

# ALGEBRAISCHE GRUNDLAGEN

## Kurzbeschreibung

Vorliegende Kurseinheit behandelt mit den Relationen (Halbgruppen, Graphen) und den Algebren (Formeln, Boolesche Algebra) zwei zentrale mathematische Hilfsmittel der Informatik.

## Schlüsselwörter

Algebraische Struktur, Halbgruppe, Monoid, Relation, Graph, Warshall-Algorithmus, Formel, Term, Schreibweisen, Boolesche Algebra, Dualitätsprinzip, Normalform

## Lernziele

1. Die Bedeutung von algebraischen Strukturen in der Informatik wird erkannt.
2. Der vielseitige Einsatz von Relationen bzw. Graphen zur formalen Beschreibung von Sachverhalten und die Möglichkeiten der darauf aufsetzenden (Graph-) Algorithmen zur Lösung von Problemstellungen werden verstanden.
3. Der Aufbau der booleschen Algebra kann wiedergegeben werden.

## Hauptquellen

- Gerhard Goos: Vorlesungen über Informatik, Band 1: Grundlagen und funktionales Programmieren, Springer Verlag 1995.

## Inhaltsverzeichnis

1	RELATIONEN .....	2
1.1	Halbgruppen und Monoide .....	3
1.2	Graphen .....	5
2	ALGEBREN .....	20
2.1	Formeln.....	21
2.2	Boolesche Algebra.....	25
	VERZEICHNISSE.....	29
	Abkürzungen und Glossar .....	29
	Index .....	30
	Informationen und Interaktionen .....	30
	Literatur .....	31

- RELATIONEN
  - Halbgruppen und Monoide, Operationen und Gesetze, algebraische Abgeschlossenheit, Einselement, Graphen, gerichtete und ungerichtete Graphen, Repräsentation von Graphen, Warshall-Algorithmus
- ALGEBREN
  - Formeln, Term, Kantorowitsch-Baum, Boolesche Algebra, Gesetze, Dualität

### Information 1: ALGEBRAISCHE GRUNDLAGEN

## 1 RELATIONEN

Relationen stellen Zusammenhänge zwischen den Elementen einer Menge oder den Elementen von zwei oder mehr Mengen dar. In praktisch jeder Aussage werden gewisse Gegenstände in Beziehung gesetzt, also Relationen gebildet [Go95].

- Durch Relationen werden Elemente einer oder mehrerer Mengen in Beziehung gesetzt
  - praktisch jede Aussage enthält Relationen
  - Beispiel: "Das Haus hat vier Außenwände"
- In der Informatik werden Relationen zur Modellierung von Systemen benötigt
  - Relationen sind ein wesentlicher Bestandteil der verschiedenen Diagramme der Unified Modeling Language
- Die einfachste im Zusammenhang mit der Zeichenverarbeitung auftretende Beziehung führt zu der algebraischen Struktur der Halbgruppen
- Aus der graphischen Darstellung von Relationen resultieren die Graphen

### Information 2: RELATIONEN - Überblick

Beispielsweise werden in der Aussage Das Haus hat vier Außenwände (siehe Information 2) die Gegenstände Haus und Außenwände in Relation gesetzt. Die zu einem Element aufstellbaren Relationen können sehr vielfältig sein.

Relationen sind in der Informatik so wichtig, weil sie eine ganz wesentliche Grundlage zur Modellierung von Systemen bilden. Anhand der im Zusammenhang mit dem Modellbegriff in der Kurseinheit GRUNDBEGRIFFE DER INFORMATIK [C&M-GI] kennen gelernten Diagrammtypen, die durch die *Unified Modeling Language* (UML) bereitgestellt werden, wird diese Bedeutung offensichtlich.

In diesem Kapitel werden mit den Halbgruppen und Graphen die konzeptionellen Grundlagen zu den Relationen gelegt.

## 1.1 Halbgruppen und Monoide

Im Folgenden werden ganz elementare mathematische Begriffe anhand eines einfachen Sachverhalts, nämlich dem Nebeneinanderschreiben von Zeichen, eingeführt. Das Ziel besteht darin, die dadurch festgelegte Relation mathematisch geeignet zu formulieren.

[www.cm-tm.uka.de/info1](http://www.cm-tm.uka.de/info1)  
Info1-Team (Prof. Abeck)

- Die betrachtete Relation ist das Nebeneinanderschreiben von Zeichen
- Mathematische Formulierung als Algebra
  - Menge: Menge aller Zeichenreihen über der Grundmenge  $\Sigma^+$
  - Operationen:
    - $\cdot$  Konkatenation
  - Gesetze
    - HG1:  $(a \cdot b) \cdot c = a \cdot (b \cdot c)$  Assoziativgesetz
- Die Eigenschaft der algebraischen Abgeschlossenheit ist erfüllt
- Die beschriebene Algebra  $(\Sigma^+, \cdot, \text{HG1})$ , kurz auch als  $\Sigma^+$  bezeichnet, bildet eine Halbgruppe

### Information 3: Halbgruppe

Die in Information 3 angestellten Überlegungen führen zum zentralen Begriff der Algebra.

#### (1) Trägermenge

Im Beispiel wird die Trägermenge gebildet durch die Zeichen bzw. genauer durch die Folgen von Zeichen, also die Zeichenreihen.

#### (2) Operationen

Die einzige Operation im Beispiel ist das Nebeneinanderschreiben, die auch als Konkatenation oder Verkettung bezeichnet wird. Hierzu wird durch das Zeichen  $\cdot$  ein geeignetes mathematisches Symbol eingeführt.

#### (3) Gesetze

Die Gesetze verbinden Trägermenge und Operationen, da durch diese die Frage beantwortet wird, welche Aussagen über die Operation, die auf der Trägermenge arbeitet, getroffen werden können. Ein offensichtlich geltendes Gesetz, die die Konkatenation erfüllt, ist das Assoziativgesetz.

Neben den Gesetzen wird die grundlegende Eigenschaft der algebraischen Abgeschlossenheit gefordert, die eine Algebra zu erfüllen hat.

Algebren, die bestimmte Strukturen haben, erhalten einen Namen. So heißt eine Algebra, die im Zusammenhang mit der Konkatenation von Zeichen erstellt wurde, eine Halbgruppe. Eine Halbgruppe ist also eine spezielle Algebra

$$\mathcal{A} = (\Sigma^+, \cdot, \{\text{HG1}\})$$

Eng mit dem Begriff der Halbgruppe ist der Begriff des Monoids verknüpft, der in Interaktion 1 näher ausgeführt wird.

- Ein Monoid ist eine Halbgruppe mit Einselement
  - Eigenschaft eines Einselements  $\varepsilon$ 
    - (1)  $\varepsilon \cdot a = a$
    - (2)  $a = a \cdot \varepsilon$
  - das Einselement ist eindeutig

Beweis: \_\_\_\_\_  
\_\_\_\_\_

- Beispiel eines Monoids:  $\Sigma^* := \Sigma^+ \cup \{\varepsilon\}$ 
  - $\varepsilon$  ist das leere Wort, d.h. die Länge  $|\varepsilon|$  ist 0
  - falls  $w^n$  die n-fache Wiederholung des Wortes  $w$  bezeichnet, so gilt  $w^0 = \varepsilon$
- Satz  
Das Monoid  $\Sigma^*$  über einem endlichen Zeichenvorrat  $\Sigma$  ist abzählbar
  - Beweis mittels Gödelnummerierung

### Interaktion 1: Monoid

Das in einem Monoid geforderte Einselement wird auch als neutrales Element bezeichnet. In einer Halbgruppe kann es Elemente geben, die nur eine der beiden Gleichungen erfüllen. Falls nur die in Interaktion 1 angegebene Eigenschaft (1) erfüllt ist, so heißen sie Linkseinselement, falls nur (2) erfüllt ist, entsprechend Rechtseinselement.

Ein Element, das die Eigenschaften (1) und (2) erfüllt, ist eindeutig, wie in Interaktion 1 zu zeigen ist.

Die Informatik behandelt vorwiegend solche Strukturen, die auf einem Rechensystem bearbeitet werden können. Hieraus resultiert die Eigenschaft, dass die Zeichenvorräte zwangsläufig endlich sein müssen. Die Konsequenz, die sich für Monoide über einem endlichen Zeichenvorrat ergibt, ist die Abzählbarkeit. Der Beweis der Abzählbarkeit erfolgt mittels der so genannten Gödelnummerierung (Kurt Gödel, Logiker des 20. Jahrhunderts).

Die algebraische Struktur der Monoide ist für die Informatik bedeutsam, da sie zum Aufbau von listen- und sequenzartigen Datenstrukturen dient (siehe Interaktion 2). Viele Algorithmen nutzen listen- bzw. sequenzartige Datenstrukturen. Daher ist es wichtig zu wissen, um welche mathematische Struktur es sich handelt und welche mathematischen Eigenschaften diese Struktur besitzt.



- In der Informatik werden Monoide  $U^*$  häufig über anderen Grundmengen als Zeichen gebildet
  - Grundmenge  $U$  können z.B. Texte sein, die aus einem anderen Monoid  $\Sigma^*$  stammen
  - Elemente von  $U^*$  heißen Listen
    - oder alternativ: Sequenzen, Folgen
  - Notation:
    - Liste:  $[x_1, x_2, \dots, x_n]$ , wobei  $x_i \in U$
    - $[]$  ist die leere Liste
    - $[U]$  als alternative Schreibweise für  $U^*$
- Beispiel:  $U = \{\text{Apfel, Birne, Pflaume, Kirsche, Traube}\}$ 
  - $[\text{Apfel, Birne}]$ ,  $[\text{Pflaume, Kirsche, Traube}]$  sind Listen
  - $\text{append}([\text{Apfel, Birne}], [\text{Pflaume, Kirsche, Traube}])$  ist eine alternative Funktionsschreibweise für

---

### Interaktion 2: Monoid $U^*$ über Monoid $\Sigma^*$

Das Vorgehen, das den Listen zugrunde liegt, besteht darin, die Trägermenge abzuändern in der Form, dass diese nicht aus einzelnen Zeichen, sondern selbst bereits aus Zeichenreihen besteht, womit das die Listen darstellende Monoid  $U^*$  auf dem Monoid  $\Sigma^*$  aufbaut.

