begin{pmatrix} x_o \\ x_u \end{pmatrix} = \begin{pmatrix} b_o \\ b_u \end{pmatrix}$
kann man nun wie folgt lösen:

$$\begin{aligned} x_u &= A_{uu}^{-1} b_u \\ x_o &= A_{oo}^{-1}(b_o - A_{ou}x_u). \end{aligned}$$

Eine Matrix kann durch eine andere Nummerierung der Zeilen und Spalten in die Blockstruktur

$$A = \begin{pmatrix} A_{oo} & A_{ou} \\ 0 & A_{uu} \end{pmatrix}$$

gebracht werden, falls A reduzierbar ist.

Definition 31. *$A = (a_{ij})_{1 \le i,j \le n}$ heißt reduzierbar, falls es eine Menge $J \overset{\neq}{\subset} \{1, 2, \ldots, n\}, J \neq \emptyset$ gibt, so dass*

$$a_{ij} = 0 \qquad \text{für alle } i \notin J,\ j \in J.$$

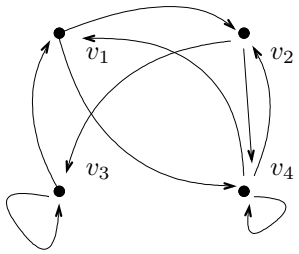
Definition 32. *Es sei $A = (a_{ij})_{1 \le i,j \le n}$ eine $n \times n$ Matrix. Der Graph zu A besteht aus den Knoten $V = \{1, 2, ..., n\}$. Die Kantenmenge E sei durch folgende Relation definiert:*

$$i \to j :\Leftrightarrow a_{i,j} \neq 0.$$

Der gerichtete Graph $G = (V, E)$ ist der zu A gehörige Graph.

Beispiel 36. *Der Graph zu*

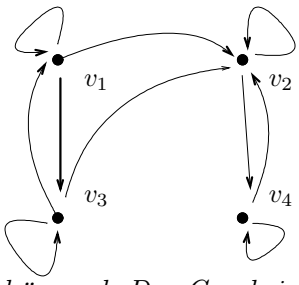
$$\begin{pmatrix} 0 & 1 & 0 & 4 \\ 0 & 0 & 3 & 4 \\ 1 & 0 & 1 & 0 \\ -1 & 2 & 0 & 3 \end{pmatrix} \text{ ist:}$$



Definition 33. *Ein gerichteter Graph heißt* stark zusammenhängend*, falls für jedes Paar unterschiedlicher Knoten v, w ein gerichteter Weg existiert. Dies bedeutet, dass ein Weg $v_0 v_1 ... v_{r-1} v_r$ mit $v_0 = v$ und$v_r = w$ existiert.*

Beispiel 37. *Der Graph zu*

$$\begin{pmatrix} 1 & 4 & 1 & 0 \\ 0 & 7 & 0 & 1 \\ 3 & 5 & 1 & 0 \\ 0 & 5 & 0 & 1 \end{pmatrix}$$ *ist:*



Dieser Graph ist stark zusammenhängend. Der Graph in Beispiel 36 ist nicht stark zusammenhängend.

Satz 10. *Eine $n \times n$ Matrix A ist nicht reduzierbar, genau dann wenn ihr gerichteter Graph G stark zusammenhängend ist.*

Beweis:
Es sei $G = (V, E)$ der Graph zu A.
Es sei A nicht reduzierbar.
Wir nehmen an, dass der Graph zu A nicht stark zusammenhängend ist. Dann gibt Knoten $i \neq j$ mit $i \nrightarrow j$. Es sei

$$I = \{k \in V \mid i \rightarrow k\}$$

und $J = V \setminus I$. Da $j \in J$, ist J nicht leer. Aber auch I ist nicht leer. Ansonsten wäre

$$\forall j \neq i \quad a_{i,j} = 0.$$

Dies wäre ein Widerspruch zu A nicht reduzierbar. Also sind I, J beide nicht leer und somit eine nicht triviale Partition. Daher ist

$$\forall i' \in I \forall j' \in J \quad a_{i,j} = 0.$$

Dies ist ein Widerspruch zu A nicht reduzierbar.
Es sei G stark zusammenhängend.
Man nehme an, dass A reduzierbar ist. Dann existieren disjunkte nicht leere Mengen J, I, so dass $a_{i,j} = 0$ für jedes $i \in I, j \in J$. Es sei nun $v_{i_0} v_{i_1} ... v_{i_{r-1}} v_{i_r}$ ein gerichteter Weg von $i_0 \in I$ nach $i_r \in J$. Dann muss ein Index s existieren, so dass $i_{s-1} \in I$ und $i_s \in J$. Dies impliziert $a_{i_{s-1}, i_s} \neq 0$. Dies ist ein Widerspruch zu den Eigenschaften von J und I. Daher ist A nicht reduzierbar. □

Beispiel 38. *Der Graph in Beispiel 37 ist nicht stark zusammenhängend. Durch vertauschen der Spalten und Reihen 2 und 3 in der Matrix in Beispiel 37 erhält*

man die Matrix

$$\begin{pmatrix} 1 & 1 & 4 & 0 \\ 3 & 1 & 5 & 0 \\ 0 & 0 & 7 & 1 \\ 0 & 0 & 5 & 1 \end{pmatrix}.$$

3.3 Weitere elementare Grundbegriffe und Sätze aus der Graphentheorie

Definition 34.

Ein einfacher Graph ist ein ungerichteter Graph $G = (V, E)$ ohne Schlaufen. Das heißt es gilt:

- $(a, b) \in E \Leftrightarrow (b, a) \in E$ *und es gilt*
- $(a, b) \in E \Rightarrow a \neq b$.

Im folgenden betrachten wir bis auf weiteres nur einfache Graphen.

Definition 35.

- *Ein Teilgraph eines Graphen $G = (V, E)$ ist ein Graph $G' = (V', E')$ mit $V' \subset V$ und $E' \subset E$.*
- *G' heißt (von E) induziert (oder aufgespannt), wenn gilt $(a, b) \in E'$ für alle $a, b \in V'$ mit $(a, b) \in E$.*

Definition 36.

- *Die Menge der Nachbarknoten eines Knoten $v \in V$ ist die Menge*

$$N(v) := \{v' \in V \mid (v, v') \in E\}.$$

- *Der Grad $d_G(v)$ eines Knoten v ist die Anzahl der Kanten, die v berühren. Im Falle eines einfachen Graphen ist dies*

$$d_G(v) = |N(v)|.$$

Definition 37.

- *Der Minimalgrad von G ist definiert als*

$$\delta(G) := \min\{d_G(v) \mid v \in V\}.$$

- *Der Maximalgrad von G ist definiert als*

$$\Delta(G) := \max\{d_G(v) \mid v \in V\}.$$

- *Der Durchschnittsgrad von G ist definiert als*

$$d(G) := \sum_{v \in V} \frac{d_G(v)}{|V|}.$$

Satz 11. *Jeder einfache Graph G enthält einen Weg der Länge $\delta(G)$ und einen Kreis der Länge mindestens $\delta(G) + 1$ (falls $\delta(G) \geq 2$).*

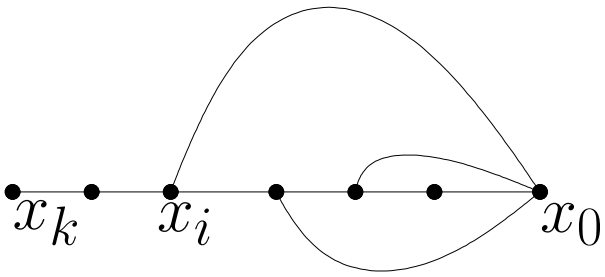


Abbildung 5: Längster Weg in G mit Nachbarkanten von x_0.

Beweis: Es sei $w = x_0...x_k$ ein längster Weg in G. Alle Nachbarn von x_0 müssen dann auf diesem Weg w liegen, da sonst w nicht der längste Weg wäre (siehe Abbildung 5). Daher ist

$$k \geq d_G(x_0) \geq \delta(G).$$

Es sei nun $i \geq k$ maximal mit $(x_i, x_0) \in E$. Dann ist $x_0...x_i x_0$ ein Kreis mit Länge mindestens $\delta(G) + 1$. □

Definition 38. *Ein einfacher Graph heißt* zusammenhängend*, falls für jedes Paar unterschiedlicher Knoten v, w ein Weg existiert. Dies bedeutet, dass ein Weg $v_{i_0} v_{i_1} ... v_{i_{r-1}} v_{i_r}$ mit $v_{i_0} = v$ und$v_{i_r} = w$ existiert.*

Beispiel 39. *Es sei V eine Menge von Computern (Prozessoren). Zwei Computer können Daten durch eine direkte Netzwerkverbindung übertragen.*

Eine direkte Netzwerkverbindung zwischen zwei Prozessoren p_1 und p_2 werde durch eine Kante dargestellt.

Um parallele Rechnungen in einer solchen Topologie von Rechnern und Netzwerkverbindungen durchführen zu können, müssen Daten von jedem Rechner zu jedem anderen übertragen werden können. Dies ist nur möglich, falls der Graph $G = (V, E)$ zusammenhängend ist.

Ziel bei der Konstruktion von Netzwerken:

- *Der Grad der Knoten soll möglichst klein sein.*
- *Die maximale Weglänge $l_{\max}(G)$ von einem Knoten zum anderen soll möglichst klein sein.*

Beispiel 40. *In Abbildung 6 sind folgende Konstruktionen dargestellt*

1. *Kreis mit l Knoten:*
   $\Delta(G_{Kreis}) = 2$ und $l_{\max}(G_{Kreis}) = [\frac{l}{2}]$.
   Hierbei ist $[\frac{l}{2}]$ der ganzteilige Anteil von $\frac{l}{2}$.
2. *Baum mit je zwei Blättern der Tiefe l und $2^{l+1} - 1$ Knoten:*
   $\Delta(G_{Baum}) = 3$ und $l_{\max}(G_{Baum}) = 2l$.
3. *n-dimensionaler Würfel mit 2^n Knoten:*
   $\Delta(G_{Baum}) = n$ und $l_{\max}(G_{Baum}) = n$.

3.4 Ebene Graphen

Definition 39. *Ein ebener Graph besteht aus einer endlichen Menge von Knoten $V \subset \mathbb{R}^2$ (genannt Ecken) und aus einer Menge von Kanten E mit den Eigenschaft*

- *Jede Kante ist ein Polygonzug deren Enden Ecken $v, w \in V$ sind.*
- *Verschiedene Kanten besitzen verschiedene Eckpunkte.*
- *Das Innere einer jeden Kante enthält weder eine Ecke noch einen Punkt einer anderen Kante.*

Jeder ebene Graph beschreibt einen einfachen Graphen $G = (V, \tilde{E})$, wobei

$$\tilde{E} = \{(v, w) \in V \times V \mid \text{Es gibt eine Kante } e \in E \text{ mit Enden } v, w\}.$$

Es sei $\Omega \subset \mathbb{R}^2$. Ω heißt (topologisch) zusammenhängend, falls je zwei Punkte $x, y \in \Omega$ durch eine Kurve verbunden werden können. Dies bedeutet, dass es eine stetige Abbildung

$$\varphi : [0, 1] \to \Omega$$

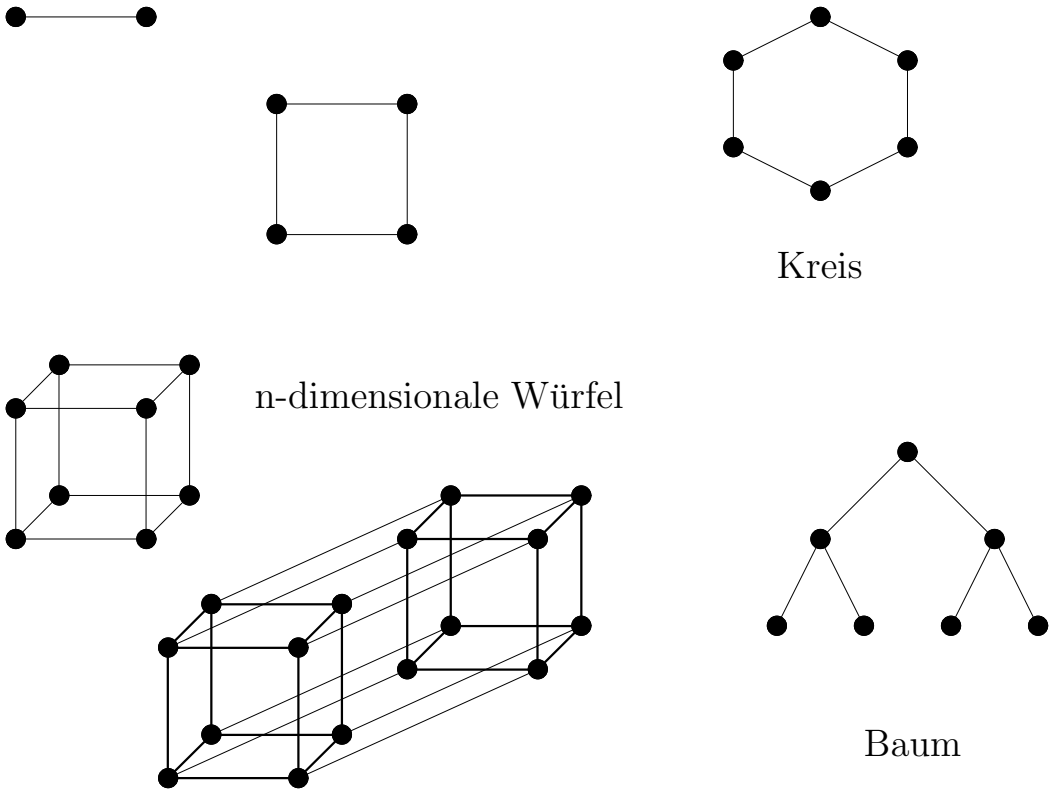


Abbildung 6: Unterschiedliche Graphen für Netzwerke.

gibt, die die Eigenschaft

$$\varphi(0) = x \quad \text{und} \quad \varphi(1) = y$$

hat.

Definition 40. *Es sei $G = (V, E)$ ein ebener Graph. Dann zerlegt $V \cup E$ die Ebene $\mathbb{R}^2$ in (topologisch) zusammenhängende disjunkte Teilgebiete $F_1, F_2, ..., F_f$:*

$$\mathbb{R}^2 \setminus (V \cup E) = \bigcup_{i=1}^{f} F_i.$$

Diese Teilgebiete werden Flächen des Graphen genannt. (Siehe Abbildung 7)

Im Gegensatz zu Definition 25 definieren wir einfache Bäume wie folgt:

Definition 41. *Ein einfacher Baum ist ein zusammenhängender einfacher Graph (V, E) der keine Kreise enthält.*

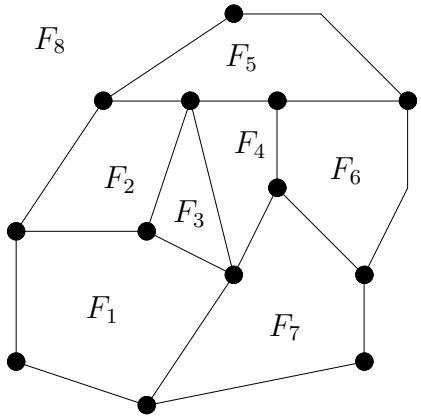


Abbildung 7: Flächen eines Graphen

Lemma 5. *Der Graph eines Baumes besitzt genau eine Fläche.*

Satz 12. *(Eulersche Polyeder Satz) Es sei G ein zusammenhängender ebener Graph mit e Ecken (Knoten), k Kanten und f Flächen. Dann gilt:*

$$e - k + f = 2.$$

Beweis: 1.Fall: G besitzt keinen Kreis. Dann ist G ein Baum.
Beweis von $e - k + f = 2$ durch Induktion nach der Anzahl der Ecken e:

$\underline{e = 1:}$ Dann ist $f = 1$ und $k = 0$.

$\underline{e - 1 \to e:}$
Wegen Satz 11 muss $\delta(G) \leq 1$ sein. Es gibt also eine Ecke c mit $d_G(c) = 1$. Der Graph ohne diese Ecke und der anliegenden Kante genügt der Eigenschaft $e - k + f = 2$ und besitzt genau eine Ecke und Kante weniger. Also auch der ursprüngliche Graph G. (Siehe Abbildung 8 linkes Bild)

2.Fall: G besitze einen Kreis oder nicht.
Beweis von $e - k + f = 2$ durch Induktion nach der Anzahl der Kanten k.

$\underline{k = 0:}$ Dann ist $f = 1$ und $e = 1$, da G zusammenhängend ist.

