Goethe-Universität Frankfurt am Main
Institut für Informatik
Theorie komplexer Systeme

Diskrete Modellierung

Skript zur Vorlesung

Prof. Dr. Nicole Schweikardt

Version vom 26. November 2008

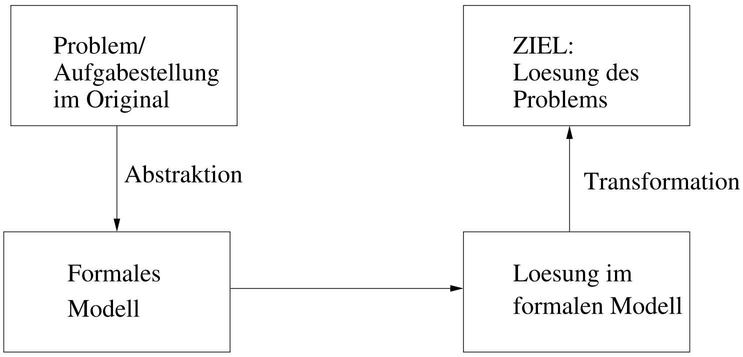
Inhaltsverzeichnis

1 Einführung ins Thema "Diskrete Modellierung"

1.1 Wozu diskrete Modellierung im Informatik-Studium?

Typische Arbeitsmethode in vielen Bereichen der Informatik: Modellierung von Problemen mittels diskreter Strukturen $\Longrightarrow$ präzise Beschreibung von Problemen – dies ist Grundvoraussetzung dafür, Probleme systematisch mit Methoden der Informatik lösen zu können.

Generelle Vorgehensweise in der Informatik:



Beispiel 1.1 (Problem "Flussüberquerung")**.**

Ein Mann steht mit einem Wolf, einer Ziege und einem Kohlkopf am linken Ufer eines Flusses, den er überqueren will. Er hat ein Boot, das gerade groß genug ist, ihn und ein weiteres Objekt zu transportieren, so dass er immer nur eines der drei mit sich hinübernehmen kann. Falls der Mann allerdings den Wolf mit der Ziege oder die Ziege mit dem Kohlkopf unbewacht an einem Ufer zurück lässt, wird einer gefressen. Ist es möglich, den Fluss zu überqueren, ohne dass die Ziege oder der Kohlkopf gefressen wird?

Lösungsansätze:

1. Knobeln $\rightarrow$ Lösung per "Geistesblitz"

2. Systematisches Vorgehen unter Verwendung von Informatik-Kalkülen

Hier: 2. Ansatz

Erste Analyse des Problems:

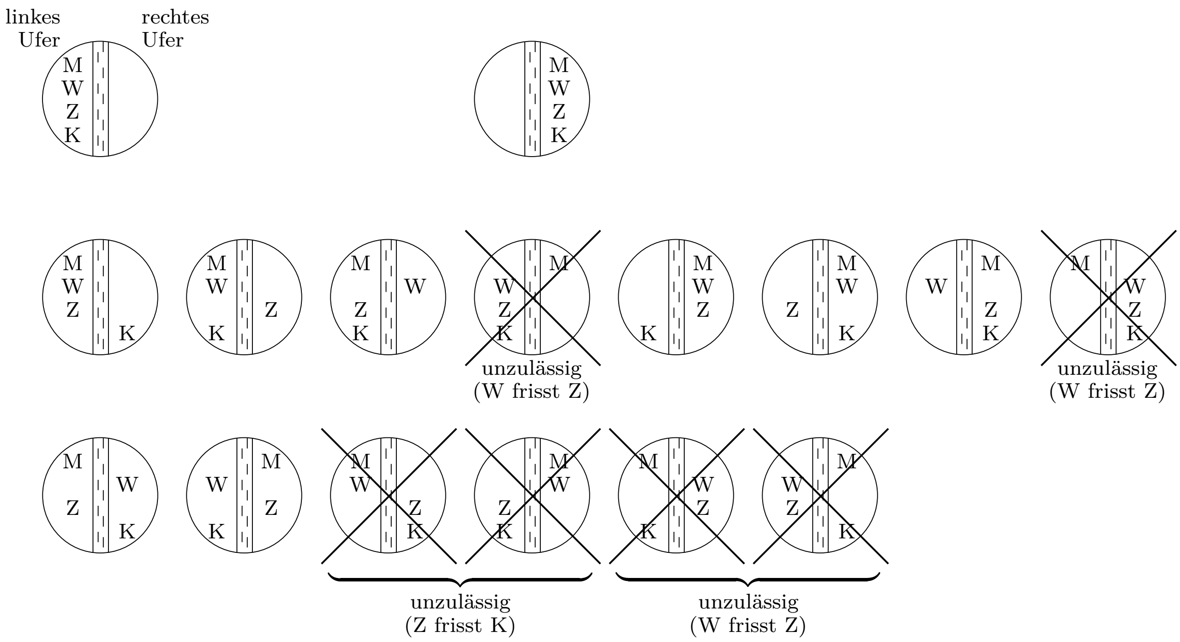
- relevante Objekte: Mann, Wolf, Ziege, Kohlkopf, Boot, Fluss, Ufer (links und rechts)
- Eigenschaften/Beziehungen:
  - Boot trägt Mann und zusätzlich maximal ein weiteres Objekt
  - Unbewacht am gleichen Ufer $\Longrightarrow$ Wolf frisst Ziege, Ziege frisst Kohlkopf
- Tätigkeit: Überqueren des Flusses
- Start: Mann, Wolf, Ziege, Kohlkopf, (Boot) am linken Ufer
- Ziel: Mann, Wolf, Ziege, Kohlkopf, (Boot) am rechten Ufer

Abstraktionen:

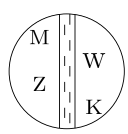
1. Nutze Abkürzungen:

   M $\widehat{=}$ Mann
   W $\widehat{=}$ Wolf
   Z $\widehat{=}$ Ziege
   K $\widehat{=}$ Kohlkopf

2. Betrachte die möglichen “Zustände”, die auftreten dürfen:



3. Formale Modellierung der “Zustände”: Repräsentiere den “Zustand”



durch das Tupel $(\{M, Z\}, \{W, K\})$.

Allgemein wird ein Zustand repräsentiert durch ein Tupel (l, r) mit $l \subseteq \{M, Z, W, K\}$ und $r \subseteq \{M, Z, W, K\}$, für das folgende Bedingungen erfüllt sind:

- $l \cup r = \{M, Z, W, K\}$
- $l \cap r = \varnothing$
- falls $Z, K \in l$, so auch $M \in l$ (um zu verhindern, dass K von Z gefressen wird)
- falls $Z, K \in r$, so auch $M \in r$ (um zu verhindern, dass K von Z gefressen wird)
- falls $W, Z \in l$, so auch $M \in l$ (um zu verhindert, dass Z von W gefressen wird)
- (*) falls $W, Z \in r$, so auch $M \in r$ (um zu verhindert, dass Z von W gefressen wird)

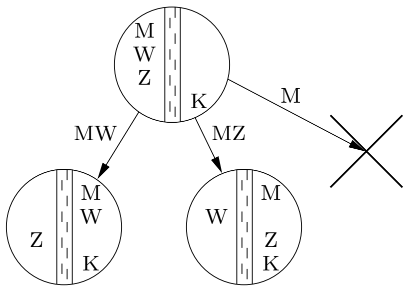
Übergänge von einem Zustand in einen anderen Zustand:

Vom Zustand $(\{M, W, Z\}, \{K\})$ aus kann man durch eine einzige Flussüberquerung in folgende Zustände gelangen:

- $(\{Z\}, \{M, W, K\})$, indem M und W im Boot fahren
- $(\{W\}, \{M, Z, K\})$, indem M und Z im Boot fahren

Beachte: wenn M allein fährt, tritt die Situation $(\{W, Z\}, \{M, K\})$ auf – dies ist aber laut (*) kein zulässiger Zustand.

Graphische Darstellung:



Insgesamt ergibt sich das in Abbildung 1.1 dargestellte Bild aus Zuständen und Zustandsübergängen.

Lösung des Problems "Flussüberquerung":

An diesem Bild lässt sich unser ursprüngliches Problem "Flussüberquerung" (Frage: Ist es möglich, den Fluss zu überqueren, ohne dass die Ziege oder der Kohlkopf gefressen werden?) ganz leicht lösen, indem man einfach einen Weg vom "Startzustand" zum "Zielzustand" sucht. Im Bild gibt es zwei verschiedene solche Wege; jeder davon kommt mit 7 Überfahrten aus.

Anmerkung:

Wir haben hier den Kalkül der **Transitionssysteme** (auch bekannt als **endliche Automaten** bzw. **Zustandsübergangsdiagramme** oder **Statecharts**) benutzt. Dieser Kalkül eignet sich besonders gut, wenn Abläufe in Systemen mit Übergängen zwischen verschiedenen Zuständen beschrieben werden sollen.

Mehr dazu: in einem späteren Kapitel dieser Vorlesung.

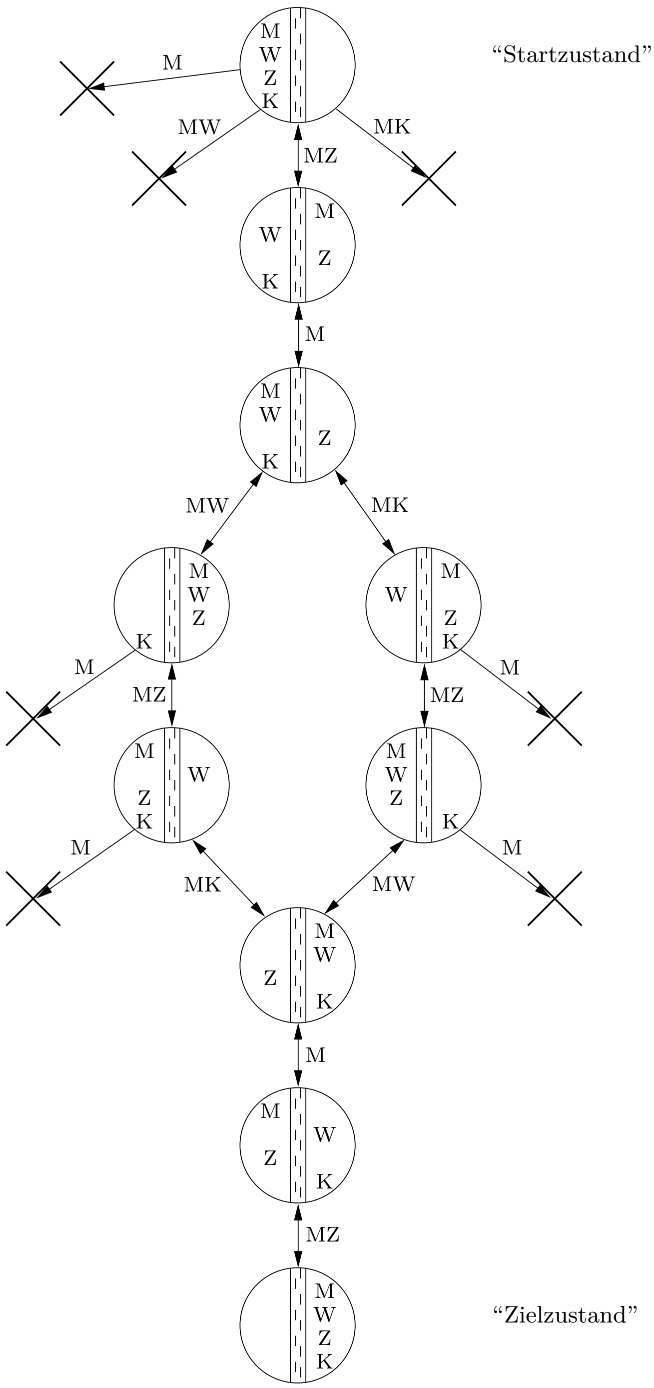


Abbildung 1.1: Übergänge zwischen den Zuständen beim Flussüberquerungsproblem

Diskussion dieses Modellierungsbeispiel:

- **Modellierung von Abläufen**/Folgen von Schritten durch ein Zustandsübergangsdiagramm
- **Abstraktion**: nur die Zustände und Übergänge interessieren
- **Relevante Objekte benannt**: M, W, Z, K
- Jeden **Zustand** haben wir hier **repräsentiert** durch **ein Paar** $(\mathbf{l}, \mathbf{r})$ **von Mengen** der Objekte (die sich am linken bzw. rechten Ufer befinden).
- Einteilung in **zulässige Zustände** und **unzulässige Zustände**
- Übergänge werden mit den transportierten Objekten beschriftet.

Besonders wichtig ist, was **nicht** modelliert wurde, weil es zur Lösung der Aufgabe irrelevant ist (z.B. Name, Breite, Tiefe des Flusses oder Länge, Geschwindigkeit des Boots etc.).

"Kreative Leistung": den Kalkül der Zustandsübergangsdiagramme wählen und die Bedeutung der Zustände und Übergänge festlegen.

Systematische Tätigkeit ("Routine-Arbeit"): das konkrete Zustandsübergangsdiagramm aufstellen und einen Weg vom Start- zum Zielzustand finden.

□Ende Beispiel 1.1

1.2 Ziele der Veranstaltung "Diskrete Modellierung"

- Überblick über grundlegende Modellierungsmethoden und -kalküle
- Verständnis des konzeptionellen Kerns der Kalküle
- Fähigkeit, Kalküle an typischen Beispielen anzuwenden
- Fähigkeit zur präzisen und formalen Ausdrucksweise bei der Analyse von Problemen
- Erkenntnis des praktischen Wertes präziser Beschreibungen

1.3 Allgemeiner Modellbegriff

Modell (lat.: modulus, "Maß, Maßstab"): allgemeines Muster, Vorbild, Entwurf Modell
Beispielweise:

- Abbild eines vorhandenen Originals (z.B. Schiffsmodell)
- Vorbild für ein herzustellendes Original (z.B. Gelände in kleinem Maßstab; Vorbild in der Kunst)
- Konkretes Modell (z.B. Schiffsmodell) oder abstraktes Modell (z.B. Rentenmodell)

Beachte:

- Modelle sind **absichtlich** nicht originalgetreu. Sie heben bestimmte Eigenschaften hervor und lassen andere weg.

Modellbegriff im Lexikon der Informatik

Modell → *Gegenstandsraum*

Modell (allgemeiner Begriff)
Teilgebiet: Modellierung
model (in general)

Während wir in den Formalwissenschaften wie Mathematik oder Physik einen präzisen Gebrauch des Wortes „Modell“ (→ *Gegenstandsraum*) vorfinden, wird das Modell-Denken in den Sozialwissenschaften weitgehend durch einen vagen Gebrauch des Ausdrucks „Modell“ gekennzeichnet. Folgende Begriffe, die sich in ihrer Intention oft stark unterscheiden, dürften die gebräuchlichsten Verwendungsweisen sein:

1. *Modell in der mathematischen Logik*
2. Modell als Bezeichnung für Theorien schlechthin
3. Modell als Resultat der Abbildung der Wirklichkeit.

Weitere Klassifizierungskriterien (→ *Klassifizierung*[2]) lassen sich nach dem Zweck, der mit den einzelnen Modellen verfolgt wird angeben (siehe Abb. S. 512).
Modell als Theorie schlechthin (2) findet sich häufig im verbalen Sprachgebrauch der Sozialwissenschaften. Insbesondere jene Teilklassen von Theorien, die mathematisiert, quantifiziert bzw. formalisiert sind, werden allgemein als Modell bezeichnet. Beispiele sind Preismodell, Rentenmodell.
Modelle als Abbild der Realität (3) stellen eine umfangreiche, sehr heterogene Klasse dar. Hierbei bilden die Beschreibungen ohne Verwendung einer Sprache, meist auf ein handliches Maß verkleinerten Nachbildungen eines vorgestellten Originals, die bekannteste Art von Modellen. Diese werden, wie z.B. der Globus, auch als ikonische oder materiale Modelle bezeichnet. *Stübel*

Modell in der mathematischen Logik
Teilgebiet: Logik
model

Es gibt zwei unterschiedliche Definitionen für Modelle der mathematischen *Logik*:

a) Eine Struktur Σ heißt Modell einer Formelmenge X, wenn jede Formel aus X in Σ gültig ist.
b) Das Paar (I, ζ), bestehend aus einer Interpretation I und einer *Belegung* ζ, heißt Modell einer Formelmenge X, wenn jede Formel aus X bei I und ζ wahr ist.

Für Mengen X von Aussagen, also Formeln ohne freie Variablen, sind beide Definitionen gleichwertig, da dann die Belegung keine Rolle spielt.
Die Modelltheorie beschäftigt sich mit gegenseitigen Beziehungen zwischen Aussagen formalisierter Theorien und mathematischen Strukturen, in denen die Aussagen gelten. *Müller; Stübel*

Modell, abstrakt symbolisches
Teilgebiet: Modellierung
abstract symbolic model

Eine vor allem in der Betriebswirtschaft sehr verbreitete Klasse von Modellen bilden die abstrakt symbolischen Abbilder eines Realitätskomplexes. Dabei kann es sich sowohl um rein verbale Reproduktionen eines Systems handeln als auch um ein künstliches Sprachsystem, das durch zunächst inhaltsleere symbolische Zeichen und syntaktische (→ *Syntax von Programmiersprachen*) Regeln gekennzeichnet ist. *Stübel*

aus
H-J. Schneider: Lexikon der Informatik und Datenverarbeitung, 3. Aufl., Oldenbourg Verlag, 1991

Abbildung 1.2: Modellbegriff im allgemeinen Lexikon

- Der intendierte Verwendungszweck des Modells bestimmt, welche Eigenschaften modelliert werden und welcher Kalkül dafür besonders geeignet ist.

Beispiel 1.2.

(a) Unterschiedliche Modelle, die beim Bau eines Hauses verwendet werden:
    - Gebäudemodell: zur Vermittlung eines optischen Eindrucks

Modell [italien., zu lat. modulus „Maß, Maßstab"], allg. Muster, Vorbild, Entwurf.
– Mensch (auch Tier), der (das) als Vorbild für künstler. Studien oder Kunstwerke dient („sitzt").
– in der Bildhauerei meist in verkleinerter Form ausgeführter Entwurf einer Plastik oder Tonarbeit, die in Bronze gegossen werden soll. - †Architekturmodell.
– in der Modebranche Bez. für 1. ein nur einmal oder in eng begrenzter Anzahl hergestelltes Kleidungsstück. (*M.kleid*); 2. die Vorlage für eine Vervielfältigung; 3. svw. Mannequin.
– im Sprachgebrauch verschiedener Wiss. (Philosophie, Naturwiss., Soziologie, Psychologie, Wirtschaftswiss., Politikwiss., Kybernetik u.a.) ein Objekt materieller oder ideeller (Gedanken-M.) Natur, das von einem Subjekt auf der Grundlage einer Struktur-, Funktions- oder Verhaltensanalogie für ein anderes Objekt (*Original*) eingesetzt und genutzt wird, um Aufgaben zu lösen, deren Durchführung unmittelbar am Original selbst nicht möglich bzw. zu aufwendig ist (z. B. Flugzeug-M. im Windkanal). Die **Modellmethode** vollzieht sich in vier Schritten: 1. Auswahl (Herstellung eines dem [geplanten] Original entsprechenden M.; 2. Bearbeitung des M., um neue Informationen über das M. zu gewinnen (**Modellversuch**; †Ähnlichkeitsgesetze); 3. Schluss auf Informationen über das Original (meist Analogieschluß); ggf. 4. Durchführung der Aufgabe am Original. Infolge der Relationen zw. Subjekt, Original und M. (**Modellsystem**) ist ein M. einsetzbar u. a. zur Gewinnung neuer Informationen über das Original (z. B. Atom-M.), zur Demonstration und Erklärung (z. B. Planetarium), zur Optimierung des Originals (z. B. Netzplan), zur Überprüfung einer Hypothese oder einer techn. Konstruktion (z. B. Laborversuch). - Abweichend von diesem M.begriff versteht die *mathemat. Logik* unter M. eine Interpretation eines Axiomensystems, bei der alle Axiome dieses Systems wahre Aussagen darstellen. Diese **Modelltheorie** liefert grundlegende Verfahren zur Behandlung von Fragen der Vollständigkeit, Widerspruchsfreiheit und Definierbarkeit.

Abbildung 1.3: Modellbegriff im Lexikon der Informatik

- Grundriss: zur Einteilung der Räume und des Grundstückes
- Kostenplan: zur Finanzierung

(b) Frankfurter S- und U-Bahn Netzplan: siehe Abbildung 1.4

Ziel: Beschreibung, welche Haltestellen von welchen Linien angefahren werden und welche Umsteigemöglichkeiten es gibt

Vernachlässigt: genauere topografische Informationen (Entfernung, genaue Lage, Straßenverläufe etc.), Abfahrtszeiten

(c) Fahrplan der U4 an der Haltestelle "Bockenheimer Warte": siehe Abbildung 1.5

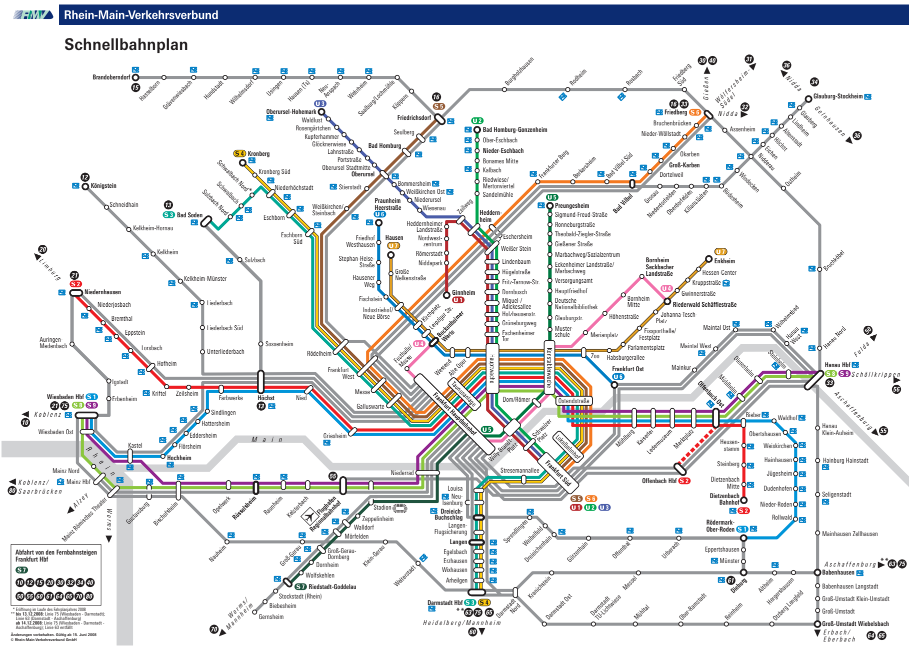


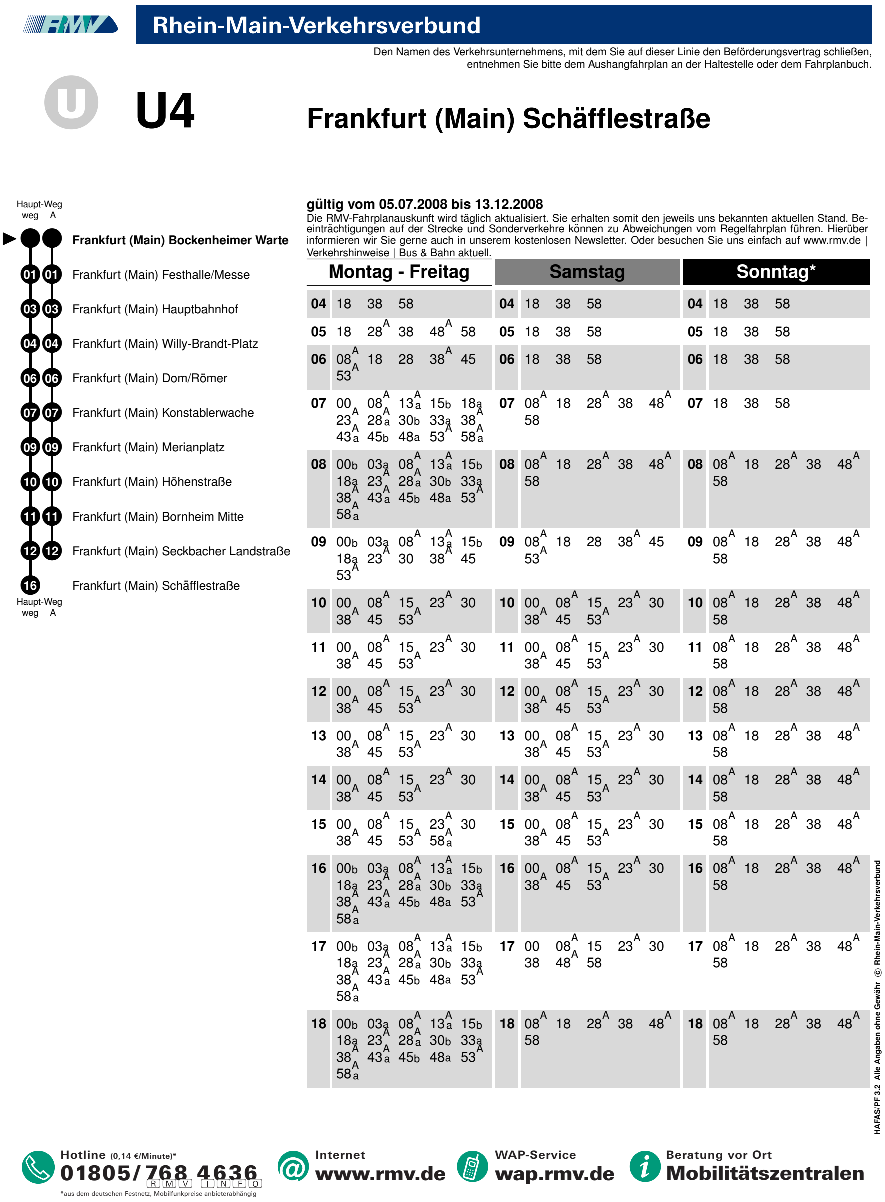
Abbildung 1.4: Schnellbahnplan des Rhein-Main-Verkehrsverbundes (RMV)

Ziel: Angabe der Abfahrtszeiten der U4 an der Haltestelle "Bockenheimer Warte" sowie Informationen darüber, wie viele Minuten die U4 von dort bis zu anderen Haltestellen auf ihrer Strecke braucht.

□Ende Beispiel 1.2

1.3.1 Arbeiten mit dem Modell

- Nutzung des Modells bei der Lösung eines Problems (z.B. Beispiel 1.1 "Flussüberquerung")
- Bestimmte Aspekte eines komplexen Gebildes untersuchen und verstehen (z.B. Geschäftsabläufe in einer Firma)
- Verständigung zwischen Auftraggeber und Hersteller des Originals (z.B. Hausbau, Software-Entwicklung)
- Fixieren von Anforderungen für die Herstellung des Originals (z.B. Software: Spezifikation, Anforderungen)
- Durchführungen von Operationen, die man am Original nicht durchführen kann (z.B. Erprobung einer neuen Flügelform im Windkanal oder durch Computer-Simulation)
- Validierung des Modells (engl. Model Checking); Nachweis, dass die relevanten Eigenschaften des Originals korrekt und vollständig im Modell erfasst sind (z.B. Prüfung, ob ein Finanzplan alle Kosten erfasst, sie korrekt aufsummiert und die vorgegebene Kostengrenze eingehalten wird)



Rhein-Main-Verkehrsverbund

Den Namen des Verkehrsunternehmens, mit dem Sie auf dieser Linie den Beförderungsvertrag schließen, entnehmen Sie bitte dem Aushangfahrplan an der Haltestelle oder dem Fahrplanbuch.

U4 Frankfurt (Main) Schäfflestraße

Haupt-Weg	A	
●	●	**Frankfurt (Main) Bockenheimer Warte**
01	01	Frankfurt (Main) Festhalle/Messe
03	03	Frankfurt (Main) Hauptbahnhof
04	04	Frankfurt (Main) Willy-Brandt-Platz
06	06	Frankfurt (Main) Dom/Römer
07	07	Frankfurt (Main) Konstablerwache
09	09	Frankfurt (Main) Merianplatz
10	10	Frankfurt (Main) Höhenstraße
11	11	Frankfurt (Main) Bornheim Mitte
12	12	Frankfurt (Main) Seckbacher Landstraße
16		Frankfurt (Main) Schäfflestraße

gültig vom 05.07.2008 bis 13.12.2008

Die RMV-Fahrplanauskunft wird täglich aktualisiert. Sie erhalten somit den jeweils uns bekannten aktuellen Stand. Beeinträchtigungen auf der Strecke und Sonderverkehre können zu Abweichungen vom Regelfahrplan führen. Hierüber informieren wir Sie gerne auch in unserem kostenlosen Newsletter. Oder besuchen Sie uns einfach auf www.rmv.de | Verkehrshinweise | Bus & Bahn aktuell.

	Montag - Freitag		Samstag		Sonntag*
04	18 38 58	**04**	18 38 58	**04**	18 38 58
05	18 28A 38 48A 58	**05**	18 38 58	**05**	18 38 58
06	08A 18 28 38A 45 53A	**06**	18 38 58	**06**	18 38 58
07	00 08A 13Aa 15b 18a 23A 28Aa 30b 33a 38A 43Aa 45b 48a 53A 58Aa	**07**	08A 18 28A 38 48A 58	**07**	18 38 58
08	00b 03a 08A 13Aa 15b 18a 23A 28Aa 30b 33a 38A 43Aa 45b 48a 53A 58Aa	**08**	08A 18 28A 38 48A 58	**08**	08A 18 28A 38 48A 58
09	00b 03a 08A 13Aa 15b 18a 23A 30 38A 45 53A	**09**	08A 18 28 38A 45 53A	**09**	08A 18 28A 38 48A 58
10	00 08A 15 23A 30 38A 45 53A	**10**	00 08A 15 23A 30 38A 45 53A	**10**	08A 18 28A 38 48A 58
11	00 08A 15 23A 30 38A 45 53A	**11**	00 08A 15 23A 30 38A 45 53A	**11**	08A 18 28A 38 48A 58
12	00 08A 15 23A 30 38A 45 53A	**12**	00 08A 15 23A 30 38A 45 53A	**12**	08A 18 28A 38 48A 58
13	00 08A 15 23A 30 38A 45 53A	**13**	00 08A 15 23A 30 38A 45 53A	**13**	08A 18 28A 38 48A 58
14	00 08A 15 23A 30 38A 45 53A	**14**	00 08A 15 23A 30 38A 45 53A	**14**	08A 18 28A 38 48A 58
15	00 08A 15 23A 30 38A 45 53A 58Aa	**15**	00 08A 15 23A 30 38A 45 53A	**15**	08A 18 28A 38 48A 58
16	00b 03a 08A 13Aa 15b 18a 23A 28Aa 30b 33a 38A 43Aa 45b 48a 53A 58Aa	**16**	00 08A 15 23A 30 38A 45 53A	**16**	08A 18 28A 38 48A 58
17	00b 03a 08A 13Aa 15b 18a 23A 28Aa 30b 33a 38A 43Aa 45b 48a 53A 58Aa	**17**	00 08A 15 23A 30 38 48A 58	**17**	08A 18 28A 38 48A 58
18	00b 03a 08A 13Aa 15b 18a 23A 28Aa 30b 33a 38A 43Aa 45b 48a 53A 58Aa	**18**	08A 18 28A 38 48A 58	**18**	08A 18 28A 38 48A 58

HAFAS/PF 3.2 Alle Angaben ohne Gewähr © Rhein-Main-Verkehrsverbund

Hotline (0,14 €/Minute)* 01805/768 4636 RMV INFO

*aus dem deutschen Festnetz, Mobilfunkpreise anbieterabhängig

Internet www.rmv.de

WAP-Service wap.rmv.de

Beratung vor Ort Mobilitätszentralen

Abbildung 1.5: Fahrplan der U4 an der Haltestelle “Bockenheimer Warte”

1.3.2 Modellierte Aspekte

Ein Modell beschreibt nur bestimmte Aspekte des Originals, z.B.

- Struktur/Zusammensetzung des Originals (z.B. Organisationsschema einer Firma)

Dafür geeignete Kalküle: Wertebereiche, Entity-Relationship-Modell, Bäume

- Eigenschaften von Teilen des Originals (z.B. Farbe und Wert einer Spielkarte)

  Dafür geeignete Kalküle: Wertebereiche, Logik, Entity-Relationship-Modell

- Beziehungen zwischen Teilen des Originals (z.B. "Wolf frisst Ziege, Ziege frisst Kohlkopf")

  Dafür geeignete Kalküle: Graphen, Logik, Entity-Relationship-Modell

- Verhalten des Originals unter Operationen (z.B. aufeinanderfolgende Zustände bei wiederholter Flussüberquerung)

  Dafür geeignete Kalküle: Zustandsübergangsdiagramme, Petri-Netze, Graphen

1.4 Übungsaufgaben zu Kapitel 1

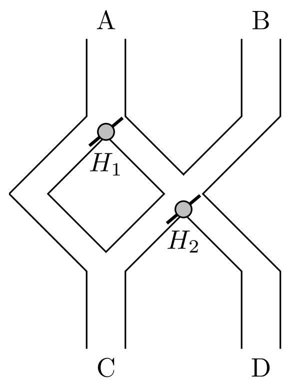
Aufgabe 1.1. Gegeben seien drei Stapel mit Büchern. Der erste besteht aus vier Büchern, der zweite aus sechs Büchern und der dritte aus 14 Büchern. Die Stapel sollen nun ausgeglichen werden, so dass auf jedem Stapel acht Bücher liegen. Allerdings dürfen in jedem Schritt nur Bücher zwischen genau zwei Stapeln umgeschichtet werden. Zudem können auf jeden Stapel immer nur so viele Bücher gelegt werden, wie bereits darauf liegen.

(a) Lassen sich die Stapel wie gewünscht ausgleichen? Modellieren Sie zur Beantwortung dieser Frage das Problem analog zum Beispiel 1.1.

(b) Nehmen wir nun an, dass der ersten Stapel aus vier Büchern, der zweite aus sechs Büchern und der dritte aus acht Büchern besteht. Lassen sich die Stapel so ausgleichen, dass auf jedem Stapel sechs Bücher liegen?

Hinweis: Es brauchen nur diejenigen Zustände betrachtet zu werden, die man vom Startzustand aus durch geeignete Zustandsübergänge erreichen kann.

Aufgabe 1.2. Die nachfolgende Abbildung zeigt ein Spiel, in dem Murmeln bei A oder B in die Spielbahn fallen gelassen werden.



Je nach Stellung der Hebel H_1 und H_2 rollen die Murmeln in der Spielbahn nach links oder rechts. Sobald eine Murmel auf einen dieser Hebel trifft, wird der Hebel nach dem Passieren der Murmel umgestellt, so dass die nächste Murmel in die andere Richtung rollt. Zu Beginn ist jeder der beiden Hebel so eingestellt, dass die nächste Murmel, die auf den Hebel trifft, nach links rollt. Wenn beispielsweise nacheinander drei Murmeln fallen gelassen werden, wobei die erste und dritte Murmel bei A und die zweite Murmel bei B fallen gelassen wird, dann kommen die ersten beiden Murmeln an der Öffnung C und die letzte Murmel an der Öffnung D heraus.

Aus welcher Öffnung fällt die letzte Murmel, wenn

(a) fünf Murmeln fallen gelassen werden, wobei die erste und die vierte Murmel bei A und alle anderen Murmeln bei B fallen gelassen werden?

(b) sieben Murmeln fallen gelassen werden, wobei die erste, zweite, vierte und letzte Murmel bei A und alle anderen Murmeln bei B fallen gelassen werden?

Modellieren Sie zur Beantwortung der Fragen das Spiel analog zu Beispiel 1.1 durch ein Transitionssystem. Überlegen Sie sich zunächst die möglichen Zustände und Zustandsübergänge, die auftreten können.

Hinweis: Jeder Zustand sollte Informationen darüber enthalten, in welche Richtung die nächste Murmel rollt, wenn sie auf H_1 trifft, in welche Richtung die nächste Murmel rollt, wenn sie auf H_2 trifft und aus welcher Öffnung die zuvor fallen gelassene Murmel herausgerollt ist.

2 Modellierung mit Wertebereichen – mathematische Grundlagen und Beweistechniken

Mathematische Notationen:

Symbol	Bedeutung
$:=$	Definition eines Wertes, z.B. $x := 5$, $M := \{1, 2, 3\}$
$:\Longleftrightarrow$	Definition einer Eigenschaft oder einer Schreibweise z.B. $m \in M \;:\Longleftrightarrow\; m$ ist Element von M
ex.	Abkürzung für “es gibt”, “es existiert”
f.a.	Abkürzung für “für alle”, “für jedes”
s.d.	Abkürzung für “so, dass”
$\Longrightarrow$	Abkürzung für “impliziert” z.B. Regen $\Longrightarrow$ nasse Straße
$\Longleftrightarrow$	Abkürzung für “genau dann, wenn” z.B. Klausur bestanden $\Longleftrightarrow$ $z \geq 50\%$
$\square$	markiert das Ende eines Beweises

Modellierung und Wertebereiche

In der Modellierung von Systemen/Aufgaben/Problemen/Lösungen kommen **Objekte unterschiedlicher Art und Zusammensetzung** vor. Für Teile des Modells wird angegeben, aus welchem Wertebereich sie stammen, aber manchmal offen gelassen, welchen konkreten Wert sie haben.

Beispiel. Gegeben 3 Karten aus einem Kartenspiel, welches ist die höchste Karte?

Wertebereich

Ein **Wertebereich** ist eine Menge gleichartiger Werte. Wertebereiche werden aus Mengen und Strukturen darüber gebildet.

Beispiel 2.1 (Modellierung der Karten eines (Skat-)Kartenspiels).
Wertebereiche:

$$\begin{aligned}
\text{KartenArten} &:= \{\text{Kreuz}, \text{Pik}, \text{Herz}, \text{Karo}\} \\
\text{KartenSymbole} &:= \{7, 8, 9, 10, \text{Bube}, \text{Dame}, \text{König}, \text{Ass}\} \\
\text{Karten} &:= \{(\text{Kreuz}, 7), (\text{Kreuz}, 8), \ldots, (\text{Kreuz}, \text{Ass}), \\
&\quad (\text{Pik}, 7), (\text{Pik}, 8), \ldots, (\text{Pik}, \text{Ass}), \\
&\quad (\text{Herz}, 7), (\text{Herz}, 8), \ldots, (\text{Herz}, \text{Ass}), \\
&\quad (\text{Karo}, 7), (\text{Karo}, 8), \ldots, (\text{Karo}, \text{Ass})\}
\end{aligned}$$

Übersicht über Begriffe, die in Kapitel 2 genauer betrachtet werden:

- Wertebereich: eine Menge gleichartiger Werte
- Grundlegender Kalkül: Mengenlehre (Mengen und Mengenoperationen)
- Strukturen über Mengen zur Bildung von zusammengesetzten Wertebereichen:
  - Potenzmengen
  - Kartesische Produkte, Tupel
  - Relationen
  - Folgen, Wörter
  - Funktionen
- Verwendung dieses Kalküls
  - Modellierung von Strukturen und Zusammenhängen
  - Grundlage für alle anderen formalen Kalküle
  - abstrakte Grundlage für Typen in Programmiersprachen

Ziel von Kapitel 2 ist, diese Begriffe genauer zu betrachten und abgesehen davon einige wichtige mathematische Grundlagen und Beweistechniken zu erklären.

2.1 Mengen, Relationen und Funktionen

2.1.1 Was ist eine Menge?

Cantors naiver Mengenbegriff (Georg Cantor, 1845–1918)
Eine Menge M ist eine Zusammenfassung von bestimmten, wohlunterschiedenen Objekten unserer Anschauung oder unseres Denkens, welche "Elemente der Menge M" genannt werden, zu einem Ganzen.

Notation. $m \in M \;:\Longleftrightarrow\; m$ ist Element der Menge M

Die Russellsche Antinomie (Bertrand Russell, 1872–1970)
Cantors Mengenbegriff ist problematisch und führt zu Widersprüchen. Russell gab folgendes Beispiel:

> Sei N die Menge aller Mengen M, die sich nicht selbst enthalten
> (d.h.: $M \in N \;:\Longleftrightarrow\; M$ ist eine Menge, für die gilt: $M \notin M$).

Frage: Enthält N sich selbst (d.h. gilt $N \in N$)?

Klar: Entweder gilt $N \in N$ oder $N \notin N$.

Fall1: $N \notin N$. Gemäß Definition der Menge N gilt dann, dass $N \in N$.
Das ist ein Widerspruch.

Fall2: $N \in N$. Gemäß Definition der Menge N gilt dann, dass $N \notin N$.
Das ist ein Widerspruch.

Somit führen beide Fälle zu einem Widerspruch, obwohl wir wissen, dass einer der beiden Fälle zutreffen müsste. $\Longrightarrow$ Irgendetwas stimmt nicht mit Cantors naivem Mengenbegriff!

Um Russells Beispiel und den daraus resultierenden Widerspruch besser zu verstehen, betrachte man folgende Geschichte vom Barbier von Sonnenthal.

Der Barbier von Sonnenthal:

Im Städtchen Sonnenthal (in dem bekanntlich viele seltsame Dinge passieren) wohnt ein Barbier, der genau diejenigen männlichen Einwohner von Sonnenthal rasiert, die sich nicht selbst rasieren.

Frage: Rasiert der Barbier sich selbst?

Um die Russellsche Antinomie zu vermeiden, muss man die Mengenlehre sehr vorsichtig axiomatisch aufbauen – dies sprengt allerdings den Rahmen dieser Vorlesung. Sofern man sich der Problematik aber bewusst ist, kann man sie im "täglichen Gebrauch" von Mengen vermeiden. Wir arbeiten daher weiter mit einem naiven Mengenbegriff.

2.1.2 Beschreibung/Definition von Mengen

- durch Aufzählen der Elemente (extensional), z.B.

$$M_1 := \{0, 1, 2, 3, 4, 5\} = \{0, 1, 2, \dots, 5\}$$

- durch Angabe von charakteristischen Eigenschaften der Elemente der Menge (intensional), z.B.

$$\begin{aligned} M_2 &:= \{x : x \in M_1 \text{ und } x \text{ ist gerade}\} \\ &= \{x \in M_1 : x \text{ ist gerade}\} \\ &= \{x : x \text{ ist eine natürliche Zahl und } x \text{ ist gerade und } 0 \leq x \leq 5\} \end{aligned}$$

  Extensional lässt sich M_2 folgendermaßen beschreiben:

$$M_2 = \{0, 2, 4\}.$$

  Oft schreibt man statt ":" auch "|" und statt "und" einfach ein "Komma", also $M_2 = \{x \mid x \in M_1,\ x \text{ gerade}\}$.