Die in Interaktion 2 angegebene Listen-Notation für eine beliebige Liste, die leere Liste und den ganzen Listen-Monoid vereinfacht die Aufschreibung, macht die Syntax intuitiver und damit besser lesbar. Die Anwendung der Notation soll das Beispiel verdeutlichen, dessen Trägermenge  $U$  aus den fünf angegebenen Texten – in diesem Fall verschiedene Sorten Obst – besteht.

Eine weitere Art von Notation ist die Funktionsschreibweise, in der das mathematische Verknüpfungssymbol gegen eine Funktion ersetzt wird.

## 1.2 Graphen

Monoid-Operationen stellen die elementarste Relation zwischen Elementen einer Trägermenge dar. Im Folgenden werden die Beziehungen in Form des Begriffs der Relation mathematisch präzisiert und mittels Graphen veranschaulicht.

- Relationen
  - Grundmengen  $U, V$
  - Notation:  $\rho \subseteq U \times V$
- Jede Relation kann als Teilmenge des kartesischen Produkts  $(U \cup V) \times (U \cup V)$  angesehen werden
- Homogene Relation
  - nur eine Grundmenge  $E$ , d.h.  $\rho \subseteq E \times E$
- Die Informatik interessiert sich speziell für homogene Relationen mit einer endlichen Grundmenge  $E$ 
  - Durchnummerieren der Elemente  $e_i$  mit  $i = 0, \dots, n - 1$
  - gerichtete Graphen ermöglichen eine anschauliche Darstellung

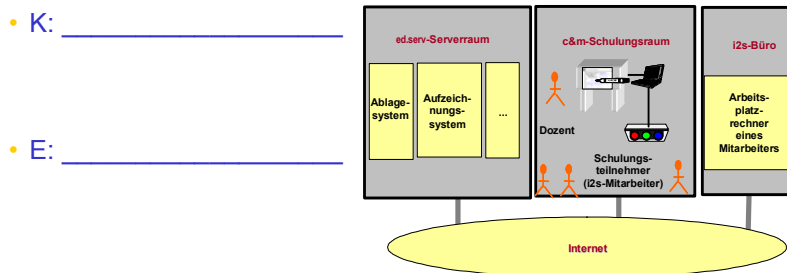
#### **Information 4: Relationen und Graphen**

Eine Relation liefert den grundlegenden Formalismus, Beziehungen zwischen Gegenständen zweier Grundmengen  $U$  und  $V$  zu beschreiben. Die in der Informatik übliche Beschränkung auf homogene Relationen ist bei näherer Betrachtung keine wirkliche Einschränkung. Wie in Information 4 ausgeführt ist, lassen sich nicht-homogene Relationen  $U \times V$  durch die Vereinigung der beiden Mengen  $U$  und  $V$  sowie die Bildung des kartesischen Produkts auf homogene Relationen zurückführen. Die nicht-homogene Relation ist dann eine Teilmenge der konstruierten homogenen Relation.

Eine homogene Relation (bezeichnet mit  $\rho$ ) ist also auf einer Grundmenge (bezeichnet mit  $E$ ) definiert.

Die Informatik stellt neben der Homogenität noch eine weitere Forderung an die Grundmenge, nämlich die Eigenschaft der Endlichkeit. Das ermöglicht dann ein Durchnummerieren der Elemente und eine anschauliche Darstellung der Relation in Form von gerichteten Graphen, auf die in Interaktion 3 näher eingegangen wird.

- Ein gerichteter Graph  $G$  ist ein Tupel  $(E, K)$  mit
  - Grundmenge  $E = \{e_i\}$  eine Menge von Ecken
  - Relation  $K \subseteq E \times E$  eine Menge von Kanten
    - alternative Notationen für Kanten
      - $(e, e') \in K$  oder  $e \rightarrow_G e'$
      - $e \rightarrow e'$
- Der Inhalt des Web-basierten Ablagesystems aus dem Beispiel-Szenario kann als gerichteter Graph aufgefasst werden



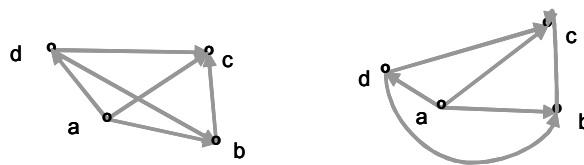
### Interaktion 3: Gerichteter Graph

Ein gerichteter Graph setzt sich aus zwei Mengen zusammen, einer Menge von Ecken (Knoten) und einer Menge von Kanten. Die Kantenmenge  $K$  stellen dabei zweistellige Relationen auf der Grundmenge  $E$  der Ecken dar.

Relationen und Graphen spielen in der Informatik eine ganz entscheidende Rolle, da zahlreiche Problemstellungen auf Graphen und entsprechende Graphenalgorithmien zurückgeführt werden können. Die Begriffe 'Relation' und 'Graph' werden dabei weitgehend synonym verwendet. In Interaktion 3 soll ein mögliches Einsatzfeld von Graphen anhand des in [C&M-IÜ] eingeführten Beispiel-Szenarios und des darin enthaltenen Web-basierten Ablagesystems skizziert werden.

Die endlichen Graphen – das sind solche mit endlicher Eckenzahl – haben die positive Eigenschaft, dass man sie zeichnen kann, d.h. die Pfeilnotation wird auf die Zeichenfläche ausgedehnt, wie das Beispiel in Information 5 zeigt.

- Ein Graph heißt endlich, wenn E endlich ist
- Graphische Darstellung von
  - Ecken als Punkte
  - Kanten als Pfeile
- Die Positionierung der Ecken und Kanten ist beliebig
- Beispiel: Zwei mögliche Darstellungen des Graphen  $\{(a,b),(a,c),(a,d),(b,c),(d,b),(d,c)\}$



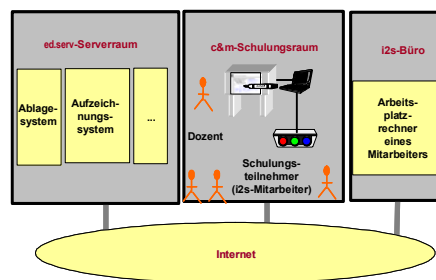
### Information 5: Graphische Darstellung eines endlichen gerichteten Graphen

Durch das Beispiel wird zudem gezeigt, dass es unterschiedliche Darstellungen von ein und demselben Graphen gibt.

- Ungerichteter (allgemeiner Graph)
  - die durch die Kanten ausgedrückte Beziehung beruht auf Gegenseitigkeit  
d.h. genau dann, wenn  $(e,e') \in K$  ist auch  $(e',e) \in K$
  - Kanten werden in der zeichnerischen Darstellung als einfache Verbindung ohne Pfeilspitzen angegeben
- Die Fähigkeit der im Beispiel-Szenario auftretenden Systeme, miteinander kommunizieren zu können, lässt sich als ungerichteter Graph auffassen

#### • Zeichnerische Darstellung

Ablage-  
system 

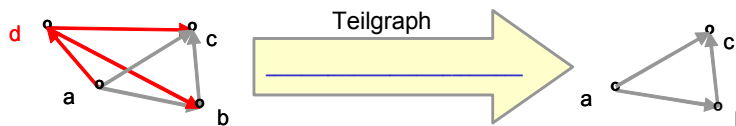


### Interaktion 4: Ungerichteter Graph

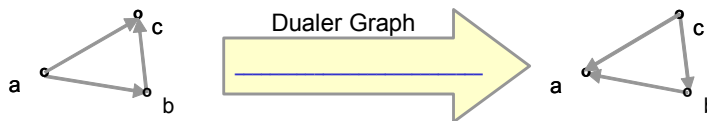
Es gibt Sachverhalte, bei denen eine Relation von a nach b zwangsläufig die Relation b nach a nach sich zieht. Ein Beispiel einer auf Gegenseitigkeit beruhenden Relation lautet: ist über einen Wanderweg verbunden mit. Wanderwege sind in aller Regel in beide Richtungen benutzbar, weshalb im Falle einer Relation zwischen einem Ort A mit einem Ort B automatisch gilt, dass auch B mit A in Relation steht, d.h. die Relationen beruhen auf Gegenseitigkeit. Zeichnerisch ist ein ungerichteter Graph daran zu erkennen, dass die Kanten keine Pfeilspitzen aufweisen.

Ein weiteres Beispiel eines ungerichteten Graphen, der aus dem Beispiel-Szenario gegriffen wurde, findet sich in Interaktion 4. Die zeichnerische Darstellung dieser Relation ist als Aufgabe im Rahmen der Interaktion anzugeben.

- Teilgraph  $G' = (E', K')$  zum Graphen  $G = (E, K)$ 
  - Einschränkung der Relation  $\rho$  auf eine Teilmenge  $E' \subseteq E$ ;  
Notation:  $\rho|_{E'}$
  - $K' = \{(e, e') \mid e, e' \in E' \text{ und } (e, e') \in K\}$



- Dualer Graph  $G^T = (E, K^T)$ 
  - $(e, e') \in K^T$  genau dann, wenn  $(e', e) \in K$
  - Der duale Graph  $G^T$  entsteht aus  $G$ , wenn alle Richtungen der Pfeile in  $G$  umgekehrt werden



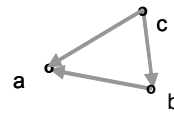
### Interaktion 5: Teilgraph und dualer Graph

Ein Teilgraph  $G'$  geht aus einem Graphen  $G$  durch Einschränkung der Relation auf gewisse Ecken aus  $E$  hervor. Zu  $K'$  gehören alle Kanten aus  $K$ , die Ecken  $e, e' \in E'$  miteinander verbinden.

