Einführung in die mathematische Logik

Prof. Dr. Holger Brenner
Universität Osnabrück
Fachbereich Mathematik/Informatik

Wintersemester 2011/2012

Inhaltsverzeichnis

Vorwort

Dieses Skript gibt die wöchentlich zweistündige Vorlesung Einführung in die mathematische Logik wieder, die ich im Wintersemester 2011/2012 an der Universität Osnabrück gehalten habe. Ich habe diese Veranstaltung zum ersten Mal durchgeführt, bei einem zweiten Durchlauf würden sicher noch viele Korrekturen und Änderungen dazukommen. Dies bitte ich bei einer kritischen Durchsicht wohlwollend zu berücksichtigen.

 Ich bedanke mich bei der Wikimedia-Gemeinschaft und insbesondere bei Benutzer Exxu für die wichtigen Beiträge im Projekt semantische Vorlagen, die eine weitgehend automatische Erstellung des Latexcodes ermöglichen.

Ich bedanke mich bei Herrn Daniel Brinkmann für Korrekturen und bei Frau Marianne Gausmann für die Erstellung der Pdf-Files. Bei Jonathan Steinbuch bedanke ich mich für Verlinkungen und Korrekturen.

Holger Brenner

Vorlesungen

1. Vorlesung

1.1. **Probleme.**

In vielen Lebensbereichen gibt es Probleme: Alltagsprobleme, Beziehungsprobleme, Gesundheitsprobleme, Gewichtsprobleme, Finanzierungsprobleme, Umweltprobleme, technische Probleme, politische Probleme, philosophische Probleme. Zu diese Problemen gehören jeweils Vorstellungen, wie eine Lösung aussehen könnte oder zumindest eine Ahnung, in welche Richtung man nach einer Lösung suchen könnte; eine präzise Formulierung, wann ein Problem gelöst wäre, fehlt allerdings in den meisten Fällen.

Für die Probleme gibt es in der Regel verschiedene Lösungsansätze oder Lösungsstrategien. Ihr Erfolg variiert und hängt stark von unbeeinflussbaren Begleitumständen, aber auch von der Unschärfe der Problemstellung und den eigenen Bewertungsmaßstäben ab. Eine Strategie, die für dieses und jenes Problem erfolgreich war, stellt sich bei einem neuen Problem plötzlich als unbrauchbar heraus.

Könnte es eine (Meta)-strategie geben, die bei allen Problemen hilft bzw. alle Probleme löst? Eine solche Strategie kann es für die oben formulierten Problembereiche allein schon wegen der angesprochenen Unschärfe nicht geben. Die Probleme sind nie so klar umrissen, dass Einigkeit darüber besteht, ob etwas eine Lösung ist oder nicht.

Dies sieht bei mathematischen Probleme anders aus. Diese sind klar formuliert, zumeist als eine offene Frage, die grundsätzlich nur eine positive oder eine negative Antwort haben kann, und wo die Schwierigkeit darin besteht, dies herauszufinden und zu begründen (beweisen), was denn nun der Fall ist.

In der Schule beschränkt man sich typischerweise auf mathematische Probleme, für die dann eine Lösungsstrategie vorgestellt wird. Daher fragen viele Menschen, ob es denn in der Mathematik noch was zu entdecken gibt. In Wahrheit sind die mathematischen Probleme die treibende Kraft der professionellen Beschäftigung mit Mathematik.

Wir wollen zunächst einige offene mathematische Probleme vorstellen. Im Laufe der Vorlesung werden wir dann die Frage präzisieren, ob es wenigstens für diesen Teilbereich des menschlichen Denkens eine universelle Lösungsstrategie geben kann. Die Antwort wurde um 1930 von Kurt Gödel bewiesen: Eine solche Strategie kann es nicht geben.

1.2. **Offene mathematische Probleme.**

Unter den natürlichen Zahlen versteht man die Menge

$$\mathbb{N} = \{0, 1, 2, 3, \ldots\}\,.$$

Wir setzen im Moment diese Menge als gegeben voraus und auch die darauf definierten Operationen, also die Addition und die Multiplikation. Die natürlichen Zahlen werden zum Zählen und Berechnen von endlichen Mengen verwendet, und im Allgemeinen gibt es dabei keine Probleme. Addition und Multiplikation sind durch einfache Algorithmen durchführbar (in der mathematischen Logik spricht man von „berechenbar“) und können auch durch eine Maschine ausgeführt werden. Man muss aber nicht viel weiter gehen, um Probleme über natürliche Zahlen formulieren zu können, für die derzeit keine Lösung bekannt ist.

Eine natürliche Zahl k heißt Teiler einer natürlichen Zahl n, wenn es eine weitere natürliche Zahl m gibt mit $n = km$.

Definition 1.1. Eine natürliche Zahl $n \geq 2$ heißt eine *Primzahl*, wenn die einzigen natürlichen Teiler von ihr 1 und n sind.

Die ersten Primzahlen sind

$$2, 3, 5, 7, 11, 13, 17, 19, 23, \ldots\,.$$

Jede natürliche Zahl lässt sich als ein Produkt von Primzahlen schreiben, z.B. ist $100 = 2 \cdot 2 \cdot 5 \cdot 5$. Dies ist ein Satz der elementaren Zahlentheorie. Ein anderer wichtiger Satz geht auf Euklid zurück und besagt, dass es unendlich viele Primzahlen gibt. Der Beweis dafür ist ein Widerspruchsbeweis.

Satz 1.2. *Es gibt unendlich viele Primzahlen.*

Beweis. Angenommen, die Menge aller Primzahlen sei endlich, sagen wir $\{p_1, p_2, \ldots, p_r\}$. Man betrachtet die Zahl

$$N = p_1 \cdot p_2 \cdot p_3 \cdot \ldots \cdot p_r \ + 1\,.$$

Diese Zahl ist durch keine der Primzahlen p_i teilbar, da bei Division von N durch p_i immer ein Rest 1 verbleibt. Damit sind die Primfaktoren von N nicht in der Ausgangsmenge enthalten - Widerspruch. □

Dieser Satz ist für den handwerklichen, alltagstauglichen Umgang mit den natürlichen Zahlen nicht besonders wichtig, er macht aber eine wichtige Aussage über die Natur der natürlichen Zahlen. Der Beweis ist recht einfach nachvollziehbar, aber es ist nicht unmittelbar klar, wie man einen solchen Beweis findet. Man beachte auch, dass es durchaus möglich ist, in einem endlichen Text Aussagen über die Unendlichkeit zu formulieren und zu beweisen.

Es gibt nun in der Mathematik, insbesondere in der Zahlentheorie, einfach zu formulierende und leicht zu verstehende Aussagen, von denen man bis heute

nicht weiß, ob sie wahr oder falsch sind. Dazu geben wir einige prominente Beispiele.

1.3. **Das Goldbach-Problem.**

Problem 1.3. Gibt es für jede gerade natürliche Zahl $n \geq 4$ Primzahlen p und q mit $n = p + q$?

Die Frage ist also, ob man jede gerade natürliche Zahl ab 4 als Summe von zwei Primzahlen schreiben kann. Dies kann wahr oder falsch sein, diese Eigenschaft kann gelten oder nicht. Bisher ist es aber niemandem gelungen, diese Eigenschaft zu beweisen oder zu widerlegen. Man spricht von einem *offenen Problem*. Die sogenannte *Goldbachsche Vermutung* besagt, dass dieses Problem eine positive Antwort besitzt. Es ist eine treibende Kraft in der Mathematik, eine Vermutung zu bestätigen (zu beweisen) oder zu widerlegen.

Für jede gegebene gerade Zahl $n \geq 4$ lässt sich in endlich vielen Schritten entscheiden, ob sie eine Summe von zwei Primzahlen ist. Dazu überprüft man einfach der Reihe nach die ungeraden Zahlen $k < n$, ob sie und der komplementäre Summand $n-k$ Primzahlen sind. Falls es ein solches Paar $(k, n-k)$ gibt, hat man die Goldbacheigenschaft für diese eine Zahl n bestätigt. Für alle geraden Zahlen $n \geq 4$, für die diese Eigenschaft überprüft wurde, hat man stets solche Primsummanden gefunden. Inzwischen sind alle Zahlen bis zur Größenordnung 10^{18} überprüft. Zum Beispiel ist (es gibt im Allgemeinen mehrere Darstellungen)

$$4 = 2+2,\ 6 = 3+3,\ 8 = 3+5,\ 10 = 5+5 = 3+7,\ 12 = 5+7,$$
$$14 = 3+11 = 7+7,\ 16 = 3+13 = 5+11,\ \textit{etc.}$$

Doch solche rechnerischen Ergebnisse sagen letztlich nichts über die Gültigkeit der Goldbachschen Vermutung aus, bei der es ja nicht darum geht, für möglichst viele und große Zahlen zu zeigen, dass die Goldbacheigenschaft gilt, sondern für alle.

Grundsätzlich gibt es mehrere Möglichkeiten, wie diese Frage beantwortet werden könnte. Der einfachste Fall wäre, wenn man eine konkrete gerade Zahl $n \geq 4$ angibt und von dieser zeigt, dass sie nicht die Summe von zwei Primzahlen ist. Der Beweis dafür wäre dann eventuell sehr lang, man müsste alle möglichen Summanden überprüfen, aber ansonsten anspruchslos. Dann wäre die Goldbachvermutung falsch. Es ist auch denkbar, dass man zeigt, dass es eine Zahl geben muss, die die Goldbacheigenschaft nicht erfüllt, ohne eine solche konkret anzugeben. Dann wäre die Goldbachvermutung ebenfalls falsch, ein solcher Beweis könnte beliebig kompliziert sein. Oder man zeigt, das es für jede gerade Zahl $n \geq 4$ eine Summendarstellung mit zwei Primzahlen geben muss, wobei ein solcher Beweis wieder beliebig kompliziert sein könnte und die Goldbachsche Vermutung beweisen würde. Oder man zeigt sogar, wie man zu einem jeden n ein Primzahlsummandenpaar explizit berechnen kann.

1.4. **Mersenne-Primzahlen.**

Marin Mersenne (1588-1648)

Definition 1.4. Eine Primzahl der Form $2^n - 1$ heißt *Mersennesche Primzahl*.

Generell nennt man die Zahl $M_n = 2^n - 1$ die n-te *Mersenne-Zahl*. Mit dieser Bezeichnung sind die Mersenne-Primzahlen genau diejenigen Mersenne-Zahlen, die Primzahlen sind. Die Mersennezahl $M_n = 2^n - 1$ hat im Dualsystem eine Entwicklung, die aus genau n Einsen besteht. Die ersten Mersenne-Primzahlen sind

$$2^2 - 1 = 3, 2^3 - 1 = 7, 2^5 - 1 = 31, 2^7 - 1 = 127\,.$$

Die Zahl $2^{11}-1 = 2047 = 23\cdot 89$ ist die erste Mersenne-Zahl, wo der Exponent zwar prim (der Exponent einer Mersenne-Primzahl muss selbst eine Primzahl sein, siehe Lemma 13.2 (Zahlentheorie (Osnabrück 2008))) ist, die aber selbst keine Mersenne-Primzahl ist. Dies wurde 1536 von Hudalrichus Regius (Walter Hermann Ryff) gezeigt. Der nächste Kandidat, nämlich $2^{13} - 1 = 8191$, ist wieder prim. Bis ca. 1950 war bekannt, dass für die Exponenten

$$2, 3, 5, 7, 13, 17, 19, 31, 61, 89, 107 \text{ und } 127$$

Mersenne-Primzahlen vorliegen, und keine weiteren unterhalb dem Exponenten 258. Von verschiedenen Leuten, unter anderem von Cataldi und Mersenne selbst, wurden falsche Behauptungen aufgestellt. Ab ca. 1950 kamen Computer zum Bestimmen von Mersenne-Primzahlen zum Einsatz, und es wurden bisher insgesamt 47 Mersenne-Primzahlen gefunden. Alle größten bekannten Primzahlen sind Mersenne-Zahlen. Das liegt daran, dass es für diese Zahlen einen vergleichsweise einfachen Primzahltest gibt, nämlich den *Lucas-Lehmer-Test*. Mit diesem Test wird etwa alle zwei Jahre eine neue größte Primzahl gefunden. Eine Rekordliste findet sich unter Mersenne-Primzahlen auf Wikipedia.

Das Auffinden von großen (Mersenne)-Primzahlen, also der konkrete Nachweis, dass eine bestimmte Zahl diese Eigenschaft besitzt, ist aber etwas anderes als der Existenznachweis, dass es innerhalb oder oberhalb gewisser Schranken solche Zahlen gibt oder dass es überhaupt nur endlich oder unendlich viele solcher Zahlen gibt. Aufgrund des Satzes von Euklid weiß man, dass es jenseits jeder beliebig großen natürlichen Zahl noch Primzahlen gibt. Für Mersenne-Primzahlen ist das unbekannt.

Problem 1.5. Gibt es unendlich viele Mersenne-Primzahlen?

Wie gesagt, dies ist unbekannt, es wird aber vermutet, dass es unendlich viele gibt.

1.5. **Primzahlzwillinge.**

Definition 1.6. Ein *Primzahlzwilling* ist ein Paar bestehend aus p und $p+2$, wobei diese beiden Zahlen Primzahlen sind.

Die ersten Beispiele für Primzahlzwillinge sind

$$(3,5),\ (5,7),\ (11,13),\ (17,19),\ (29,31),\dots\ .$$

Übrigens ist $3,5,7$ der einzige Primzahldrilling, siehe Aufgabe 1.4 .

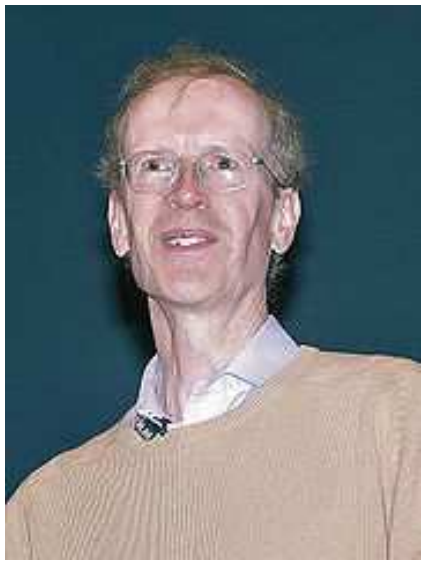
Problem 1.7. Gibt es unendlich viele Primzahlzwillinge?

1.6. **Der große Fermat und der Satz von Wiles.**

Aus dem siebzehnten Jahrhundert stammt das Problem, ob die Fermat-Gleichungen

$$x^n + y^n = z^n$$

für alle $n \geq 3$ nur triviale ganzzahlige Lösungen, bei denen $x = 0$ oder $y = 0$ ist, besitzen. Die entsprechende *Fermatsche Vermutung*, der sogenannte „Große Fermat“, galt lange Zeit als das berühmteste offene Problem der Mathematik. Nach rund 350 Jahren wurde der Große Fermat schließlich 1995 von Andrew Wiles bewiesen.



Andrew Wiles (*1953)

Satz 1.8. *Die diophantische Gleichung*

$$x^n + y^n = z^n$$

besitzt für jedes $n \geq 3$ keine ganzzahlige nichttriviale Lösung.

Mathematik-geschichtlich gesprochen kann man sagen, dass mathematische Probleme sehr hartnäckig sind, aber früher oder später doch gelöst werden. Beispielsweise wurden alle Probleme der antiken Mathematik im Laufe des 19. Jahrhunderts gelöst. Eine wichtige geschichtliche Beobachtung ist auch, dass die Lösungen zwar auch mit individuellen Höchstleistungen zusammenhängen, aber doch stark von der allgemeinen Entwicklung des mathematischen Apparats abhängen. Viele elementar formulierbare Probleme wurden nicht elementar bewiesen, sondern erst durch neue komplexe Theorien und Methoden, die neue Sichtweisen auf das Problem ermöglichten.

Eine solche geschichtliche Beobachtung trägt aber nichts zu der Frage bei, ob es eine allgemeine Lösungsstratgie für mathematische Probleme geben könnte.

1.7. **Helfen Maschinen?**

Betrachten wir das Goldbach-Problem und nehmen wir für einen Moment an, dass die Goldbachsche Vermutung nicht stimmt, dass es also eine gerade natürliche Zahl ≥ 4 gibt, die nicht die Summe von zwei Primzahlen ist. Ein Computer, eine Rechenmaschine, nennen wir sie M_1, die der Reihe nach alle geraden Zahlen auf die Goldbacheigenschaft überprüft, wird früher oder später auch diese Zahl erreichen und feststellen, dass sie nicht diese Eigenschaft besitzt und wird damit die Vermutung widerlegen.

Nehmen wir nun an, dass die Goldbachsche Vermutung stimmt, und dass es dafür einen Beweis gibt, der in normaler Sprache formuliert werden kann. Eine zweite Rechenmaschine M_2 drucke nach und nach alle möglichen Texte (in aufsteigender Länge) aus. Nehmen wir an, dass M_2 erkennen kann, ob ein Text aus sinnvollen Wörtern und Sätzen besteht, ob es sich um einen korrekten mathematischen Beweis handelt und ob dieser die Goldbachsche Vermutung beweist. Dann wird M_2 früher oder später einen Beweis für die Goldbachsche Vermutung ausgeben und dieses auch erkennen.

Da wir nicht wissen, ob die Goldbachsche Vermutung wahr ist oder nicht, kombinieren wir die beiden Maschinen zu einer einzigen Maschine M, die abwechselnd die M_1-Funktion und die M_2-Funktion ausführt. D.h., dass M abwechselnd eine Zahl überprüft, ob sie der Goldbachschen Vermutung widerspricht, und sodann einen Text überprüft, ob er einen Beweis für die Goldbachsche Vermutung beinhaltet. Da die Goldbachsche Vermutung wahr oder falsch ist, wird früher oder später M ein Gegenbeispiel oder einen Beweis finden und somit das Problem entscheiden.[1]

1.8. **Universelle Lösungsverfahren.**

Wir haben anhand einiger Beispiele gesehen, dass man mit sehr elementaren Mitteln offene Probleme formulieren kann, für die Mathematiker trotz jahrhundertelanger Bemühungen keine Antwort finden konnten. Zugleich gab es ähnliche Fragen, die lange Zeit offen waren, und dann irgendwann „plötzlich“ gelöst werden konnten.

Alles in allem ist die Lösung von mathematischen Problemen ein extrem zäher Prozess. Warum hatte Wiles den Schlüssel zum Fermatproblem, aber nicht auch zu den drei anderen oben genannten Problemen? Wird es irgendwann einmal einen Menschen geben, der alle bis dahin offenen Probleme lösen kann? Gibt es außerirdische Intelligenz, die alle mathematischen Probleme lösen kann?

In dieser Vorlesung soll es um eine Variante dieser Fragestellung gehen, nämlich um die Frage, ob es eine universelle Strategie geben kann, mit der man sämtliche mathematische Probleme angehen könnte, oder zumindest solche, die sich für die natürlichen Zahlen in einfacher Weise formulieren lassen. Von einer solchen Strategie würde man die folgenden Eigenschaften erwarten.

(1) Die Strategie ist fixiert (durch einen endlichen Text, ein Programm, eine Maschine).

[1]Die beiden anderen oben erwähnten Probleme über Mersenne-Primzahlen und Primzahlzwillinge sind von einer anderen „Bauart“ und für M_1 gibt es keine direkte Entsprechung. Man kann allerdings die Idee von M_2 radikalisieren und M_1 analog zu M_2 aufbauen, indem man M_1 ebenfalls Beweise ausgeben lässt, jetzt aber überprüft, ob es sich um einen korrekten Beweis für die Negation der Behauptung handelt. Natürlich kann man dann M_1 und M_2 unmittelbar zu einer Maschine M kombinieren, die Beweise ausgibt und überprüft, ob sie die Aussage oder ihre Negation beweist.

(2) Die Strategie ist deterministisch und ist nicht auf neue Einfälle, Intuition, Genialität angewiesen.
(3) Sie führt, angesetzt auf jedes Problem, zu einer (richtigen) Lösung.
(4) Dabei braucht die Durchführung der Strategie nur endlich viele Schritte (die Anzahl der benötigten Schritte darf vom Problem abhängen).

Ramon Llull (1232-1316)

Gottfried Wilhelm Leibniz (1646-1716)

Die Präzisierung dieser Idee führt zu der Frage, ob es einen Algorithmus geben kann, der alle (zahlentheoretischen) Probleme löst. In vielen mathematischen Teilbereichen gibt es solche Algorithmen, z.B. das eingangs erwähnte Addieren oder Multiplizieren von natürlichen Zahlen, die Bestimmung, ob eine vorgegebene natürliche Zahl eine Primzahl ist, das Lösen von linearen

Gleichungssystemen, u.s.w. Ein solcher universeller Algorithmus bzw. eine Maschine, worauf dieser universelle Algorithmus läuft, wäre sicher eine Sensation und würde die mathematische Welt enorm verändern. Selbst dann, wenn er so aufwändig wäre, dass er nie in der Zeitspanne eines Menschen (oder des Universums) zu einem einzigen Resultat gelangen würde, so wäre doch allein schon der Nachweis der prinzipiellen Existenz eine gewaltige theoretische Erkenntnis. Gedanken zu einer solchen Maschine finden sich bei Llull und bei Leibniz.

In dieser Vorlesung werden wir mathematisch beweisen, dass es einen solchen universellen Algorithmus nicht geben kann, und zwar noch nicht einmal für den Bereich der natürlichen Zahlen. Dies ist einer der Hauptsätze von Kurt Gödel. Wichtig ist an dieser Stelle zu betonen, dass es sich dabei um mathematische Sätze handelt, nicht um philosophische Sätze, auch wenn sie erkenntnistheoretisch interpretiert werden können. Es gibt auch keinen philosophischen Ersatz für diese Sätze, etwa im Sinne, „weil die Welt komplex und die Sprache unscharf ist gibt es immer Probleme“. Das wäre etwa so, wie wenn man die Relativitätstheorie mit den Worten „alles ist relativ“ gleichsetzt. Das eine ist eine mathematisch-physikalische Theorie, das andere ein nichtssagender Allgemeinplatz.

Obwohl die Unvollständigkeitssätze deutliche Schranken für die maschinelle Entscheidbarkeit und Beweisbarkeit von mathematischen Sätzen setzen, gibt es auch starke Resultate, die besagen, dass viele mathematische Tätigkeiten maschinell durchführbar sind. Der ebenfalls auf Gödel zurückgehende Vollständigkeitssatz sagt, dass Beweise für Sätze der „ersten Stufe“ als eine formale Ableitung aus Axiomen realisiert werden können, und dass damit die Korrektheit von Beweisen grundsätzlich mechanisch überprüft werden kann und dass alle korrekten Beweise mechanisch aufzählbar sind. Das oben am Beispiel der Goldbachschen Vermutung angedachte Aufzählungsprinzip für mathematische Beweise ist also prinzipiell realistisch (ein Problem dabei ist, dass die natürlichen Zahlen nicht einstufig axiomatisierbar sind, siehe Satz 12.11).

Die Behandlung der Ergebnisse von Gödel setzt mehrere mathematische Präzisierungen voraus: Eine axiomatische Präzisierung der natürlichen Zahlen, eine Präzisierung der mathematischen Sprache, in der mathematische Aussagen formuliert werden können, eine Präzisierung von Beweis, eine Präzisierung von Algorithmus (mit Hilfe von rekursiven Funktionen, Turing-Maschine, Registermaschine, etc.). Dies alles ist der Inhalt der folgenden Vorlesungen.

1.9. **Arbeitsblatt.**

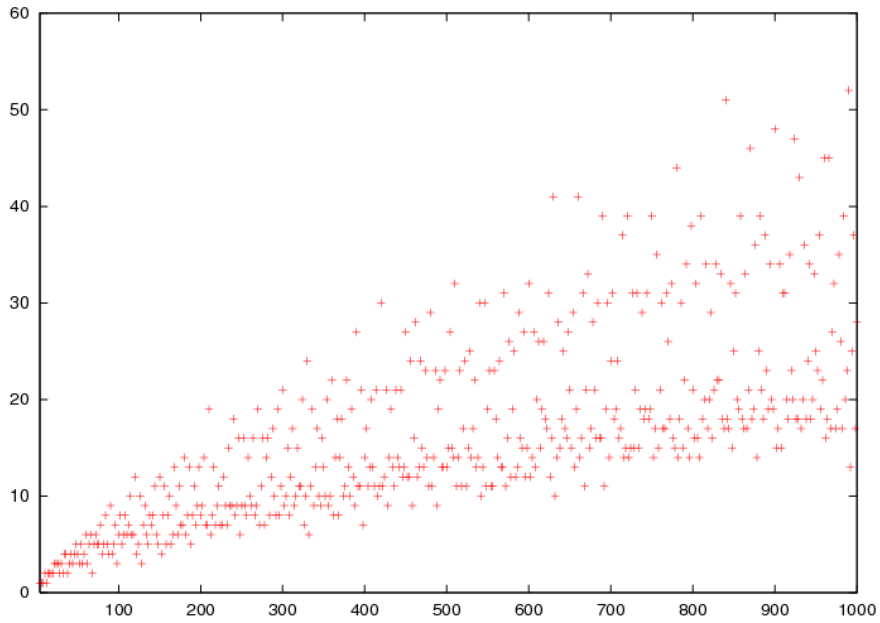
Aufgabe 1.1. Finde die kleinste Zahl N der Form $N = p_1 \cdot p_2 \cdot \ldots \cdot p_r + 1$, die keine Primzahl ist, wobei $p_1, p_2, \ldots, p_r$ die ersten r Primzahlen sind.

Aufgabe 1.2. Sei $r \in \mathbb{N}$.

a) Finde r aufeinander folgende natürliche Zahlen (also $n, n+1, \ldots, n+r-1$), die alle nicht prim sind.

b) Finde unendlich viele solcher primfreien r-„Intervalle“.

Aufgabe 1.3. Das Schaubild unten bezieht sich auf die Goldbachsche Vermutung. Was wird dadurch dargestellt?



Aufgabe 1.4. Zeige, dass es außer $3, 5, 7$ kein weiteres Zahlentripel der Form $p, p+2, p+4$ gibt, in dem alle drei Zahlen Primzahlen sind.

In der folgenden Aufgabe wird ein weiteres offenes Problem formuliert. Man mache sich die Wirkungsweise des beschriebenen Algorithmus für die Zahlen bis 20 klar.

Aufgabe 1.5. Für positive ganze Zahlen n betrachten wir folgenden Algorithmus.

Wenn n gerade ist, so ersetze n durch die Hälfte.

Wenn n ungerade ist, so multipliziere n mit 3 und addiere dann 1 dazu.

Frage (Collatz-Problem): Ist es wahr, dass man bei jeder Startzahl n früher oder später bei 1 landet?

Aufgabe 1.6. Wir betrachten eine Maschine, die nach und nach sämtliche Texte ausdruckt und damit auch früher oder später jeden Beweis ausgibt. Welche Eigenschaft eines in der Vorlesung 1 beschriebenen universellen Lösungsverfahrens besitzt diese Maschine nicht?

Aufgabe 1.7. Führe folgendes Gedankenexperiment durch: Es sei eine Maschine gegeben, die eine Aussage (eine Vermutung) über die natürlichen Zahlen nach und nach überprüft. Wenn sie alle Zahlen überprüft hätte, stünde die Antwort fest, doch da die Maschine Schritt für Schritt arbeitet, hat sie zu jedem Zeitpunkt immer nur eine endliche Teilmenge der natürlichen Zahlen überprüft und kann so, wenn die Aussage wahr ist, keinen Beweis für die Aussage liefern.

Im Allgemeinen braucht die Rechenmaschine für große Zahlen länger. Die Maschine wird jetzt beschleunigt, so dass sie für große Zahlen immer weniger Zeit braucht.

Die Maschine wird so beschleunigt, dass sie für die Überprüfung der ersten Zahl (also 1) $\frac{1}{2}$ Sekunden braucht, für die Überprüfung der zweiten Zahl $\frac{1}{4}$ Sekunden, für die Überprüfung der dritten Zahl $\frac{1}{8}$ Sekunden. Für die Überprüfung der n-ten Zahl benötigt die Maschine also genau $\left(\frac{1}{2}\right)^n$ Sekunden. Damit ist die Gesamtlaufzeit der Maschine

$$\frac{1}{2} + \left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^3 + \dots .$$

Diese Summe ist wohldefiniert, und zwar gleich 1 (im Zweiersystem ist es die Zahl $0,111111111\ldots$, deren Wert 1 ist). Nach einer Sekunde hat also die Maschine die unendlich vielen Zahlen durchgearbeitet und überprüft, und damit die Aussage bewiesen oder widerlegt.

2. Vorlesung

2.1. **Sprache als Symbolketten.**

Wir knüpfen an die Überlegungen der ersten Vorlesung an, ob es eine Maschine (einen Computer, einen Algorithmus) gibt, die mathematische Aussagen ausdrücken, ausdrucken, überprüfen, beweisen oder widerlegen kann. Eine solche Maschine operiert mit Zeichenreihen, die wir in diesem Zusammenhang eine (formale) Sprache nennen.

Definition 2.1. Es sei A eine Menge von Symbolen. Dann nennt man jede (endliche) Zeichenreihe, die man mit den Elementen aus A aufstellen kann, ein *Wort über dem Alphabet A*.

Die Menge aller Wörter über dem Alphabet A bezeichnen wir mit A^*.

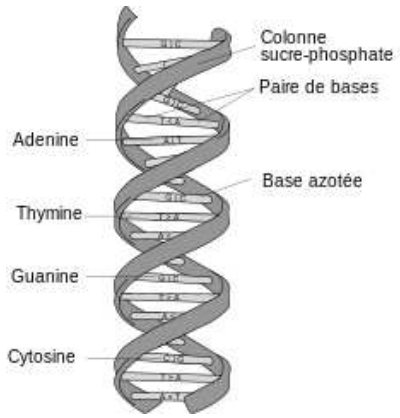
Das zugrunde liegende Alphabet kann endlich oder unendlich sein, für praktische Anwendungen reicht ein endliches Alphabet. Die Elemente des Alphabets nennt man Buchstaben, Zeichen oder Symbole. Mit einer Zeichenreihe meint man eine hintereinander geschriebene Buchstabenkette (oder Symbolkette). Dazu gehören die einelementigen Ketten, also die Elemente aus A selbst, aber auch die leere Kette (das leere Wort), die wir mit $\emptyset$ bezeichnen. Bei dieser Definition kommt es nicht auf irgendeine Sinnhaftigkeit der Wörter an, es handelt sich um eine rein formale Definition.

Beispiel 2.2. Es sei ein einelementiges Alphabet $A = \{|\}$ gegeben. Dann besitzt jedes Wort die Gestalt

$$|||\ldots|$$

mit einer gewissen Anzahl von Strichen. Zwei solche Wörter sind genau dann gleich, wenn ihre Strichanzahl übereinstimmt. In diesem Fall entsprechen also die Wörter den natürlichen Zahlen (das leere Wort entspricht der 0).

Beispiel 2.3. Die DNA-Stränge, die die Erbinformationen aller Lebewesen tragen, sind Doppelketten in Helixform aus Nukleotiden. Die entscheidenden Bestandteile der Nukleotiden sind die Basen, wofür es nur vier Möglichkeiten gibt, nämlich Adenin (A), Thymin (T), Guanin (G) und Cytosin (C). Die Nukleotiden treten in der Helix stets mit einem festen Partner (nämlich Adenin mit Thymin und Guanin mit Cytosin) auf, so dass die Struktur durch die eine Hälfte der Helix festgelegt ist. Daher entspricht die genetische Information eines DNA-Stranges einem Wort über dem Alphabet mit den Buchstaben A,T,G,C.



Wenn zu einem Alphabet A ein neues Zeichen $-$ als „Leerzeichen“ hinzugenommen wird, so werden manchmal die Wörter aus A als (eigentliche) Wörter und die Wörter aus dem Alphabet $A \cup \{-\}$ als Texte (oder Sätze) bezeichnet. Mit der Hinzunahme eines weiteren Satzbeendungssymbols kann man auch zwischen Sätzen und Texten unterscheiden.

