Logische Grundlagen der Informatik

Skript zur Vorlesung Theoretische Informatik I

© Prof. Peter H. Starke
Lehrstuhl für Automaten- und Systemtheorie

Oktober 2000

Dies ist die gedruckte Version des Wintersemesters 2000/2001. Fast alle bekannten Fehler sind beseitigt. Eine Fehlerliste befindet sich im World Wide Web (WWW) unter

`http://www.informatik.hu-berlin.de/lehrstuehle/automaten/logik/errata.html`

Zur weiteren Korrektur bitte Hinweise auf bestehende Fehler, möglichst mit genauer Lokalisierung (Seite, Zeile und Absatz, ...) und Verbesserungsvorschlag an die Übungsleiter.

Vorwort

Warum und zu welchem Ende studieren wir Theoretische Informatik?

Informatik–Studenten in den ersten Semestern empfinden häufig die Vorlesungen zur Theoretischen Informatik als lästige Zumutung. Sie verstehen die Informatik als eine Wissenschaft des Handelns und Gestaltens und sehen die Theorie als ein Gebirge von Abstraktionen. Sie vergessen dabei, daß mit sinnvollem Handeln und effizientem Gestalten untrennbar das Planen und Entwerfen verbunden sind. Planen und Entwerfen geschieht aber auf der Basis von Modellen, man macht sich ein Modell und spielt seinen Plan daran durch. Geht es um ein größeres Projekt, so muß dieses Modell mitteilbar sein, damit Zusammenarbeit möglich ist. Mitteilbar sein bedeutet aber, daß ein allen Mitarbeitern gemeinsames Netz von Abstraktionen, d. h. ein gemeinsames Begriffssystem, vorhanden ist und ausschließlich genutzt werden muß. Um die angestrebten Ziele des Projektes zu erreichen, muß dieses Begriffssystem präzise Formulierungen ermöglichen. Das einzige Begriffssystem, das diese Forderung erfüllt, ist die abstrakte Begriffswelt der Mathematik. Wie die Physik nutzt die Informatik das mathematische Begriffssystem zur Formulierung ihrer Modelle und der Semantik dieser Modelle.

Deshalb führt diese Vorlesung zunächst in die **Mengenlehre** als Grundlage der Mathematik ein. Hier geht es auch darum, intuitiv vorhandene Begriffe (wie z. B. dem der Relation) scharf zu präzisieren, d. h. eine Sprachregelung zu treffen, genau zu definieren.

Der Informatiker unterscheidet sich vom Programmierer nicht nur darin, daß er gefundene Lösungen bewerten und gegebenfalls verbessern kann. Zur Bewertung gehört neben dem Beweis, daß das geplante System die gestellte Aufgabe auch wirklich löst, auch der Vergleich der theoretisch erreichbaren mit der praktisch realisierten Effizienz. Wir studieren also in dieser Vorlesung, wie Modelle gebildet werden (wie dieser Abstraktionsprozeß verläuft) und welche Fragen bei der Modellbildung beantwortet werden müssen, welche Hilfe uns die Theorie bei der Anwendung dieser Modelle geben kann. Das geschieht an (im wesentlichen) drei Beispielen.

In der **Aussagenlogik** geht es um die Wahrheit bzw. Falschheit von Aussagen und Aussagenverbindungen. Unser Modell, der Aussagenkalkül, stellt unsere Vorstellungen auf eine berechenbare Grundlage: Wir können mit Aussagen rechnen, ihre Wahrheit beweisen. Als Hauptaufgabe der Logik betrachten wir dabei, semantische (d.h. inhaltliche) Beziehungen und Begriffe syntaktisch zu charakterisieren.

In der **Berechnungstheorie** geht es darum, was Computer eigentlich können, also auch darum, was sie nicht können. Wir geben in dieser Vorlesung einen Abschnitt der Berechnungstheorie wieder, der uns beweist, daß es für Computer unlösbare (abstrakte) Probleme gibt. Zu diesen Problemen gehört das Entscheidungsproblem der **Prädikatenlogik**, d. h. die Frage nach einem Verfahren, mit dessen Hilfe von jeder Aussage über die Elemente einer gewissen Menge festgestellt werden kann, ob diese

Aussage richtig ist. Wenn auch ein solches Verfahren (ein Algorithmus) nicht existiert, so kann doch ein Verfahren gefunden werden, das die Richtigkeit richtiger Aussagen bestätigt (für falsche Aussagen aber keine Information liefert).

Spätestens an dieser Stelle bemerkt der Leser, wie ungenaue Formulierungen Verwirrung erzeugen, wie notwendig also ein scharf definiertes Begriffssystem ist. Auch dazu dient uns die Logik (gibt es z. B. einen Unterschied zwischen Wahrheit und Beweisbarkeit?).

Inhaltsverzeichnis

1 Mengen **1**

1.1 Mengenbegriff und Elementbeziehung 1

1.1.1 Arten von Definitionen 1

1.1.2 Die Cantorschen „Definitionen“ 2

1.1.3 Die Russellsche Antinomie 3

1.1.4 Axiomatische Mengenlehre 3

1.1.5 Spezielle Mengen 6

1.2 Mengenalgebra 7

1.2.1 Mengenbeziehungen 7

1.2.2 Durchschnitt, Vereinigung und Komplement 10

1.2.3 Differenzen 16

1.2.4 Allgem. Durchschnitt/Vereinigung und Potenzmengenbildung . 17

1.3 Korrespondenzen, Relationen, Abbildungen 18

1.3.1 Geordnetes Paar und Cartesisches Produkt 19

1.3.2 Korrespondenzen 20

1.3.3 Relationen 22

1.3.4 Abbildungen 25

1.4 Äquivalenzrelationen 26

1.4.1 Äquivalenzrelationen und Zerlegungen 26

1.4.2 Feinheit von Äquivalenzrelationen 28

1.4.3 Kongruenzrelationen 29

1.5 Natürliche Zahlen 30

1.5.1 Gleichmächtigkeit 31

1.5.2 Der Endlichkeitsbegriff 31

1.5.3 Die Menge der natürlichen Zahlen 34

1.6 Wörter 38

1.6.1 Grundbegriffe 38

1.6.2 Induktive Beweise in freien Worthalbgruppen 42

1.6.3 Induktionsbeweise in Algebren 43

1.6.4 Induktive Definitionen 44

1.7 Sprachen . . . 45
1.7.1 Grundbegriffe . . . 46
1.7.2 Beispiel: Sprache der Mengengleichungen . . . 47
1.7.3 Weitere Grundbegriffe . . . 48

2 Aussagenlogik 49
2.1 Grundbegriffe . . . 49
2.2 Syntax und Semantik des Aussagenkalküls . . . 51
2.2.1 Aufbau des Kalküls — Syntax . . . 51
2.2.2 Semantik . . . 53
2.2.3 Weitere semantische Begriffe . . . 54
2.3 Anwendung in der Schaltalgebra . . . 56
2.3.1 Grundbegriffe . . . 56
2.3.2 Optimierte Schaltungen . . . 58
2.4 Folgern . . . 63
2.4.1 Grundbegriffe . . . 63
2.4.2 Beweis des Endlichkeitssatzes für das Folgern . . . 64
2.4.3 Ableitbarkeits- und Deduktionstheorem . . . 66
2.4.4 Der Kalkülbegriff . . . 67
2.5 Aussagenlogisches Schließen . . . 68
2.5.1 Grundbegriffe . . . 68
2.5.2 Nützliche aussagenlogische Schlußweisen . . . 69
2.5.3 Das Haubersche Theorem . . . 71
2.6 Ableitbarkeit . . . 71
2.6.1 Ableiten mit Abtrennungs- und Einsetzungsregel . . . 71
2.6.2 Syntaktische Charakterisierung von ag . . . 77
2.6.3 Cut–Ableitbarkeit . . . 80
2.6.4 Syntaktisches Entscheidungsverfahren für ag . . . 87
2.7 Widerspruchsfreiheit, Vollständigkeit, ... 87
2.7.1 Widerspruchsfreiheit . . . 87
2.7.2 Vollständigkeit . . . 89
2.7.3 Nichtableitbarkeit . . . 91
2.8 Dualitätstheoreme . . . 93

3 Berechnungstheorie 97
3.1 Grundbegriffe . . . 97
3.2 Axiomatische Charakterisierung . . . 99
3.2.1 Anfangsfunktionen . . . 99
3.2.2 Substitution und primitive Rekursion . . . 100
3.2.3 Beispiele für primitiv–rekursive Funktionen . . . 101

3.2.4 Die Peterfunktion 103
3.2.5 Minimum-Operator und partiell-rekursive Funktionen 104
3.3 Der Maschinen–Ansatz 106
3.3.1 Die Turing–Maschine 106
3.3.2 Bausteine für Turing–Maschinen 107
3.4 Äquivalenz der Berechenbarkeitsbegriffe 110
3.4.1 Partiell–rekursive Funktionen sind Turing–berechenbar 110
3.4.2 Turing–berechenbare Funktionen sind partiell–rekursiv 113
3.5 Folgerungen 120

4 Prädikatenlogik 123
4.1 Grundbegriffe 124
4.2 Syntax und Semantik des Prädikatenkalküls 126
4.2.1 Die Sprache des Prädikatenkalküls der ersten Stufe 126
4.2.2 Einige syntaktische Eigenschaften 127
4.2.3 Semantik elementarer Sprachen 129
4.2.4 Weitere semantische Begriffe 131
4.3 Die Unentscheidbarkeit des Prädikatenkalküls 133
4.4 Lösbare Fälle des Entscheidungsproblems 137
4.4.1 Klassenkalkül 137
4.4.2 Aussagenlogische Erfüllbarkeit und Gültigkeit 140
4.5 Folgern und Ableiten 141
4.5.1 Folgern 141
4.5.2 Ableiten 142
4.6 Prädikatenlogisches Schließen 145
4.7 Widerlegen 146
4.7.1 Klauselbildung 147
4.7.2 Herbrand–Modell 149
4.8 Unifikation und Resolution 151
4.8.1 Substitution und Unifikation 152
4.8.2 Berechnung eines allgemeinsten Unifikators 152
4.8.3 Resolution 154
4.9 Logische Programmierung 156
4.9.1 Prolog 156
4.9.2 Ein abstrakter Interpreter 159
4.9.3 Die Auswertungsstrategie von Prolog 160

Literaturverzeichnis 161

Symbolverzeichnis 162

Kapitel 1

Mengen

Die Mengenlehre gilt als Fundament der Mathematik, weil sich alle mathematischen Begriffe, wie z. B. der Zahlenbegriff, der Funktionsbegriff und der Relationsbegriff, auf mengentheoretische Begriffe zurückführen lassen. Dabei bedeutet „zurückführen“ dasselbe wie definieren. In die Denk- und Vorgehensweise der theoretischen Informatik einzudringen ist daher unmöglich, ohne sich mit der Begriffswelt der Mathematik und insbesondere ihres Fundaments, der Mengenlehre, vertraut zu machen. Insofern gehört die Mengenlehre zu den logischen Grundlagen der Informatik.

1.1 Mengenbegriff und Elementbeziehung

In diesem Abschnitt verfolgen wir das Ziel, den Mengenbegriff zu definieren. Dazu müssen wir uns zunächst Rechenschaft darüber ablegen, was wir unter einer Definition verstehen wollen.

1.1.1 Arten von Definitionen

1. *explizite* Definitionen
   Das zu definierende Objekt kommt im definierenden Ausdruck nicht vor.

   (a) Definition durch Aufzählung
   Beispiel
   - Die Menge der Ziffern besteht aus den Elementen $0, 1, \ldots, 9$

   (b) Definition durch Angabe einer charakteristischen Eigenschaft
   Beispiele
   - Die Menge aller Quadratzahlen ist die Menge aller x mit:
     x ist ganze Zahl und es gibt eine ganze Zahl y mit: $y \cdot y = x$
   - Die Menge aller Primzahlen ist die Menge aller n mit:
     n besitzt genau zwei Teiler (1 und n)

   Nachteile:

   - Aufzählung ist unmöglich bei unendlichen Mengen
   - charakteristische Eigenschaft setzt andere Begriffe voraus

2. *implizite* Definitionen
   Das zu definierende Objekt kommt im definierenden Ausdruck vor.

(a) rekursive (induktive) Definition
Beispiele
- Die Fakultätsfunktion $n!$ ist induktiv definiert
  - $0! = 1$
  - $(n+1)! = (n+1)n!$
- Die Folge $(a_i)_{i=0,1,\ldots}$ von natürlichen Zahlen wird definiert durch
  - $a_0 = 0$
  - $a_{i+1} = a_i + 2i + 1 = a_i + i + i + 1$

$\Rightarrow$ eine induktive Definition beschreibt den Aufbau aus (elementaren) Objekten mittels (elementarer) Operationen.

(b) Axiomatische Charakterisierung
Angabe eines Systems von Aussagen, welche die zu charakterisierenden Grundbegriffe in Beziehung setzen.
Beispiele
- Hilberts Axiomatisierung der Geometrie
- Definition der natürlichen Zahlen nach Peano

Definitionen sind nicht wahr oder falsch (höchstens unzweckmäßig), bedürfen aber einer Rechtfertigung hinsichtlich der Frage, ob wenigstens ein Objekt der definierten Art überhaupt existiert.

Bei expliziten Definitionen ist dies i. a. klar: wenn die Elemente $0, 1, \ldots, 9$ existieren, so existiert die Menge der Ziffern. Ganz offen ist dieses Problem bei der axiomatischen Charakterisierung; hierbei geht es doch um die Frage, ob es ein Modell des Axiomensystems gibt, d. h. Objekte, bei denen alle Aussagen dieses Systems wahr sind. (Dann nennt man dieses System widerspruchsfrei.)

1.1.2 Die Cantorschen „Definitionen“: Der naive Mengenbegriff

Der Mengenbegriff gilt als der Grundbegriff der Mathematik in dem Sinne, daß sich alle übrigen mathematischen Begriffe auf der Basis der Begriffe der Menge und der Elementbeziehung exakt begründen lassen. Dementsprechend ist eine *explizite* Definition des Begriffs „Menge“ nicht möglich, denn eine explizite Definition eines Begriffes bedeutet immer, daß dieser Begriff auf andere (grundlegendere) Begriffe zurückgeführt wird.

Die Mengenbildung hat zwei Aspekte:

- operativ-praktischer Aspekt:
  Äpfel in den Korb legen, Perlen auffädeln $\Rightarrow$ Zusammenfassen
- Abstraktionsaspekt:
  Schaffung eines neuen Objektes, Absehen von der Individualität der Elemente, gemeinsame Eigenschaft.

Georg Cantor begründete gegen Ende des vorigen Jahrhunderts die Theorie beliebiger Mengen von Dingen, die Mengenlehre, mit der folgenden intuitiven (naiven) Umschreibung:

Erklärung 1 (G. Cantor)
Eine Menge M ist eine Zusammenfassung von bestimmten wohlunterschiedenen Objekten unserer Anschauung oder unseres Denkens, welche Elemente der Menge M genannt werden, zu einem Ganzen.

Wenn wir den Abstraktionsaspekt stärker betonen wollen, können wir diese Erklärung wie folgt präzisieren:

Erklärung 2

Zu jeder Eigenschaft H der wohlunterschiedenen Objekte unserer Anschauung und unseres Denkens existiert eine Menge M derart, daß für jedes Objekt x gilt:

$$\underbrace{x \text{ ist Element von } M}_{x \in M} \text{ genau dann, wenn } \underbrace{x \text{ die Eigenschaft } H \text{ hat}}_{H(x)}$$

Als abkürzende Schreibweisen benutzen wir künftig:

- $x \in M$ für „x ist ein Element von M“
- $x \notin M$ für „x ist kein Element von M“
- $H(x)$ für „x hat die Eigenschaft H“
- $\{x|H(x)\}$ für „die Menge aller x mit $H(x)$“

Die geschweiften Klammern werden zur Bezeichnung von Mengen reserviert.

1.1.3 Die Russellsche Antinomie

Wenn wir die Erklärung 1 (oder 2) zur Grundlage eines Aufbaus der Mengenlehre (bzw. der Mathematik) verwenden wollen, müssen wir uns nach der Widerspruchsfreiheit dieser Erklärung (dieses Axioms) fragen. Russell, ein englischer Philosoph und Mathematiker, zeigte uns um die Jahrhundertwende mit folgender Überlegung, daß es damit schlecht bestellt ist.

Wir betrachten die Eigenschaft H^*, die durch die Formel $x \notin x$ bestimmt ist, d. h. ein Objekt x unserer Anschauung oder unseres Denkens hat die Eigenschaft H^* genau dann, wenn es *nicht* Element von x ist. Nach der Erklärung 2 existiert dann die Menge $M^* \equiv \{x|x \notin x\}$.

Nun ist M^* selbst ein Objekt unseres Denkens, wir können also fragen, ob $M^* \in M^*$ ist. Es ist $M^* \in M^*$ genau dann, wenn M^* die Eigenschaft H^* hat, also genau dann, wenn $M^* \notin M^*$ ist. Das ist ein (logischer) Widerspruch, die sogenannte Russellsche Antinomie.

1.1.4 Axiomatische Mengenlehre

Wir haben gesehen, daß die Erklärungen 1 und 2 zum Aufbau der Mengenlehre nicht brauchbar sind, andererseits kommen die Erklärungen unseren anschaulichen Vorstellungen von der Mengenbildung entgegen. Was haben wir also bei der Präzisierung bzw. Formulierung unserer Vorstellungen vergessen? Es ist anschaulich klar, daß eine Menge gegenüber den Elementen eine neue Qualität darstellt, etwas durch Abstraktion gewonnenes, das nicht ohne weiteres mit seinen Elementen auf die gleiche Stufe gestellt werden darf.

Eine Lösung, die Mengenbildung soweit einzuschränken, daß keine Mengen mehr gebildet werden können, die zu Antinomien führen (die Russellsche Antinomie ist nur *ein* Beispiel), besteht darin, Mengen verschiedener Stufen $(0., 1., \ldots)$ zu betrachten und die Mengenbildung nur von Stufe zu Stufe zuzulassen.

Eine andere Lösungsmöglichkeit bietet das sogenannte Klassenkalkül, auf dessen Einführung wir hier verzichten wollen. Der interessierte Leser findet bei [2][1] eine

[1] Dies ist ein Verweis auf das Literaturverzeichnis am Ende des Skriptes

Darstellung dieses Ansatzes.

Im folgenden werden wir den axiomatischen Ansatz des STUFENKALKÜLS vorstellen: Wir legen dazu einen Objektbereich E von Elementen, auch Mengen *nullter Stufe* genannt, zugrunde. Durch Zusammenfassung von Elementen bilden wir die Mengen erster Stufe, solche Mengen bilden die Elemente von Mengen zweiter Stufe und so fort.

Wir bezeichnen im folgenden:

- Elemente von E (Mengen nullter Stufe) durch kleine lateinische Buchstaben $a, b, c, \ldots$
- Mengen von Elementen von E (Mengen erster Stufe) durch große lateinische Buchstaben $A, B, \ldots, M, \ldots$
- Mengen zweiter Stufe (Mengensysteme) durch große deutsche Buchstaben $\mathfrak{M}, \mathfrak{N}, \ldots$
- allgemein: Mengen n-ter Stufe ($n = 0, 1, \ldots$) durch große lateinische Buchstaben mit dem oberen Index n : $M^{(n)}, N^{(n)}, \ldots$

Mengenbildungsaxiome
Zu jeder Eigenschaft H von Mengen n-ter Stufe ($n = 0, 1, 2, \ldots$) gibt es eine Menge $M^{(n+1)}$ $n+1$-ter Stufe derart, daß für alle Mengen $N^{(n)}$ n-ter Stufe gilt:

$$N^{(n)} \in M^{(n+1)} \text{genau dann, wenn } H\left(N^{(n)}\right).$$

Aus diesem Mengenbildungsprinzip kann die RUSSELsche Antinomie nicht hergeleitet werden, weil da die Eigenschaft H^*, d. h. $x \notin x$, keine zur Mengenbildung zugelassene Eigenschaft ist.

Zum Aufbau der Mengenlehre reichen die Mengenbildungsaxiome allein nicht aus, wie wir sofort sehen, wenn wir uns mit der Frage beschäftigen, wann Mengen gleich sind, wobei wir voraussetzen, daß uns das für Elemente von E bekannt ist, handelt es sich doch gemäß unserer Anschauung dabei um „wohlunterschiedene“ Objekte. Zwei unterschiedliche Definitionen der Gleichheit bieten sich an:

Definition 1.1.1 (Umfangsgleichheit, extensionale Gleichheit)
Mengen $M^{(n+1)}$ $N^{(n+1)}$ $(n+1)$-ter Stufe heißen umfangsgleich (notiert als Gleichung durch $M^{(n+1)} =_u N^{(n+1)}$), wenn sie dieselben Elemente haben.

Definition 1.1.2 (Eigenschaftsgleichheit)
Mengen $M^{(n+1)}$, $N^{(n+1)}$ $(n+1)$-ter Stufe heißen eigenschaftsgleich ($M^{(n+1)} =_e N^{(n+1)}$), wenn sie dieselben mengentheoretischen Eigenschaften haben, d.h. für jede Eigenschaft H von Mengen $(n+1)$-ter Stufe gilt

$$H\left(M^{(n+1)}\right) \textit{ genau dann, wenn } H\left(N^{(n+1)}\right)$$

Mit der Definition der Umfangsgleichheit präsentieren wir unsere Vorstellung, daß eine Menge durch ihre Elemente bereits eindeutig bestimmt ist, während die Eigenschaftsgleichheit besagt, daß Gleichheit die Gemeinsamkeit in allen Eigenschaften impliziert. Naheliegend ist es daher, nach der Beziehung zwischen diesen beiden Begriffen zu fragen.

Satz 1.1.3
Wenn Mengen $(n+1)$-ter Stufe eigenschaftsgleich sind, dann sind sie auch umfangsgleich,d.h. wenn $M^{(n+1)} =_e N^{(n+1)}$, so $M^{(n+1)} =_u N^{(n+1)}$ $(n = 0, 1, \ldots)$.

Beweis
Wir haben zu zeigen, daß (unter der Voraussetzung $M^{(n+1)} =_e N^{(n+1)}$) für alle Mengen $X^{(n)}$ n-ter Stufe gilt

$$X^{(n)} \in M^{(n+1)} \text{genau dann ,wenn } X^{(n)} \in N^{(n+1)},$$

denn dann und nur dann haben $M^{(n+1)}$ und $N^{(n+1)}$ dieselben Elemente.

Es sei nun $X^{(n)}$ eine beliebige Menge n-ter Stufe. Wir betrachten die Eigenschaft $H^{X^{(n)}}$ von Mengen $(n+1)$-ter Stufe, mit:

$$H^{X^{(n)}}\left(Y^{(n+1)}\right) \text{genau dann, wenn } X^{(n)} \in Y^{(n+1)}.$$

Weil $M^{(n+1)} =_e N^{(n+1)}$ ist, gilt

$$\begin{array}{lll} X^{(n)} \in M^{(n+1)} & \text{gdw.} \quad H^{X^{(n)}\left(M^{(n+1)}\right)} & (\text{nach Def. von } H^{X^{(n)}}) \\ & \text{gdw.} \quad H^{X^{(n)}\left(N^{(n+1)}\right)} & (\text{wegen } M^{(n+1)} =_e N^{(n+1)}) \\ & \text{gdw.} \quad X^{(n)} \in N^{(n+1)} & (\text{nach Def. von } H^{X^{(n)}}), \end{array}$$

also gilt, was zu zeigen war, und der Satz 1.1.3 ist bewiesen. □[2]

Die Umkehrung des Satzes 1.1.3, d. h. die Aussage:

$$\text{Wenn } M^{(n+1)} =_u N^{(n+1)}, \text{ so } M^{(n+1)} =_e N^{(n+1)}$$

kann dagegen aus den Mengenbildungsaxiomen nicht abgeleitet werden. Wir fordern daher die

Extensionalitätsaxiome
Wenn Mengen $(n+1)$-ter Stufe umfangsgleich (extensional gleich) sind, so sind sie eigenschaftsgleich.

Nach dieser Forderung sind Umfangsgleichheit und Eigenschaftsgleichheit identisch, wir sprechen daher nur noch von Gleichheit schlechthin und schreiben $M = N$, wenn Mengen M, N gleich sind.

Um genügend „Material“ für die Mengenbildung zu haben, fordern wir ferner das

Unendlichkeitsaxiom
Es gibt eine unendliche Menge erster Stufe.

Anschaulich bedeutet das, daß unser Objektbereich E unendlich viele Elemente hat, was aber „unendlich“ im Sinne der Mengenlehre bedeutet, kann an dieser Stelle noch nicht präzisiert werden. Ein weiteres Axiomenschema, das aus den bereits angegebenen Axiomen nicht bewiesen werden kann (von diesen unabhängig ist), bilden die sogenannten

[2] Mit diesem Symbol bezeichnen wir das Ende eines Beweises

Auswahlaxiome
Zu jeder nichtleeren Menge $M^{(n+2)}$ $(n+2)$-ter Stufe (zu jedem Mengensystem) mit paarweise disjunkten nichtleeren Mengen $N^{(n+1)}$ als Elementen gibt es eine Auswahlmenge $A^{(n+1)}$, die mit jedem $N^{(n+1)}$ aus $M^{(n+2)}$ genau ein Element $X^{(n)}$ gemeinsam hat.

Dabei heißt eine Menge *nichtleer*, wenn sie (wenigstens) ein Element besitzt und Mengen (gleicher Stufe) heißen *disjunkt*, wenn sie kein Element gemeinsam haben.

Mit den bisher angegebenen Axiomen kommt man nun tatsächlich beim Aufbau der Mengenlehre aus. Dabei muß an dieser Stelle die Rolle der Auswahlaxiome unklar bleiben; ihre Bedeutung kann erst eine ihrer Anwendungen beim Beweis tiefliegender Sätze der Mengenlehre klar machen.

Bevor wir uns mit dem Rechnen mit Mengen (der sogenannten Mengenalgebra) beschäftigen, wollen wir noch einige wichtige Mengen definieren und benennen.

1.1.5 Spezielle Mengen

Leere Menge und Allmenge

Betrachten wir die durch $x \neq x$ gegebene Eigenschaft $H^{\neq}$ von Mengen 0-ter Stufe (Elementen). Nach dem Mengenbildungsaxiom gibt es eine Menge M_0 mit

$$x \in M_0 \text{ gdw. } x \neq x$$

für alle Elemente x aus dem Objektbereich E. Weil jedes Element x zu sich selbst gleich ist, enthält M_0 *kein* Element. Ist andererseits M' eine Menge erster Stufe, die kein Element enthält, so enthält M' offenbar dieselben Elemente wie M_0 (nämlich keine), nach dem Extensionalitätsaxiom gilt also $M' = M_0$.

Dieselbe Überlegung für höhere Stufen $n = 1, 2, \ldots$ beweist den

Satz 1.1.4
Zu jeder Stufe $n+1$ gibt es genau eine Menge M_0 $(n+1)$-ter Stufe derart, daß für alle Mengen N n-ter Stufe gilt $N \notin M_0$.

Nachdem wir gezeigt haben, daß es eine, aber auch nur eine solche Menge gibt, dürfen wir von *der* Menge M_0 $(n+1)$-ter Stufe sprechen.

Definition 1.1.5 (Leere Menge)
Die nach Satz 1.1.4 bestimmte Menge $M_0^{(n+1)} = \left\{N^{(n)} | N^{(n)} \neq N^{(n)}\right\}$ heißt die leere Menge *$(n+1)$-ter Stufe und wird durch $\emptyset^{(n+1)}$ notiert. Für $\emptyset^{(1)}$ schreiben wir auch $\emptyset$.*

In analoger Weise zeigt man mittels der Eigenschaft $x = x$, daß es zu jeder Stufe $(n+1)$ $(n = 0, 1, \ldots)$ genau eine *Allmenge* $E^{(n+1)}$ $(n+1)$-ter Stufe gibt, die genau die Mengen n-ter Stufe als Elemente hat. Dabei ist $E^{(1)}$ gerade der Objektbereich E, d. h. $E^{(1)} = E$. Die Allmenge $E^{(2)}$ zweiter Stufe bezeichnen wir mit $\mathfrak{E}$.

Einer- und Zweiermengen

Zu jedem Element $a \in E$ existiert nach dem Mengenbildungsaxiom die Menge M für die gilt:

- $a \in M$
- Für jedes b gilt: Wenn $b \in M$, so $b = a$.

Wir schreiben M als $\{a\}$ und bezeichnen sie als *Einermenge*.

Für Elemente a, b ist dementsprechend $\{a, b\}$ die Menge M, für die gilt:

- $a \in M$, $b \in M$
- Für jedes c gilt: Wenn $c \in M$, so $c = a$ oder $c = b$.

Ist dabei $a \neq b$, so wird $\{a, b\}$ als *Zweiermenge* bezeichnet.

1.2 Mengenalgebra

In diesem Abschnitt beschäftigen wir uns mit den wichtigsten Beziehungen und Operationen zwischen Mengen. Da es sich dabei (bis auf die Potenzmengenbildung) um Beziehungen und Operationen für Mengen gleicher Stufe handelt, lassen wir die Stufenindizes weg bzw. definieren die Beziehungen und Operationen für Mengen erster Stufe.

1.2.1 Mengenbeziehungen

Definition 1.2.1 (Inklusion, Teilmengenbeziehung)
Die Menge M heißt Teilmenge der Menge N *oder* in N enthalten *(notiert durch $M \subseteq N$), wenn für alle Elemente x gilt:*

$$\textit{wenn } x \in M, \textit{ so } x \in N$$

Die grundlegenden Eigenschaften der Inklusion vermittelt der

Satz 1.2.2
Für beliebige Mengen M, N, L gilt:

1. $M \subseteq M$ *(Reflexivität)*
   (d. h. die Inklusion ist eine reflexive Relation*)*
2. *Wenn $M \subseteq N$ und $N \subseteq L$, so $M \subseteq L$* *(Transitivität)*
   (d. h. die Inklusion ist eine transitive *Relation)*
3. *Wenn $M \subseteq N$ und $N \subseteq M$, so $M = N$* *(Antisymmetrie)*
   (d. h. die Inklusion ist eine antisymmetrische Relation).

Beweis
Dieser Satz wird durch Rückgang auf die Definition der Inklusion bewiesen, d. h. man geht zurück auf die Stufe der Elemente und nimmt logische Umformungen der Aussagen auf dieser Stufe vor. Als Beispiel führen wir den Beweis der zweiten Aussage vor.

Aus der Voraussetzung $M \subseteq N$ und $N \subseteq L$ ist $M \subseteq L$ zu beweisen. Wenn wir vereinbaren, daß x ein beliebiges Element ist, so ergibt sich nach Einsetzen der Definition der Inklusion, daß folgendes zu beweisen ist:

Wenn:	wenn $x \in M$, so $x \in N$	(a)
und:	wenn $x \in N$, so $x \in L$	(b)
so:	wenn $x \in M$, so $x \in L$	(c)

Es sei nun $x \in M$. Dann haben wir

$$x \in M \text{und (wenn } x \in M\text{, so } x \in N\text{)},$$

folglich haben wir $x \in N$, zusammen mit (b) folgt $x \in L$; wir haben

$$\text{wenn } x \in M\text{, so } x \in L,$$

d. h. (c).

Die zweite Aussage des Satzes 1.2.2 ist damit bewiesen. □

Übung 1
Man beweise die restlichen Aussagen von Satz 1.2.2 sowie

Satz 1.2.3
Für alle Mengen M gilt $\emptyset \subseteq M$.

Wir führen jetzt einen Abstraktionsschritt durch:

Reflexive Halbordnung
Bei der Betrachtung des Systems $\mathfrak{E}$ aller Mengen (erster Stufe) und der Relation der Inklusion (als Beziehung zwischen den Elementen von $\mathfrak{E}$) abstrahieren wir von der Natur dieser Elemente (daß es sich um Mengen handelt). Die Struktur des betrachteten Objekts $[\mathfrak{E}, \subseteq]$ stellt sich damit dar, als eine Menge M von irgendwelchen Elementen, zwischen denen eine Relation $\sqsubseteq$ gegeben ist, die folgende Eigenschaften hat:

1. $\sqsubseteq$ ist reflexiv über M
   für alle $a \in M$ gilt $a \sqsubseteq a$
2. $\sqsubseteq$ ist transitiv
   für alle $a, b, c \in M$ gilt: Wenn $a \sqsubseteq b$ und $b \sqsubseteq c$, so $a \sqsubseteq c$
3. $\sqsubseteq$ ist antisymmetrisch
   für alle $a, b \in M$ gilt: Wenn $a \sqsubseteq b$ und $b \sqsubseteq a$, so $a = b$

Eine solche Struktur $[M, \sqsubseteq]$ wird *reflexive Halbordnung* genannt. Die Ordnung ist nicht total, weil es Elemente $a, b \in M$ geben kann, für die weder $a \sqsubseteq b$ noch $b \sqsubseteq a$ gilt.

Beispiel
Die Teilbarkeitsbeziehung in der Menge der natürlichen Zahlen

$$n \mid m \text{ gdw. es ein } k \in \mathbb{N} \text{ mit } n \cdot k = m \text{ gibt}$$

ist eine reflexive Halbordnung, aber keine Ordnung, d. h. nicht für alle $n, m \in \mathbb{N}$ gilt: Entweder $n \mid m$ oder $m \mid n$.

Folgerung 1.2.4
$[\mathfrak{E}, \subseteq]$ *ist eine reflexive Halbordnung.*

Definition 1.2.5 (echte Inklusion, echte Teilmenge)
Die Menge M wird echte Teilmenge von N *oder* in N echt enthalten *(notiert durch $M \subset N$) genannt, wenn $M \subseteq N$ und $M \neq N$ ist.*

Man zeigt nun

Satz 1.2.6
Für beliebige Mengen M, N, L gilt

1. *$M \not\subset M$* *(Irreflexivität)*
2. *Wenn $M \subset N$ und $N \subset L$, so $M \subset L$* *(Transitivität)*
3. *Wenn $M \subset N$, so $N \not\subset M$* *(Asymmetrie)*

Man sieht hier, daß die Asymmetrie der echten Inklusion aus der Irreflexivität und der Transitivität folgt:
Wäre die Asymmetrie für M, N verletzt, d.h.

$$M \subset N \text{ und } N \subset M$$

so ergäbe sich aus der Transitivität (für $L = M$)

$$M \subset M$$

also ein Widerspruch zur Irreflexivität.

Wir machen nun den gleichen Abstraktionsschritt wie oben und erhalten

Irreflexive Halbordnung
Eine Struktur $[M, \sqsubset]$, bei der die Relation $\sqsubset$ irreflexiv (für alle $a \in M$ gilt nicht $a \sqsubset a$) und transitiv ist, wird *irreflexive Halbordnung* genannt. Eine solche Relation ist immer asymmetrisch.

Folgerung 1.2.7
$[\mathfrak{E}, \subset]$ *ist eine irreflexive Halbordnung.*

Bemerkung
Wir sehen an der Definition der echten Inklusion, daß in jeder reflexiven Halbordnung $[M, \sqsubseteq]$ eine irreflexive Halbordnung $\sqsubset$ durch die Definition

$$a \sqsubset b \text{ gdw. } a \sqsubseteq b \text{ und } a \neq b$$

eingeführt werden kann. Umgekehrt kann in jeder irreflexiven Halbordnung $[M, \sqsubset]$ eine reflexive Halbordnung $\sqsubseteq$ durch

$$a \sqsubseteq b \text{ gdw. } a \sqsubset b \text{ oder } a = b$$

definiert werden.

Für die Teilmengenbeziehungen ergibt sich daher

Satz 1.2.8
Für beliebige Mengen M, N, L gilt

1. *Wenn $M \subset N$, so $M \subseteq N$*

2. *$M \subseteq N$ genau dann, wenn $M = N$ oder $M \subset N$*
3. *Wenn $M \subseteq N$ und $N \subset L$, so $M \subset L$*
4. *Wenn $M \subseteq N$, so $N \not\subset M$.*

Übung 2
Man formuliere die Beweise der Sätze 1.2.6 und 1.2.8 und eine Definition der echten Inklusion, in der die Relationen $\subseteq$, $\neq$ (für Mengen) nicht verwendet werden, d. h. in der nur die Elementbeziehung vorkommt.

1.2.2 Durchschnitt, Vereinigung und Komplement

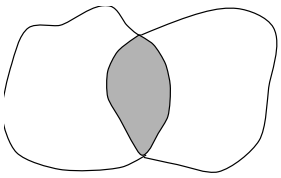
Definition 1.2.9 (Durchschnitt, Vereinigung und Komplement)
Es seien M, N beliebige Mengen. Dann ist

1. *der* Durchschnitt *$M \cap N$ von M und N die Menge*
$$M \cap N =_{Def} \{x | x \in M \text{ und } x \in N\}$$
2. *die* Vereinigung *$M \cup N$ von M und N die Menge*
$$M \cup N =_{Def} \{x | x \in M \text{ oder } x \in N\}$$
3. *das* Komplement *$\overline{M}$ von M die Menge*
$$\overline{M} =_{Def} \{x | x \notin M\}\,.$$

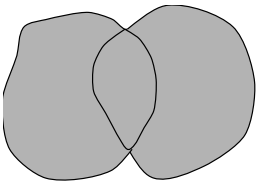
In der Definition der Vereinigung ($=_{\text{Def}}$ bedeutet, daß die Mengen nach Definition gleich sind) ist das „oder“ im nicht ausschließenden Sinn zu verstehen (d. h. nicht als „entweder-oder“); zur Vereinigung von M und N gehören also auch die Elemente x, die in beiden Mengen Element sind. Zur Veranschaulichung der Wirkungsweise der Operatoren $\cup$, $\cap$ und $\overline{}$ beschaut man sich oft Zeichnungen der folgenden Art:

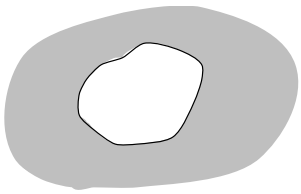
Veranschaulichung

Wenn M die Menge der Punkte innerhalb des linken und N entsprechend die Menge der Punkte innerhalb des rechten Kreises ist, so gibt die schraffierte Fläche gerade den Durchschnitt $M \cap N$ wieder.



Die entsprechenden Bilder der Vereinigung und des Komplements sehen so aus:





Wir formulieren nun eine Reihe von Sätzen über das Rechnen mit den in 1.2.9 definierten sogenannten BOOLEschen Operationen, deren Beweis wir abermals dem Leser überlassen.

Satz 1.2.10
Für beliebige Mengen M, N, L gilt:

1. $M \cap N = N \cap M$, $\quad M \cup N = N \cup M$ *(Kommutativität)*
2. $(M \cap N) \cap L = M \cap (N \cap L)$, $\quad (M \cup N) \cup L = M \cup (N \cup L)$ *(Assoziativität)*
3. $M \cap (M \cup N) = M$, $\quad M \cup (M \cap N) = M$ *(Absorptionsgesetze)*

Bemerkung
Eine Menge V, in der zweistellige Operationen $\sqcap$ und $\sqcup$ definiert sind, die beide kommutativ und assoziativ sind und die Absorptionsgesetze erfüllen, wird *Verband* genannt, d. h. wir definieren

Verband
$\mathfrak{V} = [V, \sqcap, \sqcup]$ heißt Verband, wenn V eine Menge ist und:

1. $\sqcap$, $\sqcup$ zweistellige Operationen über V sind; $\sqcap$ und $\sqcup$ ordnen also jedem $a, b \in V$ je (genau) ein Element $a \sqcap b$, $a \sqcup b \in V$ zu;
2. $\sqcap$, $\sqcup$ sind kommutativ
3. $\sqcap$, $\sqcup$ sind assoziativ
4. $\sqcap$, $\sqcup$ erfüllen die Absorptionsgesetze, d. h. für alle $a, b, \in V$ gilt

$$a \sqcap (a \sqcup b) = a \text{ und } a \sqcup (a \sqcap b) = a.$$

Bezeichnen wir mit $\mathfrak{E}$ das System aller Mengen erster Stufe, so gilt also

Folgerung 1.2.11
$[\mathfrak{E}, \cap, \cup]$ *ist ein Verband.*

Der Verbandsbegriff ist ein wichtiger algebraischer Grundbegriff. Sein wesentlicher Inhalt besteht in einer Verallgemeinerung der Struktur des Rechnens mit Mengen auf andere Bereiche. Wir werden z. B. sehen, daß auch die Menge der Äquivalenzrelationen (über einer gegebenen Menge) einen Verband bildet, so daß man gewisse Rechnungen mit Äquivalenzrelationen so durchführen kann (d. h. nach den gleichen Rechenregeln durchführen kann) als hätte man Mengen vor sich.

Satz 1.2.12
Für Mengen M, N gilt offenbar

$$\begin{array}{lll} M \cap N = M \text{ gdw. } M \subseteq N & (1) & \text{und} \\ M \cup N = N \text{ gdw. } M \subseteq N & (2) & \end{array}$$

Was passiert, wenn wir einen Verband $[V, \sqcap, \sqcup]$ hernehmen und (1) oder (2) als Definition für eine Relation in V nehmen:

$$a \sqsubseteq b \text{ gdw. } a \sqcap b = a$$

Dann gilt

Satz 1.2.13
$[V, \sqsubseteq]$ *ist eine reflexive Halbordung.*

Beweis

1. $\sqsubseteq$ ist reflexiv: Für alle $a \in V$ gilt $a \sqsubseteq a$, d. h. $a \sqcap a = a$.

   Das sieht man wie folgt:
   Nach Absorptionsgesetz ist $a = a \sqcup (a \sqcap b)$ (für beliebiges $b \in V$)
   also
   $$\begin{array}{rcll} a \sqcap a & = & a \sqcap (a \sqcup (a \sqcap b)) & \\ & = & a \sqcap (a \sqcup c) & (\text{für } c = a \sqcap b\ ) \\ & = & a & (\text{wegen Absorption}) \end{array}$$

2. $\sqsubseteq$ ist transitiv: Wenn $a \sqsubseteq b$ und $b \sqsubseteq c$, so $a \sqsubseteq c$.

   Unsere Voraussetzungen sind also: $a \sqcap b = a$ und $b \sqcap c = b$
   und zu zeigen ist: $a \sqcap c = a$

   Es ist $b \sqcap c = b$, also $a \sqcap (b \sqcap c) = a \sqcap b = a$
   und daher $a \sqcap c = (a \sqcap b) \sqcap c = a \sqcap (b \sqcap c) = a$

3. $\sqsubseteq$ ist antisymmetrisch: Wenn $a \sqsubseteq b$ und $b \sqsubseteq a$, so $a = b$.

   nämlich: $a \sqcap b = a$ und $b \sqcap a = b$

   also wegen Kommutativität: $a = a \sqcap b = b \sqcap a = b$

$\square$

In der Halbordnung $[V, \sqsubseteq]$ gilt überdies stets

$$\begin{array}{c} a \sqsubseteq a \sqcup b \text{ (d. h. } a \sqcap (a \sqcup b) = a) \quad \text{und} \\ a \sqcap b \sqsubseteq a \text{ (d. h. } a \sqcap b \sqcap a = a \sqcap b) \end{array}$$

wie sich aus den Absorptionsgesetzen ergibt.

Folglich besitzen je zwei Elemente a, b bezüglich $\sqsubseteq$ eine obere Schranke, nämlich $a \sqcup b$, und eine untere Schranke, nämlich $a \sqcap b$ und es gilt:

$$\begin{array}{l} \text{Wenn } x \sqsubseteq a \text{ und } x \sqsubseteq b, \text{ so } x \sqsubseteq a \sqcap b \quad \text{und} \\ \text{Wenn } a \sqsubseteq x \text{ und } b \sqsubseteq x, \text{ so } a \sqcup b \sqsubseteq x \end{array}$$

d. h. $a \sqcap b$ ist die bezüglich $\sqsubseteq$ größte untere Schranke von a,b und $a \sqcup b$ ist bezüglich $\sqsubseteq$ die kleinste obere Schranke von a,b.

Ist umgekehrt $[V, \sqsubseteq]$ eine reflexive Halbordnung derart, daß zu beliebigen Elementen a,b stets die bezüglich $\sqsubseteq$ größte untere und die kleinste obere Schranke existieren, so ist V mit den Operationen

$$\begin{array}{rcl} a \sqcap b & = & \text{die größte untere Schranke von } a, b \\ a \sqcup b & = & \text{die kleinste obere Schranke von } a, b \end{array}$$

ein Verband.

Übung 3
Man weise nach, daß $[N^+, ggT, kgV]$ ein Verband ist.

Satz 1.2.14 (Fortsetzung von Satz 1.2.10)

1. $\begin{array}{rcl}(M \cap N) \cup L & = & (M \cup L) \cap (N \cup L) \\ (M \cup N) \cap L & = & (M \cap L) \cup (N \cap L)\end{array}$ *(Rechtsdistributivität)*

2. $\emptyset \cap M = \emptyset$, $\emptyset \cup M = M$ *($\emptyset$ ist Nullelement)*

3. $E \cap M = M$, $E \cup M = E$ *(E ist Einselement)*

4. $M \cap \overline{M} = \emptyset$, $M \cup \overline{M} = E$

Bemerkung
Aus der Kommutativität und der Rechtsdistributivität folgt unmittelbar die Linksdistributivität, d. h. es gilt

$$\begin{array}{rcl} L \cup (M \cap N) & = & (M \cup L) \cap (N \cup L) \\ L \cap (M \cup N) & = & (M \cap L) \cup (N \cap L) \end{array}$$

Wir sprechen daher künftig nur von der Distributivität.

Ein Verband $\mathfrak{V} = [V, \sqcap, \sqcup]$, bei dem die Operationen $\sqcap$ und $\sqcup$ die distributiven Gesetze erfüllen, wird *distributiver Verband* genannt; $[\mathfrak{E}, \cap, \cup]$ ist also ein distributiver Verband.

Die Bezeichnung „Nullelement“ bzw. „Einselement“ werden klar, wenn man eine Analogie zwischen dem Durchschnitt von Mengen und der Addition von Zahlen herstellt. Diese Analogie geht aber nicht so weit, daß etwa die Menge der natürlichen Zahlen $\mathbb{N}$ mit Multiplikation und Addition einen Verband bildet, offenbar sind die Absorptionsgesetze nicht erfüllt (jedoch ist $[\mathbb{N}, \min, \max]$ ein Verband).

Ein Verband $[V, \sqcap, \sqcup]$, in dem es Elemente $0, 1 \in V$ gibt, so daß (1.2.14.2) und (1.2.14.3) erfüllt sind, wird Verband mit Null und Eins genannt.

In jedem Verband gibt es höchstens ein Nullelement und höchstens ein Einselement, so daß bei einem Verband mit Null und Eins von *dem* Nullelement und *dem* Einselement gesprochen werden kann. Man beweist die Einzigkeit des Nullelements wie folgt (analog führt man den Beweis für das Einselement):

Es seien $0, 0' \in V$ mit (5) $0 \sqcap a = 0$ und (5′) $0' \sqcap a = 0'$ für alle $a \in V$. Dann gilt:

$$\begin{array}{rcll} 0 & = & 0 \sqcap 0' & \text{(wegen (5) und } a = 0' \in V) \\ & = & 0' \sqcap 0 & \text{(wegen der Kommutativität von } \sqcap) \\ & = & 0' & \text{(wegen (5′) und } a = 0 \in V) \end{array}$$

also ist $0 = 0'$, was zu zeigen war.

Übung 4
Gibt es für $[N^+, ggT, kgV]$ ein Null- und Einselement?
Ist es ein distributiver Verband?

Ein Verband $\mathfrak{V} = [V, \sqcap, \sqcup, 0, 1]$, in dem eine dritte einstellige Operation $^{-}$ definiert ist, die jedem $a \in V$ ein solches Element $\overline{a} \in V$ zuordnet, daß stets $a \sqcap \overline{a} = 0$, $a \sqcup \overline{a} = 1$ heißt *komplementärer Verband*. Ein distributiver komplementärer Verband (mit Null und Eins) wird BOOLEsche Algebra genannt.

Folgerung 1.2.15
$[\mathfrak{E}, \cap, \cup, ^{-}, \emptyset, E]$ *ist eine* BOOLE*sche Algebra.*

Wir können bereits an dieser Stelle eine Folgerung aus unseren verbandstheoretischen Überlegungen ziehen. Aus dem Beweis der Einzigkeit des Nullelements ergibt sich nämlich:

Folgerung 1.2.16
Ist M_0 eine Menge derart, daß für alle Mengen N gilt $M_0 \cap N = \emptyset$, dann ist $M_0 = \emptyset$.

Übung 5
Wie lautet das Analogon dieser Aussage für die Allmenge E ?

Dualitätstheorem

Sieht man sich die Form der bisher aufgeführten Rechenregeln an, so fällt auf, daß man aus jeder Gleichung durch Vertauschen von $\cap$ und $\cup$ wieder gültige Gleichungen erhält. In der Tat gilt das

Dualitätstheorem
Es sei G eine gültige Gleichung, in der nur Variablen für Mengen, Klammern und die Zeichen $\cap$, $\cup$ vorkommen und G' entsteht aus G, indem an allen Stellen gleichzeitig das Zeichen $\cap$ durch das Zeichen $\cup$ und das Zeichen $\cup$ durch das Zeichen $\cap$ ersetzt wird. Dann ist G' eine gültige Geichung.

Der Beweis für das Dualitätstheorem führen wir im Kapitel Aussagenlogik ab Seite 93 mit den dort entwickelten Hilfsmitteln.

Weitere Sätze

Satz 1.2.17
Für beliebige Mengen M, N, L gilt:

1. $M \cap M = M, \quad M \cup M = M$
2. $M \cap N \subseteq M, \quad M \subseteq M \cup N$
3. $M \subseteq N$ *und* $M \subseteq K$ *gdw.* $M \subseteq N \cap K, \quad M \subseteq K$ *und* $N \subseteq K$ *gdw.* $M \cup N \subseteq K$
4. *Wenn* $M \subseteq N$, *so* $M \cap L \subseteq N \cap L$ *und* $M \cup L \subseteq N \cup L$ *(Monotonie)*
5. *Wenn* $M \supseteq K$, *so* $M \cap (N \cup K) = (M \cap N) \cup K$
   Wenn $M \subseteq K$, *so* $M \cup (N \cap K) = (M \cup N) \cap K$ *(Modularität)*
6. $\overline{\overline{M}} = M$
7. $\overline{M \cap N} = \overline{M} \cup \overline{N}$
   $\overline{M \cup N} = \overline{M} \cap \overline{N}$ *(De Morgansche Formeln)*
8. $M \subseteq N$ *gdw.* $M \cap \overline{N} = \emptyset \quad M \subseteq N$ *gdw.* $\overline{M} \cup N = E$

Die Aussagen dieses Satzes gelten in beliebigen Booleschen Algebren, sofern dort durch Verwendung des Satzes 1.2.12 als Definition eine „Inklusion“ zwischen den Verbandselementen definiert ist. Einige dieser Beweise geben wir im folgenden an.

Beweis der ersten Aussage
Aus (1.2.10.3) folgt

$$\begin{array}{rcll} M \cap M & = & M \cap [M \cup (M \cap N)] & (\text{wegen}\, M = M \cup (M \cap N)\,,\ (1.2.10.3)) \\ & = & M \cap [M \cup K] & (\text{für} K = M \cap N\ ) \\ & = & M & (\text{wegen}\ (1.2.10.3)) \end{array}$$

□

Beweis der zweiten Aussage
Wegen Satz 1.2.12 gilt

$$\begin{array}{lll} M \cap N \subseteq M & \text{gdw.} & (M \cap N) \cup M = M \\ & \text{gdw} & M \cup (M \cap N) = M \quad (\text{wegen}\ (1.2.10.1)) \end{array}$$

und $M = M \cup (M \cap N)$ ist nach (1.2.10.3) wahr. □

Beweis der dritten Aussage
Wir zeigen zuerst:

$$\text{wenn } M \subseteq N \text{ und } M \subseteq K, \text{ so } M \subseteq N \cap K.$$

Aus $M \subseteq N$, $M \subseteq K$ ergibt sich mit Satz 1.2.12

$$M \cap N = M \quad \text{und} \quad M \cap K = M,$$

woraus nach Aussage eins folgt

$$\begin{array}{rcll} M = M \cap M & = & (M \cap N) \cap (M \cap K) & \\ & = & (M \cap M) \cap (N \cap K) & (\text{wegen}\ (1.2.10.2)\ \text{und}\ (1.2.10.1)) \\ & = & M \cap (N \cap K) & (\text{wegen Aussage zwei}) \end{array}$$

also gilt nach Satz 1.2.12 auch $M \subseteq N \cap K$.

Ist umgekehrt $M \subseteq N \cap K$, d. h. $M = M \cap (N \cap K)$, so gilt nach Aussage zwei und (1.2.10.2)

$$M = (M \cap N) \cap K \subseteq M \cap N$$

Andererseits ist nach Aussage zwei $M \cap N \subseteq M$, folglich gilt bei $L = M \cap N$

$$L \subseteq M \quad \text{und} \quad M \subseteq L.$$

Aus Satz 1.2.12 ergibt sich nun mit (1.2.10.1) $L = L \cap M = M \cap L = M$ d. h. $M = M \cap N$, folglich $M \subseteq N$. Analog zeigt man $M \subseteq K$. □

Übung 6
Man führe die Beweise für die vierte und fünfte Aussage.

Beweis der sechsten Aussage
Wegen (1.2.14.4) gilt

$$\overline{M} \cap \overline{\overline{M}} = \emptyset$$

folglich ist $M = M \cup \emptyset = M \cup \left(\overline{M} \cap \overline{\overline{M}}\right)$ nach (1.2.14.2).
Mit (1.2.14.1) erhalten wir

$$M = \left(M \cup \overline{M}\right) \cap \left(M \cup \overline{\overline{M}}\right)$$

d. h. über (1.2.14.3)

$$M = E \cap \left(M \cup \overline{\overline{M}}\right) = M \cup \overline{\overline{M}}$$

Auf die gleiche Weise leitet man aus $\overline{M} \cup \overline{\overline{M}} = E$ her, daß

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

ist, woraus

$$\overline{\overline{M}} = \overline{\overline{M}} \cup \left(\overline{\overline{M}} \cap M\right) = \overline{\overline{M}} \cup M = M,$$

d. h. Aussage sechs folgt. □

Übung 7
Man beweise die siebte und achte Aussage.

1.2.3 Differenzen

Definition 1.2.18 (Differenz und symmetrische Differenz)
Die Differenz $M \setminus N$ der Mengen M, N ist die Menge

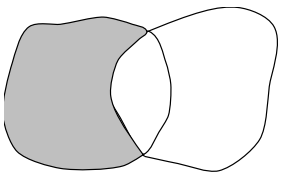
$$M \setminus N =_{Def} \{x | x \in M \text{ und } x \notin N\}$$

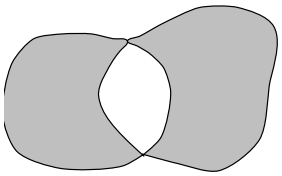
Die symmetrische Differenz $M \triangle N$ der Mengen M, N ist die Menge

$$M \triangle N =_{Def} \{x | \text{ entweder } x \in M \text{ oder } x \in N\}$$

Veranschaulichung

Folgende Bilder geben die Differenz $M \setminus N$ (links) und die symmetrischen Differenz $M \triangle N$ (rechts) wieder:





Einige Regeln für das Rechnen mit Differenz bzw. symmetrischer Differenz geben die beiden folgenden Sätze wider:

Satz 1.2.19
Für beliebige Mengen M, N gilt

1. $M \setminus N = M \cap \overline{N}$
2. $E \setminus M = \overline{M}$
3. $M \setminus \emptyset = M$
4. $M \subseteq N$ *gdw.* $M \setminus N = \emptyset$
5. *Wenn* $M \cap N = \emptyset$, *so* $M \setminus N = M$ *und* $N \setminus M = N$.

Satz 1.2.20
Für beliebige Mengen M, N gilt:

1. $M \triangle N = \left(M \cap \overline{N}\right) \cup \left(\overline{M} \cap N\right)$
2. $M \triangle N = (M \cup N) \cap \overline{(M \cap N)} = (M \cup N) \setminus (M \cap N)$
3. $M \triangle N = N \triangle M$ *(Kommutativität)*
4. $(M \triangle N) \triangle K = M \triangle (N \triangle K)$ *(Assoziativität)*
5. $M \triangle \emptyset = M$ *($\emptyset$ ist Nullelement)*
6. $M \triangle N = \emptyset$ *gdw.* $M = N$
7. $M \cap (N \triangle K) = (M \cap N) \triangle (M \cap K)$

Übung 8
Man beweise die Aussagen der vorstehenden Sätze.

Halbgruppe, Gruppe, Abelsche Gruppe
Ist in einer Menge G eine zweistellige assoziative Operation $\square$ definiert, so nennt man $[G, \square]$ eine *Halbgruppe*. Ist die Operation zudem kommutativ, wird die Halbgruppe ebenso bezeichnet. Eine kommutative Halbgruppe $[G, \square, 0]$ mit Nullelement wird ABEL*sche* oder *kommutative Gruppe* genannt, wenn zu jedem $g \in G$ (genau) ein inverses Element $g' \in G$ mit $g \square g' = 0$ existiert.

Folgerung 1.2.21
Die Aussage 4 des Satzes 1.2.20 besagt also, daß $[\mathfrak{E}, \triangle]$ eine Halbgruppe ist, wegen Aussage 3 handelt es sich sogar um eine kommutative Halbgruppe. In dieser Halbgruppe ist $\emptyset$ das Nullelement. Wegen 6 ist also $[\mathfrak{E}, \triangle, \emptyset]$ eine ABEL*sche Gruppe.*

Ebenso wie der Verbandbegriff spielt der Gruppenbegriff eine bedeutende Rolle in der Mathematik und ihren Anwendungen.

1.2.4 Allgemeiner Durchschnitt, Allgemeine Vereinigung und Potenzmengenbildung

Definition 1.2.22 (Allgemeiner Durchschnitt und Vereinigung)
Es sei $\mathfrak{M} \subseteq \mathfrak{E}$ ein beliebiges Mengensystem. Der allgemeine Durchschnitt *von $\mathfrak{M}$ ist die Menge*

$$\bigcap \mathfrak{M} =_{Def} \{x | \text{ für alle } M \in \mathfrak{M} \text{ ist } x \in M\}.$$

Die allgemeine Vereinigung *von $\mathfrak{M}$ ist die Menge*

$$\bigcup \mathfrak{M} =_{Def} \{x | \text{ es gibt ein } M \in \mathfrak{M} \text{ mit } x \in M\}.$$

Die Operationen „allgemeiner Durchschnitt“ und „ allgemeine Vereinigung“ führen also von Mengensystemen zu Mengen, verringern also die Stufe.

Satz 1.2.23
Es sei $\mathfrak{M}, \mathfrak{N} \subseteq \mathfrak{E}$

1. $\bigcap \emptyset^{(2)} = E$

2. $\bigcup \emptyset^{(2)} = \emptyset$

3. $(\bigcup \mathfrak{M}) \cap N = \bigcup \{M \cap N | M \in \mathfrak{M}\}$

4. $(\bigcap \mathfrak{M}) \cup N = \bigcap \{M \cup N | M \in \mathfrak{M}\}$

5. $\bigcap \mathfrak{M} \cap \bigcap \mathfrak{N} = \bigcap (\mathfrak{M} \cup \mathfrak{N})$

6. $\bigcup \mathfrak{M} \cup \bigcup \mathfrak{N} = \bigcup (\mathfrak{M} \cup \mathfrak{N})$

7. $\overline{\bigcap \mathfrak{M}} = \bigcup \{\overline{M} | M \in \mathfrak{M}\}$

Übung 9
Man beweise die Aussagen des vorstehenden Satzes.

Definition 1.2.24 (Potenzmenge)
Die Potenzmenge $\mathfrak{P}(M)$ *der Menge* M *ist das System aller Teilmengen von* M*, d. h.*

$$\mathfrak{P}(M) =_{Def} \{N | N \subseteq M\}$$

Die Potenzmengenbildung führt also von einer Menge zu einem Mengensystem, sie hebt die Stufe an. Es gilt der

Satz 1.2.25
Für beliebige Mengen M, N *gilt:*

1. $\mathfrak{P}(\emptyset) = \{\emptyset\}$

2. $\mathfrak{P}(E) = \mathfrak{E}$

3. $M \subseteq N$ *gdw.* $\mathfrak{P}(M) \subseteq \mathfrak{P}(N)$

4. $\mathfrak{P}(M \cup N) \supseteq \mathfrak{P}(M) \cup \mathfrak{P}(N)$

5. $\mathfrak{P}(M \cap N) = \mathfrak{P}(M) \cap \mathfrak{P}(N)$

6. $\mathfrak{P}(\overline{M}) \subseteq \{\emptyset\} \cup \overline{\mathfrak{P}(M)}$

Übung 10
Man beweise die Aussagen des vorstehenden Satzes.

1.3 Korrespondenzen, Relationen, Abbildungen

Unser nächstes Ziel besteht darin, den Begriff der Zuordnung mathematisch, d. h. mengentheoretisch, zu präzisieren. Wenn wir davon gesprochen haben, daß eine zweistellige Operation $\Box$ je zwei Elementen a, b einer Menge G ein Element $a \Box b$ aus G eindeutig zuordnet, so haben wir dabei an die Anschauung appelliert, aber nicht gesagt, was Zuordnen eigentlich ist.