$\underline{-1k \to k:}$

a) G besitzt keinen Kreis.
Dann ist G ein Baum und es gilt nach *1.Fall* $e - k + f = 2$.

b) G besitzt einen Kreis. Es sei nun β eine Kante die zu einem Kreis gehört. Dann sei G der Graph der durch die Wegnahme von β entsteht. G besitzt eine Kante und eine Fläche weniger als G. Da G der Eigenschaft $e - k + f = 2$ genügt, genügt auch G dieser Eigenschaft. (Siehe Abbildung 8 rechtes Bild)
□

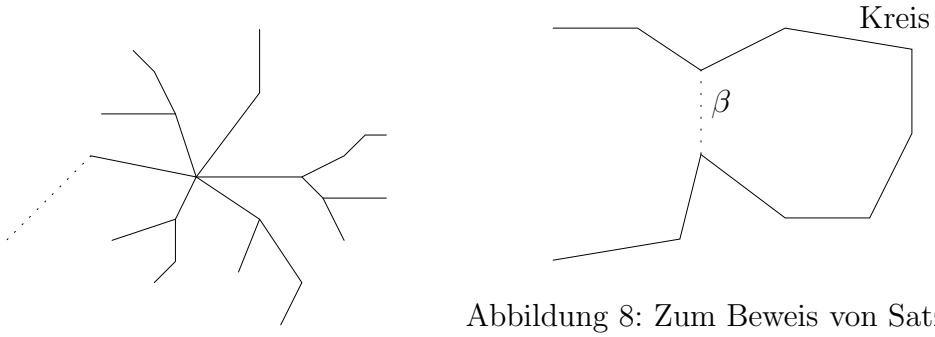


Abbildung 8: Zum Beweis von Satz 12

Korollar 1. *Es sei G ein zusammenhängender ebener Graph mit $e \geq 3$ Ecken. Dann gilt:*

1. *G hat höchstens $3e - 6$ Kanten.*
2. *Falls die beschränkten Flächen von G nur aus Dreiecken bestehen und G e_r Ecken am Rand des unbeschränkten Gebietes besitzt, dann enthält G genau $3e - e_r - 3$ Kanten.*

Anwendung: Zur Diskretisierung von partiellen Differentialgleichungen verwendet man oft lineare finite Element auf Dreiecksgittern. Besitzt der dazugehörige Graph e Ecken und e_r Randecken, dann benötigt man einen Datenspeicher der Länge $3e - e_r - 3$ um Daten auf den Kanten zu speichern.

Beweis: zu 1) Jede Fläche besitzt mindestens 3 Kanten und an jeder Kante liegen 2 Flächen. Daher gilt

$$3f \leq 2k.$$

Wegen $e - k + f = 2$ folgt daher $k \leq 3e - 6$.
zu) Hausaufgabe. □

3.5 Eckenfärbung von Graphen

Eine Landkarte kann als ebener Graph $G_K = (V_K, E_K)$ betrachtet werden, dessen Flächen die Länder der Karte darstellen. Die Kanten dieser Landkarte sind die Grenzen der Länder.

Konstruktion eines Färbungsgraphen: Es sei V die Menge der Flächen von G_K. Zwischen den Knoten $v, w \in V$ sei eine Kante $(v, w) \in E$ genau dann wenn v und w eine gemeinsame Grenze haben.

Der Graph $G = (V, E)$ ist der Färbungsgraph der Landkarte.

Problem 2. *Es sei $G = (V, E)$ ein einfacher Graph und es sei $F_n = \{1, 2, ..., n\}$. F_n ist die Menge der Farben.*

Gesucht ist eine Abbildung

$$f : V \to F_n$$

so dass

$$f(v) \neq f(w) \quad \textit{falls } (v, w) \in E.$$

Falls eine solche Abbildung existiert heißt der Graph n-färbbar.

Die Lösung dieses Problems ist im Allgemeinen sehr schwierig. Wir beschränken uns hier auf einfache Fälle.

Satz 13. *Jeder ebene Graph ist 4-färbbar.*

Satz 14. *Jeder ebene Graph ist 5-färbbar.*

Satz 15. *Jeder ebene Graph ist 6-färbbar.*

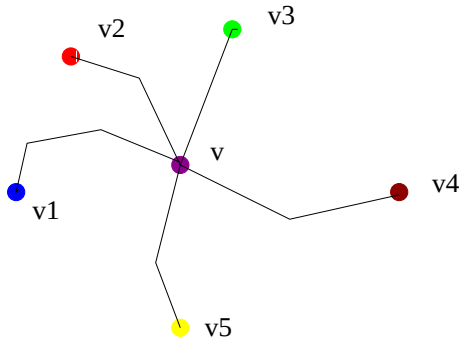


Abbildung 9: Beweis von Satz 15.

Der Beweis des Satzes 13 ist sehr schwierig. Deswegen wird dieser Satz hier nicht bewiesen.

Beweis von Satz 15:

Beweis durch Induktion nach e.

$\underline{e \leq 6}$: Jede Ecke wird mit einer anderen Farbe markiert.

$\underline{e - 1 \to e}$: Aus Definition 3.3 und Korollar 1 folgt

$$d(G) = \sum_{v \in V} \frac{d_G(v)}{|V|} = \frac{2k}{e} \leq \frac{2(3e - 6)}{e} < 6.$$

Daher gibt es mindestens einen Knoten v

$$d_G(v) \leq 5.$$

Es sei nun G' der induzierte Teilgraph von $V' := V \backslash \{v\}$. Da G' weniger Ecken als G enthält, gibt es eine Abbildung

$$f : V' \to F_6.$$

$f(N(v)) \subset F_6$ sind die so gefärbten Farben der Nachbarknoten von v.

Da $|N(v)| \leq 5$ gibt es eine Farbe $F \in F_6 \backslash f(N(v))$. Die zusätzliche Färbung

$$f : v \mapsto F$$

liefert die gewünschte Färbung von G (siehe Abbildung 9). □

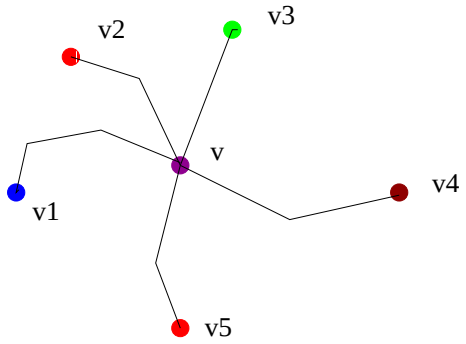


Abbildung 10: Beweis von Satz 14: einfacher Fall.

Beweis von Satz 14: Beweis durch Induktion nach e.
$e \leq 5$: Jede Ecke wird mit einer anderen Farbe markiert.
$e - 1 \to e$: Definiere v analog wie im Beweis von Satz 15. Das heißt $d_G(v) \leq 5$. Man färbe den induzierten Graph mit Ecken $V \backslash \{v\}$ mit 5 Farben.

Einfacher Fall:
Falls die Nachbarn $N(v)$ von v maximal 4 unterschiedliche Farben haben, dann färbe man v mit einer noch nicht verwendeten Farbe aus $F_5 \backslash F^{-1}(N(v))$ (siehe Abbildung 10) .

Schwieriger Fall:
Schwieriger ist der Fall dass alle Nachbarn $N(v)$ von v eine andere Farbe besitzen. Dann ist $d_G(v) = 5$. Man nummeriere die Nachbarknoten von v im Uhrzeigersinn von v_1, v_2, v_3, v_4, v_5. Durch Vertauschung der Farbennummern erreichen wir $f(v_i) = i$.

Es sei nun H_{ij} der von den mit i, j gefärbten Knoten induzierte Untergraph. Das heißt H_{ij} habe die Knotenmenge $f^{-1}(\{i, j\})$. Es sei C_1 die Zusammenhangskomponente von H_{13} die v_1 enthält.

1.Fall: C_1 enthält v_3 nicht. Dann kann man die Farben von $1, 3$ in C_1 vertauschen und für v die Farbe 1 wählen: $f(v) = 1$ (siehe Abbildung 11) .

2.Fall: C_1 enthält v_3. Dann gibt es einen Weg von $v_1 w_1 w_2 ... w_s v_3$. $v v_1 w_1 w_2 ... w_s v_3 v$ ist dann ein Kreis. Aus Abbildung ... sieht man, dass v_2 und v_4 durch den Kreis $v v_1 w_1 w_2 ... w_s v_3 v$ getrennt werden. Damit liegen v_2, v_4 in unterschiedlichen Zusammenhangskomponenten von H_{24}. Vertausche daher die Farben $2, 4$ in der Zusammenhangskomponente von H_{24}, die v_2 enthält. Für v kann man nun die Farbe 2 wählen: $f(v) = 2$ (siehe Abbildung 12).

□

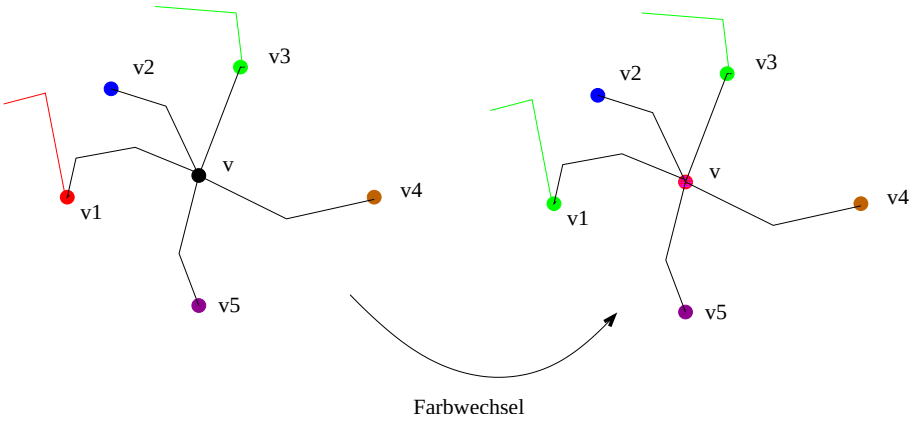


Abbildung 11: Beweis von Satz 14: 1.Fall.

Satz 16. *Jeder Graph ist* $\Delta(G) + 1$*-färbbar.*

Beweis: Greedy-Algorithmus:
Nummeriere alle Knoten $v_1, .., v_n$. Dann setze für $i = 1, ...n$:

$$f(v_i) := \min\left(\{1, 2, 3, ..., \Delta(G) + 1\} \backslash f\left(\{v_j \in N(v_i) \mid j < i\}\right)\right).$$

Eine solche Definition von $f(v_i)$ ist möglich, da

$$|f(\{v_j \in N(v_i) \mid j < i\})| < \Delta(G) + 1$$

□

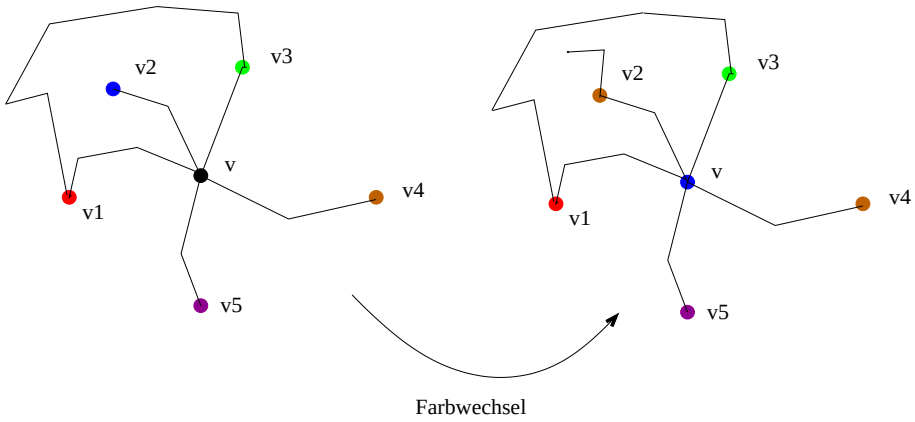


Abbildung 12: Beweis von Satz 14: 2.Fall.

3.6 Kantenfärbung von bipartiten Multigraphen

Definition 42. *Es seien V, E disjunkte Mengen. Die Menge der 2-elementigen Teilmengen von V sei $[V]^2$. Des weiteren sei eine Abbildung*

$$c : E \to [V]^2$$

gegeben. $G = (V, E, c)$ ist dann ein Multigraph ohne Schlingen.

Bemerkung:
G ist ein einfacher Graph, falls c eine injektive Abbildung ist. Die Kantenrelation E' ist dann definiert durch:

$$E' = \{(v, w) \in V \times V \mid \exists e \in E \;\; c(e) = \{v, w\}\}.$$

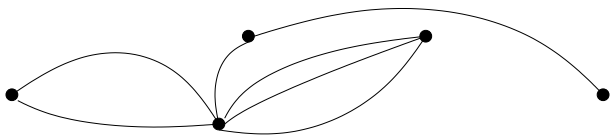


Abbildung 13: Beispiel eines Multigraphen.

Definition 43. *Ein Multigraph $G = (V, E, c)$ heißt bipartiter Graph, falls es eine Partition*

$$V = V_l \dot\cup V_r$$

gibt, so dass es keine Kante e mit $c(e) = \{a, b\}$ und $a, b \in V_l$ oder $a, b \in V_r$ gibt.

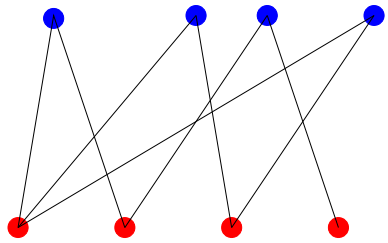


Abbildung 14: Beispiel eines bipartiten Graphen.

Satz 17. *Ein bipartiter Graph enthält keinen Kreis ungerader Länge.*

Problem 3. *Es sei $G = (V, E, c)$ ein Multigraph und es sei $F_n = \{1, 2, ..., n\}$. F_n ist die Menge der Farben.*

Gesucht ist eine Abbildung

$$f : E \to F_n$$

so dass

$$f(e) \neq f(g) \quad \textit{falls } c(e) \cap c(g) \neq \emptyset.$$

Falls eine solche Abbildung existiert heißen die Kanten des Graphen n-färbbar.

Beispiel 41. *Es sei V_l eine Menge von Lehrern und V_r eine Menge von Schulklassen. Die Menge der Knoten sei $V = V_l \dot{\cup} V_r$. Für jede Unterrichtsstunde die ein Lehrer $w \in V_l$ eine Schulklasse $v \in V_r$ unterrichten muss konstruiere man eine Kante e. E sei die Menge dieser Kanten e und $c(e) = \{w, v\}$ die Abbildung die der Unterrichtsstunde e den Lehrer w und die Klasse v zuordnet.*

Frage: *Wie viele Schulstunden muss der Stundenplan enthalten?*

Für jede Schulstunde im Stundenplan verwende man eine Farbe. Jede Kante e wird dann mit der Farbe der Schulstunde gefärbt in der die zugehörige Unterrichtsstunde stattfindet.

Satz 18. *Für jeden bipartiten Multigraphen G sind die Kanten $\Delta(G)$-färbbar.*

Beweis: Induktion nach $n = |E|$.

$\underline{n = |E| = 0.}$ Dann braucht man keine Farbe.

$\underline{n \to n+1.}$ Es sei e eine Kante mit $c(e) = \{x, y\}$.
Mit der Abkürzung $E' = E \backslash \{e\}$ sind die Kanten des Graphen $G' = (V, E', c)$ $\Delta(G')$-färbbar, wobei $\Delta(G') \leq \Delta(G)$. Es sei

$$f : E' = E \backslash \{e\} \to F_{\Delta(G')}$$

eine solche Färbung. Die Knoten x, y besitzen in G' weniger als $\Delta(G)$ anliegende Kanten. Daher gibt es Farben $\alpha, \beta \in F_{\Delta(G)}$, so dass
keine an x anliegende Kante die Färbung α besitzt und
keine an y anliegende Kante die Färbung β besitzt.

1.Fall: $\alpha = \beta$. Dann sei $f(e) := \alpha$.
2.Fall: $\alpha \neq \beta$. Betrachte einen maximal langen Kantenzug

$$z := c_1 e_1 c_2 e_2 ... c_s e_s c_{s+1},$$

so dass

$$\begin{aligned} \forall i \quad f(e_{2i-1}) &= \beta \\ \forall i \quad f(e_{2i}) &= \alpha \\ c_1 &= x. \end{aligned}$$

Die Eigenschaft Kantenzug bedeutet hierbei

- Alle c_i sind Knoten: $\forall i \;\; c_i \in V$.
- Alle e_i sind Kanten: $\forall i \;\; e_i \in E$.
- $\forall i \;\; c(e_i) = \{c_i, c_{i+1}\}$
- Alle Kanten e_i sind paarweise verschieden.