Die geschriebene natürliche Sprache umfasst das Alphabet, das aus den Großbuchstaben A,B,C,...,Z,Ä,Ö,Ü, den Kleinbuchstaben a,b,c,...,z,ä,ö,ü,ß, den Ziffern, den Satzzeichen und einem Leerzeichen für den Abstand zwischen den Wörtern besteht. Jede lineare Hintereinanderreihung dieser Zeichen gilt für uns als Text. Im Moment interessieren wir uns nicht dafür, ob die geschriebenen Texte syntaktisch richtig gebildet oder semantisch erlaubt sind. Im Moment ist also z.B.

$$!!fL33kAs.,r$$

ein erlaubter Text.

In der Definition von einem Wort über einem Alphabet haben wir von einer Menge gesprochen und somit eine naive Mengenlehre vorausgesetzt. Im endlichen Fall wird die Symbolmenge einfach durch Auflisten ihrer Elemente gegeben. Für die gebildeten Wörter haben wir implizit verwendet, dass das Bilden von linearen Zeichenreihen unproblematisch ist.

2.2. **Rekursive Definitionen.**

Ein wichtiges Prinzip, Mengen zu definieren, ist das der *rekursiven Definition*. Eine rekursive Definition besteht aus zwei Sorten von Regeln. (1) Einerseits gewisse Startregeln, die sagen, was direkt zu der Menge gehört, und (2) Rekursionsregeln, die die Form einer Bedingung haben, und besagen, dass wenn gewisse Objekte zu der Menge gehören, und wenn neue Objekte aus diesen Objekten in bestimmter Weise gebildet sind, dass dann diese neuen Objekte ebenfalls dazu gehören (die dritte stillschweigende Bedingung an eine rekursive Definition ist, dass es keine weitere Möglichkeit gibt, zu der Menge zu gehören, außer den in (1) und (2) genannten).

Die Menge der Wörter über einem Alphabet A kann man auch folgendermaßen rekursiv definieren.

(1) $\emptyset$ ist ein Wort über A.
(2) Wenn x ein Wort ist und $a \in A$ ein Buchstabe, so ist auch xa ein Wort.

Hier repräsentiert x (eine Variable) ein beliebiges schon konstruiertes Wort. Dabei ist $\emptyset a$ als a zu lesen, so dass die beiden erlaubten Konstruktionsschritte (also der Anfangsschritt und der Rekursionsschritt) sichern, dass die einzelnen Symbole aus A Wörter sind. Wenn das Alphabet durch $A = \{a, b, c\}$ gegeben ist, so würde der rekursive Nachweis, dass *abbac* ein Wort ist, folgendermaßen gehen.

(1) Wegen der Anfangsbedingung ist $\emptyset$ ein Wort.
(2) Deshalb und wegen des Rekursionsschrittes ist $\emptyset a = a$ ein Wort.
(3) Deshalb und wegen des Rekursionsschrittes ist ab ein Wort (hier ist also $x = a$ das schon nachgewiesene Wort und der Buchstabe b wird angehängt).

(4) Deshalb und wegen des Rekursionsschrittes ist *abb* ein Wort (hier ist also $x = ab$ das schon nachgewiesene Wort und der Buchstabe b wird angehängt).
(5) Deshalb und wegen des Rekursionsschrittes ist *abba* ein Wort (hier ist also $x = abb$ das schon nachgewiesene Wort und der Buchstabe a wird angehängt).
(6) Deshalb und wegen des Rekursionsschrittes ist *abbac* ein Wort (hier ist also $x = abba$ das schon nachgewiesene Wort und der Buchstabe c wird angehängt).

Natürlich kann man *abbac* sofort ansehen, dass es sich um eine linear angeordnete Zeichenreihe über $\{a, b, c\}$ handelt, und der rekursive Nachweis scheint übertrieben pedantisch zu sein. Bei komplexer gebildeten Mengen ist aber die rekursive Definition unerlässlich, vor allem auch deshalb, da sie ermöglicht, Eigenschaften der Elemente einer Menge über den rekursiven Aufbau nachzuweisen.

Eine Sprache besteht aus sinnvollen Wörtern und sinnvollen Sätzen, nicht aus der beliebigen Aneinanderreihung von Symbolen oder Buchstaben (oder Lauten). Es ist aber vorteilhaft, erstmal alle Möglichkeiten zuzulassen und daraus durch eine Vorgabe von Regeln die sinnvollen Ausdrücke, Wörter, Lautkombinationen herauszufiltern. So funktioniert auch der kleinkindliche Spracherwerb und der Aufbau der formalen Sprachen. Wir werden nun den rekursiven Aufbau von syntaktisch korrekten Termen besprechen.

2.3. **Terme.**

Betrachten wir die sinnvollen Ausdrücke, die für eine natürliche Zahl stehen können. Mit dem Ziffernalphabet $Z = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ kann man mit den obigen Vorschriften alle natürlichen Zahlen (im Zehnersystem) aufschreiben, z.B. 530386275. Allerdings gibt es hier ein paar Schwierigkeiten, es sind nämlich auch die Zahlen 0530386275, 00530386275, u.s.w. erlaubt (und untereinander verschieden, da sie eben unterschiedliche Symbolfolgen sind). Der „Zahlenwert“ steht im Moment noch nicht zur Verfügung. Ferner möchte man das leere Zahlwort nicht als erlaubte Ziffernfolge ansehen.

Mit dieser Menge an erlaubten Zahlwörtern kann man Telefonnummern oder Internetadressen bezeichnen, aber noch nicht das machen, was man eigentlich mit Zahlen machen möchte, nämlich Zählen, Rechnen, Probleme formulieren und lösen. Für die innerhalb der natürlichen Zahlen ausführbaren Rechenoperationen, insbesondere das Nachfolgernehmen (also das Zählen) und die Addition und die Multiplikation, brauchen wir neue Symbole. Eine Aussage wie

$$5 \cdot 3 = 8 + 7$$

ist natürlich wahr, da links und rechts 15 „steht“, wie man durch „ausrechnen“ (also das korrekte Anwenden der Rechenregeln) überprüfen kann. Wenn man allerdings solche Gleichungen logisch verstehen und analysieren möchte,

so sollte man die beiden Seiten nicht als 15 lesen, sondern jeweils als ein neues „komplexes Zahlwort“, das sich aus den Ziffernsymbolen 5 und 3 und dem Malzeichen $\cdot$ bzw. den Ziffernsymbolen 8 und 7 und dem Pluszeichen $+$ zusammensetzt. Noch deutlicher ist dies in einer Gleichung der Form

$$4 \cdot x = 3 \cdot (8 + y)\,,$$

wo vermutlich nach den erlaubten Werten für x und y gesucht wird, die diese Gleichung erfüllen. Die linke und die rechte Seite sind hier sogenannte Terme, also sinnvolle mathematische Ausdrücke, die einen Zahlwert annehmen können (der Vergleich der beiden Terme durch $=$ macht aus den beiden Termen eine Aussage, das spielt jetzt aber noch keine Rolle). Solche Termmengen werden, ausgehend von einer Variablenmenge, einer Konstantenmenge und verschiedenen Funktionssymbolmengen, rekursiv definiert.

Definition 2.4. Eine *Grundtermmenge* besteht aus den folgenden (untereinander disjunkten[2]) Mengen.

(1) eine Variablenmenge V,
(2) eine Konstantenmenge K,
(3) zu jedem $n \in \mathbb{N}_+$ eine Menge F_n von Funktionssymbolen.

Dabei können die auftretenden Mengen leer sein, es ist für die Funktionssymbole sogar typisch, dass es nicht zu jeder Stelligkeit (zu jedem n) ein Funktionssymbol gibt. Die Konstanten kann man auch als nullstellige Funktionssymbole auffassen. Unter dem *Termalphabet* versteht man die Vereinigung

$$A = V \cup K \cup \bigcup_{n \in \mathbb{N}_+} F_n\,.$$

Die arithmetische Grundtermmenge besteht aus den beiden Konstanten $0, 1$ und den beiden zweistelligen Funktionssymbolen $\{+, \cdot\}$. Die Variablenmenge wird häufig als $x_1, x_2, x_3, \ldots$ angesetzt.

Definition 2.5. Zu einer Grundtermmenge $G = (V, K, F_n)$ ist die zugehörige *Termmenge* (oder die Menge der G-Terme) diejenige Teilmenge $T = T(G)$ der Wörter A^* über dem Termalphabet $A = V \cup K \cup \bigcup_{n \in \mathbb{N}_+} F_n$, die durch die folgenden rekursiven Vorschriften festgelegt wird.

(1) Jede Variable $v \in V$ ist ein Term.
(2) Jede Konstante $c \in K$ ist ein Term.
(3) Für jedes $f \in F_n$ und n Terme $t_1, t_2, \ldots, t_n$ ist auch $f t_1 t_2 \ldots t_n$ ein Term.

Hierbei sind (1) und (2) die Anfangsbedingungen und (3) der Rekursionsschritt, da darin auf schon gebildete Terme Bezug genommen wird. Wie bei

[2]Zwei Mengen L und M heißen *disjunkt*, wenn ihr Durchschnitt $L \cap M = \emptyset$ ist.

jeder rekursiven Definition ist ein Wort nur dann ein Term, wenn es gemäß dieser Regeln gebildet werden kann.

Gemäß dieser Definition verzichten wir auf Klammern; schon in einfachen Beispielen ist es aber wegen der Lesbarkeit sinnvoll, auch Klammern zu verwenden. Ebenso werden die Funktionssymbole einheitlich links geschrieben[3] und daran werden rechts davon die Terme angefügt (das wird später so interpretiert, dass in n-stellige Funktionen n Terme eingesetzt werden). Auch hierbei ist es in Beispielen sinnvoll, von dieser strengen Reihenfolge abzuweichen und beispielsweise $a + b$ statt $+ab$ zu schreiben.

Beispiel 2.6. Eine Grundtermmenge sei durch die Variablenmenge $V = \{x, y, z\}$, eine Konstantenmenge $K = \{c_1, c_2\}$, die einstelligen Funktionssymbole $F_1 = \{f, g\}$ und die zweistelligen Funktionssymbole $F_2 = \{\alpha, \beta, \gamma\}$ gegeben. Dann sind die folgenden Wörter Terme.

$$x, y, z, c_1, c_2, fx, fy, fc_1, gz, \alpha xy, \alpha xx, \alpha xfy, \alpha fxgc_1, \gamma\gamma xxx, \beta\alpha xgc_2\gamma fy\alpha gzx\,.$$

Auch wenn es für das Auge etwas ungewohnt aussieht, so sind diese Terme auch ohne Klammern allesamt wohldefiniert. Davon überzeugt man sich, indem man die Terme von links nach rechts liest, und dabei bei jedem Funktionssymbol die zugehörige Stelligkeit bestimmt (zu welchem F_n gehört das Funktionssymbol) und dann die folgenden Symbole in die geforderten n Terme aufspaltet (wenn dies nicht geht, so ist das Wort kein Term). Dabei entsteht schnell eine große Verschachtelungstiefe. Den letzten angeführten Term, also

$$\beta\alpha xgc_2\gamma fy\alpha gzx\,,$$

kann man mit (suggestiven) Klammern und Kommata nach und nach lesbarer gestalten. Er beginnt mit dem zweistelligen Funktionssymbol β, also muss das Folgende aus zwei Termen bestehen. Es folgt zunächst das ebenfalls zweistellige Funktionssymbol α, worauf zwei Terme folgen müssen. Wenn diese gefunden sind, muss der verbleibende Rest (also alles, was weiter rechts steht) den zweiten Term bilden, der von β verlangt wird. Die zwei Terme des an zweiter Stelle stehenden α sind x und gc_2. Man kann also den Term nach dieser Analyse auch als

$$\beta(\alpha(x, g(c_2)), \gamma fy\alpha gzx)$$

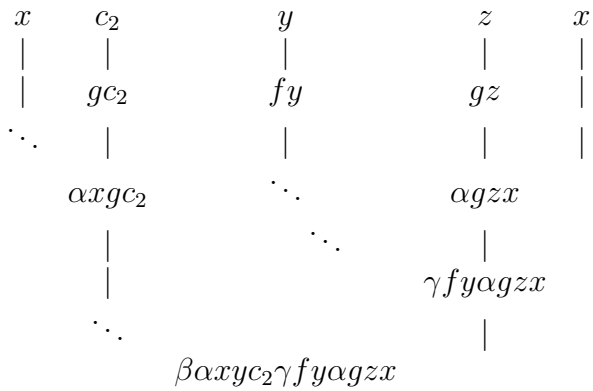
schreiben. Wenn man ebenso den zweiten Term für das äußere β auflöst, so erhält man

$$\beta(\alpha(x, g(c_2)), \gamma(fy, \alpha(g(z), x)))\,.$$

Übrigens kann man auch bei einem beliebigen Funktionssymbol mittendrin beginnen und die zugehörigen Terme, auf die es Bezug nimmt, bestimmen. Besonders übersichtlich wird die Termstruktur durch einen *Termstammbaum* ausgedrückt. Dabei werden die verwendeten Variablen und Konstanten

[3] Man spricht von *polnischer Notation*.

(mehrfach, um die unterschiedlichen Stellen, in die sie eingesetzt werden, beachten zu können) als Blätter[4] nebeneinander aufgeführt. Sie bilden die 0-te Reihe des Baumes. Wenn ein n-stelliges Funktionssymbol auf n solche Blätter angewendet wird, so zeichnet man einen Knoten, bezeichnet ihn mit dem Funktionssymbol (bzw. dem Funktionssymbol mit den eingelesenen Termen) und verbindet es mit den eingelesenen Blättern (die Einlesungsreihenfolge entspricht der Blätterreihenfolge). So entsteht aus allen Funktionssymbolen, die nur auf Variablen und Konstanten Bezug nehmen, die erste Reihe des Baumes. Die Funktionssymbole, die auf solche Knoten (und Blätter) Bezug nehmen, bilden die nächste Reihe, u.s.w. Der Stamm des Baumes ist dann der in Frage stehende Term. In unserem Beispiel sieht das so aus:



Beispiel 2.7. Wir betrachten ein Modell für die Termmenge der natürlichen Zahlen. Als Grundtermmenge nehmen wir eine Variablenmenge V, die Konstantenmenge $K = \{0\}$, die einstelligen Funktionssymbolmenge $F_1 = \{N\}$ (N steht für Nachfolger) und die zweistellige Funktionssymbolmenge $F_2 = \{\alpha, \mu\}$ (für Addition und Multiplikation). Allein aus der Konstante 0 und dem Nachfolgersymbol N kann man dann für jede natürliche Zahl eine Repräsentierung finden, nämlich

$$N0\,, NN0,\ NNN0,\ NNNN0,\ \textit{etc.}$$

Typische Terme sind dann Ausdrücke wie (u, v, w seien Variablen)

$$\alpha NN0NNNv,\ \mu NN0\alpha NN0NNN0,\ \mu\alpha NNN0\mu NNuN0NNNNw, \text{ etc.}$$

Wenn man x' statt Nx, $(x + y)$ statt αxy und $(x \cdot y)$ statt μxy schreibt, so „verschönern“ sich diese Terme zu

$$(0'' + v'''),\ (0'' \cdot (0'' + 0''')),\ ((0''' + (u'' \cdot 0')) \cdot w''''), \text{ etc.}$$

Mit den Abkürzungen $1 = 0'$, $2 = 0''$ etc. wird daraus

$$2 + v''',\ (2 \cdot (2 + 3)),\ ((3 + (u'' \cdot 1)) \cdot w''''), \text{ etc.}\,.$$

Man beachte, dass die Einführung dieser Abkürzungen nicht bedeutet, dass dadurch die üblicherweise mit diesen Symbolen verwendeten Rechenregeln

[4]Dies ist die graphentheoretische Bezeichnung für die Startpunkte eines Baumes.

erlaubt sind. Im Moment ist der zweite Term oben nicht gleich 10, dem zehnten Nachfolger der 0.

Abschließend erklären wir, was die Variablenmenge eines Terms ist. Diese ist stets eine Teilmenge der Variablenmenge V und enthält diejenigen Variablen, die in dem Term irgendwo vorkommen. Die rekursive Definition lautet folgendermaßen.

(1) Wenn $t = x$ eine Variable ist, so ist $\mathrm{Var}(x) = \{x\}$.
(2) Wenn $t = c$ eine Konstante ist, so ist $\mathrm{Var}(c) = \emptyset$.
(3) Wenn f ein n-stelliges Funktionssymbol ist und wenn $t_1, \ldots, t_n$ Terme sind, so ist $\mathrm{Var}(ft_1, \ldots, t_n) = \mathrm{Var}(t_1) \cup \ldots \cup \mathrm{Var}(t_n)$.

2.4. **Arbeitsblatt.**

Aufgabe 2.1. Entwerfe einen Termstammbamm für den Term

$$f\alpha\alpha gx\alpha c_2 f\beta gy\alpha c_1 gfz\beta gc_1 fc_1$$

wie in Beispiel 2.6.

Aufgabe 2.2. Wir betrachten die arithmetische Grundtermmenge, die aus den Konstanten 0 und 1, den Variablen x_n $n \in \mathbb{N}$, dem einstelligen Funktionssymbol N und den beiden zweistelligen Fnuktionssymbolen α und μ besteht. Entscheide, ob die folgenden Wörter über diesem Termalphabet Terme sind oder nicht.

(1) $NNNNNNN01$,
(2) $NNNNNNx_1NNNNNNNNNNNx_2$,
(3) $\alpha NNNNNN0NNNNNNNNNNN1$,
(4) $NNN\mu NNN\mu 0NNNNNNNNNNN1$,
(5) $\mu\alpha\mu\alpha\mu\alpha 0101010$,
(6) $\alpha\alpha\alpha Nx_1Nx_2x_3x_4x_3$.

Schreibe diejenigen Wörter, die Terme sind, mit Klammern, $'$, $+$ und $\cdot$.

Aufgabe 2.3. Es sei G eine Grundtermmenge und $t \in T(G)$ ein G-Term. Es sei u das am weitesten links stehende Symbol von t und v das am weitesten rechts stehende Symbol von t. Zeige die folgenden Eigenschaften.

(1) Wenn u eine Variable oder eine Konstante ist, so ist $t = u$.
(2) v ist eine Variable oder eine Konstante.
(3) Wenn t_1 und t_2 Terme sind, so ist t_1t_2 kein Term.

Aufgabe 2.4. Es sei G eine Grundtermmenge und t ein G-Term. Es sei n die Gesamtzahl der Variablen und Konstanten in t, wobei mehrfaches Vorkommen auch mehrfach gezählt wird. Es sei k die Summe über alle Stelligkeiten der in t vorkommenden Funktionssymbolen, wobei wiederum mehrfach auftretende Symbole auch mehrfach gezählt werden.

(1) Bestimme n und k im Term

$$ggxyhfxfzgyfy\,,$$

wobei f einstellig, g zweistellig und h dreistellig sei.
(2) Es sei t weder eine Variable noch eine Konstante. Zeige $k \geq n$.
(3) Zeige, dass die Differenz $n - k$ beliebig groß sein kann.

Die folgende Aufgabe verwendet den Begriff abzählbar.

Aufgabe 2.5. Es sei A ein abzählbares Alphabet. Zeige, dass auch die Menge A^* der Wörter über A abzählbar ist.

3. Vorlesung

Prädikatenlogik

Aristoteles (384-322 v.C.) gilt als Erfinder der Prädikatenlogik. Er verwendet in seiner Analytik Variablen, einstellige Prädikate, Quantoren und die logischen Junktoren.

Wir beginnen mit dem syntaktischen Aufbau der Prädikatenlogik mit Identität. Um den Aufbau dieser formalen Sprache zu motivieren und das Verständnis der zunächst rein formalen Ausdrücke zu erleichtern, ist es hilfreich, an Bildungsweisen von mathematischen Aussagen zu erinnern. Der konsequente Aufbau der Semantik folgt in der nächsten Vorlesung.

3.1. **Relationen.**

Ein Term kann weder wahr noch falsch sein, und zwar unabhängig davon, ob man ihn einfach als ein nach gewissen formalen Regeln aufgebautes Symbolwort auffasst oder ihn in einer bestimmten Menge (etwa den natürlichen Zahlen) interpretiert. Wahr oder falsch können nur Aussagen sein. Wichtig sind für uns zunächst die formalen Eigenschaften einer Aussage. In mathematischen Aussagen kommen häufig Terme zusammen mit einem Vergleichssymbol vor, z. B. in der (wahren) Gleichung

$$2 \cdot (2 + 3) = 10$$

oder der (falschen) Abschätzung

$$2 \cdot (2 + 3) < 10\,.$$

Mit zwei Termen und dem Gleichheitszeichen oder Kleinerzeichen gelangt man also zu Aussagen, man spricht von zweistelligen Relationen (in Logik und Grammatik auch von zweistelligen Prädikaten). Der Wahrheitsgehalt hängt dabei von den zwei Eingaben ab.

Eine einstellige Relation oder ein Prädikat ist eine Eigenschaftsform, die einem Element zukommen kann oder nicht, z.B. die Eigenschaft einer natürlichen Zahl, prim zu sein oder gerade zu sein oder eine Quadratzahl zu sein, oder das Positivitätsprädikat, das besagt, dass eine reelle Zahl positiv ist. Einstellige Prädikate definieren eine Teilmenge einer gegebenen Grundmenge: einem einstelligen Prädikat wird diejenige Teilmenge zugeordnet, die aus allen Elementen besteht, für die das Prädikat gilt. Daher entspricht die Mengenlehre der Prädikatenlogik mit nur einstelligen Prädikaten.

Mit n-stelligen Relationenssymbolen und n Termen gelangt man ebenfalls zu einer Aussage. Wenn z.B. A, B, C als Punkte in der Ebene interpretiert werden können, und G die Relation „bildet ein gleichseitiges Dreieck“ bedeutet, so bedeutet $G(A, B, C)$, dass diese drei Punkte ein gleichseitiges Dreieck bilden. Der Wahrheitsgehalt hängt natürlich von der Lage der Punkte A, B, C ab, hier interessiert aber lediglich, dass $G(A, B, C)$ eine sinnvolle Aussageform repräsentiert.

Andere geometrische Beispiele für dreistellige Relationen sind die Eigenschaften, dass die drei Punkte A, B, C auf einer Geraden liegen, sagen wir

$L(A, B, C)$, oder dass die drei Punkte ein rechtwinkliges Dreieck bilden, wobei der rechte Winkel an dem zuerst genannten Eckpunkt liegen muss, sagen wir $R(A, B, C)$. Man kann sich darüber streiten, ob bei einem Dreieck die Eckpunkte alle verschieden sein müssen, jedenfalls kann man die Eigenschaft der drei Punkte, dass sie paarweise verschieden sind, durch ein dreistelliges Prädikat ausdrücken, sagen wir $E(A, B, C)$.

3.2. **Quantoren.**

Mathematische Aussage enthalten häufig auch Existenzaussagen. Wenn wir bei dem eben erwähnten Beispiel bleiben, so bedeutet

$$\text{es gibt } z\, G(A, B, z)$$

die Aussage, dass es zu gegebenen festen A und B ein z gibt derart, dass die drei Punkte A, B, z ein gleichseitiges Dreieck bilden (diese Aussage ist in der reellen Zahlenebene wahr). In dem Beispielsatz wird nur über z quantifiziert, nicht über A und B. Dies kann man durch die folgenden Aussagen erreichen.

$$\text{es gibt } x \text{ und es gibt } y \text{ und es gibt } z\, G(x, y, z)\,,$$

was bedeutet, dass es Punkte x, y, z gibt, die ein gleichseitiges Dreieck bilden, die wahr ist, aber deutlich schwächer als die Aussage

$$\text{für alle } x \text{ und } \text{ für alle } y \text{ gibt es } z\, G(x, y, z)$$

ist, die behauptet, dass es zu (beliebig vorgegebenen) Eckpunkten x und y stets einen dritten Punkt gibt, so dass ein gleichseitiges Dreieck entsteht.[5] Die Ausdrücke „es gibt“ und „für alle“ nennt man *Quantoren*. Für diese Quantoren gibt es spezielle Symbole, nämlich $\exists$ für „es gibt“ und $\forall$ für „für alle“. Die obigen Beispielsätze schreibt man dann formal als

$$\exists x \exists y \exists z G(x, y, z)$$

bzw. als

$$\forall x \forall y \exists z G(x, y, z)\,.$$

Auf die Reihenfolge bei gleichartigen Quantoren kommt es nicht an (dies ist von der inhaltlichen Bedeutung her klar, wird später aber auch formal im Ableitungskalkül nachgebildet), sie ist aber bei wechselnden Quantoren entscheidend. Beispielsweise ist die Aussage

$$\exists z \forall x \forall y G(x, y, z)$$

(also die Aussage, dass es einen Punkt gibt, der mit je zwei anderen beliebigen Punkten eine gleichseitiges Dreieck bildet) im Gegensatz zur vorherigen Aussage nicht wahr.

[5] Die Gültigkeit dieser Aussagen setzt voraus, dass wir über den reellen Zahlen bzw. in der reellen Zahlenebene arbeiten.

3.3. **Junktoren.**

Eine weitere Art von mathematischen Aussagen entsteht dadurch, dass man Aussagen selbst zueinander in eine logische Beziehung setzt, indem man beispielsweise sagt, dass aus der Aussage p die Aussage q folgt, oder dass p und q zueinander äquivalent sind. Der Satz des Pythagoras besagt, dass wenn zwischen drei Punkten A, B, C in der Ebene die Beziehung der Rechtwinkligkeit am Punkt A besteht, dass dann zwischen den durch die drei Punkte definierten Streckenlängen ebenfalls eine bestimmte Beziehung[6] besteht. Wenn man die Rechtwinkligkeit wie oben mit dem dreistelligen Relationssymbol R und die Streckenbeziehung mit dem dreistelligen Relationssymbol S bezeichnet, so gilt also

$$\text{aus } R(A,B,C) \text{ folgt } S(A,B,C)\,,$$

was wir formal als

$$\forall A \forall B \forall C (R(A,B,C) \longrightarrow S(A,B,C))$$

schreiben. Gilt davon auch die Umkehrung? Folgt also aus der (für den Satz des Pythagoras typischen Streckenbeziehung) $S(A,B,C)$, dass ein rechter Winkel an A vorliegt? Dies ist in der Tat der Fall! Der Kosinussatz besagt für ein beliebiges (echtes) Dreieck mit einem an A anliegenden Winkel, dass

$$d(B,C)^2 = d(A,B)^2 + d(A,C)^2 - 2d(A,B)d(A,C)\cos\alpha$$

gilt, wobei d den Abstand zwischen zwei Punkten bezeichne. Der „Störterm“ rechts entfällt genau dann, wenn $\cos\alpha = 0$ ist, und dies ist nur bei 90 Grad der Fall. Daher können wir die Äquivalenz

$$\forall A \forall B \forall C (R(A,B,C) \longleftrightarrow S(A,B,C))$$

schreiben (ein Dreieck, bei dem zwei Eckpunkte zusammenfallen, akzeptieren wir als rechtwinklig an dem doppelten Punkt).

Unser Rechtwinkligkeitsprädikat $R(A,B,C)$ besagt, dass der Winkel am Eckpunkt A ein Rechter ist. Wenn man sich dafür interessiert, ob überhaupt ein rechtwinkliges Dreieck vorliegt, so muss $R(A,B,C)$ oder $R(B,C,A)$ oder $R(C,A,B)$ gelten. Die Oderverknüpfung wird formal als

$$(R(A,B,C) \vee R(B,C,A)) \vee R(C,A,B)$$

geschrieben (die Assoziativität der oder-Verknüpfung steht im Moment noch nicht zur Verfügung).

Für ein echtes Dreieck haben wir oben gefordert, dass die konstituierenden Punkte A, B, C paarweise verschieden sind. Die Gleichheit von zwei Punkten wird durch $A = B$ und die Negation davon, also die Verschiedenheit der beiden Punkte, wird in der Mathematik durch $A \neq B$, in der Logik aber durch $\neg\,()$ ausgedrückt. Dass drei Punkte paarweise verschieden sind, erfordert ein

[6] Zur Erinnerung: das Quadrat der Streckenlänge zwischen B und C (die Hypothenuse) ist gleich der Summe der Quadrate der beiden Streckenlängen zwischen A und B und A und C (den Katheten).

logisches und, das durch $\wedge$ symbolisiert wird, so dass sich die Echtheit eines Dreiecks durch

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

ausdrücken lässt.

3.4. **Sprachen erster Sufe.**

Die erwähnten Konstruktionsmöglichkeiten für Aussagen sind im Wesentlichen schon erschöpfend. Mit ihnen kann man formale Sprachen aufbauen, deren Aussagekraft prinzipiell groß genug ist, um die gesamte Mathematik auszudrücken (für viele Bereiche wäre es aber künstlich, sich auf diese Sprachen zu beschränken). Diese formalen Sprachen nennt man *Sprachen erster Stufe*, wir beginnen mit den zugehörigen Alphabeten.

Definition 3.1. Ein *Alphabet einer Sprache erster Stufe* umfasst die folgenden Daten.

(1) Eine Grundtermmenge, also eine Menge aus Variablen, Konstanten und Funktionssymbolen.
(2) Zu jeder natürlichen Zahl $n \in \mathbb{N}_+$ eine Menge R_n von n-stelligen Relationssymbolen.
(3) Die aussagenlogischen Junktoren
$$\neg,\ \wedge,\ \vee,\ \rightarrow,\ \leftrightarrow\ .$$
(4) Das Gleichheitszeichen $=$.
(5) Die Quantoren $\forall$ und $\exists$.
(6) Klammern, also (und).

Die aussagenlogischen Junktoren werden als *Negation*, *Konjunktion* (und), *Disjunktion* (*Alteration*, einschließliches Oder), *Implikation* (wenn, dann) und *Äquivalenz* (genau dann, wenn) bezeichnet. Der Quantor $\forall$ heißt *Allquantor* und $\exists$ heißt *Existenzquantor*. Diese Liste ist etwas redundant, da man, von der späteren Interpretation her gesehen, einige aussagenlogische Junktoren durch andere ersetzen kann, beispielsweise ist für zwei Aussagen p und q die Aussage $p \rightarrow q$ gleichwertig mit $\neg p \vee q$, und so könnte man den Implikationspfeil auch weglassen. Ebenso kann man den einen Quantor mit Hilfe des anderen und der Negation ausdrücken, es ist nämlich $\forall x p$ gleichbedeutend mit $\neg\exists x \neg p$. Um die Lesbarkeit von Ausdrücken zu erhöhen ist es aber alles in allem vorteilhaft, nicht allzu minimalistisch sein zu wollen (man könnte die unnötigen Symbole auch als Abkürzungen einführen). Das Gleichheitszeichen könnte man zwar auch als ein weiteres zweistelliges Relationssymbol auffassen, allerdings sind die weiter unten einzuführenden Schlussregeln für das Gleichheitszeichen (insbesondere die Möglichkeit einzusetzen) für die Logik erster Stufe konstitutiv. Da ein Alphabet einer Sprache erster Stufe eine Termgrundmenge enthält, ist klar, was als Term in der Sprache zu gelten hat. Als nächstes erklären wir formal, was wir als einen Ausdruck (oder eine formale Aussage) in dieser Sprache ansehen.

Definition 3.2. Es sei ein Alphabet einer Sprache erster Stufe gegeben. Dann nennt man die folgenden rekursiv definierten Wörter über diesem Alphabet die *Ausdrücke* dieser Sprache.

(1) Wenn t_1 und t_2 Terme sind, so ist
$$t_1 = t_2$$
ein Ausdruck.

(2) Wenn R ein n-stelliges Relationssymbol ist, und $t_1, \ldots, t_n$ Terme sind, so ist
$$Rt_1 \ldots t_n$$
ein Ausdruck.

(3) Wenn p und q Ausdrücke sind, so sind auch
$$\neg p,\ (p \wedge q),\ (p \vee q),\ (p \to q),\ (p \leftrightarrow q)$$
Ausdrücke.

(4) Wenn p ein Ausdruck ist und x eine Variable, so sind auch
$$\forall x p \text{ und } \exists x p$$
Ausdrücke.

