Beweise als Programme

Helmut Schwichtenberg

Mathematisches Institut der Ludwig–Maximilians–Universität München
Sommersemester 1996

Inhaltsverzeichnis

1 Einführung **2**

1.1 Ein Existenzbeweis ohne Beispielsangabe 2

1.2 Formale Systeme 2

1.3 Die Sprache der Logik erster Stufe 3

1.4 Terme und Formeln 7

2 Natürliche Herleitungen **10**

2.1 Einführungs– und Beseitigungsregeln 10

2.2 Herleitungen als Terme 13

2.3 Klassische und intuitionistische Logik 16

3 Arithmetik **20**

3.1 Sprachen der Arithmetik 20

3.2 Arithmetische Herleitungen 21

3.3 Beispiel: Obere Schranke von + und · 23

3.4 Beispiel: Das Wall Street Problem 25

4 Beweise als Programme **29**

4.1 Existenz einer Normalform 29

4.2 Folgerungen aus der Normalisierung 34

4.3 Interpretation von Beweisen 36

4.4 Programmentwicklung durch Beweistransformation 38

4.5 Ausblick 40

Index **41**

Literatur **44**

Kapitel 1

Einführung

{Einf}

1.1 Ein Existenzbeweis ohne Beispielsangabe

{Sqrt2}

Untersuchungsgegenstand dieser Vorlesung sind mathematische Beweise. Bei diesen Untersuchungen werden einige interessante Aspekte von Beweisen zum Vorschein kommen. So ist es zum Beispiel möglich und sinnvoll, zwischen Existenzbeweisen zu unterscheiden, die das als existent nachgewiesene Objekt tatsächlich liefern, und solchen, die dies nicht tun. Als Beispiel betrachten wir die folgende Aussage.

Es gibt irrationale Zahlen a, b mit a^b rational.

Einen Beweis erhält man wie folgt durch Fallunterscheidung.

Fall $\sqrt{2}^{\sqrt{2}}$ ist rational. Man wähle $a = \sqrt{2}$ und $b = \sqrt{2}$. Dann sind a, b irrational, und nach Annahme ist a^b rational.

Fall $\sqrt{2}^{\sqrt{2}}$ ist irrational. Man wähle $a = \sqrt{2}^{\sqrt{2}}$ und $b = \sqrt{2}$. Dann sind nach Annahme a, b irrational, und

$$a^b = \left(\sqrt{2}^{\sqrt{2}}\right)^{\sqrt{2}} = \left(\sqrt{2}\right)^2 = 2$$

ist rational. □

Solange wir nicht entschieden haben, ob $\sqrt{2}^{\sqrt{2}}$ nun rational ist oder nicht, wissen wir nicht, welche Zahlen a, b wir nehmen müssen. Damit haben wir ein Beispiel eines Existenzbeweises, der es nicht erlaubt, das als existent nachgewiesene Objekt tatsächlich anzugeben.

1.2 Formale Systeme

{FormSys}

Der Mathematiker studiert Axiomensysteme. Man unterscheidet zwischen Axiomensystemen, die ein intendiertes Modell möglichst vollständig beschreiben sollen (etwa die natürlichen Zahlen oder die ebene Geometrie) und "modernen" Axiomensystemen wie etwa denen für Gruppen, Ringe und Körper, die von vornherein so konzipiert sind, daß sie möglichst viele Modelle besitzen. Die Grundbegriffe der jeweiligen mathematischen Disziplin sind dann durch die Axiome implizit bestimmt. Ausgehend von den Grundbegriffen werden weitere Begriffe definiert, die aber prinzipiell entbehrlich sind. Aus den Axiomen werden dann mittels logischer Schlußregeln die Sätze der Theorie hergeleitet. — Wir werden uns hier in erster Linie mit Axiomensystemen der ersten Art befassen.

Einen mathematischen Satz kann man unter zwei Aspekten betrachten: einerseits teilt er einen Inhalt mit (semantischer Aspekt), andererseits liegt er als Zeichenreihe vor (syntaktischer

Aspekt). Der semantische Aspekt scheint auf den ersten Blick viel wichtiger zu sein. Es ist jedoch nötig und in manchen Fällen ergiebig, sich auch mit dem syntaktischen Aspekt zu befassen.

Die syntaktische Seite eines Axiomensystems fassen wir in dem Begriff eines *formalen Systems* zusammen. Ein formales System ist bestimmt durch

(1) eine (formale) Sprache, und

(2) Axiome und Schlußregeln.

Als Sprache verwendet man am besten eine künstliche, formale Sprache, um Eindeutigkeit und Einfachheit zu gewährleisten. Eine solche Sprache ist bestimmt durch einen Zeichenvorrat (Alphabet) und daraus gebildete Terme (die Objekte bedeuten) und Formeln (die Aussagen bedeuten). Eine Aussage ist nach Aristoteles ein *sprachliches Gebilde, von dem es sinnvoll ist zu sagen, es sei wahr oder falsch*. Die Axiome sind dann einfach spezielle Formeln, und die Schlußregeln gestatten den Übergang von gewissen (meist endlich vielen) Prämissen auf eine Konklusion — im allgemeinen unter zusätzlichen Bedingungen.

Wir wollen hier eine Klasse von formalen Systemen beschreiben, die auf der Logik erster Stufe basieren. Sie reichen für die Darstellung der üblichen Axiomensysteme der Mathematik aus. Als erstes wollen wir ihre Sprache einführen. Dazu beginnen wir mit einer informalen Diskussion.

1.3 Die Sprache der Logik erster Stufe

{*Sprache*}

Jede Sprache verwendet Begriffe. In unserem Fall einer Sprache über mathematische Gegenstände können wir es uns relativ einfach machen. Hier kann man zwischen logischen und nichtlogischen Begriffen unterscheiden.

Wir behandeln zunächst die nichtlogischen Begriffe. Wir legen einen sogenannten Individuenbereich zugrunde (etwa den Bereich $\mathbb{N}$ der natürlichen Zahlen oder die Trägermenge G einer Gruppe) und verstehen dann unter einem Begriff einfach eine Teilmenge des Individuenbereichs; ein Beispiel ist der Primzahlbegriff. In der Sprache muß dann für jeden relevanten Begriff ein Symbol zu seiner Benennung vorhanden sein, etwa P für den Primzahlbegriff; solche Symbole nennt man Relationssymbole. Es ist notwendig, auch mehrstellige Begriffe zuzulassen, etwa die Kleiner–oder–Gleich–Beziehung $n \leq m$ zwischen natürlichen Zahlen. Die zugehörigen Symbole nennt man Relationssymbole.

Ferner brauchen wir in der Mathematik offenbar Funktionen, etwa die Nachfolgerfunktion und die Addition auf $\mathbb{N}$ oder die Gruppenverknüpfung. In der Sprache muß dann für jede vorkommende Funktion ein Symbol zu ihrer Benennung vorhanden sein, etwa S für die Nachfolgerfunktion, $+$ für die Addition und $\circ$ für die Gruppenverknüpfung; solche Symbole nennt man Funktionssymbole. Jedes Funktionssymbol hat eine feste Stellenzahl. Ferner brauchen wir offenbar in der Sprache Konstanten für einzelne Individuen (etwa 5). Man kann Konstante als entartete (genauer: nullstellige) Funktionssymbole auffassen. Aus den Funktionssymbolen und den Konstanten kann man Terme aufbauen, etwa $\mathrm{S}(5)$ oder $5 + 7$. Mit solchen Termen lassen sich dann atomare Formeln bilden, etwa $P(7)$ (mit der Bedeutung "7 ist eine Primzahl") oder $\mathrm{S}(5) < 9$.

Wir nächstes betrachten wir die logischen Begriffe. Aus atomaren Formeln kann man mittels logischer Verknüpfungen weitere Formeln zusammensetzen, etwa $2 < 3 \wedge P(7)$. Allgemein ist mit A und B auch die Konjunktion $A \wedge B$ (gelesen "A und B") eine Formel.

Man beachte, daß die Wahrheit oder Falschheit von $A \wedge B$ allein von der Wahrheit oder Falschheit von A und von B abhängt, nicht von der Bedeutung dieser Aussagen. Wir nehmen deshalb mit Frege an, daß zwei verschiedene Wahrheitswerte W und F gegeben sind. Man beachte, daß es nicht darauf ankommt, was genau die Wahrheitswerte W und F sind, solange sie nur verschieden sind. Man könnte etwa 1 und 0 wählen. Wir betrachten jetzt Wahrheitsfunktionen $H\colon \{W,F\}^n \to \{W,F\}$. Die Konjunktion $\wedge$ wird dann vollständig durch ihre Wahrheitsfunktion $H_\wedge\colon \{W,F\}^2 \to \{W,F\}$ beschrieben, die durch

$$\begin{aligned} &H_\wedge(W,W) = W,\\ &H_\wedge(W,F) = H_\wedge(F,W) = H_\wedge(F,F) = F \end{aligned}$$

bestimmt ist.

Mit A ist auch die Negation $\neg A$ (gelesen "nicht A") eine Formel. Die zugehörige Wahrheitsfunktion $H_\neg\colon \{W,F\} \to \{W,F\}$ ist einstellig und bestimmt durch

$$H_\neg(W) = F, \qquad H_\neg(F) = W.$$

Mit A und B ist auch die Disjunktion $A \vee B$ (gelesen "A oder B") eine Formel. Man beachte, daß das "oder" hier im nicht ausschließenden Sinn gemeint ist; dies ist in der Mathematik üblich. Bei klassischer Auffassung ist die Disjunktion $\vee$ vollständig durch ihre Wahrheitsfunktion $H_\vee\colon \{W,F\}^2 \to \{W,F\}$ beschrieben, die durch

$$\begin{aligned} &H_\vee(W,W) = H_\vee(W,F) = H_\vee(F,W) = W,\\ &H_\vee(F,F) = F \end{aligned}$$

bestimmt ist. Man beachte, daß dann $A \vee B$ dasselbe bedeutet wie $\neg(\neg A \wedge \neg B)$, also $\vee$ auch durch $\wedge$ und $\neg$ definiert werden kann. Es gilt nämlich

$$\begin{aligned} &H_\neg(H_\wedge(H_\neg(W), H_\neg(W))) = H_\neg(H_\wedge(F,F)) = H_\neg(F) = W = H_\vee(W,W),\\ &H_\neg(H_\wedge(H_\neg(W), H_\neg(F))) = H_\neg(H_\wedge(F,W)) = H_\neg(F) = W = H_\vee(W,F),\\ &H_\neg(H_\wedge(H_\neg(F), H_\neg(W))) = H_\neg(H_\wedge(W,F)) = H_\neg(F) = W = H_\vee(F,W),\\ &H_\neg(H_\wedge(H_\neg(F), H_\neg(F))) = H_\neg(H_\wedge(W,W)) = H_\neg(W) = F = H_\vee(F,F). \end{aligned}$$

Mit A und B ist auch die Implikation $A \to B$ (gelesen "wenn A, so B" oder "A impliziert B") eine Formel. Man beachte, daß wenn–so hier im mathematischen Sinn gemeint ist, d.h. es braucht kein kausaler Zusammenhang zwischen A und B zu bestehen. So ist zum Beispiel $2 < 3 \to P(7)$ wahr. Die Wahrheitsfunktion $H_\to\colon \{W,F\}^2 \to \{W,F\}$ ist

$$\begin{aligned} &H_\to(W,W) = H_\to(F,W) = H_\to(F,F) = W,\\ &H_\to(W,F) = F. \end{aligned}$$

Schließlich ist mit A und B auch die Äquivalenz $A \leftrightarrow B$ (gelesen "A genau dann, wenn B" oder "A äquivalent B") eine Formel. Die Wahrheitsfunktion $H_\leftrightarrow\colon \{W,F\}^2 \to \{W,F\}$ ist

$$\begin{aligned} &H_\leftrightarrow(W,W) = H_\leftrightarrow(F,F) = W,\\ &H_\leftrightarrow(W,F) = H_\leftrightarrow(F,W) = F. \end{aligned}$$

Die Äquivalenz ist offenbar aus $\wedge$ und $\to$ definierbar durch $(A \to B) \wedge (B \to A)$.

Ferner ist es nützlich, noch das Falsum $\bot$ als logische Konstante einzuführen. Es ist bestimmt durch die "nullstellige" Wahrheitsfunktion $H_\bot := F$. Man kann dann etwa die Negation $\neg$ aus der Implikation $\to$ und dem Falsum $\bot$ definieren durch $\neg A :\equiv A \to \bot$, denn es gilt

$$H_\to(W, H_\bot) = H_\to(W, F) = F = H_\neg(W),$$
$$H_\to(F, H_\bot) = H_\to(F, F) = W = H_\neg(F).$$

Um auch allgemeine Aussagen machen zu können — etwa $\forall x(e \circ x = x)$ —, benötigen wir noch Variablen $x, y, z, \ldots$. Dabei stellen wir uns vor, daß die Variablen über die Elemente unseres Individuenbereichs laufen. Der Termbegriff ist dann so zu erweitern, daß auch Variablen darin zugelassen sind.

Mit A ist auch die Allformel $\forall x A$ (gelesen "für alle x gilt A") eine Formel. Hierbei wird im allgemeinen A eine Formel sein, in der die Variable x vorkommt.

Schließlich ist mit A auch die Existenzformel $\exists x A$ (gelesen "es gibt ein x mit A") eine Formel. Man beachte, daß bei der üblichen *klassischen* (oder schwachen) Auffassung des "es gibt" $\exists x A$ dasselbe bedeutet wie $\neg\forall x \neg A$, also $\exists$ auch durch $\forall$ und $\neg$ definiert werden kann.

Man beachte ferner, daß die Verwendung der Quantoren $\forall x$ und $\exists x$ die Unterscheidung zwischen freien und gebundenen Vorkommen von Variablen notwendig macht. So ist etwa in $\forall x(x \leq x + y)$ das zweite und dritte Vorkommen der Variablen x gebunden, aber das einzige Vorkommen der Variablen y frei. Die letzte Formel kann man unter Erhalt ihrer Bedeutung auch umschreiben in $\forall z(z \leq z + y)$, denn dies ist nur eine Umbenennung der gebundenen Variablen. Ein ähnliches Phänomen ist aus der Analysis bekannt, wo ebenfalls

$$\int_0^1 \sin(xy)\,dx = \int_0^1 \sin(ty)\,dt$$

ist; x bzw. t kommen hier gebunden vor.

Wir geben jetzt einige Beispiele zur Formalisierung mathematischer Aussagen.

1.3.1. Beispiel. Als erstes betrachten wir die Axiome der Gruppentheorie. Grundbegriffe sind die Gruppenverknüpfung $\circ$ (ein zweistelliges Funktionssymbol), die Einheit e (eine Konstante), die Inversenbildung $^{-1}$ (ein einstelliges Funktionssymbol) und schließlich die Gleichheit $=$ (ein zweistelliges Relationssymbol). Als Axiome haben wir

$$\forall x \forall y \forall z[x \circ (y \circ z) = (x \circ y) \circ z] \qquad \text{(Assoziativität)},$$
$$\forall x[e \circ x = x] \qquad \text{(Linkseinheit)},$$
$$\forall x[x^{-1} \circ x = e] \qquad \text{(Linksinverses)}.$$

1.3.2. Beispiel. Zwei Mengen sind gleich, wenn sie dieselben Elemente haben.

$\forall$	für alle Mengen
$\exists$	es gibt eine Menge
$\in$	ist ein Element von
$=$	ist gleich

$$\forall x \forall y(\forall z[z \in x \leftrightarrow z \in y] \to x = y).$$

1.3.3. BEISPIEL. f ist stetig im Punkt a

f	einstelliges Funktionssymbol
a, 0	reelle Zahlen
$\forall$	für alle reelle Zahlen
$\exists$	es gibt eine reell Zahl
$>$ ($<$)	ist größer (kleiner) als
$d(x,y)$	der Abstand von x und y

$$\forall\varepsilon(\varepsilon > 0 \to \exists\delta[\delta > 0 \wedge \forall x(d(x,a) < \delta \to d(f(x),f(a)) < \varepsilon)]).$$

1.3.4. BEISPIEL.

$\forall$	für alle positiven rationalen Zahlen
$\exists$	es gibt eine positive rationale Zahl
$+$	Addition
$\cdot$	Multiplikation
$\leq$	ist kleiner oder gleich
$2xy$	verdoppeltes Produkt von x und y

Es gibt eine gemeinsame obere Schranke für Summe und Produkt zweier positiver rationaler Zahlen. Dieses Beispiel wird später dazu dienen, einige Aspekte des Paradigmas "Beweise als Programme" zu demonstrieren. {*schranke*}

$$\forall x\forall y\exists z(x + y \leq z \wedge x \cdot y \leq z).$$

Für positive rationale Zahlen $x \leq 1$ und y ist $y + 1$ eine gemeinsame obere Schranke für ihre Summe und ihr Produkt.

$$\forall x\forall y[x \leq 1 \to (x + y \leq y + 1 \wedge x \cdot y \leq y + 1)].$$

Für positive rationale Zahlen $x, y > 1$ ist $2xy$ eine gemeinsame obere Schranke für ihre Summe und ihr Produkt.

$$\forall x\forall y[x \not\leq 1 \to y \not\leq 1 \to (x + y \leq 2xy \wedge x \cdot y \leq 2xy)].$$

1.3.5. BEISPIEL. Wir betrachten das folgende *Wall Street Problem*. Nehmen wir an, wir haben eine Million Dollar zur Verfügung. Wir wollen feststellen, in welchem Zeitintervall wir diese Summe am besten in Aktien investiert hätten. Das soll heißen, in welchem Zeitintervall hätten wir den höchsten Spekulationsgewinn gemacht. Zur Verfügung stehen uns die täglichen Angaben über den Dow–Jones Index. (Mit diesem Problem werden wir uns später noch genauer befassen. Im Moment wollen wir nur die Aufgabe durch eine logische Formel spezifizieren.) {*maxseg*}

Sei x_i die Differenz der Dow–Jones Indizes der Tage $i + 1$ und i. Dann haben wir folgendes Maximalsegmentproblem zu betrachten. Gegeben sei eine Folge $x_0, x_1, \ldots, x_n, \ldots$ ganzer Zahlen, also $x_i \in \mathbb{Z}$. Dann gibt es ein maximales Segment $x_i + \cdots + x_k$ (mit $i \leq k \leq n$).

