Beweistheorie

Prof. Dr. Thomas Streicher

SS 2000

1 Einleitung

Aufgrund der im vorigen Kapitel ausführlich dargestellten Unabhängigkeitsphänomene muß man zu dem Schluß kommen, daß die Beweisbarkeit einer Aussage A in einem bestimmten formalen System wie etwa der PA mehr Information beinhaltet als die bloße Gültigkeit von A in einem Standardmodell wie etwa $\mathbb{N}$. Auf sehr schöne Weise ist diese Fragestellung in folgendem Zitat von G. Kreisel zusammengefaßt.

> What more do we know if we have proved a theorem by restricted means than if we merely know that it is true?

Man kann nun die Stärke eines formalen Systems $\mathcal{S}$ etwa dadurch messen, indem man untersucht welche Sätze einer bestimmten einfachen Bauart in $\mathcal{S}$ hergeleitet werden können. Welche Klasse von Sätzen man dabei in Betracht zieht, ist natürlich etwas subjektiv. Folgende zwei Klassen werden traditionell in der sogenannten *Beweistheorie* betrachtet.

(1) Man untersucht, welche Π_2–Sätze ein formales System zu beweisen gestattet, d.h. welche Sätze der Form $\forall x.\exists y.R(x,y)$, wobei $R(x,y)$ quantorenfrei ist, d.h. im Falle der Arithmetik ein primitiv rekursives Prädikat. In der *Heyting Arithmetik* HA, d.h. PA aber mit *intuitionistischer Logik*, läßt sich leicht zeigen (Übung!), daß jeder Π_2–Satz äquivalent ist zu einem der Gestalt

$$\forall m.\exists k.\, T(\underline{n}, m, k)$$

für eine geeignet gewählte Gödelnummer n, d.h. der Aussage, daß der Algorithmus mit Gödelnummer n für alle Eingaben terminiert. Unter diesem Gesichtspunkt erscheint es natürlich, ein formales System dadurch zu *messen*, indem man die Klasse derjenigen total rekursiven Funktionen bestimmt, für die ein Algorithmus existiert, dessen Termination in dem betrachteten formalen System bewiesen werden kann.

Definition 1.1 *Sei $\mathcal{S}$ ein formales System, das die Sprache der primitiv rekursiven Arithmetik enthält. Die Menge $\mathcal{R}(\mathcal{S})$ der* in $\mathcal{S}$ beweisbar total rekursiven Funktionen *besteht aus allen $f \in \mathcal{R}$, sodaß ein $e \in \mathbb{N}$ existiert mit der Eigenschaft, daß $f = \{e\}$ und $\mathcal{S} \vdash \forall n.\exists k.\, T(\underline{e}, n, k)$.* $\Diamond$

Es sei erinnert, daß nicht alle Algorithmen, die ein $f \in \mathcal{R}(\mathcal{S})$ implementieren, in $\mathcal{S}$ als terminierend nachgewiesen werden können, da dies schon für die Funktion $\lambda x.0$ schiefgeht, wie wir am Ende des letzten Kapitels gesehen haben.
Es sei darauf hingewiesen, daß man für jedes formale System $\mathcal{S}$ eine offensichtlich total rekursive Funktion finden kann, deren Termination in $\mathcal{S}$ selbst nicht nachgewiesen werden kann, nämlich den Interpreter für $\mathcal{R}(\mathcal{S})$[1], welcher durchaus

[1]Sei h eine total rekursive Funktion, die $\{e \in \mathbb{N} \mid \mathcal{S} \vdash \forall n.\exists k.\, T(\underline{e}, n, k)\}$ enumeriert. Dann ist $v(n,m) := u(h(n), m)$ ein Interpreter für $\mathcal{R}(\mathcal{S})$, also offensichtlich total rekursiv. Wenn man

von "praktischem Interesse" ist. Daraus ersehen wir, daß jedes formale System $\mathcal{S}$ auch in dem starken Sinn unvollständig ist, daß es auch interessante Algorithmen nicht als terminierend nachweisen kann.