Es ist $c_{s+1} \neq y$. Ansonsten wäre s gerade und

$$c_1 c_2 ... c_s c_{s+1} c_1,$$

ein ungerader Kreis in G. Des weiteren ist natürlich $c_i \neq y$ für alle $1 \leq i \leq s$. Wegen der Maximalität des Kantenzuges kann man die Farben im Kantenzug z vertauschen. Das bedeutet

$$\begin{aligned} \forall i \in \mathbb{N} \quad \tilde{f}(e_{2i-1}) &:= \alpha \\ \forall i \in \mathbb{N} \quad \tilde{f}(e_{2i}) &:= \beta. \end{aligned}$$

Für alle anderen Kanten aus $e' \in E'$ setzen wir $\tilde{f}(e') = f(e')$. Wegen der Konstruktion von $\tilde{f}$ können wir

$$\tilde{f}(e) = \beta$$

setzen um mit $\tilde{f}$ eine $\Delta(G)$-Färbung der Kanten von G zu erhalten. $\square$

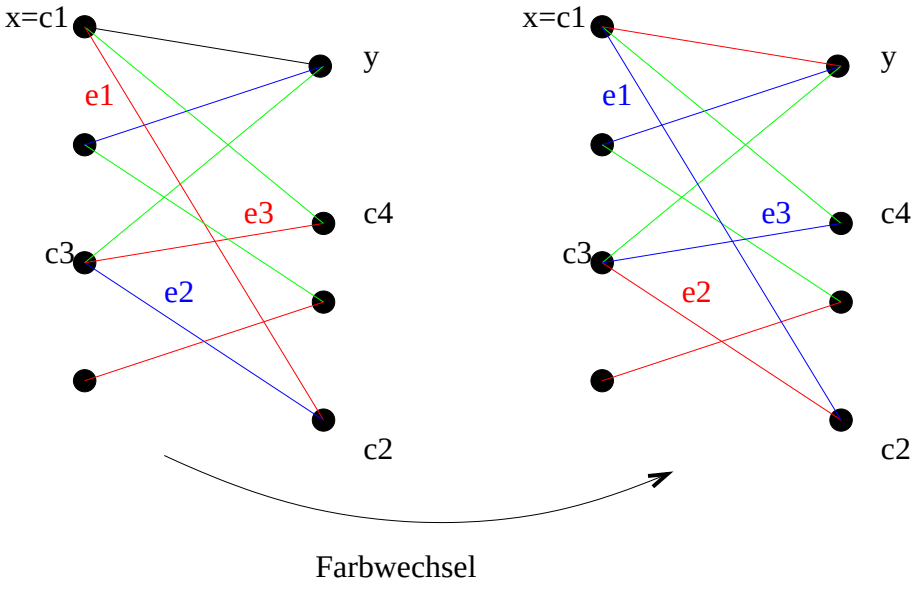


Abbildung 15: Zum Beweis von Satz 18.

4 Endliche Automaten

4.1 Der Mealy-Automat

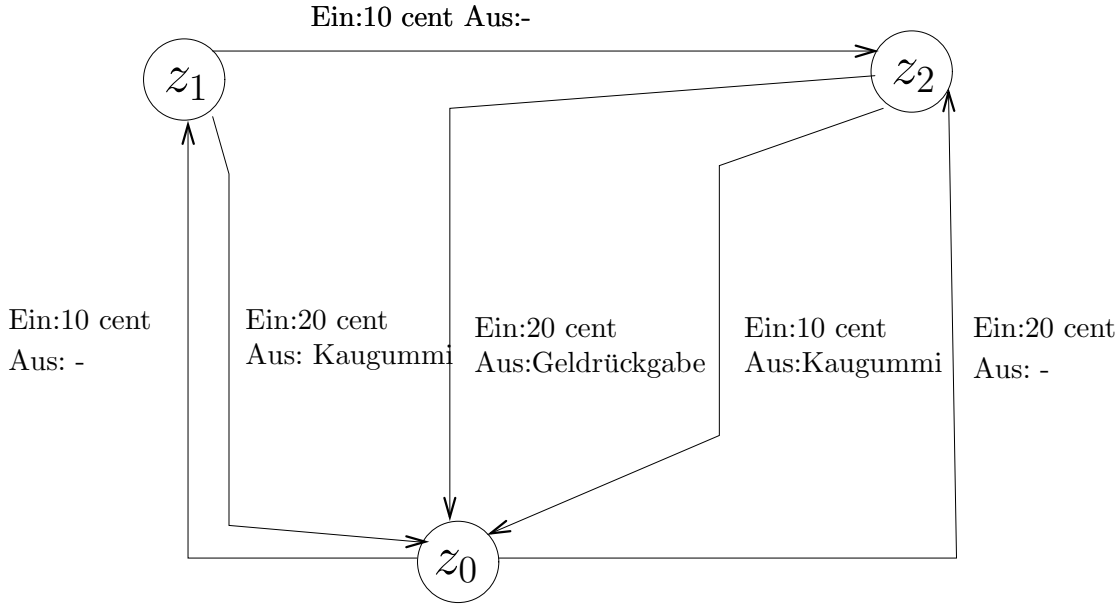
Beispiel 42.

- *Ein Automat verkaufe für* 30 Cent *einen Kaugummi.*
- *Nach Eingabe von genügend vielen* 10 Cent *und* 20 Cent *Stücken gibt der Automat einen Kaugummi heraus.*
- *Bei falscher Eingabe gibt es eine Rückgabe des Geldes.*

Aufgabe: Beschreibe den Kaugummi-Automaten durch einen gerichteten Graphen:

- *Die Menge der Zustände Z sei die Menge der Knoten.*
- $\textcircled{z_0}$ *Anfangszustand.*
- $\textcircled{z_i}$ *ein weiterer Zustand für $i > 0$.*
- $z_i \overset{Ein,Aus}{\longrightarrow} z_j$ *Übergang von einem Zustand z_i zu einem Zustand z_j nach einer bestimmten Eingabe Ein*
  und mit einer bestimmten Ausgabe Aus.

Lösung:



Formal kann man den Kaugummi-Automaten in Beispiel 42 wie folgt beschreiben:

- Menge von Zuständen
$$Z = \{z_0, z_1, z_2\}.$$
- Eingabemenge
$$\Sigma = \{1, 2\} \hat{=} \{10 \text{ cent}, 20 \text{ cent}\}.$$
- Ausgabemenge
$$\Delta = \{K, R, -\} \hat{=} \{\text{Kaugummi}, \text{Geldrückgabe}, \text{keine Ausgabe}\}.$$
- Überführungsfunktion $\delta : Z \times \Sigma \to Z$ mit
$$\begin{aligned} (z_0, 1) &\mapsto z_1 \\ (z_0, 2) &\mapsto z_2 \\ (z_1, 1) &\mapsto z_2 \\ (z_1, 2) &\mapsto z_0 \\ (z_2, 1) &\mapsto z_0 \\ (z_2, 2) &\mapsto z_0 \end{aligned}$$

- Ausgabefunktion $\lambda : Z \times \Sigma \to \Delta$ mit

$$\begin{array}{rcl} (z_0, 1) & \mapsto & - \\ (z_0, 2) & \mapsto & - \\ (z_1, 1) & \mapsto & - \\ (z_1, 2) & \mapsto & K \\ (z_2, 1) & \mapsto & K \\ (z_2, 2) & \mapsto & R \end{array}$$

Definition 44. *Ein Mealy-Automat ist ein Sechstupel $M = (Z, \Sigma, \Delta, \delta, \lambda, z_0)$, wobei*

- *Z eine endliche Menge von Zuständen ist,*
- *Σ eine endliche Eingabemenge ist,*
- *Δ eine endliche Ausgabemenge,*
- *$\delta : Z \times \Sigma \to Z$ eine Überführungsfunktion*
- *$\lambda : Z \times \Sigma \to \Delta$ eine Ausgabefunktion und*
- *$z_0 \in Z$ der Startzustand ist.*

Die Mengen Z, Σ, Δ sollten disjunkt sein.

Bei der Anwendung eines Mealy-Automaten werden mehrere Buchstaben, also ein ganzes Wort eingegeben. In welchem Zustand sich der Mealy-Automaten nach Eingabe des Wortes w befindet, gibt der Zustand $\hat{\delta}(z_0, w)$ an. Hierbei ist $\hat{\delta} : Z \times \Sigma^* \to Z$ die allgemeine Überführungsfunktion.

Die Funktion $\hat{\delta} : Z \times \Sigma^* \to Z$ sei wie folgt rekursiv definiert:

$$\begin{array}{rcl} \hat{\delta}(z, \epsilon) & = & z \quad \text{für alle } z \in Z, \\ \hat{\delta}(z, wa) & = & \delta(\hat{\delta}(z, w), a) \quad \text{für alle } a \in \Sigma, w \in \Sigma^*, z \in Z. \end{array}$$

Man schreibt kurz δ anstatt $\hat{\delta}$.

(Überlege warum das möglich ist!).

Im Falle des Kaugummi-Automaten in Beispiel 42 liefert die Eingabe $w = 2121$ den Zustand $\delta(z_0, w) = z_0$ und

die Eingabe $w = 222111$ den Zustand $\delta(z_0, w) = z_2$.

Die allgemeine Ausgabefunktion $\hat{\lambda} : \Sigma^+ \to \Delta^+$ sei wie folgt rekursiv definiert:

$$\begin{aligned} \hat{\lambda}(a) &= \lambda(z_0, a) \quad \text{für alle } a \in \Sigma, \\ \hat{\lambda}(wa) &= \hat{\lambda}(w)\lambda(\delta(z_0, w), a) \quad \text{für alle } a \in \Sigma, w \in \Sigma^*. \end{aligned}$$

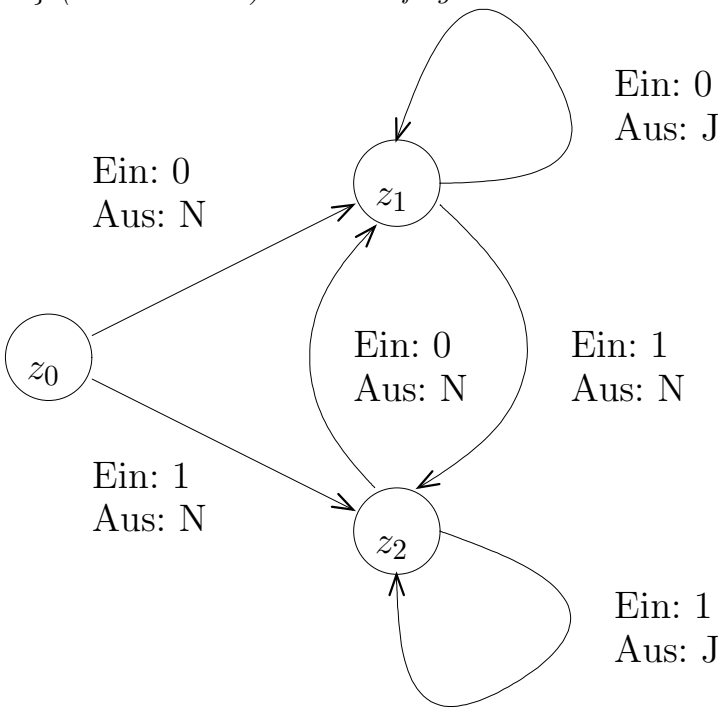
Bem: Der Mealy-Automat liefert bei der Eingabe eines Wortes $w \in \Sigma^+$ der Länge $|w|$ als Ausgabe das Wort

$$v = \hat{\lambda}(w) \in \Delta^+$$

der gleichen Länge $|w|$.

Im Falle des Kaugummi-Automaten in Beispiel 42 liefert
die Eingabe $w = 2121$ die Ausgabe $\hat{\lambda}(w) = -K - K$ und
die Eingabe $w = 2221111$ die Ausgabe $\hat{\lambda}(w) = -R - K - -K$.

Beispiel 43. *Es sei $\Sigma = \{0, 1\}$. Ein Mealy-Automat soll erkennen, ob bei der Eingabe des Wortes $w = b_1 b_2 .. b_n \in \Sigma^*$ am Ende zwei Buchstaben doppelt vorkommen (also $b_{n-1} = b_n$). Der zugehörige Mealy-Automat mit Ausgabemenge $\{J, N\}$ (Ja oder Nein) sieht wie folgt aus:*



4.2 Der Moore-Automat

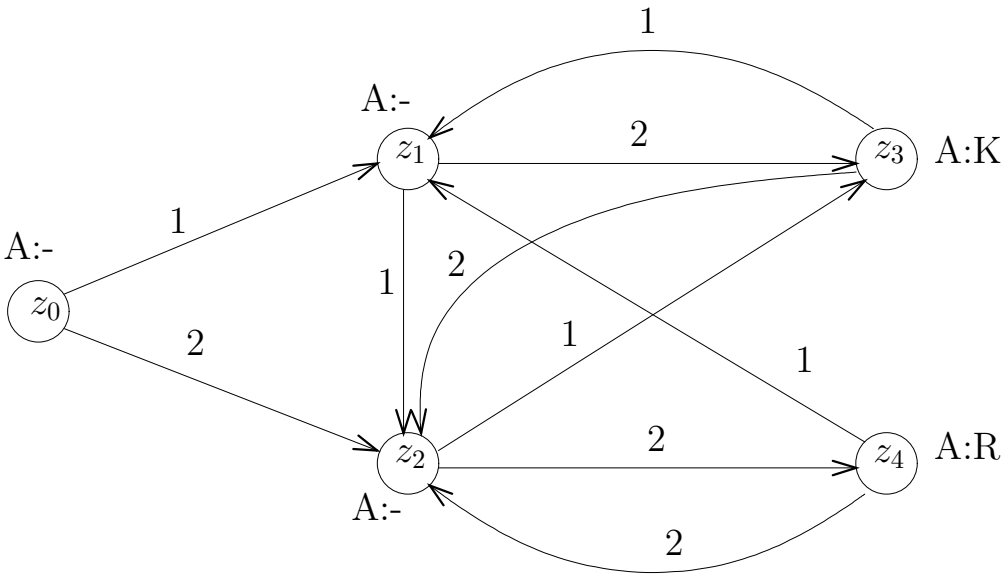
Beispiel 44.

- *Ein Automat verkaufe für* 30 Cent *einen Kaugummi.*
- *Nach Eingabe von genügend vielen* 10 Cent *und* 20 Cent *Stücken gibt der Automat einen Kaugummi heraus.*
- *Bei falscher Eingabe gibt es eine Rückgabe des Geldes.*

Aufgabe: Beschreibe den Kaugummi-Automaten durch einen gerichteten Graphen:

- *Die Menge der Zustände Z sei die Menge der Knoten.*
- $\textcircled{z_0}$ *Anfangszustand.*
- $\textcircled{z_i}$ *ein weiterer Zustand für $i > 0$ mit einer Ausgabe.*
- $z_i \xrightarrow{Ein} z_j$ *Übergang von einem Zustand z_i zu einem Zustand z_j nach einer bestimmten Eingabe Ein.*

Lösung:



- Menge von Zuständen

$$Z = \{z_0, z_1, z_2, z_3, z_4\}.$$

- Eingabemenge
$$\Sigma = \{1, 2\} \hat{=} \{10 \text{ cent}, 20 \text{ cent}\}.$$
- Ausgabemenge
$$\Delta = \{K, R, -\} \hat{=} \{\text{Kaugummi}, \text{Geldrückgabe}, \text{keine Ausgabe}\}.$$
- Überführungsfunktion $\delta : Z \times \Sigma \to Z$ mit
$$\begin{array}{rcl} (z_0, 1) & \mapsto & z_1 \\ (z_0, 2) & \mapsto & z_2 \\ (z_1, 1) & \mapsto & z_2 \\ (z_1, 2) & \mapsto & z_3 \\ (z_2, 1) & \mapsto & z_3 \\ (z_2, 2) & \mapsto & z_4 \\ (z_3, 1) & \mapsto & z_1 \\ (z_3, 2) & \mapsto & z_2 \\ (z_4, 1) & \mapsto & z_1 \\ (z_4, 2) & \mapsto & z_2 \end{array}$$
- Ausgabefunktion $\lambda : Z \to \Delta$ mit
$$\begin{array}{rcl} z_0 & \mapsto & - \\ z_1 & \mapsto & - \\ z_2 & \mapsto & - \\ z_3 & \mapsto & K \\ z_4 & \mapsto & R \end{array}$$

Definition 45. *Ein Moore-Automat ist ein Sechstupel $M = (Z, \Sigma, \Delta, \delta, \lambda, z_0)$, wobei*

- *Z eine endliche Menge von Zuständen ist,*
- *Σ eine endliche Eingabemenge ist,*

- Δ *eine endliche Ausgabemenge,*
- $\delta : Z \times \Sigma \to Z$ *eine Überführungsfunktion*
- $\lambda : Z \to \Delta$ *eine Ausgabefunktion und*
- $z_0 \in Z$ *der Startzustand.*

Die Mengen Z, Σ, Δ *sollten disjunkt sein.*

Wie beim Mealy-Automaten kann man beim Moore-Automaten, eine allgemeine Überführungsfunktion und eine allgemeine Ausgabefunktion definieren:

Die allgemeine Überführungsfunktion sei:

$$\begin{aligned} \hat{\delta} : Z \times \Sigma^* &\to Z \\ \hat{\delta}(z, \epsilon) &= z \quad \text{für alle } z \in Z, \\ \hat{\delta}(z, wa) &= \delta(\hat{\delta}(z, w), a) \quad \text{für alle } a \in \Sigma, w \in \Sigma^*, z \in Z. \end{aligned}$$

Man schreibt kurz δ anstatt $\hat{\delta}$.

