WAIT

Wernigeröder
Automatisierungs- und
Informatik-
Texte

Logik für Informatiker

Prof. Dr. Frieder Stolzenburg

Hochschule Harz
Hochschule für angewandte Wissenschaften

01
2018

Hochschule Harz
Fachbereich Automatisierung und Informatik

Logik für Informatiker

Skript zur Vorlesung

Prof. Dr. Frieder Stolzenburg
http://www.hs-harz.de/fstolzenburg/

Stand: 21. März 2019

Inhaltsverzeichnis

Einführung

Mathematik und Logik sind die Grundlagen der Informatik und schulen das abstrakt-logische Denken in Zusammenhängen. Praktische Bedeutung hat die Logik und Mengenlehre insbesondere in folgenden Bereichen:

- Formulierung von Bedingungen in Programmen (z. B. *if-then-else*-Anweisungen)
- Spezifikation und Validierung von Software (z. B. Algebraische Spezifikation, Programmverifikation)
- Logik als Programmiersprache (z. B. Constraint-Logikprogrammierung, Datenbank-Abfragen)

Zitat

Alles was überhaupt gedacht werden kann, kann klar gedacht werden. Alles, was sich aussprechen läßt, läßt sich klar aussprechen.

Meine Sätze erläutern dadurch, daß sie der, welcher mich versteht, am Ende als unsinnig erkennt, wenn er durch sie – auf ihnen – über sie hinausgestiegen ist. (Er muß sozusagen die Leiter wegwerfen, nachdem er auf ihr hinaufgestiegen ist.)

Wovon man nicht sprechen kann, darüber muß man schweigen.

—Ludwig Wittgenstein: *Tractatus logico-philosophicus* (4.116; 6.54; 7) [Wit63]

1 Mathematische Grundlagen

1.1 Mengen

Definition 1.1.1 (Cantor, 1895)**.** Eine Menge ist eine Zusammenfassung M von bestimmten, wohlunterschiedenen Objekten unserer Anschauung oder unseres Denkens (welche die Elemente von M genannt werden) zu einem Ganzen.

Abgrenzung

Axiomatische Mengenlehre betrachten wir hier nicht.

Schreibweise

- Mengen werden mit Hilfe geschweifter Klammern $\{\,\}$ notiert.
- Mengen lassen sich *extensional* definieren, d. h. durch Aufzählung der in ihr enthaltenen Elemente, z. B. die Menge der Ziffern als $\{0, 1, 2, \ldots, 9\}$, wobei hier Kommata die einzelnen Mengen-Elemente voneinander trennen.
- Mengen lassen sich aber auch *intensional* definieren, d. h. durch Angabe einer charakteristischen Eigenschaft der Mengen-Elemente, z. B. die Menge der Ziffern als $\{x \in \mathbb{N}_0 \mid 0 \leq x \leq 9\}$.

  Der Strich $\mid$ bedeutet hierbei „mit der Eigenschaft“.
  $x \in \mathbb{N}_0$ bedeutet „x ist Element der Menge der natürlichen Zahlen einschließlich 0“.
- In jedem Fall ist eine Menge M aber durch ihre Extension, d. h. durch die Menge der in M enthaltenen Elemente, charakterisiert, z. B.:

$$\{153\} = \{x \mid x \text{ ist die Anzahl der Fische im Netz}\}$$

Übung 1.1.2. Nennen Sie Beispiele von Mengen (z. B. Zahlen)!

$$\hookrightarrow \mathbb{N}, \mathbb{N}_0, \mathbb{Q}, \mathbb{R}, \mathbb{R}_0^+, \mathbb{Z}$$

Begriffe

- Element-Relation:
  $m \in M \mathrel{\widehat{=}}$ „m ist Element der Menge M“
- leere Menge (enthält keine Elemente):
  $\emptyset$, eindeutig bestimmt, für alle m gilt: $m \notin \emptyset$
  Dabei steht $\notin$ für „ist nicht Element von“.
- Teilmenge:
  $N \subseteq M$ gdw. für alle $m \in N$ gilt auch $m \in M$
- Anzahl der Elemente der Menge: $|M|$

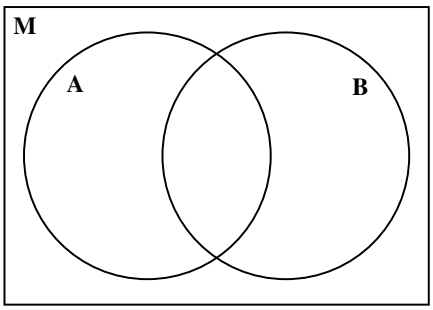
- Potenzmenge:
  Menge aller Teilmengen der Menge M
  alternative Schreibweisen: $\mathcal{P}(M), 2^M$

- Mengen-Operationen:

Vereinigung:	$A \cup B = \{x \mid x \in A \text{ oder } x \in B\}$
Schnitt:	$A \cap B = \{x \mid x \in A \text{ und } x \in B\}$
Komplement:	$\overline{A} = \{x \in M \mid x \notin A\}$ (bezogen auf Grundmenge M)
Differenz:	$A \setminus B = A \cap \overline{B}$
kartesisches Produkt:	$A \times B = \{(x, y) \mid x \in A, y \in B\}$

  (x, y) bezeichnet ein geordnetes Paar der Elemente x und dann y.

Übung 1.1.3 (Venn-Diagramm)**.** Kennzeichnen Sie im nachfolgenden Mengendiagramm für zwei Mengen A und B in der Grundmenge M die Mengen $A \cup B$, $A \cap B$, $\overline{A}$ und $A \setminus B$!



Übung 1.1.4. Zwei Väter und zwei Söhne kaufen ein, und zwar drei Smartphones. Jeder hat genau ein Smartphone. Wie geht das?

Übung 1.1.5. Sei $M = \{0, 1\}$. Was ist $|M|$, $M \times M$ und $\mathcal{P}(M)$?
Gilt $M \subseteq M$ oder $M \in M$?

Übung 1.1.6 (Russell'sche Antinomie)**.** Lässt sich die Menge M aller Mengen, die sich nicht selbst als Element enthalten, bilden?

1.2 Relationen

Definition 1.2.1 (Relation)**.** Seien M, M_1, ..., M_n nicht-leere Mengen. Eine Menge $R \subseteq M_1 \times M_2 \times \cdots \times M_n$ wird als *n-stellige Relation* bezeichnet. Eine Menge $R \subseteq M \times M$ heißt *binäre Relation* auf M. Statt $(x, y) \in R$ schreibt man auch xRy. Eine 1-stellige Relation ist eine Menge, und zwar eine Teilmenge von M_1.

Übung 1.2.2. Nennen Sie Beispiele von Relationen!

$$\hookrightarrow \quad \leq, =, \geq, <, >$$

Schreibweisen

- Menge: $\{a, b, \dots\}$
  Die Reihenfolge der Elemente ist egal.
  Identisch doppelte Elemente sind überflüssig.

- n-Tupel: $(x_1, \dots, x_n)$
  Die Reihenfolge der Elemente ist wichtig.
  Elemente können mehrfach vorkommen.

  Spezialfälle:

Paar:	(x, y)
Tripel:	(x, y, z)
Quadrupel:	(x_1, x_2, x_3, x_4)
Quintupel:	$(x_1, x_2, x_3, x_4, x_5)$

Definition 1.2.3 (Verkettung von Relationen)**.**

Komposition (Verkettung): $R_1 \circ R_2 = \{(x, y) \mid \exists z : (xR_1z \text{ und } zR_2y)\}$

n-Schritt-Relation: $R^0 = \{(x, x) \mid x \in M\}$

$R^{n+1} = R \circ R^n \qquad (n \geq 0)$

Abschlüsse: $R^+ = \bigcup_{n \geq 1} R^n = R^1 \cup R^2 \cup \dots$ (transitive Hülle)

$R^* = \bigcup_{n \geq 0} R^n$ (reflexiv-transitive Hülle)

Umkehrung: $R^{-1} = \{(y, x) \mid xRy\}$

Erläuterung: R^* heißt *reflexiv-transitive Hülle* von R, weil es die kleinste reflexive und transitive Relation (Def. s. u.) ist, die R beinhaltet. — Das Symbol $\exists$ heißt Existenzquantor, sprich „es gibt“.

Übung 1.2.4. Wir betrachten die Relation $R = \{(x, y) \mid y = x + 1,\ x, y \in \mathbb{Z}\}$.

(a) Zählen Sie einige Elemente von R auf!

(b) Was ist $R \circ R$, R^0, R^1, R^2, R^+ und R^*?

Hinweis: Sie können die binäre Relation R grafisch durch ein Pfeildiagramm darstellen. Die Relationen R^n lassen sich daraus ablesen, indem jeweils Pfade der Länge n betrachtet werden.

Übung 1.2.5. Wir betrachten die Mengen L der Teilnehmer der Lehrveranstaltung und B aller Brillenträger in der Grundmenge M aller Menschen sowie die Relation $R = \{(x, y) \mid x \text{ sitzt unmittelbar links, rechts, vor oder hinter } y\} \subseteq M \times M$. Bestimmen Sie $L \cap B$, $L \cup B$, $L \cap \overline{B}$, R^0, R^2 sowie R^*!

Mengen vs. Zahlen

Relationen	Zahlen
Komposition	Multiplikation
$A \circ B$	$a \cdot b$
$R = R^1$	$r = r^1$
$R \circ R = R^2$	$r \cdot r = r^2$
$\underbrace{R \circ \cdots \circ R}_{n\text{-fach}} = R^n$	$\underbrace{r \cdot \cdots \cdot r}_{n\text{-fach}} = r^n$
$\underbrace{(R_1 \circ R_2) \circ R_3 = R_1 \circ (R_2 \circ R_3)}_{\text{Assoziativgesetz}}$	$\underbrace{(r_1 \cdot r_2) \cdot r_3 = r_1 \cdot (r_2 \cdot r_3)}_{\text{Assoziativgesetz}}$
$R_1 \circ R_2 \neq R_2 \circ R_1$	$\underbrace{r_1 \cdot r_2 = r_2 \cdot r_1}_{\text{Kommutativgesetz}}$

Das Kommutativgesetz gilt i. A. nicht für die Komposition von Relationen. Vergleiche dazu $R_1 \circ R_2$ und $R_2 \circ R_1$ z. B. für $R_1 = \{(1, 2)\}$ und $R_2 = \{(2, 3)\}$.

Eigenschaften von Relationen

reflexiv:	$\forall x \in M : xRx$
irreflexiv:	$\forall x \in M : \neg(xRx)$
transitiv:	$\forall x, y, z \in M : xRy \wedge yRz \Longrightarrow xRz$
symmetrisch:	$\forall x, y \in M : xRy \Longleftrightarrow yRx$
antisymmetrisch:	$\forall x, y \in M : xRy \wedge yRx \Longrightarrow x = y$
asymmetrisch:	$\forall x, y \in M : xRy \Longrightarrow \neg(yRx)$
total:	$\forall x, y \in M : xRy \vee yRx$

Im Vorgriff auf die Schreibweisen in der Logik verwenden wir hier folgende Symbole:

- $\forall$ „für alle“ (Allquantor)
- $\wedge$ „und“
- $\vee$ „oder“ (nicht ausschließend)
- $\neg$ „nicht“
- $\Longrightarrow$ „impliziert“ (Wenn-Dann-Aussage)
- $\Longleftrightarrow$ „gdw.“ (genau dann, wenn)

Spezielle Relationen

Bestimmte Kombinationen der o. g. Eigenschaften führen zu besonderen Relationstypen:

- Äquivalenzrelation: reflexiv, transitiv, symmetrisch
- partielle, reflexive oder Halbordnung: reflexiv, transitiv, antisymmetrisch
- strikte oder irreflexive Ordnung: transitiv, irreflexiv
- Quasiordnung: reflexiv, transitiv
- Ordnungsrelation: Oberbegriff für reflexive und irreflexive Ordnung (total oder nicht)

Satz 1.2.6. Eine Relation ist eine strikte Ordnung gdw. sie transitiv und asymmetrisch ist.

Beispiel 1.2.7.

$(\mathbb{N}, \leq)$	totale (reflexive) Ordnung (alle Elemente sind miteinander vergleichbar)
$(\mathbb{R}, >)$	strikte Ordnung
(D'land, $\blacktriangleright$)	Äquivalenzrelation ($\blacktriangleright$ ist hier Erreichbarkeitsrelation)
$(2^{\mathbb{N}}, \subseteq)$	nicht-totale reflexive Ordnung (da z. B. $\{1\} \not\subseteq \{2\}$ und $\{2\} \not\subseteq \{1\}$)
$(\mathbb{N}, =)$	Äquivalenzrelation, reflexive Ordnung
$(\mathbb{R}^3, \ll)$	Vektoren nach Länge vergleichen gemäß folgender Definition:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \ll \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} \text{ gdw. } \left\| \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \right\| \leq \left\| \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} \right\| \text{ gdw. } \sqrt{x_1^2 + x_2^2 + x_3^2} \leq \sqrt{y_1^2 + y_2^2 + y_3^2}$$

Beweis und Gegenbeispiel

Allgemeine Aussagen sind durch einen allgemeinen Beweis (*deduktiv*) zu zeigen. Zur Widerlegung einer allgemeinen Aussage reicht bereits ein einziges Gegenbeispiel. So ist z. B. die Aussage

Alle Studenten bestehen die Logik-Klausur.

bereits durch einen Studenten widerlegt, der die Klausur nicht besteht. Das Gegenteil der obigen allgemeinen Aussage (d. h. deren *Negation*) ist also eine Existenzaussage:

Es gibt (mindestens) einen Studenten, der die Klausur nicht besteht.

Letzteres ist unbedingt zu unterscheiden von und echt verschieden zu:

Kein Student besteht die Logik-Klausur.

Definition 1.2.8. Sei $\lesssim$ eine Quasiordnung in einer Menge M. Dann lassen sich darauf aufbauend u. a. die folgenden Relationen definieren:

$$\begin{aligned}
\gtrsim &:= \{(x,y) \mid y \lesssim x\} &&= (\lesssim)^{-1} && \text{(Quasiordnung)}\\
\approx &:= \{(x,y) \mid x \lesssim y \wedge y \lesssim x\} &&= (\lesssim) \cap (\gtrsim) && \text{(Äquivalenzrelation)}\\
< &:= \{(x,y) \mid x \lesssim y \wedge x \not\approx y\} &&= (\lesssim) \setminus (\gtrsim) && \text{(irreflexive Ordnung)}\\
\leq &:= \{(x,y) \mid x < y \vee x = y\} &&= (<) \cup \{(x,x) \mid x \in M\} && \text{(reflexive Ordnung)}\\
> &:= \{(x,y) \mid y < x\} &&= (<)^{-1} && \text{(irreflexive Ordnung)}\\
\geq &:= \{(x,y) \mid x > y \vee x = y\} &&= (>) \cup \{(x,x) \mid x \in M\} && \text{(reflexive Ordnung)}
\end{aligned}$$

Definition 1.2.9 (Äquivalenzklasse)**.** Sei R eine Äquivalenzrelation auf der Menge M. Dann nennt man für ein $x \in M$ die Teilmenge $[x] = \{y \in M \mid xRy\} \subseteq M$ die *R-Äquivalenzklasse* von x in M. Eine andere Schreibweise für $[x]$ ist x/R.

Definition 1.2.10 (Quotientenmenge, kanonische Halbordnung)**.** Die Menge aller möglichen Äquivalenzklassen $M/R = \{[x] \mid x \in M\}$ nennt man *Quotientenmenge* von R auf M. Auf der Quotientenmenge $M/\approx$ ist die *kanonische Halbordnung* $\leq$ zur Quasiordnung $\lesssim$ durch die Festlegung $[x] \leq [y]$ gdw. $x \lesssim y$ definiert.

Satz 1.2.11. $(M/\approx, \leq)$ ist stets eine reflexive Ordnung, d. h. $\leq$ ist reflexiv, transitiv und antisymmetrisch auf der Quotientenmenge $M/\approx$.

1.3 Funktionen

Eine Relation $R \subseteq M \times N$ heißt

- linkstotal gdw. $\forall x \in M\, \exists y \in N : xRy$ bzw.
- rechtseindeutig gdw. $\forall x, y_1, y_2 \in M : xRy_1 \wedge xRy_2 \Longrightarrow y_1 = y_2$.

Analog: rechtstotal, linkseindeutig.

Definition 1.3.1 (Funktion, Abbildung)**.** Eine rechtseindeutige Relation $f \subseteq M \times N$ heißt *(partielle) Funktion*, i. Z. $f : M \to N$. Die Menge M nennt man *Definitionsbereich*, N *Werte- oder Zielbereich* der Funktion f. Statt xfy schreiben wir $f(x) = y$. Ist f zusätzlich linkstotal, spricht man von einer *totalen Funktion* oder auch *Abbildung*. Die Zuordnungsvorschrift einer Funktion wird in der Form $x \mapsto f(x)$ dargestellt.

Übung 1.3.2. Sind die Funktionen $f : \mathbb{R} \to \mathbb{R}$ mit (a) $x \mapsto \frac{1}{x}$ bzw. (b) $x \mapsto \sin(x)$ (echt) partiell oder total?

Beobachtungen

- Bei einer (partiellen) Funktion ist jedem $x \in M$ höchstens ein $y \in N$ zugeordnet.
- Bei einer totalen Funktion ist jedem $x \in M$ (wegen der Rechtseindeutigkeit) genau ein $y \in N$ zugeordnet.
- Jede totale Funktion ist auch eine partielle Funktion (die im engen Wortsinne natürlich nicht echt partiell ist). Die Umkehrung gilt i. A. nicht.

Was ist eine Umkehrung?

Unter der Umkehrung einer Aussage der Form $A \Longrightarrow B$ versteht man die Behauptung $B \Longrightarrow A$, wo also Vor- und Nachbedingungen vertauscht, d. h. in umgekehrter Reihenfolge sind. Eine Aussage und ihre Umkehrung sind i. A. nicht gleichbedeutend, d. h. weder folgt aus der ursprünglichen Aussage deren Umkehrung noch umgekehrt.

Übung 1.3.3. Ist die Relation $R = \{(x^2, x) \mid x \in \mathbb{R}\}$ eine Funktion?

Beispiel 1.3.4.

Exponentionalfunktion:	$\exp : \mathbb{R} \to \mathbb{R}$	mit	$x \mapsto e^x$
Wurzelfunktion:	$\mathrm{sqrt} : \mathbb{R} \to \mathbb{R}$	mit	$x \mapsto \sqrt{x}$

Definition 1.3.5 (für totale Funktionen $f : M \to N$)**.**

injektiv:	$\forall x_1, x_2 \in M : f(x_1) = f(x_2) \Longrightarrow x_1 = x_2$	(d. h., f ist linkseindeutige Relation)
	anders ausgedrückt: $\forall x_1, x_2 \in M : x_1 \neq x_2 \Longrightarrow f(x_1) \neq f(x_2)$	(Kontraposition)
surjektiv:	$\forall y \in N \; \exists x \in M : y = f(x)$	(d. h., f ist rechtstotale Relation)
bijektiv:	injektiv + surjektiv (Eineindeutigkeit)	

Bijektive Funktionen sind umkehrbar, d. h. zu jeder bijektiven Funktion f gibt es eine *Umkehrfunktion* $\overline{f} : N \to M$ mit $\overline{f}(f(x)) = x$ für alle $x \in M$ bzw. $f(\overline{f}(y)) = y$ für alle $y \in N$.

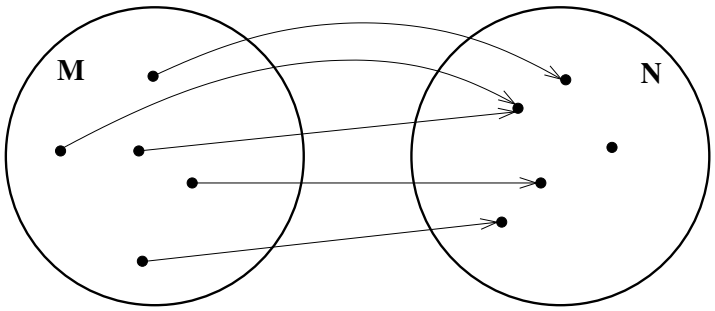
Was ist Kontraposition?

Eine Wenn-Dann-Aussage $A \Longrightarrow B$ ist gleichbedeutend (äquivalent) mit ihrer Kontraposition $\neg B \Longrightarrow \neg A$. Wenn man also die Vor- und Nachbedingungen A und B einer Aussage miteinander vertauscht und beide negiert, ändert sich die Gültigkeit der Aussage nicht. Dies wird häufig in Beweisen ausgenutzt, indem statt der ursprünglichen Behauptung deren Kontraposition gezeigt wird.

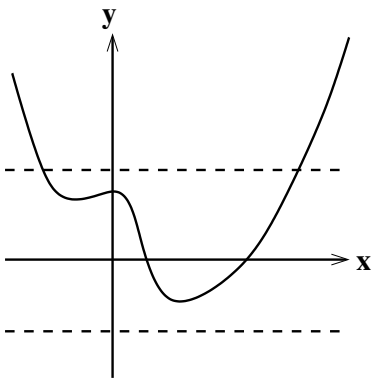
Satz 1.3.6. Für (nicht-leere) endliche Mengen M und N mit gleicher Kardinalität (also gleicher Anzahl Elemente, i. Z. $|M| = |N| < \infty$) gilt: Eine Abbildung $f : M \to N$ ist injektiv gdw. sie surjektiv ist.

Definition 1.3.7. Eine Ordnungsrelation $<$ auf der Menge M heißt *dicht* gdw. es für alle $x, y \in M$ mit $x < y$ ein $z \in M$ mit $x < z$ und $z < y$ gibt.

Bemerkung 1.3.8. Funktionen $f : M \to N$ lassen sich für endliche oder zumindest nicht dichte Mengen M und N wie binäre Relationen durch Pfeildiagramme darstellen (vgl. Übung 1.2.4). Injektivität bzw. Surjektivität ist dann daran erkennbar, dass stets höchstens bzw. mindestens eine Kante (d.h. ein Pfeil) zu den Elementen in N führt (siehe nachfolgende Skizze).



Für dichte Mengen M und N lassen sich Funktionen $f : M \to N$ als Funktionsgraphen in einem Koordinatensystem darstellen. Injektivität bzw. Surjektivität ist in diesem Fall daran erkennbar, dass jede zur x-Achse parallele Gerade höchstens bzw. mindestens einen Schnittpunkt mit dem Funktionsgraphen hat (siehe nachfolgende Skizze).



Beispiel 1.3.9.

$f : \mathbb{R} \setminus \{0\} \to \mathbb{R} \setminus \{0\}$ mit $x \mapsto \frac{1}{x}$	bijektiv
$f : \mathbb{N} \to \mathbb{N}$ mit $x \mapsto x^2$	injektiv, nicht surjektiv, da z. B. $3 \neq f(x)\ \forall x \in \mathbb{N}$
$f : \mathbb{R} \to \mathbb{R}_0^+$ mit $x \mapsto x^2$	surjektiv, nicht injektiv, da z. B. $2^2 = (-2)^2$

Definition 1.3.10 (Gleichheit von Funktionen, Identitätsfunktion)**.**

- Zwei Funktionen f_1 und f_2 heißen *gleich* gdw. sie denselben Definitionsbereich haben und $f_1(x) = f_2(x)$ für alle x aus dem Definitionsbereich gilt.
- Die Funktion f, die alle Elemente auf sich selbst abbildet, d. h. $f(x) = x$ für alle x, nennt man *Identitätsfunktion*.

1.4 Halbgruppen

Definition 1.4.1 (algebraische Struktur)**.** Mengen M, in denen eine (zweistellige) Verknüpfung zwischen zwei Elementen (durch eine Funktion $f : M \times M \to M$) definiert ist, nennt man *algebraische Struktur*.

In der Literatur wird ein ganzer Zoo von algebraischen Strukturen und deren Eigenschaften betrachtet, z. B. Gruppen (siehe Def. 1.5.1), Verbände, Ringe und Boolesche Algebren (siehe Def. 2.2.10). Im Folgenden folgt nur ein kurzer Einblick in das Gebiet. Für weiterführende Informationen sei auf die zahlreiche Literatur verwiesen, z. B. [Lau11].

Definition 1.4.2 (Verknüpfung, Abgeschlossenheit)**.** Seien M_1, M_2 nicht-leere Mengen. Eine *Verknüpfung* $\circ : M_1 \times M_1 \to M_2$ ist eine Abbildung, die jedem Paar $(a, b) \in M_1 \times M_1$ genau ein Element $a \circ b \in M_2$ zuordnet. $\circ$ muss also eine *abgeschlossene*, d. h. totale (also überall in $M_1 \times M_1$ definierte) Funktion sein. Falls $M_1 = M_2 = M$ ist, sprechen wir von einer *Verknüpfung in M* und schreiben $(M, \circ)$. $(M, \circ)$ ist eine algebraische Struktur (mit bestimmten Eigenschaften).

Definition 1.4.3 (Halbgruppe)**.** $(M, \circ)$ heißt *Halbgruppe* gdw. die Verknüpfung $\circ$ *assoziativ* ist, d. h. $\forall a, b, c \in M : (a \circ b) \circ c = a \circ (b \circ c)$. Falls zusätzlich $a \circ b = b \circ a$ gilt, heißt die Halbgruppe *kommutativ* oder auch *abelsch*.

Beispiel 1.4.4.

(a) $(\mathbb{N}_0, +)$: kommutative Halbgruppe

(b) $(\mathcal{F}, \circ)$: nicht-kommutative Halbgruppe

- $\mathcal{F}$ ist hierbei die Menge der Funktionen $f : \mathbb{N}_0 \to \mathbb{N}_0$ und $\circ$ die Komposition von Funktionen, definiert wie folgt: $(f_1 \circ f_2)(x) = f_2(f_1(x))$ (vgl. Def. 1.2.3)
- $(\mathcal{F}, \circ)$ ist nicht kommutativ, d. h. i. A. gilt $f_1 \circ f_2 \neq f_2 \circ f_1$.
  Betrachte dazu das Beispiel $f_1 : n \mapsto n + 1$ und $f_2 : n \mapsto n^2$.

Definition 1.4.5 (Einselement, neutrales Element, Monoid)**.** Sei $(H, \circ)$ eine Halbgruppe. Dann heißt $e \in H$ mit $e \circ a = a \circ e = a$ für alle $a \in H$ *Einselement* oder (besser) auch *neutrales Element*. Eine Halbgruppe $(H, \circ)$ heißt *Monoid*, wenn sie ein neutrales Element enthält.

Übung 1.4.6. Welche der algebraischen Strukturen $(\mathbb{R}, \cdot)$ und $(\mathbb{N}_0, +)$ sind Monoide? Was sind die Einselemente e der Monoide?

Satz 1.4.7. In einem Monoid ist das Einselement eindeutig bestimmt.

Beweis (indirekt)**:** Angenommen, es gibt zwei Einselemente e_1 und e_2 mit $e_1 \neq e_2$. Dann gilt $e_1 = e_1 \circ e_2 = e_2$ (aufgrund der Eigenschaft von Einselementen). Das ist aber ein Widerspruch zur Annahme. Also gilt $e_1 = e_2$. ⊠

Was ist ein indirekter Beweis?

