Theoretische Informatik I

Theoretische Informatik I, Wintersemester 2005/2006

Christoph Pflaum

Aussagen

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).

Bezeichnungen

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

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

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.

Folgerungspfeile

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

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:

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

Mengen

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.

Beispiel 4.

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

Prädikate

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.**

Prädikate

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

Quantoren

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." .

Quantoren bezüglich einer Menge

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

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

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

Formel des Beweises durch Induktion

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

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

Mengen

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.

Mengen

- $[a, b] = \{x \in \mathbb{R} | a \leq x \leq b\}$
- $]a, b] = \{x \in \mathbb{R} | a < x \leq b\}$
- $]a, b[= \{x \in \mathbb{R} | a < x < b\}$
- $(a, b) = \{x \in \mathbb{R} | a < x < b\}$
  (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.

Notationen und Operationen für Mengen

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*
- $M \subset N :\Leftrightarrow \forall x \in M \;\; x \in N$ *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 $N \neq \emptyset$.*

Morgansche Komplementenregel

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:*

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

Notationen und Operationen für Mengen

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

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

Kreuzprodukt von Mengen

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.

Was ist eine Menge?

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?

Grundmenge und Element-Zeichen

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.

Beispiele

Beispiel 7. *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

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

Anwendung: Negation von Quantoren

Formel 2. *Es gilt*

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

Funktionen

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

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

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

bijektiv, injektiv, surjektiv

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.*

Komposition von Funktionen

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

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

die Komposition von f *und* g*.*

Lemma 1.

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

Komposition von Funktionen

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

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

so dass

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

und

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

Mächtigkeit einer endlichen Menge

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 $|M| = \infty$.*

Mächtigkeit allgemeiner Mengen

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

- *M heißt mindestens genauso mächtig wie N, falls es eine surjektive Abbildung $f : M \to N$. 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.

Schubfachprinzip

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 $f(a) = f(b)$.

Beispiele

Beispiel 8.

- *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.*

Abzählbare Mengen

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: $\mathbb{N} \times \mathbb{N}$

Beispiel 9. $\mathbb{N} \times \mathbb{N}$ *ist abzählbar.*

Beweis:

Die Abbildung

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

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

ist wohl definiert, da $2, 3$ Primzahlen und die Primfaktorzerlegung eindeutig ist.

$f : \mathbb{N} \mapsto \mathbb{N} \times \mathbb{N}$ ist surjektiv. $\square$

Beispiel: Menge der Wörter einer Sprache

Beispiel 10. *Es sei* $\Sigma = \{a, b, \ldots, \rho\}$ *eine endliche Menge. Dann ist*

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

abzählbar.

Beispiel: Menge der Funktionen $\varphi : \mathbb{N} \longrightarrow \mathbb{N}$

Beispiel 11. *Die Menge M der Funktionen $\varphi : \mathbb{N} \longrightarrow \mathbb{N}$ ist überabzählbar.*

Monoid

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*
  $\forall a \in M \;\; 0_M \circ a = a \circ 0_M = 0_M$.
- $1_M \in M$ *heißt Einselement, falls* $\forall a \in M \;\; 1_M \circ a = a \circ 1_M = a$.
- $(M, \circ)$ *heißt* Halbgruppe, *falls* $\circ$ *assoziativ ist.*
- *Die Halbgruppe* $(M, \circ)$ *heißt* Monoid, *falls es in* M *ein Einselement* 1_M *gibt.*

Gruppe

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*,* $\circ$ *kommutativ.*

Eigenschaft einer Halbgruppe und Gruppe

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

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

Menge der Wörter einer Sprache

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

$$a_1 a_2 ... a_n$$

- $w = a_1 a_2 ... a_n$ heißt ein Wort über Σ.
- $|w| = n$ ist die Länge des Wortes.

Konkatenation von Wörtern

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

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

die folgende Verknüpfung, genannt Konkatenation, von Wörtern definiert

$$\begin{aligned} \circ : \Sigma^+ \times \Sigma^+ &\rightarrow \Sigma^+ \\ (a_1 a_2 ... a_n, b_1 b_2 ... b_m) &\mapsto a_1 a_2 ... a_n b_1 b_2 ... b_m \end{aligned}$$

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.*

Das leere Wort

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*

$$\begin{aligned} \circ : \Sigma^* \times \Sigma^* &\rightarrow \Sigma^* \\ (a_1 a_2 ... a_n, b_1 b_2 ... b_m) &\mapsto a_1 a_2 ... a_n b_1 b_2 ... b_m \\ (\epsilon, b_1 b_2 ... b_m) &\mapsto b_1 b_2 ... b_m \\ (a_1 a_2 ... a_n, \epsilon) &\mapsto a_1 a_2 ... a_n \end{aligned}$$

Eigenschaften des leeren Wortes

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

Beispiel

Beispiel 12. $\Sigma = \{a\}$. *Dann ist*

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

Die Abbildung

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

ist eine Bijektion und hat die Eigenschaft

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

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

Präfix, Infix, Suffix

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* $\exists r \in \Sigma^* \; ru = w$.

Homomorphismus und Isomorphismus

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

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

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.*

Eigenschaft einer Gruppe

Beispiel 13.

- *Die Abbildung*

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

ist ein Isomorphismus.

- *Die Abbildung*

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

ist kein Homomorphismus, da a und b nicht vertauschen.

Eigenschaft einer Gruppe

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

Eigenschaft der freien Halbgruppe

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

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

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

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

ein Epimorphismus.

Sprache

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

Operationen auf Sprachen

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 $L^+ := \bigcup_{i \geq 1} L^i$.*

Relationen

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 14. *Mitarbeiter einer Firma, deren Ausbildungsgrad und Geschlecht.*

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

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

Relationen

Schreibweise:

Sei $R \subset M \times M$ eine binäre Relation.

Dann schreibt man oft aRb anstelle von $(a, b) \in R$.

Beispiel 15. *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:*

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

Relationen

Definition 21. *Eine n-stellige* Relation *über den Mengen $M_1, \cdots, M_n$ ist eine Menge $R \subset \bigtimes_{i=1}^{n} M_i$.*

Beispiel 16. *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.*

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

Eigenschaften von Relationen

Definition 22. *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).*

Eigenschaften von Modulo k

- Modulo Äquivalenzrelation.

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

Weitere Eigenschaften der Relation 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 $\mathbb{Z}_k$: $a + b \equiv \phi(a+b)$ modulo k.

Eigenschaften von Relationen

Definition 23. *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 $[a]_\tau$.

Eigenschaften von Restklassen

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

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

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

$$a \oplus b = c$$

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

Äquivalenz zur Drehgruppe

Φ_0: Drehung um $0°$

Φ_2: Drehung um $180°$

Φ_1: Drehung um $90°$

Φ_3: Drehung um $270°$

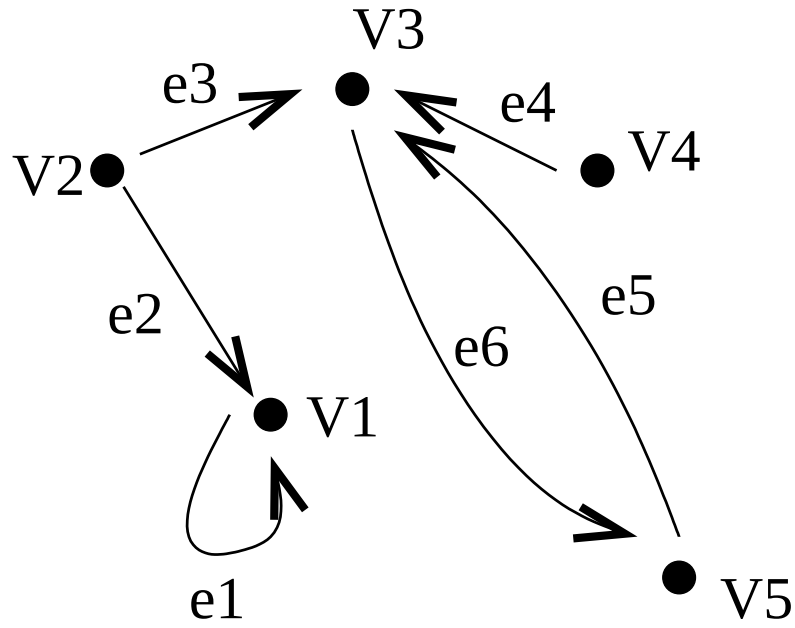
Isomorphismus:

$$\begin{aligned} \varphi : \mathbb{Z}/\equiv \text{ modulo } 4 &\rightarrow \mathcal{D} = \{\Phi_0, \Phi_1, \Phi_2, \Phi_3\} \\ [a]_{\equiv} \text{ modulo } 4 &\mapsto \Phi_a, \quad a = 0, 1, 2, 3. \end{aligned}$$

Graphen

Definition 24. *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 $\forall a, b \in V \left((a, b) \in E \Rightarrow (b, a) \in E\right)$.*
- *Ein nicht ungerichteter Graph heißt gerichteter Graph.*



Graphen

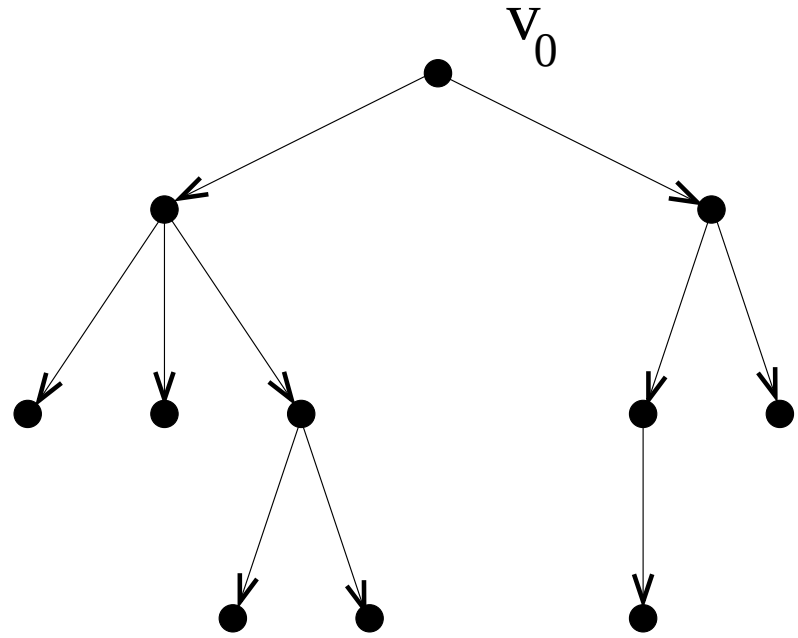
Definition 25. *Es sei (V, E) ein Graph.*

- *Falls $v, v' \in V$, dann schreibt man dafür $v \longrightarrow v'$. v heißt Vater von v' und v' Sohn von v.*
- *Ein nicht-leeres Wort $v_1 v_2 \ldots v_n$ heißt Weg (der Länge $n-1$) von v_1 nach v_n falls $\forall 1 \leq i < n \; v_i \longrightarrow v_{i+1}$.*
- *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 $v_1 v_2 \ldots v_n v_1$ heißt Kreis.*

Bäume

Definition 26. *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.*
- v_0 *hat keinen Vater.*



Satz 5.
Ein Baum enthält keinen Kreis.