Eine weitere Art von Graph, die aus einem gegebenen Graphen  $G$  hervorgeht, ist der duale Graph, der als  $G^T$  bezeichnet wird. Die Menge der Ecken  $E$  bleibt unverändert und die Menge der Kanten  $K^T$  geht durch Vertauschen der Start- und Zielecke jeder Kante hervor.

In Interaktion 5 soll der Übergang des Ausgangsgraphen in einen Teilgraphen bzw. in einen dualen Graphen durch Verwendung einer geeigneten Notation formal beschrieben werden.

- Ausgangsmenge  $e^\bullet$  und Eingangsmenge  ${}^\bullet e$ 
  - Menge der Kanten, die bzgl. der Ecke  $e$  aus- bzw. eingehen
  - $|e^\bullet|$  heißt Ausgangsgrad,  ${}^\bullet e$  heißt Eingangsgrad
  - in einem ungerichteten Graphen gilt  $|e^\bullet| = |{}^\bullet e| = \text{grad}(e)$  und heißt Grad der Ecke  $e$



$$a^\bullet = \underline{\hspace{2cm}} \quad |a^\bullet| = \underline{\hspace{2cm}}$$

$${}^\bullet a = \underline{\hspace{2cm}} \quad |{}^\bullet a| = \underline{\hspace{2cm}}$$

- Für beliebige Graphen gilt:  $\sum_{e \in E} |e^\bullet| = \sum_{e \in E} |{}^\bullet e|$
- Vollständiger (ungerichteter) Graph
  - zwischen je zwei Ecken existiert eine (ungerichtete) Kante

### Interaktion 6: Grad einer Ecke

Die Eingangs- und Ausgangsmenge beinhalten die von einer Ecke ausgehenden bzw. in eine Ecke mündenden Kanten. Zur Angabe dieser Mengen wird eine einfache Punkt-Notation eingeführt, wie in Interaktion 6 ausgeführt ist.

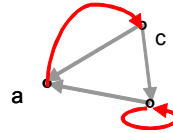
Eng verknüpft mit dem Eingangs- und Ausgangsmengen ist die Angabe des so genannten Grads. Der Grad gibt die Anzahl der Kanten an, die von der Ecke ausgehen bzw. zu dieser Ecke führen. Entsprechend dieser beiden Fälle sind der Ausgangsgrad und der Eingangsgrad zu unterscheiden. Zur Angabe dieser Grade wird die Punkt-Notation um Betrags-Striche ergänzt.

Die Übereinstimmung der Summe der Ausgangs- und Eingangsgrade aller Ecken eines Graphen lässt sich einfach erklären.

Für einen gerichteten Graphen gilt außerdem, dass die Eingangs- bzw. Ausgangsgradsumme genau der Anzahl der Kanten im Graphen entspricht.

Eine spezielle Form eines (üblicherweise ungerichteten) Graphen ist der vollständige Graph. Dieser hat die Eigenschaft, dass jede Ecke mit jeder anderen im Graphen auftretenden Ecke verbunden ist, was man auch durch den Grad einfach ausdrücken kann.

- Folge von  $e_0$  nach  $e_n$  über  $e_1, \dots, e_{n-1}$ 
  - heißt Weg  $e_0 \rightarrow^* e_n$  der Länge  $n$
  - $e_n$  heißt dann von  $e_0$  erreichbar
- Zyklus (Kreis)
  - ein Weg  $e_0 \rightarrow^* e_n$  mit  $n \geq 1$  und  $e_0 = e_n$
- Einfacher Zyklus
  - alle Ecken  $e_i$  sind verschieden
- Hamiltonscher Kreis
  - jede Ecke des (ungerichteten) Graphen ist genau einmal enthalten
- Eulerscher Zyklus
  - jede Kante ist genau einmal enthalten



### Information 6: Weg und Zyklus

Kanten können miteinander verknüpft werden, so dass Kantenzüge entstehen. Hierzu verbindet man solche zwei Kanten, bei denen die Eingangsecke der einen Kante mit der Ausgangsecke der anderen Kante übereinstimmen. Bei ungerichteten Graphen müssen die zu verbindenden Kanten nur eine gemeinsame Ecke besitzen.

Ein spezieller Weg ist der Zyklus oder Kreis, der an der Ecke ( $e_n$ ) endet, an der er beginnt ( $e_0$ ). Ein Zyklus der Länge 1 wird als Schlinge bezeichnet (d.h. ein und denselben Knoten ist Anfangs- und Endpunkt).

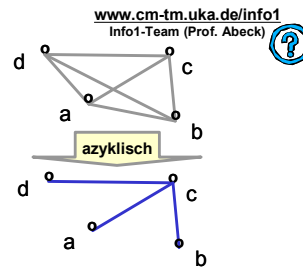
Die bislang behandelten Graphen enthalten keine Schlingen und auch keinen sonstigen längeren Zyklus. In dem in Information 6 dargestellten Beispiel-Graphen, in dem einer Kante ( $a, c$ ) zu dem in Interaktion 5 erstellten Teil-Graphen hinzugenommen wurde, bestehen die Zyklen  $a \rightarrow c \rightarrow a$  und  $a \rightarrow c \rightarrow b \rightarrow a$ . Außerdem wurde im Beispiel-Graphen zur Ecke  $b$  eine Schlinge hinzugefügt.

Es werden verschiedene Arten von Zyklen bzw. Kreise unterschieden:

- Ein einfacher Zyklus ist gegeben, wenn keine zwei gleichen Ecken auftreten.
- Ein hamiltonscher Kreis ergibt sich durch eine Verschärfung dieser Forderung, nämlich dass jede Ecke nicht nur höchstens einmal, sondern genau einmal auftreten darf.
- Der eulersche Zyklus stellt die Anforderung, dass jede Kante des betrachteten Graphen genau einmal enthalten sein muss.

Der eulersche Zyklus wurde von Leonhard Euler (schweizerischer Mathematiker des 18. Jh.) im Zusammenhang mit dem Königsberger Brückenproblem definiert, bei dem ein Weg über die damaligen 7 Brücken der Pregel gesucht wurde, bei dem keine der Brücken zweimal überquert werden musste. Bei der Darstellung als Graph sind die Brücken die Kanten und das die Brücken verbindende Land die Ecken. Einen solchen Weg gibt es genau, wenn der Grad aller Ecken gerade ist, was beim Graphen, der das Königsberger Brückenproblem darstellte, nicht gegeben war.

- Azyklischer Graph: Graph ohne Zyklen
- Wald: Azyklischer ungerichteter Graph
- Ungerichteter Baum: Wald, bei dem je zwei Ecken durch genau einen Weg verbunden sind



Welcher spezielle Graph liegt hier vor?

- Satz: Für einen ungerichteten Baum  $G = (E, K)$  mit endlich vielen Ecken gilt  $|E| = |K| + 1$ 
  - Beweis durch vollständige Induktion entlang der Eckenzahl

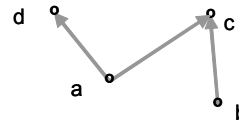
### Interaktion 7: Spezielle Graphen im Zusammenhang mit Zyklen

Im Zusammenhang mit Zyklen werden spezielle Graphen unterschieden. Eine Art von speziellen Graphen sind die in der Informatik besonders wichtigen Bäume. Die wesentliche Eigenschaft dieser Graphen-Art ist die Zyklensfreiheit und das Zusammenhängen (Verbundenseins von je zwei Ecken über beliebige Wege).

Erfüllt ein ungerichteter Graph nur die Eigenschaft der Zyklensfreiheit, nicht aber die Eigenschaft des Zusammenhängens, so spricht man von einem Wald. Offensichtlich besteht ein Wald-Graph aus einzelnen Baum-Graphen.

Bei einem ungerichteten Baum stehen die Anzahl der Ecken und der Kanten in einem bestimmten Verhältnis: Es gibt genau eine Ecke mehr als Kanten. Diese Eigenschaft ist in Interaktion 7 durch einen Induktionsbeweis zu zeigen.

- **Gerichteter Wald**  
Ein azyklischer Graph, bei dem alle Knoten einen Eingangsgrad  $|e^*| \leq 1$  besitzen



- **Gerichteter Baum**  
Ein gerichteter Wald, bei dem es genau eine Ecke  $e$  mit  $|e^*| = 0$  gibt
  - diese Ecke heißt die Wurzel des Baumes
- **Blatt**
  - bei einem ungerichteten Graphen: Ecke mit  $\text{grad}(e) = 1$
  - bei einem gerichteten Graphen: Ecke mit Ausgangsgrad  $|e^*| = 0$

Der Graph ist (1) in einen gerichteten Wald und anschließend (2) in einen gerichteten Baum zu überführen

(1) \_\_\_\_\_

(2) \_\_\_\_\_

Welche Ecken des erstellten gerichteten Baumes sind Blätter?

\_\_\_\_\_

### Interaktion 8: Gerichtete Wälder und Bäume

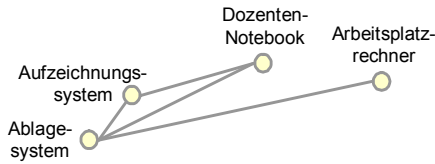
Die bislang gemachten Angaben haben sich auf ungerichtete Graphen bezogen. Im Falle von gerichteten Graphen gelten für einen Wald und einen Baum die in Interaktion 8 angegebenen Eigenschaften, die gewisse Forderungen an den Eingangsgrad der im Graphen auftretenden Ecken haben.