Bei einem indirekten Beweis zeigt man, dass das Gegenteil der zu beweisenden Behauptung nicht zutreffen kann, und folgert daraus, dass deshalb (mangels weiterer Alternativen) die ursprüngliche Behauptung wahr sein muss. Dabei ist eine zweiwertige Logik angenommen: Aussagen sind (nur) entweder wahr oder falsch.

Für eine Wenn-Dann-Aussage der Form $A \Longrightarrow B$ bedeutet dies, dass, ausgehend von der Annahme, dass die Aussage B nicht gilt, jedoch die Voraussetzung(en) A (manchmal nur implizit gegeben), ein logischer Widerspruch herzuleiten ist.

Definition 1.4.8 (Erzeugendensystem E)**.** Sei $(H, \circ)$ eine Halbgruppe. $E \subseteq H$ heißt *Erzeugendensystem* (Basis) von $A \subseteq H$ gdw. $\forall a \in A\ \exists b_1, \ldots, b_k \in E : a = b_1 \circ \ldots \circ b_k$. Falls $|E| = 1$ und $A = H$, heißt $(H, \circ)$ *zyklische* Halbgruppe.

Beispiel 1.4.9. $(\mathbb{N}, +)$ ist zyklisch mit $b = 1$.

1.5 Gruppen

Definition 1.5.1 (Gruppe)**.** $(M, \circ)$ heißt *Gruppe* gdw. gilt:

1. Abgeschlossenheit: $\forall x, y \in M : x \circ y \in M$
2. Assoziativität: $\forall x, y, z \in M : (x \circ y) \circ z = x \circ (y \circ z)$
3. Rechtsneutrales Element: $\exists e \in M \, \forall x \in M : x \circ e = x$
4. Rechtsinverses Element: $\forall x \in M \, \exists y \in M : x \circ y = e$

Falls $\circ$ zusätzlich kommutativ ist, spricht man auch von einer kommutativen oder abelschen Gruppe. Rechtsneutrale Elemente nennt man auch hier Rechtseins.

Übung 1.5.2. Untersuchen Sie, ob

(a) $(\mathbb{N}, +)$,

(b) $(\mathbb{Z}, +)$,

(c) $(\mathbb{R}, \cdot)$ bzw.

(d) $(\mathbb{R}^+, \cdot)$

Gruppen sind!

Satz 1.5.3. Sei $(M, \circ)$ eine Gruppe (mit rechtsneutralem Element e). Dann gilt:

(a) Rechtsinverse Elemente sind gleichzeitig linksinvers.

(b) Rechtsneutrale Elemente sind gleichzeitig linksneutral.

(c) Es gibt genau ein gleichzeitig rechts- und linksneutrales Element e.

(d) Jedes Element x besitzt genau ein eindeutig bestimmtes inverses Element $\overline{x}$.

(e) Doppelt inverse Elemente $\overline{\overline{x}}$ sind mit dem ursprünglichen Wert x identisch.

Beweis: Wir beweisen die einzelnen Behauptungen nacheinander größtenteils direkt durch Gleichungsketten und wenden die vier Eigenschaften einer Gruppe aus Def. 1.5.1 sowie die bereits jeweils vorher bewiesenen Behauptungen an.

(a) Sei $\overline{x}$ rechtsinvers zu einem festen, aber beliebig gewählten Element $x \in M$ und $\overline{\overline{x}}$ rechtsinvers zu $\overline{x}$, d. h. $x \circ \overline{x} = e$ und $\overline{x} \circ \overline{\overline{x}} = e$. Wir zeigen nun die Behauptung $\overline{x} \circ x = e$ durch eine Kette von Äquivalenzumformungen:

$$\overline{x} \circ x \overset{3.}{=} (\overline{x} \circ x) \circ e \overset{4.}{=} (\overline{x} \circ x) \circ (\overline{x} \circ \overline{\overline{x}}) \overset{2.}{=} ((\overline{x} \circ x) \circ \overline{x}) \circ \overline{\overline{x}} \overset{2.}{=} (\overline{x} \circ (x \circ \overline{x})) \circ \overline{\overline{x}} \overset{4.}{=} (\overline{x} \circ e) \circ \overline{\overline{x}} \overset{3.}{=} \overline{x} \circ \overline{\overline{x}} \overset{4.}{=} e$$

(b) Zu zeigen ist $e \circ x = x$ für ein beliebiges Element $x \in M$. Im Folgenden bezeichnet $\overline{x}$ wiederum ein inverses Element zu x, d. h. $x \circ \overline{x} = e$:

$$e \circ x \overset{4.}{=} (x \circ \overline{x}) \circ x \overset{2.}{=} x \circ (\overline{x} \circ x) \overset{(a)}{=} x \circ e \overset{3.}{=} x$$

(c) Aufgrund von Eigenschaft 3. aus Def. 1.5.1 gibt es (mindestens) ein rechtsneutrales Element e. Wir zeigen, dass die Annahme, dass es mehr als eins, d. h. mindestens zwei verschiedene neutrale Elemente $e_1, e_2 \in M$ mit $e_1 \neq e_2$ gibt, zu einem Widerspruch führt, nämlich $e_1 = e_2$, wie folgt (vgl. Beweis zu Satz 1.4.7):

$$e_1 \overset{\text{(b)}}{=} e_2 \circ e_1 \overset{3.}{=} e_2$$

Daraus folgt insgesamt die Eindeutigkeit des neutralen Elements e.

(d) Die Annahme, dass es zu einem $x \in M$ zwei inverse Elemente $\overline{x}_1$ und $\overline{x}_2$ mit $\overline{x}_1 \neq \overline{x}_2$ gibt, führt zu einem Widerspruch:

$$\overline{x}_1 \overset{3.}{=} \overline{x}_1 \circ e \overset{4.}{=} \overline{x}_1 \circ (x \circ \overline{x}_2) \overset{2.}{=} (\overline{x}_1 \circ x) \circ \overline{x}_2 \overset{\text{(a)}}{=} e \circ \overline{x}_2 \overset{\text{(b)}}{=} \overline{x}_2$$

Also muss $\overline{x}_1 = \overline{x}_2$ gelten, d. h. das inverse Element zu x ist eindeutig bestimmt.

(e) Wir verwenden die Variablensymbole $\overline{x}$ und $\overline{\overline{x}}$ wie in Teil (a) und zeigen $\overline{\overline{x}} = x$:

$$\overline{\overline{x}} \overset{\text{(b)}}{=} e \circ \overline{\overline{x}} \overset{4.}{=} (x \circ \overline{x}) \circ \overline{\overline{x}} \overset{2.}{=} x \circ (\overline{x} \circ \overline{\overline{x}}) \overset{4.}{=} x \circ e \overset{3.}{=} x$$ ⊠

Was ist ein direkter Beweis?

Bei einem direkten Beweis wird die Behauptung durch Anwendung von bereits bewiesenen Aussagen, den gegebenen Voraussetzungen und Definitionen, und durch logische Folgerungen bewiesen. Für eine Behauptung der Form $A \Longrightarrow B$ bedeutet dies, dass der Beweis in der Regel über mehrere Zwischenschritte erfolgt als Kette von Schlussfolgerungen $A \Longrightarrow \ldots \Longrightarrow B$. Im Falle von Äquivalenzen oder Gleichungsketten lassen sich direkte Beweise sogar vorwärts und rückwärts lesen.

1.6 Wohlfundierte Ordnungen und Induktion

Grenzelemente

Sei $\leq$ eine (reflexive) Ordnungsrelation in einer Grundmenge M, $< := (\leq) \setminus \{(x, x) \mid x \in M\}$ die daraus abgeleitete irreflexive Ordnungsrelation und N eine nicht-leere Teilmenge von M. Ein Element m, welches in der Teilmenge N enthalten sein muss, heißt

- *Maximum* von N gdw. $x \leq m$ für alle $x \in N$, i. Z. $\max(N)$ bzw.
- *maximales Element* von N gdw. es gibt kein $x \in N$ mit $x > m$.

Ein Element s, welches in der Grundmenge M enthalten sein darf, heißt

- *obere Schranke* von N gdw. $x \leq s$ für alle $x \in N$ bzw.
- *Supremum* von N gdw. s ist die (eindeutig bestimmte) kleinste obere Schranke, i. Z. $\sup(N)$.

Die Begriffe *Minimum*, *minimales Element*, *untere Schranke* und *Infimum* (= größte untere Schranke) ergeben sich analog.

Übung 1.6.1. Bestimmen Sie Maxima, maximale Elemente, obere Schranken und Suprema der folgenden Mengen N bzgl. der angegebenen Ordnungsrelation:

(a) $M = \mathbb{N}_0$ und (i) $N = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ bzw. (ii) $N = \mathbb{N}_0$ bzgl. $\leq$

(b) $M = \mathbb{R}$ und (i) $N = \{x \mid x \leq 1\}$ bzw. (ii) $N = \{x \mid x < 1\}$ bzgl. $\leq$

(c) $M = \mathcal{P}(\{0, 1\})$ und $N = M \setminus \{\{0, 1\}\}$ bzgl. $\subseteq$ (Teilmengenrelation)

Bemerkung 1.6.2. Das Minimum einer Menge ist eindeutig bestimmt, falls existent, und dann das einzige minimale Element und gleichzeitig Infimum der Menge. Analoges gilt für Maximum, maximale Elemente und Supremum. Es kann i.A. null, eins oder mehr minimale oder maximale Elemente geben.

Definition 1.6.3 (wohlfundierte Ordnung)**.** Sei M eine mittels $\leq$ (partiell) geordnete Menge. Falls jede nicht-leere Teilmenge $N \subseteq M$ mindestens ein minimales Element enthält, heißt $\leq$ *wohlfundiert*.

Definition 1.6.4 (Wohlordnung)**.** Falls jede nicht-leere Teilmenge $N \subseteq M$ ein Minimum hat (eindeutig), heißt $\leq$ *wohlgeordnet*.

Bemerkung 1.6.5.

- Die gerade für eine reflexive Ordnung $\leq$ definierten Begriffe *wohlfundiert* und *wohlgeordnet* wendet man sinngemäß auch auf eine (daraus abgeleitete) irreflexive Ordnung $<$ mit $(<) = (\leq) \setminus (=)$ an.
- Mengen können ein Minimum besitzen, ohne dass die betrachtete Ordnungsrelation wohlfundiert sein muss.
- Wohlfundierte Ordnungen (gemäß Def. 1.6.3) sind (reflexive bzw. irreflexive) Ordnungsrelationen $\leq$ bzw. $<$, die (überhaupt) keine unendlich absteigenden Ketten der Form
$$m_0 > m_1 > \ldots > m_i > \ldots$$
enthalten. Hierbei ist die Relation $>$ wie in Def. 1.2.8 aus der Relation $<$ abgeleitet. Die m_i sind klarerweise für alle $i \in \mathbb{N}_0$ verschieden.
- Die Untersuchung der Wohlfundiertheit kann insofern auch als Spiel durchgeführt werden: Es werden ausgehend von einem Wert der Menge nacheinander immer kleiner werdende Elemente genannt. Man muss versuchen, möglichst lange (unendlich) im Spiel zu bleiben.
- Wohlordnungen sind wohlfundierte Ordnungen, die außerdem total sind.

Übung 1.6.6. Untersuchen Sie, ob die folgenden reflexiven Ordnungen total, wohlfundiert und/oder wohlgeordnet sind!

(a) $(\mathbb{N}_0, \leq)$ (b) $(\mathbb{R}_0^+, \leq)$ (c) $(\mathcal{P}(\{0, 1\}), \subseteq)$

Axiom 1.6.7 (Wohlordnungssatz)**.** Jede Menge lässt sich wohlordnen. Bei überabzählbaren Mengen wie den reellen Zahlen ist dies nicht mehr konstruktiv möglich.

Definition 1.6.8 (abzählbar)**.** Eine Menge M heißt abzählbar, falls es eine surjektive Funktion $f : \mathbb{N}_0 \to M$ gibt, d. h. M ist darstellbar als $M = \{f(0), f(1), f(2), \ldots\}$. Dabei braucht die Funktion f weder total noch berechenbar zu sein. Elemente aus M werden ggf. mehrfach aufgezählt. Nicht abzählbare Mengen heißen *überabzählbar*.

Die leere Menge $\emptyset$ lässt sich durch die überall undefinierte Funktion namens Ω abzählen bzw. gilt per Definition als abzählbar. Falls f eine totale Funktion ist, handelt es sich bei M um eine nicht-leere abzählbare Menge. Falls f eine (totale) bijektive Funktion ist, heißt M *abzählbar unendlich*.

Bemerkung 1.6.9. Die Menge der nicht-negativen Bruchzahlen $\mathbb{Q}_0^+$ ist bezüglich der normalen $\leq$–Relation nicht wohlfundiert. Die Elemente in $\mathbb{Q}_0^+$ lassen sich aber wie folgt zu einer (speziellen sogenannten lexikografischen) Wohlordnung $\preceq$ umordnen: Für zwei vollständig gekürzte Brüche a_1/a_2 und b_1/b_2 gilt $a_1/a_2 \preceq b_1/b_2$ gdw. entweder $a_1 < b_1$ oder $a_1 = b_1$ und $a_2 \leq b_2$.

Was ist der Unterschied zwischen Axiom, Lemma, Theorem und Korollar?

Ein *Axiom* ist ein Grundsatz einer Theorie, der innerhalb dieses Systems nicht begründet oder (deduktiv) hergeleitet wird. Ein Axiom ist also ein Satz, der nicht in der Theorie bewiesen werden soll, sondern beweislos vorausgesetzt wird. In der Regel geht man davon aus, dass die gewählten Axiome einer Theorie logisch unabhängig sind, d. h. keines von ihnen aus den anderen hergeleitet werden kann, und sie sich nicht gegenseitig widersprechen, d. h. insgesamt kein Widerspruch hergeleitet werden kann. In der Logik werden Axiome meist durch (prädikatenlogische) Formeln beschrieben.

Axiom wird als Gegenbegriff zu *Theorem* (im engeren Sinn) verwendet. Theoreme sind Sätze, die durch formale Beweisgänge aus den Axiomen hergeleitet werden können. Ein Hilfssatz oder *Lemma* ist eine mathematische oder logische Aussage, die im Beweis eines Satzes verwendet wird, der aber selber nicht der Rang eines Theorems oder Satzes eingeräumt wird. *Korollar* bezeichnet in der Mathematik und Logik eine Aussage, die sich aus einem schon bewiesenen Satz oder dem Beweis eines schon bewiesenen Satzes ohne großen (weiteren) Beweisaufwand ergibt. Die Unterscheidung von Lemmata, Sätzen bzw. Theoremen und Korollaren ist fließend.

Satz 1.6.10 (Noether'sche Induktion)**.** Sei P eine Eigenschaft eines Elements m einer mittels $\leq$ wohlfundiert geordneten Menge M. Dann gilt:

$$\Big(\forall j \in M : \Big(\forall i < j : P(i) \Longrightarrow P(j)\Big)\Big) \Longrightarrow \Big(\forall m \in M : P(m)\Big)$$

Wie funktioniert Induktion?

Um eine Behauptung $P(j)$ für alle $j \in M$ zu beweisen, erlaubt das Induktionsprinzip die folgende Vorgehensweise: Wir nehmen an, dass die Behauptung P bereits für *alle* „kleineren" Elemente i mit $i < j$ gilt (*Induktionsannahme* oder *Induktionsvoraussetzung*, abgekürzt I.V.). Zu zeigen ist dann, dass P dann auch für j gilt (*Induktionsschritt*). Sonderfälle sind die minimalen Elemente in M, die keine kleineren Elemente haben (*Induktionsanfang*) sowie (selten) Elemente, die keine unmittelbaren Vorgänger haben (sogenannte *Limeselemente*) bei der transfiniten Induktion.

Was heißt Fallunterscheidung?

Eine Aussage der Form $A \Longrightarrow B$ kann in mehrere Fälle zerlegt werden, die ggf. leichter als die gesamte Behauptung zu beweisen sind. Dabei wird die Voraussetzung A in die Fälle $A_1, \ldots, A_n$ zerlegt. Die einzelnen Fälle müssen insgesamt alle möglichen Fälle abdecken, d. h. $A_1 \vee \cdots \vee A_n$ ist äquivalent zu A. Außerdem sind in der Regel alle Fälle disjunkt, d. h. A_i und A_j für $i \neq j$ gelten nicht gleichzeitig. Beispielsweise können Aussagen für alle natürlichen Zahlen $n \in \mathbb{N}_0$ zerlegt werden in die zwei Fälle $n = 0$ und $n > 0$.

Was ist vollständige Induktion?

Induktionsbeweise erweitern das Konzept der Fallunterscheidung um das Induktionsprinzip. Der Beweis durch vollständige Induktion ist ein oft angewendetes Verfahren zum Beweis von Sätzen der Form *Für jede natürliche Zahl n gilt ...* Dazu zeigt man zuerst, dass die Aussage für $n = 0$ (oder auch einen anderen Anfangswert) gilt (Induktionsanfang), und dann (im sogenannten Induktionsschritt), dass sie immer auch für $n+1$ gilt (Induktionsbehauptung), wenn sie für n gilt (Induktionsannahme). Die vollständige Induktion lässt sich mit einem Domino-Effekt vergleichen. Man stellt die Steine so auf, dass, wenn einer umfällt, auch immer der nächste umfällt ($n \mapsto n+1$), und stößt den ersten Stein um ($n = 0$).

Arten der Induktion

vollständige Induktion: zweiteiliger Beweis, $M = \mathbb{N}_0$

(a) $n = 0$ (Induktionsanfang)

(b) $n \mapsto n + 1$ (Induktionsschritt)

strukturelle Induktion: über den Aufbau einer Datenstruktur, z. B. Bäume oder Terme

- Beweise gehen „längs“ der Ordnung („ist Teilstruktur von“)
- Beispiel: Zerlegung von Termausdrücken

transfinite Induktion: dreiteiliger Beweis, $(M, \leq)$ ist Wohlordnung

(a) $m = \min(M)$ (Induktionsanfang)

(b) $m = n + 1$ für $n \in M$ (Induktionsschritt)

(c) $m = \sup(N)$ mit $\sup(N) \notin N$ (transfiniter Schritt)

Übung 1.6.11. Die Summe der ersten n ungeraden Zahlen ist eine Quadratzahl, d. h.:

$$\forall n \in \mathbb{N} : \sum_{k=1}^{n} (2k - 1) = n^2$$

Beweisen Sie diese Behauptung durch vollständige Induktion für $n \in \mathbb{N}$!
Hinweis: Wenden Sie (wie in jedem echten Induktionsbeweis) im Induktionsschritt irgendwo (ggf. mehrfach) die Induktionsvoraussetzung, dass also die Behauptung bis zu einem festen, aber beliebigen n bereits gilt, an!

1.7 Ordinalzahlen – Mehr als Unendlich

Mit Hilfe von sogenannten Ordinalzahlen lässt sich im Unendlichen rechnen. Ordinalzahlen werden über Wohlordnungen definiert. Sie erweitern die Menge der natürlichen Zahlen. Der Begriff von Zahlen wird so weiter abstrahiert [Fel79].

Definition 1.7.1. Homomorphismen sind Abbildungen $h : M \to N$ mit gewissen Nebenbedingungen. Seien $(M, \circ_M)$ und $(N, \circ_N)$ Gruppen. Dann heißt h

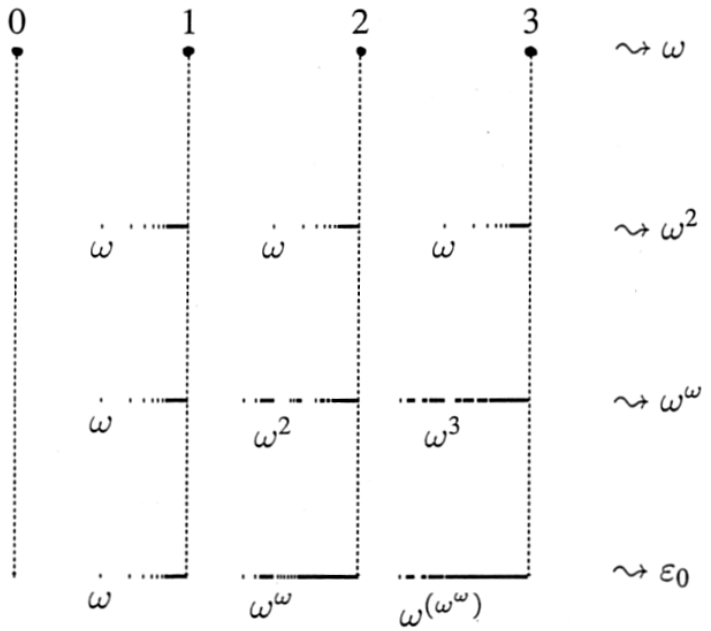
- Gruppenhomomorphismus gdw. $\forall x, y \in M : h(x \circ_M y) = h(x) \circ_N h(y)$ bzw.
- Isomorphismus gdw. h ein bijektiver Homomorphismus ist.

Beispiel 1.7.2 (Konstruktionen im Unendlichen)**.**
Gegeben:

- $\mathbb{Q}_0^+$: Menge der nicht-negativen rationalen Zahlen
- $S : \mathbb{N}_0 \to [0, 1)$ mit $x \mapsto \frac{x}{x+1}$ (Ordnungsisomorphismus)
- $T_n : [0, 1) \to [n, n + 1)$ mit $x \mapsto x + n$ (Translation)

Betrachte (vgl. Abbildung):

- $Q(\omega)$: Teilmenge der natürlichen Zahlen in $\mathbb{Q}_0^+$
- $Q(\omega^2)$: Einpassen von $Q(\omega)$ in die Intervalle von $Q(\omega)$
- $Q(\omega^\omega)$: Einpassen von $Q(\omega^n)$ in die Intervalle von $Q(\omega)$
- $Q(\varepsilon_0)$: Analoges Vorgehen für alle ω_n $[\omega_0 = 1; \omega_{n+1} = \omega^{\omega_n}]$



Wohlordnungen (Begriffe)

1. Relation R: Klasse, deren Elemente geordnete Paare (x, y) sind
2. Feld $\mathrm{fd}(R)$: Klasse aller x mit $\exists y : \ (x, y) \in R \vee (y, x) \in R$
3. Vorbereich $R^{-1}(y)$: Klasse aller x mit $(x, y) \in R$
4. R lokal: jede Klasse $R^{-1}(y)$ ist eine Menge
5. Wohlordnung $(R, \leq)$: Ordnung die lokal ist und jedes nicht-leere $X \subseteq \mathrm{fd}(R)$ enthält ein kleinstes Element $\min(X)$
6. Limeselement z: nicht Minimum oder Nachfolger eines $x \in \mathrm{fd}(R)$
7. Homomorphismus f: $R \to R'$ gdw. $x \leq_R y \Longrightarrow f(x) \leq_{R'} f(y)$

Theorem 1.7.3 (Cantor'scher Vergleichbarkeitssatz)**.** Seien R und R' zwei Wohlordnungen.

1. Dann ist eine von beiden zu einem Anfang der anderen isomorph.
2. Der Isomorphismus ist eindeutig bestimmt.

Beweis: Wir beweisen Teil 1. durch Induktion längs R:

- Induktionsanfang:
  Für $x = \min(R)$ setze $f(\min(R)) = \min(R')$. f ist trivialerweise ein Isomorphismus.
- Induktionsannahme: Es gibt einen Isomorphismus f' von $(\leftarrow, x)$ auf einen Anfang von R' oder umgekehrt. Wir zeigen, dass f' auf x fortsetzbar ist.
- Induktionsschritt: Sei $M = R' \setminus f'((\leftarrow, x))$. Falls $M = \emptyset$, ist R' zum Anfang $(\leftarrow, x)$ von R isomorph. Setze andernfalls $f(x) = \min(M)$; $f(x') = f'(x')$ sonst. f ist nach wie vor ein Isomorphismus.

Teil 2. beweisen wir indirekt: Angenommen, es gibt zwei Isomorphismen f und g.
$\Longrightarrow \overline{f} \circ g$ und $\overline{g} \circ f$ Isomorphismen von R in sich [R' Wohlordnung]
$\Longrightarrow x \leq_R (\overline{f} \circ g)(x)$ und $x \leq_R (\overline{g} \circ f)(x)$ für alle $x \in \mathrm{fd}(R)$ [Lemma]
$\Longrightarrow f(x) \leq_{R'} g(x)$ und $g(x) \leq_{R'} f(x)$ [f und g Isomorphismen]
$\Longrightarrow f(x) = g(x)$, also $f \equiv g$ [Antisymmetrie von $\leq$] ⊠

Lemma 1.7.4 (nach Zermelo)**.** Sei R eine Wohlordnung und h ein Isomorphismus von R in sich. Dann gilt $x \leq h(x)$ für alle $x \in \mathrm{fd}(R)$.

Beweis: Andernfalls existiert $x \in \mathrm{fd}(R)$ mit $x \not\leq h(x)$
$\Longrightarrow$ es gibt minimales x mit $h(x) < x$ [R ist Wohlordnung]
$\Longrightarrow h(h(x)) < h(x)$ [Widerspruch zur Minimalität] ⊠

Definition 1.7.5 (Ordinalzahl)**.** Eine Menge n heißt *Ordinalzahl*, wenn die folgenden Axiome gelten:

1. Transitivität: $x \in n \Longrightarrow x \subseteq n$

2. Konnexität: $x \in n \wedge y \in n \wedge x \neq y \Longrightarrow x \in y \vee y \in x$
3. Fundiertheit: $x \subseteq n \wedge x \neq 0 \Longrightarrow \exists y : y \in x \wedge y \cap x = 0$

Satz 1.7.6. Zu jeder wohlgeordneten Menge $(E, \leq)$ existiert genau eine Ordinalzahl n , so dass $(E, \leq)$ zu $(n, \subseteq)$ isomorph ist.

Satz 1.7.7. Jede Ordinalzahl $(n, \subseteq)$ ist eine Wohlordnung, deren Elemente selbst wieder Ordinalzahlen sind.

Feststellungen

1. ORD: Klasse aller Ordinalzahlen
2. Minimum in ORD: $0 = \emptyset$
3. Unmittelbarer Nachfolger: $n + 1 = n \cup \{n\}$
4. Für Mengen M von Ordinalzahlen gilt: $\sup(M) = \bigcup_{m \in M} m$
5. Wir können $<$ statt $\in$ schreiben

Bemerkung 1.7.8. ORD ist ebenso wie die Gesamtheit aller Isomorphietypen *echt* Klasse (also nicht als Menge definierbar).

Definition 1.7.9 (Arithmetik)**.** Im Folgenden sei z immer eine Limeszahl.

1. Addition:
   $m + 0 = m; \quad m + (x + 1) = (m + x) + 1; \quad m + z = \sup\{m + x \mid x \in z\}$
2. Multiplikation:
   $m \cdot 0 = 0; \quad m \cdot (x + 1) = (m \cdot x) + m; \quad m \cdot z = \sup\{m \cdot x \mid x \in z\}$
3. Exponentiation:
   $m^0 = 1; \quad m^{x+1} = (m^x) \cdot m; \quad m^z = \sup\{m^x \mid x \in z\}$

Achtung: Exponentiation bedeutet hier nicht das Bilden einer Potenzmenge. Außerdem sind $+$ und $\cdot$ zwar assoziativ, aber nicht kommutativ in den Ordinalzahlen. Es gelten weitere Gesetze (Monotonie, Stetigkeit usw.).