Die allgemeine Ausgabefunktion sei

$$\begin{aligned} \hat{\lambda} : \Sigma^+ &\to Z \\ \hat{\lambda}(a) &= \lambda(\delta(z_0, a)) \quad \text{für alle } a \in \Sigma, \\ \hat{\lambda}(wa) &= \hat{\lambda}(w)\lambda(\delta(z_0, wa)) \quad \text{für alle } a \in \Sigma, w \in \Sigma^*. \end{aligned}$$

Bem: Der Moore-Automat liefert bei der Eingabe eines Wortes $w \in \Sigma^+$ der Länge $|w|$ als Ausgabe das Wort

$$v = \hat{\lambda}(w) \in \Delta^+$$

der gleichen Länge $|w|$.

Beispiel 45. *Es sei* $\Sigma = \{0,1\} \times \{0,1\}$ *und* $\Delta = \{0,1\}$.

Ziel ist es zwei binär geschriebene Zahlen $a, b \in \Delta^+$ *zu addieren. Dazu schreiben wir*

$$\begin{aligned} a &= a_0 a_1 ... a_n \\ b &= b_0 b_1 ... b_n \\ w &= (a_0, b_0)(a_1, b_1)...(a_n, b_n) =: \mathcal{J}(a, b). \end{aligned}$$

Hierbei entsprechen $a_0a_1...a_n$ und $b_0b_1...b_n$ den Zahlen

$$x = \sum_{k=0}^{n} a_k 2^k \quad \textit{und} \quad y = \sum_{k=0}^{n} b_k 2^k.$$

Wir suchen nun einen Moore-Automaten der unter der Eingabe $w \in \Sigma^+$ als Ausgabe die binäre Summe $a \oplus b \in \Delta^+$ liefert. Abbildung 16 zeigt einen solchen Automaten. Hierbei ist folgendes zu beachten. Die binäre Darstellung $a \oplus b$ ist eventuell ein Wort der Länge $n+2$ und nicht nur $n+1$ wie das Wort w. Dieses Problem wird dadurch vermeiden, dass man den Moore-Automaten nur auf Worte aus

$$\{w = (a_0, b_0)(a_1, b_1)...(a_n, b_n) \in \Sigma^+ \mid a_n = b_n = 0\}$$

anwendet. Jedes gegeben Wort läßt sich natürlich so verlängern, dass $a_n = b_n = 0$.

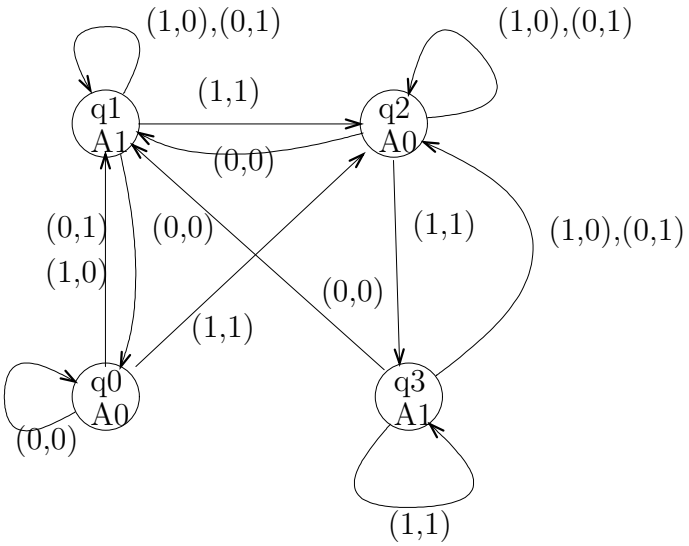


Abbildung 16: Addition durch einen Moore Automaten

Die Kaugummi-Automaten in Beispiel 42 und Beispiel 44 liefern bei der gleichen Eingabe w die gleiche Ausgabe $\hat{\lambda}(w)$, obwohl der eine Automat ein Mealy-Automat mit 3 Zuständen ist und der andere Automat ein Moore Automat mit 5 Zuständen.

Der folgende Satz zeigt, dass so eine Äquivalenz für jeden Moore- und Mealy-Automaten konstruiert werden kann.

Satz 19. *Zu jedem Moore-Automaten gibt es einen Mealy-Automaten, der bei gleicher Eingabe die gleiche Ausgabe liefert und umgekehrt gibt es zu jedem Mealy-Automaten einen Moore-Automaten der bei gleicher Eingabe die gleiche Ausgabe liefert.*

Beweis: :
1.Teil: Moore-Automat $\rightarrow$ Mealy-Automat.
Es sei $M = (Z, \Sigma, \Delta, \delta, \lambda, z_0)$ ein Moore-Automat. Dann sei $M' = (Z, \Sigma, \Delta, \delta, \lambda', z_0)$ der Mealy-Automat mit

$$\lambda'(z, b) := \lambda(\delta(z, b)) \quad \text{für alle } z \in Z, b \in \Sigma.$$

Für die zugehörigen allgemeinen Ausgabefunktionen gilt

$$\hat{\lambda}'(w) := \hat{\lambda}(w) \quad \text{für alle } w \in \Sigma^+ \ .$$

Dies folgt durch Beweis durch Induktion nach $|w|$ und mit Hilfe der Formel

$$\begin{aligned} \hat{\lambda}'(wa) &= \hat{\lambda}'(w)\lambda'(\delta(z_0, w), a) \\ &= \hat{\lambda}(w)\lambda(\delta(\delta(z_0, w), a)) \\ &= \hat{\lambda}(w)\lambda(\delta(z_0, wa)) \\ &= \hat{\lambda}(wa), \end{aligned}$$

wobei $w \in \Sigma^+, a \in \Sigma$.
2.Teil: Mealy-Automat $\rightarrow$ Moore-Automat.
Es sei $M = (Z, \Sigma, \Delta, \delta, \lambda, z_0)$ ein Mealy-Automat. Dann sei $M' = (Z', \Sigma, \Delta, \delta', \lambda', z_0')$ der Moore-Automat mit

$$\begin{aligned} Z' &:= Z \times \Delta, \\ z_0' &:= [z_0, a_0] \quad \text{wobei } a_0 \in \Delta \text{ beliebig fest gewählt}, \\ \delta'([z, a], b) &:= [\delta(z, b), \lambda(z, b)] \quad \text{für alle } [z, a] \in Z', b \in \Sigma \\ \lambda'([z, a]) &:= a \quad \text{für alle } [z, a] \in Z'. \end{aligned}$$

Für die zugehörigen allgemeinen Ausgabefunktionen gilt

$$\hat{\lambda}'(w) := \hat{\lambda}(w) \quad \text{für alle } w \in \Sigma^+ \ . \tag{1}$$

Diese Aussage zeigt man wie folgt:

Zuerst zeigt man durch Induktion, dass es für alle $[z, a] \in Z', w \in \Sigma^+$ ein $a' \in \Sigma$ gibt, so dass gilt:

$$\delta'([z, a], w) := [\delta(z, w), a'].$$

Die Gleichung (1) folgt dann durch Beweis durch Induktion nach $|w|$ und mit Hilfe der Formel

$$\begin{aligned}
\hat{\lambda}'(wa) &= \hat{\lambda}'(w)\lambda'(\delta'(z_0', wa)) \\
&= \hat{\lambda}'(w)\lambda'(\delta'([z_0, a_0], wa)) \\
&= \hat{\lambda}(w)\lambda'(\delta'(\delta'([z_0, a_0], w), a)) \\
&= \hat{\lambda}(w)\lambda'(\delta'([\delta(z_0, w), a'], a)) \\
&= \hat{\lambda}(w)\lambda'([\delta(\delta(z_0, w), a), \lambda(\delta(z_0, w), a)]) \\
&= \hat{\lambda}(w)\lambda(\delta(z_0, w), a) \\
&= \hat{\lambda}(wa),
\end{aligned}$$

wobei $w \in \Sigma^+, a \in \Sigma$. □

4.3 Der endliche Automat

Ein endlicher Automat ist ein Moore-Automat mit Ausgabemenge $\Delta = \{J, N\}$.

Formal definiert man eine endlichen Automaten wie folgt:

Definition 46 (DEA). *Ein endlicher Automat ist ein Fünftupel* $M = (Z, \Sigma, \delta, z_0, F)$*, wobei*

- Z *eine endliche Menge von Zuständen ist,*
- Σ *eine endliche Eingabemenge ist,*
- $\delta : Z \times \Sigma \to Z$ *eine Überführungsfunktion,*
- $z_0 \in \Sigma$ *der Startzustand und*
- $F \subset Z$ *eine Menge von Endzuständen ist.*

Die Mengen Z, Σ *sollten disjunkt sein.*

Die allgemeine Überführungsfunktion sei:

$$\begin{aligned}\hat{\delta} : Z \times \Sigma^* &\rightarrow Z \\ \hat{\delta}(z, \epsilon) &= z \quad \text{für alle } z \in Z, \\ \hat{\delta}(z, wa) &= \delta(\hat{\delta}(z, w), a) \quad \text{für alle } a \in \Sigma, w \in \Sigma^*, z \in Z.\end{aligned}$$

Man schreibt kurz δ anstatt $\hat{\delta}$.

Definition 47. *Es sei $M = (Z, \Sigma, \delta, z_0, F)$ ein endlicher Automat. Die von diesem Automaten akzeptierte (oder erkannte) Sprache wird definiert als*

$$L_M := \{w \in \Sigma^* \mid \delta(z_0, w) \in F\}.$$

Definition 48. *Eine Sprache $L \subset \Sigma^*$ heißt regulär, falls es einen endlichen Automaten gibt, der L akzeptiert.*

Beispiel 46. *Es sei $\Sigma = \{0, 1\}$ und*

$$L = \{w_1 w_2 ... w_n \in \Sigma^* \mid w_{n-1} = w_n\}.$$

Der endliche Automat, der diese Sprache erkennt, ist in Abbildung 17 dargestellt.

Beispiel 47. *Es sei $\Sigma = \{a, b, a^{-1}, b^{-1}\}$. Dann sei*

$$\begin{aligned}H = \{z \in \Sigma^* \mid\ & \neg\exists p, q \in \Sigma^* \ \ z = paa^{-1}q \wedge \\ & \neg\exists p, q \in \Sigma^* \ \ z = pa^{-1}aq \wedge \\ & \neg\exists p, q \in \Sigma^* \ \ z = pbb^{-1}q \wedge \\ & \neg\exists p, q \in \Sigma^* \ \ z = pb^{-1}bq \ \}.\end{aligned}$$

In Beispiel 22 wurde auf H eine Gruppenstruktur konstruiert. Der Automat in Abbildung 18 zeigt eine endlichen Automaten, der H akzeptiert.

Satz 20. *Es sei $M_M = (Z, \Sigma, \Delta, \delta, \lambda, z_0)$ ein Moore-Automat. Es seien $w = w_0 w_1 ... w_n \in \Sigma^+$ und $q = q_0 q_1 ... q_n \in \Delta^+$ Wörter gleicher Länge. Dann fassen wir das Wort (w, q) wie folgt als Wort in $(\Sigma \times \Delta)^+$ auf*

$$(w_0, q_0)(w_1, q_1)...(w_n, q_n) =: \mathcal{J}(w, q).$$

Die Diagonalsprache des Moore-Automaten M ist definiert als

$$\begin{aligned}L_D &:= \{\mathcal{J}(w, \hat{\lambda}(w)) \mid w \in \Sigma^+\} \\ &\subset (\Sigma \times \Delta)^+ \subset (\Sigma \times \Delta)^*.\end{aligned}$$

Dann gibt es einen endlichen Automaten, der diese Sprache erkennt.

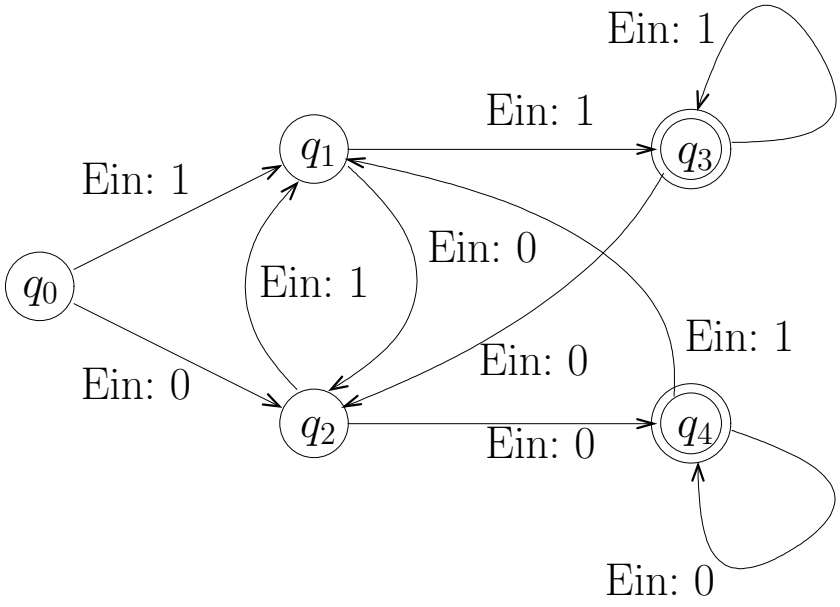


Abbildung 17: Ein endlicher Automat

Beweis: Es sei $M_e = (Z', \Sigma', \delta', z_0, Z)$ wobei z_{no} ein weiterer Zustand und

$$\begin{aligned} Z' &= Z \cup \{z_{no}\}, \\ \Sigma' &= \Sigma \times \Delta, \\ \delta'(z,(b,q)) &:= \begin{cases} \delta(z,b) & \text{falls } z \neq z_{no} \text{ und } q = \lambda(\delta(z,b)) \\ z_{no} & \text{sonst} \end{cases} \end{aligned}$$

Auf Grund der Eigenschaften eines Moore-Automaten ist

$$\begin{aligned} (w, \hat{\lambda}(w)) &= (w_1 w_2 ... w_n, q_1 q_2 ... q_n) \\ &\Updownarrow \\ \lambda(\delta(z_0, w_1 ... w_i)) &= q_i \quad \text{für } i = 1, ..., n. \end{aligned}$$

Daher erkennt Z' die Diagonalsprache L_D. □

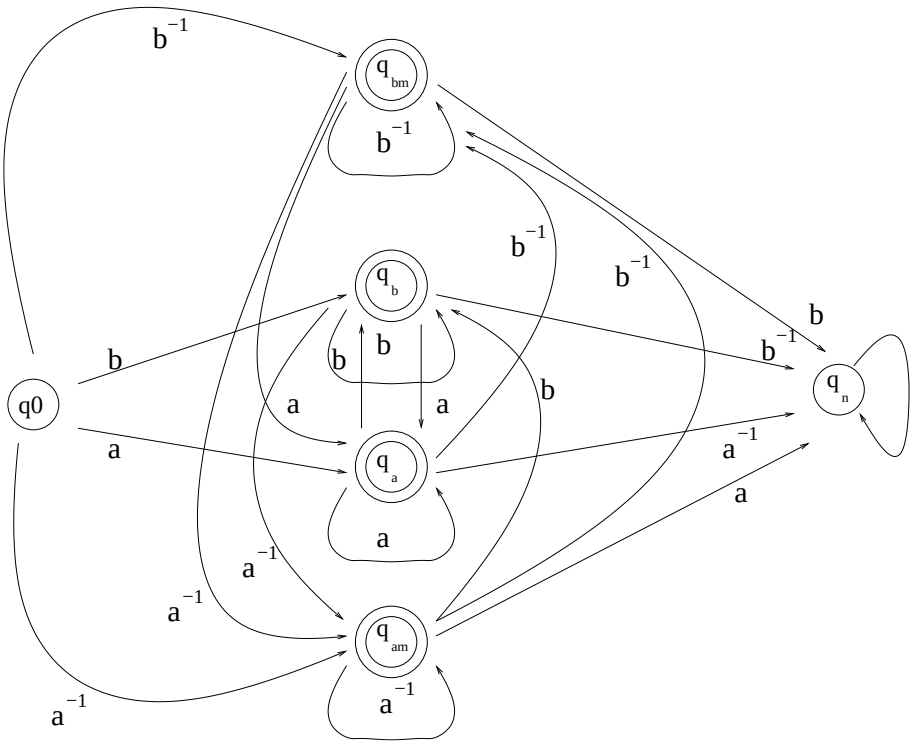


Abbildung 18: Ein endlicher Automat

4.4 Das Pumping Lemma

Satz 21. *Es sei $L \subset \Sigma^*$ eine reguläre Sprache. Dann gibt es eine Zahl n, so dass gilt:*

Für alle Wörter $x \in L$ mit $|x| \geq n$ gibt es Wörter $u, v, w \in \Sigma^*$, so dass

- $x = uvw$,
- $|v| \geq 1$,
- $|uv| \leq n$ *und*
- *für alle $i = 0, 1, 2, ...$ gilt $uv^i w \in L$.*

Proof. Es sei $M = (Z, \Sigma, \delta, z_0, F)$ der endliche Automat der L akzeptiert. Wir wählen $n := |Z|$. Es sei nun $x = x_1 x_2 .. x_m \in L$ ein Wort der Länge $|x| = m \geq n$. Bei der Anwendung des Wortes x auf den endlichen Automaten

M geht der Automat M in folgende Zustände über:

$$\delta(z_0, x_1x_2...x_i) \quad \text{für } i = 0, ..., n.$$

Dies sind $n+1$ Zustände. Nach dem Schubfachprinzip, gibt es einen Zustand $\tilde{z}$, den der endliche Automat mehrmals erreicht. Dies bedeutet es gibt $0 \leq k < j \leq n$ mit

$$\tilde{z} = \delta(z_0, x_1x_2...x_k) = \delta(z_0, x_1x_2...x_j).$$

Daher gilt

$$\delta(\tilde{z}, x_{k+1}...x_j) = \tilde{z}$$

und da M das Wort x akzeptiert

$$\delta(\tilde{z}, x_{j+1}...x_m) \in F.$$

Wir setzen daher

$$\begin{aligned} u &= x_1x_2...x_k \\ v &= x_{k+1}...x_j \\ w &= x_{j+1}...x_n. \end{aligned}$$

Durch Induktion zeigt man nun leicht

$$\delta(z_0, uv^iw) \in F$$

für alle $i \in \mathbb{N}_0$. Dies zeigt die Behauptung. □

Beispiel 48. *Es sei* $\Sigma = \{0, 1\}$ *und*

$$L = \{w_1w_2...w_n \in \Sigma^* \mid w_{n-1} = w_n\}.$$

Der endliche Automat, der diese Sprache erkennt, ist in Abbildung 17 dargestellt. Wir wählen $n = 5 = |Z|$. *Als Beispiel wählen wir das Wort* $x = 110011$. *Der Zustand* q_1 *wird bei der Eingabe* $x_1 = 1$ *und bei der Eingabe* $x_1x_2x_3x_4x_5 = 11001$ *erreicht. Daher können wir*

$$\begin{aligned} u &= 1 \\ v &= 1001 \\ w &= 1 \end{aligned}$$

wählen. Man sieht, dass auch $uv^3w = 11001100110011 \in L$.