Die Forderung an einen gerichteten Wald ist, dass zu jedem Knoten höchstens eine Kante führt. Offensichtlich ist das für den in Interaktion 8 angegebenen Beispiel-Graphen nicht erfüllt.

An einen gerichteten Baum wird die Forderung hinsichtlich der Eingangsgrade so verschärft, dass es eine ausgezeichnete Ecke gibt, zu der gar keine Kante führt. Diese Ecke heißt Wurzel und muss eindeutig sein, womit auch von einem Baum gefordert wird, dass keine isolierten Ecken oder Teilgraphen auftreten können.

In gewisser Weise das Gegenstück zu der Wurzel ist das Blatt. Der Begriff des Blattes ist nicht nur für Bäume, sondern für beliebige ungerichtete und gerichtete Graphen definiert.

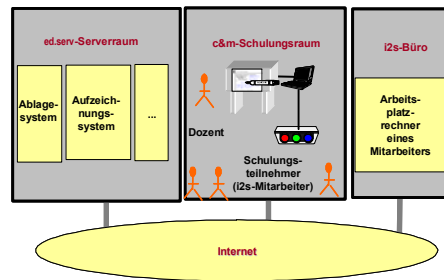
- $G = (E, K)$  heißt ein markierter Graph, wenn es Markierungsfunktionen
  - $M_E: E \rightarrow G_E$
  - $M_K: K \rightarrow G_K$
 mit geeigneten Mengen  $G_E, G_K$  gibt



- Welche Mengen liegen dem markierten Graph zugrunde:

$G_E =$  \_\_\_\_\_

$G_K =$  \_\_\_\_\_

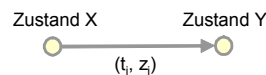
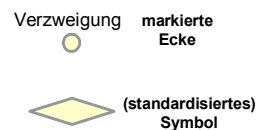
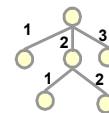
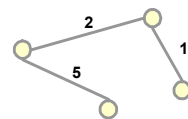


### Interaktion 9: Markierter Graph

Durch die Zuordnung von Eigenschaften zu Ecken und insbesondere zu den Kanten kann man dem Graphen eine bestimmte Bedeutung zuordnen. Diese Zuordnung wird als Markierung bezeichnet, der resultierende Graph heißt entsprechend markierter Graph. Formal bedeutet die Markierung die Einführung von zwei Markierungsfunktionen, einer Eckenmarkierung und einer Kantenmarkierung. Mit der Einführung der Funktionen müssen auch die Wertebereiche festgelegt werden, in die abgebildet werden soll.

In Interaktion 9 wird neben der formalen Definition eines markierten Graphen ein bereits in Interaktion 4 eingeführter Beispiel-Graph gezeigt, der markiert ist. Die dem Graphen zugrunde liegenden Eckenmarkierungs- und Kantenmarkierungsmengen sind anzugeben.

- Mehrfachgraph
  - Kantenmarkierung  $M_K: K \rightarrow \mathbb{N}$  gibt an, dass die Beziehung auf mehrere Weisen existiert
- Geordneter Baum
  - gerichteter Baum mit einer Kantenmarkierung  $M_K: K \rightarrow \mathbb{N}$ , die die Kanten von links nach rechts durchnummeriert und somit ordnet
- Weitere praktische Beispiele
  - Flussdiagramm zu Programmen in höheren Programmiersprachen
    - Graphen mit markierten Ecken, wobei  $G_E = \{\text{Programmstart, Programmende, Verzweigung, Anweisung}\}$
  - Netzplan zur Planung von Projektabläufen
    - Markierung der Ecken: Zustände, die im Projekt erreicht werden können
    - Markierung der Kanten: Paar (Tätigkeit, Zeitdauer der Tätigkeit)



### Information 7: Ausprägungen von markierten Graphen

Die Markierungsfunktionen können für unterschiedliche Zwecke sinnvoll genutzt werden, was zu verschiedenen Ausprägungen von markierten Graphen führt. In vielen Fällen, wie z.B. auch in dem von Euler gelösten Königsberger Brückenproblem, kommt es vor, dass zwei Ecken durch mehr als eine Kante verbunden sind.

Mit einer Kantenmarkierung, die jeder Kante eine natürliche Zahl zuordnet, lassen sich auch so genannte geordnete Bäume erstellen. Beispiele für geordnete Bäume sind Ableitungsbäume oder Kantorowitsch-Bäume.

Von besonderer praktischer Bedeutung sind die Flussdiagramme zu Programmen, die ebenfalls eine bestimmte Ausprägung markierter Graphen darstellen. Spätestens hier wird deutlich, welche umfangreichen Möglichkeiten durch Graphen geboten werden, wenn man den Ecken und/oder Kanten eine bestimmte Semantik zuordnet.

Während in den vorhergehenden zwei Ausprägungen die Kanten markiert wurden, sind es bei den Flussdiagrammen die Ecken, denen jeweils eine von vier Bedeutungen zugewiesen wird, wobei nur eine Ecke mit dem Programmanfang markiert sein darf. Zu den Flussdiagrammen sei Folgendes angemerkt:

- (1) Markierung der von der Verzweigungs-Ecke ausgehenden Kanten mit ja (then-Fall) bzw. nein (else-Fall).
- (2) Es sind ggf. mehrere Programmende-Ecken möglich (allerdings nur eine Programmanfang-Ecke).
- (3) Der Ecken-Typ legt den Ausgangsgrad eindeutig fest, z.B.  $|e^*| = 0$ , wenn  $M_E(e) = \text{Programmende}$ .

Ein weiteres und abschließendes Beispiel eines markierten Graphen ist der in zahlreichen Anwendungsszenarien genutzte Netzplan. Ein Netzplan stellt die Ablaufbeschreibung eines Projekts dar. Es handelt es sich hierbei um einen Graphen mit folgenden Eigenschaften:

- Er enthält keine geschlossenen Kantenzüge, d.h. er ist zyklensfrei.
- Er beinhaltet zwei ausgezeichnete Ecken, ähnlich wie das zuvor behandelte Flussdiagramm.
- Diese Ecken sind durch ihren Eingangs- bzw. Ausgangsgrad gekennzeichnet.

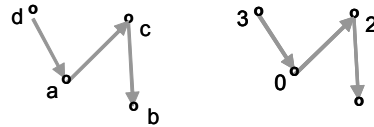
Durch die Markierungsfunktionen wird den Ecken und Kanten folgende Bedeutung zugeordnet:

- Ecken beschreiben die Projektzustände
- Kanten beschreiben, wie man von einem Projektzustand in den nächsten kommt. Hierzu ist eine bestimmte Tätigkeit erforderlich. Zu der Tätigkeit wird noch die erforderliche Zeitdauer angegeben.

Am Beispiel des Netzplans lässt sich leicht klarmachen, dass durch geeignete Auswertung der im Graph gehaltenen Informationen relevante Fragen zur Projektplanung beantwortet werden können.

Die Bearbeitung von Netzplan-Fragen durch einen Rechner setzt voraus, dass ein Graph geeignet im Speicher dieses Rechner gehalten werden kann, was zu den in Information 8 behandelten Repräsentationsmöglichkeiten von Relationen und Graphen führt.

- Gegeben sei ein Graph  $G = (E, K)$ , wobei jede Ecke durchnummeriert ist



- Darstellungsform 1: Adjazenzmatrix
  - $n \times n$  - Matrix  $A = (a_{ij})$
  - $a_{ij} = 1$ , wenn  $(i, j) \in K$ ,  $a_{ij} = 0$  sonst
  - anstelle von  $\{0, 1\}$  ist auch ein beliebiger anderer Binärcode möglich

nach	0	1	2	3
von 0	0	0	1	0
1	1	0	0	0
2	0	1	0	0
3	1	0	0	0

- Darstellungsform 2: Adjazenzliste
  - jeder Ecke  $e_i$  wird die Menge  $\{e_{i_1}, \dots, e_{i_n}\}$  der Ecken  $e_j$  mit  $e_i \rightarrow e_j$  zugeordnet
  - die jeder Ecke zugeordneten Mengen lassen sich sinnvoll durch Listen wiedergeben

0: [2, 3]  
1: []  
2: [1]  
3: [0]

### Information 8: Repräsentation von Relationen und Graphen

Zunächst sind die Bezeichnungen der Ecken geeignet zu nummerieren, d.h. der "erste" Knoten erhält die Nummer 0, der zweite Knoten erhält die Nummer 1 usw.

In den Beispiel-Graphen würde eine denkbare Durchnummerierung bedeuten, dass der Ecke mit der Bezeichnung a die Nummer 0, der Ecke mit der Bezeichnung b die Nummer 1 usw. zugewiesen werden könnte.

Die Frage ist nun, wie man einen solchen Graphen (bzw. allgemein eine Relation) in einem Rechner repräsentiert. Offensichtlich ist die bislang verwendete graphische Darstellung hierfür nicht geeignet.

Es werden zwei Darstellungsformen unterschieden, wobei die erste die so genannte Adjazenzmatrix ist. Es handelt es sich hierbei um eine Matrix mit einer Zeilen- und Spaltenzahl, die der Knotenzahl des Graphen entspricht. Die Zeilen und Spalten der Matrix sind mit den Ecken indiziert. Der Wert eines Adjazenzmatrix-Elements  $(i, j)$  bestimmt, ob eine Kante zwischen Ecke i und Ecke j besteht (Wert 1) oder nicht (Wert 0).

Somit ergibt sich für den in Information 8 gezeigten Beispiel-Graphen die o.a. Adjazenzmatrix. Statt der verwendeten Menge  $\{0, 1\}$  kann auch ein beliebiger anderer Binärcode, wie z.B.  $\{0, L\}$  oder  $\{\text{wahr}, \text{falsch}\}$  verwendet werden.