1.3.1 Geordnetes Paar und Cartesisches Produkt

Der einfachste Fall einer Zuordnung liegt vor, wenn einem einzigen Element a ein Element b zugeordnet ist, was man z. B. durch $[a, b]$ notieren kann. Die erste Komponente dieses sogenannten *geordneten Paares*, das Element a, wird in einen Zusammenhang gesetzt mit der zweiten Komponente, dem Element b; dem a wird das b zugeordnet und nicht umgekehrt — es kommt also auf die Reihenfolge der Komponenten (Elemente) an. Für geordnete Paare $[a, b]$, $[c, d]$ muß also gelten

$$[a, b] = [c, d] \text{ gdw. } a = c \text{ und } b = d$$

Diese Eigenschaft ist die grundlegende Eigenschaft geordneter Paare, sie kann zur axiomatischen Charakterisierung dieses Begriffs verwendet werden. Wir können aber diesen Begriff mengentheoretisch wie folgt definieren:

Definition 1.3.1 (Geordnetes Paar)
Es seien a, b beliebige Elemente. Dann ist das geordnete Paar $[a, b]$ das Mengensystem

$$[a, b] =_{Def} \{\{a\}, \{a, b\}\}.$$

Man kann nun beweisen

Satz 1.3.2
Für beliebige Elemente a, b, c, d gilt

$$[a, b] = [c, d] \text{ gdw. } a = c \text{ und } b = d.$$

Durch wiederholte Paarbildung kann man nun den Begriff des geordneten Tripels, Quadrupels, ... (allgemein:) den des n-Tupels einführen. Dazu überträgt man die Definition 1.3.1 von Elementen a, b auf beliebige Mengen a, b gleicher Stufe und definiert dann:

Definition 1.3.3
Es seien $a, b, c, d, a_1, a_2, \cdots$ beliebige Elemente.

- $[a, b, c] =_{Def} [[a, b], \{\{c\}\}]$ *(Tripel)*
- $[a, b, c, d] =_{Def} [[a, b, c], \{\{\{\{d\}\}\}\}]$ *(Quadrupel)*

  $\vdots$ $\vdots$
- $[a_1, \ldots, a_n] =_{Def} [[a_1, \ldots, a_{n-1}], \underbrace{\{\cdots\{}_{2(n-2)} a_n \underbrace{\}\cdots\}}_{2(n-2)}]$ *(n-Tupel)*

Man zeigt durch Induktion über $n \geq 2$:

Satz 1.3.4
$[a_1, \ldots, a_n] = [b_1, \ldots, b_n]$ *gdw.* $a_1 = b_1$ *und* ... *und* $a_n = b_n$.

Einfachste Zuordnungen der Form „a geht über in b“ können wir also durch das geordnete Paar $[a, b]$ beschreiben, kompliziertere Zuordnungen werden wir durch Mengen geordneter Paare beschreiben.

Definition 1.3.5 (Cartesisches Produkt)
Es seien $M_1, M_2, \ldots, M_n$ Mengen (gleicher Stufe). Das CARTESISCHE *Produkt (direktes Produkt, Kreuzprodukt) von $M_1, \ldots, M_n$ in dieser Reihenfolgen) ist die Menge*

$$M_1 \times M_2 =_{Def} \{[a, b] \,|\, a \in M_1 \text{ und } b \in M_2\}\,.$$

bzw.

$$M_1 \times \cdots \times M_n =_{Def} \{[a_1, \ldots, a_n] \,|\, a_1 \in M_1 \text{ und } \ldots \text{ und } a_n \in M_n\}\,.$$

Nach dieser Definition kann das CARTESISCHE Produkt nur zwischen Mengen gleicher Stufe gebildet werden. Man braucht aber auch Produkte zwischen Mengen verschiedener Stufen, z. B. wenn man eine Zuordnung beschreiben will, die jedem Element eine gewisse Menge zuordnet. Eine Definition läßt sich ebenfalls angeben, wir verzichten hier darauf und gehen im folgenden davon aus, daß für beliebige Mengen N, K das Kreuzprodukt definiert ist. Wichtige Eigenschaften des Produkts formulieren wir als

Satz 1.3.6

1. *$M \times N = \emptyset$ gdw. $M = \emptyset$ oder $N = \emptyset$*
2. *Wenn $M \subseteq N$, so $M \times K \subseteq N \times K$ und $K \times M \subseteq K \times N$.*
3. *$M \times (N \cup K) = (M \times N) \cup (M \times K)$*
4. *$M \times (N \cap K) = (M \times N) \cap (M \times K)$*

1.3.2 Korrespondenzen

Wir sind nun in der Lage, den Begriff der Zuordnung, wir sagen „Korrespondenz“ mengentheoretisch zu definieren:

Definition 1.3.7 (Korrespondenz)
Es seien M, N Mengen. Dann wird K eine Korrespondenz aus M in N (bzw. zwischen M und N) genannt, wenn $K \subseteq M \times N$ ist.

Beispiele

- Es sei $M = N = \mathbb{N}$ (die Menge der natürlichen Zahlen) und

$$\leq \;=\; \{[a, b] \,|\, a, b \in \mathbb{N} \text{ und } a \text{ kleiner oder gleich } b\} \subseteq \mathbb{N} \times \mathbb{N}$$

- Es sei $M = \mathbb{N}$, $N = \{\text{wahr}, \text{falsch}\}$ (die Menge der Wahrheitswerte) und

$$P_2 = \{[a, \alpha] \,|\, a \in \mathbb{N} \text{ und entweder } [a \text{ ist keine Primzahl und } \alpha = \text{falsch}] \text{ oder } [a \text{ ist Primzahl und } \alpha = \text{wahr}]\}$$

- Es sei $M = \{1, \ldots, 99\}$, N die Menge aller Straßennamen in Berlin und

$$F \;=\; \{[a, x] \,|\, a \in M \text{ und } x \in N \text{ und die Straßenbahnlinie Nr.} a \text{ führt durch die Straße } x\}$$

- Es sei A eine Menge, dann bezeichnet I_A die Identität über A mit

$$I_A \;=_{\text{Def}}\; \{[a, a] \mid a \in A\}$$

Ist K eine Korrespondenz, so schreiben wir für $[a, b] \in K$ auch aKb und sagen dafür „a steht zu b in Korrespondenz K“ oder „b ist ein K-Bild von a“ oder „a ist ein K-Urbild von b“. Unser erstes Beispiel zeigt, daß ein Element im allgemeinen mehrere Bilder oder Urbilder haben kann; das dritte Beispiel zeigt, daß ein Element kein Bild bzw. Urbild haben muß.

Definition 1.3.8 (Benennungen von Korrespondenzen)
Es sei $K \subseteq M \times N$ eine Korrespondenz zwischen M und N.

1. *Der* Definitionsbereich *(Argumentenbereich, Vorbereich) von K ist die Menge D_K der Elemente von M, die K-Urbilder eines Elementes von N sind, d. h.*

$$D_K =_{Def} \{a|\ \textit{Es gibt ein } b \textit{ mit } [a, b] \in K\}$$

2. *Der* Bildbereich *(Wertebereich, Nachbereich) von K ist die Menge B_K der Elemente von N, die K-Bild eines Elementes von M sind, d. h.*

$$B_K =_{Def} \{b|\ \textit{Es gibt ein } a \textit{ mit } [a, b] \in K\}$$

3. *K heißt Korrespondenz* von M in N, *wenn $D_K = M$ ist*
4. *K heißt Korrespondenz* aus M auf N, *wenn $B_K = N$ ist*
5. *K heißt Korrespondenz* von M auf N, *wenn $D_K = M$ ist und $B_K = N$ ist.*
6. *Die Korrespondenz K wird* eindeutig *genannt, wenn zu jedem $a \in M$ höchstens ein $b \in N$ (eventuell keines) mit $[a, b] \in K$ existiert.*
7. *Die Korrespondenz K wird* eindeutig umkehrbar *genannt, wenn es zu jedem $b \in N$ höchstens ein $a \in M$ mit $[a, b] \in K$ gibt*

Beispiele
Die Korrespondenzen $\leq$, P_2 und I_A sind Korrespondenzen *von–auf*, bei F ist das nicht der Fall; ferner ist P_2 eindeutig, aber nicht eindeutig umkehrbar; I_A ist dagegen eindeutig umkehrbar.

Definition 1.3.9 (Verkettung und Inversion)
Es seien K und L Korrespondenzen, $K \subseteq A \times B$, $L \subseteq C \times D$.

1. *Die* Verkettung *$K \circ L$ von K mit L ist die Menge*

$$K \circ L =_{Def} \{[a, d] \mid \textit{Es gibt ein } b \textit{ mit } [a, b] \in K \textit{ und } [b, d] \in L\}$$

2. *Die* Inversion *K^{-1} von K ist die Menge*

$$K^{-1} =_{Def} \{[b, a] \mid [a, b] \in K\}$$

Man zeigt nun:

Satz 1.3.10
Es seien $K \subseteq A \times B$, $L \subseteq C \times D$ Korrespondenzen.

1. *$K \circ L$ ist eine Korrespondenz zwischen A und D*
2. *Die Verkettung ist eine assoziative Operation*

3. *Die Identität I_E über dem Objektbereich E ist neutrales Element, d. h.* $K \circ I_E = I_E \circ K = K$
4. $K \circ L = \emptyset$ *genau dann, wenn* $B_K \cap D_L = \emptyset$.
5. K^{-1} *ist eine Korrespondenz zwischen B und A*
6. $\left(K^{-1}\right)^{-1} = K$
7. $(K \circ L)^{-1} = L^{-1} \circ K^{-1}$.

Übung 11
Man beweise die Aussagen des vorstehenden Satzes.

1.3.3 Relationen

Korrespondenzen aus einer Menge A in diese Menge selbst, werden als Relationen bezeichnet:

Definition 1.3.11 (Relation)
R wird binäre (zweistellige) Relation *in A genannt, wenn $R \subseteq A \times A$ ist. Ist dabei $D_R \cup B_R = A$, so heißt R Relation* über A.

Beispiele

- $\leq$ ist eine Relation über $\mathbb{N}$
- I_E ist eine Relation über E
- die leere Menge $\emptyset^{(3)}$ dritter Stufe ist eine Relation in jeder Menge erster Stufe
- $A \times A$ ist eine Relation über A (die Allrelation über A).

Bemerkung
Ist R eine binäre Relation, d. h. $R \subseteq A \times A$, so schreiben wir statt $[a, b] \in R$ häufig aRb.

Für Relationen werden wir einige Begriffe, die wir weiter oben schon benutzt haben, allgemein zusammenfassen mit folgender

Definition 1.3.12
Sei R eine Relation in der Menge A. R heißt

1. reflexiv *wenn für alle $a \in A$ gilt: aRa*
2. irreflexiv *wenn für kein $a \in A$ gilt: aRa*
3. transitiv *wenn für alle $a, b, c \in A$ gilt: wenn aRb und bRc, so aRc*
4. antisymmetrisch *wenn für alle $a, b \in A$ gilt: wenn aRb und bRa, so $a = b$*
5. asymmetrisch *wenn für alle $a, b \in A$ gilt: wenn aRb, so nicht bRa*
6. symmetrisch *wenn für alle $a, b \in A$ gilt: wenn aRb, so bRa*

Eine Relation R heißt

1. reflexive Halbordnung
   wenn R reflexiv, transitiv und antisymmetrisch ist
2. irreflexive Halbordnung
   wenn R irreflexiv, transitiv und asymmetrisch ist
3. Äquivalenzrelation
   wenn R reflexiv, transitiv und symmetrisch ist

Der Vollständigkeit halber haben wir hier schon den Begriff der Äquivalenzrelation genannt, der weiter unten ausführlicher behandelt wird.

Wie wir später sehen werden, spielen reflexive und transitive Relationen in der Mathematik eine große Rolle — Äquivalenzrelationen und reflexive Ordnungsrelationen haben diese Eigenschaften. Deshalb interessiert eine Operation, die (ausgehend von einer Relation ohne diese Eigenschaften) die bezüglich $\subseteq$ kleinste reflexive und transitive Relation R' mit $R \subseteq R'$ als Resultat liefert.

Definition 1.3.13 (reflexiv-transitive Hülle)
Es sei R eine Relation über *A. Wir setzen*

- $R^0 = I_A$
- $R^{i+1} = R^i \circ R$
- $\langle R \rangle =_{Def} \bigcup \{R^0, R^1, \ldots\} = \bigcup_{i \in \mathbb{N}} R^i$

Die Menge $\langle R \rangle$ wird als reflexiv-transitive Hülle *von R bezeichnet.*

Man überlegt sich sofort

Satz 1.3.14
Es sei R Relation über A

1. *$\langle R \rangle$ ist eine Relation über A*
2. *$\langle R \rangle$ ist reflexiv über A, d. h. für alle $a \in A$ gilt: $a \langle R \rangle a$*
3. *$\langle R \rangle$ ist transitiv über A, d. h. für alle $a, b, c \in A$ gilt:*
   wenn $a \langle R \rangle b$, $b \langle R \rangle c$, so $a \langle R \rangle c$

Beweis
der einzelnen Aussagen:

1. es ist zu zeigen: $\langle R \rangle \subseteq A \times A$ und $D_{\langle R \rangle} = A$, woraus $D_{\langle R \rangle} \cup B_{\langle R \rangle} = A$ folgt.
2. gilt offensichtlich, weil $I_A \subseteq \langle R \rangle$ ist.
3. Wegen $a \langle R \rangle b$, $b \langle R \rangle c$ gibt es Zahlen $i, j \geq 0$ mit aR^ib, bR^jc. Bei $i = 0$ folgt $a = b$, also aR^jc, d. h. $a \langle R \rangle c$. Analog schließt man bei $j = 0$.

   Sei $i > 0$. Man überlegt sich (durch Induktion über i) daß in diesem Fall Elemente $a_1, \ldots, a_{i+1} \in A$ existieren mit $a_1 = a$, $a_{i+1} = b$ und $a_l R a_{l+1}$ für $l = 1, \ldots, i$.

   Analog existieren Elemente $b_1, \ldots, b_{j+1}$ mit $b_1 = b$, $b_{j+1} = c$ und $b_k R b_{k+1}$ für $k = 1, \ldots, j$.

   Daraus folgt $[a, c] \in R^{i+j} \subseteq \langle R \rangle$, d. h. $a \langle R \rangle c$.

□

Der Gebrauch der Bezeichnung „Hülle“ ist in der Mathematik für solche Operationen reserviert, die die sogenannten Hülleneigenschaften haben:

Hüllenoperation
Eine Operation $Op : \mathfrak{P}(A) \to \mathfrak{P}(A)$ heißt *Hüllenoperation*, wenn folgende Eigenschaften gelten:

1. $M \subseteq Op(M)$ (Satz der Einbettung)
2. Wenn $M \subseteq N$, so $Op(M) \subseteq Op(N)$ (Satz der Monotonie)
3. $Op(Op(M)) = Op(M)$ (Satz der Abgeschlossenheit)

Op erfüllt den *Endlichkeitssatz*, wenn für alle $a \in A, M \subseteq A$ gilt:
Wenn $a \in Op(M)$, so existiert eine endliche Teilmenge $M^* \subseteq M$ mit $a \in Op(M^*)$.

Satz 1.3.15
Die Operation $\langle\ \rangle$ hat die Hülleneigenschaften, d. h. es gilt

1. $R \subseteq \langle R \rangle$
2. *Wenn* $R_1 \subseteq R_2$, *so* $\langle R_1 \rangle \subseteq \langle R_2 \rangle$
3. $\langle \langle R \rangle \rangle = \langle R \rangle$

Überdies erfüllt die Operation $\langle\ \rangle$ den Endlichkeitssatz:

$$[a, b] \in \langle R \rangle, \text{ so existiert eine endliche Teilmenge } R' \subseteq R \text{ mit } [a, b] \in \langle R' \rangle$$

Übung 12
Man beweise den vorstehenden Satz.

Ferner gilt der bereits oben erwähnte

Satz 1.3.16
Es sei R eine Relation über A. Dann ist $\langle R \rangle$ die bezüglich $\subseteq$ kleinste reflexive und transitive Relation R' mit $R \subseteq R'$, d. h. für alle Relationen R' über A gilt: Wenn $R \subseteq R'$ und R' ist reflexiv und transitiv, so ist $\langle R \rangle \subseteq R'$.

Beweis
Daß $\langle R \rangle$ reflexiv und transitiv ist, sagt Satz 1.3.14. Aus dem Satz der Einbettung haben wir $R \subseteq \langle R \rangle$. Ist $R \subseteq R'$, so ist $\langle R \rangle \subseteq \langle R' \rangle$ wegen der Monotonie. Für reflexive und transitive Relationen R' gilt aber $R' = \langle R' \rangle$, denn aus der Reflexivität von R' über A folgt $R^0 = I_A \subseteq R'$ und aus der Transitivität von R' folgt $R' \circ R' = R'$, d. h. $(R')^i \subseteq R'$. Also ist $\langle R \rangle \subseteq \langle R' \rangle = R'$. □

In analoger Weise wie zweistellige Korrespondenzen definiert man n-stellige Korrespondenzen zwischen Mengen $M_1, \ldots, M_n$ und n-stellige Relationen R in A als

$$R \subseteq A \times \cdots \times A = \bigtimes_n A.$$

1.3.4 Abbildungen

Wir wenden uns nun dem Begriff der *Abbildung* zu.

Definition 1.3.17 (Abbildung und deren Benennung)
Es seien A, B Mengen

1. *f heißt* Abbildung *von A in B, wenn*
   (a) *f eine Korrespondenz von A in B und*
   (b) *f eindeutig ist.*
2. *Eine Abbildung f wird* surjektiv *genannt (oder Abbildung* auf B*), wenn $B_f = B$ ist.*
3. *Eine Abbildung f heißt* injektiv *(oder umkehrbar eindeutig), wenn f eindeutig umkehrbar ist.*
4. *Eine Abbildung f heißt* bijektiv*, wenn f surjektiv und injektiv ist.*

Beispiel
Die oben angegebene Korrespondenz P_2 ist eine surjektive Abbildung von $\mathbb{N}$ auf {wahr, falsch}.

Bemerkung
Die Aussage „f ist eine Abbildung von A in B“ notieren wir künftig durch „$f : A \to B$“ und statt „$[a, b] \in f$“ schreiben wir „$b = f(a)$“.

Zum Abschluß dieses Abschnittes machen wir eine Bemerkung über die *Beschreibung von Korrespondenzen durch Abbildungen.* Dabei verwenden wir, daß das Kreuzprodukt auch zwischen Mengen unterschiedlicher Stufen definiert ist.

Es sei $K \subseteq A \times B$ eine Korrespondenz zwischen A und B. Die zu K gehörige Abbildung f_K ist definiert durch

$$f_K : A \to \mathfrak{P}(B)$$

mit

$$f_K(a) =_{\text{Def}} \{b | [a, b] \in K\}$$

für alle $a \in A$.

Offenbar kann K aus einer Kenntnis von f_K gewonnen werden (ist durch f_K eindeutig bestimmt), denn es gilt:

$$K = \bigcup_{a \in A} \{a\} \times f_K(a) = \bigcup \{\{a\} \times f_K(a) | a \in A\}$$

Da die Beziehung zwischen K undf_K also eineindeutig ist, kann man wirklich von einer *Beschreibung* sprechen; Beschreibung (f_K) und Beschriebenes (K) entsprechen einander eineindeutig.

1.4 Äquivalenzrelationen

1.4.1 Äquivalenzrelationen und Zerlegungen

Definition 1.4.1 (Äquivalenzrelation)
Eine Relation $R \subseteq A \times A$ in A heißt Äquivalenzrelation *über A, wenn*

1. *R eine reflexive Relation über A ist*
2. *R symmetrisch ist, d. h. $[a, b] \in R$ stets $[b, a] \in R$ impliziert (mit anderen Worten ist $R = R^{-1}$)*
3. *R transitiv ist.*

Bemerkung
Von der Forderung (1.4.1.1) ist nur wesentlich, daß R eine Relation *über* A ist, denn aus der Symmetrie folgt $D_R = B_R$, d. h. wir haben $A = D_R \cup B_R = D_R = B_R$. Ist nun $a \in A$, so gibt es ein $b \in A$ mit $[a, b] \in R$ (wegen $a \in D_R$) und $[b, a] \in R$ (wegen (1.2)) woraus mit der Transitivität $[a, a] \in R$ folgt, d. h. R ist reflexiv.

Für jede Menge A ist offenbar I_A eine Äquivalenzrelation über A, und zwar ist I_A in dem Sinne die schärfste (bzw. feinste) Äquivalenzrelation über A, daß I_A die meisten Unterschiede zwischen Elementen von A macht. Im allgemeinen nennt man ja Elemente äquivalent bzw. gleichwertig in gewisser Hinsicht, wenn sie sich in dieser Hinsicht nicht unterscheiden, z. B. gleich verhalten. Wenn es z. B. auf Zeit, Kosten, Bequemlichkeit usw. bei einer beabsichtigten Reise nicht ankommt, sondern nur auf den Ortswechsel, so ist es gleichwertig, äquivalent, ob dazu das Flugzeug oder die Bahn benutzt wird. Objekte in Äquivalenz zu setzen, bedeutet also, von gewissen ihrer Eigenschaften abzusehen (zu abstrahieren) und andere gemeinsame Eigenschaften hervorzuheben. Aus dieser Rolle bei der Abstraktion, der Bildung neuer Begriffe, leitet sich die Bedeutung des Begriffs der Äquivalenzrelation ab.

Wir betrachten nun die oben definierte Abbildung f_R für den Fall, daß R eine Äquivalenzrelation über A ist. Man zeigt leicht:

Satz 1.4.2
Es sei R eine Äquivalenzrelation über A und $a, b \in A$. Dann gilt:

1. $a \in f_R(a) \neq \emptyset$
2. $a \in f_R(b)$ *genau dann, wenn* $b \in f_R(a)$
3. *Wenn* $f_R(a) \cap f_R(b) \neq \emptyset$, *so* $f_R(a) = f_R(b)$.

Definition 1.4.3 (Zerlegung)
Es sei A eine Menge, $A \neq \emptyset$, $\mathfrak{M} \subseteq \mathfrak{P}(A)$. $\mathfrak{M}$ heißt Zerlegung *von A (Klasseneinteilung oder Partition von A), wenn gilt*

1. $\emptyset \notin \mathfrak{M}$
2. $A = \bigcup \mathfrak{M}$
3. *$\mathfrak{M}$ ist disjunkt, d. h. für alle $M, M' \in \mathfrak{M}$ gilt: wenn $M \cap M' \neq \emptyset$, so $M = M'$.*

Mit Hilfe des Begriffes der Zerlegung können wir den Satz 1.4.2 wie folgt formulieren:

Satz 1.4.4
Ist R eine Äquivalenzrelation über $A \neq \emptyset$, so ist

$$\mathfrak{M}_R =_{Def} \{f_R(a) \,|a \in A\} = B_{f_R}$$

eine Zerlegung von A.

Wir wissen bereits, daß wir eine Relation R durch die Abbildung f_R beschreiben können. Es stellt sich die Frage, ob bei Äquivalenzrelationen R dazu bereits $\mathfrak{M}_R = B_{f_R}$ genügt. Es zeigt sich zunächst:

Satz 1.4.5
Ist R eine Äquivalenzrelation über A, so gilt für $a, b \in A$: aRb genau dann, wenn es ein $M \in \mathfrak{M}_R$ mit $a, b \in M$ gibt.

Diese Aussage weist den Weg, wie zu einer Zerlegung $\mathfrak{M}$ eine Äquivalenzrelation $R_{\mathfrak{M}}$ zu definieren ist; wir setzen für $a, b \in A = \bigcup \mathfrak{M}$

$$aR_{\mathfrak{M}}b \text{ gdw. es ein } M \in \mathfrak{M} \text{ mit } a, b \in M \text{ gibt.}$$

Man zeigt ohne Schwierigkeiten:

Satz 1.4.6
Ist $\mathfrak{M}$ eine Zerlegung von A, so ist $R_{\mathfrak{M}}$ Äquivalenzrelation über A.

Die Bildung von $R_{\mathfrak{M}}$ zu einer Zerlegung $\mathfrak{M}$ erweist sich nun gerade als Umkehrung der Bildung der Zerlegung $\mathfrak{M}_R$ zu einer Äquivalenzrelation R.

Satz 1.4.7
Es sei $A \neq \emptyset$

1. *Ist R eine Äquivalenzrelation über A, so ist*

$$R = R_{(\mathfrak{M}_R)}$$

2. *Ist $\mathfrak{M}$ eine Zerlegung von A, so ist*

$$\mathfrak{M} = \mathfrak{M}_{(R_{\mathfrak{M}})}$$

Äquivalenzrelationen und Zerlegungen sind einander auf diese Weise eineindeutig zugeordnet. Man bezeichnet $f_R(a)$ als

die *Äquivalenzklasse* nach R, in der a liegt (die Äquivalenzklasse von a)

und notiert $\mathfrak{M}_R$ auch als A/R (A *zerlegt nach* R, A *faktorisiert nach* R).

1.4.2 Feinheit von Äquivalenzrelationen

Definition 1.4.8 (Feinerrelation)
Es seien $\mathfrak{M}_1$, $\mathfrak{M}_2$ Zerlegungen. $\mathfrak{M}_1$ ist feiner *als $\mathfrak{M}_2$ ($\mathfrak{M}_1 \preceq \mathfrak{M}_2$), wenn zu jeder Klasse $K_1 \in \mathfrak{M}_1$ eine Klasse $K_2 \in \mathfrak{M}_2$ existiert mit $K_1 \subseteq K_2$.*

Satz 1.4.9
Seien R_1, R_2 Äquivalenzrelationen. Dann gilt:

$$\mathfrak{M}_{R_1} \preceq \mathfrak{M}_{R_2} \text{ gdw. } R_1 \subseteq R_2$$

Folgerung 1.4.10
Wenn $\mathfrak{Z}_A$ die Menge aller Zerlegungen der Menge A bezeichnet, dann ist $[\mathfrak{Z}_A, \preceq]$ eine reflexive Halbordnung.

Definition 1.4.11 (Durchschnitt von Äquivalenzrelationen)
Seien R_1 und R_2 Äquivalenzrelationen. Dann bezeichnet $R_1 \sqcap R_2$ die gröbste Äquivalenzrelation R mit $R \subseteq R_1$ und $R \subseteq R_2$. Für die Zerlegungen $\mathfrak{M}_1$ und $\mathfrak{M}_2$ bezeichnet analog $\mathfrak{M}_1 \sqcap \mathfrak{M}_2$ die gröbste Zerlegung $\mathfrak{M}$ mit $\mathfrak{M} \preceq \mathfrak{M}_1$ und $\mathfrak{M} \preceq \mathfrak{M}_2$.

Satz 1.4.12
Es gilt: $R_1 \sqcap R_2 = R_1 \cap R_2$

Beweis
Zu zeigen: $R_1 \cap R_2$ ist Äquivalenzrelation.

1. reflexiv: $I_A \subseteq R_1 \cap R_2$
2. symmetrisch: klar
3. transitiv: $\underbrace{a(R_1 \cap R_2)b}_{\substack{aR_1b \\ aR_2b}}$ und $\underbrace{b(R_1 \cap R_2)c}_{\substack{bR_1c \\ bR_2c}} \Rightarrow \underbrace{a(R_1 \cap R_2)c}_{\substack{aR_1c \\ aR_2c}}$

□

Für die Zerlegungen gilt $\mathfrak{M}_1 \sqcap \mathfrak{M}_2 = \{N_1 \cap N_2 \mid N_1 \in \mathfrak{M}_1, N_2 \in \mathfrak{M}_2, N_1 \cap N_2 \neq \emptyset\}$. $\mathfrak{M}_{I_A}$ ist die feinste und $\mathfrak{M}_{A \times A}$ die gröbste Zerlegung von A.

Wir fragen uns nun, ob $R_1 \cup R_2$ auch eine Äquivalenzrelation ist. An einem Beispiel kann man sich sofort klar machen, daß das i. a. nicht gilt. Wir definieren deshalb:

Definition 1.4.13
$R_1 \sqcup R_2 =_{Def} < R_1 \cup R_2 >$ ist die bezüglich Inklusion kleinste (feinste) Äquivalenzrelation die R_1 und R_2 umfaßt.

Folgerung 1.4.14
Wenn $\mathfrak{R}$ die Menge aller Äquivalenzrelationen über A bezeichnet, dann ist $[\mathfrak{R}, \sqcap, \sqcup]$ ein Verband mit Nullelement I_A und Einselement $A \times A$.

1.4.3 Kongruenzrelationen

Wenn in der Menge A neben der Äquivalenzrelation R noch weitere Relationen und Operationen, d. h. Abbildungen definiert sind, die mit R „verträglich“ sind, nennt man R eine Kongruenzrelation bzgl. diesen Operationen und Relationen.

Dieser Fall ist deshalb wichtig, weil sich dann Relationen bzw. Operationen auf A auf dem einfacheren Bereich A/R nachbilden lassen.

Wir führen das genauer aus für den Fall einer n-stelligen Operation über A.

Definition 1.4.15
Es sei A eine nichtleere Menge , $n \geq 1$.

1. *$\circ$ heißt n-stellige* Operation *über A, wenn*

$$\circ : \bigtimes_n A \to A$$

2. *Eine Äquivalenzrelation R über A heißt* verträglich (invariant) *bzgl. der n-stelligen Operation $\circ$ über A (oder Kongruenzrelation bzgl. $\circ$), wenn für alle $a_1, \ldots, a_n$, $b_1, \ldots, b_n \in A$ gilt:*

$$\textit{Wenn } a_1 R b_1 \textit{ und } \ldots \textit{ und } a_n R b_n, \textit{ so } \circ(a_1, \ldots, a_n)\, R \circ (b_1, \ldots, b_n)$$

Ist nun R eine Kongruenzrelation über A bzgl. der Operation $\circ$, so können wir für $M_1, \ldots, M_n \in A/R$ definieren

$$\circ'(M_1, \cdots M_n) =_{\text{Def}} f_R(\circ(a_1, \ldots, a_n))$$

für beliebiges $a_1, \ldots a_n$ mit $a_1 \in M_1$ und $\ldots$ und $a_n \in M_n$.

Diese Definition muß gerechtfertigt werden, d. h. es ist zu zeigen:

$$\text{Wenn } [a_1, \ldots, a_n] \in M_1 \times \cdots \times M_n \text{ und } [b_1, \ldots b_n] \in M_1 \times \cdots \times M_n, \text{ so}$$

$$f_R(\circ(a_1, \ldots, a_n)) = f_R(\circ(b_1, \ldots, b_n)).$$

Das ergibt sich daraus, daß R Kongruenz bzgl. $\circ$ ist, denn aus $a_i, b_i \in M_i \in A/R$ folgt $a_i R b_i$, also $\circ(a_1, \ldots a_n)\, R \circ (b_1, \ldots, b_n)$, folglich $f_R(\circ(a_1, \ldots, a_n)) = f_R(\circ(b_1, \ldots, b_n))$. Es ergibt sich also, daß folgendes Diagramm „kommutativ“ ist:

$$\begin{array}{ccccc} A & \times \cdots \times & A & \xrightarrow{\circ} & A \\ {\scriptstyle f_R}\downarrow & & {\scriptstyle f_R}\downarrow & & {\scriptstyle f_R}\downarrow \\ (A/R) & \times \cdots \times & (A/R) & \xrightarrow{\circ'} & A/R \end{array}$$

Das Diagramm ist insofern „kommutativ“, daß man ausgehend von einem Element $[a_1, \ldots, a_n] \in \bigtimes_n A$ das gleiche Element von A/R erhält, unabhängig davon, ob man zuerst mittels $\circ$ das Element $\circ(a_1, \ldots, a_n)$ bildet und dann f_R anwendet oder zuerst mittels f_R das Element $[f_R(a_1), \ldots, f_R(a_n)] \in \bigtimes_n (A/R)$ bildet und dann $\circ'$ anwendet. Die „abstrahierende“ Abbildung f_R einer Kongruenz R liefert also zugleich eine

Vereinfachung A/R des Bereichs A (z. B. kann A/R endlich sein während A unendlich ist) und eine Vereinfachung der Operation $\circ$.

Bemerkung
Sei M eine Menge und R eine Relation über M, die reflexiv und transitiv ist. Dann ist R im allgemeinen keine Halbordnung in M, weil R nicht antisymmetrisch sein muß.

Beispiel
Definiert man „*besser*“ auf der Menge M der Studenten durch

$$S_1 \; besser \; S_2 \text{ gdw. } Notendurchschnitt(S_1) \leq Notendurchschnitt(S_2)$$

so folgt aus S_1 *besser* S_2 und S_2 *besser* S_1 nicht $S_1 = S_2$.

Wir können aber auf M die Relation $\sim$ definieren durch

$$a \sim b \text{ gdw. } aRb \text{ und } bRa$$

Offenbar ist $\sim$ eine Äquivalenzrelation und es gilt

$$\text{Wenn } aRb,\ a \sim c \text{ und } b \sim d,\ \text{ so } cRd$$

d. h. $\sim$ und R sind verträglich.

Folglich können wir die Relation R auf die Äquivalenzklassen übertragen:

$$[a], [b] \in M/\sim : [a]\overline{R}\,[b] \text{ gdw. } aRb$$

Man zeigt, daß $[M/\sim, \overline{R}\,]$ eine Halbordnung ist.

1.5 Natürliche Zahlen

In diesem Abschnitt beschäftigen wir uns mit der mengentheoretischen Begründung der natürlichen Zahlen. Wir haben diesen Begriff bisher im wesentlichen nur in Beispielen benutzen können (nicht z. B. in Definitionen), denn wir haben ihn bisher nicht definiert. Insbesondere ist noch nicht klar, ob die Menge der natürlichen Zahlen überhaupt existiert, oder ob sie vielleicht sogar zu einer Antinomie führt.

Wir geben zunächst (als weiteres Beispiel einer axiomatische Charakterisierung) eine axiomatische Charakterisierung der natürlichen Zahlen an:

Peanosches Axiomensystem der natürlichen Zahlen
Die Grundbegriffe „Null“, „Menge aller natürlichen Zahlen“ und „die natürliche Zahl b ist unmittelbarer Nachfolger der natürlichen Zahl a“, formal durch x_0, X, $a\mathrm{Nf}b$ notiert, werden wie folgt zueinander in Beziehung gesetzt:

1. $x_0 \in X$ (lies: Null ist eine natürliche Zahl)
2. Zu jedem $a \in X$ gibt es genau ein $b \in X$ mit $a\mathrm{Nf}b$
3. Es gibt kein $a \in X$ mit $a\mathrm{Nf}x_0$
4. Für alle $a, b, c \in X$ gilt: Wenn $a\mathrm{Nf}c$ und $b\mathrm{Nf}c$, so $a = b$
5. Für alle $M \subseteq X$ gilt:
   Wenn $x_0 \in M$ und (für alle $a \in M$ und jedes b mit $a\mathrm{Nf}b$ gilt $b \in M$), so ist $X \subseteq M$.

Ein Modell dieses Axiomensystems P ist ein Tripel $[\alpha, A, R]$, bestehend aus einem konkreten mathematischen Objekt α, einer Menge A von Objekten und einer Relation R über A, derart daß diese fünf Axiome erfüllt sind.

Das Tripel $[0, \mathbb{N}, R^\star]$ mit $aR^\star b$ gdw. $b = a+1$ erfüllt offenbar das PEANOsche Axiomensystem; man kann sogar zeigen, daß jedes Modell von P zu diesem Tripel isomorph ist.

Wenn wir gesagt haben, daß $[0, \mathbb{N}, R^\star]$ P erfüllt, so setzt daß natürlich voraus, daß sich die in diesem Tripel vorkommenden Objekte 0, $\mathbb{N}$, $R^\star$ mathematisch (mengentheoretisch) als existent begründen lassen.

1.5.1 Gleichmächtigkeit

Den Zugang zur Einführung des Zahlenbegriffs bildet der Begriff der Gleichzahligkeit, der Gleichmächtigkeit. Das Gemeinsame einer Menge von drei Äpfeln, einer Menge von drei Stühlen oder drei Flaschen ist die Gleichmächtigkeit, so daß man sagen kann die *Drei*, das ist das Mengensystem, in dem genau alle Mengen mit genau drei Elementen vorkommen. Nun kann man aber von einer Menge schwer feststellen, ob sie genau drei Elemente hat, ohne die drei Elemente zu zählen, es sei denn, es ist einem schon eine Menge bekannt, die genau drei Elemente enthält. Dann ist es auch anschaulich klar, daß zwei Mengen dann und nur dann gleich viele Elemente besitzen, wenn es eine Bijektion (eine eineindeutige Abbildung *von-auf*) zwischen diesen Mengen gibt.

Definition 1.5.1 (Gleichmächtigkeit)
Mengen A, B heißen gleichmächtig *(oder äquivalent, notiert durch $A \sim B$), wenn es eine Bijektion $\Phi : A \to B$ gibt.*

Man überlegt sich leicht:

Satz 1.5.2
$\sim$ ist eine Äquivalenzrelation über $\mathfrak{E}$

Definition 1.5.3 (Kardinalzahlen)
Die Äquivalenzklassen von $\mathfrak{E}$ nach $\sim$, d. h. die Elemente von $\mathfrak{E}/\sim$ heißen Kardinalzahlen *(von Mengen erster Stufe). Es ist für $M \in \mathfrak{E}$*

$$\operatorname{card}(M) = \{N | N \in \mathfrak{E} \text{ und } N \sim M\}$$

Eine Kardinalzahl ist also ein nichtleeres Mengensystem, das alle Mengen enthält, die zu einer Menge dieses Systems gleichmächtig sind. Unserer Anschauung entspricht es, die natürlichen Zahlen als Kardinalzahlen der endlichen Mengen einzuführen.

Wir müssen also dazu den Begriff der endlichen Menge klären, ohne dabei den Zahlenbegriff zu benutzen. Die Erklärung „M ist genau dann endlich, wenn es Elemente $a_1, a_2, \ldots, a_n$ so gibt, daß $M = \{a_1, \ldots, a_n\}$ ist“, ist also als Endlichkeitsdefinition nicht brauchbar.

1.5.2 Der Endlichkeitsbegriff

Eine auf RUSSEL zurückzuführende Definition des Endlichkeitsbegriffes lautet:

Definition 1.5.4 (Russel)

1. *Das System $\mathfrak{M}_{\text{fin}}$ der endlichen Mengen ist das kleinste Mengensystem $\mathfrak{M}$ welches induktiv wie folgt definiert wird*
   (a) $\emptyset \in \mathfrak{M}$ *(Anfangsschritt)*
   (b) *Ist $M \in \mathfrak{M}$ und $a \in E$, so $(M \cup \{a\}) \in \mathfrak{M}$* *(Induktionschritt)*
2. *M heißt endlich, wenn $M \in \mathfrak{M}_{\text{fin}}$ ist (sonst unendlich)*

Folgerung 1.5.5
Das Unendlichkeitsaxiom *sagt also: $E \notin \mathfrak{M}_{\text{fin}}$.*

Eine weitere Endlichkeits- bzw. Unendlichkeitsdefinition wurde von DEDEKIND angegeben:

Definition 1.5.6 (Dedekind)

1. *Eine Menge M heißt „unendlich", wenn es eine echte Teilmenge $M' \subset M$ mit $M' \sim M$ gibt.*
2. *Eine Menge M heißt „endlich", wenn M keine echte Teilmenge $M' \subset M$ mit $M' \sim M$ besitzt.*

Diese Definition beruht auf der Beobachtung, daß nur unendliche Mengen zu einer ihrer echten Teilmengen äquivalent sind. So ist z. B. die Menge $\mathbb{N}$ äquivalent mit der Menge der geraden natürlichen Zahlen, eine Bijektion $\Phi : \mathbb{N} \to \{0, 2, 4, \ldots\}$ ist gegeben durch die Gleichung $\Phi(x) = 2x$. Wir beweisen, daß beide Definitionen gleichwertig sind. Dabei bezeichnet *endlich* die RUSSELsche und *„endlich"* die DEDEKINDsche Auffassung.

Satz 1.5.7 (Gleichwertigkeit der Endlichkeitsbegriffe)
Eine Menge M ist endlich nach RUSSEL *genau dann, wenn sie „endlich" nach* DEDEKIND *ist.*

Beweis $\Longrightarrow$
Jede endliche Menge ist „endlich", d. h. wenn $M \in \mathfrak{M}_{\text{fin}}$, so gilt für jede echte Teilmenge $M' \subset M$: $M' \not\sim M$.

Wir beweisen diese Behauptung durch Induktion über M

Anfangsschritt: $M = \emptyset$
Die Behauptung ist trivial, weil die leere Menge keine echten Teilmengen besitzt.

Induktionsschritt:

Induktionsvoraussetzung:
Für jede echte Teilmenge M' von M ist $M' \not\sim M$

Induktionsbehauptung:
Für jede echte Teilmenge N' von $M \cup \{a\}$ ist $N' \not\sim M \cup \{a\}$

Wenn $a \in M$ ist, so gilt $M \cup \{a\} = M$, die Behauptung ist mit der Voraussetzung identisch. Es sei also $a \notin M$.

Wir schließen indirekt, und nehmen an, daß

$$N' \subset M \cup \{a\} \ , \ N' \sim M \cup \{a\} \, .$$

Es sei $\Phi : N' \to M \cup \{a\}$ bijektiv. Wir setzen

$$N'' =_{\text{Def}} N' \setminus \left\{\Phi^{-1}(a)\right\} \ , \ \Phi' =_{\text{Def}} \Phi \setminus \left\{\left[\Phi^{-1}(a), a\right]\right\}$$

und erhalten

$$\Phi' : N'' \to M \text{ ist bijektiv}$$

also $N'' \sim M$.

An dieser Stelle müssen wir eine Fallunterscheidung vornehmen:

Fall 1: $a \notin N''$
Wegen

$$N'' = N' \setminus \left\{\Phi^{-1}(a)\right\} \subset N' \subset M \cup \{a\},$$

gilt die Inklusion $N'' \subseteq M$, aus $N'' \sim M$ folgt nach Induktionsvoraussetzung $N'' = M$, folglich $\Phi^{-1}(a) = a$, d. h. $N' = M \cup \{a\}$ im Widerspruch zu $N' \subset M \cup \{a\}$.

Fall 2: $a \in N''$
Es sei $b =_{\text{Def}} \Phi'(a) \in M$ und $c =_{\text{Def}} \Phi^{-1}(a)$, sowie

$$\Psi(x) =_{\text{Def}} \begin{cases} a & \text{für} \quad x = a \\ b & \text{für} \quad x = c \\ \Phi(x) & \text{für} \quad x \in N' \setminus \{a, c\} \end{cases}$$

ferner $N''' = (N'' \setminus \{a\} \cup \{c\})$.
Dabei ist $N''' \sim N'' \sim M$ und $a \notin N''' \subset N' \subset M \cup \{a\}$, also $N''' \subseteq M$, woraus nach Induktionsvoraussetzung $N''' = M$ folgt. Wegen $N' = N''' \cup \{a\} = M \cup \{a\}$ erhalten wir abermals einem Widerspruch zu $N' \subset M \cup \{a\}$.

□

Damit ist Richtung $\Longrightarrow$ bewiesen. Noch zu zeigen:

Beweis $\Longleftarrow$
Jede „endliche“ Menge M ist endlich, d. h. $M \in \mathfrak{M}_{\text{fin}}$.

Wir führen den Beweis indirekt und nehmen an, daß M_0 eine Menge ist, mit

1. Wenn $M' \subset M_0$, so $M' \not\sim M_0$
2. $M_0 \notin \mathfrak{M}_{\text{fin}}$.

Dann gibt es zu jedem $N \in \mathfrak{M}_{\text{fin}}$ mit $N \subseteq M_0$ ein $a \in M_0$ mit $a \notin N$, d. h. $M_0 \setminus N \neq \emptyset$ für alle $N \in \mathfrak{M}_{\text{fin}}$ mit $N \subseteq M_0$. Wir setzen

$$\mathfrak{M}^\star =_{\text{Def}} \{M_0 \setminus N | N \subseteq M_0 \text{und } N \in \mathfrak{M}_{\text{fin}}\}$$

Es ist $\emptyset \notin \mathfrak{M}^\star$ weil $M_0 \notin \mathfrak{M}_{\text{fin}}$ und $\mathfrak{M}^\star \neq \emptyset$ weil $M_0 = M_0 \setminus \emptyset \in \mathfrak{M}^\star$.

Eine gleichwertige Formulierung des *Auswahlaxioms* lautet:

> Zu jedem nichtleeren Mengensystem $\mathfrak{M}$ nichtleerer Mengen existiert eine Auswahlfunktion, d. h. eine Abbildung $\rho : \mathfrak{M} \to \bigcup \mathfrak{M}$ mit $\rho(M) \in M$ für alle $M \in \mathfrak{M}$.

Es sei ρ eine Auswahlfunktion für $\mathfrak{M}^\star$, wir setzen

$$\begin{array}{rcl} a_0 & = & \rho\,(M_0) \\ a_{i+1} & = & \rho\,(M_0 \setminus \{a_0, a_1, \ldots, a_i\}) \quad i = 0, 1, 2, \ldots \end{array}$$

Offenbar ist $\{a_0, \ldots, a_i\} \in \mathfrak{M}_{\text{fin}}$, also $M_0 \setminus \{a_0, \ldots a_i\} \in \mathfrak{M}^\star$. Dabei sind für jedes i die Elemente $a_0, \ldots a_i$ offenbar paarweise verschieden. Es sei schließlich

$$\begin{array}{rcl} M^\star & = & \{a_0, a_1, \ldots\} \\ M^{\star\star} & = & \{a_1, a_2, \ldots\} \end{array}$$

Offenbar ist $M^\star \sim M^{\star\star}$. Ferner ist $M_0 = M^\star \cup (M_0 \setminus M^\star)$ und für $M' =_{\text{Def}} M^{\star\star} \cup (M_0 \setminus M^\star)$ gilt $M' \subset M_0$, aber es ist $M' \sim M_0$, wie man leicht sieht. Das ist ein Widerspruch zur Voraussetzung 1. □

Folgerung 1.5.8
Eine Menge ist „unendlich“ genau dann, wenn sie unendlich ist.

1.5.3 Die Menge der natürlichen Zahlen

Definition 1.5.9 (Natürliche Zahlen, Null und Nachfolger)

1. *n ist eine natürliche Zahl, wenn n die Kardinalzahl einer endlichen Menge ist.*
2. $0 = \text{card}\,(\emptyset) = \{\emptyset\}$
3. $\mathbb{N} := \mathfrak{M}_{\text{fin}}/\sim$, *wobei $\sim$ die Gleichmächtigkeitsrelation ist.*
4. *Für natürliche Zahlen n, m definieren wir die Nachfolgerelation* Nf

$$n\,\text{Nf}\,m :\leftrightarrow \text{ Es gibt ein } N \in n \text{ und } a \in E \text{ mit } a \notin N \text{ und } N \cup \{a\} \in m.$$

Satz 1.5.10
Das Tripel $[0, \mathbb{N}, \text{Nf}]$ ist ein Modell des Peanoschen Axiomensystems.

Wir führen den Beweis nicht aus, merken lediglich an, daß man zum Nachweis der Existenz des Nachfolgers einer natürlichen Zahl das Unendlichkeitaxiom braucht.

Ist nämlich $n \in \mathbb{N}$, etwa $n = \text{card}\,(N)$ wobei $N \in \mathfrak{M}_{\text{fin}}$, so ist zu zeigen, daß ein $a \in E$ mit $a \notin N$ existiert.

Gäbe es ein solches a nicht, so hätte man $E \subseteq N$, also $E = M$, d. h. $E = N \in \mathfrak{M}_{\text{fin}}$, im Widerspruch zum Unendlichkeitsaxiom.

Definition 1.5.11
Es seien m, n beliebige Kardinalzahlen. Wir setzen $m \leq n$ gdw. es Mengen $M \in m$, $N \in n$ und eine Injektion $\Phi : M \to N$ gibt.

Satz 1.5.12
$[\mathfrak{E}/\sim, \leq]$ ist eine reflexive Ordnung, d. h. für beliebige Kardinalzahlen m, n, k gilt:

1. *$m \leq m$* *(Reflexivität)*
2. *Wenn $m \leq n$ und $n \leq k$, so $m \leq k$* *(Transitivität)*
3. *Wenn $m \leq n$ und $n \leq m$, so $m = n$* *(Antisymmetrie)*

4. $m \leq n$ *oder* $n \leq m$ *(Vergleichbarkeit)*

Bemerkung
Die ersten drei Behauptungen entsprechen den Halbordnungsaxiomen.

Die erste und zweite Behauptung ist leicht einzusehen. Wir beweisen hier nur die Antisymmetrie von $\leq$ in Form des

Satz 1.5.13 (Bernsteinscher Äquivalenzsatz)
Sind M, N *Mengen und ist* $\Phi : M \to N$ *injektiv, sowie* $\Psi : N \to M$ *injektiv, so ist* $M \sim N$.

Zur Vorbereitung zeigen wir folgenden

Hilfssatz (Fixpunktsatz)
Es sei M *eine beliebige Menge und* $f : \mathfrak{P}(M) \to \mathfrak{P}(M)$ *mit: Wenn* $A, B \subseteq M$ *und* $A \subseteq B$*, so* $f(A) \subseteq f(B)$*. Dann hat* f *einen Fixpunkt* C*, d. h. es gibt eine Menge* $C \subseteq M$ *mit*

$$f(C) = C$$

Beweis des Hilfssatzes
Es sei

$$\begin{array}{lll} \mathfrak{A} & =_{\text{Def}} & \{A | A \subseteq M \text{und } A \subseteq f(A)\} \\ C & =_{\text{Def}} & \bigcup \mathfrak{A}. \end{array}$$

Zum Beweis von $C = f(C)$ zeigen wir zuerst

$$C \subseteq f(C).$$

Es sei $x \in C$ beliebig. Dann gibt es ein $A \in \mathfrak{A}$ mit $x \in A$. Wegen $A \subseteq f(A)$ ist $x \in f(A)$, ferner ist $A \subseteq C$. Aus der Monotonie von f ergibt sich $X \in f(A) \subseteq f(C)$, also $x \in f(C)$.

Aus $C \subseteq f(C)$ folgt $f(C) \subseteq f(f(C))$, also ist $f(C) \in \mathfrak{A}$, d. h. $f(C) \subseteq \bigcup \mathfrak{A} = C$. Damit ist der Hilfssatz bewiesen. □

Beweis des Bernsteinschen Äquivalenzsatzes
Es seien nun M, N, Φ, Ψ mit den angegebenen Eigenschaften gegeben. Wir betrachten die Abbildung $f : \mathfrak{P}(M) \to \mathfrak{P}(M)$ mit:

$$f(X) =_{\text{Def}} M \setminus \overline{\Psi}\left(N \setminus \overline{\Phi}(X)\right)$$

für $X \subseteq M$, wobei

$$\begin{array}{lllll} \overline{\Phi}(X) & = & \{\Phi(x) | x \in X\} & \text{für} & X \subseteq M \\ \overline{\Psi}(Y) & = & \{\Psi(y) | y \in Y\} & \text{für} & Y \subseteq N \end{array}$$

gesetzt ist.

Wir zeigen zunächst, daß f monoton ist. Ist $A \subseteq B \subseteq M$, so $\overline{\Phi}(A) \subseteq \overline{\Phi}(B) \subseteq \overline{\Phi}(M) \subseteq N$, folglich $N \setminus \overline{\Phi}(A) \supseteq N \setminus \overline{\Phi}(B)$ und also

$$M \supseteq \overline{\Psi}\left(N \setminus \overline{\Phi}(A)\right) \supseteq \overline{\Psi}\left(N \setminus \overline{\Phi}(B)\right),$$

so daß

$$f(A) = M \setminus \overline{\Psi}\left(N \setminus \overline{\Phi}(A)\right) \subseteq M \setminus \overline{\Psi}\left(N \setminus \overline{\Phi}(B)\right) = f(B)$$

folgt.

Nach unserem Hilfssatz besitzt f einen Fixpunkt C. Wir setzen nun für $x \in M$

$$\tau(x) =_{\text{Def}} \begin{cases} \Phi(x), & \text{falls } x \in C \\ \Psi^{-1}(x), & \text{falls } x \notin C \end{cases}$$

und zeigen, daß $\tau : M \to N$ bijektiv ist.

1. τ ist injektiv (eineindeutig)

   Die Eindeutigkeit von τ ist offensichtlich, es genügt also zu zeigen:

   $$\text{Wenn } \tau(x_1) = \tau(x_2), \text{ so } x_1 = x_2$$

   In den Fällen, daß $x_1, x_2 \in C$ oder $x_1, x_2 \in M \setminus C$ ist, folgt das unmittelbar aus der Eineindeutigkeit von Φ bzw. Ψ^{-1} Es sei o.B.d.A. $x_1 \in C$, $x_2 \notin C$. Dann ist

   $$y =_{\text{Def}} \tau(x_1) = \Phi(x_1) \in N \text{und } y = \tau(x_2) = \Psi^{-1}(x_2)$$

   Wegen $x_1 \in C$ ist $y = \Phi(x_1) \in \overline{\Phi}(C)$. Andererseits ist $\Psi(y) = x_2 \notin C$, also $x_2 \notin f(C) = M \setminus \overline{\Psi}\left(N \setminus \overline{\Phi}(C)\right)$, folglich $x_2 \in \overline{\Psi}\left(N \setminus \overline{\Phi}(C)\right)$ wegen $x_2 \in M$.

   Es gibt also ein $y' \in N \setminus \overline{\Phi}(C)$ mit $x_2 = \Psi(y')$. Weil Ψ eineindeutig ist, folgt aus $\Psi(y) = x_2 = \Psi(y')$, daß $y = y'$ ist. Das steht im Widerspruch zu $y \in \overline{\Phi}(C)$, $y' \notin \overline{\Phi}(C)$. Der Fall $x_1 \in C$, $x_2 \notin C$ kann also nicht eintreten.

2. τ ist Abbildung auf N.

   Es sei $y \in N$ beliebig. Wenn $x =_{\text{Def}} \Psi(y) \notin C$, so ist $\tau(x) = \Psi^{-1}(x) = y$, es ist also y ein Wert von τ. Andernfalls ist $x \in C = f(C) = M \setminus \overline{\Psi}\left(N \setminus \overline{\Phi}(C)\right)$. Folglich ist $x \in M$ und $x \notin \overline{\Psi}\left(N \setminus \overline{\Phi}(C)\right)$. Wir zeigen, daß in diesem Fall $y \in \overline{\Phi}(C)$ ist. Sonst währe nämlich $y \in N \setminus \overline{\Phi}(C)$, also $x = \Psi(y) \in \overline{\Psi}\left(N \setminus \overline{\Phi}(C)\right)$. Widerspruch. Aus $y \in \overline{\Phi}(C)$ ergibt sich die Existenz eines $x' \in C$ mit $y = \Phi(x')$. Nun ist wegen $x' \in C$

   $$\tau(x') = \Phi(x') = y,$$

   also triff y auch in diesem Fall als Wert von τ auf.

Damit ist der BERNSTEINsche Äquivalenzsatz bewiesen. □

Man kann zeigen:

Satz 1.5.14
Für beliebige Kardinalzahlen m, n gilt: $m \leq n$ genau dann, wenn es Mengen $M \in \mathfrak{M}$, $N \in \mathfrak{N}$ und eine surjektive Abbildung $\Phi : N \to M$ gibt.

Definition 1.5.15 (Abzählbarkeit, Überabzählbarkeit)
Eine Menge M heißt abzählbar unendlich, *wenn sie gleichmächtig mit $\mathbb{N}$ ist. Ist M unendlich und $M \not\sim \mathbb{N}$, so wird M* überabzählbar *genannt.*

Die Kardinalzahl der Menge $\mathbb{N}$ wird mit $\aleph_0$ (Aleph Null) bezeichnet, dies ist die kleinste unendliche Kardinalzahl. Überabzählbar ist z. B. die Menge aller Folgen von Nullen und Einsen, d. h. die Menge $M_1 = \{\Phi | \Phi : \mathbb{N} \to \{0, 1\}\}$, diese Menge hat die gleiche Mächtigkeit wie die Menge der reellen Zahlen im Intervall $[0, 1]$ bzw. wie die Menge aller reellen Zahlen. Die Kardinalzahl dieser Menge wird mit $\aleph$ (Aleph) bezeichnet, es ist also $\aleph_0 < \aleph$.

Nun können wir gewohnte Operationen auf der Menge der natürlichen Zahlen mittels Kardinalzahlen definieren:

Definition 1.5.16 (Addition, Multiplikation)
Für Kardinalzahlen m, n sei

$$m + n =_{Def} \operatorname{card}(M \cup N)$$

für beliebiges $M \in m, N \in n$ mit $M \cap N = \emptyset$ und

$$m \cdot n =_{Def} \operatorname{card}(M \times N)$$

für beliebiges $M \in m, N \in n$.

Man zeigt ohne Schwierigkeiten, daß diese Definitionen vom Repräsentanten unabhängig sind, d. h. daß gilt:

1. Wenn $M \sim M'$, $N \sim N'$, $M \cap N = \emptyset$, $M' \cap N' = \emptyset$,
   so $\operatorname{card}(M \cup N) = \operatorname{card}(M' \cup N')$.
2. Wenn $M \sim M'$, $N \sim N'$, so $\operatorname{card}(M \times N) = \operatorname{card}(M' \times N')$.

Für natürliche Zahlen gelten dabei natürlich die üblichen Rechenregeln, das ist bei unendlichen Kardinalzahlen nicht der Fall, man verifiziert z. B. leicht, daß $\aleph_0 + \aleph_0 = \aleph_0$ ist. Es gilt aber z. B.

Satz 1.5.17
Es ist für beliebige Kardinalzahlen m, n: $m \leq n$ genau dann, wenn es eine Kardinalzahl k mit $m + k = n$ gibt.

Übung 13
Man führe die offen gebliebenen Beweise und außerdem:

1. $A \cup (B \setminus C) = ((A \cup B) \setminus C) \cup (A \cap C)$
2. $A \cap B = A \setminus (A \setminus B)$
3. $(A \setminus B) \times C = (A \times C) \setminus (B \times C)$
4. $(A \times B) \setminus (C \times D) = ((A \setminus C) \times B) \cup (A \times (B \setminus D))$
5. Es sei $f : R_1 \times R_2 \to R_1 \times R_2$ (R_1 = die Menge der reellen Zahlen) mit $f(x, y) = [x + y, x - y]$. Man untersuche, ob f surjektiv, injektiv bzw. bijektiv ist.
6. Es seien $f : X \to Y$, $g_1, g_2 : Y \to X$ mit

   $$f \circ g_1 = f \circ g_2 = I_X,\ g_1 \circ f = I_Y.$$

   man beweise: $g_1 = g_2$.
7. Es sei $\mathcal{Z}$ die Menge der ganzen Zahlen und

   $$R = \{[[a, b], [c, d]] \mid a, b, c, d \in \mathcal{Z},\ b \cdot d \neq 0, ad = bc\}$$

   Man zeige, daß R eine Äquivalenzrelation ist und charakterisiere die Äquivalenzklassen.

1.6 Wörter

In diesem Abschnitt beschäftigen wir uns mit einer Verallgemeinerung der natürlichen Zahlen — den Wörtern. Wörter spielen in der Informatik eine große Rolle, sie bilden die Grundlage für den Begriff der Sprache, also für den Kalkülbegriff einerseits und für Programmiersprachen andererseits. Die rechentechnische Realisierung erfolgt je nach Anwendungsbereich durch die Datentypen String (Zeichenkette) als ARRAY OF CHARACTER oder durch den Datentyp Liste (wenn die Länge der betrachteten Ketten unbeschränkt ist). Wörter werden als endliche Folgen von Buchstaben (= Elementen eines endlichen Alphabets) verstanden.

1.6.1 Grundbegriffe

Definition 1.6.1 (Alphabet, Wort)
Zuerst definieren wir

1. *Ein* Alphabet X *ist eine endliche, nichtleere Menge (die Elemente nennt man Buchstaben).*
2. p *ist ein* Wort *über* X*, wenn* p *eine endliche Folge von Elementen von* X *ist, d. h. es gibt eine natürliche Zahl* n*, so daß* $p : \{0, \ldots, n-1\} \to X$.
3. *Das Wort* $e : \emptyset \to X$ *heißt das* leere Wort.
4. *Die Zahl* n *wird als* Länge $l(p)$ *des Wortes* p *bezeichnet.*
5. $W(X)$ *bezeichnet die* Menge aller Wörter *über* X.

Bemerkung
Die Abbildung $l : W(X) \to \mathbb{N}$ ist nicht umkehrbar, falls $\operatorname{card}(X) \geq 2$.

Um die Menge $W(X)$ zu einer Algebra zu machen, definieren wir eine Operation auf $W(X)$:

Definition 1.6.2 (Verkettung, concatenation)
Es sei $p : \{0, \ldots, n-1\} \to X$, $q : \{0, \ldots, m-1\} \to X$. *Dann ist die* Verkettung *von* p *mit* q *die Abbildung* $p^\frown q : \{0, \ldots, m+n-1\} \to X$ *mit*

$$p^\frown q(i) := \begin{cases} p(i), & i \leq n-1 \\ q(i-n), & n \leq i \leq m+n-1. \end{cases}$$

Was hat diese Operation für Eigenschaften?

Satz 1.6.3
$[W(X), ^\frown, e]$ *ist eine Halbgruppe mit dem Einselement* e.

Beweis
Man rechnet leicht nach:

1. $p^\frown(q^\frown r) = (p^\frown q)^\frown r$ ($^\frown$ ist assoziativ)
2. $p^\frown e = p = e^\frown p$ für jedes $p \in W(X)$. (e ist Einselement in $[W(X), ^\frown]$)

Übung 14
Wir kennen bereits folgende Halbgruppe: $[\mathbb{N}, +, 0]$. Frage: Haben beide etwas Gemeinsames? Sind beide homomorph? (Standardfrage in der Mathematik in solchem Fall)

Definition 1.6.4 (Homomorphismus)
f ist Homomorphismus *von der Halbgruppe $[M, \circ, 1_M]$ in $[N, \star, 1_N]$ wenn gilt:*

1. $f : M \to N$
2. $f(1_M) = 1_N$
3. $f(m_1 \circ m_2) = f(m_1) \star f(m_2)$

Ist f überdies bijektiv, so heißt f Isomorphismus.

Folgerung 1.6.5
Die in 1.6.1 definierte Wortlängenfunktion l ist ein Homomorphismus von $[W(X), \frown, e]$ auf $[\mathbb{N}, +, 0]$.

Bemerkung
Wenn $\operatorname{card}(X) = 1$, so ist l umkehrbar eindeutig, (folglich bijektiv). Mit anderen Worten, die Halbgruppen $[\mathbb{N}, +, 0]$ und $[W(\{a\}), \frown, e]$ sind isomorph, d. h. $[\mathbb{N}, +, 0]$ und $[W(\{a\}), \frown, e]$ sind algebraisch gesehen identisch. Wörter sind also eine Verallgemeinerung von Zahlen, d. h. Zahlen sind spezielle Wörter.