Definition 1.7.10 (g-adische Normalformen)**.**

1. Normalfunktion $N : \text{ORD} \to \text{ORD}$ mit $x \mapsto g^x \quad [1 < g \in \text{ORD}]$
2. N-Index: Paar (α, β) von Ordinalzahlen der Länge $k < \omega$ mit $\alpha_0 > \ldots \alpha_{k-1}$ und $\beta_i > 0$ und $N(\alpha_i) \cdot \beta_i < N(\alpha_i + 1)$ für alle $i < k$
3. Wert eines N-Index: $\sum_{i<k}(N(\alpha_i) \cdot \beta_i)$

Beispiel 1.7.11.

	g	k	α	β	Wert
1.	10	3	$\langle 3, 2, 0\rangle$	$\langle 2, 6, 1\rangle$	2601
2.	ω	2	$\langle \omega, 1\rangle$	$\langle 1, 2\rangle$	$\omega^\omega + \omega \cdot 2$

Kleiner Ankreuztest

		richig	falsch
1.	$\omega + \omega = 2 \cdot \omega$	☐	☐
2.	$\omega \cdot \omega^2 = \omega^2 \cdot \omega$	☐	☐
3.	$2^\omega = \omega$	☐	☐
4.	$1 \cdot \omega^\omega = \omega^\omega$	☐	☐
5.	$1 + 1 = 1$	☐	☐
6.	$(2 \cdot 2)^\omega = 2^\omega \cdot 2^\omega$	☐	☐
7.	$\omega \cdot (\omega + 1) = \omega^2 + \omega$	☐	☐
8.	$\omega \cdot (\omega + \omega) = (\omega + \omega) \cdot \omega$	☐	☐

2 Aussagenlogik

Motivation

Ursprüngliches Ziel der Logik ist es, das menschliche Schlussfolgern durch reine Zeichenmanipulation nachzubilden und so nachvollziehbar zu machen. Die klassische Logik betrachtet nur Aussagen, die entweder *wahr* oder *falsch* sind. Wir sprechen daher von einer *zweiwertigen* Logik. Zum Thema Logik gibt es umfangreiche Literatur, z. B. [Sch00].

Logische Kalküle können auf Computer übertragen werden und finden in der Digitaltechnik Anwendungen (z. B. beim Schaltkreisentwurf). Aussagenlogik, genauer deren Erweiterungen zu *Zeitlogiken*, durch die zeitliche Abläufe von Systemen gut dargestellt werden können, werden außerdem zum Beweis von System- oder Hardware-Eigenschaften eingesetzt (*Model Checking*) [CGP99].

2.1 Grundbegriffe

Definition 2.1.1 (Syntax)**.**

- Eine *atomare Formel* (kurz: *Atom*) hat die Form A_i für $i = 0, 1, \ldots$
  (Es ist auch ein überabzählbarer Atom-Vorrat denkbar, z. B. mit Indizes $i \in \mathbb{R}$.)
- *Formeln* sind induktiv definiert:
  1. Atomare Formeln sind Formeln.
  2. Sind F und G Formeln, so auch $(F \wedge G)$ und $(F \vee G)$.
  3. Ist H eine Formel, so auch $(\neg H)$.
- Nur was mit 1. bis 3. bildbar ist, sind (aussagenlogische) Formeln. Häufig werden jedoch zusätzlich zu den drei o. g. Operatorensymbolen $\wedge$, $\vee$ und $\neg$ weitere logische Operatoren verwendet (s. u.).
- Formeln haben immer eine endliche Länge.
- Klammern können ggf. weggelassen werden.
- $\mathcal{F}_0$ bezeichnet die Menge aller atomaren Formeln, $\mathcal{F}$ die Menge aller Formeln.

Logische Operatoren

$\wedge$	„und“	$\longrightarrow$	„impliziert“
$\vee$	„oder“	$\longleftrightarrow$	„äquivalent“
$\neg$	„nicht“	$\dot{\vee}$	„entweder-oder“

Schreibkonventionen

- Atome werden auch mit anderen Symbolen als A_i abgekürzt: A, B, C, P, Q, R
- Folgende Präzedenzen der Operatoren-Bindung gelten: $\neg \prec \wedge \prec \vee$

Definition 2.1.2. Die Menge $\mathcal{D}(F)$ aller in einer Formel F verwendeten Atome kann durch *strukturelle Induktion* (rekursiv) definiert werden:

$$\mathcal{D}(F) = \begin{cases} \{A\}, & \text{falls } F = A \text{ atomar} \\ \mathcal{D}(F_0), & \text{falls } F = \neg F_0 \\ \mathcal{D}(F_1) \cup \mathcal{D}(F_2), & \text{falls } F = F_1 \vee F_2 \\ \mathcal{D}(F_1) \cup \mathcal{D}(F_2), & \text{falls } F = F_1 \wedge F_2 \end{cases}$$

Definition 2.1.3 (Semantik)**.** Die Semantik einer aussagenlogischen Formel ist ein Wahrheitswert. Eine aussagenlogische Formel kann wahr oder falsch sein. Eine weitere Möglichkeit gibt es in der (klassischen) Logik nicht: *Tertium non datur.*

- Wir assoziieren die *Wahrheitswerte* mit den Zahlen $\{0, 1\}$: $0 \mathrel{\widehat{=}}$ *falsch* und $1 \mathrel{\widehat{=}}$ *wahr*.
- Sei $D \subseteq \mathcal{F}_0$. Dann heißt $\mathcal{A} : D \to \{0, 1\}$ *Belegung*. Man bezeichnet die Menge D, also die Menge der Atome, für die $\mathcal{A}$ definiert ist, als *Domäne* der Funktion $\mathcal{A}$, i. Z. $\mathrm{dom}(\mathcal{A})$.
- Eine Belegung $\mathcal{A}$ heißt *passend* zu einer Formel F, falls $\mathcal{A}$ (mindestens) auf allen Atomen, die in F vorkommen, definiert ist, d. h. $\mathcal{D}(F) \subseteq \mathrm{dom}(\mathcal{A})$.
- Sei $\mathcal{F}_D \subseteq \mathcal{F}$ die Menge aller Formeln F, die nur die atomaren Formeln aus D verwenden (zu denen also die Belegung $\mathcal{A}$ passt).
- Wir erweitern nun die Funktion $\mathcal{A}$ zu einer *Interpretation* von Formeln und definieren $\mathcal{A} : \mathcal{F}_D \to \{0, 1\}$ für (möglicherweise zusammengesetzte) Formeln $F, G \in \mathcal{F}_D$ gemäß folgender Tabelle:

$\mathcal{A}(F)$	$\mathcal{A}(G)$	$\mathcal{A}(\neg F)$	$\mathcal{A}(F \wedge G)$	$\mathcal{A}(F \vee G)$	$\mathcal{A}(F \longrightarrow G)$	$\mathcal{A}(F \longleftrightarrow G)$	$\mathcal{A}(F \veebar G)$
0	0	1	0	0	1	1	0
0	1	1	0	1	1	0	1
1	0	0	0	1	0	0	1
1	1	0	1	1	1	1	0

Übung 2.1.4. Formalisieren Sie die Aussage der Implikation *Wenn es regnet, nehme ich einen Regenschirm mit.* mittels Aussagenlogik! Welche Semantik haben die von Ihnen verwendeten Atome? Ist die Aussage (jetzt) wahr?

Bemerkung 2.1.5. Eine Implikation $F \longrightarrow G$ hat die Bedeutung: Wenn die Bedingung F gilt, dann auch die Schlussfolgerung G. Die Implikation als Ganzes gilt unter anderem dann als wahr, wenn die Bedingung F gar nicht erfüllt ist: *Ex falso quod libet.* Das heißt, aus einer falschen Bedingung lassen sich beliebige (wahre oder falsche) Schlussfolgerungen ziehen.

Syntax versus Semantik

- Wir unterscheiden strikt zwischen *Syntax* (Schreibung) und *Semantik* (Bedeutung) logischer Formeln. Syntaktisch gesehen sind Formeln gewisse Zeichenketten, die Atome und Operatorensysmbole mit einer bestimmten Bedeutung enthalten.
- Prinzipiell wären $2^{2^2} = 16$ zweistellige Operatoren denkbar. Allerdings sind die konstanten Funktionen $\{0,1\} \times \{0,1\} \mapsto \{0,1\}$ mit $(a_1, a_2) \mapsto 0$ bzw. 1 keine echten Operatoren, da deren Wert nicht von den Eingaben abhängig ist.

Definition 2.1.6 (Wahrheitstabelle)**.** Eine *Wahrheitstabelle*, auch Wahrheitstafel genannt ist eine tabellarische Aufstellung des Wahrheitswertsverlaufs einer logischen Aussage. Die Wahrheitstabelle zeigt für alle möglichen Belegungen, welchen Wahrheitswert die Gesamtaussage unter der jeweiligen Zuordnung annimmt.

Beispiel 2.1.7. Wahrheitstabelle zu $F = A \vee (B \wedge \neg C)$:

A	B	C	$\neg C$	$B \wedge \neg C$	$A \vee (B \wedge \neg C)$
0	0	0	1	0	0
0	0	1	0	0	0
0	1	0	1	1	1
0	1	1	0	0	0
1	0	0	1	0	1
1	0	1	0	0	1
1	1	0	1	1	1
1	1	1	0	0	1

Übung 2.1.8. Wie viele Belegungen muss man für eine gegebene Formel F unterscheiden, in der $n = |\mathcal{D}(F)|$ verschiedene Atome vorkommen, d. h. wie viele Zeilen hat die zugehörige Wahrheitstafel?

Definition 2.1.9 (Modelle)**.** Sei F eine Formel und $\mathcal{A}$ eine zu F passende Belegung. Dann gilt:

1. Falls $\mathcal{A}(F) = 1$ ist, so heißt $\mathcal{A}$ *Modell* von F, i. Z. $\mathcal{A} \models F$.
2. F heißt *erfüllbar*, falls F (mindestens) ein Modell besitzt, und *unerfüllbar* sonst (i. Z. $\not\models F$).
3. F heißt *gültig* bzw. *Tautologie*, i. Z. $\models F$, falls jede Belegung Modell von F ist.
4. Seien F und G Formeln und $\mathcal{A}$ eine zu F und G passende Belegung. Dann heißt G *Folgerung* von F, i. Z. $F \models G$, gdw. $\mathcal{A} \models F \Longrightarrow \mathcal{A} \models G$ für alle $\mathcal{A}$.

Bemerkung 2.1.10. Ob eine Formel erfüllbar, gültig bzw. unerfüllbar ist, kann man der zugehörigen Wahrheitstabelle, d.h. dem Wahrheitswerteverlauf, entnehmen: Erfüllbare Formeln weisen mindestens eine 1, gültige Formeln nur 1-en und unerfüllbare Formeln nur 0-en in der letzten Spalte der Wahrheitstabelle auf. Gültige Formeln sind also immer auch erfüllbar. Man kann also in der Aussagenlogik die Erfüllbarkeit durch Enummerieren testen, nämlich in einer Wahrheitstabelle.

Übung 2.1.11. Ist die Formel F aus Beispiel 2.1.7 unerfüllbar, erfüllbar und/oder gültig?

Übung 2.1.12. Eine Maschine hat drei Schalter A, B und C. Die Maschine läuft, wenn Schalter A aus, Schalter B und C an oder Schalter C an ist.

(a) Welche Lesarten lässt die obige Beschreibung zu? Vereindeutlichen Sie die Aussage!

(b) Formulieren Sie den dargestellten Sachverhalt als aussagenlogische Formel F! Verwenden Sie dabei die Atome A, B und C mit der Bedeutung, dass der entsprechende Schalter an ist.

(c) Stellen Sie eine Wahrheitstabelle zu F auf!

(d) Wie viele und welche Modelle besitzt F?

(e) Ist die Formel F also unerfüllbar, erfüllbar und/oder gültig?

Satz 2.1.13 (Prinzip des indirekten Beweises)**.** F ist gültig bzw. eine Tautologie gdw. $\neg F$ unerfüllbar ist.

Beweis: F ist Tautologie $\Longleftrightarrow$ jede zu F passende Belegung $\mathcal{A}$ ist Modell von $F \Longleftrightarrow$ für jede zu F passende Belegung $\mathcal{A}$ gilt $\mathcal{A}(F) = 1 \Longleftrightarrow$ für jede zu F (und damit auch zu $\neg F$) passende Belegung $\mathcal{A}$ gilt $\mathcal{A}(\neg F) = 0$ (folgt aus Semantik von $\neg$) $\Longleftrightarrow$ jede zu F passende Belegung $\mathcal{A}$ ist kein Modell von $\neg F \Longleftrightarrow$ es gibt kein Modell von $\neg F \Longleftrightarrow \neg F$ ist nicht erfüllbar $\Longleftrightarrow \neg F$ ist unerfüllbar. ⊠

Definition 2.1.14 (Formelmengen)**.** Sei M eine (möglicherweise unendliche) Formelmenge, die man auch *Theorie* nennt, und $\mathcal{A}$ eine zu allen Formeln $F \in M$ (gleichzeitig) passende Belegung. Dann heißt $\mathcal{A}$ *Modell der Formelmenge* M, i. Z. $\mathcal{A} \models M$, gdw. $\mathcal{A} \models F$ für alle $F \in M$ gilt. Die Begriffe *erfüllbar*, *unerfüllbar*, *gültig*, *Tautologie* sowie *Folgerung* (aus einer Theorie, i. Z. $M \models F$) aus Def. 2.1.9 werden sinngemäß auf Formelmengen angewendet.

Satz 2.1.15. Eine endliche Formelmenge $M = \{F_1, \ldots, F_n\}$ ist erfüllbar gdw. die Formel $F_1 \wedge \cdots \wedge F_n$ erfüllbar ist.

2.2 Äquivalenz von Formeln

Definition 2.2.1 (semantische Äquivalenz)**.** Zwei Formeln F und G heißen (semantisch) *äquivalent*, i. Z. $F \equiv G$ gdw. $\mathcal{A}(F) = \mathcal{A}(G)$ für alle zu F und G passenden Belegungen $\mathcal{A}$. Die Formeln F und G haben also den gleichen Wahrheitswerteverlauf.

Bemerkung 2.2.2. Syntaktische Gleichheit ($=$) impliziert semantische Gleichheit ($\equiv$), aber nicht umgekehrt. Es gilt also z. B. $((A \wedge B) \wedge A) \equiv (A \wedge B)$, aber $((A \wedge B) \wedge A) \neq (A \wedge B)$. $F = G \wedge H$ bedeutet „F hat die syntaktische Form $G \wedge H$“.

Lemma 2.2.3. Falls $F \equiv G$, so gilt auch:

1. $\neg F \equiv \neg G$
2. $(F \vee H) \equiv (G \vee H)$
3. $(F \wedge H) \equiv (G \wedge H)$

Beweis: Ad 1.:

- $\mathcal{A}(F) = \mathcal{A}(G) = 0 \Longrightarrow \mathcal{A}(\neg F) = 1 = \mathcal{A}(\neg G)$
- $\mathcal{A}(F) = \mathcal{A}(G) = 1 \Longrightarrow \mathcal{A}(\neg F) = 0 = \mathcal{A}(\neg G)$

Ad 2. und 3.: Hier ist eine 4-teilige Fallunterscheidung nötig (Wahrheitstafel). ⊠

Satz 2.2.4 (Ersetzbarkeitstheorem)**.** Sei $F \equiv G$. Wenn H eine Formel ist, in der F vorkommt, so ist H äquivalent zu H' (d. h. $H \equiv H'$), wobei H' aus H entsteht, indem F in H durch G (ein- oder ggf. mehrfach) ersetzt wird.

Beweis (durch strukturelle Induktion)**:**
Fallunterscheidung über den Formelaufbau (gemäß Def. 2.1.1). ⊠

Definition 2.2.5 (weitere Operatoren, vgl. S. 20)**.**

Implikation:	$F \longrightarrow F'$	$\equiv$	$\neg F \vee F'$
Biimplikation, Äquivalenz:	$F \longleftrightarrow F'$	$\equiv$	$(F \longrightarrow F') \wedge (F' \longrightarrow F)$
Entweder-Oder:	$F \dot{\vee} F'$	$\equiv$	$(F \wedge \neg F') \vee (\neg F \wedge F')$

Rechenregeln

Satz 2.2.6. Die folgenden Umformungsgesetze erhalten die Äquivalenz. Sie können durch Wahrheitstafeln bewiesen werden. Dabei sind F, F', F'' beliebige aussagenlogische Formeln, $\top$ (top) steht für eine Tautologie und $\bot$ (bottom) für eine unerfüllbare Formel. Klarerweise gilt $\neg\bot \equiv \top$ und $\neg\top \equiv \bot$.

Assoziativität: $(F \wedge F') \wedge F'' \equiv F \wedge (F' \wedge F'')$

$(F \vee F') \vee F'' \equiv F \vee (F' \vee F'')$

Kommutativität: $F \wedge F' \equiv F' \wedge F$

$F \vee F' \equiv F' \vee F$

Idempotenz: $F \wedge F \equiv F$

$F \vee F \equiv F$

Absorption: $F \wedge (F \vee F') \equiv F$

$F \vee (F \wedge F') \equiv F$

Distributivität: $F \wedge (F' \vee F'') \equiv (F \wedge F') \vee (F \wedge F'')$

$F \vee (F' \wedge F'') \equiv (F \vee F') \wedge (F \vee F'')$

Doppelnegation: $\neg\neg F \equiv F$

De Morgan-Regeln: $\neg(F \wedge F') \equiv \neg F \vee \neg F'$

$\neg(F \vee F') \equiv \neg F \wedge \neg F'$

Tautologie-Regeln: $F \wedge \top \equiv F$

$F \vee \top \equiv \top$

$F \vee \neg F \equiv \top$

Unerfüllbarkeit-Regeln: $F \wedge \bot \equiv \bot$

$F \vee \bot \equiv F$

$F \wedge \neg F \equiv \bot$

Beispiel 2.2.7 (Äquivalenzumformung und Anwendung des Ersetzbarkeitstheorems $(*)$)**.**

$$((A \wedge B) \wedge A) \overset{(*)}{\underset{\text{Komm.}}{\equiv}} ((B \wedge A) \wedge A) \overset{\text{Ass.}}{\equiv} (B \wedge (A \wedge A)) \overset{(*)}{\underset{\text{Idem.}}{\equiv}} (B \wedge A) \overset{\text{Komm.}}{\equiv} (A \wedge B)$$

Übung 2.2.8. Zeigen Sie die folgenden semantischen Gleichheiten durch Äquivalenzumformungen. Geben Sie bei allen Umformungsschritten die verwendeten Regeln an!

(a) $(A \longrightarrow (B \wedge C)) \equiv (A \longrightarrow B) \wedge (A \longrightarrow C)$

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

Übung 2.2.9. In einem Programm soll die folgende Bedingung in einer if-Anweisung realisiert werden: x ist größer oder gleich 0 und, falls x kleiner als 0 ist, so muss y kleiner als 0 sein. Formulieren Sie die Bedingung formal und vereinfachen Sie sie, sofern möglich!

Dualitätsprinzip

Alle Gesetze, die für $\wedge$ gelten, gelten auch für $\vee$. Beide Operatoren sind in diesem Sinne austauschbar (d. h. dual zueinander). Es gelten z. B. zwei Distributivgesetze (s. o.), die auseinander durch gleichzeitigen Austausch der beiden logischen Operatoren entstehen. In der Arithmetik gilt nur ein Distributivgesetz, vgl. $2 \cdot (3+4) = 2 \cdot 3 + 2 \cdot 4$, aber $2 + (3 \cdot 4) \neq (2+3) \cdot (2+4)$.

Definition 2.2.10 (Boolesche Algebra)**.** Eine algebraische Struktur mit zwei Verknüpfungen $\wedge$ und $\vee$ und den Eigenschaften aus Satz. 2.2.6 nennt man *Boole'sche Algebra*. Die Symbole $\top$ und $\bot$ sind hier Eins- bzw. Nullelement, d. h. neutrale Elemente (im Sinne von Def. 1.4.5) bezüglich $\wedge$ bzw. $\vee$.

Bemerkung 2.2.11 (Schreibvereinfachung)**.** Wegen des Assoziativitätsgesetzes lassen wir häufig Klammern weg.

Satz 2.2.12. Die Äquivalenzumformung bildet mit den in Satz 2.2.6 aufgelisteten Rechenregeln einen vollständigen Kalkül: Wenn zwei Formeln F und G semantisch äquivalent sind, so kann F in G umgeformt werden.

2.3 Normalformen

Definition 2.3.1 (Literal)**.** Ein *Literal* L ist ein Atom A oder ein negiertes Atom $\neg A$. A bezeichnet man als *positives*, $\neg A$ als *negatives* Literal.

Definition 2.3.2 (DNF)**.** Eine Formel F ist in *disjunktiver Normalform* (kurz: DNF), wenn sie eine Disjunktion von Konjunktionen von Literalen ist, d. h. wenn sie die folgende Form hat:

$$F = \bigvee_{i=1}^{n} \left(\bigwedge_{j=1}^{m_i} L_{ij} \right) = (L_{11} \wedge \cdots \wedge L_{1m_1}) \vee \cdots \vee (L_{n1} \wedge \cdots \wedge L_{nm_n})$$

Definition 2.3.3 (KNF)**.** Eine Formel F ist in *konjunktiver Normalform* (kurz: KNF), wenn sie eine Konjunktion von Disjunktionen von Literalen ist, d. h. wenn sie die folgende Form hat:

$$F = \bigwedge_{i=1}^{n} \left(\bigvee_{j=1}^{m_i} L_{ij} \right) = (L_{11} \vee \cdots \vee L_{1m_1}) \wedge \cdots \wedge (L_{n1} \vee \cdots \vee L_{nm_n})$$

Beispiel 2.3.4.

- $A \vee (B \wedge \neg C)$ ist in DNF.
- $(A \wedge B) \wedge A \equiv A \wedge B$ ist in KNF *und* DNF.

Übung 2.3.5. Betrachten Sie die folgenden Formeln:

1. $A \vee (\neg B \wedge C \wedge \neg D)$
2. $E \vee D$
3. $\neg(A \vee \neg C)$

Aufgaben:

(a) Unterstreichen Sie alle (größtmöglichen) Literale!

(b) Welche Formeln sind in DNF oder KNF?

(c) Geben Sie ggf. die n bzw. m_i aus den zwei vorigen Definitionen an!

Definition 2.3.6 (komplementäres Literal)**.** Sei L ein Literal. Dann ist das zu L komplementäre Literal folgendermaßen definiert:

$$\overline{L} = \begin{cases} A, & \text{falls } L = \neg A \\ \neg A, & \text{falls } L = A \end{cases}$$

Theorem 2.3.7. Zu jeder Formel F gibt es je eine äquivalente Formel in DNF und KNF.

Beweis (strukturelle Induktion über den Formelaufbau)**:**

- Induktionsanfang: $F = A$ atomar $\Longrightarrow$ F ist bereits in KNF und DNF ($n = m_1 = 1$).
- Induktionsschritt:
  1. $F = \neg G$
     Dann gibt es nach Induktionsvorausetzung $G_1 \equiv G \equiv G_2$ mit G_1 in DNF und G_2 in KNF.
     (a) $G_1 = \bigvee_{i=1}^{n} \left(\bigwedge_{j=1}^{m_i} L_{ij} \right)$ Dann gilt:
     $$F \overset{\text{Theorem 2.2.4}}{\equiv} \neg G_1 = \neg \left(\bigvee_{i=1}^{n} \left(\bigwedge_{j=1}^{m_i} L_{ij} \right) \right) \overset{De\ Morgan}{\underset{(mehrfach)}{\equiv}} \bigwedge_{i=1}^{n} \left(\neg \left(\bigwedge_{j=1}^{m_i} L_{ij} \right) \right) \overset{De\ Morgan}{\underset{(mehrfach)}{\equiv}} \bigwedge_{i=1}^{n} \left(\bigvee_{j=1}^{m_i} \neg L_{ij} \right) \overset{Doppelnegation}{\underset{(mehrfach)}{\equiv}} \bigwedge_{i=1}^{n} \left(\bigvee_{j=1}^{m_i} \overline{L_{ij}} \right)$$
     Letzere Formel ist in KNF.
     (b) Nachweis einer äquivalenten DNF für $\neg G_2$ analog zu (a).
  2. $F = G \vee H$
     Dann gibt es nach Induktionsvoraussetzung $G_1 \equiv G \equiv G_2$, und $H_1 \equiv H \equiv H_2$ mit G_1 und H_1 in DNF bzw. H_2 und G_2 in KNF.
     (a) $$F \equiv G_1 \vee H_1 = \left(\bigvee_{i=1}^{n} G_{1i} \right) \vee \left(\bigvee_{j=1}^{m} H_{1j} \right)$$
     ... ist bereits in DNF (nach Anwendung des Assoziativgesetzes)!
     (b) $$F \equiv G_2 \vee H_2 = \left(\bigwedge_{i=1}^{n} G_{2i} \right) \vee \left(\bigwedge_{j=1}^{m} H_{2j} \right) \overset{Distr.}{\equiv} \bigwedge_{i=1}^{n} \bigwedge_{j=1}^{m} \left(G_{2i} \vee H_{2j} \right)$$
     Letztere Formel ist in KNF.
  3. Nachweis einer äquivalenten KNF und DNF für $F = G \wedge H$ analog zu 2.

Man beachte, dass im Fall 1. die Existenz der jeweils anderen Normalform vorausgesetzt wird. Das geht nur, wenn (wie hier) gleichzeitig die Existenzen von KNF und DNF bewiesen werden. ⊠

Herstellung von Normalformen

1. Direktes Umformen (Vorgehen wie in obigem Beweis)
   - Negationen mittels De-Morgan-Regel nach „innen“ bringen
   - Doppelnegation ggf. auflösen

- Distributivgesetz anwenden

Beispiel: $A \vee \neg(\neg B \vee A) \equiv A \vee (B \wedge \neg A)$

2. Herleitung aus Wahrheitstafel:

Beispiel:

A	B	$A \vee (B \wedge \neg A)$
0	0	0
0	1	1
1	0	1
1	1	1

DNF: „1“-Zeilen betrachen
- Atome „positiv“ $\wedge$-verknüpfen (Konjunktionen)
- Alle Konjunktionen dann $\vee$-verknüpfen

KNF: „0“-Zeilen betrachten
- Atome „negativ“ $\vee$-verknüpfen (Disjunktionen)
- Alle Disjunktionen dann $\wedge$-verknüpfen

3. KV-Diagramme, Quine-McCluskey-Verfahren, ...

Definition 2.3.8 (kanonische Normalform)**.** Falls in jeder Teilformel einer DNF oder KNF jedes Atom genau einmal vorkommt (positiv oder negativ), d. h. $m_i = |\mathcal{D}(F)|$ für $1 \leq i \leq n$, so handelt es sich um eine *kanonische Normalform*. Diese sind (bis auf die Reihenfolge der Elemente) jeweils eindeutig bestimmt.