$\forall$	für alle natürlichen Zahlen
$\exists$	es gibt eine natürliche Zahl
$\leq$	ist kleiner oder gleich
$\mathrm{seg}(i,k)$	Segment $x_i + \cdots + x_k$

$$\forall n\exists i,k(i \leq k \leq n \wedge \forall i',k'[i' \leq k' \leq n \to \mathrm{seg}(i',k') \leq \mathrm{seg}(i,k)]).$$

Hier haben wir zur Abkürzung z.B. $\exists i,k$ für $\exists i\exists k$ und $i \leq k \leq n$ für $i \leq k \wedge k \leq n$ geschrieben.

1.3.6. AUFGABE. Man formalisiere die folgenden Aussagen, unter Verwendung von:

$\forall$	für alle Mengen
$\exists$	es gibt eine Menge
$\in$	ist ein Element von
$=$	ist gleich

a. Es gibt eine Menge ohne Elemente.

b. Zu je zwei Mengen x_1, x_2 gibt es eine Menge x, deren Elemente genau x_1 und x_2 sind.

c. Zu jeder Menge x gibt es eine Menge y, die aus den Elementen aller Elemente von x besteht.

d. Zu jeder Menge x gibt es eine Menge y, deren Elemente genau die Teilmengen von x sind.

1.3.7. AUFGABE. Man formalisiere die folgenden Aussagen

$\forall$	für alle Personen
a	Agatha
b	Butler
c	Charles
U	hat umgebracht
R	ist reicher als
H	haßt

a. Agatha, der Butler oder Charles haben Agatha umgebracht.

b. Man bringt nur solche Personen um, die man haßt und die reicher sind als man selbst.

c. Charles mag (= haßt nicht) alle Personen, die Agatha haßt.

d. Agatha haßt sich selbst und Charles.

e. Der Butler haßt alle Personen, die nicht reicher sind als Agatha.

f. Jeden, den Agatha haßt, haßt auch der Butler.

g. Niemand haßt sowohl Agatha als auch den Butler als auch Charles.

1.3.8. AUFGABE. Man zeige, daß jede aussagenlogische Funktion $H\colon \{W,F\}^n \to \{W,F\}$ (mit $n \geq 1$) aus $H_\neg$, $H_\wedge$ und $H_\vee$ explizit definierbar ist. Hinweis: Induktion über n.

1.4 Terme und Formeln

$\{_{TuF}\}$

Wir geben jetzt eine genaue Beschreibung der Sprache der Logik erster Stufe.

Gegeben sei eine abzählbar unendliche Menge von *Variablen* $*_0, *_1, \ldots$, die wir mit x, y, z bezeichnen. Eine *Sprache $\mathcal{L}$ erster Stufe* ist bestimmt durch ihre *Signatur*. Darunter versteht man folgendes.

- Für jede natürliche Zahl $n \geq 0$ eine (eventuell leere) Menge $\mathrm{REL}_{\mathcal{L}}^{(n)}$ von n–stelligen *Relationssymbolen* (oder auch *Prädikatensymbolen*). 0–stellige Relationssymbole heißen *Aussagensymbole*.

- Für jede natürliche Zahl $n \geq 0$ eine (eventuell leere) Menge $\textsc{Fun}_{\mathcal{L}}^{(n)}$ von n–stelligen *Funktionssymbolen*. Die 0–stelligen Funktionssymbole heißen *Konstante*.

Wir setzen voraus, daß die Menge der Variablen und alle Mengen $\textsc{Rel}_{\mathcal{L}}^{(n)}$ und $\textsc{Fun}_{\mathcal{L}}^{(n)}$ paarweise disjunkt sind.

Zum Beispiel ist die Sprache $\mathcal{L}_G$ der Gruppentheorie bestimmt durch folgende Signatur. Die Gruppenverknüpfung $\circ$ (ein zweistelliges Funktionssymbol), die Einheit e (eine Konstante), die Inversenbildung $^{-1}$ (ein einstelliges Funktionssymbol) und schließlich die Gleichheit $=$ (ein zweistelliges Relationssymbol).

$\mathcal{L}$–Terme werden induktiv wie folgt definiert.

- Jede Variable ist ein $\mathcal{L}$–Term.
- Jede Konstante von $\mathcal{L}$ ist ein $\mathcal{L}$–Term.
- Sind $t_1, \ldots, t_n$ $\mathcal{L}$–Terme und ist f ein n–stelliges Funktionssymbol von $\mathcal{L}$ mit $n \geq 1$, so ist $f(t_1, \ldots, t_n)$ ein $\mathcal{L}$–Term.

Mit $\textsc{Ter}_{\mathcal{L}}$ bezeichnen wir die Menge der $\mathcal{L}$–Terme. Für jeden $\mathcal{L}$–Term t die Menge $\mathrm{vars}(t)$ aller Variablen, die in dem Term auftreten, wie folgt definiert.

- Ist t eine Variable, so ist $\mathrm{vars}(t) := \{t\}$.
- Ist t eine Konstante von $\mathcal{L}$, so ist $\mathrm{vars}(t) := \emptyset$.
- Sind $t_1, \ldots, t_n$ $\mathcal{L}$–Terme und ist f ein n–stelliges Funktionssymbol von $\mathcal{L}$, so ist

$$\mathrm{vars}(f(t_1, \ldots, t_n)) := \mathrm{vars}(t_1) \cup \ldots \cup \mathrm{vars}(t_n).$$

Ist $\mathrm{vars}(t) = \emptyset$, so heißt der Term t *geschlossener Term* oder *Grundterm*.

Zweistellige Funktionssymbole werden meist in *Infix*–Schreibweise verwendet. Man schreibt dann $x + y$ statt $+(x, y)$. Klammern werden so gesetzt, daß sich die Struktur eines Terms eindeutig ergibt. Statt $f(t_1, \ldots, t_n)$ schreiben wir oft auch $f t_1 \ldots t_n$. Terme der Sprache der Gruppentheorie, also $\mathcal{L}_G$–Terme, sind dann zum Beispiel

$$e, \quad x \circ e, \quad (x^{-1} \circ y^{-1})^{-1}.$$

Aus $\mathcal{L}$–Termen erhält man jetzt sehr einfach *$\mathcal{L}$–Primformeln* oder *atomare Formeln* von $\mathcal{L}$: Sind $t_1, \ldots, t_n$ $\mathcal{L}$–Terme und ist R ein n–stelliges Relationssymbol von $\mathcal{L}$, so ist $R(t_1, \ldots, t_n)$ eine $\mathcal{L}$–Primformel; ferner ist $\bot$ (gelesen "das Falsche") eine $\mathcal{L}$–Primformel.

Auch zweistellige Relationssymbole werden meist in *Infix*–Schreibweise verwendet. Man schreibt dann $x < y$ statt $<(x, y)$. Klammern werden so gesetzt, daß sich die Struktur einer atomaren Formel eindeutig ergibt. Statt $R(t_1, \ldots, t_n)$ schreiben wir oft $R t_1 \ldots t_n$.

$\mathcal{L}$–Formeln werden aus $\mathcal{L}$–Primformeln wieder induktiv definiert durch

- Jede $\mathcal{L}$–Primformel ist eine $\mathcal{L}$–Formel. Insbesondere ist also $\bot$ eine $\mathcal{L}$–Formel.
- Sind A und B $\mathcal{L}$–Formeln, so auch $A \wedge B$ und $A \to B$.
- Ist A eine $\mathcal{L}$–Formel und x eine Variable, so ist $\forall x A$ eine $\mathcal{L}$–Formel.

Für jede $\mathcal{L}$–Formel A die Menge $\mathrm{FV}(A)$ aller in A *freien Variablen* wie folgt definiert.

$$\begin{aligned}
\mathrm{FV}(R(t_1,\ldots,t_n)) &:= \mathrm{vars}(t_1)\cup\ldots\cup\mathrm{vars}(t_n),\\
\mathrm{FV}(\bot) &:= \emptyset,\\
\mathrm{FV}(A\wedge B) &:= \mathrm{FV}(A)\cup\mathrm{FV}(B),\\
\mathrm{FV}(A\to B) &:= \mathrm{FV}(A)\cup\mathrm{FV}(B),\\
\mathrm{FV}(\forall x A) &:= \mathrm{FV}(A)\setminus\{x\}.
\end{aligned}$$

Falls $\mathrm{FV}(A)=\emptyset$, so heißt A *geschlossene Formel* oder *Satz*. Wir schreiben $A[\vec{x}]$, um mitzuteilen, daß alle freien Variablen von A in der Liste $\vec{x}$ enthalten sind.

Wir beschränken uns hier der Einfachheit halber auf die logischen Symbole $\bot$, $\wedge$, $\to$ und $\forall$, da Negation, Disjunktion, Äquivalenz und der Existenzquantor daraus definierbar sind, und zwar wie folgt.

$$\begin{aligned}
\neg A &:= A\to\bot,\\
A\vee B &:= \neg(\neg A\wedge\neg B),\\
A\leftrightarrow B &:= (A\to B)\wedge(B\to A),\\
\exists x A &:= \neg\forall x\neg A.
\end{aligned}$$

Wir schreiben $t\neq s$ für $\neg(t=s)$ und $t\not< s$ für $\neg(t<s)$. Klammern werden so gesetzt, daß sich die Struktur einer Formel eindeutig ergibt. Um Klammern zu sparen, vereinbaren wir folgendes.

1. Der Wirkungsbereich von $\neg$ und $\forall x$ ist so klein wie möglich. Beispiele:

$$\begin{aligned}
\neg A\wedge B \quad &\text{meint} \quad (\neg A)\wedge B, \quad &&\text{nicht} \quad \neg(A\wedge B).\\
\forall x A\to B \quad &\text{meint} \quad (\forall x A)\to B, \quad &&\text{nicht} \quad \forall x(A\to B).
\end{aligned}$$

2. Der Wirkungsbereich von $\wedge$ und $\vee$ ist so klein wie möglich, unter Berücksichtigung von 1. Beispiel:

$$\neg A\wedge B\to C \quad \text{meint} \quad ((\neg A)\wedge B)\to C.$$

3. Bei wiederholter Anwendung einer Verknüpfung ist stets Rechtsklammerung gemeint. Beispiel:

$$A\to B\to C \quad \text{meint} \quad A\to(B\to C).$$

4. Schreibt man hinter $\forall x$ einen Punkt, so ist der Wirkungsbereich von $\forall x$ ist so groß wie möglich. Beispiel:

$$\forall x.A\to B \quad \text{meint} \quad \forall x(A\to B), \quad \text{nicht} \quad (\forall x A)\to B.$$

Als Mitteilungszeichen verwenden wir (auch mit Indizes)

r,s,t	für Terme
x,y,z	für Variablen
c	für Konstanten
P,Q,R	für Relationssymbole
f,g,h	für Funktionssymbole
A,B,C,D	für Formeln

Im folgenden sei die Sprache $\mathcal{L}$ fest gewählt. Wir lassen deshalb den Bezug auf $\mathcal{L}$ weg und sprechen kurz von Termen und Formeln statt von $\mathcal{L}$–Termen und $\mathcal{L}$–Formeln.

Kapitel 2

Natürliche Herleitungen

{Herl}

Wir wollen jetzt den Begriff einer *logischen Herleitung* einer Formel A definieren. Dazu verwenden wir ein System des "natürlichen Schließens", das von Gerhard Gentzen in [5] eingeführt wurde.

2.1 Einführungs– und Beseitigungsregeln

{Regeln}

2.1.1. BEISPIELE. Wir beginnen mit einigen Beispielen für natürliche Beweise. Gegeben sei also eine Sprache $\mathcal{L}$ erster Stufe. Der Einfachheit halber betrachten wir Beweise in der reinen Logik, d.h. ohne Annahmen über die Funktionen und Prädikate.

$$(A \wedge B \to C) \to (A \to (B \to C)). \tag{1}$$

Gelte $A \wedge B \to C$. Zu zeigen: $A \to (B \to C)$. Gelte also A. Zu zeigen: $B \to C$. Gelte also B. Zu zeigen: C. Wir haben $A \wedge B$, nach den letzten beiden Annahmen. Also auch C, nach der ersten Annahme.

$$(A \to (B \to C)) \to (A \wedge B \to C). \tag{2}$$

Gelte $A \to (B \to C)$. Zu zeigen: $A \wedge B \to C$. Gelte also $A \wedge B$. Zu zeigen: C. Wir haben A, nach der letzten Annahme. Also auch $B \to C$, nach der ersten Annahme. Wir haben B, wieder nach der letzten Annahme. Daher auch C, nach den letzten beiden Aussagen.

$$\forall x(A \to B) \to (A \to \forall x B), \quad \text{falls } x \notin \mathrm{FV}(A). \tag{3}$$

Gelte $\forall x(A \to B)$. Zu zeigen: $A \to \forall x B$. Gelte also A. Zu zeigen: $\forall x B$. Sei x beliebig; man beachte, daß wir bisher keine Annahmen über x gemacht haben. Zu zeigen: B. Wir haben $A \to B$, nach der ersten Annahme. Also auch B, nach der zweiten Annahme.

$$(A \to \forall x B) \to \forall x(A \to B), \quad \text{falls } x \notin \mathrm{FV}(A). \tag{4}$$

Gelte $A \to \forall x B$. Zu zeigen: $\forall x(A \to B)$. Sei x beliebig; man beachte, daß wir bisher keine Annahmen über x gemacht haben. Zu zeigen: $A \to B$. Gelte also A. Zu zeigen: B. Wir haben $\forall x B$, nach der ersten und zweiten Annahme. Also auch B.

Ein Charakteristikum dieser Beweise ist es, daß Annahmen eingeführt und wieder beseitigt werden: Zu jedem Zeitpunkt im Beweis kennt man die jetzt freien Annahmen.

Wir reservieren das Wort *Beweis* für die Meta–Stufe; eine formale Darstellung eines Beweises nennen wir *Herleitung* oder *Ableitung*.

2.1.2. Eine anschauliche Art, Herleitungen zu definieren, besteht darin, sie als beschriftete Bäume aufzufassen. Die Beschriftungen der inneren Knoten sind Formeln, die der Blätter sind Formeln oder Terme. Die Beschriftungen der Nachfolger eines Knotens ν sind die *Prämissen* einer Regelanwendung, die Formel am Konten ν ist ihre *Konklusion*. An der Wurzel des Baums befindet sich die Konklusion der gesamten Herleitung. Im natürlichen Schließen arbeitet man mit *Annahmen*, die an Blättern des Baums stehen; sie können offen oder geschlossen (man sagt auch: *gestrichen*) sein.

Jede dieser Annahmen ist mit einer *Marke* versehen. Als Marken verwenden wir *Annahmenvariablen* $\Box_0, \Box_1, \ldots$; Mitteilungszeichen für Annahmenvariablen sind u, v, w. Die (bisherigen) Variablen nennen wir oft auch *Objektvariablen*, um sie von den Annahmenvariablen zu unterscheiden. Wenn an einer späteren Stellen (d.h. an einem Knoten im Baum unterhalb dieser Annahme) die Abhängigkeit von dieser Annahme beseitigt wird, notieren wir dies durch Angabe der Annahmenvariablen. Da wir dieselbe Annahme auch mehrfach verwenden können (dies war etwa im Beispiel (2) der Fall), darf in einem Baum eine mit u markierte Annahme A (mitgeteilt durch $u\colon A$) auch mehrfach vorkommen. Wir verlangen jedoch, daß verschiedene Annahmeformeln stets verschiedene Marken bekommen.

Einen inneren Knoten im Baum verstanden wir als Resultat eines Übergangs von *Prämissen* zu einer *Konklusion*. Die zulässigen Übergänge werden durch die *Regeln* bestimmt. Die Beschriftung des Knotens enthält dann neben der Konklusion noch den Namen der verwendeten Regel. In manchen Fällen bindet eine Regel eine Annahmenvariable u (und beseitigt damit die Abhängigkeit von allen darüber stehenden, mit u markierten Annahmen $u\colon A$) oder eine Objektvariable x (und beseitigt damit die Abhängigkeit von x). Dann wird die abgebundene Annahmen– oder Objektvariable der Beschriftung hinzugefügt.

2.1.3. *Regeln des natürlichen Schließens.* Die Regeln des natürlichen Schließens gliedern sich in Einführungs– und Beseitigungsregeln. Für die Konjuktion $\wedge$ haben wir eine Einführungsregel $\wedge^+$ und zwei Beseitigungsregeln $\wedge_0^-$ und $\wedge_1^-$. $\{_{regeln}\}$

$$\frac{\begin{array}{cc}\mathcal{D}_0 & \mathcal{D}_1\\ A & B\end{array}}{A \wedge B}\,\wedge^+ \qquad \frac{\begin{array}{c}\mathcal{D}\\ A \wedge B\end{array}}{A}\,\wedge_0^- \qquad \frac{\begin{array}{c}\mathcal{D}\\ A \wedge B\end{array}}{B}\,\wedge_1^-$$

Für die Implikation $\to$ gibt es eine Einführungsregel $\to^+ u$ und eine Beseitigungsregel $\to^-$, die man auch *modus ponens* nennt. Die linke Prämisse $A \to B$ in $\to^-$ nennt man *Hauptprämisse*, die rechte Prämisse A *Nebenprämisse*. Man beachte, daß bei Anwendung einer $\to^+ u$–Regel *alle* darüber stehenden mit u markierten Annahmen A gestrichen werden.

$$\frac{\begin{array}{c}[u\colon A]\\ \mathcal{D}\\ B\end{array}}{A \to B}\,\to^+ u \qquad \frac{\begin{array}{cc}\mathcal{D}_0 & \mathcal{D}_1\\ A \to B & A\end{array}}{B}\,\to^-$$

Für den Allquantor $\forall$ gibt es eine Einführungsregel $\forall^+ x$ und eine Beseitigungsregel $\forall^-$, die als rechte Prämisse den zu substituierenden Term t hat. Die Regel $\forall^+ x$ unterliegt der folgenden *Variablenbedingung*: Die Herleitung $\mathcal{D}$ der Prämisse A darf keine offenen Annahmen enthalten, in denen x frei vorkommt.

$$\frac{\begin{array}{c}\mathcal{D}\\ A\end{array}}{\forall x A}\,\forall^+ x \qquad \frac{\begin{array}{cc}\mathcal{D} & \\ \forall x A & t\end{array}}{A[t/x]}\,\forall^-$$

2.1.4. BEISPIELE. Wir geben Herleitungen für die oben bewiesenen Formeln (1)–(4) an. Da meist die verwendete Regel durch die an den Knoten stehenden Formeln bestimmt ist, verzichten wir im allgemeinen auf die Angabe der Regel.