Die Klammern sind hier auch nur nötig, weil wir die zweistelligen Junktoren anders als die Funktionssymbole in der Mitte schreiben. Die Menge der Konstanten, der Variablen, der Funktionssymbole und der Relationssymbole nennt man zusammen auch das *Symbolalphabet* der Sprache. Die anderen Symbole (Junktoren, Quantoren, Gleichheitszeichen, Klammern) sind immer gleich, so dass eine Sprache erster Stufe im Wesentlichen nur von der gewählten Symbolmenge S abhängt. Für die zugehörige Sprache schreibt man L^S.

3.5. **Freie und gebundene Variablen.**

In einem Ausdruck $p \in L^S$ über einem Symbolalphabet S nennt man die Variablen, die innerhalb der Reichweite eines Quantors stehen, *gebunden*, die anderen *frei*. Dies wird streng über den Aufbau der Ausdrücke definiert.

(1)
$$\text{Frei}\,(t_1 = t_2) = \text{Var}\,(t_1) \cup \text{Var}\,(t_2)$$

(2)
$$\text{Frei}\,(Rt_1 \ldots t_n) = \text{Var}\,(t_1) \cup \text{Var}\,(t_2) \cup \ldots \cup \text{Var}\,(t_n)$$
für ein n-stelliges Relationssymbol R und n Terme $t_1, t_2, \ldots, t_n$.

(3)
$$\text{Frei}\,(\neg p) = \text{Frei}\,(p)$$
für einen Ausdruck p.

(4)
$$\text{Frei}\,(p \to q) = \text{Frei}\,(p) \cup \text{Frei}\,(q)$$
für Ausdrücke p und q. Ebenso für $\leftrightarrow, \wedge, \vee$.

(5)
$$\text{Frei}\,(\forall x p) = \text{Frei}\,(p) \setminus \{x\}$$
für einen Ausdruck p und eine Variable x.

(6)
$$\text{Frei}\,(\exists x p) = \text{Frei}\,(p) \setminus \{x\}$$
für einen Ausdruck p und eine Variable x.

Einen Ausdruck ohne freie Variablen nennt man einen *Satz*, auch wenn diese Bezeichnung nicht ganz glücklich ist, da „Satz“ die Gültigkeit einer Aussage suggeriert. Die Menge der Sätze wird mit L_0^S bezeichnet, die Menge der Ausdrücke mit genau einer freien Variablen (die aber in dem Ausdruck beliebig oft vorkommen darf) mit L_1^S.

3.6. **Arbeitsblatt.**

Aufgabe 3.1. Formuliere die folgenden Beziehungen (ein- oder mehrstellige Prädikate) innerhalb der natürlichen Zahlen $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$ allein mittels Gleichheit, Addition, Multiplikation und unter der Verwendung von aussagenlogischen Junktoren und Quantoren.

(1) $x \geq y$.
(2) $x > y$.
(3) x teilt y.
(4) x teilt nicht y.
(5) x ist eine Quadratzahl.
(6) x ist eine Primzahl.
(7) x ist keine Primzahl.
(8) x ist das Produkt von genau zwei verschiedenen Primzahlen.
(9) x wird von einer Primzahl geteilt.

Aufgabe 3.2. Formalisere in der arithmetischen Sprache (mit + und $\cdot$) die folgenden (wahren) Aussagen.

(1) Wenn $x \geq y$ und $y \geq z$, so ist $x \geq z$.
(2) Wenn $x \geq y$ und $y \geq x$ gilt, so ist $x = y$.
(3) Für jede natürliche Zahl gibt es eine größere natürliche Zahl.
(4) Eine natürliche Zahl, für die es keine kleinere natürliche Zahl gibt, ist gleich 0.

Aufgabe 3.3. Schreibe die folgenden Aussagen mit Quantoren:

(1) Für jede natürliche Zahl gibt es eine größere Zahl.
(2) Für jede natürliche Zahl gibt es eine kleinere Zahl.
(3) Es gibt eine natürliche Zahl, die größer oder gleich jeder anderen natürlichen Zahl ist.

(4) Es gibt eine natürliche Zahl, die kleiner oder gleich jeder anderen natürlichen Zahl ist.

Welche sind wahr, welche falsch?

Aufgabe 3.4. Formalisiere die folgenden mengentheoretischen Fassungen einiger aristotelischer Syllogismen in der Prädikatenlogik erster Stufe.

(1) Modus Barbara: Aus $B \subseteq A$ und $C \subseteq B$ folgt $C \subseteq A$.
(2) Modus Celarent: Aus $B \cap A = \emptyset$ und $C \subseteq B$ folgt $C \cap A = \emptyset$.
(3) Modus Darii: Aus $B \subseteq A$ und $C \cap B \neq \emptyset$ folgt $C \cap A \neq \emptyset$.
(4) Modus Ferio: Aus $B \cap A = \emptyset$ und $C \cap B \neq \emptyset$ folgt $C \not\subseteq A$.
(5) Modus Baroco: Aus $B \subseteq A$ und $B \not\subseteq C$ folgt $A \not\subseteq C$.

Aufgabe 3.5. Formalisiere in der arithmetischen Sprache die folgenden wahren Aussagen.

(1) Es gibt unendlich viele Primzahlen.
(2) Jede natürliche Zahl ≥ 2 wird von einer Primzahl geteilt.

Wie sieht es mit der Aussage aus, dass jede natürliche Zahl eine Primfaktorzerlegung besitzt?

Aufgabe 3.6. Formalisiere in der arithmetischen Sprache die folgenden zahlentheoretischen Vermutungen.

(1) Die Goldbach-Vermutung.
(2) Die Vermutung über die Unendlichkeit der Primzahlzwillinge.
(3) Die Vermutung über die Unendlichkeit der Mersenne-Primzahlen.

Man beachte bei (3), dass das Potenzieren mit einem unbekannten Exponenten nicht zur arithmetischen Sprache gehört.

Aufgabe 3.7. Zeige, dass es kein gleichseitiges Dreieck gibt, dessen sämtliche Ecken rationale Koordinaten besitzen.

Diese Aufgabe ist nicht ganz einfach. Zur Lösung verwende man, dass $\sqrt{3}$ irrational ist und den Satz des Pythagoras.

4. Vorlesung

4.1. **Semantik.**

Gelegentlich haben wir schon angedeutet, was die zuletzt eingeführten prädikatenlogischen Symbole, die wir rein formal als Zeichenreihen behandelt haben, eigentlich bedeuten sollen, was also ihr logisch-mathematischer Gehalt sein soll. Bei einer solchen Interpretation werden die Junktoren, die Quantoren und das Gleichheitszeichen stets in der gleichen Weise interpretiert, die Variablen, Konstanten, Relations- und Funktionssymbole aber unterschiedlich. Dazu erinnern wir an einige mathematische Begriffe. Wir setzen eine naive Mengenlehre und die natürlichen Zahlen „zum Zählen“ (wie schon weiter oben) voraus. Wir erinnern an einige grundlegende mathematische Definitionen.

Definition 4.1. Es seien zwei Mengen L und M gegeben. Dann nennt man die Menge

$$L \times M = \{(x, y) |\, x \in L,\, y \in M\}$$

die *Produktmenge* der beiden Mengen.

Für uns ist insbesondere das n-fache Produkt einer Menge M mit sich selbst, also

$$M^n = M \times M \times \cdots \times M$$

(mit n Faktoren) wichtig.

Definition 4.2. Unter einer *n-stelligen Relation* R auf einer Menge M versteht man eine Teilmenge der n-fachen Produktmenge $M \times \cdots \times M$.

Definition 4.3. Seien L und M zwei Mengen. Eine *Abbildung* F von L nach M ist dadurch gegeben, dass jedem Element der Menge L genau ein Element der Menge M zugeordnet wird. Das zu $x \in L$ eindeutig bestimmte Element wird mit $F(x)$ bezeichnet. Die Abbildung drückt man als Ganzes häufig durch

$$F : L \longrightarrow M,\, x \longmapsto F(x),$$

aus.

Definition 4.4. Es sei M eine Menge. Unter einer *n-stelligen Funktion* auf M versteht man eine Abbildung

$$f : M \times \cdots \times M \longrightarrow M,\, (x_1, \ldots, x_n) \longmapsto f(x_1, \ldots, x_n),$$

vom n-fachen Produkt von M mit sich selbst nach M.

Eine n-stellige Funktion kann auch als eine $(n+1)$-stellige Relation aufgefasst werden, bei der es zu jedem n-Tupel $(x_1, \ldots, x_n)$ genau ein x_{n+1} gibt derart, dass $(x_1, \ldots, x_n, x_{n+1})$ zur Relation gehört. Dieses x_{n+1} ist dann der Funktionswert der zugehörigen Funktion an der Stelle $(x_1, \ldots, x_n)$.

4.2. **Interpretationen.**

Definition 4.5. Es sei S das Symbolalphabet einer Sprache erster Stufe. Unter einer *S-Struktur* versteht man eine Menge M mit den folgenden Festlegungen.

(1) Für jede Konstante $c \in C$ ist ein Element $c^M \in M$ festgelegt.
(2) Zu jedem n-stelligen Funktionssymbol f (aus S) ist eine n-stellige Funktion
$$f^M : M^n \longrightarrow M$$
festgelegt.
(3) Zu jedem n-stelligen Relationssymbol R (aus S) ist eine n-stellige Relation
$$R^M \subseteq M^n$$
festgelegt.

Unter einer *S-(Variablen)belegung* in M versteht man eine Festlegung x^M für jede Variable $x \in \mathrm{Var}(S)$.

Unter einer *S-Interpretation* versteht man eine S-Struktur zusammen mit einer S-Belegung.

Die Menge M heißt auch *Grundmenge* der S-Struktur bzw. der S-Interpretation. Die Festlegung für die Konstanten und die Variablen ist einfach eine Abbildung von C bzw. von der Variablenmenge in die Menge M. Statt c^M, x^M, F^M, R^M schreibt man auch $I(c), I(x), F^I, R^I$, wobei I eine Interpretation bezeichnet. Die Strukturen sind übliche Gegenstände der Mathematik.

Beispiel 4.6. Es sei S ein Alphabet, das außer einer Variablenmenge V aus einem einzigen einstelligen Funktionssymbol F bestehe (die Konstantenmenge und die Relationssymbolmengen seien also leer). Eine S-Struktur besteht dann aus einer Menge M zusammen mit einer Abbildung
$$f = F^M : M \longrightarrow M,\ a \longmapsto f(a).$$
Beispiele sind $M = \mathbb{N}$ mit der Nachfolgerfunktion, $M = \mathbb{R}$ mit dem Quadrieren $x \mapsto x^2$ oder der Sinusfunktion oder der Exponentialfunktion, oder eine beliebige Menge mit der Identität, u.s.w.

Beispiel 4.7. Es sei S ein Alphabet, das außer einer Variablenmenge V aus einem einzigen zweistelligen Funktionssymbol F bestehe (die Konstantenmenge und die Relationssymbolmengen seien also leer). Eine S-Struktur besteht dann aus einer Menge M zusammen mit einer Abbildung
$$f = F^M : M \times M \longrightarrow M,\ (a, b) \longmapsto f(a, b).$$
Eine solche Abbildung nennt man auch eine Verknüpfung auf M

sie ordnet (einem geordneten Paar aus) zwei Elementen der Menge ein weiteres Element der Menge zu. Die Addition oder die Multiplikation auf den natürlichen Zahlen sind jeweils eine solche Verknüpfung.

Beispiel 4.8. Es sei S ein Alphabet, das außer einer Variablenmenge V aus einem einzigen einstelligen Relationssymbol R bestehe (die Konstantenmenge und die Funktionssymbolmengen seien also leer). Eine S-Struktur besteht dann aus einer Menge M zusammen mit einer fixierten Teilmenge $U \subseteq M$. Beispiele sind $M = \mathbb{N}$ mit der Teilmenge der Primzahlen, oder der Teilmenge der Quadratzahlen, oder $M = \mathbb{R}$ mit der Teilmenge der positiven Zahlen, oder der Teilmenge der rationalen Zahlen, u.s.w.

4.3. **Interpretation von Termen.**

Mit einer solchen Interpretation wird das Symbolalphabet, das neben den Junktoren, Quantoren, dem Gleichheitszeichen und den Klammern das Alphabet der Sprache bildet, interpretiert. Man möchte aber die gesamte Sprache in M, ausgehend von der Interpretation dieser Symbole, interpretieren. Der erste Schritt dazu ist die Interpretation der Terme. Die Wohldefiniertheit der folgenden Festlegung ergibt sich durch einen Beweis über den Aufbau der Terme.

Definition 4.9. Zu einem Symbolalphabet S erster Stufe und einer S-Interpretation in einer Menge M wird induktiv über den Aufbau der Terme für jeden S-Term t eine Interpretation $I(t)$ in M definiert.

(1) Für jede Konstante c und jede Variable x ist die Terminterpretation durch die Interpretation bzw. die Belegung direkt gegeben, also $I(c) = c^M$ und $I(x) = x^M$.
(2) Wenn $t_1, \ldots, t_n$ Terme sind mit Interpretationen $I(t_1), \ldots, I(t_n)$ und wenn f ein n-stelliges Funktionsymbol ist, so wird der Term $ft_1 \cdots t_n$ als $f^M(I(t_1), \ldots, I(t_n))$ interpretiert.

Damit werden alle Terme in der Grundmenge M interpretiert. In vielen Situationen bleibt die Grundmenge und die Interpretation der Konstanten und der Relations- und Funktionssymbole gleich, während man die Variablenbelegung ändern möchte. Insbesondere möchte man Interpretationen für eine einzelne Variable abändern.

Definition 4.10. Es sei ein Symbolalphabet S erster Stufe und eine S-Interpretation I in einer Menge M gegeben. Es sei x eine Variable und $m \in M$ ein Element der Grundmenge. Dann versteht man unter $I\frac{m}{x}$ diejenige Interpretation von S in M, die strukturgleich zu I ist und für deren Variablenbelegung gilt

$$\left(I\frac{m}{x}\right)(y) = \begin{cases} I(y), & \text{falls } y \neq x \\ m, & \text{falls } y = x\,. \end{cases}$$

4.4. **Interpretation von Ausdrücken.**

Nachdem wir alle Terme bei einer gegebenen S-Interpretation interpretieren können, wenden wir uns nun den Ausdrücken zu. Es ist das Ziel, jedem

S-Ausdruck eine Aussage (unter Bezug auf die Grundmenge M und die Interpretation des Symbolalphabets) zuzuordnen, die wahr oder falsch ist.

Definition 4.11. Zu einem Symbolalphabet S erster Stufe und einer S-Interpretation I in einer Menge M werden die S-Ausdrücke folgendermaßen (induktiv über den Aufbau der Ausdrücke) interpretiert und als gültig (oder ungültig) charakterisiert (die Gültigkeit einer Aussage p unter der Interpretation wird dabei als $I \vDash p$ geschrieben). Es seien $s, t, t_1, \ldots, t_n$ Terme und p, q Ausdrücke.

(1) $I \vDash s = t$, wenn $I(s) = I(t)$.
(2) $I \vDash Rt_1 \ldots t_n$, wenn $(I(t_1), \ldots, I(t_n)) \in R^M$.
(3) $I \vDash \neg p$, wenn nicht $I \vDash p$ gilt.
(4) $I \vDash p \wedge q$, wenn $I \vDash p$ und $I \vDash q$ gilt.
(5) $I \vDash p \to q$, wenn die Gültigkeit $I \vDash p$ die Gültigkeit $I \vDash q$ impliziert.
(6) $I \vDash \exists x p$, wenn es ein $m \in M$ gibt mit $I\frac{m}{x} \vDash p$.
(7) $I \vDash \forall x p$, wenn für alle $m \in M$ die Beziehung $I\frac{m}{x} \vDash p$ gilt.

Dabei ist, wie bei jeder Definition, „wenn“ als „genau dann, wenn“ zu lesen. Auf der linken Seite stehen die formalen Ausdrücke zusammen mit der Erklärung, ob sie in der Interpretation gelten, und auf der rechten Seite steht eine logisch-mathematische Bedingung. Diese ist im Sinne des üblichen Gebrauchs in der Mathematik zu verstehen.

Da bei dieser Zuordnung alle möglichen Konstruktionsweisen für Ausdrücke auftauchen, ergibt sich eine Erklärung für jeden Ausdruck durch deren induktiven Aufbau. Für jeden Ausdruck p gilt in einer Interpretation I entweder $I \vDash p$ oder nicht, wobei die Nichtgültigkeit zur Gültigkeit von $I \vDash \neg p$ äquivalent ist. Eine Interpretation liefert also insbesondere eine *vollständige Aufteilung* der S-Ausdrücke in wahre und falsche Ausdrücke.

4.5. **Beispiele.**

Beispiel 4.12. Es sei S ein Alphabet, das außer einer Variablenmenge V aus einem einzigen einstelligen Funktionssymbol F bestehe (die Konstantenmenge und die Relationssymbolmengen seien also leer), so dass eine S-Struktur aus einer Menge M zusammen mit einer Abbildung

$$f = F^M : M \longrightarrow M,\ a \longmapsto f(a)$$

besteht. In einer solchen Interpretation wird jeder S-Ausdruck interpretiert. Der Ausdruck

$$p = \forall x(\exists y Fy = x)$$

besagt die Surjektivität von F. D.h. in einer S-Interpretation gilt

$$I \vDash p$$

genau dann, wenn die durch die Interpretation festgelegte Abbildung F^I surjektiv ist. Der Ausdruck

$$q = \forall x \forall y (Fx = Fy \to x = y)$$

besagt die Injektivität von F. D.h. in einer S-Interpretation gilt

$$I \vDash q$$

genau dann, wenn die durch die Interpretation festgelegte Abbildung F^I injektiv ist.

Beispiel 4.13. Es sei S das Symbolalphabet für einen angeordneten Körper, d.h. es gebe eine zweielementige Konstantenmenge $C = \{0, 1\}$, eine zweielementige Menge für die 2-stelligen Funktionssymbole $\{+, \cdot\}$ und eine einelementige Menge $\{\geq\}$ für ein zweistelliges Relationssymbol. Wir betrachten die Interpretation I_1 mit der Grundmenge $\mathbb{Q}$ und die Interpretation I_2 mit der Grundmenge $\mathbb{R}$, wobei Konstanten, Funktionssymbole und Relationssymbol in natürlicher Weise interpretiert werden (und die Variablenbelegung irgendwie festgelegt sei).

Der S-Ausdruck $1 + 1 \geq 1$ (also der Ausdruck $\geq +111$ in vorgestellter Notation) wird unter den Interpretationen als $1_{\mathbb{Q}} + 1_{\mathbb{Q}} \geq 1_{\mathbb{Q}}$ bzw. als $1_{\mathbb{R}} + 1_{\mathbb{R}} \geq 1_{\mathbb{R}}$ interpretiert und daher gelten $I_1 \vDash 1 + 1 \geq 1$ und $I_2 \vDash 1 + 1 \geq 1$. Dagegen ist der Ausdruck $\forall x (x \geq 0 \to \exists y (x = y \cdot y))$ unter I_1 falsch und unter I_2 richtig, also

$$I_1 \vDash \neg(\forall x (x \geq 0 \to \exists y (x = y \cdot y))) \text{ und } I_2 \vDash \forall x (x \geq 0 \to \exists y (x = y \cdot y)) .$$

Das vorstehende Beispiel zeigt, dass die Gültigkeit von Ausdrücken unter einer bestimmten Interpretation von Eigenschaften der Grundmenge abhängt und durch eine mathematische Argumentation erwiesen oder zurückgewiesen werden muss. Diese kann beliebig kompliziert sein. Insbesondere bedeutet die Modellbeziehung nicht, dass man für jeden Ausdruck entscheiden kann, ob er in einer Interpretation wahr oder falsch ist.

4.6. **Gültigkeit von Ausdrucksmengen.**

Für die Gültigkeitsbeziehung $I \vDash p$ sagt man auch, dass die Interpretation I ein *Modell* für den Ausdruck p ist oder den Ausdruck p erfüllt. Für eine Menge Γ von Ausdrücken schreibt man $I \vDash \Gamma$, wenn in I jeder Ausdruck aus Γ gilt. Man sagt, dass I ein *Modell* für Γ ist. Eine Struktur heißt ein *Modell*, wenn jede Variablenbelegung zu dieser Struktur eine Interpretation liefert, die ein Modell ist.

Diese Sprechweise wird insbesondere für Axiomensysteme Γ verwendet, die eine mathematisch wichtige Struktur festlegen. Die erfüllenden Modelle heißen dann so, wie der Definitionsname in der Definition lautet, die dieses Axiomensystem verwendet. Die Modelle sind im mathematischen Sprachgebrauch Beispiele für die Struktur, die durch die Definition festgelegt wird.

Betrachten wir beispielsweise die Definition einer Gruppe.

Definition 4.14. Eine Menge G mit einem ausgezeichneten Element $e \in G$ und mit einer Verknüpfung

$$G \times G \longrightarrow G,\ (g,h) \longmapsto g * h,$$

heißt *Gruppe*, wenn folgende Eigenschaften erfüllt sind.

(1) Die Verknüpfung ist *assoziativ*, d.h. für alle $f, g, h \in G$ gilt

$$(f * g) * h = f * (g * h)\,.$$

(2) Das Element e ist ein *neutrales Element*, d.h. für alle $g \in G$ gilt

$$g * e = g = e * g\,.$$

(3) Zu jedem $g \in G$ gibt es ein *inverses Element*, d.h. es gibt ein $h \in G$ mit

$$h * g = g * h = e\,.$$

In formal-prädikatenlogischer Formulierung besteht das Alphabet (neben den Variablen) aus einer Konstanten e und aus einem zweistelligen Funktionssymbol μ. Die in der Gruppendefinition auftretenden Axiome (die Gruppenaxiome, also die drei auftretenden Bedingungen) kann man mit diesen Symbolen einfach schreiben als

(1)
$$\forall x(\forall y(\forall z\, \mu x \mu y z = \mu \mu x y z))\,.$$

(2)
$$\forall x(\mu x e = x \wedge \mu e x = x)\,.$$

(3)
$$\forall x \exists y(\mu x y = e \wedge \mu y x = e)\,.$$

Nennen wir diese drei Ausdrücke Γ. Dann ist eine Gruppe eine Menge mit einer Interpretation I für e und für μ, d.h. es muss ein ausgezeichnetes Element e^G (häufig schreibt man e_G) geben und eine zweistellige Funktion (eine Verknüpfung), derart, dass $I \vDash \Gamma$ gilt.

4.7. **Arbeitsblatt.**

Aufgabe 4.1. Es sei das arithmetische Alphabet $\{0, 1, +, \cdot\}$ zusammen mit der Variablenmenge $\{x, y\}$ gegeben. Interpretiere den Term

$$((0 + 1) + x) \cdot (1 + (y + 1))$$

unter den folgenden Interpretationen.

(1) $M = \mathbb{N}$ mit der Standardinterpretation und der Variablenbelegung $I(x) = 5$ und $I(y) = 3$.

(2) $M = \mathrm{Mat}_2(\mathbb{R})$ mit der Standardinterpretation

$$I(0) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}, \; I(1) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

und der üblichen Matrizenaddition und Matrizenmultiplikation und der Variablenbelegung $I(x) = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ und $I(y) = \begin{pmatrix} 3 & -2 \\ 0 & 5 \end{pmatrix}$.

(3) $M = \mathbb{N}$, mit

$$I(0) = 1,\; I(1) = 4,\; I(x) = 2,\; I(y) = 1\,,$$

und wo $+$ als Multiplikation und $\cdot$ als Addition interpretiert wird.

(4) $M = \mathbb{Z}$, mit

$$I(0) = 5,\; I(1) = -1,\; I(x) = 0,\; I(y) = 0\,,$$

und wo sowohl $+$ als auch $\cdot$ als Subtraktion interpretiert werden.

(5) $M =$ Potenzmenge von $\{1, 2, 3, 4, 5\}$ mit

$$I(0) = \emptyset,\; I(1) = \{1, 2, 3, 4, 5\},\; I(x) = \emptyset,\; I(y) = \{2, 4\}\,,$$

und wo $+$ als $\cup$ und $\cdot$ als $\cap$ interpretiert wird.

Aufgabe 4.2. Es sei das arithmetische Alphabet $\{0, 1, +, \cdot\}$ zusammen mit der Variablenmenge $\{x, y\}$ gegeben. Interpretiere den Ausdruck

$$\forall x \exists y (x = y + y \vee x + 1 = y + y)$$

unter den in Aufgabe 4.1 angeführten Interpretationen und überprüfe die Gültigkeit.

Aufgabe 4.3. Erstelle einen prädikatenlogischen Ausdruck p, der in einer Struktur genau dann gilt, wenn die Grundmenge der Struktur genau 7 Elemente besitzt.

5. Vorlesung

5.1. **Weitere Axiomensysteme.**

In der letzten Vorlesung haben wir gesehen, wie man die Gruppenaxiome in der Prädikatenlogik erster Stufe formulieren kann. Eine Gruppe im herkömmlichen mathematischen Sinn ist prädikatenlogisch formuliert eine Menge zusammen mit einer Interpretation für eine Konstante und ein zweistelliges Funktionssymbol (nämlich ein ausgezeichnetes Element und eine Verknüpfung), unter der gemäß der Modellbeziehung die Gruppenaxiome gültig sind.

Wir geben ein weiteres Beispiel, das die Beziehung zwischen mathematischer und prädikatenlogischer Formulierung deutlich machen soll.

Definition 5.1. Eine Relation $\preccurlyeq$ auf einer Menge I heißt *Ordnungsrelation* oder *Ordnung*, wenn folgende drei Bedingungen erfüllt sind.

(1) Es ist $i \preccurlyeq i$ für alle $i \in I$.
(2) Aus $i \preccurlyeq j$ und $j \preccurlyeq k$ folgt stets $i \preccurlyeq k$.
(3) Aus $i \preccurlyeq j$ und $j \preccurlyeq i$ folgt $i = j$.

Neben den Variablen besteht das zugehörige Symbolalphabet allein aus einem zweistelligen Relationssymbol, das wir ebenfalls mit $\preccurlyeq$ bezeichnen. Die für eine Ordnung verlangten Eigenschaften führen zu dem folgenden einstufigen Axiomensystem Γ.

(1)
$$\forall x(x \preccurlyeq x)\,.$$
(2)
$$\forall x \forall y \forall z(x \preccurlyeq y \wedge y \preccurlyeq z \to x \preccurlyeq z)\,.$$
(3)
$$\forall x \forall y(x \preccurlyeq y \wedge y \preccurlyeq x \to x = y)\,.$$

In einer Menge mit einer zweistelligen Relation R gilt das Axiomensystem Γ genau dann, wenn die Relation eine Ordnungsrelation ist.

5.2. **Die Folgerungsbeziehung.**

Mit Axiomensystemen verbindet man die Vorstellung, dass daraus „wichtige“ weitere Eigenschaften beweisbar sind. In einer jeden Gruppe gelten nicht nur die Gruppenaxiome, sondern auch alle Gesetzmäßigkeiten, die man aus den Gruppenaxiomen folgern kann. Dies wird in der mathematischen Logik durch den Folgerungsbegriff präzisiert.

Definition 5.2. Es sei S ein Symbolalphabet erster Stufe, Γ eine Menge von S-Ausdrücken und p ein S-Ausdruck. Man sagt, dass p aus Γ *folgt*, geschrieben $\Gamma \vDash p$, wenn für jede S-Interpretation I mit $I \vDash \Gamma$ auch $I \vDash p$ gilt.

Die Folgerungsbeziehung verwendet also das gleiche Symbol wie die Gültigkeitsbeziehung. Dass aus einer gewissen Ausdrucksmenge Γ ein gewisser Audruck p folgt, erfordert eine mathematische Argumentation, die aufzeigt, dass eine Menge mit zusätzlichen Strukturen, die Γ erfüllt, stets auch p erfüllen muss.

Beispiel 5.3. In einer Gruppe ist das neutrale Element, das es aufgrund der Definition einer Gruppe geben muss, eindeutig bestimmt. Mathematisch wird dies so bewiesen: Sei e das neutrale Element der Gruppe, und sei e' ein weiteres Element, das ebenfalls die Eigenschaft des neutralen Elements erfüllt, d.h. es gilt $e'x = xe' = x$ für alle $x \in G$. Dann gilt einerseits $e' = e'e$, da e neutrales Element ist, und andererseits $e'e = e$, da auch e' neutrales Element ist. Also ist insgesamt $e' = e'e = e$ und e und e' stimmen überein.

Die Eindeutigkeit des neutralen Elementes kann man als den Ausdruck

$$p := \forall z(\forall x(zx = x \wedge xz = x) \to z = e)$$

ansetzen, und die obige mathematische Argumentation bedeutet, dass der Ausdruck p aus den Gruppenaxiomen Γ folgt, also die Folgerungsbeziehung

$$\Gamma \vDash p$$

vorliegt.

5.3. **Allgemeingültige Ausdrücke.**

Definition 5.4. Es sei S ein Symbolalphabet und p ein S-Ausdruck in der Prädikatenlogik erster Stufe. Man nennt p *allgemeingültig*, wenn er in jeder S-Interpretation I gilt, also $I \vDash p$ wahr ist.

Allgemeingültige Ausdrücke sind *Tautologien* im semantischen Sinn. Wir werden später noch Tautologien im syntaktischen Sinn kennenlernen und die Übereinstimmung der beiden Konzepte zeigen. Da ein allgemeingültiger Ausdruck p in jeder Interpretation gilt, kann man auch sagen, dass p aus der leeren Ausdrucksmenge folgt, also $\emptyset \vDash p$ gilt. Beispiele sind die Ausdrücke

$$\forall x \forall y \forall z((x = y \wedge y = z) \to x = z)$$

oder

$$(\forall x p) \to p$$

(wobei p ein Ausdruck ist). Wenn p_1, p_2, p_3 die Gruppenaxiome sind, und p die im obigen Beispiel erwähnte Eindeutigkeitsausssage für das neutrale Element ist, so ist auch

$$p_1 \wedge p_2 \wedge p_3 \to p$$

allgemeingültig.

Definition 5.5. Es sei S ein Symbolalphabet und es sei p ein S-Ausdruck in der Prädikatenlogik erster Stufe. Man nennt p *erfüllbar*, wenn es eine S-Interpretation I mit $I \vDash p$ gibt.

Für eine Ausdrucksmenge Γ bedeutet die Erfüllbarkeit, dass die darin enthaltenen Ausdrücke simultan in einer Interpretation erfüllbar sind. Zwischen Allgemeingültigkeit und Erfüllbarkeit besteht die Beziehung, dass p genau dann allgemeingültig ist, wenn die Negation $\neg p$ nicht erfüllbar ist.

Zwischen Folgerung und Erfüllbarkeit besteht der folgende Zusammenhang.

Lemma 5.6. *Es gilt $\Gamma \vDash p$ genau dann, wenn $\Gamma \cup \{\neg p\}$ nicht erfüllbar ist.*

Beweis. Siehe Aufgabe 5.4. □

5.4. **Das Koinzidenzlemma.**

Die folgende Aussage, das Koinzidenzlemma, zeigt, dass der Wert eines Terms und die Gültigkeit eines Ausdrucks unter einer Interpretation nur von den in dem Term bzw. Ausdruck vorkommenden freien Variablen abhängt. Ihr Beweis ist ein typisches Beispiel für einen Beweis durch Induktion über den Aufbau der Terme bzw. Ausdrücke.