Wir werden nun die Schreibweise vereinfachen, indem wir Struktureigenschaften der Wörter ausnutzen (z. B. erzeugendes Element)

Definition 1.6.6 (Erzeugendensystem)
E ist Erzeugendensystem *für $[W(X), \frown, e]$, wenn:*

1. $e \notin E \subseteq W(X)$
2. *zu jedem $p \in W(X) \setminus \{e\}$ gibt es Elemente $r_1, \ldots, r_k \in E$ mit $p = r_1 \frown r_2 \frown \ldots \frown r_k$.*

Man sucht natürlich einfache Elemente als erzeugende Elemente. Hierfür bieten sich Buchstaben an (sie sind keine Wörter!). Deshalb setzen wir fest:

Definition 1.6.7 (Buchstaben – Worte der Länge 1)
Für $x \in X$ sei $\underline{x} =_{Def} \{[0, x]\}$ Wort der Länge 1.

Damit haben wir die Buchstaben auf das Niveau Wort gebracht. Später werden wir $\underline{x}$ und x identifizieren.

Satz 1.6.8
$\underline{X} =_{Def} \{\underline{x} | x \in X\}$ ist ein Erzeugendensystem für $[W(X), \frown, e]$.

Beweis
Sei $p \in W(X)$ und $p : \{0, \ldots, n-1\} \to X$.
Offenbar ist $p = \underline{p(0)} \frown \underline{p(1)} \frown \ldots \frown \underline{p(n-1)}$, da $p(i) \in X$. □

Beispiele

- $\left[\mathbb{N}^+, \cdot, 1\right]$ ist eine Halbgruppe.
  Erzeugendensystem ist Pz – die Menge der Primzahlen, z. B. $6 = 2 \cdot 3 = 3 \cdot 2$
- $[\mathbb{N}, +, 0]$ ist Halbgruppe.
  Erzeugendensystem ist $\{1\}$, z. B. $3 = 1 + 1 + 1$.

Unterschied zwischen beiden Halbgruppen ist, daß die Elemente im ersten Beispiel in mehreren Weisen, im zweiten Beispiel dagegen nur auf eine Weise erzeugbar sind. Die Erzeugung der zweiten Halbgruppe ist frei von nichttrivalen Gleichungen.

Definition 1.6.9 (freie Halbgruppe)
$G = [M, \circ, 1, E]$ sei eine Halbgruppe mit Erzeugendensystem E. Die Halbgruppe G heißt frei erzeugt *(frei von nichttrivialen Gleichungen), wenn aus $m_1 \circ m_2 \circ \cdots \circ m_k = m'_1 \circ m'_2 \circ \cdots \circ m'_n$ stets $k = n$ und $m_1 = m'_1, \ldots, m_k = m'_k$ folgt (für $m_i, m'_i \in E$).*

Folgerung 1.6.10
$[W(X), \frown, e, \underline{X}]$ ist freie Halbgruppe.

Satz 1.6.11
Alle freien Halbgruppen mit derselben Zahl von erzeugenden Elementen sind isomorph.

Beweis
Idee: Erzeugende Elemente einer freien Halbgruppe sind nicht zerlegbar. Wir werden deshalb Erzeugende aufeinander abbilden.

Seien $[M_1, \circ, 1, E_1]$, $[M_2, \star, \underline{1}, E_2]$ Halbgruppen mit $\operatorname{card}(E_1) = \operatorname{card}(E_2)$.

Es sei $f : E_1 \to E_2$ bijektiv. Weiter definiert man den Homomorphismus folgendermaßen: $f(1) = \underline{1}$. Sei $m \in M_1$. m besitzt genau eine Darstellung $m = r_1 \circ_1 r_2 \circ \cdots \circ r_k$ $(r_i \in E_1)$. Dann sei $f(m) := f(r_1) \star f(r_2) \star \cdots \star f(r_k)$. Daß f ein Homomorphismus ist, ist klar.

Es bleibt zu zeigen, daß f bijektiv ist, d. h. daß f eine umkehrbare Abbildung auf M_2 ist.

Es sei $m' \in M_2$. Wenn $m' = \underline{1}$, so besitzt m' das Urbild $1 \in M_1$ wegen $f(1) = \underline{1}$. Für $m \in M_1 \setminus \{1\}$ ist $f(m) \neq \underline{1} \in M_2$, weil anderenfalls aus der Darstellung

$$m = r_1 \circ r_2 \circ \cdots \circ r_k \quad (r_i \in E_1)$$

von m in M_1 eine Darstellung

$$\underline{1} = f(m) = f(r_1) \star \cdots \star f(r_k) \quad (f(r_i) \in E_2)$$

des Einselements in M_2 gewonnen würde im Widerspruch dazu, daß $[M_2, \star, \underline{1}, E_2]$ frei ist (Die obige Darstellung wäre eine nichttriviale Gleichung). Also ist $1 \in M_1$ das einzige Urbild von $\underline{1} \in M_2$.

Ist $m' \in M_2 \setminus \{\underline{1}\}$, dann sei

$$m' = r'_1 \star \cdots \star r'_k \quad (r'_i \in E_2)$$

die Darstellung von m' aus Erzeugenden. Da f bijektiv von E_1 auf E_2 ist, gibt es eindeutig bestimmte $r_i \in E_1$ mit $f(r_i) = r'_i$ und

$$m = r_1 \circ \cdots \circ r_k$$

ist ein Element von M_1 mit $f(m) = m'$. Man überlegt sich wie oben, daß m eindeutig bestimmt ist. □

Bemerkung
Eine Abbildung $f : X \to Y$ zwischen den Alphabeten X und Y induziert immer einen Homomorphismus von $[W(X), \frown, e]$ auf $[W(Y), \frown, e]$. Allgemein kann man sogar schreiben $f : X \to W(Y)$.

Folgerung 1.6.12
Jede freie Halbgruppe mit endlichem Erzeugendensystem ist isomorph zu einer Worthalbgruppe.

$[\mathbb{N}, +, 0, \{1\}]$ ist eine Halbgruppe mit genau einem Erzeugenden und ist isomorph zu

$$\left[W\left(\{|\}\right), \frown, e, \left\{\underline{|}\right\}\right]$$

$[\mathbb{N}, +, 0, \{1\}]$ ist kommutativ, also auch

$$\left[W\left(\{|\}\right), \frown, e, \left\{\underline{|}\right\}\right]$$

folglich ist jede freie Halbgruppe mit genau einem Erzeugenden kommutativ.

Es ist bekannt, daß man bei der Übermittlung von Nachrichten mit 2 Zeichen auskommen kann (z. B. Computer: $0, 1$). Frage: Warum nicht mit einem Zeichen? Der Grund liegt in der Kommutativität, denn es gibt Halbgruppen, die nicht kommutativ sind. Dazu unternehmen wir einen kurzen Ausflug in die

Codierungstheorie

Unter einer Codierung C verstehen wir eine Abbildung $C : W(X) \to W(Y)$. Für die im Alphabet X formulierte Nachricht $p \in W(X)$ ist dann $C(p) \in W(Y)$ die im Alphabet Y (z. B. Morse–Alphabet) codierte Nachricht, der Code von p. Die Codierung C ist decodierbar, wenn die Abbildung eindeutig umkehrbar ist, d. h. aus $C(p) = C(q)$ stets $p = q$ folgt.

Die einfachsten Codierungen sind sogenannte kombinatorische Codierungen, nämlich Homomorphismen von $W(X)$ in $W(Y)$, die ja bereits durch die Teilabbildung $C : X \to W(Y)$ festgelegt sind. Für ein Wort $p = x_1 \cdots x_n$ ist dann $C(p) = C(x_1) \cdots C(x_n)$.

Eine solche Codierung ist genau dann decodierbar, wenn die Unterhalbgruppe, die von der Wertemenge $\{C(x) \mid x \in X\}$ in der freien Halbgruppe $W(Y)$ erzeugt wird, frei erzeugt ist. Klar, denn um ein Wort $q = y_1 \cdots y_m \in W(Y)$ zu decodieren, d. h. es in der Form $q = C(x_1) \cdots C(x_n)$ darzustellen, darf es nur eine solche Zerlegung von q in Elemente von $\{C(x) \mid x \in X\}$ geben.

Der folgende Satz sagt aus, daß es zu jedem endlichen Alphabet eine decodierbare Codierung in ein zweielementiges Alphabet gibt.

Satz 1.6.13 (Hauptsatz der Codierungstheorie)
In der freien Halbgruppe $[W(\{a, b\}), \frown, e, \{a, b\}]$ mit genau zwei Erzeugenden existiert zu jedem $k \geq 1$ eine freie Unterhalbgruppe mit genau k Erzeugenden.

Beweis
Sei k gegeben.

$$E_k =_{\text{Def}} \{a \underbrace{b}_{1} a, \ldots, a \underbrace{b \ldots b}_{k} a\}$$

Sei $E_k^\star$ die Menge aller Wörter, die durch endlich oftmaliges Verketten von Elementen aus E_k entstehen. $H_k = [E_k^\star, \frown, e, E_k]$ ist frei, die Codierung C mit $C(x_k) = a \underbrace{b \ldots b}_{k} a$ ist decodierbar. □

1.6.2 Induktive Beweise in $[W(X), \frown, e, X]$

Beispiel $[\mathbb{N}, +, 0, \{1\}]$
Um zu zeigen, daß eine Aussage $H(n)$ für alle n aus $\mathbb{N}$ gilt, genügt es zu zeigen:

A) $H(0)$

I) Wenn $H(n)$, so $H(n+1)$.

Wir verallgemeinern diese Aussage für Halbgruppen mit beliebigem Erzeugendensystem:

Induktionstheorem
Um zu zeigen, daß eine Aussage H auf alle Elemente p einer Halbgruppe $[M, \circ, 1, E]$ mit dem Erzeugendensystem E zutrifft, genügt es zu zeigen

A) $H(1)$

I) Für alle Erzeugenden x gilt: Wenn $H(r)$ und $x \in E$, so $H(r \circ x)$
(bzw.: Wenn $H(r)$ und $x \in E$, so $H(x \circ r)$)

Beweis des Induktionstheorems
erfolgt indirekt:

Nehmen wir an, daß es Elemente $q \in M$ gibt, auf die H nicht zutrifft. Unter diesen Elementen kommt das Einselement nicht vor, weil $H(1)$ gilt (Anfangsschritt). Jedes dieser Elemente besitzt also eine Darstellung

$$q = q_1 \circ q_2 \circ \cdots \circ q_k$$

aus Erzeugenden $q_i \in E$. Es sei $q^\star$ ein Element mit einer kürzesten Darstellung

$$q^\star = q_1^\star \circ \cdots \circ q_{k^\star}^\star \quad (k^\star \text{ minimal}).$$

Wir setzen $q^{\star\star} = q_1^\star \circ \cdots \circ q_{k^\star - 1}^\star$ falls $k^\star \geq 2$ und $q^{\star\star} = 1$ falls $k^\star = 1$. Wegen der Minimalität von $k^\star$ gilt $H(q^{\star\star})$ und es ist

$$q^\star = q^{\star\star} \circ q_{k^\star}^\star, \quad q_{k^\star}^\star \in E$$

Aus I) erhalten wir damit $H(q^\star)$; Widerspruch. □

Auch das Prinzip der ordnungstheoretischen Induktion kann auf (freie) Halbgruppen übertragen werden:

Bemerkung (Ordnungstheoretische Induktion in $[\mathbb{N}, \leq]$)
Um zu zeigen, daß eine Aussage H für alle $n \in \mathbb{N}$ gilt, ist z. z.: Wenn H auf alle $k < n$ zutrifft, so trifft H auf n zu.

Definition 1.6.14 (Relationen in $W(X)$)

1. *$p \underline{\text{in}}\, q \leftrightarrow_{\text{Def}}$ es gibt Wörter r, s mit $q = rps$*
2. *$p \sqsubseteq q$ (p ist Anfangsstück von q) $\leftrightarrow_{\text{Def}}$ Es gibt s mit $q = ps$*
3. *$p \sqsupseteq q$ (p ist Endstück von q) $\leftrightarrow_{\text{Def}}$ Es gibt r mit $q = rp$*

Satz 1.6.15 (Halbordnungen in $W(X)$)
$[W(X), \underline{\text{in}}]$, $[W(X), \sqsubseteq]$, $[W(X), \sqsupseteq]$ sind reflexive Halbordnungen.

Beweis
Wir zeigen hier nur, daß $\underline{\text{in}}$ folgende Eigenschaften hat:

- reflexiv: $p \underline{\text{in}}\, p$ $(r, s = e)$
- transitiv: $p \underline{\text{in}}\, q$ und $q \underline{\text{in}}\, r$, so $p \underline{\text{in}}\, r$
  $(q = r_1 p s_1,\ r = r_2 q s_2$ und mit $r_3 = r_2 r_1$ und $s_3 = s_1 s_2$ folgt $r = r_3 p s_3)$
- antisymmetrisch: $p \underline{\text{in}}\, q$ und $q \underline{\text{in}}\, p$ so $p = q$
  $(q = r_1 p s_1,\ p = r_2 q s_2$ und mit $r_1, r_2, s_1, s_2 = e$ folgt $p = q)$

□

Übung 15
Man führe die restlichen Beweise analog.

Folgerung 1.6.16
Wenn $p \underline{\text{in}}\, q$ so $l(p) \leq l(q)$.
Wenn $p \sqsubseteq q$ so $l(p) \leq l(q)$.
Wenn $p \sqsupseteq q$ so $l(p) \leq l(q)$.

Satz 1.6.17 (Ordnungstheoretische Induktion in $W(X)$)
Um zu zeigen, daß eine Aussage H auf alle $p \in W(X)$ zutrifft, genügt es zu zeigen, daß für alle p gilt:

> *Wenn für alle r mit $r \underline{\text{in}}\, p$ (bzw. $r \sqsubseteq p$ bzw. $r \sqsupseteq p$) und $r \neq p$ stets $H(r)$ gilt, so gilt $H(p)$.*

Nehmen wir an, daß H nicht für alle $p \in W(X)$ gilt, dann existiert ein kürzestes Wort p so, daß H nicht auf p zutrifft. Alle r mit $r \underline{\text{in}}\, p$ (bzw. $r \sqsubseteq p$ bzw. $r \sqsupseteq p$) und $r \neq p$ sind aber echt kürzer als p, d. h. H trifft auf alle Wörter r zu; folglich trifft H auf p zu. Widerspruch.

1.6.3 Induktionsbeweise in Algebren $[M, \Phi, E]$

Wir betrachten eine beliebige Algebra $[M, \Phi, E]$ mit Erzeugendensystem E. Hierbei ist M eine Menge, Φ ist eine Menge von Abbildungen $\phi : M \times \cdots \times M \to M$ und E ist eine Teilmenge von M derart, daß jedes Element von M aus Elementen von E durch Anwendung von Funktionen aus Φ erzeugt werden kann.

Beispiel $\left[\mathfrak{M}_{\text{fin}}, \{+a | a \in E\}, \{\emptyset\}\right]$
Dabei ist die Funktion $+a$ für $a \in E$ (die Allmenge erster Stufe) und $N \in \mathfrak{M}_{\text{fin}}$ definiert als $+a(N) =_{\text{Def}} N \cup \{a\}$. Es ist also zum Beispiel

$$\{a, b, c\} = +c(+b(+a(\emptyset))).$$

Um zu zeigen, daß $H(m)$ für alle $m \in M$ gilt, genügt es zu zeigen, daß

1. $H(x)$ für alle Erzeugenden $x \in E$ gilt (Anfangsschritt)
2. Für alle $\phi^n \in \Phi$, $a_1 \ldots, a_n \in M$ mit $H(a_1), \ldots, H(a_n)$ gilt:

$$H\big(\underbrace{\phi^n(a_1, \ldots a_n)}_{\in M}\big)$$

1.6.4 Induktive Definitionen

Eine induktive Definition einer $(k+1)$-stelligen Funktion f im Bereich der natürlichen Zahlen folgt dem Schema

$$\begin{aligned} f(x_1, \cdots, x_k, 0) &= g(x_1, \cdots, x_k) \\ f(x_1, \cdots, x_k, n+1) &= h(x_1, \ldots, x_k, n, f(x_1, \ldots, x_k, n)) \end{aligned}$$

Die Definition ist implizit, da f auf der rechten und auf der linken Seite vorkommt. Eine implizite Definition muß gerechtfertigt werden:

Satz 1.6.18 (Dedekindscher Rechtfertigungssatz)
Sind g, h Funktionen, so gibt es genau eine Funktion f, die das Schema erfüllt.

Beweis
Wir zeigen

1. es existiert eine Funktion f
2. es existiert genau eine Funktion f.

Einzigkeit:

Es seien f_1, f_2 Funktionen, die das Gleichungssystem erfüllen.

Wir zeigen durch Induktion über i, daß für alle $x_1, \ldots, x_k, i$

$$f_1(x_1, \ldots, x_k, i) = f_2(x_1, \ldots, x_k, i)$$

ist.

Im Anfangsschritt ist $i = 0$, also

$$\begin{aligned} f_1(x_1, \ldots, x_k, i) &= f_1(x_1, \ldots, x_k, 0) \\ &= g(x_1, \ldots, x_k) \\ &= f_2(x_1, \ldots, x_k, 0) \\ &= f_2(x_1, \ldots, x_k, i) \end{aligned}$$

Im Induktionschritt schließen wir von i auf $i+1$:

$$\begin{aligned} f_1(x_1, \ldots, x_k, i+1) &= h(x_1, \ldots, x_k, i, f_1(x_1, \ldots x_k, i)) \\ &= h(x_1, \ldots, x_k, i, f_2(x_1, \ldots x_k, i)) \text{ (Voraussetzung)} \\ &= f_2(x_1, \ldots, x_k, i+1) \end{aligned}$$

Existenz:

Zum Beweis der Lösbarkeit des Gleichungssystems lösen wir die induktive Definition in eine unendliche Folge von expliziten Definitionen auf:

$$\begin{aligned} f_0(x_1, \ldots, x_k, j) &:= \begin{cases} g(x_1, \ldots, x_k), & \text{für } j = 0 \\ \text{nicht definiert}, & \text{sonst} \end{cases} \\ f_1(x_1, \ldots, x_k, j) &:= \begin{cases} f_0(x_1, \ldots, x_k, j), & \text{für } j < 1 \\ h(x_1, \ldots, x_k, 0, f_0(x_1, \ldots x_k, 0)), & \text{für } j = 1 \\ \text{nicht definiert}, & \text{sonst} \end{cases} \\ &\vdots \\ f_{i+1}(x_1, \ldots, x_k, j) &:= \begin{cases} f_i(x_1, \ldots, x_k, j), & \text{für } j < i+1 \\ h(x_1, \ldots, x_k, i, f_i(x_1, \ldots x_k, i)), & \text{für } j = i+1 \\ \text{nicht definiert}, & \text{sonst} \end{cases} \\ &\vdots \end{aligned}$$

Diese sind alles explizite Definitionen, die Funktionen f_i existieren also gemäß den Mengenbildungsaxiomen.

Folglich existiert auch f mit

$$f(x_1, \ldots, x_k, i) \quad =_{\text{Def}} \quad f_i(x_1, \ldots, x_k, i)$$

und es gilt

$$\begin{aligned} f(x_1, \ldots, x_k, 0) &= f_0(x_1, \ldots, x_k, 0) \\ &= g(x_1, \ldots, x_k) \end{aligned}$$

sowie

$$\begin{aligned} f(x_1, \ldots, x_k, i+1) &= f_{i+1}(x_1, \ldots, x_k, i+1) \\ &= h(x_1, \ldots x_k, i, f_i(x_1, \ldots, x_k, i)) \\ &= h(x_1, \ldots x_k, i, f(x_1, \ldots, x_k, i)) \end{aligned}$$

d. h. f erfüllt das Gleichungssystem.

□

Bemerkung
Dieses Definitionsprinzip kann auf Worthalbgruppen verallgemeinert werden (also auf beliebige *freie* Halbgruppen), aber *nicht* auf beliebige Halbgruppen oder gar auf beliebig erzeugte Algebren.

Es seien $\mathfrak{X}, \mathfrak{Y}$ beliebige Mengen und X ein Alphabet. Um eine Abbildung $f : \mathfrak{X} \times W(X) \to \mathfrak{Y}$ zu definieren genügt es anzugeben

- eine Abbildung $g : \mathfrak{X} \to \mathfrak{Y}$
- zu jedem $x \in X$ eine Abbildung $h_x : \mathfrak{X} \times W(X) \times \mathfrak{Y} \to \mathfrak{Y}$

Die Definition von f sieht dann so aus:

$$\left.\begin{aligned} f(\mathfrak{x}, e) &= g(\mathfrak{x}) \\ f(\mathfrak{x}, px) &= h_x(\mathfrak{x}, p, f(\mathfrak{x}, p)) \end{aligned}\right\} \begin{array}{c} \mathfrak{x} \in \mathfrak{X} \\ p \in W(X) \\ x \in X \end{array}$$

Dieses Gleichungssystem (bestehend aus $\operatorname{card}(X) + 1$ Gleichungen) wird von genau einer Funktion $f : \mathfrak{X} \times W(X) \to \mathfrak{Y}$ erfüllt.

1.7 Sprachen

Wir haben die Theorie der Wörter mit dem Ziel entwickelt, mathematische Modelle zu bilden, d .h. Modelle, deren Elemente Wörter sind. Dazu fixieren wir ein (endliches) Alphabet X und betrachten die Menge $W(X)$. Im allgemeinen zeigt sich sofort, daß nicht allen Wörtern über X eine Bedeutung zukommt, nur gewisse wohlbestimmte Wörter haben eine Bedeutung, diese bilden die Sprache L des Modells.

1.7.1 Grundbegriffe

Definition 1.7.1 (Sprache)
Ist $X \neq \emptyset$ ein endliches Alphabet, so wird jede Teilmenge L der Menge aller Wörter über dem Alphabet X, also

$$L \subseteq W(X)$$

als Sprache über X *bezeichnet.*

Im Sinne der Sprachwissenschaft erfaßt diese Definition nur geschriebene Sprache, also Texte.

Beispiel
Für die deutsche Sprache nehmen wir als *Alphabet* die Menge aller deutschsprachigen Wörter in allen grammatischen Formen, die deutsche *Sprache* ist dann gegeben als Menge aller Wörter über diesem Alphabet (also aller endlichen Folgen von deutschen Wörtern), die (syntaktisch) richtig gebildete Sätze sind, auch wenn ihr semantischer Inhalt dunkel bleibt oder als sinnleer empfunden wird.

Satz 1.7.2
Für ein gegebenes endliches Alphabet X ist $W(X)$ eine abzählbar unendliche Menge und $\mathfrak{P}(W(X))$, die Menge aller Sprachen über X, ist überabzählbar.

Für die Bildung mathematischer Modelle interessante Sprachen sind unendlich. Damit entsteht die Frage nach (endlichen) Beschreibungen für Sprachen, denn zur Beschreibung des Modells gehört die Beschreibung „seiner Syntax“, als der Sprache des Modells.

Wir können eine Sprache $L \subseteq W(X)$ als hinreichend beschrieben (abgegrenzt) ansehen, wenn wir

(a) ein Erzeugungsverfahren angeben können, mit dessen Hilfe jedes Element $p \in L$ (und kein Element aus $W(X) \setminus L$) hergestellt werden kann

oder

(b) ein Akzeptierungsverfahren (oder Entscheidungsverfahren) angeben können, mit dessen Hilfe von jedem $p \in W(X)$ festgestellt werden kann, ob $p \in L$ ist.

Ein Vorgehen nach (a) wäre eine Formalisierung der Fähigkeit eines Sprechers von L, nur richtig gebildete Sätze der Sprache L zu produzieren, während (b) die Fähigkeit eines Hörers der Sprache L formalisiert, nur richtig gebildete Sätze zu akzeptieren.

Beispiel
Ein Programmierer in der Sprache MODULA ist in diesem Sinne ein Sprecher, während der M2-Compiler den Hörer realisiert.

Zur Formalisierung des Sprecher–Aspekts dient der Begriff der Grammatik, den wir im folgenden für den wichtigen Spezialfall der kontext-freien Grammatik definieren:

Definition 1.7.3 (kontextfreie Grammatik)
$\mathfrak{G} = [X, H, s, R]$ *ist eine* kontextfreie Grammatik*, wenn*

1. *X, H sind endliche nichtleere disjunkte Mengen, d. h. $X \cap H = \emptyset$*
2. *$s \in H$ ist das Satzsymbol oder Startsymbol von $\mathfrak{G}$*

3. R ist eine endliche Teilmenge von $H \times W(X \cup H)$.

Dabei ist X das Alphabet der zu beschreibenden Sprache, die Elemente von H werden Hilfssymbole (oder auch – in der Linguistik – syntaktische Kategorien) genannt und die Elemente von R werden als Ersetzungsregeln bezeichnet (oder Ableitungsregeln). Die Idee ist, daß jedes Wort $p \in L$ aus dem Startsymbol s mittels Regeln aus R „abgeleitet“ werden kann und nur die $p \in L$ diese Eigenschaft haben.

Definition 1.7.4 (Ableitbarkeit in einem Schritt)
Es sei w_1,$w_2 \in W(X \cup H)$. Dann ist aus w_1 das Wort w_2 in einem Schritt mittels R ableitbar:

$$w_1 \overset{1}{\underset{R}{\vdash\!\!\rightarrow}} w_2$$

wenn es $h \in H$ und Wörter $p, q, r \in W(X \cup H)$, so gibt, daß

$$w_1 = p\,h\,q,\ w_2 = p\,r\,q \text{ und } [h, r] \in R \text{ ist.}$$

w_2 entsteht also aus $w_1 = p\,h\,q$ dadurch, daß h, die linke Seite der Ersetzungsregel $[h, r] \in R$, durch das Wort r, die rechte Seite der Ersetzungsregel, ersetzt wird. Man schreibt daher statt $[h, r] \in R$ auch $h \longrightarrow r \in R$.

Definition 1.7.5 (Ableitbarkeit)
Mit $\overset{*}{\underset{R}{\vdash\!\!\rightarrow}}$ bezeichnen wir die reflexiv-transitive Hülle der Relation $\overset{1}{\underset{R}{\vdash\!\!\rightarrow}}$.

Folgerung 1.7.6
Es gilt $w \overset{*}{\underset{R}{\vdash\!\!\rightarrow}} w'$ genau dann, wenn $w = w'$ ist oder endlich viele Wörter $w_1, \ldots, w_n$ $(n \geq 0)$ existieren mit

$$w \overset{1}{\underset{R}{\vdash\!\!\rightarrow}} w_1 \overset{1}{\underset{R}{\vdash\!\!\rightarrow}} \cdots \overset{1}{\underset{R}{\vdash\!\!\rightarrow}} w_n \overset{1}{\underset{R}{\vdash\!\!\rightarrow}} w'.$$

Definition 1.7.7 (erzeugte Sprache)
Die von einer Grammatik $\mathfrak{G}$ erzeugte Sprache L ist

$$L(\mathfrak{G}) =_{Def} \left\{p \mid p \in W(X) \wedge s \overset{*}{\underset{R}{\vdash\!\!\rightarrow}} p\right\}$$

1.7.2 Beispiel: Sprache der Mengengleichungen

Als Beispiel wollen wir die Sprache der Mengengleichungen, die wir im Zusammenhang mit dem Dualitätstheorem verbal beschrieben haben, durch eine Grammatik erzeugen.

Eine Mengengleichung hat die Form $T_1 = T_2$, wobei T_1, T_2 Mengenterme sind, Terme setzen sich aus Mengenvariablen, den Operationszeichen und Klammern zusammen, Mengenvariablen M_i schreiben wir als ein Wort der Form $M\underbrace{\star\cdots\star}_{i}$.

Wir setzen also $X = \{M, \star, =, (,), \cap, \cup\}$ und $H = \{S, T, V\}$ (S ist das Startsymbol, T steht für Term, V für Variable).

Die Menge R der Regeln ist

$$\begin{array}{lll@{\qquad}lll} S & \longrightarrow & T = T & T & \longrightarrow & V \\ T & \longrightarrow & (T \cap T) & T & \longrightarrow & (T \cup T) \\ V & \longrightarrow & M & V & \longrightarrow & V\star \end{array}$$

Wir können nun die Gleichung $M_0 \cap (M_0 \cup M_1) = M_0$, d. h.

$$(M \cap (M \cup M\star)) = M$$

als Element von $L([X, H, S, R])$ wie folgt nachweisen:

$$\begin{array}{llll}
S & \overset{1}{\underset{R}{\vdash\!\!\rightarrow}} & T = T & \overset{1}{\underset{R}{\vdash\!\!\rightarrow}} \quad T = V \\
 & \overset{1}{\underset{R}{\vdash\!\!\rightarrow}} & T = M & \overset{1}{\underset{R}{\vdash\!\!\rightarrow}} \quad (T \cap T) = M \\
 & \overset{1}{\underset{R}{\vdash\!\!\rightarrow}} & (T \cap (T \cup T)) = M & \overset{1}{\underset{R}{\vdash\!\!\rightarrow}} \quad (V \cap (T \cup T)) = M \\
 & \overset{1}{\underset{R}{\vdash\!\!\rightarrow}} & (M \cap (T \cup T)) = M & \overset{1}{\underset{R}{\vdash\!\!\rightarrow}} \quad (M \cap (V \cup T)) = M \\
 & \overset{1}{\underset{R}{\vdash\!\!\rightarrow}} & (M \cap (M \cup T)) = M & \overset{1}{\underset{R}{\vdash\!\!\rightarrow}} \quad (M \cap (M \cup V)) = M \\
 & \overset{1}{\underset{R}{\vdash\!\!\rightarrow}} & (M \cap (M \cup V\star)) = M & \overset{1}{\underset{R}{\vdash\!\!\rightarrow}} \quad (M \cap (M \cup M\star)) = M
\end{array}$$

1.7.3 Weitere Grundbegriffe

Kontextfreie Grammatiken spielen eine große Rolle bei der Definition von Programmierspachen. Dabei geht man i. a. so vor, daß eine kontextfreie Grammatik angegeben wird, die alle compilierbaren Programme und einige weitere Wörter erzeugt und danach durch weitere Forderungen („ alle Variablen müssen deklariert sein“,...) die „richtigen“ Programme bestimmt werden. Die Grammatik wird uns dabei in der sogenannten Backus–Nauer–Form (BNF) angegeben:

> Die syntaktischen Kategorien sind durch spitzgeklammerte Wörter gegeben, statt „→“ schreibt man „::=“, Regeln mit gleicher linker Seite werden zusammengefaßt.

In unserem Beispiel ergibt sich

$$\begin{array}{rcl}
\langle \mathit{gleichung} \rangle & ::= & \langle \mathit{term} \rangle \text{ '=' } \langle \mathit{term} \rangle \\
\langle \mathit{term} \rangle & ::= & \langle \mathit{variable} \rangle \mid \text{'('} \langle \mathit{term} \rangle \text{ '}\cap\text{' } \langle \mathit{term} \rangle \text{ ')'} \mid \\
 & & \text{'('} \langle \mathit{term} \rangle \text{ '}\cup\text{' } \langle \mathit{term} \rangle \text{ ')'} \\
\langle \mathit{variable} \rangle & ::= & \text{'M'} \mid \langle \mathit{variable} \rangle \text{ '}\star\text{'}
\end{array}$$

Zu den kontextfreien Grammatiken gehört als Akzeptierungsmechanismus der sogenannte Kellerautomat; diese Begriff wird in der Compilertechnik intensiv behandelt.

Natürlich können Sprachen auch in anderer Weise, z.B. durch Induktion, definiert werden. Wir werden Beispiele im nächsten Kapitel kennenlernen.

Kapitel 2

Aussagenlogik

In diesem Kapitel beschäftigen wir uns mit der Formalisierung eines Teilbereichs des menschlichen Denkens, dem logischen Schließen. Wir wollen ein mathematisches Modell dieses Teilbereiches angeben, das uns gestattet, logisches Schließen durch Rechnen zu ersetzen; dadurch können logische Schlüsse überprüft werden.

Logisches Schließen ist eine Operation mit Aussagen, die auf der Basis von sogenannten Schlußregeln arbeitet. Von einer richtigen Schlußregel erwartet man, daß sie von gültigen Aussagen zu gültigen Aussagen führt.

Es sind also zunächst folgende Fragen zu klären:

- Was ist eine Aussage?
- Wann ist eine Aussage wahr?
- Was ist eine gültige Schlußregel?

2.1 Grundbegriffe

Unter einer *Aussage* verstehen wir die sprachliche Widerspiegelung eines Sachverhalts (in Form eines Aussagesatzes).

Beispiele
Es regnet; wenn es regnet, dann ist die Straße naß.

Eine Aussage wird wahr genannt, wenn der ausgedrückte Sachverhalt vorliegt. Die Aussage „es regnet“ ist also genau dann wahr, wenn es regnet.

In der Realität ist häufig unklar, ob eine Aussage wahr ist oder nicht, weil Maßstäbe unscharf sind, z. B. „Gestern war schlechtes Wetter“. Wir gehen hier von einer Idealisierung der wirklichen Verhältnisse aus und verlangen als Axiom:

Satz der Zweiwertigkeit
Jede Aussage ist entweder wahr oder falsch.

Diese Idealisierung hat sich insbesondere in der Mathematik und der Informatik bewährt. Die sogenannte Fuzzy–Logik bietet Ansätze, ohne dieses Axiom auszukommen.

Nach dem Satz der Zweiwertigkeit zerfällt die Menge aller Aussagen erschöpfend und disjunkt in zwei Klassen, die Menge W aller wahren Aussagen und die Menge F aller

falschen Aussagen. Wir nennen W und F die Wahrheitswerte; wir haben also genau zwei Wahrheitswerte.

Neben den Sachverhalten selbst beobachten wir in der Realität Beziehungen zwischen Sachverhalten, die sich in Beziehungen und Operationen mit Aussagen widerspiegeln.

Eine n-stellige *Aussagenfunktion* α ist also eine Abbildung, die jedem n-Tupel von Aussagen eine Aussage zuordnet. Einige wichtige Aussagenfunktionen spiegeln sich sprachlich sehr einfach in Bindewörtern wider, z. B.

- Konjunktion „und“ $\qquad \text{und}(A, B) = A \,\text{und}\, B$
- Alternative „oder“ $\qquad \text{oder}(A, B) = A \,\text{oder}\, B$
- Negation „nicht“ $\qquad \text{nicht}(A) = \text{nicht}\, A$
- Implikation „wenn. . . , so. . . “
- Äquivalenz „genau dann, wenn“

Aussagen, wahre wie falsche, können nach verschiedenen Gesichtspunkten geordnet werden, z. B. nach Wissensgebieten (etwa Physik, Mathematik, . . .). Wir streben hier eine Formalisierung der Logik an, die in allen Gebieten anwendbar ist, in denen logisch gedacht wird. Damit stellt sich die Frage, welche Eigenschaften von Aussagen in unserem Formalismus (dem Aussagenkalkül) widergespiegelt werden sollen.

Wir beantworten diese Frage einschränkend: die einzige Eigenschaft ist Wahrheit bzw. Falschheit. Wir berücksichtigen hier beispielsweise nicht, daß Aussagen etwas über einzelne Gegenstände aussagen („3 ist eine Primzahl“) oder die Struktur einer Allaussage haben („jede Äquivalenzrelation ist transitiv“); das wird erst in der Prädikatenlogik (ab Seite 123) geschehen.

Was geschieht bei dieser Abstraktion Aussage → Wahrheitswert mit den Aussagenverbindungen (Aussagenfunktionen)? Mit anderen Worten, kann zu jeder Aussagenfunktion

$$\alpha : \bigtimes_n \mathfrak{A} \to \mathfrak{A},$$

wobei $\mathfrak{A}$ die Menge aller Aussagen ist, eine Wahrheitsfunktion

$$\phi : \bigtimes_n \{\mathrm{W}, \mathrm{F}\} \to \{\mathrm{W}, \mathrm{F}\},$$

d. h. eine Abbildung, die jedem n-Tupel von Wahrheitswerten einen Wahrheitswert zuordnet, gefunden werden, so daß für $A_1, \ldots, A_n$ gilt:

$$\mathrm{WW}(\alpha(A_1, \ldots, A_n)) = \phi(\mathrm{WW}(A_1), \ldots, \mathrm{WW}(A_n))$$

wobei WW die Funktion ist, die jeder Aussage A ihren Wahrheitswert $\mathrm{WW}(A)$ zuordnet, also W wenn A wahr ist und F sonst.

Für die Aussagenfunktion „und“ funktioniert das offensichtlich; die Wahrheitsfunktion et mit der Tabelle

w_1	w_2	$\mathrm{et}(w_1, w_2)$
W	W	W
W	F	F
F	W	F
F	F	F

leistet das Verlangte, denn die Aussage „A und B“ ist genau dann wahr, wenn die Aussage A wahr ist und die Aussage B wahr ist.

Definition 2.1.1 (Extensionale Aussagenfunktion)
Eine Aussagenfunktion $\alpha : \bigtimes_n \mathfrak{A} \to \mathfrak{A}$ *heißt* extensional*, wenn es eine Wahrheitsfunktion (*BOOLE*sche Funktion)* $\phi : \bigtimes_n \{\mathrm{W}, \mathrm{F}\} \to \{\mathrm{W}, \mathrm{F}\}$ *so gibt, daß für alle* $A_1, \ldots, A_n \in \mathfrak{A}$ *gilt*

$$\mathrm{WW}(\alpha(A_1, \ldots, A_n)) = \phi(\mathrm{WW}(A_1), \ldots, \mathrm{WW}(A_n))$$

Die Aussagenfunktion „und“ ist also extensional, man stellt leicht fest, daß dasselbe für „oder“, „nicht“, „wenn..., so...“ und „genau dann, wenn“ gilt; die zugehörigen (sogenannten klassischen) Wahrheitsfunktionen werden als vel, non, seq (von sequitur) und aeq bezeichnet:

w_1	w_2	$\mathrm{vel}(w_1, w_2)$	$\mathrm{seq}(w_1, w_2)$	$\mathrm{aeq}(w_1, w_2)$	$\mathrm{non}(w_1)$
W	W	W	W	W	F
W	F	W	F	F	F
F	W	W	W	F	W
F	F	F	W	W	W

Gibt es überhaupt Aussagenfunktionen, die nicht extensional sind, die wir also *nicht* durch Wahrheitsfunktionen widerspiegeln können? Das ist der Fall, als Beispiel betrachte man die Aussagenfunktionen „weil“ bzw. „damit“:

$$\begin{aligned} \mathrm{weil}(A, B) &= A, \text{ weil } B \\ \mathrm{damit}(A, B) &= A, \text{ damit } B \end{aligned}$$

die eine kausale bzw. pragmatische Beziehung beschreiben. Diese Aussagenfunktionen können also nicht durch Wahrheitsfunktionen repräsentiert werden.

2.2 Syntax und Semantik des Aussagenkalküls

2.2.1 Aufbau des Kalküls — Syntax

In unserem Kalkül brauchen wir Variable für Wahrheitswerte, die wir Aussagenvariable nennen werden, und Zeichen (Funktoren) für Wahrheitsfunktionen. Aus den Zeichen p und $\star$ bilden wir Wörter, unter denen die Aussagenvariablen ausgezeichnet werden, ferner verwenden wir

$\neg$ als Zeichen für die Wahrheitsfunktion non
$\wedge$ als Zeichen für die Wahrheitsfunktion et
$\vee$ als Zeichen für die Wahrheitsfunktion vel
$\to$ als Zeichen für die Wahrheitsfunktion seq
$\leftrightarrow$ als Zeichen für die Wahrheitsfunktion aeq

sowie die technischen Zeichen (und). Das Alphabet des Aussagenkalküls ist also

$$X = \{p, \star, \neg, \wedge, \vee, \to, \leftrightarrow,), (\}$$

Aussagenvariable
wird induktiv definiert:

A) p ist eine Aussagenvariable

I) Ist das Wort w eine Aussagenvariable, so auch $w^\frown\star$

Die Menge AV der Aussagenvariablen ist die kleinste Menge von Wörtern über X, die das Wort p enthält, und abgeschlossen ist in bezug auf das Anhängen des Zeichens $\star$.

$$AV = \{p\underbrace{\star\cdots\star}_{i} | i \geq 0\}$$

Schreibweise: $p\star$ wird abgekürzt als q, $p\star\star$ als r, allgemein $p\underbrace{\star\cdots\star}_{i}$ als p_i.

Ausdruck
wird ebenfalls induktiv definiert:

A) Jede Aussagenvariable ist ein Ausdruck

I) Sind H_1, H_2 Ausdrücke, so auch die folgenden Zeichenreihen:
$\neg H_1$,$(H_1 \wedge H_2)$, $(H_1 \vee H_2)$, $(H_1 \rightarrow H_2)$, $(H_1 \leftrightarrow H_2)$.

Die Menge ausd der Ausdrücke ist die kleinste Menge von Wörtern, die alle Aussagenvariablen enthält und abgeschlossen ist bezüglich der unter I) angegebenen Konstruktionen.

Übung 16
Geben sie eine Grammatik $\mathfrak{G}$ mit $L(\mathfrak{G}) =$ ausd an.

Beispiele
Es sind $\neg p$, $(\neg p \vee p)$, $((\neg p_7 \wedge p_5) \rightarrow p_1)$ Ausdrücke; keine Ausdrücke: e, 11, (p).

Folgerung 2.2.1
Durch die Klammerung ist für jeden Ausdruck kenntlich, auf welche Weise er aus Aussagenvariablen oder einfacheren Ausdrücken aufgebaut ist, genauer: jeder Ausdruck kann auf genau eine Weise aus Aussagenvariablen aufgebaut werden. Die Menge ausd *bildet daher mit dem Erzeugendensystem AV und den angegebenen Konstruktionsoperationen eine* freie Struktur*, so daß wir im folgenden Funktionen induktiv auf* ausd *definieren können.*

Bei der Schreibweise von Ausdrücken können folgende Klammersparregeln verwendet werden:

1. Außenklammern weglassen.
2. Klammern in mehrfachen Alternativen oder Konjunktionen weglassen.
3. Vorrangregeln
   - $\wedge$ bindet stärker als $\vee$
   - $\vee$ bindet stärker als $\rightarrow$
   - $\rightarrow$ bindet stärker als $\leftrightarrow$

Beispiele

- $p \wedge q \vee r \stackrel{\wedge}{=} (p \wedge q) \vee r$
- $p \leftrightarrow r \rightarrow q \stackrel{\wedge}{=} p \leftrightarrow (r \rightarrow q)$

Satz 2.2.2
Es gibt einen Algorithmus, der für jedes Wort $w \in W(X)$ entscheidet, ob $w \in$ ausd ist.

2.2.2 Semantik

Ein Kalkül κ ist gegeben durch sein Alphabet (hier X), die Menge (Sprache) der in diesem Kalkül sinnvollen Wörter über diesem Alphabet (hier ausd) und seine Semantik, d. h. eine Zuordnung von Bedeutungen zu den Wörtern seiner Sprache. Bei Logikkalkülen besteht die Angabe der Semantik aus der Menge S der (gültigen) Sätze und einem Deduktionsoperator.

Um über Gültigkeit eines Ausdruckes reden zu können, definieren wir zunächst den Wahrheitswert, den ein Ausdruck bei einer Belegung der Aussagenvariablen annimmt.

Belegung
Eine Abbildung $\beta : AV \rightarrow \{\mathrm{W}, \mathrm{F}\}$ der Menge AV der Aussagenvariablen in die Menge $\{\mathrm{W}, \mathrm{F}\}$ der Wahrheitswerte heißt *Belegung*.

Wert eines Ausdrucks
$\mathrm{wert}(H, \beta)$ wird induktiv über den Aufbau von Ausdrücken definiert:

A) $\mathrm{wert}(p_i, \beta) = \beta(p_i)$

I)
- $\mathrm{wert}(\neg H, \beta) = \mathrm{non}(\mathrm{wert}(H, \beta))$
- $\mathrm{wert}((H_1 \wedge H_2), \beta) = \mathrm{et}(\mathrm{wert}(H_1, \beta), \mathrm{wert}(H_2, \beta))$
- $\mathrm{wert}((H_1 \vee H_2), \beta) = \mathrm{vel}(\mathrm{wert}(H_1, \beta), \mathrm{wert}(H_2, \beta))$
- $\mathrm{wert}((H_1 \rightarrow H_2), \beta) = \mathrm{seq}(\mathrm{wert}(H_1, \beta), \mathrm{wert}(H_2, \beta))$
- $\mathrm{wert}((H_1 \leftrightarrow H_2), \beta) = \mathrm{aeq}(\mathrm{wert}(H_1, \beta), \mathrm{wert}(H_2, \beta))$

Lemma
Es sei H ein Ausdruck, in dem genau die Variablen $p_{j_1}, \ldots, p_{j_k}$ vorkommen. β, β' seien Belegungen, die auf $p_{j_1}, \ldots, p_{j_k}$ übereinstimmen. Dann ist $\mathrm{wert}(H, \beta) = \mathrm{wert}(H, \beta')$.

Definition 2.2.3
Es sei $H \in$ ausd, $p_{j_1}, \ldots, p_{j_n}$ $(j_1 < \cdots < j_n)$ alle in H vorkommenden Aussagenvariablen.

$$\Phi_H(w_1, \ldots, w_n) := \mathrm{wert}(H, \beta^\star), \text{ wobei } \beta^\star(p_{j_i}) = w_i \text{ für } 1 \leq i \leq n$$

ist die von H repräsentierte Wahrheitsfunktion.

$\Phi_H(\beta^\star(p_{j_1}), \ldots, \beta^\star(p_{j_n})) = \mathrm{wert}(H, \beta^\star)$ *für alle Belegungen $\beta^\star$.*

Damit haben wir jedem Ausdruck eine Wahrheitsfunktion zugeordnet, die seine Bedeutung ist, denn er repräsentiert diese Wahrheitsfunktion im Kalkül. Es entsteht die Frage, ob alle Wahrheitsfunktionen durch Ausdrücke dargestellt werden können, oder ob wir beim Übergang von Wahrheitsfunktionen zu Ausdrücken wieder etwas verlieren wie beim Übergang von Aussagenfunktionen zu Wahrheitsfunktionen.

Repräsentantentheorem
Zu jeder Wahrheitsfunktion Φ existiert ein Ausdruck H mit $\Phi_H = \Phi$.

Beweis
Sei Φ gegeben durch die Tabelle:

w_1	$\dots$	w_n	$\Phi(w_1, \dots, w_n)$
W	$\dots$	W	$\Phi(\mathrm{W}, \dots, \mathrm{W})$
	$\vdots$		$\vdots$

Eine *Elementarkonjunktion* in $p_1, \dots, p_n$ ist ein Ausdruck H der Form:

$$H : [\neg]\, p_1 \wedge [\neg]\, p_2 \cdots \wedge [\neg]\, p_n$$

Sei H eine Elementarkonjunktion. Dann ist $\operatorname{wert}(H, \beta) = \mathrm{W}$ genau dann, wenn für $i = 1, \dots, n$ gilt:

$$\beta(p_i) = \begin{cases} \mathrm{W}, & \text{falls vor } p_i \text{ kein } \neg \text{ steht} \\ \mathrm{F}, & \text{falls vor } p_i \text{ ein } \neg \text{ steht} \end{cases}$$

Beispiel
$p_1 \wedge \neg p_2 \wedge p_3 : \beta(p_1) = \mathrm{W}$, $\beta(p_2) = \mathrm{F}$, $\beta(p_3) = \mathrm{W}$.

Fall 1: $\Phi =$ falsum, d. h. $\Phi(w_1, \dots, w_n) \equiv \mathrm{F}$, dann sei

$$H := (p_1 \wedge \neg p_1) \vee (p_2 \wedge \neg p_2) \vee \dots \vee (p_n \wedge \neg p_n)$$

also gilt $\Phi_H = \Phi$

Fall 2: $\Phi \neq$ falsum, d. h. es gibt n-Tupel $[w_1, \dots, w_n]$ mit $\Phi(w_1, \dots, w_n) = \mathrm{W}$.

Es seien $\beta_1, \dots, \beta_k$ alle Belegungen β_i der Aussagenvariablen $p_1, \dots, p_n$ so, daß $\Phi(\beta_i(p_1), \dots, \beta_i(p_n)) = \mathrm{W}$ ist. Zu β_i sei K_i die Elementarkonjunktion mit $\operatorname{wert}(K_i, \beta_i) = \mathrm{W}$. Dann leistet

$$H := K_1 \vee K_2 \vee \cdots \vee K_k$$

das Verlangte, d. h. $\Phi_H = \Phi$.

□

2.2.3 Weitere semantische Begriffe

Oben im Abschnitt 2.2.1 haben wir ohne weitere Begründung $\neg$ als Zeichen für non, $\wedge$ als Zeichen für et, ... betrachtet, zuvor haben wir uns auf zwei Wahrheitswerte eingeschränkt. Wir können die Sprache ausd des Aussagenkalküls aber auch in anderer Weise interpretieren, d. h. den Ausdrücken eine andere Bedeutung zuweisen, wenn wir festlegen,

- was die Menge M aller Wahrheitswerte ist,
- welche Wahrheitswerte wir als „gute“ Werte betrachten, diese bilden die Menge $M^\star \subseteq M$,
- welche einstellige Funktion $\phi_1 : M \to M$ das Zeichen $\neg$ bedeutet und
- welche zweistelligen Funktionen $\phi_2, \ldots, \phi_5 : M \times M \to M$ die Zeichen $\wedge$, $\vee$, $\to$ und $\leftrightarrow$ bedeuten sollen.

Ein Tupel $\mu = [M, M^\star, \phi_1, \phi_2, \phi_3, \phi_4, \phi_5]$ dessen Elemente obigen Eigenschaften genügen, also eine Zusammenfassung dieser Angaben, nennen wir eine logische Matrix.

Beispiel 1: Die klassische logische Matrix

$$[\{\mathrm{W}, \mathrm{F}\}, \{\mathrm{W}\}, \mathrm{non}, \mathrm{et}, \mathrm{vel}, \mathrm{seq}, \mathrm{aeq}]$$

Beispiel 2: Die wahrscheinlichkeitslogische Matrix

$$\begin{aligned}
M &:= \text{ das reelle Intervall } [0,1] \\
M^\star &:= \{1\} \\
\phi_1(w) &:= 1 - w \\
\phi_2(w_1, w_2) &:= w_1 \cdot w_2 \\
\phi_3(w_1, w_2) &:= 1 - (1 - w_1) \cdot (1 - w_2) \\
\phi_4(w_1, w_2) &:= \phi_3(1 - w_1, w_2) = 1 - w_1 \cdot (1 - w_2) \\
\phi_5(w_1, w_2) &:= \phi_4(w_1, w_2) \cdot \phi_4(w_2, w_1)
\end{aligned}$$

Entsprechend wie wir früher den Wert eines Ausdrucks H bei einer Belegung der Aussagenvariablen mit Wahrheitswerten definiert haben, können wir jetzt auch allgemein festlegen:

Definition 2.2.4 (Belegung und Wert)
Für eine beliebige logische Matrix μ ist

1. *die* Belegung der Aussagenvariablen *eine Abbildung $\beta : AV \to M$*
   $\mathfrak{B}_\mu = \{\beta | \beta : AV \to M\}$ *ist die Menge aller Belegungen; für die klassische logische Matrix ist* $\mathfrak{B} = \{\beta | \beta : AV \to \{\mathrm{W}, \mathrm{F}\}\}$
2. *der* Wert eines Ausdrucks, *d. h.* $\mathrm{wert}_\mu(H, \beta)$, *analog zu* $\mathrm{wert}(H, \beta)$ *induktiv definiert.*

Beispiel
Ein Teil der induktiven wert–Definition sieht wie folgt aus:

$$\mathrm{wert}_\mu((H_1 \wedge H_2), \beta) = \phi_2(\mathrm{wert}_\mu(H_1, \beta), \mathrm{wert}_\mu(H_2, \beta))$$

Wir führen nun die folgenden semantischen Begriffe für die klassische logische Matrix ein:

Erfüllbarkeit, Gültigkeit
Sei $\beta \in \mathfrak{B}$ und $H \in$ ausd.

1. β *erfüllt* H ($\beta \mathrm{erf} H$) $:\leftrightarrow \mathrm{wert}\,(H, \beta) = \mathrm{W}$

   In der logischen Matrix μ gilt: $\beta \mathrm{erf}_\mu H :\leftrightarrow \mathrm{wert}_\mu\,(H, \beta) \in M^\star$

2. H ist *erfüllbar* ($H \in \mathrm{ef}$, $\mathrm{ef} H$), wenn eine Belegung $\beta \in \mathfrak{B}$ existiert so, daß $\beta \mathrm{erf} H$.

3. H ist *(allgemein–)gültig* ($H \in \mathrm{ag}$, $\mathrm{ag} H$), wenn jede Belegung $\beta \in \mathfrak{B}$ den Ausdruck H erfüllt. (H ist eine Identität, bzw. Tautologie.)

   In der logischen Matrix μ gilt: $H \in \mathrm{ag}_\mu$ gdw. für jede Belegung β gilt $\mathrm{wert}_\mu\,(H, \beta) \in M^\star$

4. H ist *Kontradiktion* ($H \in \mathrm{kt}$, $\mathrm{kt} H$), wenn es keine Belegung β gibt, die H erfüllt.

Folgerung 2.2.5
Für ag, ef *und* erf *gilt:*

1. $\mathrm{ag} \subset \mathrm{ef}$
2. $\mathrm{ag} H$ *gdw. nicht* $\mathrm{ef} \neg H$*;* $\mathrm{ef} H$ *gdw. nicht* $\mathrm{ag} \neg H$
3. *Wenn* $\beta \mathrm{erf} H$*, so nicht* $\beta \mathrm{erf} \neg H$

Außerdem gilt:

Satz 2.2.6 (Ausgeschlossener Widerspruch für die Gültigkeit)
Es gibt keinen Ausdruck H *mit* $\mathrm{ag} H$ *und* $\mathrm{ag} \neg H$.

Satz 2.2.7 (Ausgeschlossener Dritte für die Erfüllbarkeit)
Für jedes $H \in$ ausd *gilt:* $\mathrm{ef} H$ *oder* $\mathrm{ef} \neg H$.

2.3 Anwendung in der Schaltalgebra

Als Schaltalgebra bezeichnet man die Anwendung des Aussagenkalküls auf die Beschreibung und die Berechnung von Schaltungen.

2.3.1 Grundbegriffe

Unter einer Schaltung verstehen wir eine *black box*, die Eingangssignale empfängt und in Ausgangssignale umwandelt, diese Signale haben nur zwei Signalwerte (Impuls/kein Impuls, hohes/niedriges Potential, An/Aus, ...) und können durch Zeitfunktionen (jedem Zeitpunkt wird einer der beiden Signalwerte zugeordnet) beschrieben werden. Als kombinatorische oder *logische Schaltung* bezeichnen wir eine Schaltung mit folgender Eigenschaft:

> Die Schaltung hat kein Gedächtnis, d. h. der aktuelle Wert eines Ausgangssignals hängt höchstens von den *aktuellen* Werten der Eingangssignale ab (nicht aber von zeitlich früheren oder künftigeren Eingabesignalwerten) und ist durch diese Werte eindeutig bestimmt.

Daraus folgt, daß eine Änderung der Eingabesignalwerte verzögerungsfrei zu einer eventuellen Änderung der Ausgabesignale führen muß.

Daher kann eine logische Schaltung S, die n Eingangssignale $x_1(t), \ldots x_n(t)$ in ein Ausgangssignal $y(t)$ transformiert, durch eine n–stellige BOOLEsche Funktion Φ_S charakterisiert werden:

Bezeichnen wir die beiden möglichen Signalwerte mit 0 und 1, so gilt für Φ_S:

$$\Phi_S\left(\mathrm{WW}\left(„x_1(t) = 1“\right), \ldots, \mathrm{WW}\left(„x_n(t) = 1“\right)\right) \quad = \quad \mathrm{WW}\left(„y(t) = 1“\right)$$

Weil S eine logische Schaltung ist, ist die Abhängigkeit von t in dieser Definitionsgleichung nur fiktiv.

Wir wissen bereits, daß jede BOOLEsche Funktion durch einen Ausdruck des Aussagenkalküls repräsentiert wird, wir können also jede logische Schaltung mit einem Ausgang syntaktisch durch den Ausdruck $H_S = H_{\Phi_S}$ beschreiben. Logische Schaltungen mit mehr als einem Ausgang können wir durch eine Menge von Ausdrücken beschreiben, jedem Ausgang entspricht ein Ausdruck. Im folgenden betrachten wir nur noch logische Schaltungen mit genau einem Ausgang.

Definition 2.3.1 (Äquivalente Schaltungen)
Zwei Schaltungen werden äquivalent genannt, wenn sie die gleiche Signalverarbeitung realisieren, d. h. jede von ihnen kann in jeder Umgebung durch die andere ersetzt werden, ohne daß sich das Verhalten des Gesamtsystems ändert.

Satz 2.3.2
Die Äquivalenz von Schaltungen spiegelt sich als semantische Äquivalenz zugehöriger Ausdrücke wider:

$$S_1 \sim S_2 \text{ gdw. } H_{S_1} \sim_{\mathrm{sem}} H_{S_2} \quad (\text{d. h. } (H_1 \leftrightarrow H_2) \in \mathrm{ag})$$

Diese Tatsache ermöglicht es zu unterscheiden, ob Schaltungen äquivalent sind und damit auch, in gewisser Hinsicht *optimale* Schaltungen zu finden.

Wir betrachten einen wichtigen Spezialfall, sogenannte $\wedge\vee-$ Schaltungen. Dabei handelt es sich um logische Schaltungen mit genau einem Ausgang, die

1. aus sogenannten *Gattern* aufgebaut sind, und zwar aus Konjunktions- und Alternativgattern, und
2. die zweistufig sind, wobei jedes Eingangssignal zuerst höchstens ein Konjunktionsgatter und dann höchstens ein Alternativgatter durchläuft.

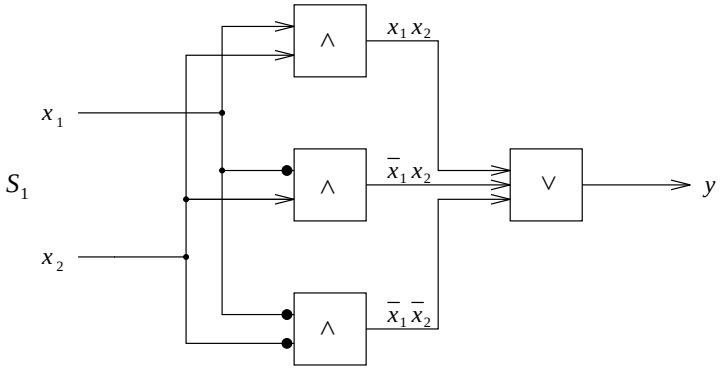
Ein *Konjunktionsgatter* ist eine logische Schaltung, die die Konjunktion der Eingangssignale oder ihrer Negation realisiert; bei n Eingangssignalen gibt es also 2^n Konjunktionsgatter; ein *Alternativgatter* realisiert die Alternative der Eingangssignale, d. h. am Ausgang steht der Signalwert 1 sobald eines der Eingangssignale den Wert 1 annimmt. Auch hier können Eingangssignale negiert sein.

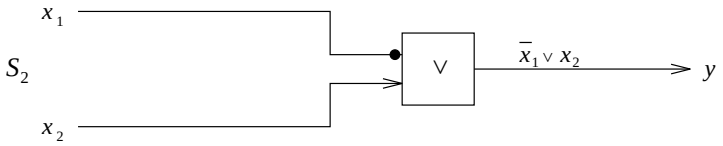
Solche Schaltungen werden strukturell beschrieben durch *Alternative Normalformen*, d. h. Ausdrücke der Form $H = K_1 \vee K_2 \vee \cdots \vee K_m$, wobei die Alternativglieder K_i *Fundamentalkonjunktionen* sind, d. h. Konjunktionen von negierten und nichtnegierten Aussagenvariablen, in denen keine Variable mehrfach vorkommt.

Beispiel

$$\begin{aligned} H_1 &= (p_1 \wedge p_2) \vee (\neg p_1 \wedge p_2) \vee (\neg p_1 \wedge \neg p_2) \\ H_2 &= \neg p_1 \vee p_2 \end{aligned}$$

sind Alternative Normalformen (ANF); H_1 ist die Kanonische Alternative Normalform der Funktion seq, H_2 ist semantisch äquivalent mit H_1, also eine ANF von seq. Die zugehörigen Schaltungen sehen so aus:





Bemerkung
Offenbar ist die Schaltung S_2 „besser“ als S_1: Sie ist übersichtlicher, benötigt weniger Gatter, usw.

Deshalb betrachten wir

2.3.2 Optimierte Schaltungen

Wir wählen als Optimierungsgröße (als „Kosten“) einer Schaltung des betrachteten Typs die Zahl der Eingänge in Konjunktionsgatter (wenn ein Signal vom Eingang direkt in das Alternativgatter geführt wird, durchläuft es sozusagen ein Konjunktionsgatter mit genau einem Eingang).

Beispiel
S_1 hat die Kosten 6 und S_2 die Kosten 2.

Die Kosten eine $\wedge\vee$–Schaltung S spiegeln sich in der zugehörigen ANF als Zahl der Stellen, an denen Variablen stehen wider.

Dementsprechend führen wir ein:

Definition 2.3.3 (Optimale ANF)
Eine ANF H *mit den Variablen* $p_1, \ldots, p_n$ *ist* optimal, *wenn für jede* ANF H', *in der höchstens die Variablen* $p_1, \ldots, p_n$ *vorkommen und die semantisch äquivalent zu* H *ist, gilt, daß die Zahl der Stellen, an denen in* H' *Aussagenvariablen stehen mindestens so groß ist wie in* H.

Folgerung 2.3.4
Die Aufgabe, eine $\wedge\vee$*–Schaltung* S *zu optimieren, bedeutet also eine optimale* ANF H *zu finden mit*

$$H \quad \sim_{\text{sem}} \quad H_{\Phi_S} = H_S$$

Diese Aufgabe ist im Prinzip durch Probieren lösbar, in der Praxis muß man das Probieren gezielt einschränken, aber ohne Probieren geht es i. a. nicht, denn i. a. gibt es viele optimale ANF zu einer gegebenen Schaltung bzw. einer gegebenen BOOLEschen Funktion.