(2) Das in der traditionellen Beweistheorie (à la Gentzen) am meisten studierte Kriterium ist, für welche primitiv rekursiven Wohlordnungen $\prec$ von $\mathbb{N}$ man das Prinzip der *transfiniten Induktion*

$$\mathrm{TI}(\prec) \qquad \forall x.((\forall y \prec x.X(y)) \to X(x)) \to \forall x.X(x)$$

im betrachteten formalen System herleitbar ist, wobei X eine frische 1–stellige Prädikatenkonstante ist, über die nichts vorausgesetzt wird. Wegen des Vollständigkeitssatzes für die Prädikatenlogik ist $\mathrm{TI}(\prec)$ in $\mathcal{S}$ herleitbar genau dann, wenn $\mathrm{TI}(\prec)$ in allen Modellen von $\mathcal{S}$ wahr ist (weil X durch eine *beliebige* Teilmenge des Grundbereichs interpretiert werden kann).

Bemerkung. In dem Buch *Basic Proof Theory* von Troelstra und Schwichtenberg findet man ein Beispiel für eine primitiv rekursive Ordnung $\prec$ auf $\mathbb{N}$, die nicht wohlfundiert ist, für die jedoch

$$\forall x.((\forall y \prec x.A(y)) \to A(x)) \to \forall x.A(x)$$

für alle arithmetischen Prädikate $A(x)$ in PA herleitbar ist. Daraus ersieht man, daß das *Schema der transfiniten Induktion* im allgemeinen schwächer ist als das Prinzip der der transfiniten Induktion.
Es sei außerdem darauf hingewiesen, daß die primitiv rekursiven Wohlordnungen $\prec$, für die $\mathrm{TI}(\prec)$ in einem formalen System herleitbar ist, nicht unter extensionaler Gleichheit abgeschloßen sind. Der Grund ist, daß extensionale Gleichheit von primitiv rekursiven Relationen nicht immer im formalen Sytem herleitbar ist.

Für ein gegebenes formales System $\mathcal{S}$ kann man sich nun die Frage stellen, was das Supremum aller Ordinalzahlen ist, für die $\mathcal{S}$ das Prinzip der transfiniten Induktion herleitet. Diese Ordinalzahl wird als "beweistheoretische Ordinalzahl von $\mathcal{S}$" oder einfach "das Ordinal von $\mathcal{S}$" bezeichnet.[2]
Für die Peanoarithmetik PA wurde ihre Ordinalzahl bereits von Gerhard GENTZEN in den 30er Jahren des 20. Jahrhunderts bestimmt, nämlich als die Ordinalzahl ε_0, die definiert ist als das kleinste Ordinal α mit $\alpha = \omega^\alpha$. Das Ordinal ε_0 kann "ganz konkret" bestimmt werden als $\sup_{n\in\mathbb{N}}\omega_n$, wobei $\omega_0 = \omega$ (das kleinste

aber in $\mathcal{S}$ die Termination von v nachweisen könnte, so auch die von $f(n) := v(n,n)+1$. Es gäbe also ein $e \in \mathbb{N}$, sodaß $v(e,n) = f(n) = v(n,n)+1$ für alle $n \in \mathbb{N}$, woraus folgen würde, daß $v(e,e) = v(e,e)+1$. Also wird v von $\mathcal{S}$ nicht als total rekursiv erkannt.

[2]Leider gibt es für Ordinalzahlen i.a. verschiedene primitiv rekursive Wohlordnungen $\prec$ desselben Ordnungstyps. Da die Beweisbarkeit von $\mathrm{TI}(\prec)$ von der Wahl von $\prec$ abhängen kann, ist der Begriff "Ordinal eines Systems" etwas vage. In der Praxis schränkt man sich auf sogenannte "natürliche Wohlordnungen" ein, was zwar inhärent vage, jedoch in konkreten Fällen immer klar ist.