(1)

$$\dfrac{\dfrac{u\colon A\wedge B\to C \qquad \dfrac{\dfrac{v\colon A \qquad w\colon B}{A\wedge B}}{C}\,w}{B\to C}\,v}{A\to(B\to C)}\,u$$
$$(A\wedge B\to C)\to(A\to(B\to C))$$

(2)

$$\dfrac{\dfrac{u\colon A\to(B\to C) \qquad \dfrac{v\colon A\wedge B}{A}}{B\to C} \qquad \dfrac{v\colon A\wedge B}{B}}{\dfrac{C}{A\wedge B\to C}\,v}\,u$$
$$(A\to(B\to C))\to(A\wedge B\to C)$$

(3)

$$\dfrac{\dfrac{u\colon \forall x(A\to B) \qquad x}{A\to B} \qquad v\colon A}{\dfrac{\dfrac{B}{\forall x B}\,x}{A\to\forall x B}\,v}\,u$$
$$\forall x(A\to B)\to(A\to\forall x B)$$

Hier ist zu beachten, daß die Variablenbedingung erfüllt ist: x kommt nicht frei in A (und auch nicht frei in $\forall x(A\to B)$) vor.

(4)

$$\dfrac{\dfrac{u\colon A\to\forall x B \qquad v\colon A}{\forall x B} \qquad x}{\dfrac{\dfrac{B}{A\to B}\,v}{\forall x(A\to B)}\,x}\,u$$
$$(A\to\forall x B)\to\forall x(A\to B)$$

Auch hier ist die Variablenbedingung erfüllt: x kommt nicht frei in A vor.

2.1.5. SCHREIBWEISE. Wir schreiben $\vdash A$ und nennen A *herleitbar* (in der *Minimallogik*), wenn es eine Herleitung von A ohne freie Annahmen gibt. Eine Formel B heißt herleitbar aus den Annahmen $A_1,\dots,A_n$, wenn es eine Herleitung mit freien Annahmen unter $A_1,\dots,A_n$ gibt. Sei S eine (endliche oder unendliche) Menge von Formeln. Wir schreiben $S\vdash B$, wenn die Formel B aus endlich vielen Annahmen $A_1,\dots,A_n\in S$ herleitbar ist.

2.1.6. AUFGABE. Man zeige

a. $\vdash A\to\neg\neg A$

b. $\vdash\neg\neg\neg A\to\neg A$

c. $\vdash\neg\neg(A\wedge B)\leftrightarrow(\neg\neg A\wedge\neg\neg B)$

d. $\vdash \neg\neg(A \to B) \to (\neg\neg A \to \neg\neg B)$

e. $\vdash \neg\neg\forall x\, A \to \forall x \neg\neg A$

2.2 Herleitungen als Terme

$\{HTerme\}$

Herleitungen kann man in natürlicher Weise als getypte Terme schreiben, sog. Herleitungsterme. Jeder Herleitung $\mathcal{D}$ entspricht dabei ein Herleitungsterm d.

Als Grundbausteine von Herleitungstermen verwenden wir mit Formeln versehene Annahmenvariable $u\colon A$; u ist wieder eines der Symbole $\square_0, \square_1, \ldots$. Wir verlangen wie bisher, daß verschiedene Annahmeformeln stets verschiedene Marken u haben. Als Mitteilungszeichen für Herleitungsterme verwenden wir d, e, f. Oft schreiben wir auch u^A anstelle von $u\colon A$ und d^A anstelle von $d\colon A$.

Herleitung	Term
u^A	u^A
$\dfrac{\begin{array}{cc}\mathcal{D}_0 & \mathcal{D}_1\\ A & B\end{array}}{A \wedge B}\,\wedge^+$	$\langle d_0^A, d_1^B\rangle^{A\wedge B}$
$\dfrac{\mathcal{D}}{\dfrac{A \wedge B}{A}\,\wedge_0^-}$	$\pi_0(d^{A\wedge B})^A$
$\dfrac{\mathcal{D}}{\dfrac{A \wedge B}{B}\,\wedge_1^-}$	$\pi_1(d^{A\wedge B})^B$
$\dfrac{\begin{array}{c}[u\colon A]\\ \mathcal{D}\\ B\end{array}}{A \to B}\,\to^+ u$	$(\lambda u^A d^B)^{A\to B}$
$\dfrac{\begin{array}{cc}\mathcal{D}_0 & \mathcal{D}_1\\ A \to B & A\end{array}}{B}\,\to^-$	$(d^{A\to B} e^A)^B$
$\dfrac{\begin{array}{c}\mathcal{D}\\ A\end{array}}{\forall x A}\,\forall^+ x$ (Var.bed.)	$(\lambda x d^A)^{\forall x A}$ (mit Var.bed.)
$\dfrac{\begin{array}{cc}\mathcal{D} & \\ \forall x A & t\end{array}}{A[t/x]}\,\forall^-$	$(d^{\forall x A} t)^{A[t/x]}$

BEMERKUNG. Man beachte, daß in Herleitungstermen viele Formeln überflüssig sind. Wenn man $(\lambda u d)^{A\to B}$ schreibt, so ist klar, daß u die Formel A und d die Formel B haben müssen. Ebenso muß in $(d^{A\to B} e)$ der Gesamtterm die Formel B und e die Formel A haben. Wir werden im folgenden häufig solche redundanten Formeln weglassen. Auf diese Weise erhält man mit den Herleitungstermen eine sehr kompakte Schreibweise für Herleitungen.

2.2.1. BEISPIELE. Wir schreiben die am Anfang von Abschnitt 2.1 gegebenen Beispielsherleitungen als Herleitungsterme auf.

(1) Mit $F := (A \wedge B \to C) \to (A \to (B \to C))$ haben wir

$$\left\{\lambda u^{A\wedge B\to C}\left(\lambda v^{A}\left[\lambda w^{B}\left(u^{A\wedge B\to C}\langle v^{A}, w^{B}\rangle^{A\wedge B}\right)^{C}\right]^{B\to C}\right)^{A\to(B\to C)}\right\}^{F}$$

Es ist klar, daß man hier viele redundante Formeln weglassen kann Wir verwenden Annahmenvariablen u, v, w mit den Typen

u	$A \wedge B \to C$
v	A
w	B

Der Herleitungsterm ist dann

$$\lambda u \lambda v \lambda w (u\langle vw\rangle).$$

(2) Wir verwenden Annahmenvariablen u, v mit den Typen

u	$A \to (B \to C)$
v	$A \wedge B$

Der Herleitungsterm ist dann

$$\lambda u \lambda v [u(\pi_0 v)](\pi_1 v).$$

(3) Wir verwenden Annahmenvariablen u, v mit den Typen

u	$\forall x(A \to B)$
v	A

Der Herleitungsterm ist dann

$$\lambda u \lambda v \lambda x [(ux)v].$$

(4) Wir verwenden Annahmenvariablen u, v mit den Typen

u	$A \to \forall x B$
v	A

Der Herleitungsterm ist dann

$$\lambda u \lambda x \lambda v [(uv)x].$$

2.2.2. *Annahmenvariablen und freie Variablen einer Herleitung.* Die Menge der in einer Herleitung $\mathcal{D}$ freien Annahmen bezeichnen wir mit $\mathrm{FA}(\mathcal{D})$. Die genaue Definition ist induktiv; wir schreiben sie für Herleitungsterme auf:

DEFINITION.

$$\begin{aligned}
\mathrm{FA}(u^A) &:= \{u^A\},\\
\mathrm{FA}(\langle d^A, e^B\rangle^{A\wedge B}) &:= \mathrm{FA}(d^A) \cup \mathrm{FA}(e^B),\\
\mathrm{FA}(\pi_0(d^{A\wedge B})^A) &:= \mathrm{FA}(d^{A\wedge B}),\\
\mathrm{FA}(\pi_1(d^{A\wedge B})^B) &:= \mathrm{FA}(d^{A\wedge B}),\\
\mathrm{FA}((\lambda u^A d^B)^{A\to B}) &:= \mathrm{FA}(d^B) \setminus \{u^A\},\\
\mathrm{FA}((d^{A\to B} e^A)^B) &:= \mathrm{FA}(d^{A\to B}) \cup \mathrm{FA}(e^A),\\
\mathrm{FA}((\lambda x d^A)^{\forall x A}) &:= \mathrm{FA}(d^A),\\
\mathrm{FA}((d^{\forall x A} t)^{A[t/x]}) &:= \mathrm{FA}(d)
\end{aligned}$$

Ein Herleitungsterm d^A heißt *geschlossen*, wenn $\mathrm{FA}(d^A) = \emptyset$. Wir schreiben

$$d^B[u_1^{A_1}, \ldots, u_n^{A_n}]$$

um mitzuteilen, daß die freien Annahmen von d^B in der Liste $u_1^{A_1}, \ldots, u_n^{A_n}$ enthalten sind.

Ferner benötigt man oft die Menge $\mathrm{FV}(\mathcal{D})$ von *freien* (Objekt–) *Variablen* in einer Herleitung $\mathcal{D}$.

BEISPIEL. Man betrachte die folgende Herleitungen.

(i) $\mathcal{D}_1$.

$$\frac{\begin{array}{c}u\colon Rx\\ \vdots\\ P \to Rx\end{array}}{v}\,{\to^+}$$

Dann ist $\mathrm{FA}(\mathcal{D}_1) = \{u^{Rx}\}$, $\mathrm{FV}(\mathcal{D}_1) = \{x\}$.

(ii) $\mathcal{D}_2$.

$$\frac{\dfrac{\begin{array}{c}u\colon Rx\\ \vdots\\ P \to Rx\end{array}}{v}\,{\to^+}}{u}\,{\to^+}$$

Dann ist $\mathrm{FA}(\mathcal{D}_2) = \emptyset$, $\mathrm{FV}(\mathcal{D}_2) = \{x\}$.

(iii) $\mathcal{D}_3$.

$$\frac{\dfrac{\dfrac{\begin{array}{c}u\colon Rx\\ \vdots\\ P \to Rx\end{array}}{v}\,{\to^+}}{u}\,{\to^+}}{\forall x(Rx \to (P \to Rx))}\,{\forall^+ x}$$

Dann ist $\mathrm{FA}(\mathcal{D}_3) = \emptyset$, $\mathrm{FV}(\mathcal{D}_3) = \emptyset$.

Die genaue Definition ist wieder induktiv; wir schreiben sie für Herleitungsterme auf:

DEFINITION.

$$\begin{aligned}
\mathrm{FV}(u^A) &:= \mathrm{FV}(A).\\
\mathrm{FV}(\langle d^A, e^B\rangle) &:= \mathrm{FV}(d^A) \cup \mathrm{FV}(e^B).\\
\mathrm{FV}(\pi_i(d^{A\wedge B})) &:= \mathrm{FV}(d^{A\wedge B}).\\
\mathrm{FV}(\lambda u^A d^B) &:= \mathrm{FV}(A) \cup \mathrm{FV}(d^B).\\
\mathrm{FV}(d^{A\to B} e^A) &:= \mathrm{FV}(d^{A\to B}) \cup \mathrm{FV}(e^A).\\
\mathrm{FV}(\lambda x d^A) &:= \mathrm{FV}(d^A) \setminus \{x\}.\\
\mathrm{FV}(d^{\forall x A} t) &:= \mathrm{FV}(d^{\forall x A}) \cup \mathrm{vars}(t).
\end{aligned}$$

BEMERKUNG. Für Herleitungsterme haben wir zwei Arten von Substitutionen: Man kann einen Herleitungsterm f^A für eine Annahmenvariable u^A substituieren, und man kann auch einen Objektterm t für eine Objektvariable x substituieren.

2.3 Klassische und intuitionistische Logik

{*Klass*}

Folgende Frage liegt nahe: haben wir logische Regeln "vergessen"? Die Antwort ist "Ja". In den bisher behandelten Regeln spielte nämlich das falsum $\bot$ keine Sonderrolle, sondern wurde behandelt wie ein gewöhnliches Aussagensymbol.

2.3.1. Wir erhalten die *intuitionistische Logik* durch Hinzunahme des folgenden *Ex–Falso–Quodlibet* Axioms EFQ_R für jedes Relationssymbol R:

$$\forall \vec{x}(\bot \to R\vec{x}). \tag{EFQ_R}$$

Ähnlich erhält man die *klassische Logik*: wir nehmen für jedes Relationssymbol R das *Prinzip des indireken Beweisens* für R als Axiom hinzu, also die Formel

$$\forall \vec{x}(\neg\neg R\vec{x} \to R\vec{x}); \tag{STAB_R}$$

diese Formel bezeichnet man auch als *Stabilität* von R.

2.3.2. SCHREIBWEISE. Sei

$$\begin{aligned} \text{EFQ} &:= \{\, \text{EFQ}_R \mid R \text{ Relationssymbol} \,\}, \\ \text{STAB} &:= \{\, \text{STAB}_R \mid R \text{ Relationssymbol} \,\}. \end{aligned}$$

Wir nennen die Formel A *klassisch (intuitionistisch) herleitbar* und schreiben $\vdash_c A$ ($\vdash_i A$), wenn es eine Herleitung von A aus Stabilitätsannahmen STAB_R (Ex–Falso–Quodlibet Annahmen EFQ_R) gibt. Ebenso nennen wir die Formel A klassisch (intuitionistisch) herleitbar aus S und schreiben $S \vdash_c A$ ($S \vdash_i A$), wenn es eine Herleitung von A aus S und Stabilitätsannahmen STAB_R (Ex–Falso–Quodlibet Annahmen EFQ_R) gibt. Wir schreiben also

$$\begin{aligned} S \vdash_i A \quad &\text{für} \quad S \cup \text{EFQ} \vdash A, \\ S \vdash_c A \quad &\text{für} \quad S \cup \text{STAB} \vdash A. \end{aligned}$$

2.3.3. LEMMA. *(Ex–falso–quodlibet). Für jede Formel A gilt $\vdash_i \bot \to A$.* {*efq*}

BEWEIS. Durch Induktion über A konstruieren wir für jede Formel A eine Herleitung $\mathcal{D}_A$ von $\bot \to A$. In den konstruierten Herleitungen lassen wir der Kürze halber Anwendungen von $\to^+$ am Schluß fort.

Fall $R(\vec{t})$. Mit EFQ_R.

Fall $\bot$. Klar.

Fall $A \wedge B$.

$$\dfrac{\dfrac{\dfrac{\mathcal{D}_A}{\bot \to A} \quad u\colon \bot}{A} \qquad \dfrac{\dfrac{\mathcal{D}_B}{\bot \to B} \quad u\colon \bot}{B}}{\dfrac{A \wedge B}{\bot \to A \wedge B}\ \to^+ u}$$

Fall $A \to B$.

$$\dfrac{\dfrac{\dfrac{\mathcal{D}_B}{\bot \to B} \qquad u\colon \bot}{B}}{\dfrac{A \to B}{\bot \to A \to B}} \to^+ u$$

Fall $\forall x\, A$.

$$\dfrac{\dfrac{\dfrac{\mathcal{D}_A}{\bot \to A} \qquad u\colon \bot}{A}}{\dfrac{\forall x\, A}{\bot \to \forall x\, A}} \to^+ u$$

Damit ist der Beweis abgeschlossen. □

2.3.4. LEMMA. *(Stabilität). Für jede Formel A (unserer $\to\wedge\bot\forall$–Sprache) gilt* $\vdash_c \neg\neg A \to A$. $\{_{stab}\}$

BEWEIS durch Induktion über A. In den konstruierten Herleitungen lassen wir der Kürze halber Anwendungen von $\to^+$ am Schluß wieder fort.

Fall $R(\vec{t})$. Mit STAB_R.