Beispiel 49. *Es sei* $\Sigma = \{a, b\}$ *und*

$$L = \{a^m b^m \in \Sigma^* \mid m \in \mathbb{N}\}.$$

Es gibt keinen endlichen Automaten der diese Sprache akzeptiert.

Beweis: Wir nehmen an es gibt einen endlichen Automaten, der L akzeptiert. Dann gibt es nach dem Pumping Lemma Satz 21 eine Zahl n mit den Eigenschaften im Pumping Lemma. Wir wählen nun $x = a^n b^n$. Es sei nun $x = uvw$ die Zerlegung nach dem Pumping Lemma. Dann folgt $uv^2w \in L$ und $v = a^k$ mit $k \geq 1$. Daher ist $uv^2w = a^{n+k}b^n \in L$. Dies ist ein Widerspruch. □

Beispiel 50. *Es sei* $\Sigma = \{0\}$ *und*

$$L = \{0^{n^2} \in \Sigma^* \mid n \in \mathbb{N}\}.$$

Es gibt keinen endlichen Automaten der diese Sprache akzeptiert.

Beweis: Wir nehmen an es gibt einen endlichen Automaten, der L akzeptiert. Dann gibt es nach dem Pumping Lemma Satz 21 eine Zahl n mit den Eigenschaften im Pumping Lemma. Wir wählen nun $x = 0^{n^2}$. Es sei nun $x = uvw$ die Zerlegung nach dem Pumping Lemma. Dann folgt $uv^2w \in L$ und

$$\begin{aligned} n^2 = |x| = |uvw| &< |uv^2w| = \\ |uv^2w| &\leq |uv| + |uvw| = \\ |uv| + |x| &\leq n + n^2 < n^2 + 2n + 1 = \\ &= (n+1)^2. \end{aligned}$$

Dies zeigt $n^2 < |uv^2w| < (n+1)^2$. Damit ist aber $uv^2w \notin L$. Dies ist ein Widerspruch. □

Satz 22. *Es sei* $\Sigma = \Delta = \{0, 1\}$. *Es gibt keinen Moore-Automaten* $M_M = (Z, \Sigma, \Delta, \delta, \lambda, z_0)$ *mit folgender Eigenschaft:*

M_M berechnet die Quadrat-Funktion in folgendem Sinne. Es sei

$$E = \{e = e_0 e_1 ... e_n \mid e_i = 0 \quad \textit{für } (n-1)/2 \leq i \leq n\}.$$

Für die Eingabe $e = e_0 e_1 ... e_n \in E$ *und*
die Ausgabe $q = q_0 q_1 ... q_n = \hat{\lambda}(e)$ *gilt*

$$y = \sum_{i=0}^{n} q_i 2^i, \qquad x = \sum_{i=0}^{n} e_i 2^i$$

$$y = x^2.$$

Beweis: Wir nehmen an es gibt eine Moore-Automaten mit obiger Eigenschaft. Dann gibt es nach Satz 20 einen endlichen Automaten, der die Diagonalsprache

$$L_D = \{\mathcal{J}(w, \hat{\lambda}(w)) \mid w \in \Sigma^+\}$$

akzeptiert.

Dann gibt es nach dem Pumping Lemma Satz 21 eine Zahl n mit den Eigenschaften im Pumping Lemma.

Es sei nun:

$$\begin{aligned} w_i = w_{i0}...w_{iN} &\in \Sigma^N \\ w_{ij} &= \begin{cases} 1 & \text{falls} \quad i = j \\ 0 & \text{sonst} \end{cases} \\ N &= 40n \end{aligned}$$

Beachte, dass $\mathcal{J}(w_i, w_{2i}) \in L_D$ für $i = 1, ..., N/4$, da N groß genug gewählt wurde.

Wir wählen nun $x = \mathcal{J}(w_{n+1}, w_{2(n+1)})$. Es sei nun $x = uvw$ die Zerlegung nach dem Pumping Lemma. Dann ist $u, v \in \{\epsilon, (0,0)\}^n$ und es gilt

$$y = uv^2w \in L_D.$$

Man sieht

$$y = \mathcal{J}(w_{n+1+|v|}, w_{2(n+1)+|v|}).$$

Da $n+1+|v| < N/4$ muß $2(n+1)+|v| = 2(n+1+|v|)$ gelten. Hieraus folgt $|v| = 0$ was ein Widerspruch zur Eigenschaft von v im Pumping Lemma ist. □

Problem 4. *Es sei $n \in \mathbb{N}$ eine feste Zahl,*

$$\begin{aligned} \Sigma &= \{0, 1\} \\ \Delta &= \{k \mid 0 \leq k \leq 2^{2(n+1)}\} \cup \{\epsilon\}. \end{aligned}$$

Gesucht ist ein Moore-Automat $M_n = (Z, \Sigma, \Delta, \delta, \lambda, z_0)$,
der in folgendem Sinne die Quadratzahlen berechnet:

$$\begin{aligned} a_0...a_{2n+1} &= \hat{\lambda}(q_0...q_n) \\ y = \sum_{i=0}^{2n+1} a_i 2^i, \qquad & x = \sum_{i=0}^{n} q_i 2^i \\ y &= x^2. \end{aligned}$$

Lösung: Es sei $M = (Z, \Sigma, \Delta, \delta, \lambda, z_0)$ der Moore-Automat

$$
\begin{aligned}
Z &= \{z_{k,q} \mid q = q_0 q_1 ... q_n \in \Sigma^n,\ 0 \le k \le n+1\} \cup \{z_r\} \\
z_0 &= z_{0,0...0} \\
\delta(z_{k,q_0...q_n}, b) &= \begin{cases} z_{k+1,q_0 q_1 ... q_{k-1} b q_{k+1} ... q_n} & \text{falls } k \le n \\ z_r & \text{falls } k = n+1 \end{cases} \\
\lambda(z_{k,q_0..q_n}) &= \begin{cases} \epsilon & \text{falls } k \le n \\ a_0 a_1 ... a_{2n+1} & \text{falls } k = n+1 \text{ wobei } y = x^2 \\ & y = \sum_{i=0}^{2n+1} a_i 2^i, \\ & x = \sum_{i=0}^{n} q_i 2^i \end{cases}
\end{aligned}
$$

Dieser Moore Automat berechnet die Quadratzahlen x^2 für $0 \le x < 2^{n+1}$.

4.5 Der nichtdeterministische endliche Automat

Definition 49 (NEA). *Ein nichtdeterministischer Automat ist ein 5-Tupel $M = (Z, \Sigma, \delta, S, E)$, wobei*

- *Z eine endliche Menge von Zuständen,*
- *Σ eine endliche Eingabemenge,*
- *$\delta : Z \times \Sigma \to \mathcal{P}(Z)$ eine Überführungsfunktion*
- *$S \subset Z$ die Menge der Startzustände und*
- *$E \subset Z$ die Menge der Endzustände ist.*

Die Mengen Z, Σ sollten disjunkt sein.

Die allgemeine Überführungsfunktion des nichtdeterministischen Automaten M sei:

$$
\begin{aligned}
\hat{\delta} : \mathcal{P}(Z) \times \Sigma^* &\to \mathcal{P}(Z) \\
\hat{\delta}(Z', \epsilon) &= Z' \quad \text{für alle } Z' \subset Z, \\
\hat{\delta}(Z', aw) &= \bigcup_{z \in Z'} \hat{\delta}(\delta(z, a), w) \quad \text{für alle } a \in \Sigma, w \in \Sigma^*, Z' \subset Z.
\end{aligned}
$$

Man schreibt kurz δ anstatt $\hat{\delta}$.

Die von M akzeptierte Sprache ist

$$L_M := \{w \in \Sigma^* \mid \delta(S, w) \cap E \neq \emptyset\}$$

Beispiel 51. *Es sei* $\Sigma = \{0, 1\}$. *Die vom nichtdeterministischen endlichen Automaten* $M = (Z, \Sigma, \delta, S, E)$ *in Abbildung 19 akzeptierte Sprache ist*

$$L = \{w_1 w_2 ... w_n \mid n \geq 2 \wedge w_n = w_{n-1} = 0\} \cup \{0\} \subset \Sigma^*.$$

Hierbei ist

$$\begin{aligned} Z &= \{z_0, z_1, z_2\} \\ E &= \{z_2\} \\ S &= \{z_0, z_1\} \\ \delta(z_0, 0) &= \{z_0, z_1\} \\ \delta(z_0, 1) &= \{z_0\} \\ \delta(z_1, 0) &= \{z_2\} \\ \delta(z_1, 1) &= \emptyset \\ \delta(z_2, 0) &= \emptyset \\ \delta(z_2, 1) &= \emptyset. \end{aligned}$$

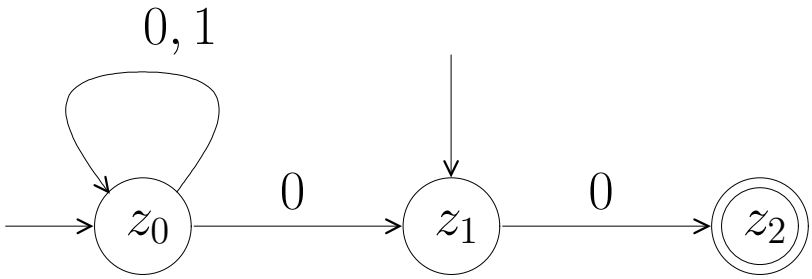


Abbildung 19: Ein nichtdeterministischer endlicher Automat

Beispiel 52. *Es sei* $\Sigma = \{0, 1\}$. *Die vom nichtdeterministischen endlichen Automaten in Abbildung 20 akzeptierte Sprache ist*

$$L = \{w_1 w_2 ... w_n \mid n \geq 2 \wedge w_n = w_{n-1}\} \subset \Sigma^*.$$

In Abbildung 17 ist ein endlicher Automat abgebildet, der diese Sprache erkennt.

Beispiel 53. *Es sei* $\Sigma = \{0, 1\}$ *und* k *eine feste Zahl. Die vom nichtdeterministischen endlichen Automaten in Abbildung 21 akzeptierte Sprache ist*

$$L_k = \{w_1 w_2 ... w_n \mid n \geq k \wedge w_{n-k+1} = 0\} \subset \Sigma^*.$$

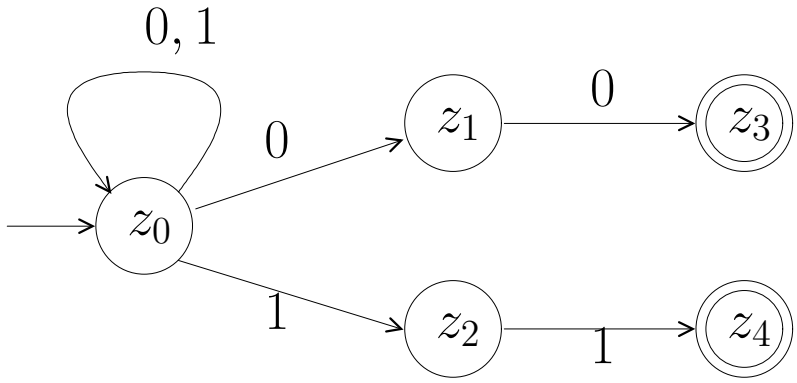


Abbildung 20: Ein nichtdeterministischer endliche Automat

Es gibt keinen endlichen Automaten mit weniger als 2^k Zuständen, der L_k akzeptiert.

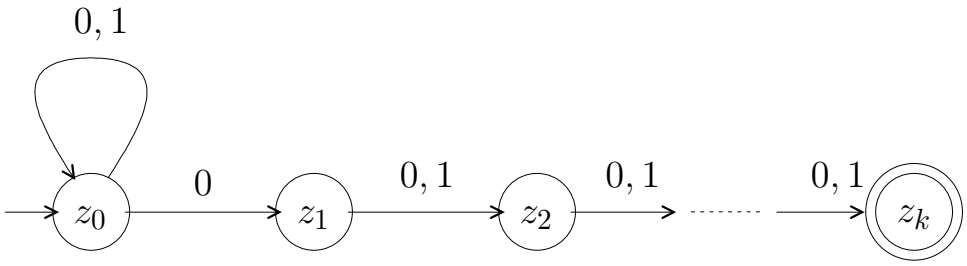


Abbildung 21: Ein nichtdeterministischer endliche Automat

Beweis: Es sei M ein endlicher Automat mit weniger als 2^k Zuständen, der L_k erkennt. Wegen des Schubfachprinzipes und $|\{0,1\}^k| = 2^k$, muss es zwei Wörter $y_1, y_2 \in \{0,1\}^k$, $y_1 \neq y_2$ geben, so dass

$$\hat{\delta}(z_0, y_1) = \hat{\delta}(z_0, y_2).$$

Es sei i die erste Position, an der sich y_1, y_2 unterscheiden ($1 \leq i \leq k$). Dies bedeutet, dass

$$y_1 = u0v, \quad y_2 = u1v',$$

wobei $|u| = i - 1$, $|v| = |v'| = k - i$ und wobei man hierzu eventuell y_1 und y_2 vertauschen muss. Nun sei $w \in \{0,1\}^{i-1}$ beliebig. Dann haben die Wörter

$$y_1 w = u0vw, \quad y_2 w = u1v'w,$$

die Eigenschaft, dass an der k-letzten Stelle der Buchstabe 0 beziehungsweise 1 vorkommt. Also ist

$$y_1 w = u0vw \in L_k, \quad y_2 w = u1v'w \notin L_k. \tag{2}$$

Außerdem folgt

$$\begin{aligned} \hat{\delta}(z_0, y_1 w) &= \hat{\delta}(\hat{\delta}(z_0, y_1), w) \\ &= \hat{\delta}(\hat{\delta}(z_0, y_2), w) \\ &= \hat{\delta}(z_0, y_2 w). \end{aligned}$$

Dies bedeutet $y_1 w \in L_k \Leftrightarrow y_2 w \in L_k$. Dies ist ein Widerspruch zu (2). □

Satz 23. *a) Es sei $L(M)$ die Sprache die von dem nichtdeterministischen Automat M akzeptiert wird.*
Dann gibt es einen deterministischen Automaten M', der die gleiche Sprache akzeptiert. Das heißt

$$L_{M'} = L_M.$$

b) Es sei $L(M')$ die Sprache die von dem deterministischen Automat M akzeptiert wird. Dann gibt es einen nichtdeterministischen Automaten M, der die gleiche Sprache akzeptiert. Das heißt

$$L_M = L_{M'}.$$

zu a) Es sei $M = (Z, \Sigma, \delta, S, E)$ ein nichtdeterministischer Automat. Dann sei

$$\begin{aligned} M' &= (\mathcal{Z}, \Sigma, \delta', z_0', E'), \\ \mathcal{Z} &= \mathcal{P}(Z), \\ \delta'(Z', a) &= \bigcup_{z \in Z'} \delta(z, a) = \hat{\delta}(Z', a), \quad Z' \in \mathcal{Z}, \\ z_0' &= S, \\ E' &= \{Z' \subset Z \mid Z' \cap E \neq \emptyset\}. \end{aligned}$$