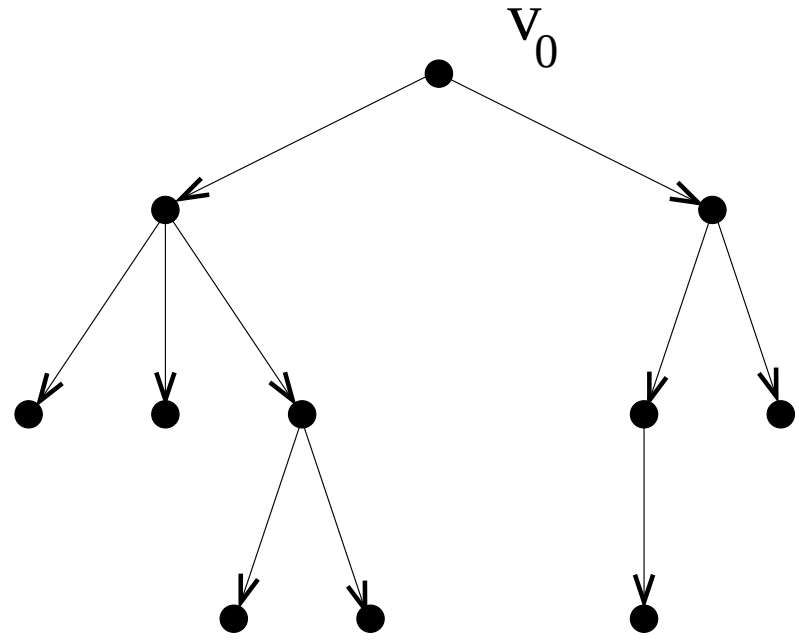
Ast, Blatt, ...

Definition 27. *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.*

Baum der Tiefe 3 mit

- 6 Blättern
- 5 inneren Knoten



Aussagenlogik

Definition 28. *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.*

Benennungen in Aussagenlogik

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

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:

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

Modell in der Aussagenlogik

Definition 29.

- *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.*

Auswertung von Formeln

Syntaxbaum der Formel $(A \Rightarrow B) \vee (A \wedge B)$:

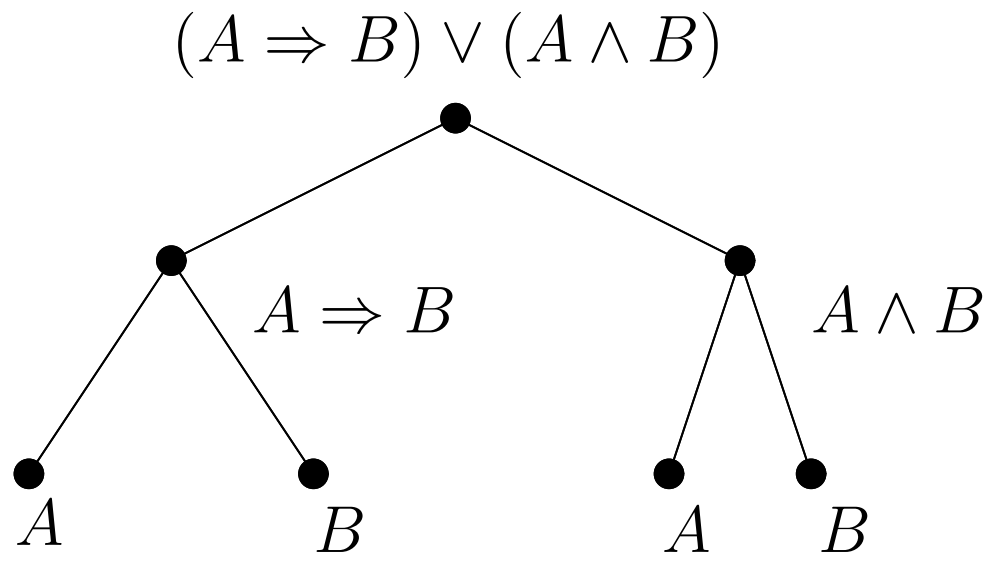


Tabelle die zu jedem Knoten eine Spalte enthält:

A	B	$A \Rightarrow B$	$A \wedge B$	$(A \Rightarrow B) \vee (A \wedge B)$
`true`	`true`	`true`	`true`	`true`
`true`	`false`	`false`	`false`	`false`
`false`	`true`	`true`	`false`	`true`
`false`	`false`	`true`	`false`	`true`

Entscheidbarkeit

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

Äquivalenz von Formeln

Definition 30. *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 17.

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:	$F \vee (G \wedge H)$	$\hat{=}$	$(F \vee G) \wedge (F \vee H)$

Allgemeine Formel von de Morgan

Satz 8.

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

KNF und DNF

Definition 31.

- *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.

Äquivalenz zu KNF und DNF

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

eispiele zur Motivation der Graphentheor

- 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?

Ein lineares 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!

Reduktion eines Gleichungssystems

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

Da A invertierbar, sind auch A_{oo} und A_{uu} invertierbar.

Wir schreiben nun x und b in der Form:

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

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:

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

Reduzierbare Matrizen

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

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

Graph einer Matrix

Definition 33. *Es sei $A = (a_{ij})_{1 \leq i,j \leq n}$ 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:*

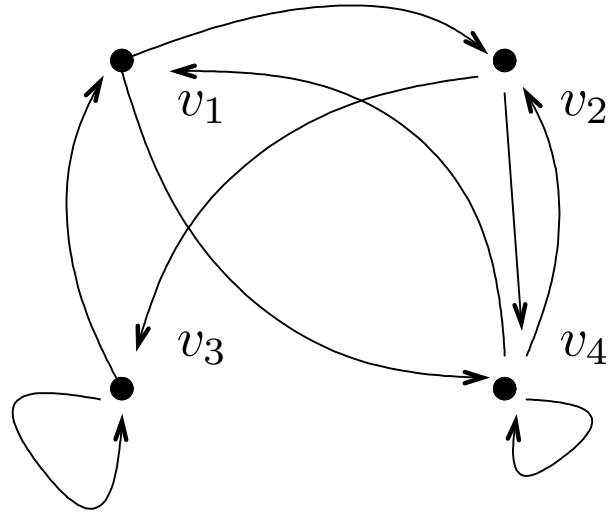
$$i \to j :\Leftrightarrow a_{i,j} \neq 0.$$

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

Beispiel 18. *Der Graph zu*

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

ist:



Graph einer Matrix

Definition 34. *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.*

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

Einfache Graphen

Definition 35.

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.

Teilgraph eines Graphen

Definition 36.

- *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 $(a, b) \in E$.*

Grad eines Knoten

Definition 37.

- *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*

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

Grad eines Graphen

Definition 38.

- *Der Minimalgrad von G ist definiert als*

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

- *Der Maximalgrad von G ist definiert als*

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

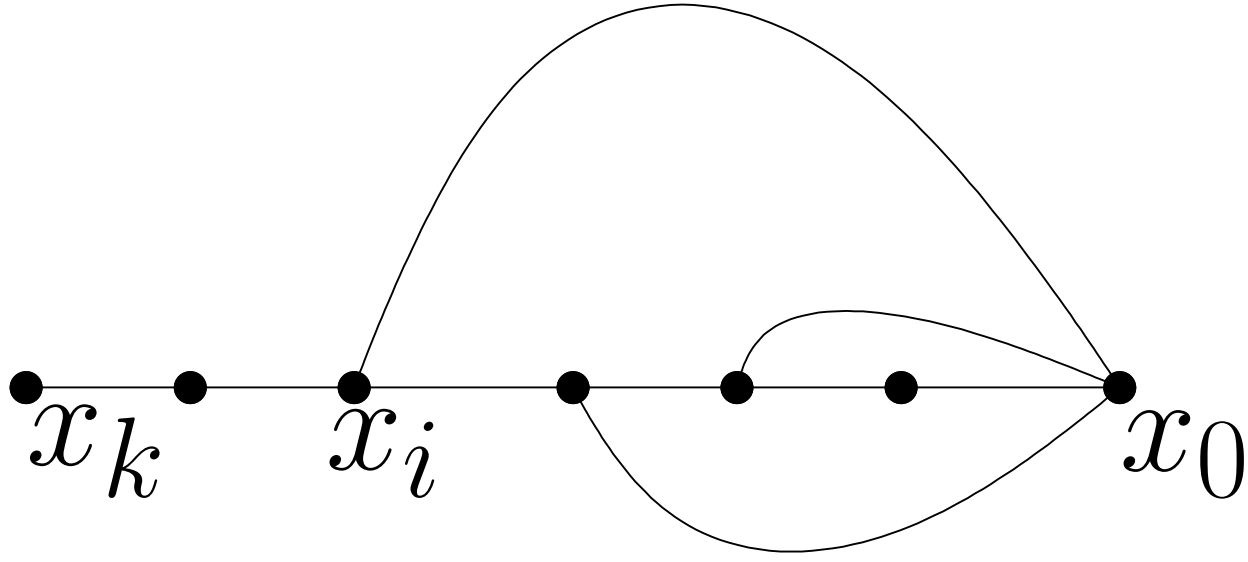
- *Der Durchschnittsgrad von G ist definiert als*

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

Existenz von Wegen und Kreisen

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

Zum Beweis: Längster Weg in G:



Zusammenhang eines Graphen

Definition 39. *Ein gerichteter Graph heißt* stark zusammenhängend*, falls für jedes Paar unterschiedlicher Knoten v, w ein gerichteter Weg existiert. Dies bedeutet, dass ein Weg $v_0 v_1 \ldots v_{r-1} v_r$ mit $v_0 = v$ und $v_r = w$ existiert.*

Definition 40. *Ein einfacher Graph heißt* zusammenhängend*, falls für jedes Paar unterschiedlicher Knoten v, w ein Weg existiert. Dies bedeutet, dass ein Weg $v_{i_0} v_{i_1} \ldots v_{i_{r-1}} v_{i_r}$ mit $v_{i_0} = v$ und $v_{i_r} = w$ existiert.*

Topologie eines Netzwerkes

Beispiel 19. *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.*

Ebene Graphen

Definition 41. *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

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

Zusammenhängende Gebiete in der Ebene

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

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

gibt, die die Eigenschaft

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

hat.