Fall $\bot$. Man beachte, daß gilt $\neg\neg\bot \to \bot = ((\bot \to \bot) \to \bot) \to \bot$. Die gesuchte Herleitung ist:

$$\dfrac{v\colon (\bot \to \bot) \to \bot \qquad \dfrac{u\colon \bot}{\bot \to \bot}\, u}{\bot}$$

Fall $A \wedge B$. Mit

$$\vdash (\neg\neg A \to A) \to (\neg\neg B \to B) \to \neg\neg(A \wedge B) \to A \wedge B.$$

Fall $A \to B$. Mit

$$\vdash (\neg\neg B \to B) \to \neg\neg(A \to B) \to A \to B.$$

Eine Herleitung ist

$$\dfrac{u\colon \neg\neg B \to B \qquad \dfrac{\dfrac{v\colon \neg\neg(A \to B) \qquad \dfrac{\dfrac{u_1\colon \neg B \qquad \dfrac{u_2\colon A \to B \qquad w\colon A}{B}}{\bot}\, u_2}{\neg(A \to B)}}{\bot}\, u_1}{\neg\neg B}}{B}$$

Fall $\forall x\, A$. Offenbar genügt es zu zeigen

$$\vdash (\neg\neg A \to A) \to \neg\neg\forall x A \to A.$$

Eine Herleitung ist

$$\dfrac{u\colon \neg\neg A \to A \qquad \dfrac{\dfrac{v\colon \neg\neg\forall x A \qquad \dfrac{\dfrac{u_1\colon \neg A \qquad \dfrac{u_2\colon \forall x A \qquad x}{A}}{\bot}\, u_2}{\neg\forall x A}}{\bot}\, u_1}{\neg\neg A}}{A}$$

Damit ist der Beweis abgeschlossen. □

2.3.5. Bemerkung. $S \vdash A \implies S \vdash_i A \implies S \vdash_c A$.

Beweis. Es genügt zu zeigen, daß $\vdash_c \mathrm{EFQ}_R$. Dies sieht man wie folgt; R sei etwa einstellig.

$$\dfrac{\dfrac{\dfrac{\forall x(\neg\neg Rx \to Rx) \qquad x}{\neg\neg Rx \to Rx} \qquad \dfrac{u\colon \bot}{\neg\neg Rx} \to^+ v^{\neg Rx}}{Rx} \to^+ u}{\bot \to Rx}}{\forall x(\bot \to Rx)} \forall^+$$

Damit ist der Beweis abgeschlossen. □

Die Umkehrungen gelten jedoch nicht; Gegenbeispiele sind:

$$\not\vdash \bot \to P, \quad \text{aber} \quad \vdash_i \bot \to P,$$
$$\not\vdash_i ((P \to Q) \to P) \to P, \quad \text{aber} \quad \vdash_c ((P \to Q) \to P) \to P.$$

2.3.6. Lemma. *(Fallunterscheidung).* $\vdash_c (A \to B) \wedge (\neg A \to B) \to B$.

Beweis.

$$\dfrac{\dfrac{\mathcal{D}_{\mathrm{STAB}}}{\neg\neg B \to B} \qquad \dfrac{\dfrac{w\colon \neg B \qquad \dfrac{\dfrac{u\colon (A \to B) \wedge (\neg A \to B)}{\neg A \to B} \qquad \dfrac{\dfrac{u\colon (A \to B) \wedge (\neg A \to B)}{A \to B} \qquad v\colon A}{B}}{\bot} v}{\neg A}}{B}}{\bot} w}{\neg\neg B}}{B}$$

wobei $\mathcal{D}_{\mathrm{STAB}}$ nach dem Stabilitätslemma 2.3.4 bekannt ist. □

Man beachte, daß sich etwa für das Maximalsegmentproblem ein Beweis noch nicht führen läßt; hierfür braucht man die Induktion. Wir gehen deshalb im nächsten Abschnitt von der reinen Logik zur Arithmetik über.

2.3.7. Aufgabe. Man leite aus den Annahmen

- u_1: $U(a,a) \vee U(b,a) \vee U(c,a)$ (Agatha, der Butler oder Charles haben Agatha umgebracht)
- u_2: $\forall x, y(U(x,y) \to H(x,y) \wedge \neg R(x,y))$ (Man bringt nur solche Personen um, die man haßt und die man an Reichtum nicht übertrifft)
- u_3: $\forall x(H(a,x) \to \neg H(c,x))$ (Charles mag (= haßt nicht) alle Personen, die Agatha haßt)
- u_4: $H(a,a) \wedge H(a,c)$ (Agatha haßt sich selbst und Charles)
- u_5: $\forall x(\neg R(x,a) \to H(b,x))$ (Der Butler haßt alle Personen, die nicht reicher sind als Agatha)
- u_6: $\forall x(H(a,x) \to H(b,x))$ (Jeden, den Agatha haßt, haßt auch der Butler)
- u_7: $\neg\exists x(H(x,a) \wedge H(x,b) \wedge H(x,c))$. (Niemand haßt sowohl Agatha als auch den Butler als auch Charles)

die Formeln (a) $\neg U(c,a)$ und (b) $\neg U(b,a)$ minimallogisch und die Formel (c) $U(a,a)$ klassisch her.

2.3.8. AUFGABE. Man zeige

a. $\vdash_c \neg(B \to C) \to B \wedge \neg C$,

b. $\vdash_c (\neg B \to A) \wedge (C \to A) \to ((B \to C) \to A)$,

c. $\vdash_c (\forall x B \to A) \to \forall y(\neg\forall x B \to \neg B[y/x] \to A) \to A$ für $y \notin \mathrm{FV}(B) \cup \mathrm{FV}(A)$ und y nicht gebunden in B.

Kapitel 3

Arithmetik

{*Arith*}

Beweise, die sich für die Verwendung als Programme eignen, finden nicht in der reinen Logik statt, die ja uninterpretierte Relations– und Funktionssymbole hat. Benötigt wird vielmehr eine "arithmetische" Sprache, deren Symbole in einem "Standardmodell" fest interpretiert sind. Die allgemeinen logischen Herleitungsregeln gelten auch für solche arithmetischen Sprachen, man benötigt aber meist noch zusätzliche Axiome, wie etwa Induktionsaxiome.

3.1 Sprachen der Arithmetik

{*Ind*}

Wir beschreiben in diesem Abschnitt Sprachen, die in einem Standardmodell fest interpretiert sind. Meist wählt man als Standardmodell die Struktur $\mathbb{N}$ der natürlichen Zahlen; dies wollen wir auch hier tun. Man kann aber auch etwa die positiven rationalen Zahlen $\mathbb{Q}^+$ zugrunde legen. Bei der Behandlung des Beispiels über die gemeinsame obere Schranke für Summe und Produkt zweier positiver rationaler Zahlen aus 1.3.4 werden wir von diesem Standardmodell ausgehen.

3.1.1. Eine *arithmetische Sprache* $\mathcal{L}$ ist ein Spezialfall der Sprachen erster Stufe, wie sie in Abschnitt 1.4 eingeführt wurden. Zusätzlich wird folgendes verlangt.

- Jedes Relationssymbol $R \in \textsc{Rel}_{\mathcal{L}}^{(n)}$ stellt eine *entscheidbare* n–stellige Relation $\hat{R} \subseteq \mathbb{N}^n$ dar (z.B. $=$, $<$).
- Jedes Funktionssymbol $f \in \textsc{Fun}_{\mathcal{L}}^{(n)}$ stellt eine *berechenbare* n–stellige Funktion $\hat{f}\colon \mathbb{N}^n \to \mathbb{N}$ dar (z.B. $+$, $\cdot$, exp). Stets sollen die Funktionssymbole 0 für die Zahl 0 und S für die Nachfolgerfunktion vorhanden sein.

Jeder geschlossene Term t hat einen *Wert* $|t| \in \mathbb{N}$; z.B. ist $|\mathrm{S}(\mathrm{S}(0)) + \mathrm{S}(0)| = 3$. Wir setzen

$$\underline{n} := \underbrace{\mathrm{S}(\ldots\mathrm{S}(}_{n\text{–mal}}0)) \quad \text{"}\textit{Ziffer}\text{" } n.$$

Offenbar gilt $|\underline{n}| = n$.

Der Einfachheit halber identifizieren wir geschlossene Terme mit ihrem Wert. Insbesondere wird jeder geschlossene Term t mit der Ziffer $\underline{|t|}$ identifiziert, also zum Beispiel $\mathrm{S}(\mathrm{S}(0)) + \mathrm{S}(0) \equiv \underline{2} + \underline{1}$ mit $\underline{3}$ ($\equiv \mathrm{S}(\mathrm{S}(\mathrm{S}(0)))$). Dies erspart uns lästige Gleichungslogik.

Für jede geschlossene atomare Formel kann man effektiv entscheiden, ob sie wahr oder falsch ist; zum Beispiel ist $\underline{3} + \underline{4} < \underline{5}$ wahr. Wir identifizieren jede zwei geschlossene atomare Formel

mit demselben Wahrheitswert, insbesondere also jede falsche derartige Formel mit dem Falsum $\bot$.

Diese Identifikationen bedeuten für Herleitungen, daß die Regeln "modulo Identifikation" zu lesen sind. Zum Beispiel ist

$$\frac{\begin{array}{cc}\mathcal{D}_0 & \mathcal{D}_1\\ \underline{3} < x \to \underline{3} < x + y & \underline{1} + \underline{2} < x\end{array}}{\underline{3} < x + y}\ \to^-$$

eine korrekte Herleitung.

3.1.2. BEMERKUNG. Anstelle des Standardmodells $\mathbb{N}$ kann man auch etwa $\mathbb{Q}^+$, also die positiven rationalen Zahlen betrachten. Unter den Funktionssymbolen soll immer neben der Null und der Nachfolgerfunktion S die Division vorkommen. Den Ziffern $\underline{n}$ entsprechen (gekürzte) Brüche $\underline{n}/\underline{m}$. Wir werden diese Sprache für die Behandlung des Beispiels aus 1.3.4 benutzen. {*ratstmod*}

3.2 Arithmetische Herleitungen

{*ArithHerl*}

Der Herleitungsbegriff der reinen Logik überträgt sich unverändert auf arithmetische Sprachen. Man benötigt jedoch zusätzliche Axiome, die das intendierte Standardmodell beschreiben.

Besonders zu beachten ist, daß in unserer fest interpretierten Sprache ein *Wahrheitsbegriff* zur Verfügung steht. Wir werden deshalb nur wahre Aussagen als Axiome zulassen.

Die hier zugelassenen Axiome sind von zwei Arten: Π–Axiome und Induktionsaxiome. *Π–Axiome* (oder *All–Axiome*) sind wahre geschlossene Formeln der Gestalt

$$\forall x_1, \ldots, x_n\, A \quad \text{mit quantorenfreiem } A.$$

Beispiele für Π–Axiome:

- Ex–Falso–Quodlibet–Axiome und Stabilitätsaxiome für atomare Formeln, also z.B.

$$\forall x, y(\neg\neg x < y \to x < y).$$

- Mathematische Axiome, z.B.

$$\begin{aligned}&\forall x(\mathrm{S}(x) \neq 0),\\ &\forall x, y(\mathrm{S}(x) = \mathrm{S}(y) \to x = y),\\ &\forall x, y(x + \mathrm{S}(y) = \mathrm{S}(x + y)),\\ &\forall x, y, z(x < y \to y < z \to x < z).\end{aligned}$$

Zur Formulierung der Induktionsaxiome führen wir noch eine suggestive Bezeichnungsweise ein. Wir schreiben $A(x_1, \ldots, x_n)$ für Formeln mit freien Variablen unter $\{x_1, \ldots, x_n\}$. Dann läßt sich das Ergebnis der (simultanen) Substitution von Termen $t_1, \ldots, t_n$ für $x_1, \ldots, x_n$ einfach als $A(t_1, \ldots, t_n)$ schreiben. Ferner schreiben wir oft $t + 1$ für den Nachfolger $\mathrm{S}(t)$.

Induktionsaxiome sind alle Formeln derGestalt

$$\forall \vec{x}\,(A(0, \vec{x}) \to \forall y[A(y, \vec{x}) \to A(y + 1, \vec{x})] \to \forall y A(y, \vec{x}))\,, \qquad (\mathrm{IND}_{A(y,\vec{x})})$$

und zwar für *jede* Formel $A(y, \vec{x})$.

3.2.1. BEISPIEL. Existenz einer angenäherten Quadratwurzel. Zu beweisen ist die Formel $\{_{wurzel}\}$

$$\forall x \exists y(y^2 \leq x < (y+1)^2),$$

also nach Auflösung der Abkürzung für den Existenzquantor

$$\forall x[\forall y(y^2 \leq x \to x < (y+1)^2 \to \bot) \to \bot].$$

Aus typographischen Gründen konstruieren wir zunächst die folgende Herleitung $\mathcal{D}$, die als freie Annahmenvariable noch $u\colon \forall y(y^2 \leq x \to x < (y+1)^2 \to \bot)$ enthält, sowie das (hier nützliche) Axiom $\forall x, y[(x<y \to \bot) \to y \leq x]$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\text{Ax}\colon \forall x, y[(x<y \to \bot) \to y \leq x] \quad x \quad (y+1)^2 \qquad \dfrac{u\colon \forall y(y^2 \leq x \to x < (y+1)^2 \to \bot) \quad y \quad v\colon y^2 \leq x}{x < (y+1)^2 \to \bot}}{(y+1)^2 \leq x}}{y^2 \leq x \to (y+1)^2 \leq x}}{\forall y(y^2 \leq x \to (y+1)^2 \leq x)}$$

Mit $\mathcal{D}$ ergibt sich jetzt die gesuchte Herleitung wie folgt. Hier verwenden wir das Induktionsaxiom

$$\forall x(0^2 \leq x \to \forall y[y^2 \leq x \to (y+1)^2 \leq x] \to \forall y(y^2 \leq x)),$$

sowie

$$\forall x[(x+1)^2 \leq x \to \bot],$$
$$\forall x(0 \leq x^2)$$

als weitere Axiome.

$$\dfrac{\dfrac{\dfrac{\text{Ax}\colon \forall x[(x+1)^2 \leq x \to \bot] \quad x \qquad \dfrac{\dfrac{\text{Ind}_{y^2 \leq x} \quad x \qquad \dfrac{\text{Ax}\colon \forall x(0 \leq x^2) \quad x}{0^2 \leq x} \qquad \dfrac{\mathcal{D}}{\forall y(y^2 \leq x \to (y+1)^2 \leq x)}}{\forall y(y^2 \leq x)} \quad x+1}{(x+1)^2 \leq x}}{\bot}}{\forall y(y^2 \leq x \to x < (y+1)^2 \to \bot) \to \bot \equiv \exists y(y^2 \leq x < (y+1)^2)}}{\forall x \exists y(y^2 \leq x < (y+1)^2)}$$

3.2.2. BEMERKUNGEN. 1. Es genügt, in der Sprache nur $0, \mathrm{S}, \mathrm{pred}, +, \cdot$ (pred soll die Vorgängerfunktion bedeuten) als Funktionssymbole und nur $=$ als Relationssymbol zuzulassen, und neben den Induktionsaxiomen nur die *Peano–Axiome*

$$\forall x(\mathrm{S}(x) \neq 0),$$
$$\forall x, y(\mathrm{S}(x) = \mathrm{S}(y) \to x = y),$$
$$\forall x(x \neq 0 \to x = \mathrm{S}(\mathrm{pred}(x))),$$
$$\forall x(x + 0 = x),$$
$$\forall x, y(x + \mathrm{S}(y) = \mathrm{S}(x + y)),$$
$$\forall x(x \cdot 0 = 0),$$
$$\forall x, y(x \cdot \mathrm{S}(y) = x \cdot y + x$$

zuzulassen, sowie die *Gleichheitsaxiome*

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

und für jedes Relations– und Funktionssymbol

$$\forall x_1, \ldots, x_n, x'_1, \ldots, x'_n(x_1 = x'_1 \wedge \cdots \wedge x_n = x'_n \wedge Rx_1, \ldots, x_n \to Rx'_1, \ldots, x'_n),$$
$$\forall x_1, \ldots, x_n, x'_1, \ldots, x'_n(x_1 = x'_1 \wedge \cdots \wedge x_n = x'_n \to fx_1, \ldots, x_n = fx'_1, \ldots, x'_n).$$

bereitzustellen. Dann kann man in einem präzisierbaren Sinn alle berechenbaren Funktionen definieren. Das entstehende formale System heißt *Peano–Arithmetik* und wird meist mit PA bezeichnet. — Für unsere Zwecke der Verwendung von Beweisen als Programmen ist ein entgegengesetzter Weg angemessen: da Π–Axiome keinen Beitrag zu dem durch den Beweis gegebenen Programm leisten werden, werden wir möglichst viele Π–Axiome verwenden. Ihre Wahrheit muß natürlich gewährleistet sein, eventuell durch einen separaten Beweis. Die Verwendung von Π–Axiomen kürzt Beweise i.a. sehr ab und macht damit ihre Verwendung als Programme erst praktikabel.

2. Unter Verwendung der Peano–Axiome

$$\forall x(\mathrm{S}(x) \neq 0),$$
$$\forall x, y(\mathrm{S}(x) = \mathrm{S}(y) \to x = y)$$

und der Gleichheitsaxiome kann man die Stabilität von $=$

$$\forall x, y(\neg\neg x = y \to x = y)$$

durch (zweifache) Induktion beweisen. Deshalb kann man sich in den Herleitungen der Peano–Arithmetik auf die Minimallogik beschränken; wir schreiben $S \vdash_{\mathrm{PA}} A$ und nennen die Formel A aus S in der Peano–Arithmetik herleitbar, wenn es eine Herleitung von A aus S und den Peano–Axiomen gibt. Für unsere $\bot\wedge\to\forall$–Sprache (das "negative Fragment" der vollen Sprache mit konstruktivem $\vee$ und $\exists$) fallen also für Herleitungen in der Peano–Arithmetik minimale, intuitionistische und klassische Logik zusammen.

3.3 Beispiel: Obere Schranke von + und ·

{*Schranke*}

Wir wollen jetzt eine arithmetische Herleitung der Existenzaussage aus Beispiel 1.3.4 angeben. {*schrankenherl*}
Anhand dieses Beispiels lassen sich verschiedene Aspekte der beabsichtigten Verwendung von Beweisen als Programm gut demonstrieren; das Beispiel stammt von Goad [7].

Wir verwenden das in Bemerkung 3.1.2 eingeführte Standardmodell der positiven rationalen Zahlen und die zugehörige arithmetische Sprache. Als Abkürzung führen wir ein

$$x \dot{+} y \leq z \quad \text{für} \quad x + y \leq z \wedge x \cdot y \leq z.$$

Zu zeigen ist also, daß es eine gemeinsame obere Schranke für Summe und Produkt zweier positiver rationaler Zahlen gibt, d.h.

$$\forall x, y \exists z(x \dot{+} y \leq z). \tag{1}$$

Im Beweis verwenden wir die folgenden Π–Axiome als Lemmata:

$$L_1\colon \forall x, y(x \leq 1 \to x \dot{+} y \leq y + 1),$$
$$L_2\colon \forall x, y(y \leq 1 \to x \dot{+} y \leq x + 1),$$
$$L_3\colon \forall x, y(x \not\leq 1 \to y \not\leq 1 \to x \dot{+} y \leq 2xy).$$

Zu beweisen ist also (1). Da der Existenzquantor $\exists z$ durch $\neg\forall z\neg$ definiert ist, müssen wir genauer zeigen

$$\forall x, y[\forall z(x \dot{+} y \not\leq z) \to \bot]. \tag{2}$$

Wir verwenden deshalb im Beweis die (falsche) Annahme

$$u\colon \forall z(x \dot{+} y \not\leq z)$$

und haben $\bot$ zu zeigen.