Die zweite Darstellungsform wird als Adjazenzliste bezeichnet. Bei dieser Darstellungsform eines Graphen werden zu jedem Knoten alle Knoten aufgelistet, zu denen eine Kante existiert. Das Vorgehen angewendet auf den Beispiel-Graphen führt zu dem in Information 8 angegebenen Ergebnis.

- Es wird davon ausgegangen, dass die Relation als Adjazenzmatrix vorliegt
- Eigenschaften einer Relation
  - Relation reflexiv  $\Leftrightarrow a_{ii} = 1$  für alle  $i = 0, \dots, n-1$
  - Relation symmetrisch  $\Leftrightarrow a_{ij} = a_{ji}$
  - Relation transitiv  $\Leftrightarrow$  aus  $a_{ij} = 1$  und  $a_{jk} = 1$  folgt  $a_{ik} = 1$
- Verknüpfung  $\rho \circ \sigma$  zweier Relationen  $\rho, \sigma \subseteq E \times E$ 
  - $\rho$  durch Adjazenzmatrix  $A = (a_{ij})$ ,  $\sigma$  durch  $B = (b_{ij})$  repräsentiert
  - $C = (c_{ij})$  ist die Adjazenzmatrix der Verknüpfung  $\rho \circ \sigma$  mit

$$c_{ij} = \begin{cases} 1, & \text{wenn } \sum_{k=0}^{n-1} a_{ik} b_{kj} \geq 1 \\ 0, & \text{sonst} \end{cases}$$

### Information 9: Adjazenzmatrix

Die Eigenschaften von Relationen spiegeln sich in Form konkreter Werte wider, die die Elemente in der zu der Relation zugehörigen Adjazenzmatrix annehmen:

- Eine reflexive Relation ist gegeben, wenn die Diagonale der Adjazenzmatrix ausschließlich aus 1 besteht.
- Eine symmetrische Relation führt zu einer symmetrischen Matrix, d.h. also  $a_{ij} = a_{ji}$ .
- Die Eigenschaft der Transitivität bedeutet, dass bei einem (aus zwei Kanten bestehenden) Weg von Knoten  $i$  zu  $j$  und von  $j$  zu  $k$  gewährleistet ist, dass eine Kante von  $i$  zu  $k$  besteht.

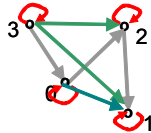
Information 9 zeigt, wie die Verknüpfung  $\rho \circ \sigma$  von zwei über  $E \times E$  definierte Relationen  $\rho$  und  $\sigma$  festgelegt ist.

Die Eigenschaften der Reflexivität und insbesondere der Transitivität können für eine beliebige Relation bzw. Graphen erzwungen werden, indem die so genannte reflexive transitive Hülle gebildet wird.



- Eine reflexive transitive Hülle zu einem Graphen entsteht dadurch, dass zu dem Graphen Kanten zwischen jeweils solchen zwei Knoten hinzugefügt werden, zwischen denen ein Weg besteht

- Beispiel:



nach	0	1	2	3
von 0	0	—	1	—
von 1	—	—	—	—
von 2	—	1	—	—
von 3	1	—	—	—

Die Adjazenzmatrix ist so zu aktualisieren, dass hierdurch die reflexive transitive Hülle des Graphen wiedergegeben wird

- Definition der reflexiven transitiven Hülle  $\rho^*$  zu einer Relation  $\rho$  :  
 $\rho^* = \{(i,j) \mid \text{es gibt einen Weg zwischen Knoten } i \text{ und Knoten } j\}$
- Bei der transitiven Hülle  $\rho^+$  werden die Schlingen (Wege der Länge 0) nicht hinzugefügt

### Interaktion 10: Reflexive transitive Hülle eines Graphen

Wie in Interaktion 10 ausgeführt ist, werden bei der Hüllen-Bildung Kanten hinzugefügt, um die beiden Eigenschaften zu erfüllen. Am Beispiel-Graphen, dessen Adjazenzmatrix entsprechend zu ergänzen ist, wird das Vorgehen deutlich gemacht.

Die reflexive transitive Hülle wird durch ein \* zu der Relationsbezeichnung hinzugefügt, die transitive Hülle durch ein + Additionszeichen kenntlich gemacht.

Von Warshall wurde 1962 ein einfaches, nicht-rekursives Verfahren entwickelt, das die Hüllenberechnung eines Graphen durchführt. Dem Verfahren liegt der folgende offensichtliche Sachverhalt zugrunde [Se02]: Falls ein Weg existiert, um von Knoten  $x$  zu Knoten  $y$  zu gelangen, und ein Weg, um von  $y$  nach  $j$  zu gelangen, so existiert auch ein Weg, um von  $x$  nach  $j$  zu gelangen.

Die Idee, die zum Warshall-Algorithmus führt, besteht darin, die Weglänge und die darin benutzten Knotennummern schrittweise (jeweils um 1) zu erhöhen. Aus dieser Überlegung resultiert die in Information 10 angegebene Relation  $\sigma$ .

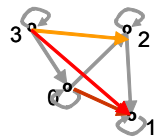
- Verfahren gemäß Warshall

Gegeben sei eine reflexive Relation  $\rho$  über einer endlichen Eckenmenge  $E = \{0, \dots, n-1\}$  mit Adjazenzmatrix  $A$

$\sigma^{(k)}$  bezeichne die Relation

$$\sigma^{(k)} = \{(i,j) \mid \text{es gibt einen Weg } i \rightarrow e_1 \rightarrow \dots \rightarrow e_{l-1} \rightarrow j, \\ l \leq k \text{ und } e_r \in \{0, \dots, k\} \text{ für } 1 \leq r \leq l-1\}$$

- Beispiel:



$\sigma^{(0)}$	$\begin{matrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{matrix}$	$\sigma^{(1)}$	$\begin{matrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{matrix}$
$\sigma^{(2)}$	$\begin{matrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{matrix}$	$\sigma^{(3)}$	$\begin{matrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{matrix}$

### Information 10: Berechnung der reflexiven transitiven Hülle

Durch den Parameter  $k$  werden alle Wege, die in dem durch die Relation  $\rho$  vorgegebenen Graphen bestehen, sukzessive gefunden. Wie die Definition der zu diesem Zweck eingeführten Relation  $\sigma$  verdeutlicht, wird die Suche durch  $k$  in den folgenden zwei Punkten eingeschränkt:

1. Durch  $l \leq k$  wird erreicht, dass der Weg von einem Anfangs- zu einem Endknoten nur aus  $k$  Zwischenknoten bestehen darf (Weglängenbegrenzung).
2. Die Festlegung  $e_r \in \{0, \dots, k\}$  besagt, dass als Zwischenknoten nur die Knoten zwischen 0 und höchstens  $k$  auftreten dürfen (Knotennummernbegrenzung).

Anhand des einfachen Beispiel-Graphen wird in Information 10 der schrittweise Aufbau der Relation entlang des Wertes  $k$  dargestellt.

[www.cm-tm.uka.de/info1](http://www.cm-tm.uka.de/info1)  
Info1-Team (Prof. Abeck)

- Anforderungsbeschreibung
  - Eingabe: Adjazenzmatrix  $A$  einer Relation  $\sigma$
  - Ausgabe: Adjazenzmatrix  $S$  von  $\sigma^*$
- Algorithmusbeschreibung in Pseudo-Code

$S := A$

für  $i = 0, \dots, n-1$  setze  $s_{ii} := 1$

für  $k = 0, \dots, n-1$

  für  $i = 0, \dots, n-1$

    für  $j = 0, \dots, n-1$

      setze  $s_{ij} := s_{ij} + s_{ik} * s_{kj}$

### Information 11: Warshall-Algorithmus

Die Umsetzung dieses Berechnungs-Vorgehens auf entsprechende auf der Adjazenzmatrix wirkende elementare Operationen zeigt Information 11. Der Algorithmus ist zwar einfach in der

Aufschreibung, aber aufgrund der drei geschachtelten Schleifen komplex im Hinblick auf dessen Berechnungsdauer.

Der in Pseudo-Code angegebene Warshall-Algorithmus lässt sich mittels der in den Kurseinheiten PROGRAMMIERGRUNDLAGEN [C&M-PG] und IMPERATIVE PROGRAMMIERUNG [C&M-IP] unmittelbar in ein Java-Programm umsetzen.

## 2 ALGEBREN

In diesem Kapitel werden die bislang eingeführten Strukturen verallgemeinert, um dadurch die mathematische Grundlage für die Beschreibung der in der Informatik relevanten Systeme zu legen. Hieraus resultieren die Algebren, zu denen Information 12 einen Überblick liefert [Go95].

[www.cm-tm.uka.de/info1](http://www.cm-tm.uka.de/info1)  
Info1-Team (Prof. Abeck)

- In der Informatik werden Algebren zur Beschreibung von Datenstrukturen und Systemen genutzt
  - durch Ausführung einer Operation wird ein System von einem Zustand in einen Folgezustand überführt
- Unterschied zwischen den in der Mathematik und in der Informatik betrachteten Algebren
  - die Operationen in Informatiksystemen sind oft wesentlich komplizierter als die Operationen in Gruppen, Ringen, Körpern , ...
- Bei einer algebraischen Formel ist zu unterscheiden zwischen
  - dem Aufbau der Formel
  - dem berechneten Ergebnis
- Eine für die Informatik wichtige Algebra ist die Boolesche Algebra

### Information 12: ALGEBREN - Überblick

Ein Beispiel eines zu beschreibenden Informatiksystems ist das in der Kurseinheit INFORMATIK I IM ÜBERBLICK [C&M-IÜ] eingeführte Ablagesystem. Operationen hierauf sind beispielsweise das Anmelden oder das Abrufen von Inhalten. Das Beispiel zeigt bereits, dass die in der Informatik betrachteten Operationen einen im Vergleich zu den aus der Mathematik bekannten Operationen vielfältiger und höherwertiger sind.

Die Formel, die im Allgemeinen als Term bezeichnet wird, ist in einer Algebra ein zentraler Begriff.