Wir vereinfachen zunächst die Schreibweise von ANF:

- das Negationszeichen $\neg$ wird als Querstrich über die Variable geschrieben
- das Konjunktionszeichen $\wedge$ entfällt

Beispiel
$\neg p \wedge q$ wird als $\overline{p}\,q$ geschrieben

Wir betrachten die folgende Aufgabe:

gegeben: Eine logische Schaltung S mit n Eingängen bzw. eine n–stellige BOOLEsche Funktion Φ

gesucht: Eine optimale ANF $H^\star$ mit

$$H^\star \sim_{\text{sem}} H_S \quad \text{bzw.} \quad H^\star \sim_{\text{sem}} H_\Phi$$

Beispiel
Φ sei gegeben durch die Tabelle

w_1	w_2	w_3	$\Phi(w_1, w_2, w_3)$
W	W	W	W
W	W	F	W
W	F	W	W
W	F	F	F
F	W	W	F
F	W	F	W
F	F	W	W
F	F	F	W

Wir wählen als Variablenmenge $\{p, q, r\}$ und bilden die KANF von Φ:

$$H_1 \quad = \quad p\,q\,r \vee p\,q\overline{r} \vee p\overline{q}\,r \vee \overline{p}\,q\,\overline{r} \vee \overline{p}\,\overline{q}\,r \vee \overline{p}\,\overline{q}\,\overline{r}$$

H_1 hat die Kosten 18; $\Phi_{H_1} = \Phi$

Den Zugang zur Vereinfachung der KANF bildet der Begriff des *Primimplikanden.*

Definition 2.3.5 (Implikand, Primimplikand)
Φ sei n–stellige BOOLE*sche Funktion, H ein Ausdruck mit $\Phi_H = \Phi$*

1. *Eine Fundamentalkonjunktion K in $\{p_1, \ldots, p_n\}$ heißt* Implikand *von Φ (bzw. H), wenn für alle Belegungen β (von $\{p_1, \ldots, p_n\}$) gilt:*

   $$\textit{Wenn } \text{wert}(K, \beta) = \text{W}, \textit{ so } \Phi\left(\beta(p_1), \ldots, \beta(p_n)\right) = \text{W}$$

   (bzw. $\text{wert}(H, \beta) = \text{W}$.

2. *K heißt* Primimplikand *von Φ (bzw. H), wenn*
   (a) *K Implikand von Φ (bzw. H) ist*
   (b) *keine Fundamentalkonjunktion K', die aus K durch Streichen einer Variable (samt Querstrich) entsteht, ein Implikand von Φ ist.*

Folgerung 2.3.6
Aus der Definition ergibt sich sofort:

1. *Alle Elementarkonjunktionen, die in der* KANF *von Φ vorkommen, sind Implikanden von Φ*
2. *K ist ein Implikand von H gdw. $(K \to H) \in \mathrm{ag}$*

In unserem Beispiel ist also $p\,q\,r$ ein Implikand, aber kein Primimplikand, denn durch Streichen der Variablen r entsteht $p\,q$ und $p\,q$ ist ein Implikand von Φ, wie man leicht nachrechnet. Allerdings ist $p\,q$ ein Primimplikand, denn weder p noch q sind Implikanden von Φ.

Satz 2.3.7
$\Phi : \bigtimes_n \{\mathrm{W}, \mathrm{F}\} \to \{\mathrm{W}, \mathrm{F}\}$ *sei nichtkonstant. Dann besitzt Φ eine* ANF*, die nur aus Primimplikanden besteht.*

Beweis
Es sei $H_1 = K_1 \vee \cdots \vee K_l$ eine KANF von Φ $(0 < l < 2^n)$. Wenn alle Elementarkonjunktionen Primimplikanden sind, ist nichts zu zeigen.

O. B. d. A. sei K_1 kein Primimplikand und K_1' sei ein Primimplikand, der aus K_1 durch Streichen von Variablen entsteht. Dann gilt für die Belegung β:

$$\text{Wenn } \mathrm{wert}(K_1, \beta) = \mathrm{W}, \text{ so } \mathrm{wert}(K_1', \beta) = \mathrm{W}.$$

Wir zeigen:

$$K_1' \vee K_2 \vee \cdots \vee K_l \text{ ist ANF von } \Phi,$$

d. h. für alle Belegungen β gilt

$$\Phi\left(\beta(p_1), \ldots, \beta(p_n)\right) = \mathrm{W} \quad \text{gdw.} \quad \mathrm{wert}(K_1' \vee K_2 \vee \cdots \vee K_l, \beta) = \mathrm{W}.$$

($\Leftarrow$): Wenn $\mathrm{wert}(K_1' \vee K_2 \vee \cdots \vee K_l, \beta) = \mathrm{W}$ ist, so ist $\mathrm{wert}(K_1', \beta) = \mathrm{W}$ oder $\mathrm{wert}(K_2, \beta) = \mathrm{W}$ oder … oder $\mathrm{wert}(K_l, \beta) = \mathrm{W}$.

Weil alle diese Konjunktionen Implikanden von Φ sind, ist

$$\Phi\left(\beta(p_1), \ldots, \beta(p_n)\right) = \mathrm{W}.$$

($\Rightarrow$): Ist $\Phi\left(\beta(p_1), \ldots, \beta(p_n)\right) = \mathrm{W}$, so ist $\mathrm{wert}(K_1 \vee K_2 \vee \cdots \vee K_l, \beta) = \mathrm{W}$, es ist also $\mathrm{wert}(K_1, \beta) = \mathrm{W}$ oder … oder $\mathrm{wert}(K_l, \beta) = \mathrm{W}$.

Folglich ist $\mathrm{wert}(K_1', \beta) = \mathrm{W}$ oder $\mathrm{wert}(K_2 \vee \cdots \vee K_l, \beta) = \mathrm{W}$, d. h.

$$\mathrm{wert}(K_1' \vee K_2 \vee \cdots \vee K_l, \beta) = \mathrm{W}.$$

Der Ausdruck $H_1' = K_1' \vee K_2 \vee \cdots \vee K_l$ enthält also mindestens einen Primimplikanden mehr als H_1, wenn H_1' noch Implikanden enthält, die nicht Primimplikanden sind, so setzen wir das Verfahren fort. $\square$

Folgerung 2.3.8
Jede optimale Normalform von Φ besteht nur aus Primimplikanden von Φ.

In unserem Beispiel sind folgende Fundamentalkonjunktionen Primimplikanden:

$$p\,q, p\,r, \overline{p}\,\overline{q}, q\,\overline{r}, \overline{q}\,r, \overline{p}\,\overline{r}$$

Es sei H_2 die ANF, die aus allen Primimplikanden von Φ besteht. Dann gilt nach Konstruktion

$$\Phi \quad = \quad \Phi_{H_2},$$

d. h. H_2 ist eine ANF von Φ.

Im allgemeinen ist aber H_2 nicht optimal. Weil jede optimale ANF von Φ nur aus Primimplikanden besteht, kann jede optimale ANF von Φ aus H_2 durch Streichen von Konjunktionen gewonnen werden.

Wir geben zunächst ein Kriterium an, wann ein Primimplikand <u>nicht</u> gestrichen werden darf, d. h. wann er in allen optimalen ANF vorkommt.

Definition 2.3.9 (Wesentlicher Primimplikand)
Es sei $\mathfrak{M}$ eine Menge von Primimplikanden von Φ und $K \in \mathfrak{M}$. K heißt wesentlich *in bezug auf $\mathfrak{M}$, wenn es eine Belegung β gibt mit*

1. $\text{wert}(K, \beta) = \text{W}$
2. *für alle $K' \in \mathfrak{M}$ mit $K' \neq K$ ist* $\text{wert}(K', \beta) = \text{F}$.

Satz 2.3.10
Es sei $\mathfrak{M}$ die Menge aller Primimplikanden von Φ und K sei wesentlich in bezug auf $\mathfrak{M}$. Dann kommt K in jeder optimalen ANF *von Φ vor.*

Beweis
Es sei $H = K_1 \vee \cdots \vee K_m$ eine beliebige optimale ANF von Φ und β eine Belegung von $p_1, \ldots p_n$ mit

1. $\text{wert}(K, \beta) = \text{W}$
2. $\text{wert}(K', \beta) = \text{F}$ für alle $K' \in \mathfrak{M}$ mit $K' \neq K$.

Weil K Implikand von Φ ist, folgt aus $\text{wert}(K, \beta) = \text{W}$

$$\text{W} = \Phi\left(\beta(p_1), \ldots, \beta(p_n)\right) = \text{wert}(K_1 \vee \cdots \vee K_m, \beta) = \text{W}$$

d. h. für wenigstens ein i ist $\text{wert}(K_i, \beta) = \text{W}$.

Aus $K_i \in \mathfrak{M}$ und (2) folgt $K_i = K$, d. h. K kommt in H vor. □

Folgerung 2.3.11 (Optimale ANF)
Es sei $\mathfrak{M}$ die Menge aller Primimplikanden von Φ und $\mathfrak{N}$ die Menge der in bezug auf $\mathfrak{M}$ wesentlichen Primimplikanden von Φ. Wenn $\mathfrak{N} \neq \emptyset$ ist, $H^\star$ die Alternative der Elemente von $\mathfrak{N}$ bezeichnet und $\Phi_{H^\star} = \Phi$ ist, dann ist $H^\star$ die optimale ANF *von Φ.*

In unserem Beispiel ist die Menge $\mathfrak{N}$ der wesentlichen Primimplikanden leer, wir kommen daher nur durch Probieren weiter, d. h. wir lassen „versuchsweise“ einen Primimplikanden weg:

$$\mathfrak{M} = \{p\,q, p\,r, \overline{p}\,\overline{q}, q\,\overline{r}, \overline{q}\,r, \overline{p}\,\overline{r}\} \qquad \mathfrak{N} = \emptyset$$

Wir streichen $p\,q$:

$$\mathfrak{M}' = \{p\,r, \overline{p}\,\overline{q}, q\,\overline{r}, \overline{q}\,r, \overline{p}\,\overline{r}\}$$

In bezug auf $\mathfrak{M}'$ sind $p\,r$ und $q\,\overline{r}$ wesentlich:

	p	q	r	$p\,r$	$\overline{p}\,\overline{q}$	$q\,\overline{r}$	$\overline{q}\,r$	$\overline{p}\,\overline{r}$
β_1	W	W	W	W	F	F	F	F
β_2	W	W	F	F	F	W	F	F

also $\mathfrak{N}' = \{p\,r, q\,\overline{r}\}$

Wir streichen $\overline{q}\,r$:

$$\mathfrak{M}'' = \{p\,r, \overline{p}\,\overline{q}, q\,\overline{r}, \overline{p}\,\overline{r}\}$$

Weil $p\,r$, $q\,\overline{r}$ wesentlich in bezug auf $\mathfrak{M}'$ sind, sind sie es auch in bezug auf $\mathfrak{M}'' \subset \mathfrak{M}'$.

In bezug auf $\mathfrak{M}''$ ist aber auch $\overline{p}\,\overline{q}$ wesentlich:

	p	q	r	$p\,r$	$\overline{p}\,\overline{q}$	$q\,\overline{r}$	$\overline{p}\,\overline{r}$
β_3	F	F	W	F	W	F	F

während $\overline{p}\,\overline{r}$ nicht wesentlich in bezug auf $\mathfrak{M}''$ ist. Durch Streichen von $\overline{p}\,\overline{r}$ erhalten wir die Menge

$$\mathfrak{M}''' = \{p\,r, \overline{p}\,\overline{q}, q\,\overline{r}\} = \mathfrak{N}'''$$

In bezug auf $\mathfrak{M}'''$ sind alle ihre Elemente wesentlich.

Man überlegt sich, daß

$$H^\star = pr \vee \overline{p}\,\overline{q} \vee q\,\overline{r}$$

eine *optimale* ANF von Φ ist (weil jeder Primimplikand für genau 2 Belegungen den Wert W liefert und Φ 6 W–Stellen hat).

Leider kann das angegebene Probeverfahren auch in eine Sackgasse führen, d. h. zu einer Menge von wesentlichen Primimplikanden, die keine optimale ANF bilden.

Wenn wir z. B. im zweiten Schritt $\overline{p}\,\overline{q}$ streichen:

$$\mathfrak{M}''_1 = \{p\,r, q\,\overline{r}, \overline{q}\,r, \overline{p}\,\overline{r}\}$$

so bleiben $p\,r$, $q\,\overline{r}$ wesentlich (in bezug auf $\mathfrak{M}''_1$) und $\overline{q}\,r$, $\overline{p}\,\overline{r}$ sind wesentlich in bezug auf $\mathfrak{M}''_1$:

	p	q	r	$p\,r$	$q\,\overline{r}$	$\overline{q}\,r$	$\overline{p}\,\overline{r}$
β_4	F	F	W	F	F	W	F
β_5	F	F	F	F	F	F	W

Wir erhalten die ANF

$$H^{\star\star} = p\,r \vee q\,\overline{r} \vee \overline{q}\,r \vee \overline{p}\,\overline{r}$$

mit größeren Kosten als $H^\star$, obwohl in $H^{\star\star}$ kein Primimplikand mehr gestrichen werden kann.

2.4 Folgern

Das logische Folgern ist eine Beziehung zwischen Mengen von Aussagen einerseits und Aussagen andererseits. Wann sagen wir, daß eine Aussage logisch aus einer Menge von Aussagen folgt? Sicher dann, wenn in jeder denkbaren Situation, in der alle Aussagen der betrachteten Menge von Aussagen wahr sind (wir nennen eine solche Situation ein Modell dieser Menge) die betreffende Aussage ebenfalls wahr ist; sicher dann nicht, wenn eine Situation vorliegt (oder denkbar ist), bei der zwar alle Aussagen in der Menge wahr, die betreffende Aussage aber falsch ist.

2.4.1 Grundbegriffe

Wir definieren:

Modell, Folgern und Folgerungsoperator
Es sei $X \subseteq \text{ausd}$

- Eine Belegung β heißt *Modell* von $X \subseteq \text{ausd}$ ($\beta \text{Mod} X$), wenn für alle $H \in X$ gilt: $\beta \text{erf} H$
- Aus X *folgt* H ($X \text{fol} H$), wenn jedes Modell β von X den Ausdruck H erfüllt.
- Der *Folgerungsoperator* fl ist wie folgt definiert: $\text{fl}(X) := \{H | X \text{fol} H\}$

Beispiel
Sei p die Aussage „Es liegt Schnee“ und q die Aussage „Es ist kalt“. Die Belegung β: p=W, q=W ist Modell der Ausdrucksmenge $X_1 = \{p, \neg p \vee q, p \rightarrow q\}$; zur Ausdrucksmenge $X_2 = \{p, \neg p\}$ gibt es dagegen kein Modell. Aus der Ausdrucksmenge X_1 folgt die Aussage q; aus der widersprüchlichen Ausdrucksmenge X_2 folgt dagegen die Menge ausd aller Ausdrücke, d. h. $\text{fl}(X_2) = \text{ausd}$.

Satz 2.4.1 (Eigenschaften des Folgerungsoperators)
fl *hat folgende Eigenschaften:*

1. $\text{fl}(\emptyset) = \text{ag}$.
2. $\text{fl}(\text{ag}) \subseteq \text{ag}$.
3. *Hülleneigenschaften von* fl*:*
   (a) $X \subseteq \text{fl}(X)$
   (b) *Wenn* $X \subseteq Y$*, so* $\text{fl}(X) \subseteq \text{fl}(Y)$.
   (c) $\text{fl}(\text{fl}(X)) = \text{fl}(X)$.
4. *Endlichkeitssatz für das Folgern:*

   Wenn $H \in \text{fl}(X)$*, so gibt es eine endliche Teilmenge* $X^\star \subseteq X$ *mit* $H \in \text{fl}(X^\star)$

Beweis von $\text{fl}(\text{ag}) \subseteq \text{ag}$:
$\text{fl}(\text{ag}) = \text{fl}(\text{fl}(\emptyset)) = \text{fl}(\emptyset) = \text{ag}$

Übung 17
Man zeige, daß $\text{fl}(\emptyset) = \text{ag}$ und $\text{fl}(\{p, \neg p\}) = \text{ausd}$ ist und beweise, daß der Folgerungsoperator fl die Hülleneigenschaften besitzt. Außerdem vollziehe man folgenden Beweis nach:

2.4.2 Beweis des Endlichkeitssatzes für das Folgern

Wir benutzen zwei Hilfssätze:

Hilfssatz 1
Voraussetzung: *Jede endliche Teilmenge $X^\star \subseteq X$ besitzt ein Modell.*
Behauptung: *Zu jeder Aussagenvariable p_i gibt es einen Wahrheitswert w_i derart, daß jede endliche Teilmenge $X^\star$ von X ein Modell $\beta_i^\star$ mit $\beta_i^\star(p_i) = w_i$ hat.*

Beweis des ersten Hilfssatzes
Es sei $p_i \in$ AV fixiert. Für jede endliche Teilmenge $X^\star$ von X kann einer von vier Fällen eintreten:

Fall 1: Es gibt ein Modell $\beta^\star$ von $X^\star$ mit $\beta^\star(p_i) = \mathrm{W}$ und es gibt ein Modell $\beta^{\star\star}$ von $X^\star$ mit $\beta^{\star\star}(p_i) = \mathrm{F}$.

Fall 2: Es gibt ein Modell $\beta^\star$ von $X^\star$ mit $\beta^\star(p_i) = \mathrm{W}$ und es gibt kein Modell $\beta^{\star\star}$ von $X^\star$ mit $\beta^{\star\star}(p_i) = \mathrm{F}$

Fall 3: Kein $\beta^\star \ldots$ und ein $\beta^{\star\star} \ldots$

Fall 4: Kein $\beta^\star \ldots$ und kein $\beta^{\star\star} \ldots$

Wir überlegen uns:

- Der Fall 4 kann nicht eintreten, da $X^\star$ ein Modell β besitzt. Wenn $\beta(p_i) = \mathrm{W}$, so existiert also ein Modell $\beta^\star = \beta$ von $X^\star$ mit $\beta^\star(p_i) = \mathrm{W}$; wenn $\beta(p_i) = \mathrm{F}$, so existiert ein Modell $\beta^{\star\star} = \beta$ von $X^\star$ mit $\beta^{\star\star}(p_i) = \mathrm{F}$.
- Es kann nicht passieren, daß für ein $X_1^\star \subseteq X$ Fall 2 und für ein $X_2^\star \subseteq X$ der Fall 3 eintritt. Begründung: $X_1^\star \cup X_2^\star \subseteq X$ ist endlich und besitzt folglich ein Modell β. β ist also zugleich Modell von $X_1^\star$ und $X_2^\star$. Weil für $X_1^\star$ der Fall 2 eintritt und β Modell von $X_1^\star$ ist, gilt $\beta(p_i) = \mathrm{W}$. Weil für $X_2^\star$ der Fall 3 eintritt, hat $X_2^\star$ kein Modell $\beta^\star$ mit $\beta^\star(p_i) = \mathrm{W}$, folglich wäre $\beta(p_i) = \mathrm{F}$, also folgt ein Widerspruch.

Also ergeben sich nur noch zwei Möglichkeiten:

1. Möglichkeit Für alle endlichen Teilmengen $X^\star$ von X tritt Fall 1 oder 2 ein. Dann setzen wir $w_i := \mathrm{W}$.

2. Möglichkeit Für alle endlichen Teilmengen $X^\star$ von X tritt Fall 1 oder 3 ein. Dann setzen wir $w_i := \mathrm{F}$.

Da w_i das Verlangte leistet, haben wir damit den ersten Hilfssatz bewiesen. □

Diesen ersten Hilfssatz benötigen wir nur für den Beweis des folgenden

Hilfssatz 2 (Endlichkeitssatz für Modelle)
Besitzt jede endliche Teilmenge $X^\star$ von X ein Modell, so besitzt X ein Modell.

Beweis des Endlichkeitssatzes für Modelle
Wir vervollständigen die Menge X unter Erhalt der Eigenschaft, daß jede endliche Teilmenge ein Modell besitzt, zu einer Menge Y, von der wir ein Modell ablesen können: Wenn eine Aussagenvariable p_i als Ausdruck in Y vorkommt, dann gilt für jedes Modell β von Y stets $\beta(p_i) = \mathrm{W}$, weil β den Ausdruck $p_i \in Y$ erfüllt, wenn dagegen $p_i \notin Y$ ist, so gilt $\beta(p_i) = \mathrm{F}$ für jedes Modell von Y.

Wir gehen induktiv vor:

Anfang: $X_0 := X$

Schritt:

$$X_{n+1} = \begin{cases} X_n \cup \{p_n\}, & \text{falls jede endliche Teilmenge } X^\star \subseteq X \\ & \text{ein Modell } \beta^\star \text{ mit } \beta^\star(p_n) = W \text{ hat} \\ X_n \cup \{\neg p_n\}, & \text{sonst} \end{cases}$$

Wir setzen nun

$$Y = \bigcup_{n=0}^{\infty} X_n$$

Behauptung 1: Für alle $n \in \mathbb{N}$ gilt: Jede endliche Teilmenge von X_n hat ein Modell.

Beweis
durch Induktion über n.

Anfang: $n = 0$ — trivial: Jede endliche Teilmenge von $X_0 = X$ hat ein Modell (nach Vor.)

Schritt: $n \to n+1$
Nach Hilfssatz 1 existiert zu p_n ein w_n so, daß jede endliche Teilmenge $X^\star \subseteq X_n$ ein Modell $\beta^\star$ mit $\beta^\star(p_n) = w_n$ hat.
Nun machen wir für w_n eine Fallunterscheidung:

Fall 1: $w_n = \mathrm{W}$
Dann hat jede endliche Teilmenge $X^\star \subseteq X_n$ ein Modell $\beta^\star$ mit $\beta^\star(p_n) = \mathrm{W}$

$$X_{n+1} = X_n \cup \{p_n\}$$

Sei $X^\star$ endliche Teilmenge von X_{n+1}, dann hat $X^\star$ ein Modell.

Fall 2: $w_n = \mathrm{F}$
$X_{n+1} := X_n \cup \{\neg p_n\}$ usw. (analog Fall 1)

□

Behauptung 2: Jede endliche Teilmenge $Y^\star$ von Y besitzt ein Modell.

Beweis
Weil $Y^\star$ endlich ist, existiert ein m mit $Y^\star \subseteq X_m$

$$m = \max\{i | p_i \in Y^\star \text{ oder } \neg p_i \in Y^\star\},$$

also hat $Y^\star$ nach Behauptung 1 ein Modell. □

Wir betrachten die Belegung $\bar{\beta} : \mathrm{AV} \to \{\mathrm{W}, \mathrm{F}\}$ mit

$$\bar{\beta}(p_i) = \begin{cases} \mathrm{W}, & p_i \in Y \\ \mathrm{F}, & \neg p_i \in Y \end{cases}$$

Behauptung 3: $\bar{\beta}\mathrm{Mod}X$, d. h. wenn $H \in X$, so $\bar{\beta}\mathrm{erf}H$

Beweis
Es sei $H \in X, p_{j_1}, \ldots, p_{j_n}$ alle Aussagenvariablen von H.

$$Y^\star = \{H\} \cup \begin{cases} \{p_{j_v}\}, & p_{j_v} \in Y \\ \{\neg p_{j_v}\}, & \neg p_{j_v} \in Y \end{cases} \quad v = 1, \ldots, n$$

Es ist $\mathrm{card}(Y^\star) \leq n+1$, d. h. $Y^\star \subseteq Y$ ist endlich, besitzt also nach Behauptung 2 ein Modell $\beta^\star$.

Dabei gilt: für $v = 1, \ldots, n$

$$\beta^\star(p_{j_v}) = \left\{ \begin{array}{ll} \text{W, falls} & p_{j_v} \in Y \\ \text{F, falls} & \neg p_{j_v} \in Y \end{array} \right\} = \bar{\beta}(p_{j_v})$$

Nun ist

$$\text{wert}(H, \beta^\star) = \text{wert}\left(H, \bar{\beta}\right) = \text{W}$$

also $\bar{\beta}\text{erf}H$ nach Lemma auf Seite 53. □

Damit haben wir den Endlichkeitssatz für Modelle bewiesen. □

Nun kommen wir zum eigentlichen

Beweis des Endlichkeitssatzes für das Folgern
Es sei $H \in \text{fl}(X)$, d. h.

jedes Modell β von X erfüllt den Ausdruck H.

Wir nehmen an, daß keine endliche Teilmenge $X^\star \subseteq X$ mit $H \in \text{fl}(X^\star)$ existiert, d. h. für jede endliche Teilmenge $X^\star \subseteq X$ gilt:

Es gibt ein Modell $\beta^\star$ von $X^\star$ mit $\beta^\star\text{erf}\neg H$

d. h.

Es gibt ein Modell von $X^\star \cup \{\neg H\}$.

Wir betrachten nun $Y := X \cup \{\neg H\}$.

Ist $Y^\star$ eine endliche Teilmenge von Y, so ist $Y^\star \setminus \{\neg H\}$ eine endliche Teilmenge von X, es gibt also ein Modell $\beta^\star$ von $(Y^\star \setminus \{\neg H\}) \cup \{\neg H\}$, d. h. ein Modell von $Y^\star$. Nach dem Endlichkeitssatz für Modelle (Hilfssatz 2) hat daher $Y = X \cup \{\neg H\}$ ein Modell.

Das steht im Widerspruch dazu, daß jedes Modell von X, also auch die Modelle von $X \cup \{\neg H\}$, den Ausdruck H erfüllen.

Hiermit haben wir den Endlichkeitssatz für das Folgern vollständig bewiesen. □

2.4.3 Ableitbarkeits- und Deduktionstheorem

Als weitere wichtige Eigenschaft des Folgerns zeigen wir, daß

das **Ableitbarkeitstheorem**

$$\text{Wenn } (H^\star \to H) \in \text{fl}(X), \text{ so } H \in \text{fl}(X \cup \{H^\star\})$$

und das **Deduktionstheorem**

$$\text{Wenn } H \in \text{fl}(X \cup \{H^\star\}), \text{ so } (H^\star \to H) \in \text{fl}(X)$$

gelten.

Das Ableitbarkeitstheorem sagt aus, daß die Abtrennungsregel für das Folgern gilt: Wenn $(H^\star \to H)$ aus einer Menge X folgt und $H^\star$ als zusätzliche Voraussetzung hinzugenommen wird, so folgt H.

Beweis des Ableitbarkeitstheorems
Vor.: Jedes Modell von X erfüllt $(H^\star \to H)$

Beh.: Jedes Modell von $X \cup \{H^\star\}$ erfüllt H.

Sei $\beta\mathrm{Mod}\,(X \cup \{H^\star\})$. $\beta\mathrm{Mod}X$, also $\beta\mathrm{erf}\,(H^\star \to H)$ nach Voraussetzung. Es gilt

$$\mathrm{W} = \mathrm{wert}\,(H^\star \to H, \beta) = \mathrm{seq}\big(\underbrace{\mathrm{wert}\,(H^\star, \beta)}_{\mathrm{W}}, \mathrm{wert}\,(H, \beta)\big)$$

also $\mathrm{wert}\,(H, \beta) = \mathrm{W}$, d. h. $\beta\mathrm{erf}H$. □

Beweis des Deduktionstheorems

Vor.: Jedes Modell von $X \cup \{H^\star\}$ erfüllt H.

Beh.: Jedes Modell von X erfüllt $(H^\star \to H)$.

Sei $\beta\mathrm{Mod}X$.

Fall 1: $\beta\mathrm{erf}\neg H^\star$, d. h. $\mathrm{wert}\,(H^\star, \beta) = \mathrm{F}$.
Dann ist $\mathrm{wert}\,((H^\star \to H)\,, \beta) = \mathrm{W}$, d. h. $\beta\mathrm{erf}\,(H^\star \to H)$

Fall 2: $\beta\mathrm{erf}H^\star$
Dann ist $\beta\mathrm{Mod}\,(X \cup \{H^\star\})$. Folglich $\beta\mathrm{erf}H$, also $\beta\mathrm{erf}\,(H^\star \to H)$.

Wir erhalten also in beiden Fällen $\beta\mathrm{erf}\,(H^\star \to H)$. □

2.4.4 Der Kalkülbegriff

Unter einem Kalkül verstehen wir ein formales System, sprich ein mathematisches Modell, in dem wir rechnen können. Als mathematisches Modell ist ein Kalkül gegeben durch ein Alphabet und eine Sprache über diesem Alphabet, sowie einem Deduktionsoperator (einer Rechenoperation). Bei Logikkalkülen hebt man die Menge der Theoreme (die wahren bzw. beweisbaren) Aussagen hervor.

Kalkül
$\mathfrak{K} = [X, A, S, F]$ heißt Kalkül, wenn

1. X ist ein endliches Alphabet.
2. $A \subseteq W\,(X)$ ist eine Sprache über X.
3. $S \subseteq A$ ist eine Satzmenge (Menge der Theoreme)
4. $F : \mathfrak{P}\,(A) \to \mathfrak{P}\,(A)$ (Deduktionsoperator des Kalküls) mit
   - Hülleneigenschaften
   - Endlichkeitssatz
   - $F\,(S) \subseteq S$

Folgerung 2.4.2
$AK = [X, \mathrm{ausd}, \mathrm{ag}, \mathrm{fl}]$ *ist ein Kalkül* (Aussagenkalkül).

Man unterscheidet:

- Kalkül mit semantisch definierter Satzmenge.

ag = Menge der in der klassischen logischen Matrix gültigen Ausdrücke ($\mathrm{fl}(\emptyset)$)

- Kalkül mit syntaktisch definierter Satzmenge.

  S_0 = Menge aller mittels rein syntaktischer Unformungsoperationen aus einer endlichen Menge von Ausdrücken zu gewinnender Ausdrücke.

Beispiel

Abtrennungsregel $\dfrac{H, (H \to H^\star)}{H^\star}$ **Einsetzungsregel** $\dfrac{H(p_i), H^\star}{H\left(p_i/_{H^\star}\right)}$

Definition 2.4.3 (Entscheidbarkeit eines Kalküls)
Ein Kalkül heißt entscheidbar*, wenn es einen Algorithmus gibt, der für jedes $H \in A$ entscheidet, ob $H \in S$ ist.*

Satz 2.4.4
Der Aussagenkalkül ist entscheidbar.

Beweis
O. B. d. A. seien $p_1, \ldots p_n$ alle in $H \in$ ausd vorkommenden Aussagenvariablen.

Es gilt:

$$H \in \text{ag gdw. für alle } \beta \in \mathfrak{B}\text{: wert}(H, \beta) = \mathrm{W}$$

Dabei ist wert nur von $\beta(p_1), \ldots, \beta(p_n)$ abhängig.

Also ist $H \in$ ag gdw. für jede Belegung β von $\{p_1, \ldots p_n\}$ gilt

$$\text{wert}(H, \beta) = \mathrm{W}$$

Es müssen also „nur“ 2^n Belegungen geprüft werden, um die Allgemeingültigkeit eines Ausdrucks zu entscheiden. □

2.5 Aussagenlogisches Schließen

2.5.1 Grundbegriffe

Wozu ist die Menge ag gut?
Es sei $H \in$ ag; $p_1, \ldots, p_n$ alle Aussagenvariablen in H, dann ist $\Phi_H(w_1, \ldots, w_n) = \mathrm{W} = \text{wert}(H, \beta)$ mit $\beta(p_i) = w_i$. $H \in$ ag *spiegelt* eine verum-Funktion, d. h. *spiegelt* eine extensionale Aussagenfunktion, die stets eine wahre Aussage als Funktionswert hat, eine sogenannte Tautologie.

Was ist eine Schlußregel?
„Wenn $A_1, \ldots, A_n$ gilt, so gilt auch $B_1, \ldots, B_m$“. Oder „Wenn $A_1, \ldots, A_n$, so $\alpha(A_1, \ldots, A_n)$“. Eine Schlußregel ist eine Tautologie, wobei die Implikation Hauptverknüpfung ist.

Wie kann man feststellen, ob eine Schlußregel gültig ist?
Wenn die Schlußregel die allgemeine Form

$$\text{wenn } A_1, \ldots, A_n \text{ so } \alpha(A_1, \ldots, A_n) \qquad (\star)$$

hat und α eine extensionale Aussagenfunktion ist, dann können wir zu der zu α gehörigen Wahrheitsfunktion Φ einen repräsentierenden Ausdruck H des Aussagenkalküls finden. Wir nehmen an, daß dieser Ausdruck genau die Variablen $p_1, \ldots, p_n$ enthält und betrachten den Ausdruck

$$H^\star = p_1 \wedge p_2 \wedge \cdots \wedge p_n \to H.$$

Wenn $H^\star \in \mathrm{ag}$ ist, dann ist die Schlußregel $(\star)$ gültig, und umgekehrt. Wir können sagen, daß die Allgemeingültigkeit von $H^\star$ die Schlußregel $(\star)$ rechtfertigt.

2.5.2 Nützliche aussagenlogische Schlußweisen

Im folgenden geben wir einige wichtige aussagenlogische Schlußregeln an. Ist dabei der „Schluß"–Strich fett gedruckt, dann gilt die Regel auch in die umgekehrte Richtung.

1. Konjunktionsregel

$$\frac{\begin{array}{c}\text{Wenn } A\text{, so } B\\ \text{Wenn } A\text{, so } C\end{array}}{\text{Wenn } A\text{, so } (B \text{ und } C)}$$

Rechtfertigung: $(p \to q) \wedge (p \to r) \leftrightarrow (p \to q \wedge r) \in \mathrm{ag}$

2. Alternative

(a) Schluß auf eine Alternative

$$\frac{\text{Wenn } A\text{, so } B}{\text{Wenn } A\text{, so } (B \text{ oder } C)}$$

Rechtfertigung: $(p \to q) \to (p \to q \vee r) \in \mathrm{ag}$

(b) Schluß aus einer Alternative

$$\frac{\begin{array}{c}\text{Wenn } A\text{, so } B \text{ oder } C\\ \text{Wenn } A \text{ und } B\text{, so } D\\ \text{Wenn } A \text{ und } C\text{, so } D\end{array}}{\text{Wenn } A\text{, so } D}$$

Rechtfertigung:
$((p \to q \vee r) \wedge (p \wedge q \to s) \wedge (p \wedge r \to s)) \to (p \to s) \in \mathrm{ag}$

3. Implikation

(a)

$$\frac{\text{Wenn } A \text{ und } B\text{, so } C}{\text{Wenn } A\text{, so (wenn } B\text{, so } C)}$$

Rechtfertigung: $(p \wedge q \to r) \to (p \to (q \to r)) \in \mathrm{ag}$

(b) Fregescher Kettenschluß

$$\frac{\begin{array}{c}\text{Wenn } A\text{, so (wenn } B\text{, so } C)\\ \text{Wenn } A\text{, so } B\end{array}}{\text{Wenn } A\text{, so } C}$$

Rechtfertigung: $(p \to (q \to r)) \wedge (p \to q) \to (p \to r) \in \text{ag}$

(c) Kettenschluß

$$\frac{\begin{array}{c}\text{Wenn } A\text{, so } B\\ \text{Wenn } B\text{, so } C\end{array}}{\text{Wenn } A\text{, so } C}$$

Rechtfertigung: $(p \to q) \wedge (q \to r) \to (p \to r) \in \text{ag}$

(d) Abtrennungsregel (modus ponens)

$$\frac{\begin{array}{c}A\\ \text{Wenn } A\text{, so } B\end{array}}{B}$$

Rechtfertigung: $(p \wedge (p \to q)) \to q \in \text{ag}$

4. Äquivalenz

$$\frac{\begin{array}{c}\text{Wenn } A\text{, so (wenn } B\text{, so } C)\\ \text{Wenn } A\text{, so (wenn } C\text{, so } B)\end{array}}{\text{Wenn } A\text{, so } (B \text{ gdw. } C)}$$

Rechtfertigung:$(p \to (q \to r)) \wedge (p \to (r \to q)) \leftrightarrow (p \to (q \leftrightarrow r)) \in \text{ag}$

5. Negation

(a) Kontraposition

$$\frac{\text{Wenn } A\text{, so } B}{\text{Wenn nicht } B\text{, so nicht } A.}$$

Rechtfertigung: $(p \to q) \leftrightarrow (\neg q \to \neg p) \in \text{ag}$

(b) Doppelte Negation

$$\frac{A}{\text{nicht nicht } A}$$

Rechtfertigung: $p \leftrightarrow \neg\neg p \in \text{ag}$

(c)

$$\frac{\begin{array}{c}\text{Wenn } A \text{ und } B\text{, so } C\\ \text{Wenn } A \text{ und } B\text{, so nicht } C\end{array}}{\text{Wenn } A\text{, so nicht } B}$$

Rechtfertigung: $(((p \wedge q) \to r) \wedge ((p \wedge q) \to \neg r)) \to (p \to \neg q) \in \text{ag}$

2.5.3 Das Haubersche Theorem

$n \geq 2$

$$\begin{array}{c}
\text{Wenn } A_1\text{, so } B_1 \\
\vdots \\
\text{Wenn } A_n\text{, so } B_n \\
\\
A_1 \text{ oder } A_2 \text{ oder} \ldots \text{oder } A_n \text{ (Die Voraussetz. } A_i \text{ erschöpfen alle Fälle).} \\
\\
\text{nicht } (B_1 \text{ und } B_2)\text{, nicht } (B_2 \text{ und } B_3)\text{, } \ldots \\
\text{nicht } (B_1 \text{ und } B_3)\text{, } \ldots \\
\vdots \\
\text{nicht } (B_1 \text{ und } B_n)\text{. (Die Behauptungen } B_i \text{ schließen sich paarweise aus)} \\
\hline
A_1 \text{ gdw. } B_1 \\
\vdots \\
A_n \text{ gdw. } B_n
\end{array}$$

Beweis (indirekt)
Annahme: Es gibt j so, daß A_j gdw. B_j falsch.

Wenn A_j, so B_j ist wahr nach Voraussetzung, d. h. wenn B_j, so A_j ist falsch, d. h. B_j ist wahr und A_j falsch. Es gibt ein $i \neq j$ so, daß A_i wahr ist.

Wenn A_i, so B_i ist wahr, also B_i ist wahr und $i \neq j$ $\lightning$ nicht (B_i und B_j) □

2.6 Ableitbarkeit

Das Ziel dieses Abschnittes besteht darin, die Folgerungsrelation $X \operatorname{fol} H$ *syntaktisch* zu charakterisieren, d. h. wir suchen nach einer Möglichkeit, die Berechnung der Modelle von X zu vermeiden, um danach festzustellen, ob alle diese Modelle den Ausdruck H erfüllen.

Syntaktisch zu charakterisieren bedeutet, daß wir die Menge X und den Ausdruck H rein strukturell bearbeiten (mit Regeln wie zum Beispiel der Abtrennungsregel), ohne uns dabei um Erfüllbarkeit, Gültigkeit oder Modelle (also die gesamte *semantische* Begriffswelt) zu kümmern.

Ziel ist also eine Gleichung der Form

$$\mathrm{fl}(X) \quad = \quad \text{„syntaktisch definierte Hüllenoperation“}(X).$$

2.6.1 Ableiten mit Abtrennungs- und Einsetzungsregel

Auf Seite 68 hatten wir als Beispiel für syntaktische Umformungsoperationen die Abtrennungs– und Einsetzungsregel kennengelernt. Ausgehend von diesen Regeln wollen wir jetzt entsprechende Korrespondenzen und syntaktische Operationen definieren:

Ableiten mit Abtrennungsregel
Aus Ausdrücken der Form H,$(H \to H^\star)$ ist $H^\star$ ableitbar.
abla
Sei $X \subseteq$ ausd, $H \in$ ausd. Die Korrespondenz $X \operatorname{abla} H$ wird induktiv definiert:

A) Wenn $H \in X$, so $X \operatorname{abla} H$

I) Wenn $X \operatorname{abla} (H \to H^\star)$ und $X \operatorname{abla} H$, so $X \operatorname{abla} H^\star$

aba
Die Operation aba ist definiert als:

$$\operatorname{aba}(X) := \{H | X \operatorname{abla} H\}$$

Satz 2.6.1 (Widerspruchsfreiheit der Abtrennungsregel)
aba(ag) = ag, *d. h. wenn* $H, (H \to H^\star) \in$ ag, *so* $H^\star \in$ ag.

Beweis
Die Abtrennungsregel ist eine Schlußregel, muß also gerechtfertigt werden:
$p \wedge (p \to q) \to q \in$ ag. □

Ableiten mit Einsetzungsregel
Aus dem Ausdruck H, der die Aussagenvariable p_i enthält, entsteht der Ausdruck $H\left(p_i/_{H^\star}\right)$, indem an allen Stellen von H zugleich die Variable p_i durch den Ausdruck $H^\star$ ersetzt wird.
able
Sei $X \subseteq$ ausd, $H \in$ ausd. Die Korrespondenz $X \operatorname{able} H$ wird induktiv definiert:

A) Wenn $H \in X$, so $X \operatorname{able} H$

I) Wenn $X \operatorname{able} H$ und $H^\star \in$ ausd, so $X \operatorname{able} H\left(p_i/_{H^\star}\right)$

abe
Die Operation abe ist definiert als:

$$\operatorname{abe}(X) := \{H | X \operatorname{able} H\}$$

Satz 2.6.2 (Widerspruchsfreiheit der Einsetzungsregel)
abe(ag) = ag, *d. h. wenn* $H \in$ ag, *dann* $H\left(p_i/_{H^\star}\right) \in$ ag.

Beweis
Sei $H \in$ ag und p_i komme in H vor, $H^\star \in$ ausd beliebig. Zu zeigen ist, daß $H\left(p_i/_{H^\star}\right) \in$ ag.

Nun ist

$$\operatorname{wert}\left(H\left(p_i/_{H^\star}\right), \beta\right) = \underbrace{\operatorname{wert}(H, \beta^\star)}_{\text{W weil } H \in \text{ag}}$$

mit der Belegung

$$\beta^\star(p) = \begin{cases} \beta(p) & \text{falls } p \neq p_i \\ \operatorname{wert}(H^\star, \beta) & \text{falls } p = p_i \end{cases}$$

also ist $H\left(p_i/_{H^\star}\right) \in$ ag. □

Ableiten
mit Abtrennungs- und Einsetzungsregel
abl
Die Korrespondenz abl wird induktiv definiert:

A) Wenn $H \in X$, so $X \operatorname{abl} H$

I) (a) Wenn $X \operatorname{abl} (H \to H^\star)$ und $X \operatorname{abl} H$, so $X \operatorname{abl} H^\star$ (mit Abtrennungsregel)

(b) Wenn $X \operatorname{abl} H$ und $H^\star \in \text{ausd}$, so $X \operatorname{abl} H \left(p_i/_{H^\star}\right)$ (mit Einsetzungsregel)

ab
Die Operation ab ist definiert als:

$$\operatorname{ab}(X) := \{H | X \operatorname{abl} H\}$$

Satz 2.6.3 (von der Rückverlegbarkeit der Einsetzung)
Es gilt:

$$\operatorname{ab}(X) = \operatorname{aba}(\operatorname{abe}(X))$$

Beweis
Wenn $H \in \operatorname{aba}(\operatorname{abe}(X))$, dann ist nach Definition auch $H \in \operatorname{ab}(X)$. Bleibt die Umkehrung **zu zeigen:** Wenn $X \operatorname{abl} H$, so $\operatorname{abe}(X) \operatorname{abla} H$.

durch Induktion über H.

A) $H \in X$, dann ist $H \in \operatorname{abe}(X)$, also $\operatorname{abe}(X) \operatorname{abla} H$.

I) (a) Wir haben $X \operatorname{abl} (H' \to H)$ und $X \operatorname{abl} H'$, nach Induktionsvoraussetzung ist also $\operatorname{abe}(X) \operatorname{abla} (H' \to H)$, $\operatorname{abe}(X) \operatorname{abla} H'$, wir erhalten also $\operatorname{abe}(X) \operatorname{abla} H$.

(b) Wir haben $X \operatorname{abl} H'$, $H = H' \left(p_i/_{H^\star}\right)$. Also gilt nach Induktionsvoraussetzung $\operatorname{abe}(X) \operatorname{abla} H'$.
Wenn hier $H' \in \operatorname{abe}(X)$ ist, dann ist $H = H' \left(p_i/_{H^\star}\right) \in \operatorname{abe}(X)$ also $\operatorname{abe}(X) \operatorname{abla} H$. Wenn $H' \notin \operatorname{abe}(X)$, so ist wegen $\operatorname{abe}(X) \operatorname{abla} H'$ auch $\operatorname{abe}(X) \operatorname{abla} H' \left(p_i/_{H^\star}\right)$, d. h. $\operatorname{abe}(X) \operatorname{abla} H$.

□

Satz 2.6.4 (Widerspruchsfreiheit von ab)
Wenn $X \subseteq \text{ag}$, *so* $\operatorname{ab}(X) \subseteq \text{ag}$.

Beweis
Es gilt:

$$\operatorname{ab}(X) = \operatorname{aba}(\operatorname{abe}(X)) \subseteq \operatorname{aba}(\text{ag}) \subseteq \text{ag}$$

□

Satz 2.6.5 (Eigenschaften von ab, aba, abe)
ab *:* $\mathfrak{P}(\text{ausd}) \to \mathfrak{P}(\text{ausd})$

1. $\operatorname{ab}(\emptyset) = \operatorname{aba}(\emptyset) = \operatorname{abe}(\emptyset) = \emptyset$

2. ab, aba, abe *haben die Hülleneigenschaften*
3. *Endlichkeitssatz:*
   Wenn $H \in \mathrm{ab}(X)$, *so gibt es eine endliche Teilmenge* $X^\star \subseteq X$ *mit* $H \in \mathrm{ab}(X^\star)$
4. *Ableitbarkeitstheorem (gilt für* ab *und* aba*):*
   Wenn $(H^\star \to H) \in \mathrm{ab}(X)$, *dann ist* $H \in \mathrm{ab}(X \cup \{H^\star\})$.

   Das Deduktionstheorem (die Umkehrung des Ableitbarkeitstheorems) gilt nicht, denn wir finden ein Gegenbeispiel mit $X = \emptyset$ *und* $H = H^\star$*:*

$$H^\star \in \mathrm{ab}(\{H^\star\}) = \mathrm{ab}(\emptyset \cup \{H^\star\}), \text{ aber } (H^\star \to H^\star) \notin \mathrm{ab}(\emptyset)$$

5. $\mathrm{ab}(\mathrm{ag}) \subseteq \mathrm{ag}$ *(sogar gleich)*

Folgerung 2.6.6
$[X, \mathrm{ausd}, \mathrm{ag}, \mathrm{ab}]$ *ist ein Kalkül.*

Haben wir jetzt schon eine vollständige syntaktische Charakterisierung des Folgerns? Nein, denn $\mathrm{ab}(\emptyset) = \emptyset \neq \mathrm{fl}(\emptyset) = \mathrm{ag}$ und $\mathrm{fl}(X) = \mathrm{ab}(\mathrm{ag} \cup X)$ gilt ebenfalls nicht.

Satz 2.6.7 (Syntaktische Vollständigkeit von ag bezüglich ab)
Ist $H \notin \mathrm{ag}$, *so* $\mathrm{ab}(\mathrm{ag} \cup \{H\}) = \mathrm{ausd}$.

Beweis
Sei H fest, $p_1, \ldots, p_n$ alle Aussagenvariablen in H.

$H \notin \mathrm{ag} \Rightarrow \mathrm{ef}\neg H$, d. h. es existiert eine Belegung $\beta^\star \in \mathfrak{B}$ mit $\mathrm{wert}(H, \beta^\star) = \mathrm{F}$.

Für $i = 1, \ldots, n$ sei $H_i := \begin{cases} p_i, & \text{falls } \beta^\star(p_i) = \mathrm{F} \\ \neg p_i, & \text{sonst.} \end{cases}$

Behauptung
Für alle $\beta \in \mathfrak{B}$ gilt: $\mathrm{wert}(H_1 \vee \cdots \vee H_n, \beta) = \mathrm{F}$ gdw. β stimmt auf $p_1, \ldots, p_n$ mit $\beta^\star$ überein.

Beweis der Behauptung
$\mathrm{wert}(H_1 \vee \cdots \vee H_n, \beta) = \mathrm{F}$ genau dann, wenn für alle $i = 1, \ldots, n$ $\mathrm{wert}(H_i, \beta) = \mathrm{F}$, d. h.

$$\beta(p_i) = \left\{ \begin{array}{ll} \mathrm{F}, & \text{falls } H_i = p_i \\ \mathrm{W}, & \text{sonst} \end{array} \right\} = \beta^\star(p_i)$$

□

Behauptung
$(H \to H_1 \vee \cdots \vee H_n) \in \mathrm{ag}$

Beweis der Behauptung (indirekt)
Annahme:$(H \to H_1 \vee \cdots \vee H_n) \notin \mathrm{ag}$
dann existiert $\beta \in \mathfrak{B}$ mit $\mathrm{wert}(H \to H_1 \vee \cdots \vee H_n, \beta) = \mathrm{F}$

$\mathrm{wert}(H, \beta) = \mathrm{W}$ und $\underbrace{\mathrm{wert}(H_1 \vee \cdots H_n, \beta) = \mathrm{F}}_{\beta = \beta^\star \text{auf } p_1, \ldots p_n}$

$\mathrm{W} = \mathrm{wert}(H, \beta) = \mathrm{wert}(H, \beta^\star) = \mathrm{F}$ ↯ Widerspruch □

Also ist $(H \to H_1 \vee \cdots \vee H_n) \in \mathrm{ag}$.

$$\frac{\begin{array}{c} H \in \mathrm{ab}(\mathrm{ag} \cup \{H\}) \\ (H \to H_1 \vee \cdots \vee H_n) \in \mathrm{ab}(\mathrm{ag} \cup \{H\}). \end{array}}{H_1 \vee \cdots \vee H_n \in \mathrm{ab}(\mathrm{ag} \cup \{H\}).}$$

Wir setzen ein für $p_i \begin{cases} p \text{ , falls } H_i = p_i \\ \neg p, \text{ falls } H_i = \neg p_i \end{cases}$ in $H_1 \vee \cdots \vee H_n$ und erhalten

$$H^\star = [\neg\neg]\, p \vee [\neg\neg]\, p \vee \cdots \vee [\neg\neg]\, p$$

$$\Rightarrow \qquad \frac{\begin{array}{c} H^\star \in \mathrm{ab}\,(\mathrm{ag} \cup \{H\}) \\ (H^\star \to p) \in \mathrm{ab}\,(\mathrm{ag} \cup \{H\}) \end{array}}{p \in \mathrm{ab}\,(\mathrm{ag} \cup \{H\})\,.} \qquad (\text{weil } H^\star \to p \in \mathrm{ag})$$

$H^{\star\star} \in$ ausd: $H^{\star\star} = p\left[p/_{H^{\star\star}}\right] \in \mathrm{ab}\,(\mathrm{ag} \cup \{H\})$. □

Bemerkung
ag ist nicht syntaktisch vollständig bezüglich fl.

Beispiele

- $p \notin \mathrm{ag}$; $H \in \mathrm{fl}\,(\mathrm{ag} \cup \{p\})$ gdw. $(p \to H) \in \mathrm{fl}\,(\mathrm{ag}) = \mathrm{ag}$ gdw. $(p \to H) \in \mathrm{ag}$
- $(p \to q) \notin \mathrm{ag}$, also $q \notin \mathrm{fl}\,(\mathrm{ag} \cup \{p\})$.

Die Gleichung $\mathrm{fl}\,(X) = \mathrm{ab}\,(\mathrm{ag} \cup X)$ gilt also nicht, aber vielleicht gilt

$$\mathrm{fl}\,(X) = \mathrm{aba}\,(\mathrm{ag} \cup X)$$

denn offensichtlich ist es gerade die Einsetzungsregel, die die syntaktische Vollständigkeit von ag bzgl. ab impliziert.

Damit stellt sich die Frage: fl erfüllt das Deduktionstheorem, aba nicht, kann die Gleichung trotzdem gelten?

Die Antwort gibt der

Satz 2.6.8 (Schwaches Deduktionstheorem für aba)

Voraussetzung:

1. *Wenn der Ausdruck H eine der Formen*
   - $H_1 \to (H_2 \to H_1)$ (1)
   - $H_1 \to H_1$ (2)
   - $(H_1 \to (H_2 \to H_3)) \to ((H_1 \to H_2) \to (H_1 \to H_3))$ (3)

   hat, so ist $X \mathrm{abla}\, H$.
2. $(X \cup \{H^{\star\star}\})\, \mathrm{abla}\, H^\star$

Behauptung: $X \mathrm{abla}\,(H^{\star\star} \to H^\star)$

Beweis
Induktion über $H^\star$.

A) $H^\star \in X \cup \{H^{\star\star}\}$.

Fall 1: $H^\star = H^{\star\star}$
Nach Voraussetzung $X \mathrm{abla}\,(H^\star \to H^\star)$, also $X \mathrm{abla}\,(H^{\star\star} \to H^\star)$

Fall 2: $H^\star \neq H^{\star\star}$
Dann ist $H^\star \in X$. Nach Voraussetzung

$$\frac{\begin{array}{c} X \operatorname{abla} H^\star \to (H^{\star\star} \to H^\star) \\ X \operatorname{abla} H^\star \end{array}}{X \operatorname{abla} (H^{\star\star} \to H^\star)}$$

I)

$$\left.\begin{array}{c} X \cup \{H^{\star\star}\} \operatorname{abla} H_1 \to H^\star \\ X \cup \{H^{\star\star}\} \operatorname{abla} H_1 \end{array}\right\} \to X \cup \{H^{\star\star}\} \operatorname{abla} H^\star$$

Nach Induktionsvoraussetzung gilt $X \operatorname{abla} H^{\star\star} \to (H_1 \to H^\star)$ und $X \operatorname{abla} H^{\star\star} \to H_1$.

Nach Voraussetzung (2.6.8.1.3) ist
$X \operatorname{abla} (H^{\star\star} \to (H_1 \to H^\star)) \to ((H^{\star\star} \to H_1) \to (H^{\star\star} \to H^\star))$
wegen Abtrennungsregel folgt:
$X \operatorname{abla} (H^{\star\star} \to H_1) \to (H^{\star\star} \to H^\star)$, also $X \operatorname{abla} H^{\star\star} \to H^\star$.

□

Die Ausdrücke (2.6.8.1.1), (2.6.8.1.2) und (2.6.8.1.3) sind aus ag, also gültig!

Satz 2.6.9
$\operatorname{fl}(X) = \operatorname{aba}(\operatorname{ag} \cup X)$.

Um den Beweis zu führen benötigen wir folgendes

Lemma
Es seien $F_1, F_2 : \mathfrak{P}(\text{ausd}) \to \mathfrak{P}(\text{ausd})$ *mit*

1. $F_1(\emptyset) \subseteq F_2(\emptyset)$
2. F_1 *erfüllt den Endlichkeitssatz*
3. *Wenn* $X^\star$ *endlich, so gilt Deduktionstheorem für* F_1*:*
$$\text{Wenn } H_2 \in F_1(X^\star \cup \{H_1\}), \text{ so } (H_1 \to H_2) \in F_1(X^\star)$$
4. *Wenn* $X^\star$ *endlich,* X *beliebig,* $X^\star \subseteq X$*, so* $F_2(X^\star) \subseteq F_2(X)$.
5. *Wenn* $X^\star$ *endlich, so gilt Ableitbarkeitstheorem für* F_2*:*
$$\text{Wenn } (H_1 \to H_2) \in F_2(X^\star), \text{ so } H_2 \in F_2(X \cup \{H_1\})$$

Dann ist für alle $X \in \text{ausd}$*:*
$$F_1(X) \subseteq F_2(X)$$

Beweis des Lemmas
Es sei $H \in F_1(X)$. Da F_1 den Endlichkeitssatz erfüllt, existiert endliches $X^\star = \{H_1, \ldots, H_n\}$ mit $H \in F_1(X^\star)$.

Fall 1: $n = 0$, $X^\star = \emptyset$
$H \in F_1(\emptyset) \subseteq F_2(\emptyset) \subseteq F_2(X)$ Dies ist doch nicht nötig, oder?

Fall 2: $n > 0$ $H \in F_1\big(\underbrace{\{H_1, \ldots, H_{n-1}\} \cup \{H_n\}}_{X^\star}\big)$. Mit Deduktionstheorem für F_1 ergibt sich $(H_n \to H) \in F_1(\{H_1, \ldots, H_{n-2}\} \cup \{H_{n-1}\})$, usw.

Wir erhalten also $(H_1 \to (H_2 \to \cdots (H_n \to H) \cdots) \in F_1(\emptyset) \subseteq F_2(\emptyset)$.

Mit Ableitbarkeitstheorem für F_2 ergibt sich dann

$$H \in F_2(\{H_1, \ldots, H_n\}) \subseteq F_2(X)$$

□

Beweis des Satzes 2.6.9

Wir wenden zweimal das Lemma an:

$$\left.\begin{array}{l} \bullet\ F_1(X) = \mathrm{fl}(X) \\ \bullet\ F_2(X) = \mathrm{aba}(\mathrm{ag} \cup X). \end{array}\right\} \mathrm{fl}(X) \subseteq \mathrm{aba}(\mathrm{ag} \cup X)$$

$$\left.\begin{array}{l} \bullet\ F_1(X) = \mathrm{aba}(\mathrm{ag} \cup X) \\ \bullet\ F_2(X) = \mathrm{fl}(X) \end{array}\right\} \mathrm{aba}(\mathrm{ag} \cup X) \subseteq \mathrm{fl}(X)$$

Wir erhalten also $\mathrm{fl}(X) = \mathrm{aba}(\mathrm{ag} \cup X)$. □

2.6.2 Syntaktische Charakterisierung von ag

Die mit Satz 2.6.9 gegebene Charakterisierung des Folgerns durch das Ableiten mit der Abtrennungsregel allein ist noch keine rein syntaktische Charakterisierung des Folgerns, weil auf der rechten Seite der Gleichung noch die Menge ag vorkommt, die unter Bezug auf die Semantik definiert ist. Es bleibt also übrig, diese Menge syntaktisch zu charakterisieren.

Das Axiomensystem axa
Die Menge axa besteht aus den folgenden 15 Ausdrücken:

1. $p \to (q \to p)$ (Prämissenbelastung)
2. $((p \to q) \to p) \to p$ (Satz von Peirce)
3. $(p \to q) \to ((q \to r) \to (p \to r))$ (Kettenschluß)
4. $p \wedge q \to p$
5. $p \wedge q \to q$
6. $(p \to q) \to ((p \to r) \to (p \to (q \wedge r)))$
7. $p \to p \vee q$
8. $q \to p \vee q$
9. $(p \to r) \to ((q \to r) \to ((p \vee q) \to r))$
10. $(p \leftrightarrow q) \to (p \to q)$
11. $(p \leftrightarrow q) \to (q \to p)$
12. $(p \to q) \to ((q \to p) \to (p \leftrightarrow q))$
13. $(p \to q) \to (\neg q \to \neg p)$
14. $p \to \neg\neg p$
15. $\neg\neg p \to p$

Man sieht sofort

Satz 2.6.10
axa $\subseteq$ ag

Satz 2.6.11 (Widerspruchsfreiheit von axa bezüglich ab)
Es gilt:

$$\mathrm{ab}\,(\mathrm{axa}) \subseteq \mathrm{ag}$$

Beweis
Es ist axa $\subseteq$ ag und nach Satz 2.6.4 gilt für ab die Widerspruchsfreiheit. □

Satz 2.6.12 (Vollständigkeit von axa bezüglich ab)
Es gilt:

$$\mathrm{ab}\,(\mathrm{axa}) \supseteq \mathrm{ag}$$

Beweis
Zeigen für $H^\star \in$ ausd:

$$\text{Wenn } H^\star \notin \mathrm{ab}\,(\mathrm{axa})\,, \text{ so } H^\star \notin \mathrm{ag}$$

d. h. wir suchen eine Belegung $\beta^\star$ mit $\mathrm{wert}(H^\star, \beta^\star) = \mathrm{F}$

Schritt 1: Satz 2.6.13 (Lindenbaumscher Ergänzungssatz)
$M \subseteq$ ausd, $H^\star \notin F\,(M)$. Es sei F ein Deduktionsoperator, dann gibt es eine Menge $M^\star$ mit

1. $M^\star \supseteq M$
2. $H^\star \notin F\,(M^\star)$
3. *Für alle $H \in$ ausd gilt: Wenn $H \notin F\,(M^\star)$, so $H^\star \in F\,(M^\star \cup \{H\})$.*
4. $F\,(M^\star) = M^\star$.

Beweis des Lindenbaumschen Ergänzungssatzes
Wir gehen von einer Aufzählung aller Ausdrücke aus: ausd $= \{H_0, H_1, \ldots\}$.

$N_0 := M$.

$$N_{i+1} := \begin{cases} N_i \cup \{H_i\}\,, & H^\star \notin F\,(N_i \cup \{H_i\}) \\ N_i, & \text{sonst} \end{cases}$$

$M^\star = \bigcup_{i \geq 0} N_i$.

Zu 1: $M \subseteq M^\star$ ist klar.

Zu 2: $H^\star \notin F\,(M^\star)$
Wäre $H^\star \in F\,(M^\star)$, so würde es $X^\star \subseteq M^\star$ und $X^\star$ – endlich geben, mit $H^\star \in F\,(X^\star)$. Weil $X^\star$ endlich, existiert ein k mit $X^\star \subseteq N_k$, also mit $H^\star \in F\,(N_k)$ ↯ zur Konstruktion.

Zu 3: Sei $H = H_k \in$ ausd; $H \notin F\,(M^\star)$. Nach Konstruktion ist $H^\star \in F\,(N_k \cup \{H\})$, sonst wäre $H = H_k \in N_k \subseteq M^\star \subseteq F\,(M^\star)$. Also $H^\star \in F\,(M^\star \cup \{H\})$.

Zu 4: Es sei $H \in F\,(M^\star)$.
Behauptung: $H \in M^\star$
Es sei $H = H_k$.
zu zeigen: $H^\star \notin F\,(N_k \cup \{H_k\})$.
Nun ist $H^\star \notin F\,(M^\star)$, also $H^\star \notin F\,(M^\star \cup \{H\})$, weil $H \in F\,(M^\star)$. $H^\star \notin F\,(N_k \cup \{H\})$, weil $N_k \subseteq M^\star$.