Lemma 5.7. *Es sei S ein Symbolalphabet erster Stufe und $U \subseteq S$ eine Teilmenge. Es sei t ein U-Term und p ein U-Ausdruck. Es seien zwei S-Interpretationen I_1 und I_2 in einer gemeinsamen Grundmenge M gegeben, die auf U identisch seien. Dann gelten folgende Aussagen.*

(1) *Es ist $I_1(t) = I_2(t)$.*
(2) *Es ist $I_1 \models p$ genau dann, wenn $I_2 \models p$ (dazu genügt bereits, dass die Interpretationen auf den Symbolen aus U und auf den in p frei vorkommenden Variablen identisch sind).*

Beweis. (1). Wir führen Induktion über den Aufbau der Terme. Für den Induktionsanfang müssen wir Variablen und Konstanten aus U betrachten. Für eine Variable x (oder eine Konstante) aus U ist nach Voraussetzung $I_1(x) = I_2(x)$. Im Induktionsschritt können wir annehmen, dass ein n-stelliges Funktionssymbol f aus U gegeben ist und U-Terme $t_1, \ldots, t_n$, für die die Interpretationsgleichheit schon gezeigt wurde. Nach Voraussetzung wird f in beiden Interpretationen durch die gleiche Funktion f^M interpretiert. Daher ist

$$\begin{aligned} I_1(ft_1 \ldots t_n) &= f^M(I_1(t_1), \ldots, I_1(t_n)) \\ &= f^M(I_2(t_1), \ldots, I_2(t_n)) \\ &= I_2(ft_1 \ldots t_n). \end{aligned}$$

(2). Wir führen Induktion über den Aufbau der U-Ausdrücke, wobei die Aussage über je zwei Interpretationen zu verstehen ist. Für die Gleichheit und ein Relationssymbol R aus U folgt die Aussage unmittelbar aus (1), da ja R in beiden Interpretationen als die gleiche Relation zu interpretieren ist. Der Induktionsschritt ist für Ausdrücke der Form $\neg p$, $p \wedge q$, $p \to q$ aufgrund der Modellbeziehung unmittelbar klar. Sei nun ein U-Ausdruck der Form $\exists x p$ gegeben, und es gelte $I_1 \models \exists x p$. Dies bedeutet aufgrund der Modellbeziehung, dass es ein $m \in M$ gibt derart, dass $I_1 \frac{m}{x} \models p$ gilt. Die beiden umbelegten Interpretationen $I_1 \frac{m}{x}$ und $I_2 \frac{m}{x}$ stimmen auf den Symbolen aus U und den in p frei vorkommenden Variablen überein: die Variable x wird so oder so als m interpretiert und die anderen freien Variablen aus p sind auch in $\exists x p$ frei. Nach Induktionsvoraussetzung gilt $I_2 \frac{m}{x} \models p$ und daher wiederum $I_2 \models \exists x p$. □

5.5. **Substitution.**

Wir besprechen nun die Variablensubstitution, wobei wir weitgehend der Darstellung von Ebbinghaus, Flum, Thomas folgen.

Variablen repräsentieren verschiedene Werte, die man für sie einsetzen kann. Auf formaler Ebene bedeutet dies, dass eine oder mehrere Variablen durch gewisse Terme ersetzt werden. In der Ersetzung macht es einen großen Unterschied, ob gebundene oder freie Variablen vorliegen. Der Ausdruck

$$x \geq 0 \to \exists y(x = y \cdot y)$$

bedeutet in einem angeordneten Körper interpretiert, dass die nichtnegative Zahl x als Quadrat darstellbar ist (also eine Quadratwurzel besitzt), was für $\mathbb{R}$ wahr ist, für $\mathbb{Q}$ im Allgemeinen (das hängt von der Interpretation für x ab) nicht. Gleichbedeutend (bei einer inhaltlichen Interpretation) mit diesem Ausdruck ist

$$x \geq 0 \to \exists z(x = z \cdot z)\,,$$

aber nicht

$$x \geq 0 \to \exists x(x = x \cdot x)\,,$$

das nur bei $x = 0$ oder $x = 1$ wahr ist. Von daher wird die weiter unten zu gebende Definition für die Substitution von Ausdrücken berücksichtigen, ob Variablen frei oder gebunden sind. Ferner wird es wichtig sein, in einem Ausdruck neue Variablen einzuführen. Damit diese Konstruktion eindeutig definiert ist, legen wir eine durchnummerierte (und abzählbare) Variablenmenge $v_1, v_2, v_3 \ldots$ zugrunde.

Definition 5.8. Es sei ein Symbolalphabet S einer Sprache erster Stufe gegeben. Es seien $x_1, \ldots, x_k$ paarweise verschiedene Variablen und $t_1, \ldots, t_k$ fixierte S-Terme. Dann definiert man rekursiv über den Aufbau der Terme die Substitutionen $s\frac{t_1,\ldots,t_k}{x_1,\ldots,x_k}$ für jeden S-Term s.

(1) Für eine Variable x ist

$$x\frac{t_1, \ldots, t_k}{x_1, \ldots, x_k} = \begin{cases} x \text{ falls } x \neq x_i \text{ für alle } i \\ t_i \text{ falls } x = x_i\,. \end{cases}$$

(2) Für eine Konstante c ist

$$c\frac{t_1, \ldots, t_k}{x_1, \ldots, x_k} = c\,.$$

(3) Für ein n-stelliges Funktionssymbol f und n Terme $s_1, \ldots, s_n$ ist

$$fs_1 \ldots s_n\frac{t_1, \ldots, t_k}{x_1, \ldots, x_k} = fs_1\frac{t_1, \ldots, t_k}{x_1, \ldots, x_k} \ldots s_n\frac{t_1, \ldots, t_k}{x_1, \ldots, x_k}\,.$$

Definition 5.9. Es sei ein Symbolalphabet S einer Sprache erster Stufe gegeben. Es seien $x_1, \ldots, x_k$ paarweise verschiedene Variablen und $t_1, \ldots, t_k$ fixierte S-Terme. Dann definiert man rekursiv über den Aufbau der S-Ausdrücke die Substitutionen $p\frac{t_1,\ldots,t_k}{x_1,\ldots,x_k}$ für jeden S-Ausdruck p.

(1) Für Terme s_1, s_2 setzt man

$$(s_1 = s_2)\frac{t_1, \ldots, t_k}{x_1, \ldots, x_k} := s_1\frac{t_1, \ldots, t_k}{x_1, \ldots, x_k} = s_1\frac{t_1, \ldots, t_k}{x_1, \ldots, x_k}\,.$$

(2) Für eine n-stelliges Relationssymbol R und n Terme $s_1, \ldots, s_n$ setzt man

$$(Rs_1 \ldots s_n)\frac{t_1, \ldots, t_k}{x_1, \ldots, x_k} := Rs_1\frac{t_1, \ldots, t_k}{x_1, \ldots, x_k} \ldots s_n\frac{t_1, \ldots, t_k}{x_1, \ldots, x_k}\,.$$

(3) Für einen Ausdruck p setzt man

$$(\neg p)\frac{t_1, \ldots, t_k}{x_1, \ldots, x_k} := \neg p\frac{t_1, \ldots, t_k}{x_1, \ldots, x_k}\,.$$

(4) Für Ausdrücke p und q setzt man

$$(p \wedge q)\frac{t_1, \ldots, t_k}{x_1, \ldots, x_k} := p\frac{t_1, \ldots, t_k}{x_1, \ldots, x_k} \wedge q\frac{t_1, \ldots, t_k}{x_1, \ldots, x_k}$$

und ebenso für die anderen zweistelligen Junktoren.

(5) Für einen Ausdruck p seien $x_{i_1}, \ldots, x_{i_r}$ diejenigen Variablen (unter $x_1, \ldots, x_k$), die in $\forall xp$ frei vorkommen. Es sei $v = x$, falls x nicht in $t_{i_1}, \ldots, t_{i_r}$ vorkommt. Andernfalls sei v die erste Variable (in der fixierten Variablenaufzählung), die weder in p noch in $t_{i_1}, \ldots, t_{i_r}$ vorkommt. Dann setzt man

$$(\forall xp)\frac{t_1, \ldots, t_k}{x_1, \ldots, x_k} := \forall vp\frac{t_{i_1}, \ldots, t_{i_r}, v}{x_{i_1}, \ldots, x_{i_r}, x}$$

und ebenso für den Existenzquantor.

Die folgende Aussage, das Substitutionslemma, geben wir ohne Beweis. Es stiftet eine Beziehung zwischen Substitutionen und Uminterpretationen. In Verallgemeinerung der Schreibweise $I(\frac{m}{x})$ für eine Uminterpretation schreiben wir $I(\frac{m_1,\ldots,m_k}{x_1,\ldots,x_k})$ für die sukzessive Uminterpretation der untereinander verschiedenen Variablen $x_1, \ldots, x_k$ (dabei seien $m_1, \ldots, m_k$ Elemente der Grundmenge M der Interpretation). Es werden also die x_i als m_i interpretiert und alle anderen Variablen werden gemäß I interpretiert.

Lemma 5.10. *Es sei ein Symbolalphabet S einer Sprache erster Stufe gegeben und es seien $x_1, \ldots, x_k$ paarweise verschiedene Variablen und $t_1, \ldots, t_k$ fixierte S-Terme. Es sei eine S-Interpretation I gegeben. Dann gelten folgende Aussagen.*

(1) *Für jeden S-Term s gilt*

$$I(s\frac{t_1, \ldots, t_k}{x_1, \ldots, x_k}) = (I\frac{I(t_1), \ldots, I(t_k)}{x_1, \ldots, x_k})(s)\,.$$

(2) *Für jeden S-Ausdruck p gilt*

$$I \vDash p\frac{t_1, \ldots, t_k}{x_1, \ldots, x_k} \text{ genau dann, wenn } (I\frac{I(t_1), \ldots, I(t_k)}{x_1, \ldots, x_k}) \vDash p\,.$$

5.6. **Arbeitsblatt.**

Aufgabe 5.1. Axiomatisiere den Körperbegriff in einer geeigneten Sprache erster Stufe.

Eine Menge K heißt ein *Körper*, wenn es zwei Verknüpfungen (genannt Addition und Multiplikation)

$$+ : K \times K \longrightarrow K \text{ und } \cdot : K \times K \longrightarrow K$$

und zwei verschiedene Elemente $0, 1 \in K$ gibt, die die folgenden Eigenschaften erfüllen.

(1) Axiome der Addition
   (a) Assoziativgesetz: Für alle $a, b, c \in K$ gilt: $(a+b)+c = a+(b+c)$.
   (b) Kommutativgesetz: Für alle $a, b \in K$ gilt $a + b = b + a$.
   (c) 0 ist das neutrale Element der Addition, d.h. für alle $a \in K$ ist $a + 0 = a$.
   (d) Existenz des Negativen: Zu jedem $a \in K$ gibt es ein Element $b \in K$ mit $a + b = 0$.

(2) Axiome der Multiplikation
   (a) Assoziativgesetz: Für alle $a, b, c \in K$ gilt: $(a \cdot b) \cdot c = a \cdot (b \cdot c)$.
   (b) Kommutativgesetz: Für alle $a, b \in K$ gilt $a \cdot b = b \cdot a$.
   (c) 1 ist das neutrale Element der Multiplikation, d.h. für alle $a \in K$ ist $a \cdot 1 = a$.
   (d) Existenz des Inversen: Zu jedem $a \in K$ mit $a \neq 0$ gibt es ein Element $c \in K$ mit $a \cdot c = 1$.

(3) Distributivgesetz: Für alle $a, b, c \in K$ gilt $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$.

Aufgabe 5.2. Axiomatisiere den Begriff eines angeordneten Körpers in einer geeigneten Sprache erster Stufe.

Ein Körper K heißt *angeordnet*, wenn es eine totale Ordnung „$\geq$“ auf K gibt, die die beiden Eigenschaften

(1) Aus $a \geq b$ folgt $a + c \geq b + c$ (für beliebige $a, b, c \in K$)
(2) Aus $a \geq 0$ und $b \geq 0$ folgt $ab \geq 0$ (für beliebige $a, b \in K$)

erfüllt.

Aufgabe 5.3. Zeige, dass die folgenden prädikatenlogischen Ausdrücke allgemeingültig sind.

(1)
$$\forall x \forall y \forall z((x = y \wedge y = z) \rightarrow x = z)\,.$$

(2)
$$(\forall x p) \to p$$
(wobei p ein Ausdruck ist).

(3)
$$p_1 \wedge p_2 \wedge p_3 \to p\,,$$
wobei p_1, p_2, p_3 die Gruppenaxiome sind und
$$p := \forall z(\forall x(zx = x \wedge xz = x) \to z = e)$$
ist.

Aufgabe 5.4. Es sei Γ eine Ausdrucksmenge und p ein Ausdruck in einer Sprache erster Stufe. Zeige, dass $\Gamma \vDash p$ genau dann gilt, wenn $\Gamma \cup \{\neg p\}$ nicht erfüllbar ist.

6. Vorlesung

6.1. **Peano-Axiome.**

Giuseppe Peano (1858 -1932)

Wir besprechen nun, inwiefern man die natürlichen Zahlen $\mathbb{N}$ axiomatisieren kann, und was davon erststufig durchführbar ist. Dazu diskutieren wir die Peano-Axiome, wobei wir mit der zweistufigen Version beginnen.

Axiom 6.1. Eine Menge N mit einem ausgezeichneten Element $0 \in N$ (die *Null*) und einer (Nachfolger-)Abbildung
$$' : N \longrightarrow N,\ n \longmapsto n',$$
heißt *natürliche Zahlen* (oder *Peano-Modell* für die natürlichen Zahlen), wenn die folgenden *Peano-Axiome* erfüllt sind.

(1) Das Element 0 ist kein Nachfolger (die Null liegt also nicht im Bild der Nachfolgerabbildung).
(2) Jedes $n \in N$ ist Nachfolger höchstens eines Elementes (d.h. die Nachfolgerabbildung ist injektiv).
(3) Für jede Teilmenge $T \subseteq N$ gilt: Wenn die beiden Eigenschaften
   - $0 \in T$,
   - mit jedem Element $n \in T$ ist auch $n' \in T$,

   gelten, so ist $T = N$.

Mit zweitstufig ist gemeint, dass nicht nur über die Elemente der Menge $\mathbb{N}$, die man axiomatisch charakterisieren will, quantifiziert wird, sondern auch über beliebige Teilmengen dieser Menge. Mit dieser Axiomatik lassen sich ausgehend von der Nachfolgerfunktion die Addition und die Multiplikation rekursiv einführen, und es lässt sich zeigen, dass je zwei Modelle für diese zweistufigen Peano-Axiome „isomorph“ sind, dass es also zwischen ihnen eine strukturerhaltende Bijektion gibt. Das im Wesentlichen eindeutig bestimmte Modell für diese Arithmetik bezeichnen wir mit $\mathbb{N}$.

Wir betrachten zwei erststufige Varianten. Dabei wird die Nachfolgerfunktion beibehalten und das Induktionsaxiom, das oben für beliebige Teilmengen formuliert war, wird durch ein Induktionsaxiom für die in der Sprache formulierbaren Ausdrücke ersetzt. Das Induktionsaxiom gilt somit lediglich für Teilmengen, die in der gegebenen Sprache charakterisierbar sind.

Axiom 6.2. Die *Peano-Axiome für die Nachfolgerfunktion in der ersten Stufe* werden (in der Sprache L zur Symbolmenge mit einer Konstanten 0 und einem einstelligen Funktionssymbol N) folgendermaßen definiert.

(1) $\forall x(\neg(Nx = 0))$.
(2) $\forall x \forall y((Nx = Ny) \to (x = y))$.
(3) Für jeden Ausdruck p von L mit der einzigen freien Variablen x gilt

$$p\frac{0}{x} \wedge \forall x(p \to p\frac{Nx}{x}) \to \forall x p\,.$$

Aus der obigen zweitstufigen Formulierung der Axiomatik, die nur die Nachfolgerabbildung verwendet, kann man in jedem Modell in eindeutiger Weise eine Addition und eine Multiplikation definieren. Dafür ist das obige erststufige Axiomensystem zu schwach. Stattdessen werden wir unter der Peano-Arithmetik das folgende Axiomensystem verstehen, das mit zwei Konstanten 0 und 1 und zwei zweistelligen Operationen $+$ und $\cdot$ auskommt. Die Nachfolgerfunktion ist dann durch $Nx = x + 1$ definiert und es braucht dafür kein eigenes Funktionssymbol.

Axiom 6.3. Die *Peano-Axiome für Addition und Multiplikation in der ersten Stufe* werden (in der Sprache L^{Ar} zur Symbolmenge mit den beiden Konstanten 0 und 1 und zwei zweistelligen Funktionssymbolen $+$ und $\cdot$) folgendermaßen definiert.

(1) $\forall x(\neg(x+1=0))$.
(2) $\forall x\forall y((x+1=y+1)\to(x=y))$.
(3) $\forall x(x+0=x)$.
(4) $\forall x\forall y(x+(y+1)=(x+y)+1))$.
(5) $\forall x(x\cdot 0=0)$.
(6) $\forall x\forall y(x\cdot(y+1)=(x\cdot y)+x)$.
(7) Für jeden Ausdruck p von L^{Ar} mit der einzigen freien Variablen x gilt

$$p\frac{0}{x}\wedge\forall x(p\to p\frac{x+1}{x})\to\forall xp\,.$$

Die Axiome (1), (2) und (7) entsprechen dabei direkt den Nachfolgeraxiomen von oben. Die Axiome (3) und (4) spiegeln die Grundregeln in der zweistufigen Peano-Arithmetik für die rekursive Definition der Addition wider, und die Axiome (5) und (6) entsprechen den Grundregeln für die rekursive Definition der Multiplikation. Bekanntlich gelten diese Axiome für die natürlichen Zahlen. Anders als bei der obigen zweitstufigen Axiomatik gibt es aber von $\mathbb{N}$ verschiedene Modelle (nicht Standard-Arithmetiken), die die erststufige Peano-Arithmetik erfüllen. Dies ist aber kein „zufälliges“ Defizit der gewählten Axiomatik, sondern dahinter verbirgt sich eine grundsätzliche Schwäche der Sprache erster Stufe, die durch die Gödelschen Unvollständigkeitssätze präzisiert werden wird.

6.2. **Kalkül der Prädikatenlogik.**

Gegeben sei ein Symbolalphabet einer Sprache erster Stufe und damit die zugehörige Termmenge und die zugehörige Ausdrucksmenge. Wir möchten die logisch wahren Aussagen einer solchen Sprache syntaktisch charakterisieren. Mathematische Aussagen sind im Allgemeinen „wenn-dann“-Aussagen, d.h. sie behaupten, dass, wenn gewisse Voraussetzungen erfüllt sind, dann auch eine gewisse Folgerung erfüllt ist.

Wenn man einen Beweis eines Satzes der Gruppentheorie oder der elementaren Arithmetik entwirft, so sind dabei die Axiome der Gruppentheorie bzw. die Peano-Axiome stets präsent. Wenn p_1, p_2, p_3 die Gruppenaxiome bezeichnen und p die Aussage, dass das neutrale Element eindeutig bestimmt ist, bezeichnet, so folgt p aus p_1, p_2, p_3. Mit der Folgerungsbeziehung kann man dies als

$$\{p_1, p_2, p_3\}\vDash p$$

formulieren. Dies kann man auch so ausdrücken, dass

$$p_1\wedge p_2\wedge p_3\to p$$

allgemeingültig ist, also dass

$$\vDash p_1\wedge p_2\wedge p_3\to p$$

gilt. So kann man jede Folgerung $\Gamma \models p$ aus einer endlichen Ausdrucksmenge Γ „internalisieren“, also durch einen allgemeingültigen Ausdruck der Form

$$\models p_1 \wedge \ldots \wedge p_n \to p$$

wiedergegeben, wobei vorne die Ausdrücke aus Γ konjugiert werden. Die Folgerungsbeziehung (zumindest aus endlichen Ausdrucksmengen) kann also vollständig durch allgemeingültige Ausdrücke verstanden werden.

Wir besprechen nun die syntaktische Variante der allgemeingültigen Ausdrücke, nämlich die syntaktischen prädikatenlogischen Tautologien. Über den soeben besprochenen Zusammenhang ergibt sich daraus auch ein Ableitungskalkül, der das syntaktische Analogon zur Folgerungsbeziehung ist. Da wir Ausdrücke der Form $p_1 \wedge \ldots \wedge p_n \to p$ als Grundtyp für eine mathematische Aussage ansehen, arbeiten wir allein mit den Junktoren $\neg, \wedge, \to$ und lesen $\vee$ und $\leftrightarrow$ als Abkürzungen. Man könnte auch noch $\wedge$ bzw. $\to$ eliminieren und durch die verbleibenden beiden Junktoren ausdrücken, doch würde dies zu recht unleserlichen Formulierungen führen.

Der prädikatenlogische Kalkül, den wir vorstellen wollen, soll es erlauben, „alle“ prädikatenlogischen allgemeingültigen Ausdrücke formal abzuleiten. Der Aufbau dieses Kalküls geschieht wiederum rekursiv (und für beliebige Symbolalphabete gleichzeitig). D.h. man hat eine Reihe von Anfangstautologien (oder Grundtautologien) und gewisse Schlussregeln, um aus schon nachgewiesenen Tautologien neue zu produzieren. Sowohl die Anfangstautologien als auch die Schlussregeln sind aus der mathematischen Beweispraxis vertraut.

Zur Formulierung dieses Kalküls verwenden wir die Schreibweise

$$\vdash p\,.$$

Sie bedeutet, dass der Ausdruck p in der Prädikatenlogik (erster Stufe zu einem gegebenen Alphabet) ableitbar ist, also eine Tautologie (im syntaktischen Sinne) ist.

6.3. **Aussagenlogische Tautologien.**

In den folgenden aussagenlogischen Tautologien sind p und q beliebige Ausdrücke. Um Klammern zu sparen verwenden wir die Konvention, dass die Negation sich auf das folgende Zeichen bezieht und dass die Konjunktion stärker bindet als die Implikation.

Axiom 6.4. (1)

$$\vdash p \to p\,.$$

(2)

$$\vdash p \to (q \to p)\,.$$

(3)

$$\vdash (p \to q) \wedge (q \to r) \to (p \to r)\,.$$

(4)

$$\vdash p \wedge q \to p$$

und

$$\vdash p \wedge q \to q\,.$$

(5)

$$\vdash (p \to q) \wedge (p \to r) \to (p \to q \wedge r)\,.$$

(6)

$$\vdash (p \wedge q \to r) \to (p \to (q \to r))$$

und

$$\vdash (p \to (q \to r)) \to (p \wedge q \to r)\,.$$

(7)

$$\vdash p \wedge \neg p \to q\,.$$

(8)

$$\vdash (p \to q) \wedge (\neg p \to q) \to q\,.$$

Diese Tautologien sind also die Startglieder. Dabei stehen p, q, r für beliebige Ausdrücke der Prädikatenlogik erster Stufe (in der Aussagenlogik werden diese Tautologien einfach mit Aussagenvariablen formuliert). Das Axiom (3) besagt die Transitivität der Implikation, Axiom (7) heißt Widerspruchsaxiom und Axiom (8) heißt Fallunterscheidungsaxiom.

Um überhaupt aus diesen Axiomen weitere Tautologien generieren zu können, braucht man Ableitungsregeln. Davon gibt es lediglich eine.

Modus Ponens

Aus $\vdash p$ und $\vdash p \to q$ folgt $\vdash q$.

Wir wollen uns nicht lange an aussagenlogischen Tautologien aufhalten. Eine Durchsicht der angeführten Tautologien zeigt, dass es sich auch um semantische Tautologien, also allgemeingültige Ausdrücke, handelt.

Bemerkung 6.5. Die prädikatenlogischen Axiome der Form $\vdash \alpha \to \beta$ führen zu entsprechenden Schlussregeln, d.h. Vorschriften, wie man aus (schon etablierten) syntaktischen Tautologien neue Tautologien erhält. Wir gehen unter diesem Gesichtspunkt die Axiome durch.

Aus $\vdash p$ folgt $\vdash q \to p$.

Dies ergibt sich aus der Voraussetzung $\vdash p$ aus $\vdash p \to (q \to p)$ und dem Modus ponens.

Aus $\vdash p \wedge q$ folgt $\vdash p$ (und ebenso $\vdash q$).

Dies ergibt sich aus dem Axiom $\vdash p \wedge q \to p$ und der Voraussetzung $\vdash p \wedge q$ mittels Modus Ponens. Umgekehrt folgt aus $\vdash p$ und $\vdash q$ auch $\vdash p \wedge q$. Dies ergibt sich aus

$$\vdash p \to (q \to p \wedge q)$$

(was aus den Axiomen folgt, siehe Aufgabe 6.4) aus den Voraussetzungen durch eine zweifache Anwendung des Modus Ponens.

Aus $\vdash p \to q$ und $\vdash q \to r$ ergibt sich $\vdash p \to r$. Diese Regel heißt *Kettenschlussregel*. Nach der obigen abgeleiteten Konjunktionsregel folgt aus den Voraussetzungen direkt $\vdash (p \to q) \wedge (q \to r)$ und daraus mit dem Modus Ponens $\vdash p \to r$.

Lemma 6.6. *Es ist*

$$\vdash p \wedge q \to q \wedge p\,.$$

Beweis. Nach Axiom 6.4 ist

$$\vdash ((p \wedge q) \to q) \wedge ((p \wedge q) \to p) \to (p \wedge q \to q \wedge p)\,.$$

Die beiden Vordersätze gelten nach Axiom 6.4, so dass auch ihre Konjunktion ableitbar ist. Daher ist auch der Nachsatz ableitbar. □

Lemma 6.7. (1)

$$\vdash (p \to q) \to (p \wedge r \to q)\,.$$

(2)

$$\vdash (p \to q) \wedge (r \to s) \to (p \wedge r \to q \wedge s)\,.$$

Beweis. (1) Nach Axiom 6.4 ist

$$\vdash (p \wedge r \to p) \to ((p \to q) \to (p \wedge r \to q))\,.$$

Der Vordersatz ist nach Axiom 6.4 ableitbar, also auch der Nachsatz.

(2) Nach Teil (1) ist

$$\vdash (p \to q) \to (p \wedge r \to q)$$

und

$$\vdash (r \to s) \to (p \wedge r \to s)\,.$$

Daher gilt auch

$$\vdash (p \to q) \wedge (r \to s) \to (p \wedge r \to q)$$

und

$$\vdash (p \to q) \wedge (r \to s) \to (p \wedge r \to s)$$

bzw.

$$\vdash (p \to q) \wedge (r \to s) \wedge p \wedge r \to q$$

und

$$\vdash (p \to q) \wedge (r \to s) \wedge p \wedge r \to s\,.$$

Nach Axiom 6.4 ist mit der Abkürzung $\alpha = (p \to q) \wedge (r \to s) \wedge p \wedge r$

$$\vdash (\alpha \to q) \wedge (\alpha \to s) \to (\alpha \to q \wedge r)\,.$$

Da die beiden Vordersätze ableitbar sind, ist auch der Nachsatz ableitbar, was unter Verwendung von Axiom 6.4 zur Behauptung umformulierbar ist.

□

Die folgende Aussage gibt eine „interne Version“ des Modus Ponens, der ja nach Definition eine Schlussregel ist.

Lemma 6.8. *Es ist*

$$\vdash p \wedge (p \to q) \to q\,.$$

Beweis. Nach Axiom 6.4 ist

$$\vdash (p \to q) \wedge (\neg p \to q) \to q\,,$$

und Axiom 6.4 kann man wegen Axiom 6.4 zu

$$\vdash p \to (\neg p \to q)$$

umformulieren. Daraus ergibt sich mit Lemma 6.7

$$\vdash p \wedge (p \to q) \to (\neg p \to q) \wedge (p \to q)$$

und daraus durch den Kettenschluss die Behauptung. □

Lemma 6.9. (1) *Aus* $\vdash \alpha \to (\beta \to \gamma)$ *und* $\vdash \gamma \to \delta$ *folgt* $\vdash \alpha \to (\beta \to \delta)$.
(2) *Aus* $\vdash \alpha$ *und* $\vdash \alpha \wedge \beta \to \gamma$ *ergibt sich* $\vdash \beta \to \gamma$.

Beweis. (1) Sei

$$\vdash \alpha \to (\beta \to \gamma)$$

und

$$\vdash \gamma \to \delta\,.$$

Nach Bemerkung 6.5 gilt auch

$$\vdash \alpha \to (\gamma \to \delta)$$

und daraus ergibt sich mit Axiom 6.4, der Konjunktionsregel und dem Modus Ponens

$$\vdash \alpha \to (\beta \to \gamma) \wedge (\gamma \to \delta)\,.$$

Mittels des Kettenschlusses ergibt sich daraus und aus Axiom 6.4 die Behauptung.

(2) Siehe Aufgabe 6.6.

□

Die folgenden Tautologien machen wichtige Aussagen über das Negationszeichen. Die Tautologie (2) ist eine wichtige Variante der *Widerspruchstautologie* und die Tautologie (5) heißt *Kontraposition*.

Lemma 6.10. (1)

$$\vdash (\neg p \to p) \to p\,.$$

(2)

$$\vdash (\neg q \to \neg p) \wedge (\neg q \to p) \to q$$

(3)

$$\vdash p \to \neg\neg p\,.$$

(4)
$$\vdash \neg\neg p \to p\,.$$
(5)
$$\vdash (p \to q) \to (\neg q \to \neg p)\,.$$

Beweis. (1) Die Fallunterscheidungstautologie liefert
$$\vdash (p \to p) \wedge (\neg p \to p) \to p\,.$$
Aus
$$\vdash p \to p$$
ergibt sich daraus die Behauptung.

(2) Nach Axiom 6.4 gilt
$$\vdash (\neg q \to \neg p) \wedge (\neg q \to p) \to (\neg q \to \neg p \wedge p)$$
und nach Axiom 6.4 gilt
$$\vdash \neg p \wedge p \to q\,.$$
Nach Lemma 6.8 folgt
$$\vdash (\neg q \to \neg p) \wedge (\neg q \to p) \to (\neg q \to q)\,,$$
woraus nach Teil (1) die Behauptung folgt.

(3) Nach Axiom 6.4 ist
$$\vdash \neg\neg p \to (p \to \neg\neg p)$$
und nach Axiom 6.4 ist
$$\vdash \neg p \wedge p \to \neg\neg p\,,$$
was wir zu
$$\vdash \neg p \to (p \to \neg\neg p)\,,$$
umformulieren können. Daraus ergibt sich
$$\vdash p \to \neg\neg p$$
mit der Fallunterscheidungsregel.

(4) Nach Axiom 6.4 ist
$$\vdash p \to (\neg\neg p \to p)$$
und nach Axiom 6.4 ist
$$\vdash \neg p \wedge \neg\neg p \to p\,,$$
was wir zu
$$\vdash \neg p \to (\neg\neg p \to p)\,,$$
umformulieren können. Daraus ergibt sich
$$\vdash \neg\neg p \to p$$
mit der Fallunterscheidungsregel.

(5) Es ist

$$\vdash \neg p \to (\neg q \to \neg p)$$

und damit auch

$$\vdash \neg p \to ((p \to q) \to (\neg q \to \neg p))\,.$$

Ferner ist

$$\vdash q \to (\neg q \to \neg p)\,.$$

Nach Lemma 6.7 ist

$$\vdash p \to ((p \to q) \to q)\,,$$

woraus sich

$$\vdash p \to ((p \to q) \to (\neg q \to \neg p)$$

ergibt. Mit der Fallunterscheidungsregel folgt die Behauptung.

□

6.4. **Gleichheitstautologien.**

Es gelten die beiden folgenden Tautologien für die Gleichheit.

Axiom 6.11. Es sei S ein Symbolalphabet, s, t seien S-Terme und p sei ein S-Ausdruck. Dann sind die beiden folgenden Ausdrücke syntaktische Tautologien.

(1)

$$\vdash t = t\,.$$

(2)

$$\vdash s = t \wedge p\frac{s}{x} \to p\frac{t}{x}\,.$$

Diese beiden Axiome heißen *Gleichheitsaxiom* und *Substitutionsaxiom*.

In der folgenden Aussage wird ein wichtiger Begriff für eine syntaktische Tautologie, eine Ableitungsregel oder einen ganzen formalen Kalkül verwendet, den der *Korrektheit*. Er besagt, dass die Tautologie auch (im semantischen Sinn) allgemeingültig ist bzw. dass der Kalkül nur wahre Aussagen liefert. Die weiter oben axiomatisch formulierten aussagenlogischen Tautologien sind korrekt, d.h. sie (und auch jede weitere daraus mittels Modus Ponens ableitbare Tautologie) sind allgemeingültig, wie eine direkte Durchsicht zeigt. Die folgende Aussage zeigt, dass auch die eben postulierten Gleichheitsaxiome allgemeingültig sind und dass der Kalkül daher korrekt ist.