Übung 2.3.9. *Alle durch* 4*, aber nicht durch* 100 *teilbaren Jahre sind Schaltjahre, außer wenn sie durch* 400 *teilbar sind.* Formalisieren Sie diese Bedingung für Schaltjahre in Aussagenlogik unter Verwendung der Atome D_4, D_{100} und D_{400} mit der Bedeutung, dass das Jahr durch die jeweilige Zahl im Index (ohne Rest) teilbar ist! Welche Kombinationen von Wahrheitswerten der Atome sind gar nicht möglich?

2.4 Resolution

Klauselmengendarstellung zu einer KNF

KNFs können als *Mengen von Mengen von Literalen* dargestellt werden. Diese Darstellung ist eindeutig, da die Struktur einer KNF (Konjunktionen von Disjunktionen) fest vorgegeben ist (daher die Darstellung als Menge von Mengen von Literalen) und Kommutativ-, Idempotenz- und Assoziativgesetz gelten (daher sind Mengen adäquat). *Klauseln* sind die Elemente der *Klauselmenge* M, Literale die Elemente der Klauseln.

Beispiel 2.4.1 (Konjunktive Normalform und Darstellung als Klauselmenge)**.**

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

$$\widehat{=}$$

$$\big\{\{A, B\}, \{\neg A, B\}, \{\neg A, \neg B\}, \{A, \neg B\}\big\}$$

Definition 2.4.2 (binäre Resolvente)**.** Eine Klausel R heißt *binäre Resolvente* der Klauseln K_1 und K_2 gdw. es zwei komplementäre Literale $L \in K_1$ und $\overline{L} \in K_2$ gibt (d. h. $L = A$ und $\overline{L} = \neg A$ oder umgekehrt für ein Atom A) und es gilt:

$$R = K_1 \setminus \{L\} \cup K_2 \setminus \{\overline{L}\}$$

Bemerkung 2.4.3 (Vorgriff auf Lemma 2.5.3)**.** Eine Resolvente R der Klauseln K_1 und K_2 ist immer eine logische Konsequenz aus K_1 und K_2, d. h. es gilt stets $K_1 \wedge K_2 \equiv K_1 \wedge K_2 \wedge R$. Letzteres lässt sich durch Äquivalenzumformungen auf Grundlage der Umformungsregeln aus Satz 2.2.6 nur umständlich herleiten. Betrachte hierzu $K_1 = A \vee B$, $K_2 = \neg B \vee C$ und $R = A \vee C$.

Definition 2.4.4 (leere Klausel)**.** Die *leere Klausel* $\{\}$ (auch mit $\square$ bezeichnet) enthält keine Literale und ist äquivalent zu $\bot$ – der unerfüllbaren Formel.

Definition 2.4.5 (Deduktion mittels Resolution)**.** Wir sagen K ist *mittels Resolution deduzierbar* aus der gegebenen Klauselmenge $\mathcal{G}$ gdw. es gibt eine endliche Sequenz von Klauseln $K_1, \ldots, K_n$ mit

1. $K = K_n$ und
2. für alle K_m mit $1 \leq m \leq n$ gilt $K_m \in \mathcal{G}$ oder K_m ist binäre Resolvente aus K_i und K_j mit $1 \leq i, j < m$.

Beispiel 2.4.6 (Fortsetzung von Bsp. 2.4.1)**.**

$K_1 \quad \{A, B\}$
$K_2 \quad \{\neg A, B\}$
$\Longrightarrow \quad K_3 \quad \{B\}$

- K_3 ist Resolvente aus $K_1, K_2 \in \mathcal{G}$
- $K_3 = K_1 \setminus \{A\} \cup K_2 \setminus \{\neg A\}$

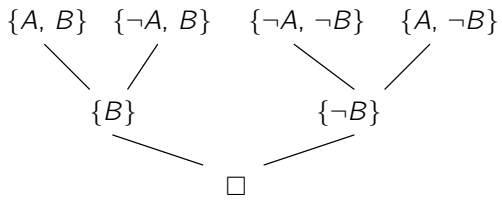
$K_4 \quad \{\neg A, \neg B\}$
$K_5 \quad \{A, \neg B\}$
$\Longrightarrow \quad K_6 \quad \{\neg B\}$

- K_6 ist Resolvente aus $K_4, K_5 \in \mathcal{G}$
- $K_6 = K_4 \setminus \{\neg A\} \cup K_5 \setminus \{A\}$

$K_3 \quad \{B\}$
$K_6 \quad \{\neg B\}$
$\Longrightarrow \quad K_7 \quad \{\,\} = \square$

- K_7 ist Resolvente aus K_3, K_6
- $K_7 = K_3 \setminus \{B\} \cup K_6 \setminus \{\neg B\}$

Darstellung als Graph:



Die Deduktion entspricht einer sogenannten topologischen Sortierung der Knoten im Graph.

Definition 2.4.7 (Refutation)**.** Eine Deduktion der leeren Klausel (d. h. $\square$ als letztes Element des Resolutionsbeweises) nennt man *Refutation*, d. h. Widerlegung, denn M muss in diesem Fall in sich widersprüchlich sein (vgl. Bemerkung 2.4.3).

Wie wir später sehen werden (in Korollar 2.5.5), gilt:

M unerfüllbar gdw. $\square$ deduzierbar aus M.

Satz 2.4.8 (Prinzip des indirekten Beweises, vgl. Satz 2.1.13)**.** Sei F eine Formel und M eine Formelmenge. Dann gilt:

$$M \models F \text{ gdw. } M \cup \{\neg F\} \text{ unerfüllbar.}$$

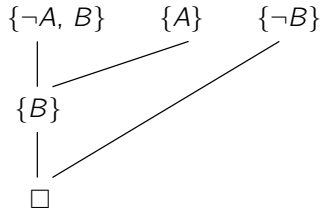
Beweis (Anwendung von Def. 2.1.14)**:** $M \models F$ gdw. $\forall \mathcal{A} : \mathcal{A} \models M \Longrightarrow \mathcal{A} \models F$ gdw. $\forall \mathcal{A} : \mathcal{A} \models M \Longrightarrow \mathcal{A} \not\models \neg F$ gdw. $\nexists \mathcal{A} : \mathcal{A} \models M \wedge \mathcal{A} \models \neg F$ gdw. $\nexists \mathcal{A} : \mathcal{A} \models M \cup \{\neg F\}$ gdw. $M \cup \{\neg F\}$ unerfüllbar. $\boxtimes$

Beispiel 2.4.9. Die bereits aus der Antike stammende logische Schlussfigur *modus ponens*

$$\frac{A \longrightarrow B \quad A}{B}$$

besagt, dass aus zwei Aussagen der Form *Wenn A, dann B* und A (den beiden *Prämissen* der Schlussfigur) eine Aussage der Form B (die *Konklusion* der Schlussfigur) hergeleitet werden kann.

Die Gültigkeit dieser Schlussweise kann durch aussagenlogische Resolution bewiesen werden: Setze dazu $M = \{A \longrightarrow B, A\}$ und $F = B$. Dann ist zu zeigen, dass die Behauptung F aus den Voraussetzungen in M folgt, d. h. $M \models F$. Dies tun wir indirekt, indem wir (gemäß Satz 2.4.8) die Unerfüllbarkeit von $M \cup \{\neg F\}$ zeigen. Letzteres gelingt durch eine Refutation (gemäß Def. 2.4.7), d.h. Herleitung der leeren Klausel $\square$ mittels aussagenlogischer Resolution (unter Berücksichtigung der Äquivalenz $A \longrightarrow B \equiv \neg A \vee B$) wie folgt:



Übung 2.4.10. Zeigen Sie die Unerfüllbarkeit der Formel

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

(in KNF) durch aussagenlogische Resolution, ggf. durch Einsatz eines Theorembeweisers!

Bemerkung 2.4.11.

- Im Verlaufe einer Resolutionsherleitung können Klauseln sowie bereits gebildete Resolventen einmal, mehrmals oder auch gar nicht verwendet werden.
- Es darf immer nur über ein Literal je Klausel resolviert werden. Die Klauseln $K_1 = \{A, B\}$ und $K_2 = \{\neg A, \neg B\}$ z. B. dürfen keinesfalls in einem Schritt zur leeren Klausel $\{\} = \square$ resolviert werden, was einer Refutation entspräche. Dies würde nämlich fälschlicherweise zu dem Schluss führen, dass die Klauselmenge $M = \{K_1, K_2\}$ unerfüllbar ist. M hat aber zwei Modelle $\mathcal{A}$, nämlich $[A \mapsto 0, B \mapsto 1]$ und $[A \mapsto 1, B \mapsto 0]$.

- Die leere Klauselmenge $M_1 = \{\}$ muss unbedingt unterschieden werden von einer (nicht-leeren) Klauselmenge, die die leere Klausel enthält, z. B. $M_2 = \{\{\}\}$. M_1 gilt als erfüllbar, sogar tautologisch, während M_2 unerfüllbar ist.

Definition 2.4.12 (Resolutionsfunktion Res)**.** Wir definieren für Klauselmengen M:

$$\begin{aligned} \mathrm{Res}(M) &= M \cup \{R \mid R \text{ ist binäre Resolvente zweier Klauseln aus } M\} \\ \mathrm{Res}^0(M) &= M \\ \mathrm{Res}^{i+1}(M) &= \mathrm{Res}(\mathrm{Res}^i(M)) \\ \mathrm{Res}^*(M) &= \bigcup_{n \geq 0} \mathrm{Res}^n(M) \end{aligned}$$

Bemerkung 2.4.13. $\mathrm{Res}^*(M)$ ist die Menge aller Resolventen aus M einschließlich der Elemente in M, die sich in $r \geq 0$ (also null, einem oder mehr) Resolutionsschritten herleiten lassen. $\mathrm{Res}^n(M)$ enthält i. d. R. Resolventen, die sich in $r > n$ Resolutionsschritten herleiten lassen. Für alle $n \in \mathbb{N}_0$ gilt: $\mathrm{Res}^n(M) \subseteq \mathrm{Res}^{n+1}(M)$.

Beispiel 2.4.14 (Fortsetzung von Bsp. 2.4.1)**.**

$$\begin{aligned} \mathrm{Res}^0(M) &= \big\{\{A, B\}, \{\neg A, B\}, \{\neg A, \neg B\}, \{A, \neg B\}\big\} \\ \mathrm{Res}^1(M) &= \mathrm{Res}(\mathrm{Res}^0(M)) = \mathrm{Res}(M) = M \cup \big\{\{B\}, \{\neg B\}, \underline{\{B, \neg B\}}, \{A\}, \{\neg A\}, \underline{\{A, \neg A\}}\big\} \\ &\quad \text{Tautologien werden normalerweise in Resolventenmengen weggelassen.} \\ \mathrm{Res}^2(M) &= \mathrm{Res}(\mathrm{Res}^1(M)) = \mathrm{Res}^1(M) \cup \big\{\{\}\big\} \qquad \text{Deduktion der leeren Klausel (s. o).} \\ \mathrm{Res}^3(M) &= \mathrm{Res}^2(M) = \mathrm{Res}^*(M) \qquad \text{Es kommen keine neuen Resolventen hinzu.} \end{aligned}$$

2.5 Korrektheit und Vollständigkeit der Resolution

Wenn die leere Klausel hergeleitet werden kann, ist die gegebene Formelmenge unerfüllbar. Wenn eine Formelmenge unerfüllbar ist, so lässt sich immer die leere Klausel deduzieren. Diese beiden Sachverhalte bezeichnet man als Korrektheit und Vollständigkeit der Resolution. Die folgenden Beweise folgen [Sto98]. Sie gelten für Formelmengen beliebiger Größe und verwenden dabei zur Nummerierung der Atome Ordinalzahlen (vgl. Abschnitt 1.7).

Lemma 2.5.1 (Stetigkeit)**.** Sei Γ eine Menge von Klauselmengen, so dass alle endlichen Vereinigungen existieren, d. h. $\mathcal{G}_1, \ldots, \mathcal{G}_n \in \Gamma$ impliziert $\mathcal{G}_1 \cup \cdots \cup \mathcal{G}_n \in \Gamma$. Dann gilt:

$$\mathrm{Res}^*\Big(\bigcup_{\mathcal{G} \in \Gamma} \mathcal{G}\Big) = \bigcup_{\mathcal{G} \in \Gamma} \mathrm{Res}^*(\mathcal{G})$$

Bemerkung 2.5.2. Obige Aussage bedeutet, dass eine Vertauschung der Reihenfolge von Res^* und Vereinigung $\bigcup_{\mathcal{G} \in \Gamma}$ möglich ist. Zum Vergleich: In der Analysis bedeutet Stetigkeit, dass eine Vertauschung der Reihenfolge von Funktions- und Grenzwertbildung möglich ist:

$$\lim_{x \to x_0} f(x) = f\Big(\lim_{x \to x_0} x\Big) = f(x_0)$$

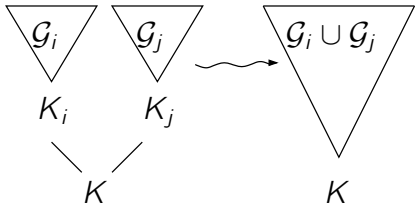
Beweis (von Lemma 2.5.1)**:**

$$\begin{aligned} K \in \mathrm{Res}^*(\bigcup_{\mathcal{G}\in\Gamma} \mathcal{G}) &\iff K \text{ ist deduzierbar aus } \bigcup_{\mathcal{G}\in\Gamma} \mathcal{G} \\ &\overset{(\dagger)}{\iff} K \text{ ist deduzierbar aus einem } \mathcal{G}^* \in \Gamma \\ &\iff K \in \bigcup_{\mathcal{G}\in\Gamma} \mathrm{Res}^*(\mathcal{G}) \end{aligned}$$

Zu zeigen bleibt (†): K ist aus $\bigcup_{\mathcal{G}\in\Gamma} \mathcal{G}$ durch die Sequenz $K_1, \ldots, K_n$ deduzierbar $\iff K$ ist deduzierbar aus $\mathcal{G}^* \in \Gamma$.

$\boxed{\Longrightarrow}$ Vollständige Induktion über die Länge n (Anzahl der Schritte) der Herleitung:

- $\underline{n = 1}$: Die Sequenz besteht in diesem Fall nur aus $K = K_1 \in \bigcup_{\mathcal{G}\in\Gamma} \mathcal{G}$. Es muss also $K \in \mathcal{G}^*$ für ein $\mathcal{G}^* \in \Gamma$ sein.
- $\underline{n > 1}$: Fallunterscheidung:
  (a) $K = K_n \in \bigcup_{\mathcal{G}\in\Gamma} \mathcal{G}$: analog zu Fall $n = 1$.
  (b) $K = K_n$ ist Resolvente aus K_i, K_j $(1 \leq i, j < n)$:
  Nach Induktionsvoraussetzung gibt es $\mathcal{G}_i, \mathcal{G}_j \in \Gamma$, so dass K_i aus $\mathcal{G}_i$ und K_j aus $\mathcal{G}_j$ deduzierbar sind. Wegen $\mathcal{G}_i \cup \mathcal{G}_j \in \Gamma$ (endliche Vereinigung) folgt, dass K aus $\mathcal{G}^* = \mathcal{G}_i \cup \mathcal{G}_j$ deduzierbar ist.



$\boxed{\Longleftarrow}$ Falls K aus $\mathcal{G}^* \in \Gamma$ deduzierbar ist, so erst recht aus $\bigcup_{\mathcal{G}\in\Gamma} \mathcal{G}$.

Res^* ist also eine *monotone Funktion*, d. h. $\mathcal{G}_1 \subseteq \mathcal{G}_2 \Longrightarrow \mathrm{Res}^*(\mathcal{G}_1) \subseteq \mathrm{Res}^*(\mathcal{G}_2)$. ⊠

Lemma 2.5.3 (Resolutionslemma)**.** Eine Klauselmenge $\mathcal{G}$ ist erfüllbar gdw. $\mathrm{Res}^*(\mathcal{G})$ erfüllbar ist.

Beweis:
$\boxed{\Longrightarrow}$ Sei K aus $\mathcal{G}$ mittels der Sequenz $K_1, \ldots, K_n$ mit $K = K_n$ deduziert. Wir zeigen

$$\mathcal{A} \models \mathcal{G} \implies \mathcal{A} \models K$$

mittels vollständiger Induktion über die Länge n der Herleitung:

- $\underline{n = 1}$: $K = K_1 \in \mathcal{G}$. $\mathcal{A} \models K$ muss wegen $\mathcal{A} \models \mathcal{G}$ gelten (vgl. Def. 2.1.14).
- $\underline{n > 1}$: Fallunterscheidung:

(a) $K = K_n \in \mathcal{G}$: wie Fall $n = 1$.

(b) $K = K_i \setminus \{L_i\} \cup K_j \setminus \{L_j\}$ ist binäre Resolvente ($1 \leq i, j < n$):
Nach Induktionsvoraussetzung gilt: $\mathcal{A} \models K_i$ und $\mathcal{A} \models K_j$. Da L_i und L_j komplementäre Literale sind, erfüllt $\mathcal{A}$ entweder L_i oder L_j (aber nicht beide). Daher muss $\mathcal{A} \models K_i \setminus \{L_i\}$ oder $\mathcal{A} \models K_j \setminus \{L_j\}$ gelten, folglich $\mathcal{A} \models K$.

$\boxed{\Longleftarrow}$ Wegen $\mathcal{G} \subseteq \text{Res}^*(\mathcal{G})$ folgt sofort die Behauptung. ⊠

Lindenbaums Erweiterung

Idee: Erweitere $\mathcal{G}$ zu einer „maximalen", konsistenten Formelmenge. Wir nummerieren alle Atome in $\mathcal{G}$ mit Ordinalzahlen und bilden die Kette $\mathcal{G} = \mathcal{G}_0 \subseteq \mathcal{G}_1 \subseteq \ldots \subseteq \mathcal{G}_\alpha \subseteq \ldots$ Wegen des sog. Auswahlaxioms (äquivalent zum Wohlordnungssatz 1.6.7) lässt sich die Funktion $\mathcal{G}_\alpha$ wie folgt definieren:

$$\mathcal{G}_\alpha = \begin{cases} \mathcal{G} & \text{, falls } \alpha = 0 \\ \mathcal{G}_{\alpha-1} \cup \mathcal{G}^*_{\alpha-1} & \text{, falls } \alpha \text{ Successor-Zahl ist} \\ \bigcup_{\beta<\alpha} \mathcal{G}_\beta & \text{, falls } \alpha \text{ Limeszahl ist} \end{cases}$$

$$\mathcal{G}^*_\alpha = \begin{cases} \{\neg A_\alpha\} & \text{, falls } \square \notin \text{Res}^*(\mathcal{G}_\alpha \cup \{\neg A_\alpha\}) \text{ (*)} \\ \{A_\alpha\} & \text{, falls } \square \notin \text{Res}^*(\mathcal{G}_\alpha \cup \{A_\alpha\}) \text{ und (*) gilt nicht} \\ \emptyset & \text{, andernfalls} \end{cases}$$

Sei nun $\overline{\mathcal{G}}$ die Vereinigung aller $\mathcal{G}_\alpha$ (bis zu einem hinreichend großen Index $\beta > \alpha$). Jede endliche Vereinigung in $\overline{\mathcal{G}}$ existiert:

$$\mathcal{G}_{\alpha_1} \cup \ldots \cup \mathcal{G}_{\alpha_n} = \mathcal{G}_{\max\{\alpha_1, \ldots, \alpha_n\}}$$

Lemma 2.5.4. $\text{Res}^*(\mathcal{G})$ unerfüllbar gdw. $\square \in \text{Res}^*(\mathcal{G})$.

Beweis:
$\boxed{\Longrightarrow}$ Wir zeigen die Kontraposition

$$\square \notin \text{Res}^*(\mathcal{G}) \Longrightarrow \text{Res}^*(\mathcal{G}) \text{ erfüllbar, d. h. es gibt } \mathcal{A} \text{ mit } \mathcal{A} \models \text{Res}^*(\mathcal{G})$$

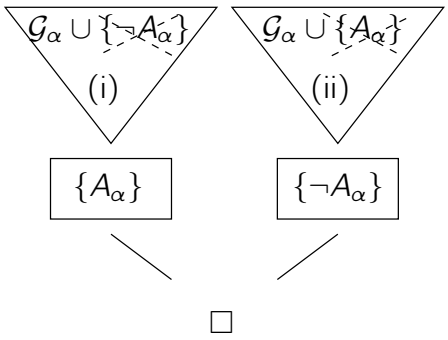
in drei Schritten, ausgehend von der Voraussetzung $\square \notin \text{Res}^*(\mathcal{G})$:

1. Wir zeigen $\square \notin \text{Res}^*(\overline{\mathcal{G}})$, d. h. $\square \notin \text{Res}^*(\mathcal{G}_\alpha)$ für alle Ordinalzahlen α (wegen Stetigkeit von Res^*, vgl. Lemma 2.5.1) durch transfinite Induktion:
   - $\underline{\alpha = 0}$: Wegen $\mathcal{G} = \mathcal{G}_0$ gilt $\square \notin \text{Res}^*(\mathcal{G}_0)$. Anderfalls wäre $\square \in \text{Res}^*(\mathcal{G})$.
   - $\underline{\alpha \text{ ist Nachfolgerzahl}}$: Nach Induktionsvoraussetzung gilt $\square \notin \text{Res}^*(\mathcal{G}_{\alpha-1})$.
     Wir betrachten alle drei Fälle der Definition von $\mathcal{G}^*_{\alpha-1}$:
     - $\mathcal{G}^*_{\alpha-1} = \{A_{\alpha-1}\}$ oder $\{\neg A_{\alpha-1}\}$: Dann gilt $\square \notin \text{Res}^*(\mathcal{G}_\alpha)$ per Definition.
     - $\mathcal{G}^*_{\alpha-1} = \emptyset$: In diesem Fall gilt $\mathcal{G}_\alpha = \mathcal{G}_{\alpha-1}$, also ebenfalls $\square \notin \text{Res}^*(\mathcal{G}_\alpha)$.

- α ist Limeszahl: Nach Voraussetzung gilt $\square \notin \text{Res}^*(\mathcal{G}_\beta)$ für $\beta < \alpha$.
  Es folgt: $\square \notin \bigcup_{\beta<\alpha} \text{Res}^*(\mathcal{G}_\beta) \overset{\text{Lemma 2.5.1}}{=} \text{Res}^*(\bigcup_{\beta<\alpha} \mathcal{G}_\beta) \overset{Def.}{=} \text{Res}^*(\mathcal{G}_\alpha)$

2. Für Ordinalzahlen α gilt entweder $A_\alpha \in \overline{\mathcal{G}}$ oder $\neg A_\alpha \in \overline{\mathcal{G}}$.

   Beweis (indirekt): Andernfalls müsste es nach Definition von $\mathcal{G}^*_\alpha$ ein α geben mit (i) $\square \in \text{Res}^*(\mathcal{G}_\alpha \cup \{\neg A_\alpha\})$ und (ii) $\square \in \text{Res}^*(\mathcal{G}_\alpha \cup \{A_\alpha\})$. Also sind A_α bzw. $\neg A_\alpha$ aus $\mathcal{G}_\alpha$ deduzierbar. Hierzu muss man nur auf den Gebrauch von $\neg A_\alpha$ bzw. A_α in den Herleitungen von $\square$ verzichten (vgl. Skizze). Folglich ist $\square \in \text{Res}^*(\mathcal{G}_\alpha)$ im Widerspruch zu 1.



3. Wir definieren: $\mathcal{A}(A_\alpha) = 1 \Longleftrightarrow A_\alpha \in \overline{\mathcal{G}}$. $\mathcal{A}$ ist ein Modell von $\overline{\mathcal{G}}$.

   Beweis (indirekt): Andernfalls existiert Klausel $K \in \overline{\mathcal{G}}$ mit $\mathcal{A} \not\models K$, d. h. $\mathcal{A}(L) = 0$ für alle $L \in K$. Dann muss wegen 2. $\overline{L} \in \overline{\mathcal{G}}$ sein. Dann lässt sich aber aus K und den $\overline{L}$ die leere Klausel $\square$ herleiten. Dies widerspricht aber 1.

$\mathcal{A}$ ist wegen $\mathcal{G} \subseteq \overline{\mathcal{G}}$ auch Modell von $\mathcal{G}$. Damit ist $\mathcal{G}$ erfüllbar und wegen Lemma 2.5.3 auch $\text{Res}^*(\mathcal{G})$.

$\boxed{\Longleftarrow}$ Wir zeigen wieder die Kontraposition:

$$\text{Res}^*(\mathcal{G}) \text{ erfüllbar} \Longrightarrow \square \notin \text{Res}^*(\mathcal{G})$$

Da die leere Klausel einer unerfüllbaren Formel entspricht, kann wegen Def. 2.1.14 nicht $\square \in \text{Res}^*(\mathcal{G})$ gelten. $\boxtimes$

Korollar 2.5.5 (aus Lemma 2.5.3 und 2.5.4)**.**

1. *Korrektheit* der Resolution: $\square \in \text{Res}^*(\mathcal{G}) \Longrightarrow \mathcal{G}$ unerfüllbar
2. *Vollständigkeit* der Resolution: $\mathcal{G}$ ist unerfüllbar $\Longrightarrow \square \in \text{Res}^*(\mathcal{G})$

Theorem 2.5.6 (Kompaktheit, Endlichkeitssatz)**.** Eine Klauselmenge $\mathcal{G}$ ist erfüllbar gdw. jede endliche Teilmenge von $\mathcal{G}$ erfüllbar ist.

Beweis: Sei $\Gamma = 2^{(\mathcal{G})}$ die Menge aller endlichen Teilmengen von $\mathcal{G}$ (genannt beschränkte Potenzmenge). Klarerweise existieren alle endlichen Vereinigungen in Γ und es gilt (†) $\mathcal{G} = \bigcup_{\mathcal{G}^* \in 2^{(\mathcal{G})}} \mathcal{G}^*$. Wir zeigen die Kontraposition des Theorems:

$$
\begin{array}{lll}
\mathcal{G} \text{ unerfüllbar} & \overset{\text{Lemma 2.5.3}}{\Longleftrightarrow} & \mathrm{Res}^*(\mathcal{G}) \text{ unerfüllbar} \\
& \overset{(\dagger)}{\Longleftrightarrow} & \square \in \mathrm{Res}^*(\bigcup_{\mathcal{G}^* \in 2^{(\mathcal{G})}} \mathcal{G}^*) \\
& \Longleftrightarrow & \exists\, \mathcal{G}^* \in 2^{(\mathcal{G})} : \square \in \mathrm{Res}^*(\mathcal{G}^*) \\
& \overset{\text{Lemma 2.5.4}}{\Longleftrightarrow} & \exists\, \mathcal{G}^* \in 2^{(\mathcal{G})} : \mathrm{Res}^*(\mathcal{G}^*) \text{ unerfüllbar} \\
& \overset{\text{Lemma 2.5.3}}{\Longleftrightarrow} & \exists \text{ endliche unerfüllbare Menge } \mathcal{G}^* \subseteq \mathcal{G}
\end{array}
$$

$$
\overset{\text{Lemma 2.5.4}}{\Longleftrightarrow} \square \in \mathrm{Res}^*(\mathcal{G}) \qquad \overset{\text{Lemma 2.5.1}}{\Longleftrightarrow} \square \in \bigcup_{\mathcal{G}^* \in 2^{(\mathcal{G})}} \mathrm{Res}^*(\mathcal{G}^*)
$$

⊠

Korollar 2.5.7 (Kontraposition von Theorem 2.5.6)**.** Eine (möglicherweise unendliche) Klauselmenge $\mathcal{G}$ ist unerfüllbar gdw. es (mindestens) eine endliche unerfüllbare Teilmenge von $\mathcal{G}$ gibt.