Als erstes konstruieren wir eine Herleitung $\mathcal{D}_{12}$, die außer u noch $u_1\colon x \not\leq 1$ und $u_2\colon y \not\leq 1$ als freie Annahmen enthält.

$$\dfrac{\dfrac{u\colon \forall z(x \dot{+} y \not\leq z) \qquad 2xy}{x \dot{+} y \not\leq 2xy} \qquad \dfrac{\overline{\overline{L_3 \quad x \quad y \quad u_1\colon x \not\leq 1 \quad u_2\colon y \not\leq 1}}}{x \dot{+} y \leq 2xy}\,{-}^*}{\bot}$$

Aus $\mathcal{D}_{12}$ konstruieren wir eine Herleitung $\mathcal{D}_1$, die außer u nur noch $u_1\colon x \not\leq 1$ als freie Annahme enthält.

$$\dfrac{\dfrac{u\colon \forall z(x \dot{+} y \not\leq z) \qquad x + 1}{x \dot{+} y \not\leq x + 1} \qquad \dfrac{\overline{\overline{L_2 \quad x \quad y}} \qquad \dfrac{\textsc{Stab} \qquad \dfrac{\dfrac{\mathcal{D}_{12}}{\bot}\,{\to}^+ u_2}{\neg\neg y \leq 1}}{y \leq 1}}{x \dot{+} y \leq x + 1}\,{-}^*}{\bot}$$

Schließlich ergibt sich die folgende Herleitung $\mathcal{D}$, die nur noch u als freie Annahme enthält.

$$\dfrac{\dfrac{u\colon \forall z(x \dot{+} y \not\leq z) \qquad y + 1}{x \dot{+} y \not\leq y + 1} \qquad \dfrac{\overline{\overline{L_1 \quad x \quad y}} \qquad \dfrac{\textsc{Stab} \qquad \dfrac{\dfrac{\mathcal{D}_1}{\bot}\,{\to}^+ u_1}{\neg\neg x \leq 1}}{x \leq 1}}{x \dot{+} y \leq y + 1}\,{-}^*}{\bot}$$

Damit ist die Herleitung vollständig angegeben. Wir schreiben sie noch einmal insgesamt auf.

$$\dfrac{\dfrac{u\colon \forall z(x \dot{+} y \not\leq z)\ \ 2xy}{x \dot{+} y \not\leq 2xy}\,\forall^- \qquad \dfrac{\overline{\overline{L_3\ x\ y\ \ u_1\colon x \not\leq 1\ \ u_2\colon y \not\leq 1}}}{x \dot{+} y \leq 2xy}\,{-}^*}{\dfrac{\bot}{\neg\neg y \leq 1}\,{\to}^+ u_2}\,{\to}^-$$

$$\dfrac{\dfrac{u\colon \forall z(x \dot{+} y \not\leq z)\ \ x + 1}{x \dot{+} y \not\leq x + 1}\,\forall^- \qquad \dfrac{\overline{\overline{L_2\ x\ y}} \qquad \dfrac{\textsc{Stab} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\mathcal{D}_{12}}{\bot}\,{\to}^+ u_2}{\neg\neg y \leq 1}}{y \leq 1}}{x \dot{+} y \leq x + 1}\,{-}^*}{\bot}\,{\to}^-}{\neg\neg x \leq 1}\,{\to}^+ u_1}\,{\to}^-$$

$$\dfrac{\dfrac{u\colon \forall z(x \dot{+} y \not\leq z)\ \ y + 1}{x \dot{+} y \not\leq y + 1}\,\forall^- \qquad \dfrac{\overline{\overline{L_1\ x\ y}} \qquad \dfrac{\textsc{Stab} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathcal{D}_1}{\bot}\,{\to}^+ u_1}{\neg\neg x \leq 1}}{x \leq 1}}{x \dot{+} y \leq y + 1}\,{-}^*}{\bot}\,{\to}^-}{\neg\neg y \leq 1}\,{\to}^+ u_2}\,{\to}^-$$

Wir werden später (in Abschnitt 4.3) eine "*direkte Methode*" kennenlernen, mit der man einen Beweis einer $\forall\exists$–Formel mit quantorenfreiem Kern als Programm verwenden kann. Im Vorgriff darauf wollen wir schon hier das diesem Beweis entsprechende Programm angeben:

$$\begin{array}{llllll} \textbf{if} & x \dot{+} y \leq y+1 & \textbf{then} & y+1 & \textbf{else} & \\ \textbf{if} & x \dot{+} y \leq x+1 & \textbf{then} & x+1 & \textbf{else} & 2xy. \end{array}$$

Wir betrachten jetzt den Spezialfall $y = \frac{1}{2}$. In $\mathcal{D}_1$ kommt dann die Formel $\frac{1}{2} \leq 1$ vor; wir nehmen an, daß wir ein Lemma $L\colon \frac{1}{2} \leq 1$ zur Verfügung haben. Indem wir in $\mathcal{D}_1$ den Herleitungsteil über $\frac{1}{2} \leq 1$ streichen und stattdessen das Lemma $L\colon \frac{1}{2} \leq 1$ verwenden, erhalten wir insgesamt die folgende Herleitungs $\mathcal{D}'$.

$$\dfrac{\dfrac{u\colon \forall z\ x \dot{+} \frac{1}{2} \not\leq z \qquad \frac{3}{2}}{x \dot{+} \frac{1}{2} \not\leq \frac{3}{2}}\,\forall^- \qquad \dfrac{L_1 x\ \frac{1}{2} \qquad \dfrac{\textsc{Stab} \qquad \dfrac{\dfrac{\dfrac{u\colon \forall z\ x \dot{+} \frac{1}{2} \not\leq z \qquad x+1}{x \dot{+} \frac{1}{2} \not\leq x+1}\,\forall^- \qquad \dfrac{L_2\ x\ \frac{1}{2}\ L\colon \frac{1}{2} \leq 1}{x \dot{+} \frac{1}{2} \leq x+1}\,-^*}{\bot}\,\to^-}{\neg\neg x \leq 1}\,\to^+ u_1}{x \leq 1}\,\to^-}{x \dot{+} \frac{1}{2} \leq \frac{3}{2}}\,-^*}{\bot}\,\to^-$$

Damit ist die Herleitung im Spezialfall vollständig angegeben.

Entsprechendes Programm:

$$\textbf{if} \quad x \dot{+} \frac{1}{2} \leq \frac{3}{2} \quad \textbf{then} \quad \frac{3}{2} \quad \textbf{else} \quad x+1.$$

Was hat man nun durch die Spezialisierung gewonnen? Man beachte, daß in $\mathcal{D}'$ die Regel $\to^+ u_1$ leer läuft: sie bindet eine überhaupt nicht vorkommende Annahme. Man kann sich also den Herleitungteil unter $\mathcal{D}'_1$ sparen und erhält so eine veränderte Herleitung:

$$\dfrac{\dfrac{u\colon \forall z\ x \dot{+} \frac{1}{2} \not\leq z \qquad x+1}{x \dot{+} \frac{1}{2} \not\leq x+1}\,\forall^- \qquad \dfrac{L_2\ x\ \frac{1}{2}\ L\colon \frac{1}{2} \leq 1}{x \dot{+} \frac{1}{2} \leq x+1}\,-^*}{\bot}\,\to^-$$

Entsprechendes Programm:

$$x+1.$$

Diese Herleitung liefert also einen Algorithmus, der eine andere obere Schranke zu x und $\frac{1}{2}$ berechnet als der Originalalgorithmus.

3.4 Beispiel: Das Wall Street Problem

{*Maxseg*}

Wir geben jetzt noch eine arithmetische Herleitung der Existenzaussage aus Beispiel 1.3.5 an. Hier spielen (im Gegensatz zu dem eben behandelten Beispiel) Induktionsschlüsse eine wesentliche Rolle.

Wir wollen hier anhand dieses einfachen Beispiels einen weiteren Aspekt dieser Art von Programmiersprache demonstrieren, nämlich die Möglichkeit, allgemeine Programme an spezielle Situationen anzupassen und sie dabei — auch extensional — zu verändern. Daß dies möglich ist, wurde zuerst von Goad in seiner Dissertation [7] bemerkt.

Unser Beispiel ist ein einfaches aber reales Programmierproblem, das sogenannte Wall Street Problem (oder Maximalsegmentproblem) aus [1]: Zu einer gegebenen Folge $x_0, x_1, \ldots, x_n$ ganzer Zahlen ist ein maximales Segment zu finden, also eine durch Indizes i und k mit $i \leq k$ bestimmte Teilfolge aufeinander folgender Glieder, so daß die Summe $x_i + \cdots + x_k$ maximal ist.

Zunächst formulieren wir die Behauptung etwas allgemeiner und verwenden anstelle von $x_i + \cdots + x_k$ ein beliebiges Maß $\mathrm{seg}(i,k)$ für $x_i, \ldots, x_k$, also eine Abbildung $\mathrm{seg}\colon \mathbb{N} \times \mathbb{N} \to X$, wobei X eine beliebige (partiell) geordnete Menge ist. Gesucht sind dann i, k so daß $\mathrm{seg}(i,k)$ maximal in X ist. Ein Beispiel ist etwa $x_i \in \mathbb{Q}^+$ und $\mathrm{seg}(i,k) := \prod_{i \leq j \leq k} x_j$.

Natürlich kann man dieses Problem einfach durch Probieren aller Möglichkeiten lösen; dies sind $O(n^2)$ viele. Dieser trivialen Lösung entspricht auch der im folgenden angegebene Beweis für die allgemeine Behauptung. Genauer heißt dies: Aus dem konstruktiv geführten Beweis kann man einen Algorithmus ablesen, der dem simplen Durchprobieren entspricht.

Wir wollen dann anschließend zeigen, daß für unser konkretes Problem mit der Summe $x_i + \cdots + x_k$ als Maß der Beweis sich vereinfachen läßt, indem man an geeigneten Stellen die Monotonie der Summe verwendet. Aus diesem vereinfachten Beweis läßt sich dann ein besserer Algorithmus ablesen, der nur noch $O(n)$ viele Schritte benötigt. Wir haben also hier durch Anpassen eines allgemeinen Beweises an eine spezielle Situation erreicht, daß der aus dem Beweis abzulesende Algorithmus wesentlich verbessert wurde.

Wir skizzieren zuerst den allgemeinen (und trivialen) Beweis. Zu zeigen ist

3.4.1. Lemma. *(Spezifikation).*

$$\forall n \exists i,k (i \leq k \leq n \wedge \forall i',k' [i' \leq k' \leq n \to \mathrm{seg}(i',k') \leq \mathrm{seg}(i,k)]).$$

Beweis durch Induktion über n. Der Induktionsanfang ist trivial, und im Schritt bestimmt man zunächst ein maximales Endsegment von $x_0, \ldots, x_n, x_{n+1}$ und bildet dann das Maximum von diesem maximalen Endsegment und dem nach IH gegebenen maximalen Segment. □

Die erste Feststellung ist hier, daß man sowieso neben dem gesuchten maximalen Segment auch noch das maximale Endsegment berechnen muß. Es liegt deshalb nahe, die Spezifikation so zu erweitern, daß neben dem maximalen Segment auch noch das maximale Endsegment zu finden ist.

Lemma. *(Erweiterte Spezifikation). Für alle n gilt*

$$\exists i,k (i \leq k \leq n \wedge \forall i',k' [i' \leq k' \leq n \to \mathrm{seg}(i',k') \leq \mathrm{seg}(i,k)]) \tag{1}$$

$$(\exists j \leq n)(\forall j' \leq n)(\mathrm{seg}(j',n) \leq \mathrm{seg}(j,n)) \tag{2}$$

Beweis durch Induktion über n. *Fall* 0. Wähle $i = k = 0$ und $j = 0$. *Fall* $n+1$. Nach IH haben wir bereits i_n, k_n sowie j_n.

Wir wollen zunächst j_{n+1} konstruieren, haben also zu zeigen

$$(\exists j \leq n+1)(\forall j' \leq n+1)(\mathrm{seg}(j',n+1) \leq \mathrm{seg}(j,n+1)). \tag{3}$$

Da uns j_n hier offensichtlich nicht weiterhilft, verallgemeinern wir die Behauptung (3) so, daß wir sie durch eine Nebeninduktion beweisen können, nämlich zu

$$(\forall m \leq n+1)(\exists \ell \leq m)(\forall \ell' \leq m)(\mathrm{seg}(\ell',n+1) \leq \mathrm{seg}(\ell,n+1)). \tag{3$^+$}$$

Den Beweis führen wir durch Nebeninduktion nach m, wobei n als Parameter festgehalten wird. *Fall* 0. Wähle $\ell = 0$. *Fall* $m+1$. Nach NIH haben wir bereits ein ℓ_m. Ist $\mathrm{seg}(\ell_m, n+1) < \mathrm{seg}(m+1, n+1)$, so sei $\ell_{m+1} := m+1$; andernfalls setzen wir $\ell_{m+1} := \ell_m$. — Damit ist (3^+) bewiesen, und mit $m := n+1$ können wir unser gesuchtes j_{n+1} definieren als $j_{n+1} := \ell_{n+1}$.

Jetzt können wir i_{n+1}, k_{n+1} konstruieren. Ist $\mathrm{seg}(j_{n+1}, n+1) < \mathrm{seg}(i_n, k_n)$, so nehmen wir das alte maximale Segment und setzen $i_{n+1} := i_n$, $k_{n+1} := k_n$; andernfalls ist das maximale Endsegment das maximale Segment und wir setzen $i_{n+1} := j_{n+1}$, $k_{n+1} := n+1$. □

Offenbar ist der diesem Beweis entsprechende Algorithmus quadratisch.

3.4.2. Wir wollen jetzt zeigen, wie man bei zusätzlichen Kenntnissen über die Eingangsdaten den Beweis vereinfachen kann. In unserer Ausgangssituation galt

$$(x_i + \cdots + x_k \le x_j + \cdots + x_k) \to (x_i + \cdots + x_k + x_{k+1} \le x_j + \cdots + x_k + x_{k+1}).$$

Wir betrachten deshalb den Spezialfall

$$\forall i, j, k[\mathrm{seg}(i,k) \le \mathrm{seg}(j,k) \to \mathrm{seg}(i, k+1) \le \mathrm{seg}(j, k+1)]. \qquad (*)$$

Diese Zusatzannahme ermöglicht es uns, die Existenz (3) eines maximalen Endsegments direkter zu beweisen, und zwar

- ohne Verwendung der Verallgemeinerung (3^+) und der zu ihrem Beweis nötigen Nebeninduktion, aber
- mit Hilfe der IH — also j_n — und der Fallunterscheidung $\mathrm{seg}(j_n, n+1) < \mathrm{seg}(n+1, n+1)$ bzw. $\ge$.

Diesen durch die Zusatzannahme $(*)$ ermöglichten verkürzten Beweis wollen wir jetzt genauer durchführen. Dazu modifizieren wir den obigen Beweis wie eben beschrieben.

Beweis (Verkürzte Form). Zu zeigen ist wieder (1) und (2). Verwendet wird Induktion über n. *Fall* 0. Wähle $i = k = 0$ und $j = 0$. *Fall* $n+1$. Nach IH haben wir bereits i_n, k_n sowie j_n.

Wir wollen zunächst j_{n+1} konstruieren, haben also zu zeigen

$$(\exists j \le n+1)(\forall j' \le n+1)(\mathrm{seg}(j', n+1) \le \mathrm{seg}(j, n+1)). \qquad (3)$$

Die IH liefert uns für beliebige $j' \le n$

$$\mathrm{seg}(j', n) \le \mathrm{seg}(j_n, n)$$

und deshalb aufgrund der Zusatzannahme $(*)$ auch

$$\mathrm{seg}(j', n+1) \le \mathrm{seg}(j_n, n+1). \qquad (4)$$

Ist nun $\mathrm{seg}(j_n, n+1) < \mathrm{seg}(n+1, n+1)$, so sei $j_{n+1} := n+1$. Dann haben wir für beliebiges $j' \le n+1$ wie gewünscht stets $\mathrm{seg}(j', n+1) \le \mathrm{seg}(j_{n+1}, n+1)$: im Fall $j' \le n$ folgt dies aus (4) und der Fallunterscheidungsannahme, und im Fall $j' = n+1$ ist es trivial.

Ist andererseits $\mathrm{seg}(n+1, n+1) \le \mathrm{seg}(j_n, n+1)$, so setzen wir $j_{n+1} := j_n$. Dann haben wir für beliebiges $j' \le n+1$ wieder $\mathrm{seg}(j', n+1) \le \mathrm{seg}(j_{n+1}, n+1)$: im Fall $j' \le n$ ist dies gerade (4), und im Fall $j' = n+1$ ist es die Fallunterscheidungsannahme.

Jetzt können wir wie vorher i_{n+1}, k_{n+1} konstruieren. □

Offenbar ist der diesem Beweis entsprechende Algorithmus linear.

BEMERKUNG. Wir wollen an dieser Stelle kurz die Frage diskutieren, ob und inwieweit diese Beweistransformation automatisierbar ist. Ein vollautomatisches Verfahren ohne Hilfe durch den Benutzer scheint mir kaum erreichbar zu sein. Andererseits sollte es als Hilfe genügen, daß man die folgenden aus der informalen Analyse des Problems offensichtlichen Hinweise gibt.

- Zu ersetzen ist der Beweisteil, der im Induktionsschritt die Existenz eines maximalen Endsegments liefert, also der zu (3) führt.
- Der neu zu führende, kürzere Beweis von (3) soll — neben der Zusatzannahme $(*)$ und der IH — die Fallunterscheidung $\mathrm{seg}(j_n, n+1) < \mathrm{seg}(n+1, n+1)$ bzw. $\geq$ verwenden.