Vorsicht:

(a) $\{x : 0 \leq x \leq 5\}$ definiert nicht eindeutig eine Menge, weil nicht festgelegt ist, ob x beispielsweise eine ganze oder eine reelle Zahl ist.

(b) $\{M : M \text{ ist eine Menge}, M \notin M\}$ führt zur Russellschen Antinomie.

Fazit: Um solche Probleme zu vermeiden, sollte man bei intensionalen Mengendefinitionen immer angeben, aus welcher anderen Menge die ausgewählten Elemente kommen sollen, also:

$$\{x \in M : x \text{ hat Eigenschaft(en) } E\},$$

wobei M eine Menge und E eine Eigenschaft oder eine Liste von Eigenschaften ist, die jedes einzelne Element aus M haben kann oder nicht.

2.1.3 Wichtige grundsätzliche Eigenschaften von Mengen

- Alle Elemente einer Menge sind verschieden. D.h. ein Wert ist entweder Element der Menge oder eben nicht – aber er kann nicht "mehrfach" in der Menge vorkommen.

- Die Elemente einer Menge haben keine feste Reihenfolge.

- Dieselbe Menge kann auf verschiedene Weisen beschrieben werden, z.B.

$$\{1, 2, 3\} \;=\; \{1, 2, 2, 3\} \;=\; \{2, 1, 3\} \;=\; \{i : i \text{ ist eine ganze Zahl, } 0 \leq x \leq 3\}.$$

Insbesondere können Mengen aus atomaren oder aus zusammengesetzten Elementen gebildet werden, und eine Menge kann auch verschiedenartige Elemente enthalten.

Beispiel. Die Menge $M := \{1, (\text{Pik}, 8), \{\text{rot}, \text{blau}\}, 5\}$ besteht aus 4 Elementen:

- den atomaren Werten 1 und 5

- dem Tupel $(\text{Pik}, 8)$

- der Menge $\{\text{rot}, \text{blau}\}$

Notationen für bestimmte Zahlenmengen

$$\begin{array}{lll}
\mathbb{N} & := & \text{Menge der natürlichen Zahlen} := \{0, 1, 2, 3, ...\} \\
\mathbb{N}_{>0} & := & \text{Menge der positiven natürlichen Zahlen} := \{1, 2, 3, ...\} \\
\mathbb{Z} & := & \text{Menge der ganzen Zahlen} := \{0, 1, -1, 2, -2, 3, -3, ...\} \\
\mathbb{Q} & := & \text{Menge der rationalen Zahlen} := \{\frac{a}{b} : a, b \in \mathbb{Z},\, b \neq 0\} \\
\mathbb{R} & := & \text{Menge der reellen Zahlen}
\end{array}$$

Beobachtung 2.2. Es gibt genau eine Menge, die keine Elemente enthält.

Definition 2.3 (leere Menge)**.** Die **leere Menge** ist die (eindeutig bestimmte) Menge, die kein(e) Element(e) enthält. Wir bezeichnen sie mit $\varnothing$. leere Menge

2.4 Frage: Gibt es eine "Menge aller Mengen"?

Nein! Denn wäre U die Menge aller Mengen, so wäre auch $N := \{M \in U : M \notin M\}$ eine Menge. Dies führt aber wieder zur Russellschen Antinomie (da die Frage "Ist $N \in N$" nicht geklärt werden kann).

2.1.4 Mengenalgebra

Definition 2.5 (Gleichheit von Mengen)**.** Zwei Mengen M und N sind gleich (kurz: $M = N$), falls sie dieselben Elemente enthalten, d.h. falls gilt:

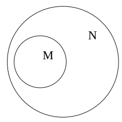
- f.a. $x \in M$ gilt $x \in N$, und
- f.a. $x \in N$ gilt $x \in M$.

Beachte: $\varnothing \neq \{\varnothing\}$, denn $\varnothing$ ist die Menge, die keine Elemente enthält, während $\{\varnothing\}$ eine Menge ist, die ein Element (nämlich $\varnothing$) enthält.

Definition 2.6 (Teilmengen)**.** Seien M, N Mengen.

Teilmenge

(a) M ist eine **Teilmenge** von N (kurz: $M \subseteq N$), wenn jedes Element von M auch ein Element von N ist.



echte Teilmenge

(b) M ist eine **echte Teilmenge** von N (kurz: $M \subsetneq N$), wenn $M \subseteq N$ und $M \neq N$.

Obermenge

(c) M ist eine **Obermenge** von N (kurz: $M \supseteq N$), wenn $N \subseteq M$.

echte Obermenge

(d) M ist eine **echte Obermenge** von N (kurz: $M \supsetneq N$), wenn $M \supseteq N$ und $M \neq N$.

Satz 2.7. *Seien M, N, P Mengen. Dann gilt:*

(a) $M = N \iff M \subseteq N$ *und* $M \supseteq N$.

(b) $M \subseteq N$ *und* $N \subseteq P \implies M \subseteq P$.

Beweis:

(a)

$$\begin{aligned}
M = N \quad &\overset{\text{Def. 2.5}}{\iff} \quad M \text{ und } N \text{ enthalten dieselben Elemente} \\
&\overset{\text{Def. 2.5}}{\iff} \quad \text{f.a. } x \in M \text{ gilt } x \in N \text{ und} \\
&\qquad\quad \text{f.a. } x \in N \text{ gilt } x \in M \\
&\iff \quad \text{jedes Element von } M \text{ ist auch ein Element von } N \text{ und} \\
&\qquad\quad \text{jedes Element von } N \text{ ist auch ein Element von } M \\
&\overset{\text{Def. 2.6(a)}}{\iff} \quad M \subseteq N \text{ und } N \subseteq M \\
&\overset{\text{Def. 2.6(c)}}{\iff} \quad M \subseteq N \text{ und } M \supseteq N
\end{aligned}$$

(b) Es gelte $M \subseteq N$ und $N \subseteq P$.

Behauptung: $M \subseteq P$, d.h. f.a. $m \in M$ gilt $m \in P$.

Beweis: Sei $m \in M$ beliebig. Wir zeigen, dass $m \in P$:

$$m \in M \overset{\text{nach Vor.: } M \subseteq N}{\implies} m \in N \overset{\text{nach Vor.: } N \subseteq P}{\implies} m \in P.$$

□

Definition 2.8. Seien M und N Mengen.

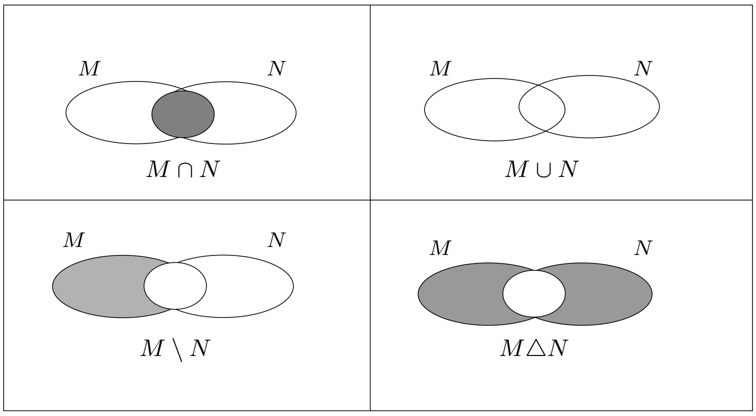
(a) (**Durchschnitt**) Durchschnitt
$M \cap N \ := \ \{x : x \in M \text{ und } x \in N\}$

(b) (**Vereinigung**) Vereinigung
$M \cup N \ := \ \{x : x \in M \text{ oder } x \in N\}$

(c) (**Differenz**) Differenz
$M \setminus N \ := \ M - N := \{x : x \in M \text{ und } x \notin N\}$

(d) (**Symmetrische Differenz**) Symmetrische Differenz
$M \triangle N \ := \ (M \setminus N) \cup (N \setminus M)$

Veranschaulichung durch Venn-Diagramme:



Notation 2.9. Zwei Mengen M und N heißen **disjunkt**, falls $M \cap N = \varnothing$, d.h. falls sie keine gemeinsamen Elemente besitzen. Manchmal schreiben wir $M \dot\cup N$, um $M \cup N$ zu bezeichnen und gleichzeitig auszudrücken, dass $M \cap N = \varnothing$.

Rechenregeln für Durchschnitt und Vereinigung

Satz 2.10. *Seien M, N, P Mengen. Dann gelten:*

(a) ***(Idempotenz)*** Idempotenz
$M \cap M = M$
$M \cup M = M$

(b) ***(Kommutativität)*** Kommutativität
$M \cap N = N \cap M$
$M \cup N = N \cup M$

(c) ***(Assoziativität)*** Assoziativität
$M \cap (N \cap P) = (M \cap N) \cap P$
$M \cup (N \cup P) = (M \cup N) \cup P$

(d) ***(Absorption)*** Absorption
$M \cap (M \cup N) = M$
$M \cup (M \cap N) = M$

Distributivität

(e) ***(Distributivität)***
$M \cap (N \cup P) = (M \cap N) \cup (M \cap P)$
$M \cup (N \cap P) = (M \cup N) \cap (M \cup P)$

Beweis:

(a)

$$\begin{aligned} M \cap M &\overset{\text{Def. 2.8(a)}}{=} \{x : x \in M \text{ und } x \in M\} \\ &= \{x : x \in M\} \\ &= M. \end{aligned}$$

Analog: $M \cup M = M$.

(b)

$$\begin{aligned} M \cap N &\overset{\text{Def. 2.8(a)}}{=} \{x : x \in M \text{ und } x \in N\} \\ &= \{x : x \in N \text{ und } x \in M\} \\ &\overset{\text{Def. 2.8(a)}}{=} N \cap M. \end{aligned}$$

Analog: $M \cup N = N \cup M$.

(c)

$$\begin{aligned} M \cap (N \cap P) &\overset{\text{Def. 2.8(a)}}{=} \{x : x \in M \text{ und } x \in N \cap P\} \\ &\overset{\text{Def. 2.8(a)}}{=} \{x : x \in M \text{ und } (x \in N \text{ und } x \in P)\} \\ &= \{x : (x \in M \text{ und } x \in N) \text{ und } x \in P\} \\ &\overset{\text{Def. 2.8(a)}}{=} \{x : x \in M \cap N \text{ und } x \in P\} \\ &\overset{\text{Def. 2.8(a)}}{=} (M \cap N) \cap P. \end{aligned}$$

Analog: $M \cup (N \cup P) = (M \cup N) \cup P$.

(d) Wir beweisen, dass $M \cap (M \cup N) = M$ in 2 Schritten:
Schritt 1: Zeige, dass $M \subseteq M \cap (M \cup N)$.
Schritt 2: Zeige, dass $M \cap (M \cup N) \subseteq M$.
Aus Satz 2.7(a) folgt dann, dass $M \cap (M \cup N) = M$.

Zu Schritt 1:

Behauptung: $M \subseteq M \cap (M \cup N)$, d.h. f.a. $m \in M$ gilt $m \in M \cap (M \cup N)$.

Beweis: Sei $m \in M$ beliebig. Zu zeigen: $m \in M \cap (M \cup N)$. Wegen $m \in M$ gilt auch $m \in M \cup N$ (gemäß Definition 2.8(b)). Wegen $m \in M$ und $m \in M \cup N$ gilt gemäß Definition 2.8(a), dass $m \in M \cap (M \cup N)$.

Zu Schritt 2:

Behauptung: $M \cap (M \cup N) \subseteq M$, d.h. f.a. $m \in M \cap (M \cup N)$ gilt $m \in M$.

Beweis: Sei $m \in M \cap (M \cup N)$ beliebig. Zu zeigen: $m \in M$. Wegen $m \in M \cap (M \cup N)$ gilt gemäß Definition 2.8(a), dass $m \in M$ und $m \in M \cup N$. Insbesondere ist also $m \in M$.

Insgesamt haben wir damit gezeigt, dass $M \cap (M \cup N) = M$.

Analog: $M \cup (M \cap N) = M$.

(e) Analog. Details: Übung.

□

2.1.5 Komplemente

Das **Komplement** einer Menge M (kurz: $\overline{M}$) soll die Menge aller Elemente sein, die **nicht** zu M gehören. Bei der präzisen Definition von $\overline{M}$ ist allerdings wieder Vorsicht geboten, denn wenn wir einfach

Komplement

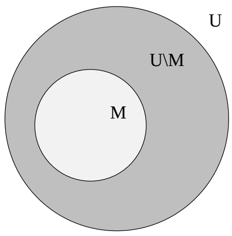
$$\overline{M} \ := \ \{x : x \notin M\}$$

setzen, so gilt für die leere Menge $\varnothing$, dass ihr Komplement $\overline{\varnothing}$ einfach **alles** enthält – und dann wäre

$$\{M : M \in \overline{\varnothing} \text{ und } M \text{ ist eine Menge}\}$$

die "Menge aller Mengen" … und dass es die nicht geben kann, haben wir in Frage 2.4 gesehen.

Daher betrachten wir Mengen stets innerhalb eines festen Universums U, das selber eine Menge ist. Für $M \subseteq U$ setzen wir dann $\overline{M} := U \setminus M$ und bezeichnen $\overline{M}$ als das Komplement von M in U.

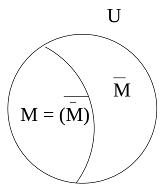


Rechenregeln für Komplemente

Satz 2.11. *Sei U unser festes Universum, das selbst eine Menge ist, und seien $M, N \subseteq U$. Dann gelten:*

(a) ***(Doppelte Negation)***
$\overline{(\overline{M})} = M$

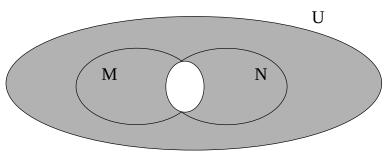
Doppelte Negation



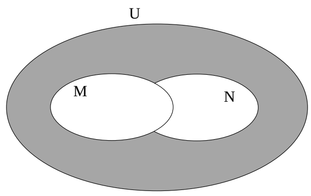
(b) ***(De Morgansche Regeln)***

De Morgansche Regeln

$$\overline{M \cap N} = \overline{M} \cup \overline{N}$$

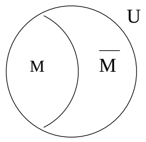


$$\overline{M \cup N} = \overline{M} \cap \overline{N}$$



(c) ***(Inversionsregeln)***
$M \cap \overline{M} = \varnothing$
$M \cup \overline{M} = U$

Inversionsregeln



Identitätsregeln

(d) ***(Identitätsregeln)***
$M \cap \overline{U} = M$
$M \cup \varnothing = M$

Beweis: Übung. □

2.1.6 Mächtigkeit/Kardinalität

Definition 2.12.

endlich

(a) Eine Menge heißt **endlich**, wenn sie nur endlich viele Elemente enthält, d.h. wenn es eine Zahl $n \in \mathbb{N}$ gibt, so dass die Menge genau n Elemente enthält.

Mächtigkeit
Kardinalität

(b) Die **Mächtigkeit** (oder **Kardinalität**) einer Menge M ist

$$|M| \;:=\; \begin{cases} \text{Anzahl der Elemente in } M, & \text{falls } M \text{ endlich ist} \\ \infty \text{ (unendlich)}, & \text{sonst.} \end{cases}$$

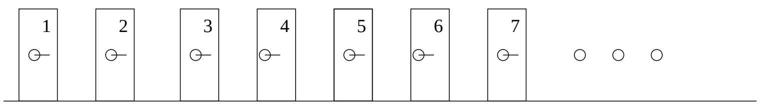
Notation 2.13. $\mathrm{Card}(M) \;:=\; |M|$.

Beispiel 2.14.

- $|\{2,4,6\}| = 3$
- $|\varnothing| = 0$
- $|\{\varnothing\}| = 1$
- $|\mathbb{N}| = \infty$
- $|\mathbb{Z}| = \infty$
- $|\{2,4,6,4\}| = 3$
- $|\{2,\{a,b\}\}| = 2$

Vorsicht beim Vergleich der Mächtigkeit unendlicher Mengen:

Hilberts Hotel (David Hilbert, 1862–1943)
Hilberts Hotel hat unendlich viele Zimmer, die fortlaufend mit $1, 2, 3, \dots$ (also mit allen Zahlen aus $\mathbb{N}_{>0}$) nummeriert sind. Obwohl alle Zimmer belegt sind, schafft der Angestellte an der Rezeption es, für jeden neuen Gast Platz zu schaffen.



Wie? – Er bittet alle Gäste, in das Zimmer mit der nächsthöheren Nummer umzuziehen und gibt dem neuen Gast das Zimmer mit der Nummer 1.

Fügt man also zu einer unendlichen Menge ein Element hinzu, so erhält man keine “wirklich größere” Menge.

2.1.7 Die Potenzmenge

Definition 2.15. Die **Potenzmenge** (engl.: power set) einer Menge M (kurz: $\mathcal{P}(M)$) ist die Menge aller Teilmengen von M. D.h.: Potenzmenge

$$\mathcal{P}(M) \;:=\; \{N : N \subseteq M\}.$$

Notation 2.16. In manchen Büchern wird $\mathcal{P}(M)$ auch mit $\mathrm{Pow}(M)$ (für "power set") oder mit 2^M bezeichnet.

Beispiel 2.17.

- $\mathcal{P}(\{a, b\}) = \{\varnothing, \{a\}, \{b\}, \{a, b\}\}$
- $\mathcal{P}(\{1, 2, 3\}) = \{\varnothing, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$
- $\mathcal{P}(\varnothing) = \{\varnothing\}$
  Insbesondere: $\mathcal{P}(\varnothing) \neq \varnothing$

2.1.8 Paare, Tupel und kartesisches Produkt

Definition 2.18 (Paare und Tupel)**.**

(a) Für beliebige a, b bezeichnet (a, b) das geordnete **Paar** mit Komponenten a und b. Paar

(b) Für $k \in \mathbb{N}$ und beliebige $a_1, \dots, a_k$ bezeichnet $(a_1, \dots, a_k)$ das k**-Tupel** mit Komponenten $a_1, \dots, a_k$. k-Tupel

(c) (Gleichheit zweier Tupel) F.a. $k, l \in \mathbb{N}$ und $a_1, \dots, a_k, b_1, \dots, b_l$ gilt:

$$(a_1, \dots, a_k) = (b_1, \dots, b_l) \;\;:\Longleftrightarrow\;\; k = l \text{ und } a_1 = b_1 \text{ und } a_2 = b_2 \text{ und } \dots \text{ und } a_k = b_k.$$

Bemerkung 2.19.

(a) Für $k = 0$ gibt es genau ein k-Tupel, nämlich das **leere Tupel** $()$, das keine Komponente(n) hat. leeres Tupel

(b) Beachte den Unterschied zwischen Tupeln und Mengen: z.B.
- $(1, 2) \neq (2, 1)$, aber $\{1, 2\} = \{2, 1\}$
- $(1, 1, 2) \neq (1, 2)$, aber $\{1, 1, 2\} = \{1, 2\}$

Definition 2.20.

(a) Sei $k \in \mathbb{N}$ und sei M eine Menge. Die k**-te Potenz** von M ist die Menge k-te Potenz

$$M^k \;:=\; \{(m_1, ..., m_k) : m_1 \in M, \dots, m_k \in M\}.$$

Insbesondere: $M^0 = \{()\}$ besteht genau aus einem Element, dem leeren Tupel.

(b) Das **kartesische Produkt** (bzw. **Kreuzprodukt**) zweier Mengen M, N ist die Menge kartesisches Produkt Kreuzprodukt

$$M \times N \;:=\; \{(m, n) : m \in M,\, n \in N\}.$$

(c) Sei $k \in \mathbb{N}_{>0}$ und seien $M_1, \ldots, M_k$ Mengen. Das kartesische Produkt von $M_1, \ldots, M_k$ ist die Menge
$$M_1 \times \cdots \times M_k \ := \ \{(m_1, \ldots, m_k) : m_1 \in M_1, \ldots, m_k \in M_k\}.$$

Beispiel 2.21. Sei $M = \{a, b\}$ und $N = \{1, 2, 3\}$.

- $M \times N = \{(a,1), (a,2), (a,3), (b,1), (b,2), (b,3)\}$
- $M \times \{1\} = \{(a,1), (b,1)\}$
- $M \times \varnothing = \varnothing$
- $M^2 = \{(a,a), (a,b), (b,a), (b,b)\}$
- $M^1 = \{(a), (b)\}$
- $M^0 = \{()\}$
- $\varnothing^2 = \varnothing$
- $\varnothing^1 = \varnothing$
- $\varnothing^0 = \{()\}$
- Im Beispiel 2.1 hatten wir die Karten eines Skat-Kartenspiels durch folgende Wertebereiche modelliert:
$$\begin{aligned} \text{KartenArten} &= \{\text{Kreuz}, \text{Pik}, \text{Herz}, \text{Karo}\} \\ \text{KartenSymbole} &= \{7, 8, 9, 10, \text{Bube}, \text{Dame}, \text{König}, \text{Ass}\} \\ \text{Karten} &= \text{KartenArten} \times \text{KartenSymbole} \end{aligned}$$
- Uhrzeiten kann man repräsentieren durch Elemente der Menge
$$\text{Uhrzeiten} \ := \ \text{Stunden} \times \text{Minuten} \times \text{Sekunden},$$
wobei
$$\begin{aligned} \text{Stunden} &:= \{0, 1, 2, ..., 23\} \\ \text{Minuten} &:= \{0, 1, 2, ..., 59\} \\ \text{Sekunden} &:= \{0, 1, 2, ..., 59\} \end{aligned}$$
Das Tupel $(9, 45, 0)$ repräsentiert dann die Uhrzeit "9 Uhr, 45 Minuten und 0 Sekunden".

Die Mächtigkeit von kartesischen Produkten

Satz 2.22.

(a) Seien M und N zwei endliche Mengen. Dann gilt:
$$|M \times N| \ = \ |M| \cdot |N|.$$

(b) Sei $k \in \mathbb{N}_{>0}$ und seien $M_1, \ldots, M_k$ endliche Mengen. Dann gilt:
$$|M_1 \times \cdots \times M_k| \ = \ |M_1| \ \cdot \ \cdots \ \cdot \ |M_k| \ = \ \prod_{i=1}^{k} |M_i|.$$

(c) Sei $k \in \mathbb{N}$ und sei M eine endliche Menge. Dann gilt:

$$|M^k| \;=\; |M|^k.$$

Beweis:

(a)

$$\begin{aligned} |M \times N| \;&=\; \Big|\dot{\bigcup}_{m \in M} \big(\{m\} \times N\big)\Big| \;=\; \sum_{m \in M} |\{m\} \times N| \\ &=\; \sum_{m \in M} |N| \;=\; \underbrace{|N| + \cdots + |N|}_{|M|\text{-mal}} \;=\; |M| \cdot |N|. \end{aligned}$$

(b) analog

(c)

$$|M^k| \;=\; \underbrace{|M \times \cdots \times M|}_{k\text{-mal}} \;\overset{(b)}{=}\; \underbrace{|M| \cdot \;\cdots\; \cdot |M|}_{k\text{-mal}} \;=\; |M|^k.$$

□

2.1.9 Worte bzw. endliche Folgen

Bemerkung 2.23. Sei A eine Menge.

- Gelegentlich fassen wir ein Tupel $(a_1, \ldots, a_k) \in A^k$ als **Wort** auf, dessen “Buchstaben” $a_1, \ldots, a_k$ sind. Um diese Sichtweise zu betonen, schreiben wir oft $a_1 \cdots a_k$. Wort

  Beispiel: Das Tupel $(\mathrm{m}, \mathrm{o}, \mathrm{d}, \mathrm{e}, \mathrm{l}, \mathrm{l})$ identifizieren wir mit dem Wort modell.

- A ist dann das **Alphabet**, über dem die Worte gebildet werden, und $a_1 \cdots a_k$ wird “Wort über A” genannt. Alphabet

- Das **leere Tupel** $() \in A^0$ heißt auch **leeres Wort** und wird oft als ε (epsilon) bezeichnet. leeres Wort (ε)

- Die **Länge** eines Wortes $a_1 \cdots a_k$ ist die Zahl Länge

  $$|a_1 \cdots a_k| \;:=\; k.$$

  Insbesondere ist $|\varepsilon| = 0$, d.h. das leere Wort hat die Länge 0.

- Sind $v = a_1 \cdots a_k$ und $w = b_1 \cdots b_l$ zwei Worte über A, so ist die **Konkatenation** von v und w das Wort Konkatenation

  $$vw \;:=\; a_1 \cdots a_k b_1 \cdots b_l.$$

- Manchmal wird ein Wort $a_1 \cdots a_k$ auch als **Folge** der Länge k aufgefasst.

Definition 2.24 (A^*, A^+, Sprache)**.** Sei A ein Alphabet (d.h. eine Menge).

(a) Die **Menge aller Worte über** A (von beliebiger endlicher Länge) bezeichnen wir mit A^*. Es gilt also: Menge aller Worte über A (A^*)

$$A^* \;=\; \bigcup_{k \in \mathbb{N}} A^k \;=\; \{a_1 \cdots a_k : k \in \mathbb{N},\, a_1, \ldots, a_k \in A\}.$$

Beachte: Wegen $0 \in \mathbb{N}$ und $A^0 = \{()\} = \{\varepsilon\}$ enthält A^* insbesondere das leere Wort.

A^+

(b) Die Menge aller **nicht-leeren** Worte über A (von beliebiger endlicher Länge) bezeichnen wir mit A^+. Es gilt:

$$A^+ \;=\; A^* \setminus \{\varepsilon\} \;=\; \{a_1 \cdots a_k : k \in \mathbb{N}_{>0},\, a_1, \ldots, a_k \in A\}.$$

Sprache

(c) Eine **Sprache** über A ist eine Teilmenge von A^*.

Bemerkung. In vielen Büchern werden Sprachen mit dem Buchstaben L (für **L**anguage) oder Varianten wie L' oder L_1 bezeichnet.

Beispiel 2.25 (Natürliche Sprachen)**.**
Alphabet:

$$A_{\text{deutsch}} \;:=\; \{\text{A, B, } \ldots, \text{ Z, Ä, Ö, Ü, a, b, } \ldots, \text{ z, ä, ö, ü, ß, ., ,, :, ;, !, ?, -, \_}\}$$

Sprachen über $A_{deutsch}$:

- $L_1 :=$ Menge aller grammatikalisch korrekten Sätze der deutschen Sprache (aufgefasst als Zeichenketten über A_{deutsch})
- $L_2 :=$ Menge aller Wörter der deutschen Sprache

Beispiel 2.26 (Programmiersprachen)**.**
Alphabet:

$$\text{ASCII} \;:=\; \text{Menge aller ASCII-Symbole}$$

Sprachen:

- $L_1 :=$ Menge aller JAVA-Schlüsselwörter
- $L_2 :=$ Menge aller erlaubten Variablennamen in JAVA
- $L_3 :=$ Menge aller syntaktisch korrekten JAVA-Programme

2.1.10 Relationen

Relationen sind Teilmengen von kartesischen Produkten. Präzise:

Definition 2.27.

Relation

(a) Seien M, N Mengen. Eine **Relation** von M nach N ist eine Teilmenge von $M \times N$.

Stelligkeit

(b) Sei $k \in \mathbb{N}_{>0}$ und seien $M_1, \ldots, M_k$ Mengen. Eine **Relation auf** $M_1, \ldots, M_k$ ist eine Teilmenge von $M_1 \times \cdots \times M_k$. Die **Stelligkeit** einer solchen Relation ist k.

(c) Sei M eine Menge und sei $k \in \mathbb{N}$. Eine **k-stellige Relation über M** ist eine Teilmenge von M^k.

Beispiel 2.28. Um Datumsangaben im Format (Tag, Monat, Jahr) anzugeben, nutzen wir die Wertebereiche

$$\begin{aligned} \text{TagWerte} &:= \{1, 2, \ldots, 31\} \\ \text{MonatsWerte} &:= \{1, 2, \ldots, 12\} \\ \text{JahresWerte} &:= \mathbb{Z}. \end{aligned}$$

Die Menge "Gültig" aller **gültigen** Daten ist dann eine **Teilmenge** von

$$\text{TagWerte} \times \text{MonatsWerte} \times \text{JahresWerte},$$

d.h. eine **Relation** auf TagWerte, MonatsWerte, JahresWerte, zu der beispielsweise das Tupel $(24, 12, 2007)$ gehört, nicht aber das Tupel $(30, 2, 2008)$.

Notation 2.29.

- Ist R eine Relation von M nach N (für zwei Mengen M, N), so schreiben wir oft

$$mRn \quad \text{anstatt} \quad (m, n) \in R.$$

  Beispiel:
  - $m \leq n$, für natürliche Zahlen m, n
  - $m \neq n$

- Ist R eine Relation auf $M_1, \ldots, M_k$, so schreiben wir manchmal

$$R(m_1, \ldots, m_k) \quad \text{anstatt} \quad (m_1, \ldots, m_k) \in R.$$

  Das soll verdeutlichen, dass R eine "Eigenschaft" ist, die ein Tupel aus $M_1 \times \cdots \times M_k$ haben kann – oder eben nicht haben kann.

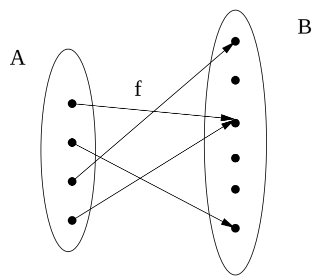
  Im Datums-Beispiel gilt: Gültig$(24, 12, 2007)$, aber es gilt **nicht**: Gültig$(30, 2, 2008)$.

2.1.11 Funktionen

Definition 2.30. Seien A, B Mengen. Eine **Funktion** (oder **Abbildung**) von A nach B ist eine Relation f von A nach B (d.h. $f \subseteq A \times B$) mit der Eigenschaft, dass für jedes $a \in A$ **genau ein** $b \in B$ mit $(a, b) \in f$ existiert.

Funktion
Abbildung

Anschaulich:



Notation 2.31.

(a) Wir schreiben $f\colon A \to B$, um auszudrücken, dass f eine Funktion von A nach B ist.

(b) Ist $f\colon A \to B$ und ist $a \in A$, so bezeichnet $f(a)$ das (eindeutig bestimmte) $b \in B$ mit $(a, b) \in f$. Insbesondere schreiben wir meistens $f(a) = b$ an Stelle von $(a, b) \in f$.

(c) Für $f\colon A \to B$ und $A' \subseteq A$ sei

$$f(A') \ := \ \{f(a) : a \in A'\}.$$

Abb(A, B)

(d) Die Menge aller Funktionen von A nach B bezeichnen wir mit Abb(A, B).

Beachte: In manchen Büchern wird Abb(A, B) auch mit $A \to B$ oder mit B^A bezeichnet.

Bemerkung. Zwei Funktionen $f\colon A \to B$ und $g\colon A \to B$ sind **gleich** (kurz: $f = g$), falls f.a. $a \in A$ gilt: $f(a) = g(a)$.

Definitionsbereich
Bildbereich
Bild

Definition 2.32 (Definitionsbereich, Bildbereich, Bild)**.** Sei $f\colon A \to B$. Der **Definitionsbereich** von f ist die Menge $\text{Def}(f) := A$. Der **Bildbereich** von f ist die Menge B. Das **Bild** von f (genauer: das Bild von A unter f) ist die Menge $\text{Bild}(f) := f(A) \overset{\text{(Def)}}{=} \{f(a) : a \in A\} \subseteq B$.

Restriktion
Einschränkung

Definition 2.33 (Restriktionen)**.** Sei $f\colon A \to B$ eine Funktion und sei $A' \subseteq A$. Die **Restriktion** (oder **Einschränkung**) von f auf A' ist die Funktion

$$f|_{A'}\colon A' \to B,$$

die folgendermaßen definiert ist: f.a. $a \in A'$ ist $f|_{A'}(a) := f(a)$.

2.1.12 Partielle Funktionen

partielle Funktion

Definition 2.34. Eine **partielle Funktion** von einer Menge A in eine Menge B ist eine Funktion f mit $\text{Def}(f) \subseteq A$ und $\text{Bild}(f) \subseteq B$.

Bemerkung 2.35.

totale Funktion

(a) Im Gegensatz zu partiellen Funktionen nennt man Funktionen, wie wir sie in Definition 2.30 definiert haben, auch **totale Funktionen**.

Sprechen wir von "Funktionen", ohne sie explizit als "partiell" zu bezeichnen, so meinen wir in dieser Vorlesung immer "totale" Funktionen.

(b) Jede partielle Funktion von einer Menge A in eine Menge B lässt sich auch als totale Funktion von A nach $B \,\dot\cup\, \{\perp\}$ auffassen, wobei $\perp$ ein spezielles Zeichen ist, das für "undefiniert" steht (und das nicht zur Menge B gehört).

2.1.13 Eigenschaften von Funktionen

Definition 2.36. Sei $f\colon A \to B$.

injektiv

(a) f heißt **injektiv**, falls es für jedes $b \in B$ höchstens ein $a \in A$ mit $f(a) = b$ gibt.

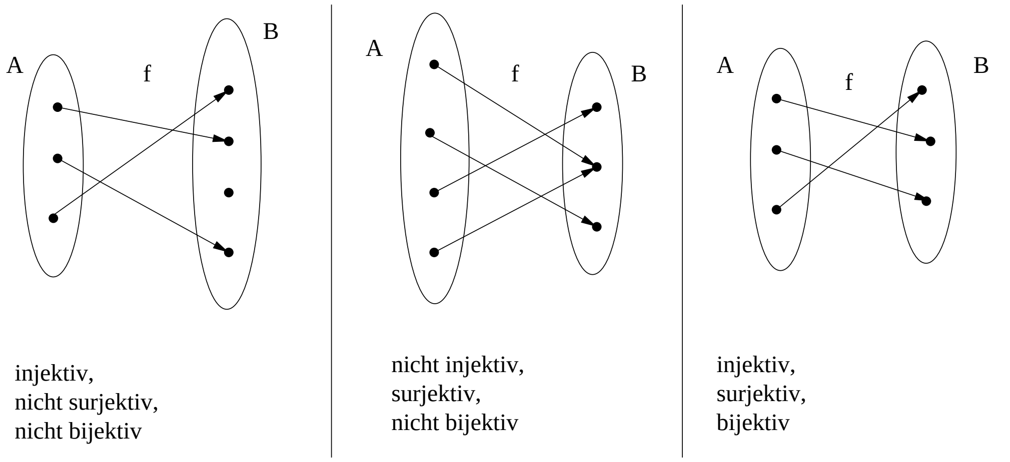
surjektiv

(b) f heißt **surjektiv**, falls es für jedes $b \in B$ mindestens ein $a \in A$ mit $f(a) = b$ gibt.

bijektiv

(c) f heißt **bijektiv**, falls es für jedes $b \in B$ genau ein $a \in A$ mit $f(a) = b$ gibt.

Anschaulich:



Beobachtung 2.37.

(a) Für jede Funktion $f\colon A \to B$ gilt:

$$f \text{ ist bijektiv} \iff f \text{ ist injektiv und surjektiv.}$$

(b) Seien A und B **endliche** Mengen. Dann gilt:

$$|A| = |B| \iff \text{es gibt eine bijektive Funktion von } A \text{ nach } B.$$

Satz 2.38.

(a) Für jede Menge M gibt es eine bijektive Funktion von $\mathcal{P}(M)$ nach $\mathrm{Abb}(M, \{0,1\})$.

(b) Sei B eine Menge, sei A eine endliche Menge und sei $k := |A|$. Dann gibt es eine bijektive Funktion von $\mathrm{Abb}(A, B)$ nach B^k.

Beweis:

(a) Repräsentiere jedes $M' \in \mathcal{P}(M)$ (d.h. $M' \subseteq M$) durch die so genannte **charakteristische Funktion** $\chi_{M'}\colon M \to \{0,1\}$ mit charakteristische Funktion

$$\chi_{M'}(m) \;:=\; \begin{cases} 1, & \text{falls } m \in M' \\ 0, & \text{sonst.} \end{cases} \qquad (*)$$

Sei nun $f\colon \mathcal{P}(M) \to \mathrm{Abb}(M, \{0,1\})$ definiert durch

$$f(M') \;:=\; \chi_{M'}, \quad \text{für jedes } M' \in \mathcal{P}(M). \qquad (**)$$

Behauptung. f ist bijektiv.

Wir zeigen dies in 2 Schritten (und nutzen Beobachtung 2.37(a)).

Schritt 1: f ist injektiv:

Seien $M', M'' \in \mathcal{P}(M)$ mit $f(M') = f(M'')$.

Ziel: Zeige, dass $M' = M''$.

Wegen $f(M') = f(M'')$ gilt gemäß (**), dass $\chi_{M'} = \chi_{M''}$. D.h. f.a. $m \in M$ gilt $\chi_{M'}(m) = \chi_{M''}(m)$. Gemäß (*) gilt daher f.a. $m \in M$, dass

$$m \in M' \iff m \in M''.$$

Somit ist $M' = M''$.

Schritt 2: f ist surjektiv:

Sei $h \in \mathrm{Abb}(M, \{0,1\})$, d.h. $h\colon M \to \{0,1\}$.

Ziel: Finde ein $M' \in \mathcal{P}(M)$ mit $f(M') = h$.

Wir wählen: $M' := \{m \in M : h(m) = 1\}$. Dann ist klar: $M' \in \mathcal{P}(M)$. Gemäß (*) gilt $\chi_{M'} = h$. Gemäß (**) ist daher $f(M') = h$.

(b) *Idee:* Sei $a_1, \ldots, a_k$ eine Liste aller Elemente in A. Repräsentiere jede Funktion $h \in \mathrm{Abb}(A, B)$ durch das k-Tupel $t_h := \big(h(a_1), \ldots, h(a_k)\big)$.

Rest: Übung.

□

Folgerung 2.39. Seien A, B, M endliche Mengen. Dann gilt:

(a) $|\mathrm{Abb}(A, B)| = |B|^{|A|}$.

(b) $|\mathcal{P}(M)| = 2^{|M|}$.

Beweis:

(a) Gemäß Satz 2.38(b) und Beobachtung 2.37(b) gilt für $k := |A|$, dass

$$|\mathrm{Abb}(A, B)| \;=\; |B^k|.$$

Laut Satz 2.22(c) ist $|B^k| = |B|^k$. Somit $|\mathrm{Abb}(A, B)| = |B|^k = |B|^{|A|}$.

(b) Gemäß Satz 2.38(a) und Beobachtung 2.37(b) ist

$$|\mathcal{P}(M)| \;=\; |\mathrm{Abb}(M, \{0,1\})|.$$

Gemäß (a) ist

$$|\mathrm{Abb}(M, \{0,1\})| \;=\; |\{0,1\}|^{|M|} \;=\; 2^{|M|}.$$

□

2.1.14 Spezielle Funktionen

Identitätsfunktion

Definition 2.40. Die **Identitätsfunktion** auf einer Menge M ist die Funktion $\mathrm{id}_M\colon M \to M$ mit $\mathrm{id}_M(m) := m$, f.a. $m \in M$.

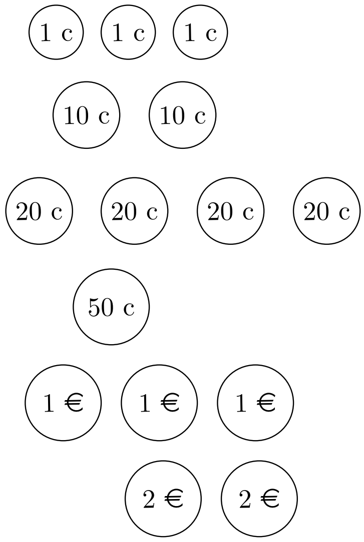
Multimenge

Definition 2.41 (Multimenge, engl.: bag)**.** Eine **Multimenge** über einer Menge M ist eine Funktion $f\colon M \to \mathbb{N}$.

Mit solchen Funktionen kann man “Mengen” beschreiben, in denen einzelne Elemente mehrfach vorkommen können: Für jedes $m \in M$ gibt $f(m)$ an, wie oft m in der “Multimenge” vorkommt.

Beispiel 2.42. Ein Geldbeutel mit

- 3 1-Cent-Münzen
- 0 2-Cent-Münzen
- 0 5-Cent-Münzen
- 2 10-Cent-Münzen
- 4 20-Cent-Münzen
- 1 50-Cent-Münzen
- 3 1-Euro-Münzen
- 2 2-Euro-Münzen



kann repräsentiert werden durch die Multimenge

$$\text{Geldbeutelinhalt}\colon \text{MünzenArten} \to \mathbb{N},$$

wobei

$$\text{MünzenArten} \;:=\; \{1\text{c}, 2\text{c}, 5\text{c}, 10\text{c}, 20\text{c}, 50\text{c}, 1€, 2€\}$$

und

$$\begin{array}{lcl}
\text{GeldbeutelInhalt}(1\text{c}) & := & 3 \\
\text{GeldbeutelInhalt}(2\text{c}) & := & 0 \\
\text{GeldbeutelInhalt}(5\text{c}) & := & 0 \\
\text{GeldbeutelInhalt}(10\text{c}) & := & 2 \\
\text{GeldbeutelInhalt}(20\text{c}) & := & 4 \\
\text{GeldbeutelInhalt}(50\text{c}) & := & 1 \\
\text{GeldbeutelInhalt}(1€) & := & 3 \\
\text{GeldbeutelInhalt}(2€) & := & 2.
\end{array}$$

Bequemere Schreibweise:

$$\text{GeldbeutelInhalt} \;:=\; \{(1\text{c}, 3), (2\text{c}, 0), (5\text{c}, 0), (10\text{c}, 2), (20\text{c}, 4), (50\text{c}, 1), (1€, 3), (2€, 2)\}.$$

2.2 Ein Beispiel zur Modellierung mit Wertebereichen

Beispiel 2.43 (Arbeitskreise der EU)**.**

In der EU-Kommission sollen drei Arbeitskreise gebildet werden. Dazu entsendet jede der Nationen Deutschland, Frankreich, Österreich und Spanien drei Delegierte. Die Arbeitskreise sollen so gebildet werden, dass in jedem Arbeitskreis jede Nation vertreten ist und dass es unter Berücksichtigung der Fremdsprachenkenntnisse der Delegierten in jedem Arbeitskreis eine gemeinsame Sprache gibt, die alle beherrschen.