unendliche Ordinal) und $\omega_{n+1} = \omega^{\omega_n}$. Basierend auf dem "Cantorschen Normalformentheorem" hat Gentzen eine "natürliche Wohlordnung" angegeben, die ε_0 repräsentiert, und gezeigt, daß für $\alpha < \varepsilon_0$ das Prinzip der transfiniten Induktion in PA herleitbar ist, nicht jedoch für ε_0 selbst, da andernfalls in PA durch Induktion über ε_0 der Schnitteliminationssatz für die PA hergeleitet werden könnte, was unmöglich ist, da aus dem Schnitteliminationssatz für PA unmittelbar die Konsistenz der PA folgt und diese nach Gödel ja nicht in der PA selbst bewiesen werden kann.
Seitdem haben sich die Beweistheoretiker dahingehend amüsiert, für immer stärkere Systeme die Ordinalzahl zu bestimmen, was zunehmend schwieriger wird je stärker die Systeme werden. Ein magischer Grenzwert ist PA_2, die Arithmetik zweiter Stufe, für die bisher noch keine Ordinalzahl bestimmt werden konnte. Es stellt sich nämlich heraus, daß die Größe der Ordinalzahlen mit der Mächtigkeit der Komprehensionsprinzipien rapide zunehmen (und PA_2 hat ja uneingeschränkte Komprehension).
Da die Ordinalzahlanalyse ein zwar schönes, jedoch technisch sehr anspruchsvolles Gebiet der Beweistheorie ist, wenden wir uns im weiteren den in (1) beschriebenen Fragestellungen zu, die überdies für die Mathematik und Informatik als fruchtbarer und auch anwendungsorientierter erscheinen.[3]

2 Die beweisbar rekursiven Funktionen von PA

Wir definieren im folgenden eine *Programmiersprache*, die genau diejenigen total rekursiven Funktionen zu programmieren gestattet, die durch einen Algorithmus implementiert werden können, dessen Termination in PA bewiesen werden kann. Diese Programmiersprache ist als "Gödels System T" bekannt und basiert auf dem typisierten λ–Kalkül, der um arithmetische Konstanten und einen Rekursor R angereichert wird.
Wir setzen die Sprache des typisierten λ–Kalküls im weiteren als bekannt voraus. Es sei bloß erwähnt, daß wir bloß folgende Gleichheitsaxiome postulieren

(β) $(\lambda x{:}\sigma.t)s = t[s/x]$

(c) $t_1 = t_2 \wedge s_1 = s_2 \to t_1 s_1 = t_2 s_2$

die für unsere Zwecke voll ausreichen, da sie gestatten, Programme "auszuwerten".[4]
Gödels T stellt einen Basistyp ι der natürlichen Zahlen bereit sowie die Konstruktoren 0 und succ, die es gestatten, natürliche Zahlen kanonisch als Terme

[3] Da ja terminierende Programme einem meist etwas näherliegen als Ordinalzahlen. Es sei jedoch bemerkt, daß transfinite Induktion über wohlfundierte Relationen sehr oft für Terminationsbeweise in der (theoretischen) Informatik herangezogen werden. Man kann z.B. auch nachweisen, daß die in PA beweisbar rekursiven Funktionen gerade die ε_0–rekursiven sind.

[4] Die bekannte η–Regel $\lambda x{:}\sigma.tx = t$ (wobei $x \notin \mathrm{FV}(t)$) wird zu diesem Zweck nicht benötigt.

zu repräsentieren. Darüber hinaus gibt es für jeden Typen σ einen "primitiven Rekursor"

$$R_\sigma : \sigma \to (\iota \to \sigma \to \sigma) \to \iota \to \sigma$$

der folgenden Gleichungen genügt

$$Raf0 = a \qquad\qquad Raf\mathrm{succ}(n) = fn(Rafn)$$

der es gestattet unter anderem auch alle primitiv rekursiven Funktionen zu programmieren.
Aus Gründen der Bequemlichkeit, um unnötiges Codieren zu vermeiden, gestatten wir uns auch Produkttypen $\sigma \times \tau$ und einen "unit type" υ wie sie in funktionalen Programmiersprachen üblich sind. Für Terme $t : \sigma$ und $s : \tau$ ist $\langle t, s\rangle$ ein Term vom Typ $\sigma \times \tau$. Wenn $t : \sigma \times \tau$, so können wir auf seine Komponenten vermittels $\pi_1(t) : \sigma$ und $\pi_2(t) : \tau$ zugreifen. Wir postulieren folgende β–Regel für Produkte

$$\pi_1(\langle t, s\rangle) = t \qquad\qquad \pi_2(\langle t, s\rangle) = s \quad .$$