Flächen eines ebenen Graphen

Definition 42. *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.

Bäume

Definition 43. *Ein einfacher Baum ist ein zusammenhängender einfacher Graph (V, E) der keine Kreise enthält.*

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

Eulerscher Polyeder Satz für die Ebene

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

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

Eulersche Polyeder Formel für die Ebene

Satz 13. *Es sei G ein zusammenhängender ebener Graph mit e Ecken (Knoten), k Kanten und f Flächen. Dann gilt:*

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

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 k_r Kanten am Rand des unbeschränkten Gebietes besitzt, dann enthält G genau $3e - k_r - 3$ Kanten.*

Färben einer Landkarte

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.

Färben der Knoten eines Graphen

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

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

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

Färben eines ebenen Graphen

Satz 14. *Jeder ebene Graph ist 4-färbbar.*

Satz 15. *Jeder ebene Graph ist 5-färbbar.*

Satz 16. *Jeder ebene Graph ist 6-färbbar.*

Beweis des 6-Farben-Satzes

Satz 17. *Jeder ebene Graph ist* 6*-färbbar.*

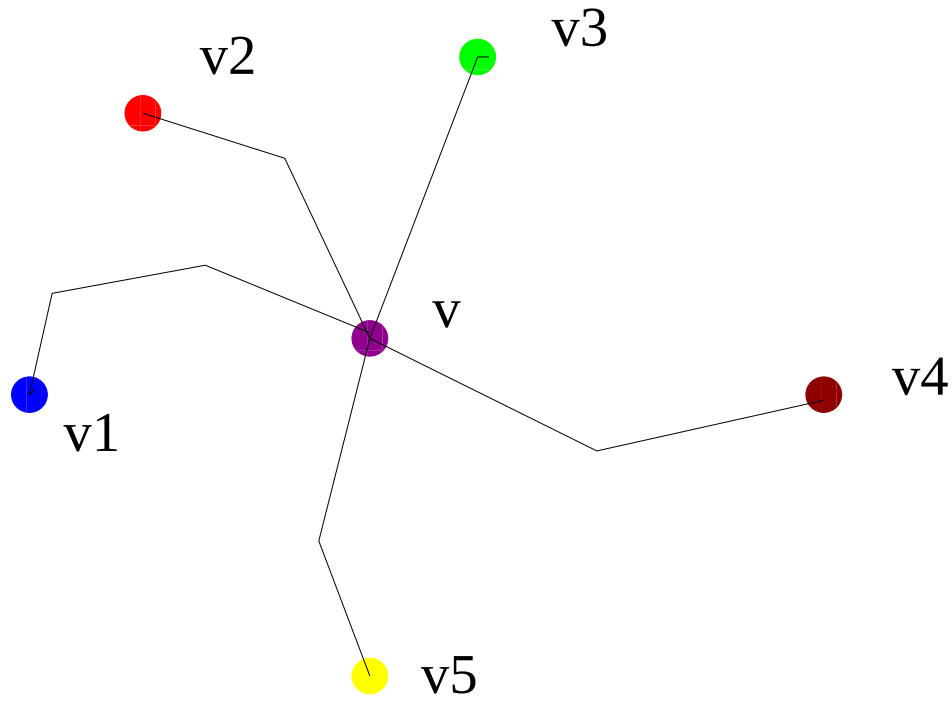


Abbildung 1: Beweis von Satz **??**.

Beweis des 5-Farben-Satzes

Satz 18. *Jeder ebene Graph ist* 5*-färbbar.*

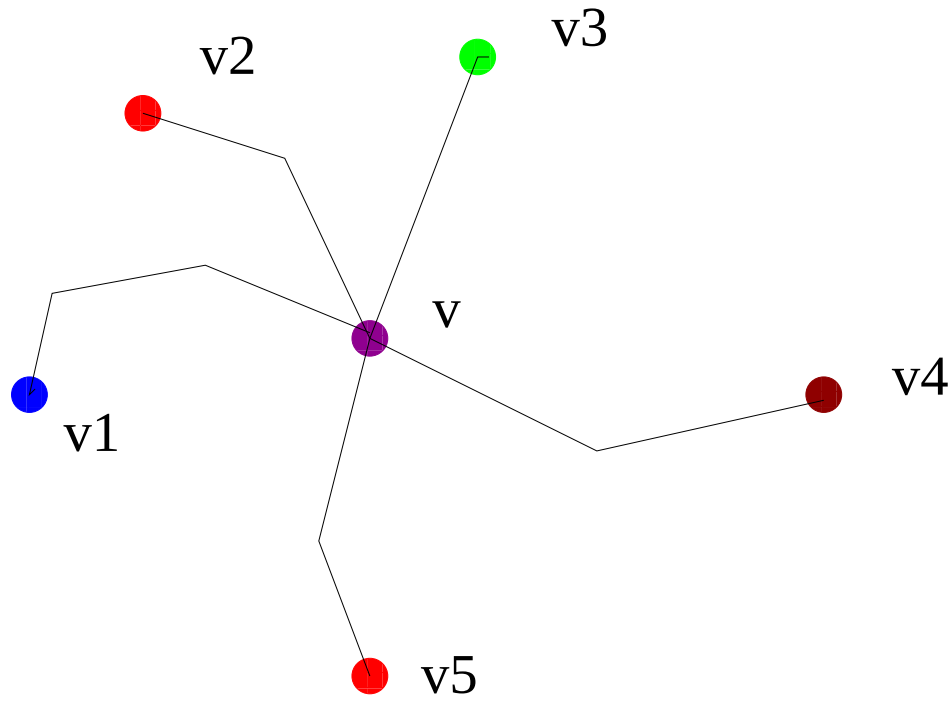


Abbildung 2: Beweis von Satz **??**: einfacher Fall.

Beweis des 5-Farben-Satzes

Satz 19. *Jeder ebene Graph ist* 5*-färbbar.*

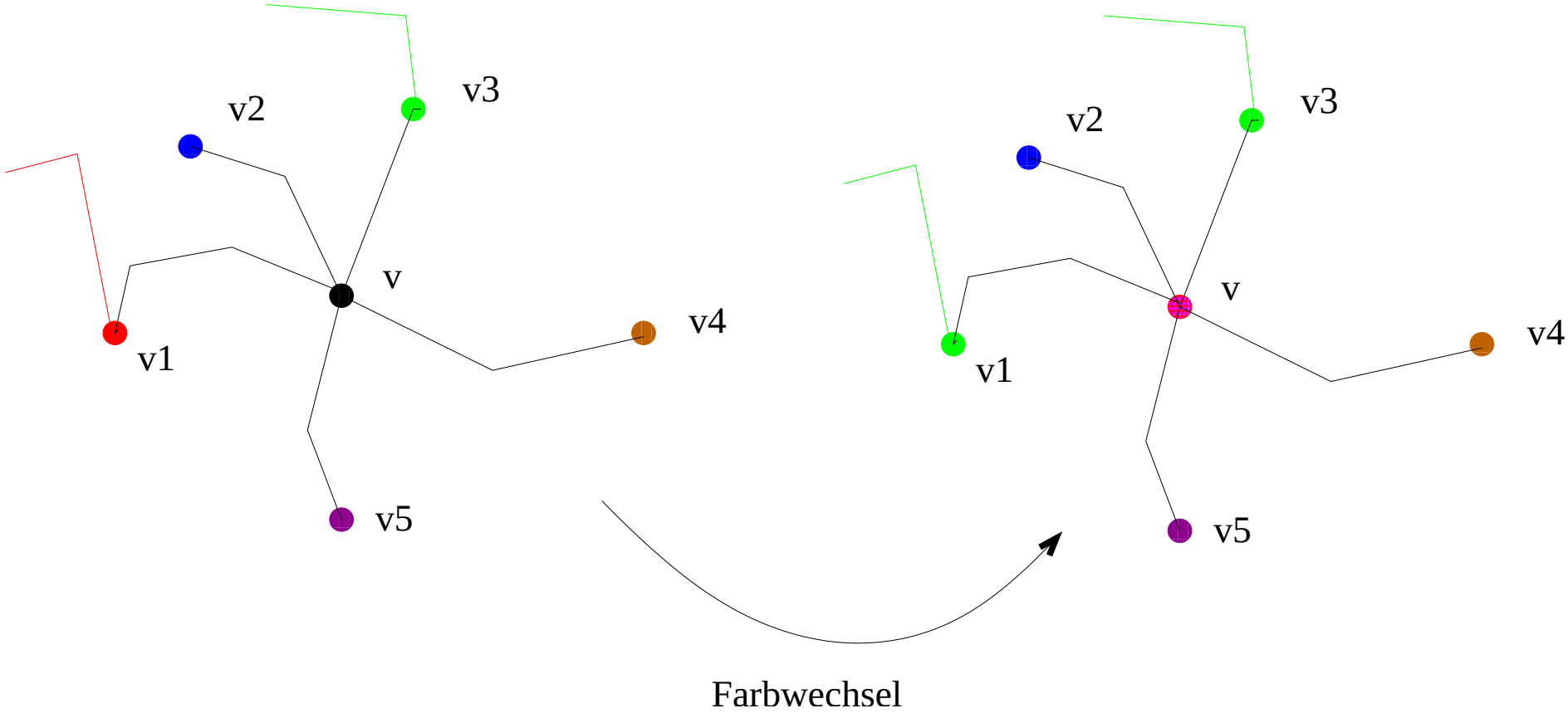


Abbildung 3: Beweis von Satz **??**: 1.Fall.