Aufgabe: Es soll nur die Situation modelliert werden – ein Lösungsverfahren wird hier erst mal nicht gesucht.

Formale Modellierung:

- Menge der **Nationen**:
$$\text{Nationen} \;:=\; \{\text{D}, \text{F}, \text{Ö}, \text{S}\},$$
wobei D für Deutschland, F für Frankreich, Ö für Österreich und S für Spanien steht.

- Die **Delegierten** können wir repräsentieren als Paare, die aus einer Nation und einem Index aus $\{1, 2, 3\}$ bestehen, so dass beispielsweise die drei Delegierten aus Deutschland durch die Paare $(\text{D}, 1)$, $(\text{D}, 2)$ und $(\text{D}, 3)$ modelliert werden. Also:
$$\text{Delegierte} \;:=\; \text{Nationen} \times \text{DelegiertenIndex},$$
wobei $\text{DelegiertenIndex} := \{1, 2, 3\}$.

- Wir nutzen eine Funktion "spricht", die jedem Delegierten die Menge von **Sprachen** zuordnet, die er beherrscht. Formal:
$$\text{spricht}\colon \text{Delegierte} \to \mathcal{P}(\text{Sprachen}),$$
wobei
$$\text{Sprachen} \;:=\; \{\text{deutsch}, \text{französisch}, \text{spanisch}, \text{englisch}, \text{italienisch}, \text{chinesisch}, \ldots\}.$$

- Die drei Arbeitskreise bezeichnen wir mit AK1, AK2, AK3 und setzen
$$\text{Arbeitskreise} \;:=\; \{\text{AK1}, \text{AK2}, \text{AK3}\}.$$

- Eine konkrete Besetzung der drei Arbeitskreise repräsentieren wir durch eine Funktion
$$\text{AK-Besetzung}\colon \text{Arbeitskreise} \to \mathcal{P}(\text{Delegierte}),$$
die jedem der 3 Arbeitskreise die Menge der Delegierten zuordnet, die Mitglied des Arbeitskreises sind.

- Die Bedingung, dass jede Nation in jedem Arbeitskreis vertreten ist, lässt sich folgendermaßen formulieren:
$$\text{f.a. } \; a \in \text{Arbeitskreise} \;\text{ ist }\; \text{Vertretene\_Nationen\_in\_}a \;:=\; \text{Nationen},$$
wobei
$$\begin{aligned}\text{Vertretene\_Nationen\_in\_}a \;:=\; \{n \in \text{Nationen} : \text{es ex. ein } i \in \text{DelegiertenIndex}\\ \text{s.d. } (n, i) \in \text{AK-Besetzung}(a)\}.\end{aligned}$$

- Die Bedingung, dass es für jeden Arbeitskreis eine Sprache gibt, die alle Mitglieder des Arbeitskreises beherrschen, lässt sich folgendermaßen formulieren:
$$\text{f.a. } \; a \in \text{Arbeitskreise} \;\text{ ist }\; \text{Gemeinsame\_Sprachen\_in\_}a \neq \varnothing,$$
wobei
$$\begin{aligned}\text{Gemeinsame\_Sprachen\_in\_}a \;:=\; \{\text{sp} \in \text{Sprachen} : \text{f.a. } d \in \text{AK-Besetzung}(a)\\ \text{ist sp} \in \text{spricht}(d)\}.\end{aligned}$$

□Ende Beispiel 2.43

2.3 Beweise verstehen und selbst formulieren

Ziel dieses Abschnitts ist, einen kurzen Überblick über grundlegende Beweistechniken zu geben, insbesondere:

- direkter Beweis
- Beweis durch Kontraposition
- Beweis durch Widerspruch (indirekter Beweis)
- vollständige Induktion.

2.3.1 Was sind "Sätze" und "Beweise"?

Ein **Satz** (bzw. **Theorem**) besteht aus Voraussetzungen und einer Behauptung. Voraussetzungen und Behauptung sind Aussagen, so dass folgendes gilt: Wenn alle Voraussetzungen erfüllt sind, dann muss auch die Behauptung wahr sein. Der **Beweis** eines Satzes muss nachweisen, dass die Behauptung des Satzes wahr ist und kann dabei verwenden:

Satz
Theorem
Beweis

- die Voraussetzungen des Satzes
- Definitionen und bereits bekannte Tatsachen und Sätze
- im Beweis selbst oder anderswo bereits als wahr bewiesene Aussagen
- logische Schlussregeln

Typische Fehler, die man beim Versuch, Beweise zu formulieren, **vermeiden** sollte, sind:

- unzulässiges Argumentieren mit Beispielen
- Verwendung gleicher Symbole zur Bezeichnung verschiedener Dinge
- Hantieren mit nicht exakt oder gar widersprüchlich definierten Begriffsbildungen
- unzulässige Gedankensprünge beim Schlussfolgern
- Ausnutzung von bis dahin noch unbewiesenen Behauptungen zur Begründung von einzelnen Beweisschritten.

2.3.2 Beweistechnik "direkter Beweis"

Ansatz: die Behauptung eines Satzes wird "direkt" (d.h. ohne "Umwege") bewiesen.

Beispiele für direkte Beweise haben wir in dieser Vorlesung bereits kennengelernt, z.B.

- der Beweis von Satz 2.7
- der Beweis von Satz 2.22
- der Beweis von Satz 2.38
- der Beweis von Folgerung 2.39.

2.3.3 Beweistechnik "Beweis durch Kontraposition"

Beim Beweis durch Kontraposition wird ein Satz der Form "Falls Aussage A gilt, so gilt auch Aussage B" dadurch bewiesen, dass man zeigt: "Falls Aussage B **nicht** gilt, so kann auch Aussage A **nicht** gelten." Als Beispiel für einen Beweis durch Kontraposition betrachten wir folgenden Satz.

Satz 2.44. *Für jedes $n \in \mathbb{N}$ gilt: Falls n^2 eine ungerade Zahl ist, so ist auch n eine ungerade Zahl.*

Beweis: Durch Kontraposition. Sei $n \in \mathbb{N}$ beliebig.

Wir zeigen: Falls n **keine** ungerade Zahl ist, so ist auch n^2 **keine** ungerade Zahl.

$n \in \mathbb{N}$ war beliebig gewählt. Falls n ungerade ist, so ist nichts weiter zu beweisen. Wir betrachten daher nun den Fall, dass n **nicht** ungerade ist (d.h. n ist gerade) und zeigen, dass dann auch n^2 gerade ist.

Beachte: Per Definition ist eine natürliche Zahl m genau dann **gerade**, wenn es ein $k \in \mathbb{N}$ gibt, s.d. $m = 2 \cdot k$.

Daher gilt:

$$
\begin{aligned}
n \text{ ist gerade} &\implies \text{es ex. } k \in \mathbb{N} \text{ s.d. } n = 2 \cdot k \quad \text{(gemäß Def. von "gerade")} \\
&\implies \text{es ex. } k \in \mathbb{N} \text{ s.d. } n^2 = n \cdot (2 \cdot k) \\
&\implies \text{es ex. } k \in \mathbb{N} \text{ s.d. } n^2 = 2 \cdot (n \cdot k) \\
&\implies \text{es ex. } k' \in \mathbb{N} \text{ s.d. } n^2 = 2 \cdot k' \\
&\implies n^2 \text{ ist gerade.} \quad \text{(gemäß Def. von "gerade")}
\end{aligned}
$$

Somit ist n^2 gerade, d.h. n^2 ist keine ungerade Zahl. □

2.3.4 Beweistechnik "Beweis durch Widerspruch" (indirekter Beweis)

Beim Beweis durch Widerspruch wird ein Satz der Form "Falls die Voraussetzungen A erfüllt sind, so gilt Aussage B" dadurch bewiesen, dass man

- annimmt, dass die Voraussetzungen A erfüllt sind, aber die Aussage B **nicht** gilt, und
- daraus einen Widerspruch herleitet.

Als Beispiel für einen Beweis durch Widerspruch betrachten wir folgenden Satz:

Satz 2.45. *Für alle geraden natürlichen Zahlen a und b gilt: $a \cdot b$ ist gerade.*

Beweis: Durch Widerspruch. Angenommen, a und b sind gerade natürlichen Zahlen, so dass $a \cdot b$ **nicht** gerade ist. Da a und b gerade sind, gibt es $k, l \in \mathbb{N}$ s.d. $a = 2 \cdot k$ und $b = 2 \cdot l$. Dann ist $a \cdot b = (2 \cdot k) \cdot (2 \cdot l)$. Insbesondere gibt es also ein $k' \in \mathbb{N}$, s.d. $a \cdot b = 2 \cdot k'$. Gemäß der Definition von "gerade" ist also $a \cdot b$ gerade. Dies ist ein Widerspruch zur Annahme, dass $a \cdot b$ **nicht** gerade ist. □

Ein weiteres, etwas anspruchsvolleres Beispiel für einen Beweis durch Widerspruch ist der Beweis des folgenden Satzes, der "anschaulich" besagt, dass die Potenzmenge von $\mathbb{N}$ **viel** größer ist als die Menge $\mathbb{N}$ selbst.

Satz 2.46 (“$\mathcal{P}(\mathbb{N})$ ist nicht abzählbar”)**.** *Es gibt keine surjektive Funktion von* $\mathbb{N}$ *nach* $\mathcal{P}(\mathbb{N})$*.*

Beweis: Durch Widerspruch. Angenommen, $f\colon \mathbb{N} \to \mathcal{P}(\mathbb{N})$ ist surjektiv. Sei

$$M \ := \ \{n \in \mathbb{N} : n \notin f(n)\}. \tag{*}$$

Klar: $M \in \mathcal{P}(\mathbb{N})$. Da f surjektiv ist, muss es ein $m \in \mathbb{N}$ geben mit $f(m) = M$. Klar: Entweder gilt $m \in M$ oder es gilt $m \notin M$.

Fall 1: $m \notin M$:
Wegen $f(m) = M$ gilt also $m \notin f(m)$. Gemäß (*) für $n := m$ gilt also $m \in M$. ↯ (Widerspruch zu “Fall 1: $m \notin M$”).

Fall 2: $m \in M$:
Wegen $f(m) = M$ gilt also: $m \in f(m)$. Gemäß (*) für $n := m$ gilt also $m \notin M$. ↯ (Widerspruch zu “Fall 2: $m \in M$”).

Somit führen beide Fälle zu einem Widerspruch. Daher kann es keine surjektive Funktion f von $\mathbb{N}$ nach $\mathcal{P}(\mathbb{N})$ geben. □

Ein weiteres, sehr ähnliches Beispiel für einen Beweis durch Widerspruch haben wir bereits im Zusammenhang mit der Russellschen Antinomie kennengelernt.

Satz 2.47 (“Es gibt keine Menge aller Mengen”)**.** *Es gibt keine Menge U, so dass für jede Menge M gilt: $M \in U$.*

Beweis: Durch Widerspruch. Angenommen, U ist eine Menge, so dass für jede Menge M gilt: $M \in U$. Dann ist auch

$$N \ := \ \{M \in U : M \text{ ist eine Menge und } M \notin M\} \tag{*}$$

eine Menge. Insbesondere gilt entweder $N \in N$ oder $N \notin N$.

Fall 1: $N \notin N$: Wir wissen: N ist eine Menge, also insbesondere $N \in U$. Und da wir in Fall 1 sind, gilt außerdem: $N \notin N$. Gemäß (*) (für $M := N$) muss dann aber gelten: $N \in N$. ↯ (Widerspruch zu “Fall 1: $N \notin N$”).

Fall 2: $N \in N$: Wegen $N \in N$ gilt gemäß (*) für $M := N$, dass $N \in U$ ist, dass N eine Menge ist, und dass $N \notin N$ ist. ↯ (Widerspruch zu “Fall 2: $N \in N$”).

Somit führen beide Fälle zu einem Widerspruch. Daher kann es keine Menge U geben, so dass für jede Menge M gilt: $M \in U$. □

2.3.5 Beweistechnik “Beweis durch vollständige Induktion”

Grundidee der vollständigen Induktion

Für $n \in \mathbb{N}$ sei $A(n)$ eine Aussage über die natürliche Zahl n.

Ziel: Zeige, dass für alle $n \in \mathbb{N}$ die Aussage $A(n)$ wahr ist.

Eine Möglichkeit, dies zu zeigen ist, sich das so genannte Induktionsprinzip zu Nutze zu machen.

Induktionsprinzip

Man zeigt, dass eine Aussage $A(n)$ für alle $n \in \mathbb{N}$ wahr ist, indem man folgendermaßen vorgeht:

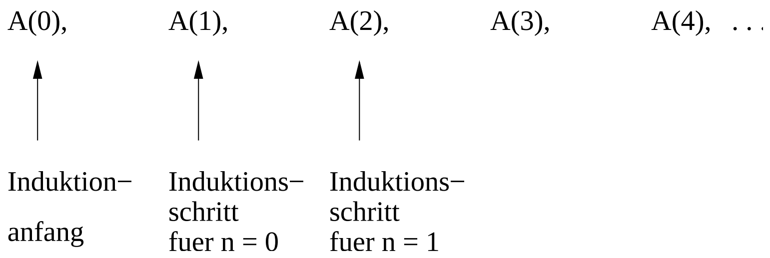
(1) Zuerst zeigt man, dass die Aussage $A(n)$ für die Zahl $n = 0$ gilt.

Diesen Schritt nennt man "Induktionsanfang" bzw. "Induktionsbasis".

(2) Danach zeigt man, dass für jede beliebige natürliche Zahl $n \in \mathbb{N}$ gilt: Falls die Aussage $A(n)$ wahr ist, so ist auch die Aussage $A(n+1)$ wahr.

Diesen Schritt nennt man "Induktionsschritt".

Beachte: Wenn man die Schritte (1) und (2) bewiesen hat, so weiß man, dass die folgenden Aussagen wahr sind:



d.h. man hat gezeigt, dass **für alle** $n \in \mathbb{N}$ die Aussage $A(n)$ wahr ist.

Beispiel für einen Beweis durch vollständige Induktion

Satz 2.48. *F.a.* $n \in \mathbb{N}$ *gilt:* $\sum_{i=0}^{n} 2^i \;=\; 2^{(n+1)} - 1.$

(Bemerkung: Die "Aussage $A(n)$*", deren Gültigkeit hier f.a.* $n \in \mathbb{N}$ *bewiesen werden soll, besagt also: "*$\sum_{i=0}^{n} 2^i = 2^{(n+1)} - 1$*".*

Zur Erinnerung: $\sum_{i=0}^{n} 2^i$ *ist eine abkürzende Schreibweise für* $2^0 + 2^1 + 2^2 + \cdots + 2^n$*.)*

Beweis: Per Induktion nach n.

INDUKTIONSANFANG: $n = 0$

Behauptung: $\sum_{i=0}^{0} 2^i \;=\; 2^{0+1} - 1.$

Beweis:

- $\sum_{i=0}^{0} 2^i \;=\; 2^0 \;=\; 1$
- $2^{0+1} - 1 \;=\; 2^1 - 1 \;=\; 2 - 1 \;=\; 1$

Also: $\sum_{i=0}^{0} 2^i \;=\; 1 \;=\; 2^{0+1} - 1.$

INDUKTIONSSCHRITT: $n \to n+1$

Sei $n \in \mathbb{N}$ beliebig.

Induktionsannahme: $\sum_{i=0}^{n} 2^i \; = \; 2^{n+1} - 1$
(D.h. wir gehen davon aus, dass die Aussage $A(n)$ wahr ist.)

Behauptung: $\sum_{i=0}^{n+1} 2^i \; = \; 2^{(n+1)+1} - 1$
(D.h. wir müssen zeigen, dass dann auch die Aussage $A(n+1)$ wahr ist.)

Beweis:

$$\begin{aligned}
\sum_{i=0}^{n+1} 2^i &\overset{\text{Def.}\sum}{=} \left(\sum_{i=0}^{n} 2^i \right) + 2^{(n+1)} \\
&\overset{\text{Ind.ann.}}{=} 2^{n+1} - 1 + 2^{n+1} \\
&= 2 \cdot 2^{n+1} - 1 \\
&= 2^{(n+1)+1} - 1.
\end{aligned}$$

□

Zwei nützliche Varianten des Induktionsprinzips

Um zu zeigen, dass eine Aussage $A(n)$ für alle $n \in \mathbb{N}$ mit $n \geq n_0$ wahr ist (wobei n_0 eine geeignete natürliche Zahl ist), kann man nach einem der beiden folgenden Schemata vorgehen:

Variante 1:

INDUKTIONSANFANG: $n = n_0$
Behauptung: Die Aussage $A(n_0)$ ist wahr.
Beweis: ...

INDUKTIONSSCHRITT: $n \to n+1$
Sei $n \in \mathbb{N}$ mit $n \geq n_0$ beliebig.
Induktionsannahme: Die Aussage $A(n)$ ist wahr.
Behauptung: Die Aussage $A(n+1)$ ist wahr.
Beweis: ...

Variante 2:

INDUKTIONSANFANG: $n = n_0$
Behauptung: Die Aussage $A(n_0)$ ist wahr.
Beweis: ...

INDUKTIONSSCHRITT: $n \to n+1$
Sei $n \in \mathbb{N}$ mit $n \geq n_0$ beliebig.
Induktionsannahme: Die Aussagen $A(n_0), A(n_0+1), \ldots, A(n)$ sind wahr.
Behauptung: Die Aussage $A(n+1)$ ist wahr.
Beweis: ...

Beispiel 2.49. Welche der Funktionen $f\colon \mathbb{N} \to \mathbb{Z}$ und $g\colon \mathbb{N} \to \mathbb{Z}$ mit $f(n) := n^2 - 7$ und $g(n) := 4 \cdot n$ (f.a. $n \in \mathbb{N}$) liefert größere Funktionswerte?

n	0	1	2	3	4	5	6	7	8	9
$f(n)$	-7	-6	-3	2	9	18	29	42	57	74
$g(n)$	0	4	8	12	16	20	24	28	32	36

Vermutung. F.a. $n \in \mathbb{N}$ mit $n \geq 6$ gilt: $f(n) > g(n)$.

Beweis: per Induktion nach n.

Induktionsanfang: $n = 6$
Behauptung: $f(6) > g(6)$
Beweis:

- $f(6) = 6^2 - 7 = 29$
- $g(6) = 4 \cdot 6 = 24$

Also: $f(6) = 29 > 24 = g(6)$.

Induktionsschritt: $n \to n+1$
Sei $n \in \mathbb{N}$ mit $n \geq 6$ beliebig.
Induktionsannahme: $f(n) > g(n)$, d.h. $n^2 - 7 > 4 \cdot n$.
Behauptung: $f(n+1) > g(n+1)$, d.h. $(n+1)^2 - 7 > 4 \cdot (n+1)$.
Beweis:

$$\begin{aligned}
(n+1)^2 - 7 &= n^2 + 2n + 1 - 7 \\
&= (n^2 - 7) + 2n + 1 \\
&\overset{\text{Ind.ann}}{>} 4n + 2n + 1 \\
&\overset{n \geq 6,\text{ also } 2n+1 \geq 13 > 4}{\geq} 4n + 4 \\
&= 4(n+1).
\end{aligned}$$

□

Auf ähnliche Weise kann man per Induktion auch Folgendes beweisen.

Satz 2.50.

(a) F.a. $n \in \mathbb{N}$ mit $n \geq 1$ gilt: $\sum\limits_{i=1}^{n} i \;=\; \frac{n \cdot (n+1)}{2}$.

(b) F.a. $n \in \mathbb{N}$ mit $n \geq 1$ gilt: $\sum\limits_{i=1}^{n} (2i - 1) \;=\; n^2$
(d.h. die Summe der ersten n ungeraden Zahlen ergibt gerade die Zahl n^2).

(c) F.a. $n \in \mathbb{N}$ mit $n \geq 1$ gilt: $\sum\limits_{i=1}^{n} i^2 \;=\; \frac{n \cdot (n+1) \cdot (2n+1)}{6}$.

Beweis: Übung. □

Das folgende Beispiel zeigt, dass man beim Führen von Induktionsbeweisen vorsichtig bzw. sehr sorgfältig sein sollte:

Beispiel 2.51. Der folgende Satz ist offensichtlich nicht wahr – aber wo steckt der Fehler im Beweis?

"Satz": *F.a. $n \in \mathbb{N}$ mit $n \geq 1$ gilt: Ist M eine Menge von Menschen mit $|M| = n$, so haben alle Menschen in M die gleiche Größe.*

"Beweis": Per Induktion nach n.

INDUKTIONSANFANG: $n = 1$

Behauptung: Ist M eine Menge von Menschen mit $|M| = 1$, so haben alle Menschen in M die gleiche Größe.

Beweis: Sei M eine Menge von Menschen mit $|M| = 1$. D.h. M besteht aus genau einem Menschen. Daher haben offensichtlich alle Menschen in M die gleiche Größe.

INDUKTIONSSCHRITT: $n \to n + 1$

Sei $n \in \mathbb{N}$ mit $n \geq 1$ beliebig.

Induktionsannahme: Ist M' eine Menge von Menschen mit $|M'| = n$, so haben alle Menschen in M' die gleiche Größe.

Behauptung: Ist M eine Menge von Menschen mit $|M| = n + 1$, so haben alle Menschen in M die gleiche Größe.

Beweis: Sei M eine Menge von Menschen mit $|M| = n + 1$. Sei $a_1, a_2, \ldots, a_n, a_{n+1}$ eine Liste aller Menschen in M, d.h. $M = \{a_1, a_2, \ldots, a_n, a_{n+1}\}$. Sei

$$M' := \{a_1, a_2, \ldots, a_n\} \quad \text{und} \quad M'' := \{a_2, \ldots, a_n, a_{n+1}\}.$$

Offensichtlich sind M' und M'' Mengen von Menschen mit $|M'| = n$ und $|M''| = n$. Gemäß der Induktionsannahme gilt daher:

(1) Alle Menschen in M' haben die gleiche Größe, und

(2) alle Menschen in M'' haben die gleiche Größe.

Sei g' die Größe, die gemäß (1) jeder Mensch in M' hat, und sei g'' die Größe, die gemäß (2) jeder Mensch in M'' hat. Laut Definition von M' und M'' gilt: $a_2 \in M'$ und $a_2 \in M''$. Da jeder einzelne Mensch (und daher insbes. der Mensch a_2) nur eine Größe haben kann, gilt: $g' = g''$. Wegen $M = M' \cup M''$ gilt daher, dass alle Menschen in M die gleiche Größe haben, nämlich die Größe $g := g' = g''$. □

Frage: Wo steckt der Fehler im Beweis? □Ende Beispiel 2.51

2.4 Rekursive Definitionen von Funktionen und Mengen

2.4.1 Rekursive Definitionen von Funktionen

Das Induktionsprinzip lässt sich auch zur "induktiven" (bzw. "rekursiven") Definition von Funktionen $f \colon \mathbb{N} \to M$ (wobei M eine beliebige Menge ist) nutzen, indem man folgendermaßen vorgeht:

(1) Definiere $f(0)$. ("Rekursionsanfang")

(2) Definiere, f.a. $n \in \mathbb{N}$, $f(n+1)$ unter Verwendung des Werts $f(n)$ (bzw. unter Verwendung der Werte $f(n), f(n-1), \ldots, f(1), f(0)$). ("Rekursionsschritt")

Auch hier ist wieder eine Reihe von Varianten möglich.

Beispiel 2.52.

Fakultätsfunktion

(a) Rekursive Definition der **Fakultätsfunktion**:

Aufgabe: n Studenten $s_1, \ldots, s_n$ sollen so auf n PCs $c_1, \ldots, c_n$ verteilt werden, dass an jedem PC genau ein Student sitzt.

Frage: Wie viele Möglichkeiten gibt es?

Antwort: $\text{fak}(n)$, wobei

- $\text{fak}(1) \ = \ 1$ und
- $\text{fak}(n+1) \ = \ (n+1) \cdot \text{fak}(n)$ (für alle $n \in \mathbb{N}_{>0}$).

(Insbesondere ist $\text{fak}\colon \mathbb{N}_{>0} \to \mathbb{N}_{>0}$.)

Beachte:

$$\text{fak}(4) \ = \ 4 \cdot \text{fak}(3) \ = \ 4 \cdot 3 \cdot \text{fak}(2) \ = \ 4 \cdot 3 \cdot 2 \cdot \text{fak}(1) \ = \ 4 \cdot 3 \cdot 2 \cdot 1.$$

Allgemein gilt f.a. $n \in \mathbb{N}_{>0}$:

$$\text{fak}(n) \ = \ n \cdot (n-1) \cdot (n-2) \cdot \cdots \cdot 2 \cdot 1 \ = \ \prod_{i=1}^{n} i.$$

Notation. $n! \ := \ \text{fak}(n)$

Fibonacci-Folge

(b) Rekursive Definition der so genannten **Fibonacci-Folge**:

(Leonardo Fibonacci, ital. Mathematiker, 13. Jh.)

Fragestellung: Ein Bauer züchtet Kaninchen. Jedes weibliche Kaninchen bringt im Alter von zwei Monaten ein weibliches Kaninchen zur Welt, und danach jeden Monat ein weiteres.

Wie viele weibliche Kaninchen hat der Bauer am Ende des n-ten Monats, wenn er mit einem neu geborenen weiblichen Kaninchen startet?

Antwort: $\text{fib}(n)$

Rekursive Definition der Fibonacci-Folge: $\text{fib}\colon \mathbb{N}_{>0} \to \mathbb{N}_{>0}$ mit

- $\text{fib}(1) \ := \ 1$,
- $\text{fib}(2) \ := \ 1$ und
- $\text{fib}(n+1) \ := \ \text{fib}(n) + \text{fib}(n-1)$ (f.a. $n \in \mathbb{N}$, $n \geq 2$)

Somit:

n	1	2	3	4	5	6	7	8	9	10	11	12
$\text{fib}(n)$	1	1	2	3	5	8	13	21	34	55	89	144

Um Aussagen über rekursiv definierte Funktionen zu beweisen, kann man wieder das Induktionsprinzip nutzen. Beispiel:

Satz 2.53. *Sei* $\operatorname{fib}\colon \mathbb{N}_{>0} \to \mathbb{N}_{>0}$ *die Fibonacci-Folge. Dann gilt f.a.* $n \in \mathbb{N}_{>0}$*:* $\operatorname{fib}(n) \leq 2^n$*.*

Beweis: Per Induktion nach n.

INDUKTIONSANFANG: betrachte $n = 1$ und $n = 2$

Behauptung: $\operatorname{fib}(1) \leq 2^1$ und $\operatorname{fib}(2) \leq 2^2$.

Beweis: $\operatorname{fib}(1) \overset{\text{Def.}}{=} 1 \leq 2 = 2^1, \quad \operatorname{fib}(2) \overset{\text{Def.}}{=} 1 \leq 4 = 2^2$.

INDUKTIONSSCHRITT: $n \to n+1$

Sei $n \in \mathbb{N}$ mit $n \geq 2$ beliebig.

Induktionsannahme: F.a. $i \in \mathbb{N}_{>0}$ mit $i \leq n$ gilt: $\operatorname{fib}(i) \leq 2^i$.

Behauptung: $\operatorname{fib}(n+1) \leq 2^{n+1}$.

Beweis: $\operatorname{fib}(n+1) \overset{\text{Def.}}{=} \operatorname{fib}(n) + \operatorname{fib}(n-1) \overset{\text{Ind.ann.}}{\leq} 2^n + 2^{n-1} \leq 2 \cdot 2^n = 2^{n+1}$. □

Bemerkung 2.54. Es gibt auch eine "geschlossene Formel", mit der man den n-ten Wert der Fibonacci-Folge, d.h. die Zahl $\operatorname{fib}(n)$ direkt ausrechnen kann, ohne dafür sämtliche Werte $\operatorname{fib}(0), \operatorname{fib}(1), \ldots, \operatorname{fib}(n-1)$ ausrechnen zu müssen:

F.a. $n \in \mathbb{N}_{>0}$ gilt:

$$\operatorname{fib}(n) = \frac{1}{\sqrt{5}} \cdot \left(\left(\frac{1+\sqrt{5}}{2} \right)^n - \left(\frac{1-\sqrt{5}}{2} \right)^n \right).$$

Beweis: Übung (... per Induktion nach n).

(Details: siehe Buch von Meinel und Mundhenk) □

2.4.2 Rekursive Definitionen von Mengen

Oft ist es nützlich, auch **Mengen** rekursiv (bzw. induktiv) zu definieren. Eine rekursive Definition einer Menge M besteht aus:

(a) **Basisregeln** der Form "$m \in M$".

(D.h. die Basisregeln listen explizit bestimmte Elemente auf, die zur Menge M gehören).

(b) **Rekursiven Regeln** der Form:

"Wenn $m_1, \ldots, m_k \in M$, dann $m \in M$",

wobei m von $m_1, \ldots, m_k$ abhängt.

Die dadurch definierte Menge M ist dann die Menge aller Elemente, deren Zugehörigkeit zu M durch endlich-maliges Anwenden der Regeln gezeigt werden kann.

Beispiel 2.55 ("Palindrome")**.** Betrachte das Alphabet $A := \{a, b\}$. Die Menge $\mathrm{PAL} \subseteq A^*$ sei wie folgt rekursiv definiert:

Basisregeln:

(B1) $\varepsilon \in \mathrm{PAL}$

(B2) $a \in \mathrm{PAL}$

(B3) $b \in \mathrm{PAL}$

Rekursive Regeln:

(R1) Ist $w \in \mathrm{PAL}$, so ist auch $awa \in \mathrm{PAL}$

(R2) Ist $w \in \mathrm{PAL}$, so ist auch $bwb \in \mathrm{PAL}$.

Beispiele für Worte, die zur Menge PAL gehören:

$$\underbrace{\varepsilon,\ a,\ b}_{\text{durch Basisregeln}} \qquad \underbrace{aa,\ bb}_{\text{durch rek. Regeln mit } w := \varepsilon} \qquad \underbrace{aaa,\ bab}_{\text{durch rek. Regeln mit } w := a} \qquad \underbrace{aba,\ bbb}_{\text{durch rek. Regeln mit } w := b}$$

Es gilt beispielsweise auch: $aababaa \in \mathrm{PAL}$.

Beweis:

- $a \in \mathrm{PAL}$ (Basisregel (B1))
- Rek. Regel (R2) mit $w := a \implies bab \in \mathrm{PAL}$
- Rek. Regel (R1) mit $w := bab \implies ababa \in \mathrm{PAL}$
- Rek. Regel (R1) mit $w := ababa \implies aababaa \in \mathrm{PAL}$

□

Aber beispielsweise $aab \notin \mathrm{PAL}$, denn gemäß Basisregeln und rekursiven Regeln gilt für jedes Wort $w \in \mathrm{PAL}$: der erste und der letzte Buchstabe von w sind identisch. □Ende Beispiel 2.55

Induktionsprinzip für rekursiv definierte Mengen

Sei M eine rekursiv definierte Menge. Dass eine Aussage $A(m)$ für alle $m \in M$ wahr ist, kann man folgendermaßen zeigen:

(1) Zuerst betrachtet man nacheinander jede Basisregel der Form "$m \in M$" und zeigt, dass die Aussage $A(m)$ wahr ist. ("Induktionsanfang")

(2) Danach betrachtet man nacheinander jede rekursive Regel der Form "Wenn $m_1, \ldots, m_k \in M$, dann $m \in M$" und zeigt folgendes: Wenn die Aussagen $A(m_1), \ldots, A(m_k)$ wahr sind, dann ist auch die Aussage $A(m)$ wahr. ("Induktionsschritt")

Beachte: Wenn man die Schritte (1) und (2) bewiesen hat, so weiß man, dass die Aussage $A(m)$ für alle $m \in M$ wahr ist.

Beispiel 2.56. Sei $A := \{a, b\}$. Für jedes Wort $w \in A^*$ sei w^R das Wort, das durch "Rückwärtslesen" von w entsteht, d.h.:

- Ist $w = \varepsilon$, so ist $w^R = \varepsilon$.
- Ist $w = w_1 \cdots w_k$ mit $k \in \mathbb{N}_{>0}$ und $w_1, \ldots, w_k \in A$, so ist $w^R := w_k \cdots w_1$.

(*Beispiel:* $aaab^R = baaa$.)

Sei PAL die im Beispiel 2.55 rekursiv definierte Teilmenge von A^*.

Behauptung 1: Für jedes Wort $w \in \mathrm{PAL}$ gilt: $w = w^R$.

Beweis: Per Induktion über den Aufbau von PAL.

INDUKTIONSANFANG: Betrachte diejenigen Worte, die aufgrund von Basisregeln zur Menge PAL gehören.
Behauptung: $\varepsilon = \varepsilon^R$, $a = a^R$ und $b = b^R$.
Beweis: Gemäß der Definition von w^R gilt offensichtlich, dass $\varepsilon = \varepsilon^R$, $a = a^R$ und $b = b^R$.

INDUKTIONSSCHRITT: Betrachte die rekursiven Regeln.

- (R1): Sei $w \in \text{PAL}$ und sei $v := awa$. Nach (R1) ist damit auch $v \in \text{PAL}$.

  Induktionsannahme: $w = w^R$

  Behauptung: $v = v^R$

  Beweis: $v^R \overset{\text{Def. } v}{=} (awa)^R \overset{\text{Def. } (\cdot)^R}{=} aw^Ra \overset{\text{Ind.ann.: } w = w^R}{=} awa \overset{\text{Def. } v}{=} v.$

- (R2): Sei $w \in \text{PAL}$ und sei $v := bwb$. Nach (R2) ist damit auch $v \in \text{PAL}$.

  Induktionsannahme: $w = w^R$

  Behauptung: $v = v^R$

  Beweis: $v^R \overset{\text{Def. } v}{=} (bwb)^R \overset{\text{Def. } (\cdot)^R}{=} bw^Rb \overset{\text{Ind.ann.: } w = w^R}{=} bwb \overset{\text{Def. } v}{=} v.$

$\square$

Behauptung 2: Für jedes $w \in A^*$ mit $w = w^R$ gilt: $w \in \text{PAL}$.

Beweisansatz: Zeige folgende Aussage per Induktion nach n:

> Für alle $n \in \mathbb{N}$ gilt: Ist $w \in A^*$ mit $w = w^R$ und $|w| \leq n$, so gilt $w \in \text{PAL}$.

Im Induktionsanfang werden $n = 0$ und $n = 1$ betrachtet; im Induktionsschritt $n \to n+1$ werden alle $n \geq 1$ betrachtet.
Details: Übung. $\square$

Aus Behauptung 1 und Behauptung 2 folgt: $\text{PAL} = \{w \in A^* : w = w^R\}$. $\square$Ende Beispiel 2.56

Antwort auf die Frage aus Beispiel 2.51:

Der "Induktionsschritt $n \to n + 1$" ist für den Wert $n = 1$ nicht schlüssig, denn in diesem Fall gilt $n + 1 = 2$ und

- $M = \{a_1, a_2\}$
- $M' = \{a1\}$
- $M'' = \{a2\}$

Insbesondere gilt also zwar, dass $a_2 \in M''$, aber es gilt nicht, dass $a_2 \in M'$!

2.5 Übungsaufgaben zu Kapitel 2

Aufgabe 2.1. Es sei $M := \{2, 5, 8\}$ und $N := \{3, 5, 7, 11\}$. Schreiben Sie die folgenden Mengen in extensionaler Form auf und geben Sie ihre Kardinalität an.

(a) $M \cup N$ (b) $M \setminus N$ (c) $\mathcal{P}(M)$
(d) $\mathcal{P}(\{\varnothing\})$ (e) $M \times \{a, b\}$ (f) $\{M\} \times \{a, b\}$
(g) $\{P : P \subseteq N \text{ und } |P| = 2\}$ (h) $N^2 \setminus \{(x, x) : x \in N\}$

Aufgabe 2.2. Sei $U := \{1, 2, \ldots, 10\}$ ein festes Universum, und seien $M := \{1, 3, 5\}$, $N := \{2, 3, 5, 7\}$ und $P := \{1, 4, 9\}$. Schreiben Sie jede der folgenden Mengen in extensionaler Form auf und geben Sie ihre Kardinalität an.

(a) $M \setminus (N \cup P)$
(b) $(M \setminus N) \cup (M \setminus P)$
(c) $(M \cup N) \cap \overline{P}$
(d) $(M \cap \overline{P}) \cup (N \cap \overline{P})$
(e) $M^2 \setminus (N \times P)$
(f) $\mathcal{P}(N)$
(g) $M \times P \times \{a, b\}$
(h) $\{Q : Q \subseteq N, |Q| = 3\}$

Aufgabe 2.3. Für jede der folgenden Behauptungen beweisen Sie, dass die Behauptung für alle Mengen M, N, P gilt, oder widerlegen Sie die Behauptung, indem Sie Mengen M, N, P angeben und zeigen, dass die Behauptung für diese Mengen nicht gilt:

(a) Falls $M \subseteq N$ und $N \subsetneqq P$, dann $M \subsetneqq P$.

(b) Falls $M \subseteq N$ und $N \not\subseteq P$, dann $M \not\subseteq P$.

(c) Falls $M \in N$ und $N \in P$, dann $M \in P$.

Aufgabe 2.4.

(a) Welche der Gleichungen stimmt, welche stimmt nicht?
  a) $(M \cap N) \setminus P = (M \setminus P) \cap (N \setminus P)$
  b) $(M \cap N) \setminus P = (M \setminus P) \cup (N \setminus P)$

(b) Begründen Sie Ihre Antwort aus (a) durch Betrachtung von Venn-Diagrammen.

(c) Beweisen Sie Ihre Antworten aus Teil (a).

Aufgabe 2.5.

(a) Bestimmen Sie mit Hilfe von Venn-Diagrammen, welche der folgenden Behauptungen für alle Mengen M, N, P gilt, und welche nicht für alle Mengen M, N, P gilt:
  (i) $M \setminus (N \cup P) = (M \setminus N) \cup (M \setminus P)$
  (ii) $M \cap N = M \setminus (M \setminus N)$

(b) Beweisen Sie, dass Ihre Antworten aus (a) korrekt sind.

Aufgabe 2.6. Seien A, B, C, D, E Teilmengen von $\mathbb{N}$, die wie folgt definiert sind:

$$A = \{3n : n \in \mathbb{N}\} \quad B = \{5n : n \in \mathbb{N}\} \quad C = \{15n : n \in \mathbb{N}\}$$
$$D = \{6n : n \in \mathbb{N}\} \quad E = \{12n : n \in \mathbb{N}\}$$

(a) Welche der folgenden Aussagen sind richtig und welche sind falsch?

$$\text{(i) } E \subseteq D \subseteq A \quad \text{(ii) } E \subseteq C \quad \text{(iii) } A \cap B \subseteq C \quad \text{(iv) } A \cup B \subseteq C$$

(b) Berechnen Sie die folgenden Mengen:

$$\text{(i) } A \cup C \quad \text{(ii) } A \cap E \quad \text{(iii) } B \cap D \quad \text{(iv) } C \setminus B$$

Aufgabe 2.7. Ein Informatikstudent hat 30 Informatikbücher von der Bibliothek ausgeliehen, die sich u.a. mit den Gebieten Algorithmik, Betriebssysteme und Compilerbau beschäftigen. Sei A die Menge der Bücher, die sich u.a. mit Algorithmik beschäftigen, B die Menge der Bücher, die sich u.a. mit Betriebssystemen beschäftigen und C die Menge der Bücher, die sich u.a. mit Compilerbau beschäftigen. Folgende Information über die Anzahl der Bücher und die von ihnen behandelten Themen ist bekannt:

$$|A| = 14, \quad |B| = 18, \quad |C| = 16, \quad |A \cap B| = 8, \quad |A \cap C| = 7, \quad |B \cap C| = 10, \quad |A \cap B \cap C| = 3.$$

(a) Wie viele der Bücher enthalten Material aus mindestens einem der genannten Gebiete? D.h. berechnen Sie $|A \cup B \cup C|$.

(b) Wie viele der Bücher enthalten Material aus mindestens zwei der genannten Gebiete? D.h. berechnen Sie $|D|$, wobei $D := (A \cap B) \cup (A \cap C) \cup (B \cap C)$.

(c) Wie viele der Bücher enthalten Material aus genau einem der genannten Gebiete? D.h. berechnen Sie $|(A \cup B \cup C) \setminus D|$, wobei D die Menge aus (b) ist.

Hinweis: Überlegen Sie sich zunächst anhand von Venn-Diagrammen, wie man die Kardinalitäten der Mengen berechnen kann.

Aufgabe 2.8.

(a) Geben Sie alle Relationen von $A := \{x, y\}$ nach $B := \{c, d\}$ an. Geben Sie für jede Relation an, ob sie eine Funktion von A nach B oder eine partielle Funktion von A nach B oder keines von beiden ist. Geben Sie außerdem für jede Funktion an, ob sie injektiv, surjektiv und/oder bijektiv ist.

(b) Seien M und N beliebige endliche Mengen. Wieviele Relationen von M nach N gibt es?

(c) Geben Sie für jede der folgenden Funktionen f an, ob die Funktion injektiv, surjektiv und/oder bijektiv ist. Geben Sie jeweils auch das Bild von f an.

   a) $f\colon \mathbb{Z} \to \mathbb{Z}$ mit $f(x) := x - 4$ für alle $x \in \mathbb{Z}$
   b) $f\colon \mathbb{Z} \to \mathbb{Z}$ mit $f(x) := 2 \cdot x$ für alle $x \in \mathbb{Z}$
   c) $f\colon \mathbb{Z} \to \mathbb{Z}$ mit $f(x) := x^2$ für alle $x \in \mathbb{Z}$
   d) $f\colon A^* \to \mathbb{N}$ für eine beliebige Menge A mit $|A| \geq 2$ und $f(w) := |w|$ für alle $w \in A^*$

(d) Wie viele Möglichkeiten gibt es,

   a) zwei Bälle B_1, B_2 so auf drei Körbe K_1, K_2, K_3 zu verteilen, dass jeder Ball in einem anderen Korb landet? D.h. wie viele injektive Funktionen von $\{B_1, B_2\}$ nach $\{K_1, K_2, K_3\}$ gibt es?
   b) drei Bälle B_1, B_2, B_3 so auf zwei Körbe K_1, K_2 zu verteilen, dass kein Korb leer bleibt? D.h. wie viele surjektive Funktionen von $\{B_1, B_2, B_3\}$ nach $\{K_1, K_2\}$ gibt es?