Damit haben wir den LINDENBAUMschen Ergänzungssatz bewiesen. □

Schritt 2: Anwendung mit $M = \mathrm{ab}\,(\mathrm{axa})\,; F = \mathrm{aba}$
$H^\star \notin \mathrm{aba}\,(\mathrm{ab}\,(\mathrm{axa}))$ (sonst $H^\star \in \mathrm{ab}(\mathrm{axa}) = \mathrm{aba}(\mathrm{abe}(\mathrm{axa}))$).

Es gibt $M^\star \subseteq \mathrm{ausd}$ mit

1. $\mathrm{ab}\,(\mathrm{axa}) \subseteq M^\star$
2. $H^\star \notin \mathrm{aba}\,(M^\star)$
3. Für alle H: Wenn $H \notin \mathrm{aba}\,(M^\star)$, so $H^\star \in \mathrm{aba}\,(M^\star \cup \{H\})$.
4. $\mathrm{aba}\,(M^\star) = M^\star$.

Schritt 3: Man zeigt, daß die folgenden Ausdrücke aus axa ableitbar sind:

$$p \to p, (p \to (q \to r)) \to ((p \to q) \to (p \to r)) \in \mathrm{ab}\,(\mathrm{axa})$$

außerdem ist $p \to (q \to p) \in \mathrm{axa}$.

Also gilt nach dem schwachen Deduktionstheorem für aba:

$$\text{Wenn } (M^\star \cup \{H_1\})\,\mathrm{abla}\,H_2, \text{ so } M^\star \mathrm{abla}\,(H_1 \to H_2)$$

Schritt 4: Lemma 1

$$\begin{array}{rl} (H_1 \to H_2) \in M^\star & \textit{gdw. Wenn } H_1 \in M^\star, \textit{ so } H_2 \in M^\star \\ H_1 \wedge H_2 \in M^\star & \textit{gdw. } H_1 \in M^\star \textit{ und } H_2 \in M^\star \\ H_1 \vee H_2 \in M^\star & \textit{gdw. } H_1 \in M^\star \textit{ oder } H_2 \in M^\star \\ H_1 \leftrightarrow H_2 \in M^\star & \textit{gdw. } H_1 \in M^\star \textit{ gdw. } H_2 \in M^\star \\ \neg H_1 \in M^\star & \textit{gdw. } H_1 \notin M^\star \end{array}$$

Beweis
der ersten Behauptung von Lemma 1

$\Rightarrow$ $\left.\begin{array}{l} (H_1 \to H_2) \in M^\star \\ H_1 \in M^\star \end{array}\right\} \Rightarrow H_2 \in M^\star$ wegen Abtrennungsregel.

$\Leftarrow$ Wenn $H_1 \in M^\star$, so $H_2 \in M^\star$, d. h. $H_1 \notin M^\star$ oder $H_2 \in M^\star$.

Fall 1: Sei zunächst $H_1 \notin M^\star$, dann ist $H^\star \in \mathrm{aba}\,(M^\star \cup \{H_1\})$, also $(H_1 \to H^\star) \in \mathrm{aba}\,(M^\star)$
indirekt: Sei $(H_1 \to H_2) \notin M^\star$
Dann gilt: $(H_1 \to H_2) \to H^\star \in \mathrm{aba}\,(M^\star)$ und $(H_1 \to H^\star) \to ((H_1 \to H_2) \to H^\star) \to H^\star) \in \mathrm{ab}\,(\mathrm{axa}) \subseteq M^\star$, d. h. weil $M^\star$ abgeschlossen gegen Abtrennung: $H^\star \in M^\star$ ↯ Widerspruch.

Fall 2: $H_2 \in M^\star$
Es ist $H_2 \to (H_1 \to H_2) \in \mathrm{ab}\,(\mathrm{axa}) \subseteq M^\star$ (Prämissenbelastung) also $H_1 \to H_2 \in M^\star$.

Die restlichen Beweise führen wir hier nicht aus. □

Schritt 5: Jetzt geben wir eine Belegung $\beta^\star$ an, die $H^\star$ falsch macht:

$$\beta^\star\,(p_i) = \begin{cases} \mathrm{W}, & p_i \in M^\star \\ \mathrm{F}, & \text{sonst} \end{cases}$$

Lemma 2
$\mathrm{wert}\,(H, \beta^\star) = \mathrm{W}$ *gdw.* $H \in M^\star$ *(dann folgt unmittelbar* $\mathrm{wert}\,(H^\star, \beta^\star) = \mathrm{F}$*, d. h.* $H^\star \notin \mathrm{ag}$*).*

Beweis
durch Induktion über H

A) H ist AV p_i
wert $(H, \beta^\star) = \beta^\star(p_i) = \mathrm{W}$ gdw. $p_i \in M^\star$

I) $H \to \neg H$
wert $(\neg H, \beta^\star) = \mathrm{W}$ gdw. wert $(H, \beta^\star) = \mathrm{F}$ gdw. $H \notin M^\star$ gdw. $\neg H \in M^\star$
analog weiter mit Ausdrücken der Form $H_1 \wedge H_2$, $H_1 \vee H_2$, $H_1 \to H_2$ und $H_1 \leftrightarrow H_2$.

□

Damit haben wir den Vollständigkeitssatz bewiesen. □

2.6.3 Cut–Ableitbarkeit

Wir haben also ag = ab(axa). Liefert uns diese syntaktische Charakterisierung von ag auch einen Algorithmus, mit dem wir für jeden Ausdruck $H \in$ ausd die Frage „$H \in$ ag?“ entscheiden können, also ein *syntaktisches Entscheidungsverfahren* für ag?

Nein, eine Axiomatisierung liefert im allgemeinen kein Entscheidungsverfahren, durch Ableiten können wir nur im Falle $H \in$ ag zur Antwort „ja“ kommen, wenn H nicht allgemeingültig ist, können wir das durch Ableiten i. a. nicht herausfinden.

Wir können aber die syntaktische Vollständigkeit von ag ausnutzen:

$$\text{Wenn } H \notin \text{ag, so ab}\,(\text{ag} \cup \{H\}) = \text{ausd.}$$

Wenn $H \notin$ ag, so ist insbesondere der Widerspruch (die Kontradiktion) $p \wedge \neg p$ aus ag $\cup \{H\}$ bzw. aus axa $\cup \{H\}$ ableitbar.

Um dies gezielt tun zu können, erweitern wir den Kalkül um das Zeichen $\perp$ für den Widerspruch:

Der Anfangsschritt in der Ausdrucksdefinition (siehe Seite 52) lautet jetzt:

A) 1. Jede Aussagenvariable ist ein Ausdruck.
2. $\perp$ ist ein Ausdruck.

Wir setzen fest, daß für jede Belegung β gilt:

$$\text{wert}(\perp, \beta) \quad = \quad \mathrm{F}$$

Folgerung 2.6.14
Für $\perp$ gilt:

1. $\neg\perp \in$ ag
2. $\{\perp\}$ *hat kein Modell*
3. $\perp \in \mathrm{fl}(X)$ *genau dann, wenn X kein Modell besitzt, also X widersprüchlich ist.*

Damit gilt dann

$$\begin{aligned} H \in \text{ag} \quad & \text{gdw.} \quad \neg H \text{ nicht erfüllbar ist} \\ & \text{gdw.} \quad \{\neg H\} \text{ kein Modell hat} \\ & \text{gdw.} \quad \bot \in \text{fl}\,(\{\neg H\}) \end{aligned}$$

Wir benötigen also „nur noch“ eine syntaktische Relation $\vdash$ (ähnlich wie zuvor abla, ...) mit

$$Y \vdash \bot \quad \text{gdw.} \quad \text{Y kein Modell besitzt,}$$

für die entscheidbar ist, ob $Y \vdash \bot$ gilt oder nicht.

Eine solche Relation ist die *Cut–Ableitbarkeit* (mittels der Schnittregel), die von Gentzen angegeben wurde.

Ihr Nachteil besteht darin, daß die Schnittregel nur auf spezielle Ausdrücke, die *Klauseln*, angewendet werden kann.

Gentzen–Ausdruck (Klausel)
Ein Ausdruck $H \in$ ausd heißt *Klausel*, wenn H eine der Formen hat:

- $p_1 \wedge p_2 \wedge \cdots \wedge p_m \to q_1 \vee q_2 \vee \cdots \vee q_n \;\; (m, n \geq 1)$
- $q_1 \vee q_2 \vee \cdots \vee q_n \, (n \geq 1)$
- $p_1 \wedge \cdots \wedge p_m \to \bot \, (m \geq 1)$
- $\bot$

Dabei sind $p_1, \ldots, p_m$ bzw. $q_1, \ldots, q_n$ jeweils paarweise verschiedene Aussagenvariablen.

Mit einer Einschränkung der Ausdrucksmenge ist stets die Frage verbunden, ob dadurch die Ausdrucksfähigkeit des Kalküls (hier die Repräsentierbarkeit von Wahrheitsfunktionen) ebenfalls echt eingeschränkt wird.

Zur Beantwortung führen wir folgenden Begriff ein:

Definition 2.6.15 (semantische Äquivalenz)
Ausdrücke H_1, H_2 *heißen* semantisch äquivalent $H_1 \sim_{\text{sem}} H_2$ *gdw.* $(H_1 \leftrightarrow H_2) \in \text{ag}$

Satz 2.6.16
Jeder Ausdruck H *ist semantisch äquivalent zu einer* Konjunktion von Klauseln*:*

$$H \sim_{\text{sem}} K_1 \wedge K_2 \wedge \cdots \wedge K_m$$

Beweis
Der Ausdruck $H = \bot$ ist eine Klausel, für $\neg\bot$ gilt offenbar $\neg\bot \sim_{\text{sem}} (p \to p)$ und das ist eine Klausel.

Alle Aussagenvariablen sind Klauseln und $\neg p_i \sim_{\text{sem}} (p_i \to \bot)$.

Nun sei H ein beliebiger Ausdruck, in dem genau die Variablen $p_1, \ldots, p_n$ vorkommen. Man beachte, daß $n = 0$ sein kann, etwa

$$H = \bot \to (\neg\bot \vee \bot)$$

Solche Ausdrücke sind allerdings entweder allgemeingültig oder Kontradiktion.

Fall 1: $H \in \text{ag}$.
Dann gilt

$$H \sim_{\text{sem}} (p_1 \to p_1) \wedge (p_2 \to p_2) \wedge \cdots \wedge (p_n \to p_n)$$

und das ist eine Konjunktion von Klauseln

Fall 2: H ist eine Kontradiktion
d. h. für alle β ist $\text{wert}(H, \beta) = \text{F}$.

Dann ist

$$H \sim_{\text{sem}} \bot$$

Fall 3: $H \notin \text{ag}$, $H \notin \text{kt}$
Zu jeder Belegung β von $\{p_1, \ldots, p_n\}$ sei

$$A_\beta := [\neg]^{\beta(p_1)} p_1 \vee \cdots \vee [\neg]^{\beta(p_n)} p_n$$

die zu β gehörige *Elementaralternative*, wobei $[\neg]^{\text{F}} := e \quad [\neg]^{\text{W}} := \neg$ ist.

Beispiel

$$\begin{matrix} p_1 & p_2 & p_3 \\ \text{F} & \text{W} & \text{F} \end{matrix} \to A_\beta = p_1 \vee \neg p_2 \vee p_3$$

Man überlegt sich (wie für Elementarkonjunktionen – siehe Seite 54), daß folgendes gilt:

Lemma
$\text{wert}\,(A_\beta, \beta^\star) = \text{F}$ *gdw.* $\beta\,(p_i) = \beta^\star\,(p_i)\,, i = 1, \ldots, n$

Wir zeigen nun, daß

$$H \sim_{\text{sem}} \bigwedge_{\text{wert}(H,\beta)=\text{F}} A_\beta$$

Beweis
Es ist zu zeigen, daß für jede Belegung $\beta^\star$ gilt

$$\text{wert}\,(H, \beta^\star) = \text{wert}\left(\bigwedge_{\text{wert}(H,\beta)=\text{F}} A_\beta, \beta^\star\right)$$

d. h.

$$\text{wert}\,(H, \beta^\star) = \text{F} \text{ gdw. } \text{wert}\left(\bigwedge_{\text{wert}(H,\beta)=\text{F}} A_\beta, \beta^\star\right) = \text{F}$$

Nun ist

$$\text{wert}\left(\bigwedge_{\text{wert}(H,\beta)=\text{F}} A_\beta, \beta^\star\right) = \text{F} \text{ gdw. } \begin{array}{l} \text{es ein } \beta \text{ mit } \text{wert}(H, \beta) = \text{F} \\ \text{derart gibt, daß } \text{wert}\,(A_\beta, \beta^\star) = \text{F} \end{array}$$

Nach dem Lemma ist das der Fall genau dann, wenn $\text{wert}(H, \beta^\star) = \text{F}$, was zu zeigen war. □

Der Ausdruck

$$\bigwedge_{\text{wert}(H,\beta)=\text{F}} A_\beta$$

heißt *Kanonische Konjunktive Normalform* (KKNF) von H.

Nachdem wir zu H die KKNF als eine Konjunktion von Elementaralternativen bestimmt haben, genügt es zu zeigen, daß jede solche Alternative semantisch äquivalent zu einer Klausel ist:

1. In A_β kommt $\neg$ nicht vor:

$$p_1 \vee p_2 \vee \cdots \vee p_n$$

   Das ist eine Klausel.

2. alle Variablen sind mit $\neg$ versehen:

$$\neg p_1 \vee \neg p_2 \vee \cdots \vee \neg p_n$$

   Dieser Ausdruck ist semantisch äquivalent zu

$$p_1 \wedge p_2 \wedge \cdots \wedge p_n \rightarrow \bot$$

3. sonst:
   Es seien $p_{i_1}, \ldots, p_{i_k}$ die Variablen, die in A_β mit $\neg$ versehen sind und $p_{i_{k+1}}, \ldots, p_{i_n}$ die Variablen, die ohne $\neg$ vorkommen. Dann ist

$$A_\beta \sim_{\text{sem}} \left(p_{i_1} \wedge \cdots \wedge p_{i_k} \rightarrow p_{i_{k+1}} \vee \cdots \vee p_{i_n}\right)$$

   und dieser Ausdruck ist eine Klausel.

Damit haben wir gezeigt, daß jeder Ausdruck semantisch äquivalent ist zu einer Konjunktion von Klauseln. □

Nun müssen wir die Abtrennungsregel entsprechend anpassen:

Die Schnittregel Cut (H_1, H_2, H)
Sofern die Klauseln H_1, H_2 den Voraussetzungen genügen, entsteht durch Anwendung der Schnittregel auf H_1, H_2 eine Klausel H durch folgende Definition:

$$\frac{\left.\begin{array}{r} H_1 : [L \rightarrow]\, q_1 \vee \cdots \vee q_n \\ H_2 : p_1 \wedge \cdots \wedge p_m \rightarrow R \end{array}\right\} \text{Klauseln} \qquad \text{es gibt } i, k \text{ mit } 1 \leq i \leq n, 1 \leq k \leq m \text{ und } q_i = p_k.}{H : [L\wedge]\, p_1 \wedge \cdots \wedge p_{k-1} \wedge p_{k+1} \wedge \cdots \wedge p_m \rightarrow R \vee q_1 \vee \cdots \vee q_{i-1} \vee q_{i+1} \cdots \vee q_n}$$

Die Schnittregel vereinfacht Klauselmengen semantisch äquivalent. Beachte: Das Ergebnis ist i. a. noch keine Klausel, notwendig ist das Streichen mehrfach auf der gleichen Seite vorkommender Literale.

Dabei gilt als Bezug zum Folgern:

Satz 2.6.17 (Rechtfertigung der Schnittregel)
Wenn Cut (H_1, H_2, H), *so* $H \in \text{fl}\,(\{H_1, H_2\})$.

Es gilt also insbesondere auch: Wenn agH_1, agH_2 und Cut (H_1, H_2, H), dann ag$H$.

Beweis des Satzes (indirekt)
Sei β Modell von $\{H_1, H_2\}$, d. h. wert $(H_1, \beta) = \text{W}$ und wert $(H_2, \beta) = \text{W}$. Wir haben zu zeigen: $\beta \text{erf} H$.

Wir nehmen an, daß wert $(H, \beta) = \text{F}$.

Dann ist $\begin{array}{lll} \text{wert}\,(L,\beta) = \text{W}, & \text{für } j = 1,\ldots,m \text{ mit } j \neq k & \beta\,(p_j) = \text{W} \\ \text{wert}\,(R,\beta) = \text{F}, & \text{für } j = 1,\ldots,n \text{ mit } j \neq i & \beta\,(q_j) = \text{W} \end{array}$

Wir betrachten jetzt H_2. Nach Voraussetzung ist wert $(H_2, \beta) = \text{W}$.
Wegen wert $(R, \beta) = \text{F}$ ist wert $(p_1 \wedge \cdots \wedge p_m, \beta) = \text{F}$, also

$$\beta\,(p_k) = \text{F} = \beta\,(q_i)$$

Wir erhalten wert $(H_1, \beta) = \text{F}$, also $\lightning$ zur Voraussetzung. □

Wir definieren nun den wichtigen Begriff der

Cut–Ableitbarkeit $Y \vdash H$
Y sei eine Menge von Klauseln, $H \in$ ausd. Dann ist $Y \vdash H$ induktiv definiert durch

A) Wenn $H \in Y$, so $Y \vdash H$

I) Wenn $Y \vdash H_1$, $Y \vdash H_2$ und Cut (H_1, H_2, H), so $Y \vdash H$

Folgerung 2.6.18
Wenn $Y \vdash H$, so $H \in \text{fl}\,(Y)$

Uns interessiert natürlich vor allem, wann $Y \vdash \bot$ gilt.

Lemma
Wenn $(Y \cup \{p_i\}) \vdash \bot$, so $Y \vdash \bot$ oder $Y \vdash (p_i \to \bot)$.

Beweis
Wir betrachten eine Cut–Ableitung von $\bot$ aus $Y \cup \{p_i\}$.

Fall 1: Bei einem Ableitungsschritt geht der Ausdruck p_i (als H_1) in die Schnittregel ein.
Wir lassen diese Ableitungsschritte aus dem Beweis heraus, dann ist entweder $Y \vdash (p_i \to \bot)$ oder sogar $Y \vdash \bot$.

Fall 2: Der Ausdruck p_i wird bei keinem Ableitungsschritt verwendet, dann ist $Y \vdash \bot$.
Also ist $Y \vdash \bot$ oder $Y \vdash (p_i \to \bot)$.

□

Folgerung 2.6.19
Wenn $(Y \cup \{p_i\}) \vdash \bot$ und $(Y \cup \{p_i \to \bot\}) \vdash \bot$, so $(Y \vdash \bot)$.

Beweis
Aus $(Y \cup \{p_i\}) \vdash \bot$ haben wir nach Lemma $Y \vdash \bot$ oder $Y \vdash (p_i \to \bot)$.
Aus $Y \vdash (p_i \to \bot)$ und $(Y \cup \{p_i \to \bot\}) \vdash \bot$ ergibt sich sofort $Y \vdash \bot$. □

Satz 2.6.20
Ist Y eine Menge von Klauseln, die kein Modell besitzt, d. h. $\bot \in \text{fl}(Y)$, dann ist $Y \vdash \bot$.

Beweis
Wir zeigen, daß Y ein Modell hat ($\perp \notin \mathrm{fl}(Y)$), wenn $Y \not\vdash \perp$ gilt.

Wir definieren induktiv

$$\begin{array}{lcl} Y_0 & := & Y \\ Y_{n+1} & := & \begin{cases} Y_n \cup \{p_n\}, & \text{falls } (Y_n \cup \{p_n\} \not\vdash \perp) \\ Y_n \cup \{p_n \to \perp\}, & \text{sonst.} \end{cases} \end{array}$$

und setzen

$$Y^\star \quad := \quad \begin{array}{l} \text{die kleinste Menge von Klauseln, die alle } Y_i \text{ enthält} \\ \text{und abgeschlossen ist in bezug auf die Schnittregel.} \end{array}$$

(die *Lindenbaumsche Ergänzungsmenge*)

Es gilt

1. $Y \subseteq Y^\star$
2. Für alle i ist $p_i \in Y^\star$ oder $(p_i \to \perp) \in Y^\star$
3. $Y^\star \not\vdash \perp$

Die dritte Aussage ergibt sich daraus, daß für alle Y_i gilt $Y_i \not\vdash \perp$ (und dem Endlichkeitssatz für $\perp$), das beweisen wir durch Induktion über i wie folgt:

Anfangsschritt: $Y_0 \not\vdash \perp$ (weil $Y_0 = Y$).

Induktionsschritt: Wenn $Y_i \not\vdash \perp$, so $Y_{i+1} \not\vdash \perp$.
Damit gleichwertig ist

$$\text{Wenn } Y_{i+1} \vdash \perp, \text{ so } Y_i \vdash \perp$$

Wegen $Y_{i+1} \vdash \perp$ ist nach Konstruktion

$$Y_{i+1} = Y_i \cup \{p_i \to \perp\}$$

und

$$(Y_i \cup \{p_i\}) \vdash \perp$$

Wir haben also

$$(Y_i \cup \{p_i \to \perp\}) \vdash \perp \text{ und } (Y_i \cup \{p_i\}) \vdash \perp$$

d. h. nach Folgerung $Y_i \vdash \perp$, was zu zeigen war.

Wir betrachten jetzt die Belegung β mit

$$\beta(p_i) := \begin{cases} \mathrm{W}, & \text{falls } p_i \in Y^\star \\ \mathrm{F}, & \text{falls } (p_i \to \perp) \in Y^\star \end{cases}$$

Behauptung: β ist Modell von Y.

Zu zeigen: Wenn $\mathrm{wert}(H, \beta) = \mathrm{F}$, so $H \notin Y$. $H : p_1 \wedge \cdots \wedge p_m \to q_1 \vee \cdots \vee q_n$ $\mathrm{wert}(H, \beta) = \mathrm{F}$

Für $i = 1, \ldots, m$ ist $\beta(p_i) = \mathrm{W}$, d. h. $p_i \in Y^\star$
$j = 1, \ldots, n$ ist $\beta(q_i) = \mathrm{F}$, d. h. $(q_j \to \perp) \in Y^\star$

Es gilt:
$\text{Cut}\,(p_1 \wedge \cdots \wedge p_m \to q_1 \vee \cdots \vee q_n, p_1, p_2 \wedge \cdots \wedge p_m \to q_1 \vee \cdots \vee q_n)$
$\text{Cut}\,(p_2 \wedge \cdots \wedge p_m \to q_1 \vee \cdots \vee q_n, p_2, p_3 \wedge \cdots \wedge p_m \to q_1 \vee \cdots \vee q_n)$
$\vdots$
$\text{Cut}\,(p_m \to q_1 \vee \cdots \vee q_n, p_m, q_1 \vee \cdots \vee q_n)$
$\text{Cut}\,(q_1 \vee \cdots \vee q_n, q_1 \to \bot, q_2 \vee \cdots \vee q_n)$
$\vdots$
$\text{Cut}\,(q_n, q_n \to \bot, \bot)$

Wäre $H \in Y$, so hätten wir $H \in Y^\star$ und folglich $Y^\star \vdash \bot$ in Widerspruch zu 3. Also ist $H \notin Y$, was zu zeigen war. Damit haben wir den Satz bewiesen. □

Wegen obiger Folgerung haben wir also

$$\bot \in \text{fl}(Y) \text{ gdw. } Y \vdash \bot$$

Satz 2.6.21
Es sei Y eine Menge von Klauseln, die ein Modell hat; $\bot \notin \text{fl}\,(Y)$.

Dann gilt für alle Aussagenvariablen p_i

1. *Wenn $p_i \in \text{fl}\,(Y)$, so $Y \vdash p_i$*
2. *Wenn $(p_i \to \bot) \in \text{fl}\,(Y)$, so $Y \vdash (p_i \to \bot)$.*

Die Schnittregel ist also vollständig für das Folgern von Formeln der Form p_i bzw. $(p_i \to \bot)$.

Beweis
Sei $p_i \in \text{fl}\,(Y)$, d. h. jedes Modell von Y erfüllt p_i, folglich

$$\bot \in \text{fl}\,(Y \cup \{p_i \to \bot\})\,,$$

d. h. kein Modell von Y erfüllt $(p_i \to \bot)$.

Nach Satz 2.6.20 folgt: $(Y \cup \{p_i \to \bot\}) \vdash \bot$. Eine entsprechende Ableitung nehmen wir her und lassen die Schritte heraus, wo $p_i \to \bot$ benutzt wird. Es entsteht ein Beweis für $Y \vdash \bot$ oder für $Y \vdash p_i$. $Y \vdash \bot$ kann nicht sein, weil Y ein Modell hat, also $Y \vdash p_i$.

Damit ist 1. bewiesen, 2. zeigt man analog. □

Satz 2.6.22
Wenn Y eine endliche Menge von Klauseln ist, so ist der Schnittregelabschluß *$\{H \mid Y \vdash H\}$ von Y endlich.*

Beweis
Da Y eine endliche Menge von Klauseln ist, ist die Menge der AV, die in den Klauseln in Y vorkommen endlich. Daher ist die Menge der AV, die im Schnittregelabschluß von Y vorkommen, ebenfalls endlich. □

Wir haben nun ein

2.6.4 Syntaktisches Entscheidungsverfahren für ag

Der Algorithmus, der feststellt ob $H \in \mathrm{ag}$ ist, arbeitet wie folgt:

Schritt 1: Berechne eine endliche Menge $Y = \{K_1, \ldots, K_n\}$ von Klauseln derart, daß

$$\neg H \sim_{\mathrm{sem}} \bigwedge_{i=1}^{n} K_i$$

(Idee: eliminiere $\leftrightarrow$ und $\rightarrow$, treibe $\neg$ rein und klammere $\vee$ ein)

Schritt 2: Berechne $Y^\star = \{H^\star \mid Y \vdash H^\star\}$, den Schnittregelabschluß von Y

($Y^\star$ ist eine endliche Menge nach Satz 2.6.22)

Schritt 3: Wenn der Ausdruck $\bot$ Element von $Y^\star$ ist, dann ist $\mathrm{ag}H$, sonst ist H nicht allgemeingültig.

Rechtfertigung

$$\begin{array}{lll} \text{Es ist } H \in \mathrm{ag} & \text{gdw.} & \{\neg H\} \text{ kein Modell hat} \\ & \text{gdw.} & Y \text{ kein Modell hat} \\ & \text{gdw.} & \bot \in \mathrm{fl}(Y) \\ & \text{gdw.} & Y \vdash \bot \\ & \text{gdw.} & \bot \in Y^\star \end{array}$$

Beispiel

$$\begin{array}{rcl} H & = & p \rightarrow (q \rightarrow p) \\ \neg H & = & \neg\,(p \rightarrow (q \rightarrow p)) \end{array}$$

Berechnung von Y:

1. Elimination von $\rightarrow$:

$$\neg\,(p \rightarrow (\neg q \vee p))$$
$$\neg\,(\neg p \vee (\neg q \vee p))$$

2. $\neg$ vor die Variablen treiben, $\neg\neg$ weglassen

$$p \wedge \neg\,(\neg q \vee p)$$
$$p \wedge q \wedge \neg p$$

$Y = \{p, q, p \rightarrow \bot\}$

Offenbar Cut $(p, p \rightarrow \bot, \bot)$, also $\bot \in Y^\star$, d. h. $H \in \mathrm{ag}$.

2.7 Widerspruchsfreiheit, Vollständigkeit und Nichtableitbarkeit

2.7.1 Widerspruchsfreiheit

Widerspruchsfreiheit im Sinne des Wortes besteht dann, wenn in der gegebenen Menge von Aussagen „kein Widerspruch steckt“, d. h. aus ihr kein Widerspruch bewiesen

bzw. gefolgert werden kann. Für den Aussagenkalkül wird dies durch HILBERTs Definition der *klassischen Widerspruchsfreiheit* (bzgl. des Ableitens ab) gefaßt. Man kann schärfer die *semantische Widerspruchsfreiheit* mit der Forderung definieren, daß alles was beweisbar ist, auch wahr ist oder schwächer, daß nicht alles beweisbar sein darf (*syntaktische Widerspruchsfreiheit*).

Definition 2.7.1 (Widerspruchsfreiheit)

1. (GÖDEL)
   $X \subseteq \text{ausd}$ *heißt* semantisch widerspruchsfrei *genau dann, wenn* $\text{ab}(X) \subseteq \text{ag}$
2. (HILBERT)
   X *heißt* klassisch widerspruchsfrei *genau dann, wenn es keinen Ausdruck* $H \in \text{ausd}$ *mit* $H \in \text{ab}(X)$ *und* $\neg H \in \text{ab}(X)$ *gibt.*
3. (POST)
   X *heißt* syntaktisch widerspruchsfrei *genau dann, wenn* $\text{ab}(X) \neq \text{ausd}$.

Satz 2.7.2

1. *Wenn X semantisch widerspruchsfrei, so ist X klassisch widerspruchsfrei.*
2. *Wenn X klassisch widerspruchsfrei, so X syntaktisch widerspruchsfrei.*
3. *Die Umkehrungen gelten nicht.*

Beweis
der einzelnen Aussagen des Satzes

1. weil niemals H und $\neg H$ zugleich allgemeingültig
2. p, $\neg p$ sind nicht beide ableitbar, also $\text{ab}(X) \neq \text{ausd}$.
3. $X_1 = \{\neg(p \to p)\} \not\subseteq \text{ag}$. X_1 nicht semantisch widerspruchsfrei.
   $\text{ab}(X_1) = \{\neg(H \to H) \mid H \in \text{ausd}\}$ klassisch widerspruchsfrei.
   $X_2 = \{p \to p, \neg(p \to p)\}$ nicht klassisch widerspruchsfrei
   Behauptung: $p \notin \text{ab}(X_2)$. d. h. X_2 syntaktisch widerspruchsfrei.
   Beweis später (siehe unten), nicht ableitbar.

□

Satz 2.7.3 (Endlichkeitssatz für Widerspruchsfreiheit)
Die semantische/klassische/syntaktische Widerspruchsfreiheit hat finiten Charakter, d. h. eine Menge X ist genau dann semantisch bzw. klassisch bzw. syntaktisch widerspruchsfrei, wenn jede endliche Teilmenge $X^\star \subseteq X$ diese Eigenschaft hat.

Beweis
Wir zeigen zuerst: Wenn X die Eigenschaft hat, dann auch jede endliche Teilmenge $X^\star \subseteq X$.

semantisch:

$$\text{Wegen } X^\star \subseteq X \text{ ist } \text{ab}(X^\star) \subseteq \text{ab}(X) \subseteq \text{ag}$$

klassisch:
Wenn $X^\star$ nicht klassisch widerspruchsfrei ist, dann gibt es ein $H \in$ ausd mit $H, \neg H \in \mathrm{ab}(X^\star) \subseteq \mathrm{ab}(X)$, also ist X nicht klassisch widerspruchsfrei.

syntaktisch:

$$\text{Wegen } X^\star \subseteq X \text{ ist } \mathrm{ab}\,(X^\star) \subseteq \mathrm{ab}\,(X) \subset \text{ausd}$$

Jetzt setzen wir voraus, daß jede endliche Teilmenge $X^\star \subseteq X$ die Eigenschaft hat und zeigen, daß auch X die Eigenschaft hat.

semantisch:
Für $\mathrm{ab}(X) \subseteq \mathrm{ag}$ genügt es zu zeigen, daß $X \subseteq \mathrm{ag}$ ist. Sei $H \in X$, dann ist $\{H\} \subseteq X$, also $\mathrm{ab}\,(\{H\}) \subseteq \mathrm{ag}$, folglich $H \in \mathrm{ag}$ (wegen $H \in \mathrm{ab}\,(\{H\})$), also ist $X \subseteq \mathrm{ag}$, was zu zeigen war.

klassisch: indirekt
Nehmen wir an, daß ein $H \in$ ausd mit

$$H, \neg H \in \mathrm{ab}\,(X)$$

existiert, d. h. X ist nicht klassisch widerspruchsfrei.

Nach dem Endlichkeitssatz für das Ableiten existieren endliche $X_1^\star, X_2^\star \subseteq X$ mit

$$H \in \mathrm{ab}\,(X_1^\star) \quad \text{und} \quad \neg H \in \mathrm{ab}\,(X_2^\star)$$

Dann ist $X_1^\star \cup X_2^\star$ endliche Teilmenge von X und

$$H, \neg H \in \mathrm{ab}\,(X_1^\star \cup X_2^\star)\,.$$

im Widerspruch dazu, daß jede endliche Teilmenge von X klassisch widerspruchsfrei ist.

syntaktisch: indirekt
Nehmen wir an, daß X nicht syntaktisch widerspruchsfrei ist: $\mathrm{ab}\,(X) = \text{ausd}$.

Dann ist $p \in \text{ausd} = \mathrm{ab}\,(X)$, es gibt also eine endliche Teilmenge $X^\star \subseteq X$ mit

$$p \in \mathrm{ab}\,(X^\star)\,,$$

folglich ist $\mathrm{ab}\,(X^\star) = \text{ausd}$, also ist $X^\star$ nicht syntaktisch widerspruchsfrei. ↯

□

2.7.2 Vollständigkeit

Die Vollständigkeit einer Menge besagt die Umkehrung der Widerspruchsfreiheit, d. h.

semantisch: alles was wahr ist, ist beweisbar.

klassisch: von zwei Aussagen A und nicht A ist stets eine beweisbar.

syntaktisch: wenn eine nicht beweisbare Aussage als Axiom genommen wird, so ist die Widerspruchsfreiheit verletzt, d. h. alles wird beweisbar.

Definition 2.7.4 (Vollständigkeit)

1. $X \subseteq \text{ausd}$ *heißt* semantisch vollständig *genau dann, wenn* $\text{ab}(X) \supseteq \text{ag}$
2. $X \subseteq \text{ausd}$ *heißt* klassisch vollständig *genau dann, wenn für jedes* $H \in \text{ausd}$ *gilt* $H \in \text{ab}(X)$ *oder* $\neg H \in \text{ab}(X)$.
3. $X \subseteq \text{ausd}$ *heißt* syntaktisch vollständig *genau dann, wenn für jedes* $H \in \text{ausd}$ *gilt: Wenn* $H \notin \text{ab}(X)$, *so* $\text{ab}(X \cup \{H\}) = \text{ausd}$

Satz 2.7.5

1. axa *ist semantisch widerspruchsfrei und semantisch vollständig.*
2. *Es gibt im AK keine klassisch vollständige und klassisch widerspruchsfreie Menge* X.

Beweis der zweiten Behauptung
Wir zeigen:

> Wenn $X \subseteq \text{ausd}$ klassisch vollständig ist, so ist X nicht klassisch widerspruchsfrei

Sei X klassisch vollständig, dann ist

$$p \in \text{ab}(X) \quad \text{oder} \quad \neg p \in \text{ab}(X)$$

Fall 1: $p \in \text{ab}(X)$
Es ist $\text{ab}(X) = \text{ausd}$, da in p jeder Ausdruck eingesetzt werden darf, also ist X nicht syntaktisch widerspruchsfrei, also nach Satz (2.7.2.2) nicht klassisch widerspruchsfrei.

Fall 2: $\neg p \in \text{ab}(X)$
Es ist

$$\neg p\left(p/\neg p\right) = \neg\neg p \in \text{ab}(X)$$

also $\neg p, \neg\neg p \in \text{ab}(X)$, d. h. X ist ebenfalls nicht klassisch widerspruchsfrei.

□

Satz 2.7.6

1. *Wenn* X *semantisch vollständig, so ist* X *syntaktisch vollständig.*
2. *Umkehrung gilt nicht*

Beweis der ersten Behauptung
Es sei X semantisch vollständig, d. h. $\text{ag} \subseteq \text{ab}(X)$ und $H \notin \text{ab}(X)$. Folglich ist $H \notin \text{ag}$, also

$$\text{ab}(\text{ag} \cup \{H\}) = \text{ausd}.$$

Also

$$\text{ab}(X \cup \{H\}) = \text{ab}(\text{ab}(X \cup \{H\})) \supseteq \text{ab}(\text{ag} \cup \{H\}) = \text{ausd}$$

d. h. X ist syntaktisch vollständig. □

Beweis der zweiten Behauptung
Betrachten: $X = \{\neg p\}$.

$\text{ab}(X) = \{\neg H | H \in \text{ausd}\} \subset \text{ausd}, X$ syntaktisch widerspruchsfrei.

Nach Lindenbaum existiert eine Menge Y mit

1. $Y \supseteq X = \{\neg p\}$
2. Y ist syntaktisch widerspruchsfrei und syntaktisch vollständig
3. $\mathrm{ab}\,(Y) = Y$

Y ist nicht semantisch vollständig, d. h. $\mathrm{ag} \not\subseteq \mathrm{ab}\,(Y)$
sonst wäre $\mathrm{axa} \subseteq \mathrm{ab}\,(Y)$ und $\neg p \in Y \setminus \mathrm{axa} \subseteq Y$ d. h. $\mathrm{ab}\,(\mathrm{axa} \cup \{\neg p\}) = \mathrm{ausd} \subseteq Y$
d. h. Y – nicht syntaktisch widerspruchsfrei $\lightning$. □

2.7.3 Nichtableitbarkeit

Im Zusammenhang mit Satz 2.7.2 sind wir auf das Problem gestoßen

Wie beweist man $p \notin \mathrm{ab}\,(\{p \to p, \neg(p \to p)\})$?

allgemein:

Wie zeigt man $H \notin \mathrm{ab}(X)$?

Für den Spezialfall $X = \mathrm{axa}$ ist das leicht, denn wegen $\mathrm{ab}(\mathrm{axa}) = \mathrm{ag}$ handelt es sich dann um die Frage $H \notin \mathrm{ag}$, das können wir entscheiden.

Wenn wir statt des Ableitens das Folgern betrachten, d. h. die Frage $H \notin \mathrm{fl}(X)$? betrachten, so ist klar, wie sie zu beantworten ist: Wir müssen ein Modell von X finden, das H falsch macht, dann ist $H \notin \mathrm{fl}(X)$.

Diese Methode können wir zum Beweis von $H \notin \mathrm{ab}(X)$ nutzbar machen, wenn es uns gelingt, das Ableiten semantisch (durch das Folgern) zu charakterisieren.

Wir ersetzen die klassische logische Matrix $[\{\mathrm{W}, \mathrm{F}\}, \{\mathrm{W}\}, \mathrm{non}, \mathrm{et}, \mathrm{vel}, \mathrm{seq}, \mathrm{aeq}]$ durch eine Matrix $\mu = [M, M^\star, \Phi_1, \Phi_2, \ldots, \Phi_5]$ und definieren

- $\beta\mathrm{erf}_\mu H$ (über μ) genau dann, wenn $\mathrm{wert}_\mu\,(H, \beta) \in M^\star$.
- $\begin{array}{lcl} \mathrm{ag}_\mu & = & \text{Menge der } H \in \mathrm{ausd} \text{ mit: Für alle } \beta \in \mathfrak{B}_\mu \text{ ist } \mathrm{wert}_\mu\,(H, \beta) \in M^\star \\ & = & \mathrm{fl}_\mu\,(\emptyset). \end{array}$

Satz 2.7.7
$\mathrm{abe}\left(\mathrm{ag}_\mu\right) = \mathrm{ag}_\mu$ *für jede logische Matrix* μ

Beweis
Man zeigt durch Induktion über H:
Wenn $H = H_1\left(p/_{H^\star}\right)$ und $H_1 \in \mathrm{ag}_\mu$, so $H \in \mathrm{ag}_\mu$. □

Bemerkung
Für aba gibt es keinen äquivalenten Satz, betrachten wir als Beispiel die Matrix $\mu_0 = [\{\mathrm{W}, \mathrm{F}\}, \{\mathrm{W}\}, \mathrm{non}, \mathrm{et}, \mathrm{vel}, \Phi_4, \mathrm{aeq}]$ mit $\Phi_4\,(w, w') \equiv \mathrm{W}$. Hierbei ist $(p \to p) \in \mathrm{ag}_{\mu_0}$ und $((p \to p) \to p) \in \mathrm{ag}_{\mu_0}$, d. h. $p \in \mathrm{aba}\left(\mathrm{ag}_{\mu_0}\right)$, aber $p \notin \mathrm{ag}_{\mu_0}$.

Hinreichend für die Widerspruchsfreiheit der Abtrennungsregel:

$$\text{Wenn } w \in M^\star \text{ und } \Phi_4\,(w, w') \in M^\star, \text{ so } w' \in M^\star.$$

$$\text{d. h. wenn } w \in M^\star \text{ und } w' \in M \setminus M^\star, \text{ so } \Phi_4\,(w, w') \in M \setminus M^\star \qquad (\star)$$

Definition 2.7.8
Eine logische Matrix μ heißt normal*, wenn Φ_4 die Bedingung $(\star)$ erfüllt.*

Folgerung 2.7.9
Wenn μ eine normale logische Matrix ist, so ist $\mathrm{ab}\left(\mathrm{ag}_\mu\right) = \mathrm{ag}_\mu$.

Satz 2.7.10 (Satz von Lindenbaum)
Zu jeder Menge $X \subseteq \mathrm{ausd}$ gibt es eine höchstens abzählbare normale logische Matrix μ mit $\mathrm{ab}\,(X) = \mathrm{ag}_\mu$.

Beweis
Wir betrachten die Matrix $\mu = [M, M^\star, \Phi_1, \ldots, \Phi_5]$ mit

$$\begin{array}{rcl} M & := & \mathrm{ausd} \\ M^\star & := & \mathrm{ab}\,(X) \\ \Phi_1\,(H) & := & \neg H \\ \Phi_2\,(H_1, H_2) & := & (H_1 \wedge H_2) \\ \Phi_3\,(H_1, H_2) & := & (H_1 \vee H_2) \\ \Phi_4\,(H_1, H_2) & := & (H_1 \rightarrow H_2) \\ \Phi_5\,(H_1, H_2) & := & (H_1, \leftrightarrow H_2) \end{array}$$

Die Menge M ist abzählbar unendlich. Wir zeigen, daß μ normal ist, d. h.

$$\text{wenn } H \in \mathrm{ab}\,(X) \text{ und } \Phi_4\,(H, H') = (H \rightarrow H') \in \mathrm{ab}\,(X)\,, \text{ so } H' \in \mathrm{ab}\,(X)$$

das gilt nach Definition von ab (Abtrennungsregel).
Es bleibt zu zeigen, daß gilt:

$$\mathrm{ab}\,(X) \quad = \quad \mathrm{ag}_\mu$$

Es sei $H \in \mathrm{ag}_\mu$, d. h. $\mathrm{wert}_\mu\,(H, \beta) \in \mathrm{ab}\,(X)$ für alle $\beta \in \mathfrak{B}_\mu$.
Wir betrachten die Belegung $\beta^\star \in \mathfrak{B}_\mu$ mit

$$\beta^\star\,(p_i) \quad := \quad p_i \in \mathrm{ausd}$$

Offenbar ist

$$\mathrm{wert}_\mu\,(H, \beta^\star) \quad = \quad H$$

also ist $H \in \mathrm{ab}\,(X)$.
Es sei umgekehrt $H \in \mathrm{ab}\,(X)$, dann ist

$$\mathrm{abe}\,(\{H\}) \quad \subseteq \quad \mathrm{ab}\,(X)$$

Es sei β eine beliebige Belegung der Aussagenvariablen mit Ausdrücken. Dann ist (wenn genau $p_1, \ldots, p_n$ in H vorkommen)

$$\mathrm{wert}_\mu\,(H, \beta) = H\left[p_1/_{\beta(p_1)}, \ldots, p_n/_{\beta(p_n)}\right] \in \mathrm{abe}\,(\{H\}) \subseteq \mathrm{ab}\,(X)$$

d. h. $H \in \mathrm{ag}_\mu$. □

Bemerkung: Es geht im allgemeinen nicht mit einer endlichen Matrix.

Satz 2.7.11 (Semantische Charakterisierung von ab)
Es ist $H \in \mathrm{ab}(X)$ *genau dann, wenn für jede normale höchstens abzählbare logische Matrix* μ *mit* $X \subseteq \mathrm{ag}_\mu$ *stets* $H \in \mathrm{ag}_\mu$.

Beweis

($\Rightarrow$) Es sei $X \mathrm{abl} H$ ($H \in \mathrm{ab}(X)$) μ–normal $X \subseteq \mathrm{ag}_\mu$
Dann ist $H \in \mathrm{ab}(X) \subseteq \mathrm{ag}_\mu$ (weil μ–normal) also $H \in \mathrm{ag}_\mu$

($\Leftarrow$)

$$H \in \bigcap_{\substack{\mu\text{–normal} \\ X \subseteq \mathrm{ag}_\mu}} \mathrm{ag}_\mu$$

Nach Satz von Lindenbaum gibt es $\mu^\star$ mit $\mathrm{ab}(X) = \mathrm{ag}_{\mu^\star}$. Dabei $X \subseteq \mathrm{ag}_{\mu^\star}$. Folglich $H \in \mathrm{ag}_{\mu^\star} = \mathrm{ab}(X)$. d. h. $H \in \mathrm{ab}(X)$.

□

Folgerung 2.7.12 (Satz über Nichtableitbarkeit)
$H \notin \mathrm{ab}(X)$ *genau dann, wenn es eine höchstens abzählbare logische Matrix* μ *mit* $X \subseteq \mathrm{ag}_\mu$ *und* $H \notin \mathrm{ag}_\mu$ *gibt.*

Beispiel

$$p \notin \mathrm{ab}(\{p \to p, \neg(p \to p)\})$$
$$\mu = [\{W, F\}, \{W\}, \Phi_1, \mathrm{et}, \mathrm{vel}, \mathrm{seq}, \mathrm{aeq}]$$
$$\Phi_1 : \Phi_1(W) = W, \Phi_1(F) = F.$$

μ normal, da $\Phi_4 = \mathrm{seq}$.

$$\mathrm{wert}_\mu(p \to p, \beta) = \mathrm{seq}(\beta(p), \beta(p)) = W \in \{W\}$$
$$p \to p \in \mathrm{ag}_\mu$$
$$\mathrm{wert}_\mu(\neg(p \to p), \beta) = \Phi_1(\mathrm{wert}(p \to p, \beta)) = \mathrm{wert}_\mu(p \to p, \beta) \in \{W\}$$

aber $\beta^\star(p) = F \to \mathrm{wert}_\mu(p, \beta^\star) = F$ d. h. $p \notin \mathrm{ag}_\mu$.

Satz 2.7.13
axa *ist unabhängig, d. h. wenn* $H \in \mathrm{axa}$, *so* $\mathrm{ab}(\mathrm{axa} \setminus \{H\}) \neq \mathrm{ag}$ *und es gilt:* $H \notin \mathrm{ab}(\mathrm{axa} \setminus \{H\})$.

2.8 Dualitätstheoreme

Mit den im Aussagenkalkül entwickelten Hilfsmitteln kann nun der Beweis des Dualitätstheorems für Mengenterme angegangen werden. Dazu geben wir eine *induktive* Definition (im Gegensatz zur Erzeugung durch eine Grammatik — siehe Seite 47) für Mengenvariablen, –terme und –gleichungen an:

Alphabet: $A = \{M, \star, \cup, \cap, \bar{\ }, =, (,)\}$

Mengenvariable:

A) M ist eine Mengenvariable.

I) Wenn das Wort $\mathcal{Z}$ eine Mengenvariable ist, so auch $\mathcal{Z}\star$.

E) Sonst nichts.

$$\begin{array}{lcllll} MV & := & \{ & M, & M\star, & M\star\star, & \ldots & \} \\ & & & = & = & = & \text{usw.} & \\ & & & M_1 & M_2 & M_3 & & \end{array}$$

Mengenterme:

A) Jede Mengenvariable ist ein Mengenterm

I) 1. Ist T_1 ein Mengenterm, so ist $\overline{T_1}$ ein Mengenterm
   2. Sind T_1, T_2 Mengenterme, so sind $(T_1 \cup T_2)$, $(T_1 \cap T_2)$ Mengenterme.

E) Sonst nichts.

Gleichungen: Sind T_1, T_2 Mengenterme, so heißt $T_1 = T_2$ Gleichung.

Weiterhin definieren wir:

Belegung: Eine Abbildung $f : \{M_0, \ldots\} \to \mathfrak{P}(E)$ heißt Belegung.

Set: Die von einem Term T bei einer Belegung f beschriebene Menge $\mathrm{Set}(T, f)$

A) $\mathrm{Set}(M_i, f) := f(M_i)$

I) $\mathrm{Set}\left(\overline{T}, f\right) := \overline{\mathrm{Set}(T, f)} = E \setminus \mathrm{Set}(T, f)$
$\mathrm{Set}\left(T_1 \begin{smallmatrix}\cup\\ \cap\end{smallmatrix} T_2, f\right) = \mathrm{Set}(T_1, f) \begin{smallmatrix}\cup\\ \cap\end{smallmatrix} \mathrm{Set}(T_2, f)$

Gültigkeit: Eine Gleichung $G \equiv T_1 = T_2$ heißt gültig gdw. für alle Belegungen f $\mathrm{Set}(T_1, f) = \mathrm{Set}(T_2, f)$ ist.

Satz 2.8.1 (Dualitätstheorem für Mengengleichungen)
Ist G eine gültige Gleichung und entsteht G' aus G indem gleichzeitig an allen Stellen „$\cap$“ durch „$\cup$“ und „$\cup$“ durch „$\cap$“ ersetzt wird, dann ist G' gültig.

Beispiel

$$\begin{array}{rcl} (M_0 \cup M_1) & = & \overline{\left(\overline{M_0} \cap \overline{M_1}\right)} \\ & \downarrow & \\ (M_0 \cap M_1) & = & \overline{\left(\overline{M_0} \cup \overline{M_1}\right)} \end{array}$$

Beweis durch Reduktion auf das Dualitätstheorem der Aussagenlogik
Zu jedem Term T konstruieren wir einen Ausdruck H_T:

A) $T \equiv M_i \quad H_T \equiv p_i$

I) $T \equiv \overline{T'} \qquad H_T \equiv \neg H_{T'}$
$T \equiv (T_1 \cup T_2) \quad H_T \equiv (H_{T_1} \vee H_{T_2})$
$T \equiv (T_1 \cap T_2) \quad H_T \equiv (H_{T_1} \wedge H_{T_2})$

Hilfssatz
Sei T Mengenterm, $a \in E$ und f Belegung, ferner

$$\beta_{a,f}(p_i) := \begin{cases} \mathrm{W}, & \textit{falls } a \in f(M_i) \\ \mathrm{F}, & \textit{sonst} \end{cases}$$

Dann gilt:

$$a \in \mathrm{Set}(T, f) \ \textit{ gdw. } \mathrm{wert}(H_T, \beta_{a,f}) = \mathrm{W}$$

Beweis des Hilfssatzes
durch Induktion über T

A) $T = M_i$
$a \in \mathrm{Set}(M_i, f) = f(M_i)$ gdw. $\mathrm{wert}(p_i, \beta_{a,f}) = \mathrm{W}$ gilt.

I)
$$\begin{array}{rll} a \in \mathrm{Set}\left(\overline{T}, f\right) & \text{gdw.} & a \in \overline{\mathrm{Set}(T, f)} \\ & \text{gdw.} & \text{nicht } a \in \mathrm{Set}(T, f) \\ & \text{gdw.} & \text{nicht wert}(H_T, \beta_{a,f}) = \mathrm{W} \\ & \text{gdw.} & \mathrm{wert}(H_T, \beta_{a,f}) = \mathrm{F} \\ & \text{gdw.} & \mathrm{wert}(\neg H_T, \beta_{a,f}) = \mathrm{W} \\ & \text{gdw.} & \mathrm{wert}\left(H_{\overline{T}}, \beta_{a,f}\right) = \mathrm{W} \\ a \in \mathrm{Set}(T_1 \cap T_2, f) & \text{gdw.} & a \in \mathrm{Set}(T_1, f) \text{ und } a \in \mathrm{Set}(T_2, f) \\ & \text{gdw.} & \mathrm{wert}(H_{T_1}, \beta_{a,f}) = W \text{ und wert}(H_{T_2}, \beta_{a,f}) = \mathrm{W} \\ & \text{gdw.} & \mathrm{wert}(H_{T_1} \wedge H_{T_2}, \beta_{a,f}) = \mathrm{W} \\ & \text{gdw.} & \mathrm{wert}(H_{T_1 \cap T_2}, \beta_{a,f}) = \mathrm{W} \end{array}$$

analog für $T = T_1 \cup T_2$. □

Satz 2.8.2
$T_1 = T_2$ gültig gdw. H_{T_1}, H_{T_2} semantisch äquivalent.

Beweis
Es ist $T_1 = T_2$ gültig gdw.

$$\begin{array}{c} \text{für alle } f : \mathrm{Set}(T_1, f) = \mathrm{Set}(T_2, f) \\ \updownarrow \\ \text{für alle } f, \text{ alle } a \in E : a \in \mathrm{Set}(T_1, f) \text{ gdw. } a \in \mathrm{Set}(T_2, f) \\ \updownarrow \\ \text{für alle } \beta : \mathrm{wert}(H_{T_1}, \beta) = \mathrm{W} \text{ gdw. } \mathrm{wert}(H_{T_2}, \beta) = \mathrm{W} \end{array}$$

□

Satz 2.8.3 (Dualitätstheorem der Aussagenlogik)
Es seien H_1, H_2 Ausdrücke, in denen die Zeichen $\rightarrow, \leftrightarrow$ nicht vorkommen, ferner sei H_i' der aus H_i durch simultanes Ersetzen von $\wedge$ durch $\vee$ und von $\vee$ durch $\wedge$ entstehende Ausdruck. Dann gilt:

$$\textit{Wenn } \mathrm{ag}(H_1 \leftrightarrow H_2), \textit{ so } \mathrm{ag}(H_1' \leftrightarrow H_2').$$

Offenbar gilt dann:

$$\begin{aligned} G \equiv T_1 = T_2 \text{ gültig} \quad & \to \operatorname{ag}\left(H_{T_1} \leftrightarrow H_{T_2}\right) \\ & \qquad\qquad\qquad\qquad \leftarrow \text{Dualität} \\ & \to \operatorname{ag}\left(H'_{T_1} \leftrightarrow H'_{T_2}\right) \\ & \to T'_1 = T'_2 \text{ gültig.} \end{aligned}$$

Damit haben wir das Dualitätstheorem für Mengenterme bewiesen. □

Beweis des Dualitätstheorems der Aussagenlogik
Wir zeigen: Wenn $H_1 \to H_2 \in \operatorname{ag}$, so $H'_2 \to H'_1 \in \operatorname{ag}$. Es seien o. B. d. A. $p_1, \ldots, p_n$ alle AV in $H_1 \to H_2$. Dann ist

$$(H_1 \to H_2)\left[p_1/\neg p_1, \ldots, p_n/\neg p_n\right] = (H_1^\star \to H_2^\star) \in \operatorname{ag}$$

wobei $H_i^\star = H_i\left[p_1/\neg p_1, \ldots, p_n/\neg p_n\right]$.
Nun ist $\neg\left(H_0 \wedge H_{00}\right) \sim_{\text{sem}} \neg H_0 \vee \neg H_{00}$ und $\neg\left(H_0 \vee H_{00}\right) \sim_{\text{sem}} \neg H_0 \wedge \neg H_{00}$.
Mittels dieser Äquivalenz treiben wir die Negationszeichen nach außen. ($\leftarrow$ anwenden von rechts nach links) Dabei geht $H_1^\star$ in $\neg H'_1$, $H_2^\star$ in $\neg H'_2$ über, wir erhalten also

$$(\neg H'_1 \to \neg H'_2) \in \operatorname{ag}$$

also $(H'_2 \to H'_1) \in \operatorname{ag}$ (Kontraposition). □

Beispiel

$$\begin{gathered} M = M \cup (M \cap N) \qquad \text{Absorptionsgesetz} \\ \text{übersetzen} \to \\ p \leftrightarrow p \wedge (p \vee q) \\ p \leftrightarrow p \vee (p \wedge q) \\ \to \\ M = M \cap (M \cup N). \end{gathered}$$

Kapitel 3

Berechnungstheorie

Warum unternehmen wir an dieser Stelle einen Abstecher in die Berechnungstheorie, d. h. in die Theorie der Algorithmen? Bisher sind wir doch mit dem intuitiven Algorithmenbegriff gut ausgekommen, z. B. als wir zeigten, daß es einen Algorithmus gibt, der auf jeden Ausdruck H des Aussagenkalküls angewendet werden kann und stets nach endlich vielen Schritten terminiert mit einem der Resultate „$H \in$ ag“ oder „$H \notin$ ag“ und mit „$H \in$ ag“ genau dann endet, wenn H allgemeingültig ist. Es zeigt sich aber, daß ein solcher Algorithmus zur Entscheidung der Gültigkeit im Prädikatenkalkül (ab Seite 123) nicht gefunden wurde und es stellt sich damit die Frage nach einem Beweis dafür, daß ein solcher Algorithmus nicht existiert. Ein exakter Beweis kann nur auf der Basis exakt definierter Begriffe geführt werden; wir müssen also den Algorithmenbegriff exakt definieren, d. h. Theorie der Algorithmen betreiben.

Obwohl hier die Logik zur Motivierung der Algorithmentheorie angeführt wird (sie ist aus dem Bedürfnis der Logik entstanden), bildet die Algorithmentheorie heute den Kern der Informatik, auch die Definition einer beliebigen Programmiersprache (und ihrer Semantik) ist nichts anderes als die Festlegung eines Algorithmenbegriffs. Insofern kann die Theorie der Programmierung als Teilgebiet der Algorithmentheorie bzw. als angewandte Algorithmentheorie verstanden werden.

Betrachten wir z. B. folgendes praktisches Problem: Sie haben ein Programm geschrieben und es mit einem Datensatz gestartet. Auf dem Bildschirm passiert nichts. Nach einiger Zeit des Wartens fragen Sie sich: Ist das Programm in eine Endlosschleife geraten (durch einen Programmierfehler) oder dauert die Berechnung wirklich so lange? Es wäre doch schön, wenn man ein Programm hätte, das einen Programmtext und einen Datensatz einliest und dann die Frage beantwortet, ob das eingelesene Programm bei dem eingelesenen Datensatz terminiert. Die Algorithmentheorie beweist, daß ein solches Programm nicht existiert, das „Halteproblem“ ist nicht entscheidbar.

Neben solchen prinzipiellen Fragen („existiert ein Algorithmus, der ...“) behandelt die Berechnungstheorie aber auch die Frage nach der Qualität von Algorithmen, die durch Laufzeit- und Speicherplatz–Bedarf gemessen wird.

3.1 Grundbegriffe

Jeder kennt eine Vielzahl von Algorithmen (Rechenverfahren, Bedienungsanleitungen, Kochrezepte, ...) und hat daraus einen intuitiven Begriff des Machbaren (mit endlichen Mitteln und in endlicher Zeit) abstrahiert. Dieser Begriff reicht aus, um zu verifizieren, daß ein angegebenes Verfahren ein bestimmtes Problem löst (z. B. das

Entscheidungsproblem im Aussagenkalkül). Dieser Begriff bietet aber keinen Ansatzpunkt für einen Beweis dessen, daß ein Problem unlösbar ist.

Gibt es überhaupt Probleme, die algorithmisch unlösbar sind? Betrachten wir als Beispiel Entscheidungsprobleme für Sprachen; wir fixieren ein endliches Alphabet X und betrachten alle Sprachen über X, also $\mathfrak{P}(W(X))$. Die Kardinalzahl dieser Menge ist größer als $\aleph_0$, die Kardinalzahl der Menge $\mathbb{N}$.

Zu jeder Sprache $L \subseteq W(X)$ gehört das Entscheidungsproblem von L, d. h. die Menge

$$\{p \in L?|p \in W(X)\}$$

der Aufgaben festzustellen, ob ein spezifisches Wort p zu L gehört oder nicht.

Ein Algorithmus löst dieses Problem, wenn er auf jedes Wort $p \in W(X)$ angewendet werden kann und seine Anwendung stets nach endlich vielen Schritten mit der richtigen Antwort auf die Frage $p \in L?$ terminiert.

Ein Algorithmus selbst hat eine endliche Beschreibung, etwa durch ein Wort über einem endlichen Alphabet Y. Die Menge der Wörter über Y kann bijektiv auf $\mathbb{N}$ abgebildet werden, ist also abzählbar unendlich, d. h. es gibt „weniger“ Algorithmen als Entscheidungsprobleme, es muß also unlösbare Probleme geben.

Der intuitive Ansatz bietet, wie gesagt, keinen Ansatz für einen exakten Beweis dafür, daß ein bestimmtes Problem unlösbar ist. Dazu muß der Algorithmenbegriff präzisiert werden. Jede Präzisierung bringt aber unvermeidlich eine Einschränkung der zugelassenen Vorgehensweisen mit sich, so daß nach einem Unlösbarkeitsbeweis die Frage entsteht, ob es an der Präzisierung oder am Problem liegt, daß keine Lösung existiert.

Dieser Frage ist man dadurch entkommen, daß man den Algorithmenbegriff mehrfach in unterschiedlicher Weise präzisiert und die Gleichwertigkeit der verschiedenen Präzisierungen bewiesen hat. Diese Ansätze können wie folgt sortiert werden:

1. Axiomatischer Ansatz (Church, Kleene)
   Man erklärt gewisse Funktionen per Axiom für berechenbar und definiert Operationen mit Funktionen, die per definitionem die Berechenbarkeit erhalten.

2. Maschinen–Ansatz (Turing)
   Man definiert einen Typ von Rechenmaschinen und erklärt als berechenbar genau das, was diese Maschinen berechnen können.

3. Semiotischer Ansatz (Markow, Post)
   Rechnen bzw. algorithmisches Vorgehen wird als Manipulation von Zeichenreihen (Wörtern) nach gewissen Regeln verstanden.

4. Beweistheoretischer Ansatz (Gödel)
   Das Berechnen eines Funktionswertes $k = f(i)$ wird als Beweisen der Gleichung $f(i) = k$ verstanden.

Wir werden uns im folgenden nur mit dem axiomatischen und dem Maschinen–Ansatz befassen.

Was ist ein Problem?

Als Problem bezeichnen wir eine Familie von Aufgaben, die von einem oder mehreren Parametern abhängen. Dabei unterscheiden wir Berechnungsprobleme

$$\{f(x_1, \ldots, x_n) =? \mid x_1, \ldots, x_n \in \mathbb{N}\}$$

und Entscheidungsprobleme, wie z. B.

$$\{H \in \text{ag}?|H \in \text{ausd}\}$$

In dem einen Fall geht es um die Berechnung eines Wertes (z. B. einer natürlichen Zahl), im anderen um die Bestimmung eines Wahrheitswertes.