Lemma 6.12. *Die Gleichheitsaxiome sind korrekt.*

Beweis. Sei I eine beliebige S-Interpretation. (1). Aufgrund der Bedeutung des Gleichheitszeichens unter jeder Interpretation gilt

$$I(t) = I(t)\,,$$

also

$$I \vDash t = t\,.$$

(2). Es gelte

$$I \vDash s = t \wedge p\frac{s}{x}\,,$$

also $I \vDash s = t$ und $I \vDash p\frac{s}{x}$. Das bedeutet einerseits $I(s) = I(t)$. Andererseits gilt nach dem Substitutionslemma

$$I\frac{I(s)}{x} \vDash p\,.$$

Wegen der Termgleichheit gilt somit auch

$$I\frac{I(t)}{x} \vDash p$$

und daher, wiederum aufgrund des Substitutionslemmas, auch

$$I \vDash p\frac{t}{x}\,.$$

□

Korollar 6.13. *Aus den Gleichheitsaxiomen lassen sich folgende Gleichheitstautologien ableiten (dabei sind $r, s, t, s_1, \ldots, s_n, t_1, \ldots, t_n$ Terme, f ein n-stelliges Funktionssymbol und R ein n-stelliges Relationssymbol).*

(1)
$$\vdash s = t \to t = s\,.$$

(2)
$$\vdash r = s \wedge s = t \to r = t\,.$$

(3)
$$\vdash s_1 = t_1 \wedge \ldots \wedge s_n = t_n \to fs_1 \ldots s_n = ft_1 \ldots t_n\,.$$

(4)
$$\vdash s_1 = t_1 \wedge \ldots \wedge s_n = t_n \wedge Rs_1 \ldots s_n \to Rt_1 \ldots t_n\,.$$

Beweis. (1). Aufgrund der Gleichheitsaxiome haben wir

$$\vdash s = s$$

und

$$\vdash s = t \wedge (x = s)\frac{s}{x} \to (x = s)\frac{t}{x}\,,$$

wobei x eine Variable sei, die weder in s noch in t vorkomme. Daher sind die Substitutionen gleich $s = s$ bzw. $t = s$. Eine aussagenlogische Umstellung der zweiten Zeile ist

$$\vdash s = s \to (s = t \to t = s)\,,$$

so dass sich aus der ersten Zeile mittels Modus ponens

$$\vdash s = t \to t = s$$

ergibt. (2). Es sei wieder x eine Variable, die weder in r noch in s noch in t vorkomme. Eine Anwendung des Substitutionsaxioms liefert

$$\vdash s = t \wedge (r = x)\frac{s}{x} \to (r = x)\frac{t}{x}\,.$$

Nach Einsetzen und einer aussagenlogischen Umstellung unter Verwendung der Eigenschaft (1) ist dies die Behauptung. Für (3) und (4) siehe Aufgabe 6.7. □

6.5. **Arbeitsblatt.**

Aufgabe 5.1. Axiomatisiere den Körperbegriff in einer geeigneten Sprache erster Stufe.

Eine Menge K heißt ein *Körper*, wenn es zwei Verknüpfungen (genannt Addition und Multiplikation)

$$+ : K \times K \longrightarrow K \text{ und } \cdot : K \times K \longrightarrow K$$

und zwei verschiedene Elemente $0, 1 \in K$ gibt, die die folgenden Eigenschaften erfüllen.

(1) Axiome der Addition
   (a) Assoziativgesetz: Für alle $a, b, c \in K$ gilt: $(a+b)+c = a+(b+c)$.
   (b) Kommutativgesetz: Für alle $a, b \in K$ gilt $a + b = b + a$.
   (c) 0 ist das neutrale Element der Addition, d.h. für alle $a \in K$ ist $a + 0 = a$.
   (d) Existenz des Negativen: Zu jedem $a \in K$ gibt es ein Element $b \in K$ mit $a + b = 0$.

(2) Axiome der Multiplikation
   (a) Assoziativgesetz: Für alle $a, b, c \in K$ gilt: $(a \cdot b) \cdot c = a \cdot (b \cdot c)$.
   (b) Kommutativgesetz: Für alle $a, b \in K$ gilt $a \cdot b = b \cdot a$.
   (c) 1 ist das neutrale Element der Multiplikation, d.h. für alle $a \in K$ ist $a \cdot 1 = a$.
   (d) Existenz des Inversen: Zu jedem $a \in K$ mit $a \neq 0$ gibt es ein Element $c \in K$ mit $a \cdot c = 1$.

(3) Distributivgesetz: Für alle $a, b, c \in K$ gilt $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$.

Aufgabe 5.2. Axiomatisiere den Begriff eines angeordneten Körpers in einer geeigneten Sprache erster Stufe.

Ein Körper K heißt *angeordnet*, wenn es eine totale Ordnung „$\geq$“ auf K gibt, die die beiden Eigenschaften

(1) Aus $a \geq b$ folgt $a + c \geq b + c$ (für beliebige $a, b, c \in K$)
(2) Aus $a \geq 0$ und $b \geq 0$ folgt $ab \geq 0$ (für beliebige $a, b \in K$)

erfüllt.

Aufgabe 5.3. Zeige, dass die folgenden prädikatenlogischen Ausdrücke allgemeingültig sind.

(1)
$$\forall x \forall y \forall z((x = y \wedge y = z) \to x = z)\,.$$

(2)
$$(\forall x p) \to p$$
(wobei p ein Ausdruck ist).

(3)
$$p_1 \wedge p_2 \wedge p_3 \to p\,,$$
wobei p_1, p_2, p_3 die Gruppenaxiome sind und
$$p := \forall z(\forall x(zx = x \wedge xz = x) \to z = e)$$
ist.

Aufgabe 5.4. Es sei Γ eine Ausdrucksmenge und p ein Ausdruck in einer Sprache erster Stufe. Zeige, dass $\Gamma \vDash p$ genau dann gilt, wenn $\Gamma \cup \{\neg p\}$ nicht erfüllbar ist.

6. Vorlesung

7.1. **Quantorenaxiome und -regeln.**

Wir besprechen nun die Tautologien und Ableitungsregeln, die mit den Quantoren zusammenhängen. Wir arbeiten allein mit dem Existenzquantor und wir arbeiten nur mit nichtleeren Grundmengen. Letzteres ist Voraussetzung dafür, dass es überhaupt eine Variablenbelegung geben kann. Bei den jetzt einzuführenden Axiomen handelt es sich um eine Tautologie (genauer gesagt um ein Schema von Tautologien), nämlich die *Existenzeinführung im Sukzedens* und um eine Schlussregel, nämlich die *Existenzeinführung im Antezedens*. Für letztere ist die exakte Formulierung und der Korrektheitsnachweis nicht trivial.

Axiom 7.1. Es sei S ein Symbolalphabet erster Stufe, p ein S-Ausdruck, x eine Variable und t ein S-Term. Dann ist
$$\vdash p\frac{t}{x} \to \exists x p\,.$$

Diese Tautologie bedeutet inhaltlich gesprochen, dass ein Ausdruck, für den man einen erfüllenden Term gefunden hat, auf die entsprechende Existenzaussage schließen kann. Diese Tautologie ist allgemeingültig: Wenn in einer Interpretation I die Beziehung
$$I \vDash p\frac{t}{x}$$

gilt, so ist dies nach dem Substitutionslemma äquivalent zu

$$I\frac{I(t)}{x} \vDash p\,,$$

und das bedeutet wiederum

$$I \vDash \exists x p\,.$$

Einen wichtigen Spezialfall dieser Tautologie erhält man für $t = x$, nämlich

$$\vdash p \to \exists x p\,.$$

Für den Allquantor (den wir als Abkürzung verstehen) ergibt sich die entsprechende Tautologie

$$\vdash \forall x p \to p\frac{t}{x}\,.$$

Axiom 7.2. Es sei S ein Symbolalphabet erster Stufe, p und q seien S-Ausdrücke, x und y seien Variablen. Dann gilt die folgende Regel: Wenn

$$\vdash p\frac{y}{x} \to q$$

gilt und wenn y weder in $\exists x p$ noch in q frei vorkommt, so gilt auch

$$\vdash \exists x p \to q\,.$$

Ein Spezialfall dieser Ableitungsregel ist, dass man aus $\vdash p \to q$ unter der Bedingung, dass x nicht frei in q vorkommt, auf $\vdash \exists x p \to q$ schließen kann.

Die Allvariante dieser Schlussregel ist die *Alleinführung im Sukzedens*. Sie besagt, dass man aus

$$\vdash q \to p\frac{y}{x}$$

unter der Bedingung, dass y weder in $\forall x p$ noch in q frei vorkommt, auf

$$\vdash q \to \forall x p$$

schließen kann.

Die Existenzeinführung im Antezedens ist die einzige syntaktische Gesetzmäßigkeit, deren Korrektheit nicht unmittelbar klar ist.

Lemma 7.3. *Die Existenzeinführung im Antezedens ist eine korrekte Regel.*

Beweis. Es sei $p\frac{y}{x} \to q$ allgemeingültig, d.h.

$$I \vDash p\frac{y}{x} \to q$$

für jede S-Interpretation I. Wir müssen zeigen, dass dann auch $\exists x p \to q$ allgemeingültig ist. Sei dazu I eine Interpretation mit

$$I \vDash \exists x p\,.$$

Aufgrund der Modellbeziehung bedeutet dies, dass es ein $m \in M$ (aus der Grundmenge der Interpretation) gibt mit

$$I\frac{m}{x} \vDash p\,.$$

Die Variable y kommt nach Voraussetzung in $\exists x p$ nicht frei vor, d.h. bei $y \neq x$, dass y in p nicht frei vorkommt. Wir können daher das Koinzidenzlemma anwenden und erhalten

$$\left(I\frac{m}{x}\right)\frac{m}{y} \vDash p\,.$$

Diese Aussage gilt trivialerweise auch bei $x = y$. Damit gilt auch

$$\left(I\frac{m}{y}\right)\frac{m}{x} \vDash p\,.$$

Wir schreiben dies (etwas künstlich) als

$$\left(I\frac{m}{y}\right)\frac{\left(I\frac{m}{y}\right)(y)}{x} \vDash p\,.$$

Darauf können wir das Substitutionslemma (für die Interpretation $J = I\frac{m}{y}$ und den Term y) anwenden und erhalten

$$I\frac{m}{y} \vDash p\frac{y}{x}\,.$$

Wegen der vorausgesetzten Allgemeingültigkeit von $p\frac{y}{x} \to q$ folgt

$$I\frac{m}{y} \vDash q\,.$$

Da y in q nicht frei vorkommt, liefert das Koinzidenzlemma

$$I \vDash q\,.$$

□

Bemerkung 7.4. Die Variablenbedingung in der Existenzeinführung im Antezedenz ist wesentlich. Das zeigt am besten die Betrachtung $q = p$, wobei darin die Variable $x = y$ frei vorkommen möge (also z.B. $p = Rx$, wobei R ein einstelliges Relationssymbol sei). Dann ist natürlich

$$\vdash p \to p$$

richtig, und die Variablenbedingung an x bezogen auf diesen Ausdruck ist nicht erfüllt. Die Aussage

$$\exists x p \to p\,,$$

die man unter Missachtung dieser Variablenbedingung erhalten würde, ist keine Tautologie. Diese Ableitungsregel lässt sich also insbesondere nicht durch eine interne Tautologie ersetzen.

7.2. **Abgeleitete Regeln.**

Lemma 7.5. *Es sei S ein Symbolalphabet erster Stufe, p ein S-Ausdruck und x eine Variable Dann ist $\vdash p$ genau dann, wenn $\vdash \forall x p$ ist.*

Beweis. Nach der Allquantorversion von Axiom 7.1 ist

$$\vdash \forall x p \to p\frac{x}{x}\,,$$

also

$$\vdash \forall x p \to p\,.$$

Daher folgt aus

$$\vdash \forall x p$$

mittels Modus Ponens direkt

$$\vdash p\,.$$

Sei umgekehrt $\vdash p$ gegeben. Es sei q ein beliebiger Ausdruck, in dem x nicht vorkomme. Nach Axiom 6.4 und Modus Ponens ergibt sich

$$\vdash q \to p$$

und

$$\vdash \neg q \to p\,.$$

Auf diese beiden abgeleiteten Ausdrücke wird nun die Allquantorversion der Existenzeinführung im Antezedens (also die Alleinführung im Sukzedens) angewendet. Dies ist möglich, da x in q überhaupt nicht und in $\forall x p$ nicht frei vorkommt. Man erhält

$$\vdash q \to \forall x p$$

und

$$\vdash \neg q \to \forall x p\,.$$

Daraus ergibt sich mit der Fallunterscheidungsregel

$$\vdash \forall x p\,.$$

□

Diese Aussage bedeutet aber keineswegs, dass man den Allquantor überall weglassen oder hinzufügen könnte. Sie bedeutet lediglich, dass bei einem Ausdruck, der als Ganzes als eine Tautologie erwiesen ist, auch der entsprechende Allausdruck eine Tautologie ist und umgekehrt. Semantisch betrachtet beruht diese Äquivalenz darauf, dass die Allgemeingültigkeit von p bedeutet, dass bei einer beliebigen (Struktur- und) Variablenbelegung die entstehende Aussage ohne freie Variable wahr wird. Da ist also eine Allaussage schon miteingebunden.

Für den Existenzquantor gilt die entsprechende Äquivalenz nicht. Zwar ergibt sich aus $\vdash p$ direkt $\vdash \exists x p$ (und zwar unabhängig davon, ob x in p vorkommt oder nicht; die Allgemeingültigkeit beruht darauf, dass nur nichtleere Grundmengen betrachtet werden), aber nicht umgekehrt. Beispielsweise ist

$$\vdash \exists x(x = y)\,,$$

aber $x = y$ ist keine Tautologie.

Lemma 7.6. *Die folgenden Ausdrücke sind im Prädikatenkalkül ableitbar.*

(1)
$$\vdash \exists x \exists y p \to \exists y \exists x p \,.$$
(2)
$$\vdash \forall x p \wedge \forall x (p \to q) \to \forall x q \,.$$
(3)
$$\vdash \exists x p \wedge \forall x (p \to q) \to \exists x q \,.$$
(4)
$$\vdash \exists x (p \wedge q) \to \exists x p \wedge \exists x q \,.$$

Beweis. (1). Durch Existenzeinführung im Sukzedenz haben wir
$$\vdash p \to \exists x p$$
und
$$\vdash \exists x p \to \exists y \exists x p$$
und daraus
$$\vdash p \to \exists y \exists x p \,.$$
Dabei ist y hinten gebunden und somit kann man mit der Existenzeinführung im Antezedens auf
$$\vdash \exists y p \to \exists y \exists x p$$
schließen. Da auch x hinten gebunden ist, ergibt sich
$$\vdash \exists x \exists y p \to \exists y \exists x p \,.$$

(2). Aufgrund der Alleinführung im Antezedens ist
$$\vdash \forall x p \to p$$
und
$$\vdash \forall x (p \to q) \to (p \to q) \,.$$
Dies konjugiert ergibt
$$\vdash \forall x p \wedge \forall x (p \to q) \to p \wedge (p \to q) \,.$$
Ferner haben wir die aussagenlogische Tautologie
$$\vdash p \wedge (p \to q) \to q \,.$$
Damit ergibt sich aufgrund der Transitivität der Implikation die Ableitung
$$\vdash \forall x p \wedge \forall x (p \to q) \to q \,.$$
Da x vorne und in $\forall x q$ gebunden vorkommt, gilt nach der Alleinführung im Sukzedens auch
$$\vdash \forall x p \wedge \forall x (p \to q) \to \forall x q \,.$$

(3). Aufgrund der Alleinführung im Sukzedens ist
$$\vdash \forall x (p \to q) \to (p \to q) \,,$$
was wir als
$$\vdash p \wedge \forall x (p \to q) \to q$$

schreiben. Wegen $\vdash q \to \exists x q$ ist auch

$$\vdash p \wedge \forall x(p \to q) \to \exists x q\,,$$

was wir als

$$\vdash p \to (\forall x(p \to q) \to \exists x q)$$

schreiben. Im Sukzedens ist x gebunden, daher folgt aus der Existenzeinführung im Antezedens

$$\vdash \exists x p \to (\forall x(p \to q) \to \exists x q)\,,$$

was aussagenlogisch äquivalent zur Behauptung ist.

Zu (4) siehe Aufgabe 7.3. □

7.3. **Die Ableitungsbeziehung.**

Analog zur Folgerungsbeziehung definieren wir die Ableitungsbeziehung aus einer Ausdrucksmenge.

Definition 7.7. Es sei S ein Symbolalphabet, Γ eine Menge an S-Ausdrücken und p ein weiterer S-Ausdruck. Man sagt, dass p aus Γ *ableitbar* ist, geschrieben

$$\Gamma \vdash p\,,$$

wenn es endlich viele Ausdrücke $p_1, \ldots, p_n \in \Gamma$ gibt derart, dass

$$\vdash p_1 \wedge \ldots \wedge p_n \to p$$

gilt.

Man kann sich also wieder fragen, welche Ausdrücke aus einer vorgegebenen Ausdrucksmenge Γ, beispielsweise einem Axiomensystem einer Sprache erster Stufe, ableitbar sind. Unser „unbedingter“ Prädikatenkalkül, der die syntaktischen Tautologien generiert, führt zu einem entsprechenden Regelsatz für die Ableitbarkeit aus Γ. Dies ist näher an der mathematischen Praxis, da man sich dort in einem bestimmten mathematischen Kontext bewegt (z.B. der Gruppentheorie) und daher unter der Voraussetzung arbeitet, dass eine gewisse Ausdrucksmenge (z.B. die Gruppenaxiome) vorliegt, aus der heraus man etwas beweisen möchte.

7.4. **Der Vollständigkeitssatz.**

Im Laufe der Einführung des syntaktischen Prädikatenkalküls haben wir gesehen, dass die in ihm ableitbaren Ausdrücke allgemeingültig sind, dass also sämtliche durch den Prädikatenkalkül generierten formalen Tautologien auch semantische Tautologien sind. Daraus ergibt sich insbesondere, dass sich aus der Ableitbarkeitsbeziehung

$$\Gamma \vdash p$$

die Folgerungsbeziehung

$$\Gamma \vDash p$$

ergibt. Diese Aussage nennt man auch den *Korrektheitssatz*. Der entworfene Kalkül produziert also nur korrekte mathematische Aussagen.

Die Umkehrung ist deutlich schwieriger: Es geht um die Frage, ob der Kalkül jeden allgemeingültigen Ausdruck formal ableiten kann, ob es also für jeden mathematischen Beweis eines Ausdrucks einer Sprache erster Stufe auch einen formalen Beweis gibt. Es ist die Frage, ob der Kalkül *vollständig* ist. Dies ist in der Tat der Fall. Für diesen *Vollständigkeitssatz*, der auf Gödel zurückgeht, geben wir nur eine kurze Beweisidee.

Satz 7.8. *Es sei S ein Symbolalphabet, Γ eine Menge an S-Ausdrücken und p ein weiterer S-Ausdruck. Dann gilt $\Gamma \vDash p$ genau dann, wenn $\Gamma \vdash p$ gilt.*

Beweis. Die Implikation von rechts nach links, dass also ein aus Γ ableitbarer Ausdruck auch aus Γ folgt, beruht auf der Korrektheit des Prädikatenkalküls. Die umgekehrte Richtung wird durch Kontraposition bewiesen. Es sei also p ein Ausdruck, der nicht aus Γ ableitbar ist. Man muss dann zeigen, dass er auch nicht aus Γ folgt. D.h. man muss zeigen, dass es eine Interpretation I

(also insbesondere eine S-Stuktur) gibt, unter der Γ gilt, aber nicht p. Wegen der Unableitbarkeit kann man aus der Ausdrucksmenge $\Gamma \cup \{\neg p\}$ keinen Widerspruch ableiten. Daher muss man zu einer (syntaktisch) widerspruchsfreien Ausdrucksmenge ein erfüllendes Modell konstruieren. Die Grundidee dazu ist, auf der Menge der S-Terme eine Äquivalenzrelation unter Berücksichtigung der Ausdrucksmenge einzuführen und die resultierende Quotientenmenge als Grundmenge der Struktur zu nehmen. Dahinter stecken aber einige Feinheiten, die wir hier nicht ausführen. □

Das folgende Korollar, der sogenannte *Endlichkeitssatz*, demonstriert, dass der Vollständigkeitssatz keineswegs selbstverständlich ist. Es sei eine Folgerungsbeziehung $\Gamma \vDash p$ bewiesen, also gezeigt, dass jede Interpretation, die Γ erfüllt, auch p erfüllen muss. Dabei sei Γ unendlich, man denke etwa an ein unendliches Axiomenschema, wie es im Induktionsschema der einstufigen Peano-Arithmetik vorliegt. Ist es vorstellbar, dass in einem Beweis irgendwie auf all diese unendlich vielen Voraussetzungen Bezug genommen wird?

Korollar 7.9. *Es sei S ein Symbolalphabet, Γ eine Menge an S-Ausdrücken und p ein weiterer S-Ausdruck. Dann gilt $\Gamma \vDash p$ genau dann, wenn es eine endliche Teilmenge $\Gamma_e \subseteq \Gamma$ gibt mit $\Gamma_e \vDash p$.*

Beweis. Dies folgt direkt aus Satz 7.8, da die Endlichkeitsbeziehung für das Ableiten nach Definition gilt. □

7.5. **Arbeitsblatt.**

Aufgabe 7.1. Beweise aus der Existenzeinführung im Antezedens die *Alleinführung im Sukzedens*. Sie besagt, dass man aus

$$\vdash q \to p\frac{y}{x}$$

unter der Bedingung, dass y weder in $\forall xp$ noch in q frei vorkommt, auf

$$\vdash q \to \forall xp$$

schließen kann.

Aufgabe 7.2. Zeige

$$\vdash \exists x(x = y)\,.$$

Aufgabe 7.3. a) Zeige

$$\vdash \exists x(p \wedge q) \to \exists xp \wedge \exists xq\,.$$

b) Zeige, dass

$$\exists xp \wedge \exists xq \to \exists x(p \wedge q)$$

keine Tautologie ist.

Die beiden folgenden Aufgaben sind vermutlich mühselig.

Aufgabe 7.4. Man gebe einen formalen Beweis für die Aussage, dass die Hintereinanderschaltung von zwei surjektiven Abbildungen auf einer Menge wieder surjektiv ist.

Aufgabe 7.5. Man gebe einen formalen Beweis für die Aussage, dass die Hintereinanderschaltung von zwei injektiven Abbildungen auf einer Menge wieder injektiv ist.

Aufgabe 7.6. Es sei Γ eine Ausdrucksmenge aus einer Sprache erster Stufe und p ein weiterer Ausdruck. Es sei p nicht aus Γ ableitbar. Zeige, dass man aus $\Gamma \cup \{\neg p\}$ keinen Widerspruch (also keinen Ausdruck der Form $q \wedge \neg q$) ableiten kann.

Aufgabe 7.7. Es sei Γ eine Menge von S-Ausdrücken, die über beliebig großen endlichen Grundmengen erfüllbar ist. Zeige, dass Γ auch über einer unendlichen Menge erfüllbar ist.

Wegen der vorstehenden Aussage gibt es keinen Ausdruck, der genau in allen endlichen Grundmengen gilt. Dennoch kann man die (Un-)endlichkeit prädikatenlogisch charakterisieren.

Aufgabe 7.8. Zeige, dass es eine Ausdrucksmenge Γ gibt mit der Eigenschaft, dass für jede Interpretation I genau dann $I \vDash \Gamma$ gilt, wenn die Grundmenge der Interpretation unendlich ist.

8. Vorlesung

Wir kehren nun zur Ausgangsfrage dieses Kurses zurück, ob es eine Maschine geben kann, die mathematische Probleme (etwa aus der Zahlentheorie) löst. In den vorhergehenden Vorlesungen haben wir eine formale Sprache entwickelt, in der man solche nichttrivialen Probleme präzise formulieren kann. Ferner haben wir gesehen, wie ein formaler Beweis (eine Ableitung im Prädikatenkalkül) in dieser Sprache aussieht, und dass es nach dem Vollständigkeitssatz für jeden mathematisch beweisbaren Ausdruck der Sprache auch einen formalen Beweis gibt.

In dem vorgestellten Ableitungskalkül der Prädikatenlogik sind die Starttautologien und die Ableitungsregeln übersichtlich strukturiert. Zwar nehmen die Starttautologien häufig Bezug auf beliebige Ausdrücke (und Variablen) der Sprache, doch da die Ausdrücke prinzipiell auflistbar sind, gilt dies auch für die Starttautologien. Daher kann man sich auch einen Algorithmus vorstellen, der nach und nach alle formalen Beweise und somit auch alle formal-beweisbaren Ausdrücke ausgibt. Ein andersgelagertes Problem ist die Fragestellung, ob es ein Entscheidungsverfahren für die Prädikatenlogik gibt, ob es also ein algorithmisches Verfahren gibt, dass zu einem gegebenen Ausdruck überprüfen kann, ob es dafür einen formalen Beweis gibt oder nicht.

Wenn wir bisher von Algorithmen gesprochen haben, so haben wir dabei immer an intuitiv durchführbare Algorithmen gedacht, ohne ein konkretes Durchführungsmodell vor Augen zu haben. In dieser Vorlesung stellen wir die Arbeitsweise einer konkreten algorithmischen Maschine vor, der Registermaschine, die wir von nun an als mechanische Realisierung unserer intuitiven Vorstellung von Algorithmen auffassen wollen.

8.1. **Registermaschinen.**

Statue von Alan Turing (1912-1954).

Es gibt verschiedene Möglichkeiten, eine deterministisch arbeitende Maschine zu modellieren. Wir arbeiten hier mit Registermaschinen, da diese einem wirklichen Computer ziemlich nahe kommen und daher etwas vertrauter sind als Turingmaschinen oder rekursive Funktionen (wobei letztere vom mathematischen Standpunkt her eleganter sind).

Definition 8.1. Unter einer *Registermaschine* versteht man eine endliche Folge von Registern $R_1, R_2, \ldots, R_m$ (oder Speichern), deren Inhalt jeweils eine natürliche Zahl ist, die durch eine endliche (eventuell leere) Folge von Strichen repräsentiert wird.

Ein *Programm für eine Registermaschine* ist eine endliche durchnummerierte Folge von Befehlen $B_1, B_2, \ldots, B_h$, wobei es für die einzelnen Befehle B_j die folgenden Möglichkeiten gibt.

(1) $i+$ (erhöhe den Inhalt des Registers R_i um 1, d.h. um einen Strich).
(2) $i-$ (reduziere den Inhalt des Registers R_i um 1, d.h. ziehe einen Strich ab; wenn der Inhalt leer ist, so lasse ihn leer).
(3) $C(ij)$ (wenn der i-te Register leer ist, so gehe zum Befehl B_j, andernfalls zum nächsten Befehl).
(4) Drucke (drucke den Inhalt des ersten Registers).
(5) Halte an.

Dabei muss $i \leq m$ für alle in einer Programmzeile adressierten Register und $j \leq h$ für alle adressierten Befehlszeilen gelten. Die letzte Befehlszeile B_h ist ein Haltebefehl und sonst gibt es keinen Haltebefehl.

Die beiden ersten Befehle nennt man *Inkrementierung* bzw. *Dekrementierung*. Der dritte Befehl ist der *Abfragebefehl* oder die (bedingte) *Sprunganweisung*. Es folgen *Druckbefehl* und *Haltebefehl*.

Ein Programm für eine Registermaschine arbeitet die Befehle der Reihe nach ab und zwar unter den jeweiligen zum Bearbeitungszeitpunkt vorgefundenen Registerbelegungen. Wenn die aktuelle Programmzeile ein bedingter Sprungbefehl $C(ij)$ ist, so wird, falls die Bedingung zu diesem Zeitpunkt erfüllt ist (also falls das Register R_i leer ist), zur Programmzeile B_j gewechselt. Wenn die Endzeile B_h, also der Haltebefehl erreicht wird, so ist die Bearbeitung beendet.

Die Belegung (oder der Inhalt) des Registers R_i, die sich im Laufe des Programmdurchlaufs mehrfach ändern kann, werden wir häufig mit r_i bezeichnen. Dies ist stets eine natürliche Zahl. Wenn das Register R_i leer ist, so ist sein Inhalt $r_i = 0$.

Die Möglichkeiten einer Registermaschine scheinen auf den ersten Blick recht bescheiden zu sein. Man sieht aber recht schnell, dass man aus diesen wenigen Befehlen Programmabschnitte zusammensetzen kann, die zunehmend komplexere Befehle ausführen. Komplexe Befehle, von denen schon gezeigt wurde, dass sie sich mit Hilfe der Grundbefehle realisieren lassen, werden ohne weiteren Kommentar weiterverwendet.

Man sagt, dass ein Programm *korrekt* ist, wenn es das tut, was es tun soll. Wenn beispielsweise gesagt wird, dass ein Programm zwei Zahlen addiert, so wird die Korrektheit dadurch bewiesen, dass man eben durch Analyse des Programmcodes nachweist, dass bei beliebiger Belegung der beiden Register, deren Inhalte addiert werden sollen, das Programm schließlich anhält und in einem weiteren Register wirklich die Summe der beiden Zahlen gespeichert ist. Ein Korrektheitsnachweis ist häufig eine mühevolle Kleinarbeit mit aufwändigen Fallunterscheidungen, in den natürlich auch mathematische Überlegungen eingehen, wie z.B. bei der Addition die Eigenschaft, dass $s+t = s+(t-1)+1$ ist, was einen induktiven Korrektheitsbeweis ermöglicht. Wir werden diese Korrektheitsüberlegungen häufig abkürzen.

8.2. **Programmbeispiele.**

Wir beschreiben nun einige Programme bzw. Programmabschnitte für Registermaschinen. Wenn man Programme aus schon entwickelten Programmabschnitte zusammensetzt, so ändern sich natürlich die absoluten Befehlsnummern im Programm, was wir aber ignorieren werden.

Beispiel 8.2. Einen unbedingten Sprung (ein „Go to-Befehl“) zu einer bestimmten Programmzeile, der also nicht von einer Abfrage abhängt, kann man dadurch realisieren, dass man einen neuen Register R_k hinzunimmt, der von keiner anderen Programmzeile adressiert wird und dessen Inhalt auf 0 gesetzt wird. Dann bewirkt der Befehl $C(kj)$, dass zur j-ten Programmzeile gewechselt wird, da der Inhalt des Registers R_k im gesamten Programmverlauf gleich 0 bleibt.

Beispiel 8.3. Ein Programm soll sämtliche natürlichen Zahlen der Reihe nach ausdrucken. Dazu brauchen wir eine Registermaschine mit zwei Registern R_1 und R_2, die zum Start beide leer sind. Der zweite Register bleibt unverändert und wird nur für den unbedingten Sprungbefehl verwendet. Die Haltezeile wird nie erreicht.

(1) Drucke
(2) 1+
(3) Gehe zu 1
(4) Halte an

Beispiel 8.4. Das Register R_i soll geleert werden. Dies geschieht durch das folgende Programm.

(1) $C(i, 4)$
(2) $i-$
(3) Gehe zu 1
(4) Halte an

Bemerkung 8.5. Wir erlauben, dass bei einer Registermaschine die Anfangsbelegung der Register von außen festgelegt wird. Man könnte aber auch festlegen, dass die Anfangsbelegung stets die Nullbelegung ist, ohne die Berechnungsmöglichkeiten der Registermaschine einzuschränken. Dann kann man die eigentlich gewünschte Anfangsbelegung dadurch errreichen, dass man dem Programm einen „Belegungsprogramm“ voranstellt, das den einzelnen Registern R_i durch die s_i Befehle $i+, \ldots, i+$ die gewünschte Belegung s_i zuweist.

Man könnte auch erstmal ein „Entleerungsprogramm“ vorschalten, das alle Register leert und daran anschließend die Belegung durchführt, doch muss man für den Entleerungsvorgang, der nach Beispiel 8.4 einen unbedingten Sprungbefehl verwendet, zumindest ein leeres Register zur Verfügung haben.