Bemerkung 2.5.8. Der Endlichkeitssatz (Theorem 2.5.6) und somit auch dessen Kontraposition (Korollar 2.5.7) gelten nicht nur für Klauselmengen, sondern für Formelmengen allgemein.

2.6 Hornformeln

Definition 2.6.1 (Hornformel)**.** Eine Formel F in KNF, bei der jede Klausel höchstens ein positives Literal enthält, heißt *Hornformel*.

Darstellungen

1. KNF: $(A \vee \neg D) \wedge (\neg A \vee \neg B \vee C) \wedge (\neg C \vee \neg D) \wedge D$
2. Klauselmenge: $\big\{\{A, \neg D\}, \{\neg A, \neg B, C\}, \{\neg C, \neg D\}, \{D\}\big\}$
3. Implikationsform: $(\underline{D} \longrightarrow \underline{A}) \wedge (\underline{A} \wedge B \longrightarrow C) \wedge (C \wedge \underline{D} \longrightarrow \bot) \wedge (\top \longrightarrow \underline{D})$

Die folgenden Beispiele zeigen, wie sich die Implikationsform aus der KNF herleiten lässt. Bei Klauseln ohne positive bzw. ohne negative Literale wird mit den Symbolen $\bot$ (*bottom* = unerfüllbare Formel) bzw. $\top$ (*top* = Tautologie) gearbeitet.

- $(\neg A \vee \neg B) \vee C \equiv \neg(A \wedge B) \vee C \equiv (A \wedge B) \longrightarrow C$
- $\neg C \vee \neg D \equiv \neg(C \wedge D) \vee \bot \equiv (C \wedge D) \longrightarrow \bot$
- $D \equiv \bot \vee D \equiv \neg\top \vee D \equiv \top \longrightarrow D$

Markierungsalgorithmus

1. Markiere alle *Fakten*, d. h. alle Atome A in Implikationen der Form $\top \longrightarrow A$, und zwar alle Vorkommen in der gesamten Hornformel.
2. Falls $A_1, \ldots, A_n$ in einer *Regel* $A_1 \wedge \cdots \wedge A_n \longrightarrow B$ bereits markiert sind, markiere B (ebenfalls alle Vorkommen).

3. Wiederhole 2., bis $\bot$ markiert wird, oder es nicht mehr weitergeht.

Beispiel 2.6.2 (Fortsetzung von s. o.)**.** D und dann A werden markiert.

Satz 2.6.3. Eine Hornformel F ist unerfüllbar gdw. $\bot$ markiert wird. Andernfalls ist $\mathcal{A}$ mit $\mathcal{A}(A) = 1$ gdw. A markiert wurde, ein Modell von F.

Darstellung von Modellen als Mengen von Atomen

Das Modell $\mathcal{A}$ für das obige Beispiel mit

	A	B	C	D
$\mathcal{A}$	1	0	0	1

lässt sich als Menge der markierten Atome $\{A, D\}$ darstellen.

Satz 2.6.4 (minimales Modell)**.** Sofern existent, ist das durch den Markierungsalgorithmus bestimmte Modell das (eindeutige) *minimale Modell* der Hornformel F (d. h. Minimum bzgl. $\subseteq$).

Bemerkung 2.6.5. Für aussagenlogische Formeln, insbesondere Hornformeln, kann es größere Modelle als die minimalen geben, muss es aber nicht. Für Hornformeln gibt es genau ein minimales Modell. Im Allgemeinen gilt dies jedoch nicht. Betrachte dazu $A \vee B$ (ist keine Hornformel), welches zwei minimale Modelle (d. h. minimale Elemente bzgl. $\subseteq$) hat – nämlich $\{A\}$ und $\{B\}$.

Aufwand

Jedes der n Atome wird maximal einmal markiert in der gesamten Formel F. Damit liegt der Aufwand des Markierungsalgorithmus in $O(n \cdot |F|) = O(|F|^2)$ wegen $n \leq |F|$, wobei $|F|$ die Anzahl der Symbole in der Formel F bezeichnet. Das Erfüllbarkeitsproblem für Hornformeln ist also effizient in quadratischer, d. h. polynomieller Zeit bezogen auf die Formellänge $|F|$ lösbar und liegt somit in der Komplexitätsklasse P, während das allgemeine Erfüllbarkeitsproblem (ob eine allgemeine aussagenlogische Formel erfüllbar ist, in der Literatur SAT genannt), NP-vollständig ist, was heißt, dass es nach heutigem Wissensstand algorithmisch nicht effizient lösbar ist.

Anwendungen

Die Hornlogik liefert die Grundlage für die Programmierspache *Prolog*. Hornformeln dienen bei der Spezifikation in vielen Anwendungen wie

- Expertensysteme (z.B. Diagnose bei Kfz-Reparaturen),
- Konfiguration (z.B. die Auswahl von Modulen in Betriebs- oder Lernsystemen) oder
- Recommender-Systeme (z.B. in Online-Shops).

Zusammenfassend spricht man in diesem Kontext von *Wissensbasierten Systemen*.

3 Prädikatenlogik

Was ist Prädikatenlogik?

Die *Prädikatenlogik* ist eine Erweiterung der Aussagenlogik. In der Aussagenlogik werden auf atomarer Ebene nur wahre und falsche Aussagen zugelassen. Durch diese Zweiwertigkeit der Aussagenlogik ist es nicht ohne Weiteres möglich, Aussagen über eine Vielzahl von Objekten einfach zu formalisieren, z. B. Aussagen über alle natürlichen Zahlen oder alle Lebewesen. Darum werden in der Prädikatenlogik atomare Aussagen auch hinsichtlich ihrer inneren Struktur untersucht, indem Prädikate eingeführt werden.

Prädikate

Ein *Prädikat* ist – sprachlich gesehen – eine Folge von Wörtern mit Leerstellen (Argumenten), die zu einer wahren oder falschen Aussage wird, wenn in jede Leerstelle ein Objekt eingesetzt wird. Zum Beispiel ist die Wortfolge

$$\_\_ \textit{ ist ein Pinguin}$$

ein Prädikat, weil durch Einsetzen eines Eigennamens – z. B. *Tux* – ein Aussagesatz entsteht (der wahr oder falsch sein kann), hier etwa:

$$\textit{Tux ist ein Pinguin.}$$

Die Argumente eines Prädikats, das man als Attribut oder Relation auffassen kann, werden in Formeln üblicherweise in runde Klammern gesetzt. Der obige Aussagesatz könnte also z. B. wie folgt in Prädikatenlogik formalisiert werden:

$$\mathit{Pinguin}(\mathit{Tux})$$

Variablen und Quantoren

Statt eines bestimmten (konstanten) Objekts kann in ein Prädikat auch eine *Variable* eingesetzt werden, die dann für ein bestimmtes Objekt aus einer Grundmenge steht. Variablen können nicht nur als Argumente von Prädikaten, sondern auch von Funktionen auftreten (siehe Definition 3.1.1 und 3.1.3).

Innerhalb einer Formel können Variablen mit Quantoren versehen sein. Ein *Quantor* gibt an, von wie vielen Individuen – z. B. einige oder alle – aus der Grundmenge ein Prädikat erfüllt wird. Er bindet die Variable einer Satzfunktion, so dass wieder ein Satz entsteht. Der Begriff *Pinguin* kann etwa (anders als oben) als Variable für ein Tier gleichen Namens stehen. Dann lässt sich die Aussage

$$\textit{Alle Pinguine sind Vögel.} \qquad (\star)$$

in Prädikatenlogik bezogen auf die Grundmenge aller Pinguine wie folgt mit einem Allquantor ($\forall$) formalisieren:

$$\forall\, \mathit{Pinguin}\ \mathit{Vogel}(\mathit{Pinguin})$$

Häufig betrachtet man aber größere, allgemeinere Grundmengen, z. B. die Menge aller Lebewesen. Dann kann die obige Aussage ($\star$) mit Hilfe zweier Prädikate *Pinguin* und *Vogel* wie folgt formalisiert werden:

$$\forall x\, (\mathit{Pinguin}(x) \longrightarrow \mathit{Vogel}(x))$$

Anwendungen der Prädikatenlogik

Prädikatenlogik wird zur Formulierung von Bedingungen in Computerprogrammen verwendet. In Prädikatenlogik formulierte mathematische Sätze können mittels Theorembeweisern (z. B. Implementierung des Resolutionsverfahrens, vgl. [McC94]) automatisch bewiesen werden, ebenso Programmeigenschaften (Invarianten) im Zusammenhang der Software-Verifikation. Prädikatenlogik kann außerdem mehr oder weniger direkt als vollwertige Programmiersprache verwendet werden (*Prolog*) [CM94]. Ausführlichere Darstellungen der Prädikatenlogik sowie ihrer Anwendung in der Logikprogrammierung finden sich z. B. in [CL73] und [Llo87].

3.1 Syntax

Definition 3.1.1 (Terme)**.** Seien x_i $(i \in \mathbb{N})$ *Variablensymbole* und f_i^k $(i \in \mathbb{N}, k \in \mathbb{N}_0)$ *Funktionssymbole*. So gilt:

1. Jedes x_i ist ein Term.
2. Ist f_i^k ein Funktionssymbol und $t_1, \ldots, t_k$ Terme, so ist $f_i^k(t_1, \ldots, t_k)$, falls $k \geq 1$, bzw. f_i^0, falls $k = 0$, auch ein Term.
3. Terme werden ausschließlich gemäß 1. und 2. gebildet.

Bemerkung 3.1.2. Im Fall von f_i^k bezeichnet k die *Stelligkeit* des Funktionssymbols. 0-stellige Funktionssymbole f_i^0 (ohne anschließende runde Klammern) werden als *Konstanten* bezeichnet.

Definition 3.1.3 (Formeln)**.** Seien x_i, f_i^k wie oben definiert und P_i^k $(i \in \mathbb{N}, k \in \mathbb{N}_0)$ *Prädikatensymbole*. So gilt:

1. $P_i^k(t_1, \ldots, t_k)$ für $k \geq 1$, bzw. P_i^0 für $k = 0$, ist eine Formel ($t_1, \ldots, t_k$ sind Terme).
2. Ist F_0 eine Formel, so auch $\neg F_0$.
3. Sind F_1 und F_2 Formeln, so auch $F_1 \wedge F_2$ und $F_1 \vee F_2$.
4. Ist x_i ein Variablensymbol und F_0 eine Formel, so sind auch $\exists x_i F_0$ und $\forall x_i F_0$ Formeln. Die Formel F_0 heißt *Geltungsbereich* des Quantors, der allerdings durch weitere in F_0 enthaltene Quantoren zur gleichen Variable x_i unterbrochen sein kann.

Bemerkung 3.1.4. Formeln nach 1. werden als *Atome* (atomare Formeln) bezeichnet. Formeln nach 4. heißen *quantisiert*. Wir unterscheiden:

- Existenzquantor: $\exists \mathrel{\widehat{=}}$ „es gibt bzw. existiert (mindestens ein)“
- Allquantor: $\forall \mathrel{\widehat{=}}$ „für alle“

In quantisierten Formeln heißt x_i *gebunden* (gebundene Variable). Nicht gebundene (Vorkommen von) Variablen heißen *frei*.

Übung 3.1.5. Betrachte $F = \forall x \exists y \big(P(x, z) \wedge P(z, y) \longrightarrow P(c, c)\big)$!

(a) Was sind Variablen-, Funktions- und Prädikatensymbole in F?

(b) Welche Quantoren und Operatoren kommen in F vor?

(c) Welche Variablen kommen in F frei vor?

Konventionen für Bezeichner

- Variablen: $x, y, z, \ldots$
- Funktionen: $f, g, h, \ldots$ (Stelligkeit > 0)
- Konstanten: $a, b, c, \ldots$
- Prädikate: $P, Q, R, \ldots$

Übung 3.1.6. Geben Sie für die folgende Formel zu jeder gebundenen Variablen den Geltungsbereich bzw. den zugehörigen Quantor an! Welche Variablen bzw. Vorkommen von Variablen sind frei?

$$F = \big(\forall x R(x,x)\big) \wedge \big(\forall x \forall y (R(x,y) \longrightarrow R(y,x))\big) \wedge R(y,y)$$

Bemerkung 3.1.7. Eine Variable x kann innerhalb einer Formel sowohl frei als auch gebunden vorkommen und/oder mehrere verschiedene, ggf. ineinander verschachtelte Geltungsbereiche haben, z. B. in $F = \forall x(P(x) \wedge \exists x Q(x)) \longrightarrow R(x,x)$. Um die Lesbarkeit zu erleichtern, sollten in solchen Fällen möglichst verschiedene Variablenbezeichner verwendet werden, z.B. hier $\forall x(P(x) \wedge \exists y Q(y)) \longrightarrow R(z,z)$.

Beispiel 3.1.8. Wir wollen die Funktion $V(F)$ für prädikatenlogische Formeln F definieren, welche die Menge der Variablen berechnet, die in F frei vorkommen. Dazu sind alle Fälle zu betrachten, wie eine prädikatenlogische Formel F aufgebaut sein kann (siehe Def. 3.1.3). Da Formeln ihrerseits Terme t enthalten können, definieren wir zunächst eine Hilfsfunktion $\tilde{V}(t)$ für Terme t, welche die Menge aller in t vorkommenden Variablen bestimmt, und zwar in dem Fall induktiv über den Aufbau von Termen (siehe Def. 3.1.1).

$$\tilde{V}(t) = \begin{cases} \{x\}, & \text{falls } t = x \text{ Variable} \\ \tilde{V}(t_1) \cup \cdots \cup \tilde{V}(t_n), & \text{falls } t = f(t_1, \ldots, t_n) \text{ für } n \geq 1 \\ \emptyset, & \text{falls } t = c \text{ (Konstante)} \end{cases}$$

$$V(F) = \begin{cases} \tilde{V}(t_1) \cup \cdots \cup \tilde{V}(t_n), & \text{falls } F = P(t_1, \ldots, t_n) \text{ atomar } (n \geq 0) \\ V(F_0), & \text{falls } F = \neg F_0 \\ V(F_1) \cup V(F_2), & \text{falls } F = F_1 \wedge F_2 \\ V(F_1) \cup V(F_2), & \text{falls } F = F_1 \vee F_2 \\ V(F_0) \setminus \{x\}, & \text{falls } F = \exists x F_0 \\ V(F_0) \setminus \{x\}, & \text{falls } F = \forall x F_0 \end{cases}$$

Übung 3.1.9. Berechnen Sie anhand der Definition aus Beispiel 3.1.8 $V\big(\exists y P(x,y)\big)$!

3.2 Semantik

Alle Symbole in einer Formel F sind zu interpretieren, also insbesondere auch Variablen und Konstanten. Der Formel als Ganzes soll (wie in der Aussagenlogik) ein Wahrheitswert zugewiesen werden.

Definition 3.2.1 (Struktur)**.** Eine *Struktur* $\mathcal{A} = (\mathcal{U}_\mathcal{A}, \mathcal{I}_\mathcal{A})$ besteht aus

1. einer nicht-leeren Menge $\mathcal{U}_\mathcal{A}$ (Grundmenge, Universum, Domäne oder Individuenbereich genannt) und
2. einer partiellen Funktion $\mathcal{I}_\mathcal{A}$, welche den in den Formeln vorkommenden Symbolen (Variable, Funktionen und Prädikate) eine Interpretation zuordnet. Dabei gilt:
   (a) Variablen werden durch Elemente aus der Grundmenge $\mathcal{U}_\mathcal{A}$ interpretiert:
   $$\mathcal{I}_\mathcal{A}(x) \in \mathcal{U}_\mathcal{A}$$
   (b) Funktionssymbole stehen für (totale) Funktionen mit entsprechender Stelligkeit:
   $$\mathcal{I}_\mathcal{A}(f_i^k) \in \{f \mid f : {\mathcal{U}_\mathcal{A}}^k \to \mathcal{U}_\mathcal{A}\}$$
   (c) Prädikate entsprechen k-stelligen Relationen in der Grundmenge $\mathcal{U}_\mathcal{A}$:
   $$\mathcal{I}_\mathcal{A}(P_i^k) \subseteq {\mathcal{U}_\mathcal{A}}^k$$

   Hierbei steht ${\mathcal{U}_\mathcal{A}}^k$ für $\underbrace{\mathcal{U}_\mathcal{A} \times \ldots \times \mathcal{U}_\mathcal{A}}_{\text{k-mal}}$.

Bemerkung 3.2.2.

- Alternativ kann man definieren:
  $$\mathcal{I}_\mathcal{A}(P_i^k) \in \{P \mid P : {\mathcal{U}_\mathcal{A}}^k \to \{0,1\}\}$$
  Dabei gilt:
  $$(u_1, \ldots, u_k) \in \mathcal{I}_\mathcal{A}(P_i^k) \Longleftrightarrow P(u_1, \ldots, u_k) = 1$$
- Sonderfälle:
  $$\begin{aligned} &\mathcal{I}_\mathcal{A}(f_i^0) \in \mathcal{U}_\mathcal{A} && \text{(Konstante)} \\ &\mathcal{I}_\mathcal{A}(P_i^0) \in \{0,1\} && \text{(Atom im aussagenlogischen Sinn)} \end{aligned}$$

Definition 3.2.3 (passende Struktur)**.** Eine Struktur $\mathcal{A} = (\mathcal{U}_\mathcal{A}, \mathcal{I}_\mathcal{A})$ heißt *passend* zu einer Formel F gdw. $\mathcal{I}_\mathcal{A}$ (mindestens) für alle freien Variablen und die Funktions- und Prädikatensymbole, die in F vorkommen, definiert ist.

Schreibabkürzungen

Wir schreiben oft kürzer:

$P^{\mathcal{A}}$ statt $\mathcal{I}_{\mathcal{A}}(P)$

$f^{\mathcal{A}}$ statt $\mathcal{I}_{\mathcal{A}}(f)$

$x^{\mathcal{A}}$ statt $\mathcal{I}_{\mathcal{A}}(x)$

Die Indizes ${}^{k}_{i}$ werden außerdem häufig weggelassen.

Beispiel 3.2.4. Wir betrachten die Formel

$$F = \forall x \forall y \big(P(x,y) \longrightarrow Q(f(x,y), f(g(x,y),y)) \big)$$

und die Struktur $\mathcal{A} = (\mathcal{U}_{\mathcal{A}}, \mathcal{I}_{\mathcal{A}})$ mit $\mathcal{U}_{\mathcal{A}} = \mathbb{N}_0$ und $\mathcal{I}_{\mathcal{A}}$ wie folgt:

	P	Q	f	g	x
$\mathcal{I}_{\mathcal{A}}$	$>$	$=$	ggT	$\ominus$	5

mit $x \ominus y = \begin{cases} x - y, & \text{falls } x > y \\ 0, & \text{sonst} \end{cases}$ (modifizierte Subtraktion)

Wir stellen fest:

- $\mathcal{A}$ ist *passend* zu F.
- x und y sind nicht frei.

Übung 3.2.5. Übersetzen Sie die Formel F aus dem vorigen Beispiel 3.2.4 in die übliche mathematische Notation! Für welche Aussage steht also die Formel F? Handelt es sich um eine wahre Aussage?

Definition 3.2.6 (Semantik)**.**

1. Sei t ein Term.
   (a) $t = x \Longrightarrow \mathcal{A}(t) = x^{\mathcal{A}}$
   (b) $t = f(t_1, \ldots, t_n) \Longrightarrow \mathcal{A}(t) = f^{\mathcal{A}}(\mathcal{A}(t_1), \ldots, \mathcal{A}(t_n))$
2. Sei F eine Formel.
   (a) $F = P(t_1, \ldots, t_n)$ atomar $\Longrightarrow \mathcal{A}(F) = \begin{cases} 1, & \text{falls } (\mathcal{A}(t_1), \ldots, \mathcal{A}(t_n)) \in P^{\mathcal{A}} \\ 0, & \text{sonst} \end{cases}$
   (b) $\neg, \vee, \wedge$ werden wie in der Aussagenlogik behandelt (siehe Def. 2.1.3)
   (c) $F = \exists x G \Longrightarrow \mathcal{A}(F) = \begin{cases} 1, & \text{falls es ein } d \in \mathcal{U}_{\mathcal{A}} \text{ gibt mit } \mathcal{A}_{[x/d]}(G) = 1 \\ 0, & \text{sonst} \end{cases}$
   (d) $F = \forall x G \Longrightarrow \mathcal{A}(F) = \begin{cases} 1, & \text{falls für alle } d \in \mathcal{U}_{\mathcal{A}} \text{ gilt: } \mathcal{A}_{[x/d]}(G) = 1 \\ 0, & \text{sonst} \end{cases}$

Hierbei stimmt $\mathcal{A}' = \mathcal{A}_{[x/d]}$ mit $\mathcal{A}$ überein, bis auf die Definition von $x^{\mathcal{A}}$. Es ist nämlich $x^{\mathcal{A}'} = d \in \mathcal{U}_{\mathcal{A}} = \mathcal{U}_{\mathcal{A}'}$ unabhängig davon, ob $\mathcal{I}_{\mathcal{A}}$ für x bereits definiert ist oder nicht. Der Wert von x wird durch d überschrieben.

Beispiel 3.2.7 (Fortsetzung von Beispiel 3.2.4)**.**

$\mathcal{A}(g(x,x)) = g^{\mathcal{A}}(\mathcal{A}(x), \mathcal{A}(x)) = x^{\mathcal{A}} \ominus x^{\mathcal{A}} = 5 - 5 = 0$

$\mathcal{A}(\forall x P(x,x)) = 0$, da nicht für alle $x^{\mathcal{A}} \in \mathcal{U}_{\mathcal{A}}$ $P^{\mathcal{A}}(x^{\mathcal{A}}, x^{\mathcal{A}})$ bzw. $x^{\mathcal{A}} > x^{\mathcal{A}}$ gilt.

$\mathcal{A}(F) = 1$, da für alle $x^{\mathcal{A}}, y^{\mathcal{A}} \in \mathbb{N}_0$ gilt:

$$x^{\mathcal{A}} > y^{\mathcal{A}} \Longrightarrow ggT(x^{\mathcal{A}}, y^{\mathcal{A}}) = ggT(x^{\mathcal{A}} \ominus y^{\mathcal{A}}, y^{\mathcal{A}})$$

Bemerkung 3.2.8. Terme werden durch Elemente aus dem Universum $\mathcal{U}_{\mathcal{A}}$ interpretiert, Formeln durch Wahrheitswerte (0 oder 1).

Übung 3.2.9. Berechnen Sie unter Verwendung der Struktur aus Beispiel 3.2.4:

(a) $\mathcal{A}\big(f(x,x)\big)$

(b) $\mathcal{A}\big(\exists y P(x,y)\big)$

(c) $\mathcal{A}\big(\forall x \exists y P(y,x)\big)$

(d) $\mathcal{A}\big(\exists y \forall x P(y,x)\big)$

Übung 3.2.10. Sei $\mathcal{U}_{\mathcal{A}} = \mathbb{N}$, $R^{\mathcal{A}} = \{x \mid x \text{ ist gerade Zahl}\}$ sowie $x^{\mathcal{A}} = 5$.
Bestimmen Sie die Wahrheitswerte folgender Formeln:

(a) $R(x)$

(b) $\exists x\, R(x)$

(c) $\forall x\, R(x)$

(d) $\exists x R(x) \vee \exists x \neg R(x)$

(e) $\exists x (R(x) \vee \neg R(x))$

(f) $\exists x R(x) \wedge \exists x \neg R(x)$

(g) $\exists x (R(x) \wedge \neg R(x))$

Bemerkung 3.2.11.

- 0-stellige Prädikate entsprechen den aussagenlogischen Konstanten (Atome).
- 1-stellige Prädikate nennt man Eigenschaften oder Attribute, z. B. *gerade*.
- 2-stellige Prädikate entsprechen binären Relationen, z. B. $=$, $<$, die meist in Infix-Notation dargestellt werden, d. h. in der Form $x = y$, selten in Präfix-Notation: $=(x,y)$

Bemerkung 3.2.12. Die ausschließliche Verwendung nullstelliger Prädikate bewirkt eine Degenerierung der Prädikaten- zur Aussagenlogik. Prädikatenlogik ist zwar ausdruckskräftiger als Aussagenlogik, doch auch mit ihr lässt sich nicht jede mathematische Aussage formalisieren. So kann man Quantoren der Art *genau ein* oder *mindestens zwei* erst durch Einführung der Identitätsrelation (=) formulieren. Außerdem fehlen Quantifizierungen über Prädikaten- oder Funktionssymbole. Dies führt zur Prädikatenlogik höherer Stufen.

Definition 3.2.13 (Modelle)**.** Sei F eine prädikatenlogische Formel und $\mathcal{A}$ eine dazu passende Struktur. $\mathcal{A}$ heißt *Modell* von F, i. Z. $\mathcal{A} \models F$, falls $\mathcal{A}(F) = 1$. Die Begriffe *erfüllbar*, *unerfüllbar*, *gültig*, *Tautologie* sowie Folgerung aus Def. 2.1.9 werden samt Schreibweisen ganz analog auf prädikatenlogische Formeln angewendet.

Beispiel 3.2.14 (Fortsetzung Beispiel 3.2.4)**.** $\mathcal{A}$ ist Modell von F (vgl. Beispiel 3.2.7).

3.3 Das Überführungslemma

Definition 3.3.1 (Substitution in Formeln bzw. Termen)**.** Sei F eine Formel, x eine Variable und t ein Term. Dann bezeichnet $F[x/t]$ diejenige Formel, die man aus F erhält, indem jedes *freie* Vorkommen von x in F durch t ersetzt wird. Durch $[x/t]$ wird eine *Substitution* beschrieben (vgl. Def. 3.7.2 f). Eine Substitution kann statt auf Formeln F auch auf Terme s angewendet werden.

Übung 3.3.2. Sei $F = P(x, y)$ und $\mathcal{A} = (\mathcal{U}_\mathcal{A}, \mathcal{I}_\mathcal{A})$ mit $\mathcal{U}_\mathcal{A} = \mathbb{N}_0$ und $\mathcal{I}_\mathcal{A}$ wie folgt:

	P	x	y
$\mathcal{I}_\mathcal{A}$	$>$	3	2

Bestimmen Sie (a) $F[x/y]$, (b) $\mathcal{A}(F)$, (c) $\mathcal{A}(F[x/y])$ und (d) $\mathcal{A}_{[x/\mathcal{A}(y)]}(F)$!

Lemma 3.3.3 (Überführungslemma)**.** Sei F eine Formel, $[x/t]$ eine Substitution und $\mathcal{A}$ eine zu F und $F[x/t]$ passende Struktur. Dann gilt:

$$\mathcal{A}(F[x/t]) = \mathcal{A}_{[x/\mathcal{A}(t)]}(F) \tag{1}$$

Dabei muss allerdings die folgende *Variablenbedingung* gelten: Eine in t vorkommende Variable wird durch die Substitution $[x/t]$ in F nicht gebunden.
Hinweis: Die linke Seite der obigen Gleichung beschreibt eine *syntaktische Ersetzung* (Substitution). Die rechte Seite ist eine *semantische Modifikation*.