Das Berechnungsproblem $\{f(x_1, \ldots, x_n) =? \mid x_1, \ldots, x_n \in \mathbb{N}\}$ der Funktion f heißt *lösbar* bzw. die zahlentheoretische Funktion f heißt *berechenbar*, wenn es einen Algorithmus gibt, der für jedes $x_1, \ldots, x_n \in \mathbb{N}$ folgendes leistet:

Wenn f für $[x_1, \ldots, x_n]$ definiert ist, dann stoppt die Anwendung des Algorithmus auf $[x_1, \ldots, x_n]$ nach endlich vielen Schritten mit dem Wert $f(x_1, \ldots, x_n)$ als Resultat; wenn f für $[x_1, \ldots, x_n]$ nicht definiert ist, dann stoppt die Anwendung des Algorithmus niemals.

Wir verlangen also *nicht*, daß der Algorithmus erkennt, daß f für $[x_1, \ldots, x_n]$ nicht definiert ist und mit einer Fehlermeldung stoppt.

Jedes Entscheidungsproblem, also auch

$$\{H \in \text{ag}? \mid H \in \text{ausd}\},$$

kann auf ein Berechnungsproblem zurückgeführt werden; dazu geht man zur charakteristischen Funktion der zu entscheidenden Menge über, hier:

$$\chi_{\text{ag}}(H) = \begin{cases} 0, \text{falls } H \in \text{ag} \\ 1, \text{sonst} \end{cases}$$

und hat das Problem

$$\{\chi_{\text{ag}}(H) =? \mid H \in \text{ausd}\}$$

zu lösen. Durch eineindeutige Numerierung der Menge aller Ausdrücke kommt man dazu, daß man nur die Berechenbarkeit zahlentheoretischer Funktionen untersuchen muß

3.2 Axiomatische Charakterisierung der Berechenbarkeit

Einer der historisch ersten Ansätze, den Berechenbarkeitsbegriff zu fassen, definiert induktiv eine Klasse von Funktionen, indem zunächst einige einfache Anfangsfunktionen angegeben werden, die per Definition in der Klasse enthalten sind. Sodann werden verschiedene Prinzipien angegeben, wie aus Funktionen, die in der Klasse liegen, neue Funktionen gewonnen werden können.

Wir nennen eine n-stellige zahlentheoretische Funktion *total*, wenn sie für alle n-Tupel von natürlichen Zahlen definiert ist.

3.2.1 Anfangsfunktionen

Anfangsfunktionen
ANF = Menge der folgenden zahlentheoretischen Funktionen

1. $N(x) = x + 1$ (Nachfolgerfunktion)
2. $C_i^n(x_1, \ldots, x_n) = i$ (n-stellige Konstanten)
3. $I_k^n(x_1, \ldots, x_n) = x_k$ (Argumentauswahl)

Satz 3.2.1
Alle Funktionen aus ANF *sind intuitiv–anschaulich berechenbar (und total).*

3.2.2 Substitution und primitive Rekursion

Substitution
Die Funktion $\phi^{(n)}$ entsteht durch Substitution aus $\psi^{(m)}$ und $\chi_1^{(n)}, \ldots, \chi_m^{(n)}$ gdw.

$$\phi^{(n)}(x_1, \ldots, x_n) = \psi^{(m)}\left(\chi_1^{(n)}(x_1, \ldots, x_n), \ldots, \chi_m^{(n)}(x_1, \ldots, x_n)\right)$$

Die Substitution kann auch auf *partielle* (d. h. nicht totale) Funktionen angewendet werden. Die Funktion $\phi^{(n)}$ ist für $[x_1, \ldots, x_n]$ (durch die angegebene Gleichung) definiert genau dann, wenn $\chi_1^{(n)}, \ldots, \chi_m^{(n)}$ für $[x_1, \ldots, x_n]$ definiert sind und $\psi^{(m)}$ für das m-Tupel $\left[\chi_1^{(n)}(x_1, \ldots, x_n), \ldots, \chi_m^{(n)}(x_1, \ldots, x_n)\right]$ definiert ist.

Satz 3.2.2
Sind $\psi^{(m)}$ und $\chi_1^{(n)}, \ldots, \chi_m^{(n)}$ intuitiv–anschaulich berechenbare Funktionen und entsteht $\phi^{(n)}$ durch Substitution von $\psi^{(m)}$ und $\chi_1^{(n)}, \ldots, \chi_m^{(n)}$, so ist $\phi^{(n)}$ intuitiv–anschaulich berechenbar. Sind dabei $\psi^{(m)}, \chi_1^{(n)}, \ldots, \chi_m^{(n)}$ total, so ist auch $\phi^{(n)}$ total.

Da wir die Substitution als Erzeugungsoperation verwenden werden, können wir auch mit einer kleineren Menge von Anfangsfunktionen auskommen, nämlich von den Konstanten C_i^n nur die Konstanten C_0^n als Anfangsfunktion behalten, die übrigen können ja durch Substitution aus C_0^n und der Nachfolgerfunktion erzeugt werden:

$$\begin{aligned} C_1^n(x_1, \ldots x_n) &= N(C_0^n(x_1, \ldots, x_n)) \\ C_2^n(x_1, \ldots x_n) &= N(N(C_0^n(x_1, \ldots, x_n))) \\ &\vdots \end{aligned}$$

Induktive Definition, die primitive Rekursion
Die Funktion $\phi^{(n+1)}$ entsteht durch primitive Rekursion aus $\psi^{(n)}$ und $\chi^{(n+2)}$, wenn $\phi^{(n+1)}$ die Lösung des folgenden Gleichungssystems ist:

$$\begin{aligned} \phi^{(n+1)}(x_1, \ldots, x_n, 0) &= \psi^{(n)}(x_1, \ldots, x_n) \\ \phi^{(n+1)}(x_1, \ldots, x_n, i+1) &= \chi^{(n+2)}\left(x_1, \ldots, x_n, i, \phi^{(n+1)}(x_1, \ldots, x_n, i)\right). \end{aligned}$$

Sind $\psi^{(n)}$ und $\chi^{(n+2)}$ partielle Funktionen, so wird i. a. auch $\phi^{(n+1)}$ partiell sein. Auch hier pflanzt sich das Nicht–Definiert–Sein fort, ist z. B. $\phi^{(n+1)}(x_1, \ldots, x_n, j)$ nicht definiert, so auch $\phi^{(n+1)}(x_1, \ldots, x_n, j+1)$. Der folgende Satz verallgemeinert den Dedekindschen Rechtfertigungssatz (vgl. S. 44).

Satz 3.2.3
Sind $\psi^{(n)}$ und $\chi^{(n+2)}$ gegebene Funktionen, so existiert genau ein $\phi^{(n+1)}$, welches das Schema erfüllt.

Satz 3.2.4
Sind $\psi^{(n)}, \chi^{(n+2)}$ intuitiv–anschaulich berechenbar und $\phi^{(n+1)}$ entsteht durch primitive Rekursion aus $\psi^{(n)}$ und $\chi^{(n+2)}$, so ist ϕ intuitiv–anschaulich berechenbar. Sind $\psi^{(n)}, \chi^{(n+2)}$ total, so ist auch $\phi^{(n+1)}$ total.

primitiv-rekursive Funktionen
PRM = die kleinste Klasse von zahlentheoretischen Funktionen, die ANF enthält und abgeschlossen bezüglich Substitution und primitiver Rekursion ist.

Satz 3.2.5
Die primitiv-rekursiven Funktionen sind total.

3.2.3 Beispiele für primitiv–rekursive Funktionen

Die folgenden zahlentheoretischen Funktionen sind primitiv–rekursiv:

- Addition $m + n$
- Multiplikation $m \cdot n$
- Vorgänger $V(n) := \begin{cases} n-1, & n > 0 \\ 0, & \text{sonst} \end{cases}$
- Arithmetische Differenz

$$m \dot{-} n := \begin{cases} m-n, & m \geq n \\ 0, & \text{sonst} \end{cases}$$

$$\begin{aligned} m \dot{-} 0 &= m \\ m \dot{-} (n+1) &= V\left(m \dot{-} n\right) \end{aligned}$$

- Betrag einer Differenz

$$|m-n| = \left(m \dot{-} n\right) + \left(n \dot{-} m\right)$$

- sg (signum)

$$\begin{aligned} \mathrm{sg}(0) &= 0 \\ \mathrm{sg}(m+1) &= 1 \end{aligned}$$

- $\overline{\mathrm{sg}}$

$$\overline{\mathrm{sg}}(n) = 1 \dot{-} \mathrm{sg}(n)$$

- Allgemeine Summe
  Sei g eine $(n+1)$-stellige primitiv–rekursive Funktion.
  Dann ist

$$f(m_1, \ldots, m_n, k) \quad := \quad \sum_{i=0}^{k} g(m_1, \ldots, m_n, i)$$

  primitiv–rekursiv.
  Bemerkung:

$$\begin{aligned} f(m_1, \ldots, m_n, 0) &= g(m_1, \ldots, m_n, 0) \\ f(m_1, \ldots, m_n, k+1) &= f(m_1, \ldots, m_n, k) + g(m_1, \ldots, m_n, k+1) \end{aligned}$$

- allgemeines Produkt
  Sei g wie oben, dann ist

$$f(m_1, \ldots, m_n, k) = \prod_{i=0}^{k} g(m_1, \ldots, m_n, i)$$

  primitiv-rekursiv.

- Fakultät

$$0! = 1$$
$$(n+1)! = n! \cdot (n+1)$$

- Division mit Rest

$$\left[\frac{x}{y}\right] := \begin{cases} 0, & y = 0 \\ x\mathrm{DIV}y, & \text{sonst} \end{cases}$$

$$\left[\frac{x}{y}\right] = \mathrm{sg}(y) \cdot \sum_{j=0}^{x} \mathrm{sg}\left(\prod_{i=0}^{j} \left[(x+1) \dot{-} (i+1) \cdot y\right]\right)$$

- Rest der Division

$$\mathrm{rest}\,(x, y) = x \dot{-} \left[\frac{x}{y}\right] \cdot y$$

Definition 3.2.6 (Beschränkter Minimum–Operator)
Es sei $g^{(n+1)}$ eine $(n+1)$-stellige totale zahlentheoretische Funktion.

Für $m_1, \ldots, m_n, k \in \mathbb{N}$ sei

$$f\,(m_1, \ldots, m_n, k) := \begin{cases} \textit{die kleinste Zahl } i \textit{ mit } i \leq k \textit{ und } g\,(m_1, \ldots, m_n, i) = 0, \textit{ falls ein solches } i \textit{ existiert} \\ 0, \textit{ sonst} \end{cases}$$

Wir notieren die Funktion f durch

$$f(m_1, \ldots, m_n, k) = \mu i\,(i \leq k \wedge g(m_1, \ldots, m_n, i) = 0)$$

und sagen, daß sie aus g durch Anwendung des beschränkten Minimumoperators *entsteht.*

Satz 3.2.7
Die Anwendung des beschränkten Minimumoperators auf eine primitiv–rekursive Funktion g erzeugt eine primitiv–rekursive Funktion f (führt nicht aus der Klasse PRM *heraus).*

Beweis
Man überlegt sich sofort, daß folgendes gilt

$$f\,(m_1, \ldots, m_n, k) = \left[\sum_{i=0}^{k} \mathrm{sg}\left(\prod_{j=0}^{i} g\,(m_1, \ldots, m_n, j)\right)\right] \cdot \overline{\mathrm{sg}}\left(\left[\sum_{i=0}^{k} \mathrm{sg}\left(\prod_{j=0}^{i} g\,(m_1, \ldots, m_n, j)\right)\right] \dot{-}\, k\right)$$

□

Folgerung 3.2.8
Die Funktion $q\,(n) :=$ die n-te Primzahl *und* $\exp(x, n) :=$ der Exponent der n-ten Primzahl in der Primzahldarstellung von x *sind primitiv–rekursiv.*

Beweis

$$\begin{aligned} q(0) &= 2 = C_2^0 \\ q(n+1) &= \mu j \left(j \leq (q(n)!+1) \wedge \left(q(n)+1 \dot{-} j \right) + \sum_{k=2}^{j-1} \overline{\mathrm{sg}}\,(\mathrm{rest}\,(j,k)) = 0 \right). \\ \exp(x,n) &= \sum_{j=1}^{x} \overline{\mathrm{sg}}\big(\mathrm{rest}\,\big(x, q(n)^j\big)\big). \end{aligned}$$

3.2.4 Die Peterfunktion

Offensichtlich sind die primitiv-rekursiven Funktionen nicht alle berechenbaren, denn z. B. die partiellen Funktionen gehören nicht zu PRM. Es gibt aber auch totale Funktionen, die intuitiv–anschaulich berechenbar, aber nicht primitiv–rekursiv sind.

Ackermann gab 1928 eine solche nicht primitiv–rekursive, aber totale und berechenbare Funktion an. In der später von R. Peter vereinfachten Version ist die Funktion wie folgt definiert:

Peterfunktion
$P : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$

$$\begin{aligned} P(0,n) &= n+1 \\ P(m+1,0) &= P(m,1) \\ P(m+1,n+1) &= P(m, P(m+1,n)) \end{aligned}$$

Beweis der Berechenbarkeit der Peterfunktion

```
PROCEDURE Peter(m,n : CARDINAL) : CARDINAL
BEGIN
  IF m = 0 THEN RETURN(n + 1); ELSE
    IF n = 0 THEN RETURN Peter(m − 1, 1);
    ELSE RETURN Peter(m − 1, Peter(m, n − 1));
    END;
  END;
END Peter;
```

Wir können also ein Programm angeben, das die Peterfunktion berechnet. □

Beispiel
Einige Werte der Peterfunktion:

$$\begin{array}{lll} P(0,n) = n+1 & P(1,n) = n+2 & \\ P(2,n) = 2n+3 & P(3,n) = 2^{n+3} - 3 & \\ P(4,0) = 13 & P(4,1) = 65533 & P(4,2) = 2^{65536} - 3 \end{array}$$

Um zu zeigen, daß die Peterfunktion nicht primitiv-rekursiv ist, benutzen wir folgendes Lemma, das die Existenz einer Majorante, d. h. einer Funktion, die schneller wächst als alle primitiv-rekursiven Funktionen, sichert.

Lemma
Zu jeder einstelligen primitiv-rekursiven Funktion ϕ gibt es eine Zahl m derart, daß für alle $n \in \mathbb{N}$ gilt $\phi(n) < P(m,n)$.

Zum Beweis des Lemmas siehe R.Peter „Rekursive Funktionen“ [6, Seite 68–73].

Mit der Existenz einer Majorante können wir nun indirekt zeigen, daß die Peterfunktion nicht primitiv-rekursiv ist.

Beweis
Annahme: P primitiv-rekursiv

Dann definieren wir

$$Q(n) := P\left(I_1^1(n), I_1^1(n)\right) = P(n,n)$$

und Q ist ebenfalls primitiv-rekursiv.

Nach Lemma existiert eine Zahl $m^\star$ mit

$$Q(n) < P(m^\star, n)$$

Für $n = m^\star$ ergibt sich

$$Q(m^\star) < P(m^\star, m^\star) = Q(m^\star) \qquad ↯$$

Die Peterfunktion ist also nicht primitiv-rekursiv. □

Folglich ist eine Erweiterung nötig, um alle berechenbaren Funktionen zu bekommen:

3.2.5 Minimum-Operator und partiell-rekursive Funktionen

Am Beispiel der Peterfunktion sieht man, daß eine verschränkte Rekursion über zwei Variable offenbar mächtiger ist als eine primitive Rekursion, die nur abwärts über eine Variable geht. Eine Rekursion „aufwärts“ dürfte dagegen niemals zu einem Ende kommen. Der Kompromiß besteht darin, daß man zwar aufwärts geht, aber nicht rekursiv, sondern aufwärts nach einer Nullstelle (d. h. der kleinsten Nullstelle) sucht. Wir gehen damit vom beschränkten zum unbeschränkten Minimumoperator über und lassen auch partielle Funktionen als Argument zu.

Minimum-Operator
Es sei ψ eine $(n+1)$-stellige zahlentheoretische Funktion. Die (partielle) Funktion ϕ entsteht durch Anwenden des Minimum-Operators auf ψ, wenn

$$\phi(x_1,\ldots,x_n) = \mu j\left(\psi(x_1,\ldots,x_n,j) = 0\right) = \begin{cases} \text{das kleinste } j \in \mathbb{N}\text{, mit} \\ \psi(x_1,\ldots,x_n,j) = 0 \text{ und für} \\ \text{alle } i < j \text{ ist } \psi(x_1,\ldots,x_n,i) \\ \text{definiert, falls solches } j \text{ existiert.} \\ \\ \text{nicht definiert, sonst} \end{cases}$$

(Das ist Formalisierung des systematischen Probierens.)

Satz 3.2.9
Entsteht ϕ durch Anwendung des Minimum-Operators auf ψ und ist ψ intuitiv–anschaulich berechenbar, so ist ϕ intuitiv–anschaulich berechenbar.

Beweis

```
PROCEDURE PHI(x_1, ... x_n : CARDINAL) : CARDINAL;
VAR i, j:CARDINAL;
BEGIN
  i := 0;
  LOOP
                                        (⋆ wenn ψ(x_1, ..., x_n i) nicht de-
    j := PSI(x_1, ... x_n, i);          finiert ist, wird die Prozedur PSI
                                        niemals terminieren ⋆)
    IF j = 0 THEN RETURN i;
    ELSE INC(i)
    END
  END
END PHI;
```

Da wir ein Programm angeben können, ist ϕ intuitiv–anschaulich berechenbar. □

partiell-rekursive Funktionen
PREK ist die kleinste Klasse von Funktionen, die ANF enthält und abgeschlossen ist bezüglich Substitution, primitiver Rekursion und Anwendung des Minimum-Operators.

Folgerung 3.2.10
ANF $\subseteq$ PRM $\subseteq$ PREK

Es ist z. B. die Addition primitiv–rekursiv, aber nicht Anfangsfunktion, und die einstellige leere (also nirgends definierte) Funktion l ist nicht primitiv–rekursiv (weil nicht total), aber partiell rekursiv:

$$l(m) = \mu j\,(N\,(+\,(m, j)) = 0)$$

Definition 3.2.11 (allgemein–rekursive Funktionen)
Eine zahlentheoretische Funktion heißt allgemein–rekursiv*, wenn sie partiell–rekursiv und total ist.*

Bei der Anwendung des μ-Operators entsteht in der Regel aus einer totalen Funktion eine partielle, aber auch umgekehrtes ist möglich, aus einer partiellen Funktion entsteht eine totale.

Ist z. B.

$$g(x, i) := \begin{cases} 0, & \text{falls } i = 0 \\ \\ \text{nicht definiert}, & \text{sonst} \end{cases}$$

so ist $\mu i\,(g(x, i) = 0) = C_0^1(x)$.

Offenbar ist $\mu i\,(g(x_1, \ldots, x_n, i) = 0)$ eine totale Funktion genau dann, wenn

1. zu jedem $x_1, \ldots, x_n$ ein i existiert mit $g(x_1, \ldots, x_n, i) = 0$
2. für alle $j < i$ ist $g(x_1, \ldots, x_n, j)$ definiert.

Ist g eine totale Funktion, so ist die zweite Bedingung trivial.

Wenn die erste Bedingung erfüllt ist, so sagt man, daß die Anwendung des μ-Operators „im Normalfall“ erfolgte. Man kann dann zeigen, daß man die Klasse der allgemein–rekursiven Funktionen auch ohne Bezug auf partielle Funktionen definieren kann:

Satz 3.2.12 (allgemein–rekursive Funktionen)
REK *ist die kleinste Klasse von zahlentheoretischen Funktionen, die alle Anfangsfunktionen enthält und abgeschlossen ist in bezug auf Substitution, primitive Rekursion und Anwendung des Minimum–Operators im Normalfall.*

3.3 Der Maschinen–Ansatz

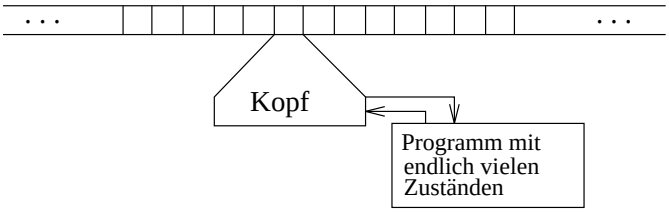
A.M. TURING schlug 1936 eine auf der — nach ihm benannten — TURING–Maschine basierende formale Definition des Berechenbarkeitsbegriffs vor.

3.3.1 Die Turing–Maschine

Eine TURING–Maschine läßt sich wie folgt beschreiben:

- ein beidseitig unbegrenztes Band, das in Felder aufgeteilt ist
- jedes Feld kann ein Zeichen aus einem endlichen Alphabet oder ein Leerzeichen aufnehmen
- auf dem Band bewegt sich ein Schreib–Lesekopf, dabei kann genau das Feld unter dem Kopf gelesen oder beschrieben werden; danach bewegt sich der Kopf maximal um eine Position nach links oder rechts
- ein Programm (auch Maschinentafel genannt) gibt die Berechnungsschritte an
- bei der Berechnung kann nur eine endliche Menge von Zuständen angenommen werden

Veranschaulichung



Formal definieren wir

Turing–Maschine
$T = [X, Z, z_0, \delta]$ ist TURING–Maschine, wenn:

1. X endliches Alphabet, $\star \notin X$ (Leerzeichen)
2. Z endliche Menge von Zuständen, z_0 ist der Anfangszustand
3. $\delta : Z \times (X \cup \{\star\}) \to (X \cup \{\star\}) \times B \times Z$ ist die Überführungsfunktion, wobei $B = \{R, L, N, S\}$ die Menge der Kopfbewegungen ist.

Arbeitsweise

1. T arbeitet taktweise

2. In Takt 1 ist T im Zustand z_0

3. Ist z_{i-1} der Zustand von T im Takt i und x das Zeichen im betrachteten Feld und $\delta(z_{i-1}, x) = [y, b, z]$, so wird

   - y in das betrachtete Feld geschrieben
   - die Kopfbewegung b ausgeführt:

$b = L$	Kopf nach links	$b = R$	Kopf nach rechts
$b = N$	keine Kopfbewegung	$b = S$	Stop der Berechnung

   - in den Zustand $z_i = z$ übergegangen.

Davon ausgehend definieren wir (indem wir die Definition der TURING–Maschine weiter einschränken und präzisieren) den Begriff der

Turing–Berechenbarkeit
Sei $\phi : \underset{n}{\times} \mathbb{N} \to \mathbb{N}$ eine zahlentheoretische Funktion.
ϕ heißt TURING–berechenbar, wenn es eine TURING–Maschine mit $X = \{|\}$,also

$$T \quad = \quad [\{|\}, Z, z_0, \delta]$$

so gibt, daß für alle $x_1, x_2, \ldots, x_n \in \mathbb{N}$ gilt:
Wenn $[x_1, \ldots, x_n] \in D_\phi$ und T wird in der Situation

$$\cdots \star \star \underbrace{|\cdots|}_{x_1+1} \star \underbrace{|\cdots|}_{x_2+1} \star \cdots \star \underbrace{|\cdots|}_{x_n+1} \underset{\substack{\triangle \\ z_0}}{\star} \quad \star \cdots$$

gestartet, so stoppt T nach endlich vielen Takten in der Situation

$$\cdots \star \star \underbrace{|\cdots|}_{x_1+1} \star \underbrace{|\cdots|}_{x_2+1} \star \cdots \star \underbrace{|\cdots|}_{x_n+1} \star \quad \underbrace{|\cdots|}_{\phi(x_1,\ldots,x_n)+1} \underset{\substack{\triangle \\ z}}{\star} \quad \cdots$$

Wenn dagegen $[x_1, \ldots, x_n] \notin D_\phi$, so kommt T nicht zum Stop.
Wir verlangen noch zusätzlich, daß der Kopf bei der Berechnung nicht über das zweite leere Feld links von den Argumenten hinausläuft.

Wie man sieht, codieren wir die natürlichen Zahlen *unär*, d. h. jede Zahl n wird durch $n + 1$ Striche dargestellt. Dadurch können wir 0 als einen Strich schreiben.

Turing–berechenbare Funktionen
TM = die Klasse der TURING–berechenbaren zahlentheoretischen Funktionen.

3.3.2 Bausteine für Turing–Maschinen

Wir geben nun einige (später benötigte) simple Bausteine, sogenannte *Makros*, für TURING–Maschinen–Programme an. Dazu sei $X = \{|\}$. Der Anfangszustand ist 1. Bei einem Stoppbefehl geben wir keinen Folgezustand an, obwohl das formal gefordert wäre, ebenso füllen wir Felder in den Tabellen nicht aus, die nicht zum Zuge kommen können.

- r : „Gehe ein Feld nach rechts“

r	$\star$	|
1	$\star$R2	|R2
2	$\star$S	|S

- l : „Gehe ein Feld nach links“

l	$\star$	|
1	$\star$L2	|L2
2	$\star$S	|S

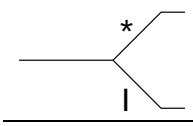
- „|“ : „Schreibe | auf das betrachtete Feld“

„|“	$\star$	|
1	|N2	|N2
2	–	|S

- „$\star$“ : „Schreibe $\star$ auf das betrachtete Feld“

„$\star$“	$\star$	|
1	$\star$N2	$\star$N2
2	$\star$S	–

- —<: „bedingter Sprung“

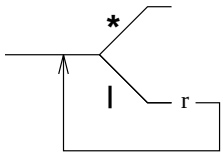


	$\star$	|
1	$\star$N2	|N3
2	$\star$S	–
3	–	|S

- $P_1; P_2$ bzw. $P_1 \to P_2$: Programmverkettung als Operation mit Programmen
  - P_1 habe Zustände $1, \ldots, l$
  - P_2 habe Zustände $1, \ldots, k$
  - In P_2 alle Adressen um l erhöhen: $l+1, \ldots, l+k$
  - In P_1 alle Stoppanweisungen $\star$S bzw. |S ersetzen durch $\star$N$(l+1)$ bzw. |N$(l+1)$ (Übergang in den Anfangszustand von P_2)

- R : „Nach rechts auf den ersten $\star$ gehen“ (WHILE not $\star$ DO r END)

  Dies ist eine Verkettung von zwei Programmen:



R	$\star$	|
1	$\star$N2	|N3
2	$\star$S	–
3	–	|S/N4
4	$\star$R5	|R5
5	$\star$S/N1	|S/N1

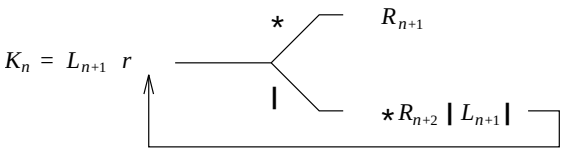
wobei / für „ersetzen“ steht.

- R_n (für $n \geq 1$) : „Gehe nach rechts auf den n-ten $\star$, der $\star$ unter dem Kopf mitgezählt.“
  - $R_1 := R$
  - $R_n := (Rr)^{n-1}\mathrm{R}$
- L_n : „Gehe nach links auf den n-ten $\star$, der $\star$ unter dem Kopf mitgezählt.“
  - $L_1 = L$
  - $L_n = (Ll)^{n-1}\, L.$
- K_n : „Kopiere die n-te Strichfolge links vom Kopf“

  Dabei gilt:

  $$\star\underbrace{|\cdots|}_{k_1}\star\cdots\star\underbrace{|\cdots|}_{k_n}\star\ \star\cdots\star \quad \text{wird zu} \quad \star\underbrace{|\cdots|}_{k_1}\star\cdots\star\underbrace{|\cdots|}_{k_n}\star\underbrace{|\cdots|}_{k_1}\star\ \star\cdots\star$$

  und $k_1 = 0$ ist erlaubt.



- Lö : Löschprogramm

  Das Löschprogramm brauchen wir, um die normierte Endsituation „Argument, Leerzeichen, Funktionswert“ auf dem Band herzustellen, d. h. Zwischenergebnisse zu löschen.

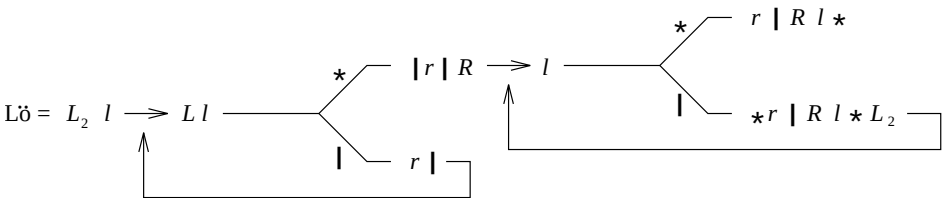
  Folgende Situation auf dem Band liege vor:

  $$\underbrace{\star\star\star\underbrace{|\cdots|}_{k_1}\star\cdots\star\underbrace{|\cdots|}_{k_n}}_{\text{Argumente}}\ \underbrace{\star\star\star}_{3\ \text{Leerz.}}\ \underbrace{|\cdots|\star|\cdots|\star\cdots\star|\cdots|}_{\text{Zwischenergebnisse}}\star\underbrace{|\cdots|}_{\text{Wert}}\star\ \star\cdots\star$$

  Das Löschprogramm soll daraus die folgende Situation herstellen:

  $$\star\star\star\underbrace{|\cdots|}_{k_1}\star\cdots\star\underbrace{|\cdots|}_{k_n}\star\underbrace{|\cdots|}_{\text{Wert}}\star\ \star\cdots\star$$

  Dazu wird zunächst die Folge der Zwischenergebnisse in eine einzige Strichfolge verwandelt (die trennenden Leerzeichen $\star$ werden in | verwandelt) und danach sukzessive abgebaut.



3.4 Äquivalenz der Berechenbarkeitsbegriffe

Nachdem wir uns mit dem axiomatischen und dem Maschinen–Ansatz befaßt haben, wollen wir nun nachweisen, daß beide — so unterschiedlich ihre Konstruktion ist — äquivalent sind.

Wir zeigen für eine beliebige Funktion ϕ, daß

$$\phi \text{ TURING–berechenbar} \qquad \text{gdw.} \qquad \phi \text{ partiell-rekursiv}$$

3.4.1 Partiell–rekursive Funktionen sind Turing–berechenbar

Da wir die Klasse der partiell–rekursiven Funktionen induktiv definiert haben, führen wir den Beweis über deren Aufbau. Dazu werden wir einige der ab Seite 107 aufgeführten TURING–Maschinen–Makros benutzen.

Anfangsfunktionen

Satz 3.4.1
Alle Anfangsfunktionen sind TURING*–berechenbar*

Beweis

1. $N(x) = x + 1$

   Das Programm $K_1|r$ berechnet N.

$$\star\star\underbrace{|\cdots|}_{x+1}\triangle\star\star \quad \text{wird zu} \quad \star\star\underbrace{|\cdots|}_{x+1}\star\underbrace{|\cdots|}_{x+2}\triangle\star$$

2. $C_i^n(x_1, \ldots, , x_n) = i$

   Das Programm $(r|)^{i+1}\, r$ berechnet C_i^n.

3. $I_k^n(x_1, \ldots, x_n) = x_k$

   K_{n-k+1} leistet das Verlangte.

Also sind alle Anfangsfunktionen TURING–berechenbar. □

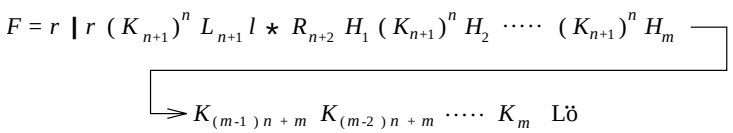
Substitution

Satz 3.4.2
Wenn ϕ durch Substitution aus ψ und $\chi_1, \ldots, \chi_m$ entsteht und $\psi, \chi_1, \ldots, \chi_m$ sind TURING*–berechenbar, so ist ϕ* TURING*–berechenbar.*

Beweis
$\phi(x_1, \ldots, x_n) = \psi(\chi_1(x_1, \ldots, x_n), \ldots, \chi_m(x_1, \ldots, x_n))$

Sei χ_i berechnet durch H_i und ψ berechnet durch P.

Dann wird ϕ berechnet durch



□

Primitive Rekursion

Satz 3.4.3
Entsteht ϕ durch primitive Rekursion aus ψ und χ und sind ψ, χ TURING*–berechenbar, so ist ϕ* TURING*–berechenbar.*

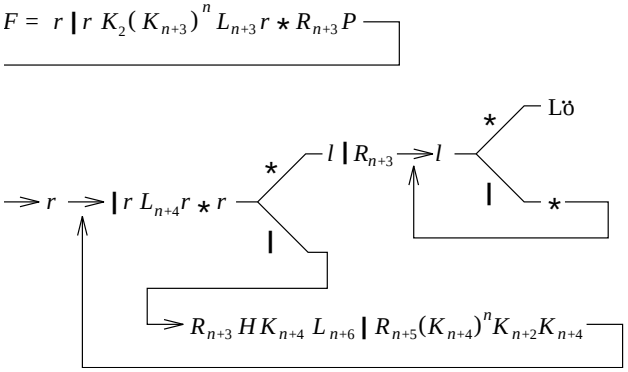
Beweis
Es ist

$$\begin{aligned}\phi^{(n+1)}(x_1,\ldots,x_n,0) &= \psi^{(n)}(x_1,\ldots,x_n)\\ \phi^{(n+1)}(x_1,\ldots,x_n,j+1) &= \chi^{(n+2)}\left(x_1,\ldots,x_n,\phi^{(n+1)}(x_1,\ldots,x_n,j),j\right)\end{aligned}$$

Man beachte das modifizierte Schema im Rekursionsschritt; durch geeignete Auswahlfunktionen kann man die Äquivalenz zum bekannten Rekursionsschema nachweisen.

Sei ψ berechnet durch P und χ berechnet durch das Programm H.

Dann wird ϕ berechnet durch F mit



□

Beispiel
Sei $n = 2$; $\phi(a,b,2) = ?$

Anfangssituation:

$$\cdots \star a \star b \star |||\star$$
$$\triangle$$

$r|rK_2(K_5)^2L_5r\star R_5P$:

$$\cdots \star a \star b \star ||| \star\star\star ||| \star a \star b \star \psi(a,b) \star$$
$$\triangle$$

$r|rL_6r\star rR_5H$:

$$\cdots | \star\star\star\star || \star a \star b \star \psi(a,b) \star | \star \chi(a,b,\psi(a,b),0) \star$$
$$\triangle$$

$K_6L_8|R_7(K_6)^2K_4K_6$:

$$\cdots|\star\star\star|||\star a\star b\star\phi(a,b,0)\star|\star\phi(a,b,1)\star||\star a\star b\star\phi(a,b,1)\star|\star$$
$$\triangle$$

$|rL_6r\star rR_5H$:

$$\cdots\phi(a,b,1)\star\star|\star a\star b\star\phi(a,b,1)\star||\star\underbrace{\chi(a,b,\phi(a,b,1),1)}_{\phi(a,b,2)}\star$$
$$\triangle$$

$K_6L_8|R_7(K_6)^2K_4K_6$:

$$\cdots\phi(a,b,1)\star||\star a\star b\star\phi(a,b,2)\star|\star a\star b\star\phi(a,b,2)\star||\star$$
$$\triangle$$

$|rL_6r\star r$

$$\cdots\phi(a,b,2)\star\star\star\ a\star b\star\phi(a,b,2)\star|||\star$$
$$\triangle$$

$l|R_5l$

$$\cdots\phi(a,b,2)\star|\star a\star b\star\phi(a,b,2)\star|||\ \star$$
$$\triangle$$

$\star l\star l\star l$Lö:

$$\cdots\star\star\star a\star b\star|||\star\phi(a,b,2)\star \quad \text{Stop}$$
$$\triangle$$

Minimum–Operator

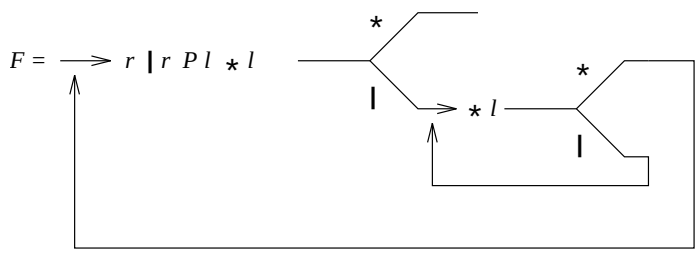
Satz 3.4.4
Entsteht ϕ durch Anwendung des μ-Operators auf ψ und ψ ist TURING*–berechenbar, so ist ϕ* TURING*–berechenbar.*

Beweis
Es ist

$$\phi(x_1,\ldots,x_n) \quad = \quad \mu j\,(\psi(x_1,\ldots x_n,j)=0)$$

Sei ψ berechnet durch P.

Dann wird ϕ berechnet durch



□

Folgerung 3.4.5
Alle partiell–rekursiven Funktionen sind TURING*–berechenbar.*

3.4.2 Turing–berechenbare Funktionen sind partiell–rekursiv

Um die Umkehrung zu beweisen, verwenden wir ein Verfahren, das nach Kurt GÖDEL auch Gödelisierung genannt wird: Mit Hilfe eineindeutiger und intuitiv–anschaulich berechenbarer Abbildungen wird der Formalismus der TURING–Maschinen in die Arithmetik der natürlichen Zahlen abgebildet. Dadurch gehen Beziehungen zwischen TURING–Maschinen und ihren Konfigurationen in Beziehungen zwischen Zahlen über, die unter Umständen durch partiell–rekursive Funktionen beschrieben werden. Daß die von einer gegebenen TURING–Maschine berechnete zahlentheoretische Funktion partiell–rekursiv ist, bedeutet ja nichts anderes, als daß die Gödelzahl der Stop–Konfiguration partiell–rekursiv von der Gödelzahl der Start–Konfiguration abhängt.

Dazu gebrauchen wir einige der auf Seite 101 aufgeführten primitiv–rekursiven Funktionen. Insbesondere werden wir die Eindeutigkeit der Primzahlzerlegung ausnutzen, um Tupel von Zahlen in einer Zahl zu codieren.

Bevor wir den TURING–Maschinen $T = [X, Z, z_0, \delta]$ ihre Gödelzahl zuordnen, machen wir einige Einschränkungen, die ihre Berechnungsfähigkeit nicht beeinträchtigen (wir *normieren* die TURING–Maschine):

Wir setzen voraus, daß $X = \{|\}$ ist und setzen

$$x_0 := \star \quad , \quad x_1 := |.$$

Die Zustandsmenge sei numeriert als

$$Z = \{z_1, \ldots, z_m\}$$

wobei $z_0 = z_1$ der Anfangszustand ist. Die Menge $B = \{N, R, L, S\}$ sei als

$$b_0 = N \quad , \quad b_1 = R \quad , \quad b_2 = L \quad \text{und} \quad b_3 = S$$

numeriert und wir setzen voraus, daß die Maschine beim Stop nichts mehr ändert: Wenn $\delta(z, x) = [x', S, z']$, dann $x' := x$ und $z' := z$.

Es sei $\mathcal{TM}$ die Menge aller TURING–Maschinen der angegebenen Art.

1. Die Gödelzahl der Turing–Maschine

Die TURING–Maschine T ist durch die Tabelle der Funktion δ vollständig beschrieben. Deshalb codieren wir

1. Tabelleneintrag: neue Beschriftung, Kopfbewegung, Zustandsänderung

$$\text{gö}([x_i, b_j, z_k]) \quad := \quad 2^i 3^j 5^k.$$

   mit $i \in \{0, 1\}$, $j \in \{0, 1, 2, 3\}$ und $k \in \{1, \ldots, m\}$

2. Zeile in der Tabelle von δ: $\quad [\delta(z_\eta, \star), \delta(z_\eta, |)]$

$$\text{gö}(\delta, z_\eta) \quad := \quad 2^{\text{gö}(\delta(z_\eta, x_0))} \cdot 3^{\text{gö}(\delta(z_\eta, x_1))}.$$

3. Gödelzahl der TURING–Maschine: alle Zeilen der Tabelle

$$\text{gö}(T) \quad = \quad \prod_{\eta=1}^{m} q_{\eta-1}^{\text{gö}(\delta, z_\eta)}$$

Lemma 1

1. *Die Abbildung* $\text{gö} : \mathcal{TM} \to \mathbb{N}$ *ist eineindeutig.*
2. *Die charakteristische Funktion*

$$\chi_{\mathcal{TM}}(i) = \left\{ \begin{array}{ll} 0, & \textit{falls eine } \textsc{Turing}\text{–} \\ & \textit{Maschine } T \in \mathcal{TM} \textit{ mit} \\ & \text{gö}(T) = i \textit{ existiert} \\ 1, & \textit{sonst} \end{array} \right\} \textit{ist primitiv–rekursiv}$$

d. h. die Menge der Gödelzahlen ist primitiv–rekursiv entscheidbar.

Beweis
Wenn i die Gödelzahl einer Turing–Maschine T ist, dann (und nur dann) gilt:

1. $i > 0$
2. i ist Produkt von positiven Potenzen der Primzahlen $q_0, \ldots, q_{m-1}$, wobei m die Zahl der Zustände der Turing–Maschine T ist und der Exponent von $q_{\eta-1}$ ist Gödelzahl der η-ten Zeile in der Tabelle von T, d. h.

$$i = \prod_{j=0}^{\mu m(m \leq i \wedge \exp(i,m)=0) \dot{-} 1} q_j^{\exp(i,j)}$$

und für alle j mit $0 \leq j < \mu m\,(m \leq i \wedge \exp(i,m) = 0)$ gibt es Zahlen l_1, l_2 mit

- $\exp(i,j) = 2^{l_1} \cdot 3^{l_2}$
  (d. h. $l_1 = \exp(\exp(i,j),0)$; $l_2 = \exp(\exp(i,j),1)$)
- $l_1 = 2^{\exp(l_1,0)} \cdot 3^{\exp(l_1,1)} \cdot 5^{\exp(l_1,2)}$
  $l_2 = 2^{\exp(l_2,0)} \cdot 3^{\exp(l_2,1)} \cdot 5^{\exp(l_2,2)}$
  (d. h. l_1, l_2 sind Gödelzahlen von Tripeln)
- $\exp(l_1,0) \leq 1$ (die erste Komponente ist x_0 oder x_1)
  $\exp(l_1,1) \leq 3$ (die zweite Komponente ist b_0, b_1, b_2 oder b_3)
  $1 \leq \exp(l_1,2) < \mu m(m \leq i \wedge \exp(i,m) = 0)$ (die dritte Komponente ist eine Zustandsnummer)
- $\exp(l_2,0) \leq 1$
  $\exp(l_2,1) \leq 3$
  $1 \leq \exp(l_2,2) \leq \mu m(m \leq i \wedge \exp(i,m) = 0)$

Um zu zeigen, daß $\chi_{\mathcal{TM}}$ primitiv–rekursiv ist, müssen wir diese logischen Zusammenhänge arithmetisch ausdrücken. Dazu benutzen wir

$a = b$	gdw.	$(a \dot{-} b) + (b \dot{-} a) = 0$	gdw.	$\lvert a - b \rvert = 0$
$a \leq b$	gdw.	$a \dot{-} b = 0$		
$a = 0$ und $b = 0$	gdw.	$a + b = 0$		

Aus der Aussage

> Für alle j mit $0 \leq j < \mu m\,(m \leq i \wedge \exp(i,m) = 0)$ gibt es Zahlen l_1, l_2 so, daß ...

wird durch Elimination von l_1, l_2 logisch äquivalent:

$$
\begin{aligned}
&\sum_{j=0}^{\mu m(m\leq i\wedge \exp(i,m)=0)\dot{-}1} \Bigg[\left|\exp(i,j) - 2^{\exp(\exp(i,j),0)} \cdot 3^{\exp(\exp(i,j),1)}\right| + \\
&+ \left|\exp\left(\exp(i,j),0\right) - 2^{\exp(\exp\exp((i,j),0),0)} \cdot 3^{\exp(\exp\exp(i,j),0),1)} \cdot 5^{\exp(\exp\exp(i,j),0),2)}\right| + \\
&+ \left|\exp\left(\exp(i,j),1\right) - 2^{\exp(\exp\exp((i,j),1),0)} \cdot 3^{\exp(\exp\exp(i,j),1),1)} \cdot 5^{\exp(\exp\exp(i,j),1),2)}\right| + \\
&+ \Big(\exp\left(\exp\left(\exp\left(i,j\right),0\right),0\right) \dot{-} 1\Big) + \\
&+ \Big(\exp\left(\exp\left(\exp\left(i,j\right),1\right),0\right) \dot{-} 1\Big) + \\
&+ \Big(\exp\left(\exp\left(\exp\left(i,j\right),0\right),1\right) \dot{-} 3\Big) + \\
&+ \Big(1 \dot{-} \exp\left(\exp\left(\exp\left(i,j\right),0\right),2\right)\Big) + \\
&+ \Big(1 \dot{-} \exp\left(\exp\left(\exp\left(i,j\right),1\right),2\right)\Big) + \\
&+ \Big(\exp\left(\exp\left(\exp\left(i,j\right),0\right),2\right) \dot{-} \Big[\mu m\left(m \leq i \wedge \exp(i,m) = 0\right) \dot{-} 1\Big]\Big) + \\
&+ \Big(\exp\left(\exp\left(\exp\left(i,j\right),1\right),2\right) \dot{-} \Big[\mu m\left(m \leq i \wedge \exp(i,m) = 0\right) \dot{-} 1\Big]\Big)\Bigg] = 0
\end{aligned}
$$

Wenn wir den Term auf der linken Seite dieser Gleichung mit $\sum[\ldots]$ abkürzen, so haben wir:

$$
\begin{aligned}
\chi_{\mathcal{TM}}(i) = 0 \quad \text{gdw.} \quad & i > 0 \text{ und} \\
& i = \prod_{j=0}^{\mu m(m\leq i\wedge \exp(i,m)=0)\dot{-}1} \cdot q_j^{\exp(i,j)} \\
& \text{und } \sum[\ldots] = 0
\end{aligned}
$$

also ist

$$
\chi_{\mathcal{TM}}\left(i\right) = \operatorname{sg}\left(\overline{\operatorname{sg}}\left(i\right) + \left|i - \prod_{j=0}^{\mu m(\cdots)\dot{-}1}\right| + \sum\left[\cdots\right]\right)
$$

Offensichtlich ist $\chi_{\mathcal{TM}}$ primitiv–rekursiv. □

2. Die Gödelzahl einer endlichen Bandbeschriftung

Wir numerieren die Felder unseres („halben“) Bandes von links nach rechts mit $0, 1, 2, \ldots$

Eine Bandbeschriftung σ ist dann eine Abbildung, die jeder Feldnummer die Nummer des in diesem Feld stehenden Buchstabens zuordnet, d. h.

$$
\sigma \quad : \quad \mathbb{N} \to \{0, 1\}
$$

σ ist endlich, wenn auf fast allen Feldern das Zeichen $\star$ steht, also wenn

$$
j_\sigma \quad := \quad \text{SUP}\{i | \sigma(i) = 1\}
$$

existiert (das Supremum der leeren Menge ist 0), d. h. es gibt einen „letzten“ Strich auf dem Band.

Wir setzen

$$\text{gö}(\sigma) \quad = \quad \prod_{j=0}^{j_\sigma} q_j^{\sigma(j)}$$

Lemma 2
Σ sei die Menge aller endlichen Bandbeschriftungen

1. *Die Abbildung* $\text{gö} : \Sigma \to \mathbb{N}$ *ist eineindeutig.*
2. *Die charakteristische Funktion*

$$\chi_\Sigma(i) = \begin{cases} 0, & \textit{falls ein } \sigma \in \Sigma \textit{ mit} \\ & \text{gö}(\sigma) = i \textit{ existiert} \\ 1, & \textit{sonst} \end{cases}$$

ist primitiv–rekursiv.

Beweis
$\chi_\Sigma(i) = 0$ gdw. $i > 0$ und für alle $j\,(0 \le j \le i)$ gilt $\exp(i,j) \le 1$.
$\chi_\Sigma(i) = \text{sg}\left(\overline{\text{sg}}(i) + \sum_{j=0}^{i}\left(\exp(i,j) \dot{-} 1\right)\right)$. □

3. Die Gödelzahl einer Konfiguration einer Turing–Maschine

Eine Konfiguration K der Turing–Maschine T ist ein Tripel

$$K \quad = \quad [j, \sigma, z_i]$$

wobei j die Nummer des Feldes auf dem der Kopf steht, σ die Beschriftung des Bandes und z_i der aktuelle Zustand ist.

$$\text{gö}(K) \quad = \quad 2^j \cdot 3^{\text{gö}(\sigma)} \cdot 5^i$$

Lemma 3

1. *Die Abbildung* $\text{gö} : \mathfrak{K} \to \mathbb{N}$ *ist eineindeutig.*
2. *Die charakteristische Funktion*

$$\chi_K(k) := \begin{cases} 0, & \textit{falls es ein } K \in \mathfrak{K} \textit{ gibt} \\ & \textit{mit } k = \text{gö}(K) \\ 1, & \textit{sonst} \end{cases}$$

ist primitiv-rekursiv.

Beweis
Die Zahl k ist genau dann die Gödelzahl einer Konfiguration, wenn

1. $k > 0$

2. k die Gödelzahl eines Tripels ist:

$$k = 2^{\exp(k,0)} \cdot 3^{\exp(k,1)} \cdot 5^{\exp(k,2)}$$

und $\exp(k,1)$ Gödelzahl einer Bandbeschriftung ist, d. h.

$$\chi_\Sigma\left(\exp(k,1)\right) = 0$$

und $1 \leq \exp(k,2)$, d. h. $\exp(k,2)$ ist eine Zustandsnummer.

Also ist

$$\begin{aligned}\chi_K\left(k\right) = & \\ & \operatorname{sg}\Big(\overline{\operatorname{sg}}\left(k\right) + \left|k - 2^{\exp(k,0)} \cdot 3^{\exp(k,1)} \cdot 5^{\exp(k,2)}\right| \\ & + \chi_\Sigma\left(\exp\left(k,1\right)\right) + \overline{\operatorname{sg}}\left(\exp\left(k,2\right)\right)\Big).\end{aligned}$$

□

4. Die Gödelzahl der Folgekonfiguration einer Konfiguration

Sei N die Gödelzahl einer Konfiguration der TURING–Maschine mit der Gödelzahl t.

Um die Folgekonfiguration zu berechnen, müssen wir diese Gödelzahl N zuerst decodieren, d. h. wir simulieren die Ausführung eines Programmschrittes der TURING–Maschine, indem wir die Gödelzahl der Konfiguration auswerten und daraus über die Gödelzahl t der TURING–Maschine die neue Gödelzahl der Folgekonfiguration bilden.

1. $N = 2^j \cdot 3^s \cdot 5^i$ — $N = \text{gö}\left([j, \sigma, z_i]\right)$ (Konfiguration); $s = \text{gö}(\sigma)$ (Bandbeschriftung)
2. $\kappa = \exp\left(s, j\right)$ — Nummer des Zeichens im Feld j bei σ: $\sigma\left(j\right) = \kappa$
3. $\lambda = \exp\left(t, i-1\right)$ — Zeile des Zustands z_i in TURING–Maschine T
4. $\tau = \exp\left(\lambda, \kappa\right)$ — $\delta\left(z_i, x_\kappa\right)$ (Tabelleneintrag)

Dann ist die Folgekonfiguration bestimmt als Funktion

$$F\left(t, N\right) \quad := \quad 2^u \cdot 3^v \cdot 5^w$$

mit

$$\begin{aligned} u &= \begin{cases} j, & \text{falls } \exp\left(\tau,1\right) = 0 \quad (\star N) \\ j+1, & \text{falls } \exp\left(\tau,1\right) = 1 \quad (\star R) \\ j-1, & \text{falls } \exp\left(\tau,1\right) = 2 \quad (\star L) \end{cases} \\ &= j + \exp\left(\tau,1\right) \dot{-} \Big(\exp\left(\tau,1\right) \dot{-} 1\Big) \cdot 3 \\ v &= \frac{s}{q_j^\kappa} \cdot q_j^{\exp(\tau,0)} \\ w &= \exp\left(\tau,2\right) \end{aligned}$$

und u ist die neue Kopfposition, v die geänderte Bandbeschriftung und w der neue Zustand.

Bemerkung
Die Stopsituation $\exp\left(\tau,1\right) = 3$ wird nicht berücksichtigt, da wir vereinbart haben, daß dann $\delta\left(z_i, x\right) = [x, S, z_i]$ gilt.

Lemma 4

$$\tilde{F}(t,N) := \begin{cases} F(t,N), & \textit{falls } \chi_{\mathcal{TM}}(t) = 0 \textit{ und } \chi_K(N) = 0 \textit{ und } \text{gö}^{-1}(N) \\ & \textit{keine Stop–Konfiguration von } \text{gö}^{-1}(t) \textit{ ist} \\ 0, & \textit{sonst} \end{cases}$$

ist primitiv-rekursiv.

5. Die Folge der Gödelzahlen der Konfigurationen bei der Berechnung

Nachdem wir aus einer beliebigen Konfiguration die Folgekonfiguration berechnen können, läßt sich jetzt auch die Folge der Gödelzahlen aller Konfigurationen, die von der TURING–Maschine T bei Start in einer Anfangs–Konfiguration durchlaufen werden, berechnen.

Eine *Anfangs–Konfiguration* bei der Berechnung einer n-stelligen Funktion sieht wie folgt aus:

$$\star|^{a_1+1}\star\cdots\star|^{a_n+1}\star\ \star\cdots$$
$$\triangle$$

d. h.

$$K_{a_1a_2\cdots a_n} := \Big[\ \underbrace{n+\sum_{v=1}^{n}(a_v+1)}_{\text{Kopfposition}},\ \underbrace{\sigma_{a_1,\ldots a_n}}_{\text{Bandkod.}},\ \underbrace{z_1}_{\text{Anfangszust.}}\ \Big]$$

Wir bezeichnen die Gödelzahl der i-ten Konfiguration in der Folge der Konfigurationen, die T bei Start in $K_{a_1,\ldots,a_n}$ durchläuft mit

$$A_T(a_1,\ldots,a_n,i)$$

Dann ist für die Anfangs–Konfiguration

$$A_T(a_1,\ldots,a_n,0) \quad = \quad \text{gö}(K_{a_1,\ldots,a_n})$$

und da bei der Stop–Konfiguration die letzte Konfiguration wiederholt wird, erhalten wir für die Konfigurationsfolge

$$A_T(a_1,\ldots,a_n,i+1) \quad = \quad \begin{cases} A_T(a_1,\ldots,a_n,i), \text{ falls dies Stop–Konfig. ist} \\ F(\text{gö}(T), A_T(a_1,\ldots,a_n,i)), \text{ sonst} \end{cases}$$

und A_T ist primitiv-rekursiv.

Lemma 5
Die Funktion

$$A(t,a_1,\ldots,a_n,j) \quad := \quad \begin{cases} A_T(a_1,\ldots,a_n,j), \textit{ falls } \chi_{\mathcal{TM}}(t) = 0, T = \text{gö}^{-1}(t) \\ 0, \textit{ sonst} \end{cases}$$

ist primitiv-rekursiv.

6. Die Funktion β

Wir definieren nun eine Funktion β, die folgende Eigenschaften hat:

$\beta\,(t, a_1, \ldots, a_n, j) = 0$ gdw.

- t ist Gödelnummer einer TURING–Maschine T,
- es gibt Zahlen k, l mit $j = 2^k \cdot 3^l$ und
  - $k \geq 0$
  - l ist Gödelzahl der Konfiguration, die T nach k Takten erreicht hat, wenn T in $K_{a_1,\ldots,a_n}$ gestartet wurde.
  - Im Takt k hat T schon gestoppt.

Im Takt $k = \exp(j, 0)$ ist die Konfiguration mit der Gödelzahl $l = \exp(j, 1)$ erreicht; dies ist eine Stop–Konfiguration genau dann, wenn (vgl. oben 4.)

$$\begin{aligned}
\tau &= \exp(\lambda, \kappa) \\
&= \exp\Big(\exp(t, i \dot{-} 1), \exp(s, j)\Big) \\
&= \exp\Big(\exp(t, \exp(l, 2) \dot{-} 1), \exp(\exp(l, 1), \exp(l, 0)\Big) \\
&= \exp\Big(\exp(t, \exp(\exp(j, 1), 2) \dot{-} 1), \exp(\exp(\exp(j, 1), 1), \exp(\exp(j, 1), 0)\Big) \\
\exp(\tau, 1) &= 3
\end{aligned} \qquad \circledast$$

Also ist

$$\beta\,(t, a_1, \ldots, a_n, j) := \begin{cases} 0, & \text{falls } j = 2^{\exp(j,0)} \cdot 3^{\exp(j,1)} \text{ und } exp\,(j, 0) \geq 0 \text{ und} \\ & \exp(j, 1) = A\,(t, a_1, \ldots, a_n, \exp(j, 0)) > 0 \text{ und } \circledast \\ \\ 1, & \text{sonst} \end{cases}$$

Man zeigt leicht:

Lemma 6
β ist primitiv-rekursiv.

7. Die Resultatfunktion α

Wir benötigen noch eine Funktion α, die uns das Resultat einer Berechnung auf einer TURING–Maschine anhand der Gödelzahlen liefert. Dabei soll die Endkonfiguration wie folgt ausgewertet werden:

Ist $n = \exp\,(j, 1) = \text{gö}\,([k, \sigma, z_l])$ die Gödelzahl der Konfiguration $[k, \sigma, z_l]$, wobei die Bandbeschriftung die Form

$$\sigma = \star w \star \underbrace{|\cdots|}_{b+1} \star\ \star\star\cdots \qquad \circledast$$

$\triangle$

hat, wobei w ein beliebiges Wort über $\{\star, |\}$ ist, dann soll $\alpha\,(j) = b$ und sonst $= 0$ sein.

Offenbar ist $\alpha\,(j) = b$ gdw. $j > 0$ und für die Zahl $n = \exp(j, 1)$ gilt

1. $\chi_K^{(n)} = 0$
2. Für $k = \exp(n, 0)$ und die Bandbeschriftung mit $\sigma(i) = \exp\,(\exp(n, 1), i)$ gilt

$$b = \left[\sum_{l=0}^{k \dot{-} 2} \left(\prod_{i=l+1}^{k \dot{-} 1} \sigma\,(i) \dot{-} \prod_{i=l}^{k \dot{-} 1} \sigma\,(i)\right) \Big(\big(k \dot{-} l\big) \dot{-} 1\Big)\right] \dot{-} 1$$

Lemma 7
α ist primitiv-rekursiv.

Nun haben wir das nötige Handwerkszeug, um den Beweis zu führen, daß jede TURING–berechenbare Funktion partiell–rekursiv ist.

Beweis
Es sei f eine n-stellige zahlentheoretische Funktion, berechnet von der TURING–Maschine $T = [\{|\}, Z, z_1, \delta]$, ferner sei $t = \text{gö}\,(T)$.

Dann gilt für alle $a_1, \ldots, a_n \in \mathbb{N}$:

$$\begin{array}{lll} [a_1, \ldots, a_n] \in D_f & \text{gdw.} & T \text{ bei Start in der Konfiguration } K_{a_1,\ldots,a_n} \text{ mit} \\ & & \text{einer Konfiguration der Form } \circledast \text{ von oben} \\ & & \text{stoppt} \\ & \text{gdw.} & \text{es ein } j \text{ gibt mit } \beta\,(t, a_1, \ldots, a_n, j) = 0. \end{array}$$

Wenn $[a_1, \ldots, a_n] \in D_f$, so ist $f\,(a_1, \ldots, a_n) = \alpha\,(\mu j\,(\beta\,(t, a_1, \ldots, a_n, j) = 0))$.
Ist dagegen $[a_1, \ldots, a_n] \notin D_f$, so

- stoppt T bei Start in $K_{a_1,\ldots,a_n}$ nicht
- gibt es kein j mit $\beta\,(\cdots, j) = 0$
- ist $\alpha\,(\cdots)$ nicht definiert
- ist $f\,(a_1, \ldots, a_n) = \alpha\,(\mu j\,(\beta\,(t, a_1, \ldots, a_n, j) = 0))$.

Die Funktion f entsteht durch einmalige Anwendung des Minimum–Operators mit den primitiv–rekursiven Funktionen α, β; f ist also partiell–rekursiv. □

3.5 Folgerungen

Im Sinne der Logik haben wir soeben einen Vollständigkeitssatz bewiesen, wir haben gezeigt, daß jede TURING–berechenbare Funktion partiell–rekursiv ist, die Menge der aus den Anfangsfunktionen mittels Substitution, primitiver Rekursion und Anwendung des Minimumoperators ableitbaren Funktionen, also die Menge der partiell–rekursiven Funktionen die Menge der TURING–berechenbaren Funktionen umfaßt. Das haben wir getan, indem wir eine Normalform für TURING–berechenbare Funktionen hergeleitet haben:

Kleenescher Darstellungssatz
Es gibt eine einstellige primitiv–rekursive Funktion α so, daß zu jeder Stellenzahl n eine $(n+2)$-stellige primitiv–rekursive Funktion β derart existiert, daß zu jeder n-stelligen TURING–berechenbaren (bzw. partiell–rekursiven) Funktion f eine Zahl t existiert so, daß für alle $[a_1, \ldots, a_n]$ gilt:

$$f\,(a_1, \ldots, a_n) = \alpha\,(\mu j\,(\beta\,(t, a_1, \ldots, a_n, j) = 0))$$

Das besondere dieser Darstellung, dieser Normalform, besteht offensichtlich darin, daß die primitiv–rekursiven (also totalen) Funktionen α, β von f (bis auf die Stellenzahl) gar nicht abhängen und daß der Minimumoperator genau einmal vorkommt.

Wir haben damit gezeigt, daß eine zahlentheoretische Funktion genau dann TURING–berechenbar ist, wenn sie partiell–rekursiv ist. Analoge Sätze wurden für alle weiteren Präzisierungen des Berechenbarkeitsbegriffes bewiesen. Das stützt die

Churchsche These
Eine zahlentheoretische Funktion ist genau dann intuitiv–anschaulich berechenbar, wenn sie partiell-rekursiv ist.