Wenn der Registerinhalt r_i um eine natürliche Zahl k erhöht werden soll, also k-fach direkt hintereinander inkrementiert werden soll, so schreiben wir dafür auch $i + \ldots +$ mit k Pluszeichen.

Beispiel 8.6. Es soll mit einer Registermaschine festgestellt werden, ob der Inhalt r_i des Registers R_i größer oder gleich dem Inhalt r_j des Registers R_j ist. Dazu reserviert man den leeren Register R_k (i, j, k seien paarweise verschieden) und baut einen Programmabschnitt der folgenden Art.

(1) $C(j, 6)$
(2) $j-$
(3) $C(i, 7)$
(4) $i-$
(5) Gehe zu 1
(6) $k+$
(7) Halte an

Wenn dieser Programmabschnitt abgelaufen ist, so steht im Register R_k der Wert $r_k = 1$ oder $r_k = 0$, je nachdem, ob $r_i \geq r_j$ ist oder nicht. Die Korrektheit dieses Programms beruht darauf, dass $r \geq s$ genau dann gilt, wenn $r - 1 \geq s - 1$ ist. Dies ermöglicht einen Induktionsbeweis.

Beispiel 8.7. Wir wollen überprüfen, ob die Inhalte von zwei Registern R_i und R_j übereinstimmen. Dazu kann man das Programm aus Beispiel 8.5 einfach abändern zu

(1) $C(j, 6)$
(2) $j-$
(3) $C(i, 9)$
(4) $i-$
(5) Gehe zu 1
(6) $C(i, 8)$
(7) Gehe zu 9
(8) $k+$
(9) Halte an

Bei Gleichheit erhält man $r_k = 1$, bei Ungleichheit $r_k = 0$.

In den obigen beiden Beispielen wurde die Antwort im Register R_k (in der Form 0 oder 1 abgespeichert). Der Druckbefehl nimmt aber immer Bezug auf R_1. Daher ist es nötig, Registerinhalte in andere Register zu verschieben.

Beispiel 8.8. Wir wollen den Registerinhalt r_i des Registers R_i in den Register R_j übertragen (unabhängig von dessen Inhalt). Dies leistet das folgende Programm.

(1) Leere R_j
(2) $C(i, 6)$
(3) $i-$

(4) $j+$
(5) Gehe zu 2
(6) Halte an

Bei diesem Programm wird im Laufe der Durchführung der Ausgangsregister der Übertragung leer gemacht. Dies ist nicht immer erwünscht, häufig möchte man den Inhalt eines Registers kopieren und sich den Inhalt zugleich merken.

Beispiel 8.9. Wir wollen den Registerinhalt r_i des Registers R_i in den Register R_j übertragen (unabhängig von dessen Inhalt), ohne R_i zu leeren. Dazu brauchen wir einen dritten Register R_k und das folgende Programm.

(1) Leere R_j
(2) Leere R_k
(3) $C(i, 8)$
(4) $i-$
(5) $j+$
(6) $k+$
(7) Gehe zu 3
(8) Übertrage den Inhalt von R_k nach R_i
(9) Halte an

Hier wird zwar im Laufe des Programms der Inhalt von R_i verändert, zum Schluss wird der ursprüngliche Inhalt aber wieder hergestellt.

Beispiel 8.10. Die zwei Registerinhalte r_i (von R_i) und r_j (von R_j) sollen addiert werden, wobei die Summe zum Schluss in R_k stehen soll (es seien i, j, k paarweise verschieden). Dies leistet das folgende Programm.

(1) Leere R_k
(2) Übertrage r_i nach R_k
(3) $C(j, 7)$
(4) $j-$
(5) $k+$
(6) Gehe zu 3
(7) Halte an

Mit der Addition und der Kopie von Inhalten kann man auch den Inhalt eines Registers zu einem anderen Register dazuaddieren. Dies kann man natürlich auch einfach direkt realisieren.

Beispiel 8.11. Die zwei Registerinhalte r_i (von R_i) und r_j (von R_j) sollen multipliziert werden, wobei das Produkt zum Schluss in R_k stehen soll (es seien i, j, k paarweise verschieden). Dies leistet das folgende Programm mit dem Hilfsregister R_ℓ.

(1) Leere R_k
(2) Übertrage den Inhalt von R_i nach R_ℓ ohne R_i zu leeren

(3) $C(j, 7)$
(4) Addiere den Inhalt von R_ℓ zu R_k hinzu
(5) $j-$
(6) Gehe zu 2
(7) Halte an

Die Korrektheit dieses Programms beruht auf $r \cdot s = (r-1)s + s$; für das Produkt rs muss man r-mal s mit sich selbst addieren.

Beispiel 8.12. Es soll überprüft werden, ob der Registerinhalt r_t (von R_t) den Registerinhalt r_j (von R_j) teilt. Falls ja soll das Programm 1 ausgeben, andernfalls 0. Dies leistet das folgende Programm mit den Hilfsregistern R_k und R_ℓ (für Teilprogramme braucht man noch weitere Hilfsregister). Der Ausgaberegister R_1 soll zu Beginn leer sein.

(1) Leere R_ℓ
(2) Berechne $r_t \cdot r_\ell$ und schreibe das Ergebnis in R_k (ohne r_t, r_ℓ zu verändern)
(3) Bei $r_k > r_j$ gehe zu 8
(4) Bei $r_k = r_j$ gehe zu 7
(5) $\ell+$
(6) Gehe zu 2
(7) $1+$
(8) Drucke
(9) Halte an

Beispiel 8.13. Es soll überprüft werden, ob der Registerinhalt r_j (von R_j) eine Primzahl ist. Falls ja soll das Programm 1 ausgeben, andernfalls 0. Dies leistet das folgende Programm mit dem Hilfsregister R_t (für Teilprogramme braucht man noch weitere Hilfsregister). Das Ausgaberegister R_1 soll zu Beginn leer sein

(1) Leere R_t
(2) $t+$
(3) $t+$
(4) Wenn $r_t = r_j$, so gehe zu 8
(5) Wenn $r_t \geq r_j$, so gehe zu 9.[7]
(6) Wenn r_j von r_t geteilt wird, so gehe zu 9
(7) Gehe zu 3
(8) $1+$
(9) Drucke
(10) Halte an

Beispiel 8.14. Es sollen die geraden Zahlen ≥ 4 daraufhin überprüft werden, ob sie die Eigenschaft in der Goldbachvermutung erfüllen, also ob sie die Summe von zwei Primzahlen sind. Das Programm soll die Ausgabe 0 machen,

[7] Die Programmzeile (5) ist nur für $r_j = 0, 1$ von Bedeutung.

falls ein Gegenbeispiel gefunden wurde. Dies leistet das folgende Programm mit den Registern R_n, R_k und R_i, die alle zu Beginn auf 0 gesetzt seien. Auch der Ausgaberegister R_1 soll zu Beginn leer sein.

(1) $n++$
(2) $n++$
(3) Leere R_k
(4) $k+$
(5) $k++$
(6) Wenn $r_k \geq r_n$, so gehe zu 12
(7) Wenn r_k eine Primzahl ist, so gehe zu 9
(8) Gehe zu 5
(9) Berechne $r_n - r_k$, schreibe das Ergebnis in R_i
(10) Wenn r_i eine Primzahl ist, so gehe zu 2
(11) Gehe zu 5
(12) Drucke
(13) Halte an

8.3. **Arbeitsblatt.**

Aufgabe 8.1. Entwickle ein Entscheidungsverfahren für aussagenlogische Tautologien.

Aufgabe 8.2. Beschreibe ein Verfahren, das alle prädikatenlogischen Ausdrücke ausgibt (dabei sei vorausgesetzt, dass die Variablen, die Konstanten, die Relationssymbole und die Funktionssymbole in einer aufgezählten Form vorliegen).

Aufgabe 8.3. Zeige, dass es kein Programm für eine Registermaschine gibt, das bei jeder Anfangsbelegung sämtliche Register leert.

Aufgabe 8.4. Entwerfe ein Programm für eine Registermaschine, die für $r_i \geq r_j$ die Differenz $r_i - r_j$ von zwei Registerinhalten berechnet.

Aufgabe 8.5. Entwerfe ein Programm für eine Registermaschine, das die Potenz $r_i^{r_j}$ berechnet (und ausgibt), wobei r_i bzw. r_j die Registerinhalte der Register R_i, R_j, $i \neq j$, sind.

Aufgabe 8.6. Entwerfe ein Programm für eine Registermaschine, das entscheidet, ob der Registerinhalt r_i des Registers R_i die Potenz einer natürlichen Zahl ist.

Aufgabe 8.7. Entwerfe ein Programm für eine Registermaschine, das nach und nach alle Mersenne-Primzahlen ausdruckt.

9. Vorlesung

9.1. **Entscheidbarkeit und Berechenbarkeit.**

In der letzten Vorlesung haben wir verschiedene mathematische Operationen (wie Addition und Multiplikation) durch Registerprogramme berechnet und ebenso mathematische Prädikate (etwa das Prädikat, eine Primzahl zu sein) durch Registerprogramme charakterisiert. Die Fähigkeit eines Registerprogramms, bestimmte Funktionen bzw. Prädikate zu berechnen bzw. zu charakterisieren, führt zu den folgenden Begriffen.

Definition 9.1. Eine k-stellige Funktion

$$F : \mathbb{N}^k \longrightarrow \mathbb{N}$$

heißt *R-berechenbar* (oder *Register-berechenbar*), wenn es ein Programm P für eine Registermaschine gibt, die bei jeder Eingabe $(r_1, \ldots, r_k)$ (in den ersten k Registern) anhält und $F(r_1, \ldots, r_k)$ als (einzige) Ausgabe besitzt.

Definition 9.2. Es sei $T \subseteq \mathbb{N}$ eine Teilmenge der natürlichen Zahlen. Man sagt, dass diese Menge *R-entscheidbar* (oder *Register-entscheidbar*) ist, wenn es ein Programm P für eine Registermaschine gibt, die bei jeder Eingabe anhält und für die die Äquivalenz

$$n \in T \text{ genau dann, wenn } P(n) \text{ die Ausgabe } 0 \text{ besitzt}$$

gilt.

Eine Teilmenge $T \subseteq \mathbb{N}$ ist genau dann R-entscheidbar, wenn die zugehörige Indikatorfunktion R-berechenbar ist.

9.2. **Die Churchsche These.**

Wir haben in der letzten Vorlesung für einige recht einfache Aufgaben Registerprogramme angegeben, die diese Aufgabe lösen. Diese Beispiele vermitteln eine Vorstellung davon, was alles mit Registermaschinen berechnet werden kann. Zur Tragweite von algorithmischer Berechenbarkeit überhaupt ist die sogenannte *Churchsche These* von Bedeutung.

Bemerkung 9.3. Die *Churchsche These* (nach Alonzo Church, manchmal auch *Church-Turing-These*) behauptet, dass die intuitiv berechenbaren Funktionen (bzw. die intuitiv entscheidbaren Prädikate) mit den Registerberechenbaren Funktionen übereinstimmt. Da es sich bei „intuitiv berechenbar“ um einen nicht präzisen Begriff handelt, lässt sich diese These nicht beweisen. Sie ist dennoch weitgehend akzeptiert, wobei die folgenden Gründe angeführt werden.

Alle Präzisierungen des Berechenbarkeitsbegriffs, nämlich durch Registermaschine, Turingmaschine, primitiv-rekursive Funktionen, λ-Kalkül, führen zu einer übereinstimmenden Klasse von berechenbaren Funktionen. Dies beruht darauf, dass man die algorithmischen Verfahren wechselseitig simulieren kann.

Konkrete, intuitiv berechenbare Funktionen lassen sich stets durch ein Registerprogramm realisieren.

In der Praxis ist die Churchsche These vor allem eine Erleichterung, da man aufgrund eines häufig naheliegenden intuitiven Algorithmus auf die Existenz eines Registerprogramms schließen kann, und so die oft mühevolle „Programmier-Arbeit“ umgeht.

9.3. **Das Halteproblem.**

Nicht jedes Programm hält an. Ein einfaches Beispiel mit zwei Registern R_1, R_2 und leerer Belegung für R_2 ist

(1) $1+$
(2) $C(2,1)$
(3) Halte an

Im Allgemeinen wird es sehr schnell schwierig, zu einem gegebenen konkreten Programm zur Eingabe $r_1 = 0$ zu entscheiden, ob es den Haltebefehl schließlich erreicht oder nicht. Ebenso ist es schwierig zu entscheiden, für welche Eingabedaten in R_1 (den *Input*) das konkrete Programm stoppt.

Ein qualitativ anderes Problem ist allerdings die Frage, ob es ein Verfahren gibt, mit dem man für jedes Programm (bzw. jedes Programm und jede Eingabe) entscheiden kann, ob es anhält oder nicht.

Hier deutet sich eine selbstbezügliche Fragestellung an: Gibt es ein Programm, das Aussagen über alle Programme machen kann? Welche Aussage macht dann dieses Programm über sich selbst?

Um einen solchen Ansatz präzise machen zu können, müssen wir Programme als Eingabe für ein Programm interpretieren können. Das Programm einer Registermaschine erlaubt nur die Eingabe einer Zahl. Daher müssen wir ein Programm durch eine Zahl kodieren. Dies geschieht in zwei Schritten.

Zuerst führen wir für die erlaubten Befehle abkürzende Schreibweisen ein. Wir arbeiten mit dem Alphabet

$$\prime - \; I\, D\, C\, P\, H\,,$$

Die einzelnen Befehle werden folgendermaßen notiert

(1) Inkrementierung von R_i
$I\prime\prime\cdots\prime$ (mit i Strichen).
(2) Dekrementierung von R_i
$D\prime\prime\cdots\prime$ (mit i Strichen).

(3) Sprunganweisung $C(i,j)$
$C\prime\prime\cdots\prime,\prime\prime\cdots\prime$ (mit i Strichen vor dem Komma und j Strichen nach dem Komma).
(4) Druckanweisung: P.
(5) Halteanweisung: H.

Das Symbol $\prime$ wird also benutzt, um sowohl die Registernummern als auch die Zeilennummern (in der Sprunganweisung) auszudrücken. Da in jeder Befehlszeile eines konkreten Programmes konkrete Register bzw. Zeilen adressiert werden, stehen da jeweils natürliche Zahlen (keine Variablen), die problemlos durch eine Strichfolge ausgedrückt werden können.

Ein Programm, das aus den durchnummerierten Befehlszeilen $B_1, B_2, \ldots, B_h$ besteht, wird dann insgesamt durch die Zeichenfolge

$$b_1 - b_2 - \ldots - b_h$$

wiedergegeben, wobei die b_j die soeben angeführte Kodierung der j-ten Befehlszeile ist. Das Zeichen $-$ wird also verwendet, um die Zeilen voneinander zu trennen. Das Mitschleppen der Zeilennummern ist nicht nötig, da sich die Nummer aus der Reihenfolge rekonstruieren lässt.

Das oben angegebene Programm hätte demnach die symbolische Kodierung

$$I\prime - C\prime\prime,\prime - H$$

In einem zweiten Schritt ersetzen wir diese symbolische Kodierung durch eine numerische Kodierung. Dafür gibt es verschiedene Möglichkeiten. Da unser Alphabet, mit dem wir jedes Programm schreiben können, 8 Symbole verwendet, liegt eine Repräsentierung im Achtersystem nahe. Da die 0 als Anfangsnummer etwas problematisch ist, arbeiten wir lieber im Neunersystem (man kann die folgenden Zahlen genau so gut im Zehnersystem auffassen) und ordnen den Symbolen von oben in der obigen Reihenfolge die Ziffern

$$1, 2, 3, 4, 5, 6, 7, 8$$

zu. Das Programm von oben wird dann zur Ziffernfolge

$$3125118127\,.$$

Die einem jeden Programm P auf diese Weise zugeordnete Zahl (also der Zahlwert, nicht die Ziffernfolge) nennen wir $c(P)$. Man spricht von der *Gödelnummer* des Programms.

Bemerkung 9.4. Es ist algorithmisch überprüfbar, ob eine als Strichfolge gegebene natürliche Zahl ein Code für ein Registerprogramm ist. Dazu muss zuerst die Zahl in ihre Ziffernentwicklung (im Neunersystem) übersetzt werden. Da der Trennstrich, der die einzelnen Befehle trennt, durch eine bestimmte Ziffer codiert wird, muss die Ziffernfolge zwischen zwei Trennstrichziffern einen Befehl codieren. Die syntaktische Korrektheit dieser einzelnen Befehlsziffernfolgen muss dann der Reihe nach überprüft werden. Dazu muss

man für jeden der Einzelbefehle einen Algorithmus entwerfen. Wenn beispielsweise die Anfangsziffer einer Befehlsziffernfolge eine 3 (also ein I) ist, so muss es sich um einen Inkrementierungsbefehl handeln und alle nachfolgenden Ziffern (bis zum nächsten Trennstrich) müssen eine 1 sein.

Für ein Registerprogramm P und eine natürliche Zahl n verstehen wir unter $P(n)$ das Programm angesetzt auf n im ersten Register (und leeren anderen Registern).

Lemma 9.5. *Die Menge*

$$\{n \in \mathbb{N} |\, n \textit{ ist die Nummer eines Registerprogramms } P \textit{ und } P(n) \textit{ hält an}\}$$

ist nicht R-entscheidbar.

Beweis. Wir nehmen an, dass es ein Programm U gibt, das diese Menge entscheidet (der erste Teilaspekt, ob es sich überhaupt um ein valides Programm handelt, ist entscheidbar). Wir ändern dieses Programm ab zu einem Programm U', indem wir den letzten Befehl von U (also den Haltebefehl) durch den Programmabschnitt (mit der relativen Nummerierung und einem neuen Register R_i)

(1) $C(1,3)$
(2) Gehe zu 5
(3) $i+$
(4) $C(1,3)$
(5) Halte an

ersetzen. Dies bedeutet, dass das Programm U' genau dann in eine Endlosschleife hineinkommt und nicht anhält, wenn das Programm U die Ausgabe 0 hat. Daher gilt die Äquivalenz, dass ein Programm P bei Eingabe der eigenen Programmnummer $c(P)$ genau dann anhält, wenn U' bei Eingabe der Programmnummer $c(P)$ von P nicht anhält. Diese Äquivalenz ergibt bei Anwendung auf das Programm $P = U'$ einen Widerspruch. □

Satz 9.6. *Die Menge*

$$\{n \in \mathbb{N} |\, n \textit{ ist die Nummer eines Registerprogramms } P \textit{ und } P(0) \textit{ hält an}\}$$

ist nicht R-entscheidbar.

Beweis. Wir nehmen an, dass es ein Registerprogramm V gibt, dass die in Frage stehende Menge entscheidet, also stets anhält und angesetzt auf eine Zahl n genau dann die Ausgabe 0 liefert, wenn $n = c(P)$ für ein Programm P ist (also wenn n die Programmnummer eines Registerprogramms ist) und wenn dieses Programm P, angesetzt auf 0, anhält. Wir entwickeln aus V ein Programm U, das genau dann die Ausgabe 0 hat, wenn $n = c(P)$ für ein Programm P ist und wenn P, angesetzt auf n, anhält. Dies ergibt einen Widerspruch zu Lemma 9.5.

Dazu wird U folgendermaßen konstruiert: Wenn n keine Programmnummer ist, so hält das Programm U mit der Ausgabe 1 an (hier gibt es also keinen Unterschied zu V). Wenn $n = c(P)$ eine Programmnummer ist, so wird das Programm P' aufgestellt, das dem Programm P die n-fache Inkrementierung des ersten Registers voranstellt und dessen (in einem bedingten Sprungbefehl in einer Befehlszeile) adressierten Befehlszeilennummern sich um n erhöhen. Für die Programmnummer $n' = C(P')$ wird nun mittels V überprüft, welche Ausgabe P', angesetzt auf 0, besitzt. Aufgrund der Konstruktion von P' besitzt P' bei Eingabe 0 die Ausgabe 0 genau dann, wenn P bei Eingabe von n die Ausgabe 0 besitzt. □

9.4. **Aufzählbarkeit von Programmen.**

Wir führen einen weiteren Berechenbarkeitsbegriff ein.

Definition 9.7. Es sei $T \subseteq \mathbb{N}$ eine Teilmenge der natürlichen Zahlen. Man sagt, dass diese Menge *R-aufzählbar* (oder *Register-aufzählbar*) ist, wenn es ein Programm P für eine Registermaschine gibt, die bei Eingabe von 0 nach und nach genau die Zahlen aus T ausdruckt (dabei dürfen Zahlen aus T auch mehrfach ausgedruckt werden).

Zwischen Entscheidbarkeit und Aufzählbarkeit besteht der folgende Zusammenhang.

Lemma 9.8. *Es sei $T \subseteq \mathbb{N}$ eine Teilmenge von natürlichen Zahlen. Dann ist T genau dann R-entscheidbar, wenn sowohl T als auch das Komplement $\mathbb{N} \setminus T$ R-aufzählbar ist.*

Beweis. Wenn P ein Programm ist, das T entscheidet, so kann man einfach ein T aufzählendes Programm konstruieren. Man lässt der Reihe nach jede natürliche Zahl mittels P auf ihre Zugehörigkeit zu T überprüfen und druckt sie aus, falls sie dazu gehört (dazu muss man den Haltebefehl von P zu einer Druckausgabe modifizieren). Entsprechend konstruiert man ein Aufzählungsprogramm für das Komplement.

Es seien nun T als auch $\mathbb{N} \setminus T$ aufzählbar, und es seien P und Q Programme, die dies leisten. Dann liefert die folgende Kombination der beiden Programme ein Entscheidungsverfahren: Man schreibt die Programme P und Q hintereinander (wobei man natürlich die adressierten Register und Programmzeilen umnummerieren muss) und lässt sie abwechselnd bis zu einer Druckausgabe laufen. Sobald eine Druckausgabe eines Programmteils mit der zu überprüfenden Zahl n übereinstimmt, weiß man, ob n zu T gehört oder nicht. Da n entweder zu T oder zum Komplement gehört, muss einer dieser Fälle eintreten. □

Lemma 9.9. *Die Menge der Programmnummern von Register-Programmen, die angesetzt auf 0 anhalten, ist R-aufzählbar.*

Beweis. Die Idee für ein algorithmisches Aufzählverfahren geht so: Zu jeder natürlichen Zahl n berechnet man sämtliche Programme P mit $c(P) \leq n$. Jedes dieser Programme lässt man, angesetzt auf 0, n Schritte (also n Befehlszeilenwechsel) lang laufen. Wenn P anhält, so druckt man $c(P)$ aus. Wenn all diese Programme n Schritte gelaufen sind, so erhöht man auf $n+1$. □

Daraus folgt insbesondere, dass die nicht haltenden Programme nicht aufzählbar sind.

9.5. **Arbeitsblatt.**

Aufgabe 8.1. Entwickle ein Entscheidungsverfahren für aussagenlogische Tautologien.

Aufgabe 8.2. Beschreibe ein Verfahren, das alle prädikatenlogischen Ausdrücke ausgibt (dabei sei vorausgesetzt, dass die Variablen, die Konstanten, die Relationssymbole und die Funktionssymbole in einer aufgezählten Form vorliegen).

Aufgabe 8.3. Zeige, dass es kein Programm für eine Registermaschine gibt, das bei jeder Anfangsbelegung sämtliche Register leert.

Aufgabe 8.4. Entwerfe ein Programm für eine Registermaschine, die für $r_i \geq r_j$ die Differenz $r_i - r_j$ von zwei Registerinhalten berechnet.

Aufgabe 8.5. Entwerfe ein Programm für eine Registermaschine, das die Potenz $r_i^{r_j}$ berechnet (und ausgibt), wobei r_i bzw. r_j die Registerinhalte der Register R_i, R_j, $i \neq j$, sind.

Aufgabe 8.6. Entwerfe ein Programm für eine Registermaschine, das entscheidet, ob der Registerinhalt r_i des Registers R_i die Potenz einer natürlichen Zahl ist.

Aufgabe 8.7. Entwerfe ein Programm für eine Registermaschine, das nach und nach alle Mersenne-Primzahlen ausdruckt.

9. Vorlesung

10.1. **Arithmetische Repräsentierbarkeit.**

Definition 10.1. Eine Abbildung

$$\varphi : \mathbb{N}^r \longrightarrow \mathbb{N}^s$$

heißt *arithmetisch repräsentierbar*, wenn es einen L^{Ar}-Ausdruck ψ in $r + s$ freien Variablen gibt derart, dass für alle $(r + s)$-Tupel $(n_1, \ldots, n_{r+s}) \in \mathbb{N}^{r+s}$ die Äquivalenz $\varphi(n_1, \ldots, n_r) = (n_{r+1}, \ldots, n_{r+s})$ genau dann, wenn $\mathbb{N} \vDash \psi(n_1, \ldots, n_{r+s})$ gilt.

Definition 10.2. Eine Relation $R \subseteq \mathbb{N}^r$ heißt *arithmetisch repräsentierbar*, wenn es einen L^{Ar}-Ausdruck ψ in r freien Variablen gibt derart, dass für alle r-Tupel $(n_1, \ldots, n_r) \in \mathbb{N}^r$ die Äquivalenz $(n_1, \ldots, n_r) \in R$ genau dann, wenn $\mathbb{N} \vDash \psi(n_1, \ldots, n_r)$ gilt.

Da die repräsentierenden Ausdrücke genau $r+s$ bzw. r freie Variablen besitzen, entsteht durch Substitution der freien Variablen durch die Terme eine Aussage ohne freie Variablen. Diese sind bei Interpretation über den natürlichen Zahlen wahr oder falsch.

Wir wollen zeigen, dass Registerprogramme, oder besser gesagt die durch ein Registerprogramm festgelegte Abbildung, arithmetisch repräsentierbar sind.

10.2. **Registerprogramme als Abbildungen.**

Ein Registerprogramm P, das aus h Programmzeilen besteht und m Register anspricht, möchten wir als eine Abbildung auffassen. Die Wirkungsweise einer jeden Programmzeile hängt dabei nur von den Belegungen der Register zu dem Zeitpunkt ab, an dem diese Zeile aufgerufen wird. Sie ist geschichtsunabhängig, d.h. unabhängig von dem bisherigen Verlauf des Programmes. Man kann daher ein Programm vollständig durch die Abbildung

$$\varphi : \{1, 2, \ldots, h\} \times \mathbb{N}^m \longrightarrow \{1, 2, \ldots, h\} \times \mathbb{N}^m,\ (z, r_1, \ldots, r_m) \longmapsto \varphi(z, r_1, \ldots, r_m),$$

auffassen. Diese Abbildung nennen wir die *Programmabbildung*. Dabei steht z für die Programmzeilennummer und r_j steht für den Inhalt des Registers R_j (von denen es ja m Stück gibt). Dem Tupel $(z, r_1, \ldots, r_m)$ wird dasjenige Tupel $\varphi(z, r_1, \ldots, r_m)$ zugeordnet, das bei Abruf des in der z-ten Programmzeile stehenden Befehls B_z bei der Belegung $(r_1, \ldots, r_m)$ entsteht. Die Abbildung φ besteht dabei aus den $k+1$ Komponentenfunktionen $\varphi_0, \varphi_1, \ldots, \varphi_k$, wobei φ_0 die Wirkungsweise auf die Programmzeilennummer und die φ_j, $1 \leq j \leq k$, die Wirkungsweise auf dem j-ten Register beschreibt. Die Wirkung der einzelnen Befehle sieht folgendermaßen aus.

Bei $B_z = i+$ ist

$$\varphi(z, r_1, \ldots, r_m) = (z + 1, r_1, \ldots, r_{i-1}, r_i + 1, r_{i+1}, \ldots, r_m)\,.$$

Bei $B_z = i-$ ist

$$\varphi(z, r_1, \ldots, r_m) = (z+1, r_1, \ldots, r_{i-1}, r_i - 1, r_{i+1}, \ldots, r_m)$$

bei $r_i \geq 1$ und

$$\varphi(z, r_1, \ldots, r_m) = (z+1, r_1, \ldots, r_{i-1}, r_i, r_{i+1}, \ldots, r_m)$$

bei $r_i = 0$. Bei $B_z = C(ij)$ ist

$$\varphi(z, r_1, \ldots, r_m) = \begin{cases} (j, r_1, \ldots, r_m) \text{ falls } r_i = 0 \\ (z+1, r_1, \ldots, r_m) \text{ sonst}\,. \end{cases}$$

Bei $B_z = H$ (also bei $z = h$) ist

$$\varphi(h, r_1, \ldots, r_m) = (h, r_1, \ldots, r_m)\,,$$

die Abbildung wirkt dort also wie die Identität. Der Druckbefehl ist für den Programmablauf nicht relevant und wird hier ignoriert.

10.3. **Repräsentierbarkeit der Registerbefehle.**

Ein Registerprogramm kann also in eine Abbildung übersetzt werden, die die Wirkungsweise des Programms widerspiegelt. Die dabei auftretenden Abbildungen sind prinzipiell einfach beschreibbar, auch wenn dafür eine lange Abbildungsdefinition und tief verschachtelte Fallunterscheidungen nötig sind.

Der Ablauf eines Programms P zur Anfangseingabe $e = (1, r_1, \ldots, r_k)$ (die Anfangszeile besitzt die Zeilennummer 1!) wird durch die Hintereinanderschaltung der Programmabbildung $\varphi = \varphi_P$ beschrieben. Nach dem ersten Programmschritt, bei dem der Befehl in der ersten Programmzeile aufgerufen wird, erhält man die Folgekonfiguration $\varphi(e)$. Die nullte Komponente von $\varphi(e)$ gibt an, mit welcher Programmzeile weitergearbeitet wird. Dies ist aber alles in φ kodiert, so dass das Ergebnis nach dem nächsten Schritt einfach $\varphi(\varphi(e))$ ist. Das Ergebnis nach dem s-ten Rechenschritt ist also

$$\varphi(\ldots(\varphi(\varphi(e)))\ldots)\,,$$

wobei s-mal φ angewendet wird. Dafür schreiben wir auch $\varphi^s(e)$. Die aktuelle Zeilennummer ist dabei stets als nullte Komponente von $\varphi^s(e)$ ablesbar, wofür wir $(\varphi^s(e))_0$ schreiben.

Wie wirkt sich nun die Eigenschaft eines Programms, anzuhalten oder nicht, auf diese Iterationen von φ aus? Das Programm hält genau dann an, wenn es bei Eingabe von e ein s gibt mit

$$(\varphi^s(e))_0 = h\,.$$

Wir möchten die Wirkungsweise von Programmen in der Sprache der Arithmetik selbst repräsentieren, um dort das Halteproblem (und seine Unentscheidbarkeit) nachbilden zu können. Dafür müssen wir zunächst die einzelnen Programmschritte arithmetisch erfassen.