In unserem Beweissystem MINLOG haben wir diese Fallstudie implementiert, und zwar mit folgendem Ergebnis. Aus dem Originalbeweis und dem unter der Zusatzannahme verkürzten Beweis haben wir Programme extrahiert, die die erwartete Form haben: der Originalbeweis liefert einen quadratischen und der verkürzte Beweis einen linearen Algorithmus. Ferner haben wir verifiziert, daß sich mit den angegebenen Hilfen der verkürzte Beweis aus dem Originalbeweis auch automatisch gewinnen läßt (falls man in (3) $(\forall j' \leq n+1)A(j)$ umformuliert in $(\forall j' \leq n)A(j) \wedge A(n+1)$).

Kapitel 4

Beweise als Programme

{BaP}

Wir wollen in diesem letzten Kapitel zeigen, wie man Existenzbeweise als Programme verwenden (d.h. interpretieren) kann. Als Beispiele werden die in den Anschnitten 3.3 und 3.4 entwickelten Herleitungen verwendet. Ferner zeigen wir, daß und wie diese in Beweisform gegebenen und deshalb "extrem kommentierten" Programme bei Vorliegen von zusätzlichen Informationen über die Daten verändert werden können.

Ein Wort der Vorsicht ist jedoch am Platz: es handelt sich hier um eine "im Prinzip" anwendbare Methode, deren praktische Relevanz noch nicht abschätzbar ist.

4.1 Existenz einer Normalform

{ExNf}

Wir zeigen in diesem Abschnitt, daß jeder Herleitungsterm d der Minimallogik durch geeignete Umformungsschritte (Konversionen) in eine gewisse Normalform gebracht werden kann; dies ist für die beabsichtige Extraktion von Programmen aus Beweisen wesentlich. Die Normalform zeichnet sich dadurch aus, daß eine Herleitung in Normalform keine "Umwege" enthält, genauer, daß es nicht mehr vorkommen kann, daß eine Beseitigungsregel unmittelbar auf eine Einführungsregel folgt. Herleitungen in Normalform haben besonders angenehme Eigenschaften, aus denen sich viel ablesen läßt.

Für unsere beabsichtigen Anwendungen ist es nötig, ein entsprechendes Resultat auch für die formalen Systeme der Arithmetik zur Verfügung zu haben, die wir in Kapitel 3 eingeführt hatten. Hier sind also noch Π–Axiome und Induktionsaxiome zugelassen, und wir müssen festlegen, was "Umwege" bei der Verwendung dieser Axiome sein sollen. Da es für unsere Zwecke ausreicht, wollen wir eine Konversionsregel für Induktionsaxiome nur dann vorsehen, wenn eine Anwendung auf eine Ziffer vorliegt.

4.1.1. *Konversionen.* Wir betrachten folgende *Konversionen* zwischen Herleitungen.

$$
\begin{array}{ll}
\pi_0(\langle d^A, e^B\rangle^{A\wedge B})^A & \mapsto d^A,\\
\pi_1(\langle d^A, e^B\rangle^{A\wedge B})^B & \mapsto e^B,\\
((\lambda u^A d^B)^{A\to B} e^A)^B & \mapsto d[e^A/u^A]^B,\\
((\lambda x d^A)^{\forall x A} t)^{A[t/x]} & \mapsto d[t/x]^{A[t/x]},\\
(\mathrm{IND}_{A(x)} d^{A(0)} e^{\forall x(A(x)\to A(x+1))} 0)^{A(0)} & \mapsto d^{A(0)},\\
(\mathrm{IND}_{A(x)} d^{A(0)} e^{\forall x(A(x)\to A(x+1))} (\underline{n+1}))^{A(\underline{n+1})} & \mapsto (e^{\forall x(A(x)\to A(x+1))} \underline{n} (\mathrm{IND}_{A(x)} d e \underline{n})^{A(\underline{n})})^{A(\underline{n+1})}.
\end{array}
$$

Eine Herleitung heißt *normal*, falls sie keine konvertierbare Teilherleitung (*Redex*) enthält.

Mit einer Methode von Tait zeigen wir, daß jede Herleitung durch fortwährendes Konvertieren von Redexen in Normalform gebracht werden kann. Wir wählen dabei stets den am weitesten links stehenden maximalen Redex aus (*leftmost–outermost reduction*). In der Term–Notation von Herleitungen ist das der Redex, der am weitesten links beginnt.

Wir definieren zu jeder nicht normalen Herleitung d^A eine Herleitung $\mathcal{R}(d)^A$, die aus d durch Konversion des ausgewählten Redexes entsteht. $\mathcal{R}(d)$ wird durch Rekursion nach dem Aufbau von d definiert.

(i) $\mathcal{R}(d) := e$, falls $d \mapsto e$.

(ii) Ist d nicht konvertierbar, so definieren wir rekursiv

$$\begin{aligned}
\mathcal{R}(\langle d, e\rangle) &:= \begin{cases} \langle \mathcal{R}(d), e\rangle & \text{falls } d \text{ nicht normal ist,} \\ \langle d, \mathcal{R}(e)\rangle & \text{falls } d \text{ normal ist,} \end{cases} \\
\mathcal{R}(\pi_i(d)) &:= \pi_i(\mathcal{R}(d)) \quad \text{für } i = 0, 1 \\
\mathcal{R}(\lambda u d) &:= \lambda u \mathcal{R}(d), \\
\mathcal{R}(de) &:= \begin{cases} \mathcal{R}(d)e & \text{falls } d \text{ nicht normal ist,} \\ d\mathcal{R}(e) & \text{falls } d \text{ normal ist,} \end{cases} \\
\mathcal{R}(\lambda x d) &:= \lambda x \mathcal{R}(d), \\
\mathcal{R}(dt) &:= \mathcal{R}(d)t.
\end{aligned}$$

Wir schreiben $d \longrightarrow e$ (d *reduziert* zu e), falls d nicht normal ist und $\mathcal{R}(d) = e$.

Eine Herleitung d^A heißt *normalisierbar*, falls es Herleitungen $d_0^A, \ldots, d_n^A$ gibt, so daß

$$d^A = d_0^A \longrightarrow \cdots \longrightarrow d_n^A$$

und d_n^A normal ist. d_n^A heißt dann die *Normalform* von d^A und $d^A = d_0^A \longrightarrow \cdots \longrightarrow d_n^A$ die *normalisierende Reduktionsfolge* für d. Man beachte, daß die Normalform und die normalisierende Reduktionsfolge eindeutig bestimmt sind, falls sie existieren.

4.1.2. Da wir hier eine spezielle Reduktionsstrategie verwenden (und nicht zeigen — was aber auch richtig ist —, daß *jede* Reduktionsfolge abbricht), müssen wir das folgende Lemma beweisen.

LEMMA. $\{normal\}$

(i) Sind $\pi_0(d^{A\wedge B})^A$ und $\pi_1(d^{A\wedge B})^B$ normalisierbar, so auch $d^{A\wedge B}$.

(ii) Ist $(d^{A\to B}u^A)^B$ normalisierbar, wobei u^A eine Annahmenvariable ist, die in d nicht frei vorkommt, so ist auch $d^{A\to B}$ normalisierbar.

(iii) Ist $(d^{\forall x A}x)^A$ normalisierbar, so ist auch $d^{\forall x A}$ normalisierbar.

BEWEIS. (i) Nach Voraussetzung hat man normalisierende Reduktionsfolgen für $\pi_0(d)$ und $\pi_1(d)$. Falls in einer der beiden Reduktionsfolgen keine äußere Konversion stattfindet, so hat man sofort eine normalisierende Reduktionsfolge für d. Andernfalls hat man

$$\begin{aligned}
\pi_0(d) &\longrightarrow \cdots \longrightarrow \pi_0(\langle d_0, d_1\rangle) \longrightarrow d_0 \longrightarrow \cdots \longrightarrow e_0, \\
\pi_1(d) &\longrightarrow \cdots \longrightarrow \pi_1(\langle d_0, d_1\rangle) \longrightarrow d_1 \longrightarrow \cdots \longrightarrow e_1
\end{aligned}$$

mit normalen e_0, e_1. Es folgt

$$d \longrightarrow \cdots \longrightarrow \langle d_0, d_1\rangle \longrightarrow \cdots \longrightarrow \langle e_0, d_1\rangle \longrightarrow \cdots \longrightarrow \langle e_0, e_1\rangle.$$

Die Teile (ii) und (iii) sind genauso einfach zu zeigen. □

4.1.3. *Neutrale Herleitungen.* Gewisse Herleitungen gestatten keine "äußere" Konversion und sind deshalb besonders einfach zu diskutieren. Ein typisches Beispiel ist eine Herleitung $ud_1 \dots d_n$ mit einer Annahmenvariable u. Sind $d_1, \dots, d_n$ normalisierbar, so werden wir erwarten können, daß auch $ud_1 \dots d_n$ normalisierbar ist. Solche Herleitungen wollen wir *neutral* nennen. Genauer definieren wir folgendes.

DEFINITION. (Neutrale Herleitung). {*defneutral*}

(i) Jede Annahme u^A ist neutral.

(ii) Ist $d^{A\wedge B}$ neutral, so auch $\pi_0(d^{A\wedge B})^A$und $\pi_1(d^{A\wedge B})^B$.

(iii) Ist $d^{A\to B}$ neutral und e^A normalisierbar, so ist $(d^{A\to B}e^A)^B$ neutral.

(iv) Ist $d^{\forall x A}$ neutral, so ist $(d^{\forall x A}t)^{A[t/x]}$ neutral für jeden Term t.

(v) Sind $d^{A(0)}$ und $e^{\forall x(A(x)\to A(x+1))}$ normalisierbar und ist t ein *nicht* geschlossener Term, so ist $(\mathrm{IND}_{A(x)}d^{A(0)}e^{\forall x(A(x)\to A(x+1))}t)^{A(t)}$ neutral.

(vi) Alle sonstigen Axiome (für die es keine Konversionsregeln gibt) sind neutral.

Aus der Definition ergibt sich unmittelbar, daß eine neutrale Herleitung nicht mit einer Einführungsregel endet.

LEMMA. *Jede neutrale Herleitung ist normalisierbar und keine Herleitung in der Reduktionsfolge endet mit einer Einführungsregel.* {*neutral*}

BEWEIS durch Induktion nach der Definition von "neutral". Wir beschränken uns auf den Fall (iii). Sei also $d^{A\to B}$ neutral und e normalisierbar. Zu zeigen ist, daß $(d^{A\to B}e^A)^B$ normalisierbar ist, und daß keine Herleitung in der Reduktionsfolge mit einer Einführungsregel endet. Sei $d = d_0 \longrightarrow \cdots \longrightarrow d_n$ die Reduktionsfolge zu d und $e = e_0 \longrightarrow \cdots \longrightarrow e_m$ die Reduktionsfolge zu e. Da kein d_i mit einer Eiführungsregel endet, ist

$$de = d_0e \longrightarrow \cdots \longrightarrow d_ne = d_ne_0 \longrightarrow \cdots \longrightarrow d_ne_m$$

die Reduktionsfolge zu de. □

4.1.4. Der zentrale Begriff, mit dessen Hilfe wir die Existenz der Normalform einer Herleitung beweisen können, ist der einer *berechenbaren* Herleitung d^A; er geht auf Tait zurück. Wir definieren berechenbare Herleitungen durch Induktion nach der Komplexität von A, d.h. der Anzahl der logischen Zeichen $\wedge$, $\to$ und $\forall$ in A.

DEFINITION. (Berechenbare Herleitungen). {*defber*}

(i) $d^{R\vec{t}}$ ist genau dann berechenbar, wenn $d^{R\vec{t}}$ normalisierbar ist.

(ii) $d^{A\wedge B}$ ist genau dann berechenbar, wenn $\pi_0(d^{A\wedge B})^A$ und $\pi_1(d^{A\wedge B})^B$ berechenbar sind.

(iii) $d^{A\to B}$ ist genau dann berechenbar, wenn für jede berechenbare Herleitung e^A die Herleitung $(d^{A\to B}e^A)^B$ berechenbar ist.

(iv) $d^{\forall x A}$ ist genau dann berechenbar, wenn für jeden Term t die Herleitung $(d^{\forall x A}t)^{A[t/x]}$ berechenbar ist.

4.1.5. LEMMA. *Endet d^A nicht mit einer Einführungsregel, gilt $d^A \longrightarrow e^A$ und ist e^A berechenbar, so ist auch d^A berechenbar.* $\{berbar\}$

BEWEIS durch Induktion nach A.

Fall $R\vec{t}$. Für atomare Formeln ist Berechenbarkeit dasselbe wie Normalisierbarkeit. Deshalb gilt die Behauptung offensichtlich.

Fall $A_0 \wedge A_1$. Da $e^{A_0 \wedge A_1}$ berechenbar ist, sind $\pi_i(e^{A_0 \wedge A_1})$ für $i = 0, 1$ berechenbar. Da $d^{A_0 \wedge A_1}$ keine Einführung ist, gilt $\pi_i(d) \longrightarrow \pi_i(e)$ wegen $d \longrightarrow e$. Da $\pi_i(d)^{A_i}$ keine Einführung ist, ist nach IH $\pi_i(d)$ für $i = 0, 1$ berechenbar. Folglich ist d berechenbar.

Fall $A \to B$. Sei f^A berechenbar. Z.z.: $(d^{A \to B} f^A)^B$ ist berechenbar. Da $e^{A \to B}$ berechenbar ist, ist $(e^{A \to B} f^A)^B$ berechenbar. Da $d^{A \to B}$ keine Einführung ist, gilt $df \longrightarrow ef$ wegen $d \longrightarrow e$. Da df keine Einführung ist, ist nach IH df berechenbar. Folglich ist d berechenbar.

Fall $\forall x A$. Leicht. □

4.1.6. LEMMA. $\{bernormneut\}$

(i) Jede berechenbare Herleitung ist normalisierbar.

(ii) Jede neutrale Herleitung ist berechenbar.

BEWEIS. (i) und (ii) werden simultan durch Induktion nach der Komplexität der hergeleiteten Formel bewiesen.

Fall $R\vec{t}$. (i) ist trivial. (ii) gilt nach Lemma 4.1.3.

Fall $A \wedge B$. (i) Sei $d^{A \wedge B}$ berechenbar. Nach Definition sind dann $\pi_0(d)^A$ und $\pi_1(d)^B$ berechenbar und damit nach IH (i) normalisierbar. Nach Lemma 4.1.2(i) ist dann auch d normalisierbar.

(ii) Sei $d^{A \wedge B}$ neutral. Nach Definition sind dann $\pi_0(d)^A$ und $\pi_1(d)^B$ neutral und damit nach IH (ii) berechenbar. Nach Definition ist dann auch d berechenbar.

Fall $A \to B$. (i) Sei $d^{A \to B}$ berechenbar. Sei u^A eine neue Annahmenvariable. Da nach IH (ii) u^A berechenbar ist, ist auch $(du)^B$ berechenbar. Nach IH (i) ist nun $(du)^B$ normalisierbar und damit nach Lemma 4.1.2(ii) auch d.

(ii) Sei $d^{A \to B}$ neutral. Sei e^A berechenbar. Z.z.: $(de)^B$ ist berechenbar. Nach IH (i) ist e^A normalisierbar und damit ist $(de)^B$ neutral. Nach IH (ii) ist $(de)^B$ berechenbar.

Fall $\forall x A$. (i) Sei $d^{\forall x A}$ berechenbar. Nach Definition ist $(dx)^A$ berechenbar und damit, nach IH (i) normalisierbar. Mit Lemma 4.1.2(iii) folgt, daß auch d normalisierbar ist.

(ii) Sei $d^{\forall x A}$ neutral und t ein beliebiger Term. Z.z.: $(de)^B$ ist berechenbar. $(dt)^{A[t/x]}$ ist neutral und damit nach IH (ii) berechenbar. □

4.1.7. LEMMA. *Jedes Axiom ist berechenbar.* $\{berax\}$

BEWEIS. Um zu zeigen, daß ein Induktionsaxiom $\mathrm{IND}_{A(x)}$ berechenbar ist, geben wir uns berechenbare Herleitungen $d^{A(0)}$ und $e^{\forall x(A(x) \to A(x+1))}$ und einen Term t vor und zeigen, daß die Herleitung

$$d_t := (\mathrm{IND}_{A(x)} d^{A(0)} e^{\forall x(A(x) \to A(x+1))} t)^{A(t)}$$

berechenbar ist. Ist t nicht geschlossen, so ist d_t neutral und damit nach Lemma 4.1.6 berechenbar. Ist t geschlossen, so argumentieren wir durch Induktion nach dem Wert von t. Ist $t = 0$, so reduziert d_t zu d. Da d nicht mit einer Einführung endet, ist nach Lemma 4.1.5

d_t berechenbar. Ist $t = \underline{n} + 1$, so reduziert d_t zu $(e\underline{n}(\text{IND}_{A(x)}de\underline{n}))$. Nach IH ist $\text{IND}_{A(x)}de\underline{n}$ berechenbar und damit auch $(e\underline{n}(\text{IND}_{A(x)}de\underline{n}))$. Wieder nach Lemma 4.1.5 ist d_t berechenbar.

Die restlichen Axiome sind neutral und damit nach Lemma 4.1.6(ii) berechenbar. □

4.1.8. DEFINITION. Eine Herleitung d^A mit $\text{FA}(d) \subseteq \vec{u}$, $\text{FV}(d) \subseteq \vec{x}$ heißt *berechenbar unter Substitutionen*, wenn für alle berechenbaren Herleitungen $\vec{e}$ und alle Terme $\vec{t}$ die Herleitung $d[\vec{e}, \vec{t}/\vec{u}, \vec{x}]^{A[\vec{t}/\vec{x}]}$ berechenbar ist.

LEMMA. *Jede Herleitung ist berechenbar unter Substitutionen.* {*bersubst*}

BEWEIS durch Induktion nach dem Aufbau der Herleitung. Gelte $\text{FA}(d^A) \subseteq \vec{u}$, $\text{FV}(d^A) \subseteq \vec{x}$, seien $\vec{e}$ berechenbaren Herleitungen, $\vec{t}$ Terme und $(d^*)^{A[\vec{t}/\vec{x}]} := d[\vec{e}, \vec{t}/\vec{u}, \vec{x}]$.