Beweis des 5-Farben-Satzes

Satz 20. *Jeder ebene Graph ist* 5*-färbbar.*

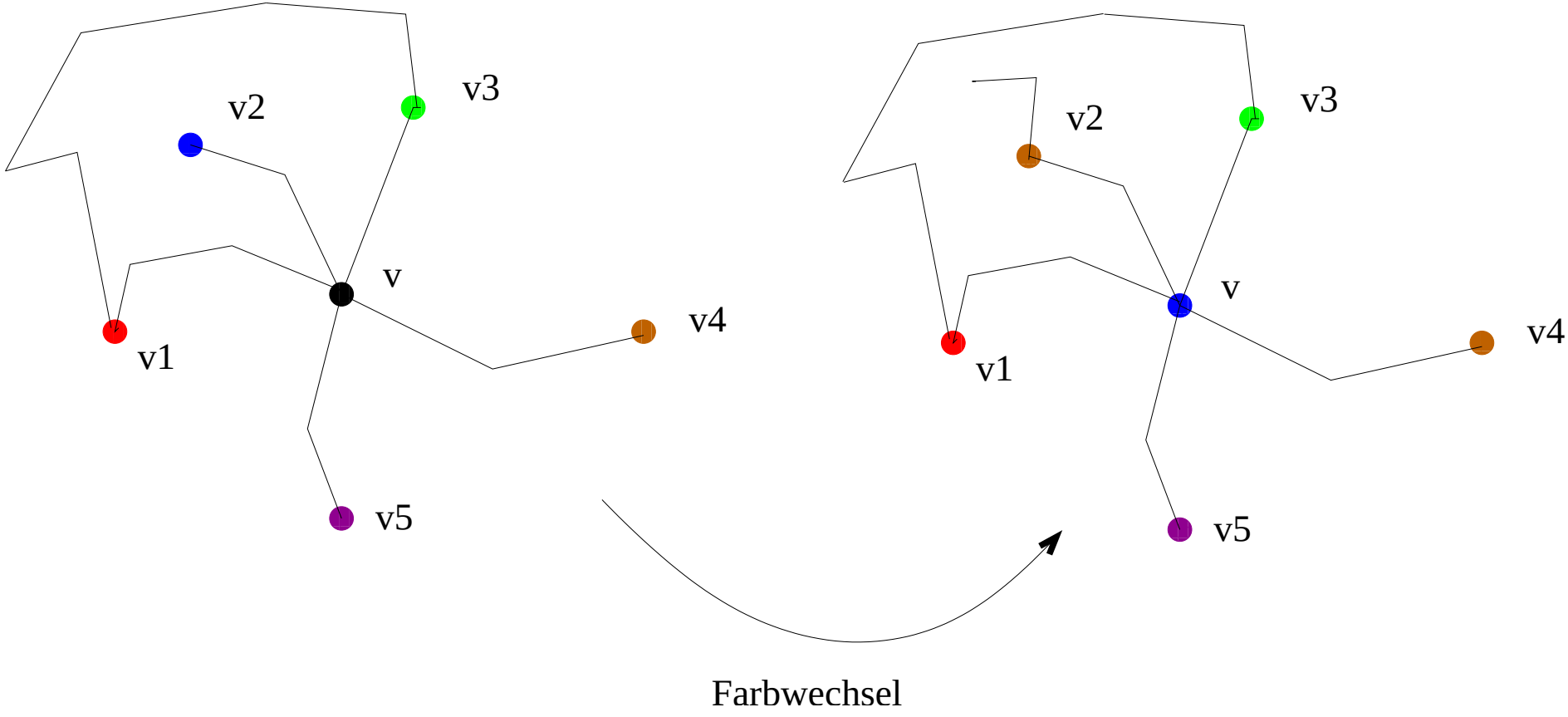


Abbildung 4: Beweis von Satz **??**: 2.Fall.

Allgemeiner Färbungssatz

Satz 21. *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

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

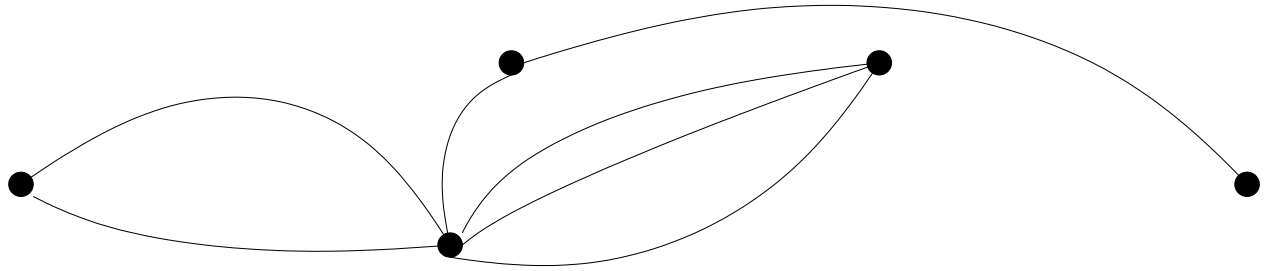
□

Multigraphen

Definition 44. *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:

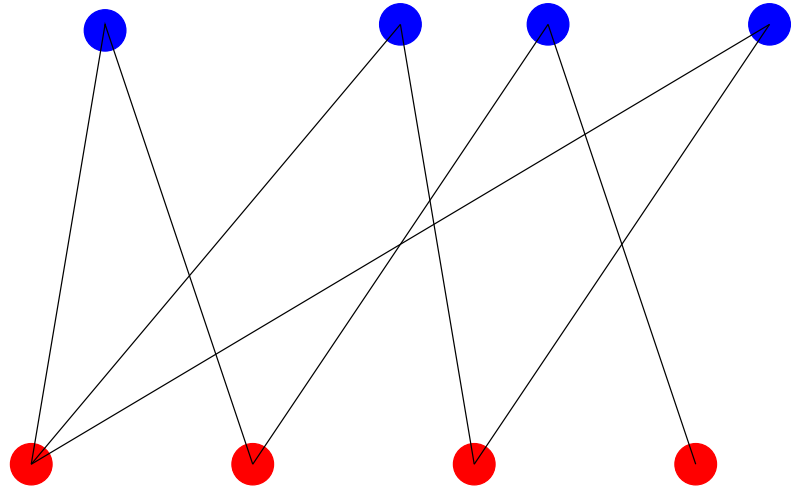
$$E' = \{(v, w) \in V \times V \mid \exists e \in E \;\; c(e) = \{v, w\}\}.$$

Bipartite Graphen

Definition 45. *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.



Satz 22. *Ein bipartiter Graph enthält keinen Kreis ungerader Länge.*

Kantenfärbung eines Multigraphen

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

$$f : E \to F_n$$

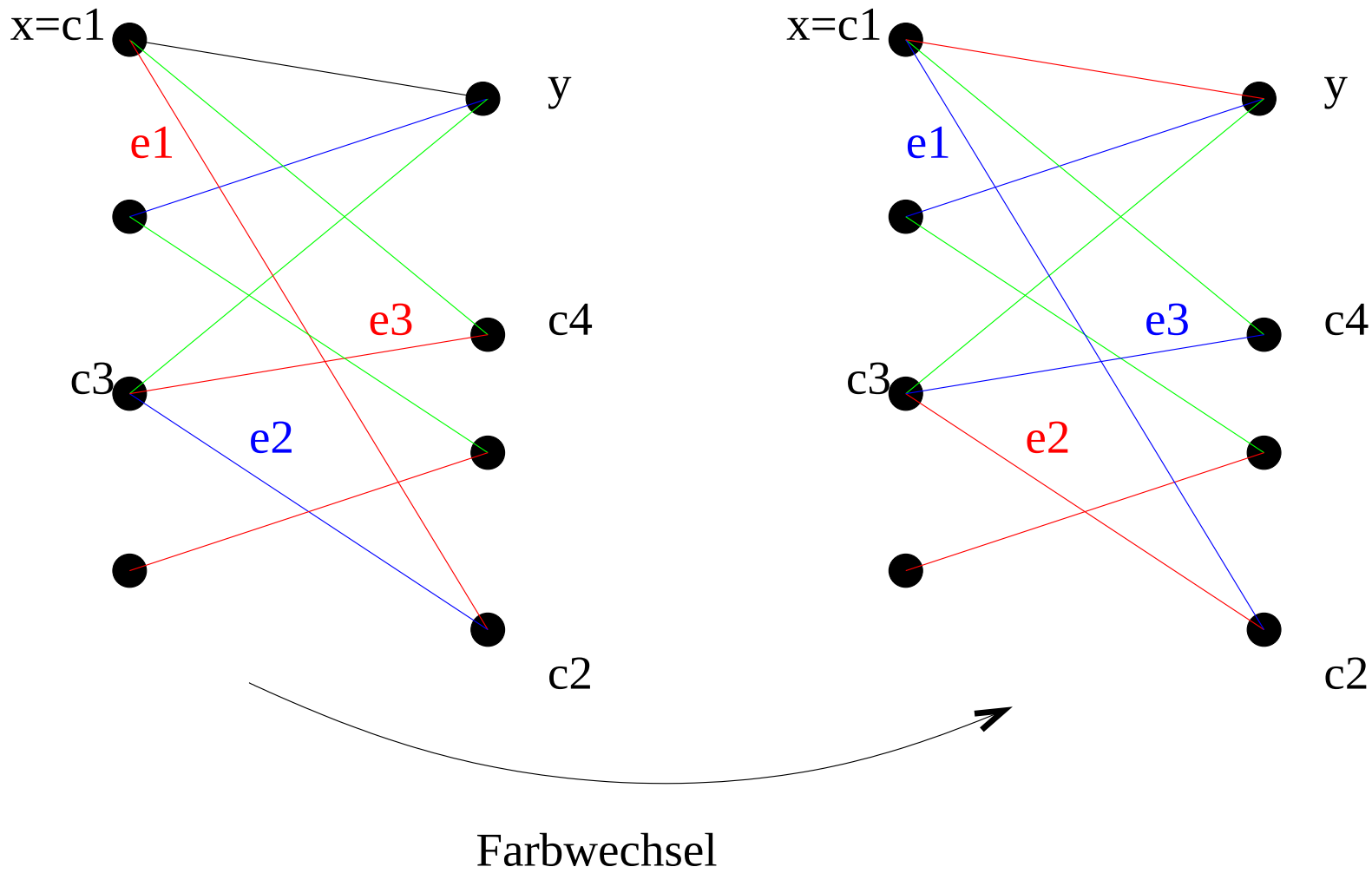
so dass

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

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

Satz zur Kantenfärbung

Satz 23. *Für jeden bipartiten Multigraphen G sind die Kanten $\Delta(G)$-färbbar.*



Endliche Automaten

Einführung: Der Kaugummi-Automat

- 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.