Die Menge der aus einer Algebra hervorgehenden Terme bildet ebenfalls eine Algebra, die so genannte Termalgebra.

Zwischen diesen beiden Ebenen liegt eine weitere in der Informatik zu berücksichtigende Ebene der Realisierung. Ein Algorithmus liefert neben dem Ergebnis auch den Lösungsweg, der zum Ergebnis führt. Dabei ist dieser Weg nicht zwingend eindeutig, da Gesetze (wie z.B. das Distributivgesetz) alternative Vorgehensweisen ermöglichen. Außerdem führen auch unterschiedliche Repräsentationen (z.B. Graph als Adjazenzmatrix oder Adjazenzliste) zu

weiteren Alternativen der Realisierung. Daraus ergibt sich die Notwendigkeit, zwischen der Spezifikation eines Algorithmus und dessen tatsächlicher Realisierung sauber zu trennen.

Für die Informatik hat die Boolesche Algebra, auf die in diesem Kapitel näher eingegangen wird, eine besondere Bedeutung.

## 2.1 Formeln

Formeln werden gebildet auf der Grundlage von Operationen und der Signatur einer Algebra (siehe Information 13).

[www.cm-tm.uka.de/info1](http://www.cm-tm.uka.de/info1)  
Info1-Team (Prof. Abeck)

- Abbildung  $f: A^n \rightarrow A$   $f(a_1, \dots, a_n)$ 
  - n-stellige Operation auf A
  - Operanden  $a_i$  heißen Argumente der Operation
  - n heißt Stelligkeit von f
    - n=1: unäre Operation
    - n=2: binäre Operation
- Beispiel: Potenzmenge  $A = \mathcal{P}(U)$ 
  - U: Grundmenge
  - Operationen im booleschen Verband über  $\mathcal{P}(U)$  :
    - unär:  $h(M) = \complement M$
    - binär:  $f(M, N) = M \cup N$   $g(M, N) = M \cap N$
- Signatur  $\Sigma = \Sigma^{(0)} \cup \Sigma^{(1)} \cup \Sigma^{(2)} \cup \dots$ 
  - $f \in \Sigma^{(n)}$  wenn n die Stelligkeit von f ist

### Information 13: Operationen, Signatur

Um eine Abbildung definieren zu können, wird eine Menge A von Elementen benötigt. Aus dieser Menge werden n Elemente herausgegriffen und auf ein Element dieser Menge abgebildet.

Die Elemente, auf der die Abbildung arbeitet, heißen Argumente. Die Anzahl der Elemente heißt die Stelligkeit der Abbildung. Einstellige bzw. zweistellige Abbildungen werden als unäre bzw. binäre Operationen bezeichnet.

Beispiele für solche unären bzw. binären Operationen liefert der boolesche Verband über der Potenzmenge  $\mathcal{P}(U)$  der Grundmenge U. Die Potenzmenge  $\mathcal{P}(U)$  beinhaltet sämtliche Teilmengen aus der Grundmenge U (inklusive der leeren Menge). Hier ist also jeweils eine Teilmenge ein Argument der Abbildungen. Beispiel einer einstelligen, also unären Operation ist die Komplementbildung. Beispiele für zweistellige Operationen im booleschen Verband sind die Vereinigung und der Durchschnitt zweier Mengen. Diese Operationen lassen sich auf n-stellige Operationen ausdehnen.

Mit der Signatur wird eine Notation zur Beschreibung solcher Strukturen, die aus Mengen und darauf definierten Abbildungen bzw. Operationen entstehen, eingeführt. Die Signatur  $\Sigma$  ist eine nach Stelligkeit geordnete Auflistung der Operationen.

- Boolesche Algebra
  - $B = B(A, \perp, \top, \complement, \vee, \wedge)$
- Zugehörige Signatur
  - $\Sigma = (\perp, \top, \complement, \vee, \wedge)$
  - $\Sigma^{(0)} = \{\perp, \top\}$      $\Sigma^{(1)} = \{\complement\}$      $\Sigma^{(2)} = \{\vee, \wedge\}$
- $f^{(n)}$  oder  $f/n$  ist eine Kurzschreibweise von  $f \in \Sigma^{(n)}$
- Operationen aus  $\Sigma^{(0)}$ 
  - haben keine Argumente und liefern daher immer dasselbe Resultat  $op = c$
  - diese Operationen heißen daher Konstante
  - es muss mindestens eine solche Operation vorhanden sein

#### Information 14: Beispiel einer Signatur

Ein wichtiges Beispiel einer Algebra, zu der in Information 14 die Signatur angegeben wird, ist die Boolesche Algebra. Dabei bezeichnet  $A$  die Menge der Bezeichnungen, die in den booleschen Ausdrücken auftreten dürfen.

Die danach folgenden Operationen lassen sich gemäß ihrer Stelligkeit in disjunkte Operationenmengen aufteilen. Die Notation  $\Sigma^{(i)}$  liefert eine disjunkte Zerlegung der Operationen der Signatur, wobei  $i$  die Stelligkeit der auftretenden angibt.

Im ersten Moment erscheint eine nicht endliche Anzahl von Operationen nicht sinnvoll; dieser Sachverhalt zielt auf die konstanten Operationen  $\Sigma^{(0)}$  ab, die unten näher betrachtet werden.

$\mathcal{P}$

Im Beispiel sind die 0-stelligen Operationen der booleschen Algebra die Konstanten  $\perp$  ("Bottom") und  $\top$  ("Top"), als einzige 1-stellige Operation tritt  $\complement$  ("Komplement") auf und die zwei zweistelligen Operationen sind  $\wedge$  ("und") und  $\vee$  ("oder").

Für  $f \in \Sigma^{(n)}$  kann eine vereinfachende Kurzschreibweise verwendet werden, wie in Information 14 ausgeführt ist.

Die 0-stelligen "Operationen", also  $\Sigma^{(0)}$ , haben keine Argumente, weshalb sie immer dasselbe Ergebnis liefern. Sie werden als Konstanten bezeichnet und sind innerhalb einer Formel bzw. eines Terms elementare, nicht weiter zerlegbare Operanden. Warum die Menge der Konstanten nicht leer sein darf, wird anhand der in Interaktion 11 vorgestellten Bildungsgesetze für Terme deutlich.



- $(M \wedge \top) \vee \mathbb{C} P$ 
  - ist ein Beispiel einer Formel zur Signatur  $\Sigma = (\perp, \top, \mathbb{C}, \vee, \wedge)$
  - heißt auch Term oder Ausdruck
  - Operationen: \_\_\_\_\_ Operanden: \_\_\_\_\_
- Ein (korrekter) Term ist
  - (1) entweder eine Konstante  $a \in X$
  - (2) oder die Anwendung eines Operators  $f(a,b,\dots)$  mit  $f$ , wobei die Anzahl der Operanden  $a,b,\dots$  der Stelligkeit von  $f$  entspricht und jeder Operand wieder ein korrekter Term ist
- Unterterm
  - Term, der als Argument eines anderen Terms auftritt
- n-stelliger Term
  - Term, der durch Anwendung einer n-stelligen Operation entsteht

### Interaktion 11: Formel und Term

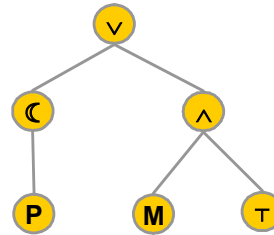
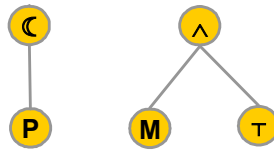
Auf der Basis der Signatur kann jetzt der Begriff der Formel eingeführt werden. Formeln bestehen aus nicht weiter zerlegbaren Operanden, die durch die in der Signatur angegebenen Operationen unter Berücksichtigung von deren Stelligkeit verknüpft sind. Bei der in Interaktion 11 angegebenen Beispiel-Formel ist zu beachten, dass das Element  $\top$  nicht als (0-stellige) Operation, sondern als elementarer Operand angesehen wird.

Für den Begriff der Formel existieren noch weitere Begriffe wie Term und Ausdruck, die zunächst alle unterschiedslos verwendet werden. In der Beispielformel hat der elementare, nicht weiter zerlegbare Operand  $\top$  die Bedeutung, dass er das größte Element bezeichnet.

Es gibt zwei Regeln (siehe Interaktion 11), gemäß derer ein korrekter Term (bzw. eine korrekte Formel) gebildet werden kann. Im Englischen werden solche Ausdrücke als *well-formed formula* (bzw. *term*) bezeichnet.

Diese als Operanden enthaltenen Terme werden auch als Unterterme bezeichnet. Ein Term, der auf  $n$  solchen Untertermen und einer  $n$ -stelligen Operation aufgebaut ist, wird entsprechend als  $n$ -stelliger Term bezeichnet.

- n-stellige Operation  $f$  wird als Baum der Höhe 1 dargestellt
  - Wurzel wird mit dem Operationssymbol bezeichnet
  - wird als Kantorowitsch-Baum bezeichnet
- Zusammenfügen der einzelnen Operations-Bäume



Warum gilt die Baum-Eigenschaft?

---



---

### Interaktion 12: Termdarstellung als Kantorowitsch-Baum

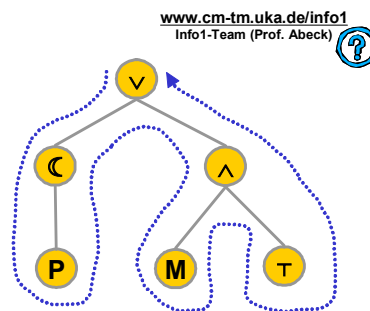
Terme lassen sich auch graphisch in Form von so genannten Kantorowitsch-Bäumen darstellen. Der Aufbau dieser Bäume erfolgt entlang des Termaufbaus beginnend bei den Konstanten, die einzelne Ecken darstellen und den Operationen, die als Kantorowitsch-Bäume der Höhe 1 dargestellt werden.