Fall $d = u_i$. Dann ist $d^* = e_i$ und damit berechenbar.

Fall $d = \langle d_0, d_1 \rangle$. Dann ist $d^* = \langle d_0^*, d_1^* \rangle$. Nach IH sind d_0^* und d_1^* berechenbar. Ferner gilt $\pi_0(d^*) \longrightarrow d_0^*$ und $\pi_1(d^*) \longrightarrow d_1^*$. Es folgt mit Lemma 4.1.5, daß $\pi_0(d^*)$ und $\pi_1(d^*)$ berechenbar sind. Damit ist auch d^* berechenbar.

Fall $d = \pi_i(e)$. Nach IH ist e^* berechenbar. Damit ist auch $d^* = \pi_i(e^*)$ berechenbar.

Fall $d = \lambda u^A d_0^B$. Es gilt $d^* = \lambda u^{A[\vec{t}/\vec{x}]} d_0[\vec{e}, \vec{t}/\vec{u}, \vec{x}]^{B[\vec{t}/\vec{x}]}$. Sei $e^{A[\vec{t}/\vec{x}]}$ berechenbar. Z.z.: d^*e ist berechenbar. Es gilt $d^*e \longrightarrow d_0[\vec{e}, e, \vec{t}/\vec{u}, u, \vec{x}]$. Nach IH ist das Redukt berechenbar und damit nach Lemma 4.1.5 auch d^*e.

Fall $d = d_1 d_2$. Nach IH sind d_1^* und d_2^* berechenbar, also auch $d^* = d_1^* d_2^*$.

Die *Fälle* $d = \lambda x\, d_0$ und $d = d_0 t$ sind ähnlich einfach zu behandeln.

Fall d ein Axiom. Die Behauptung folgt aus Lemma 4.1.7. □

4.1.9. SATZ. *Jede Herleitung ist normalisierbar.* {*normsatz*}

BEWEIS. Sei d eine Herleitung mit freien Annahmen $u_1, \ldots, u_n$. Da die u_i neutral und deshalb nach Lemma 4.1.6(ii) berechenbar sind, ist d nach Lemma 4.1.8 berechenbar und damit nach Lemma 4.1.6(i) normalisierbar. □

BEMERKUNG. Mit einer ähnlichen Methode kann man den sogenannten *starken Normalisierungssatz* beweisen. Er sagt aus, daß *jede* Reduktionsfolge terminiert, also nicht nur diejenige, die durch unsere spezielle Reduktionsstrategie (leftmost–outermost reduction) ausgewählt wird. Einen Beweis dazu findet man bei Troelstra [15, 17] oder auch in der Vorlesungsausarbeitung [11]. Mit derselben Beweistechnik kann man auch einen starken Normalisierungssatz für ein Termsystem beweisen, das anstelle der Induktionsaxiome den allgemeinen Fixpunktoperator (siehe Plotkin [9]) oder einen beschränkten Fixpunktoperator (siehe Schwichtenberg/Wainer [12]) enthält. Ferner kann man Abschätzungen der Länge der notwendigen Reduktionsfolgen angeben, wenn man anstelle der Taitschen Methode ein auf Sanchis, Diller und Howard zurückgehendes Verfahren verwendet (siehe [10]).

Man kann auch zeigen, daß bei jeder Reduktionsfolge die Normalform eindeutig bestimmt ist. Ein Beweis, der die auf Tait zurückgehende Technik der parallelen Reduktion verwendet und die zugehörigen Überlegungen noch einmal sehr vereinfacht, ist von Takahashi in [14] veröffentlicht worden.

4.2 Folgerungen aus der Normalisierung

$\{AnwNor\}$

Wir wollen jetzt einige Folgerungen aus der Tatsache ziehen, daß sich jede Herleitung in Normalform bringen läßt, und zu diesem Zweck die Form normaler Herleitungen genauer analysieren. Zunächst benötigen wir einige terminologische Vereinbarungen.

4.2.1. Mit $p, q \in \{0,1\}^*$ bezeichnen wir *Positionen* in einer Herleitung d. $[]$ sei die leere Liste, also die Wurzelposition, und $O(d)$ sei die Menge der Positionen in d. Mit $p0$ und $p1$ bezeichnen wir die Verlängerung der Position p um 0 bzw. 1. Der Teilherleitung von d an der Position p wird mit $d|_p$ bezeichnet, und $d[e]_p$ entstehe aus d, indem man in d die Teilherleitung an der Position p durch e ersetzt. Wir schreiben $p \leq q$, wenn p Präfix oder Anfangsstück von q ist. p und q heißen *unabhängig*, geschrieben $p \parallel q$, wenn $p \not\leq q$ und $q \not\leq p$. Weiter sei $\{pos\}$

$$\begin{aligned}
\textsc{Leafpos}(d) &:= \{\, p \in O(d) \mid d|_p \text{ Variable oder Konstante} \,\},\\
\textsc{Elimpos}(d) &:= \{\, p \in O(d) \mid d|_p \text{ durch Beseitigungsregel gebildet} \,\},\\
\textsc{Intropos}(d) &:= \{\, p \in O(d) \mid d|_p \text{ durch Einführungsregel gebildet} \,\}.
\end{aligned}$$

$\textsc{Leafpos}(d)$ heißt die Menge der *Blattpositionen* von d. Ferner definieren wir für jedes $p \in \textsc{Elimpos}(d) \cup \textsc{Leafpos}(d)$ die zugehörige *Minimalposition* durch

$$\begin{aligned}
&\mathrm{minpos}_d([]) := [].\\
&\mathrm{minpos}_d(pi) := \begin{cases} \mathrm{minpos}_d(p), & \text{falls } i = 0 \text{ und } p \in \textsc{Elimpos}(d);\\ pi, & \text{falls } i = 1 \text{ und } p \in \textsc{Elimpos}(d), \text{ oder}\\ & \quad p \in \textsc{Intropos}(d). \end{cases}
\end{aligned}$$

Offenbar bildet minpos_d die Menge $\textsc{Elimpos}(d) \cup \textsc{Leafpos}(d)$ in sich ab. Wir setzen

$$\textsc{Minpos}(d) := \{\, p \in \textsc{Elimpos}(d) \cup \textsc{Leafpos}(d) \mid \mathrm{minpos}_d(p) = p \,\}.$$

Schließlich definieren wir für jedes $p \in O(d)$ die zugehörige *Endposition* durch

$$\begin{aligned}
&\mathrm{endpos}_d([]) := [].\\
&\mathrm{endpos}_d(pi) := \begin{cases} \mathrm{endpos}_d(p), & \text{falls } p \in \textsc{Intropos}(d), \text{ oder}\\ & \quad i = 0 \text{ und } p \in \textsc{Elimpos}(d);\\ pi, & \text{falls } i = 1 \text{ und } p \in \textsc{Elimpos}(d). \end{cases}
\end{aligned}$$

Wir setzen

$$\textsc{Endpos}(d) := \{\, p \in O(d) \mid \mathrm{endpos}_d(p) = p \,\}.$$

Im Fall einer normalen Herleitung d entsprechen sich die Blattpositionen und die Minimalpositionen von d eineindeutig. Man beachte jedoch, daß zwei Minimalpositionen dieselbe Endposition liefern können, wie etwa in

$$\frac{P \qquad Q}{P \wedge Q}.$$

Gegeben sei ein Herleitung d und $p \in \textsc{Leafpos}(d)$. Dann nennt man

$$\{\, q \in O(d) \mid p \geq q \geq \mathrm{endpos}_d(p) \,\}$$

den durch die Blattposition p bestimmten *Zweig* in d. Insbesondere liegt $\mathrm{minpos}_d(p)$ auf dem durch p bestimmten Zweig. Im Fall $\mathrm{endpos}_d(p) = []$ spricht man von einem *Hauptzweig*. In einer normalen Herleitung hat jeder Zweig $p_1 \geq \cdots \geq p_n$ eine besonders übersichtliche Form:

Alle Beseitigungsregeln müssen vor allen Einführungsregeln kommen.

Hierbei bedeutet "vor", daß die entsprechende Position weiter oben im Herleitungsbaum liegt, also länger ist.

4.2.2. Im Fall einer normalen Herleitung wollen wir hieraus einige Schlußfolgerungen über die an den Positionen angehefteten Formeln ziehen. Dazu benötigen wir die folgenden Begriffe.

Die Relation "A ist eine *Teilformel* von B" soll der reflexive transitive Abschluß der Relation "*unmittelbare Teilformel*" sein, die wie folgt definiert ist.

- A und B sind unmittelbare Teilformeln von $A \to B$,
- A und B sind unmittelbare Teilformeln von $A \wedge B$,
- $A[t/x]$ ist eine unmittelbare Teilformel von $\forall x A$.

Ferner benötigen wir den Begriff "A ist eine *strikt positive Teilformel* von B". Dies soll der reflexive transitive Abschluß der Relation "*unmittelbare strikt positive Teilformel*" sein, die wie folgt definiert ist.

- B ist eine unmittelbare strikt positive Teilformel von $A \to B$,
- A und B sind unmittelbare strikt positive Teilformeln von $A \wedge B$,
- $A[t/x]$ ist eine unmittelbare strikt positive Teilformel von $\forall x A$.

Sei jetzt d eine normale Herleitung, $p_1 \geq \cdots \geq p_n$ ein Zweig in d, $p_m = \mathrm{minpos}_d(p_1)$ und A_i die von $d|_{p_i}$ hergeleitete Formel. Dann muß A_m eine strikt positive Teilformel aller A_i mit $i \neq m$ sein. Dieses A_m heißt die *Minimalformel* des Zweiges. Ferner ist jedes A_i mit $i \leq m$ eine strikt positive Teilformel von A_1, und jedes A_i mit $i \geq m$ ist eine strikt positive Teilformel von A_n.

SATZ. *(Teilformel–Eigenschaft). Ist $d^B[u_1^{A_1}, \ldots, u_n^{A_n}]$ eine normale Herleitung und ist e^C eine Teilherleitung von d^B, so ist C eine Teilformel von B oder von einem A_i.*

BEWEIS durch Induktion über die Länge der Endposition eines Zweiges in d, wobei die eben erwähnten Eigenschaften von Zweigen in normalen Herleitung benötigt werden. □

4.2.3. SATZ. *(Herbrand). Sei $d[u_1^{\forall \vec{x}_1 A_1}, \ldots, u_n^{\forall \vec{x}_m A_m}]: B$ eine normale Herleitung mit quantorenfreien Formeln $A_1, \ldots, A_m, B$. Dann ist jedes Vorkommen einer Annahmevariablen u_i in d in einem Kontext* {herbrand}

$$\frac{u_i: \forall \vec{x}_i A_i \qquad \vec{t}_{ij}}{A_i[\vec{t}_{ij}]}.$$

Insbesondere findet man Terme $\vec{t}_{11}, \ldots, \vec{t}_{1n_1}, \ldots, \vec{t}_{m1}, \ldots, \vec{t}_{mn_m}$ mit

$$A_1[\vec{t}_{11}], \ldots, A_1[\vec{t}_{1n_1}], \ldots, A_m[\vec{t}_{m1}], \ldots, A_m[\vec{t}_{mn_m}] \vdash B.$$

BEWEIS. Zur Vereinfachung der Schreibweise behandeln wir nur den Fall $d[u^{\forall x A}]: B$. Durch Induktion über die Länge der Endposition eines Zweiges zeigt man leicht, daß jeder Zweig mit einer Herleitung einer quantorenfreien Formel endet und beginnen muß

- mit der Regel $\forall^-$, d.h. mit $u^{\forall x A} t_i$, oder
- mit einer gebundenen Annahmevariablen v^C mit quantorenfreiem C.

Man ersetzt im erstgenannten Fall die Teilherleitung $u^{\forall x A} t_i$ durch $v_i^{A[t_i]}$ mit einer neuen Annahmevariablen v_i. □

4.3 Interpretation von Beweisen

{Direkt}

Ausgerüstet mit den in 4.1 und 4.2 erworbenen Kenntnissen können wir jetzt daran gehen, Beweise von Existenzsätzen als Programme zu interpretieren.

4.3.1. Zunächst ist ein Wort der Vorsicht am Platz. Ein (klassischer) Beweis für $\forall x \exists y\, A$ liefert i.a. *kein* Programm zur Berechnung von y aus x. Ein einfaches Gegenbeispiel wurde von Kreisel angegeben. Dazu verwenden wir ein Ergebnis aus der Theorie der Berechenbarkeit (Rekursionstheorie): man kann eine entscheidbare Relation R angeben, für die $\{\, x \mid \exists y R(x,y) \,\}$ unentscheidbar ist. {kreisel}

Offenbar gilt schon in der Minimallogik

$$\vdash \forall x \exists y \forall z [R(x,z) \to R(x,y)].$$

(dies ergibt sich aus $\vdash \exists z\, R(x,z) \to \exists y\, R(x,y)$ mit den bekannten Regeln über das Herausziehen von Quantoren aus Implikationen). Gäbe es nun eine berechenbare Funktion f so daß

$$\forall x, z [R(x,z) \to R(x, f(x))],$$

in $\mathbb{N}$ gilt, so wäre $\{\, x \mid \exists y R(x,y) \,\}$ entscheidbar, denn $\exists z\, R(x,z)$ wäre wahr genau dann, wenn $R(x, f(x))$ ist. — Wir müssen uns also auf Formeln der Gestalt

$$\forall x \exists y\, A \quad \text{mit } A \text{ quantorenfrei}$$

beschränken.

4.3.2. *Interpretation von Beweisen als Programme.* Gegeben sei jetzt eine arithmetische Herleitung von $\forall x \exists y A(x,y)$ mit quantorenfreiem $A(x,y)$. Offenbar können wir annehmen, daß diese Herleitung keine freien Objektvariablen enthält (andernfalls ersetze man sie durch 0). Wir geben jetzt eine "*direkte Methode*" an mit der sich diese Herleitung als Programm interpretieren läßt. {direkt}

Schritt 1. Wir wählen eine Ziffer $\underline{n}$ and Eingabe und spezialisieren die Herleitung auf $\underline{n}$. Damit haben wir eine Herleitung von $\exists y A(\underline{n}, y)$, also nach Auflösung der Abkürzung für den Existenzquantor eine Herleitung

$$d_n[u{:}\forall y(A(\underline{n}, y) \to \bot)]{:}\, \bot.$$

Schritt 2. Bilde die Normalform d_n^* von d_n. Da d_n keine freien Objektvariablen enthält, kommen in d_n^* keine Induktionsaxiome vor. Man beweist dazu durch Induktion über die Ordnung eines Zweiges (= Länge seiner Endposition), daß er nicht mit einem Induktionsaxiom beginnen kann.

Schritt 3. Aufgrund des Herbrandscher Satzes 4.2.3 ist jedes Vorkommen der Annahmevariablen u in d_n^* in einem Kontext

$$\frac{u{:}\forall y(A(\underline{n}, y) \to \bot) \qquad \underline{m_i}}{A(\underline{n}, \underline{m_i}) \to \bot}.$$

Insbesondere hat man also eine Herleitung von $\bot$ aus $A(\underline{n}, \underline{m_1}) \to \bot, \ldots, A(\underline{n}, \underline{m_k}) \to \bot$ und (wahren) Π–Axiomen.

Schritt 4. Da $\bot$ nicht aus wahren Annahmen herleitbar sein kann, muß eine der Formeln $A(\underline{n}, \underline{m_i}) \to \bot$ falsch sein, also $A(\underline{n}, \underline{m_i})$ wahr; man wähle ein solches $\underline{m_i}$ aus. Damit ist zu der Eingabe $\underline{n}$ ein $\underline{m_i}$ mit $A(\underline{n}, \underline{m_i})$ gefunden.