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 $i > 0$.
- $z_i \overset{\text{Ein,Aus}}{\longrightarrow} z_j$ Übergang von einem Zustand z_i zu einem Zustand z_j nach einer bestimmten Eingabe Ein und mit einer bestimmten Ausgabe Aus.

Formale Beschreibung des K.-Automaten

- Menge von Zuständen

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

- Eingabemenge

$$\Sigma = \{1, 2\} \hat{=} \{\text{10 cent}, \text{20 cent}\}.$$

- Ausgabemenge

$$\Delta = \{K, R, -\} \hat{=} \{\text{Kaugummi}, \text{Geldrückgabe}, \text{keine Ausgabe}\}.$$

- Überführungsfunktion $\delta : Z \times \Sigma \rightarrow Z$
- Ausgabefunktion $\lambda : Z \times \Sigma \rightarrow \Delta$

Der Mealy-Automat

Definition 46. *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.*

Eingabe des Mealy-Automaten

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

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

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

Ausgabe des Mealy-Automaten

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 $|w|$.

augummi-Automat mit Ausgabe an Knote

- 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.

Beschreibe den Kaugummi-Automaten durch einen gerichteten Graphen:

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

Formale Beschreibung des K.-Automaten

- Menge von Zuständen

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

- Eingabemenge

$$\Sigma = \{1, 2\} \hat{=} \{\text{10 cent}, \text{20 cent}\}.$$

- Ausgabemenge

$$\Delta = \{K, R, -\} \hat{=} \{\text{Kaugummi}, \text{Geldrückgabe}, \text{keine Ausgabe}\}.$$

- Überführungsfunktion $\delta : Z \times \Sigma \rightarrow Z$
- Ausgabefunktion $\lambda : Z \rightarrow \Delta$

Der Moore-Automat

Definition 47. *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.*

Eingabe und Ausgabe beim Moore-Autom

Die allgemeine Überführungsfunktion sei:

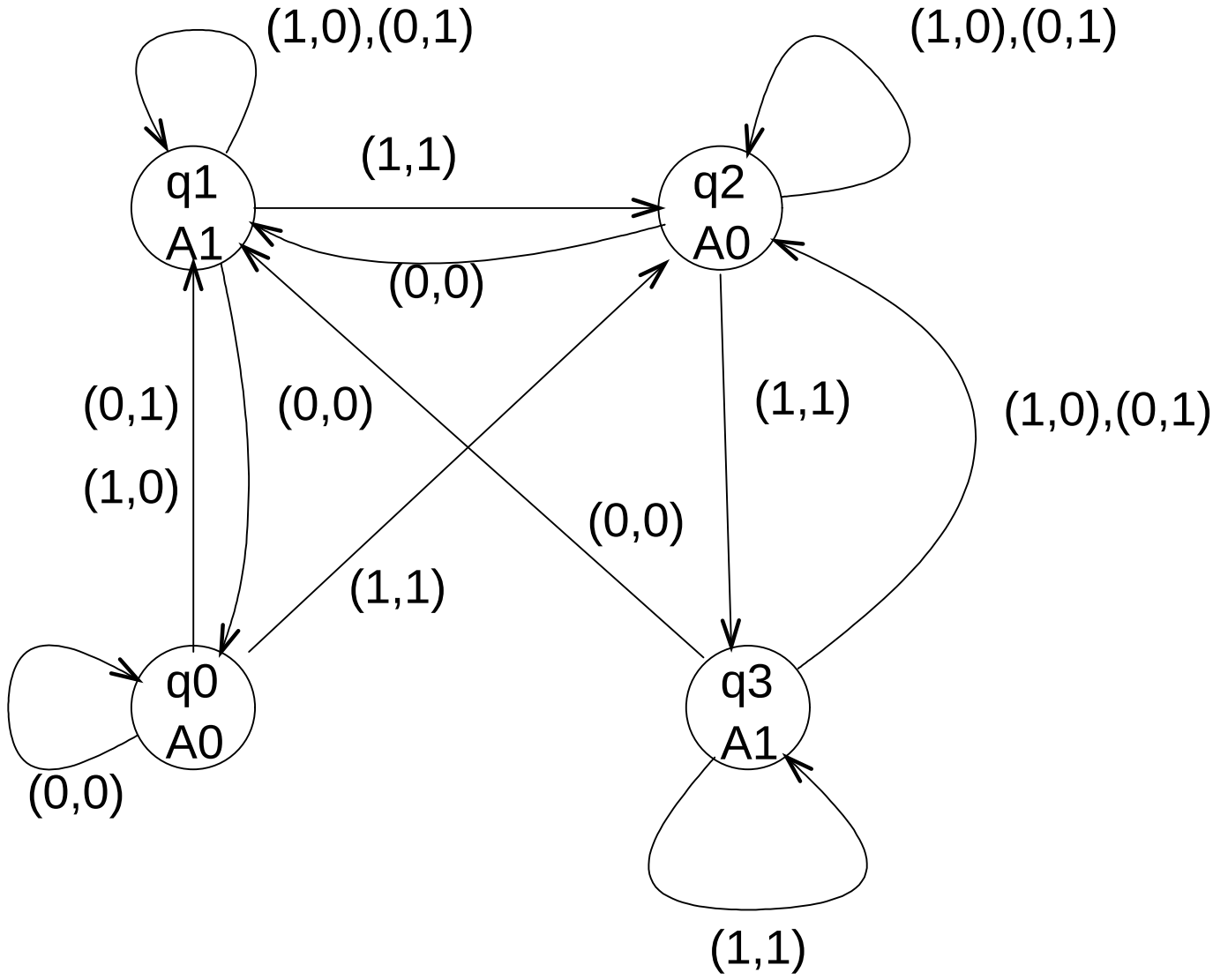
$$\begin{aligned}
\hat{\delta} : Z \times \Sigma^* &\rightarrow Z \\
\hat{\delta}(z, \epsilon) &= z \quad \text{für alle } z \in Z, \\
\hat{\delta}(z, wa) &= \delta(\hat{\delta}(z, w), a) \quad \text{für alle } a \in \Sigma, w \in \Sigma^*, z \in Z.
\end{aligned}$$

Man schreibt kurz δ anstatt $\hat{\delta}$.
Die allgemeine Ausgabefunktion sei

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

Addition mit einem Moore-Autom.

Beispiel 20. *Es sei* $\Sigma = \{0,1\} \times \{0,1\}$ *und* $\Delta = \{0,1\}$.



Äquivalenz des Moore- und Mealy-Auto.

Satz 24. *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.*

Der endliche Automat

Ein endlicher Automat ist ein Moore-Automat mit Ausgabemenge $\Delta = \{J, N\}$.
Formal definiert man eine endlichen Automaten wie folgt:

Definition 48 (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,*
- $z_0 \in Z$ *der Startzustand und*
- $F \subset Z$ *eine Menge von Endzuständen ist.*

Die Mengen Z, Σ *sollten disjunkt sein.*

Allgemeine Überführungsfunktion

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 $\hat{\delta}$.

Akzeptierende Sprache

Definition 49. *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 50. *Eine Sprache $L \subset \Sigma^*$ heißt regulär, falls es einen endlichen Automaten gibt, der L akzeptiert.*

Ein weiteres Beispiel

Beispiel 21. *Es sei* $\Sigma = \{a, b, a^{-1}, b^{-1}\}$*. Dann sei*

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

Diagonalsprache des Moore-Automaten

Satz 25. *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.

Das Pumping Lemma

Satz 26. *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

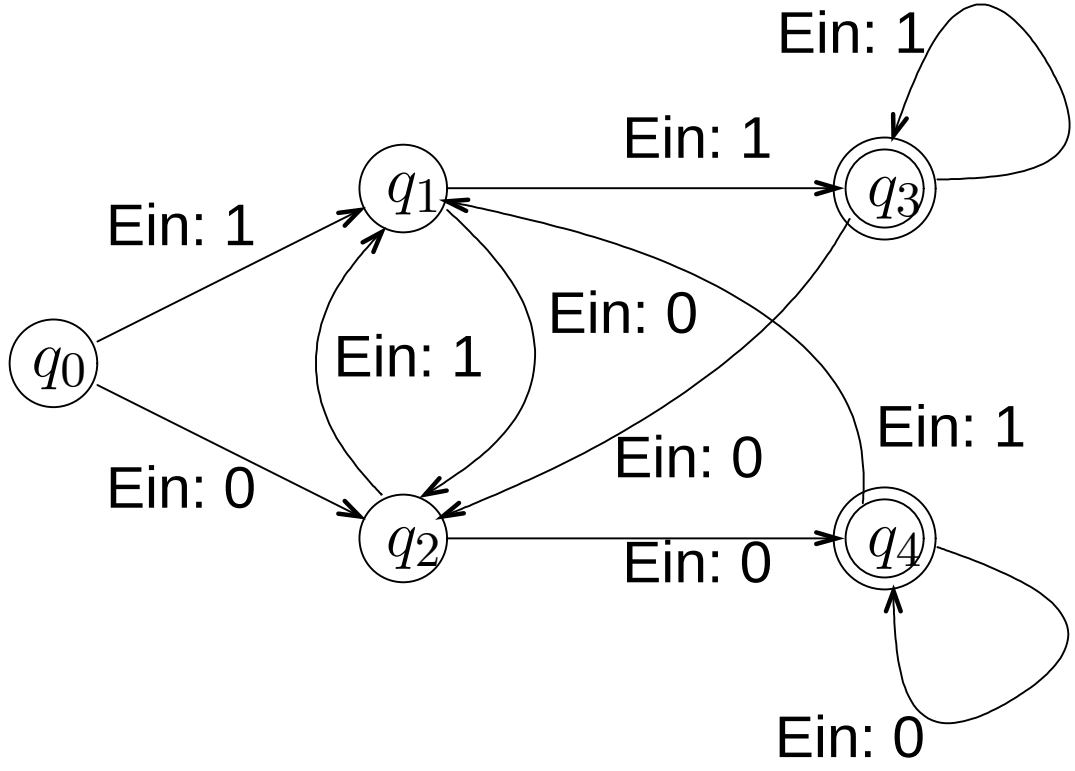
- $x = uvw$,
- $|v| \geq 1$,
- $|uv| \leq n$ *und*
- *für alle $i = 0, 1, 2, \ldots$ gilt $uv^i w \in L$.*

Beispiel Pumping Lemma 1

Beispiel 22. *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:



Beispiel Pumping Lemma 2

Beispiel 23. *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.