Für ein Wort $x = a_1 ... a_n \in \Sigma^*$ gilt

$$x \quad \in \quad L_M$$

$$
\begin{array}{rcl}
 & \Updownarrow & \\
\delta(S,x) \cap E & \neq & \emptyset \\
 & \Updownarrow & \\
\text{Es gibt eine Folge } Z_1, Z_2, ...Z_n \subset Z & \text{mit:} & \\
\delta'(S,a_1) = Z_1, \delta'(Z_1,a_2) = Z_2 & ,..., & \delta'(Z_{n-1},a_n) = Z_n, \quad Z_n \cap E \neq \emptyset \\
 & \Updownarrow & \\
\hat{\delta}'(S,x) & \in & E' \\
 & \Updownarrow & \\
x & \in & L_{M'}.
\end{array}
$$

zu b) Es sei $M' = (Z, \Sigma, \delta', z_0', E)$ ein deterministischer Automat. Dann sei

$$
\begin{aligned}
M &= (Z, \Sigma, \delta, \{z_0'\}, E), \\
\delta(z,a) &= \{\delta'(z,a)\}.
\end{aligned}
$$

Man sieht, dass $L_M = L_{M'}$. $\square$

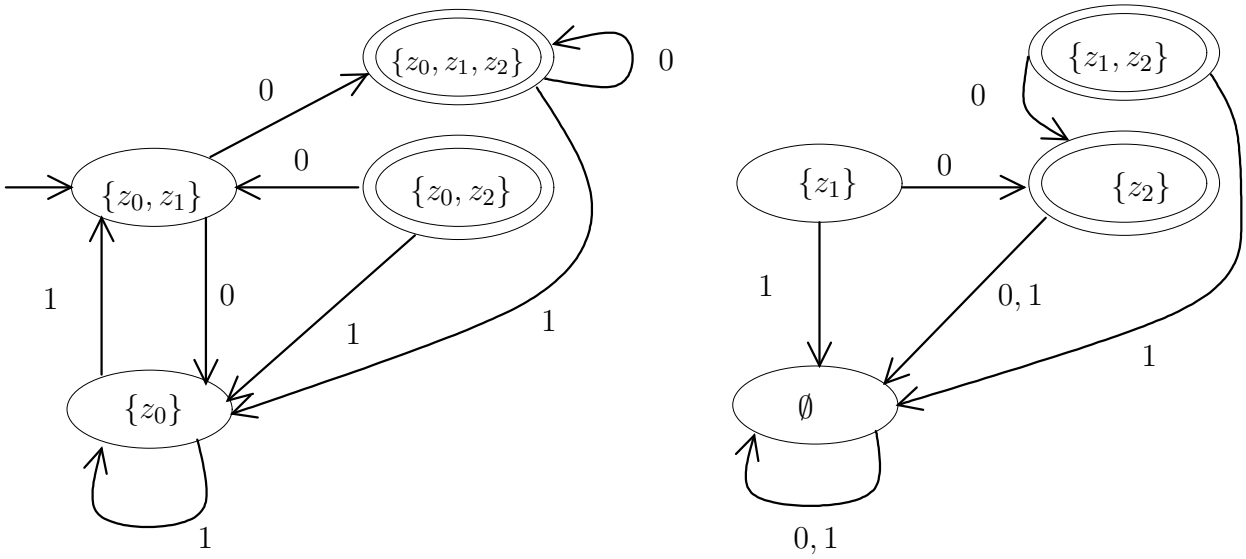


Abbildung 22: Deterministischer endlicher Automat zum nichtdeterministischen endlichen Automaten in Abbildung 19.

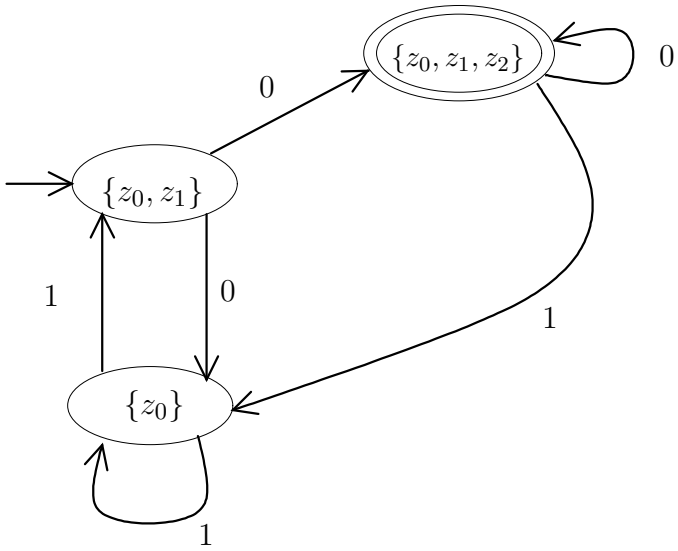


Abbildung 23: Deterministischer endlicher Automat von Abbildung 19 ohne überflüssige Zustände.

Beispiel 54. *Der nach dem Beweis von Satz 23 konstruierte deterministische endliche Automat zum nichtdeterministischen endlichen Automaten in Beispiel 51 ist in Abbildung 22 dargestellt.*

Unter Wegnahme der überflüssigen Zustände erhält man den endlichen Automaten in Abbildung 23.

4.6 Satz von Myhill-Nerode und Minimalautomaten

Definition 50. *Es sei $\tau \in R \times R$ eine Äquivalenzrelation. Die Anzahl der Äquivalenzklassen von τ heißt der Index von τ.*

Beispiel 55.

- *Die Restklassenrelation Modulo p hat Index p.*
- $=\ \in \mathbb{N} \times \mathbb{N}$ *hat Index* ∞.

Definition 51. *Eine binäre Relation $\tau \in H \times H$ auf einer Halbgruppe H heißt rechts-invariant, falls*

$$\forall z \in H \quad (x \tau y \Rightarrow xz \tau yz).$$

Beispiel 56.

- $=\ \in \mathbb{R} \times \mathbb{R}$ *ist rechts-invariant. (Äquivalenzrelation)*

- $<\ \in \mathbb{N} \times \mathbb{N}$ *ist rechts-invariant. (keine Äquivalenzrelation)*
- $<\ \in \mathbb{Z} \times \mathbb{Z}$ *ist* nicht *rechts-invariant. (keine Äquivalenzrelation)*

Konstruktion 1. *Es sei $M = (Z, \Sigma, \delta, z_0, F)$ ein endlicher Automat. Die Äquivalenzrelation R_M auf Σ^* zu M ist wie folgt definiert:*

$$xR_My \ :\Leftrightarrow\ \delta(z_0, x) = \delta(z_0, y).$$

Bemerkung:

- R_M ist eine rechts-invariante Äquivalenzrelation.
- R_M hat endlichen Index.
- Es sei L die von M akzeptierte Sprache. Dann ist L die Vereinigung einiger Äquivalenzklassen von R_M.

Konstruktion 2. *Es sei $L \subset \Sigma^*$ eine Sprache. Die Äquivalenzrelation R_L zu der Sprache L ist wie folgt definiert:*

$$xR_Ly \ :\Leftrightarrow\ (\forall z \in \Sigma^*\ \ (xz \in L \Leftrightarrow yz \in L)\,.$$

Bemerkung: R_L ist eine rechts-invariante Äquivalenzrelation.

Konstruktion 3. *Es sei R_L die rechts-invariante Äquivalenzrelation zu L auf Σ^*.*
R_L habe endlichen Index.
Dann sei $M_{R_L} = (Z', \Sigma, \delta', z_0', F')$ der endliche Automat mit

$$\begin{aligned} Z' &= \{[x] \mid x \in \Sigma^*\}, \\ \delta'([z], a) &= [za], \quad \forall a \in \Sigma \\ z_0' &= [\epsilon], \\ F' &= \{[z] \mid z \in L\}. \end{aligned}$$

Bemerkung: Es gilt:

$$\begin{aligned} \delta([z], w) &= [zw] \quad \forall w \in \Sigma^* \\ R_{M_{R_L}} &= R_L. \end{aligned}$$

Satz 24 (Satz von Myhill-Nerode). *Folgende Aussagen sind äquivalent:*

a) Die Sprache $L \subset \Sigma^*$ ist regulär.

b) Es gibt eine rechts-invariante Äquivalenzrelation R mit endlichem Index,
so dass L die Vereinigung einiger Äquivalenzklassen von R ist.

c) Der Index von R_L ist endlich.

Beweis:
a) $\rightarrow$ b) Es sei $M = (Z, \Sigma, \delta, z_0, F)$ der endliche Automat, der L akzeptiert. Der Index von R_M ist endlich, da M nur endlich viele Zustände hat. Man sieht

$$L \;=\; \{x \mid \delta(z_0, x) \in F\} \;=\; \bigcup_{f \in F} [f]_{R_M}.$$

b) $\rightarrow$ c) Dass L die Vereinigung einiger Äquivalenzklassen von R ist, bedeutet dass es $x_1, x_2, ..., x_n$ gibt, so dass

$$L \;=\; \bigcup_{i=1}^{n} [x_i]_R \tag{3}$$

Es sei xRy. Da R rechts-invariant, gilt $\forall z \in \Sigma^* \quad (xzRyz)$. Wegen (3) gilt daher

$$xz \in L \;\Leftrightarrow\; yz \in L.$$

Dies bedeutet xR_Ly. Also ist

$$R \subset R_L. \tag{4}$$

Daher ist der Index von R_L kleiner gleich dem Index von R. Also ist der Index von R_L endlich.

c) $\rightarrow$ a) Betrachte den endlichen Automaten M_m nach Konstruktion 3. Dann gilt $\delta([\epsilon], w) = [w]$. Also akzeptiert M_m das Wort w genau dann wenn $w \in L$.
□

Satz 25 (Minimalautomat). *Es sei $L \subset \Sigma^*$ eine reguläre Sprache. M_{R_L} ist der endliche Automat mit der geringsten Anzahl von Zuständen, der L akzeptiert (Minimalautomat). Jeder L akzeptierende endliche Automat M*

mit der selben Anzahl von Zuständen wie M_{R_L} ist isomorph zu M_{R_L} und es gilt

$$R_M \subset R_{M_{R_L}}.$$

.

Beweis: Es seien M und M' endliche Automaten bei denen man jeden Zustand durch die Eingabe eines Wortes erreichen kann. Falls

$$R_M \subset R_{M'}$$

für zwei endliche Automaten, dann hat M mindestens so viele Zustände wie M'. Wegen (4) gilt aber

$$R_M \subset R_L = R_{M_{R_L}}.$$

Daher hat M mindestens so viele Zustände wie M_{R_L}.

Falls M' die gleiche Anzahl von Zuständen wie $M = M_{R_L}$ hat, dann ist

$$R_M = R_{M'}.$$

Ein Isomorphismus ist nun durch

$$\begin{array}{rcl} Z' & \to & Z \\ z' & \mapsto & z \quad \text{falls es ein } w \in \Sigma^* \text{ mit} \\ & & \delta'(z'_0, w) = z' \\ & & \delta(z_0, w) = z \\ & & \text{gibt.} \end{array}$$

gegeben. □

Beispiel 57.

$$L = \{a^n b^n \mid n \geq 1\}$$

ist nicht regulär.

Beweis: Die folgenden Mengen sind Äquivalenzklassen von R_L:

$$\begin{array}{rcl} [ab] & = & L \\ [a^2b] & = & \{a^2b, a^3b^2, a^4b^3, ...\} \\ [a^3b] & = & \{a^3b, a^4b^2, a^5b^3, ...\} \\ & \vdots & \\ [a^kb] & = & \{a^{k+i-1}b^i \mid i \geq 1\} \\ & \vdots & \end{array}$$

□

4.7 Konstruktion von Minimalautomaten

Konstruktion 4. *Es sei $M = (Z, \Sigma, \delta, z_0, F)$ ein endlicher Automat, bei dem jeder Zustand durch Eingabe eines Wortes erreicht werden kann.*

Dann definiere folgende Äquivalenzrelation auf Z:

$$z \equiv z' \;:\Leftrightarrow\; (\forall w \in \Sigma^* \;\; (\delta(z, w) \in F \Leftrightarrow \delta(z', w) \in F)\,.$$

Der Minimalautomat zu L ist nun:

$$\begin{aligned} M' &= (Z', \Sigma, \delta', z_0', F') \\ Z' &= \{[z]_\equiv \mid z \in Z\} \\ \delta'([z]_\equiv, a) &= [\delta(z, a)]_\equiv \\ z_0' &= [z_0]_\equiv \\ F' &= \{[z]_\equiv \mid z \in F\}. \end{aligned}$$

Bemerkung: M' ist der Minimalautomat von L. Dies erkennt man aus $R_{M'} \subset R_{M_{R_L}}$ der Definition von $R_{M'}$ und der Konstruktion von M'.

Der Minimalautomat lässt sich wie folgt konstruieren:

1. Stelle eine Tabelle für alle Paare von Zuständen $\{z, z'\}$, mit $z \neq z'$ auf.
2. Markiere alle Paare $\{z, z'\}$ mit $z \in F$ und $z' \notin F$. Eine Markierung bedeutet dass für diese Zustände gilt $z \not\equiv z'$.
3. Markiere $\{z, z'\}$, falls es ein $a \in \Sigma$ gibt, so dass

   $$\{\delta(z, a), \delta(z', a)\}$$

   markiert ist.
4. Falls keine weitere Markierung mehr möglich ist, kann man alle unmarkierten Paare $\{z, z'\}$ zu einem Paar verschmelzen.

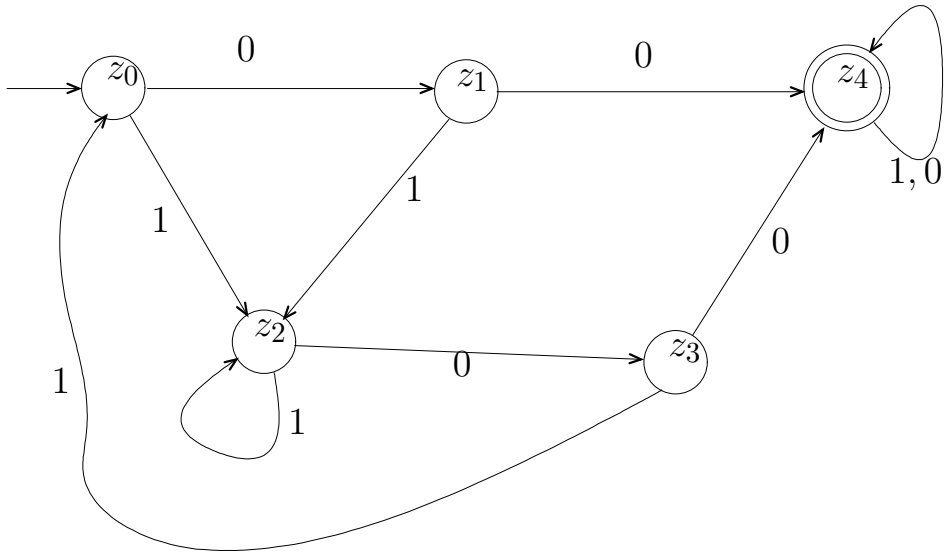
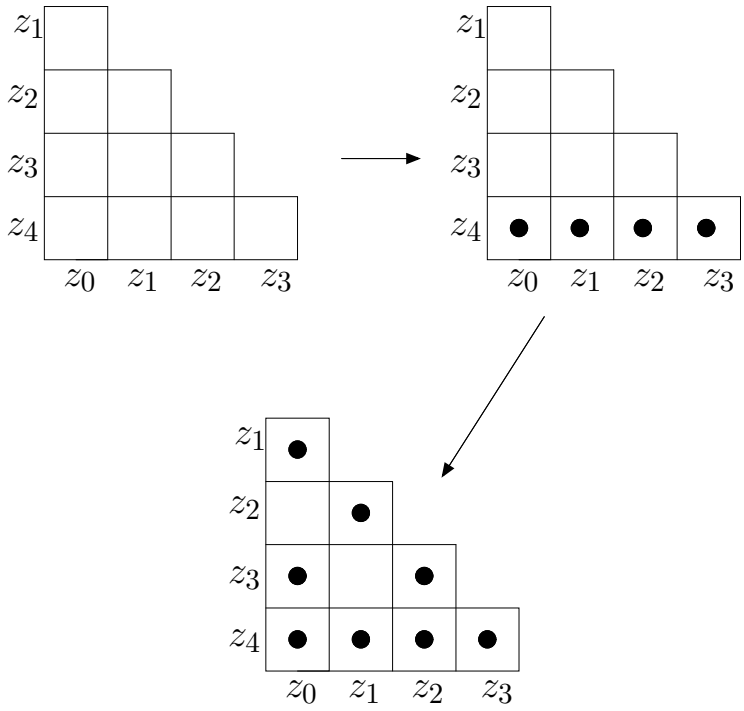