Beispiel 3.3.4.

$$\begin{aligned} F &= \exists y P(x, y) \\ F[x/y] &= \exists y P(y, y) \end{aligned}$$

Durch die Substitution wird das neue y in $F[x/y]$ gebunden! Die obige Variablenbedingung gilt also nicht. Somit ist das Überführungslemma in diesem Fall nicht anwendbar. Die Wahrheitswerte von F und $F[x/y]$ sind i. A. verschieden.

Der Beweis des Überführungslemmas (s. u.) kann als typischer Induktionsbeweis über die Struktur von Formeln (siehe Def. 3.1.3) angesehen werden. Durch die vielfachen Fallunterscheidungen ist er jedoch recht umfangreich. Da Formeln Terme enthalten können, und Terme selbst wieder induktiv definiert sind (siehe Def. 3.1.1), sollte die zu beweisende Behauptung auch für Terme formuliert werden. Der folgende Hilfssatz besagt daher genau das Gleiche wie das Lemma 3.3.3, nur eben für Terme statt für Formeln.

Lemma 3.3.5 (Hilfssatz)**.** Sei s ein Term. Dann gilt:

$$\mathcal{A}(s[x/t]) = \mathcal{A}_{[x/\mathcal{A}(t)]}(s) \tag{2}$$

Beweis (Induktion über den Termaufbau)**:** Wir müssen alle Möglichkeiten in Betracht ziehen, wie der Term s aufgebaut sein kann (gemäß Def. 3.1.1).

1. s ist eine Variable. Wir unterscheiden weiter – das darf man, solange die Unterscheidung alle Möglichkeiten abdeckt:

   (a) $s = x$. Wir setzen in die linke Seite von (2) ein und formen um:

   $$\mathcal{A}(x[x/t]) \overset{\text{Def. 3.3.1}}{=} \mathcal{A}(t) \overset{\text{Def. 3.2.6}}{=} \mathcal{A}_{[x/\mathcal{A}(t)]}(x)$$

   (b) $s \neq x$, d. h. s ist eine von x verschiedene Variable $s = y$. Wir setzen wiederum in die linke Seite von (2) ein und formen um:

   $$\mathcal{A}(y[x/t]) \overset{\text{Def. 3.3.1}}{=} \mathcal{A}(y) \overset{\text{Def. 3.2.6}}{=} \mathcal{A}_{[x/\mathcal{A}(t)]}(y)$$

2. s ist ein Funktionsterm $s = f(s_1, \ldots, s_n)$. Eingesetzt in die linke Seite von (2) ergibt das:

$$\begin{aligned}
\mathcal{A}(f(s_1, \ldots, s_n)[x/t]) &\overset{\text{Def. 3.3.1}}{=} \mathcal{A}(f(s_1[x/t], \ldots, s_n[x/t])) \\
&\overset{\text{Def. 3.2.6}}{=} f^{\mathcal{A}}(\mathcal{A}(s_1[x/t]), \ldots, \mathcal{A}(s_n[x/t])) \\
&\overset{\text{I.V.}}{=} f^{\mathcal{A}}(\mathcal{A}_{[x/\mathcal{A}(t)]}(s_1), \ldots, \mathcal{A}_{[x/\mathcal{A}(t)]}(s_n)) \\
&\overset{f^{\mathcal{A}} = f^{\mathcal{A}_{[x/\mathcal{A}(t)]}}}{=} f^{\mathcal{A}_{[x/\mathcal{A}(t)]}}(\mathcal{A}_{[x/\mathcal{A}(t)]}(s_1), \ldots, \mathcal{A}_{[x/\mathcal{A}(t)]}(s_n)) \\
&\overset{\text{Def. 3.2.6}}{=} \mathcal{A}_{[x/\mathcal{A}(t)]}(f(s_1, \ldots, s_n))
\end{aligned}$$

Der Fall, dass s eine Konstante ist, ist bereits mit $n = 0$ im letzten Fall enthalten. Damit ist der Hilfssatz bewiesen. ⊠

Beweis (von Lemma 3.3.3, durch Induktion über den Formelaufbau)**:** Wir müssen alle Möglichkeiten in Betracht ziehen, wie die Formel F aufgebaut sein kann.

1. F ist ein Atom $F = P(s_1, \ldots, s_n)$. Das geht analog zum Fall 2. des Hilfssatzes, nur dass statt der Induktionsvoraussetzung der Hilfssatz (Lemma 3.3.5) benutzt wird. Wir setzen diesmal in die linke Seite von (1) ein:

$$\begin{aligned}
\mathcal{A}(P(s_1, \ldots, s_n)[x/t]) &\overset{\text{Def. 3.3.1}}{=} \mathcal{A}(P(s_1[x/t], \ldots, s_n[x/t])) \\
&\overset{\text{Def. 3.2.6}}{=} (\mathcal{A}(s_1[x/t]), \ldots, \mathcal{A}(s_n[x/t])) \in P^{\mathcal{A}} \\
&\overset{\text{Lemma 3.3.5}}{=} (\mathcal{A}_{[x/\mathcal{A}(t)]}(s_1), \ldots, \mathcal{A}_{[x/\mathcal{A}(t)]}(s_n)) \in P^{\mathcal{A}} \\
&\overset{P^{\mathcal{A}} = P^{\mathcal{A}_{[x/\mathcal{A}(t)]}}}{=} (\mathcal{A}_{[x/\mathcal{A}(t)]}(s_1), \ldots, \mathcal{A}_{[x/\mathcal{A}(t)]}(s_n)) \in P^{\mathcal{A}_{[x/\mathcal{A}(t)]}} \\
&\overset{\text{Def. 3.2.6}}{=} \mathcal{A}_{[x/\mathcal{A}(t)]}(P(s_1, \ldots, s_n))
\end{aligned}$$

2. F ist eine Formel $F = \neg F_0$.

$$\begin{aligned}
\mathcal{A}((\neg F_0)[x/t]) &\overset{\text{Def. 3.3.1}}{=} \mathcal{A}(\neg(F_0[x/t])) \\
&\overset{\text{Def. 3.2.6}}{=} 1 - \mathcal{A}(F_0[x/t]) \\
&\overset{\text{I.V.}}{=} 1 - \mathcal{A}_{[x/\mathcal{A}(t)]}(F_0) \\
&\overset{\text{Def. 3.2.6}}{=} \mathcal{A}_{[x/\mathcal{A}(t)]}(\neg F_0)
\end{aligned}$$

3. F ist eine Formel $F = F_1 \wedge F_2$. Ähnlich 2., aus Platzgründen weggelassen.

4. F ist eine Formel $F = F_1 \vee F_2$. Analog zu 3.

5. F ist eine allquantifizierte Formel. Wir unterscheiden weiter:

   (a) $F = \forall x F_0$. Das heißt, die allquantifizierte Variable ist identisch mit x, das somit in F nicht frei vorkommt.

   $$\mathcal{A}((\forall x F_0)[x/t]) \overset{\text{Def. 3.3.1}}{=} \mathcal{A}(\forall x F_0) \overset{\text{Def. 3.2.6}}{=} \mathcal{A}_{[x/\mathcal{A}(t)]}(\forall x F_0)$$

   (b) $F = \forall y F_0$ und $y \neq x$. Das heißt, die allquantifizierte Variable ist nicht identisch mit dem x aus der Substitution $[x/t]$.

   $$\begin{aligned}
   \mathcal{A}((\forall y F_0)[x/t]) = 1 &\overset{x \neq y}{\iff} \mathcal{A}(\forall y(F_0[x/t])) = 1 \\
   &\overset{\text{Def. 3.2.6}}{\iff} \forall v \in \mathcal{U}_\mathcal{A} : \mathcal{A}_{[y/v]}(F_0[x/t]) = 1 \\
   &\overset{\text{I.V.}}{\iff} \forall v \in \mathcal{U}_\mathcal{A} : \mathcal{A}_{[y/v][x/\mathcal{A}_{[y/v]}(t)]}(F_0) = 1 \\
   &\overset{\text{Variablenbedingung}}{\iff} \forall v \in \mathcal{U}_\mathcal{A} : \mathcal{A}_{[y/v][x/\mathcal{A}(t)]}(F_0) = 1 \\
   &\overset{x \neq y}{\iff} \forall v \in \mathcal{U}_\mathcal{A} : \mathcal{A}_{[x/\mathcal{A}(t)][y/v]}(F_0) = 1 \\
   &\overset{\text{Def. 3.2.6}}{\iff} \mathcal{A}_{[x/\mathcal{A}(t)]}(\forall y F_0) = 1
   \end{aligned}$$

6. F ist eine existentiell quantifizierte Formel. Analog zu 5.

Damit sind alle Aufschichtungsmöglichkeiten für Formeln erschöpft, und das Überführungslemma ist bewiesen. ⊠

3.4 Äquivalenz von Formeln

Definition 3.4.1 (Äquivalenz in der Prädikatenlogik)**.** $F \equiv G$ gdw. $\mathcal{A}(F) = \mathcal{A}(G)$ für alle zu F und G passenden Strukturen $\mathcal{A}$.

Satz 3.4.2 (prädikatenlogische Äquivalenzen)**.**

1. Sämtliche aussagenlogischen Äquivalenzen (Satz 2.2.6) sowie das Ersetzbarkeitstheorem (Satz 2.2.4) gelten.

2. Zusätzlich gibt es folgende Regeln für Quantoren:

Umbenennung von Variablen: $\exists x\, F \equiv \exists y\, F[x/y]$

$$\forall x\, F \equiv \forall y\, F[x/y]$$

Hinweis: Die zu ersetzenden Vorkommen von x dürfen sich nicht im Geltungsbereich von quantifizierten y befinden, das darüber hinaus nicht freie Variable sein darf.

Quantoren-Gesetze:

$$\exists x\, \neg F \equiv \neg \forall x\, F$$

$$\forall x\, \neg F \equiv \neg \exists x\, F$$

$$\exists x \exists y\, F \equiv \exists y \exists x\, F$$

$$\forall x \forall y\, F \equiv \forall y \forall x\, F$$

$$\exists x\, F \vee \exists x\, F' \equiv \exists x\, (F \vee F')$$

$$\forall x\, F \wedge \forall x\, F' \equiv \forall x\, (F \wedge F')$$

Falls x nicht (frei) in F vorkommt:

$$\exists x\, F \equiv F$$

$$\forall x\, F \equiv F$$

$$\exists x\, (F \wedge F') \equiv F \wedge \exists x\, F'$$

$$\forall x\, (F \wedge F') \equiv F \wedge \forall x\, F'$$

$$\exists x\, (F \vee F') \equiv F \vee \exists x\, F'$$

$$\forall x\, (F \vee F') \equiv F \vee \forall x\, F'$$

Im Allgemeinen gilt <u>nicht</u>:

$$\forall x \exists y\, F \equiv \exists y \forall x\, F$$

$$\forall x\, F \vee \forall x\, F' \equiv \forall x\, (F \vee F')$$

$$\exists x\, F \wedge \exists x\, F' \equiv \exists x\, (F \wedge F')$$

Beweis (exemplarisch für $\exists x \neg F \equiv \neg \forall x F$)**:**
Sei $\mathcal{A}$ eine zu beiden Seiten von $\equiv$ passende Struktur. $\mathcal{A}(\exists x \neg F) = 1 \iff$ es gibt $d \in \mathcal{U}_\mathcal{A}$ mit $\mathcal{A}_{[x/d]}(\neg F) = 1 \iff$ es gibt $d \in \mathcal{U}_\mathcal{A}$ mit $\mathcal{A}_{[x/d]}(F) = 0 \iff$ nicht für alle $d \in \mathcal{U}_\mathcal{A}$ gilt $\mathcal{A}_{[x/d]}(F) = 1 \iff \mathcal{A}(\neg \forall x F) = 1$. ⊠

Bemerkung 3.4.3. Die Reihenfolge gleichartiger Quantoren ist beliebig vertauschbar. Es gilt sowohl $\exists x \exists y F \equiv \exists y \exists x F$ als auch $\forall x \forall y F \equiv \forall y \forall x F$. Die Reihenfolge von verschiedenen Quantoren wirkt sich dagegen i. A. auf die Semantik aus (vgl. Übung 3.2.9).

Faustregel zur Formalisierung natürlichsprachlicher Aussagen in Prädikatenlogik

Bei der Formalisierung natürlichsprachlicher Aussagen in Prädikatenlogik werden i.d.R.

- universelle Aussagen mit Allquantoren ($\forall$) und Implikation ($\longrightarrow$) und
- Existenz-Aussagen mit Existenzquantoren ($\exists$) und Konjunktion ($\wedge$)

ausgedrückt.

Beispiel:

- *Alle Vögel können fliegen.* $\rightsquigarrow \forall x(\mathit{Vogel}(x) \longrightarrow \mathit{fliegen}(x))$
- *Es gibt Vögel, die nicht fliegen können.* $\rightsquigarrow \exists x(\mathit{Vogel}(x) \wedge \neg \mathit{fliegen}(x))$

Hierbei ist eine Struktur $\mathcal{A}$ mit dem Universum $\mathcal{U}_\mathcal{A}$, der Menge aller Lebewesen, und zwei Prädikatensymbolen *Vogel* und *fliegen* mit den nachfolgenden Bedeutungen unterstellt:

1. $\mathit{Vogel}(x) \mathrel{\widehat{=}} x$ ist ein Vogel
2. $\mathit{fliegen}(x) \mathrel{\widehat{=}} x$ kann fliegen

Übung 3.4.4. Zeigen Sie formal, dass im obigen Beispiel die Negation der universellen Aussage gleichbedeutend ist mit der Existenz-Aussage, indem Sie Umformungen auf Grundlage der Äquivalenzen aus Satz 3.4.2 durchführen!

3.5 Normalformen

Idee: Wie in der Aussagenlogik kann man auch für prädikatenlogische Formeln Normalformen definieren. Ziel ist es hier, eine der KNF (vgl. Def. 2.3.3) verwandte Normalform zu erzeugen, auf die man das – für die Prädikatenlogik erweiterte – Resolutionsverfahren anwenden kann (siehe Def. 3.8.1). Dazu werden zunächst die Quantoren in einer Formel vorne in einem Block zusammengefasst, und der Rest der Formel, genannt Matrix, kann dann in eine KNF umgeformt werden (vgl. Def. 3.5.4).

Lemma 3.5.1 (gebundene Umbenennung)**.** Sei $F = Qx\,G$ eine Formel mit $Q \in \{\exists, \forall\}$ und y eine Variable, die in G nicht vorkommt. Dann ist:

$$F \equiv Qy\,G[x/y]$$

Beweis: Anwendung von Satz 3.4.2 (Umbenennung von Variablen). ⊠

Bemerkung 3.5.2. Hintergrund der gebundenen Umbenennung ist die Verwendung verschiedener Bezeichner in der Mathematik bzw. verschiedener Geltungssbereiche in der Informatik (lokale Variablen). Ziel ist es, dass jedes Variablensymbol genau eine Bedeutung innerhalb der gesamten Formel hat.

Lemma 3.5.3 (bereinigte Form)**.** Zu jeder Formel F gibt es eine äquivalente Formel in *bereinigter Form*, d. h. hinter allen Quantoren stehen verschiedene Variablen und keine Variable kommt sowohl frei als auch gebunden vor.

Beweis: Mehrfachanwendung von Lemma 3.5.1. ⊠

Definition 3.5.4 (Pränexform, Matrix)**.** Eine Formel heißt *pränex* oder in *Pränexform*, falls sie die Gestalt

$$Q_1 x_1 Q_2 x_2 \cdots Q_n x_n F$$

hat mit den Variablen x_i und den Quantoren $Q_i \in \{\exists, \forall\}$ für $0 \leq i \leq n$. F enthält keine weiteren Quantoren und heißt in diesem Fall *Matrix*.

Satz 3.5.5. Ist F eine Formel, so gibt es eine zu F äquivalente, bereinigte Formel G in Pränexform.

Beweis (strukturelle Induktion über Aufbau prädikatenlogischer Formeln)**:**

- Induktionsanfang: F atomar – ist klarerweise pränex.
- Induktionsschritt: Fallunterscheidung:

  $F = \neg F_0$**:** Nach I.V. existiert Pränexform $Q_1 x_1 \cdots Q_n x_n \, F_0' \equiv F_0$

  $\Longrightarrow F \equiv \overline{Q_1} x_1 \cdots \overline{Q_n} x_n \neg F_0'$ gilt wegen Äquivalenzen $\neg \forall x F \equiv \exists x \neg F$ und $\neg \exists x F \equiv \forall x \neg F$ (siehe Satz 3.4.2, Quantoren-Gesetze). $\overline{Q}$ ist hierbei folgendermaßen definiert:

Q	$\exists$	$\forall$
$\overline{Q}$	$\forall$	$\exists$

  $F = F_1 \vee F_2$**:** Nach I.V. existieren Pränexformen

  $$Q_1 x_1 \cdots Q_n x_n F_1' \equiv F_1$$

  $$Q_1^* x_1^* \cdots Q_m^* x_m^* F_2' \equiv F_2$$

  Ohne Beschränkung der Allgemeinheit seien $x_1, \ldots, x_n$ und $x_1^*, \ldots, x_n^*$ verschieden (ggf. Lemma 3.5.1 anwenden).

  $\Longrightarrow F \equiv Q_1 x_1 \cdots Q_n x_n Q_1^* x_1^* \cdots Q_n^* x_n^* \, (F_1' \vee F_2')$
  (wegen $\exists x (F \vee F') \equiv (\exists x F) \vee F'$ etc.)

  $F = F_1 \wedge F_2$**:** analog zu $F = F_1 \vee F_2$

  $F = \forall x F_0$**:** Nach I.V. existiert Pränexform $Q_1 x_1 \cdots Q_n x_n F_0'$ zu F_0.

  $\Longrightarrow$ Man erhält durch adäquate gebundene Umbenennung:

  $$F \equiv \forall y \left(Q_1 x_1 \cdots Q_n x_n F_0' \, [x/y] \right)$$

  $F = \exists x F_0$**:** analog zu $F = \forall x F_0$.

Bemerkung: Da Terme hier keine Rolle spielen, muss die Induktion nicht über den Termaufbau fortgesetzt werden. ⊠

Bemerkung 3.5.6. Bei der Erzeugung der Pränexform zieht man am besten alle Quantoren der Reihe nach von links nach rechts nach vorne. Im Wirkungsbereich von Negationen ($\neg$) sind jedoch unbedingt die Quantoren-Gesetze (Satz 3.4.2) zu beachten. Dies gilt auch für Implikationen ($\longrightarrow$), die implizit auf der linken Seite eine Negation enthalten (vgl. Def. 2.2.5).

Übung 3.5.7. Zeigen Sie

$$\exists x \left((\neg \forall x P(x)) \longrightarrow P(x) \right) \equiv \exists x \forall y \left((\neg P(y)) \longrightarrow P(x) \right)$$

durch Äquivalenzumformungen gemäß Satz 3.4.2! Was geschieht bei der Erzeugung der Pränexform (allgemein) mit den Quantoren im Wirkungsbereich einer (i) geraden bzw. (ii) ungeraden Zahl von Negationen? Was ist die Matrix der rechts stehenden Formel?

Skolemisierung

Falls man nur eine bezüglich Erfüllbarkeit äquivalente Formel sucht, gibt es eine noch einfachere Normalform, die *Skolemform*. Sie enthält nur eine einzige Quantoren-Art, nämlich nur noch Allquantoren ($\forall$) und keine Existenzquantoren ($\exists$) mehr.

Definition 3.5.8 (Skolemform)**.** Eine Formel F ist in *Skolemform*, falls sie die Gestalt $\forall y_1 \cdots \forall y_n F^*$ hat mit $n \geq 0$, wobei F^* die Matrix der Formel F ist und daher keine weiteren Quantoren enthält (vgl. Def. 3.5.4).

Satz 3.5.9 (Erfüllbarkeitsäquivalenz für Skolemformen)**.** Für jede Formel F gibt es eine Formel F' in Skolemform, die erfüllbar ist gdw. F erfüllbar ist. Diese Äquivalenz (nur) bezüglich Erfülbarkeit ist eine schwächere Eigenschaft als semantische Äquivalenz, d.h. es gilt i. A. nicht $F \equiv F'$.

Der folgende Algorithmus erzeugt zu einer Formel F in bereinigter Pränexform eine zugehörige Skolemform F'. Er eliminiert alle durch einen Existenzquantor gebundenen Variablen y und ersetzt in der Matrix F^* alle Vorkommen von y durch einen Funktionsausdruck $f(x_1, \ldots, x_k)$ mit:

- f ist ein neues, bisher in der ganzen Formel noch nicht vorkommendes Funktionssymbol mit Stelligkeit $k \geq 0$.
- Die Argumente von f sind dabei genau die mit einem Allquantor gebundenen Variablen $x_1, \ldots, x_k$, die in F vor dem zu eliminierenden Existenzquantor $\exists y$ stehen.
- Für $k = 0$ wird die Variable y durch eine Konstante c ersetzt, die in diesem Zusammenhang *Skolemkonstante* genannt wird.

Algorithmus zur Erzeugung der Skolemform

Eingabe: F in bereinigter Pränexform
Ausgabe: skolemisierte Formel F' mit Matrix F^*

$F' = F$
repeat
  let $F' = Q_1 y_1 \cdots Q_n y_n \, F^*$
    $i = \min(\{r \mid Q_r = \exists\} \cup \{\infty\})$ ▷ Position des vordersten $\exists$ in F'
    if $i \neq \infty$ **then**
      let f = neues $(i-1)$-stelliges Funktionssymbol
        $F' = \forall y_1 \cdots \forall y_{i-1} Q_{i+1} y_{i+1} \cdots Q_n y_n \, F^*[y_i / f(y_1, \ldots, y_{i-1})]$
        ▷ $\exists$ Eliminierung
      end let
    end if
  end let
  $\{F'$ erfüllbar $\iff F$ erfüllbar$\}$ ▷ Schleifeninvariante (noch zu beweisen)
until $i = \infty$

Beweis (der Schleifeninvariante)**:**

$\boxed{\Longrightarrow}$ Sei $F' = \forall x_1 \cdots \forall x_{i-1} Q_{i+1} x_{i+1} \cdots Q_n x_n \, F^*[x_i / f(x_1, \ldots, x_{i-1})]$ die Formel nach einem Schleifendurchlauf. $\mathcal{A}$ sei eine zu F' passende Struktur mit $\mathcal{A}(F') = 1$. Dann gilt für alle $u_1, \ldots, u_{i-1} \in \mathcal{U}_\mathcal{A}$

$$\mathcal{A}_{[x_1/u_1]\cdots[x_{i-1}/u_{i-1}]} \left(Q_{i+1} x_{i+1} \cdots Q_n x_n \, F^*[x_i / f(x_1, \ldots, x_{i-1})] \right) = 1$$

gdw. für alle $u_1, \ldots, u_{i-1} \in \mathcal{U}_\mathcal{A}$

$$\mathcal{A}_{[x_1/u_1]\cdots[x_{i-1}/u_{i-1}][x_i/\mathcal{A}(f(u_1,\ldots,u_{i-1}))]} (\underbrace{Q_{i+1} x_{i+1} \cdots Q_n x_n \, F^*}_{\tilde{F}}) = 1$$

(mit Lemma 3.3.3) gdw. es für alle $u_1, \ldots, u_{i-1} \in \mathcal{U}_\mathcal{A}$ ein $v \in \mathcal{U}_\mathcal{A}$ gibt, nämlich z. B. $v = \mathcal{A}(f(u_1, \ldots, u_{i-1}))$ mit:

$$\mathcal{A}_{[x_1/u_1]\cdots[x_{i-1}/u_{i-1}][x_i/v]}(\tilde{F}) = 1 \Longleftrightarrow \mathcal{A}(\forall x_1 \cdots \forall x_{i-1} \exists x_i \, \tilde{F}) = 1 \Longleftrightarrow \mathcal{A}(F) = 1$$

Bemerkung: F und F' sind mit gleichem Modell $\mathcal{A}$ erfüllbar.

$\boxed{\Longleftarrow}$ Sei nun F die Formel vor dem Schleifendurchlauf und $\mathcal{A}$ eine zu F passende Struktur mit $\mathcal{A}(F) = 1$. $x_1, \ldots, x_{i-1}$ sind bereits ausschließlich durch Allquantoren gebunden. Dann gilt für alle $u_1, \ldots, u_{i-1} \in \mathcal{U}_\mathcal{A}$: Es gibt ein $v \in \mathcal{U}_\mathcal{A}$ mit $\mathcal{A}_{[x_1/u_1]\cdots[x_{i-1}/u_{i-1}][x_i/v]}(\tilde{F}) = 1$.

Wir erweitern $\mathcal{A}$ zu $\mathcal{A}'$ mit einem zusätzlichen, neuen Funktionssymbol f und setzen $f^{\mathcal{A}'}(u_1, \ldots, u_{i-1}) = v$ für alle $u_1, \ldots, u_{i-1} \in \mathcal{U}_\mathcal{A}$. Hierbei wird das sogenannte *Auswahlaxiom* angewendet, welches äquivalent zum Wohlordnungssatz (Axiom 1.6.7) ist. Der Beweis kann nun wie im Fall $\boxed{\Longrightarrow}$ fortgesetzt werden, nur rückwärts.

Bemerkung: $\mathcal{A}$ und $\mathcal{A}'$ sind verschieden. Daher liegt nur Erfüllbarkeitsäquivalenz vor. $\mathcal{A}$ darf f nicht interpretieren, sollte also minimal gewählt sein. ⊠

Beispiel 3.5.10 (vgl. Übung 3.2.9)**.** Betrachte $F = \forall x \exists y P(y, x)$. Eine zugehörige Skolemform ist:

$$F' = \forall x \big(P(y, x)[y / f(x)] \big) = \forall x \underbrace{P(f(x), x)}_{\text{Matrix } F^*}$$

Bezogen auf die Struktur $\mathcal{A} = (\mathcal{U}_\mathcal{A}, \mathcal{I}_\mathcal{A})$ mit $\mathcal{U}_\mathcal{A} = \mathbb{N}$ und $P = (>)$ (aus Beispiel 3.2.4) gilt $\mathcal{A} \models F$, da für alle $u \in \mathbb{N}$ z. B. $v = u + 1$ existiert mit $v > u$. v wäre aber auch z. B. als 2^u wählbar. Wie kann also $f^{\mathcal{A}'}$ gewählt werden, damit $\mathcal{A}'(F') = 1$ gilt?

$$f^{\mathcal{A}'}(n) = n + 1 \text{ oder auch } f^{\mathcal{A}'}(n) = 2^n$$

Übung 3.5.11. Eine Skolemform zu $G = \exists y \forall x P(y, x)$ ist – wie zu erwarten (vgl. Übung 3.2.9)– nicht äquivalent zu F bzw. F' aus Beispiel 3.5.10 (nichtsdestotrotz aber erfüllbarkeitsäquivalent zu G). Wie sieht eine Skolemform G' zu G aus?

Übung 3.5.12. Skolemisieren Sie die Formel $F = \forall x \exists y \exists z P(x, f(y, z))$!