Beispiel Pumping Lemma 3

Beispiel 24. *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.

Beispiel Pumping Lemma 4

Satz 27. *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*

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

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*

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

$$y = x^2.$$

Berechnung endlich vieler Quadratzahlen

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:

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

Der nichtdeterministische Automat

Definition 51 (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.*

Allgemeine Überführungsfunktion

Die allgemeine Überführungsfunktion des nichtdeterministischen Automaten M sei:

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

$$w \in \Sigma^*, Z' \subset Z.$$

Man schreibt kurz δ anstatt $\hat{\delta}$.
Die von M akzeptierte Sprache ist

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

Äquivalenz zwischen DEA und NEA

Satz 28. *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*

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

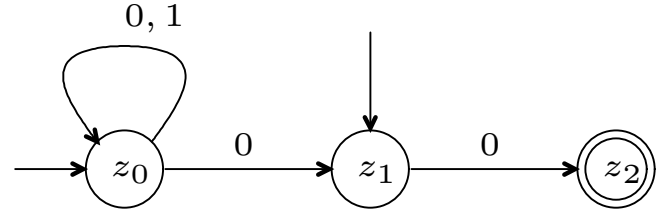
b) Es sei $L(M')$ die Sprache die von dem deterministischen Automat M akzeptiert wird. Dann gibt es einen nichtdeterministischen Automaten M, der die gleiche Sprache akzeptiert. Das heißt

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

Beispiel zur Transformation NEA $\longrightarrow$ DEA

NEA; $M = (Z, \Sigma, \delta, S, E)$

$$
\begin{array}{llll}
\delta(z_0, 0) & = & \{z_0, z_1\} & \delta(z_1, 1) = \emptyset \\
\delta(z_0, 1) & = & \{z_0\} & \delta(z_2, 0) = \emptyset \\
\delta(z_1, 0) & = & \{z_2\} & \delta(z_2, 1) = \emptyset.
\end{array}
$$



DEA: $M' = (\mathcal{Z}, \Sigma, \delta', z_0', E')$

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

Index einer Relation

Definition 52. *Es sei $\tau \in R \times R$ eine Äquivalenzrelation. Die Anzahl der Äquivalenzklassen von τ heißt der Index von τ.*

Beispiel 25.

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

Rechts-invariante Relation

Definition 53. *Eine binäre Relation $\tau \in H \times H$ auf einer Halbgruppe H heißt rechts-invariant, falls*

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

Beispiel 26.

- $=$ $\in \mathbb{R} \times \mathbb{R}$ *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)*

Ä.-relation R_M eines endlichen Automaten

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:*

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

Eine Äquivalenzklasse ist dann $[z]_{R_M} := \{w \in \Sigma^* \mid \delta(z_0, w) = z\}$.

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.

Äquivalenzrelation R_L zu einer Sprache L

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.

Endlicher Automat zu R_L

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

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

Hierbei ist $[z]$ die Äquivalenzklasse bezüglich R_L.

Bemerkung: Es gilt:

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

Satz von Myhill-Nerode

Satz 29 (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.*

Minimalautomat

Satz 30 (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*

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

.

Anwendung des Satzes von Myhill Nerode

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

Satz 31 (Satz von Myhill-Nerode)**.** *Eine Sprache $L \subset \Sigma^*$ ist genau dann regulär, wenn der Index von R_L endlich ist.*

Beispiel 27.

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

ist nicht regulär.

Äquivalenzrelation $\equiv$ auf M

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:

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

Der Minimalautomat zu L ist nun:

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

Konstruktion von Minimalautomaten

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

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.

Suchen regulärer Ausdrücke mit `egrep`

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`

Reguläre Ausdrücke in der Theo. Info.

Definition 54. *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 $(\alpha)*$.*

Sprache zu einem regulären Ausdruck

Definition 55. *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* $\alpha\beta$.
- $L((\alpha)^*) = L(\alpha)^*$.

Eigenschaften regulärer Ausdrücke

Lemma 6.

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

Satz von Kleene

Satz 32 (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.

Beweis des Satzes von Kleene

"$\Rightarrow$": $\underline{l-1 \to l}$:

FALL 1: $\gamma = \alpha\beta$. Seien M_1, M_2 NEA's, die $L(\alpha)$ und $L(\beta)$ akzeptieren.
Wir konstruieren folgenden Automaten M in "Serie" :

- 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.

Beweis des Satzes von Kleene

"$\Rightarrow$": 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 $L(\gamma) = L(\alpha|\beta)$.

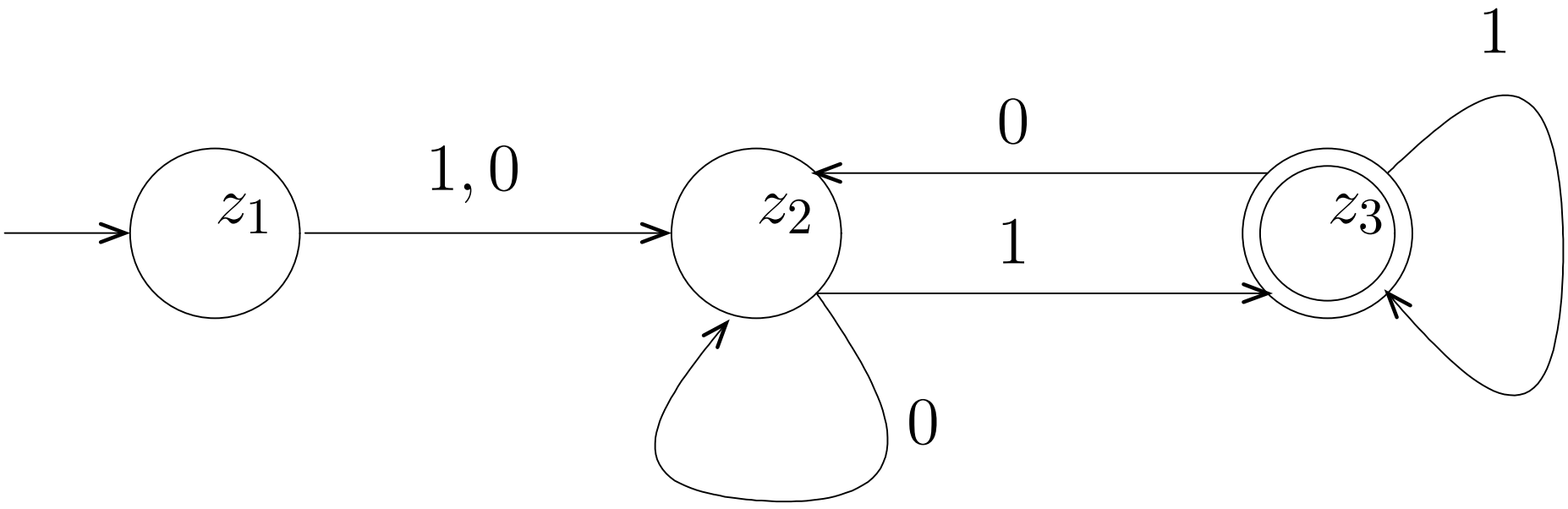
Beweis des Satzes von Kleene

"$\Rightarrow$": FALL 3: $\gamma = (\alpha)^*$. Sei M_1 ein NEA, der $L(\alpha)$ akzeptiert. Wir konstruieren folgenden Automaten M:

- M hat die Zustände von M_1 und einen weiteren Zustand z_e der zum 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.

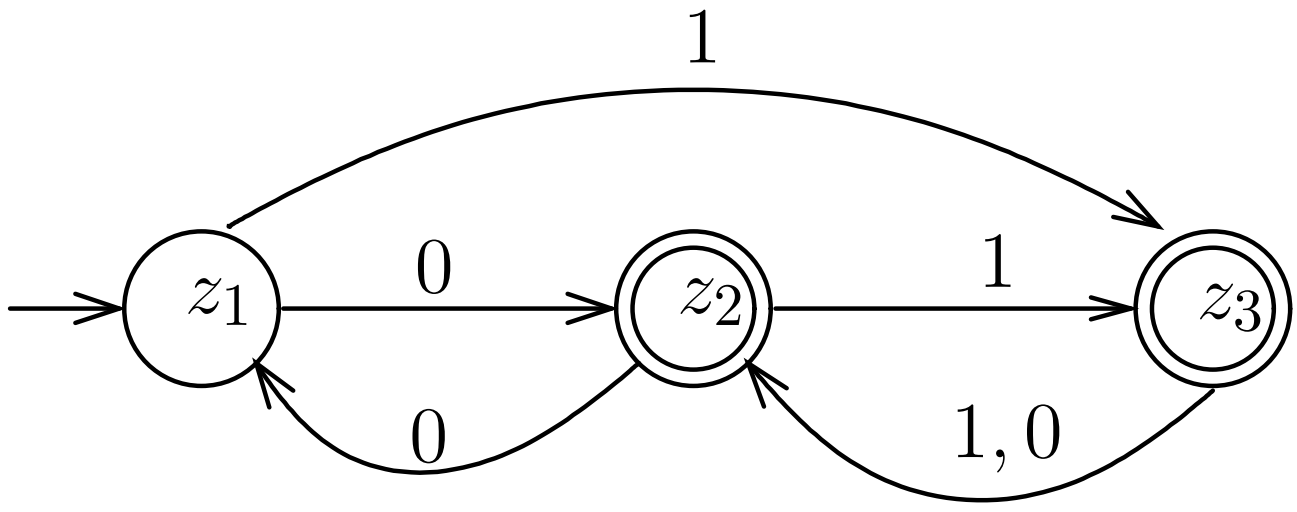
Man sieht dass M die Sprache $L(\gamma) = L((\alpha)^*)$ akzeptiert.