Aufgabe 2.9. Beweisen Sie Satz 2.38(b), d.h.:

Sei B eine Menge, sei A eine endliche Menge und sei $k := |A|$. Zeigen Sie, dass es eine bijektive Funktion von $\mathrm{Abb}(A, B)$ nach B^k gibt.

Aufgabe 2.10. In den folgenden Teilaufgaben sollen einige Aspekte einer Variante des Spiels Monopoly mit Wertebereichen modelliert werden. Setzen Sie dabei nur die Menge $\mathbb{N}$ als vordefiniert voraus.

(a) Auf dem Spielbrett gibt es 40 Felder, wobei 22 von diesen Feldern Straßen und 18 Felder Plätze sind. Die Straßen und Plätze sind von 1 bis 22 bzw. von 1 bis 18 durchnummeriert.

Definieren Sie drei Mengen Strassen, Plätze und Felder, deren Elemente Straßen, Plätze bzw. Felder repräsentieren.

(b) Auf ein Feld vom Typ 'Straße' können beliebig viele Häuser und Hotels platziert werden, deren Anordnung aber keine Rolle spielt.

  (i) Definieren Sie eine Menge Bebauungszustände, von der jedes Element den Bebauungszustand einer einzelnen Straße (d.h. die Anzahl der Häuser und die Anzahl der Hotels) repräsentiert.

  (ii) Welches Element von Bebauungszustände beschreibt, dass sich drei Häuser und vier Hotels auf der Straße befinden?

(c) Der Zustand eines Spielers ist zu jedem Zeitpunkt bestimmt durch den Geldbetrag, der ihm zur Verfügung steht, der Menge der Straßen, die er besitzt, und dem Feld, auf dem er sich gerade befindet.

  (i) Definieren Sie eine Menge Spielerzustände, von der jedes Element den Zustand eines Spielers repräsentiert.

  (ii) Welches Element von Spielerzustände beschreibt, dass dem Spieler 1000 Euro zur Verfügung stehen, dass er die Straßen 4, 6 und 7 besitzt, und dass er gerade auf der 17. Straße steht?

(d) Ein Spieler, der eine Straße betritt, die bereits einem anderen Spieler gehört, muss Miete an den Besitzer der Straße entrichten. Die Höhe der Miete hängt von der Straße und deren Bebauungszustand ab.

Geben Sie Mengen A und B an, so dass der oben beschriebene Zusammenhang durch eine Funktion $\mathit{miete}\colon A \to B$ modelliert werden kann, d.h. *miete* soll die Miete für die Straße in Abhängigkeit von der Straße selbst und deren Bebauungszustand angeben.

Aufgabe 2.11. Beweisen Sie: Falls M eine endliche Teilmenge einer unendlichen Menge U ist, so ist das Komplement von M in U unendlich.

Aufgabe 2.12. Beweisen Sie, dass für alle Mengen A, B, C mit $A = B \cup C$ gilt: Falls A unendlich ist, so ist B oder C unendlich.

Aufgabe 2.13. Beweisen Sie folgendes durch vollständige Induktion nach n.

(a) Für alle $n \in \mathbb{N}$ mit $n \geq 1$ gilt: $\displaystyle\sum_{i=1}^{n} (2i - 1) \;=\; n^2.$

(b) Für alle $n \in \mathbb{N}$ mit $n \geq 1$ gilt: $\displaystyle\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}.$

(c) Für alle $n \in \mathbb{N}$ mit $n \geq 1$ gilt: $\sum_{i=1}^{n}(4i-1) \; = \; 2n^2 + n$.

(d) Für alle $x \in \mathbb{R}$ mit $x \geq -1$ und alle $n \in \mathbb{N}$ mit $n \geq 1$ gilt: $1 + n \cdot x \; \leq \; (1+x)^n$.

Aufgabe 2.14. Ein möglicher Algorithmus, um für eine Zahl $n \in \mathbb{N}_{>0}$ den Wert $\mathrm{fib}(n)$ der Fibonacci-Folge zu berechnen, ist:

Algo 1 (bei Eingabe einer Zahl $n \in \mathbb{N}_{>0}$):

1. Falls $n = 1$ oder $n = 2$, dann gib 1 als Ergebnis zurück.
2. Falls $n \geq 3$, dann:
3. Sei x_1 die Ausgabe von *Algo 1* bei Eingabe der Zahl $n-1$.
4. Sei x_2 die Ausgabe von *Algo 1* bei Eingabe der Zahl $n-2$.
5. Gib den Wert $(x_1 + x_2)$ als Ergebnis zurück.

Der Algorithmus benötigt bei Eingabe einer Zahl n höchstens $g_1(n)$ Schritte, wobei

$$\begin{aligned} g_1(1) \; &= \; 1 \quad \text{und} \quad g_1(2) \; = \; 1 \quad \text{und} \\ g_1(n) \; &= \; 3 + g_1(n-1) + g_1(n-2) \quad \text{für alle } n \in \mathbb{N} \text{ mit } n \geq 3 \end{aligned}$$

(der Einfachheit halber zählen die Zeilen 1, 2 und 5 hier jeweils nur als ein Schritt).

Ein anderer Algorithmus, um für eine Zahl $n \in \mathbb{N}_{>0}$ den Wert $\mathrm{fib}(n)$ zu berechnen, ist:

Algo 2 (bei Eingabe einer Zahl $n \in \mathbb{N}_{>0}$):

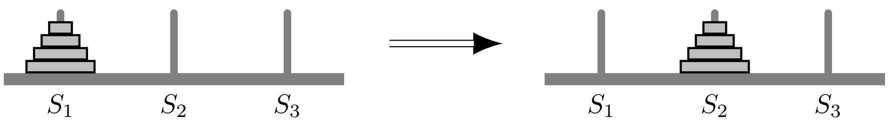
1. Falls $n = 1$ oder $n = 2$, dann gib 1 als Ergebnis zurück.
2. Seien $a_0 := 0$, $a_1 := 1$ und $a_2 := 1$.
3. Wiederhole für alle i von 3 bis n:
4. Ersetze a_0 durch a_1 und a_1 durch a_2.
5. Ersetze a_2 durch $a_0 + a_1$.
6. Gib den Wert a_2 als Ergebnis zurück.

Dieser Algorithmus benötigt bei Eingabe $n \in \mathbb{N}_{>0}$ höchstens $g_2(n) \; := \; 2 \cdot (n-2) + 3$ Schritte (wie oben zählen wir die Zeilen 1, 2, 4, 5 und 6 jeweils als nur einen Schritt).

(a) Welcher der beiden Algorithmen läuft im Allgemeinen schneller? D.h. welche der beiden Funktionen g_1 und g_2 liefert kleinere Funktionswerte?

(b) Beweisen Sie, dass Ihre Antwort aus (a) korrekt ist. D.h. falls Sie in (a) geantwortet haben, dass *Algo i* im Allgemeinen schneller als *Algo j* ist, dann finden Sie eine Zahl $n_0 \in \mathbb{N}_{>0}$ und beweisen Sie per Induktion nach n, dass für alle $n \in \mathbb{N}$ mit $n \geq n_0$ gilt: $g_i(n) < g_j(n)$.

Aufgabe 2.15 (Türme von Hanoi)**.** Ein Turm aus $n \in \mathbb{N}_{>0}$ unterschiedlich großen gelochten Scheiben soll von einem Stab (S_1) auf einen zweiten Stab (S_2) unter Zuhilfenahme eines Hilfsstabes (S_3) verschoben werden (das folgende Bild zeigt die Situation für den Fall $n = 4$).



Dabei müssen die folgenden Regeln beachtet werden:

- Pro Zug darf nur eine Scheibe bewegt werden. Es kann also immer nur die oberste Scheibe eines Turmes bewegt werden.
- Es darf nie eine größere Scheibe auf einer kleineren Scheibe liegen.

(a) Beschreiben Sie, wie der Turm im Fall $n = 4$ von S_1 nach S_2 verschoben werden kann.

(b) Beweisen Sie, dass es für alle $n \in \mathbb{N}_{>0}$ möglich ist, die n Scheiben von S_1 nach S_2 zu verschieben.

Hinweis: Beweisen Sie zuerst durch vollständige Induktion nach n, dass die folgende Aussage für alle $n \in \mathbb{N}$ mit $n \geq 1$ gilt:

A(n): Seien $i, j \in \{1, 2, 3\}$ mit $i \neq j$, sei $m \in \mathbb{N}$ mit $m \geq n$, und seien m Scheiben so auf die drei Stäbe verteilt, dass gilt:

- Auf S_i liegen mindestens n Scheiben.
- Die Scheiben auf den beiden anderen Stäben sind größer als die obersten n Scheiben auf S_i.

Dann lassen sich die obersten n Scheiben von S_i so nach S_j verschieben, dass keine der anderen Scheiben bewegt wird.

Aufgabe 2.16. Sei die Sprache L über dem Alphabet $A := \{(,)\}$ wie folgt rekursiv definiert:

Basisregel:	(B)	$\varepsilon \in L$
Rekursive Regeln:	(R1)	Ist $w \in L$, so ist auch $(w) \in L$.
	(R2)	Sind $w_1, w_2 \in L$, so ist auch $w_1 w_2 \in L$.

(a) Welche der folgenden Wörter gehören zu L und welche nicht?

- ()
- ()()
- ((
- (())
- ())(
- ((())()

(b) Beweisen Sie, dass $(()((()())) \in L$ ist.

(c) Für jedes Symbol $s \in A$ und jedes Wort $w \in A^*$ bezeichne $|w|_s$ die Anzahl der Vorkommen des Symbols s in w. Beweisen Sie durch Induktion, dass für alle Wörter $w \in L$ gilt: $|w|_(= |w|_)$.

(d) Beweisen Sie, dass $()(()() \notin L$ ist.

Aufgabe 2.17. Im Folgenden wird die Syntax einer sehr einfachen Programmiersprache definiert, der so genannten WHILE-Programme. Die Menge L, die hier definiert wird, ist die Menge aller Zeichenketten über dem Alphabet A, die syntaktisch korrekte WHILE-Programme sind. Hierbei ist $A := \{\mathbf{x}, \mathbf{:=}, +, -, \neq, \mathbf{;}, \mathbf{while}, \mathbf{do}, \mathbf{end}\} \cup \mathbb{N}$, und L ist die folgendermaßen rekursiv definierte Menge:

Basisregeln:	(B1)	Für Zahlen $i, j, c \in \mathbb{N}$ gilt: $\mathbf{x}i \mathbin{\mathbf{:=}} \mathbf{x}j + c \ \in L$.
	(B2)	Für Zahlen $i, j, c \in \mathbb{N}$ gilt: $\mathbf{x}i \mathbin{\mathbf{:=}} \mathbf{x}j - c \ \in L$.
Rekursive Regeln:	(R1)	Sind $w_1 \in L$ und $w_2 \in L$, so ist auch $w_1\mathbf{;}\, w_2 \ \in L$.
	(R2)	Ist $w \in L$ und $i \in \mathbb{N}$, so ist $\mathbf{while}\ \mathbf{x}i \neq 0\ \mathbf{do}\ w\ \mathbf{end} \ \in L$.

(a) Welche der folgenden Wörter aus A^* gehören zu L und welche nicht? Begründen Sie jeweils Ihre Antwort.

(i) $\mathbf{x}3 \mathbin{\mathbf{:=}} \mathbf{x}7 - 2$

(ii) $\mathbf{x}3 \mathbin{\mathbf{:=}} 1\mathbf{;}\ \mathbf{x}2 \mathbin{\mathbf{:=}} \mathbf{x}3 + 5$

(iii) $\mathbf{while}\ \mathbf{x}1 \neq 0\ \mathbf{do}\ \mathbf{x}0 \mathbin{\mathbf{:=}} \mathbf{x}0 + 1\mathbf{;}\ \mathbf{x}1 \mathbin{\mathbf{:=}} \mathbf{x}1 - 1\ \mathbf{end}$

(iv) $\mathbf{x}1 \mathbin{\mathbf{:=}} \mathbf{x}1 + 42\mathbf{;}\ \mathbf{while}\ \mathbf{x}1 \neq 0\ \mathbf{do}\ \mathbf{x}1 \mathbin{\mathbf{:=}} \mathbf{x}1 - 1$

(b) Für jedes Wort $w \in A^*$ und jedes Symbol $s \in A$ bezeichne $|w|_s$ die Anzahl der Vorkommen des Symbols s in w. Beweisen Sie durch Induktion, dass für alle Wörter $w \in L$ gilt: $|w|_{\mathbf{do}} = |w|_{\mathbf{end}}$.

3 Aussagenlogik

3.1 Wozu "Logik" im Informatik-Studium?

Logik

Logik (altgriechisch: "Logos": "Vernunft")

- "die Lehre des vernünftigen Schlussfolgerns"
- Teilgebiete der
  - Philosophie
  - Mathematik
  - Informatik
- zentrale Frage: "Wie kann man Aussagen miteinander verknüpfen, und auf welche Weise kann man formal Schlüsse ziehen und Beweise durchführen?"
- These: Logik spielt für die Informatik eine ähnlich wichtige Rolle wie die Differentialrechnung für die Physik
- Anwendungsbereiche der Logik in der Informatik: z.B.
  - zur Repräsentation von statischem Wissen (z.B. im Bereich der künstlichen Intelligenz)
  - zur Verifikation von
    - Schaltkreisen (*Ziel*: beweise, dass ein Schaltkreis bzw. Chip "richtig" funktioniert)
    - Programmen (*Ziel*: beweise, dass ein Programm gewisse wünschenswerte Eigenschaften hat)
    - Protokollen (*Ziel*: beweise, dass die Kommunikation zwischen 2 "Agenten", die nach einem gewissen "Protokoll" abläuft, "sicher" ist – etwa gegen Abhören, Anwendungsbeispiel: Internet-Banking)
  - als Grundlage für Datenbank-Anfragesprachen
  - als Bestandteil von Programmiersprachen (z.B. um "Bedingungen" in "IF-Anweisungen" zu formulieren)
  - zur automatischen Generierung von Beweisen (so genannte "Theorembeweiser")

Aussagenlogik

Aussagen
Junktoren
Aussagenlogik

Aussagen (im Sinne der Aussagenlogik) sind sprachliche Gebilde, die entweder **wahr** oder **falsch** sind. Aussagen können mit **Junktoren** wie "nicht", "und", "oder", "wenn ... dann" etc. zu komplexeren Aussagen verknüpft werden. Die **Aussagenlogik** beschäftigt sich mit allgemeinen Prinzipien des korrekten Argumentierens und Schließens mit Aussagen und Kombinationen von Aussagen.

Beispiel 3.1 ("Geburtstagsfeier").

Fred möchte mit möglichst vielen seiner Freunde Anne, Bernd, Christine, Dirk und Eva seinen Geburtstag feiern. Er weiß, dass Eva nur dann kommt, wenn Christine und Dirk kommen. Andererseits kommt Christine nur dann, wenn auch Anne kommt; und Dirk wird auf keinen Fall kommen, wenn Bernd und Eva beide zur Feier kommen. Anne wiederum wird nur dann kommen, wenn auch Bernd oder Christine dabei sind. Wenn allerdings Bernd und Anne beide zur Party kommen, dann wird Eva auf keinen Fall dabei sein.

Frage: Wie viele Freunde (und welche) werden im besten Fall zur Party kommen?

Das Wissen, das im obigen Text wiedergegeben ist, lässt sich in "atomare Aussagen" zerlegen, die mit Junktoren verknüpft werden können. Die "atomaren Aussagen", um die sich der Text dreht, kürzen wir folgendermaßen ab:

A	$\widehat{=}$	Anne kommt zur Feier
B	$\widehat{=}$	Bernd kommt zur Feier
C	$\widehat{=}$	Christine kommt zur Feier
D	$\widehat{=}$	Dirk kommt zur Feier
E	$\widehat{=}$	Eva kommt zur Feier

Das im Text zusammengefasste "Wissen" lässt sich wie folgt repräsentieren:

	(Wenn E, dann (C und D))	Eva kommt nur dann, wenn Christine und Dirk kommen
und	(Wenn C, dann A)	Christine kommt nur dann, wenn auch Anne kommt
und	(Wenn (B und E), dann nicht D)	Dirk wird auf keinen Fall kommen, wenn Bernd und Eva beide kommen
und	(Wenn A, dann (B oder C))	Anne kommt nur dann, wenn auch Bernd oder Christine dabei sind
und	(Wenn (B und A), dann nicht E)	Wenn Bernd und Anne beide kommen, dann wird Eva auf keinen Fall dabei sein.

Die Aussagenlogik liefert einen Formalismus, mit dessen Hilfe man solches "Wissen" modellieren und Schlüsse daraus ziehen kann – insbesondere z.B. um die Frage, mit wie vielen (und welchen) Gästen Fred bei seiner Feier rechnen kann, zu beantworten. □Ende von Beispiel 3.1

3.2 Syntax und Semantik der Aussagenlogik

Syntax: legt fest, welche Zeichenketten (Worte) Formeln der Aussagenlogik sind — Syntax

Semantik: legt fest, welche "Bedeutung" einzelne Formeln haben — Semantik

(vgl. "Syntax" und "Semantik" von JAVA-Programmen: die Syntax legt fest, welche Zeichenketten JAVA-Programme sind; Semantik bestimmt, was das Programm tut)

Definition 3.2 (Aussagenvariablen und Alphabet der Aussagenlogik).

(a) Eine **Aussagenvariable** (kurz: Variable) hat die Form V_i, für $i \in \mathbb{N}$. Die Menge aller Variablen bezeichnen wir mit AVAR, d.h. $\text{AVAR} := \{V_i : i \in \mathbb{N}\}$. — Aussagenvariable

(b) Das Alphabet der Aussagenlogik ist

$$A_{\mathrm{AL}} \;:=\; \textsc{Avar} \cup \{\mathbf{0}, \mathbf{1}, \neg, \wedge, \vee, \rightarrow, \leftrightarrow, (,)\}.$$

Definition 3.3 (aussagenlogische Formeln: Syntax)**.** Die Menge AL der aussagenlogischen Formeln (kurz: Formeln) ist die folgendermaßen rekursiv definierte Teilmenge von A_{AL}^*:

Basisregeln:

(B0) $\mathbf{0} \in \mathrm{AL}$

(B1) $\mathbf{1} \in \mathrm{AL}$

(BV) Für jede Variable $X \in \textsc{Avar}$ gilt: $X \in \mathrm{AL}$.

Rekursive Regeln:

(R1) Ist $\varphi \in \mathrm{AL}$, so ist auch $\neg\varphi \in \mathrm{AL}$.

(R2) Ist $\varphi \in \mathrm{AL}$ und $\psi \in \mathrm{AL}$, so ist auch
- $(\varphi \wedge \psi) \in \mathrm{AL}$
- $(\varphi \vee \psi) \in \mathrm{AL}$
- $(\varphi \rightarrow \psi) \in \mathrm{AL}$
- $(\varphi \leftrightarrow \psi) \in \mathrm{AL}$.

Notation 3.4.

atomare Formel
Atom

(a) $\mathbf{0}$, $\mathbf{1}$ und die Variablen (d.h. die Elemente aus $\textsc{Avar}$) bezeichnen wir als **atomare Formeln** bzw. **Atome**.

Junktor

(b) Die Symbole $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$ heißen **Junktoren**.

(c) Sind φ und ψ Formeln (d.h. $\varphi \in \mathrm{AL}$ und $\psi \in \mathrm{AL}$), so heißt:

Konjunktion
Disjunktion
Negation

- $(\varphi \wedge \psi)$ **Konjunktion** (bzw. ver-UND-ung) von φ und ψ
- $(\varphi \vee \psi)$ **Disjunktion** (bzw. ver-ODER-ung) von φ und ψ
- $\neg\varphi$ **Negation** (bzw. ver-NEIN-ung) von φ

Beispiel 3.5. Beispiele für Formeln (d.h. Worte über A_{AL}, die zur Menge AL gehören):

- $\big(\neg V_0 \vee (V_5 \rightarrow V_1)\big)$
- $\neg\big((V_0 \wedge \mathbf{0}) \leftrightarrow \neg V_3\big)$

Aber beispielsweise ist

$$V_1 \vee V_2 \wedge V_3$$

keine Formel (d.h. kein Element in AL), da die Klammern fehlen. Auch $(\neg V_1)$ ist **keine** Formel, da die Klammern "zu viel sind". □Ende von Beispiel 3.5

Wir wissen nun, welche Zeichenketten (über dem Alphabet A_{AL}) **Formeln** genannt werden. Um festlegen zu können, welche Bedeutung (d.h. Semantik) solche Formeln haben, brauchen wir folgende Definition:

Variablenmenge

Definition 3.6. Die **Variablenmenge** einer aussagenlogischen Formel φ (kurz: $\mathrm{Var}(\varphi)$) ist die Menge aller Variablen $X \in \textsc{Avar}$, die in φ vorkommen.

Beispiel.

- $\text{Var}\Big((\neg V_0 \vee (V_5 \to V_1)) \Big) = \{V_0, V_1, V_5\}$
- $\text{Var}\Big(\neg((V_0 \wedge \mathbf{0}) \leftrightarrow \neg V_3) \Big) = \{V_0, V_3\}$
- $\text{Var}\Big((\mathbf{0} \vee \mathbf{1}) \Big) = \varnothing$

Definition 3.7.

(a) Eine **Belegung** (bzw. **Wahrheitsbelegung**) ist eine partielle Funktion von AVAR nach $\{0, 1\}$. Belegung Wahrheitsbelegung

(b) Eine Belegung $\mathcal{B}$ ist eine **Belegung für die Formel** φ (bzw. **passend zu** φ), wenn passend zu φ

$$\text{Def}(\mathcal{B}) \;\supseteq\; \text{Var}(\varphi).$$

Intuitive Bedeutung: 1 steht für "wahr" und 0 steht für "falsch".

Definition 3.8 (Semantik der Aussagenlogik)**.** Rekursiv über den Aufbau von AL definieren wir eine Funktion $[\![\cdot]\!]^{\cdot}$, die jeder Formel $\varphi \in$ AL und jeder zu φ passenden Belegung $\mathcal{B}$ einen **Wahrheitswert** (kurz: **Wert**) $[\![\varphi]\!]^{\mathcal{B}} \in \{0, 1\}$ zuordnet: Wahrheitswert

Rekursionsanfang:

- $[\![\mathbf{0}]\!]^{\mathcal{B}} := 0$
- $[\![\mathbf{1}]\!]^{\mathcal{B}} := 1$
- F.a. $X \in$ AVAR gilt: $[\![X]\!]^{\mathcal{B}} := \mathcal{B}(X)$.

Rekursionsschritt:

- Ist $\varphi \in$ AL, so ist $[\![\neg\varphi]\!]^{\mathcal{B}} \;:=\; \begin{cases} 1, & \text{falls } [\![\varphi]\!]^{\mathcal{B}} = 0 \\ 0, & \text{falls } [\![\varphi]\!]^{\mathcal{B}} = 1 \end{cases}$
- Ist $\varphi \in$ AL und $\psi \in$ AL, so ist
  - $[\![(\varphi \wedge \psi)]\!]^{\mathcal{B}} \;:=\; \begin{cases} 1, & \text{falls } [\![\varphi]\!]^{\mathcal{B}} = 1 \text{ und } [\![\psi]\!]^{\mathcal{B}} = 1 \\ 0, & \text{sonst} \end{cases}$
  - $[\![(\varphi \vee \psi)]\!]^{\mathcal{B}} \;:=\; \begin{cases} 0, & \text{falls } [\![\varphi]\!]^{\mathcal{B}} = 0 \text{ und } [\![\psi]\!]^{\mathcal{B}} = 0 \\ 1, & \text{sonst} \end{cases}$
  - $[\![(\varphi \to \psi)]\!]^{\mathcal{B}} \;:=\; \begin{cases} 1, & \text{falls } [\![\varphi]\!]^{\mathcal{B}} = 0 \text{ oder } [\![\psi]\!]^{\mathcal{B}} = 1 \\ 0, & \text{sonst} \end{cases}$
  - $[\![(\varphi \leftrightarrow \psi)]\!]^{\mathcal{B}} \;:=\; \begin{cases} 1, & \text{falls } [\![\varphi]\!]^{\mathcal{B}} = [\![\psi]\!]^{\mathcal{B}} \\ 0, & \text{sonst} \end{cases}$

Intuitive Bedeutung der Semantik:

- **Atome: 1** und **0** bedeuten einfach "wahr" und "falsch".

  Die Variablen $X \in \textsc{Avar}$ stehen für irgendwelche Aussagen. Uns interessiert hier nur, ob diese Aussagen "wahr" oder "falsch" sind – und dies wird durch eine Belegung $\mathcal{B}$ angegeben.

- **Negation:** $\neg\varphi$ bedeutet "nicht φ".

  D.h. $\neg\varphi$ ist wahr (unter Belegung $\mathcal{B}$) $\iff$ φ ist falsch (unter Belegung $\mathcal{B}$). Darstellung durch eine so genannte **Verknüpfungstafel** (bzw. **Wahrheitstafel**):

$[\![\varphi]\!]^{\mathcal{B}}$	$[\![\neg\varphi]\!]^{\mathcal{B}}$
0	1
1	0

- **Konjunktion:** $(\varphi \wedge \psi)$ bedeutet "φ und ψ".

  D.h. $(\varphi \wedge \psi)$ ist wahr (unter Belegung $\mathcal{B}$) $\iff$ φ ist wahr und ψ ist wahr (unter Belegung $\mathcal{B}$).

  Zugehörige Verknüpfungstafel:

$[\![\varphi]\!]^{\mathcal{B}}$	$[\![\psi]\!]^{\mathcal{B}}$	$[\![(\varphi \wedge \psi)]\!]^{\mathcal{B}}$
0	0	0
0	1	0
1	0	0
1	1	1

- **Disjunktion:** $(\varphi \vee \psi)$ bedeutet "φ oder ψ".

  D.h. $(\varphi \vee \psi)$ ist wahr (unter Belegung $\mathcal{B}$) $\iff$ φ ist wahr oder ψ ist wahr (unter Belegung $\mathcal{B}$).

  Zugehörige Verknüpfungstafel:

$[\![\varphi]\!]^{\mathcal{B}}$	$[\![\psi]\!]^{\mathcal{B}}$	$[\![(\varphi \vee \psi)]\!]^{\mathcal{B}}$
0	0	0
0	1	1
1	0	1
1	1	1

- **Implikation:** $(\varphi \rightarrow \psi)$ bedeutet "φ impliziert ψ", d.h. "wenn φ, dann auch ψ".

  D.h. $(\varphi \rightarrow \psi)$ ist wahr (unter Belegung $\mathcal{B}$) $\iff$ wenn φ wahr ist, dann ist auch ψ wahr (unter Belegung von $\mathcal{B}$).

  Zugehörige Verknüpfungstafel:

$[\![\varphi]\!]^{\mathcal{B}}$	$[\![\psi]\!]^{\mathcal{B}}$	$[\![(\varphi \rightarrow \psi)]\!]^{\mathcal{B}}$
0	0	1
0	1	1
1	0	0
1	1	1

- **Biimplikation:** $(\varphi \leftrightarrow \psi)$ bedeutet "φ genau dann, wenn ψ".

  D.h. $(\varphi \leftrightarrow \psi)$ ist wahr (unter Belegung $\mathcal{B}$) $\iff$ φ ist genau dann wahr, wenn ψ wahr ist (unter Belegung von $\mathcal{B}$).

  Zugehörige Verknüpfungstafel:

$[\![\varphi]\!]^{\mathcal{B}}$	$[\![\psi]\!]^{\mathcal{B}}$	$[\![(\varphi \leftrightarrow \psi)]\!]^{\mathcal{B}}$
0	0	1
0	1	0
1	0	0
1	1	1

Beispiel 3.9. Betrachte die Formel $\varphi := \big(\neg V_0 \vee (V_5 \to V_1)\big)$. Dann ist beispielsweise die Funktion $\mathcal{B}\colon \{V_0, V_1, V_5\} \to \{0,1\}$ mit $\mathcal{B}(V_0) := 1$, $\mathcal{B}(V_1) := 1$ und $\mathcal{B}(V_5) := 0$ eine Belegung für φ. Der Wahrheitswert von φ unter Belegung $\mathcal{B}$ ist der Wert

$$\begin{aligned}
[\![\varphi]\!]^{\mathcal{B}} &\overset{\text{Def. 3.8}}{=} \begin{cases} 1, & \text{falls } [\![\neg V_0]\!]^{\mathcal{B}} = 1 \text{ oder } [\![(V_5 \to V_1)]\!]^{\mathcal{B}} = 1 \\ 0, & \text{sonst} \end{cases} \\
&\overset{\text{Def. 3.8}}{=} \begin{cases} 1, & \text{falls } [\![V_0]\!]^{\mathcal{B}} = 0 \text{ oder } ([\![V_5]\!]^{\mathcal{B}} = 0 \text{ oder } [\![V_1]\!]^{\mathcal{B}} = 1) \\ 0, & \text{sonst} \end{cases} \\
&\overset{\text{Def. 3.8}}{=} \begin{cases} 1, & \text{falls } \mathcal{B}(V_0) = 0 \text{ oder } \mathcal{B}(V_5) = 0 \text{ oder } \mathcal{B}(V_1) = 1 \\ 0, & \text{sonst} \end{cases} \\
&= \quad 1 \qquad \text{(denn gemäß obiger Wahl von } \mathcal{B} \text{ gilt } \mathcal{B}(V_5) = 0\text{).}
\end{aligned}$$

Beobachtung 3.10. Sind $\mathcal{B}$ und $\mathcal{B}'$ zwei Belegungen für eine Formel φ, die auf $\mathrm{Var}(\varphi)$ übereinstimmen (d.h. f.a. $X \in \mathrm{Var}(\varphi)$ gilt $\mathcal{B}(X) = \mathcal{B}'(X)$), so ist $[\![\varphi]\!]^{\mathcal{B}} = [\![\varphi]\!]^{\mathcal{B}'}$.

In der Literatur wird diese Beobachtung oft unter dem Namen "**Koinzidenzlemma**" geführt. Intuitiv ist die Beobachtung "offensichtlich richtig", denn in der Definition von $[\![\varphi]\!]^{\mathcal{B}}$ werden ja nur diejenigen Variablen verwendet, die in φ vorkommen (also zu $\mathrm{Var}(\varphi)$ gehören). Einen formalen Beweis der Beobachtung kann man leicht per Induktion über den Aufbau von AL führen. Aufgrund der Beobachtung des "Koinzidenzlemmas" werden wir im Folgenden, wenn wir Belegungen $\mathcal{B}$ für eine Formel φ betrachten, uns meistens nur für diejenigen Werte $\mathcal{B}(X)$ interessieren, für die $X \in \mathrm{Var}(\varphi)$ ist.

Anmerkung 3.11 (griechische Buchstaben)**.** In der Literatur werden Formeln einer Logik traditionell meistens mit griechischen Buchstaben bezeichnet. Hier eine Liste der gebräuchlichsten Buchstaben:

Buchstabe	φ	ψ	χ	θ bzw. ϑ	λ	μ	ν	τ	κ
Aussprache	phi	psi	chi	theta	lambda	mü	nü	tau	kappa

Buchstabe	σ	ρ	ξ	ζ	α	β	γ	δ	ω
Aussprache	sigma	rho	xi	zeta	alpha	beta	gamma	delta	omega

Buchstabe	ε	ι	π	Δ	Γ	Σ	Π	Φ
Aussprache	epsilon	iota	pi	Delta	Gamma	Sigma	Pi	Phi

Um umgangssprachlich formuliertes Wissen (vgl. Beispiel 3.1 “Geburtstagsfeier”) durch aussagenlogische Formeln zu repräsentieren, sind folgende Konventionen bequem:

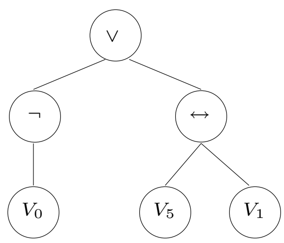
Notation 3.12.

- Statt $V_0, V_1, V_2, \ldots$ bezeichnen wir Variablen oft auch mit $A, B, C, \ldots, X, Y, Z, \ldots$ oder mit Variablen wie $X', Y_1, \ldots$
- Wir schreiben $\bigwedge_{i=1}^{n} \varphi_i$ bzw. $\varphi_1 \wedge \cdots \wedge \varphi_n$ an Stelle von $\big(((\varphi_1 \wedge \varphi_2) \wedge \varphi_3\big) \wedge \cdots \wedge \varphi_n\big)$ (analog für “$\vee$” an Stelle von “$\wedge$”).
- Die äußeren Klammern einer Formel lassen wir manchmal weg und schreiben z.B. $(A \wedge B) \rightarrow C$ an Stelle des (formal korrekten) $\big((A \wedge B) \rightarrow C\big)$.
- Ist φ eine Formel und $\mathcal{B}$ eine Belegung für φ, so sagen wir “$\mathcal{B}$ **erfüllt** φ” (bzw. “$\mathcal{B}$ ist eine **erfüllende Belegung** für φ”), falls $[\![\varphi]\!]^{\mathcal{B}} = 1$.

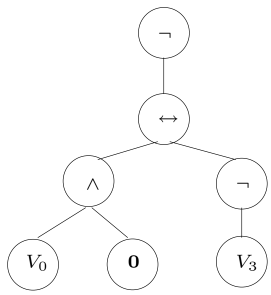
Bemerkung 3.13 (Syntaxbäume zur graphischen Darstellung von Formeln)**.** Die Struktur einer Formel lässt sich bequem durch einen **Syntaxbaum** (englisch: **parse tree**) darstellen.

Beispiele:

- Syntaxbaum der Formel $\big(\neg V_0 \vee (V_5 \leftrightarrow V_1)\big)$:



- Syntaxbaum der Formel $\neg\big((V_0 \wedge \mathbf{0}) \leftrightarrow \neg V_3\big)$:



Computerlesbare Darstellung von Formeln:

Definition 3.14 (ASCII-Syntax für die Aussagenlogik)**.**

(a) Wir betrachten das folgende Alphabet:

$$\text{ASCII} := \text{Menge aller ASCII-Symbole.}$$

(b) Die Menge $\text{AVAR}_{\text{ASCII}}$ aller ASCII-Repräsentationen von Aussagenvariablen ist wie folgt definiert:

$$\text{AVAR}_{\text{ASCII}} := \big\{ w \in \text{ASCII}^+ : \text{das erste Symbol in } w \text{ ist ein Buchstabe, alle weiteren Symbole in } w \text{ sind Buchstaben oder Ziffern} \big\}.$$

(c) Die Menge AL_{ASCII} aller ASCII-Repräsentationen von aussagenlogischen Formeln ist die rekursiv wie folgt definierte Teilmenge von ASCII^*:

Basisregeln:

- $\texttt{0} \in \text{AL}_{\text{ASCII}}$
- $\texttt{1} \in \text{AL}_{\text{ASCII}}$
- Für alle $w \in \text{AVAR}_{\text{ASCII}}$ gilt: $w \in \text{AL}_{\text{ASCII}}$.

Rekursive Regeln:

- Ist $\varphi \in \text{AL}_{\text{ASCII}}$, so ist auch $\texttt{\textasciitilde}\varphi \in \text{AL}_{\text{ASCII}}$.
- Ist $\varphi \in \text{AL}_{\text{ASCII}}$ und $\psi \in \text{AL}_{\text{ASCII}}$, so ist auch
  - $\texttt{(}\varphi\,\texttt{/\textbackslash}\,\psi\texttt{)} \in \text{AL}_{\text{ASCII}}$
  - $\texttt{(}\varphi\,\texttt{\textbackslash/}\,\psi\texttt{)} \in \text{AL}_{\text{ASCII}}$
  - $\texttt{(}\varphi\,\texttt{->}\,\psi\texttt{)} \in \text{AL}_{\text{ASCII}}$
  - $\texttt{(}\varphi\,\texttt{<->}\,\psi\texttt{)} \in \text{AL}_{\text{ASCII}}$.

Bemerkung 3.15. Es ist offensichtlich, wie man Formeln aus AL in ihre entsprechende ASCII-Repräsentation übersetzt und umgekehrt.

Beispiel:

Formel in AL:	$\big((V_0 \wedge \mathbf{0}) \to \neg V_{13}\big)$
entsprechende Formel in AL_{ASCII}:	`( (V0 /\ 0) -> ~V13 )`

In der Vorlesung werden wir nur mit der "abstrakten Syntax", d.h. mit AL, arbeiten. Um aber Formeln in Computer-Programmen eingeben zu können (siehe Webseite der Vorlesung) werden wir die ASCII-Repräsentation verwenden.

Umgangssprachliche Aussagen lassen sich wie folgt durch aussagenlogische Formeln repräsentieren:

Beispiel 3.16.
"Das Fluchtauto war rot oder grün und hatte weder vorne noch hinten ein Nummernschild."

Atomare Aussagen:

- X_R: Das Fluchtauto war rot.

- X_G: Das Fluchtauto war grün.
- X_V: Das Fluchtauto hatte vorne ein Nummernschild.
- X_H: Das Fluchtauto hatte hinten ein Nummernschild.

Obige Aussage wird dann durch folgende aussagenlogische Formel repräsentiert:

$$\big((X_R \vee X_G) \wedge (\neg X_V \wedge \neg X_H)\big).$$

Beispiel 3.17.
Das in Beispiel 3.1 (“Geburtstagsfeier”) aufgelistete Wissen kann folgendemaßen repräsentiert werden:
Atomare Aussagen:

- A: Anne kommt zur Feier
- B: Bernd kommt zur Feier
- C: Christine kommt zur Feier
- D: Dirk kommt zur Feier
- E: Eva kommt zur Feier

Die Aussage des gesamten Textes in Beispiel 3.1 wird durch folgende Formel repräsentiert:

$$\varphi \ := \ \big(E \to (C \wedge D)\big) \ \wedge \ (C \to A) \ \wedge \ \big((B \wedge E) \to \neg D\big) \ \wedge \ \big(A \to (B \vee C)\big) \ \wedge \ \big((B \wedge A) \to \neg E\big).$$

Die Frage “Wie viele (und welche) Freunde werden im besten Fall zur Party kommen?” kann dann durch Lösen der folgenden Aufgabe beantwortet werden: Finde eine Belegung $\mathcal{B}$ für φ, so dass

(1) φ von $\mathcal{B}$ erfüllt wird, d.h. $[\![\varphi]\!]^{\mathcal{B}} = 1$ und

(2) $|\{X \in \{A, B, C, D, E\} : \mathcal{B}(X) = 1\}|$ so groß wie möglich ist.

Um Aufgaben solcher Art lösen zu können, brauchen wir also eine Methode zum Finden der erfüllenden Belegungen für eine Formel. Eine Möglichkeit dafür ist, so genannte Wahrheitstafeln zu benutzen.

Wahrheitstafeln:

Für jede Formel φ kann man die Werte unter allen möglichen Belegungen in einer Wahrheitstafel darstellen. Für jede Belegung $\mathcal{B} : \mathrm{Var}(\varphi) \to \{0, 1\}$ hat die Wahrheitstafel eine Zeile, die die Werte $\mathcal{B}(X)$ f.a. $X \in \mathrm{Var}(\varphi)$ und den Wert $[\![\varphi]\!]^{\mathcal{B}}$ enthält. Um die Wahrheitstafel für φ auszufüllen, ist es bequem, auch Spalten für (alle oder einige) “Teilformeln” von φ einzufügen.

Beispiel 3.18. Wahrheitstafel für $\varphi := \big(\neg V_0 \vee (V_5 \to V_1)\big)$:

V_0	V_1	V_5	$\neg V_0$	$(V_5 \to V_1)$	φ
0	0	0	1	1	1
0	0	1	1	0	1
0	1	0	1	1	1
0	1	1	1	1	1
1	0	0	0	1	1
1	0	1	0	0	0
1	1	0	0	1	1
1	1	1	0	1	1

Beispiel. Wahrheitstafel für $\varphi := \Big(X \wedge \big((\mathbf{1} \to \mathbf{0}) \to \mathbf{0}\big)\Big)$

X	$\mathbf{1}$	$\mathbf{0}$	$(\mathbf{1} \to \mathbf{0})$	$((\mathbf{1} \to \mathbf{0}) \to \mathbf{0})$	φ
0	1	0	0	1	0
1	1	0	0	1	1

Die **erfüllenden** Belegungen für eine Formel φ entsprechen also gerade denjenigen Zeilen der Wahrheitstafel für φ, in denen in der mit "φ" beschrifteten Spalte der Wert 1 steht. Das liefert uns ein Werkzeug, um die in Beispiel 3.17 beschriebene Aufgabe zur "Geburtstagsfeier" zu lösen.

Beispiel 3.19. Sei φ die Formel aus Beispiel 3.17. Die Frage "Wie viel (und welche) Freunde werden bestenfalls zur Party kommen?" können wir lösen, in dem wir

(1) die Wahrheitstafel für φ ermitteln,

(2) alle Zeilen raussuchen, in denen in der mit "φ" beschrifteten Spalte der Wert 1 steht und

(3) aus diesen Zeilen all jene raussuchen, bei denen in den mit A, B, C, D, E beschrifteten Spalten möglichst viele Einsen stehen – jede dieser Zeilen repräsentiert dann eine größtmögliche Konstellation von gleichzeitigen Partybesuchern.

Prinzipiell führt diese Vorgehensweise zum Ziel – leider ist das Verfahren aber recht aufwendig, da die Wahrheitstafel, die man dabei aufstellen muss, sehr groß wird, wie man am Beispiel der Wahrheitstafel für die Formel φ (siehe Tabelle 3.1) sieht. **Erfüllende Belegungen** für φ werden in Tabelle 3.1 durch Zeilen repräsentiert, die grau unterlegt sind.

In der Wahrheitstafel sieht man, dass es **keine** erfüllende Belegung gibt, bei der in den mit A bis E beschrifteten Spalten insgesamt 5 Einsen stehen, und dass es genau **zwei** erfüllende Belegung gibt, bei denen in den mit A bis E beschrifteten Spalten insgesamt 4 Einsen stehen, nämlich die beiden Belegungen $\mathcal{B}_1$ und $\mathcal{B}_2$ mit

$$\mathcal{B}_1(A) = \mathcal{B}_1(C) = \mathcal{B}_1(D) = \mathcal{B}_1(E) = 1 \quad \text{und} \quad \mathcal{B}_1(B) = 0$$

und

$$\mathcal{B}_2(A) = \mathcal{B}_2(B) = \mathcal{B}_2(C) = \mathcal{B}_2(D) = 1 \quad \text{und} \quad \mathcal{B}_2(E) = 0.$$

Die Antworten auf die Frage "Wie viel (und welche) Freunde werden bestenfalls zur Party kommen?" lautet also: Bestenfalls werden 4 der 5 Freunde kommen, und dafür gibt es zwei Möglichkeiten, nämlich

(1) dass alle außer Bernd kommen, und

(2) dass alle außer Eva kommen. □Ende Beispiel 3.19

Angesichts der Wahrheitstafel aus Beispiel 3.19 stellt sich die Frage, wie groß die Wahrheitstafel für eine gegebene Formel φ ist. Die Antwort darauf gibt der folgende Satz.