Wir ziehen noch einige Folgerungen aus dem Satz von KLEENE. Insbesondere betrachten wir die primitiv–rekursiven Funktionen α und β im Falle $n = 1$ und setzen

$$A\,(t,i) := \alpha\,(\mu j\,(\beta\,(t,i,j) = 0))$$

Offenbar ist A eine zweistellige partiell–rekursive Funktion, mit der folgenden Eigenschaft:

> Zu jeder einstelligen partiell–rekursiven Funktion f existiert eine Nummer t derart, daß für alle $i \in \mathbb{N}$
>
> $$f\,(i) = A\,(t,i)$$

Dafür sagt man auch, A ist eine *Numerierung* der einstelligen partiell–rekursiven Funktionen.

A ist partiell–rekursiv, also auch TURING–berechenbar. Ist nun T eine TURING–Maschine, die A berechnet, so kann T offenbar alle einstelligen partiell–rekursiven Funktionen berechnen: Um die Funktion f mit der Nummer t für das Argument i zu berechnen, startet T in der Situation

$$\ldots \star \underbrace{|\cdots|}_{t+1} \star \underbrace{|\cdots|}_{i+1} \star \cdots$$
$$\triangle$$

Es gibt also eine „universelle“ TURING–Maschine, die alle einstelligen berechenbaren Funktionen berechnen kann. Bedenkt man, daß durch Gödelnumerierung der n-Tupel $[a_1, \ldots, a_n]$ als $2^{a_1} \cdots q_{n-1}^{a_n}$ jede n-stellige berechenbare Funktion auf eine einstellige berechenbare reduziert werden kann, so ist klar, daß die universelle TURING–Maschine jede Berechnung durchführen kann, deren Nummer ihr mitgeteilt wird.

$A\,(t,i)$ ist eine partielle Funktion, sie ist z. B. nicht definiert, wenn t nicht Gödelzahl einer TURING–Maschine ist. Wir betrachten

$$h\,(t,i) := \begin{cases} A(t,i) & \text{,falls ein } j \text{ existiert mit } \beta(t,i,j) = 0 \\ 0 & \text{,sonst.} \end{cases}$$

Wenn es ein j mit $\beta(t,i,j) = 0$ gibt, so gibt es auch ein kleinstes solches j, also ist $A(t,i)$ definiert, folglich ist h total.

Wir zeigen: h ist nicht berechenbar, d. h. h ist nicht allgemein–rekursiv.

Beweis
Wäre $h \in$ REK, dann wäre die Funktion g mit

$$g(n) := h(n,n) + 1$$

ebenfalls allgemein–rekursiv, also partiell–rekursiv, folglich gäbe es eine Nummer $t^\star$ mit

$$g(i) = A\,(t^\star, i) \text{ für alle } i \in \mathbb{N}.$$

Nun hätten wir für $t = i = t^\star$

$$h\,(t^\star, t^\star) + 1 = g\,(t^\star) = A\,(t^\star, t^\star) = h\,(t^\star, t^\star) \quad \lightning$$

Wir erhalten also einen Widerspruch, d. h. h ist nicht berechenbar. $\square$

Woran liegt es, daß h nicht berechenbar ist?

Betrachten wir die Menge

$$M := \{[t, i] |\ \text{Es gibt ein } j \text{ mit } \beta(t, i, j) = 0\}$$

der Argumentenpaare, wo h denselben Wert wie A hat, d. h. wo A definiert ist:

> Die Menge M ist nicht entscheidbar, d. h. die charakteristische Funktion χ_M ist nicht allgemein–rekursiv.

Sonst wäre

$$h(t, i) = \overline{\mathrm{sg}}\,(\chi_M(t, i) \cdot \alpha\,(\mu j\,(\beta(t, i, j) = 0)))$$

allgemein–rekursiv (hierbei gilt die Konvention „Null · Irgendwas = Null“, auch wenn „Irgendwas“ nicht definiert ist).

Was bedeutet die Unentscheidbarkeit von M?

Ist t die Gödelzahl einer Turing–Maschine T, so ist $[t, i] \in M$ genau dann, wenn die Turing–Maschine T bei Start in der Situation

$$\star \underbrace{|\cdots|}_{i+1} \star \ \star\star\cdots$$
$$\triangle$$

irgendwann stoppt.

Daß M nicht entscheidbar ist, bedeutet also:

Es gibt keinen Algorithmus, der für eine Turing–Maschine und eine Anfangssituation entscheidet, ob die Maschine nach dem Start in dieser Situation jemals stoppt. Kurz: Das *Halteproblem* für Turing–Maschinen ist nicht entscheidbar (ist unlösbar).

Wenn also unsere Programmiersprache so ausdrucksfähig ist, daß alle partiell–rekursiven Funktionen programmiert werden können (oder alle Turing–Maschinen simuliert), dann kann es kein Programm geben, das Terminierung für alle Programme entscheidet.

Allgemein wird dieses Resultat durch den Satz von Rice wiedergegeben: Jedes nicht-triviale Problem für Turing–Maschinen ist unentscheidbar.

Satz von Rice
Es sei $\mathfrak{F}$ eine Menge von partiell–rekursiven Funktionen und $N_{\mathfrak{F}} = \{t | A(t, i) \in \mathfrak{F}\}$ die Menge der Gödelnummern der Funktionen aus $\mathfrak{F}$. Wenn $\mathfrak{F} \neq \emptyset$ und $\mathfrak{F} \neq$ PREK (dann ist das Problem $\{t \in N_{\mathfrak{F}} ? | t \in \mathbb{N}\}$ nichttrivial) so ist $\chi_{N_{\mathfrak{F}}} \notin$ REK.

Kapitel 4

Prädikatenlogik

Die tägliche Praxis des logischen Schließens zeigt, daß der mit dem Aussagenkalkül objektivierte Teilbereich (nämlich das aussagenlogische Schließen) wichtig ist, aber im allgemeinen nicht ausreicht. Als Unzulänglichkeit des Aussagenkalküls stellt sich dabei heraus, daß zuwenig von der inneren Struktur von Aussagen (nämlich nur die aussagenlogische Struktur, d. h. der Aufbau mittels Negation, Konjunktion, Alternative,. . .) widergespiegelt werden kann. Ein Schluß der Form

$$\frac{\text{Es gibt ein } x\text{, das nicht die Eigenschaft } E \text{ hat}}{\text{Es gilt nicht: Alle } x \text{ haben die Eigenschaft } E}$$

kann im Aussagenkalkül nicht verifiziert werden.

Was wird im Aussagenkalkül nicht widergespiegelt:

1. Aussagen betreffen Gegenstände, sie entstehen aus „Aussageformen“

   - „x ist gerade“
   - „k ist Primzahl“ (Aussageformen)
   - „x ist größer als z“

   Dies sind keine Aussagen, ihnen kann ein Wahrheitswert erst zugeordnet werden, wenn die Variablen x, k, z (z. B. an feste Werte) gebunden sind. „3 ist Primzahl“ ist eine wahre, „4 ist Primzahl“ eine falsche Aussage.

2. Aussagen betreffen Eigenschaften von und Beziehungen zwischen Gegenständen

   - „Die Menge $\mathbb{N}$ ist unendlich“
   - „Die Relation $<$ ist transitiv“

3. Aussagen können die Form von All- oder Existenzaussagen haben

   - „Jede natürliche Zahl ist Summe von vier Quadraten“
   - „Es gibt eine gerade Primzahl“

Logische Schlüsse, bei denen solche Strukturen eine Rolle spielen, können im Aussagenkalkül nicht überprüft werden, weil sie nicht widergespiegelt werden.

4.1 Grundbegriffe

Wir brauchen in unserem Modell insbesondere die Möglichkeit der *Quantifikation*, d. h. eines Überganges von einer Aussagenform zu einer Aussage durch Bindung der Variablen, wie zum Beispiel

„x ist Primzahl“ $\longrightarrow$ „alle x sind Primzahl“
„x ist gerade“ $\longrightarrow$ „es gibt ein x mit x ist gerade“

Damit entsteht die Frage: Was darf quantifiziert werden?

In unseren Beispielen ist stets die Variable x, also eine Variable für Gegenstände quantifiziert worden, d. h. eine Variable erster Stufe.

Das 5. PEANOsche Axiom lautet:

> Für jede Menge $M \subseteq \mathbb{N}$ gilt:
> Wenn $0 \in M$ und für alle k gilt: Wenn $k \in M$, so $(k+1) \in M$, dann ist $\mathbb{N} \subseteq M$

Offenbar wird hier auch eine Variable (nämlich M) quantifiziert, die über Mengen von Gegenständen, also sozusagen über Gegenständen zweiter Stufe, variiert. Man sagt daher, daß diese Aussage zur Prädikatenlogik der zweiten Stufe gehört.

Wir werden hier nur die Prädikatenlogik der ersten Stufe behandeln, die sogenannte *elementare* Prädikatenlogik, in der nur Variable für Gegenstände quantifiziert werden können. Dementsprechend nehmen wir in die Sprache des Prädikatenkalküls auch nur Variable für Gegenstände, nicht aber für Mengen von bzw. Relationen zwischen Gegenständen auf.

Also was wollen wir in unserem Modell widerspiegeln, d. h. in unserer Sprache ausdrücken?

a) Gegenstände (Individuen) $\longrightarrow$ Individuenzeichen (Individuenkonstanten)

   Beispiele
   Null, Eins $\longrightarrow$ 0, 1

b) Beziehungen, Eigenschaften zwischen (bzw. von) Individuen $\longrightarrow$ Attributenzeichen

   Beispiele
   Kleiner–Beziehung, Primzahleigenschaft $\longrightarrow$ $<$, Pz

c) funktionale Abhängigkeiten zwischen Individuen $\longrightarrow$ Funktionszeichen

   Beispiele
   Nachfolgerfunktion, Addition $\longrightarrow$ Nf, $+$

d) Quantifiziert werden $\longrightarrow$ Individuenvariable

e) aussagenlogische Struktur $\longrightarrow$ $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$

f) prädikatenlogische Struktur $\longrightarrow$ Quantoren

g) Gleichheitsattribut für Individuen $\longrightarrow$ $=$

In dieser Liste ist der Punkt f) unklar, was ist „prädikatenlogische Struktur“; welche prädikatenlogischen Verknüpfungen gibt es überhaupt?

Definition 4.1.1 (Individuenbereich, Attribut, Funktion)

1. *Jede nichtleere Menge I heißt* Individuenbereich.
2. α^i *ist ein i-stelliges* Attribut *über I, wenn:*
$$\alpha^i : \bigtimes_i I \to \{\mathrm{W}, \mathrm{F}\}$$
Die Menge $R_{\alpha^i} := \left\{[x_1, \ldots, x_i] \,|\alpha^i(x_1, \ldots, x_i) = \mathrm{W}\right\}$ wird die i-stellige Relation *von α^i genannt.*
3. *Das* Gleichheitsattribut Id_I *über I ist für $x, y \in I$ definiert durch*
$$\mathrm{Id}_I(x, y) = \begin{cases} \mathrm{W}, & \textit{falls } x \textit{ gleich } y \textit{ ist} \\ \mathrm{F}, & \textit{sonst} \end{cases}$$
4. ϕ^i *ist eine i-stellige* Funktion *über I, wenn*
$$\phi^i : \bigtimes_i I \to I$$

Bemerkung

1. Es ist sehr wichtig, daß die Grundmenge I von Gegenständen, über die wir Aussagen formulieren, als nichtleer vorausgesetzt ist, andernfalls würde der Schluß
$$\frac{\text{Alle } x \text{ haben die Eigenschaft } E}{\text{Es gibt ein } x \text{ mit der Eigenschaft } E}$$
nicht richtig sein.
2. Attribute ordnen also Tupeln von Gegenständen Wahrheitswerte zu. Einstellige Attribute werden auch Prädikate (Eigenschaften) genannt, die zugehörige einstellige Relation ist dabei einfach eine Menge von Gegenständen.

Was ist nun eine prädikatenlogische Verknüpfung?

Eine aussagenlogische Verknüpfung (wie z. B. die Konjunktion) ordnet Aussagen eine neue Aussage zu, im extensionalen Fall wird diese Zuordnung durch eine Wahrheitsfunktion charakterisiert.

Eine prädikatenlogische Verknüpfung (erster Stufe) ordnet Aussageformen (mit Variablen für Gegenstände) Aussagen zu und wird (im extensionalen Fall) beschrieben durch ein Attributenattribut, d. h. eine Abbildung, die Attributen einen Wahrheitswert zuordnet:

Definition 4.1.2 (Prädikatenlogische Wahrheitsfunktionen)
Prädikatenlogische Wahrheitsfunktionen ordnen Tupeln von Attributen einen Wahrheitswert zu:

$$\begin{aligned}
\mathrm{Om}\left(\alpha^1\right) &= \begin{cases} \mathrm{W}, & \textit{falls für alle } x \in I \textit{ gilt: } \alpha^1(x) = \mathrm{W} \\ \mathrm{F}, & \textit{sonst} \end{cases} \\
\mathrm{Ex}\left(\alpha^1\right) &= \begin{cases} \mathrm{W}, & \textit{falls für wenigstens ein } x \in I \textit{ gilt: } \alpha^1(x) = \mathrm{W} \\ \mathrm{F}, & \textit{sonst} \end{cases} \\
\mathrm{gv}\left(\alpha^1, \beta^1\right) &= \begin{cases} \mathrm{W}, & \textit{falls } \mathrm{card}\left(\left\{x | x \in I \wedge \alpha^1(x) = \mathrm{W}\right\}\right) = \\ & \qquad \mathrm{card}\left(\left\{x | x \in I \wedge \beta^1(x) = \mathrm{W}\right\}\right) \\ \mathrm{F}, & \textit{sonst} \end{cases} \\
\mathrm{Ex}^n!!\left(\alpha^1\right) &= \begin{cases} \mathrm{W}, & \textit{falls es genau } n \textit{ Elemente } x \textit{ aus } I \textit{ mit} \\ & \alpha^1(x) = \mathrm{W} \textit{ gibt} \\ \mathrm{F}, & \textit{sonst} \end{cases}
\end{aligned}$$

In unserem Kalkül werden wir nur die einstelligen Attributenattribute Om und Ex verwenden. gv läßt sich nicht durch Om und Ex ausdrücken.

4.2 Syntax und Semantik des Prädikatenkalküls

4.2.1 Die Sprache des Prädikatenkalküls der ersten Stufe

Als Alphabet des Prädikatenkalküls verwenden wir die Menge

$$Y \quad := \quad \{\neg, \wedge, \vee, \rightarrow, \leftrightarrow,), (, \text{,}, =, \forall, \exists, a, x, A, F, \star, |\}$$

die offenbar aus 17 Grundzeichen besteht.

Die Wörter der Form

$$x\underbrace{\star\cdots\star}_{i}$$

bezeichnen wir kurz durch x_i und nennen sie Individuenvariable. Die Menge der Individuenvariablen ist $X = \{x_i | i \in \mathbb{N}\}$.

Die Wörter der Form

$$A\underbrace{\star\cdots\star}_{i}\underbrace{|\cdots|}_{j}$$

mit $i \geq 0, j \geq 1$ bezeichnen wir als (j-stelliges) Attributenzeichen und schreiben dafür kürzer A_i^j.

Die Wörter der Form

$$F\underbrace{\star\cdots\star}_{i}\underbrace{|\cdots|}_{j}$$

mit $i \geq 0, j \geq 1$ werden kurz als F_i^j geschrieben und (j-stelliges) Funktionszeichen genannt.

Die Wörter der Form

$$a\underbrace{\star\cdots\star}_{i}$$

geschrieben a_i werden Individuenzeichen bzw. Individuenkonstanten genannt.

Term
wird induktiv definiert:

A) Jede Individuenkonstante und jede Individuenvariable ist ein Term.

I) Ist F_i^j ein j-stelliges Funktionszeichen und sind $T_1, \ldots, T_j$ Terme, so ist $F_i^j(T_1, \ldots, T_j)$ ein Term

Beispiel
$F_1^2\left(F_5^1(x_j), a_3\right)$ ist ein Term.

Ausdruck
wird ebenfalls induktiv definiert:

A) (Prädikative Ausdrücke)

1. Sind T_1, T_2 Terme, so ist $T_1 = T_2$ ein Ausdruck (Gleichung)
2. Ist A_i^j ein j-stelliges Attributenzeichen und sind $T_1, \ldots, T_j$ Terme, so ist $A_i^j T_1 \cdots T_j$ ein Ausdruck.

I) 1. (Aussagenlogische Zusammensetzung)
Sind H_1, H_2 Ausdrücke, so auch $\neg H_1$, $(H_1 \wedge H_2)$, $(H_1 \vee H_2)$, $(H_1 \rightarrow H_2)$,
$(H_1 \leftrightarrow H_2)$.

2. (prädikatenlogische Zusammensetzung)
Ist H ein Ausdruck, x_j eine Variable und in H kommt x_j vor, aber keines der Wörter $\forall x_j, \exists x_j$, so sind $\forall x_j H, \exists x_j H$ Ausdrücke.

E) Sonst nichts

AUSD = Menge aller so gebildeten Ausdrücke
= Sprache des Prädikatenkalküls der ersten Stufe

Bemerkung: Elementare Sprache mit der Signatur $[K, P, F]$

In der Sprache des Prädikatenkalküls kommen unendlich viele Individuenkonstanten, Attributen– und Funktionszeichen vor. Für viele Anwendungen ist das nicht notwendig, wenn man z. B. die Arithmetik der natürlichen Zahlen formalisieren will, kommt man mit Zeichen für die Zahlen Null und Eins, die Kleiner–Beziehung und die Funktionen Addition und Multiplikation aus; eine solche Vorgabe nennt man eine *Signatur*. Eine Signatur $[K, P, F]$ besteht also aus einer Menge K von Individuenzeichen, einer Menge P von Attributenzeichen und einer Menge F von Funktionszeichen. In unserem Beispiel ist $K = \{a_0, a_1\}(= \{0, 1\})$, $P = \{A_0^2\}(= \{<\})$ und $F = \{F_0^2, F_1^2\}(= \{+, \cdot\})$.

Die elementare Sprache mit der Signatur $[K, P, F]$ ist dann die Menge aller Ausdrücke $H \in$ AUSD des Prädikatenkalküls der ersten Stufe, in denen höchstens Individuenkonstanten aus K, Attributenzeichen aus P und Funktionszeichen aus F vorkommen.

Beispiel
Der Ausdruck

$$\forall x_0 \exists x_1 \, (x_1 = N(x_0))$$

ist nicht Element unserer elementaren Sprache der Arithmetik, weil N kein Element von F ist.

4.2.2 Einige syntaktische Eigenschaften

Definition 4.2.1 (vollfreie und quantifizierte Individuenvariablen)
Es sei H ein Ausdruck, x_j eine Individuenvariable.

1. *x_j kommt in H* vollfrei *vor*
   (a) *x_j kommt in H vor.*
   (b) *Die Zeichenreihen $\forall x_j, \exists x_j$ kommen nicht in H vor.*
2. *x_j kommt in H an der n-ten Stelle* quantifiziert *vor, wenn*

(a) *x_j steht in H an der n-ten Stelle, und $n \geq 2$.*

(b) *An der $(n-1)$-ten Stelle in H kommt $\exists$ oder $\forall$ vor.*

Satz 4.2.2
H ist ein Ausdruck, in dem x_j an der n-ten Stelle quantifiziert vorkommt. Dann gibt es einen Ausdruck H_1 mit

1. *in H_1 kommt x_j vollfrei vor.*
2. *Wörtern Z_1, Z_2 mit $H = Z_1 Q x_j H_1 Z_2$, wobei $Q \in \{\forall, \exists\}$.*

Bevor wir den Satz beweisen, führen wir noch eine Bezeichnung ein:

Definition 4.2.3 (Wirkungsbereich eines Quantors)
Der im Satz 4.2.2 mit H_1 bezeichnete Ausdruck heißt Wirkungsbereich *des Quantors, der in H an der $(n-1)$-ten Stelle steht.*

Beweis des Satzes
durch Induktion über den Aufbau des Ausdrucks H

A) Wenn H prädikativer Ausdruck ist, d. h. $T_1 = T_2$ oder $A_j T_1 \ldots T_j$, so kommt x_j nicht quantifiziert in H vor, also ist nichts zu zeigen.

I) 1. (H durch aussagenlogische Zusammensetzung entstanden)
Wenn $H \equiv \neg H_0$, so ist x_j in H_0 an der $(n-1)$-ten Stelle quantifiziert; nach Induktionsvoraussetzung hat H_0 die Form

$$H_0 \equiv Z_1 Q x_j H_1 Z_2$$

also

$$H \equiv \neg Z_1 Q x_j H_1 Z_2.$$

Wenn $H \equiv (H_0 \otimes H_{00})$ ist, wobei $\otimes$ eines der Zeichen $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$ ist, so befindet sich die n-te Stelle (an der x_j quantifiziert steht) entweder im Ausdruck H_0 oder in H_{00}, dieser kann nach Induktionsvoraussetzung als $Z_1 Q x_j H_1 Z_2$ geschrieben werden, womit sich eine entsprechende Zerlegung für H ergibt, etwa bei $H_0 \equiv Z_1 Q x_j H_1 Z_2$ als

$$H = \underbrace{(Z_1}_{Z_1'} Q x_j H_1 \underbrace{Z_2 \otimes H_2)}_{Z_2'}$$

2. ($H \equiv Q' x_i H_{00}$ durch prädikatenlogische Zusammensetzung entstanden)
Entweder ist $x_i = x_j$, also $H \equiv Q x_j H_{00}$, dann ist $n = 2$ und $Q' = Q$, $H_1 = H_{00}$ und $Z_1 = Z_2 =$ leeres Wort, oder es gilt $x_i \neq x_j$, dann kommt x_j in H_{00} an der $(n-2)$-ten Stelle quantifiziert vor. Nach Induktionsvoraussetzung ist $H_{00} \equiv Z_1 Q x_j H_1 Z_2$, also $H \equiv Q' x_i Z_1 Q x_j H_1 Z_2$.

□

Definition 4.2.4 (gebundenes und freies Vorkommen, Ausdruck)

1. *x_j kommt in H an der n-ten Stelle* gebunden *vor, wenn*

   (a) $l(H) \geq n \geq 2$

   (b) *x_j steht in H an der n-ten Stelle.*
   (c) *x_j kommt an der n-ten Stelle quantifiziert vor oder steht im Wirkungsbereich eines mit x_j versehenen Quantors.*

2. *x_j kommt in H an der n-ten Stelle* frei *vor, wenn*
   (a) *x_j steht in H an der n-ten Stelle,*
   (b) *x_j ist an der n-ten Stelle nicht gebunden.*

3. *H heißt* Aussage *(abgeschlossen), wenn H keine freien Variablen enthält.*

Beispiel
Wir betrachten den Ausdruck $H \equiv \exists x_0 \left(\forall x_1 \left(x_0 = x_1 \wedge A_1^2 x_1 x_2\right) \wedge A_2^3 x_0 x_1 x_2\right)$:
Zuerst kennzeichnen wir die Wirkungsbereiche der Quantoren:

$$\exists x_0 \underline{\Big(\forall x_1 \underline{\left(x_0 = x_1 \wedge A_1^2 x_1 x_2\right)} \wedge A_2^3 x_0 x_1 x_2\Big)}$$

Wir numerieren nun die einzelnen Bestandteile des Ausdrucks H:

$$\begin{array}{cccccccccccccccccccc} \exists & x_0 & (& \forall & x_1 & (& x_0 & = & x_1 & \wedge & A_1^2 & x_1 & x_2 &) & \wedge & A_2^3 & x_0 & x_1 & x_2 &) \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 \end{array}$$

Die Variable x_0 kommt an der 2., 7. und 17. Stelle gebunden, an der Stelle 2 sogar quantifiziert vor. x_1 ist an 5., 9. und 12. Stelle gebunden und an 5. Stelle quantifiziert. An Stelle 18 ist x_1 frei. Die Variable x_2 kommt an 13. und 19. Stelle frei vor, x_2 ist sogar vollfrei in H.

Am Beispiel des allquantifizierten x_1 an 5. Stelle zerlegen wir den Ausdruck H in Z_1, Z_2, Q und H_1:

$$\underbrace{\exists x_0(}_{Z_1} \underbrace{\forall}_{Q} x_1 \underbrace{\left(x_0 = x_1 \wedge A_1^2 x_1 x_2\right)}_{H_1} \underbrace{\wedge A_2^3 x_0 x_1 x_2)}_{Z_2}$$

und x_1 kommt in H_1 vollfrei vor.

4.2.3 Semantik elementarer Sprachen

Interpretation
$[I, g]$ ist eine *Interpretation* der elementaren Sprache mit der Signatur $[K, P, F]$, wenn

1. $I \neq \emptyset$ ein nichtleerer Individuenbereich ist
2. g eine Abbildung ist, die
   (a) jedem Individuenzeichen a_i ein Element $g(a_i) \in I$
   (b) jedem Funktionszeichen F_i^k eine k-stellige Funktion $g\left(F_i^k\right) : \bigtimes_k I \to I$
   (c) jedem Attributzeichen A_i^k ein k-stelliges Attribut $g\left(A_i^k\right) : \bigtimes_k I \to \{\mathrm{W}, \mathrm{F}\}$

   zuordnet.

Man bezeichnet g auch als die *Interpretation* von AUSD über I. Zu jedem I, g gehört die interpretierende Algebra $\left[I, \left\{g\left(F_i^k\right) | i \in \mathbb{N}\right\}, R_{g(A^k)}, \ldots\right]$

Belegung
Die Abbildung f heißt *Belegung* über I, wenn f jeder Individuenvariablen x_i ein Individuum $f(x_i) \in I$ zuordnet.

Um Terme auszuwerten, d. h. das Element aus I zu erhalten, das ein Term T bei der Interpretation I, g und der Belegung f beschreibt, führen wir folgende Funktion ein:

element
$\text{element}_{I,g}(T, f)$ wird wie folgt induktiv definiert:

A) Für Variable x_i: $\text{element}_{I,g}(x_i, f) = f(x_i)$
Für Konstanten a_i: $\text{element}_{I,g}(a_i, f) = g(a_i)$

I) $$\text{element}_{I,g}\left(F_i^j(T_1, \ldots, T_j), f\right) = g\left(F_i^j\right)(\text{element}_{I,g}(T_1, f), \ldots, \text{element}_{I,g}(T_j, f)).$$

Analog wie im Aussagenkalkül bestimmen wir nun den Wert eines Ausdrucks:

wert
$\text{wert}_{I,g}(H, f)$ wird induktiv über den Aufbau von Ausdrücken definiert:

A) $\text{wert}_{I,g}(T_1 = T_2, f) = \text{Id}_I(\text{element}_{I,g}(T_1, f), \text{element}_{I,g}(T_2, f))$
$\text{wert}_{I,g}\left(A_i^k T_1 \ldots T_k, f\right) = g\left(A_i^k\right)(\text{element}_{I,g}(T_1, f), \ldots, \text{element}_{I,g}(T_k, f))$

I) $\text{wert}_{I,g}(\neg H, f) = \text{non}\ (\text{wert}_{I,g}(H, f))$
$\text{wert}_{I,g}((H_1 \wedge H_2), f) = \text{et}(\text{wert}_{I,g}(H_1, f), \text{wert}_{I,g}(H_2, f))$
$\text{wert}_{I,g}((H_1 \vee H_2), f) = \text{vel}(\text{wert}_{I,g}(H_1, f), \text{wert}_{I,g}(H_2, f))$
$\text{wert}_{I,g}((H_1 \rightarrow H_2), f) = \text{seq}(\text{wert}_{I,g}(H_1, f), \text{wert}_{I,g}(H_2, f))$
$\text{wert}_{I,g}((H_1 \leftrightarrow H_2), f) = \text{aeq}(\text{wert}_{I,g}(H_1, f), \text{wert}_{I,g}(H_2, f))$

$\text{wert}_{I,g}(\forall x_j H, f) = \text{Om}\left(\alpha^1_{H,I,g,f,x_j}\right)$
$\text{wert}_{I,g}(\exists x_j H, f) = \text{Ex}\left(\alpha^1_{H,I,g,f,x_j}\right)$

wobei:
$\alpha^1_{H,I,g,f,x_j}(y) = \text{wert}_{I,g}\left(H, [f]_y^j\right)$
mit $[f]_y^j(x_i) = \begin{cases} y, & \text{falls } i = j \\ f(x_i), & \text{sonst} \end{cases}$

Beispiel
Wir bestimmen den Wert des Ausdrucks $H \equiv \exists x_0 A_0^2 x_0 x_1$, $I = \mathbb{N}$, $g\left(A_0^2\right) =$ „ $<$ “, $f(x_1) = 5$.

Dann ist
$$\text{wert}_{I,g}\left(\exists x_0 A_0^2 x_0 x_1, f\right) = \text{Ex}\left(\alpha^1_{A_0^2 x_0 x_1, \mathbb{N}, g, f, x_0}\right)$$

wobei für $k \in \mathbb{N}$

$$\begin{aligned} \alpha^1_{A_0^2 x_0 x_1, \mathbb{N}, g, f, x_0}(k) &= \text{wert}\left(A_0^2 x_0 x_1, [f]_k^0\right) \\ &= g\left(A_0^2\right)\left([f]_k^0(x_0), [f]_k^0(x_1)\right) \\ &= \text{„} < \text{“}(k, 5) = \begin{cases} \text{W}, & \text{falls } k < 5 \\ \text{F}, & \text{sonst} \end{cases} \end{aligned}$$

Es gibt also ein $k \in I = \mathbb{N}$ mit

$$\alpha^1_{A_0^2 x_0 x_1, \mathbb{N}, g, f, x_0}(k) = \mathrm{W}$$

d. h.

$$\mathrm{wert}_{I,g}\left(\exists x_0 A_0^2 x_0 x_1, f\right) = \mathrm{Ex}\left(\alpha^1_{A_0^2 x_0 x_1, \mathbb{N}, g, f, x_0}\right) = \mathrm{W}.$$

Bei einer Belegung f mit $f(x_1) = 0$ ist jedoch

$$\mathrm{wert}_{I,g}\left(\exists x_0 A_0^2 x_0 x_1, f\right) = \mathrm{F}.$$

4.2.4 Weitere semantische Begriffe

Erfüllbarkeit
Sei $H \in \mathrm{AUSD}$

1. *f erfüllt H in der Interpretation g über Individuenbereich I*; $(f \mathrm{erf}_{I,g} H)$, wenn $\mathrm{wert}_{I,g}(H, f) = \mathrm{W}$.
2. H ist *erfüllbar in g über I*, wenn es eine Belegung $f : X \to I$ gibt, mit $\mathrm{wert}_{I,g}(H, f) = \mathrm{W}$.
3. H heißt *erfüllbar über I* $(H \in \mathrm{ef}_I,\ \mathrm{ef}_I H)$, wenn es g, f gibt, mit: $\mathrm{wert}_{I,g}(H, f) = \mathrm{W}$.
4. H heißt *erfüllbar* $(H \in \mathrm{ef},\ \mathrm{ef} H)$, wenn es I, g, f gibt, mit $\mathrm{wert}_{I,g}(H, f) = \mathrm{W}$.

Beispiel
Wir betrachten den Ausdruck $H \equiv \exists x_0 \exists x_1\ \neg x_0 = x_1$. Wenn I eine Einermenge ist, so ist H nicht erfüllbar über I. Der Ausdruck $H \equiv \forall x_0 \forall x_1\ x_0 = x_1$ ist dagegen erfüllbar über I gdw. I Einermenge.

Gültigkeit
Sei $H \in \mathrm{AUSD}$

1. H ist *allgemeingültig in der Interpretation g über Individuenbereich I* $\left(H \in \mathrm{ag}_{I,g}\right)$, wenn für alle Belegungen f : $\mathrm{wert}_{I,g}(H, f) = \mathrm{W}$.
2. H ist *allgemeingültig über I* $(H \in \mathrm{ag}_I,\ \mathrm{ag}_I H)$, wenn für alle g, f gilt: $\mathrm{wert}_{I,g}(H, f) = \mathrm{W}$.
3. H ist *allgemeingültig* $(H \in \mathrm{ag},\ \mathrm{ag} H)$, wenn für alle I, f, g gilt: $\mathrm{wert}_{I,g}(H, f) = \mathrm{W}$.

Für das folgende erinnern wir daran, daß ein Ausdruck H als *Aussage* bezeichnet wird, wenn in H keine freien Variablen vorkommen.

Satz 4.2.5
Wenn H Aussage ist, dann hängt $\mathrm{wert}_{I,g}(H, f)$ *nicht von der Belegung f ab.*

Definition 4.2.6 (wahre und falsche Aussagen)
Sei H Aussage.

1. *H heißt* wahr *in der Interpretation g über I, wenn H erfüllbar in g über I ist.*

2. *H heißt* falsch *in g über I, wenn $\neg H$ erfüllbar in g über I ist.*

Zentrales Lemma
Der Wahrheitswert $\text{wert}_{I,g}(H, f)$ hängt nur ab von der Interpretation der Zeichen, die in H vorkommen, und von der Belegung der Variablen, die in H frei vorkommen

Mit anderen Worten: Kommt die Variable x_j in H nicht frei vor (also quantifiziert oder gebunden), so gilt für alle $a \in I$:

$$\text{wert}_{I,g}(H, f) = \text{wert}_{I,g}\left(H, [f]_a^j\right)$$

Für Aussagen gilt weiterhin:

Satz 4.2.7
Ist H Aussage, so gilt: H ist erfüllbar in g über I gdw. H allgemeingültig in g über I ist. Enthält H zudem keine Zeichen, so ist H allgemeingültig über I gdw. H erfüllbar über I ist.

Es gilt

Folgerung 4.2.8

1. *$H \in \text{ef}_I$ gdw. $\neg H \notin \text{ag}_I$*
2. *$H \in \text{ef}$ gdw. $\neg H \notin \text{ag}$*
3. $\text{ag}_I \subset \text{ef}_I$, $\text{ag} \subset \text{ef}$

Für die dritte Aussage ist die Festsetzung wichtig, daß $I \neq \emptyset$ ist, denn $\neg x_j = x_j$ würde über $I = \emptyset$ gültig sein.

Die Frage nach der Gültigkeit eines Ausdrucks ist durch

$$H \in \text{ag} \quad \text{gdw.} \quad \neg H \notin \text{ef}$$

auf die Frage nach der Erfüllbarkeit seiner Negation reduziert.

Um prädikatenlogische Schlußregeln zu verifizieren, brauchen wir also ein Verfahren, um die Erfüllbarkeit von Ausdrücken zu entscheiden. Die Frage „$H \in \text{ef}$?“ ist das *Entscheidungsproblem des Prädikatenkalküls.*

Wir wissen, daß im Aussagenkalkül die Mengen ef, ag entscheidbar sind. Wie sieht es nun im Prädikatenkalkül aus? Gibt es ein solches Verfahren?

Wir werden zeigen:

Wenn I eine unendliche Menge ist, so ist ef_I nicht entscheidbar.

Es gibt also keinen Algorithmus, der für jedes $H \in \text{AUSD}$ nach endlicher Zeit abbricht mit dem Resultat „$H \in \text{ef}$“ bzw. „$H \notin \text{ef}$“.

4.3 Die Unentscheidbarkeit des Prädikatenkalküls

Wir wollen beweisen, daß die Menge ef_I der über einem unendlichen Individuenbereich erfüllbaren Ausdrücke des Prädikatenkalküls nicht entscheidbar ist, mit anderen Worten, daß die charakteristische Funktion χ_{ef_I} nicht allgemein–rekursiv ist. Dazu bilden wir die partiell–rekursiven Funktionen im Prädikatenkalkül nach (repräsentieren sie durch Ausdrücke) und numerieren die Ausdrücke.

1. Darstellung zahlentheoretischer Funktionen im Prädikatenkalkül

Dazu brauchen wir zunächst eine Möglichkeit, natürliche Zahlen im Prädikatenkalkül darzustellen. Wir erinnern uns daran, daß die Zahl *Drei* die Menge aller Mengen mit genau drei Elementen ist, also jede Dreiermenge als Repräsentant der Zahl *Drei* herhalten kann. Eine Menge M kann beschrieben werden als ein einstelliges Attribut α_M, das genau auf die Elemente der Menge M zutrifft:

$$x \in M \quad \text{gdw.} \quad \alpha_M(x)$$

Für Attribute haben wir Zeichen in unserem Kalkül, wir können also festlegen, daß das einstellige Attributenzeichen A_j^1 in einer Interpretation g die Zahl n repräsentiert, wenn das Attribut $g\left(A_j^1\right)$ auf genau n Elemente von I zutrifft. Dies können wir durch Anzahlaussagen ausdrücken:

Es sei $H \in \mathrm{AUSD}$, x eine Individuenvariable, die in H frei vorkommt. Dann sei $\exists i!!xH(x)$ Abkürzung für den Ausdruck

$$\begin{array}{ll}
\neg\exists x H\,(x) & \text{falls } i = 0 \\
\exists x_1\,(H\,(x_1) \wedge \forall x_2\,(H\,(x_2) \rightarrow x_2 = x_1)) & \text{falls } i = 1 \\
\vdots & \\
\exists x_1 \cdots \exists x_i \big(H\,(x_1) \wedge \cdots \wedge H\,(x_i) \wedge \neg x_1 = x_2 \wedge \cdots & \text{für beliebiges } i \\
\cdots \neg x_1 = x_3 \wedge \cdots \wedge \neg x_{i-1} = x_i \wedge \forall x_{i+1}\,(H\,(x_{i+1}) \rightarrow & \\
x_{i+1} = x_1 \vee \cdots \vee x_{i+1} = x_i)\big) &
\end{array}$$

Offenbar gilt:

> $\mathrm{wert}_{I,g}\,(\exists i!!xH, f) = \mathrm{W}$ genau dann, wenn es genau i Elemente $a_1, \ldots, a_i \in I$ gibt mit $\mathrm{wert}_{I,g}\left(H, [f]_{a_i}^x\right) = \mathrm{W}$. Deshalb liest man den Ausdruck $\exists i!!xH(x)$ als „es gibt genau i Elemente x mit $H(x)$“.

Ist A_j^1 ein einstelliges Attributenzeichen, dann beschreibt also $\exists n!!xA_j^1x$ die Zahl n in dem Sinne, daß der Ausdruck genau in den Interpretationen g wahr ist, bei denen das Attribut $g\left(A_j^1\right)$ auf genau n Elemente zutrifft.

Definition 4.3.1
Die Funktion $\phi : \mathbb{N}^m \rightarrow \mathbb{N}$ heißt durch $H \in \mathrm{AUSD}$ bzgl. $\left\{A_1^1, \ldots, A_{m+1}^1\right\}$ in I dargestellt, wenn für alle $j_1, \ldots, j_{m+1} \in \mathbb{N}$ gilt:

$$\begin{array}{l}
\phi\,(j_1, \ldots, j_m) = j_{m+1} \ \textit{gdw.} \\
\qquad (H \wedge \exists j_1!!xA_1^1x \wedge \cdots \wedge \exists j_m!!xA_m^1x \wedge \exists j_{m+1}!!xA_{m+1}^1x) \in \mathrm{ef}_I
\end{array}$$

Es ist klar, daß eine solche Darstellung nur in einem unendlichen Individuenbereich I existieren kann, weil bei $\mathrm{card}(I) = n \in \mathbb{N}$ der Ausdruck $\exists(n+1)!!xA_j^1x$ nicht über I erfüllbar ist.

Beispiele

1. Nachfolgerfunktion $N(x) = x + 1$
   Die Nachfolgerfunktion N wird durch
   $$H \equiv \exists x_1 \left(\neg A_1^1 x_1 \wedge \forall x_2 \left(A_2^1 x_2 \leftrightarrow \left(A_1^1 x_2 \vee x_1 = x_2\right)\right)\right)$$
   bzgl. $\left\{A_1^1, A_2^1\right\}$ dargestellt.
2. Konstanten $C_i^n(j_1, \ldots, j_n) = i$
   Die Konstante C_i^n wird durch
   $$H \equiv \exists i!!x A_{n+1}^1 x$$
   bzgl. $\left\{A_1^1, \ldots, A_{n+1}^1\right\}$ dargestellt.
3. Argumentauswahlfunktionen $I_i^n(j_1, \ldots, j_n) = j_i$
   Die Argumentauswahlfunktion I_i^n wird durch
   $$H \equiv \forall x \left(A_i^1 x \leftrightarrow A_{n+1}^1 x\right)$$
   bzgl. $\left\{A_1^1, \ldots, A_{n+1}^1\right\}$ dargestellt.

Folgerung 4.3.2
Die ANF–Funktionen sind im Prädikatenkalkül darstellbar.

Man zeigt durch Induktion

Satz 4.3.3
Alle partiell–rekursiven Funktionen können im Prädikatenkalkül dargestellt werden.

2. Numerierung (Gödelisierung) von AUSD
Wir numerieren unser Alphabet wie folgt:

$\xi \in X:$	$\neg$	$\wedge$	$\vee$	$\rightarrow$	$\leftrightarrow$	(	)	,	=	$\forall$	$\exists$	a	x	A	F	$\star$	|
$N(\xi):$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

Definition 4.3.4 (Gödelzahl eines Wortes über dem Alphabet X)
Für $\xi_1 \xi_2 \cdots \xi_n \in W(X)$ sei
$$\text{gö}(\xi_1 \ldots \xi_n) = \prod_{j=0}^{n-1} q_j^{N(\xi_j)}$$
die Gödelzahl eines Wortes über Alphabet X.

Beispiel
$\text{gö}(x = x\star) = 2^{13} \cdot 3^9 \cdot 5^{13} \cdot 7^{16}$

Satz 4.3.5

1. *Die Funktion* gö *ist eineindeutig.*
2. *Die charakteristische Funktion χ_{AUSD} mit*
   $$\chi_{\text{AUSD}}(n) = \begin{cases} 0, & \textit{falls es } H \in \text{AUSD } \textit{gibt, mit } n = \text{gö}(H) \\ 1, & \textit{sonst} \end{cases}$$
   ist primitiv-rekursiv.

Es ist also „$H \in \mathrm{ef}_I$“ logisch gleichwertig mit „$\text{gö}(H) \in \text{gö}\,(\mathrm{ef}_I)$“, d. h. genau dann existiert ein Algorithmus, der das Entscheidungsproblem $\{H \in \mathrm{ef}_I ? | H \in \mathrm{AUSD}\}$ löst, wenn die charakteristische Funktion χ der Menge der Gödelzahlen der Ausdrücke aus ef_I

$$\chi\,(n) = \begin{cases} 0, & \text{falls } n = \text{gö}\,(H) \text{ für } H \in \mathrm{ef}_I \\ 1, & \text{sonst} \end{cases}$$

berechenbar ist.

Wir zeigen, daß χ nicht allgemein–rekursiv ist, nach der CHURCHschen These also nicht berechenbar, sobald I unendlich ist.

Satz 4.3.6
Ist I ein unendlicher Individuenbereich, dann ist χ nicht rekursiv.

Beweis
Wir nehmen an, daß χ allgemein–rekursiv ist.

Zu $H \in \mathrm{AUSD}$ sei

$$S(H) \equiv H \wedge \exists \text{gö}(H)!!x_1 A_1^1 x_1 \wedge \exists 0!!x_1 A_2^1 x_1$$

Dieser Ausdruck kann als Darstellung einer einstelligen Funktion verwendet werden, die der Gödelzahl von H die Zahl 0 zuordnet. Wäre die dargestellte Funktion gerade χ, so könnte $S(H)$ gerade als „H ist erfüllbar über I“ gelesen werden.

Wir definieren die zahlentheoretische Funktion σ durch

$$\sigma\,(n) := \begin{cases} \text{gö}\,(S\,(H))\,, & \text{falls } n = \text{gö}\,(H) \\ 0, & \text{sonst (d. h. } \chi_{\mathrm{AUSD}}\,(n) = 1) \end{cases}$$

Man zeigt leicht:

Lemma
$\sigma\,(n)$ ist allgemein-rekursiv.

Für $H \in \mathrm{AUSD}$ gilt:

$$H \in \mathrm{ef}_I \quad \text{gdw.} \quad \chi\,(\text{gö}(H)) = 0$$

also speziell für Ausdrücke der Form $S(H)$

$$\begin{aligned} S(H) \in \mathrm{ef}_I \quad & \text{gdw.} \quad \chi\,(\text{gö}\,(S(H))) = 0 \\ & \text{gdw.} \quad \chi\,(\sigma\,(\text{gö}(H))) = 0 \end{aligned}$$

d. h.

$$S(H) \notin \mathrm{ef}_I \quad \text{gdw.} \quad \overline{\mathrm{sg}}\,(\chi\,(\sigma\,(\text{gö}(H)))) = 0$$

Wir setzen

$$\psi\,(n) := \overline{\mathrm{sg}}\,(\chi\,(\sigma\,(n)))$$

und offenbar ist ψ allgemein–rekursiv.

Weil I unendlich ist, kann ψ im Prädikatenkalkül (bzgl. I) dargestellt werden, es gibt also ein $H^\star \in \mathrm{AUSD}$ derart, daß für alle $m, n \in \mathbb{N}$ gilt:

$$\psi(m) = n \quad \text{gdw.} \quad \left(H^\star \wedge \exists m!!x A_1^1 x \wedge \exists n!!x A_2^1 x\right) \in \mathrm{ef}_I$$

Folglich ist

$$\psi(m) = 0 \quad \text{gdw.} \quad \left(H^\star \wedge \exists m!!xA_1^1x \wedge \exists 0!!xA_2^1x\right) \in \mathrm{ef}_I$$

und insbesondere für $m = \text{gö}(H)$ ist

$$\psi\left(\text{gö}(H)\right) = 0 \quad \text{gdw.} \quad \left(H^\star \wedge \exists \text{gö}(H)!!xA_1^1x \wedge \exists 0!!xA_2^1x\right) \in \mathrm{ef}_I$$

also ist

$$S(H) \notin \mathrm{ef}_I \quad \text{gdw.} \quad \left(H^\star \wedge \exists \text{gö}(H)!!xA_1^1x \wedge \exists 0!!xA_2^1x\right) \in \mathrm{ef}_I$$

Nun setzen wir für H speziell $H^\star$ ein und erhalten einen Widerspruch:

$$S(H^\star) \notin \mathrm{ef}_I \quad \text{gdw.} \quad \underbrace{\left(H^\star \wedge \exists \text{gö}(H^\star)!!xA_1^1x \wedge \exists 0!!xA_2^1x\right)}_{= S(H^\star)} \in \mathrm{ef}_I \quad \lightning$$

Also ist χ nicht rekursiv, d. h. ef_I ist nicht entscheidbar. □

Wie steht es mit der Entscheidbarkeit der Menge aller erfüllbaren Ausdrücke

$$\mathrm{ef} = \bigcup_{I \neq \emptyset} \mathrm{ef}_I \qquad ?$$

Auch diese Menge ist nicht entscheidbar.

Um das einzusehen, betrachten wir den Ausdruck H_∞:

$$\begin{aligned} H_\infty \quad \equiv \quad & (\forall x_1 \neg A_0^2 x_1 x_1 \\ & \wedge \forall x_1 \forall x_2 \forall x_3 (A_0^2 x_1 x_2 \wedge A_0^2 x_2 x_3 \rightarrow A_0^2 x_1 x_3) \\ & \forall x_1 \exists x_2 A_0^2 x_1 x_2) \end{aligned}$$

Es gilt:

Lemma 1
Für alle $I \neq \emptyset$ gilt $\mathrm{ef}_I H_\infty$ gdw. I unendlich ist.

weil nämlich nur in einem unendlichen Bereich eine binäre Relation existiert, die irreflexiv und transitiv (also irreflexive Halbordnung) ist und kein maximales Element hat.

Lemma 2
Wenn ef *entscheidbar ist, so ist auch* ef_{N} *entscheidbar.*

Beweis
Wir nehmen also an, daß ein Algorithmus zur Lösung von $\{H \in \mathrm{ef}? | H \in \mathrm{AUSD}\}$ gegeben ist.

Um für einen Ausdruck $H \in \mathrm{AUSD}$ herauszufinden, ob $H \in \mathrm{ef}_{\mathsf{N}}$ (also im Bereich der natürlichen Zahlen erfüllbar) ist, wenden wir diesen Algorithmus auf den Ausdruck $H \wedge H_\infty$ an (wobei wir o. B. d. A. voraussetzen, daß das Zeichen A_0^2 in H nicht vorkommt).

Fall 1: $(H \wedge H_\infty) \notin \mathrm{ef}$
d. h. es gibt kein $I \neq \emptyset$ mit $(H \wedge H_\infty) \in \mathrm{ef}_I$, wegen $H_\infty \in \mathrm{ef}_\mathsf{N}$ gilt also $H \notin \mathrm{ef}_\mathsf{N}$.

Fall 2: $(H \wedge H_\infty) \in \mathrm{ef}$
Dann gibt es ein $I^\star \neq \emptyset$ mit $(H \wedge H_\infty) \in \mathrm{ef}_{I^\star}$, wegen $H_\infty \in \mathrm{ef}_{I^\star}$ ist $I^\star$ unendlich. Nach dem

Satz 4.3.7 (von Löwenheim–Skolem)
Wenn $\mathrm{ef}_I H$ *und* I *unendlich, so* $\mathrm{ef}_\mathsf{N} H$.

ist $H \in \mathrm{ef}_\mathsf{N}$.

Also gilt $H \in \mathrm{ef}_\mathsf{N}$ gdw. $(H \wedge H_\infty) \in \mathrm{ef}_I$.

Mit ef wäre also auch ef_N entscheidbar, was wir bereits als falsch nachgewiesen haben. Folglich ist auch ef nicht entscheidbar. □

4.4 Lösbare Fälle des Entscheidungsproblems

Die Unlösbarkeit des Entscheidungsproblems wirft natürlich die Frage auf, in welchen Fällen, d. h. zum Beispiel für welche spezielle Klassen von Ausdrücken, man dennoch die Erfüllbarkeit bzw. Gültigkeit entscheiden kann.

Trivial ist der Fall, daß der Individuenbereich I endlich ist. Dann gibt es zu gegebener Stellenzahl n nur endlich viele n-stellige Attribute und nur endlich viele n-stellige Funktionen über I. Da in einem Ausdruck H nur endlich viele (freie) Individuenvariable vorkommen, sind also auch nur endlich viele Interpretationen g und Belegungen f über I daraufhin zu prüfen, ob $\mathrm{wert}_{I,g}(H, f) = \mathrm{W}$ ist.

Im folgenden betrachten wir zwei unterschiedliche Spezialfälle, zum einen schränken wir die Menge der betrachteten Ausdrücke ein (Klassenkalkül), zum anderen übertragen wir den Gültigkeitsbegriff des Aussagenkalküls in den Prädikatenkalkül.

4.4.1 Klassenkalkül

Als Klassenkalkül bezeichnen wir die Menge der Ausdrücke $H \in \mathrm{AUSD}$

- ohne Funktionszeichen
- ohne Konstanten
- ohne Gleichheitszeichen

in denen nur einstellige Attributenzeichen vorkommen.

Beispiel
$\forall x\, ((Ax \wedge (Ax \vee Bx)) \leftrightarrow Ax)$

In der Sprache des Klassenkalküls kommen als Terme nur Individuenvariable in Frage und alle prädikativen Ausdrücke haben die Form $A_j x_k$.

Satz 4.4.1
Es sei $H \in \mathrm{AUSD}$ *ohne Funktionszeichen, ohne Konstanten, ohne* $=$ *und mit* k *verschiedenen einstelligen Attributenzeichen, o. B. d. A.* $A_1, \ldots, A_k$. *Dann ist* $\mathrm{ef} H$ *gdw. ein* I *mit* $0 < \mathrm{card}\,(I) \leq 2^k$ *existiert, mit* $\mathrm{ef}_I H$.

Beweis
Wenn $H \in \mathrm{ef}_I$ ist, dann gilt nach Definition $H \in \mathrm{ef}$. Umgekehrt sei H erfüllbar; es gibt also ein $I \neq \emptyset$, eine Interpretation g und eine Belegung f über I mit $\mathrm{wert}_{I,g}(H, f) = \mathrm{W}$.

Es sei für $i = 1, \ldots, k$

$$\alpha_i \quad := \quad g(A_i),$$

wobei $A_1, \ldots, A_k$ die in H vorkommenden einstelligen Attributenzeichen sind.

Wir konstruieren einen Individuenbereich $\overline{I}$, eine Interpretation $\overline{g}$ und eine Belegung $\overline{f}$ mit

$$\begin{aligned} \mathrm{card}\left(\overline{I}\right) &\leq 2^k \\ \mathrm{wert}_{\overline{I},\overline{g}}\left(H, \overline{f}\right) &= \mathrm{W} \end{aligned}$$

durch Faktorisierung der Algebra $[I, \alpha_1, \ldots, \alpha_k]$:

Für $a, b \in I$ definieren wir

$$a \sim b \quad :\leftrightarrow \quad \alpha_1(a) = \alpha_1(b) \wedge \cdots \wedge \alpha_k(a) = \alpha_k(b)$$

Lemma 1
$\sim$ ist Äquivalenzrelation in I.

Für $a \in I$ bezeichne $[a] := \{b | b \in I \wedge b \sim a\}$ die Äquivalenzklasse von a und

$$\overline{I} \quad := \quad \{[a] | a \in I\}$$

die Zerlegung $I/\sim$.

Lemma 2
$1 \leq \mathrm{card}\left(\overline{I}\right) \leq 2^k$.

Jetzt überlegen wir uns, daß $\sim$ sogar Kongruenz bezüglich $\alpha_1, \ldots, \alpha_k$ ist:

$$\text{wenn } a \sim b, \text{ so } \alpha_i(a) = \alpha_i(b)$$

folglich können wir auf $\overline{I}$ definieren:

$$\overline{\alpha}_i([a]) = \alpha_i(a) \qquad (i = 1, \ldots, k)$$

Wir betrachten $\overline{g}, \overline{f}$ mit

$$\overline{g}(A_i) = \overline{a}_i \qquad \overline{f}(x_j) = [f(x_j)]$$

und behaupten

Lemma 3
$\mathrm{wert}_{\overline{I},\overline{g}}\left(H', \overline{f}\right) = \mathrm{wert}_{I,g}(H', f)$ *für alle Ausdrücke H' des Klassenkalküls, die höchstens die Attributenzeichen $A_1, \ldots, A_k$ enthalten.*

Beweis
durch Induktion über H'

Anfangsschritt: H' ist prädikativ: $H' \equiv A_i x_j \ (1 \leq i \leq k)$

Dann ist

$$\begin{array}{rclcl} \text{wert}_{\overline{I},\overline{g}}\left(H', \overline{f}\right) & = & \overline{g}\left(A_i\right)\left(\overline{f}\left[x_j\right]\right) & = & \overline{\alpha}_i\left(\left[f\left(x_j\right)\right]\right) \\ & = & \alpha_i\left(f\left(x_j\right)\right) & = & \text{wert}_{I,g}\left(H', f\right). \end{array}$$

Induktionsschritt:

(a) aussagenlogische Zusammensetzung:

$$\text{wert}_{\overline{I},\overline{g}}\left(\neg H, \overline{f}\right) = \text{non}\left(\text{wert}_{\overline{I},\overline{g}}\left(H, \overline{f}\right)\right) = \text{non}\left(\text{wert}_{I,g}\left(H, f\right)\right) = \text{wert}_{I,g}\left(\neg H, f\right)$$

$$\text{wert}_{\overline{I},\overline{g}}\left(\left(\left(H_1 \begin{array}{c} \wedge \\ \vee \\ \rightarrow \\ \leftrightarrow \end{array} H_2\right), \overline{f}\right)\right) = \left(\text{wert}_{\overline{I},\overline{g}}\left(H_1, \overline{f}\right), \text{wert}_{\overline{I},\overline{g}}\left(H_1, \overline{f}\right)\right) = \text{wert}_{I,g}\left(\left(H_1 \begin{array}{c} \wedge \\ \vdots \\ \leftrightarrow \end{array} H_2, f\right)\right)$$

et
$\vdots$
aeq

(b) prädikatenlogische Zusammensetzung:

$$\begin{array}{rcl} \text{wert}_{\overline{I},\overline{g}}\left(\forall x_j H, \overline{f}\right) & = & \text{Om}\left(\alpha^1_{H,\overline{I},\overline{g},\overline{f},x_j}\right) \\ & = & \begin{cases} \text{W}, & \text{falls } \alpha_{H,\overline{I},\overline{g},\overline{f},x_j}\left([a]\right) = \text{W für alle } [a] \in \overline{I} \\ \text{F}, & \text{sonst} \end{cases} \end{array}$$

Nun ist

$$\begin{array}{rcll} \alpha_{H,\overline{I},\overline{g},\overline{f},x_j}\left([a]\right) & = & \text{wert}_{\overline{I},\overline{g}}\left(H, \left[\overline{f}\right]^j_{[a]}\right) & \\ & = & \text{wert}_{I,g}\left(H, \left[\overline{f}\right]^j_a\right) & \text{(nach Ind.–Vor.)} \\ & = & \alpha_{H,I,g,f,x_j}\left(a\right) & \end{array}$$

Also ist

$$\begin{array}{rcl} \text{wert}_{\overline{I},\overline{g}}\left(\forall x_j H, \overline{f}\right) & = & \begin{cases} \text{W}, & \text{falls } \alpha_{H,I,g,f,x_j}\left(a\right) = \text{W für alle } [a] \in I \\ \text{F}, & \text{sonst} \end{cases} \\ & = & \text{wert}_{I,g}\left(\forall x_j H, f\right). \end{array}$$

Analog behandelt man den Fall $H' \equiv \exists x_j H$.

□

Folgerung 4.4.2 (aus Satz 4.4.1)
Um zu entscheiden, ob ein Ausdruck des Klassenkalküls erfüllbar (in irgendeinem nichtleeren Bereich) ist, genügt es, die Erfüllbarkeit in dem Individuenbereich mit 2^k Elementen zu überprüfen (wobei k die Zahl der Attributenzeichen ist).

Das gilt deshalb, weil für Ausdrücke ohne Gleichheitszeichen gilt: Wenn $H \in \text{ef}_I$ und $\text{card}(I) \leq \text{card}(I')$, so ist $H \in \text{ef}_{I'}$.

4.4.2 Aussagenlogische Erfüllbarkeit und Gültigkeit

Die aussagenlogisch gültigen Ausdrücke des Prädikatenkalküls bilden eine wichtige Teilmenge von ag, die entscheidbar ist.

Definition 4.4.3 (aussagenlogisch einfache Ausdrücke)
Ein Ausdruck $H \in \mathrm{AUSD}$ *heißt* aussagenlogisch einfach *oder* unzerlegbar*, wenn er entweder prädikativer Ausdruck ist oder mit einem Quantor beginnt.*

AE *ist die Menge der aussagenlogisch einfachen Ausdrücke.*

Es ist klar, daß jeder Ausdruck $H \in \mathrm{AUSD}$ des Prädikatenkalküls in eindeutiger Weise aus aussagenlogisch einfachen Ausdrücken mit Hilfe der aussagenlogischen Zusammensetzungen aufgebaut ist.

Definition 4.4.4 (aussagenlogische Belegung)
Jede Abbildung $\beta : \mathrm{AE} \to \{\mathrm{W}, \mathrm{F}\}$ *wird* aussagenlogische Belegung *genannt.*

Definition 4.4.5 (aussagenlogischer Wert)
Die Abbildung $\mathrm{werta}(H, \beta)$ *wird für* $H \in \mathrm{AUSD}$*,* $\beta : \mathrm{AE} \to \{\mathrm{W}, \mathrm{F}\}$ *wie folgt induktiv definiert:*

Anfangsschritt:

$$\textit{Wenn } H \in \mathrm{AE}, \textit{ so } \mathrm{werta}(H, \beta) = \beta(H)$$

Induktionsschritte:

$$\textit{Wenn } H \equiv \neg H_1, \textit{ so } \mathrm{werta}(H, \beta) = \mathrm{non}(\mathrm{werta}(H_1, \beta))$$
$$\textit{Wenn } H \equiv (H_1 \wedge H_2)\,, \textit{ so } \mathrm{werta}(H, \beta) = \mathrm{et}(\mathrm{werta}(H_1, \beta), \mathrm{werta}(H_2, \beta))$$

und analog für $\vee$*,* $\to$*,* $\leftrightarrow$*.*

Definition 4.4.6 (aussagenlogische Erfüllbarkeit, Gültigkeit)
Ein Ausdruck $H \in \mathrm{AUSD}$ *heißt* aussagenlogisch erfüllbar *(bzw.* gültig*), wenn* $\mathrm{werta}(H, \beta) = \mathrm{W}$ *für ein (bzw. alle) aussagenlogischen Belegungen* β *gilt. Es sei* efa *bzw.* aga *die Menge aller aussagenlogisch erfüllbaren bzw. gültigen Ausdrücke.*

Satz 4.4.7
Es sei $I \neq \emptyset$*,* g *eine Interpretation in* I *und* f *eine Belegung über* I *und für* $H \in \mathrm{AE}$ *sei* $\beta(H) = \mathrm{wert}_{I,g}(H, f)$*. Dann gilt für* $H^\star \in \mathrm{AUSD}$

$$\mathrm{wert}_{I,g}(H^\star, f) = \mathrm{werta}(H^\star, \beta)$$

Diese Aussage beweist man leicht durch Induktion über den Aufbau von $H^\star$ aus aussagenlogisch einfachen Ausdrücken.

Folgerung 4.4.8

- $\mathrm{ef} \subset \mathrm{efa}$

  Beispiel
  $\exists x(Ax \wedge \neg Ax) \in \mathrm{efa} \setminus \mathrm{ef}$

- $\mathrm{aga} \subset \mathrm{ag}$

  Beispiel
  $\forall x(Ax \vee \neg Ax) \in \mathrm{ag} \setminus \mathrm{aga}$

Wichtig ist die

Folgerung 4.4.9
Die Mengen aga *und* efa *sind entscheidbar.*

4.5 Folgern und Ableiten

4.5.1 Folgern

Modell, Folgern und Folgerungsoperator
Sei $X \subseteq \text{AUSD}$

- $[g, f]$ ist ein *Modell* von X in I, wenn für alle $H \in X$ gilt: $\text{wert}_{I,g}(H, f) = \text{W}$.
- Aus X *folgt* H in I, $(X \text{fol}_I H)$, wenn jedes Modell von X in I den Ausdruck H erfüllt. Aus X folgt H $(X \text{fol} H)$, wenn für jedes $I \neq \emptyset$ gilt: $X \text{fol}_I H$.
- Der *Folgerungsoperator* fl_I ist wie folgt definiert: $\text{fl}_I(X) := \{H | X \text{fol}_I H\}$. Analog ist $\text{fl}(X)$ definiert.