Beispiel 1



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

Beispiel 2



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	0	1	0	ϵ	1	$\emptyset$	$0\|1$	ϵ
$k=1$	ϵ	0	1	0	$\epsilon\|00$	$1\|01$	$\emptyset$	$0\|1$	ϵ
$k=2$	$(00)^*$	$0(00)^*$	0^*1	$0(00)^*$	$(00)^*$	0^*1	$(0\|1)(00)^*0$	$(0\|1)00^*$	$\epsilon\|(0\|1)0^*1$

$$
\begin{aligned}
R^3_{12} &= L(0^*1((0|1)0^*1)^*(0|1)(00)^*|0(00)^*) \\
R^3_{13} &= L(0^*1((0|1)0^*1)^*) \\
R^3_{12} \cup R^3_{13} &= L(0^*1((0|1)0^*1)^*(\epsilon|(0|1)(00)^*)|0(00)^*
\end{aligned}
$$

Abschlusseigenschaften

Satz 33. *Die regulären Sprachen sind abgeschlossen unter Vereinigung, Produkt, Stern, Komplement und Schnitt.*

Grammatikalisch richtig oder falsch

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)

Beispiel einer Grammatik

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

Grammatik

Definition 56. *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.*

Sprache einer Grammatik

Definition 57. *Sei $G = (V, \Sigma, P, S)$ eine Grammatik. Die Ableitungsrelation $\Rightarrow_G$ auf $(V \cup \Sigma)^*$ ist definiert durch*

$u \Rightarrow_G v$ *genau dann wenn es*
$x, y, y', z \in (V \cup \Sigma)^*$ *gibt, so dass*
$y \to y' \qquad u = xyz \qquad v = xy'z.$

Des weiteren schreiben wir

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

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

Chomsky-Hierarchie

Definition 58. *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 $w_1 \longrightarrow w_2$ gilt:*
$$|w_1| \leq |w_2|.$$
- *G heißt vom Typ 2 oder kontextfrei, falls G vom Typ 1 und für jede Regel $w_1 \longrightarrow w_2$ gilt, dass w_1 eine einzelne Variable ist.*
- *G heißt vom Typ 3 oder regulär, falls G vom Typ 2 und für jede Regel $w_1 \longrightarrow w_2$ 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 $L(G) = L$.

Grammatik

Satz 34. *Die Sprachen zu einer regulären Grammatik sind genau die regulären Sprachen.*

Backus-Naur-Form

Anstelle von

$$A \to \beta_1, \qquad A \to \beta_2$$
$$\vdots$$
$$A \to \beta_n$$

schreibt man

$$A \to \beta_1 | \beta_2 | ... | \beta_n$$

In der erweiterten Backus-Naur-Form schreibt man anstatt

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

gerne

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

Beschreibung von C

Anstelle von

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

schreibt man

A :

β_1

...

β_n

und anstatt

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

schreibt man

A :

$\alpha\ \beta_{opt}\ \gamma$

Beispiel: C-Grammatik

Beispiel 28 (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

...

equality-expression:

relational-expression

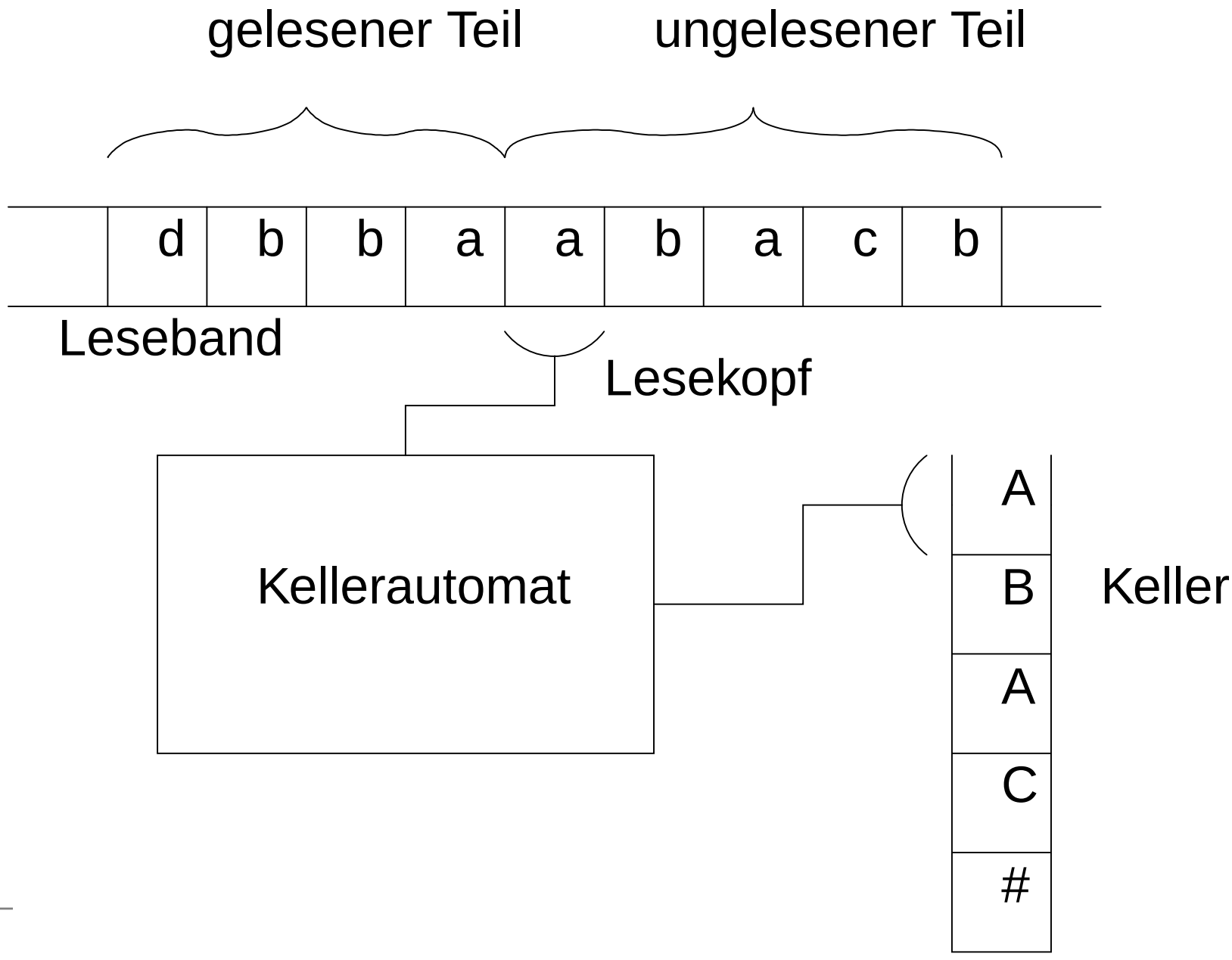
equality-expression `==` *relational-expression*

equality-expression `!=` *relational-expression*

Übersicht

- Die regulären Grammatiken beschreiben die Sprachen, die die endlichen (deterministischen) Automaten akzeptieren.
  Beispiel: $\{a^n \mid n \in \mathbb{N}\}$.
- Die kontext-freien Grammatiken beschreiben die Sprachen, die die nicht deterministischen Kellerautomaten akzeptieren.
  Beispiel: $\{a^n b^n \mid n \in \mathbb{N}\}$.
- Die kontext-sensitiven Grammatiken beschreiben die Sprachen, die die linear beschränkten nicht deterministischen Turing-Maschinen akzeptieren.
  Beispiel: $\{a^n b^n c^n \mid n \in \mathbb{N}\}$.
- Die Typ 0-Grammatiken beschreiben die Sprachen, die die allgemeinen Turing-Maschinen akzeptieren.

Kellerautomat



Deterministischer Kellerautomat

Definition 59. *Ein deterministischer Kellerautomat ist ein Sechstupel* $KA = (Z, \Sigma, \Gamma, \delta, z_0, \#)$*, wobei*

- Z *eine endliche Menge von Zuständen,*
- Σ *eine endliche Eingabemenge,*
- Γ *eine endliche Menge von "Kellerbuchstaben",*
- $\delta : Z \times \Sigma \times \Gamma \rightarrow Z \times \Gamma^*$ *eine Überführungsfunktion,*
- $z_0 \in \Sigma$ *der Startzustand und*
- $\# \in \Gamma$ *das unterste Kellerzeichen.*

$\longrightarrow$ *wichtiger ist der nicht deterministische Kellerautomat!*

$\longrightarrow$ *Ein deterministischer Kellerautomat wird teilweise anders definiert!*

Konfiguration eines Kellerautomaten

Definition 60. $k = (z, w, W) \in Z \times \Sigma^* \times \Gamma^*$ *heißt Konfiguration des Kellerautomaten. Dies bedeutet:*

- *KA ist im Zustand* z*.*
- *KA muss noch das Wort* $w = a_1 ... a_n$ *lesen.*
- *Im Keller des KA ist das Wort* $W = A_1 ... A_m$*.*

Zum Deterministischen Kellerautomaten

Ein deterministischer Kellerautomat KA habe die Konfiguration $(z, w, W) = (z, a_1...a_n, A_1...A_m)$.
Dann geht KA in die Konfiguration

$$(z', a_2...a_n, b_1...b_s A_2...A_m)$$

über, wobei $\delta(z, a_1, A_1) = (z', b_1...b_s) \in Z \times \Gamma^*$.

Ein Wort w wird akzeptiert, wenn KA von der Konfiguration $(w, \#)$ in ein oder mehreren Schritten in die Konfiguration $(\epsilon, \#)$ übergeht.