Zusammenfassung: Prädikatenlogik – Umformung in Normalform

Wir können nun ein Verfahren zur Erzeugung einer Normalform für prädikatenlogische Formeln F angeben. Die Matrix der entstehenden Formel ist dabei in Konjunktiver Normalform, so dass wir wie in der Aussagenlogik eine letztendlich eine Darstellung als Klauselmenge erhalten.

Eingabe: prädikatenlogische Formel F

$$F = (\forall x\, P(x, f(x))) \wedge (\neg\forall y(P(y, x) \wedge R(y)))$$

1. Bereinige F durch gebundenes Umbenennen!

$$(\forall z\, P(z, f(z))) \wedge (\neg\forall y\, (P(y, x) \wedge R(y)))$$

2. Binde freie Variablen durch Existenzquantoren!*

$$\exists x\, ((\forall z\, P(z, f(z))) \wedge (\neg\forall y(P(y, x) \wedge R(y))))$$

3. Stelle Pränexform her!

$$\exists x\forall z\exists y\, (P(z, f(z)) \wedge \neg(P(y, x) \wedge R(y)))$$

4. Ersetze die mit Existenzquantor gebundenen Variablen durch Skolemkonstanten bzw. -funktionen!*

$$\forall z\, (P(z, f(z)) \wedge \neg(P(h(z), c) \wedge R(h(z))))$$

5. Forme die Matrix in Konjunktive Normalform um!

$$\forall z\, (P(z, f(z)) \wedge (\neg P(h(z), c) \vee \neg R(h(z))))$$

Ausgabe: (erfüllbarkeits*)äquivalente Klauselmenge M (ergibt sich aus Matrix):

$$M = \big\{\{P(z, f(z))\}, \{\neg P(h(z), c), \neg R(h(z))\}\big\}$$

Bei den mit * gekennzeichneten Schritten handelt es sich um Umformungen, die zwar nicht die Äquivalenz, aber die Erfüllbarkeit erhalten (vgl. Satz 3.5.9).

Übung 3.5.13. Überführen Sie die Formel $F = \forall x R(x, x) \wedge \forall x \forall y \big(R(x, y) \longrightarrow R(y, x)\big)$ in Normalform (Klauseldarstellung)!

3.6 Herbrand-Theorie

- Idee: Suchraum nach potentiellen Modellen einschränken
- Entwickler: Jacques Herbrand, Kurt Gödel, Thoralf Skolem

Definition 3.6.1. Eine Formel ohne freie Variablen heißt *geschlossen*, sonst *offen*. Wir betrachten im Folgenden nur geschlossene Formeln in Skolemform, genannt *Aussagen*.

Definition 3.6.2 (Herbrand-Universum)**.** Als *Herbrand-Universum* $D(F)$ wird die Menge aller variablenfreier Terme bezeichnet, welche aus den Bestandteilen einer Formel F wie folgt gebildet werden können (induktive Definition):

$D(F)$ enthält genau die Elemente folgender Bauart:

1. Alle in F vorkommenden Konstanten sind in $D(F)$. Falls F keine Konstanten enthält, so ist $a \in D(F)$, wobei a ein neues Konstantensymbol ist.

2. Sei f ein Funktionssymbol in F mit Stelligkeit $k \geq 1$ und die Terme $t_1, \ldots, t_k \in D(F)$. Dann ist auch $f(t_1, \ldots, t_k) \in D(F)$.

Satz 3.6.3. $D(F)$ ist stets eine nicht-leere Teilmenge der Menge aller Terme gemäß Def. 3.1.1. Außerdem enthält $D(F)$ unendlich viele Elemente gdw. in F mindestens ein Funktionssymbol f mit Stelligkeit $k \geq 1$ vorkommt.

Beispiel 3.6.4. Sei $F = \forall x \forall y (P(x, f(x, y)) \vee P(x, g(x)))$.
Dann ist $D(F) = \{a, f(a, a), g(a), f(g(a), a), g(f(a, a)), \ldots\}$.

Übung 3.6.5. Nennen Sie mindestens je vier Elemente aus dem Herbrand-Universum der folgenden Formeln in Skolemform, sofern möglich!

(a) $\forall x \forall y \big(R(x, x) \wedge (R(x, y) \longrightarrow R(y, x))\big)$

(b) $\forall x \forall y \forall z \big(P(x, f(x)) \vee P(c, c)\big)$

(c) $\forall v Q(g(h(v), v))$

Definition 3.6.6 (Herbrand-Struktur)**.** Sei F eine Aussage. Dann heißt eine zu F passende Struktur $\mathcal{A} = (\mathcal{U}_\mathcal{A}, \mathcal{I}_\mathcal{A})$ *Herbrand-Struktur* für F, falls gilt:

1. $\mathcal{U}_\mathcal{A} = D(F)$

2. Für jeden Funktor f und Terme $t_1, \ldots, t_n$ $(n \geq 0)$ ist $f^\mathcal{A}(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$.

Bemerkung 3.6.7. Die Gleichsetzung $f^\mathcal{A}(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$ entspricht einem „Kurzschließen“ von Syntax und Semantik. Die Interpretation der Prädikatensymbole ist in einer Herbrand-Struktur nicht festgelegt. Für F (siehe Beispiel 3.6.4) wähle z. B.

$$P^\mathcal{A} = \{(t_1, t_2) \mid t_1, t_2 \in D(F),\ t_2 = g(t_1)\} = \{(t, g(t)) \mid t \in D(F)\}$$

Es gilt dann $\mathcal{A} \models F$ und man bezeichnet $\mathcal{A}$ als *Herbrand-Modell*.

Satz 3.6.8. Sei F eine Aussage in Skolemform. F ist erfüllbar gdw. F ein Herbrand-Modell besitzt.

Beweis:

$\boxed{\Longleftarrow}$ Da es ein Herbrand-Modell $\mathcal{A}$ gibt, ist F klarerweise erfüllbar.

$\boxed{\Longrightarrow}$ Wir beweisen diese Richtung in 3 Schritten:

1. Sei $\mathcal{A} = (\mathcal{U}_\mathcal{A}, \mathcal{I}_\mathcal{A})$ Modell von F. Falls in F keine Konstante vorkommt, so erweitern wir $\mathcal{A}$ um die Festlegung $a^\mathcal{A} = m$ für ein beliebiges $m \in \mathcal{U}_\mathcal{A}$.
2. Wir definieren die Herbrand-Struktur $\mathcal{B} = (D(F), \mathcal{I}_\mathcal{B})$ wie folgt. Sei P ein n-stelliges Prädikatensymbol in F und $t_1, \ldots, t_n \in D(F)$. Dann definieren wir:

$$(t_1, \ldots, t_n) \in P^\mathcal{B} \Longleftrightarrow (\mathcal{A}(t_1), \ldots, \mathcal{A}(t_n)) \in P^\mathcal{A}$$

Die Definition der Prädikate aus $\mathcal{A}$ wird sozusagen in $\mathcal{B}$ übernommen.

3. Wir zeigen nun: $\mathcal{B}$ ist Herbrand-Modell von F.
   Trick: Wir zeigen die stärkere Behauptung: Für jede Aussage G in Skolemform, die aus den Bestandteilen von F aufgebaut ist, gilt $\mathcal{A} \models G \Longrightarrow \mathcal{B} \models G$ (siehe Beweis unten). Die eigentliche zu beweisende Behauptung ergibt sich dann mit $F = G$.

Beweis zu Schritt 3.: Induktion über die Zahl der Allquantoren:

- $\underline{n = 0:}$ $\mathcal{A}(G) = \mathcal{B}(G)$ nach Definition von $\mathcal{B}$.
- $\underline{n \mapsto n+1:}$ Sei G eine Formel mit $n+1$ Allquantoren:

  $$G = \forall x H \quad (H \text{ hat } n \text{ Allquantoren})$$

  Aus der Voraussetzung $\mathcal{A} \models G$ folgt

  $$\mathcal{A}_{[x/u]}(H) = 1 \quad \text{für alle } u \in \mathcal{U}_{\mathcal{A}}$$

  und damit erst recht für alle $u = \mathcal{A}(t)$ mit $t \in D(F)$. Anders ausgedrückt: Für alle $t \in D(F)$ gilt mit dem Überführungslemma (Lemma 3.3.3)

  $$\mathcal{A}_{[x/\mathcal{A}(t)]}(H) = 1 = \mathcal{A}(H[x/t])$$

  Nach I.V. muss daher $\mathcal{B}(H[x/t]) = 1$ für alle $t \in D(F)$ gelten. Mit dem Überführungslemma folgt

  $$\mathcal{B}_{[x/\mathcal{B}(t)]}(H) = 1 \text{ für alle } t \in D(F)$$

  und schließlich

  $$\mathcal{B}(\forall x H) = \mathcal{B}(G) = 1$$ ⊠

Korollar 3.6.9 (Satz von Löwenheim-Skolem)**.** Jede erfüllbare prädikatenlogische Formel besitzt bereits ein abzählbares Modell.

Beweis: $D(F)$ ist stets abzählbar (unendlich). Die Abzählung kann über die Anzahl der Funktionssymbole in den Termen aus den Elementen von F konstruiert werden. ⊠

Übung 3.6.10 (Fortsetzung von Übung 3.6.5)**.** Die Formel

$$\forall x \forall y \big(R(x,x) \wedge (R(x,y) \longrightarrow R(y,x))\big)$$

hat z. B. das Modell $\mathcal{A}$ mit $\mathcal{U}_{\mathcal{A}} = \mathbb{N}$ und $R^{\mathcal{A}} = (=) = \{(x,x) \mid x \in \mathbb{N}\}$. Daraus lässt sich ein Herbrand-Modell $\mathcal{B}$ ableiten. Welches?

Definition 3.6.11 (Herbrand-Expansion, Grundinstanz)**.** Sei $F = \forall y_1 \cdots \forall y_n F^*$ eine Aussage in Skolemform. Dann ist die *Herbrand-Expansion* $E(F)$ definiert durch:

$$E(F) = \{F^*[y_1/t_1] \cdots [y_n/t_n] \mid t_1, \ldots, t_n \in D(F)\}$$

$E(F)$ ist eine abzählbare, i.d.R. unendliche Formelmenge. Die Elemente von $E(F)$ heißen *Grundinstanzen* von F.

Beispiel 3.6.12 (Fortsetzung von Beispiel 3.6.4)**.**
Für $F = \forall x \forall y (P(x, f(x, y)) \vee P(x, g(x)))$ gilt:

$$E(F) = \{P(a, f(a, a)) \vee P(a, g(a)), P(a, f(a, g(a))) \vee P(a, g(a)), \dots\}$$

Die (mit $\forall$ quantifizierten) Variablen (hier: x und y) sind unabhängig voneinander durch *alle* möglichen Kombinationen von Grundtermen zu ersetzen (hier: $[x/a, y/a]$, $[x/a, y/g(a)]$ etc.).

Übung 3.6.13. Geben Sie, sofern möglich, mindestens je zwei Elemente aus $E(F)$ zu den Formeln F aus Übung 3.6.5 an!

Satz 3.6.14 (Gödel-Herbrand-Skolem)**.** Für jede Aussage F in Skolemform gilt: F erfüllbar gdw. $E(F)$ im aussagenlogischen Sinn erfüllbar ist, wobei die (variablenfreien) prädikatenlogischen Atome in $E(F)$ als aussagenlogische Atome aufzufassen sind.

Beweis: Wegen Satz 3.6.8 genügt es zu zeigen, dass F ein Herbrand-Modell $\mathcal{A}$ besitzt gdw. $E(F)$ erfüllbar ist. Für eine Formel F in Skolemform, d. h. $F = \forall y_1 \cdots \forall y_n F^*$, gilt aber:

$$\begin{aligned}
\mathcal{A} \models F \quad &\overset{\text{Def. 3.2.6}}{\Longleftrightarrow} \quad \forall t_1, \dots, t_n \in D(F) : \mathcal{A}_{[y_1/t_1]\cdots[y_n/t_n]}(F^*) = 1 \\
&\overset{\text{Lemma 3.3.3}}{\Longleftrightarrow} \quad \forall t_1, \dots, t_n \in D(F) : \mathcal{A}(F^*[y_1/t_1]\cdots[y_n/t_n]) = 1 \\
&\overset{\text{Def. 3.6.11}}{\Longleftrightarrow} \quad \forall G \in E(F) : \mathcal{A}(G) = 1 \\
&\overset{\text{Def. 2.1.14}}{\Longleftrightarrow} \quad \mathcal{A} \models E(F)
\end{aligned}$$

In der letzten Aussage kann das Herbrand-Modell $\mathcal{A}$ wie folgt als Belegung $\mathcal{A}'$ im aussagenlogischen Sinne nach Def. 2.1.3 verstanden werden: Jedes (variablenfreie) prädikatenlogische Atom in $E(F)$ der Form $P(t_1, \dots, t_n)$ mit $t_1, \dots, t_n \in D(F)$ wird als aussagenlogisches Atom A nach Def. 2.1.1 aufgefasst, und es gilt $\mathcal{A}'(A) = 1$ gdw. $(t_1, \dots, t_n) \in P^{\mathcal{A}}$. ⊠

Korollar 3.6.15 (Herbrand)**.** Eine Aussage F ist unerfüllbar gdw. es eine endliche Teilmenge von $E(F)$ gibt, die im aussagenlogischen Sinne unerfüllbar ist.

Beweis: F unerfüllbar $\overset{\text{Satz 3.6.14}}{\Longleftrightarrow}$ $E(F)$ unerfüllbar $\overset{\text{Kor. 2.5.7}}{\Longleftrightarrow}$ es gibt eine endliche Teilmenge von $E(F)$, die im aussagenlogischen Sinne unerfüllbar ist. ⊠

Semi-Entscheidungsverfahren für Unerfüllbarkeit nach Gilmore

Eingabe: Aussage F (Skolemform)
$n = 0$
repeat
  $n \mapsto n + 1$
  Erzeuge die ersten n Formeln in $E(F)$
until $F_1 \wedge F_2 \wedge \cdots \wedge F_n$ unerfüllbar
write „unerfüllbar“

Der Algorithmus von Gilmore gerät in eine Endlosschleife, sofern die Formel erfüllbar und das Herbrand-Universum unendlich ist. Daher ist das Problem herauszufinden, ob eine Formel unerfüllbar ist, nur semi-entscheidbar (vgl. Def. 3.9.2).

Beispiel 3.6.16. Falls die Matrix von F in KNF ist, lässt sich die aussagenlogische Resolution beim Testen auf Unerfüllbarkeit einsetzen:

$$\begin{aligned} F &= \forall x \forall y (P(g(f(x))) \wedge \neg P(g(y))) \qquad (F^* \text{ in KNF}) \\ &\hookrightarrow \big\{\{P(g(f(x)))\}, \{\neg P(g(y))\}\big\} \qquad (\text{Klauselmenge zu } F^*) \end{aligned}$$

$$D(F) = \{a, f(a), g(a), f(g(a)), g(f(a)), f(f(a)), g(g(a)), \dots\}$$

$$E(F) = \big\{\{P(g(f(a)))\}, \{\neg P(g(a))\}, \{\neg P(g(f(a)))\}, \dots\big\}$$

$$\begin{array}{ccc} [x/a] & [y/a] & [y/f(a)] \\ \searrow & & \swarrow \\ & \square & \end{array}$$

Die obige Vorgehensweise nennt man *Grundresolution*. Die Bezeichnung leitet sich aus der Verwendung variablenfreier Terme (Grundinstanzen gemäß Def. 3.6.11) bei der (letztendlich aussagenlogischen) Resolution ab.

Fazit: Im Ergebnis führen wir die Unerfüllbarkeit einer Formel F in Skolemform, die gemäß Def. 3.5.4 nur noch Allquantoren enthält, auf die Untersuchung aller Grundinstanzen von F zurück, genau genommen einer endlichen Teilmenge davon (vgl. Korolllar 3.6.15).

Aber: Bei der Herbrand-Expansion werden u. U. sehr viele Grundinstanzen erzeugt, bevor man zwei komplementäre Atome erhält, wie sie für die Grundresolution benötigt werden. Hier wäre es besser, gezielt nach geeigneten (Grund)Instanzen zu suchen.

3.7 Unifikation

Idee: Wir arbeiten bei der Resolution mit Variablen, deren Ersetzung (Substitution) verzögert wird. Dies führt zum Begriff der *Unifikation*. Ziel ist es, eine *allgemeinste* Substitution zu finden, welche je zwei Terme gleichmacht (unifiziert).

Beispiel 3.7.1.

$$\begin{array}{cc} \{P(g(f(x)))\} & \{\neg P(g(y))\} \\ \big| \ [x/a] & \big| \ [y/f(a)] \\ \{P(g(f(a)))\} & \{\neg P(g(f(a)))\} \\ \diagdown \qquad \diagup & \\ \square & \end{array}$$

Im obigen Beispiel würde man bereits durch die Substitution $[y/f(x)]$ das gewünschte Ergebnis – zwei komplementäre Atome – erhalten.

Definition 3.7.2 (Substitution – Funktion)**.** Sei V eine Menge von Variablen und $T(V)$ eine Menge von Termen. Eine Abbildung $\sigma : V \to T(V)$ heißt *Substitution*, falls $(\star)$ $\sigma(x) \neq x$ für höchstens endlich viele $x \in V$ gilt. Die Menge der Variablen $\{x_1, \ldots, x_n\}$, für die $(\star)$ gilt, heißt *Domäne* von σ, abgekürzt $\mathrm{dom}(\sigma)$.

Bemerkung 3.7.3 (Schreibweisen)**.** Die Substitution σ mit

1. $\sigma(x_i) = t_i$ und
2. $\sigma(x) = x$ für $x \neq x_i$

für alle i mit $1 \leq i \leq n$ wird häufig in der Form $[x_1/t_1, \ldots, x_n/t_n]$ notiert.
$\epsilon = [\,]$ heißt *leere Substitution*.

Definition 3.7.4. Sei $\sigma = [x_1/t_1, \ldots, x_n/t_n]$ eine Substitution und E ein Ausdruck (Term, Literal oder Formel). Dann ist $E\sigma$ der Ausdruck, den man erhält, indem alle Vorkommen der Variablen x_i für $1 \leq i \leq n$ simultan durch t_i ersetzt werden.

Beispiel 3.7.5.

$$\begin{aligned} \sigma &= [x/y, y/f(a), z/b] \\ E &= P(x, g(y, z)) \\ E\sigma &= P(y, g(f(a), b)) \end{aligned}$$

Definition 3.7.6. Die *Komposition* zweier Substitutionen σ und λ, geschrieben $\sigma\lambda$, ist definiert durch $x(\sigma\lambda) = (x\sigma)\lambda$ für alle $x \in V$.

Beispiel 3.7.7.

$$\begin{aligned} \sigma &= [x/f(y), y/z] \\ \lambda &= [x/a, y/b, z/y] \\ \sigma\lambda &= [x/f(y)\lambda, y/z\lambda, \cancel{x/a}, \cancel{y/b}, z/y] \\ &= [x/f(b), \cancel{y/y}, z/y] \\ &= [x/f(b), z/y] \end{aligned}$$

Satz 3.7.8. Für Substitutionen ρ, σ, τ gilt $(\rho\sigma)\tau = \rho(\sigma\tau)$, i. A. aber nicht $\sigma\tau = \tau\sigma$. Außerdem gilt $(E\sigma)\tau = E(\sigma\tau)$ (gemischtes Assoziativgesetz).

Bemerkung 3.7.9. Wir betrachten die algebraische Struktur $(\mathcal{S}, \circ)$, wobei $\mathcal{S}$ die Menge aller Substitutionen und $\circ$ die Komposition von Substitutionen sei. Laut Satz 3.7.8 ist $(\mathcal{S}, \circ)$ assoziativ, aber nicht kommutativ. Die leere Substitution ϵ ist das neutrale Element. Es gibt aber nicht zu jeder Substitution eine inverse, z. B. kann $[x/f(y)]$ nicht rückgängig gemacht werden, denn Substitutionen ersetzen nur Variablen, nicht Funktionsausdrücke! $(\mathcal{S}, \circ)$ ist also ein *Monoid*, aber keine Gruppe.

Letzteres gilt im Übrigen allgemein für Mengen $\mathcal{F}$ von Abbildungen $f : M \to M$ bzgl. der Komposition als Verknüpfung (vgl. Beispiel 1.4.4). Das neutrale Element ist stets die Identitätsfunktion id mit $x \mapsto x$ für alle $x \in M$. Im Falle der gerade eingeführten Menge $\mathcal{S}$ von Substitutionsabbildungen, die aufgrund Def. 3.7.2 spezielle Abbildungen sind, ist das neutrale Element die leere Substitution ϵ.

Übung 3.7.10. Sei $\sigma = [x/y, z/f(a)]$, $\tau = [x/a, y/z, z/x]$ und $E = g(x, y, z)$.
Bestimmen Sie:

(a) $\sigma \circ \tau$ (b) $E\sigma\tau$ (c) $\tau \circ \sigma$ (d) $E\tau\sigma$

Bemerkung 3.7.11. Hintereinanderausführen (Komposition) von Substitutionen $\neq$ Hintereinanderschreiben von Substitutionen, z. B. $[x/y] \circ [y/c] = [x/c, y/c] \neq [x/y, y/c]$.

Definition 3.7.12 (Unifikator)**.** Sei $\{E_1, \ldots, E_n\}$ eine Menge von Ausdrücken (Terme oder Literale) und σ eine Substitution. Dann heißt σ ein *Unifikator* für $\{E_1, \ldots, E_n\}$ genau dann, wenn $E_1\sigma = \cdots = E_n\sigma$. Eine Substitution σ heißt *allgemeinster Unifikator*, falls sie allgemeiner ist als jeder (andere) Unifikator τ. Eine Substitution σ heißt *allgemeiner* als eine Substitution τ, i. Z. $\sigma \lesssim \tau$, falls es eine Substitution λ gibt mit $\tau = \sigma\lambda$.

Bemerkung 3.7.13. Eine Substitution σ heißt *idempotent*, falls $\sigma\sigma = \sigma$ (vgl. Satz 2.2.6). Idempotente Substitutionen enthalten keine Variablen, die sowohl auf einer linken Seite als auch innerhalb einer rechten Seite von / vorkommen. Unifikatoren sind stets idempotent. Der allgemeinste Unifikator ist i. A. nicht eindeutig: $f(x)$ und $f(y)$ haben z. B. $[x/y]$, $[y/x]$ und $[x/z, y/z]$ als allgemeinste Unifikatoren. Sie sind aber bis auf reine Variablenumbenennungen gleich.

Übung 3.7.14 (vgl. Beispiel 3.7.1 und Bemerkung 3.7.13)**.** Geben Sie je eine Substitution λ gemäß Definition 3.7.12 an, die folgende Aussagen belegt:

(a) $[y/f(x)] \lesssim [x/a, y/f(a)]$

(b) $[x/y] \lesssim [y/x]$

(c) $[y/x] \lesssim [x/z, y/z]$

(d) $[x/z, y/z] \lesssim [x/y]$

Gleichungssysteme

Darstellung: Eine Menge von zu unifizierenden Ausdrücken (Terme oder Literale) $M = \{t_1, \ldots, t_n\}$ wird überführt in die Gleichungsmenge $N = \{t_1 = t_2, \ldots, t_{n-1} = t_n\}$.

Definition 3.7.15. Ein Unifikator σ für die Gleichungsmenge $N = \{s_1 = t_1, \ldots, s_n = t_n\}$, ist eine Substitution derart, dass $s_1\sigma = t_1\sigma, \ldots, s_n\sigma = t_n\sigma$ zutrifft.

Man kann Unifikatoren durch Lösen von (Systemen von Term-)Gleichungen bestimmen – ähnlich wie in der Arithmetik. Die Idee des folgenden Algorithmus nach [JK91] ist, die Gleichungsmenge N solange Unifikator-erhaltend zu transformieren, bis der allgemeinste Unifikator unmittelbar abgelesen werden kann. Ein anderer Algorithmus, der aber nicht so effizient implementiert werden kann, ist z. B. in [Sch00] dargestellt.

Theorem 3.7.16 (Unifikationsalgorithmus)**.** Sei N eine endliche Gleichungsmenge. Der nachfolgende Unifikationsalgorithmus terminiert stets, angewendet auf N. Falls der Algorithmus mit FAIL anhält, hat N keinen Unifikator. Falls der Algorithmus erfolgreich terminiert, ist N in die Gleichungsmenge $\{y_1 = u_1, \ldots, y_m = u_m\}$ überführt, welche den (allgemeinsten) Unifikator von N wiedergibt.

Unifikationsalgorithmus

Der folgende Algorithmus arbeitet mit Transformationsregeln. Dabei bedeutet $\frac{R \uplus \{s = t\}}{R'}$, dass eine syntaktische Gleichung $s = t$ zusammen mit einer Menge R weiterer Gleichungen überführt (transformiert) wird in das Gleichungsystem R'. Das Symbol $\uplus$ steht hier für *disjunkte Vereinigung*, d. h. es gilt stets $R \cap \{s = t\} = \emptyset$, was in diesem Fall gleichbedeutend ist mit $(s = t) \notin R$.

Eingabe: Gleichungsmenge R
Ausgabe: allgemeinster Unifikator σ für R oder FAIL, falls σ nicht existiert

Wende die folgenden Transformationen so lange wie möglich an:

1. *Orient:* $\frac{R \uplus \{t = x\}}{R \cup \{x = t\}}$, wobei x Variable und t echter Term (d. h. keine Variable)
2. *Delete:* $\frac{R \uplus \{s = s\}}{R}$
3. *Decompose* (Termreduktion):

$$\frac{R \uplus \{f(s_1, \ldots, s_n) = f(t_1, \ldots, t_n)\}}{R \cup \{s_1 = t_1, \ldots, s_n = t_n\}}$$

4. *Eliminate* (Variablenelimination I):

$$\frac{R \uplus \{x = t\}}{R[x/t] \cup \{x = t\}}, \text{ falls } x \text{ nicht in } t, \text{ aber in } R \text{ vorkommt}$$

5. *Coalesce* (Variablenelimination II):

$$\frac{R \uplus \{x = y\}}{R[x/y] \cup \{x = y\}}, \text{ falls } x \neq y \text{ in } R \text{ vorkommt}$$

6. *Conflict:*

$$\frac{R \uplus \{f(s_1, \ldots, s_m) = g(t_1, \ldots, t_n)\}}{\text{FAIL}}, \text{ falls } f \neq g \text{ oder } m \neq n$$

7. *Occur Check:* $\frac{R \uplus \{x = t\}}{\text{FAIL}}$, falls x in t vorkommt

Bemerkung 3.7.17 (zum Unifikationsalgorithmus)**.**

- Der Algorithmus hat, sofern er effizient implementiert wird, nur lineare Laufzeit bezogen auf die Gesamtlänge aller zu unifizierenden Terme.