Satz 3.20. *Sei φ eine aussagenlogische Formel und sei $n := |\mathrm{Var}(\varphi)|$ die Anzahl der in φ vorkommenden Variablen. Dann gibt es 2^n verschiedene zu φ passende Belegungen $\mathcal{B}$ mit $\mathrm{Def}(\mathcal{B}) = \mathrm{Var}(\varphi)$.*

Beweis: Es gilt

$$\begin{aligned}
&\{\mathcal{B} : \mathcal{B} \text{ ist eine zu } \varphi \text{ passende Belegung mit } \mathrm{Def}(\mathcal{B}) = \mathrm{Var}(\varphi)\} \\
\stackrel{\text{Def. 3.7}}{=} &\{\mathcal{B} : \mathcal{B}\colon \mathrm{Var}(\varphi) \to \{0,1\} \text{ ist eine Funktion}\} \\
\stackrel{\text{Not. 2.31}}{=} &\mathrm{Abb}(\mathrm{Var}(\varphi), \{0,1\}).
\end{aligned}$$

A	B	C	D	E	$E \to (C \wedge D)$	$C \to A$	$(B \wedge E) \to \neg D$	$A \to (B \vee C)$	$(B \wedge A) \to \neg E$	φ
0	0	0	0	0	1	1	1	1	1	1
0	0	0	0	1	0	1	1	1	1	0
0	0	0	1	0	1	1	1	1	1	1
0	0	0	1	1	0	1	1	1	1	0
0	0	1	0	0	1	0	1	1	1	0
0	0	1	0	1	0	0	1	1	1	0
0	0	1	1	0	1	0	1	1	1	0
0	0	1	1	1	1	0	1	1	1	0
0	1	0	0	0	1	1	1	1	1	1
0	1	0	0	1	0	1	1	1	1	0
0	1	0	1	0	1	1	1	1	1	1
0	1	0	1	1	0	1	0	1	1	0
0	1	1	0	0	1	0	1	1	1	0
0	1	1	0	1	0	0	1	1	1	0
0	1	1	1	0	1	0	1	1	1	0
0	1	1	1	1	1	0	0	1	1	0
1	0	0	0	0	1	1	1	0	1	0
1	0	0	0	1	0	1	1	0	1	0
1	0	0	1	0	1	1	1	0	1	0
1	0	0	1	1	0	1	1	0	1	0
1	0	1	0	0	1	1	1	1	1	1
1	0	1	0	1	0	1	1	1	1	0
1	0	1	1	0	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1
1	1	0	0	0	1	1	1	1	1	1
1	1	0	0	1	0	1	1	1	0	0
1	1	0	1	0	1	1	1	1	1	1
1	1	0	1	1	0	1	0	1	0	0
1	1	1	0	0	1	1	1	1	1	1
1	1	1	0	1	0	1	1	1	0	0
1	1	1	1	0	1	1	1	1	1	1
1	1	1	1	1	1	1	0	1	0	0

Tabelle 3.1: Wahrheitstafel für die Formel φ aus Beispiel 3.17

Wir wissen außerdem, dass

$$|\mathrm{Abb}(\mathrm{Var}(\varphi), \{0,1\})| \overset{\text{Fol. 2.39(a)}}{=} |\{0,1\}|^{|\mathrm{Var}(\varphi)|} \overset{n=|\mathrm{Var}(\varphi)|}{=} 2^n.$$

$\square$

Satz 3.20 besagt, dass die Wahrheitstafel einer Formel mit n Variablen genau 2^n Zeilen hat. Wie die folgende Tabelle zeigt, ergibt das bereits bei relativ kleinen Werten von n schon riesige Wahrheitstafeln:

n = Anzahl Variablen	2^n =	Anzahl Zeilen der Wahrheitstafel
10	2^{10} =	$1.024 \approx 10^3$
20	2^{20} =	$1.048.576 \approx 10^6$
30	2^{30} =	$1.073.741.824 \approx 10^9$
40	2^{40} =	$1.099.511.627.776 \approx 10^{12}$
50	2^{50} =	$1.125.899.906.842.624 \approx 10^{15}$
60	2^{60} =	$1.152.921.504.606.846.976 \approx 10^{18}$

Zum Vergleich: Das Alter des Universums wird auf 10^{10} Jahre $< 10^{18}$ Sekunden geschätzt.

Ein ganzer Zweig der Informatik (z.B. unter dem Stichwort “SAT-Solving”) und viele internationale Forschungsgruppen beschäftigen sich mit der Aufgabe, Verfahren zu entwickeln, die die erfüllenden Belegungen von aussagenlogischen Formeln ermitteln und dabei wesentlich effizienter sind als das vorgestellte Wahrheitstafel-Verfahren. Ein relativ ernüchterndes Resultat, das Sie in der “Algorithmentheorie”-Vorlesung kennenlernen werden, ist der folgende Satz:

Satz. *Das aussagenlogische Erfüllbarkeitsproblem ist NP-vollständig.*

Das **aussagenlogische Erfüllbarkeitsproblem** (Kurz: **SAT**, für englisch: “satisfiability”) ist dabei das folgende Berechnungsproblem: aussagenlogisches Erfüllbarkeits-problem (SAT)

AUSSAGENLOGISCHES ERFÜLLBARKEITSPROBLEM (SAT)
Eingabe: eine aussagenlogische Formel φ
Frage: Gibt es eine erfüllende Belegung für φ?

Was der Begriff “NP-vollständig” **genau** bedeutet, werden Sie in der “Algorithmentheorie”-Vorlesung lernen; grob gesagt bedeutet “NP-vollständig”, dass es “Wahrscheinlich keinen **effizienten** Algorithmus gibt, der das aussagenlogische Erfüllbarkeitsproblem löst.” Andererseits wurden (besonders in den letzten Jahren) Heuristiken und randomisierte Algorithmen entwickelt, die das aussagenlogische Erfüllbarkeitsproblem trotzdem in vielen Fällen erstaunlich effizient lösen können.

Die folgenden Begriffe werden Ihnen in späteren Vorlesungen immer wieder begegnen:

Definition 3.21. Sei φ eine aussagenlogische Formel.

(a) φ heißt **erfüllbar**, wenn es (mindestens) eine erfüllende Belegung für φ gibt, d.h. wenn es (mindestens) eine zu φ passende Belegung $\mathcal{B}$ gibt mit $[\![\varphi]\!]^{\mathcal{B}} = 1$. erfüllbar

(b) φ heißt **unerfüllbar**, wenn es **keine** erfüllende Belegung für φ gibt. unerfüllbar

(c) φ heißt **allgemeingültig** (bzw. **Tautologie**), wenn **jede** zu φ passende Belegung φ erfüllt, d.h. wenn für jede zu φ passende Belegung $\mathcal{B}$ gilt: $[\![\varphi]\!]^{\mathcal{B}} = 1$. allgemeingültig Tautologie

Beispiel 3.22.

(a) Die Formel $\big((X \vee Y) \wedge (\neg X \vee Y)\big)$ ist
- **erfüllbar**, da z.B. die Belegung $\mathcal{B}$ mit $\mathcal{B}(X) = 0$ und $\mathcal{B}(Y) = 1$ die Formel erfüllt.
- **nicht allgemeingültig**, da z.B. die Belegung $\mathcal{B}'$ mit $\mathcal{B}'(X) = 0$ und $\mathcal{B}'(Y) = 0$ die Formel nicht erfüllt.

(b) Die Formel $(X \wedge \neg X)$ ist **unerfüllbar**, da für jede zur Formel passenden Belegung $\mathcal{B}$ entweder $\mathcal{B}(X) = 1$ oder $\mathcal{B}(X) = 0$ gilt.

Fall 1: $\mathcal{B}(X) = 1$:

$$\begin{aligned} [\![(X \wedge \neg X)]\!]^{\mathcal{B}} &= \begin{cases} 1, & \text{falls } \mathcal{B}(X) = 1 \text{ und } \mathcal{B}(X) = 0 \\ 0, & \text{sonst} \end{cases} \\ &= 0, \text{ da } \mathcal{B}(X) = 1 \neq 0. \end{aligned}$$

Fall 2: $\mathcal{B}(X) = 0$:

$$\begin{aligned} [\![(X \wedge \neg X)]\!]^{\mathcal{B}} &= \begin{cases} 1, & \text{falls } \mathcal{B}(X) = 1 \text{ und } \mathcal{B}(X) = 0 \\ 0, & \text{sonst} \end{cases} \\ &= 0, \text{ da } \mathcal{B}(X) = 0 \neq 1. \end{aligned}$$

(c) Die Formel $(X \vee \neg X)$ ist **allgemeingültig**, da für jede zur Formel passenden Belegung $\mathcal{B}$ entweder $\mathcal{B}(X) = 1$ oder $\mathcal{B}(X) = 0$ gilt. Somit gilt $[\![(X \wedge \neg X)]\!]^{\mathcal{B}} = 1$, für alle zur Formel passenden Belegungen $\mathcal{B}$.

Beobachtung 3.23. Sei φ eine aussagenlogische Formel.

(a) φ ist erfüllbar $\iff$ in der Wahrheitstafel für φ steht in der mit "φ" beschrifteten Spalte mindestens eine 1.

(b) φ ist unerfüllbar $\iff$ in der Wahrheitstafel für φ stehen in der mit "φ" beschrifteten Spalte nur Nullen.

(c) φ ist allgemeingültig $\iff$ in der Wahrheitstafel für φ stehen in der mit "φ" beschrifteten Spalte nur Einsen.

Folgerung 3.24. Für alle aussagenlogischen Formeln φ gilt:

$$\varphi \text{ ist allgemeingültig} \iff \neg\varphi \text{ ist unerfüllbar.}$$

3.3 Folgerung und Äquivalenz

folgt aus

Definition 3.25 (semantische Folgerung)**.** Seien φ und ψ zwei aussagenlogische Formeln. Wir sagen ψ **folgt aus** φ (kurz: $\varphi \models \psi$, "φ impliziert ψ"), falls für jede zu φ und ψ passende Belegung $\mathcal{B}$ gilt:

$$\text{Falls } [\![\varphi]\!]^{\mathcal{B}} = 1, \text{ so auch } [\![\psi]\!]^{\mathcal{B}} = 1.$$

D.h.: $\varphi \models \psi \iff$ jede Belegung, die zu φ und ψ passt und die φ erfüllt, erfüllt auch ψ.

Beispiel 3.26. Sei $\varphi := \big((X \vee Y) \wedge (\neg X \vee Y)\big)$ und $\psi := \big(Y \vee (\neg X \wedge \neg Y)\big)$. Dann gilt $\varphi \models \psi$, aber **nicht** "$\psi \models \varphi$" (kurz: $\psi \not\models \varphi$), denn:

X	Y	$(X \vee Y)$	$(\neg X \vee Y)$	φ	ψ
0	0	0	1	0	1
0	1	1	1	1	1
1	0	1	0	0	0
1	1	1	1	1	1

Hier steht in jeder Zeile (d.h. jede zu φ und ψ passende Belegung), in der in der mit "φ" beschrifteten Spalte eine 1 steht, auch in der mit "ψ" beschrifteten Spalte eine 1. Somit gilt $\varphi \models \psi$.

Andererseits steht in Zeile 1 in der mit ψ beschrifteten Spalte eine 1 und in der mit φ beschrifteten Spalte eine 0. Für die entsprechende Belegung $\mathcal{B}$ (mit $\mathcal{B}(X) = 0$ und $\mathcal{B}(Y) = 0$) gilt also $[\![\psi]\!]^{\mathcal{B}} = 1$ und $[\![\varphi]\!]^{\mathcal{B}} = 0$. Daher gilt $\psi \not\models \varphi$.

Beobachtung 3.27. Seien φ und ψ aussagenlogische Formeln. Dann gilt:

(a) $\mathbf{1} \models \varphi \iff \varphi$ ist allgemeingültig.

(b) $\varphi \models \mathbf{0} \iff \varphi$ ist unerfüllbar.

(c) $\varphi \models \psi \iff (\varphi \to \psi)$ ist allgemeingültig.

(d) $\varphi \models \psi \iff (\varphi \wedge \neg\psi)$ ist unerfüllbar.

Beweis: Übung. □

Definition 3.28 (logische Äquivalenz)**.** Zwei aussagenlogische Formeln φ und ψ heißen **äquivalent** (kurz: $\varphi \equiv \psi$), wenn für alle zu φ und ψ passenden Belegungen $\mathcal{B}$ gilt: $[\![\varphi]\!]^{\mathcal{B}} = [\![\psi]\!]^{\mathcal{B}}$. äquivalent

Beispiel 3.29. Sei $\varphi := \big(X \wedge (X \vee Y)\big)$ und $\psi := X$. Dann ist $\varphi \equiv \psi$, denn

X	Y	$(X \vee Y)$	φ	ψ
0	0	0	0	0
0	1	1	0	0
1	0	1	1	1
1	1	1	1	1

Hier ist die mit "φ" beschriftete Spalte identisch zur mit "ψ" beschrifteten Spalte. D.h. für alle zu φ und ψ passenden Belegungen $\mathcal{B}$ gilt $[\![\varphi]\!]^{\mathcal{B}} = [\![\psi]\!]^{\mathcal{B}}$. D.h.: $\varphi \equiv \psi$.

Beobachtung 3.30. Seien φ und ψ aussagenlogische Formeln. Dann gilt:

(a) $\varphi \equiv \psi \iff (\varphi \leftrightarrow \psi)$ ist allgemeingültig $\iff \varphi \models \psi$ und $\psi \models \varphi$.

(b) φ ist allgemeingültig $\iff \varphi \equiv \mathbf{1}$.

(c) φ ist erfüllbar $\iff \varphi \not\equiv \mathbf{0}$ (d.h. "$\varphi \equiv \mathbf{0}$" gilt nicht).

Beweis: Übung. □

Fundamentale Äquivalenzen der Aussagenlogik

Satz 3.31. *Seien φ, ψ und χ aussagenlogische Formeln. Dann gilt:*

(a) (Idempotenz)

- $(\varphi \wedge \varphi) \equiv \varphi$
- $(\varphi \vee \varphi) \equiv \varphi$

(b) (Kommutativität)

- $(\varphi \wedge \psi) \equiv (\psi \wedge \varphi)$
- $(\varphi \vee \psi) \equiv (\psi \vee \varphi)$

(c) (Assoziativität)

- $((\varphi \wedge \psi) \wedge \chi) \equiv (\varphi \wedge (\psi \wedge \chi))$
- $((\varphi \vee \psi) \vee \chi) \equiv (\varphi \vee (\psi \vee \chi))$

(d) (Absorption)

- $(\varphi \wedge (\varphi \vee \psi)) \equiv \varphi$
- $(\varphi \vee (\varphi \wedge \psi)) \equiv \varphi$

(e) (Distributivität)

- $(\varphi \wedge (\psi \vee \chi)) \equiv ((\varphi \wedge \psi) \vee (\varphi \wedge \chi))$
- $(\varphi \vee (\psi \wedge \chi)) \equiv ((\varphi \vee \psi) \wedge (\varphi \vee \chi))$

(f) (doppelte Negation)

- $\neg\neg\varphi \equiv \varphi$

(g) (De Morgansche Regeln)

- $\neg(\varphi \wedge \psi) \equiv (\neg\varphi \vee \neg\psi)$
- $\neg(\varphi \vee \psi) \equiv (\neg\varphi \wedge \neg\psi)$

(h) (Tertium non Datur)

- $(\varphi \wedge \neg\varphi) \equiv \mathbf{0}$
- $(\varphi \vee \neg\varphi) \equiv \mathbf{1}$

(i)
- $(\varphi \wedge \mathbf{1}) \equiv \varphi$
- $(\varphi \wedge \mathbf{0}) \equiv \mathbf{0}$
- $(\varphi \vee \mathbf{1}) \equiv \mathbf{1}$
- $(\varphi \vee \mathbf{0}) \equiv \varphi$

(j)
- $\mathbf{1} \equiv \neg\mathbf{0}$
- $\mathbf{0} \equiv \neg\mathbf{1}$

(k) (Elimination der Implikation)

- $(\varphi \to \psi) \equiv (\neg\varphi \vee \psi)$

(l) (Elimination der Biimplikation)

- $(\varphi \leftrightarrow \psi) \equiv ((\varphi \to \psi) \wedge (\psi \to \varphi))$

Beweis: Durch einfaches Nachrechnen. Details: Übung. □

Bemerkung 3.32. Durch schrittweises Anwenden der in Satz 3.31 aufgelisteten Äquivalenzen kann man eine gegebene aussagenlogische Formel in eine zu ihr äquivalente Formel umformen.

Beispiel. Sind φ und ψ aussagenlogische Formeln, so gilt:

$$\begin{aligned}
(\varphi \leftrightarrow \psi) &\equiv \big((\varphi \to \psi) \wedge (\psi \to \varphi)\big) && \text{(Satz 3.31(l))} \\
&\equiv \big((\neg\varphi \vee \psi) \wedge (\neg\psi \vee \varphi)\big) && \text{(Satz 3.31(k))}
\end{aligned}$$

3.4 Normalformen

Bisher haben wir gesehen, wie man für eine gegebene aussagenlogische Formel φ eine Wahrheitstafel aufstellen kann.

Frage: Wie kann man umgekehrt zu einer gegebenen Wahrheitstafel eine Formel φ finden, zu der die Wahrheitstafel passt?

Beispiel 3.33. Betrachte die Wahrheitstafel T:

X	Y	Z	φ
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Eine Formel φ, so dass T die Wahrheitstafel für φ ist, kann man folgendermaßen erzeugen:

- Betrachte alle Zeilen von T, in denen in der mit "φ" beschrifteten Spalte eine 1 steht.
- Für jede solche Zeile konstruiere eine Formel, die genau von der zur Zeile gehörenden Belegung erfüllt wird.
- Bilde die Disjunktion (d.h. Ver-ODER-ung) über all diese Formeln. Dies liefert die gesuchte Formel φ.

In unserer Beispiel-Wahrheitstafel T gibt es genau 3 Zeilen, in denen in der mit φ beschrifteten Spalte eine 1 steht, nämlich die Zeilen

X	Y	Z	φ	zur Belegung der jeweiligen Zeile gehörende Formel:
0	0	0	1	$(\neg X \wedge \neg Y \wedge \neg Z)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	
1	0	0	1	$(\ X \wedge \neg Y \wedge \neg Z)$
1	0	1	1	$(\ X \wedge \neg Y \wedge \ \ Z)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	

$\Longrightarrow$ zur Wahrheitstafel T passende Formel:

$$\varphi \ := \ (\neg X \wedge \neg Y \wedge \neg Z) \ \vee \ (X \wedge \neg Y \wedge \neg Z) \ \vee \ (X \wedge \neg Y \wedge Z).$$

Generell kann man auf die beschriebene Art zu jeder beliebigen Wahrheitstafel eine aussagenlogische Formel konstruieren, die zur Wahrheitstafel passt. Die so konstruierten Formeln haben eine besonders einfache Form. Sie sind Disjunktionen von Formeln, die aus Konjunktionen von Variablen oder negierten Variablen bestehen. Formeln, die diese spezielle Struktur besitzen, nennt man auch Formeln in **disjunktiver Normalform** (kurz: **DNF**).

Definition 3.34 (disjunktive Normalform, konjunktive Normalform)**.**

(a) Ein **Literal** ist eine Formel der Form X oder $\neg X$, wobei $X \in \text{AVAR}$ (d.h. X ist eine Aussagenvariable). Ein Literal der Form X, mit $X \in \text{AVAR}$, wird auch **positives Literal** genannt; eine Formel der Form $\neg X$, mit $x \in \text{AVAR}$, **negatives Literal**.

Literal
positives Literal
negatives Literal

(b) Eine aussagenlogische Formel ist in **disjunktiver Normalform** (**DNF**), wenn sie eine Disjunktion von Konjunktionen von Literalen ist, d.h. wenn sie die Gestalt

disjunktive
Normalform
DNF

$$\bigvee_{i=1}^{n} \left(\bigwedge_{j=1}^{m_i} l_{i,j} \right)$$

hat, wobei $n, m_1, \ldots, m_n \in \mathbb{N}_{>0}$ und $l_{i,j}$ ein Literal ist (für jedes $i \in \{1, \ldots, n\}$ und $j \in \{1, \ldots, m_i\}$).

Die Teilformeln $\kappa_i := \bigwedge_{j=1}^{m_i} l_{i,j}$ (für $i \in \{1, \ldots, n\}$) heißen **konjunktive Klauseln**.

konjunktive
Klausel

(c) Eine aussagenlogische Formel ist in **konjunktiver Normalform** (**KNF**), wenn sie eine Konjunktion von Disjunktionen von Literalen ist, d.h. wenn sie die Gestalt

konjunktive
Normalform
KNF

$$\bigwedge_{i=1}^{n} \left(\bigvee_{j=1}^{m_i} l_{i,j} \right)$$

hat, wobei $n, m_1, \ldots, m_n \in \mathbb{N}_{>0}$ und $l_{i,j}$ ein Literal ist (für jedes $i \in \{1, \ldots, n\}$ und $j \in \{1, \ldots, m_i\}$).

disjunktive Klausel

Die Teilformeln $\kappa_i := \bigvee_{j=1}^{m_i} l_{i,j}$ (für $i \in \{1, \ldots, n\}$) heißen **disjunktive Klauseln**.

Normalformen spielen in vielen Anwendungsgebieten eine wichtige Rolle. Beispielsweise geht man in der Schaltungstechnik (Hardware-Entwurf) oft von DNF-Formeln aus, während bei der aussagenlogischen Modellbildung oftmals KNF-Formeln auftreten, da sich eine Sammlung von einfach strukturierten Aussagen sehr gut durch eine Konjunktion von Klauseln ausdrücken lässt.

Satz 3.35. *Für jede aussagenlogische Formel φ gibt es eine Formel ψ_D in DNF und eine Formel ψ_K in KNF, so dass $\varphi \equiv \psi_D \equiv \psi_K$.*
(D.h.: Jede Formel ist äquivalent zu einer Formel in DNF und zu einer Formel in KNF.)

Beweisidee:

- Zur Konstruktion einer zu φ äquivalenten Formel ψ_D in DNF stellen wir zunächst die Wahrheitstafel für φ auf. Falls diese in der mit "φ" beschrifteten Spalte nur Nullen hat (d.h. φ ist unerfüllbar), so setzen wir $\psi_D := (V_0 \wedge \neg V_0)$ – offensichtlich ist ψ_D in DNF und unerfüllbar, also äquivalent zu φ.

  Falls die mit "φ" beschriftete Spalte der Wahrheitstafel mindestens eine 1 enthält, so gehen wir wie in Beispiel 3.33 vor, um eine zu φ äquivalente Formel ψ_D in DNF zu konstruieren.

- Zur Konstruktion einer zu φ äquivalenten Formel ψ_K in KNF können wir folgendermaßen vorgehen:
  (1) Sei $\varphi' := \neg\varphi$.
  (2) Konstruiere eine zu φ' äquivalente Formel ψ'_D in DNF. Sei $\bigvee_{i=1}^{n} \left(\bigwedge_{j=1}^{m_i} l_{i,j} \right)$ die Gestalt von ψ'_D.
  (3) Für alle i, j sei $\tilde{l}_{i,j} := \begin{cases} \neg X, & \text{falls } l_{i,j} = X \text{ für ein } X \in \textsc{Avar} \\ X, & \text{falls } l_{i,j} = \neg X \text{ für ein } X \in \textsc{Avar}. \end{cases}$
  (4) Setze $\psi_K := \bigwedge_{i=1}^{n} \left(\bigvee_{j=1}^{m_i} \tilde{l}_{i,j} \right)$.

  Offensichtlich ist ψ_K eine Formel in KNF. Außerdem gilt:

$$\begin{aligned}
\varphi &\equiv \neg\varphi' \\
&\equiv \neg\psi'_D \\
&\equiv \neg\left(\bigvee_{i=1}^{n} \left(\bigwedge_{j=1}^{m_i} l_{i,j} \right) \right) \\
&\overset{\text{Satz 3.31(g)}}{\equiv} \left(\bigwedge_{i=1}^{n} \neg \left(\bigwedge_{j=1}^{m_i} l_{i,j} \right) \right) \\
&\overset{\text{Satz 3.31(g)}}{\equiv} \bigwedge_{i=1}^{n} \left(\bigvee_{j=1}^{m_i} \neg l_{i,j} \right) \\
&\overset{\text{Def. } \tilde{l}_{i,j}}{\equiv} \bigwedge_{i=1}^{n} \left(\bigvee_{j=1}^{m_i} \tilde{l}_{i,j} \right) \\
&\overset{\text{Def.}}{=} \psi_K .
\end{aligned}$$

□

Abgesehen von DNF und KNF gibt es noch eine weitere wichtige Normalform, die so genannte Negationsnormalform.

Definition 3.36. Eine aussagenlogische Formel ist in **Negationsnormalform** (**NNF**), wenn sie keines der Symbole $\to$, $\leftrightarrow$, $\mathbf{0}$, $\mathbf{1}$ enthält und Negationszeichen nur unmittelbar vor Variablen auftreten.

Negations-normalform NNF

Rekursiv lässt sich die Menge der Formeln in NNF folgendermaßen definieren.

Basisregeln: Für jedes $X \in \textsc{Avar}$ ist sowohl X als auch $\neg X$ eine Formel in NNF.

Rekursive Regeln: Sind φ und ψ Formeln in NNF, so sind auch $(\varphi \wedge \psi)$ und $(\varphi \vee \psi)$ Formeln in NNF.

Beobachtung 3.37. Jede Formel, die in KNF oder in DNF ist, ist auch in NNF. Aus Satz 3.35 folgt also insbesondere, dass jede aussagenlogische Formel äquivalent zu einer Formel in NNF ist.

Beachte: Nicht jede Formel in NNF ist auch in KNF oder in DNF.

Beispiel: $\Big(\big((X \wedge \neg Y) \vee (\neg X \wedge Y)\big) \wedge \neg Z\Big)$ ist in NNF, aber weder in KNF noch in DNF.

Ein einfaches Verfahren zur Transformation einer gegebenen aussagenlogischen Formel in eine äquivalente Formel in NNF beruht auf der wiederholten Anwendung der De Morganschen Regeln (Satz 3.31(g)) und der Regel für "doppelte Negation" (Satz 3.31(f)): Mit den De Morganschen Regeln ($\neg(\varphi \wedge \psi) \equiv (\neg\varphi \vee \neg\psi)$ bzw. $\neg(\varphi \vee \psi) \equiv (\neg\varphi \wedge \neg\psi)$) ziehen wir das Negationszeichen nach innen; mit der "doppelte Negation"-Regel ($\neg\neg\varphi \equiv \varphi$) können wir Schritt für Schritt mehrfach hintereinander vorkommende Negationszeichen eliminieren. Eventuell in der Formel vorkommende Implikationspfeile "$\to$" oder Biimplikationspfeile "$\leftrightarrow$"eliminieren wir durch Verwenden von Satz 3.31(k) und (l). Eventuelle Vorkommen der Symbole $\mathbf{0}$ bzw. $\mathbf{1}$ ersetzen wir durch die Formel $(V_0 \wedge \neg V_0)$ bzw. $(V_0 \vee \neg V_0)$.

Beispiel 3.38.

Ziel: Bringe die Formel $\Big(\big(\neg V_0 \wedge \neg((V_0 \vee V_1) \to V_0)\big) \to \mathbf{0}\Big)$ in NNF, d.h. finde eine zur gegebenen Formel äquivalente Formel in NNF.

Lösung:

$$
\begin{aligned}
\Big(\big(\neg V_0 \wedge \neg((V_0 \vee V_1) \to V_0)\big) \to \underline{\mathbf{0}}\Big) &\equiv \Big(\big(\neg V_0 \wedge \neg((V_0 \vee V_1) \to V_0)\big) \underline{\to} (V_0 \wedge \neg V_0)\Big)\\
&\equiv \Big(\neg\big(\neg V_0 \wedge \neg((V_0 \vee V_1) \underline{\to} V_0)\big) \vee (V_0 \wedge \neg V_0)\Big)\\
&\equiv \Big(\underline{\neg}\big(\neg V_0 \wedge \neg(\neg(V_0 \vee V_1) \vee V_0)\big) \vee (V_0 \wedge \neg V_0)\Big)\\
&\equiv \Big(\big(\underline{\neg\neg} V_0 \vee \underline{\neg\neg}(\neg(V_0 \vee V_1) \vee V_0)\big) \vee (V_0 \wedge \neg V_0)\Big)\\
&\equiv \Big(\big(V_0 \vee (\underline{\neg}(V_0 \vee V_1) \vee V_0)\big) \vee (V_0 \wedge \neg V_0)\Big)\\
&\equiv \Big(\big(V_0 \vee ((\neg V_0 \wedge \neg V_1) \vee V_0)\big) \vee (V_0 \wedge \neg V_0)\Big) \quad \text{in NNF.}
\end{aligned}
$$

Unter zusätzlicher Verwendung der "Distributivitätsregel" (Satz 3.31(e)) erhält man Verfahren zur Transformation einer gegebenen Formel in eine äquivalente Formel in DNF bzw. KNF, bei denen man nicht zuerst eine Wahrheitstafel aufstellen muss. Diese Verfahren sind vor allem dann ratsam, wenn die gegebene Formel sehr viele verschiedene Variablen enthält, die zugehörige Wahrheitstafel also sehr groß wird.

Algorithmus 3.39 (Ein KNF-Algorithmus)**.**

Eingabe: Eine aussagenlogische Formel φ.

Ausgabe: Eine zu φ äquivalente Formel φ' in KNF.

Verfahren:

(1) Konstruiere eine zu φ äquivalente Formel φ' in NNF (beispielsweise mit dem in Beobachtung 3.37 beschriebenen Verfahren).

(2) Wiederhole folgende Schritte:

   (i) Falls φ' in KNF ist, so halte mit Ausgabe φ'.

   (ii) Ersetze eine Teilformel von φ' der Gestalt $\big(\psi_1 \vee (\psi_2 \wedge \psi_3)\big)$ durch $\big((\psi_1 \vee \psi_2) \wedge (\psi_1 \vee \psi_3)\big)$ oder ersetze eine Teilformel von φ' der Gestalt $\big((\psi_2 \wedge \psi_3) \vee \psi_1\big)$ durch $\big((\psi_2 \vee \psi_1) \wedge (\psi_3 \vee \psi_1)\big)$. Sei φ'' die resultierende Formel.

   (iii) Setze $\varphi' := \varphi''$.

Algorithmus 3.40 (Ein DNF-Algorithmus)**.**

Eingabe: Eine aussagenlogische Formel φ

Ausgabe: Eine zu φ äquivalente Formel φ' in DNF.

Verfahren:

(1) Konstruiere eine zu φ äquivalente Formel φ' in NNF.

(2) Wiederhole folgende Schritte:

   (i) Falls φ' in DNF ist, so halte mit Ausgabe φ'.

   (ii) Ersetze eine Teilformel von φ' der Gestalt $\big(\psi_1 \wedge (\psi_2 \vee \psi_3)\big)$ durch $\big((\psi_1 \wedge \psi_2) \vee (\psi_1 \wedge \psi_3)\big)$ oder ersetze eine Teilformel von φ' der Gestalt $\big((\psi_2 \vee \psi_3) \wedge \psi_1\big)$ durch $\big((\psi_2 \wedge \psi_1) \vee (\psi_3 \wedge \psi_1)\big)$. Sei φ'' die resultierende Formel.

   (iii) Setze $\varphi' := \varphi''$.

Satz 3.41 (Korrektheit der Algorithmen 3.39 und 3.40)**.** *Für jede aussagenlogische Formel φ gilt:*

(a) Algorithmus 3.39 hält bei Eingabe der Formel φ nach endlich vielen Schritten an und gibt eine zu φ äquivalente Formel in KNF aus.

(b) Algorithmus 3.40 hält bei Eingabe der Formel φ nach endlich vielen Schritten an und gibt eine zu φ äquivalente Formel in DNF aus.

(hier ohne Beweis)

Beispiel 3.42. Sei $\varphi := \Big(\big(\neg V_0 \wedge (V_0 \to V_1)\big) \vee (V_2 \to V_3)\Big)$.

Transformation von φ in NNF:

$$\varphi \;=\; \Big(\big(\neg V_0 \wedge (V_0 \underline{\to} V_1)\big) \vee (V_2 \underline{\to} V_3)\Big) \;\equiv\; \Big(\big(\neg V_0 \wedge (\neg V_0 \vee V_1)\big) \vee (\neg V_2 \vee V_3)\Big) \;=:\; \varphi'.$$

Transformation von φ in DNF mittels Algorithmus 3.40:

(1) Transformation von φ in NNF $\rightsquigarrow$ liefert $\varphi' = \Big(\underline{\big(\neg V_0 \wedge (\neg V_0 \vee V_1)\big)} \vee (\neg V_2 \vee V_3)\Big)$

(2) 1-maliges Anwenden von Zeile (ii) des Algorithmus auf φ' liefert:

$$\varphi'' \ := \ \Big(\big((\underline{\neg V_0 \wedge \neg V_0}) \vee (\underline{\neg V_0 \wedge V_1})\big) \vee (\underline{\neg V_2 \vee V_3})\Big).$$

Diese Formel ist die DNF-Formel, die von dem Algorithmus ausgegeben wird.

Transformation von φ in KNF mittels Algorithmus 3.39:

(1) Transformation von φ in NNF $\rightsquigarrow$ liefert $\varphi' = \Big(\big(\neg V_0 \wedge (\neg V_0 \vee V_1)\big) \underline{\vee} (\neg V_2 \vee V_3)\Big)$

(2) 1-maliges Anwenden von Zeile (ii) des Algorithmus auf φ' liefert:

$$\varphi'' \ := \ \Big(\underline{\big(\neg V_0 \vee (\neg V_2 \vee V_3)\big)} \wedge \underline{\big((\neg V_0 \vee V_1) \vee (\neg V_2 \vee V_3)\big)}\Big).$$

Dies ist die KNF-Formel, die von dem Algorithmus ausgegeben wird.

Unmittelbar vor Definition 3.21 wurde darauf hingewiesen, dass die Aufgabe, für eine gegebene Formel φ herauszufinden, ob sie erfüllbar ist, im Allgemeinen ein recht schwieriges Problem ist. Für den Spezialfall, dass φ eine Formel in DNF ist, lässt sich das Erfüllbarkeitsproblem allerdings sehr effizient lösen, wie die folgende Beobachtung zeigt.

Beobachtung 3.43 (effizienter Erfüllbarkeitstest für DNF-Formeln)**.** Sei φ eine Formel in DNF, d.h. φ ist von der Form

$$\bigvee_{i=1}^{n} \left(\bigwedge_{j=1}^{m_i} l_{i,j} \right), \quad \text{für Literale } l_{i,j}.$$

D.h. φ ist von der Form

$$\kappa_1 \vee \cdots \vee \kappa_n,$$

wobei für jedes $i \in \{1, \ldots, n\}$ κ_i die konjunktive Klausel

$$\kappa_i \ := \ l_{i,1} \wedge \cdots \wedge l_{i,n}$$

ist.

Klar: φ ist erfüllbar $\iff$ für mindestens ein $i \in \{1, \ldots, n\}$ ist κ_i erfüllbar. Da κ_i eine Konjunktion von Literalen (d.h. von Variablen und/oder negierten Variablen) ist, gilt:

$$\kappa_i \text{ ist erfüllbar} \iff \text{es gibt keine } j, j' \in \{1, \ldots, m_i\}, \text{ so dass } l_{i,j} = \neg l_{i,j'}.$$

Daher ist der folgende Algorithmus dazu geeignet, zu testen, ob eine gegebene DNF-Formel erfüllbar ist.

Eingabe: Eine aussagenlogische Formel $\varphi = \bigvee_{i=1}^{n} \left(\bigwedge_{j=1}^{m_i} l_{i,j} \right)$ in DNF

Ziel: Entscheide, ob φ erfüllbar ist.

Verfahren:

(1) Für $i = 1, \ldots, n$

(2) Für $j = 1, \ldots, m_i$

(3) Für $j' = j+1, \ldots, m_i$

(4) Falls $l_{i,j} = \neg l_{i,j'}$ oder $l_{i,j'} = \neg l_{i,j}$, dann:

(5) Falls $i = n$ ist, so mache in Zeile 7 weiter;
ansonsten setze $i := i+1$ und mache in Zeile 2 weiter.

(6) Halte mit Ausgabe "φ ist erfüllbar".

(7) Halte mit Ausgabe "φ ist unerfüllbar".

Um aussagenlogische Formeln φ von **beliebiger** Form auf Erfüllbarkeit zu testen, kann man dann folgendermaßen vorgehen:

Schritt 1: Transformiere φ in eine äquivalente Formel φ' in DNF (z.B. mit Algorithmus 3.40).

Schritt 2: Entscheide, ob φ' erfüllbar ist (z.B. mit dem obigen Verfahren).

Das Ausführen von Schritt 1 kann dabei u.U. aber leider wieder sehr lange dauern, da es einige Formeln gibt, zu denen äquivalente Formeln in DNF zwangsläufig sehr groß sind. Dies wird durch den folgenden Satz präzisiert:

Satz 3.44. *Sei $n \in \mathbb{N}_{>0}$, seien $X_1, \ldots, X_n, Y_1, \ldots, Y_n$ genau $2 \cdot n$ verschiedene aussagenlogische Variablen, und sei*

$$\varphi_n \;:=\; \bigwedge_{i=1}^{n} (X_i \leftrightarrow Y_i).$$

Dann hat jede zu φ_n äquivalente Formel in DNF mindestens 2^n konjunktive Klauseln.

Beweis: Übung. □

3.5 Übungsaufgaben zu Kapitel 3

Aufgabe 3.1.

(a) Welche der folgenden Wörter gehören gemäß Definition 3.3 zur Sprache AL, welche nicht?

- $(V_1 \wedge \mathbf{1})$
- $(V_1 \wedge \mathbf{101})$
- $(\neg(V_1 \wedge V_2) \vee V_3)$
- $\neg(V_1 \wedge V_2) \vee V_3$
- $(V_1 \to V_2)$
- $(V_1 \to V_2 \to V_3)$
- $(V_1 \leftarrow V_2)$
- $(V_1 \leftrightarrow V_2)$

(b) Beweisen Sie, dass für die Formel $\varphi := \big((V_1 \leftrightarrow \mathbf{1}) \wedge (V_1 \to (V_2 \wedge \mathbf{0}))\big)$ gilt: $\varphi \in \mathrm{AL}$.

(c) Betrachten Sie die Formel φ aus (b) und die Belegung $\mathcal{B}\colon \mathrm{Var}(\varphi) \to \{0,1\}$ mit $\mathcal{B}(V_1) = 1$ und $\mathcal{B}(V_2) = 0$. Berechnen Sie den Wert $[\![\varphi]\!]^{\mathcal{B}}$.

(d) Geben Sie den Syntaxbaum der Formel φ aus (b) an.

Aufgabe 3.2.

(a) Betrachten Sie die folgenden Wörter und beweisen Sie jeweils, dass das Wort gemäß Definition 3.3 zur Sprache AL gehört oder begründen Sie, warum das Wort nicht zu AL gehört.

(i) $\neg\Big((V_3 \wedge \neg\mathbf{0}) \to \big(V_0 \vee (\neg\neg V_1 \wedge V_4)\big)\Big)$

(ii) $(V_5 \leftrightarrow X) \wedge \big(V_{23} \to (V_1 \wedge \mathbf{0})\big)$

(iii) $\Big((V_{11} \leftarrow V_7) \vee \neg\neg V_5\Big)$

(iv) $\big((V_9 \vee \neg(\neg V_{42}) \vee \neg V_2) \to \mathbf{1}\big)$

(b) Betrachten Sie die aussagenlogische Formel

$$\varphi \;:=\; \Big((\neg V_0 \wedge V_1) \to \big(V_0 \wedge (V_1 \vee \neg V_2)\big)\Big)$$

und die Belegung $\mathcal{B}\colon \operatorname{Var}(\varphi) \to \{0,1\}$ mit $\mathcal{B}(V_0) = 1$ und $\mathcal{B}(V_1) = \mathcal{B}(V_2) = 0$. Berechnen Sie den Wert $[\![\varphi]\!]^{\mathcal{B}}$ in nachvollziehbaren Schritten analog zu Beispiel 3.9.

(c) Geben Sie den Syntaxbaum und die ASCII-Darstellung der Formel φ aus (b) an.

Aufgabe 3.3. Schon kurz nach der Geburt von Herakles und Eurystheus entstand ein Streit, wer von den beiden der rechtmäßige Herrscher sei. Dazu wurden die drei bekanntesten Orakel Griechenlands befragt.

Das Ammonion gab bekannt, dass die Orakelsprüche aus Klaros grundsätzlich falsch seien. Ebenso ließ das Orakel aus Klaros verlauten, dass die Orakelsprüche aus Delphi samt und sonders unzutreffend seien. Das Orakel aus Delphi jedoch behauptete, sowohl die Sprüche des Ammonions als auch die des Orakels in Klaros seien unwahr.

Wem sollen die armen Griechen nun glauben?

(a) Zerlegen Sie den obigen Text in atomare Aussagen und geben Sie eine aussagenlogische Formel φ an, die das im Text zusammengefasste Wissen repräsentiert (ähnlich wie in den Beispielen 3.1, 3.17 und 3.19).

(b) Geben Sie für Ihre Formel φ aus (a) eine Belegung $\mathcal{B}$ an, die besagt, dass das Ammonion die Wahrheit sagt und die beiden anderen Orakel lügen. Erfüllt $\mathcal{B}$ die Formel φ?

(c) Welchen Orakeln können die Griechen glauben, welchen nicht? Falls es mehrere Möglichkeiten gibt, geben Sie alle an.