Lemma 10.3. *Die Programmzeilen eines Registerprogramms bzw. die zugehörige Programmabbildung lassen sich folgendermaßen mit den Variablen $z, r_1, \ldots, r_m, z', r'_1, \ldots, r'_m$ arithmetisch repräsentieren (dabei sei ℓ die aktuelle Programmzeile).*

(1) *$B_\ell = i+$ wird arithmetisiert durch*

$$A_\ell := (\ (z = \ell) \to (z' = z+1) \wedge (r'_1 = r_1) \wedge \ldots \wedge (r'_{i-1} = r_{i-1}) \wedge (r'_i = r_i + 1) \\ \wedge (r'_{i+1} = r_{i+1}) \wedge \ldots \wedge (r'_m = r_m)\ ) .$$

(2) *$B_\ell = i-$ wird arithmetisiert durch*

$$A_\ell := (\ (z = \ell) \to (z' = z+1) \wedge (r'_1 = r_1) \wedge \ldots \wedge (r'_{i-1} = r_{i-1}) \wedge ((r_i = 0) \to r'_i = r_i) \\ \wedge (\neg(r_i = 0) \to r'_i + 1 = r_i) \wedge (r'_{i+1} = r_{i+1}) \wedge \ldots \wedge (r'_m = r_m)\ ) .$$

(3) *$B_\ell = C(i,j)$ wird arithmetisiert durch*

$$A_\ell := (\ (z = \ell) \to ((r_i = 0 \to (z' = j)) \wedge (\neg(r_i = 0) \to (z' = z+1))) \\ \wedge (r'_1 = r_1) \wedge \ldots \wedge (r'_m = r_m)\ ) .$$

(4) *$B_\ell = B_h = H$ wird arithmetisiert durch*

$$A_h := ((z = h) \to (z' = h) \wedge (r'_1 = r_1) \wedge \ldots \wedge (r'_m = r_m)) .$$

Beweis. Die arithmetische Repräsentierbarkeit bedeutet, dass $\varphi_P(z, r_1, \ldots, r_m) = (z', r'_1, \ldots, r'_m)$ genau dann gilt, wenn die entsprechende arithmetische Aussage A_z in $\mathbb{N}$ gilt. Genau so wurden aber die A_z definiert. □

Zu einem gegebenen Programm bestehend aus den Programmzeilen $P_1, \ldots, P_h$ betrachtet man die Konjunktion der soeben eingeführten zugehörigen arithmetischen Repräsentierungen, also

$$A_P = A_1 \wedge A_2 \wedge \ldots \wedge A_h .$$

Die Aussage A_P ist somit für eine Variablenbelegung (der Variablen z, z', r_j, r'_j mit Werten in $\mathbb{N}$) genau dann gültig, wenn $\ell = z^{\mathbb{N}} > h$ ist (da dann keine Bedingung der einzelnen konjugierten Aussage erfüllt ist) oder wenn $\ell \leq h$ ist und A_ℓ erfüllt ist, und dies ist nach dem Lemma genau dann der Fall, wenn die (Variablen)-Belegung von z' die Programmzeilennummer ist, die durch den aktuellen (durch die Belegung von z festgelegten) Befehl B_ℓ als nächste Programmzeile aufgerufen wird, und wenn die Belegungen der r'_i die aus diesem Befehl resultierenden Belegungen der r_i sind.

10.4. **Die β-Funktion.**

Das Halteproblem führte zu der Existenzaussage, dass es eine Iteration der Programmabbildung gibt, für die die 0-te Komponente gleich der Haltezeilennummer ist. Die arithmetische Repräsentierung dieser Existenzaussage bedarf einiger Vorbereitungen.

Eine natürliche Zahl n lässt sich bekanntlich im Zehnersystem als

$$n = a_0 1 + a_1 10 + a_2 10^2 + \ldots + a_k 10^k$$

schreiben, wobei die a_i zwischen 0 und 9 liegen. Umgekehrt definiert eine endliche Ziffernfolge $(a_0, a_1, \ldots, a_k)$ (bzw. in alltäglicher Schreibweise $a_k a_{k-1} \ldots a_1 a_0$) eine natürliche Zahl. Anstatt der Basis 10 kann man jede natürliche Zahl $p \geq 2$ als Basis nehmen (für viele Zwecke ist auch die Basis 1 erlaubt, eine Zahl n wird dann einfach durch das n-fache Hintereinanderschreiben der 1 repräsentiert). Man spricht dann von der p-adischen Entwicklung (oder Darstellung) der Zahl. Die p-adische Entwicklung einer natürlichen Zahl ist unter der Voraussetzung, dass nur Ziffern zwischen 0 und $p-1$ verwendet werden, eindeutig.

Sei $p \geq 2$ fixiert. Wie berechnet man die Ziffernfolge einer gegebenen Zahl n? Zuerst betrachten wir die Ziffer $a_0(n) = a(n, 0)$. Es gilt die rekursive Beziehung

$$a(n,0) = \begin{cases} n, \text{falls } n < p \\ a(n-p, 0) \text{ sonst}\,. \end{cases}$$

Dies beruht einfach darauf, dass bei $n \geq p$ das Abziehen von p die Ziffer zu p^0 nicht ändert. Man beachte, dass sowohl die Abfrage, die die Fallunterscheidung in dieser Definition konstituiert, als auch die Subtraktion im Fall 2 mit einer Registermaschine durchführbar ist, und dass dadurch eine erlaubte rekursive Definition einer Funktion vorliegt.

Auch die Definition der anderen Ziffern geschieht rekursiv. Wenn man von n die Ziffer zu p^0 abzieht, so erhält man eine durch p teilbare Zahl. Zwischen der Zifferenentwicklung von n und von $m = (n - a(n,0))/p$ besteht ein direkter Zusammenhang, die Ziffer a_{i+1} von n ist einfach die Ziffer a_i von m. Daher ist für $i \geq 0$

$$a(n, i+1) = \begin{cases} 0, \text{falls } n < p^{i+1} \\ a((n - a(n,0))/p, i) \text{ sonst}\,. \end{cases}$$

Damit ist die Berechnung der $(i+1)$-ten Ziffer auf die Berechnung der i-ten Ziffer rekursiv zurückgeführt. Die Bedingung in der Abfrage und die Subtraktion und die Division in der Definition sind durch eine Registermaschine durchführbar. Diese Funktionsvorschrift berechnet nicht nur die „benötigten“ Ziffern, sondern auch alle höheren, wobei natürlich für alle unbenötigten 0 herauskommt.

Wir führen nun die β-Funktion ein. Hauptzweck einer solchen Funktion ist es, endliche Folgen von natürlichen Zahlen unterschiedlicher Länge durch drei Zahlen zu kodieren. Die Grundidee ist, dies über die p-adische Entwicklung zu tun, wobei die drei Eingabezahlen einen Zahlwert, eine Basis und eine Ziffernstelle repräsentieren, und die Ausgabe die Ziffernfolge ist. Zugleich soll diese Funktion arithmetisch repräsentierbar sein, so dass die folgende

Funktion etwas komplizierter aussieht. Wir folgen weitgehend dem Zugang von Ebbinghaus, Flum, Thomas.

Definition 10.4. Unter der *β-Funktion* versteht man die Abbildung

$$\mathbb{N}^3 \longrightarrow \mathbb{N},\ (p,n,i) \longmapsto \beta(p,n,i),$$

die folgendermaßen festgelegt ist. $\beta(p,n,i)$ ist die kleinste Zahl $a \in \mathbb{N}$, die die Bedingung erfüllt, dass es natürliche Zahlen b_0, b_1, b_2 gibt, die die folgenden Eigenschaften erfüllen:

(1) $n = b_0 + b_1((i+1) + ap + b_2p^2)$.
(2) $a < p$.
(3) $b_0 < b_1$.
(4) b_1 ist eine Quadratzahl.
(5) Alle Teiler $d \neq 1$ von b_1 sind ein Vielfaches von p.

Wenn kein solches a existiert, so ist $\beta(p,n,i) = 0$.

Zunächst ist klar, dass diese Funktion arithmetisch repräsentierbar ist. Wenn p eine Primzahl ist, so bedeutet Teil (5), dass b_1 eine Primzahlpotenz ist, und Teil (4), dass der Exponent geradzahlig ist. Das folgende Lemma sichert die gewünschte Eigenschaft der β-Funktion, nämlich die Eigenschaft, endliche Folgen zu repräsentieren.

Lemma 10.5. *Zu jeder endlichen Folge $(a_0, \ldots, a_s)$ aus $\mathbb{N}$ gibt es natürliche Zahlen p, n derart, dass $\beta(p,n,i) = a_i$ ist für $i \leq s$.*

Beweis. Es sei die endliche Folge $(a_0, a_1, \ldots, a_s)$ vorgegeben. Wir wählen eine Primzahl p, die größer als alle a_i und größer als $s+1$ ist. Es sei

$$\begin{aligned} n &:= 1 \cdot p^0 + a_0p^1 + 2p^2 + a_1p^3 + \ldots + (s+1)p^{2s} + a_sp^{2s+1} \\ &= \sum_{i=0}^{s} a_ip^{2i+1} + \sum_{i=0}^{s}(i+1)p^{2i} \\ &= \sum_{i=0}^{s}(i+1+a_ip)p^{2i}. \end{aligned}$$

Die vorgegebene Folge ist also die Folge der Ziffern der ungeraden Stellen in der p-adischen Ziffernentwicklung von n. Wir behaupten $\beta(p,n,k) = a_k$ für $k \leq s$. Zunächst erfüllt a_k die in der Definition der β-Funktion formulierten Eigenschaften, und zwar mit

$$b_0 = 1p^0 + a_0p^1 + 2p^2 + a_1p^3 + \ldots + kp^{2k-2} + a_{k-1}p^{2k-1}\,,$$

$$b_1 = p^{2k}\,,$$

$$b_2 = (k+2) + a_{k+1}p + (k+3)p^2 + \ldots + (s+1)p^{2(s-k)-2} + a_sp^{2(s-k)-1}\,.$$

Die erste Eigenschaft ergibt sich aus

$$\begin{aligned} n &= \sum_{i=0}^{s} a_i p^{2i+1} + \sum_{i=0}^{s} (i+1)p^{2i} \\ &= \sum_{i=0}^{k-1} a_i p^{2i+1} + \sum_{i=0}^{k-1} (i+1)p^{2i} + \sum_{i=k}^{s} a_i p^{2i+1} + \sum_{i=k}^{s} (i+1)p^{2i} \\ &= b_0 + p^{2k}(\sum_{i=0}^{s-k} a_{k+i} p^{2i+1} + \sum_{i=0}^{s-k} (k+i+1)p^{2i}) \\ &= b_0 + p^{2k}(k+1+a_k p + \sum_{i=1}^{s-k} a_{k+i} p^{2i+1} + \sum_{i=1}^{s-k} (k+i+1)p^{2i}) \\ &= b_0 + b_1(k+1+a_k p + b_2 p^2), \end{aligned}$$

die anderen sind klar. Wenn umgekehrt ein a die Bedingungen erfüllt (mit c_0, c_1, c_2), wobei $c_i = p^{2\ell}$ ist, so ist

$$\begin{aligned} n &= b_0 + (k+1)p^{2k} + a_k p^{2k+1} + b_2 p^{2k+2} \\ &= c_0 + (k+1)p^{2\ell} + a p^{2\ell+1} + c_2 p^{2\ell+2}. \end{aligned}$$

Da die p-adische Entwicklung von n eindeutig ist, folgen daraus und aus den weiteren Bedingungen die Gleichheiten $\ell = k$ und $a = a_k$. □

10.5. **Arbeitsblatt.**

Aufgabe 10.1. Zeige, dass die folgenden Teilmengen T der natürlichen Zahlen arithmetisch repräsentierbar sind.

(1) Eine konkrete endliche Menge $\{n_1, \ldots, n_k\}$.
(2) Die Menge aller Vielfachen von 5.
(3) Die Menge der Primzahlen.
(4) Die Menge der Quadratzahlen.
(5) Die Menge der Zahlen, in deren Primfaktorzerlegung jeder Exponent maximal 1 ist.

Aufgabe 10.2. Zeige, dass die folgenden Abbildungen $\varphi : \mathbb{N}^r \to \mathbb{N}$ arithmetisch repräsentierbar sind.

(1) Die Addition
$$\mathbb{N}^2 \longrightarrow \mathbb{N},\ (x,y) \longmapsto x+y.$$
(2) Die Multiplikation
$$\mathbb{N}^2 \longrightarrow \mathbb{N},\ (x,y) \longmapsto x \cdot y.$$
(3) Die eingeschränkte Subtraktion
$$\mathbb{N}^2 \longrightarrow \mathbb{N},\ (x,y) \longmapsto \max(x-y, 0),$$
die bei $y > x$ den Wert 0 besitzt.

(4) Die Restfunktion

$$\mathbb{N}^2 \longrightarrow \mathbb{N},\ (n,t) \longmapsto r(n,t),$$

die den Rest (zwischen 0 und $t-1$) bei Division von n durch t angibt.

Aufgabe 10.3. Es sei

$$\varphi : \mathbb{N}^r \longrightarrow \mathbb{N}^s$$

eine Abbildung. Zeige, dass φ genau dann arithmetisch repräsentierbar ist, wenn sämtliche Komponentenfunktionen φ_i, $1 \leq i \leq s$, arithmetisch repräsentierbar sind.

Aufgabe 10.4. Zeige explizit, dass die in Vorlesung 8 besprochenen Registerprogramme (also ihre zugehörigen Programmabbildungen) arithmetisch repräsentierbar sind.

Aufgabe 10.5. Zeige, dass die β-Funktion arithmetisch repräsentierbar ist.

11. Vorlesung

11.1. **Repräsentierbarkeit der Halteeigenschaft.**

Ein Durchlauf eines Registerprogramms P bis zum Rechenschritt t wird am einfachsten dokumentiert durch die Folge der Programmkonfigurationen K_s, $1 \leq s \leq t$, wobei jede Programmkonfiguration K_s aus der Nummer der im Rechenschritt s abzuarbeitenden Programmzeile und der Folge der Registerinhalte (zu diesem Zeitpunkt) besteht. Wenn man diese Konfigurationen einfach hintereinander schreibt, so erhält man eine Folge von $t(m+1)$ Zahlen. Wenn umgekehrt eine solche Zahlenfolge gegeben ist, so kann man einfach überprüfen, ob sie den Durchlauf eines Programms bis zum Schritt t korrekt dokumentiert. Man muss sicher stellen, dass sich jeder Abschnitt $(s+1)(m+1)+1, \ldots, (s+1)(m+1)+m+1$ aus dem Vorgängerabschnitt $s(m+1)+1, \ldots, s(m+1)+m+1$ ergibt, wenn die Programmzeile $s(m+1)+1$ angewendet wird.

Lemma 11.1. *Für ein Programm P für eine Registermaschine gibt es einen arithmetischen Ausdruck ψ_P, der genau dann (bei der Standardinterpretation in den natürlichen Zahlen) gilt, wenn das Programm anhält. Genauer gesagt: Wenn das Programm h Programmzeilen besitzt und m Register verwendet, so gibt es einen arithmetischen Ausdruck ψ_P in $2m$ freien Variablen derart, dass*

$$\mathbb{N} \vDash \psi_P(e_1, \ldots, e_m, a_1, \ldots, a_m)$$

genau dann gilt, wenn das Programm bei Eingabe von $(1, e_1, \ldots, e_m)$ nach endlich vielen Schritten bei der Konfiguration $(h, a_1, \ldots, a_m)$ anlangt (und insbesondere anhält).

Beweis. Zur Notationsvereinfachung schreiben wir r_0 statt z und r'_0 statt z'. Es sei ϑ der Ausdruck (in vier freien Variablen), der die β-Funktion arithmetisch repräsentiert. Der Ausdruck

$$\vartheta(p, n, i, r)$$

ist also genau dann wahr in $\mathbb{N}$, wenn $\beta(p, n, i) = r$ ist. Diese Beziehung verwenden wir für $i = s(m + 1) + j$ (bzw. $i = (s + 1)(m + 1) + j$) und $r = r_j$ (bzw. $r = r'_j$) und $j = 0, \ldots, m$. Daher besagt der Ausdruck (bei Interpretation in $\mathbb{N}$)

$$\begin{aligned} T(p, n, s) := \vartheta(p, n, s(m + 1), r_0) \wedge \ldots \wedge \vartheta(p, n, s(m + 1) + m, r_m) \\ \wedge \vartheta(p, n, (s + 1)(m + 1), r'_0) \wedge \ldots \wedge \vartheta(p, n, (s + 1)(m + 1) + m, r'_m), \end{aligned}$$

dass die β-Funktion $\beta(p, n, -)$ für die $m + 1$ aufeinander folgenden Zahlen (eingesetzt in die dritte Komponente der β-Funktion) $s(m + 1), s(m + 1) + 1, \ldots, s(m+1)+m$ gleich $r_0, r_1, \ldots, r_m$ und für die $m+1$ aufeinander folgenden Zahlen $(s + 1)(m + 1), (s + 1)(m + 1) + 1, \ldots, (s + 1)(m + 1) + m$ gleich $r'_0, r'_1, \ldots, r'_m$ ist. Hierbei sind p, n, s und die r_j, r'_j Variablen und m ist eine (vom Programm abhängige) Zahl. An der mit $s(m + 1) + j$ bezeichneten Stelle steht die $(m + 1)$-fache Addition der Variablen s mit sich selbst plus die j-fache Addition der 1.

Mit diesem Ausdruck soll der Konfigurationsübergang beim s-ten Rechenschritt beschrieben werden. Da man die Registerbelegung beim s-ten Rechenschritt nicht von vornherein kennt, muss man den Übergang mit Allquantoren ansetzen. Der Ausdruck

$$E(p, n, s) := \forall r_0 \forall r_1 \ldots \forall r_m \forall r'_0 \forall r'_1 \ldots \forall r'_m (T(p, n, s) \to A_P)$$

besagt (A_P repräsentiere das Programm), dass wenn $(r_0, r_1, \ldots, r_m)$ die Konfiguration beim s-ten Rechenschritt und $(r'_0, r'_1, \ldots, r'_m)$ die Konfiguration beim $(s + 1)$-ten Rechenschritt beschreibt, dass dann dieser Konfigurationsübergang durch das Programm bewirkt wird.

In analoger Weise besagt der Ausdruck

$$D(p, n)(x_1, \ldots, x_m) := \vartheta(p, n, 0, 1) \wedge \vartheta(p, n, 1, x_1) \wedge \ldots \wedge \vartheta(p, n, m, x_m)\,,$$

dass $\beta(p, n, 0) = 1$ und $\beta(p, n, j) = x_j$ für $j = 1, \ldots, m$ ist, und der Ausdruck

$$\begin{aligned} F(p, n, t)(y_1, \ldots, y_m) := \vartheta(p, n, t(m + 1), h) \wedge \vartheta(p, n, t(m + 1) + 1, y_1) \\ \wedge \ldots \wedge \vartheta(p, n, t(m + 1) + m, y_m), \end{aligned}$$

besagt, dass $\beta(p, n, t(m + 1)) = h$ und $\beta(p, n, t(m + 1) + j) = y_j$ für $j = 1, \ldots, m$ ist.

Somit besagt der Ausdruck

$$\begin{aligned} \exists p \exists n \exists t\,(\; D(p, n)(x_1, \ldots, x_m) \wedge \forall s(1 \leq s < t \to E(p, n, s)) \\ \wedge F(p, n, t)(y_1, \ldots, y_m)\;)\,, \end{aligned}$$

dass das Programm mit der Startkonfiguration $(1, x_1, \ldots, x_m)$ anhält und dabei die Konfiguration $(h, y_1, \ldots, y_m)$ erreicht. □

11.2. **Die Unentscheidbarkeit der Arithmetik.**

Die Idee des folgenden Beweises beruht darauf, dass man, wie wir in der letzten Vorlesung gezeigt haben, die Arbeitsweise von Registerprogrammen mit arithmetischen Ausdrücken repräsentieren und damit die Unentscheidbarkeit des Halteproblems arithmetisch modellieren kann.

Satz 11.2. *Die Menge der wahren arithmetischen Ausdrücke (ohne freie Variablen) ist nicht R-entscheidbar. D.h. es gibt kein R-Entscheidungsverfahren, mit dem man von einem beliebigen vorgegebenen Ausdruck $p \in L_0^{\mathrm{Ar}}$ der arithmetischen Sprache bestimmen kann, ob er wahr oder falsch ist.*

Beweis. Nach Lemma 11.1 gibt es zu jedem Programm P (mit h Befehlen und m Registern) einen arithmetischen Ausdruck ψ_P in $2m$ freien Variablen $x_1, \ldots, x_m, y_1, \ldots, y_m$, der bei Belegung mit $e_1, \ldots, e_m, a_1, \ldots, a_m$ genau dann wahr ist, wenn das Programm, angesetzt auf $(1, e_1, \ldots, e_m)$, schließlich mit der Konfiguration $(h, a_1, \ldots, a_m)$ anhält. Der Ausdruck

$$\varphi_P = \psi_P(0, 0, \ldots, 0, y_1, \ldots, y_m)$$

besagt daher, dass das Programm bei Nulleingabe mit der Registerbelegung $(y_1, \ldots, y_m)$ anhält und der Ausdruck (ohne freie Variablen)

$$\theta_P = \exists y_1 \exists y_2 \ldots \exists y_m \varphi_P$$

besagt, dass das Programm überhaupt anhält. Es gilt also

$$\mathbb{N} \vDash \theta_P$$

genau dann, wenn P bei Nulleingabe anhält. Man beachte, dass die Abbildung, die einem jeden Programm P dieses θ_P zuordnet, effektiv durch eine Registermaschine durchführbar ist.

Wenn es ein Entscheidungsverfahren für arithmetische Sätze geben würde, das die Richtigkeit von $\mathbb{N} \vDash \theta_P$ entscheiden könnte, so würde es auch ein Entscheidungsverfahren für das Halteproblem geben im Widerspruch zu Satz 9.6. □

11.3. **Folgerungen aus der Unentscheidbarkeit.**

Wir werden aus der Unentscheidbarkeit weitere Folgerungen über die Aufzählbarkeit und die Axiomatisierbarkeit der Arithmetik in der ersten Stufe ziehen. Dazu werden wir diese Begriffe allgemein für sogenannte Theorien einführen.

Definition 11.3. Es sei A ein Symbolalphabet und L^A die zugehörige Sprache erster Stufe. Eine Teilmenge $T \subseteq L_0^A$ heißt *Theorie*, wenn T abgeschlossenen unter der Ableitungsbeziehung ist, d.h. wenn aus $T \vdash p$ bereits $p \in T$ folgt.

Zu jeder Ausdrucksmenge Γ ist die Menge $\Gamma^\vdash$ der aus Γ ableitbaren Sätze eine Theorie. Häufig wählt man „kleine“ und „handhabbare“ Mengen, um übersichtliche Theorien zu erhalten. Mengen, die eine Theorie erzeugen, heißen auch *Axiomensysteme* für diese Theorie. Es ist im Allgemeinen schwierig zu entscheiden, ob ein bestimmter Satz aus einem Axiomensystem ableitbar ist, also zu der entsprechenden Theorie dazugehört.

Wenn I eine Interpretation einer Sprache erster Stufe ist, so ist $I^\vDash$, also die Menge der in dem Modell gültigen Sätze, ebenfalls eine Theorie. Dies folgt direkt aus der Korrektheit des Ableitungskalküls. So ist $\mathbb{N}^\vDash$ eine Theorie zur Sprache L_0^{Ar}, die alle bei der Standardinterpretation gültigen Sätze beinhaltet.

Die Menge aller aus den Peano-Axiomen ableitbaren Sätze bildet die *Peano-Arithmetik*, die wir hier PA nennen. Es ist $\mathrm{PA} \subseteq \mathbb{N}^\vDash$.

Die Gesamtmenge L_0^A ist natürlich ebenfalls abgeschlossen unter der Ableitungsbeziehung. Sie ist widersprüchlich im Sinne der folgenden Definition.

Definition 11.4. Es sei A ein Symbolalphabet und L^A die zugehörige Sprache erster Stufe. Eine Theorie $T \subseteq L_0^A$ heißt *widersprüchlich*, wenn es einen Satz $p \in L_0^A$ gibt mit $p \in T$ und $\neg p \in T$.

Lemma 11.5. *Es sei A ein Symbolalphabet und L^A die zugehörige Sprache erster Stufe, wobei die Sprache zumindest eine Variable besitzen möge. Es sei $T \subseteq L_0^A$ eine Theorie. Dann ist T genau dann widersprüchlich, wenn $T = L_0^A$ ist.*

Beweis. Siehe Aufgabe 11.5. □

Man interessiert sich natürlich hauptsächlich für widerspruchsfreie (also nicht widersprüchliche) Theorien.

Definition 11.6. Es sei A ein Symbolalphabet und L^A die zugehörige Sprache erster Stufe. Eine Theorie T heißt *vollständig*, wenn für jeden Satz $p \in L_0^A$ gilt $p \in T$ oder $\neg p \in T$.

Dabei ist grundsätzlich auch erlaubt, dass sowohl p als auch $\neg p$ zu T gehört, doch liegt dann bereits eine widersprüchliche Theorie vor.

Zu einer Interpretation I einer Sprache erster Stufe ist die Gültigkeitsmenge $I^\vDash$ eine widerspruchsfreie vollständige Theorie. Dies ergibt sich aus dem rekursiven Aufbau der Gültigkeitsbeziehung (die beinhaltet, dass wir das Tertium non datur anerkennen - sonst wäre eine mathematische Argumentation nicht möglich).

Definition 11.7. Es sei A ein Symbolalphabet und L^A die zugehörige Sprache erster Stufe. Eine Theorie $T \subseteq L_0^A$ heißt *endlich axiomatisierbar*, wenn es endlich viele Sätze $p_1, \ldots, p_s \in L_0^A$ gibt mit $T = \{p_1, \ldots, p_s\}^\vdash$.

Das ist häufig zu viel verlangt, wie die einstufige Peano-Arithmetik zeigt (zumindest haben wir sie nicht durch ein endliches Axiomensystem eingeführt). Eine schwächere Variante wird in der folgenden Definition beschrieben.

Definition 11.8. Es sei A ein Symbolalphabet und L^A die zugehörige Sprache erster Stufe. Eine Theorie $T \subseteq L_0^A$ heißt *aufzählbar axiomatisierbar*, wenn es eine R-aufzählbare Satzmenge $\Gamma \subseteq L_0^A$ gibt mit $T = \Gamma^\vdash$.

Lemma 11.9. *Es sei A ein Symbolalphabet und L^A die zugehörige Sprache erster Stufe. Eine aufzählbar axiomatisierbare Theorie $T \subseteq L_0^A$ ist R-aufzählbar.*

Beweis. Es sei Γ eine aufzählbare Satzmenge, die T axiomatisiert, und es sei p_n, $n \in \mathbb{N}_+$, eine Aufzählung von Γ. Es sei q_n, $n \in \mathbb{N}_+$, eine Aufzählung der prädikatenlogischen Tautologien aus L^A. Wenn ein Satz r aus Γ ableitbar ist, so gibt es eine endliche Auswahl $p_1, \ldots, p_n$ aus Γ (bzw. aus der gewählten Aufzählung) derart, dass

$$\vdash p_1 \wedge \ldots \wedge p_n \to r$$

eine prädikatenlogische Tautologie ist. Daher leistet das folgende Verfahren, bei dem n wächst, das Gewünschte: Für jedes n notiert man die Tautologien $q_1, \ldots, q_n$ in der Form

$$q_i = a_1 \wedge \ldots \wedge a_s \to b\,.$$

Wenn q_i überhaupt diese Form besitzt, so ist diese eindeutig betimmt. Danach überprüft man für jedes $i \leq n$, ob alle $a_1, \ldots, a_s$ zu $\{p_1, \ldots, p_n\}$ gehören. Falls ja, und wenn b ein Satz ist, so wird b notiert. Danach geht man zum nächsten i. Wenn man $i = n$ erreicht hat, so geht man zu $n + 1$, wobei man aber wieder bei $i = 1$ anfängt. □

Satz 11.10. *Es sei A ein Symbolalphabet und L^A die zugehörige Sprache erster Stufe. Jede aufzählbare (oder aufzählbar axiomatisierbare), widerspruchsfreie und vollständige Theorie $T \subseteq L_0^A$ ist entscheidbar.*

Beweis. Nach Lemma 11.9 bedeutet die aufzählbare Axiomatisierbarkeit, dass schon die Theorie selbst aufzählbar ist. Sei also T aufzählbar, vollständig und widerspruchsfrei, und sei p_n, $n \in \mathbb{N}_+$, eine Aufzählung von T. Es sei $q \in L_0^A$ ein Satz. Wegen der Widerspruchsfreiheit und der Vollständigkeit gilt entweder $q \in T$ oder $\neg q \in T$. Daher kommt entweder q oder $\neg q$ in der Aufzählung von T vor. Bei $p_n = q$ ist $q \in T$ und bei $p_n = \neg q$ ist $q \notin T$. □

Bemerkung 11.11. Ohne die Voraussetzung der Widerspruchsfreiheit ist obiges Argument nicht durchführbar. Eine widersprüchliche Theorie ist natürlich aufzählbar und vollständig. Es lässt sich aber an einer Aufzählung

zu keinem Zeitpunkt mit Sicherheit ablesen, ob die Theorie widersprüchlich ist. Wenn bis zu einem bestimmten Zeitpunkt weder eine widersprüchliche Aussage noch eine Aussage und ihre Negation ausgegeben wurden, so lässt sich nicht entscheiden, ob dies an der Widerspruchsfreiheit der Theorie oder der Art der Aufzählung liegt.

Satz 11.12. *Die Menge der wahren arithmetischen Ausdrücke ist nicht R-aufzählbar. D.h. es gibt kein R-Verfahren, das alle in $\mathbb{N}$ wahren Sätze der arithmetischen Sprache auflistet.*

Beweis. Dies folgt direkt aus Satz 11.10 und aus Satz 11.2. □

Korollar 11.13. *Die (erststufige) Peano-Arithmetik ist unvollständig.*

Beweis. Wegen $PA \subseteq \mathbb{N}^{\models}$ würde die Vollständigkeit hier die Gleichheit bedeuten. Da die Peano-Arithmetik R-aufzählbar ist, würde aus Satz 11.10 die Entscheidbarkeit folgen im Widerspruch zu Satz 11.2. □

11.4. **Arbeitsblatt.**

Aufgabe 11.1. Beschreibe für die in Vorlesung 8 besprochenen Registerprogramme die Konfigurationsfolge bei Nulleingabe.

Aufgabe 11.2. Erstelle für das Registerprogramm (mit keinem Register und leerer Anfangsbelegung)

(1) Halte an

den zugehörigen arithmetischen Ausdruck, der die Anhalteeigenschaft beschreibt.

Aufgabe 11.3. Erstelle für das Registerprogramm (mit zwei Registern R_1, R_2 und leerer Anfangsbelegung)

(1) $1+$
(2) $2-$
(3) Halte an

den zugehörigen arithmetischen Ausdruck, der die Anhalteeigenschaft beschreibt.

Aufgabe 11.4. Erstelle für das Registerprogramm (mit zwei Registern R_1, R_2 und leerer Anfangsbelegung)

(1) $1+$
(2) $C(2,1)$

(3) Halte an

den zugehörigen arithmetischen Ausdruck, der die Anhalteeigenschaft beschreibt.

Aufgabe 11.5. Es sei A ein Symbolalphabet und L^A die zugehörige Sprache erster Stufe, wobei die Sprache zumindest eine Variable besitzen möge. Es sei $T \subseteq L_0^A$ eine Theorie. Zeige, dass T genau dann widersprüchlich ist, wenn $T = L_0^A$ ist.

Aufgabe 11.6. Begründe, dass die (durch die Peano-Axiome definierte) Peano-Arithmetik aufzählbar-axiomatisierbar ist.

12. Vorlesung

12.1. **Repräsentierbarkeit in einer Theorie.**

Wir haben schon in der zehnten Vorlesung davon gesprochen, wann eine arithmetische n-stellige Relation R (bzw. Funktion) in $\mathbb{N}$ arithmetisch repräsentierbar ist, wann es also einen arithmetischen Ausdruck ψ mit r freien Variablen derart gibt, dass dieser Ausdruck für jede Belegung genau dann wahr wird, wenn die Relation auf das Belegungstupel zutrifft. Da $\mathbb{N}^{\vdash}$ vollständig ist, ergibt sich daraus die Äquivalenz, dass $(n_1, \ldots, n_r) \notin R$ äquivalent zur Nichtgültigkeit $\mathbb{N} \not\models \psi(n_1, \ldots, n_r)$ und somit auch zur Gültigkeit der Negation $\mathbb{N} \models \neg\psi(n_1, \ldots, n_r)$ ist. Bei nichtvollständigen Ausdrucksmengen bzw. Theorien wollen wir auch von Repräsentierungen sprechen, wobei wir diese zweite Eigenschaft explizit fordern müssen.