- Die Regeln 1.–5. sind in beliebiger Reihenfolge anwendbar.
- Falls Regeln 6. und 7. greifen (FAIL), ist das Gleichungssystem nicht lösbar.
- Das allgemeine Vorgehen ist folgendermaßen:
  1. Wende *Decompose* an!
  2. Wende weitere Ersetzungsregeln so lange an, bis entweder FAIL auftritt oder keine Regel mehr anwendbar ist!
- Die Decompose-Regel besagt, dass im Kontext der syntaktischen Unifikation von Termen im Herbrand-Universum alle Funktionen injektiv sind, d. h. $f(x) = f(y)$ impliziert stets $x = y$.
- Die Terme x und $f(x)$ scheinen zwar unifizierbar zu sein durch die (nicht idempotente) Substitution $[x/f(x)]$, aber deren ein oder auch mehrfache Anwendung (simultan) auf x und $f(x)$ macht die Terme nicht gleich. Im Unifikationsalgorithmus werden derartige Fälle durch den *Occur Check* abgefangen.

Beispiel 3.7.18. $\{f(g(x), x) = f(g(g(y)), g(z))\}$

1. Decompose: $\Downarrow$

$$\{g(x) = g(g(y)), x = g(z)\}$$

2. Decompose: $\Downarrow$

$$\{x = g(y), x = g(z)\}$$

3. Eliminate: $\Downarrow$

$$\{x = g(y), g(y) = g(z)\}$$

4. Decompose: $\Downarrow$

$$\{x = g(y), y = z\}$$

5. Coalesce: $\Downarrow$

$$\{x = g(z), y = z\}$$

Aus der letzten Gleichungsmenge, die nicht mehr weiter umgeformt werden kann, lässt sich – wie stets im positiven Falle der Unifizierbarkeit – der allgemeinste Unifikator unmittelbar ablesen:

$$\sigma = [x/g(z), y/z]$$

Es gilt wie gewünscht $f(g(x), x)\sigma = f(g(g(y)), g(z))\sigma$.

Übung 3.7.19. Berechnen Sie die allgemeinsten Unifikatoren folgender Termgleichungen oder begründen Sie, warum dies nicht möglich ist.

(a) $\{h(x, y, c) = h(z, x, y)\}$

(b) $\{f(x, c) = f(g(y), x)\}$

(c) $\{g(y) = y\}$

Übung 3.7.20. Was ist der allgemeinste Unifikator für $f(x) = f(y)$ gemäß Algorithmus?

$$[x/y] \text{ oder } [y/x] \text{ oder } [x/z, y/z]\,?$$

3.8 Prädikatenlogische Resolution

Definition 3.8.1 (Resolventenbildung)**.** Seien K_1 und K_2 prädikatenlogische Klauseln. Die beiden Klauseln müssen variablendisjunkt sein, d. h. sie dürfen keine Variablen mit gleichem Namen enthalten. Daher benennen wir, sofern nötig, zuerst die Variablen in einer oder auch beiden Klauseln geeignet um (gemäß Satz 3.4.2), wobei stets komplett neue Variablenbezeichner zu verwenden sind. Wir erhalten die Klauseln $K_1\tau_1$ und $K_2\tau_2$, wobei τ_1 und τ_2 die jeweiligen Variablenumbenennungen sind. Es gilt:

1. Innerhalb einer Klausel (K_1 bzw. K_2) werden alle Variablen simultan ersetzt, d. h. in allen Literalen dieser Klausel.

2. Nach der Variablenumbenennung enthalten die beiden Klauseln $K_1\tau_1$ und $K_2\tau_2$ keine gemeinsamen Variablen mehr.

Nun sei $\{L_1, \ldots, L_m\} \subseteq K_1\tau_1$ und $\{L'_1, \ldots, L'_n\} \subseteq K_2\tau_2$ mit $m, n \geq 1$ und σ der allgemeinste Unifikator der Literalmenge $\{\overline{L}_1, \ldots, \overline{L}_m, L'_1, \ldots, L'_n\}$. Dann ist

$$R = \left(K_1\tau_1 \backslash \{L_1, \ldots, L_m\} \cup K_2\tau_2 \backslash \{L'_1, \ldots, L'_n\}\right)\sigma$$

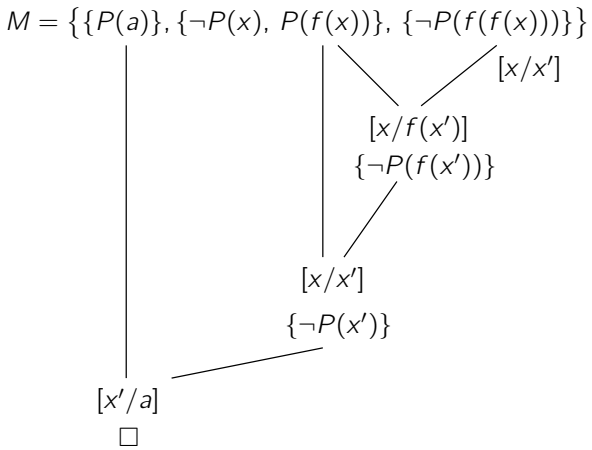
die (direkte) *Resolvente* aus K_1 und K_2. Falls $m = n = 1$ ist, so heißt R *binär*. Andernfalls handelt es sich um Resolventenbildung mit *Faktorisierung*.

Bemerkung 3.8.2 (Korrektheit und Vollständigkeit der prädikatenlogischen Resolution)**.** Es gilt (im Vorgriff auf Theorem 3.8.8):

> Eine prädikatenlogische Klauselmenge M ist unerfüllbar gdw. die leere Klausel $\square$ durch sukzessives Bilden von prädikatenlogischen Resolventen (gemäß Def. 3.8.1) herleitbar ist.

Der Begriff Herleitung (Deduktion) ist hierbei analog zur Aussagenlogik definiert (siehe Def. 2.4.5).

Beispiel 3.8.3 (Herleitung mit prädikatenlogischer Resolution)**.**



Übung 3.8.4. Geben Sie für die folgenden Klauselmengen M jeweils eine Herleitung der leeren Klausel mit prädikatenlogischer Resolution an, sofern das möglich ist!

(a) $\{\{P(x)\}, \{\neg P(f(x))\}\}$

(b) $\{\{P(f(a))\}, \{\neg P(x), \neg P(f(x))\}\}$

Faktorisierung

Satz 3.8.5. Die binäre Resolution ist ohne Faktorisierung nicht vollständig.

Beweis (durch Gegenbeispiel)**:**

$$\{P(x), P(y)\} \qquad \{\neg P(u), \neg P(v)\}$$

Die schrittweise Unifikation je zweier Atome liefert hier nicht die leere Klausel. Jedoch ergibt sich $\square$ als Resolvent, wenn $\sigma = [x/v, y/v, u/v]$ gewählt wird. ⊠

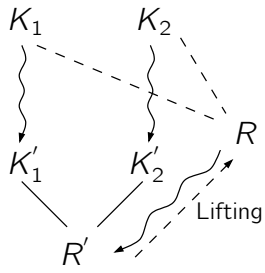
Korrektheit und Vollständigkeit der prädikatenlogischen Resolution

Satz 3.8.6 (Grundresolutionssatz)**.** Eine Aussage $F = \forall y_1 \cdots \forall y_n F^*$ in geschlossener Skolemform mit Matrix F^* ist unerfüllbar gdw. eine Ableitung von Klauseln $K_1, \ldots, K_n$ existiert mit:

1. $K_n = \square$
2. Für alle m mit $1 \leq m \leq n$ gilt: K_m ist entweder eine Grundinstanz einer Klausel K der in KNF überführten Matrix F^* oder K_m ist aussagenlogische Resolvente zweier Klauseln K_i, K_j mit $i, j < m$.

Beweis: Sei F' die Formel, die aus F durch Umformung der Matrix F^* in eine äquivalente KNF entsteht (Theorem 2.3.7). Dann gilt mit Satz 3.6.14: $F \equiv F'$ ist erfüllbar gdw. $E(F')$ im aussagenlogischen Sinne erfüllbar ist. $E(F')$ enthält durchweg Formeln in KNF, also von der Form $G = G_1 \wedge \cdots \wedge G_k$. Aufgrund der Semantik der Konjunktion (Def. 2.1.3) ist G erfüllbar gdw. die (aussagenlogische) Klauselmenge $\{G_1, \ldots, G_k\}$ erfüllbar ist (gemäß Def. 2.1.14). Aus der Korrektheit und Vollständigkeit der aussagenlogischen Resolution (Korollar 2.5.5), angewendet auf die Vereinigung aller Klauselmengen der obigen Form, folgt dann sofort die Behauptung des Grundresolutionssatzes. ⊠

Lemma 3.8.7 (Lifting-Lemma)**.** Seien K_1 und K_2 prädikatenlogische Klauseln und K_1', K_2' zugehörige (variablenfreie) Grundinstanzen. Falls R' Resolvent im aussagenlogischen Sinn von K_1' und K_2' ist, dann gibt es eine prädikatenlogische Resolvente R von K_1 und K_2, von dem R' eine Grundinstanz ist.



Beweis: Seien zunächst τ_1 und τ_2 Variablenumbenennungen nach Def. 3.8.1, so dass $K_1\tau_1$ und $K_2\tau_2$ keine gemeinsamen Variablen mehr enthalten. Da K_1' bzw. K_2' Grundinstanzen von K_1 bzw. K_2 sind, sind sie auch Grundinstanzen von $K_1\tau_1$ bzw. $K_2\tau_2$. Seien also σ_1 und σ_2 Grundsubstitutionen, welche alle vorkommenden Variablen durch variablenfreie Terme ersetzen, mit $K_1' = K_1\tau_1\sigma_1$ und $K_2' = K_2\tau_2\sigma_2$. Da es keine Variablen gibt, die sowohl in σ_1 als auch σ_2 ersetzt werden, d. h. $\mathrm{dom}(\sigma_1) \cap \mathrm{dom}(\sigma_2) = \emptyset$, setzen wir $\sigma_0 = \sigma_1\sigma_2$, und es gilt $K_1' = K_1\tau_1\sigma_0$ und $K_2' = K_2\tau_2\sigma_0$.

K_1' und K_2' sind nach Voraussetzung aussagenlogisch resolvierbar. Daher muss es nach Def. 2.4.2 ein Literal $L \in K_1'$ und $\overline{L} \in K_2'$ mit $R' = K_1' \setminus \{L\} \cup K_2' \setminus \{\overline{L}\}$ geben. Das Literal L entstammt aus einem oder mehreren Literalen aus $K_1\tau_1$ durch die Grundsubstitution σ_0. Analoges gilt für $\overline{L}$ und $K_2\tau_2$. Es gilt also für gewisse Literale $L_1, \ldots, L_m \in K_1\tau_1$ und $L_1', \ldots, L_n' \in K_2\tau_2$ mit $m, n \geq 1$, dass $L = L_1\sigma_0 = \cdots = L_m\sigma_0$ und $\overline{L} = L_1'\sigma_0 = \cdots = L_n'\sigma_0$. Daher sind $K_1\tau_1$ und $K_2\tau_2$ resolvierbar, denn σ_0 ist ein Unifikator für die Literalmenge $\{\overline{L}_1, \ldots, \overline{L}_m, L_1', \ldots, L_n'\}$. Dann gibt es aber auch einen allgemeinsten Unifikator σ gemäß Def. 3.7.12, der diese Literale unifiziert, mit $\sigma_0 = \sigma\lambda$. Es ist deshalb

$$R = \left(K_1\tau_1 \setminus \{L_1, \ldots, L_m\} \cup K_2\tau_2 \setminus \{L_1', \ldots, L_n'\}\right)\sigma$$

eine prädikatenlogische Resolvente von K_1 und K_2 gemäß Def. 3.8.1. Wir erhalten damit:

$$\begin{aligned} R' &= K_1' \setminus \{L\} \cup K_2' \setminus \{\overline{L}\} \\ &= K_1\tau_1\sigma_0 \setminus \{L\} \cup K_2\tau_2\sigma_0 \setminus \{\overline{L}\} \\ &= \left(K_1\tau_1 \setminus \{L_1, \ldots, L_m\} \cup K_2\tau_2 \setminus \{L_1, \ldots, L_n'\}\right)\sigma_0 \\ &= \left(K_1\tau_1 \setminus \{L_1, \ldots, L_m\} \cup K_2\tau_2 \setminus \{L_1, \ldots, L_n'\}\right)\sigma\lambda \\ &= R\lambda \end{aligned}$$

Somit ist gezeigt, dass R' eine Grundinstanz von R ist (mittels Substitution λ). ⊠

Theorem 3.8.8 (Resolutionssatz)**.** Sei F eine prädikatenlogische Formel in geschlossener Skolemform mit Matrix F^* in KNF. Dann gilt:

$$F \text{ unerfüllbar} \iff \square \in \mathrm{Res}^*(F^*)$$

Letzteres heißt, $\square$ lässt sich durch prädikatenlogische Resolution aus F^* herleiten.

Beweis: F unerfüllbar $\overset{\text{Satz 3.6.14}}{\Longleftrightarrow}$ $E(F)$ unerfüllbar (im aussagenlogischen Sinne) $\overset{\text{Satz 3.8.6}}{\Longleftrightarrow}$ $\square \in \mathrm{Res}^*(E(F)) \overset{(\dagger)}{\Longleftrightarrow} \square \in \mathrm{Res}^*(F^*)$. Für die letzte Äquivalenz (†) beweisen wir die beiden Richtungen einzeln:

$\boxed{\Longrightarrow}$ Der Beweis kann mittels Induktion über die Länge der Grundresolutionsherleitung erbracht werden. Dabei ist das Lifting-Lemma (Lemma 3.8.7) auf jeden der aussagenlogischen Resolutionsschritte in $E(F)$ anzuwenden.

$\boxed{\Longleftarrow}$ Sei eine prädikatenlogische Deduktion der leeren Klausel $\square$ aus F^* (verstanden als Klauselmenge) gegeben. Wir schreiben $K_1 \overset{\theta}{\dashrightarrow} K_2$, falls K_2 eine prädikatenlogische Resolvente aus K_1 und einer weiteren Klausel gemäß Def. 3.8.1 ist mit $\theta = \tau\sigma$, wobei τ die auf die Klausel K_1 angewendete Variablenumbenennung und σ der im betreffenden Resolutionsschritt verwendete allgemeinste Unifikator ist. Wir betrachten nun Pfade im Resolutionsgraphen, d. h. alle Folgen von Klauseln aus der prädikatenlogischen Deduktion der leeren Klausel der Gestalt

$$K = K_1 \overset{\theta_1}{\dashrightarrow} \cdots \overset{\theta_{n-1}}{\dashrightarrow} K_n = \square$$

und bilden alle möglichen Grundinstanzen der Form $K' = K\theta_1 \cdots \theta_n$ für alle derartigen Folgen, wobei θ_n eine Substitution ist, die alle in der prädikatenlogischen Deduktion vorkommenden Variablen auf ein und dieselbe, aber beliebig wählbare Konstante $a \in D(F)$ abbildet. Jede dieser (variablenfreien) Klauseln K' ist entweder eine Grundinstanz einer Klausel in F^* oder eines prädikatenlogischen Resolutionsschritts aus der prädikatenlogischen Deduktion und entspricht somit einem aussagenlogischen Resolutionsschritt in $E(F)$.

Den eigentlichen Beweis erhält man durch Induktion über die Länge der prädikatenlogischen Resolutionsherleitung. Man beachte, dass hierbei einem einzelnen prädikatenlogischen Resolutionsschritt in F^* mehrere Grundresolutionsschritte in $E(F)$ entsprechen können (vgl. Übung 3.8.9).

Somit ist insgesamt die prädikatenlogische Resolution – wie die aussagenlogische Resolution (siehe Korollar 2.5.5) – in Bezug auf die Fragestellung, ob eine Aussage F unerfüllbar ist, *korrekt* ($\Longleftarrow$) und *vollständig* ($\Longrightarrow$). ⊠

Übung 3.8.9. Wir betrachten die folgende prädikatenlogische Klauselmenge M:

$$\big\{\{P(x)\}, \{\neg P(y), Q(y)\}, \{\neg Q(a), \neg Q(b)\}\big\}$$

Leiten Sie die leere Klausel (a) mit prädikatenlogischer Resolution sowie (b) durch Grundresolution her! Welche Herleitung ist länger?

Beweisen mit Prädikatenlogik – Vorgehensweise

Falls die Gültigkeit einer allgemeinen Aussage zu zeigen ist, kann wie folgt vorgegangen werden:

1. Problem in Prädikatenlogik formulieren $\rightsquigarrow F$

2. Formel F negieren und $\neg F$ auf Normalform (Skolemform mit Matrix in KNF) bringen, ggf. unter Anwendung der folgenden Äquivalenz:

$$\neg(A_1 \wedge A_2 \wedge \cdots \wedge A_n \longrightarrow B) \equiv (A_1 \wedge A_2 \wedge \cdots \wedge A_n \wedge \neg B)$$

3. Unerfüllbarkeit von $\neg F$ zeigen (mit prädikatenlogischer Resolution)

Mit Hilfe der Resolution können also auch in der Prädikatenlogik indirekte Beweise geführt werden. Die Unerfüllbarkeit der negierten allgemeinen Aussage wird quasi durch ein Gegenbeispiel gezeigt, welches sich aus den verwendeten Variablenbindungen (Unifikatoren) ablesen lässt.

Übung 3.8.10. Zeigen Sie die folgende Implikation mittels prädikatenlogischer Resolution:

$$\forall x P(x) \longrightarrow \exists x P(x)$$

3.9 Unentscheidbarkeit der Prädikatenlogik

Bemerkung 3.9.1. Prädikatenlogik kann als Programmiersprache verwendet werden. Wie bei jeder hinreichend mächtigen Programmiersprache ist es jedoch nicht möglich, alle Probleme zu lösen. Bei sogenannten unentscheidbaren Problemen läuft jede Implementierung u. U. in eine Endlosschleife, so auch Theorembeweiser.

Exkurs: Theorie der Berechenbarkeit

Problem: allgemeine Frage, deren Antwort gesucht ist

Entscheidungsproblem: hat die Antwort *ja* oder *nein*

Abstraktion: Ein Problem kann durch eine Funktion $f : S \to T$ charakterisiert werden, für die eine Berechnungsprozedur gesucht wird. S ist hierbei die Eingabemenge (Probleme bzw. Fragestellungen) und T die Menge der möglichen Antworten.

Definition 3.9.2. Ein Entscheidungsproblem heißt *entscheidbar* bzw. *semi-entscheidbar* gdw. die zugehörige charakteristische Funktion f total bzw. partiell und berechenbar ist, d. h. es gibt ein Rechenverfahren (z. B. ein Java-Programm), das die Funktion $f : S \to T$ mit $T = \{0, 1\}$ berechnet, im Falle der Semi-Entscheidbarkeit aber nicht immer terminieren muss. Für alle $x \in S$ muss gelten $f(x) = 1$ gdw. die Antwort zum gegebenen Entscheidungsproblem *ja* ist. Ein Entscheidungsproblem heißt *unentscheidbar* bzw. *strikt unentscheidbar* gdw. es nicht entscheidbar bzw. nicht einmal semi-entscheidbar ist.

Satz 3.9.3. Es gibt unentscheidbare Probleme.

Beweis: durch Widerspruch, genauer Diagonalisierung, hier nicht. ⊠

Entscheidungsproblem der Prädikatenlogik

Gegeben: eine beliebige prädikatenlogische Formel F

Frage: Ist F eine gültige Formel?

Satz 3.9.4 (Church)**.** Das Gültigkeitsproblem ist unentscheidbar, d. h. es lässt sich kein Algorithmus formulieren, der das Problem allgemein löst.

Bemerkung 3.9.5. Das Gültigkeitproblem sowie das Unerfüllbarkeitsproblem ist semi-entscheidbar. Folglich ist das Erfüllbarkeitsproblem nicht einmal semi-entscheidbar, sonst wäre das Gültigkeitsproblem nämlich entscheidbar.

Problem:	Erfüllbarkeit	Unerfüllbarkeit	Gültigkeit
Aussagenlogik	entscheidbar	entscheidbar	entscheidbar
Prädikatenlogik	strikt unentscheidbar	semi-entscheidbar	semi-entscheidbar

3.10 Aristotelische Syllogismen

Syllogismen sind korrekte Schlüsse, deren Prämissen oder Konklusion entweder bejahend oder verneinend sind und allgemein (allen oder keinen) oder partikulär (einigen, nicht einigen oder nicht jedem) sind.

Urteilsarten

A	universell bejahend	alle P sind Q	$\forall x(P(x) \longrightarrow Q(x))$
E	universell verneinend	kein P ist ein Q	$\forall x\,\neg(P(x) \wedge Q(x))$
I	partikulär bejahend	einige P sind Q	$\exists x(P(x) \wedge Q(x))$
O	partikulär verneinend	einige P sind nicht Q	$\exists x\,\neg(P(x) \longrightarrow Q(x))$

Schema eines Syllogismus

M	u	P
S	v	M
S	w	P

Gültige Syllogismen sind (u.a) $(u, v, w) =$ (A, A, A) Barbara

(E, A, E) Celarent

(A, I, I) Darii

(E, I, O) Ferio (que)

Resolutionsbeweise von Syllogismen

Beispiel 3.10.1. Syllogismus *Celarent*

E:	kein	B	ist	A
A:	Alle	C	sind	B
E:	kein	C	ist	A

1. Wir wandeln in prädikatenlogische Formeln um. A, B, C werden hier als einstellige Prädikate betrachtet. Die Symbole A, B, C stehen dabei für beliebige einstellige Prädikate. Man kann sie als (universell) quantifizierte Prädikatensymbole auffassen. Insofern kann man die Syllogismen als Formeln der Prädikatenlogik zweiter Stufe auffassen, wo Quantifizierung von Funktionen und Prädikaten erlaubt ist.

   E: $\forall x\, \neg(B(x) \wedge A(x))$

   A: $\forall x(C(x) \longrightarrow B(x))$

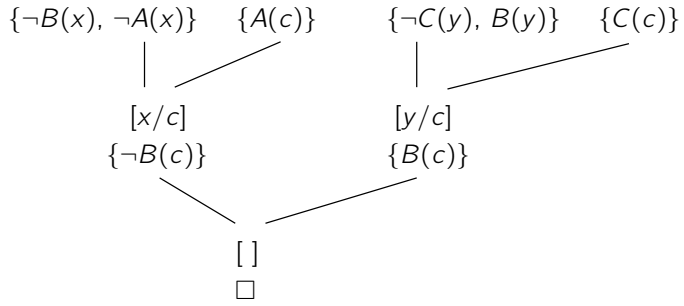
   E: $\forall x\, \neg(C(x) \wedge A(x))$

   $F = E \wedge A \longrightarrow E$

2. Wir negieren F und formen um:

$$\begin{aligned}\neg F &\equiv \forall x\,(\neg B(x) \vee \neg A(x)) \wedge \forall x(C(x) \longrightarrow B(x)) \wedge \exists x(C(x) \wedge A(x)) \\ &(\equiv)\forall x \forall y\,((\neg B(x) \vee \neg A(x)) \wedge (C(y) \longrightarrow B(y)) \wedge (C(c) \wedge A(c))) \\ &(\equiv)\big\{\{\neg B(x), \neg A(x)\}, \{\neg C(y), B(y)\}, \{C(c)\}, \{A(c)\}\big\}\end{aligned}$$

3. Resolutionsbeweis:



Übung 3.10.2 (Wiederholung)**.** Welche der folgenden Aussagen sind wahr bzw. falsch? Begründen Sie bitte Ihre Antworten!

(a) Eine Klauselmenge M ist erfüllbar gdw. jede endliche Teilmenge von M erfüllbar ist.

(b) Formeln in Pränexform enthalten keine Existenzquantoren.

(c) Das Bibelzitat *Ein Kreter sagt: Alle Kreter lügen.* ist in sich widersprüchlich.

(d) Dieser Satz ist falsch.

(e) Wenn 2+2=5 ist, dann schreiben wir das Jahr 1984.

Antworten von Studierenden zum letzten Aufgabenteil:

Die Aussage ist...

- wahr, da ich diese Annahme nicht wiederlegen kann (auch wenn die beiden Aussagen willkürlich erscheinen) und auch nicht beweisen.
- wahr, denn wenn [man] die grundlegenden mathematischen Strukturen als falsch ansieht kann man auch die Jahreszahlen zählen wie man will.
- Wir definieren:

  A: 2+2=5
  B: 1984

  Die Aussagen A und B lassen sich nicht unifizieren, weil sie keine gemeinsame Basis haben.
- falsch, selbst wenn 2+2 fünf wäre, ließe sich daraus nicht gleich auf das Jahr schließen
- falsch, dann schreiben wir 5-2=2
- falsch:

  alle 5 Jahre – 1
  alle 10 Jahre – 2
  alle 100 Jahre – 20
  wäre 2000 $\sim 160^6$
- falsch: $a \cdot a = -b \quad a \cdot a < b \Longrightarrow a \cdot a = c \Longrightarrow b < c$
- falsch, da die beiden Aussagen in keinem Zusammenhang stehen (wenn es so wäre, wäre ich auch noch gar nicht geboren)
- falsch (2+2=4 < 2+2=5 also müsste das Jahr > 2008 sein)
- falsch, da wir, wenn wir immer nur 2+2 rechnen, generell auf 5 kommen. Sagen wir, wir rechnen 4 Jahre nach Christus 2+2 und erhalten 5. Wir merken uns das Ergebnis, und rechnen alle weiteren 4 Jahre 2+2=5, kommen wir nach acht Jahren auf 10, die Jahreszahl 1984 ist aber nicht durch 5 teilbar, daher sage ich die Aussage ist falsch.

George Orwell's Big Brother lässt grüßen...

Literatur

[CGP99] Edmund M. Clarke, Orna Grumberg, and Doron Peled. *Model Checking*. MIT Press, Cambridge, MA, London, 1999.

[CL73] Chin-Liang Chang and Richard Char-Tung Lee. *Symbolic Logic and Mechanical Theorem Proving*. Academic Press, London, 1973.

[CM94] W. F. Clocksin and C. S. Mellish. *Programming in Prolog*. Springer, Berlin, Heidelberg, New York, 4th edition, 1994.

[Fel79] Walter Felscher. *Naive Mengen und abstrakte Zahlen III: Transfinite Methoden*. BI-Wissenschaftsverlag, Mannheim, Wien, Zürich, 1979.

[JK91] Jean-Pierre Jouannaud and Claude Kirchner. Solving equations in abstract algebras: A rule-based survey of unification. In Jean-Louis Lassez and Gordon Plotkin, editors, *Computational Logic: Essays in Honor of Alan Robinson*, pages 257–321. MIT Press, Cambridge, MA, London, 1991.

[Lau11] Dietlinde Lau. *Algebra und Diskrete Mathematik 1*. Springer, Berlin, Heidelberg, New York, 2011.

[Llo87] John W. Lloyd. *Foundations of Logic Programming*. Springer, Berlin, Heidelberg, New York, 1987.

[McC94] William McCune. OTTER 3.0 reference manual and guide. Technical Report ANL-94/6, National Laboratory, Argonne, IL, 1994.

[Sch00] Uwe Schöning. *Logik für Informatiker*. Spektrum Akademischer Verlag, 5th edition, 2000.

[Sto98] Frieder Stolzenburg. *Disjunctive Logic Programming with Constraints and its Applications*. Koblenzer Schriften zur Informatik 7. Fölbach, Koblenz, 1998. Dissertation. Reviewers: Ulrich Furbach and Joxan Jaffar.

[Wit63] Ludwig Wittgenstein. *Tractatus logico-philosophicus – Logisch-philosophische Abhandlung*. Suhrkamp, Frankfurt a. M., 1963.