Aufgabe 3.4. USA, 4. November 2008. Vor einem Wahllokal befragt ein Journalist vier Freunde A, B, C und D, die gerade das Wahllokal verlassen haben, wie sie gewählt haben. A sagt: „Falls B für Obama gestimmt hat, dann haben auch C und D für Obama gestimmt.“ B sagt: „A hat auf keinen Fall für Obama gestimmt, aber D.“ C sagt: „B hat nur dann für McCain gestimmt, wenn A für Obama gestimmt hat.“ D sagt schließlich: „Wenn C für Obama gestimmt hat, dann hat A für McCain oder B für Obama gestimmt.“ Wir nehmen an, dass jeder die Wahrheit gesagt und entweder Obama oder McCain gewählt hat.

(a) Zerlegen Sie den obigen Text in atomare Aussagen und geben Sie eine aussagenlogische Formel φ an, die das im Text zusammengefasste Wissen repräsentiert (ähnlich wie in den Beispielen 3.1, 3.17 und 3.19).

(b) Geben Sie für Ihre Formel φ aus (a) eine Belegung $\mathcal{B}$ an, die besagt, dass A, B und C Obama gewählt haben und D für McCain gestimmt hat. Erfüllt $\mathcal{B}$ die Formel φ?

(c) Wen haben A, B, C und D jeweils gewählt? Falls es mehrere Möglichkeiten gibt, geben Sie alle an.

Aufgabe 3.5. Auf der Insel Wafa leben zwei Stämme: Die Was, die immer die Wahrheit sagen, und die Fas, die immer lügen. Ein Reisender besucht die Insel und kommt mit drei Einwohnern A, B, C ins Gespräch. Der Reisende schreibt daraufhin folgende atomare Aussagen in sein Notizbuch:

- X_A: A sagt die Wahrheit
- X_B: B sagt die Wahrheit
- X_C: C sagt die Wahrheit

(a) Sei $\mathcal{B}\colon \{X_A, X_B, X_C\} \to \{0,1\}$ die Belegung mit $\mathcal{B}(X_A) = 1$, $\mathcal{B}(X_B) = 0$ und $\mathcal{B}(X_C) = 0$. Beschreiben Sie umgangssprachlich, welcher Sachverhalt durch die Belegung $\mathcal{B}$ ausgedrückt wird. Was folgt daraus über die Stammesangehörigkeit der drei Einwohner A, B und C?

Die Informationen, die der Reisende im Gespräch erhalten hat, fasst er durch folgende aussagenlogische Formeln zusammen:

- $\varphi_A := \big(X_A \leftrightarrow (\neg X_B \vee \neg X_C)\big)$
- $\varphi_B := \big(X_B \leftrightarrow (X_A \to X_C)\big)$
- $\varphi_C := \big(X_C \leftrightarrow (\neg X_B \to X_A)\big)$

Er merkt an, dass die durch $\varphi_A, \varphi_B, \varphi_C$ formalisierten Aussagen der Wahrheit entsprechen.

(b) Beschreiben Sie umgangssprachlich, was jede der Formeln $\varphi_A, \varphi_B, \varphi_C$ aussagt.

(b) Zu welchen Stämmen gehören A, B und C?

Aufgabe 3.6. Zwei Analysten streiten sich, wer von ihnen denn nun am besten Aktienkurse voraussagen kann. Dazu wollen sie drei zufällig anwesende Anleger A, B und C befragen. Das wäre nicht weiter schwierig, wenn sich A, B und C nicht folgendes (repräsentiert durch aussagenlogische Formeln) vorwerfen würden:

- A behauptet: $\varphi_A := (\neg B \vee \neg C)$
- B behauptet: $\varphi_B := \neg A$
- C behauptet: $\varphi_C := (A \wedge \neg B)$

Hierbei bedeuten die Aussagenvariablen:

- A: A sagt die Wahrheit.
- B: B sagt die Wahrheit.
- C: C sagt die Wahrheit.

(a) Beschreiben Sie umgangssprachlich, was jede der Formeln φ_A, φ_B, φ_C aussagt.

(b) Wem können die Analysten glauben und wem nicht? Falls es mehrere Möglichkeiten gibt, geben Sie alle an.

Aufgabe 3.7. Geben Sie für jede der folgenden aussagenlogischen Formeln eine Wahrheitstafel und alle erfüllenden Belegungen $\mathcal{B}\colon \operatorname{Var}(\varphi_1) \to \{0,1\}$ (für (a)) bzw. $\mathcal{B}\colon \operatorname{Var}(\varphi_2) \to \{0,1\}$ (für (b)) an.

(a) $\varphi_1 := \Big(\big((V_1 \leftrightarrow \neg V_2) \wedge (\neg V_2 \vee V_3)\big) \wedge (V_3 \to V_1)\Big)$

(b) $\varphi_2 := \Big((V_1 \leftrightarrow \mathbf{1}) \wedge \big(V_1 \to (V_2 \wedge \mathbf{0})\big)\Big)$

Aufgabe 3.8.

(a) Geben Sie für jede der folgenden aussagenlogischen Formeln an, ob sie erfüllbar, unerfüllbar und/oder allgemeingültig ist.

- $(V_0 \wedge \neg V_1)$
- $\big(V_0 \leftrightarrow (\mathbf{1} \to V_0)\big)$
- $\big(V_0 \leftrightarrow (V_0 \to \mathbf{0})\big)$
- $\Big(V_1 \vee \big((V_0 \wedge V_1) \to V_2\big)\Big)$
- $\big((V_0 \to V_1) \leftrightarrow (\neg V_1 \to \neg V_0)\big)$

(b) Für jedes $n \in \mathbb{N}$ sei die aussagenlogische Formel φ_n definiert durch

$$\varphi_n \ := \begin{cases} (V_n \vee V_{n+1}), & \text{falls } n \text{ gerade} \\ (V_n \to \neg V_{n+1}), & \text{falls } n \text{ ungerade.} \end{cases}$$

Es gilt also

$$\varphi_0 = (V_0 \vee V_1), \quad \varphi_1 = (V_1 \to \neg V_2), \quad \varphi_2 = (V_2 \vee V_3), \quad \varphi_3 = (V_3 \to \neg V_4), \quad \cdots$$

Geben Sie eine Belegung $\mathcal{B}$ an, so dass für alle $n \in \mathbb{N}$ gilt: $\mathcal{B}$ erfüllt φ_n.

(c) Beweisen oder widerlegen Sie die folgende Behauptung:

$$\big((\neg V_0 \vee V_2) \wedge (V_1 \to \neg V_2)\big) \ \models \ \neg\big((V_0 \wedge \neg V_1) \to \neg(V_0 \to V_2)\big)$$

Aufgabe 3.9.

(a) Geben Sie für jede der folgenden aussagenlogischen Formeln an, ob sie erfüllbar, unerfüllbar und/oder allgemeingültig ist.

- $\neg V_1$
- $\big((V_0 \vee \neg V_1) \leftrightarrow V_2\big)$
- $\big(\neg V_0 \to (V_0 \to V_1)\big)$
- $\big(V_0 \wedge (V_0 \to \neg V_0)\big)$
- $\Big((V_0 \to V_1) \leftrightarrow \big((V_0 \wedge \neg V_1) \to \mathbf{0}\big)\Big)$

(b) Für jedes $n \in \mathbb{N}$ sei die aussagenlogische Formel φ_n definiert durch

$$\varphi_n \ := \begin{cases} (V_n \leftrightarrow V_{n+2}), & \text{falls } n \text{ gerade} \\ (V_n \leftrightarrow \neg V_{n-1}), & \text{falls } n \text{ ungerade.} \end{cases}$$

Es gilt also

$$\varphi_0 = (V_0 \leftrightarrow V_2), \quad \varphi_1 = (V_1 \leftrightarrow \neg V_0), \quad \varphi_2 = (V_2 \leftrightarrow V_4), \quad \varphi_3 = (V_3 \leftrightarrow \neg V_2), \quad \cdots$$

Geben Sie eine Belegung $\mathcal{B}\colon \operatorname{Var} \to \{0,1\}$ an, so dass für alle $n \in \mathbb{N}$ gilt: $\mathcal{B}$ erfüllt φ_n.

(c) Beweisen oder widerlegen Sie die folgenden Behauptungen:

(i) $\big(\neg(V_0 \leftrightarrow V_1) \wedge (\neg V_2 \vee V_0)\big) \;\models\; \big(V_0 \vee (V_1 \wedge \neg V_2)\big)$

(ii) $\big(\neg(V_0 \leftrightarrow V_1) \wedge (\neg V_2 \vee V_0)\big) \;\equiv\; \big(V_0 \vee (V_1 \wedge \neg V_2)\big)$

Aufgabe 3.10. Beweisen Sie Beobachtung 3.27 (b) und (d), d.h. beweisen Sie, dass für alle aussagenlogischen Formeln φ und ψ gilt:

(a) $\varphi \models \mathbf{0} \iff \varphi$ ist unerfüllbar.

(b) $\varphi \models \psi \iff (\varphi \wedge \neg\psi)$ ist unerfüllbar.

Aufgabe 3.11. Betrachten Sie die folgenden beiden Aussagen:

(a) Wenn der Rechner einen Virus hat oder nicht mehr funktioniert, und wenn der Administrator erreichbar ist, dann rufen wir den Administrator.

(b) Wenn der Rechner einen Virus hat, so rufen wir den Administrator, falls wir ihn erreichen; und wenn der Administrator erreichbar ist und der Rechner nicht funktioniert, so rufen wir den Administrator.

(a) Formalisieren Sie jede der beiden Aussagen (1), (2) durch eine aussagenlogische Formel.

(b) Zeigen Sie, dass die beiden Aussagen (1) und (2) äquivalent sind.

Aufgabe 3.12 (Modellierung und Folgerung)**.** Einer Ihrer Bekannten berichtet von seiner Zimmersuche in Frankfurt und äußert Ihnen gegenüber folgende Aussagen, die auf alle der von ihm besichtigten Wohnungen zutreffen:

- Wenn es sich um eine 1-Zimmer-Wohnung handelt, dann stehen höchstens 26 m^2 Wohnraum zur Verfügung oder der Mietpreis ist höher als 400 €.
- Wenn sich das Zimmer nicht in einer 1-Zimmer-Wohnung befindet, dann ist das Zimmer in einer WG.
- Wenn mehr als 26 m^2 Wohnraum zur Verfügung stehen, dann liegt das Zimmer nicht in einer WG.
- Wenn mehr als 26 m^2 Wohnraum zur Verfügung stehen und der Mietpreis höher als 400 € ist, dann handelt es sich nicht um eine 1-Zimmer-Wohnung.

(a) Zerlegen Sie den obigen Text in atomare Aussagen und geben Sie eine aussagenlogische Formel φ an, die das im Text zusammengefasste Wissen repräsentiert.

Betrachten Sie nun die nachfolgenden Aussagen:

- In jeder besichtigten Wohnung stehen Ihrem Bekannten maximal 26 m^2 zur Verfügung.
- Für jede besichtigte Wohnung gilt: Wenn die Wohnung in einer WG liegt, dann beträgt der Mietpreis höchstens 400 €.
- Für jede besichtigte Wohnung gilt: Wenn der verlangte Mietpreis höchstens 400 € beträgt, dann handelt es sich um eine WG oder um eine 1-Zimmer-Wohnung.

(b) Geben Sie für jede der drei Aussagen eine aussagenlogische Formel an, die die Aussage repräsentiert.

(c) Entscheiden Sie für jede der aussagenlogischen Formeln aus (b), ob sie aus der Formel φ in (a) folgt.

Aufgabe 3.13. Es sei $\varphi := \big((V_0 \vee \neg V_2) \to V_1\big)$.

(a) Wandeln Sie φ mittels Wahrheitstafel in eine äquivalente aussagenlogische Formel in DNF um.

(b) Wenden Sie Algorithmus 3.39 an, um eine zu φ äquivalente Formel in KNF zu finden.

Aufgabe 3.14. Betrachten Sie die aussagenlogische Formel

$$\varphi \;:=\; \big(\neg(V_0 \leftrightarrow V_1) \wedge (\neg V_2 \vee V_0)\big).$$

(a) Wandeln Sie φ mittels Wahrheitstafel in eine äquivalente aussagenlogische Formel in DNF um.

(b) Wenden Sie Algorithmus 3.39 an, um eine zu φ äquivalente Formel in KNF zu finden.

Aufgabe 3.15. Für jedes $n \in \mathbb{N}_{>0}$ sei die aussagenlogische Formel φ_n definiert durch

$$\varphi_n \;:=\; \bigwedge_{i=1}^{n} (X_i \leftrightarrow Y_i).$$

(a) Beschreiben Sie die erfüllenden Belegungen $\mathcal{B}\colon \operatorname{Var}(\varphi_n) \to \{0,1\}$ für φ_n. Wie viele solche Belegungen gibt es?

(b) Geben Sie eine zu φ_n äquivalente Formel in DNF an.

(c) Beweisen Sie Satz 3.44, d.h. zeigen Sie, dass jede zu φ_n äquivalente Formel in DNF mindestens 2^n konjunktive Klauseln hat.

Hinweis: Eine Möglichkeit, dies zu zeigen, ist einen Beweis durch Widerspruch zu führen. Nehmen Sie dazu an, dass ψ_n eine zu φ_n äquivalente Formel in DNF ist, die aus weniger als 2^n konjunktiven Klauseln besteht. D.h. es gibt eine natürliche Zahl $N < 2^n$ und N konjunktive Klauseln $\kappa_1, \dots, \kappa_N$, so dass $\psi_n = \kappa_1 \vee \cdots \vee \kappa_N$. Folgern Sie aus Ihrer Antwort aus Teil (a), dass mindestens eine der Klauseln $\kappa_1, \dots, \kappa_N$ von mindestens zwei verschiedenen die Formel φ_n erfüllenden Belegungen wahr gemacht wird. Leiten Sie daraus einen Widerspruch her.

4 Graphen und Bäume

Bei Modellierungsaufgaben geht es oft darum, **Objekte** sowie **Beziehungen zwischen Objekten** zu beschreiben. **Graphen** und **Bäume** (Bäume sind Graphen mit bestimmten Eigenschaften) eignen sich dazu oft besonders gut.

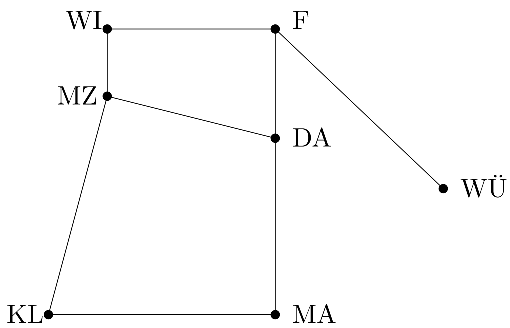
Anschaulich besteht ein Graph aus **Knoten** und **Kanten:**

- “Knoten” repräsentieren dabei “gleichartige Objekte”.
- “Kanten” repräsentieren Beziehungen zwischen je zwei “Objekten”.

Je nach Aufgabenstellung werden **ungerichtete Graphen** oder **gerichtete Graphen** verwendet.

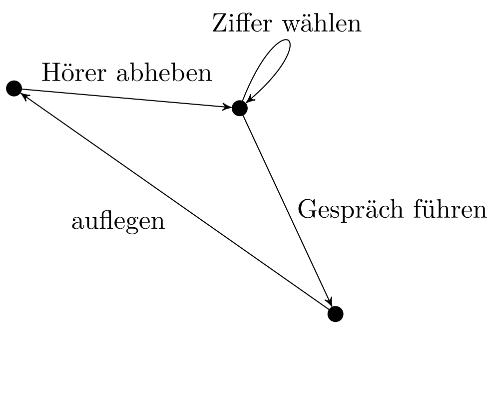
Beispiel 4.1.

(a) **Skizze eines ungerichteten Graphen**, der die Autobahnverbindungen zwischen einigen Städten darstellt:



DA	$\widehat{=}$	Darmstadt
F	$\widehat{=}$	Frankfurt
KL	$\widehat{=}$	Kaiserslautern
MA	$\widehat{=}$	Mannheim
MZ	$\widehat{=}$	Mainz
WI	$\widehat{=}$	Wiesbaden
WÜ	$\widehat{=}$	Würzburg

(b) **Skizze eines gerichteten Graphen**, der den prinzipiellen Ablauf eines Telefonats darstellt:



4.1 Graphen

4.1.1 Grundlegende Definitionen

Definition 4.2 (ungerichteter Graph)**.** Ein **ungerichteter Graph** $G = (V, E)$ besteht aus einer Menge V, die **Knotenmenge von G** genannt wird, und einer Menge ungerichteter Graph

$$E \subseteq \big\{\{i, j\} : i \in V,\, j \in V,\, i \neq j\big\},$$

die **Kantenmenge von G** genannt wird. Die Elemente aus V heißen **Knoten** von G (auch: "Ecken"; englisch: **vertices**, singular: vertex); die Elemente aus E heißen **Kanten** von G (englisch: **edges**, singular: edge). Knoten Kanten

Beispiel 4.3. $G = (V, E)$ mit

$$\begin{aligned} V &:= \big\{\mathrm{MZ}, \mathrm{WI}, \mathrm{MA}, \mathrm{DA}, \mathrm{KL}, \mathrm{F}, \mathrm{WÜ}\big\} \quad \text{und} \\ E &:= \big\{\{\mathrm{MZ}, \mathrm{WI}\}, \{\mathrm{WI}, \mathrm{F}\}, \{\mathrm{F}, \mathrm{DA}\}, \{\mathrm{F}, \mathrm{WÜ}\}, \{\mathrm{MZ}, \mathrm{DA}\}, \{\mathrm{MZ}, \mathrm{KL}\}, \{\mathrm{KL}, \mathrm{MA}\}, \{\mathrm{DA}, \mathrm{MA}\}\big\} \end{aligned}$$

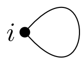
ist ein ungerichteter Graph, der die Autobahnverbindungen zwischen Mainz (MZ), Wiesbaden (WI), Mannheim (MA), Darmstadt (DA), Kaiserslautern (KL), Frankfurt (F) und Würzburg (WÜ) repräsentiert.

Beispiel 4.1(a) zeigt diesen Graphen G in **graphischer Darstellung**: Knoten werden als Punkte dargestellt, Kanten als Verbindungslinien zwischen Punkten.

Beachte: Laut Definition 4.2 gibt es zwischen zwei Knoten i und j aus V

- **höchstens** eine Kante; diese wird mit $\{i, j\}$ bezeichnet und graphisch dargestellt als

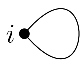
- **keine** Kante, falls $i = j$ ist. In der graphischen Darstellung eines ungerichteten Graphs sind also "Schleifen" der Form



  nicht erlaubt.

Jede Kante $\{i, j\}$ eines ungerichteten Graphen ist also eine 2-elementige Menge von Knoten des Graphen.

Bemerkung. Diese Definition ungerichteter Graphen wird in den meisten Büchern verwendet. In manchen Büchern werden davon abweichend in ungerichteten Graphen auch "Schleifen" der Form



erlaubt.

Notation 4.4. Sei $G = (V, E)$ ein ungerichteter Graph.

- Ein Knoten $v \in V$ heißt **inzident** mit einer Kante $e \in E$, falls $v \in e$. inzident
- Die beiden mit einer Kante $e \in E$ inzidenten Knoten nennen wir die **Endknoten** von e, und wir sagen, dass e diese beiden Knoten verbindet. Endknoten

benachbart
adjazent

- Zwei Knoten $v, v' \in V$ heißen **benachbart** (bzw. **adjazent**), falls es eine Kante $e \in E$ gibt, deren Endknoten v und v' sind (d.h. $e = \{v, v'\}$).

Grad
$\text{Grad}_G(v)$

Definition 4.5 (Grad)**.** Sei $G = (V, E)$ ein ungerichteter Graph und sei $v \in V$ ein Knoten von G. Der **Grad von v in G** (engl.: degree), kurz: $\text{Grad}_G(v)$, ist die Anzahl der Kanten, die v als Endknoten haben. D.h.

$$\text{Grad}_G(v) \;:=\; |\{e \in E : v \in e\}|.$$

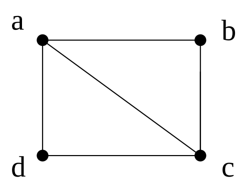
Grad(G)

Der **Grad von G** ist

$$\text{Grad}(G) \;:=\; \max\{\text{Grad}_G(v) : v \in V\},$$

d.h. Grad(G) gibt den maximalen Grad eines Knotens von G an.

Beispiel.



$\text{Grad}_G(a) = 3$
$\text{Grad}_G(b) = 2$
$\text{Grad}_G(c) = 3$
$\text{Grad}_G(d) = 2$

$\text{Grad}(G) = 3$

gerichteter Graph

Definition 4.6 (gerichteter Graph)**.** Ein **gerichteter Graph** $G = (V, E)$ besteht aus einer Menge V, die **Knotenmenge von G** genannt wird, und einer Menge

$$E \;\subseteq\; \big\{(i, j) : i \in V,\, j \in V\big\},$$

Knoten
(gerichtete) Kante

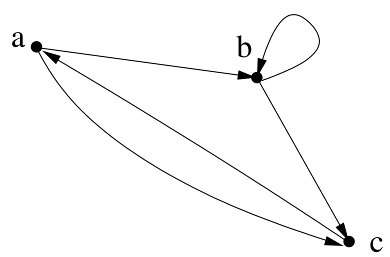
die **Kantenmenge von G** genannt wird. Die Elemente aus V heißen **Knoten** (bzw. "Ecken"), die Elemente aus E heißen (gerichtete) **Kanten** von G.

Beispiel 4.7. $G = (V, E)$ mit

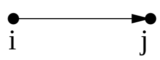
$$\begin{aligned} V &:= \{a, b, c\} \quad \text{und} \\ E &:= \big\{(a,b), (b,b), (b,c), (c,a), (a,c)\big\} \end{aligned}$$

ist ein gerichteter Graph.

Graphische Darstellung:



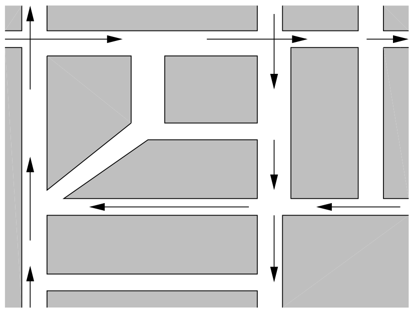
Knoten werden dabei als Punkte dargestellt; eine Kante der Form (i, j) wird als Pfeil von Knoten i nach Knoten j dargestellt, also



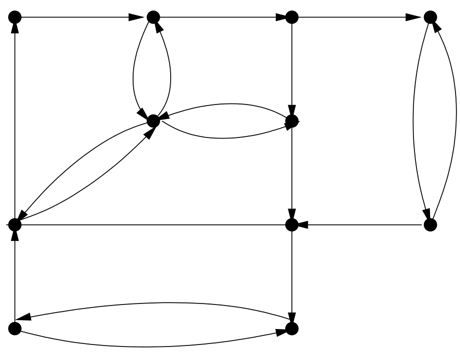
Notation 4.8. Sei $G = (V, E)$ ein gerichteter Graph.

- Ist $e = (i, j) \in E$, so heißt i der Ausgangsknoten von e und j der Endknoten von e, und wir sagen, dass e von i nach j verläuft.
- Ein Knoten V heißt **inzident** mit einer Kante $e \in E$, falls v der Ausgangs- oder Endknoten von e ist. inzident
- Zwei Knoten $v, v' \in V$ heißen **benachbart** (bzw. **adjazent**), falls $(v, v') \in E$ oder $(v', v) \in E$. benachbart adjazent
- Eine Kante der Form (v, v) (d.h. deren Ausgangs- und Endpunkt identisch ist) wird **Schleife** bzw. **Schlinge** genannt. Schleife Schlinge

Beispiel 4.9 (Modellierung durch gerichtete Graphen)**.** In der folgenden Straßenkarte sind Einbahnstraßen durch Pfeile markiert.



Diese Straßenkarte können wir durch einen gerichteten Graphen repräsentieren, der für jede Straßenkreuzung einen Knoten enthält, und in dem es eine Kante von “Kreuzung” i zu “Kreuzung” j gibt, falls man von i nach j fahren kann, ohne zwischendurch eine weitere Kreuzung zu passieren. Graphisch lässt sich dieser gerichtete Graph folgendermaßen darstellen:



Weitere Beispiele zur Modellierung durch Graphen:

- Computer-Netzwerk:
  Knoten repräsentieren Computer; Kanten repräsentieren Netzwerkverbindungen.
- das World Wide Web:
  Knoten repräsentieren Webseiten; Kanten repräsentieren Hyperlinks.

Definition 4.10. Sei $G = (V, E)$ ein gerichteter Graph und sei $v \in V$ ein Knoten von G.

Ausgangsgrad
Aus-Grad$_G(v)$

- Der **Ausgangsgrad von v in G** (engl.: out-degree), kurz: Aus-Grad$_G(v)$, ist die Anzahl der Kanten, die v als Ausgangsknoten haben. D.h.:

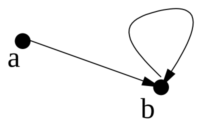
$$\text{Aus-Grad}_G(v) \ := \ |\{e \in E : \text{es ex. } v' \in V \text{ s.d. } e = (v, v')\}|$$

Eingangsgrad
Ein-Grad$_G(v)$

- Der **Eingangsgrad von v in G** (engl.: in-degree), kurz: Ein-Grad$_G(v)$, ist die Anzahl der Kanten, die v als Eingangsknoten haben. D.h.:

$$\text{Ein-Grad}_G(v) \ := \ |\{e \in E : \text{es ex. } v' \in V \text{ s.d. } e = (v', v)\}|.$$

Beispiel.

$G =$ 

Ein-Grad$_G(a) = 0$
Ein-Grad$_G(b) = 2$
Aus-Grad$_G(a) = 1$
Aus-Grad$_G(a) = 1$

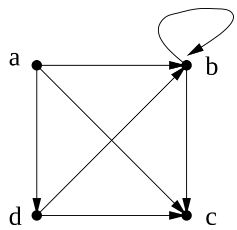
Bemerkung 4.11 (Darstellung von Graphen)**.** Es gibt mehrere Arten Graphen darzustellen, zum Beispiel

- **abstrakt**, durch Angabe der Knotenmenge V und der Kantenmenge E.

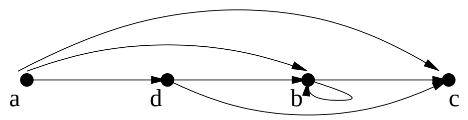
  Beispiel: $G_1 = (V_1, E_1)$ mit

$$V_1 = \{a, b, c, d\} \quad \text{und} \quad E_1 = \{(a, b), (a, c), (a, d), (b, b), (b, c), (d, b), (d, c)\}.$$

- **graphisch** (bzw. **anschaulich**). Der Beispiel-Graph G_1 wird graphisch dargestellt durch:



  oder, äquivalent dazu, durch

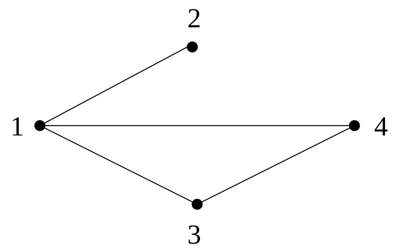


Adjazenzliste

- durch Angabe einer **Adjazenzliste**, die zu jedem Knoten i eine Liste aller Knoten angibt, zu denen eine von i ausgehende Kante führt. Der Beispiel-Graph G_1 wird durch folgende Adjazenzliste repräsentiert:

Knoten	Nachfolger
a	(b, c, d)
b	(b, c)
c	()
d	(b, c)

Auf die gleiche Art können auch **ungerichtete** Graphen durch eine Adjazenzliste repräsentiert werden. Beispielweise der Graph $G_2 :=$



durch die Adjazenzliste

Knoten	Nachbarn
1	(2, 3, 4)
2	(1)
3	(1, 4)
4	(1, 3)

- durch Angabe einer **Adjazenzmatrix**, d.h. eine Tabelle, deren Zeilen und Spalten mit Knoten beschriftet sind, und die in der mit Knoten i beschrifteten Zeile und der mit Knoten j beschrifteten Spalte den Eintrag 1 hat, falls es eine Kante von Knoten i nach Knoten j gibt – und den Eintrag 0, falls es keine Kante von i nach j gibt. Adjazenzmatrix

  Adjazenzmatrix von G_1:

	a	b	c	d
a	0	1	1	1
b	0	1	1	0
c	0	0	0	0
d	0	1	1	0

  Adjazenzmatrix von G_2:

	1	2	3	4
1	0	1	1	1
2	1	0	0	0
3	1	0	0	1
4	1	0	1	0

4.1.2 Wege in Graphen

Definition 4.12. Sei $G = (V, E)$ ein (gerichteter oder ungerichteter) Graph.

(a) Ein **Weg** in G ist ein Tupel Weg

$$(v_0, \dots, v_l) \in V^{l+1},$$

für ein $l \in \mathbb{N}$, so dass f.a. i mit $0 \leq i \leq l$ gilt:

- falls G ein gerichteter Graph ist, so ist $(v_i, v_{i+1}) \in E$
- falls G ein ungerichteter Graph ist, so ist $\{v_i, v_{i+1}\} \in E$.

Weglänge

Das Tupel $(v_0, \dots, v_l)$ wird dann "ein Weg von v_0 nach v_l" genannt; l ist die **Länge des Weges** (d.h.: die **Länge** des Weges gibt gerade an, wie viele **Kanten** auf dem Weg durchlaufen werden).

Beachte: Gemäß dieser Definition ist für jedes $v \in V$ das Tupel (v) ein Weg der Länge 0 von v nach v.

einfacher Weg

(b) Ein Weg heißt **einfach**, wenn kein Knoten mehr als einmal in dem Weg vorkommt.

Kreis

(c) Ein Weg $(v_0, \dots, v_l)$ heißt **Kreis**, wenn $l \geq 1$ und $v_l = v_0$ ist.

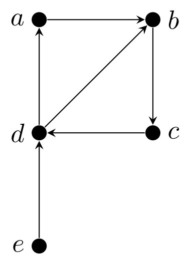
einfacher Kreis

(d) Ein Kreis $(v_0, \dots, v_l)$ heißt **einfach**, wenn keine Kante mehrfach durchlaufen wird und – abgesehen vom Start- und Endknoten – kein Knoten mehrfach besucht wird. D.h.:

- In einem **gerichteten** Graphen G sind **einfache** Kreise genau die Wege der Form $(v_0, \dots, v_l)$, für die gilt: $l \geq 1$ und $v_l = v_0$ und $|\{v_0, \dots, v_{l-1}\}| = l$.
- In einem **ungerichteten** Graphen G sind **einfache** Kreise genau die Wege der Form $(v_0, \dots, v_l)$, für die gilt: $l \geq 3$ und $v_l = v_0$ und $|\{v_0, \dots, v_{l-1}\}| = l$.

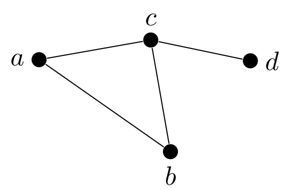
Beispiel 4.13.

(a) Wir betrachten den Graph



- (e, d, b, c, d) ist ein Weg der Länge 4, aber kein einfacher Weg.
- (d, b, c, d) ist ein einfacher Kreis.
- (e, d, a, b) ist ein einfacher Weg.
- (b, d, a) ist kein Weg.
- (a, b, c, d, b, c, d, a) ist ein Kreis, aber kein einfacher Kreis.

(b) Wir betrachten den Graph



- (a, b, c, a) ist ein einfacher Kreis.
- (c, d, c) ist ein Kreis, aber kein einfacher Kreis.

Definition 4.14.

azyklisch

(a) Ein Graph heißt **azyklisch**, falls er keinen einfachen Kreis enthält.

DAG

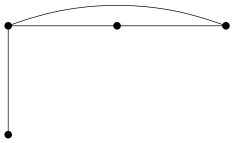
(b) Gerichtete azyklische Graphen werden im Englischen "directed acyclic graph", kurz: DAG, genannt.

Definition 4.15 (zusammenhängend, stark zusammenhängend)**.**

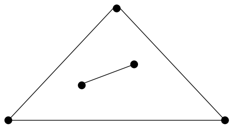
zusammenhängend

(a) Ein ungerichteter Graph $G = (V, E)$ heißt **zusammenhängend**, wenn für alle Knoten $v, w \in V$ gilt: Es gibt in G einen Weg von v nach w.

Beispiel.



ist zusammenhängend.

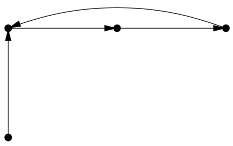


ist nicht zusammenhängend.

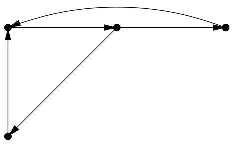
(b) Ein gerichteter Graph $G = (V, E)$ heißt **stark zusammenhängend**, wenn für alle Knoten $v, w \in V$ gilt: Es gibt in G einen Weg von v nach w.

stark zusammenhängend

Beispiel.



ist **nicht** stark zusammenhängend (da es z.B. keinen Weg vom Knoten links oben zum Knoten links unten gibt).



ist stark zusammenhängend.

Definition 4.16 (Hamilton-Kreis und Hamilton-Weg)**.** Sei $G = (V, E)$ ein (gerichteter oder ein ungerichteter) Graph.

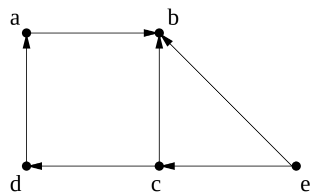
(a) Ein Weg $W = (v_0, \dots, v_l)$ heißt **Hamilton-Weg**, wenn jeder **Knoten** aus V genau einmal in W vorkommt.

Hamilton-Weg

(b) Ein Weg $W = (v_0, \dots, v_l)$ heißt **Hamilton-Kreis**, wenn $l \geq 1$ und $v_l = v_0$ und $(v_0, \dots, v_{l-1})$ ein Hamilton-Weg ist.

Hamilton-Kreis

Beispiel. Der Graph G



hat einen Hamilton-Weg, nämlich (e, c, d, a, b), aber keinen Hamilton-Kreis (da Aus-Grad$_G(b) =$ 0 ist).

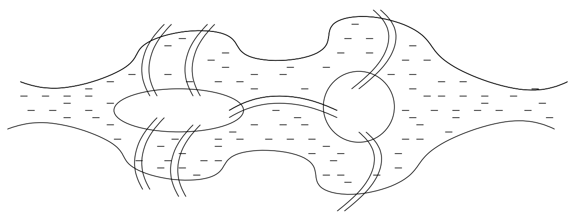
Ein Anwendungsbeispiel: Beim Problem des **Handlungsreisenden** (engl.: Travelling Salesman's Problem) geht es darum, eine Rundreise durch n Städte so durchzuführen, dass jede Stadt genau 1 mal besucht wird. Es geht also darum, einen Hamilton-Kreis zu finden. Das Problem, zu einem gegebenen Graphen zu entscheiden, ob er einen Hamilton-Kreis besitzt, ist algorithmisch ein

Handlungsreisendenproblem

schwieriges Problem: man kann zeigen, dass es (genau wie das aussagenlogische Erfüllbarkeitsproblem) NP-vollständig ist.

Im Gegensatz zu Hamilton-Wegen (bei denen es darum geht, einen Weg zu finden, der jeden **Knoten** des Graphen genau einmal besucht), geht es bei den im Folgenden betrachteten **Euler-Wegen** darum, einen Weg zu finden, der jede **Kante** des Graphen genau einmal besucht.

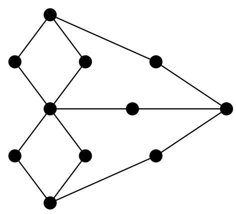
Beispiel 4.17 (Königsberger Brückenproblem)**.** In der Stadt Königsberg gab es im 18 Jahrhundert 7 Brücken über den Fluss Pregel, die die Ufer und 2 Inseln miteinander verbanden. Skizze:



Frage: Gibt es einen Spaziergang, der jede der 7 Brücken genau einmal überquert und zum Ausgangspunkt zurückkehrt?

Die obige Skizze lässt sich folgendermaßen durch einen ungerichteten Graphen modellieren: für jedes Ufer, jede Insel und jede Brücke gibt es einen Knoten; Kanten zeigen direkte Verbindungen an. Die Skizze wird also durch folgenden Graphen repräsentiert:

$G_{\text{Königsberg}} :=$



Die Frage nach dem "Spaziergang" entspricht dann gerade der Frage: Gibt es in $G_{\text{Königsberg}}$ einen Euler-Kreis?

Definition 4.18 (Euler-Kreise und Euler-Wege)**.** Sei $G = (V, E)$ ein ungerichteter Graph.

Euler-Weg

(a) Ein Weg $W = (v_0, \ldots, v_l)$ heißt **Euler-Weg**, wenn W jede Kante aus E genau einmal enthält, d.h. wenn es für jedes $e \in E$ genau ein $i \in \{0, \ldots, l-1\}$ gibt, so dass $e = \{v_i, v_{i+1}\}$.

Euler-Kreis

(b) Ein Weg $w = (v_0, \ldots, v_l)$ heißt **Euler-Kreis**, wenn W ein Euler-Weg ist und $v_0 = v_l$ ist.

Satz 4.19 (Existenz von Euler-Kreisen und Euler-Wegen)**.** *Sei $G = (V, E)$ ein ungerichteter, zusammenhängender Graph, dessen Knotenmenge endlich ist. Dann gilt:*

(a) G besitzt einen Euler-Kreis $\iff$ jeder Knoten von G hat einen geraden Grad (d.h. ist mit einer geraden Anzahl von Kanten inzident).

(b) G besitzt einen Euler-Weg, der kein Euler-Kreis ist $\iff$ es gibt in G genau zwei Knoten mit ungeradem Grad.

Beweis:

(a) "$\Longrightarrow$" Sei $K = (v_0, \dots, v_l)$ ein Euler-Kreis. Insbesondere: $v_0 = v_l$.

Schritt 1: Jeder Knoten $v \in \{v_0, \dots, v_{l-1}\}$ hat geraden Grad, denn: Sei $v \in \{v_0, \dots, v_{l-1}\}$ beliebig. Zu jedem $i \in \{0, \dots, l-1\}$ mit $v = v_i$ gibt es im Euler-Kreis K zwei verschiedene Kanten, nämlich $\{v_{i-1}, v_i\}$ und $\{v_i, v_{i+1}\}$ (falls $i \neq 0$) bzw. falls $i = 0$, $\{v_0, v_1\}$ und $\{v_{l-1}, v_0\}$ (beachte: $v_0 = v_l$).

Da der Euler-Kreis K jede Kante von G genau einmal enthält, gilt somit folgendes: Ist $k = |\{i \in \{0, \dots, l-1\} : v = v_i\}|$ (d.h. k gibt an, wie oft v im Tupel $(v_0, \dots, v_{l-1})$ vorkommt), so ist $\text{Grad}_G(v) = 2 \cdot k$. Daher hat jeder Knoten $v \in \{v_0, \dots, v_{l-1}\}$ geraden Grad.

Schritt 2: $\{v_0, \dots, v_{l-1}\} = V$, denn laut Voraussetzung ist G zusammenhängend. Für beliebige Knoten $v, w \in V$ gilt daher: es gibt in G einen Weg von v nach w. Da K ein Euler-Kreis ist, enthält K sämtliche Kanten, die auf dem Weg von v nach w vorkommen. Insbesondere gilt also f.a. $v, w \in V$, dass $v, w \in \{v_0, \dots, v_{l-1}\}$.

Schritt 3: Aus Schritt 1 und Schritt 2 folgt direkt, dass jeder Knoten von G geraden Grad hat.

"$\Longleftarrow$" Sei G ein zusammenhängender ungerichteter Graph, in dem jeder Knoten geraden Grad hat. Es sei

$$W = (v_0, \dots, v_l)$$

ein Weg **maximaler Länge** in G, der **keine Kante(n) mehrfach** enthält. Da wir W nicht mehr verlängern können, liegen alle mit v_l inzidenten Kanten auf W. Da laut unserer Voraussetzung die Anzahl dieser Kanten gerade ist, folgt $v_l = v_0$.

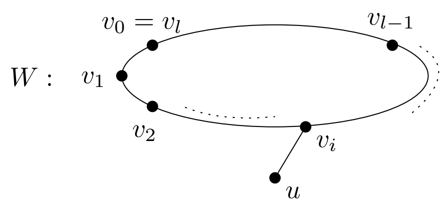
Zu zeigen: W ist ein Euler-Kreis.

Angenommen, W ist **kein** Euler-Kreis. Dann gibt es in G eine Kante e, die nicht auf W liegt, die aber mit mindestens einem Knoten auf W inzident ist (um dies zu sehen, nutzen wir, dass G zusammenhängend ist). Sei v_i der zu e inzidente Knoten aus W und sei $u \in V$ der andere zu e inzidente Knoten, d.h. $e = \{u, v_i\}$. Dann ist der Weg

$$W' := (u, v_i, v_{i+1}, \dots, v_{l-1}, v_0, v_1, \dots, v_i)$$

ein Weg der Länge $l + 1$, der keine Kante(n) mehrfach enthält.

Skizze:



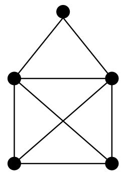
Dies widerspricht aber der Tatsache, dass W ein Weg **maximaler** Länge ist.

(b) Folgt leicht aus (a). Details: Übung. □

Beispiel 4.20. Mit Hilfe von Satz 4.19 können wir das Königsberger Brückenproblem aus Beispiel 4.17 leicht lösen: Es gibt **keinen** Spaziergang, der jede der 7 Brücken genau einmal überquert und zum Ausgangspunkt zurückkehrt, denn: ein solcher Spaziergang würde gerade einem Euler-Kreis im Graphen $G_{\text{Königsberg}}$ entsprechen. Dieser Graph besitzt aber 4 Knoten von ungeradem Grad und kann daher laut Satz 4.19(a) keinen Euler-Kreis besitzen.

Beispiel 4.21.

Frage: Kann man die Figur



in einem Zug nachzeichnen? D.h: Besitzt dieser Graph einen Euler-Weg?

Unter Verwendung von Satz 4.19 kann man die Frage leicht beantworten, indem man nachzählt, wie viele Knoten von ungeradem Grad es gibt. Im obigen Graphen gibt es genau 2 Knoten von ungeradem Grad. Gemäß Satz 4.19 besitzt G also einen Euler-Weg, der kein Euler-Kreis ist.