Nachteil des Deterministischen KA

Es gibt keinen deterministischen Kellerautomaten der die Sprache

$$\{ww^R \mid w \in (0|1)^*\}$$

akzeptiert. Hierbei ist

$$w^R = (w_n ... w_1)$$

für $w = (w_1 ... w_n)$.

Nicht deterministischer Kellerautomat

Definition 61. *Ein nicht deterministischer Kellerautomat ist ein Sechstupel* $KA = (Z, \Sigma, \Gamma, \delta, z_0, \#)$*, wobei*

- Z *eine endliche Menge von Zuständen,*
- Σ *eine endliche Eingabemenge,*
- Γ *eine endliche Menge von "Kellerbuchstaben",*
- $\delta : Z \times (\Sigma \cup \{\epsilon\}) \times \Gamma \rightarrow \mathcal{P}_e(Z \times \Gamma^*)$ *eine Überführungsfunktion,*
- $z_0 \in \Sigma$ *der Startzustand und*
- $\# \in \Gamma$ *das unterste Kellerzeichen.*

$\longrightarrow \mathcal{P}_e(M)$ bedeutet hierbei die Menge der endlichen Teilmengen von M.

Sprache eines Kellerautomaten

Definition 62. $k = (z, w, W) \in Z \times \Sigma^* \times \Gamma^*$ *heißt Konfiguration des Kellerautomaten.*
Auf der Menge der Konfigurationen definieren wir die Relation $\vdash$

falls $(z', B_1...B_k) \in \delta(z, a_1, A_1)$*:*

$$(z, a_1...a_n, A_1...A_m) \vdash (z', a_2...a_n, B_1...B_k A_2...A_m)$$

falls $(z', B_1...B_k) \in \delta(z, \epsilon, A_1)$*:*

$$(z, a_1...a_n, A_1...A_m) \vdash (z', a_1...a_n, B_1...B_k A_2...A_m).$$

Es sei $\vdash^*$ *die reflexive und transitive Hülle von* $\vdash$*.*
Ein Kellerautomat akzeptiert die Sprache

$$\{x \in \Sigma^* \mid \exists z \in Z : \ (z_0, x, \#) \vdash^* (z, \epsilon, \epsilon)\}.$$

Äquivalenter Kellerautomat

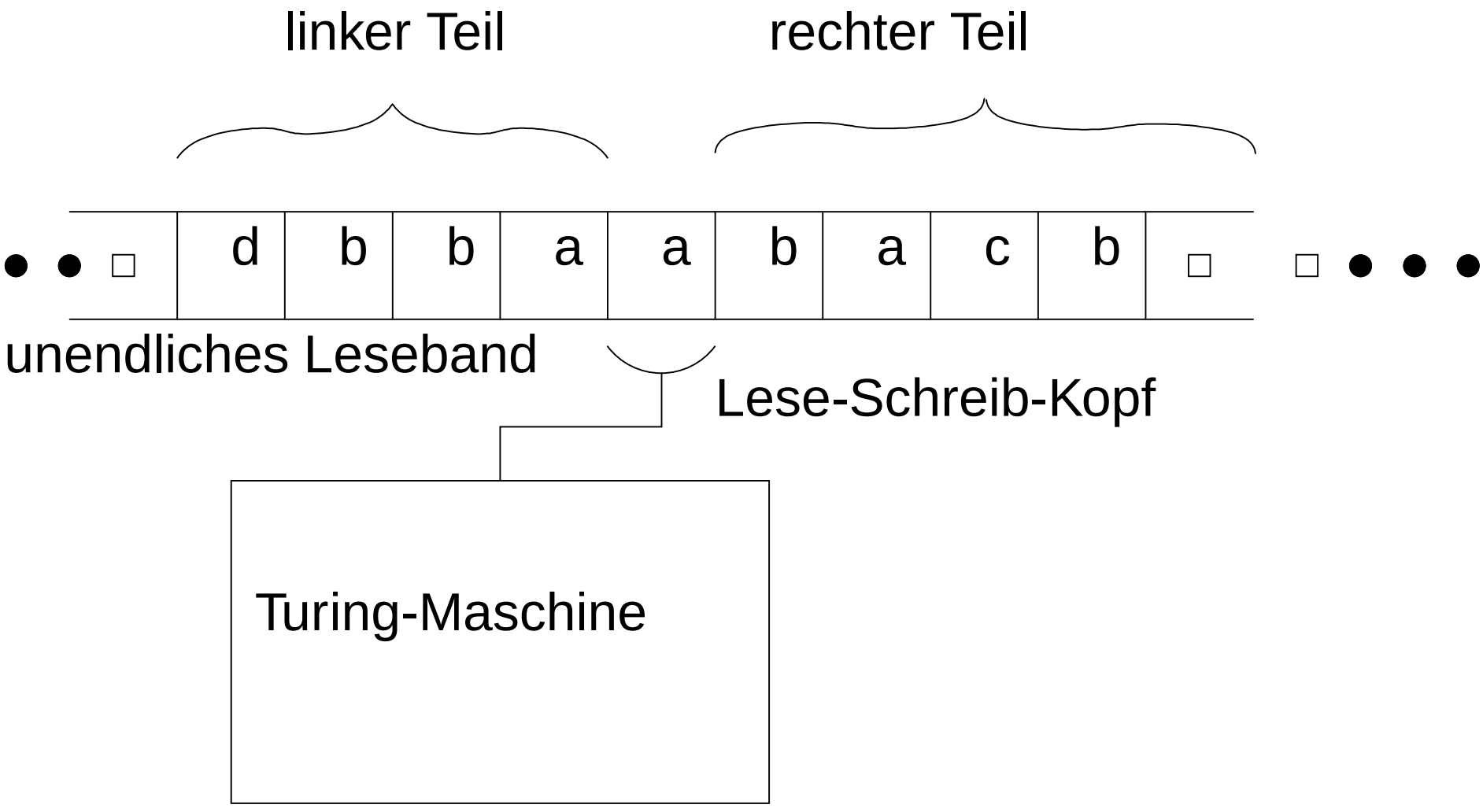
Definition 63. *Ein nicht deterministischer Kellerautomat mit Endzuständen ist ein Siebentupel* $KA = (Z, \Sigma, \Gamma, \delta, z_0, \#, E)$*, wobei*

- Z *eine endliche Menge von Zuständen,*
- Σ *eine endliche Eingabemenge,*
- Γ *eine endliche Menge von "Kellerbuchstaben",*
- $\delta : Z \times (\Sigma \cup \{\epsilon\}) \times \Gamma \to \mathcal{P}_e(Z \times \Gamma^*)$ *eine Überführungsfunktion,*
- $z_0 \in \Sigma$ *der Startzustand,*
- $\# \in \Gamma$ *das unterste Kellerzeichen und*
- $E \subset Z$ *eine Menge von Endzuständen.*

Dieser Kellerautomat akzeptiert die Sprache

$$\{x \in \Sigma^* \mid \exists(\gamma \in \Gamma^*,\ z \in E) :\ (z_0, x, \#) \vdash^* (z, \epsilon, \gamma)\}.$$

Turing-Maschine



Turing-Maschine

Definition 64. *Eine Turing-Maschine ist ein Siebentupel* $T = (Z, \Sigma, \Gamma, \delta, z_0, \Box, E)$*, wobei*

- Z *eine endliche Menge von Zuständen,*
- Σ *eine endliche Eingabemenge,*
- $\Gamma \supset \Sigma$ *eine endliche Menge (Arbeitsalphabet),*
- $\delta : Z \times \Gamma \to Z \times \Gamma \times \{L, R, N\}$ *eine Überführungsfunktion, falls* T *deterministisch,*
- $\delta : Z \times \Gamma \to \mathcal{P}(Z \times \Gamma \times \{L, R, N\})$ *eine Überführungsfunktion, falls* T *nicht deterministisch,*
- $z_0 \in Z$ *der Startzustand und*
- $\Box \in \Gamma \backslash \Sigma$ *das Blank,*
- $E \subset Z$ *eine Menge von Endzuständen.*

Konfiguration einer Turing-Maschine

Definition 65. $k = (w_l, z, w_r) \in \Gamma^* \times Z \times \Gamma^*$ *heißt Konfiguration der Turing-Maschine* T*. Dies bedeutet:*

- *T ist im Zustand* z*.*
- *Das Wort links vom Schreibkopf ist* w_l*.*
- *Das Wort rechts vom Schreibkopf ist* $a_2 \ldots a_s$*, wobei* $w_r = a_1 a_2 \ldots a_s$*.*

Bewegung einer Turing-Maschine

Definition 66. *Wir definieren die Relation (Funktion)* $\vdash$

falls $\delta(z, b_1) = (z', c, N)$, $m \geq 0$, $n \geq 1$:

$$(a_1 ... a_m, z, b_1 ... b_n) \vdash (a_1 ... a_m, z', cb_2 ... b_n)$$

falls $\delta(z, b_1) = (z', c, R)$, $m \geq 0$, $n \geq 2$:

$$(a_1 ... a_m, z, b_1 ... b_n) \vdash (a_1 ... a_m c, z', b_2 ... b_n)$$

falls $\delta(z, b_1) = (z', c, R)$, $m \geq 0$, $n = 1$:

$$(a_1 ... a_m, z, b_1) \vdash (a_1 ... a_m c, z', \Box)$$

falls $\delta(z, b_1) = (z', c, L)$, $m \geq 1$, $n \geq 1$:

$$(a_1 ... a_m, z, b_1 ... b_n) \vdash (a_1 ... a_{m-1}, z', a_m c b_2 ... b_n)$$

falls $\delta(z, b_1) = (z', c, L)$, $m = 0$, $n \geq 1$:

$$(, z, b_1 ... b_n) \vdash (, z', \Box c b_2 ... b_n).$$

Sprache einer Turing-Maschine

Definition 67. *Es sei $\vdash^*$ die reflexive und transitive Hülle von $\vdash$.*
Ein Turing-Maschine akzeptiert die Sprache

$$\{x \in \Sigma^* \mid \exists(z \in E, \alpha, \beta \in \Gamma^*) : \ (z_0, x) \vdash^* (\alpha, z, \beta)\}.$$