Satz 4.5.1 (Eigenschaften der Folgerungsoperatoren)
fl_I *und* fl *haben folgende Eigenschaften:*

0. $H \in \text{fl}_I(X)$ *gdw. die Menge* $X \cup \{\neg H\}$ *kein Modell in* I *hat.*
1. $\text{fl}_I(\emptyset) = \text{ag}_I$
2. fl_I *hat die Hülleneigenschaften*
3. fl_I *erfüllt den Endlichkeitssatz*
   Beweis über den Endlichkeitssatz für Modelle, analog wie im Aussagenkalkül.
4. *Ableitbarkeitstheorem: Wenn* $(H_1 \to H_2) \in \text{fl}_I(X)$, *so* $H_2 \in \text{fl}_I(X \cup \{H_1\})$.
5. *Deduktionstheorem: Wenn* $H_2 \in \text{fl}_I(X \cup \{H_1\})$, *so* $(H_1 \to H_2) \in \text{fl}_I(X)$.
6. $\text{fl}_I(\text{ag}_I) \subseteq \text{ag}_I$, $\text{fl}(\text{ag}) \subseteq \text{ag}$

Beweis des Deduktionstheorems

Voraussetzung: Es sei $H_2 \in \text{fl}(X \cup \{H_1\})$ und $[g, f]$ ein Modell von X in I.

Behauptung: $\text{wert}_{I,g}(H_1 \to H_2, f) = \text{W}$

Fall 1: $[g, f]\text{erf}_I H_1$ $(\text{wert}_{I,g}(H_1, f) = \text{W})$

Dann ist $[g, f]$ Modell von $X \cup \{H_1\}$ in I. Es ist also $\text{wert}_{I,g}(H_2, f) = \text{W}$ nach Voraussetzung.

Daraus folgt die Behauptung.

Fall 2: $[g, f]\text{erf}_I \neg H_1$ $(\text{wert}_{I,g}(H_1, f) = \text{F})$

Dann ist $\text{wert}_{I,g}((H_1 \to H_2), f) = \text{W}$.

Also gilt die Behauptung. □

Bemerkung
Die Relation $X \text{fol}_I H$ ist unentscheidbar, auch wenn X endlich ist ($\emptyset \text{fol}_I H$ gdw. $\text{ag}_I H$).

4.5.2 Ableiten

Wie im Aussagenkalkül wollen wir für $X \subseteq$ AUSD und $H \in$ AUSD das Folgern syntaktisch charakterisieren und definieren deshalb $X\text{abl}H$.

abl
wird induktiv definiert:

A) Wenn $H \in X$, so $X\text{abl}H$

I) 1. Wenn $X\text{abl}H_1$ und $X\text{abl}(H_1 \to H_2)$, so $X\text{abl}H_2$ (modus ponens)

$$\frac{\begin{array}{c} H_1 \\ (H_1 \to H_2) \end{array}}{H_2}$$

2. Regel der vorderen Generalisierung

$$\frac{X\text{abl}(H_1\,(x) \to H_2)}{X\text{abl}(\forall x H_1 \to H_2)}$$

3. Regel der hinteren Generalisierung

$$\frac{X\text{abl}(H_1 \to H_2\,(x)),\ \text{und } x \text{ nicht in } H_1.}{X\text{abl}(H_1 \to \forall x H_2)}$$

4. Regel der vorderen Partikularisierung

$$\frac{X\text{abl}(H_1\,(x) \to H_2),\ x \text{ nicht in } H_2}{X\text{abl}(\exists x H_1 \to H_2)}$$

5. Regel der hinteren Partikularisierung

$$\frac{X\text{abl}(H_1 \to H_2\,(x))}{X\text{abl}(H_1 \to \exists x H_2)}$$

6. Regel der gebundenen Umbenennung
Aus H entsteht H' durch gebundene Umbenennung, wenn gilt:
Es gibt Individuenvariablen x_k, x_l mit
   (a) x_k kommt in H gebunden vor
   (b) x_l kommt im Wirkungsbereich $H^\star$ von x_k in H nicht quantifiziert und nicht frei vor,
   (c) $H^\star$ liegt in keinem Wirkungsbereich von x_l. (Insgesamt wird also gefordert, daß x_l nicht in $H^\star$ vorkommt.)
   (d) H' entsteht aus H, indem in $Qx_kH^\star$ überall x_k durch x_l ersetzt wird (mit Quantor $Q \in \{\forall, \exists\}$)

$$\frac{\begin{array}{c} X\text{abl}H \\ H \text{ geht durch gebundene Umbenennung in } H' \text{ über} \end{array}}{X\text{abl}H'}$$

7. Termeinsetzung

$$\frac{\begin{array}{c} X\text{abl}H\,(x) \\ T \text{ Term, wobei} \end{array} \quad \begin{array}{l} \text{alle Stellen in } H\text{, an denen } x \text{ steht, stehen in keinem Wirkungsbereich eines} \\ \text{Quantors einer Variable, die in } T \text{ vorkommt} \end{array}}{X\text{abl}H\,(x/_T)}$$

Die gebundene Umbenennung eines Ausdrucks H veranschaulichen wir an folgendem Beispiel:

$$H \equiv \exists y \forall x \overbrace{A^2 xy}^{H^\star} \qquad H' \equiv \exists y \forall z A^2 zy$$

Ein Beispiel für die Termeinsetzung ist

$$H \equiv \forall z A^2 zy \quad \text{mit} \quad T \equiv F(x, a)$$

Der Term T erfüllt alle Bedingungen für die Termeinsetzung und wir erhalten

$$H\left(y/_T\right) \equiv \forall z A^2 z F(x, a)$$

Wir definieren wieder die Operation

ab

$$\mathrm{ab}\,(X) := \{H | X \mathrm{abl} H\}$$

und die Relation und Operation

abla und aba
Die Relation $X \mathrm{abla} H$ gilt gdw. H ist nur mit der Abtrennungsregel (modus ponens) aus X ableitbar.

$$\mathrm{aba}\,(X) := \{H | X \mathrm{abla} H\}$$

Satz 4.5.2 (Eigenschaften von ab und aba)

1. $\mathrm{ab}\,(\emptyset) = \mathrm{aba}\,(\emptyset) = \emptyset$;
2. ab *und* aba *haben die Hülleneigenschaften.*
3. ab *und* aba *erfüllen den Endlichkeitssatz.*
4. *Ableitbarkeitstheorem:* $X \mathrm{abl}\,(H_1 \to H_2)$, *so* $X \cup \{H_1\} \mathrm{abl} H_2$.
5. *Deduktionstheorem gilt nicht.*
6. *Wenn* $X \subseteq \mathrm{ag}_I$, *so* $\mathrm{ab}\,(X) \subseteq \mathrm{ag}_I$ *und wenn* $X \subseteq \mathrm{ag}_I$, *so* $\mathrm{aba}\,(X) \subseteq \mathrm{ag}_I$.

Schwaches Deduktionstheorem

Wie im Aussagenkalkül wollen wir zeigen, unter welchen Voraussetzungen abl das Deduktionstheorem erfüllt.

Satz 4.5.3
Wenn für alle $H_1, H_2 \in$ AUSD *gilt:* $X \mathrm{abl} H_1 \to (H_2 \to H_1)$ *und* $X \mathrm{abl} H\,(x_j)$, *so* $X \mathrm{abl} \forall x_j H$.

Beweis
Es sei $H_0 \in$ AUSD mit x_j kommt in H_0 nicht vor und $X \mathrm{abl} H_0$.

$$\begin{array}{cl}
X \mathrm{abl} H \to (H_0 \to H) & \\
X \mathrm{abl} H & \\
\hline
X \mathrm{abl} H_0 \to H\,(x_j) & \text{(nach modus ponens)} \\
\hline
X \mathrm{abl} H_0 \to \forall x_j H & \text{(hintere Generalisierung)} \\
X \mathrm{abl} H_0 & \\
\hline
X \mathrm{abl} \forall x_j H & \text{(nach modus ponens)}
\end{array}$$

□

Definition 4.5.4 („Die“ Generalisierte eines Ausdrucks)
Für $H \in$ AUSD ist $\mathrm{Gen}\,(H)$ *einer der Ausdrücke (vielleicht der mit der kleinsten Gödelzahl), die aus H entstehen, wenn zuerst alle gebundenen Variablen nacheinander in nicht vorkommende Variablen umbenannt werden, dann alle vollfrei vorkommenden Variablen x mit $\forall x$ quantifiziert werden.*

Beispiel
Wir bestimmen die Generalisierte des Ausdrucks $H \equiv \forall x \left(A_1^2 yz \to A_2^2 xy\right) \wedge A_3^1 x$.

Zuerst nennen wir die gebundenen Variablen in nicht vorkommende um:

$$\forall u \left(A_1^2 yz \to A_2^2 uy\right) \wedge A_3^1 x$$

Alle freien Variablen sind jetzt vollfrei. Nun quantifizieren wir alle vollfreien Variablen und erhalten

$$\mathrm{Gen}(H) \equiv \forall x \forall y \forall z \forall u \left(\left(A_1^2 yz \to A_2^2 uy\right) \wedge A_3^1 x\right)$$

Dual zu $\mathrm{Gen}\,(H)$ definieren wir „die“ Partikularisierte $\mathrm{Part}\,(H)$ eines Ausdrucks H, indem wir x mit $\exists$ quantifizieren.

Satz 4.5.5 (Schwaches Deduktionstheorem)

Voraussetzung: *Es gelte*

- $X \mathrm{abl}\,(H' \to (H'' \to H'))$ (1)
- $X \mathrm{abl}\,(H' \to H')$ (2)
- $X \mathrm{abl}\,(H' \to (H'' \to H''')) \leftrightarrow ((H' \to H'') \to (H' \to H'''))$ (3)
- $X \mathrm{abl}\,(H' \to (H'' \to H''')) \to (H'' \to (H' \to H'''))$ (4)
- $X \mathrm{abl}\,(H' \to (H'' \to H''')) \leftrightarrow ((H' \wedge H'') \to H''')$ (5)

Behauptung: *Wenn* $(X \cup \{H_1\})\,\mathrm{abl} H_2$, *so* $X \mathrm{abl}\,(\mathrm{Gen}\,(H_1) \to H_2)$.

Bemerkung
Wenn H_1 Aussage ist, dann ist $\mathrm{Gen}\,(H_1) = H_1$, da dann alle vorkommenden Aussagenvariablen quantifiziert sind. Unter den Voraussetzungen (1)–(5) erfüllt abl das Deduktionstheorem.

Beweis des Satzes durch Induktion über H_2.

Syntaktische Charakterisierung des Folgerns

Die syntaktische Charakterisierung des Folgerns, die wir im Aussagenkalkül durch die Gleichung

$$\mathrm{fl}(X) = \mathrm{aba}(\mathrm{abe}(\mathrm{axa}) \cup X)$$

vorgenommen haben, erfolgt ähnlich auch im Prädikatenkalkül. Man zeigt

Satz 4.5.6

$$\mathrm{fl}(X) = \mathrm{ab}(\mathrm{ag} \cup X)$$

Der Beweis verläuft ähnlich wie im Aussagenkalkül mit Hilfe des Endlichkeitssatzes für das Folgern.

Syntaktische Charakterisierung von ag

Es bleibt, die Menge ag syntaktisch zu charakterisieren:

Die Menge AXA

$$\text{AXA} = \{H|\ H \text{ entsteht aus einem Axiom aus axa durch Ersetzen der Aussagenvariablen durch Ausdrücke aus AUSD}\}.$$

Folgerung 4.5.7
Durch Einsetzen von Aussagen des Prädikatenkalküls in die Menge axa *des Aussagenkalküls erhalten wir Axiome des Prädikatenkalküls und es gilt:*

$$\text{AXA} \subseteq \text{aga} \subseteq \text{ag}$$

Weiterhin definieren wir:

Die Menge AXI

$$\begin{aligned}
\text{AXI} \quad = \quad & \{x_0 = x_0, x_0 = x_1 \to x_1 = x_0, x_0 = x_1 \wedge x_1 = x_2 \to x_0 = x_2\} \cup \\
& \{x_0 = x_1 \to A_i^j x_2 x_3 \cdots x_k x_0 x_{k+1} \cdots x_j \leftrightarrow \\
& \qquad A_i^j x_2 \cdots x_k x_1 x_{k+1} \cdots x_j | i \geq 0, k = 1, \ldots, j\} \cup \\
& \{x_0 = x_1 \to F_i^j(x_2, \ldots, x_k, x_0, x_{k+1}, \ldots, x_j) = \\
& \qquad F_i^j(x_2, \ldots, x_k, x_1, x_{k+1}, \ldots, x_j) | i \geq 0, k = 1, \ldots, j\}
\end{aligned}$$

Dann ist

Die Menge AX

$$\text{AX} \quad := \quad \text{AXA} \cup \text{AXI}$$

Man zeigt

Satz 4.5.8 (Axiomatisierbarkeit des Prädikatenkalküls)

$$\text{ag} = \text{ab}(\text{AX})$$

Bemerkung
Die Widerspruchsfreiheit $\text{ab}(\text{AX}) \subseteq \text{ag}$ ist dabei leicht zu sehen, während die Vollständigkeit $\text{ag} \subseteq \text{ab}(\text{AX})$ schwieriger zu beweisen ist.

4.6 Prädikatenlogisches Schließen

Wir geben nun noch einige abgeleitete Schlußregeln für den Umgang mit Quantoren an. Dabei erinnern wir daran, daß Schlußregeln mit fetten „Schlußstrichen“ in beiden Richtungen gelten.

1. Quantoren–Einführung und –Beseitigung

Voraussetzung: Die Individuenvariable x kommt vollfrei in $H \in$ AUSD vor

$$\frac{(X \cup \mathrm{AX})\mathrm{abl}\, H(x)}{(X \cup \mathrm{AX})\mathrm{abl}\, \forall x\, H}$$

$$\frac{(X \cup \mathrm{AX})\mathrm{abl}\, H(x)}{(X \cup \mathrm{AX})\mathrm{abl}\, \exists x\, H}\ ,\quad \frac{(X \cup \mathrm{AX})\mathrm{abl}\, \forall x\, H}{(X \cup \mathrm{AX})\mathrm{abl}\, \exists x\, H}$$

2. Quantoren–Vertauschung

$$\frac{(X \cup \mathrm{AX})\mathrm{abl}\, \forall x \forall y\, H}{(X \cup \mathrm{AX})\mathrm{abl}\, \forall y \forall x\, H}\ ,\quad \frac{(X \cup \mathrm{AX})\mathrm{abl}\, \exists x \exists y\, H}{(X \cup \mathrm{AX})\mathrm{abl}\, \exists y \exists x\, H}$$

$$\frac{(X \cup \mathrm{AX})\mathrm{abl}\, \exists x \forall y\, H}{(X \cup \mathrm{AX})\mathrm{abl}\, \forall y \exists x\, H}$$

3. Quantoren–Verteilung

$$\frac{(X \cup \mathrm{AX})\mathrm{abl}\, \forall x\ (H_1(x) \wedge H_2(x))}{(X \cup \mathrm{AX})\mathrm{abl}\, \forall x\, H_1 \wedge \forall x\, H_2}\ ,\quad \frac{(X \cup \mathrm{AX})\mathrm{abl}\, \exists x\ (H_1(x) \wedge H_2(x))}{(X \cup \mathrm{AX})\mathrm{abl}\, \exists x\, H_1 \wedge \exists x\, H_2}$$

$$\frac{(X \cup \mathrm{AX})\mathrm{abl}\, \exists x\ (H_1(x) \vee H_2(x))}{(X \cup \mathrm{AX})\mathrm{abl}\, \exists x\, H_1 \vee \exists x\, H_2}\ ,\quad \frac{(X \cup \mathrm{AX})\mathrm{abl}\, \forall x\, H_1 \vee \forall x\, H_2}{(X \cup \mathrm{AX})\mathrm{abl}\, \forall x\ (H_1(x) \vee H_2(x))}$$

4. Quantoren–Verschiebung

Die Individuenvariable x komme in H_1 *nicht* vor:

$$\frac{(X \cup \mathrm{AX})\mathrm{abl}\, \forall x\ (H_1 \rightarrow H_2(x))}{(X \cup \mathrm{AX})\mathrm{abl}\ (H_1 \rightarrow \forall x\, H_2)}\ ,\quad \frac{(X \cup \mathrm{AX})\mathrm{abl}\, \exists x\ (H_1 \rightarrow H_2(x))}{(X \cup \mathrm{AX})\mathrm{abl}\ (H_1 \rightarrow \exists x\, H_2)}$$

$$\frac{(X \cup \mathrm{AX})\mathrm{abl}\, \forall x\ (H_2(x) \rightarrow H_1)}{(X \cup \mathrm{AX})\mathrm{abl}\, \exists x\, H_2 \rightarrow H_1}\ ,\quad \frac{(X \cup \mathrm{AX})\mathrm{abl}\, \exists x\ (H_2(x) \rightarrow H_1)}{(X \cup \mathrm{AX})\mathrm{abl}\ (\forall x\, H_2 \rightarrow H_1)}$$

$$\frac{(X \cup \mathrm{AX})\mathrm{abl}\, \forall x\ (H_1 \wedge H_2(x))}{(X \cup \mathrm{AX})\mathrm{abl}\ (H_1 \wedge \forall x\, H_2)}\ ,\quad \frac{(X \cup \mathrm{AX})\mathrm{abl}\, \exists x\ (H_1 \wedge H_2(x))}{(X \cup \mathrm{AX})\mathrm{abl}\, H_1 \wedge \exists x\, H_2}$$

$$\frac{(X \cup \mathrm{AX})\mathrm{abl}\, \forall x\ (H_1 \vee H_2(x))}{(X \cup \mathrm{AX})\mathrm{abl}\ (H_1 \vee \forall x\, H_2)}\ ,\quad \frac{(X \cup \mathrm{AX})\mathrm{abl}\, \exists x\ (H_1 \vee H_2(x))}{(X \cup \mathrm{AX})\mathrm{abl}\, H_1 \vee \exists x\, H_2}$$

$$\frac{(X \cup \mathrm{AX})\mathrm{abl}\, \forall x\, \neg H}{(X \cup \mathrm{AX})\mathrm{abl}\, \neg\, \exists x\, H}\ ,\quad \frac{(X \cup \mathrm{AX})\mathrm{abl}\, \exists x\, \neg H}{(X \cup \mathrm{AX})\mathrm{abl}\, \neg\, \forall x\, H}$$

4.7 Widerlegen

Da es kein Entscheidungsverfahren für die Menge ag gibt, ag kann nur axiomatisiert werden als

$$\mathrm{ag} = \mathrm{ab}(\mathrm{AX}),$$

können wir auch nicht durch ein algorithmisches Verfahren garantieren, daß eine Widerlegung für einen Ausdruck H gefunden wird, wenn er nicht allgemeingültig ist.

Wir gehen hier auf das Widerlegen ein, weil es eine große Rolle in der Wissensverarbeitung (in der künstlichen Intelligenz) spielt.

Im folgenden sei $X \subseteq$ AUSD eine Menge von Aussagen ohne Identität, d. h. in denen das Gleichheitszeichen nicht vorkommt. Die Menge X möge das Wissen über ein Teilgebiet darstellen, also eine Art von Axiomensystem. Eine Vermutung sei durch eine Aussage H ohne Identität gegeben.

Die Aussage H folgt aus X ($H \in \mathrm{fl}(X)$), wenn jedes Modell von X H erfüllt, bzw. wenn $X \cup \{\neg H\}$ kein Modell hat. (Dabei spielen die Belegungen der Individuenvariablen keine Rolle, weil wir nur Aussagen betrachten.)

Zum Widerlegen brauchen wir einen syntaktischen Ableitungsbegriff, notiert durch $\vdash$, mit

$$X \vdash \bot \text{ gdw. } X \text{ kein Modell hat.}$$

4.7.1 Klauselbildung

Wir gehen vor wie im Aussagenkalkül und definieren

Literal und Klausel

- Ausdrücke der Form $[\neg]A^i_j T_1 \cdots T_i$ (wobei A^i_j ein i-stelliges Attributensymbol ist und $T_1, \ldots, T_i$ Terme sind) werden *Literal* genannt.
- Als *Klausel* wird ein Literal oder eine Alternative von Literalen, bei der alle Individuenvariablen durch $\forall$ gebunden (generalisiert) sind, bezeichnet.

Die allgemeine Form einer Klausel ist also

$$\forall x_1 \cdots \forall x_k \; (\neg L_1 \vee \neg L_2 \vee \cdots \vee \neg L_n \vee L'_1 \vee \cdots \vee L'_m)$$

wobei die L_i, L'_j Literale und $x_1, \ldots, x_k$ alle vorkommenden Individuenvariablen sind. Als logisch äquivalente Schreibweise notiert man die Klausel wie folgt

$$L_1 \wedge L_2 \wedge \cdots \wedge L_n \;\rightarrow\; L'_1 \vee \cdots \vee L'_m$$

Es sei X eine Menge von Aussagen ohne Identität. Nach dem Endlichkeitssatz für Modelle hat X genau dann kein Modell, wenn eine endliche Teilmenge $X^\star = \{H_1, \ldots, H_k\} \subseteq X$ existiert, die kein Modell hat, d. h. genau dann, wenn die Einermenge $\{H^\star\}$ mit

$$H^\star = H_1 \wedge \cdots \wedge H_k$$

kein Modell hat.

Wir konstruieren zu $H^\star$ eine Menge K von Klauseln mit

$$\mathrm{ef}\,H^\star \text{ gdw. } K \text{ hat ein Modell.}$$

Satz 4.7.1
Zu jeder Aussage H ohne Identität existiert eine Menge K von Klauseln mit

$$\mathrm{ef}\,H \textit{ gdw. } K \textit{ hat ein Modell.}$$

Beweis
Wir formen die Aussage H wie folgt semantisch äquivalent um:

Schritt 1: Die Zeichen $\rightarrow$, $\leftrightarrow$ eliminieren

$$\begin{array}{lcl} H_1 \leftrightarrow H_2 & \sim & (H_1 \rightarrow H_2) \wedge (H_2 \rightarrow H_1) \\ H_1 \rightarrow H_2 & \sim & \neg H_1 \vee H_2 \end{array}$$

Betrachten wir als Beispiel

$$H \equiv \forall x \forall y\, (\exists z\, (Axz \vee Ayz) \rightarrow \exists z\, Bxyz);$$

wir erhalten nach dem ersten Schritt

$$H \sim \forall x \forall y\, (\neg \exists z\, (Axz \vee Ayz) \vee \exists z\, Bxyz)$$

Schritt 2: Das Zeichen $\neg$ wird vor die Attributenzeichen getrieben

$$\begin{array}{rcl} \neg(H_1 \vee H_2) & \sim & \neg H_1 \wedge \neg H_2 \\ \neg(H_1 \wedge H_2) & \sim & \neg H_1 \vee \neg H_2 \\ \neg\neg H_1 & \sim & H_1 \\ \neg \forall x\, H_1 & \sim & \exists x\, \neg H_1 \\ \neg \exists x\, H_1 & \sim & \forall x\, \neg H_1 \end{array}$$

Im Beispiel erhalten wir

$$H \sim \forall x \forall y\, (\forall z\, (\neg Axz \wedge \neg Ayz) \vee \exists z\, Bxyz)$$

Schritt 3: Die Quantoren nach vorne ziehen

$$\left.\begin{array}{lcl} Qx\, H_1(x) \wedge H_2 & \sim & Qx\ (H_1(x) \wedge H_2) \\ Qx\, H_1(x) \vee H_2 & \sim & Qx\ (H_1(x) \vee H_2) \end{array}\right\} \text{wobei } x \text{ nicht in } H_2 \text{ vorkommt}$$

$$\begin{array}{lcl} \forall x\, H_1(x) \wedge \forall x\, H_2(x) & \sim & \forall x\ (H_1(x) \wedge H_2(x)) \\ \exists x\, H_1(x) \vee \exists x\, H_2(x) & \sim & \exists x\ (H_1(x) \vee H_2(x)) \end{array}$$

Die Variable z komme weder in H_1 noch in H_2 vor

$$Q_1 x\, H_1(x) \overset{\wedge}{\vee} Q_2 x\, H_2(x) \quad \sim \quad Q_1 x Q_2 z\ \left(H_1(x) \overset{\wedge}{\vee} H_2\left({}^{x}\!/_{z}\right)\right)$$

Im Beispiel erhalten wir

$$\begin{array}{rcl} H & \sim & \forall x \forall y\, (\forall z\, (\neg Axz \wedge \neg Ayz) \vee \exists u\, Bxyu) \\ & \sim & \forall x \forall y \forall z \exists u\, ((\neg Axz \wedge \neg Ayz) \vee Bxyu) \end{array}$$

Schritt 4: Den quantorenfreien Ausdruck in Form einer Konjunktion von Alternativen bringen

$$\begin{array}{lcl} (H_1 \wedge H_2) \vee H_3 & \sim & (H_1 \vee H_3) \wedge (H_2 \vee H_3) \\ H_1 \vee (H_2 \wedge H_3) & \sim & (H_1 \vee H_2) \wedge (H_1 \vee H_3) \end{array}$$

Im Beispiel:

$$H \sim \forall x \forall y \forall z \exists u\, ((\neg Axz \vee Bxyu) \wedge (\neg Ayz \vee Bxyu))$$

Bemerkung
Der jetzt erhaltene Ausdruck wird als *pränexe Normalform* ($\mathrm{PNf}(H)$) bezeichnet; bisher sind alle Umformungen semantisch äquivalent erfolgt, d. h. $\mathrm{ag}(H \leftrightarrow \mathrm{PNf}(H))$.

Schritt 5: Elimination der $\exists$-Quantoren (Skolemisierung)

Dieser Schritt führt von $H' = \mathrm{PNf}(H)$ nicht zu einem semantisch äquivalenten Ausdruck H'', sondern zu einem H'' mit

$$\mathrm{ef}H' \text{ gdw. } \mathrm{ef}H'',$$

d. h. einem erfüllbarkeitsäquivalenten Ausdruck, was für unsere Zwecke ausreicht.

Sei $H = Q_1 x_1 \cdots Q_n x_n \underbrace{H_0(x_1, \ldots, x_n)}_{\text{quantorenfrei}}$, Q_i sei das von links erste $\exists$

Fall 1: $i = 1$, $H \equiv \exists x_1 \ldots$

Wir nehmen eine (neue) Individuenkonstante a_j (nullstelliges Funktionszeichen) und setzen

$$H' \quad \equiv \quad Q_2 x_2 \cdots Q_n x_n \, H_0 \left[{}^{x_1}\!/_{a_j} \right]$$

Offenbar gilt $\mathrm{ef}H$ gdw. $\mathrm{ef}H'$

Fall 2: $i > 1$, $H \equiv \forall x_1 \cdots \forall x_{i-1} \exists x_i Q_{i+1} x_{i+1} \cdots$

Wir nehmen ein (neues) $(i-1)$-stelliges Funktionssymbol F_j^{i-1} her und setzen

$$H' \quad \equiv \quad \forall x_1 \cdots \forall x_{i-1} Q_{i+1} x_{i+1} \cdots Q_n x_n \, H_0 \left[{}^{x_i}\!/_{F_j^{i-1}(x_1, \ldots, x_{i-1})} \right]$$

Diese Operation wird wiederholt bis alle $\exists$ eliminiert sind

Im Beispiel erhalten wir

$$H \approx \forall x \forall y \forall z \, ((\neg Axz \vee BxyF(x,y,z)) \wedge (\neg Ayz \vee BxyF(x,y,z)))$$

Schritt 6: Zerlegung in Klauseln

Es erfolgt eine Umbenennung der Variablen, so daß in jedem Konjunktionsglied verschiedene Variable sind.

$$\forall x \, (H_1(x) \wedge H_2(x)) \quad \sim \quad \forall x \, H_1 \wedge \forall y \, H_2 \left[{}^{x}\!/_{y} \right]$$

Im Beispiel:

$$H \approx \forall x \forall y \forall z \, (\neg Axz \vee BxyF(x,y,z)) \wedge \forall x' \forall y' \forall z' \, (\neg Ay'z' \vee Bx'y'F(x',y',z'))$$

Die Konjunktionsglieder liefern jetzt die Klauseln; in unserem Beispiel erhalten wir

$$\begin{array}{rll} K = & \{Axz & \rightarrow \quad BxyF(x,y,z), \\ & Ay'z' & \rightarrow \quad Bx'y'F(x',y',z')\} \end{array}$$

Damit haben wir die Erfüllbarkeitsäquivalenz gezeigt. □

Die Klauseln in unserem Beispiel haben eine besondere Form:

Horn–Klauseln
Klauseln, die nur ein Literal auf der rechten Seite haben, heißen HORN–Klauseln.

HORN–Klauseln spielen (wie wir später sehen werden) eine wichtige Rolle in der logischen Programmierung (PROLOG).

4.7.2 Herbrand–Modell

Wir betrachten nun spezielle (syntaktische) Modelle und zeigen, daß diese zum Nachweis, daß eine Klauselmenge kein Modell hat, ausreichen.

Herbrand–Modell
H sei eine Konjunktion von Klauseln.
Wir setzen

$$\begin{aligned} I_0 &= \{a_0\} \cup \text{ Menge aller Konstantenzeichen, die in } H \text{ vorkommen} \\ I_{j+1} &= I_j \cup \{F_k^n(t_1,\ldots,t_n) \mid F_k^n \text{ ist ein in } H \text{ vorkommendes Funktionszeichen, } t_1,\ldots,t_n \in I_j\ \} \end{aligned}$$

$I_H = \bigcup_{j\geq 0} I_j$ ist eine Menge von Wörtern (Grundtermen), die wir als Individuenbereich verwenden.
Das Tripel $[I_H, g_H, f_H]$ heißt HERBRAND–Modell, wenn

1. $g_H(a_i) = a_i$ für alle a_i, die in H vorkommen)
2. $g_H(F_k^n)(t_1,\ldots,t_n) = F_k^n(t_1,\ldots,t_n)$ (für alle F_k^n, die in H vorkommen)
3. $[g_H, f_H]$ ist Modell in I_H, d. h. $\text{wert}_{I_H,g_H}(H, f_H) = \text{W}$

In unserem Beispiel ist

$$\begin{aligned} I_0 &= \{a_0\} \qquad\qquad \text{(in } K \text{ kommen keine Konstanten vor)} \\ I_1 &= \{a_0, F(a_0,a_0,a_0)\} \\ I_2 &= \{a_0, F(a_0,a_0,a_0), F(F(a_0,a_0,a_0),a_0,a_0),\ldots\} \end{aligned}$$

Satz 4.7.2
H sei eine Konjunktion von Klauseln. Dann gilt

$$\text{ef}H \quad \textit{gdw. } H \textit{ ein } \textsc{Herbrand}\textit{–Modell hat.}$$

Beweis
Es genügt zu zeigen, daß jede erfüllbare Konjunktion von Klauseln ein HERBRAND–Modell hat.

Es sei $[I, g, f]$ gegeben mit $\text{wert}_{I,g}(H, f) = \text{W}$, d. h. für jede Klausel

$$\forall x_1 \cdots \forall x_n\, H_0(x_1,\ldots,x_n)$$

gilt bei beliebiger Belegung $f' : \{x_1,\ldots,x_n\} \to I$

$$\text{wert}_{I,g}(H_0, f') = \text{W}.$$

Entsprechend der Definition bestimmen wir I_H und g_H und vervollständigen g_H für Attributenzeichen wie folgt:

$$g_H(A_j^i)(t_1,\ldots,t_i) := \text{wert}_{I,g}(A_j^i t_1 \cdots t_i, f)$$

Dabei hängt die rechte Seite (der zugeordnete Wahrheitswert) nicht von f ab, weil $A_j^i t_1 \cdots t_i$ keine Individuenvariablen enthält.

Es sei nun $f_H : X \to I_H$ eine Belegung und $A_j^i T_1 \cdots T_i$ ein Literal aus H_0. Dann gilt

$$\begin{aligned} &\text{wert}_{I_H,g_H}(A_j^i T_1 \cdots T_i, f_H) = \\ &\quad = g_H(A_j^i)\,(\text{element}_{I_H,g_H}(T_1, f_H),\ldots,\text{element}_{I_H,g_H}(T_1, f_H)) \\ &\quad = \text{wert}_{I_H,g_H}\left(A_j^i T_1 \left[{}^{x_\gamma}/_{f_H(x_\gamma)}\right] \cdots T_i \left[{}^{x_\gamma}/_{f_H(x_\gamma)}\right], f\right) \end{aligned}$$

wobei f eine beliebige I-Belegung ist.

Wir betrachten zu f_H die I-Belegung $f^\star$ mit

$$f^\star(x_\gamma) := \text{element}_{I,g}\,(f_H(x_\gamma), f)$$

Die rechte Seite hängt von f nicht ab, weil $f_H(x_\gamma)$ ein Term ohne Individuenvariable ist. Ist z. B. $f_H(x_\gamma) = a_0$, so ist $\text{element}_{I,g}\,(f_H(x_\gamma), f) = g(a_0)$. Nun ist

$$\begin{aligned}\text{wert}_{I,g}(A^i_j T_1 \cdots T_i, f^\star) &= \text{wert}_{I,g}\left(A^i_j T_1 \left[{}^{x_\gamma}\!/_{f_H(x_\gamma)}\right] \cdots T_i \left[{}^{x_\gamma}\!/_{f_H(x_\gamma)}\right], f\right)\\ &= \text{wert}_{I_H,g_H}(A^i_j T_1 \cdots T_i, f_H)\end{aligned}$$

folglich gilt für alle I_H-Belegungen f_H

$$\text{wert}_{I_H,g_H}(H_0, f_H) = \text{wert}_{I,g}(H_0, f^\star) = \text{W},$$

denn alle Literale in H_0 haben bei $[I_H, g_H, f_H]$ denselben Wert wie bei $[I, g, f^\star]$; d. h.

$$\text{wert}_{I_H,g_H}(\forall x_1 \cdots \forall x_n\, H_0, f_H) = \text{W}$$

also hat H das HERBRAND–Modell $[I_H, g_H, f_H]$. □

Der Test, ob eine Konjunktion von Klauseln ein HERBRAND–Modell besitzt, ist i. a. sehr aufwendig. Es gibt aber Fälle, wo man sofort sieht, daß kein Modell existiert:

Beispiel

$$H \equiv Aa_0 \wedge \neg Aa_0$$

d. h. als Klauselmenge geschrieben

$$\begin{array}{ccc} & & Aa_0 \\ Aa_0 & \rightarrow & \bot \end{array}$$

Die Schnittregel liefert hier sofort $\bot$.

Im Beispiel

$$H \equiv \forall y\, Aaf(y) \wedge \forall x\, \neg Axf(b)$$

bzw. als Klauselmenge

$$\begin{array}{ccc} & & Aaf(y) \\ Axf(b) & \rightarrow & \bot \end{array}$$

ist die Schnittregel nicht unmittelbar anwendbar, erst wenn man die Substitution ${}^{x}\!/_{a}$, ${}^{y}\!/_{b}$ ausgeführt hat, kommt man mit der Schnittregel zum Widerspruch $\bot$.

4.8 Unifikation und Resolution

Im vorhergehenden Beispiel haben wir gesehen, daß die Schnittregel nicht immer unmittelbar anwendbar ist. Wir definieren deshalb den Begriff der Substitution und der Unifikation und bilden mittels der Resolution davon ausgehend einen syntaktischen Ableitungsoperator.

4.8.1 Substitution und Unifikation

Substitution
Eine *Substitution* σ ist eine Abbildung, die jeder Individuenvariablen einen Term zuordnet. Ist T ein Term, so bezeichne $T[\sigma]$ den Term, der entsteht, wenn simultan für jede Variable x_i der Term $\sigma(x_i)$ eingesetzt wird und $H[\sigma]$ bezeichnet den Ausdruck, der aus H in diesem Fall entsteht.

Unifikator
Eine Substitution σ heißt *Unifikator* für Ausdrücke H_1, H_2, wenn $H_1[\sigma] \equiv H_2[\sigma]$ ist.

Beispiel
Wir betrachten die Ausdrücke $H_1 \equiv Aaxf(g(y))$ und $H_2 \equiv Azf(z)f(v)$ und wollen beide unifizieren.

	x	y	z	v
σ_1	$f(a)$	a	a	$g(a)$

$$H_1[\sigma_1] \equiv Aaf(a)f(g(a)) \equiv H_2[\sigma_1]$$

also σ_1 ist ein Unifikator für H_1, H_2.

	x	y	z	v
σ_2	$f(a)$	y	a	$g(y)$

$$H_1[\sigma_2] \equiv Aaf(a)f(g(y)) \equiv H_2[\sigma_2]$$

Auch σ_2 ist ein Unifikator für H_1, H_2, aber σ_2 ist allgemeiner als σ_1, weil eine Substitution σ_3 so existiert, daß

$$\sigma_3(\sigma_2(.)) = \sigma_1(.)$$

ist.

allgemeinster Unifikator
σ heißt *allgemeinster Unifikator* für H_1, H_2, wenn

1. σ ein Unifikator für H_1, H_2 ist
2. Zu jedem Unifikator σ' für H_1, H_2 eine Substitution $\sigma^\star$ mit

$$\sigma'(.) = \sigma^\star(\sigma(.))$$

existiert.

4.8.2 Berechnung eines allgemeinsten Unifikators

Wir formulieren einen Algorithmus, der folgendes leistet:

Eingabe: $H_1 = A^i_j T_1 \cdots T_i$, $H_2 = A^i_j T'_1 \cdots T'_i$
zwei prädikative Ausdrücke mit dem gleichen Attributensymbol und disjunkten Variablenmengen

Ausgabe: allgemeinster Unifikator falls existent, „$\ominus$“ sonst.

PROCEDURE Unify (H_1, H_2: Literal): Substitution;
$k := 0$; $M_0 := \{H_1, H_2\}$; $\sigma_0 = \text{id}$
LOOP
  IF card(M_k) = 1 THEN RETURN σ_k END;
  $T_k := \{t_1, t_2 | t_1, t_2$ bestimmen den von links ersten Unterschied in
        Ausdrücken von $M_k\}$;
  IF Es gibt eine Variable y in T_k und einen Term t in T_k, in
    dem diese Variable nicht vorkommt
  THEN
    $\sigma_{k+1}(x) = \begin{cases} t, & \text{falls } x = y \\ \sigma_k(x), & \text{sonst} \end{cases}$
    $M_{k+1} := \sigma_{k+1}(M_k)$;
    INC(k);
  ELSE RETURN „$\ominus$“
  END
END;

Beispiel
Wir betrachten

$$k = 0,\ M_0 = \{Aaxf(g(y)), Azf(z)f(v)\};\ \sigma_0 = \begin{array}{l} v \to v \\ x \to x \\ y \to y \\ z \to z \end{array}$$

und zeigen, wie die Prozedur Unify den allgemeinsten Unifikator bestimmt:

$$\begin{array}{lcll}
T_0 & = & \{a, z\} & \\
 & \Downarrow & & \\
\sigma_1 & = & \begin{array}{l} v \to v \\ x \to x \\ y \to y \\ z \to a \end{array} & M_1 = \{Aaxf(g(y)), Aaf(a)f(v)\} \\
 & \Downarrow & & \\
T_1 & = & \{x, f(a)\} & \\
\sigma_2 & = & \begin{array}{l} v \to v \\ x \to f(a) \\ y \to y \\ z \to a \end{array} & M_2 = \{Aaf(a)f(g(y)), Aaf(a)f(v)\} \\
 & \Downarrow & & \\
T_2 & = & \{g(y), v\} & \\
\sigma_3 & = & \begin{array}{l} v \to g(y) \\ x \to f(a) \\ y \to y \\ z \to a \end{array} & M_3 = \{Aaf(a)f(g(y))\} \\
 & \Downarrow & & \\
 & \text{Stop} & &
\end{array}$$

Ein Beispiel für das Scheitern der Unifikation zeigen wir mit

$$k = 0,\ M_0 = \{Af(a)g(x), Ayy\}$$

Die Prozedur Unify führt folgende Berechnung aus:

$$\begin{array}{rcll} T_0 & = & \{f(a), y\} & \\ & \Downarrow & & \\ \sigma_1 & = & \begin{array}{l} x \to x \\ y \to f(a) \end{array} & M_1 = \{Af(a)g(x), Af(a)f(a)\} \\ & \Downarrow & & \\ T_1 & = & \{g(x), f(a)\} & \Longrightarrow \text{ enthält keine Variable} \\ & & & \Longrightarrow \ominus \end{array}$$

4.8.3 Resolution

Die soeben definierte Unifikation ermöglicht die Anwendung der Schnittregel, aber auch die Vereinfachung von Klauseln.

Beispiel

$$\begin{array}{rcl} Ax \wedge Af(y) & \to & \bot \\ & & Af(g(a)) \end{array}$$

Unifizieren $\quad Ax \quad Af(y) \quad$ durch $\sigma : x \to f(y)$ und erhalten

$$\begin{array}{rcl} Af(y) & \to & \bot \\ & & Af(g(a)) \end{array}$$

Unifizieren durch $\sigma : y \to g(a)$

$$\begin{array}{rcl} Af(g(a)) & \to & \bot \\ & & Af(g(a)) \end{array}$$

und Cut liefert $\bot$, also ist die ursprüngliche Klauselmenge nicht erfüllbar und hat kein Modell.

Resolutionsregel

Seien $K_1 : L_1 \wedge \cdots \wedge L_n \to L'_1 \vee \cdots \vee L'_m$ und $K_2 : L^\star_1 \wedge \cdots \wedge L^\star_{n^\star} \to L''_1 \vee \cdots \vee L''_{m'}$ Klauseln mit disjunkten Variablenmengen.

Wir führen folgende Begriffe ein:

Resolvente
σ sei allgemeinster Unifikator von L'_μ und $L^\star_\gamma$. Die Klausel

$$\begin{array}{c} \sigma(L_1 \wedge \cdots \wedge L_n \wedge L^\star_1 \wedge \cdots \wedge L^\star_{\gamma-1} \wedge L^\star_{\gamma+1} \wedge \cdots \wedge L^\star_{n^\star} \to \\ L'_1 \vee \cdots \vee L'_{\mu-1} \vee L'_{\mu+1} \vee \cdots \vee L'_m \vee L''_1 \vee \cdots \vee L''_{m'}) \end{array}$$

heißt *Resolvente* von K_1, K_2 bzgl. μ, γ und σ.

rechter Faktor
Ist σ ein allgemeinster Unifikator von L'_i, L'_j $(1 \leq i < j \leq m)$, so heißt

$$\sigma(L_1 \wedge \cdots \wedge L_n \rightarrow L'_1 \vee \cdots \vee L'_{i-1} \vee L'_{i+1} \vee \cdots \vee L'_m)$$

ein *rechter Faktor* von K_1.
linker Faktor
Ist σ ein allgemeinster Unifikator von L_i, L_j $(1 \leq i < j \leq n)$, so ist

$$\sigma(L_1 \wedge \cdots \wedge L_{i-1} \wedge L_{i+1} \wedge \cdots \wedge L_n \rightarrow L'_1 \vee \cdots \vee L'_m)$$

ein *linker Faktor* von K_1.

Ableiten mit der Resolutionsregel

Mit der Resolution definieren wir jetzt wieder induktiv einen syntaktischen Ableitbarkeitsbegriff (wie die Cut–Ableitbarkeit im Aussagenkalkül):

Ableiten mit der Resolutionsregel $Y \vdash H$
Es sei Y eine Menge von Klauseln, H eine Klausel.

A) Wenn $H \in Y$, so $Y \vdash H$

I) 1. Wenn $Y \vdash K_1$ und $Y \vdash K_2$ und H ist Resolvente von K_1, K_2, so $Y \vdash H$.
2. Wenn $Y \vdash K$ und H ist rechter oder linker Faktor von K, so $Y \vdash H$.

Es gilt der

Satz 4.8.1
Es sei Y eine Menge von Klauseln, die kein Modell hat. Dann ist $Y \vdash \bot$.

Beweis (Idee)
Wenn Y kein Modell hat, so besitzt Y kein Herbrand–Modell. Also gibt es Klauseln $K_1, \ldots, K_n \in Y$ und Substitutionen $\sigma_1, \ldots, \sigma_n$ derart, daß aus $\{K_1[\sigma_1], \ldots, K_n[\sigma_n]\}$ aussagenlogisch (mit der Schnittregel) der Widerspruch $\bot$ abgeleitet werden kann. Aus einer solchen Ableitung konstruiert man eine Ableitung gemäß $\vdash$. $\square$

Das Ableiten mit der Resolutionsregel veranschaulichen wir uns an folgendem

Beispiel

$$\begin{array}{rrcl}
K_1: & Ax \wedge Ab & \rightarrow & \bot \\
K_2: & Aa & \rightarrow & Ab \\
K_3: & Ab & \rightarrow & Aa \\
K_4: & & & Aa \vee Ay \\
\text{Resolution } K_1, K_2,\ x \rightarrow a & & & \\
 & Aa \wedge Ab & \rightarrow & \bot \\
 & Aa & \rightarrow & Ab \\
\hline
K_5: & Aa & \rightarrow & \bot \\
\text{Cut}(K_5, K_3): & Ab & \rightarrow & Aa \\
\hline
K_6: & Ab & \rightarrow \bot & \\
\text{Resolution } K_4, K_6,\ y \rightarrow b & & & \\
 & & & Aa \vee Ab \\
\hline
K_7: & & & Aa \\
\text{Cut}(K_7, K_5): & Aa & \rightarrow & \bot \\
\hline
 & & & \bot
\end{array}$$

4.9 Logische Programmierung

Jeder Programmierer einer Programmiersprache, wie z. B. MODULA, muß, nachdem er sich die Frage gestellt hat: „**Was** ist das Problem?“, überlegen: „**Wie** ist das Problem zu lösen?“. Algorithmen werden erstellt, indem eine Abfolge von Befehlen entwickelt wird.

Grundgedanke des logischen Programmierens ist, daß die Ausführung eines Programms auch als das automatische Herleiten eines Widerspruchs aus einer gegebenen Klauselmenge aufgefaßt werden kann. Dabei wird das schon skizzierte Widerlegen mittels Resolution unter Anwendung der Unifikation benutzt. Eine aus dem Resolutionsbeweis extrahierte Antwort wird als das Rechenergebnis des Logik–Programms aufgefaßt.

Oft findet man folgende schematische Darstellung des Grundgedankens der logischen Programmierung:

$$algorithm = logic + control$$

wobei *logic* dem rein logischen Aspekt, also dem Wissen über die Zielsetzung des Algorithmus (auch Spezifikation genannt) und *control* dem prozeduralen Aspekt, also der Strategie der Anwendung von Ableitungsregeln, entspricht.

4.9.1 Prolog

Die logik–orientierte Sprache PROLOG (PROgramming in LOGic) wurde in den 70er Jahren an verschiedenen Universitäten entwickelt und erfreut sich großer Beliebtheit im Bereich der Künstlichen Intelligenz (z. B. zur Entwicklung von Expertensystemen).

PROLOG basiert auf dem Prädikatenkalkül der ersten Stufe und arbeitet nur mit Horn–Klauseln (zur Erinnerung: Horn–Klauseln haben höchstens ein positives Literal; alle Variablen sind gebunden).

Die Beschränkung auf Horn–Klauseln bedeutet nur eine geringe Einschränkung der Ausdrucksfähigkeit, da die meisten mathematischen Theorien (wenn überhaupt) mit Horn–Klauseln axiomatisierbar sind. Außerdem würden sich sonst große Effizienzprobleme bei der Auswertung ergeben.

Ein PROLOG–System besteht aus einer Wissensbasis, d. h. einer Menge von Horn–Klauseln (sogenannte Regeln und Fakten) und einem Interpreter, der problemunabhängig Abfragen der Wissensbasis auswerten und so Schlußfolgerungen ziehen kann. Der Interpreter ist ein (auf Resolution beruhendes) Beweisprogramm, das die Unerfüllbarkeit einer Klauselmenge nachweist.

Wie werden nun Horn–Klauseln in PROLOG notiert?

Variablen werden mit einem Großbuchstaben am Bezeichneranfang gekennzeichnet. Ihr Gültigkeitsbereich bezieht sich nur auf eine Klausel. Mit Kleinbuchstaben werden dagegen **Prädikate**, **Funktionssymbole** und **Konstanten** benannt. Wie man sieht, wird die semantische Unterscheidung zwischen Funktionen und Prädikaten (im Gegensatz zum Prädikatenkalkül) syntaktisch nicht unterstützt. Das ist Aufgabe des Programmierers. Klauseln werden mit einem Punkt am Ende versehen und in Fakten und Regeln unterteilt. Objekte, über die wir Aussagen machen, werden als **Atome** bezeichnet.

Fakten sind Aussagen über Objekte der Form

funktor (*objekt*).

und entsprechen allgemeinen Tatsachen, d. h. HORN–Klauseln, die nur aus einem positiven Literal bestehen, die Vorbedingung ist also leer.

Beispiel
Einfache Aussage (als Fakten) in unserer Wissensbasis sind:

```
vater(peter,harald).
mutter(anna,harald).
eltern(adam,eva,kain).
```

Ebenso können wir Allaussagen wie `mag(X,blumen).` machen (eine mögliche Interpretation dieser Allaussage ist „Jeder mag Blumen“).

Regeln sind logische Ausdrücke, mit denen aus bekannten Fakten neue Fakten gefolgert werden können. Die Schreibweise lautet

kopf`:-`*rumpf*.

wobei der Rumpf–Teil der Regel eine Konjunktion (mit Komma unterteilt) von Vorbedingungen ist, die erfüllt sein müssen, bevor der Kopf–Teil erfüllt ist.

Beispiel
Wir geben in unsere Wissensbasis folgende Regeln ein:

```
kind(K,V):-vater(V,K).
kind(K,M):-mutter(M,K).
eltern(V,M,K):-vater(V,K),mutter(M,K).
```

Das Zeichen `:-` hat den Charakter einer Implikation (es soll ein symbolisierter Pfeil nach links sein), denn eine HORN–Klausel

$$R_1 \wedge \cdots \wedge R_n \rightarrow K$$

wird in PROLOG als

$$K : -R_1, \ldots, R_n$$

notiert.

Als PROLOG–**Prädikat** wird eine Menge von Klauseln (also eine Menge von Fakten und Regeln) mit demselben Funktor und der gleichen Stelligkeit bezeichnet.

Beispiel
Wir haben den Fakt

```
eltern(adam,eva,kain).
```

und die Regel

```
eltern(V,M,K):-vater(V,K),mutter(M,K)..
```

Beide zusammen bilden das PROLOG–Prädikat `eltern\3`.

Induktiv definieren wir nun den *Term*–Begriff (nicht zu verwechseln mit der Term–Definition im Prädikatenkalkül):

A) Alle Atome und Variablen sind Terme.

I) Ist α ein n-stelliges Prädikat und $t_1, \ldots, t_n$ sind Terme, so ist auch $\alpha(t_1, \ldots, t_n)$ ein Term.

Mit Prädikaten beschreiben wir einen Sachverhalt, ein Problem. Wichtig dabei ist, daß PROLOG die Bedeutung der Klauseln nicht versteht.

Beispiel
Die Klausel `eltern(adam,eva,kain).` wird von uns als „Adam und Eva sind die Eltern von Kain“ interpretiert. Wir müssen also sehr genau die Reihenfolge der Argumente beachten.

Ausgehend von der geschaffenen Wissensbasis können wir nun Anfragen an das PROLOG–System stellen. Dabei wird die Gültigkeit von Ausdrücken anhand der vorher im Einfügemodus eingegebenen Klauseln geprüft.

Diese Prinzip wird als *„closed world assumption“* bezeichnet, d. h. selbst wenn wir wissen, daß ein bestimmter Sachverhalt in der wirklichen Welt vorliegt, muß noch nicht garantiert sein, daß er aus unseren Klauseln auch abgeleitet werden kann.

Abfragen werden wie folgt eingegeben:

`?-`*ziel.*

Das zu beweisende Ziel wird als *„goal“* bezeichnet.

Beispiel
Eine einfache Frage ist z. B.

```
?-vater(peter,harald).
```

PROLOG antwortet erwartungsgemäß mit `YES`. Die Abfrage

```
?-vater(peter,paul).
```

dagegen erzeugt `NO`.

Fragen, die eine oder mehrere Variablen enthalten, entsprechen Existenzfragen:

```
?-vater(X,harald).
```

wird beantwortet, indem die Bindung der Variable `X` ausgegeben wird, die zum Erfolg geführt hat: `X= peter`. Konjunktiv verknüpfte Fragen werden mit einem Komma unterteilt:

```
?-vater(X,harald),mag(X,blumen).
```

Die Antwort ist `X= peter`.

Bei der Abarbeitung einer Frage, die eine Variable enthält, überprüft PROLOG, ob die Variable geeignet mit einem Wert instantiiert werden kann. Dies wird über die bereits oben erwähnte Unifikation gelöst, die hier auf Termen arbeitet.

4.9.2 Ein abstrakter Interpreter

Ein abstrakter Interpreter für Logik–Programme sieht wie folgt aus:

Eingabe: P Menge von Klauseln, Z Ziel

Ausgabe: eine Instanz von Z, die aus P abgeleitet wurde, falls diese existiert, „$\ominus$“ sonst.

PROCEDURE Logische_Berechnung (P: Programm, Z: Ziel): Antwort;
$k := 0$; $RS_0 := \{Z\}$; $Z_0 = Z$
LOOP
  IF card(RS_k) $= 0$ THEN RETURN Z_k END;
  $A := \{$Ziel in $RS_k\}$;
  $A' := \{$Klausel in P, die mit A unifizierbar ist; die
      Klausel hat die Form $A : -B_1, \ldots, B_n\}$;
  $\sigma_{k+1} :=$ Unify(A, A');
  IF $\sigma_{k+1} = \ominus$
  THEN RETURN$\ominus$
  ELSE
    $RS_{k+1} := \sigma_{k+1}(\{B_1, \ldots, B_n\} \cup RS_k \setminus \{A\})$;
    $Z_{k+1} := \sigma_{k+1}(Z_k)$;
    INC(k);
  END
END;

Die Berechnung beginnt mit der Anfrage Z; falls das Programm terminiert, wird entweder die bewiesene Instanz von Z oder das Scheitern ausgegeben. Die Berechnung schreitet mittels Ziel–Reduktion fort. In jedem Schritt gibt es eine Resolvente, die bewiesen werden soll. Wird mit einer Regel resolviert, so wird der Kopf der Regel nach der Unifikation durch den Rumpf ersetzt. Die Berechnung endet, wenn die Resolvente leer ist.

Variablen müssen bei der Berechnung vorsichtig behandelt werden, um Namenskonflikte zu vermeiden. Da Variablen nur lokal in Klauseln gelten, sind gleichnamige Variablen in verschiedenen Klauseln voneinander verschieden. Eine Lösung ist, bei jeder Reduktion die Variablen einer Klausel in noch nicht verwendete Variablen umzubenennen.

Unser Interpreter ist nicht–deterministisch, d. h. nach jedem Ableitungsschritt gibt es eventuell mehr als eine Möglichkeit, die Berechnung fortzufahren. Unbestimmt ist:

- die Wahl des zu reduzierenden Ziels aus RS_K
- die Wahl der Klausel in P, mit der reduziert wird

Durch alternative Auswahlen wird implizit ein Suchbaum definiert. Da Computer nur deterministisch vorgehen, müssen die Auswahlmöglichkeiten gezielt eingeschränkt werden.

Eine Lösung ist, den Suchbaum erst in der Breite zu durchsuchen (*breadth–first*), das heißt, alle endlichen erfolgreichen Pfade im Suchbaum (mögliche Beweise des Ziels) werden auch gefunden. Die Breitensuche ist vollständig, aber ineffizient (exponentieller Aufwand).

Eine andere Möglichkeit ist, den Suchbaum zuerst in der Tiefe zu durchsuchen (*depth–first*), wobei früher durchgeführte Beweisschritte eventuell zurückgenommen werden müssen (*Backtracking*). Im Gegensatz zur Breitensuche garantiert die Tiefensuche nicht, daß ein eventuell vorhandener Beweis auch gefunden wird, da der Suchbaum unendliche Pfade enthalten kann, die unendlichen Berechnungen der Prozedur entsprechen.

Die Tiefensuche kann auf einen unendlichen Pfad geraten, bevor ein endlicher Pfad gefunden wurde, selbst wenn er existiert.

4.9.3 Die Auswertungsstrategie von Prolog

Prolog realisiert die Tiefensuche, indem der Nichtdeterminismus (Freiheit in der Wahl der Berechnungsschritte) wie folgt aufgelöst wird:

- RS_k wird als Stack verwaltet, d. h. es wird immer das oberste Ziel zur Reduktion entfernt und die abgeleiteten Ziele werden wieder oben auf den Stack abgelegt.
- Die Klauseln in P sind linear sortiert, d. h. sie werden von oben nach unten durchsucht.

Durch diese Einschränkung werden Klauseln nicht als Mengen, sondern als Folgen aufgefaßt, da die Resolution von Literalen in der Zielklausel von links nach rechts fortschreitet.

Die Auswertungsstrategie von Prolog läuft der Idealvorstellung der logischen Programmierung entgegen, denn wir müssen uns bei der Programmformulierung Gedanken machen, wie wir unsere Klauseln im Programm und die Teilziele der Regeln anordnen. Trotz logischer Äquivalenz ergeben sich verschiedene Suchbäume; eventuell kann durch Umstellung von Klauseln und Umformulierung von Regeln ein vorher gefundener Beweis nicht mehr gefunden werden, da die Berechnung in einen unendlichen Pfad des Suchbaumes geraten ist. Natürlich kann eine geschickte Formulierung unseres Logikprogramms auch das Umgekehrte erreichen (dies ist unsere Absicht); dazu müssen wir die Auswertungsstrategie von Prolog genau kennen und dementsprechend umsichtig bei der Aufstellung unseres Programms sein.

Literaturverzeichnis

[1] Asser, G.: Einführung in die Mathematische Logik (Teil 1: Aussagenkalkül; Teil 2: Prädikatenkalkül 1.Stufe); B. G. Teubner, Leipzig, 1972 und 1982.

Grundlage der Kapitel „Aussagenlogik“ und „Prädikatenlogik“

[2] Bachmann, P.: Mathematische Grundlagen der Informatik; Akademie–Verlag, Berlin, 1992.

allgemeine Einführung (mit einer Darstellung des Klassenkalküls)

[3] Cordes/Kruse/Langendörfer/Rust: Prolog; Vieweg Verlag, Braunschweig, 1992.

Ergänzung des Abschnittes „Logische Programmierung“

[4] Hermes, H.: Aufzählbarkeit, Entscheidbarkeit, Berechenbarkeit; Teubner Verlag, Stuttgart, 1976.

Ergänzung des Kapitels „Berechnungstheorie“

[5] Klaua D.: Einführung in die allgemeine Mengenlehre (Band 1: Elementare Axiome der Mengenlehre; Band 2: Grundbegriffe der axiomatischen Mengenlehre); Akademie–Verlag, Berlin, 1971 und 1972.

Grundlage des Kapitels „Mengenlehre“ (mit Stufenkalkül, Mengenalgebra etc.)

[6] Peter, R.: Rekursive Funktionen; Budapest, 1951.

Grundlage des Kapitels „Berechnungstheorie“ (mit Beweis der Existenz einer Majorante für primitiv–rekursive Funktionen)

[7] Rusza J.: Die Begriffswelt der Mathematik; Volk und Wissen Verlag, Berlin, 1975.

[8] Schöning, U.: Logik für Informatiker; Theoretische Informatik kurz gefaßt; BI Wissenschaft, Mannheim, beide 1992.

Aussagenlogik, Prädikatenlogik, Logische Programmierung bzw. Berechnungstheorie

[9] Siefkes, D.: Formalisieren und Beweisen; Vieweg Verlag, Braunschweig, 1992.

weniger formalistischer Zugang zur Logik mit vielen Beispielen

[10] Tuschik/Wolter: Mathematische Logik — kurzgefaßt: Grundlagen, Modelltheorie, Entscheidbarkeit , Mengenlehre; BI Wissenschaft, Mannheim, 1994.

allgemeine Einführung

[11] Sterling/Shapiro: Prolog — Fortgeschrittenen Programmiertechniken; Addison–Wesley Verlag, Bonn, 1988.

Ergänzung des Abschnittes „Logische Programmierung“

Symbolverzeichnis

Die folgende Tabelle verzeichnet das griechische Alphabet, dabei ist der erste Eintrag in der Tabelle immer der Kleinbuchstabe, eventuell gefolgt von einer alternativen Schreibweise. Der letzte Eintrag für einen Buchstaben zeigt die Großschreibweise, falls ein besonderes Symbol dafür existiert.

α	Alpha	ι	Jota	ρ	Rho
β	Beta	κ	Kappa	$\sigma, \varsigma, \Sigma$	Sigma
γ, Γ	Gamma	λ, Λ	Lambda	τ	Tau
δ, Δ	Delta	μ	My	υ, Υ	Ypsilon
ε	Epsilon	ν	Ny	φ, ϕ, Φ	Phi
ζ	Zeta	ξ, Ξ	Xi	χ	Chi
η	Eta	o	Omikron	ψ, Ψ	Psi
ϑ, Θ	Theta	π, Π	Pi	ω, Ω	Omega

Die folgende Liste zeigt die Großbuchstaben der Frakturschrift:

$$\mathfrak{A}, \mathfrak{B}, \mathfrak{C}, \mathfrak{D}, \mathfrak{E}, \mathfrak{F}, \mathfrak{G}, \mathfrak{H}, \mathfrak{I}, \mathfrak{J}, \mathfrak{K}, \mathfrak{L}, \mathfrak{M}, \mathfrak{N}, \mathfrak{O}, \mathfrak{P}, \mathfrak{Q}, \mathfrak{R}, \mathfrak{S}, \mathfrak{T}, \mathfrak{U}, \mathfrak{V}, \mathfrak{W}, \mathfrak{X}, \mathfrak{Y}, \mathfrak{Z}$$

Die meisten anderen Symbole werden an der Stelle ihrer Definition bzw. bei der ersten Verwendung erklärt und eingeführt. $\aleph$ ist der erste Buchstabe des hebräischen Alphabets und wird als Aleph bezeichnet.