4.1.3 Ähnlichkeit zweier Graphen

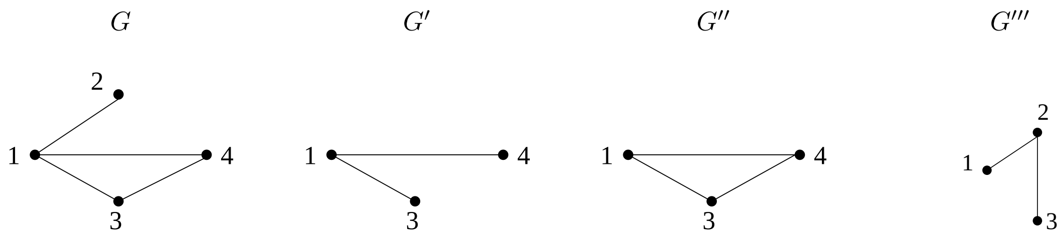
Die folgende Definition formalisiert, wann ein Graph G' in einem Graphen G "enthalten" ist.

Teilgraph
induzierter Teilgraph

Definition 4.22 (Teilgraph)**.** Seien $G = (V, E)$ und $G' = (V', E')$ zwei (gerichtete oder ungerichtete) Graphen. G' heißt **Teilgraph von** G, falls $V' \subseteq V$ und $E' \subseteq E$. G' heißt **induzierter Teilgraph von** G, falls

$$V' \subseteq V \quad \text{und} \quad E' = \{e \in E : \text{die mit } e \text{ inzidenten Knoten liegen in } V'\}.$$

Beispiel 4.23. Wir betrachten die folgenden Graphen:



Dann ist

- G' ein Teilgraph von G, aber kein induzierter Teilgraph von G.
- G'' ein induzierter Teilgraph von G.
- G''' kein Teilgraph von G.

Bemerkung 4.24. Zwei Graphen $G = (V, E)$ und $G' = (V', E')$ sind **gleich** (kurz: $G = G'$), falls sie dieselbe Knotenmenge und dieselbe Kantenmenge besitzen. D.h.:

$$G = G' \;:\Longleftrightarrow\; V = V' \text{ und } E = E'.$$

Zwei Graphen G und G' sind *"prinzipiell gleich"* (Fachbegriff: **isomorph**, kurz: $G \cong G'$), falls G' aus G durch Umbenennung der Knoten entsteht. Dies wird durch die folgende Definition präzisiert:

Definition 4.25 (Isomorphie von Graphen)**.** Seien $G = (V, E)$ und $G' = (V', E')$ zwei (gerichtete oder ungerichtete) Graphen. G und G' heißen **isomorph** (kurz: $G \cong G'$, in Worten: G ist isomorph zu G'), falls es eine bijektive Abbildung $f : V \to V'$ gibt, so dass für alle Knoten $i \in V$ und $j \in V$ gilt:

isomorph
$G \cong G'$

- falls G und G' gerichtet sind:

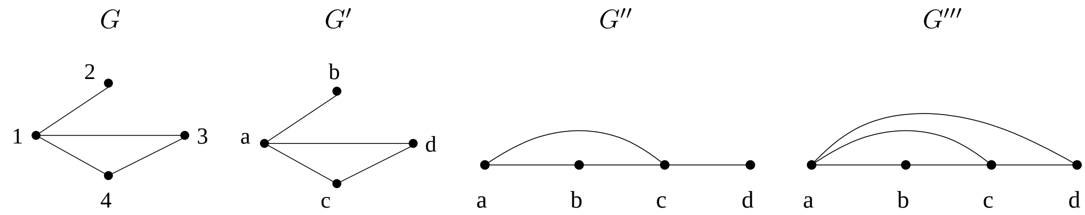
$$(i, j) \in E \iff \big(f(i), f(j)\big) \in E'$$

- falls G und G' ungerichtet sind:

$$\{i, j\} \in E \iff \{f(i), f(j)\} \in E'$$

Eine solche Abbildung f wird **Isomorphismus von G nach G′** genannt. Isomorphismus

Beispiel 4.26. Es seien:



Dann gilt:

- $G \cong G'$ via $f\colon \{1, 2, 3, 4\} \to \{a, b, c, d\}$ mit $f(1) = a,\ f(2) = b,\ f(3) = d,\ f(4) = c$.
- $G \cong G''$ via $f\colon \{1, 2, 3, 4\} \to \{a, b, c, d\}$ mit $f(1) = c,\ f(2) = d,\ f(3) = a,\ f(4) = b$.
- G'' ist nicht isomorph zu G''', kurz: $G'' \not\cong G'''$, da G''' **mehr** Kanten als G'' hat.

4.1.4 Markierte Graphen

Bemerkung 4.27. Viele Modellierungsaufgaben erfordern, dass den Knoten oder den Kanten eines Graphen weitere Informationen zugeordnet werden. Dies wird durch so genannte **Markierungsfunktionen** für Knoten oder Kanten formalisiert.

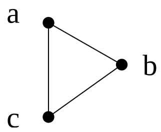
Eine **Knotenmarkierung** eines (gerichteten oder ungerichteten) Graphen $G = (V, E)$ ist eine Abbildung $\mathrm{MV}\colon V \to \mathrm{WV}$, wobei WV ein geeigneter Wertebereich ist. In dem Graph aus Beispiel 4.1(a) könnte man beispielweise eine Knotenmarkierung $\text{Einwohnerzahl}\colon V \to \mathbb{N}$ einführen, die jedem Knoten die Einwohnerzahl der zugehörigen Stadt zuordnet. Eine **Kantenmarkierung** von G ist eine Abbildung $\mathrm{ME}\colon E \to \mathrm{WE}$, wobei WE ein geeigneter Wertebereich ist. In dem Graph aus Beispiel 4.1(a) könnte man beispielweise eine Kantenmarkierung $\text{Entfernung}\colon E \to \mathbb{N}$ einführen, die jeder Kante die Länge (in km) des von der Kante repräsentierten Autobahnteilstücks zuordnet.

Knotenmarkierung
Kantenmarkierung

Kantenmarkierungen kann man auch dazu verwenden, um auszudrücken, dass es zwischen zwei Knoten mehr als eine Kante gibt: die Markierungsfunktion gibt dann an, für wie viele Verbindungen die eine Kante des Graphen steht.

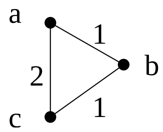
Definition 4.28 (Multigraph)**.** Ein **Multigraph** (G, m) besteht aus einem (gerichteten oder ungerichteten) Graphen $G = (V, E)$ und einer Kantenmarkierung $m\colon E \to \mathbb{N}$. Multigraph

Beispiel. Multigraph (G, m) mit

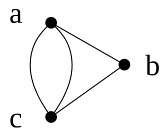


und $m(\{a,b\}) = 1$, $m(\{b,c\}) = 1$, $m(\{c,a\}) = 2$

graphische Darstellung von (G, m):



bzw.



4.1.5 Zuordnungsprobleme

Beispiel 4.29. Typische Aufgabenstellungen:

(a) In einem Tennisverein sollen die Vereinsmitglieder für ein Turnier zu Doppelpaarungen zusammengestellt werden. Dabei möchte man jeweils nur befreundete Personen als "Doppel" zusammen spielen lassen.

(b) Eine Gruppe unterschiedlich ausgebildeter Piloten soll so auf Flugzeuge verteilt werden, dass jeder das ihm zugeteilte Flugzeug fliegen kann.

Beide Situationen lassen sich gut durch ungerichtete Graphen modellieren:

Zu (a): Modelliere die Situation durch den Graphen $G_T := (V_T, E_T)$ mit

$$\begin{aligned} V_T &:= \{x : x \text{ ist ein Vereinsmitglied}\} \\ E_T &:= \{\{x, y\} : x \text{ und } y \text{ sind befreundete Vereinsmitglieder}\}. \end{aligned}$$

Ziel: Finde eine größtmögliche Anzahl von Doppelpaarungen, d.h. finde eine möglichst große Menge $E' \subseteq E_T$, so dass kein Vereinsmitglied Endpunkt von mehr als einer Kante aus E' ist.

Zu (b): Modelliere die Situation durch den Graphen $G_F := (V_F, E_F)$ mit

$$\begin{aligned} V_F &:= \{x : x \text{ ist ein Pilot}\} \,\dot\cup\, \{y : y \text{ ist ein Flugzeug}\}, \\ E_F &:= \{\{x, y\} : \text{Pilot } x \text{ kann Flugzeug } y \text{ fliegen}\}. \end{aligned}$$

Ziel: Stelle einen Flugplan auf, so dass jeder Pilot das ihm zugeteilte Flugzeug fliegen kann, d.h. finde eine möglichst große Menge $E' \subseteq E_F$, so dass kein Element aus V_F Endpunkt von mehr als einer Kante in E' ist.

Die gesuchten Kantenmengen E' aus (a) und (b) werden auch **Matching** genannt:

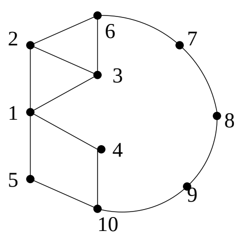
Matching

Definition 4.30. Sei $G = (V, E)$ ein ungerichteter Graph. Eine Kantenmenge $E' \subseteq E$ heißt **Matching** (bzw. **Menge unabhängiger Kanten**), falls gilt: kein Knoten aus V ist Endpunkt von mehr als einer Kante aus E'.

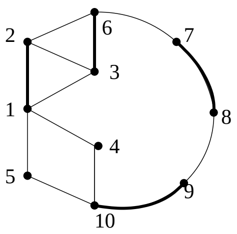
Ziel in Beispiel 4.29 (a) und (b) ist es, ein Matching maximaler Größe (d.h. eins, das so viele Kanten wie möglich enthält) zu finden.

Beispiel 4.31. In einem Tennisverein mit 10 Mitgliedern und "Freundschaftsgraph"

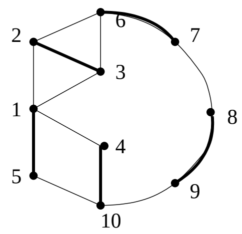
G_T =



sind z.B. die folgenden beiden Kantenmengen Matchings:



und



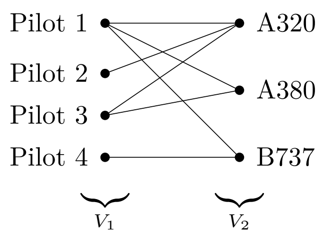
$$E' = \big\{\{1,2\},\{3,6\},\{7,8\},\{9,10\}\big\} \qquad E'' = \big\{\{1,5\},\{4,10\},\{8,9\},\{6,7\},\{2,3\}\big\}.$$

In Beispiel 4.29(b) sollten Piloten auf Flugzeuge verteilt werden. Die Knotenmenge des zugehörigen Graphen G_F bestand aus zwei verschiedenen Arten von Objekten (nämlich einerseits Piloten und andererseits Flugzeuge), und Kanten konnten jeweils nur zwischen Objekten unterschiedlicher Art verlaufen (also zwischen Piloten und Flugzeugen, nicht aber zwischen Piloten und Piloten bzw. Flugzeugen und Flugzeugen). Solche Graphen werden *bipartite Graphen* genannt:

Definition 4.32. Ein ungerichteter Graph $G = (V, E)$ heißt **bipartit**, wenn seine Knotenmenge bipartit
V so in zwei disjunkte Teilmengen V_1 und V_2 zerlegt werden kann, dass jede Kante aus E einen Endknoten in V_1 und einen Endknoten in V_2 hat.

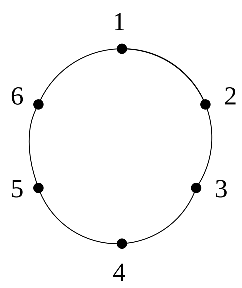
Beispiel 4.33.

(a) Der Graph

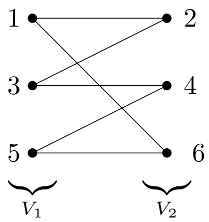


ist bipartit.

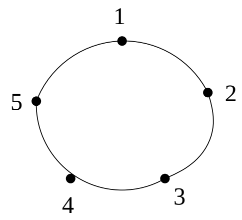
(b) Der Graph



ist bipartit, denn wähle $V_1 = \{1, 3, 5\}$ und $V_2 = \{2, 4, 6\}$; andere graphische Darstellung des Graphen:



(c) Der Graph



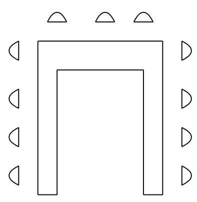
ist **nicht** bipartit (denn angenommen doch, seien V_1 und V_2 die beiden disjunkten Teilmengen der Knotenmenge, so dass jede Kante des Graphen einen Endknoten in V_1 und einen Endknoten in V_2 hat; o.B.d.A. nehmen wir an, dass $1 \in V_1$ ist. Dann muss aber gelten: $2 \in V_2$, $3 \in V_1$, $4 \in V_2$, $5 \in V_1$, also $V_1 = \{1, 3, 5\}$ und $V_2 = \{2, 4\}$. **Aber:** es gibt eine Kante zwischen 1 und 5, und beide Knoten gehören zu V_1. $\lightning$)

Allgemein gilt: Ist $n \in \mathbb{N}_{>0}$ und ist G ein Kreis auf n Knoten (wie in (b) für $n = 6$ und in (c) für $n = 5$), so gilt:

$$G \text{ ist bipartit} \iff n \text{ ist gerade.}$$

Ein weiteres typisches Beispiel für ein Zuordnungsproblem:

Beispiel 4.34. Die Gäste einer Familienfeier sollen so an einer hufeisenförmigen Tafel



platziert werden, dass niemand neben jemanden sitzt, den er nicht leiden kann.

Lösungsansatz:

Schritt 1: Stelle den **Konfliktgraphen** $G = (V, E)$ auf mit

$$\begin{aligned} V &= \{x : \text{Person } x \text{ soll zur Feier eingeladen werden}\} \quad \text{und} \\ E &= \big\{\{x, y\} : \text{Person } x \text{ kann Person } y \text{ nicht leiden oder} \\ &\qquad\qquad \text{Person } y \text{ kann Person } x \text{ nicht leiden}\big\}, \end{aligned}$$

d.h. Kanten im Konfliktgraphen zeigen auf, wer im Konflikt mit wem steht.

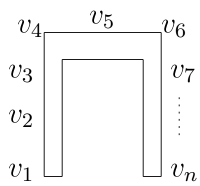
Schritt 2: Bilde das Komplement $\tilde{G} = (\tilde{V}, \tilde{E})$ des Konfliktgraphen, d.h. betrachte

$$\begin{aligned} \tilde{V} &:= V, \\ \tilde{E} &:= \big\{\{x,y\} : x, y \in V,\ x \neq y,\ \{x,y\} \notin E\big\}, \end{aligned}$$

d.h. Kanten in $\tilde{G}$ zeigen an, wer prinzipiell neben wem platziert werden könnte.

Schritt 3: Suche einen Hamilton-Weg in $\tilde{G}$.

Wenn $(v_1, \ldots, v_n)$ (mit $n = |\tilde{V}|$) ein Hamilton-Weg in $\tilde{G}$ ist, dann kann man die Sitzordnung folgendermaßen festlegen:



Falls es in $\tilde{G}$ keinen Hamilton-Weg gibt, so weiß man, dass es **keine Möglichkeit gibt**, die geladenen Gäste so an einer hufeisenförmigen Tafel zu platzieren, dass niemand neben jemandem sitzt, den er nicht leiden kann.

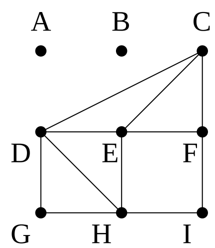
Ein möglicher Ausweg: Verteile die Gäste auf **mehrere** Tische:

Beispiel 4.35. Die Gäste einer Familienfeier sollen so an mehreren (möglichst wenigen) Tischen platziert werden, dass Personen, die sich nicht ausstehen können, an verschiedenen Tischen sitzen. Diese Aufgabe kann folgendermaßen modelliert werden. Die verfügbaren Tische werden durchnummeriert mit den Zahlen $1, 2, 3, \ldots$. Die geladenen Gäste und die herrschenden Konflikte zwischen Gästen werden durch den in Beispiel 4.34 betrachteten Konfliktgraphen $G = (V, E)$ repräsentiert. Die Zuordnung, wer an welchem Tisch sitzen soll, wird durch eine Knotenmarkierung $m\colon V \to \mathbb{N}_{>0}$ repräsentiert ($m(x) = i$ bedeutet dann, dass Person x am Tisch i sitzen soll).

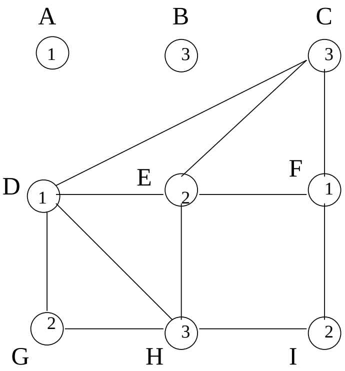
Ziel: Finde eine **konfliktfreie Knotenmarkierung** $m\colon V \to \mathbb{N}_{>0}$, d.h. eine Knotenmarkierung, so dass für jede Kante $\{x,y\} \in E$ gilt: $m(x) \neq m(y)$. Dabei soll $|\text{Bild}(m)|$ möglichst klein sein (dies entspricht dem Ziel, die Gäste auf möglichst **wenige** Tische zu verteilen).

Definition 4.36. Sei $G = (V, E)$ ein ungerichteter Graph. Eine Funktion $m\colon V \to \mathbb{N}$ heißt **konfliktfreie Knotenmarkierung** (oder: konfliktfreie **Färbung**), wenn für jede Kante $\{x,y\} \in E$ gilt: $m(x) \neq m(y)$. konfliktfreie Knotenmarkierung

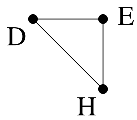
Beispiel 4.37. Familienfeier mit Gästen A, B, C, D, E, F, G, H, I und folgendem Konfliktgraphen



Eine konfliktfreie Knotenmarkierung (d.h. Platzierung an verschiedene Tische) $m\colon V \to \mathbb{N}$:



Hier ist für jeden Knoten $v \in V$ der Wert $m(v)$ in den Kreis geschrieben, der den Knoten v repräsentiert. Für die hier gegebene Markierung m gilt $|\mathrm{Bild}(m)| = 3$, die Gäste werden also auf 3 Tische verteilt. Dies ist "optimal", da der Konfliktgraph ein Dreieck, z.B.



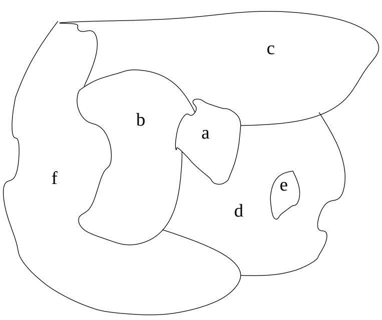
als Teilgraph enthält – deshalb muss für jede konfliktfreie Knotenmarkierung m' gelten: $|\mathrm{Bild}(m')| \geq 3$.

4-Farben-Problem

Bemerkung 4.38. Die wohl berühmteste Aufgabe dieser Art von Markierungs- oder Färbungsaufgaben ist das so genannte **4-Farben-Problem**. Dabei handelt es sich um die Hypothese, dass vier verschiedene Farben ausreichen, um eine Landkarte so einzufärben, dass zwei Staaten, die ein Stück gemeinsamer Grenze haben, durch unterschiedliche Farben dargestellt werden. Erst 1976 wurde diese Hypothese bewiesen, und zwar durch eine Fallunterscheidung mit mehr als 1000 Fällen, die mit Hilfe eines Computerprogramms gelöst wurde.

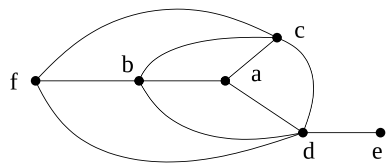
Beispiel.

Eine (kleine) Landkarte:



zugehöriger Konfliktgraph:

Knoten $\widehat{=}$ Staaten
Kanten $\widehat{=}$ Staaten mit gemeinsamer Grenze

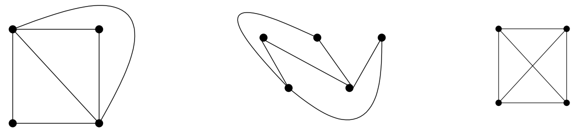


Da bei den vier Knoten a, b, c, d paarweise jeder zu jedem benachbart ist, muss eine konfliktfreie Färbung diesen 4 Knoten 4 verschiedene Farben zuordnen – für a, b, c, d etwa rot, gelb, grün, blau. Da f außerdem mit b, c, d benachbart ist, muss f dann wieder rot gefärbt sein; e kann jede Farbe außer blau erhalten.

Die zu Landkarten gehörenden Konfliktgraphen haben eine besondere Eigenschaft: Sie sind planar.

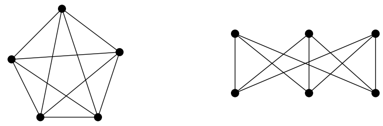
Definition 4.39. Ein Graph G heißt **planar**, wenn er so in die Ebene gezeichnet werden kann, dass seine Kanten sich nicht kreuzen. planar

Beispiel. Planare Graphen sind:



(Der dritte Graph kann wie der erste Graph kreuzungsfrei in die Ebene gezeichnet werden.)

Nicht-planare Graphen sind:



Beispiel 4.40. Weitere Beispiele von Anwendungen, die durch Finden konfliktfreier Färbungen im Konfliktgraphen gelöst werden können:

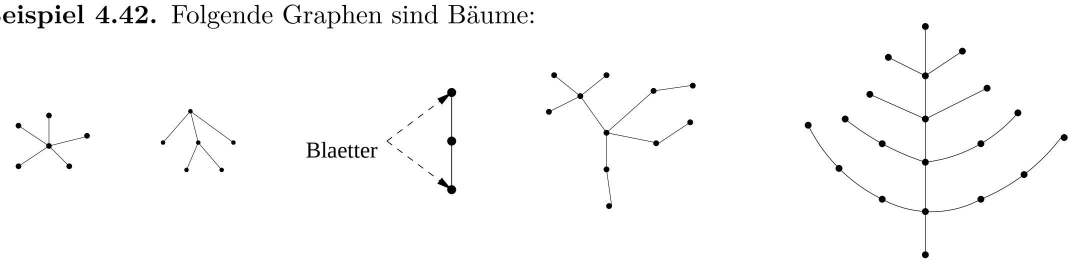
Knoten	Kante zwischen x und y	Farbe bzw. Markierung
Staat auf Karte	haben gemeinsame Grenze	Farbe
Gast auf Familienfeier	können sich nicht leiden	Tischnummer
Vorlesung	haben gemeinsame Teilnehmer	Termin
Variable im Programm	ihre Werte werden gleichzeitig benötigt	Registerspeicher
Prozess	benötigt dieselben Ressourcen	Ausführungstermin

4.2 Bäume

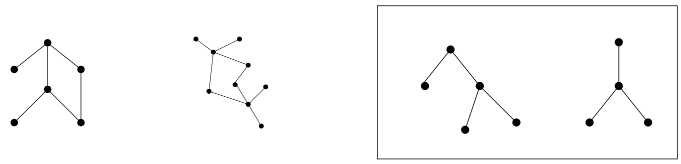
4.2.1 Ungerichtete Bäume

Definition 4.41. Ein **ungerichteter Baum** ist ein ungerichteter, zusammenhängender Graph $G = (V, E)$, der keinen einfachen Kreis enthält. Diejenigen Knoten in V, die den Grad 1 haben, heißen **Blätter** des Baums. ungerichteter Baum Blätter

Beispiel 4.42. Folgende Graphen sind Bäume:



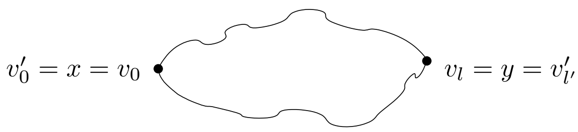
Folgende Graphen sind keine Bäume:



Beobachtung 4.43. Sei $B = (V, E)$ ein ungerichteter Baum. Da B zusammenhängend ist und keinen einfachen Kreis enthält, gilt f.a. Knoten $x, y \in V$:

es gibt in B genau einen einfachen Weg von x nach y

(denn: Da B zusammenhängend ist, gibt es mindestens einen einfachen Weg von x nach y. Angenommen, $(v_0, \ldots, v_l)$ und $(v'_0, \ldots, v'_{l'})$ sind zwei verschiedene einfache Wege von x nach y. Insbesondere gilt dann $v_0 = x = v'_0$ und $v_l = y = v'_{l'}$.
Skizze:

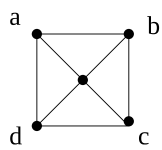


Dann ist aber $(v_0, \ldots, v_l, v'_{l'-1}, \ldots, v'_0)$ ein Kreis. Dieser Kreis enthält einen einfachen Kreis. Dann kann B aber kein Baum sein. $\lightning$)

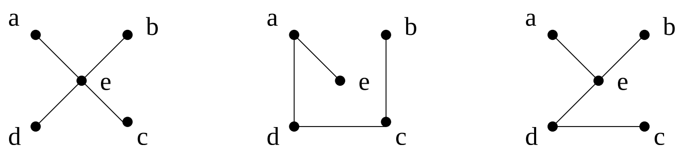
Spannbaum

Definition 4.44. Sei $G = (V, E)$ ein ungerichteter Graph. Ein Graph $G' = (V', E')$ heißt **Spannbaum von** G, falls G' ein ungerichteter Baum mit $V' = V$ und $E' \subseteq E$ ist.

Beispiel 4.45. Der Graph



hat z.B. folgende Spannbäume:



Satz 4.46. *Sei $G = (V, E)$ ein ungerichteter Graph, dessen Knotenmenge endlich ist. Dann gilt:*

$$\textit{Es gibt (mindestens) einen Spannbaum von } G \iff G \textit{ ist zusammenhängend.}$$

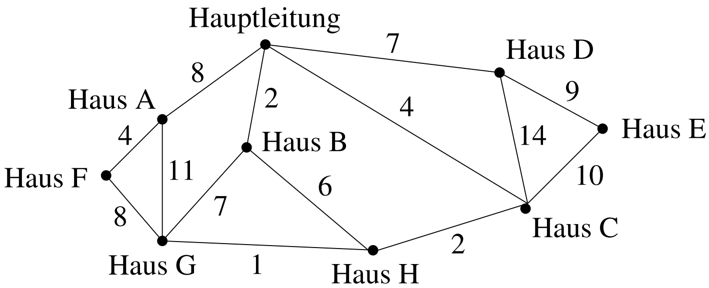
Beweis: "$\Longrightarrow$" klar, "$\Longleftarrow$" Übung. $\square$

Geht man von einem zusammenhängenden Graphen zu einem seiner Spannbäume über, so verkleinert man die Kantenmenge von $|E|$ auf – gemäß dem folgenden Satz 4.48 – $|V| - 1$ Kanten, ohne dabei den Zusammenhang des Graphen aufzugeben. Mit dem Begriff des Spannbaums wird also ein "bezüglich der Kantenzahl kostengünstigerer Zusammenhang" modelliert.

Manche konkreten Probleme lassen sich durch Graphen modellieren, deren Kanten mit bestimmten Werten markiert sind, so dass zur Lösung des Problems ein Spannbaum gesucht wird, bei dem die Summe seiner Kantenmarkierungen so klein wie möglich ist (engl: "**minimum spanning tree** problem").

Beispiel 4.47 (Kabelfernsehen)**.** Eine Firma will Leitungen zum Empfang von Kabelfernsehen in einem neuen Wohngebiet verlegen. Der folgende Graph skizziert das Wohngebiet:

- Knoten entsprechen dabei einzelnen Häusern bzw. der Hauptleitung, die aus einem bereits verkabelten Gebiet heranführt,
- eine Kante zwischen zwei Knoten zeigt an, dass es prinzipiell möglich ist, eine direkte Leitung zwischen den beiden Häusern zu verlegen, und
- der Wert, mit dem die Kante markiert ist, beschreibt, wie teuer (in 1000 €) es ist, diese Leitung unterirdisch zu verlegen.



Ziel ist, Leitungen so zu verlegen, dass

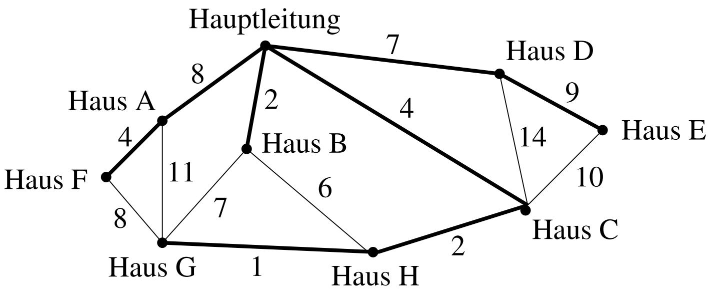
(1) jedes Haus ans Kabelfernsehen angeschlossen ist und

(2) die Kosten für das Verlegen der Leitungen so gering wie möglich sind.

Es wird also ein Spannbaum gesucht, bei dem die Summe seiner Kantenmarkierungen so klein wie möglich ist. Ein solcher Spannbaum wird **minimaler Spannbaum** (engl.: minimum spanning tree) genannt.

minimaler Spannbaum

Die im Folgenden **fett** gezeichneten Kanten geben die Kanten eines minimalen Spannbaums an:



Verlegt die Firma genau diese Leitungen, so hat sie das neue Wohngebiet mit den geringstmöglichen Kosten ans Kabelfernsehen angeschlossen.

Bemerkung. Effiziente Verfahren zum Finden minimaler Spannbäume werden Sie in der Vorlesung "Algorithmentheorie" (GL-1) kennenlernen.

Satz 4.48 (Anzahl der Kanten eines Baums)**.** *Sei $B = (V, E)$ ein* ***ungerichteter Baum****, dessen Knotenmenge endlich und nicht-leer ist. Dann gilt*

$$|E| \;=\; |V| - 1.$$

Beweis: Per Induktion nach $n = |V|$.

INDUKTIONSANFANG: $n = 1$

Der einzige ungerichtete Baum $B = (V, E)$ mit $|V| = 1$ ist der Graph $\bullet$ mit $E = \emptyset$. Für diesen Graphen gilt:

$$|E| \;=\; 0 \;=\; 1 - 1 \;=\; |V| - 1.$$

INDUKTIONSSCHRITT: $n \to n + 1$

Sei $n \in \mathbb{N}$ mit $n \geq 1$ beliebig.

Induktionsannahme: Für jeden ungerichteten Baum $B' = (V', E')$ mit $|V'| \leq n$ gilt:

$$|E'| \;=\; |V'| - 1.$$

Behauptung: Für jeden ungerichteten Baum $B = (V, E)$ mit $|V| = n + 1$ gilt $|E| \;=\; |V| - 1$.

Beweis: Sei $B = (V, E)$ ein ungerichteter Baum mit $|V| = n + 1$. Da B zusammenhängend ist und $|V| \geq 1 + 1 = 2$ Knoten besitzt, muss E mindestens eine Kante enthalten. Sei $\{x, y\}$ eine Kante in E.

Sei G der Graph, der aus B durch Löschen der Kante $\{x, y\}$ entsteht. Da B zusammenhängend ist und keinen einfachen Kreis enthält, muss G aus genau 2 Zusammenhangskomponenten bestehen. Seien $G_1 = (V_1, E_1)$ und $G_2 = (V_2, E_2)$ diese beiden Zusammenhangskomponenten von G. Es gilt:

(1) $V = V_1 \,\dot\cup\, V_2$, $V_1 \neq 0$ und $V_2 \neq 0$.

(2) $E = E_1 \,\dot\cup\, E_2 \,\dot\cup\, \{\{x, y\}\}$.

(3) G_1 ist ein ungerichteter Baum.

(4) G_2 ist ein ungerichteter Baum.

Wegen (1) gilt insbesondere, dass $|V_1|, |V_2| \leq n$. Wegen (3) und (4) gilt daher gemäß Induktionsannahme:

(5) $|E_1| = |V_1| - 1$ und $|E_2| = |V_2| - 1$.

Aus (2) und (1) folgt daher:

$$|E| \;\stackrel{(2)}{=}\; |E_1| + |E_2| + 1 \;\stackrel{(5)}{=}\; |V_1| - 1 + |V_2| - 1 + 1 \;\stackrel{(1)}{=}\; |V| - 1.$$

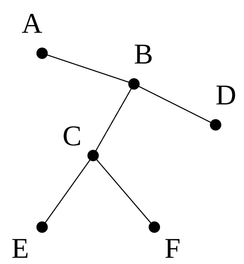
□

4.2.2 Gerichtete Bäume

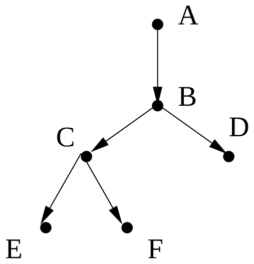
gerichteten Baum

Einen **gerichteten Baum** erhält man, indem man in einem ungerichteten Baum einen Knoten als "Wurzel" auswählt und alle Kanten in die Richtung orientiert, die von der Wurzel weg führt.

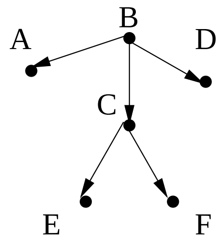
Beispiel 4.49. Ungerichteter Baum:



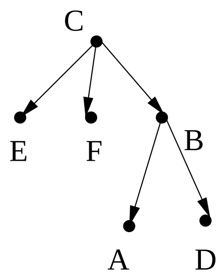
- Zugehöriger gerichteter Baum mit Wurzel A:

$G_A \ :=$ 

- Zugehöriger gerichteter Baum mit Wurzel B:

$G_B \ :=$ 

- Zugehöriger gerichteter Baum mit Wurzel C:

$G_C \ :=$ 

Definition 4.50.

(a) Ein gerichteter Graph $G = (V, E)$ heißt **gerichteter Baum**, falls er folgende Eigenschaften hat: — gerichteter Baum

(1) G besitzt genau einen Knoten $w \in V$ mit $\text{Ein-Grad}_G(w) = 0$. Dieser Knoten wird **Wurzel** genannt. — Wurzel

(2) Für jeden Knoten $v \in V$ gilt: Es gibt in G einen Weg von der Wurzel zum Knoten v.

(3) Für jeden Knoten $v \in V$ gilt: $\text{Ein-Grad}_G(v) \leq 1$.

(b) Sei $B = (V, E)$ ein gerichteter Baum. Die **Höhe** (bzw. **Tiefe**, engl.: height, depth) von B ist die Länge eines längsten Weges in B. — Höhe, Tiefe

Beispiel: In Beispiel 4.49 hat G_A die Höhe 3, G_B die Höhe 2 und G_C die Höhe 2.

(c) Sei $B = (V, E)$ ein gerichteter Baum. Diejenigen Knoten, deren Aus-Grad 0 ist, heißen **Blätter**.

Blätter

Beispiel: In Beispiel 4.49 hat G_A die Blätter D, E, F; G_B die Blätter A, D, E, F und G_C die Blätter A, D, E, F.

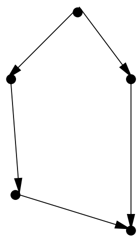
(d) Diejenigen Knoten eines gerichteten Baums, die weder Wurzel noch Blätter sind, heißen **innere Knoten**.

innere Knoten

Beobachtung 4.51.

(a) Jeder gerichtete Baum ist ein gerichteter azyklischer Graph (kurz: DAG, vgl. Definition 4.14). Aber es gibt gerichtete Graphen, die keine gerichteten Bäume sind.

Beispiel:



ist ein DAG, aber kein gerichteter Baum.

(b) Für jeden gerichteten Baum $B = (V, E)$, dessen Knotenmenge endlich und nicht-leer ist, gilt:

$$|E| \ = \ |V| - 1.$$

Dies folgt unmittelbar aus Satz 4.48, da der ungerichtete Graph, der entsteht, indem man in B die Kantenorientierung "vergisst" (d.h. jede gerichtete Kante (i, j) durch die ungerichtete Kante $\{i, j\}$ ersetzt), ein ungerichteter Baum ist.

Alternativ zu Definition 4.50 kann man die gerichteten Bäume, deren Knotenmenge endlich und nicht-leer ist, auch folgendermaßen definieren:

Definition 4.52. Die Klasse der gerichteten Bäume mit endlicher, nicht-leerer Knotenmenge ist rekursiv wie folgt definiert:

Basisregel: Ist V eine Menge mit $|V| = 1$, so ist $B := (V, \varnothing)$ ein gerichteter Baum.

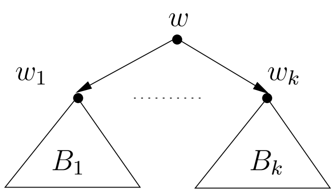
Skizze: $B := \bullet$

Der (eindeutig bestimmte) Knoten in V heißt **Wurzel** von B. Die **Höhe** von B ist 0.

Rekursive Regel: Ist $k \in \mathbb{N}_{>0}$, sind $B_1 = (V_1, E_1), \ldots, B_k = (V_k, E_k)$ gerichtete Bäume mit paarweise disjunkten Knotenmengen (d.h. $V_i \cap V_j = \varnothing$ f.a. $i, j \in \{1, \ldots, k\}$ mit $i \neq j$), sind $w_1 \in V_1, \ldots, w_k \in V_k$ die Wurzeln von $B_1, \ldots, B_k$, und ist w ein Element, das nicht in $V_1 \cup \cdots \cup V_k$ liegt, dann ist der Graph $B = (V, E)$ mit $V := \{w\} \cup V_1 \cup \cdots \cup V_k$ und $E := E_1 \cup \cdots \cup E_k \cup \{(w, w_i) : i \in \{1, \ldots, k\}\}$ ein gerichteter Baum.

Skizze:

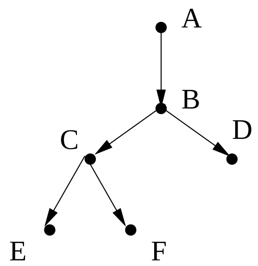
$B \ :=$



Der Knoten w heißt **Wurzel** von B. Die **Höhe** von B ist $1+\max\{h_1, \ldots, h_k\}$, wobei $h_1, \ldots, h_k \in \mathbb{N}$ die Höhen der gerichteten Bäume $B_1, \ldots, B_k$ sind.

Notation 4.53. Sei $B = (V, E)$ ein gerichteter Baum und sei $v \in V$ ein beliebiger Knoten in B. Die Knoten $v' \in V$, zu denen von v aus eine Kante führt (d.h. $(v, v') \in E$), heißen **Kinder** von v. Kinder

Beispiel. Im Graphen $G_A :=$



aus Beispiel 4.49 gilt: Knoten A hat genau ein Kind, nämlich B; Knoten B hat genau zwei Kinder, nämlich C und D; Knoten C hat genau zwei Kinder, nämlich C und F; und die Knoten D, E, F haben keine Kinder.

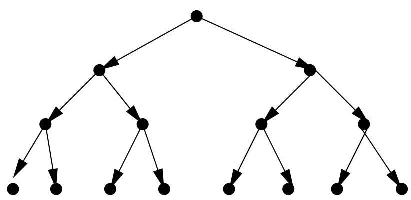
Eine besondere Rolle bei der Modellierung spielen Bäume, bei denen jeder Knoten höchstens 2 Kinder hat. Mit solchen Bäumen kann man z.B. Kaskaden von JA-NEIN-Entscheidungen oder Binär-Codierung beschreiben.

Definition 4.54 (Binärbaum, vollständiger Binärbaum)**.**

(a) Ein gerichteter Baum $B = (V, E)$ heißt **Binärbaum**, falls für jeden Knoten $v \in V$ gilt: Aus-Grad$_B(v) \leq 2$. Binärbaum

(b) Ein Binärbaum $B = (V, E)$ heißt **vollständiger Binärbaum**, falls gilt: vollständiger Binärbaum
   (1) Jeder Knoten, der kein Blatt ist, hat Aus-Grad 2.
   (2) Es gibt eine Zahl $h \in \mathbb{N}$, so dass für jedes Blatt $v \in V$ gilt: Der Weg von der Wurzel zum Blatt v hat die Länge h.

Beispiel 4.55. Der Graph G_A aus Beispiel 4.49 ist ein Binärbaum, aber kein vollständiger Binärbaum. Der Graph G_B aus Beispiel 4.49 ist kein Binärbaum. Der folgende Graph B_3 ist ein **vollständiger Binärbaum** der Höhe 3:



Zwischen der Höhe, der Anzahl der Blätter und der Anzahl der Knoten eines Binärbaums besteht der folgende wichtige Zusammenhang:

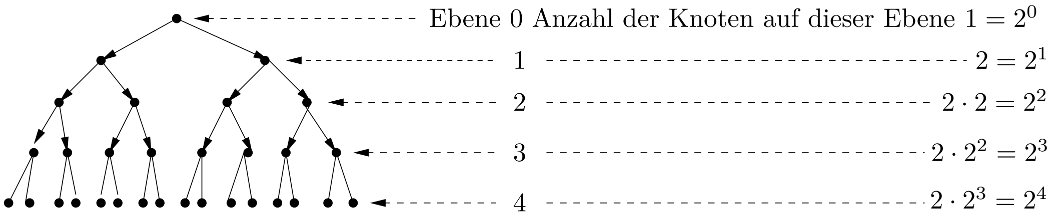
Satz 4.56. *Sei $h \in \mathbb{N}$.*

(a) Jeder ***vollständige Binärbaum der Höhe*** *h hat genau 2^h* ***Blätter*** *und genau $2^{h+1} - 1$* ***Knoten****.*

(b) Jeder ***Binärbaum der Höhe*** *h hat* ***höchstens*** *2^h* ***Blätter*** *und* ***höchstens*** *$2^{h+1} - 1$* ***Knoten****.*

Beweis:

(a) **Anschaulich:**



$\rightsquigarrow$ vollständiger Binärbaum der Höhe h hat 2^h Blätter und

$$2^0 + 2^1 + 2^2 + \cdots + 2^h \overset{\text{Satz 2.48}}{=} 2^{h+1} - 1$$

Knoten.

Formaler Beweis: Per Induktion nach h:

Induktionsanfang: $h = 0$:
Für jeden gerichteten Baum $B = (V, E)$ der Höhe 0 gilt: $|V| = 1$ und $|E| = 0$. D.h. B besteht aus genau einem Knoten, der gleichzeitig Wurzel und (einziges) Blatt des Baums ist. D.h: B hat genau $1 = 2^0 = 2^h$ Blätter und genau $1 = 2 - 1 = 2^1 - 1 = 2^{h+1} - 1$ Knoten.