Der unit type υ enthält ein ausgezeichnetes Element $*$.[5]

Der Beweis, daß die PA beweisbar rekursiven Funktionen genau die in Gödel T programmierbaren sind, unterteilt sich in zwei Schritte. Zuerst zeigen wir, daß PA bzgl. Π_2–Sätzen konservativ ist über HA, vermittels des sogenannten "Friedman Tricks". Anschließend weisen wir mithilfe der Kreiselschen Modified Realizability nach, daß die HA beweisbar rekursiven Funktionen genau die in Gödel T programmierbaren sind.

2.1 Friedman Trick

Sei R eine frische propositionale Konstante, für die wir *a posteriori* beliebige Formeln einsetzen können, sofern diese Ersetzung zu keiner Verletzung der Variablenbedingungen führt. Um die gewünschte Konservativität zu zeigen, definieren wir folgende R–Übersetzung, die die wohlbekannte "double negation translation" verallgemeinert.

$P^R \equiv (P \to R) \to R \quad$ für atomare Formeln P

$\bot^R \equiv R$

$(A \wedge B)^R \equiv A^R \wedge B^R$

$(A \to B)^R \equiv A^R \to B^R$

$(\forall x.\, A)^R \equiv \forall x.\, A^R$

[5] Wir könnten für υ die η–Regel $t =_\upsilon *$ postulieren, was jedoch für unser Zwecke überflüssig ist.

$$(A \vee B)^R \equiv ((A^R \to R) \wedge (B^R \to R)) \to R$$

$$(\exists x.\, A)^R \equiv (\forall x.\, A^R \to R) \to R$$

Man beachte, daß $(-) \to R$ Rolle einer Art verallgemeinerter Negation spielt.

Lemma 2.1 *Für alle Formeln A ist die logische Äquivalenz von A^R und $(A^R \to R) \to R$ in der* intuitionistischen *Prädikatenlogik beweisbar.*

Beweis: Die behauptete Eigenschaft gilt offensichtlich für alle Formeln, die intuitionistisch beweisbar äquivalent sind zu einer Formel der Gestalt $B \to R$. Im Falle atomarer Formeln und von R–Übersetzungen von Disjunktionen und existenziellen Quantifikationen ist dies offensichtlich der Fall. Für die restlichen Fälle muß man die Induktionshypothese verwenden, die besagt, daß die gewünschte Eigenschaft bereits für die unmittelbaren Konstituenten gilt, sowie folgende intutionistische Tautologien

$$((A \to R) \wedge (B \to R)) \leftrightarrow ((A \vee B) \to R)$$

$$(A \to (B \to R)) \leftrightarrow ((A \wedge B) \to R)$$

$$((\exists x.A) \to R) \leftrightarrow (\forall x.(A \to R))$$

□

Mithilfe diese Lemmas läßt sich nun völlig überraschungsfrei der folgende Satz herleiten.

Satz 2.1 *Wenn $\Gamma \vdash A$ in der PA herleitbar ist, so ist $\Gamma^R \vdash A^R$ in der HA herleitbar.*[6]

Beweis: Ganz allgemein ist diese Aussage für prädikatenlogische Formeln herleitbar durch Induktion über die Struktur klassischer Herleitungen. Wir betrachten nur folgende nicht ganz trivialen Fälle.
(∃E) Angenommen $\Gamma^R \vdash (\exists x.A)^R$ und $\Gamma^R, A(x)^R \vdash B^R$ sind intuitionistisch herleitbar, wobei x eine frische Variable ist, die weder in Γ noch in B frei vorkommt. Wenn wir die Definition der R–Übersetzung ausfalten, heißt das, daß