Definition 12.1. Es sei Γ eine Menge von arithmetischen Ausdrücken. Eine Relation $R \subseteq \mathbb{N}^r$ heißt *repräsentierbar* in Γ, wenn es einen L^{Ar}-Ausdruck ψ in r freien Variablen gibt derart, dass für alle r-Tupel $(n_1, \ldots, n_r) \in \mathbb{N}^r$ die beiden Eigenschaften

(1) Wenn $(n_1, \ldots, n_r) \in R$, so ist $\Gamma \vdash \psi(n_1, \ldots, n_r)$,
(2) Wenn $(n_1, \ldots, n_r) \notin R$, so ist $\Gamma \vdash \neg\psi(n_1, \ldots, n_r)$,

gelten.

Definition 12.2. Es sei Γ eine Menge von arithmetischen Ausdrücken. Eine Funktion

$$F : \mathbb{N}^r \longrightarrow \mathbb{N}^s$$

heißt *repräsentierbar* in Γ, wenn es einen L^{Ar}-Ausdruck ψ in $r + s$ freien Variablen gibt derart, dass für alle $(r + s)$-Tupel $(n_1, \ldots, n_{r+s}) \in \mathbb{N}^{r+s}$ die folgenden Eigenschaften

(1) Wenn $F(n_1, \ldots, n_r) = (n_{r+1}, \ldots, n_{r+s})$, so ist $\Gamma \vdash \psi(n_1, \ldots, n_{r+s})$,

(2) Wenn $F(n_1, \ldots, n_r) \neq (n_{r+1}, \ldots, n_{r+s})$, so ist $\Gamma \vdash \neg\psi(n_1, \ldots, n_{r+s})$,
(3) $\Gamma \vdash \exists! x_{r+1} \ldots \exists! x_{r+s} \psi(n_1, \ldots, n_r, x_{r+1}, \ldots, x_{r+s})$,

gelten.

Die dritte Eigenschaft besagt, dass die Theorie beweisen kann, dass es sich um eine Funktion handelt. Diese Eigenschaft folgt nicht aus den beiden ersten Eigenschaften.

Definition 12.3. Es sei Γ eine Menge von arithmetischen Ausdrücken. Man sagt, dass Γ *Repräsentierungen erlaubt*, wenn Γ jede R-berechenbare Relation und jede R-berechenbare Funktion repräsentiert.

Korollar 12.4. *Die natürliche Arithmetik, also die Menge der in $\mathbb{N}$ wahren Ausdrücke $\mathbb{N}^{\models}$, erlaubt Repräsentierungen.*

Beweis. Es sei $R \subseteq \mathbb{N}^r$ eine R-entscheidbare Relation und es sei P ein Registerprogramm, das diese Relation entscheidet. Aufgrund von Lemma 11.1 gibt es einen arithmetischen Ausdruck ψ_P, der den Programmablauf arithmetisch modelliert. Es gilt also $(n_1, \ldots, n_r) \in R$ genau dann, wenn P, angesetzt auf $(n_1, \ldots, n_r)$ anhält mit der Ausgabe 0 (und andernfalls anhält mit der Ausgabe 1), genau dann, wenn $\mathbb{N} \models \psi_P(n_1, \ldots, n_r, 0)$ gilt. Wegen der Vollständigkeit von $\mathbb{N}$ bedeutet dies, dass ψ_P die Relation repräsentiert. Es sei

$$F : \mathbb{N}^r \longrightarrow \mathbb{N}^s$$

eine R-berechenbare Abbildung und es sei P ein Registerprogramm, dass F berechnet. Aufgrund von Lemma 11.1 gibt es einen arithmetischen Ausdruck ψ_P, der den Programmablauf arithmetisch modelliert. D.h. für jedes $(r+s)$-Tupel $n_1, \ldots, n_{r+s}$ gilt

$$F(n_1, \ldots, n_r) \;=\; (n_{r+1}, \ldots, n_{r+s})$$

genau dann, wenn das Programm P bei jeder Eingabe anhält und angesetzt auf $(n_1, \ldots, n_r)$ die Ausgabe $(n_{r+1}, \ldots, n_{r+s})$ besitzt, genau dann, wenn $\mathbb{N} \models \psi(n_1, \ldots, n_{r+s})$ gilt. Da eine Funktion vorliegt und $\mathbb{N}^{\models}$ vollständig ist, gilt auch $\mathbb{N} \models \exists! x_{r+1} \ldots \exists! x_{r+s} \psi(n_1, \ldots, n_r, x_{r+1}, \ldots, x_{r+s})$. □

Bemerkung 12.5. Man kann zeigen, dass auch die erststufige Peano-Arithmetik Repräsentierungen erlaubt. Dazu muss man zeigen, dass die in Lemma 10.3 und in Lemma 11.1 konstruierten Ausdrücke, die die Wirkungsweise von Registerprogrammen beschreiben, nicht nur in $\mathbb{N}$ gelten, sondern aus den erststufigen Peano-Axiomen ableitbar sind. Es ist noch nicht einmal selbstverständlich, dass die Addition der natürlichen Zahlen in der Peano-Arithmetik repräsentierbar ist, obwohl dafür direkt das Additionssymbol zur Verfügung steht, siehe Aufgabe 12.4.

12.2. **Der Fixpunktsatz.**

Schon beim Halteproblem haben wir die Programmcodes durch eine natürliche Zahl effektiv repräsentiert, was uns ermöglichte, in ein Programm die eigene Programmnummer einzusetzen und so eine Selbstbezüglichkeit abzubilden, die zur Unlösbarkeit des Halteproblems führte. Ähnliches haben wir mit der Arithmetik vor, wobei die arithmetische Sprache durch die Symbole $0, 1, +, \cdot$ gegeben sei.

Den Ausdrücken der Sprache ordnen wir eine natürliche Zahl, ihre sogenannte *Gödelnummer* zu. Die Gödelnummer eines Ausdrucks p bezeichnen wir mit $GN(p)$. Wichtig ist dabei nicht die konkrete Gestalt, sondern allein ihre Effektivität in dem Sinne, dass diese Zuordnung durch eine Registermaschine ausführbar sein muss. Bei einem endlichen Alphabet ist die einfachste Möglichkeit, die Symbole mit Ziffern durchzunummerieren und die Ausdrücke durch die Hintereinanderschreibung der Ziffern in einem hinreichend großen Ziffernsystem zu realisieren. Da wir die Anzahl der Variablen nicht beschränken wollen, ist dies nicht direkt durchführbar. Im Falle von Programmen konnten wir die Register, deren Anzahl ebenfalls nicht beschränkt war, durch $R\prime\prime\ldots\prime$ benennen. Es ist auch möglich, in einem Zwischenschritt die Variablen $x_1, x_2, x_3 \ldots$ mit $x, x\prime, x\prime\prime, \ldots$ zu benennen und so ein endliches Alphabet zu erhalten. Eine andere Möglichkeit besteht darin, abzählbar unendlich viele Symbole mit den natürlichen Zahlen durchzunummerieren und einen Ausdruck der Form $s_{i_1} s_{i_2} \ldots s_{i_n}$ (i_j sei die Nummer des j-ten Symbols im Ausdruck) durch das Produkt $p_1^{i_1} p_2^{i_2} \cdots p_n^{i_n}$ wiederzugeben, wobei die $p_1, p_2, p_3, \ldots$ die Folge der Primzahlen sei.

Beispiel 12.6. Zu einer Ausdrucksmenge Γ kann man die Menge

$$\{GN(p) | \, \Gamma \vdash p\}$$

betrachten, also die Menge der Gödelnummern von Ausdrücken, die aus Γ ableitbar sind. Dies ist eine Teilmenge der natürlichen Zahlen, daher kann man auf diese Menge den Begriff der Repräsentierbarkeit anwenden. Eine natürliche Frage ist, ob diese Menge in Γ selbst repräsentierbar ist und welche Konsequenzen das hat.

Wir legen im Folgenden eine algorithmische Gödelisierung zu Grunde.

Satz 12.7. *Es sei $\Gamma \subseteq L^{\mathrm{Ar}}$ eine Menge von arithmetischen Ausdrücken, die Repräsentierungen erlaube. Dann gibt es zu jedem $p \in L_1^{\mathrm{Ar}}$ einen Satz $q \in L_0^{Ar}$ mit*

$$\Gamma \vdash q \leftrightarrow p(GN(q)) \, .$$

Beweis. Wir betrachten die Abbildung

$$F : \mathbb{N} \times \mathbb{N} \longrightarrow \mathbb{N}, \, (m, n) \longmapsto F(m, n),$$

die durch

$$F(m,n) := \begin{cases} GN(p(n)), & \text{falls } m \text{ die } GN \text{ eines } p \in L_1^{\mathrm{Ar}} \text{ ist}\,, \\ 0 \text{ sonst}\,. \end{cases}$$

Bei der Berechnung von F wird also zuerst geschaut, ob das erste Argument, also m, die Gödelnummer eines arithmetischen Ausdrucks mit genau einer freien Variablen ist. Falls nicht, so ist $F(m,n) = 0$, unabhängig von n. Falls ja, so ist also $m = GN(p)$ mit $p \in L_1^{\mathrm{Ar}}$. In diesem Ausdruck wird dann die einzige freie Variable durch das zweite Argument der Abbildung, also n, ersetzt, wobei man einen Satz $p(n)$ erhält. Dessen Gödelnummer ist nach Definition der Wert der Abbildung $F(m,n)$. Diese Erläuterung zeigt zugleich, dass F berechenbar ist.Da Γ nach Voraussetzung Repräsentierungen erlaubt, gibt es einen Ausdruck $\varphi(x,y,z)$ mit drei freien Variablen, der diese Abbildung repräsentiert. D.h. es gilt für jede Belegung der Variablen mit natürlichen Zahlen m,n,k die Beziehungen (wir können annehmen, dass Γ widerspruchsfrei ist, da andernfalls das Resultat trivial ist)

$$F(m,n) = k \text{ genau dann, wenn } \Gamma \vdash \varphi(m,n,k)\,,$$

$$F(m,n) \neq k \text{ genau dann, wenn } \Gamma \vdash \neg\varphi(m,n,k)$$

und (für jede Belegung für x und y)

$$\Gamma \vdash \exists! z \varphi(m,n,z)\,.$$

Den Fixpunkt zu einem vorgegebenen $p \in L_1^{\mathrm{Ar}}$ erhalten wir nun durch eine trickreiche Anwendung von φ. Wir setzen

$$s := \forall z(\varphi(x,x,z) \to p(z))\,.$$

Der Ausdruck s besitzt die Gödelnummer $GN(s)$. Wir behaupten nun, dass der Satz

$$q \;:=\; s\frac{GN(s)}{x} \;=\; \forall z(\varphi(GN(s),GN(s),z) \to p(z))$$

die zu beweisende Ableitungsbeziehung $\Gamma \vdash q \leftrightarrow p(GN(q))$ erfüllt. Der Ausdruck s besitzt die einzige freie Variable x, daher gilt

$$F(GN(s),GN(s)) \;=\; GN(s\frac{GN(s)}{x}) \;=\; GN(q).$$

Aufgrund der Repräsentierungseigenschaft ist daher

$$\Gamma \vdash \varphi(GN(s),GN(s),GN(q))\,.$$

Aus der Allaussage q erhält man durch Spezialisierung (man ersetzt die Variable z durch den Term $GN(q)$)

$$\vdash q \to (\varphi(GN(s),GN(s),GN(q)) \to p(GN(q)))\,.$$

Da das Antezedens der rechten Implikation aus Γ ableitbar ist, folgt

$$\Gamma \vdash q \to p(GN(q))\,.$$

Dies besagt also die Ableitbarkeit der Hinrichtung. Die aufgrund der Repräsentierbarkeit oben angeführte eindeutige Existenzaussage führt zu

$$\Gamma \vdash \forall z(\varphi(GN(s), GN(s), z) \to (z = GN(q)))\,.$$

Durch Substitution ergibt sich

$$\vdash (z = GN(q)) \to (p(GN(q)) \to p(z))$$

und somit

$$\Gamma \vdash \forall z(\varphi(GN(s), GN(s), z) \wedge p(GN(q)) \to p(z))\,.$$

Dies kann man als

$$\Gamma \vdash p(GN(q)) \to (\forall z(\varphi(GN(s), GN(s), z) \to p(z)))$$

schreiben, und das Sukzedens ist gerade q, so dass auch die Rückrichtung ableitbar ist. □

12.3. **Arbeitsblatt.**

Aufgabe 12.1. Zeige, dass eine widersprüchliche Ausdrucksmenge $\Gamma \subseteq L^{\mathrm{Ar}}$ Repräsentierungen erlaubt.

Aufgabe 12.2. Es sei $\Gamma \subseteq L^{\mathrm{ar}}$ eine Ausdrucksmenge, die Repräsentierungen erlaube. Zeige, dass jede größere Ausdrucksmenge $\Gamma' \supseteq \Gamma$ ebenfalls Repräsentierungen erlaubt.

Aufgabe 12.3. Es sei $\Gamma \subseteq L^{\mathrm{ar}}$ eine widerspruchsfreie und R-entscheidbare Ausdrucksmenge.

a) Zeige, dass jede in Γ repräsentierbare Relation $R \subseteq \mathbb{N}^r$ R-entscheidbar ist.

b) Zeige, dass jede in Γ repräsentierbare Abbildung

$$\varphi : \mathbb{N}^r \longrightarrow \mathbb{N}^s$$

R-berechenbar ist.

Aufgabe 12.4. Zeige, dass in der erststufigen Peano-Arithmetik die Addition von natürlichen Zahlen repräsentierbar ist.

13. Vorlesung

13.1. **Der erste Gödelsche Unvollständigkeitssatz.**

Wir haben gesehen, dass die Unentscheidbarkeit des Halteproblems über die arithmetische Repräsentierbarkeit von Registerprogrammen zur Unentscheidbarkeit der Arithmetik führt. Beim Beweis des ersten Gödelschen Unvollständigkeitssatzes arbeitet man mit einem Fixpunkt zu einem negierten Ableitungsprädikat, um eine „paradoxe“ Situation zu erhalten. Ein Ableitungsprädikat $a(x)$ soll die Eigenschaft haben, dass $\Gamma \vdash s$ genau dann gilt, wenn $\Gamma \vdash a(GN(s))$ gilt. Ein solches Ableitungsprädikat muss es im Allgemeinen nicht geben. Im folgenden *Unvollständigkeitslemma* gehört die Existenz eines Ableitungsprädikates zur Voraussetzung.

Lemma 13.1. *Es sei Γ eine widerspruchsfreie, arithmetische Ausdrucksmenge, die Repräsentierungen erlaube. Die Ableitungsmenge $\Gamma^{\vdash}$ (also die Menge der zugehörigen Gödelnummern) sei repräsentierbar in Γ. Dann gibt es einen arithmetischen Satz q derart, dass weder q noch seine Negation $\neg q$ aus Γ ableitbar ist. Die Ableitungsmenge $\Gamma^{\vdash}$ ist also nicht vollständig.*

Beweis. Die Repräsentierbarkeit von $\Gamma^{\vdash}$ bedeutet, dass es einen arithmetischen Ausdruck in einer freien Variablen gibt, sagen wir $a(x)$, mit der Eigenschaft, dass

$$\Gamma \vdash s$$

genau dann gilt, wenn

$$\Gamma \vdash a(GN(s))$$

gilt. Wir betrachten die Negation $p = \neg a$. Nach Satz 12.7 gibt es für p einen Fixpunkt, also einen Satz q mit

$$\Gamma \vdash q \longleftrightarrow p(GN(q))$$

bzw.

$$\Gamma \vdash q \longleftrightarrow \neg a(GN(q))\,.$$

Sowohl aus $\Gamma \vdash q$ als auch aus $\Gamma \vdash \neg q$ ergibt sich dann direkt ein ableitbarer Widerspruch, was der Widerspruchsfreiheit des Systems widerspricht. □

Man beachte, dass die Repräsentierbarkeit der Ableitungsmenge hier eine explizite Voraussetzung ist, die nicht aus der allgemein vorausgesetzten Eigenschaft, Repräsentierungnen zu erlauben, folgt. Letztere bezieht sich nur auf rekursive Relationen und Funktionen, es wird aber nicht vorausgesetzt, dass Γ selbst oder $\Gamma^{\vdash}$ rekursiv ist.

Was passiert, wenn man den Satz q (oder seine Negation) einfach zu Γ hinzunimmt? Kann man so nicht Γ „vollständig auffüllen“? Das Problem ist hierbei, dass $(\Gamma \cup \{q\})^{\vdash}$ nicht mehr repräsentierbar in $\Gamma \cup \{q\}$ sein muss.

Kurt Gödel (1906-1978) bewies im Alter von 24 Jahren seine Unvollständigkeitssätze.

Satz 13.2. *Es sei* Γ *eine arithmetische Ausdrucksmenge, die widerspruchsfrei und aufzählbar ist und Repräsentierungen erlaube. Dann ist* $\Gamma^{\vdash}$ *unvollständig. Es gibt also einen arithmetischen Satz, für den weder* $\Gamma \vdash q$ *noch* $\Gamma \vdash \neg q$ *gilt.*

Beweis. Wir nehmen an, dass $\Gamma^{\vdash}$ vollständig ist. Da Γ aufzählbar ist, ist $\Gamma^{\vdash}$ nach Lemma 11.9 aufzählbar und nach Satz 11.10 auch entscheidbar. Da Γ Repräsentierungen erlaubt, ist insbesondere $\Gamma^{\vdash}$ repräsentierbar. Daher sind die Voraussetzungen von Lemma 13.1 erfüllt und es ergibt sich ein Widerspruch zur angenommenen Vollständigkeit. □

Korollar 13.3. *Es sei* Γ *eine arithmetische korrekte Ausdrucksmenge, die aufzählbar sei und Repräsentierungen erlaube. Dann gibt es einen in (der Standardinterpretation)* $\mathbb{N}$ *wahren Satz, der nicht zu* $\Gamma^{\vdash}$ *gehört, der also nicht aus* Γ *formal ableitbar ist.*

Beweis. Die Korrektheit bedeutet, dass $\Gamma^{\vdash} \subseteq \mathbb{N}^{\models}$ gilt. Dies sichert zugleich die Widerspruchsfreiheit von Γ. Gemäß Satz 13.2 gibt es einen Satz q, der weder selbst noch seine Negation $\neg q$ aus Γ ableitbar ist. Da aber $\mathbb{N}^{\models}$ vollständig ist, muss entweder q oder $\neg q$ in $\mathbb{N}$ wahr sein. □

Diese Aussage ist für die Peano-Arithmetik und jedes größere aufzählbare widerspruchsfreie System anwendbar.

13.2. **Der zweite Gödelsche Unvollständigkeitssatz.**

Wenn die Ableitungsrelation $\Gamma^{\vdash}$ repräsentierbar ist und der zugehörige repräsentierende arithmetische Ausdruck a bekannt ist, so ist auch der im Beweis zu Lemma 13.1 verwendete Ausdruck q, (also der Fixpunkt zu $\neg a(x)$)

prinzipiell bekannt, da der Fixpunktsatz konstruktiv ist. Im Beweis des ersten Gödelschen Unvollständigkeitssatz war ein solches Ableitungsprädikat a aber nur aufgrund der angenommenen Vollständigkeit vorhanden, die dann zum Widerspruch geführt wurde. Aus diesen Überlegungen ergibt sich weder die Existenz eines Ableitungsprädikates noch die eines Fixpunktes zum negierten Ableitungsprädikat.

Der zweite Gödelsche Unvollständigkeitssatz gibt hingegen explizit einen Satz an, der weder selbst noch seine Negation beweisbar ist, und zwar einen Satz von großer inhaltlicher Bedeutung: Es geht um den Satz, der die Widerspruchsfreiheit des gegebenen Systems behauptet.

Betrachten wir zunächst eine beliebige korrekte arithmetische Theorie $T \subseteq L^{\mathrm{Ar}}$, also eine deduktiv abgeschlossene Satzmenge, die bei der Standardinterpretation in $\mathbb{N}$ nur wahre Sätze ergibt (dazu genügt es wegen der Korrektheit des Ableitungskalküls, dass sämtliche Sätze aus einem Axiomensystem Γ für T (also $T = \Gamma^{\vdash}$) in $\mathbb{N}$ wahr sind). Da $\mathbb{N}^{\models}$, wie jede Gültigkeitsmenge eines Modells, vollständig und widerspruchsfrei ist, ist auch T (als Teilmenge von $\mathbb{N}^{\models}$) widerspruchsfrei. Daher gehört zu T kein Satz der Form $p \wedge \neg p$ und auch nicht der Satz $\neg(0 = 0)$ (da ja die Identität $0 = 0$ dazugehört). Eine andere Frage ist es, ob das System bzw. die Theorie oder das Axiomensystem diese Unableitbarkeit eines widersprüchlichen Satzes auch „weiß“.

Schon im Unvollständigkeitslemma und im ersten Gödelschen Unvollständigkeitssatz kam wesentlich ein Ableitungsprädikat a vor. Dieses hatte die Eigenschaft

$$\Gamma \vdash s \text{ genau dann, wenn } \Gamma \vdash a(GN(s))\,,$$

allerdings unter der Bedingung, dass $\Gamma^{\vdash}$ entscheidbar und damit in Γ (das Repräsentierungen erlaube) repräsentierbar ist. Aus der Entscheidbarkeit von Γ folgt zwar die Aufzählbarkeit von $\Gamma^{\vdash}$, und daraus, wenn $\Gamma^{\vdash}$ zusätzlich vollständig ist, auch die Entscheidbarkeit von $\Gamma^{\vdash}$, sonst aber nicht. Diese Überlegung haben wir schon in umgekehrter Richtung angewendet, indem wir aus der Unentscheidbarkeit der Arithmetik auf die Unvollständigkeit der Peano-Arithmetik geschlossen haben (siehe Korollar 11.13). Es ist also keineswegs selbstverständlich, dass es ein sinnvolles entscheidbares Ableitungsprädikat gibt.

Allerdings ist ein schwächeres Ableitungsprädikat entscheidbar und damit repräsentierbar, nämlich die folgende zweistellige Ableitungsrelation. Dazu sei die Gödelisierung auf endliche Folgen von Ausdrücken (die mögliche Ableitungsketten repräsentieren möge) ausgedehnt. Wir betrachten dann das zweistellige Prädikat

$$A \subseteq \mathbb{N} \times \mathbb{N}$$

(eigentlich $A_\Gamma(x, y)$, da diese Teilmenge von Γ abhängt), mit der Eigenschaft, dass $(m, n) \in A$ genau dann gilt, wenn m die Gödelnummer einer korrekten Ableitung im Prädikatenkalkül aus Γ ist, deren letzter Ausdruck (also der in

der Ableitung bewiesene Ausdruck) die Gödelnummer n besitzt. Diese Relation ist unter der Voraussetzung, dass Γ entscheidbar ist, selbst entscheidbar. Man kann ja zum ersten Eintrag m die Ableitung rekonstruieren, ihre Korrektheit im Prädikatenkalkül überprüfen und aufgrund der Entscheidbarkeit von Γ feststellen, ob nur Ausdrücke aus Γ als Voraussetzungen verwendet wurden. Wenn Γ Repräsentierungen erlaubt, so gibt es einen arithmetischen Ausdruck mit zwei freien Variablen, sagen wir $\delta(x,y)$ (eigentlich $\delta_\Gamma(x,y)$, da dieses Prädikat von Γ abhängt), der für jede Belegung $(m,n) \in \mathbb{N}^2$ genau dann aus Γ ableitbar ist, wenn m einen Beweis für die Aussage zu n kodiert.

Wie formuliert man die Eigenschaft, dass es einen prädikatenlogischen Beweis aus Γ für die Aussage zu n gibt? Innerhalb der natürlichen Zahlen ist dies äquivalent dazu, dass es ein $m \in \mathbb{N}$ gibt mit $A(m,n)$. Dies muss aber *nicht* äquivalent zu $\Gamma \vdash \exists x \delta(x,y)$ sein. Wir setzen

$$\alpha(y) = \exists x \delta(x,y)\,.$$

Wenn Γ Repräsentierungen erlaubt, so gibt es aufgrund des Fixpunktsatzes, angewendet auf den negierten Ausdruck $\neg\alpha(y)$, einen Ausdruck $q \in L^{\mathrm{Ar}}$ mit

$$\Gamma \vdash q \longrightarrow \neg\alpha(GN(q))\,.$$

Dieser Satz q kann nun aus Γ nicht ableitbar sein, es sei denn, dass Γ widersprüchlich ist. Wenn nämlich $\Gamma \vdash q$ gilt, so bedeutet dies, dass es eine korrekte Ableitung von q aus Γ gibt. Diese Ableitung wird durch eine Zahl m kodiert und daher gilt

$$\Gamma \vdash \delta(m, GN(q))\,,$$

da ja δ das zweistellige Ableitungsprädikat repräsentiert. Daher ist auch

$$\Gamma \vdash \exists x \delta(x, GN(q))\,,$$

also $\Gamma \vdash \alpha(GN(q))$. Die Negation der Fixpunkteigenschaft ergibt somit

$$\Gamma \vdash \neg q\,,$$

so dass ein Widerspruch vorliegt.

Das Beweisprädikat $\alpha(y)$ besitzt, zumindest, wenn Γ die Peano-Arithmetik umfasst, einige ausdruckstarke Eigenschaften, die auch in Γ ableitbar sind. Der Beweis von diesen Eigenschaften ist aufwändig, da sie nicht abstrakt aus der Repräsentierbarkeit folgen, sondern im Beweiskalkül erarbeitet werden müssen. Wichtige Eigenschaften sind (Γ sei entscheidbar und enthalte die Peano-Arithmetik)

- Wenn $\Gamma \vdash s$, so ist $\Gamma \vdash \alpha(GN(s))$ für jeden Ausdruck $s \in L^{\mathrm{Ar}}$.

- Für je zwei Ausdrücke $s,t \in L^{\mathrm{Ar}}$ ist $\Gamma \vdash \alpha(GN(s \to t)) \to (\alpha(GN(s)) \to \alpha(GN(t)))$.

- Für jeden Ausdruck $s \in L^{\mathrm{Ar}}$ ist $\Gamma \vdash \alpha(GN(s)) \to \alpha(GN(\alpha(GN(s))))$.

Diese und ähnliche Gesetzmäßigkeiten sind der Ausgangspunkt der *Beweisbarkeitslogik*, die in der Sprache der Modallogik beweistheoretische Fragestellungen untersucht.

Die aufgelisteten Eigenschaften sind für ein Ableitungsprädikat natürlich wünschenswert; der naive Wunsch $\vdash s \longleftrightarrow \alpha(GN(s))$ ist nicht realisierbar, da er in Verbindung mit dem Satz q von oben (der Fixpunkt zur Negation $\neg\alpha(GN(s))$) sofort einen internen Widerspruch ergibt. Die Verbindung der „positiven“ Eigenschaften des Ableitungsprädikates mit dem „paradoxen“ q aus dem Fixpunktsatz liefert einen Beweis für den zweiten Unvollständigkeitssatz. Dazu braucht man nicht die volle Liste von oben, sondern es genügt zu wissen, dass die weiter oben auf Grundlage der Widerspruchsfreiheit von Γ gezeigte Unableitbarkeit von q aus Γ sich in der Peano-Arithmetik selbst nachvollziehen lässt. D.h. es gilt

$$PA \vdash WF(\Gamma) \longrightarrow \neg\alpha(GN(q))$$

Dabei realisieren wir die Widerspruchsfreiheit $WF(\Gamma)$ intern durch die Unableitbarkeit des weiter oben schon erwähnten widersprüchlichen Satzes $r = \neg(0 = 0)$, also durch

$$WF(\Gamma) = \neg\alpha(GN(r))\,.$$

Satz 13.4. *Es sei Γ eine arithmetische Ausdrucksmenge, die widerspruchsfrei und entscheidbar sei und die Peano-Arithmetik umfasse. Dann ist die Widerspruchsfreiheit $WF(\Gamma)$ nicht aus Γ ableitbar, d.h. es ist*

$$\Gamma \nvdash WF(\Gamma)\,.$$

Beweis. Aus der Annahme $\Gamma \vdash WF(\Gamma)$ folgt wegen

$$PA \vdash WF(\Gamma) \longrightarrow \neg\alpha(GN(q))$$

(was wir allerdings nicht bewiesen haben) direkt

$$\Gamma \vdash \neg\alpha(GN(q))\,.$$

Aus der Fixpunkteigenschaft von q folgt somit $\Gamma \vdash q$, was aber in dem widerspruchsfreien System Γ nach obiger Überlegung nicht sein kann. □

Anhang 1: Bildlizenzen

Die Bilder dieses Textes stammen aus Commons (also http://commons.wikimedia.org), und stehen unter unterschiedlichen Lizenzen, die zwar alle die Verwendung hier erlauben, aber unterschiedliche Bedingungen an die Verwendung und Weitergabe stellen. Es folgt eine Auflistung der verwendeten Bilder dieses Textes (nach der Seitenzahl geordnet, von links nach rechts, von oben nach unten) zusammen mit ihren Quellen, Urhebern (Autoren) und Lizenzen. Dabei ist *Quelle* so zu verstehen, dass sich, wenn man

http://commons.wikimedia.org/wiki/File:

unmittelbar davor setzt, die entsprechende Datei auf Commons ergibt. *Autor* benennt den Urheber des Werkes, falls dieser bekannt ist. *Benutzer* meint den Hochlader der Datei; wenn keine weitere Information über den Autor vorliegt, so gilt der Benutzer als Urheber. Die Angabe des Benutzernamen ist so zu verstehen, dass sich, wenn man

http://commons.wikimedia.org/wiki/User:

unmittelbar davor setzt, die Benutzerseite ergibt. Wenn das Bild ursprünglich in einem anderen Wikimedia-Projekt hochgeladen wurde, so wird die Domäne (bspw. *de.wikipedia.org*) explizit angegeben.

Die *Lizenz* ist die auf der Dateiseite auf Commons angegebene Lizenz. Dabei bedeuten

- GFDL: Gnu Free Documentation License (siehe den angehängten Text, falls diese Lizenz vorkommt)
- CC-BY-SA-2.5 (3.0): Creative Commons Attribution ShareAlike 2.5 (oder 3.0)
- PD: gemeinfrei (public domain)

Abbildungsverzeichnis

Quelle = Gottfried Wilhelm Leibniz c1700.jpg , Autor = Johann Friedrich Wentzel d. Ä. (= Benutzer AndreasPraefcke auf Commons), Lizenz = PD 11

Quelle = Goldbach-1000.svg , Autor = Benutzer Mucfish auf Commons, Lizenz = PD 13

Quelle = DNA structure and bases FR.svg , Autor = Benutzer Dosto auf Commons, Lizenz = CC-by-sa 2.5 15

Quelle = Uni Freiburg - Philosophen 4.jpg , Autor = Cipri Adolf Bermann (= Benutzer Michael Sch. auf Commons), Lizenz = CC-BY-SA-2.5 23

Quelle = Giuseppe Peano.jpg , Autor = Benutzer Kalki auf Commons, Lizenz = PD? 43

Quelle = Alan Turing cropped.jpg , Autor = Jon Callas (= Benutzer Compro auf Commons), Lizenz = CC by sa 2.0 63

Quelle = 1925 kurt gödel.png , Autor = Benutzer Kl833x9 auf Commons, Lizenz = PD 94

Anhang 2: GFDL-Lizenz

GNU Free Documentation License Version 1.2,

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others. This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software. We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holdersaying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law. A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language. A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal,

commercial, philosophical, ethical or political position regarding them. The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none. The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words. A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straight forwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text for matters or for automatic translation to a variety of formats suitable for input to text for matters. A copy made in anotherwise Transparent file format whose mark up, or absence of mark up, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque". Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only. The"Title Page" means, for a printed book, the title page itself, plussuch following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formants which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text. A section"Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when youmodify the Document means that it remains a section "Entitled XYZ" according to this definition. The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in

this License, butonly as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this

License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3. You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, youmust enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on thecovers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects. If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages. If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public. It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to who ever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document. E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be replaced in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of thesection, and preserve in the section all the substance and tone of each the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such as section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles. You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties–for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard. You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one. The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers. The combined work need only contain one copy of this License, and multiple identical Invariant Sections maybe replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher

of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work. In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects. You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individualworks permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document. If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you maydistribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version

will prevail. If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement(section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/. Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of anylater version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOURNAME. Permission is granted to copy, distribute and/or modifythis document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation. If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.