Abbildung 24: Kein minimaler endlicher Automat

Beispiel 58. *Der endliche Automat in Abbildung 24 ist nicht minimal. Durch nebenstehenden Minimalisierungsprozess erhält man einen minimalen endlichen Automaten, der in Abbildung 25 dargestellt ist. Die von diesem Automaten akzeptierte Sprache ist*

$$L = \{x \in \{0,1\}^* \mid x \text{ enthält Infix } 00\}.$$



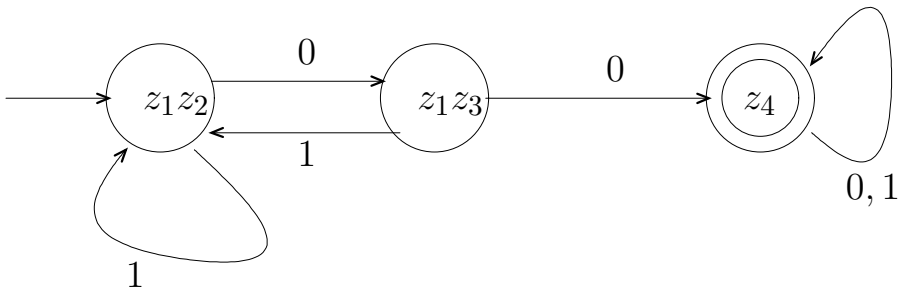


Abbildung 25: Minimaler endlicher Automat

5 Reguläre Ausdrücke und Grammatiken

5.1 Reguläre Ausdrücke

Mittels des Befehls `egrep` (Linux Betriebssystem, na klar!) kann man reguläre Ausdrücke in einem Textfile finden.

file.txt:

```
aab
aaa
bba
```

Eingabe		Ausgabe
`egrep '(aa)'`	`file.txt`	`aab, aaa`
`egrep '( aa )'`	`file.txt`	
`egrep '( aab )'`	`file.txt`	`aab`
`egrep '(aa){1}'`	`file.txt`	`aab, aaa`
`egrep ' (b)(ba|ab)* '`	`file.txt`	`bba`

Definition 52. *Reguläre Ausdrücke über einem Alphabet Σ sind wie folgt rekursiv definierte Formeln:*

- *$\emptyset$ ist ein regulärer Ausdruck.*
- *ϵ ist ein regulärer Ausdruck.*
- *Für jedes $a \in \Sigma$ ist a ein regulärer Ausdruck.*
- *Falls α und β reguläre Ausdrücke sind, dann auch $\alpha\beta, (\alpha|\beta)$ und $(\alpha)*$.*

Beispiel 59. *Für $\Sigma = \{0,1\}$ sind 00 und $1(10|01)^*$ reguläre Ausdrücke.*

Definition 53. *Die Sprache $L(\gamma)$ zu einem regulären Ausdruck γ über einem Alphabet Σ ist wie folgt rekursiv definiert:*

- $L(\emptyset) = \{\emptyset\}$.
- $L(\epsilon) = \{\epsilon\}$.
- $L(a) = a$ *für $a \in \Sigma$.*
- $L(\alpha\beta) = L(\alpha)L(\beta)$.
- $L(\alpha|\beta) = L(\alpha) \cup L(\beta)$ *für reguläre Ausdrücke $\alpha\beta$.*
- $L((\alpha)^*) = L(\alpha)^*$.

Beispiel 60. *Für $\Sigma = \{0,1\}$ ist*

$$\begin{aligned} L(00) &= \{00\} \\ L(1(10|01)^*) &= \{1\}\{10,01\}^* = \{110, 101, 11010, ...\} \\ L((0|1)^*(00|11)) &= \{w_1 w_2 ... w_n \in \Sigma^* \mid w_{n-1} = w_n\} \end{aligned}$$

$L((0|1)^*(00|11))$ ist die vom Automaten in Abbildung 46 akzeptierte Sprache.

Lemma 6.

$$\begin{aligned} L((\alpha|\beta)\gamma) &= L(\alpha\gamma|\beta\gamma) \\ L((\epsilon|\alpha)^*) &= L(\alpha^*). \end{aligned}$$

Satz 26 (Kleene). *Die Menge der durch reguläre Ausdrücke beschreibbaren Sprachen ist genau die Menge der regulären Sprachen.*

Beweis: "$\Rightarrow$" Es sei γ ein regulärer Ausdruck und $L(\gamma) \subset \Sigma^*$ die Sprache, die dieser Ausdruck beschreibt.

Wir führen Induktion über die Länge l des regulären Ausdruck γ.

Länge $l = 1$: Für $\gamma = \emptyset$, $\gamma = \epsilon$ und $\gamma = a$, $\quad a \in \Sigma$ findet man leicht einen endlichen Automaten, der $L(\gamma)$ akzeptiert.

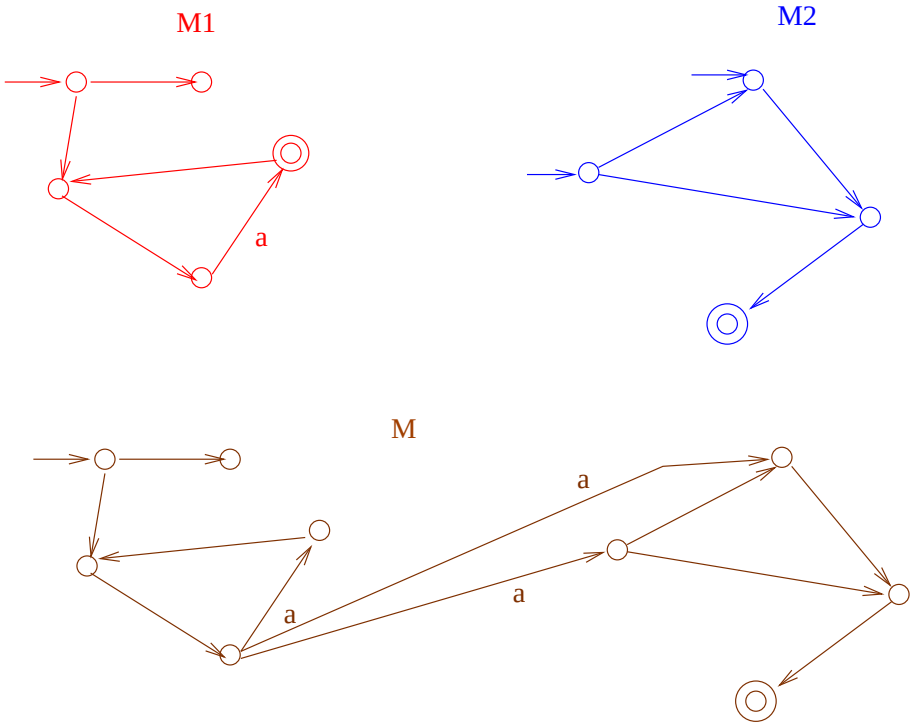


Abbildung 26: Konstruktion eines Serienautomaten.

$\underline{l-1 \to l}$:
FALL 1: $\gamma = \alpha\beta$. Nach Induktionsvoraussetzung gibt es NEA's M_1, M_2, die $L(\alpha)$ und $L(\beta)$ akzeptieren. Wir konstruieren folgenden Automaten M in "Serie" (siehe auch Abbildung 26):

- M hat die Zustände von M_1 und M_2.
- Startzustände sind die Startzustände von M_1. Falls $\epsilon \in L(\alpha)$, dann sind zusätzlich die Startzustände von M_2 Startzustände.
- Endzustände sind die Endzustände von M_2.
- Man beschreibe die Überführungsfunktion von M mit Pfeilen in einem Graphen. M habe dann die Pfeile von M_1 und M_2 und folgende zusätzlichen Pfeile: Falls ein Pfeil in M_1 von einem Zustand zu einem Endzustand geht, dann führe von diesem Zustand Pfeile zu jedem Startzustand in M_2 mit der gleichen Markierung.

Man sieht dass M die Sprache $L(\gamma) = L(\alpha\beta)$ akzeptiert.

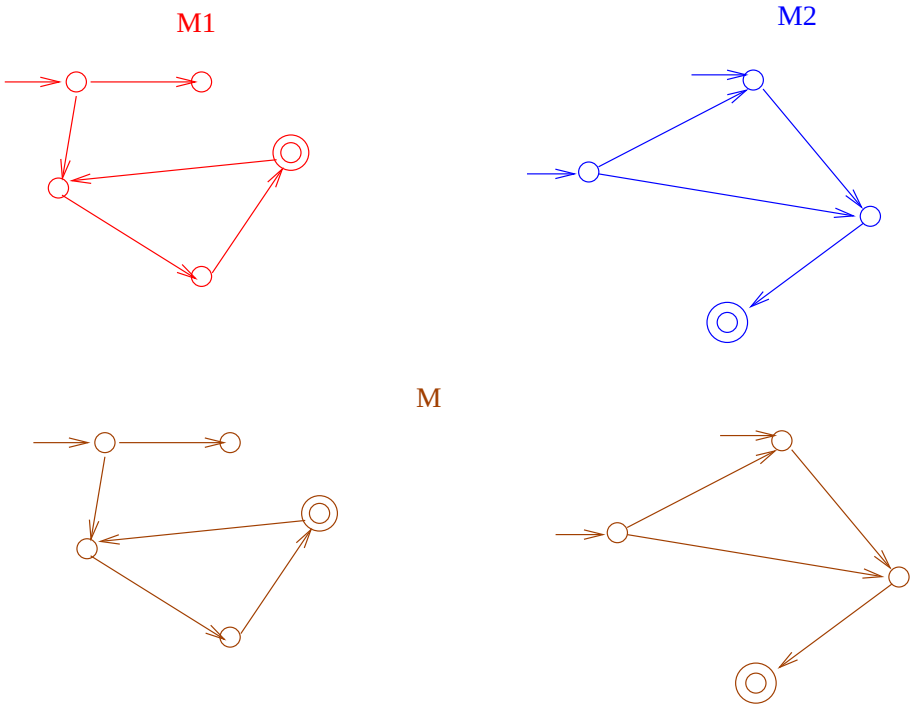


Abbildung 27: Konstruktion eines Vereinigungsautomaten.

FALL 2: $\gamma = (\alpha|\beta)$. Nach Induktionsvoraussetzung gibt es NEA's $M_1 = (Z_1, \Sigma, \delta_1, S_1, E_1)$ und $M_2 = (Z_2, \Sigma, \delta_2, S_2, E_2)$, $Z_1 \cap Z_2 = \emptyset$, die $L(\alpha)$ und $L(\beta)$ akzeptieren. Dann akzeptiert

$$M = (Z_1 \cup Z_2, \Sigma, \delta_1 \cup \delta_2, S_1 \cup S_2, E_1 \cup E_2)$$

die Sprache $L(\gamma) = L(\alpha|\beta)$. (siehe Abbildung 27).

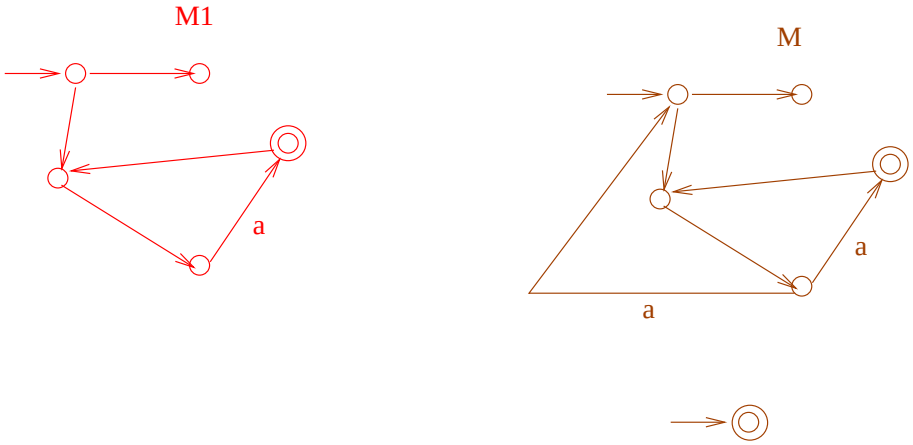


Abbildung 28: Konstruktion eines $*$-Automaten.

FALL 3: $\gamma = (\alpha)^*$. Nach Induktionsvoraussetzung gibt es einen NEA M_1, der $L(\alpha)$ akzeptiert. Wir konstruieren folgenden Automaten M (siehe auch Abbildung 28):

- M hat die Zustände von M_1 und einen weiteren Zustand z_e der zu akzeptieren des leeren Wortes ϵ benötigt wird.
- Startzustände sind die Startzustände von M_1 und z_e.
- Endzustände sind die Endzustände von M_1 und z_e.
- Man beschreibe die Überführungsfunktion von M mit Pfeilen in einem Graphen. M habe dann die Pfeile von M_1 und folgende zusätzlichen Pfeile: Falls ein Pfeil in M_1 von einem Zustand zu einem Endzustand geht, dann führe von diesem Zustand Pfeile zu jedem Startzustand in M_1 mit der gleichen Markierung.

Man sieht dass M die Sprache $L(\gamma) = L((\alpha)^*)$ akzeptiert.

"$\Leftarrow$" Es sei M ein DEA, der die Sprache $L \subset \Sigma^*$ akzeptiert. Die Zustände von M seien $z_1, ..., z_n$, wobei z_1 der Startzustand.

Für $i, j \in \{1, ..., n\}$ und $k \in \{0, 1, ..., n\}$ definiere die Sprache $R^k_{i,j}$ wie folgt:

$$\begin{aligned} R^k_{i,j} \quad := \quad & \{x = x_1...x_s \in \Sigma^* \mid \delta(z_i, x) = z_j \text{ und} \\ & \text{für jeden Zwischenzustand } \delta(z_i, x_1...x_l) = z_{k'},\ 1 \leq l < s \\ & \text{gilt } k' \leq k\}. \end{aligned}$$

Durch Induktion nach k zeigen wir, dass $R^k_{i,j}$ sich durch einen regulären Ausdruck beschreiben lässt.

$\underline{k = 0:}$ Für $k = 0$ und $i \neq j$ gilt:

$$R^0_{i,j} = \{a \in \Sigma \mid \delta(z_i, a) = z_j\}.$$

Für $k = 0$ und $i = j$ gilt:

$$R^0_{i,i} = \{\epsilon\} \cup \{a \in \Sigma \mid \delta(z_i, a) = z_i\}.$$

$R^0_{i,j}$ ist also immer endlich und lässt sich daher durch einen regulären Ausdruck beschreiben.

$\underline{k \to k+1:}$ Beobachte

$$R^{k+1}_{i,j} = R^k_{i,j} \cup R^k_{i,k+1}(R^k_{k+1,k+1})^* R^k_{k+1,j}.$$

Es sei $\alpha^k_{i,j}$ ein regulärer Ausdruck für $R^k_{i,j}$. Dann ist

$$\alpha^{k+1}_{i,j} = (\alpha^k_{i,j} \mid \alpha^k_{i,k+1}(\alpha^k_{k+1,k+1})^* \alpha^k_{k+1,j}).$$

Damit ist bewiesen, dass $R^k_{i,j}$ sich durch einen regulären Ausdruck beschreiben lassen.

Es seien $i_1, ..., i_m$ die Indizes der Endzustände. Dann gilt

$$L = \bigcup_{z_i \in E} R^n_{1,i} = \bigcup_{s=1}^{m} R^n_{1,i_s}.$$

Also ist

$$L(\alpha^n_{1,i_1} \mid \alpha^n_{1,i_2} \mid ... \mid \alpha^n_{1,i_s}) = L.$$

□

Beispiel 61. *Folgende Tabelle zeigt die Konstruktion des regulären Ausdrucks zum endlichen Automaten in Abbildung 29. Hierbei wurde die Formel*

$$R^{k+1}_{i,j} = R^k_{i,j} \cup R^k_{i,k+1}(R^k_{k+1,k+1})^* R^k_{k+1,j}$$

und einige Vereinfachungsformeln für reguläre Ausdrücke verwendet:

$L(...) =$	R^k_{11}	R^k_{12}	R^k_{13}	R^k_{21}	R^k_{22}	R^k_{23}	R^k_{31}	R^k_{32}	R^k_{33}
$k = 0$	ϵ	$1\|0$	$\emptyset$	$\emptyset$	$0\|\epsilon$	1	$\emptyset$	0	$1\|\epsilon$
$k = 1$	ϵ	$1\|0$	$\emptyset$	$\emptyset$	$0\|\epsilon$	1	$\emptyset$	0	$1\|\epsilon$
$k = 2$	ϵ	$(1\|0)0^*$	$(1\|0)1$	$\emptyset$	0^*	0^*1	$\emptyset$	00^*	$1\|\epsilon$
$k = 3$			$(1\|0)(0\|1)^*1$						

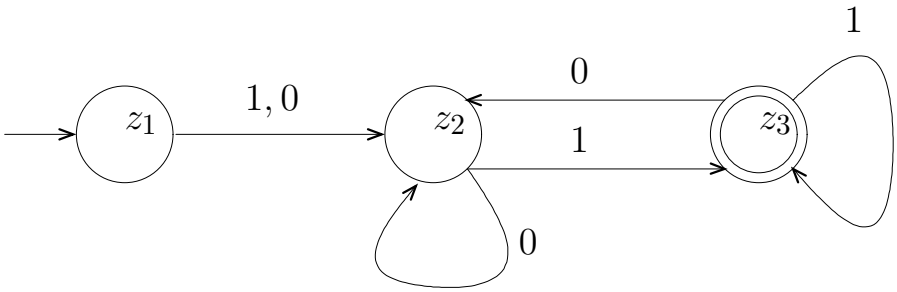


Abbildung 29: Endlicher Automat zu Beispiel 61

Satz 27. *Die regulären Sprachen sind abgeschlossen unter Vereinigung, Produkt, Stern, Komplement und Schnitt.*

Beweis: Die Abschlusseigenschaft unter Vereinigung, Produkt und Stern folgt aus dem Satz von Kleene und dass für reguläre Ausdrücke α, β gilt

$$\begin{aligned} L(\alpha\beta) &= L(\alpha)L(\beta) \\ L(\alpha|\beta) &= L(\alpha) \cup L(\beta) \\ L((\alpha)^*) &= L(\alpha)^*. \end{aligned}$$

Falls $M = (Z, \Sigma, \delta, z_0, E)$ ein DEA ist, dann ist $M = (Z, \Sigma, \delta, z_0, Z \backslash E)$ ein DEA, der die Komplementsprache erkennt.