4.3.3. *Beispiel: Obere Schranke von $+$ und $\cdot$.* Betrachten wir noch einmal die in 3.3 angegebene Herleitung

$$d[u{:}\forall z(x\dot{+}y \not\leq z)]{:}\bot.$$

Diese Herleitung ist bereits normal, und in ihr kommen keine Induktionsaxiome vor. In ihr gibt es drei Vorkommen der Annahmevariablen u, und zwar in den Kontexten

$$\frac{u{:}\forall z(x\dot{+}y \not\leq z) \qquad y+1}{x\dot{+}y \not\leq y+1} \qquad \frac{u{:}\forall z(x\dot{+}y \not\leq z) \qquad x+1}{x\dot{+}y \not\leq x+1} \qquad \frac{u{:}\forall z(x\dot{+}y \not\leq z) \qquad 2xy}{x\dot{+}y \not\leq 2xy}$$

Aus der allgemeinen Theorie wissen wir, daß eine dieser Formeln falsch sein muß. Die Reihenfolge der Suche nach einer falschen Formel haben wir nicht festgelegt; wählt man eine natürliche Reihenfolge (etwa von links nach rechts), so liefert die Herleitung das schon in 3.3 angegebene Programm

$$\begin{array}{llllll} \textbf{if} & x\dot{+}y \leq y+1 & \textbf{then} & y+1 & \textbf{else} & \\ \textbf{if} & x\dot{+}y \leq x+1 & \textbf{then} & x+1 & \textbf{else} & 2xy. \end{array}$$

4.3.4. *Beispiel: Existenz einer angenäherten Quadratwurzel.* In 3.3 haben wir eine Herleitung von

$$\forall x \exists y(y^2 \leq x < (y+1)^2),$$

angegeben, in der ein Induktionsaxiom verwendet wird. Spezialisiert man diese Herleitung für $x = 2$, so ergibt sich mit $B(x) := 0^2 \leq x \to \forall y[y^2 \leq x \to (y+1)^2 \leq x] \to \forall y(y^2 \leq x)$ die Herleitung

$$\dfrac{\dfrac{\dfrac{\dfrac{\textsc{Ind}{:}\forall x B(x) \qquad 2}{B(2)} \qquad \dfrac{\mathcal{D}_0(2)}{0^2 \leq 2}}{\forall y[y^2 \leq 2 \to (y+1)^2 \leq 2] \to \forall y(y^2 \leq 2)} \qquad \dfrac{\mathcal{D}'(2)}{\forall y[y^2 \leq 2 \to (y+1)^2 \leq 2]}}{\forall y(y^2 \leq 2)} \qquad 3}{3^2 \leq 2}$$

die in der spezialisierten Schrittherleitung $\mathcal{D}'(2)$ die falsche Annahme

$$u{:}\forall y(y^2 \leq 2 \to 2 < (y+1)^2 \to \bot)$$

enthält. Diese Herleitung ist nicht in Normalform, da ein Induktionsaxiom auf 2 angewandt ist. "Ausfalten" des Induktionsaxioms, also die alleinige Verwendung der Konversionsregeln für die Induktion, ergibt

$$\dfrac{\dfrac{\dfrac{\mathcal{D}'(2)}{\forall y[y^2 \leq 2 \to (y+1)^2 \leq 2]} \qquad 2}{2^2 \leq 2 \to 3^2 \leq 2} \qquad \dfrac{\dfrac{\mathcal{D}'(2) \qquad 1}{1^2 \leq 2 \to 2^2 \leq 2} \qquad \dfrac{\dfrac{\mathcal{D}'(2) \qquad 0}{0^2 \leq 2 \to 1^2 \leq 2} \qquad \dfrac{\mathcal{D}_0(2)}{0^2 \leq 2}}{1^2 \leq 2}}{2^2 \leq 2}}{3^2 \leq 2}$$

Erinnern wir uns nun an die Gestalt von $\mathcal{D}'(2)$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\textsc{Ax}{:}\forall x,y[(x<y \to \bot) \to y \leq x] \quad 2 \quad (y+1)^2}{(y+1)^2 \leq 2} \qquad \dfrac{u{:}\forall y(y^2 \leq 2 \to 2<(y+1)^2 \to \bot) \quad y \quad v{:}y^2 \leq 2}{2<(y+1)^2 \to \bot}}{y^2 \leq 2 \to (y+1)^2 \leq 2}}{\forall y(y^2 \leq 2 \to (y+1)^2 \leq 2)}$$

Es ist damit klar, daß die ausgefaltete Herleitung noch nicht in Normalform ist. Wir wollen diese Normalform hier nicht explizit angeben, sondern nur feststellen, daß in ihr die Annahme u auf $2,1$ und 0 spezialisiert sein muß. Also muß eine der Formeln

$$2^2 \le 2 \to 2 < (2+1)^2 \to \bot,$$
$$1^2 \le 2 \to 2 < (1+1)^2 \to \bot,$$
$$0^2 \le 2 \to 2 < (0+1)^2 \to \bot$$

falsch sein. Offenbar ist es die mittlere; wir haben also zu unserer Eingabe 2 das Ergebnis 1 als angenäherte Quadratwurzel erhalten.

4.4 Programmentwicklung durch Beweistransformation

{*Entw*}

In dem zuletzt behandelten Beispiel der Existenz einer angenäherten Quadratwurzel stellt sich der Verdacht ein, daß unsere Interpretation von Herleitungen als Programmen nur auf ein Probierverfahren hinausläuft. Dann wäre aber der Aufwand überflüssig, denn bei einer wahren Formel $\forall x \exists y A(x,y)$ mit quantorenfreiem $A(x,y)$ kann man zu gegebenem $\underline{n}$ offensichtlich das kleinste $\underline{m}$ mit $A(\underline{n},\underline{m})$ einfach durch Probieren finden.

Hierzu ist folgendes zu sagen. Es hängt wesentlich von dem Beweis ab, welche Instanzen der spezialisierten falschen Annahme u in der Herleitung vorkommen, und in welcher Reihenfolge das natürliche Suchverfahren (von links nach rechts) sie abarbeitet. Dies kann ein Probierverfahren sein, muß es aber nicht. Im Quadratwurzel–Beispiel beginnt das natürliche Suchverfahren nicht etwa bei 0 und zählt aufwärts, sondern es zählt von der durch den Beweis vorgegebenen Schranke (nämlich $x+1$) abwärts.

Die Möglichkeit, Beweise als Programme zu interpretieren, also Programme in einer "extrem kommentierten" Form vorliegen zu haben, hat aber noch einen weiteren Aspekt, den wir bereits in 3.3 und 3.4 erwähnt haben: man kann Programme an spezielle Situationen anpassen und sie dabei — auch extensional — verändern. Daß dies möglich ist, wurde zuerst von Goad in seiner Dissertation [7] bemerkt.

Wir wollen dies am Beispiel des obigen Existenzbeweises einer oberen Schranke demonstrieren. Die mit $\frac{1}{2}$ spezialisierte Herleitung war

$$\dfrac{\dfrac{\dfrac{u{:}\forall z\ x\dot{+}\frac{1}{2} \not\le z \qquad \frac{3}{2}}{x\dot{+}\frac{1}{2} \not\le \frac{3}{2}}\forall^- \qquad \dfrac{L_1\, x\ \frac{1}{2}}{x\dot{+}\frac{1}{2} \le \frac{3}{2}}-^*}{\bot}\to^- \qquad \dfrac{\dfrac{\dfrac{u{:}\forall z\ x\dot{+}\frac{1}{2} \not\le z \qquad x+1}{x\dot{+}\frac{1}{2} \not\le x+1}\forall^- \qquad \dfrac{L_2\ x\ \frac{1}{2}\ \ L{:}\frac{1}{2} \le 1}{x\dot{+}\frac{1}{2} \le x+1}-^*}{\bot}\to^- \qquad \dfrac{\dfrac{\text{STAB}}{\neg\neg x \le 1}\to^+ u_1}{x \le 1}\to^-}{x\dot{+}\frac{1}{2} \le \frac{3}{2}}-^*$$

Diese Herleitung ist wieder bereits normal, und ihr kommen keine Induktionsaxiome vor. In ihr gibt es zwei Vorkommen der Annahmevariablen u, und zwar in den Kontexten

$$\dfrac{u{:}\forall z(x\dot{+}\frac{1}{2} \not\le z) \qquad \frac{3}{2}}{x\dot{+}\frac{1}{2} \not\le \frac{3}{2}} \qquad\qquad \dfrac{u{:}\forall z(x\dot{+}\frac{1}{2} \not\le z) \qquad x+1}{x\dot{+}\frac{1}{2} \not\le x+1}$$

Aus der allgemeinen Theorie wissen wir, daß eine dieser Formeln falsch sein muß. Die Reihenfolge der Suche nach einer falschen Formel haben wir nicht festgelegt; wählt man eine natürliche

Reihenfolge (etwa von links nach rechts), so liefert die Herleitung das schon in 3.3 angegebene Programm

$$\textbf{if} \quad x \dot{+} \frac{1}{2} \leq \frac{3}{2} \quad \textbf{then} \quad \frac{3}{2} \quad \textbf{else} \quad x+1.$$

Was hat man nun durch die Spezialisierung gewonnen? Man beachte, daß die Regel $\rightarrow^+ u_1$ leer läuft: sie bindet eine überhaupt nicht vorkommende Annahme. Man kann deshalb den Herleitungteil zwischen den beiden Vorkommen von $\bot$ herausschneiden ("*pruning*") und erhält so eine veränderte Herleitung:

$$\dfrac{\dfrac{u\colon \forall z\ x \dot{+} \frac{1}{2} \not\leq z \qquad x+1}{x \dot{+} \frac{1}{2} \not\leq x+1}\,\forall^- \qquad \dfrac{\begin{array}{c}[L_2 \quad x\ \frac{1}{2} \quad L\colon \frac{1}{2} \leq 1] \\ \hline x \dot{+} \frac{1}{2} \leq x+1\end{array}}{\bot}\,{-}^* \rightarrow^-$$

Jetzt gibt es nur noch ein Vorkommen der Annahmenvariablen u, und zwar im Kontext

$$\dfrac{u\colon \forall z (x \dot{+} \frac{1}{2} \not\leq z) \qquad x+1}{x \dot{+} \frac{1}{2} \not\leq x+1}$$

Das entsprechende Programm ist also einfach

$$x+1.$$

Diese Herleitung liefert demnach einen Algorithmus, der eine andere obere Schranke zu x und $\frac{1}{2}$ berechnet als der Originalalgorithmus.

Als abschließendes Beispiel für die Programmentwicklung durch Beweistransformation kommen wir noch einmal auf das Wall Street Problem zurück. Wir hatten in 3.4 zunächst einen Existenzbeweis angegeben, der dem einfachen Durchprobieren entsprach; er enthielt zwei übereinanderliegende Induktionen. Der durch Interpretation dieses Beweises als Programm entstehende Algorithmus ordnet jedem n Zahlen i_n, k_n, j_n wie folgt zu.

$n = 0$: $\quad i_0 = k_0 = j_0 = 0$

$n+1$: $\quad$ Gegeben: i_n, k_n, j_n

$$j_{n+1} := \ell_{n+1} \text{ mit}$$

$$\begin{cases} \ell_0 := 0 \\ \ell_{m+1} := \begin{cases} m+1 & \text{falls } \mathrm{seg}(\ell_m, n+1) < \mathrm{seg}(m+1, n+1) \\ \ell_m & \text{sonst} \end{cases} \end{cases}$$

$$(i_{n+1}, k_{n+1}) := \begin{cases} (i_n, k_n) & \text{falls } \mathrm{seg}(j_{n+1}, n+1) < \mathrm{seg}(i_n, k_n) \\ (j_{n+1}, n+1) & \text{sonst} \end{cases}$$

Wir hatten dann zusätzliche Kenntnisse über die Daten angenommen, nämlich daß die Segmentfunktion monoton ist (wie dies bei der Summe der Fall ist). Unter dieser Zusatzannahme ließ sich der Beweis vereinfachen, und zwar so, daß nur noch eine Induktion auftrat. Dem vereinfachten Beweis entspricht folgender Algorithmus:

$n = 0$: $\quad i_0 = k_0 = j_0 = 0$

$n+1$: $\quad$ Gegeben: i_n, k_n, j_n

$$j_{n+1} := \begin{cases} n+1 & \text{falls } \mathrm{seg}(j_n, n+1) < \mathrm{seg}(n+1, n+1) \\ j_n & \text{sonst} \end{cases}$$

(i_{n+1}, k_{n+1}) $\quad$ wie eben

Wir haben also einen linearen (statt quadratischen) Algorithmus gewonnen. Man beachte, daß das von dem verbesserten Algorithmus gelieferte maximale Segment *verschieden* sein kann von dem durch den ursprünglichen Algorithmus gefundenen. Dies ist ein Efekt, der durch reine Programmtransformation (etwa partielle Auswertung) prinzipiell nicht erreicht werden kann.

4.5 Ausblick

{*Ausblick*}

Ein Nachteil der hier dargestellten Interpretation von Beweisen als Programmen ist sicherlich, daß bei jeder Eingabe wieder der gesamte Beweis normalisiert werden muß. Zwar läßt sich die Normalisierung von in unserem Formalismus des natürlichen Schließens geführten Beweisen effizient implementieren (siehe etwa [2]), aber sicherlich sind hier noch viele etwa im Compilerbau übliche Optimierungen möglich.

Es liegt deshalb nahe, unsere Logiksprache durch den *starken* oder *konstruktiven Existenzquantor* zu erweitern, wie er in der intuitionistischen Logik betrachtet wird. Dann kann man aus einem Beweis einer starken Existenzaussage mittels der sogenannten "*modifizierten Realisierbarkeitsübersetzung*" ein Programm in einer funktionalen Programmsprache extrahieren, das wiederum aufgrund seiner Konstruktion korrekt sein muß.

Diese Art der Programmierung bezeichnen Bates und Constable in [1] als "very high level programming". Sie ermöglicht es, den bewährten mathematisch–logischen Formalismus als Programmiersprache zu benutzen; auf diese Weise werden nachträgliche Korrektheitsbeweise überflüssig. Der höhere Aufwand bei der Erstellung von Programmen wird dadurch erträglicher, daß man Beweise von reinen Allsätzen nicht zu formalisieren braucht, falls man ihre Richtigkeit auf andere Weise eingesehen hat.

Index

Π–Axiom, 21
Äquivalenz, 4

Ableitung, 10
All–Axiome, 21
Allformel, 5
Alphabet, 3
Annahme, 11
    geschlossene, 11
    offene, 11
Annahmenvariablen, 11
Aristoteles, 3
Aussage, 3
Aussagensymbol, 7
Axiom, 3

Beseitigungsregel, 11
Beweis, 10
Blattposition, 34

direkte Methode, 25, 36
Disjunktion, 4

Einführungsregel, 11
Endposition, 34
Ex–Falso–Quodlibet, 16
Existenzformel, 5
Existenzquantor
    konstruktiver, 40
    starker, 40

Falsum, 5
formales System, 3
Formel, 3, 8
    atomare, 8
    geschlossene, 9
Frege, 4
Funktion
    berechenbare, 20
Funktionssymbol, 3, 8

genau dann wenn, 4
Gleichheitsaxiome, 23
Grundterm, 8
Gruppentheorie, 5

Hauptprämisse, 11
Hauptzweig, 34
Herbrandscher Satz, 35
herleitbar, 12
Herleitung, 10
    berechenbare, 31
    berechenbare, unter Substitutionen, 33
    neutrale, 31
    normale, 29
    normalisierbare, 30

Implikation, 4
Individuenbereich, 3
Induktionsaxiom, 21
Infix, 8

Klammerkonventionen, 9
Konjunktion, 3
Konklusion, 3, 11
Konstante, 3, 8
Konversion, 29

leftmost–outermost reduction, 30
Logik
    intuitionistische, 16
    klassische, 16
    minimale, 12

Marke, 11
Minimalformel, 35
Minimalposition, 34
modifizierte Realisierbarkeit, 40
modus ponens, 11

Nebenprämisse, 11
Negation, 4
negatives Fragment, 23
Normalform, 30
Normalisierungssatz
    starker, 33

Objektvariablen, 11
oder, 4

partielle Auswertung, 40
Peano–Arithmetik, 23
Peano–Axiome, 22
Position, 34
    unabhängige, 34
Prädikatensymbol, 7
Prämisse, 3, 11
Primformel, 8
Prinzip des indirekten Beweisens, 16
pruning, 39

Quadratwurzel–Beispiel, 22
Quantor, 5

Redex, 29
reduziert, 30
Regel, 11
Relation
    entscheidbare, 20
Relationssymbol, 3, 7

Satz, 9
Schlußregel, 3
Semantik, 2
Signatur, 7
Sprache
    arithmetische, 20
    erster Stufe, 7
Stabilität, 16
Syntax, 2

Teilformel, 35
    strikt positive, 35
    unmittelbare strikt positive, 35
Teilformel–Eigenschaft, 35
Term, 3, 8
    geschlossener, 8

Umbenennung, 5
und, 3

Variable, 5, 7
    Annahmen–, 11
    freie, 5, 9, 15
    gebundene, 5
Variablenbedingung, 11

Wahrheitsbegriff, 21
Wahrheitsfunktion, 4
Wahrheitswert, 4
wenn–so, 4
Wert eines Terms, 20

Ziffer, 20
Zweig, 34

Literaturverzeichnis

[1] Joseph L. Bates and Robert L. Constable. Proofs as programs. *ACM Transactions on Programming Languages and Systems*, 7(1):113–136, January 1985.

[2] Ulrich Berger and Helmut Schwichtenberg. An inverse of the evaluation functional for typed λ–calculus. In Rao Vemuri, editor, *Proceedings of the Sixth Annual IEEE Symposium on Logic in Computer Science*, pages 203–211. IEEE Computer Society Press, Los Alamitos, 1991.

[3] Hans-Dieter Ebbinghaus, Jörg Flum, and Wolfgang Thomas. *Einführung in die mathematische Logik*. Wissenschaftliche Buchgesellschaft, Darmstadt, 3. edition, 1992.

[4] Herbert B. Enderton. *A Mathematical Introduction to Logic*. Academic Press, New York, 1972.

[5] Gerhard Gentzen. Untersuchungen über das logische Schließen. *Mathematische Zeitschrift*, 39:176–210, 405–431, 1934.

[6] Jean-Yves Girard, Paul Taylor, and Yves Lafont. *Proofs and Types*. Cambridge University Press, 1989.

[7] Christopher Alan Goad. *Computational uses of the manipulation of formal proofs*. PhD thesis, Stanford University, August 1980. Stanford Department of Computer Science Report No. STAN–CS–80–819.

[8] Robert Gray. Computer programs and mathematical proofs. *The Mathematical Intelligencer*, 13(4), 1991.

[9] Gordon D. Plotkin. LCF considered as a programming language. *Theoretical Computer Science*, 5:223–255, 1977.

[10] Helmut Schwichtenberg. Primitive recursion on the partial continuous functionals. In M. Broy, editor, *Informatik und Mathematik*, pages 251–269. Springer, Berlin, 1991.

[11] Helmut Schwichtenberg. Proof theory. Lecture notes, Mathematisches Institut der Universität München, 1994. http://rz.mathematik.uni-muenchen.de/ schwicht.

[12] Helmut Schwichtenberg and Stanley S. Wainer. Ordinal bounds for programs. In P. Clote and J. Remmel, editors, *Feasible Mathematics II*, pages 387–406. Birkhäuser, Boston, 1995.

[13] Joseph R. Shoenfield. *Mathematical Logic*. Addison–Wesley Publ. Comp., Reading, Massachusetts, 1967.

[14] Masako Takahashi. Parallel reductions in λ–calculus. *Information and Computation*, 118:120–127, 1995.

[15] Anne S. Troelstra. *Metamathematical Investigation of Intuitionistic Arithmetic and Analysis. Chapters 1–4*. Lecture Notes in Mathematics 344. Springer Verlag, Berlin, Heidelberg, New York, 1973.

[16] Anne S. Troelstra and Helmut Schwichtenberg. *Basic Proof Theory*. Cambridge University Press, to appear.

[17] Anne S. Troelstra and Dirk van Dalen. *Constructivism in Mathematics. An Introduction*, volume 121, 123 of *Studies in Logic and the Foundations of Mathematics*. North–Holland, Amsterdam, 1988.

[18] Dirk van Dalen. *Logic and Structure*. Springer, Berlin, 3. edition, 1994.