Zur Darstellung komplexerer Terme, wie der zuvor eingeführten Beispiel-Term

$$(M \wedge T) \vee C P$$

sind Kantorowitsch-Bäume zusammen zu setzen, wie in Interaktion 12 gezeigt wird. Gemäß den Ausführungen im Kapitel zu den RELATIONEN (siehe Interaktion 7), erfüllen die Kantorowitsch-Bäume die an Bäume gestellten Eigenschaften, wie in Interaktion 12 näher ausgeführt werden soll.

- Infix  $a f b$ 
  - nur für binäre Operationen
  - z.B.  $a+b$ ,  $M \cap N$
  - beim Durchlauf durch den Kantorowitsch-Baum werden die Ecken bei deren vorletzten Besuch angeschrieben (bzw. beim ersten Besuch, falls nur einmal besucht)
- Präfix  $f a b$ 
  - Operationssymbol vorne
  - z.B.  $+ 2 3$ ,  $C \cap N$
  - Anschreiben beim ersten Besuch
- Postfix  $a b f$ 
  - Operationssymbol hinten
  - z.B.  $A B \text{ union}$
  - Anschreiben beim letzten Besuch



Die gepunktete Linie beschreibt den Durchlauf durch der Kantorowitsch-Baum

Infix: \_\_\_\_\_

Präfix: \_\_\_\_\_

Postfix: \_\_\_\_\_

### Interaktion 13: Schreibweisen für Operationen

Der Kantorowitsch-Baum ist auch dazu nützlich, die verschiedenen in Interaktion 13 aufgezeigten Schreibweisen für Operationen zu verdeutlichen. Jede der drei genannten wichtigsten Schreibweisen Infix, Präfix und Postfix entspricht einer Strategie, gemäß der die Ecken beim Durchlauf durch den Kantorowitsch-Baum aufgeschrieben werden.

[www.cm-tm.uka.de/info1](http://www.cm-tm.uka.de/info1)  
Info1-Team (Prof. Abeck)

- Zwei weitere Schreibweisen
  - Funktionsform
    - ähnlich der Präfix-Notation
    - z.B.  $\text{union}(X, Y)$
  - Spezialnotation
    - nur für unäre Operationen
    - z.B.  $\bar{E}$  (Komplement)
- Lösungen zur Aufhebung von Mehrdeutigkeiten
  - Vorrangregeln zwischen Operatoren
  - Klammerung

### **Information 15: Weitere Schreibweisen und Vorrangregeln**

Daneben bestehen mit der Funktionsform und den Spezialnotationen noch zwei weitere Schreibweisen (siehe Information 15). Die Funktionsform weist bzgl. der Stellung des Operationssymbols Ähnlichkeiten mit der Präfix-Form auf. Die Spezialnotation (wozu beispielsweise auch die Wurzelschreibweise gehört) tritt nur bei unären Operatoren auf.

Ein Beispiel für eine Vorrangregel ist, dass  $*$  stärker bindet als  $+$ . Zur Auflösung von Mehrdeutigkeiten durch Klammern werden in der Informatik ausschließlich die runden Klammern benutzt, da anderen Klammerformen (z.B. geschweift oder eckig) für andere Zwecke genutzt werden.

## **2.2 Boolesche Algebra**

Die boolesche Algebra ist ein vollständiger, komplementärer, distributiver Verband. In Information 16 sind alle Gesetze, die in dieser Algebra mit der angegebenen Signatur gelten, aufgeführt.

- Signatur der booleschen Algebra  $B = B(A, \perp, \top, \complement, \vee, \wedge)$ 
  - $\perp$  ist kleinstes Element und  $\top$  ist größtes Element
- Gesetze ( $x, y, z \in A$ )
 

V1 Assoziativität	$(x \wedge y) \wedge z = x \wedge (y \wedge z)$	$(x \vee y) \vee z = x \vee (y \vee z)$
V2 Kommutativität	$x \wedge y = y \wedge x$	$x \vee y = y \vee x$
V3 Idempotenz	$x \wedge x = x$	$x \vee x = x$
V4 Verschmelzung	$(x \vee y) \wedge x = x$	$(x \wedge y) \vee y = y$
V5 Distributivität	$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$	$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$
V6 Modularität	falls $x \leq z$ gilt: $x \wedge (y \vee z) = (x \wedge y) \vee z$	
V7 Neutrales Element	$x \wedge \perp = \perp$	$x \vee \perp = x$
	$x \wedge \top = x$	$x \vee \top = \top$
V8 Komplement	$x \wedge \complement x = \perp$	$x \vee \complement x = \top$
V9 Involution	$\complement(\complement x) = x$	
V10 DeMorgan	$\complement(x \wedge y) = \complement x \vee \complement y$	$\complement(x \vee y) = \complement x \wedge \complement y$

### Information 16: Signatur und Gesetze der booleschen Algebra

Wegen V9 ist jedes Element  $x \in A$  ein Komplement eines anderen Elements  $x' = \complement x$ . Hieraus resultiert das in der booleschen Algebra geltende Dualitätsprinzip.

Durch die zehn aufgeführten Vorschriften ist die boolesche Algebra überspezifiziert, d.h. man kann auch gewisse Gesetze weglassen, da sich diese zwangsläufig aus den übrigen Gesetzen ableiten lassen. So kann man beispielsweise zu den Vorschriften V1 bis V5 entweder jeweils das erste oder jeweils das zweite Gesetz ersatzlos streichen.



- Die Mengenalgebra ist ein bekanntes Beispiel einer booleschen Algebra
  - Gegeben: eine beliebige Menge  $U$
  - $(\mathcal{P}(U), \cup, \cap, \complement)$  ist eine boolesche Algebra und heißt Mengenalgebra, wenn  $\cup$  die Mengenvereinigung,  $\cap$  der Mengendurchschnitt und  $\complement$  das Mengenkomplement ist
- $\mathcal{L}$  ist ein Mengensystem, das als eine Teilmenge (von Mengen) aus der Potenzmenge  $\mathcal{P}(U)$  hervorgeht
  - $\mathcal{L}$  ist eine Unteralgebra, falls diese die an eine Algebra gestellte Anforderung erfüllt
  - Welche Anforderung muss  $\mathcal{L}$  erfüllen?

---
- Bei endlicher Grundmenge  $A$  sind die Begriffe boolesche Algebra und Mengenalgebra äquivalent
  - formuliert im Satz von Stone

### Interaktion 14: Mengenalgebra und Unteralgebra $\mathcal{L}$

Das wohl bedeutendste und bekannteste Beispiel einer booleschen Algebra ist die Mengenalgebra, die auf der Potenzmenge einer beliebigen endlichen Menge definiert ist und der

die bekannten Mengenoperationen (Vereinigung, Durchschnitt, Komplement) zugrunde gelegt sind.

Eine Unteralgebra  $\mathcal{L}$  wird dadurch gebildet, dass aus der Potenzmenge (Menge aller Teilmengen) nur gewisse dieser Teilmengen herausgegriffen werden. Diese Teilmenge muss die an eine Algebra gestellte Anforderung erfüllen, nach der in Interaktion 14 gefragt wird.

Im Satz von Stone ist der Übergang von der booleschen Algebra zur Mengenalgebra und umgekehrt durch bijektive Abbildungen festgelegt.

[www.cm-tm.uka.de/info1](http://www.cm-tm.uka.de/info1)  
Info1-Team (Prof. Abeck)

- Betrachtet wird die Mengenalgebra (boolesche Algebra) über einer einelementigen Grundmenge  $U = \{a\}$ 
  - $\mathcal{P}(U) = \{\emptyset, \{a\}\}$  ist zweielementig
- Diese boolesche Algebra wird mit  $\mathcal{B} = \{0, L\}$  bezeichnet
  - 0 heißt Nullelement, L heißt Einselement
- $\mathcal{B}$  bildet die Grundlage der Aussagenlogik und aller digitalen Codierungen und Schaltungen
- Kartesisches Produkt  $\mathcal{B}^n = \{0, L\}^n = \mathcal{B} \times \dots \times \mathcal{B}$ 
  - Elementweises Ausführen der Operationen auf den n-Tupeln  $(a_1, \dots, a_n)$ ,  $a_i \in \{0, L\}$  ergibt, dass auch  $\mathcal{B}^n$  eine boolesche Algebra ist

### Information 17: Spezielle boolesche Algebra

Die Potenzmenge einer 1-elementigen Grundmenge besteht aus

- (1) dem Nullelement, also der leeren Menge  $\emptyset$ ,
- (2) dem Einselement  $\{a\}$ , also der gesamten Grundmenge.

Aufgrund dieser speziellen Eigenschaft und der großen Bedeutung, die gerade diese Ausprägung einer booleschen Algebra in der Informatik hat, wird hierfür eine spezielle Bezeichnung  $\mathcal{B}$  eingeführt.

Konkret bildet diese zweiwertige boolesche Algebra die Grundlage für die in der Kurseinheit RECHENSTRUKTUREN [C&M-RS] behandelte Aussagenlogik sowie für alle digitalen Codierungen und Schaltungen.

Die Operation des Kartesischen Produkts führt bei Anwendung auf die zweiwertige boolesche Algebra  $\mathcal{B}$  zu n-Tupeln, wobei diese Konstruktion  $\mathcal{B}^n$  wieder eine boolesche Algebra ergibt.

Abschließend wird in Interaktion 15 eine wichtige Normalform zu Termen der booleschen Algebra vorgestellt.