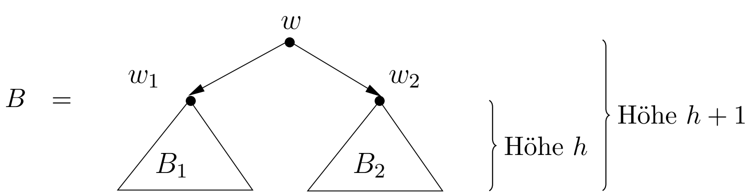
Induktionsschritt: $h \to h + 1$:
Sei $h \in \mathbb{N}$ beliebig.

Induktionsannahme: Jeder vollständige Binärbaum der Höhe h hat genau 2^h Blätter und genau $2^{h+1} - 1$ Knoten.

Behauptung: Jeder vollständige Binärbaum der Höhe $h + 1$ hat genau 2^{h+1} Blätter und genau $2^{h+2} - 1$ Knoten.

Beweis: Sei $B = (V, E)$ ein vollständiger Binärbaum der Höhe $h + 1$, und sei $w \in V$ die Wurzel von B. Wegen $h + 1 \geq 1$ hat w genau 2 Kinder; seien $w_1 \in V$ und $w_2 \in V$ diese beiden Kinder von w. Für $i \in \{1, 2\}$ sei V_i die Menge aller Knoten aus V, zu denen von w_i aus ein Weg führt; und sei $B_i := (V_i, E_i)$ der induzierte Teilgraph von B mit Knotenmenge V_i.

Skizze:



Offensichtlich ist sowohl B_1 als auch B_2 ein vollständiger Binärbaum der Höhe h. Gemäß Induktionsannahme hat jeder der beiden Bäume B_1 und B_2 genau 2^h Blätter und genau $2^{h+1} - 1$ Knoten.

Der Baum B hat daher genau $2^h + 2^h = 2^{h+1}$ Blätter und genau $1 + (2^{h+1} - 1) + (2^{h+1} - 1) = 2 \cdot 2^{h+1} - 1 = 2^{h+2} - 1$ Knoten.

(b) Analog. Details: Übung.

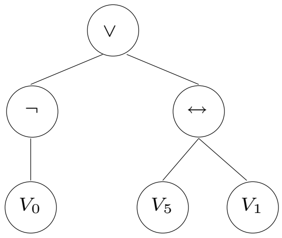
□

4.2.3 Modellierungsbeispiele

Gerichtete Bäume mit zahlreichen Knoten- und/oder Kantenmarkierungen können auf vielfältige Arten zur Modellierung genutzt werden:

Beispiel 4.57. In Kapitel 3 (Seite 58) haben wir Bäume bereits genutzt, um die Struktur einer aussagenlogischen Formel übersichtlich darzustellen (der entsprechende Baum heißt **Syntaxbaum** der Formel).

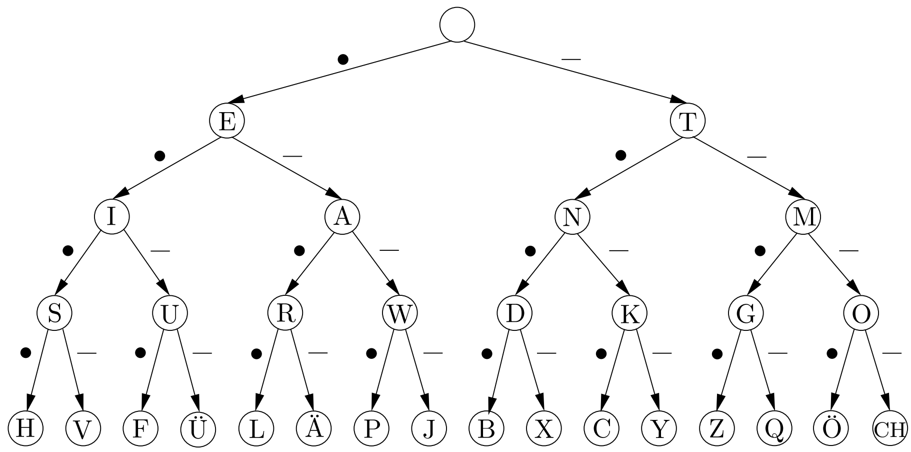
Beispiel: Syntaxbaum der Formel $\big(\neg V_0 \vee (V_5 \leftrightarrow V_1)\big)$:



Auf ähnliche Art werden markierte Bäume genutzt, um die Struktur vieler anderer Objekte (an Stelle von aussagenlogischen Formeln) zu beschreiben – z.B. für arithmetische Terme, zur Darstellung von Klassen- und Objekthierarchien oder zur Beschreibung der Struktur von Computerprogrammen oder umgangssprachlichen Texten (↗ Linguistik).

Beispiel 4.58. Folgen von Entscheidungen können in vielen Zusammenhängen durch gerichtete markierte Bäume modelliert werden – so genannte *Entscheidungsbäume*. Durch einen solchen Entscheidungsbaum erhält man beispielsweise eine kompakte Darstellung des **Morse-Codes:**

M =



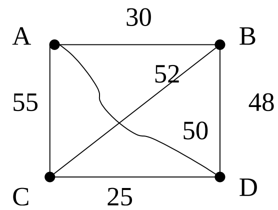
Im Morse-Code wird jeder Buchstabe durch eine Folge von kurzen und langen Signalen (• $\widehat{=}$ kurz, — $\widehat{=}$ lang) repräsentiert. Ein eingehende Meldung aus kurzen und langen Signalen wird entschlüsselt, indem man an der Wurzel des Baums M beginnt und bei einem kurzen Signal nach links, bei einem langen nach rechts weitergeht. Eine längere Pause zeigt an, dass ein Buchstabe vollständig übermittelt ist.

In jedem Entscheidungsbaum modellieren die Knoten einen Zwischenstand bei der Entscheidungsfindung. Sie können entsprechend markiert sein, z.B. mit dem codierten Buchstaben des Morse-Codes. Die Kanten, die von einem Knoten ausgehen, modellieren die Alternativen, aus denen in dem durch den Knoten repräsentierten "Zustand" eine ausgewählt werden kann – im

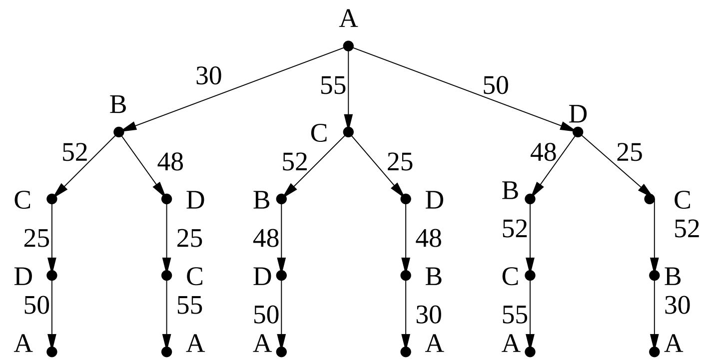
Morse-Code ist das jeweils ein kurzes oder ein langes Signal, das als Kantenmarkierung angegeben wird.

Beispiel 4.59. Markierte Bäume können auch genutzt werden, um den Lösungsraum kombinatorischer Probleme darzustellen.

Beispiel: Ein Handlungsreisender soll einen möglichst kurzen Rundweg finden, auf dem er jede der Städte A, B, C, D besucht. Die Entfernungen (in km) zwischen den Städten sind als Kantenmarkierungen des folgenden Graphen gegeben:



Alle möglichen in Stadt A startenden Rundwege werden durch den folgenden Baum dargestellt:



Jeder Weg von der Wurzel zu einem Blatt repräsentiert dabei einen Rundweg, auf dem jede der Städte genau einmal besucht wird. Die Kantenmarkierungen geben die Entfernungen zwischen einzelnen Städten wieder. Eine zusätzliche Knotenmarkierung an jedem Blatt gibt die Gesamtlänge des entsprechenden Rundwegs an. Die beiden kürzesten Rundwege für unseren Handlungsreisenden sind also

$$(A, B, C, D, A) \quad \text{und} \quad (A, D, C, B, A).$$

Bemerkung 4.60. Nach dem gleichen Schema kann man auch Zugfolgen in Spielen modellieren: Jeder Knoten des Entscheidungsbaums modelliert einen Spielzustand. Die von dort ausgehenden Kanten geben an, welche Möglichkeiten für den nächsten Zug bestehen. Solche Darstellungen werden z.B. in Schachprogrammen verwendet, um die Folgen der anstehenden Entscheidung zu analysieren und zu bewerten.

Beachte: Bei der Modellierung von Spielabläufen können manche "Spielzustände" (z.B. Konfigurationen eines Schachbretts) auf unterschiedlichen Wegen (d.h. Spielverläufen) erreicht werden, und trotzdem "im Sinne des Spiels" den selben Zustand beschreiben. In solchen Fällen könnte man im Entscheidungsbaum die zugehörigen Knoten zu einem einzigen Knoten zusammenfassen. Damit geht dann allerdings die Baum-Eigenschaft verloren, und es entsteht ein allgemeiner gerichteter Graph, der auch Kreise enthalten kann. Ein Kreis entspricht dann der Situation, dass eine Folge von Spielzügen in einen Zustand zurückführt, der früher schon einmal durchlaufen wurde.

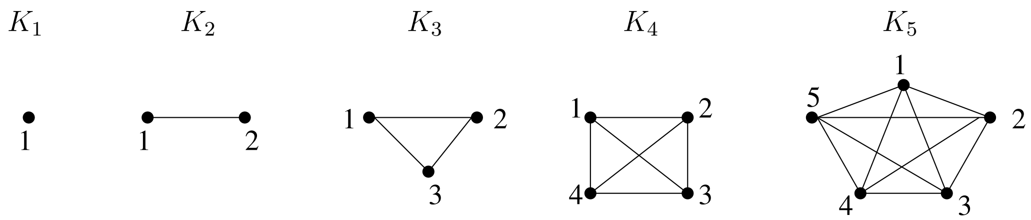
4.3 Einige spezielle Arten von Graphen

In diesem Abschnitt werden einige spezielle Arten von Graphen vorgestellt, die Ihnen im Laufe der nächsten Semester immer wieder begegnen werden.

4.3.1 Spezielle ungerichtete Graphen

Definition 4.61 (Der vollständige Graph K_n)**.** Sei $n \in \mathbb{N}_{>0}$. Der **vollständige ungerichtete Graph** K_n hat Knotenmenge $\{1,\ldots,n\}$ und Kantenmenge $\big\{\{i,j\} : i,j \in \{1,\ldots,n\},\, i \neq j\big\}$. vollständiger ungerichteter Graph

Beispiele.



Beobachtung. Der vollständige Graph K_n hat n Knoten und $\frac{n(n-1)}{2}$ Kanten.

Definition 4.62 (Der vollständige bipartite Graph $K_{m,n}$)**.** Seien $m,n \in \mathbb{N}_{>0}$. Der **vollständige ungerichtete bipartite Graph** $K_{m,n}$ hat Knotenmenge vollständiger ungerichteter bipartiter Graph

$$\big\{(1,i) : i \in \{1,\ldots,n\}\big\} \ \dot\cup\ \big\{(2,j) : j \in \{1,\ldots,m\}\big\}$$

und Kantenmenge

$$\big\{\{(1,i),(2,j)\} \ : \ i \in \{1,\ldots,n\},\, j \in \{1,\ldots,m\}\big\}.$$

Beispiele. Siehe Abbildung 4.1.

Beobachtung. Der Graph $K_{m,n}$ hat $m+n$ Knoten und $m \cdot n$ Kanten.

Notation 4.63. Ein ungerichteter Graph G mit endlicher, nicht-leerer Knotenmenge heißt

(a) **vollständig**, falls es ein $n \in \mathbb{N}_{>0}$ gibt, so dass $G \cong K_n$ (d.h. G ist isomorph zu K_n). vollständig

(b) **vollständig bipartit**, falls es Zahlen $m,n \in \mathbb{N}_{>0}$ gibt, so dass $G \cong K_{m,n}$. vollständig bipartit

4.3.2 Spezielle gerichtete Graphen

Gemäß Definition 4.6 ("gerichteter Graph") und Definition 2.27(c) ("k-stellige Relation") kann jeder gerichtete Graph $G = (V,E)$ als eine 2-stellige Relation über V aufgefasst werden, da die Kantenmenge E von G ja gerade eine Teilmenge von $V^2 = V \times V$ ist. Umgekehrt können wir natürlich auch jede 2-stellige Relation R über einer Menge M als gerichteten Graph auffassen, dessen Knotenmenge M und dessen Kantenmenge R ist. Gerichtete Graphen sind also dasselbe wie 2-stellige Relationen über einer Menge V.

Von besonderem Interesse sind 2-stellige Relationen, die eine oder mehrere der folgenden Eigenschaften besitzen:

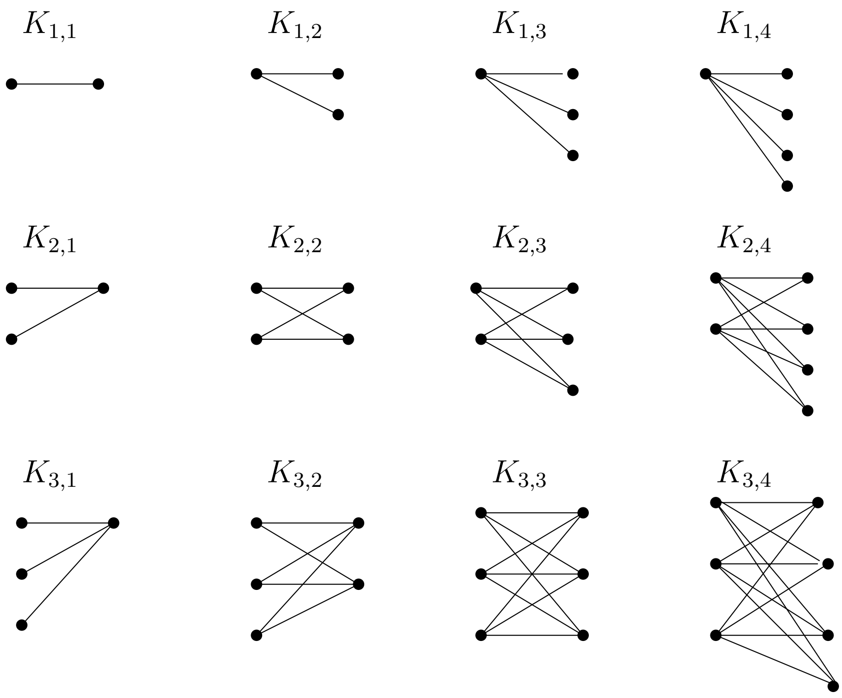


Abbildung 4.1: einige vollständige bipartite Graphen

Definition 4.64. Sei E eine 2-stellige Relation über einer Menge V (d.h. $E \subseteq V \times V$, d.h. $G = (V, E)$ ist ein gerichteter Graph).

reflexiv

(a) E heißt **reflexiv**, falls für alle $v \in V$ gilt:

$$(v, v) \in E. \qquad \text{(Skizze: } v \circlearrowleft \text{)}$$

symmetrisch

(b) E heißt **symmetrisch**, falls f.a. $v, w \in V$ gilt:

$$\text{Wenn } (v, w) \in E, \text{ dann auch } (w, v) \in E.$$

(d.h. zu jeder Kante $v \longrightarrow w$ gibt es auch eine "Rückwärtskante" $v \longleftarrow w$)

antisymmetrisch

(c) E heißt **antisymmetrisch**, falls f.a. $v, w \in V$ gilt:

$$\text{Wenn } (v, w) \in E \text{ und } (w, v) \in E, \text{ dann } v = w.$$

(d.h: Ist $v \neq w$, so gibt es in E allenfalls eine der beiden Kanten $v \longrightarrow w$ und $v \longleftarrow w$)

konnex

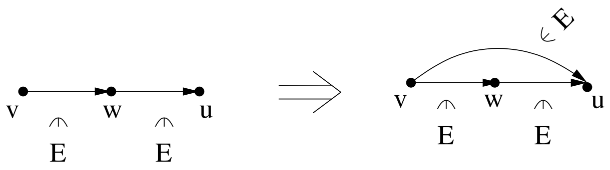
(d) E heißt **konnex**, falls f.a. $v, w \in V$ gilt:

$$(v, w) \in E \text{ oder } (w, v) \in E.$$

(d.h. mindestens eine der beiden Kanten $v \longrightarrow w$ und $v \longleftarrow w$ liegt in E)

(e) E heißt **transitiv**, falls f.a. $v, w, u \in V$ gilt: transitiv

$$\text{Ist } (v,w) \in E \text{ und } (w,u) \in E, \text{ so auch } (v,u) \in E.$$

(Skizze: 

)

4.3.3 Äquivalenzrelationen

Definition 4.65. Eine **Äquivalenzrelation** ist eine 2-stellige Relation, die **reflexiv**, **symmetrisch** und **transitiv** ist. Äquivalenzrelation

Beispiel 4.66. Beispiele für Äquivalenzrelationen:

(a) **Gleichheit:** Für jede Menge M ist

$$E \;:=\; \{(m,m) : m \in M\}$$

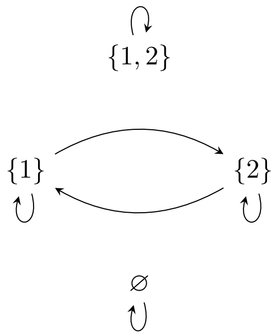
eine Äquivalenzrelation. Die Aussage “$(x,y) \in E$” entspricht gerade der Aussage “$x = y$”.

(b) **Gleichmächtigkeit:** Für jede endliche Menge M ist

$$E \;:=\; \{(A,B) : A \subseteq M,\, B \subseteq M,\, |A| = |B|\}$$

eine Äquivalenzrelation über der Potenzmenge $\mathcal{P}(M)$.

Skizze für $M = \{1,2\}$:



(c) **Logische Äquivalenz:** Die Relation

$$E \;:=\; \{(\varphi,\psi) : \varphi \in \text{AL},\, \varphi \equiv \psi\}$$

ist eine Äquivalenzrelation über der Menge AL aller aussagenlogischen Formeln.

4.3.4 Ordnungsrelationen

Definition 4.67 (Ordnungen)**.** Sei E eine 2-stellige Relation über einer Menge V.

(a) E heißt **Präordnung**, falls E **reflexiv und transitiv** ist. Präordnung

(b) E heißt **partielle Ordnung**, falls E **reflexiv, transitiv und antisymmetrisch** ist. partielle Ordnung

lineare Ordnung
totale Ordnung

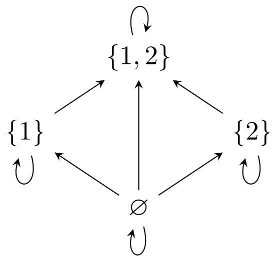
(c) E heißt **lineare Ordnung** oder **totale Ordnung**, falls E **reflexiv, transitiv, antisymmetrisch und konnex** ist.

Beispiel 4.68.

(a) $\leq$ ist eine **lineare Ordnung** auf $\mathbb{N}$ (und $\mathbb{Z}, \mathbb{Q}, \mathbb{R}$). Ebenso ist $\geq$ eine lineare Ordnung auf $\mathbb{N}$ (und $\mathbb{Z}, \mathbb{Q}, \mathbb{R}$).

(b) Für jede Menge M sind $\subseteq$ und $\supseteq$ **partielle Ordnungen** auf der Potenzmenge $\mathcal{P}(M)$.

Skizze für $M = \{1, 2\}$:



(c) Die "Folgerungsrelation"

$$E \;:=\; \{(\varphi, \psi) : \varphi \in \mathrm{AL},\, \psi \in \mathrm{AL},\, \varphi \models \psi\}$$

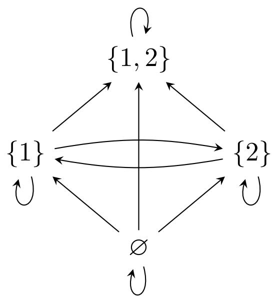
ist eine **Präordnung auf AL**.

(d) Für jede endliche Menge M ist

$$E \;:=\; \{(A, B) : A \subseteq M,\, B \subseteq M,\, |A| \leq |B|\}$$

eine **Präordnung** auf $\mathcal{P}(M)$.

Skizze für $M = \{1, 2\}$:



4.3.5 Die reflexive transitive Hülle einer Relation

reflexive transitive Hülle
reflexiver transitiver Abschluss

Definition 4.69. Sei $G = (V, E)$ ein gerichteter Graph. Die **reflexive transitive Hülle** (bzw. der **reflexive transitive Abschluss**) von E auf V ist die rekursiv wie folgt definierte Relation $E^* \subseteq V \times V$:

Basisregeln:

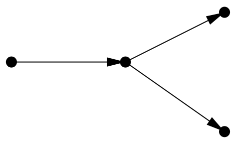
- F.a. $v \in V$ ist $(v, v) \in E^*$.
- F.a. $(v, w) \in E$ ist $(v, w) \in E^*$

Rekursive Regel:

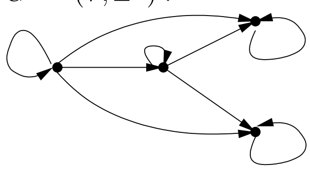
- Sind $(v, w) \in E^*$ und $(w, u) \in E^*$, so ist auch $(v, u) \in E^*$.

Beispiel.

$G = (V, E) \; :=$



$G^* = (V, E^*) :$



Beobachtung 4.70. Sei $G = (V, E)$ ein gerichteter Graph und seien $v, w \in V$. Dann sind die beiden folgenden Aussagen äquivalent:

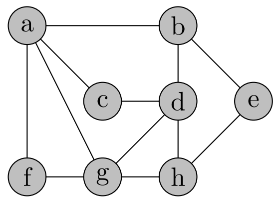
(a) Es gibt in G einen Weg von v nach w.

(b) $(v, w) \in E^*$, wobei E^* die reflexive transitive Hülle von E auf V ist.

Beweis: Übung. □

4.4 Übungsaufgaben zu Kapitel 4

Aufgabe 4.1. Betrachten Sie den ungerichteten Graphen G auf der rechten Seite.



(a) Geben Sie die Knotenmenge V und die Kantenmenge E des Graphen G an. Repräsentieren Sie G außerdem durch eine Adjazenzmatrix und eine Adjazenzliste.

(b) Geben Sie einen Euler-Weg in G an. Besitzt G auch einen Euler-Kreis?

(c) Geben Sie einen Hamilton-Kreis in G an.

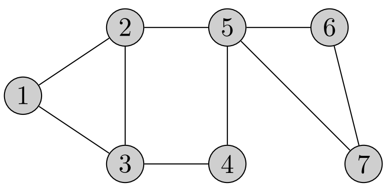
(d) Zeigen Sie, dass G orientierbar ist.

(e) Geben Sie einen Spannbaum von G an, den man so wurzeln kann, dass er die Höhe 2 hat. Kennzeichnen Sie die Wurzel in Ihrer Lösung.

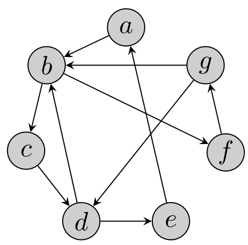
(f) Geben Sie den Kanten von G Richtungen, so dass der entstehende gerichtete Graph genau zwei starke Zusammenhangskomponenten besitzt. Geben Sie die Knotenmengen der beiden starken Zusammenhangskomponenten an.

Aufgabe 4.2. Es seien die folgenden beiden Graphen G_1 und G_2 gegeben:

G_1:



G_2:

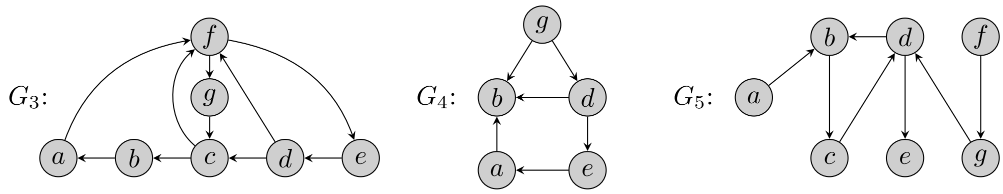


(a) Geben Sie für jeden der beiden Graphen G_1 und G_2 die Knotenmenge und die Kantenmenge an. Repräsentieren Sie außerdem jeden der beiden Graphen durch eine Adjazenzmatrix und eine Adjazenzliste.

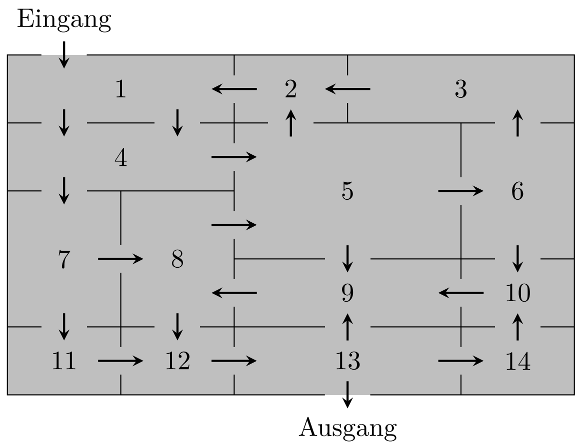
(b) Geben Sie einen Weg von 2 nach 4 in G_1 an, der *nicht* einfach ist. Geben Sie außerdem einen Kreis in G_1 an, der *nicht* einfach ist und durch den Knoten 2 verläuft.

(c) Ist G_1 zusammenhängend? Ist G_2 stark zusammenhängend? Ist G_2 azyklisch?

(d) Überprüfen Sie für jeden der folgenden Graphen G, ob folgendes gilt: (i) $G = G_2$, (ii) G ist ein Teilgraph von G_2, (iii) G ist ein induzierter Teilgraph von G_2, (iv) G ist isomorph zu G_2. Geben Sie bei (d) auch einen Isomorphismus von G nach G_2 an, falls dieser existiert.



Aufgabe 4.3. Die folgende Abbildung stellt den Grundriss eines Irrgartens dar.



Die Türen in diesem Irrgarten schwingen nur zu einer Seite auf und haben keine Klinken o.ä. Nachdem also ein Besucher die Eingangstür oder eine nachfolgende Tür durchschritten hat und die Tür hinter ihm zugefallen ist, kann der Besucher nicht mehr durch diese Tür zurück. Die Tür bleibt aber für weitere Durchgänge in der ursprünglichen Richtung benutzbar. Die allgemeinen Sicherheitsbestimmungen für Irrgärten schreiben vor, dass jeder Besucher, der den Irrgarten betritt, – egal wie er läuft – den Ausgang erreichen kann.

(a) Modellieren Sie den Irrgarten durch einen Graphen.

(b) Formulieren Sie die allgemeinen Sicherheitsbestimmungen für Irrgärten mit Begriffen der Graphentheorie.

(c) Überprüfen Sie anhand der Formulierungen aus (b), ob der angegebene Irrgarten den allgemeinen Sicherheitsbestimmungen entspricht.

Aufgabe 4.4. Sie bekommen die Aufgabe, $n \in \mathbb{N}_{>0}$ Rechner zu vernetzen. Ihr Auftraggeber verlangt folgende Eigenschaften des Netzwerkes:

(1) Von jedem Rechner muss jeder andere Rechner über einen Leitungsweg erreichbar sein.

(2) Auch wenn genau eine Leitung zwischen zwei Rechnern ausfällt, muss jeder Rechner über einen Leitungsweg mit jedem anderen Rechner verbunden sein.

(3) An jedem Rechner können maximal vier Leitungen angeschlossen werden.

Dabei können auf einer Leitung Daten in beide Richtungen gesendet werden. Ein solches Netzwerk lässt sich leicht als ungerichteter Graph darstellen: ein Knoten repräsentiert einen Rechner, und eine Kante repräsentiert eine Leitung.

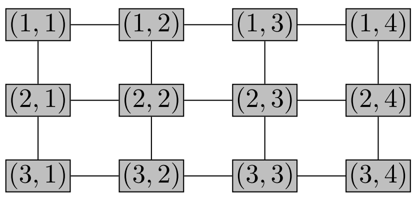
(a) Formulieren Sie die Eigenschaften (1), (2) und (3) mit Begriffen der Graphentheorie.

(b) Untersuchen Sie die folgenden Graphen G_1, G_2 und G_3 auf Ihre Tauglichkeit bezüglich der Eigenschaften (1), (2) bzw. (3):
- $G_1 = (V_1, E_1)$ mit $V_1 = \{1, 2, \ldots, n\}$ und $E_1 = \{\{1, i\} : 2 \leq i \leq n\}$
- $G_2 = (V_2, E_2)$ mit $V_2 = V_1$ und $E_2 = \{\{i, i+1\} : 1 \leq i < n\}$
- $G_3 = (V_3, E_3)$ mit $V_3 = V_1$ und $E_3 = E_2 \cup \{\{n, 1\}\}$

Aufgabe 4.5. Für $m, n \in \mathbb{N}_{>0}$ sei das $m \times n$-*Gitter* der Graph $G_{m\times n} = (V_{m\times n}, E_{m\times n})$ mit

$$\begin{aligned} V_{m\times n} &:= \big\{ (i,j) \,:\, 1 \leq i \leq m,\, 1 \leq j \leq n \big\}, \\ E_{m\times n} &:= \big\{ \{(i,j),(i,j+1)\} \,:\, 1 \leq i \leq m,\, 1 \leq j < n \big\} \cup \\ & \big\{ \{(i,j),(i+1,j)\} \,:\, 1 \leq i < m,\, 1 \leq j \leq n \big\}. \end{aligned}$$

Das 3×4-Gitter $G_{3\times 4}$ sieht z.B. wie folgt aus:



(a) Überprüfen Sie, ob $G_{3\times 4}$ bipartit ist. Falls $G_{3\times 4}$ bipartit ist, so geben Sie zwei disjunkte Knotenmengen $V_1, V_2 \subseteq V_{3\times 4}$ mit $V_1 \cup V_2 = V_{3\times 4}$ an, so dass jede Kante aus $E_{3\times 4}$ einen Knoten aus V_1 und einen Knoten aus V_2 miteinander verbindet. Falls $G_{3\times 4}$ nicht bipartit ist, so begründen Sie dies.

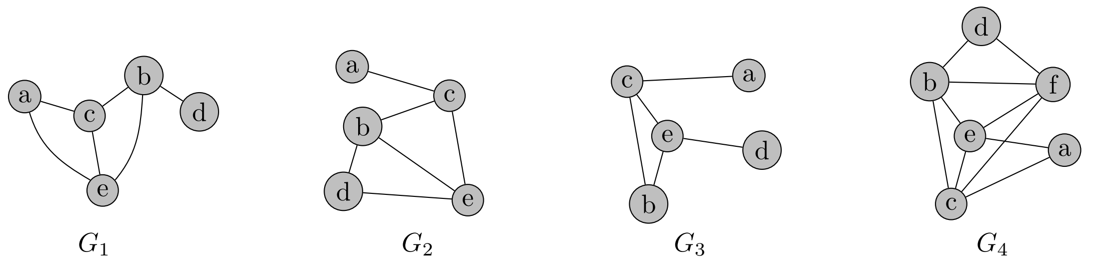
(b) Geben Sie ein Matching maximaler Größe in $G_{3\times 4}$ an.

(c) Geben Sie einen Hamilton-Kreis in $G_{3\times 4}$ an.

(d) Für welche $m, n \in \mathbb{N}_{>0}$ besitzt $G_{m\times n}$ einen Hamilton-Kreis, für welche nicht?

Hinweis: Verallgemeinern Sie die Argumentation auf Seite 136 (oben) im Buch „Modellierung – Grundlagen und formale Methoden" von Uwe Kastens und Hans Kleine Büning. Das Buch befindet sich unter Anderem im Semesterapparat zur Vorlesung in der Informatik-Bibliothek.

Aufgabe 4.6. Es seien die folgenden ungerichteten Graphen G_1, G_2, G_3 und G_4 gegeben:



(a) Überprüfen Sie für alle $i, j \in \{1, 2, 3, 4\}$ mit $i \neq j$, ob Folgendes gilt:

- $G_i = G_j$
- G_i ist ein Teilgraph von G_j
- G_i ist ein induzierter Teilgraph von G_j

(b) Überprüfen Sie, welche der Graphen isomorph zueinander sind. Falls zwei Graphen G_i und G_j isomorph sind, so geben Sie einen Isomorphismus von G_i nach G_j an. Falls hingegen G_i und G_j nicht isomorph sind, so begründen Sie dies.

(c) Welche der Graphen kann man nachzeichnen, ohne den Stift abzusetzen oder eine Kante doppelt zu ziehen?

Aufgabe 4.7. König Artus will für die Tafelrunde eine Sitzordnung für sich und neun seiner Ritter festlegen, bei der er und die neun Ritter im Kreis an einem runden Tisch sitzen. Das wäre nicht schwer, gäbe es nicht diese Rivalitäten und Eifersüchteleien zwischen den Rittern. König Artus möchte, dass Lancelot zu seiner Rechten und Kay zu seiner Linken sitzt. Erec weigert sich, neben jemand anderem als Lyonel oder Tristan zu sitzen. Galahad will weder neben Tristan noch neben Lancelot oder Lyonel sitzen. Parzival lehnt es ab, neben Gawain, Lancelot oder Lyonel zu sitzen. Gaheris möchte auf keinen Fall neben Gawain, Lancelot oder Kay sitzen. Tristan weigert sich, neben Lancelot, Parzival oder Kay zu sitzen. Gawain würde sich neben jeden anderen setzen, aber nicht neben Galahad oder Kay. Und Lyonel ist dagegen, neben Gawain zu sitzen.

(a) Stellen Sie den Konfliktgraphen auf.

(b) Verwenden Sie den Konfliktgraphen aus (a), um eine Tischordnung aufzustellen, die von allen akzeptiert wird. Zeichnen Sie den entsprechenden Graph und die Sitzordnung.

Aufgabe 4.8. Auf dem Weihnachtsmarkt von Großdorf sollen insgesamt 8 Stände rund um den Marktplatz arrangiert werden. Die 8 Stände setzen sich folgendermaßen zusammen:

- Ein Stand, in dem die traditionelle Weihnachtskrippe aufgebaut ist.
- Zwei Stände, an denen Kunsthandwerk verkauft wird: einer der beiden Stände ist die Töpferei, der andere bietet Holzschmuck aus dem Erzgebirge an.
- Zwei Glühweinstände; einer davon wird von Herrn Max, der andere von Frau Peters betrieben.
- Drei Essensstände; einer davon verkauft Crêpes, der andere Waffeln und der dritte Steaks vom Holzkohlegrill.

Bei der Platzierung der 8 Stände um den Marktplatz ist folgendes zu beachten: Neben der Weihnachtskrippe darf keiner der Glühweinstände platziert werden. Essensstände dürfen nicht nebeneinander stehen, die beiden Glühweinstände dürfen nicht nebeneinander stehen, und die beiden Kunsthandwerkstände dürfen nicht nebeneinander stehen. Aus Sicherheitsgründen darf der Holzkohlegrill weder neben der Weihnachtskrippe noch neben dem Stand mit dem Holzschmuck aus dem Erzgebirge stehen. Herr Max ist mit den Besitzern des Holzkohlegrills und der Töpferei befreundet und möchte daher unbedingt die beiden als Nachbarn haben. Außerdem ist zu beachten, dass sich der Betreiber des Waffelstands weder mit Frau Peters noch mit dem Besitzer der Töpferei verträgt und daher auf keinen Fall neben einem der beiden platziert werden will.

(a) Stellen Sie den Konfliktgraphen und das Komplement des Konfliktgraphen auf.

(b) Gibt es im Komplement des Konfliktgraphen einen Hamiltonkreis? Falls ja, dann geben Sie einen solchen Hamiltonkreis an. Falls nein, dann begründen Sie, warum es keinen gibt.

(c) Geben Sie eine Platzierung der 8 Stände rund um den Marktplatz an, mit der alle zufrieden sind.

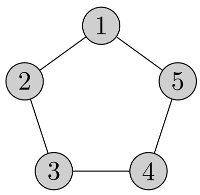
Aufgabe 4.9. Sei $G = (V, E)$ ein ungerichteter, zusammenhängender Graph, dessen Knotenmenge endlich ist. Beweisen Sie, dass G genau dann einen Euler-Weg besitzt, der kein Euler-Kreis ist, wenn es in G genau zwei Knoten mit ungeradem Grad gibt.

Aufgabe 4.10. Zwei Personen A und B spielen ein Spiel auf einem zusammenhängenden ungerichteten Graphen $G = (V, E)$. Die Spieler wählen abwechselnd Knoten $v_1, v_2, v_3, \dots$ aus V, so dass $v_1, v_2, v_3, \dots$ verschiedene Knoten sind und jeweils gilt: $\{v_i, v_{i+1}\} \in E$. Den ersten Knoten wählt A. Der letzte Spieler, der einen Knoten wählen kann, gewinnt.

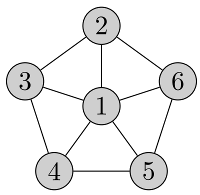
Ein Spieler hat eine *Gewinnstrategie* in dem Spiel genau dann, wenn der Spieler das Spiel, unabhängig davon wie der andere Spieler spielt, gewinnen kann.

(a) Geben Sie für jeden der beiden folgenden Graphen G_1 und G_2 ein Matching maximaler Größe an und entscheiden Sie, welcher der beiden Spieler in dem Spiel auf dem entsprechenden Graph eine Gewinnstrategie hat.

G_1:



G_2:



(b) Beweisen Sie, dass die beiden folgenden Aussagen äquivalent sind:

(i) G besitzt ein Matching M, so dass jeder Knoten aus V zu mindestens einer Kante aus M inzident ist.

(ii) Spieler B hat eine Gewinnstrategie in dem oben beschriebenen Spiel auf G.

Aufgabe 4.11. Es soll ein Klausurplan für 7 Klausuren A–G aufgestellt werden, bei dem kein Student mehr als eine Klausur pro Tag schreiben muss. Über die Teilnehmer an den Klausuren ist Folgendes bekannt:

- Für jede der Klausuren B,C,E und G gibt es mindestens einen Studenten, der sich für diese Klausur und A angemeldet hat.

- Es gibt Studenten, die sich für B und C angemeldet haben, als auch Studenten, die die Kombination B und E gewählt haben.
- Jeder Student, der D mitschreibt, hat sich auch für C und G angemeldet.
- Mindestens ein Teilnehmer der Klausur G nimmt an F teil.

(a) Stellen Sie den Konfliktgraphen auf.

(b) Geben Sie einen Klausurplan an, bei dem kein Student an mehr als einer Klausur pro Tag teilnimmt.

(c) Wie viele Tage werden für einen solchen Klausurplan benötigt?

Aufgabe 4.12. Beweisen Sie, dass jeder ungerichtete, zusammenhängende Graph $G = (V, E)$, dessen Knotenmenge V endlich ist, einen Spannbaum besitzt.

Hinweis: Verwenden Sie Induktion nach $n := |E|$.

Aufgabe 4.13. Zwei Spieler A und B spielen das folgende Spiel. Das Spiel ist in Runden aufgeteilt, wobei Spieler A in den geraden Runden und Spieler B in den ungeraden Runden spielt. In der ersten Runde wählt Spieler B eine Zahl aus $\{1, 2\}$. In jeder der nachfolgenden Runden wählt der jeweilige Spieler eine Zahl aus $\{1, 2, 3\}$ mit der Einschränkung, dass die Zahl aus der vorhergehenden Runde nicht gewählt werden darf. Nach jeder Runde wird die Summe der bereits gewählten Zahlen berechnet. Nimmt diese Summe den Wert 6 an, so gewinnt der Spieler der jeweiligen Runde; übersteigt sie den Wert 6, so verliert er.

(a) Beschreiben Sie das Spiel durch einen Entscheidungsbaum.

(b) Wer gewinnt, wenn beide Spieler optimal spielen, d.h. wenn jeder Spieler immer nur diejenigen Zahlen wählt, mit denen er – falls dies noch möglich ist – gewinnen kann?

Aufgabe 4.14. Sei $\mathcal{G}$ die Menge der ungerichteten Graphen $G = (V, E)$ mit $V \subseteq \mathbb{N}$.

(a) Unter (i) bis (iii) sind zweistellige Relationen über $\mathcal{G}$ gegeben. Überprüfen Sie für jede dieser Relationen, ob sie reflexiv, symmetrisch, antisymmetrisch, konnex bzw. transitiv ist.

   (i) $\{\, (G, G') \in \mathcal{G}^2 \,:\, G \cong G' \,\}$

   (ii) $\{\, (G, G') \in \mathcal{G}^2 \,:\, G$ hat höchstens so viele Knoten wie $G' \,\}$

   (iii) $\{\, (G, G') \in \mathcal{G}^2 \,:\, G'$ besitzt einen Teilgraph G'' mit $G'' \cong G \,\}$

(b) Zeigen Sie, dass die zweistellige Relation

$$R \;:=\; \{\, (G, G') \in \mathcal{G}^2 \,:\, G \text{ ist ein induzierter Teilgraph von } G' \,\}$$

eine partielle Ordnung, aber keine lineare Ordnung, auf $\mathcal{G}$ ist.

Literaturverzeichnis

[1] U. Kastens und H. Kleine Büning. *Modellierung. Grundlagen und formale Methoden*
Hanser, 2005.

[2] C. Meinel und M. Mundhenk. *Mathematische Grundlagen der Informatik. Mathematisches Denken und Beweisen.*
Teubner, 2002.

[3] A. Beutelspacher. *„Das ist o.B.d.A. trivial!“ Tipps und Tricks zur Formulierung mathematischer Gedanken.*
Vieweg Studium.

[4] M. Kreuzer und S. Kühlig. *Logik für Informatiker.*
Pearson Studium, 2006.

[5] U. Schöning. *Logik für Informatiker.*
Springer, 2000.

[6] R. Diestel. *Graphentheorie.*
Springer, 2006 (dritte Auflage).

[7] L. Lovasz, J. Pelikan und K. Vesztergombi. *Discrete Mathematics. Elementary and Beyond*
Springer, 2003.

[8] J. E. Hopcroft, R. Motwani und J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation.*
Addison-Wesley, 2001.

[9] U. Schöning. *Theoretische Informatik - kurzgrfasst.*
Springer, 2001.

[10] I. Wegener. *Kompendium Theoretische Informatik - eine Ideensammlung.*
Teubner, 1996.

[11] I. Wegener. *Theoretische Informatik.*
Teubner, 1993.