Aus der Morganschen Komplementenregel folgt die Abschlusseigenschaft bezüglich Schnitt.

5.2 Grammatiken

Deutsche Sätze:

- Der Baum hat schöne Blätter. (grammatikalisch richtig)
- Schöne der hat Blätter Baum. (grammatikalisch falsch)

C-code:

- ```
  for(i=0;i<5;++i) {
    a[i] = b[i] + c[i];
  }
  ```
  (grammatikalisch richtig)
  ```

- ```
  for(i=1...5) {
    a(i) = b(i) + c(i);
  }
  ```
  (grammatikalisch falsch)

$$
\begin{aligned}
< \text{Satz} > &\rightarrow < \text{Subjekt} >< \text{Prädikat} >< \text{Objekt} > \\
< \text{Subjekt} > &\rightarrow < \text{Artikel} >< \text{Attribut} >< \text{Substantiv} > \\
< \text{Objekt} > &\rightarrow < \text{Artikel} >< \text{Attribut} >< \text{Substantiv} > \\
< \text{Artikel} > &\rightarrow \epsilon \\
< \text{Artikel} > &\rightarrow \text{der} \\
< \text{Artikel} > &\rightarrow \text{die} \\
< \text{Artikel} > &\rightarrow \text{das} \\
< \text{Adjektiv} > &\rightarrow \text{schöne} \\
< \text{Adjektiv} > &\rightarrow \epsilon \\
< \text{Substantiv} > &\rightarrow \text{Blätter} \\
< \text{Substantiv} > &\rightarrow \text{Baum} \\
< \text{Prädikat} > &\rightarrow \text{hat}
\end{aligned}
$$

Definition 54. *Eine Grammatik ist ein 4-Tupel* $G = (V, \Sigma, P, S)$*, wobei*

- V *endliche Menge der* Variablen*,*
- Σ *endliche Menge des* Terminalalphabets*,*
- $\Sigma \cap V = \emptyset$*,*
- $\xrightarrow{P} \subset (V \cup \Sigma)^+ \times (V \cup \Sigma)^*$ *Relation der Regeln,*
- $S \in V$ *Startvariable.*
  ```

Definition 55. *Sei $G = (V, \Sigma, P, S)$ eine Grammatik. Die Ableitungsrelation $\Rightarrow_G$ auf $(V \cup \Sigma)^*$ ist definiert durch*

$$\begin{array}{lll} u \Rightarrow_G v & \text{genau dann wenn es} & \\ & x, y, y', z \in (V \cup \Sigma)^* \quad \text{gibt, so dass} & \\ & y \to y' \qquad u = xyz \qquad v = xy'z. & \end{array}$$

Des weiteren schreiben wir

$$\begin{array}{ll} u \Rightarrow_G^* v & \text{genau dann wenn } u = v \text{ oder wenn es} \\ & y_1, ..., y_n \in (V \cup \Sigma)^* \quad \text{gibt, so dass} \\ & u \Rightarrow_G y_1,\ y_1 \Rightarrow_G y_2,\ \ldots\ y_{n-1} \Rightarrow_G y_n,\ y_n \Rightarrow_G v. \\ & (\Rightarrow_G^* \text{ ist die reflexive transitive Hülle von } \Rightarrow_G.) \end{array}$$

Die Sprache $L(G)$ zur Grammatik $G = (V, \Sigma, P, S)$ ist

$$L(G) = \{w \in \Sigma^* \mid S \Rightarrow_G^* w\}.$$

Oft schreibt man kurz: $u \Rightarrow_G v$ anstatt $u \Rightarrow_G^* v$.

Beispiel 62. *$G = (\{E\}, \{+, *, (,), a\}, P, E)$ wobei*

$$\begin{array}{rll} P & = \{ & E \to E + E, \\ & & E \to E * E, \\ & & E \to (E), \\ & & E \to a \qquad \}. \end{array}$$

$L(G)$ besteht aus allen Ausdrücken der Form $a{+}a, a{*}a{+}a, a, ...$ Zum Beispiel ist

$$\begin{array}{rl} E & \Rightarrow_G \quad E * E \Rightarrow_G E * (E + E) \Rightarrow_G a * (E + E) \Rightarrow_G a * (a + E) \\ & \Rightarrow_G \quad a * (a + a). \end{array}$$

Beispiel 63. *$G = (\{S, A\}, \{0, 1\}, P, S)$, wobei*

$$\begin{array}{rll} P & = \{ & S \to 1A, \\ & & S \to 0S, \\ & & A \to 0S, \\ & & A \to 1A, \\ & & A \to 1, \\ & & S \to 1 \qquad \}. \end{array}$$

Dann ist

$$L(G) = \{w_1 w_2 ... w_n \in \{0,1\}^* \mid w_n = 1 \wedge n \geq 1\}.$$

Zum Beispiel ist

$$E \Rightarrow_G 1A \Rightarrow_G 11S \Rightarrow_G 110S \Rightarrow_G 1101.$$

Beispiel 64.

$$\begin{array}{lcl} G & = & (V, \Sigma, P, S) \\ V & = & \{S, B\} \\ \Sigma & = & \{a, b\} \\ P & = \{ & S \to aSb, \\ & & S \to ab, \qquad \}. \end{array}$$

Dann ist

$$L(G) = \{a^n b^n \mid n \geq 1\}.$$

Zum Beispiel ist

$$\begin{array}{rcl} S & \Rightarrow_G & aSb \Rightarrow_G aaSbb \\ & \Rightarrow_G & aaabbb. \end{array}$$

Analog zeigt man $a^n b^n \in L(G)$ für beliebiges $n \in \mathbb{N}$.

Nach der Anwendung jeder Regel erhält die gleiche Anzahl von a wie b.

Beispiel 65.

$$\begin{array}{lcl} G & = & (V, \Sigma, P, S) \\ V & = & \{S, B, C\} \\ \Sigma & = & \{a, b, c\} \\ P & = \{ & S \to aSBC, S \to aBC, \\ & & CB \to BC, aB \to ab \\ & & bB \to bb, bC \to bc, \\ & & cC \to cc \qquad \}. \end{array}$$

Dann ist

$$L(G) = \{a^n b^n c^n \mid n \geq 1\}.$$

Zum Beispiel ist

$$\begin{aligned}
\textit{(a's einbauen:)}\ S &\Rightarrow_G aSBC \Rightarrow_G aaSBCBC \\
&\Rightarrow_G aaaBCBCBC \\
\textit{(BC's vertauschen:)} &\Rightarrow_G aaaBBCCBC \Rightarrow_G aaaBBCBCC \\
&\Rightarrow_G aaaBBBCCC \\
\textit{(b's einbauen:)} &\Rightarrow_G aaabBBCCC \Rightarrow_G aaabbBCCC \\
&\Rightarrow_G aaabbbCCC \\
\textit{(c's einbauen:)} &\Rightarrow_G aaabbbcCC \Rightarrow_G aaabbbccC \Rightarrow_G aaabbbccc.
\end{aligned}$$

Analog zeigt man $a^n b^n c^n \in L(G)$ für beliebiges $n \in \mathbb{N}$.

Nach der Anwendung jeder Regel erhält man die gleiche Anzahl von a (oder A) wie b (oder B) und c (oder C). Durch die ersten beiden Regeln bekommt man a^n als Präfix. b's kann man nur einfügen, wenn $a^n b^m$ mit $0 \leq m < n$ Präfix ist. Man erhält dann nach Einfügen eines b das Präfix $a^n b^{m+1}$. Das gleiche gilt für das Einfügen von c's. Also haben alle Wörter in L die Form $a^n b^n c^n$.

Beispiel 66 (L-Systeme). *Mit Hilfe L-Systemen kann man Pflanzenwachstum beschreiben ein ganz einfaches Modell hierzu ist das 0L-System zur Entwicklung des Bakterium Anabaena Catenula.*

$$\begin{aligned}
G &= (V, \Sigma, P, \overrightarrow{A}) \\
V &= \{\overleftarrow{A}, \overrightarrow{A}, \overleftarrow{B}, \overrightarrow{B}\} \\
\Sigma &= \{\overleftarrow{a}, \overrightarrow{a}, \overleftarrow{b}, \overrightarrow{b}\} \\
P &= \{\overrightarrow{A} \to \overleftarrow{A}\,\overrightarrow{B}, \overleftarrow{A} \to \overleftarrow{B}\,\overrightarrow{A}, \\
&\quad \overrightarrow{A} \to \overrightarrow{B}, \overleftarrow{A} \to \overleftarrow{B}, \\
&\quad \overleftarrow{A} \to \overleftarrow{a}, \overrightarrow{A} \to \overrightarrow{a}, \\
&\quad \overleftarrow{B} \to \overleftarrow{b}, \overrightarrow{B} \to \overrightarrow{b}.
\end{aligned}$$

Zum Beispiel ist

$$\begin{aligned} \overrightarrow{A} &\Rightarrow_G \overleftarrow{A}\,\overrightarrow{B} \Rightarrow_G \cdots \\ &\Rightarrow_G \overleftarrow{b}\,\overrightarrow{a}\,\overleftarrow{b}\,\overrightarrow{a}\,\overrightarrow{a}\,\overleftarrow{b}\,\overrightarrow{a}\,\overrightarrow{a} \end{aligned}$$

5.3 Chomsky-Hierarchie

Definition 56. *Es sei $G = (V, \Sigma, P, S)$ eine Grammatik.*

- *G heißt vom Typ 0.*
- *G heißt vom Typ 1 oder kontextsensitiv, falls für jede Regel $w_1 \to w_2$ gilt:*
$$|w_1| \leq |w_2|$$
oder die Regel ist die Regel $S \to \epsilon$.
- *G heißt vom Typ 2 oder kontextfrei, falls G vom Typ 1 und für jede Regel $w_1 \to w_2$ gilt, dass w_1 eine einzelne Variable ist.*
- *G heißt vom Typ 3 oder regulär, falls G vom Typ 2 und für jede Regel $w_1 \to w_2$ gilt, dass $w_2 \in \Sigma \cup \Sigma V$.*

Die Sprache L heißt vom Typ i, falls es eine Grammatik G vom Typ i gibt, so dass $L(G) = L$.

Beispiel 67.
Die Grammatik in Beispiel 65 ist kontextsensitiv.
Die Grammatik in Beispiel 64 ist kontextfrei.
Die Grammatik in Beispiel 63 ist regulär.

Satz 28. *Die Sprachen zu einer regulären Grammatik sind genau die regulären Sprachen.*

Beweis: "⇒" Es sei $G = (V, \Sigma, P, S)$ eine reguläre Grammatik. Dann sei $M = (Z, \Sigma, \delta, S', E)$ folgender NEA:

$$\begin{aligned} Z &= V \cup \{X\}, \quad X \notin V \\ S' &= \{S\} \end{aligned}$$

$$
\begin{array}{rcl}
E & = & \begin{cases} \{S, X\}, & S \to \epsilon \in P \\ \{X\}, & S \to \epsilon \notin P \end{cases} \\
B & \in & \delta(A, a) \quad \text{falls } A \to aB \in P \\
X & \in & \delta(A, a) \quad \text{falls } A \to a \in P.
\end{array}
$$

Man sieht nun:

$$
\begin{array}{rcl}
a_1 a_2 ... a_n & \in & L(G) \\
& \Updownarrow & \\
\text{es gibt Variablen } A_1, ..., A_n \text{ mit:} & & \\
S \Rightarrow a_1 A_1 \Rightarrow a_1 a_2 A_2 \Rightarrow & ... & \Rightarrow a_1 a_2 ... a_{n-1} A_{n-1} \Rightarrow a_1 a_2 ... a_n A_n \\
& \Updownarrow & \\
\text{es gibt Zustände } A_1, ..., A_n \text{ mit:} & & \\
A_1 \in \delta(S, a_1), A_2 \in \delta(A_1, a_2), & ... & X \in \delta(A_{n-1}, a_n) \\
& \Updownarrow & \\
a_1 a_2 ... a_n & \in & L(M).
\end{array}
$$

"$\Leftarrow$" Es sei $M = (Z, \Sigma, \delta, S, E)$ ein DEA. Dann sei $G = (V, \Sigma, P, S)$ folgende reguläre Grammatik:

$$
\begin{array}{rcl}
V & = & Z \\
S & = & z_0 \\
z_0 & \to & \epsilon \quad \text{falls} \quad \epsilon \in L(M) \\
z_1 & \to & a z_2 \quad \text{falls} \quad \delta(z_1, a) = z_2 \\
z_1 & \to & a \quad \text{falls} \quad \delta(z_1, a) = z_2 \wedge z_2 \in E.
\end{array}
$$

Man sieht nun:

$$
\begin{array}{rcl}
a_1 a_2 ... a_n & \in & L(M) \\
& \Updownarrow & \\
\text{es gibt Zustände } z_1, ..., z_n \text{ mit:} & & \\
z_n \in E \quad \text{und} & & i = 1, .., n : \ \delta(z_{i-1}, a_i) = z_i \\
& \Updownarrow & \\
\text{es gibt Variablen } z_1, ..., z_n \text{ mit:} & & \\
z_0 \Rightarrow a_1 z_1 \Rightarrow a_1 a_2 z_2 \Rightarrow & ... & \Rightarrow a_1 a_2 ... a_{n-1} z_{n-1} \Rightarrow a_1 a_2 ... a_n \\
& \Updownarrow & \\
a_1 a_2 ... a_n & \in & L(G).
\end{array}
$$

□

Siehe Beispiel 63.

5.4 Andere Beschreibungen von Grammatiken

Die Beschreibung von Grammatiken wie in Abschnitt 5.2 ist manchmal etwas unübersichtlich. Eine Verbesserung kann durch die Backus-Naur-Form erreicht werden, mit der ALGOL 60 beschrieben wurde.

Anstelle von schreibt man

$$A \to \beta_1, \quad A \to \beta_2 \qquad A \to \beta_1|\beta_2|...|\beta_n$$
$$\vdots$$
$$A \to \beta_n$$

In der erweiterten Backus-Naur-Form schreibt man anstatt gerne

$$A \to \alpha[\beta]\gamma$$

$$A \to \alpha\gamma$$
$$A \to \alpha\beta\gamma$$

Zur Beschreibung von `C` verwendet man folgende Notation:

Anstelle von schreibt man

$$A \to \beta_1, \quad A \to \beta_2 \qquad A:\ \beta_1$$
$$\vdots \qquad\qquad\qquad\qquad\quad ...$$
$$A \to \beta_n \qquad\qquad\qquad\quad \beta_n$$

und anstatt schreibt man

$$A \to \alpha\gamma \qquad A:\ \alpha\ \beta_{opt}\ \gamma$$
$$A \to \alpha\beta\gamma$$

Beispiel 68 (C-Grammatik).
(Siehe Kernighan/Ritchie, Programmieren in C)

iteration-statement:
`while` *(expression) statement*
`do` *statement* `while` *(expression)*
`for` *(expression*$_{opt}$*, expression*$_{opt}$*, expression*$_{opt}$*) statement*
expression:
assignment-expression
expression, assignment-expression
assignment-expression:
conditional-expression
unary-expression assignment-operator assignment-expression
...
AND-expression
equality-expression
AND-expression & equality-expression
equality-expression:
relational-expression
equality-expression `==` *relational-expression*
equality-expression `!=` *relational-expression*

Literatur

[1] C. H. Cap. *Theoretische Grundlagen der Informatik*. Springer 1993.

[2] K. Erk und L. Priese. *Theoretische Informatik*. Springer 2000.

[3] R.Diestel. *Graphentheorie*. Springer 2000.

[4] J.E. Hopcroft, J.D. Ullman. Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie. (oder die englische Orginalversion)

[5] U. Schöning. Theoretische Informatik - kurzgefasst. Spektrum Akademischer Verlag. 2001.