- Ziel ist die standardisierte Darstellung von booleschen Termen
- Disjunktive Normalform
  - durch Disjunktionen ( $\vee$ ) verknüpfte Teilterme
  - Teilterme bestehen aus durch Konjunktionen ( $\wedge$ ) verknüpften negierten oder nicht-negierten Elementen
  - Beispiel:  $(a \wedge b \wedge c) \vee (\neg a \wedge \neg b \wedge c) \vee (\neg a \wedge b \wedge \neg c) \vee (a \wedge \neg b \wedge \neg c)$
- Konjunktive Normalform
  - Konjunktion von Teiltermen, die aus disjunktiv verknüpften negierten oder nicht-negierten Elementen zusammengesetzt sind
  - Beispiel:  $(a \vee b \vee c) \wedge (\neg a \vee \neg b \vee c) = f(a, b, c)$

Wertetabelle:

a	0	0	0	0	1	1	1	1
b	0	0	1	1	0	0	1	1
c	0	1	0	1	0	1	0	1
f								

### Interaktion 15: Disjunktive und konjunktive Normalform

Die disjunktive Normalform erhält ihren Namen aufgrund der durch Disjunktionen (also durch die  $\vee$ -Operationen) verbundenen Teiltermen. Die von der Normalform zu erfüllende Anforderung steckt dabei in der scharfen Restriktion der disjunktiv verknüpften Teilterme. Es wird nämlich von jedem Term verlangt, dass

- ausschließlich die Konjunktion (also die  $\wedge$ -Operation) sowie die Negation auftreten darf und
- keine weiteren geklammerten Teilterme enthalten sein dürfen, d.h. der Teilterm ausschließlich aus Elementen aufgebaut sein darf.

Die konjunktive Normalform ist die duale Form der disjunktiven Normalform, d.h. es handelt sich um eine Konjunktion von Teiltermen, die aus disjunktiv verknüpften negierten oder nicht-negierten Elementen zusammengesetzt sind.

Die Beispiele verdeutlichen den Aufbau einer disjunktiven bzw. konjunktiven Normalform. Anhand der auszufüllenden Wertetabelle kann man sich den Zusammenhang zwischen den Normalformen und dem Werteverlauf von "normalisierten" booleschen Termen klar machen.

Algebren entsprechen in der Informatik den Rechenstrukturen. Dieser Zusammenhang wird in der Kurseinheit RECHENSTRUKTUREN [C&M-RS] aufgegriffen und am Beispiel verschiedener für die Informatik besonders wichtigen Algebren verdeutlicht.

## VERZEICHNISSE

### Abkürzungen und Glossar

<b>Abkürzung oder Begriff</b>	<b>Langbezeichnung und/oder Begriffserklärung</b>
Adjazenzmatrix	Eine Matrix, durch die ein Graph repräsentiert wird. Die Zeilen- und Spaltenzahl entspricht dabei der Knotenzahl des Graphen. Die Zeilen und Spalten der Matrix sind mit den Ecken indiziert. Der Wert eines Adjazenzmatrix-Elements $(i, j)$ bestimmt, ob eine Kante zwischen Ecke $i$ und Ecke $j$ besteht (Wert 1) oder nicht (Wert 0).
Algebra	Ein Tripel bestehend aus (Trägermenge, Operationen, Gesetze).
Algebraische Abgeschlossenheit	Das Ergebnis der Anwendung einer Operation der Algebra auf Elemente der Trägermenge liefert wieder ein Element der Trägermenge.
Baum (ungerichtet)	Graph, der die Eigenschaften der Zyklenfreiheit und des Zusammenhängens erfüllt.
Boolesche Algebra	Eine für die Informatik besonders wichtige Algebra. Es handelt sich hierbei um einen vollständigen, komplementären, distributiven Verband.
Dualitätsprinzip	Prinzip, dass in einer Algebra gilt, falls jedes ihrer Gesetz doppelt auftritt. Das duale Gesetz erhält man durch Ersetzen einer Operation gegen sein duales Gegenüber.
Einselement	Ein Element $\varepsilon$ , für das gilt: (1) $\varepsilon \times a = a$ (2) $a = a \times \varepsilon$ Synonymer Begriff: Neutrales Element
Grad	Angabe zu einem Graphen, der die Anzahl der Kanten angibt, die von der Ecke ausgehen bzw. zu dieser Ecke führen.
Graph	Anschauliche Darstellung einer Relation in Form von Ecken und den bestehenden Beziehungen zwischen den Ecken. Die Begriffe 'Relation' und 'Graph' werden in der Informatik weitgehend synonym verwendet.
Halbgruppe	Eine spezielle Algebra mit der Konkatenation als einziger Operation, in der das Assoziativgesetz gilt.
Kantorowitsch- Baum	Baum, durch der den Aufbau eines Terms in Teilterme dargestellt wird.
Konkatenation	Bezeichnung einer Operation, durch die zwei Elemente (i.d.R. Zeichenketten) verkettet werden.
Monoid	Halbgruppe, in der es ein Einselement gibt.
Relation	Liefert den grundlegenden Formalismus, Beziehungen zwischen

	Gegenständen zu beschreiben.
Schlinge	Ein Zyklus der Länge 1.
UML	<i>Unified Modeling Language</i> Sprache, die aus graphischen Elemente zur semi-formalen Beschreibung von beliebigen Gegenständen (z.B. Software-Systeme oder Geschäftsbereiche) besteht.
Wald (ungerichtet)	Graph, der die Eigenschaft der Zyklensfreiheit, nicht aber die Eigenschaft des Zusammenhängens erfüllt.
Weg	Ein Kantenzug mit einer Folge von Kanten, bei denen die Eingangsecke der einen Kante mit der Ausgangsecke der anderen Kante übereinstimmen.
Zyklus	Ein spezieller Weg ist der Zyklus oder Kreis, der an der Ecke endet, an der er beginnt. Synonymer Begriff: Kreis

## Index

Adjazenzmatrix	16	Halbgruppe	3
Algebra	3	Kantorowitsch-Bäumen	24
algebraischen Abgeschlossenheit	3	Konkatenation	3
Bäume	12	Monoids	4
Dualitätsprinzip	26	Relation	6
Gödelnummerierung	4	Unified Modeling Language	2
Grad	10	Wald	12

## Informationen und Interaktionen

Information 1: ALGEBRAISCHE GRUNDLAGEN .....	2
Information 2: RELATIONEN - Überblick.....	2
Information 3: Halbgruppe.....	3
Information 4: Relationen und Graphen.....	6
Information 5: Graphische Darstellung eines endlichen gerichteten Graphen.....	8
Information 6: Weg und Zyklus.....	11
Information 7: Ausprägungen von markierten Graphen .....	14
Information 8: Repräsentation von Relationen und Graphen.....	16
Information 9: Adjazenzmatrix .....	17
Information 10: Berechnung der reflexiven transitiven Hülle .....	19
Information 11: Warshall-Algorithmus.....	19
Information 12: ALGEBREN - Überblick.....	20
Information 13: Operationen, Signatur .....	21
Information 14: Beispiel einer Signatur .....	22
Information 15: Weitere Schreibweisen und Vorrangregeln.....	25
Information 16: Signatur und Gesetze der booleschen Algebra.....	26
Information 17: Spezielle boolesche Algebra .....	27
Interaktion 1: Monoid .....	4
Interaktion 2: Monoid $U^*$ über Monoid $\Sigma^*$ .....	5
Interaktion 3: Gerichteter Graph .....	7

Interaktion 4: Ungerichteter Graph .....	8
Interaktion 5: Teilgraph und dualer Graph.....	9
Interaktion 6: Grad einer Ecke .....	10
Interaktion 7: Spezielle Graphen im Zusammenhang mit Zyklen.....	12
Interaktion 8: Gerichtete Wälder und Bäume.....	13
Interaktion 9: Markierter Graph .....	14
Interaktion 10: Reflexive transitive Hülle eines Graphen .....	18
Interaktion 11: Formel und Term .....	23
Interaktion 12: Termdarstellung als Kantorowitsch-Baum .....	24
Interaktion 13: Schreibweisen für Operationen.....	24
Interaktion 14: Mengenalgebra und Unteralgebra $\mathcal{L}$ .....	26
Interaktion 15: Disjunktive und konjunktive Normalform.....	28

## Literatur

- [Br98] Manfred Broy: Informatik – Eine grundlegende Einführung, Band 1: Programmierung und Rechstrukturen, Springer Verlag 1998.
- [C&M-GI] Cooperation&Management: GRUNDBEGRIFFE DER INFORMATIK, Kursdokument zur Vorlesung "INFORMATIK I", <http://www.cm-tm.uka.de/info1>, Universität Karlsruhe (TH), C&M (Prof. Abeck).
- [C&M-IP] Cooperation&Management: IMPERATIVE PROGRAMMIERUNG, Kursdokument zur Vorlesung "INFORMATIK I", <http://www.cm-tm.uka.de/info1>, Universität Karlsruhe (TH), C&M (Prof. Abeck).
- [C&M-IÜ] Cooperation&Management: INFORMATIK I IM ÜBERBLICK, Kursdokument zur Vorlesung "INFORMATIK I", <http://www.cm-tm.uka.de/info1>, Universität Karlsruhe (TH), C&M (Prof. Abeck).
- [C&M-PG] Cooperation&Management: PROGRAMMIERGRUNDLAGEN, Kursdokument zur Vorlesung "INFORMATIK I", <http://www.cm-tm.uka.de/info1>, Universität Karlsruhe (TH), C&M (Prof. Abeck).
- [C&M-RS] Cooperation&Management: RECHENSTRUKTUREN, Kursdokument zur Vorlesung "INFORMATIK I", <http://www.cm-tm.uka.de/info1>, Universität Karlsruhe (TH), C&M (Prof. Abeck).
- [Go95] Gerhard Goos: Vorlesungen über Informatik, Band 1: Grundlagen und funktionales Programmieren, Springer Verlag 1995.
- [Se02] Robert Sedgewick: Algorithmen, Addison-Wesley – Pearson Studium, 2002.