(1) $\Gamma^R \vdash (\forall x.(A(x)^R \to R)) \to R$

(2) $\Gamma^R, A(x)^R \vdash (B^R \to R) \to R$

wobei wir für (2) bereits Lemma 2.1 verwendet haben. Aus (2) folgt unmittelbar

(3) $\Gamma^R, B^R \to R \vdash \forall x.(A(x)^R \to R)$

woraus mithilfe von (1) folgt, daß $\Gamma^R, B^R \to R \vdash R$ und somit

[6]Wobei Γ^R als Abkürzung steht für $A_1^R, \ldots, A_n^R$, falls $\Gamma \equiv A_1, \ldots, A_n$.

(4) $\Gamma^R \vdash (B^R \to R) \to R$

Somit folgt aber mit Lemma 2.1 (angewendet auf B), daß $\Gamma^R \vdash B^R$, wie gewünscht.
($\bot$) Wenn $\Gamma \vdash \bot$, dann aufgrund der Induktionshypothese auch $\Gamma^R \vdash R$ und somit $\Gamma^R \vdash (A^R \to R) \to R$. Mit Lemma 2.1 folgt, dann aber $\Gamma^R \vdash A^R$, wie erwünscht.
(raa) Wenn $\Gamma^R \vdash (\neg\neg A)^R$, dann gilt aufgrund der Induktionshypothese, daß $\Gamma^R \vdash (A^R \to R) \to R$, da die R–Übersetzung von $(\neg\neg A)^R \equiv (A^R \to R) \to R$. Wiederum mit Lemma 2.1 ergibt sich $\Gamma^R \vdash A^R$.
Aus diesem letzten Fall ergibt sich zwangsläufig die Notwendigkeit von Lemma 2.1, da ansonsten die Übersetzung von *reductio ad absurdum* nicht intuitionistisch beweisbar wäre!
Die R–Übersetzungen der nichtlogischen Axiome von HA sind offensichtlich in der HA beweisbar. □

Korollar 2.1 *Wenn die* PA *eine Formel der Gestalt* $\exists x.P(x)$ *beweist, wobei* P *atomar ist, dann beweist bereits* HA *die Formel* $\exists x.P(x)$.

Beweis: Wenn PA die Formel $\exists x.P(x)$ beweist, dann beweist HA ihre R–Übersetzung $(\exists x.P(x))^R \equiv (\forall x.\, P(x)^R \to R) \to R$ und somit die dazu äquivalente Formel $(\forall x.\, P(x) \to R) \to R$. Also beweist HA auch

$$(\forall x.\, P(x) \to \exists x.P(x)) \to \exists x.P(x)$$

indem wir $R \equiv \exists x.P(x)$ setzen.[7] Da aber $\forall x.\, P(x) \to \exists x.P(x)$ intuitionistisch (ganz einfach) herleitbar ist, ist somit auch $\exists x.\, P(x)$ in der HA herleitbar. □

2.2 Kreisel's Modified Realizability

ist eine allgemeine Methode, um aus konstruktiven Beweisen einen *algorithmischen Gehalt* zu *extrahieren*. Insbesondere liefert diese Methode für jede Herleitung in der HA eines Π_2–Satz $\forall x.\exists y.P(x,y)$ in der HA einen Gödel T Term $t : \iota \to \iota$ mit $\forall x.\, A(x, t(x))$, d.h. eine Gödel T programmierbare Skolemfunktion. Aus Gründen der Zweckmässigkeit definieren wir Modified Realizability nicht bloß für die HA sondern für ihre Erweiterung HA_ω, deren Termsprache gerade Gödel's T ist und deren Axiome neben den Gleichungen von Gödel's T das Axiom $0 \neq \mathrm{succ}(x)$ und das (uneingeschränkte) Induktionsschema umfassen.
Aus Gründen der Ökonomie werden wir im weiteren die Disjunktion aus unseren Betrachtungen ausklammern. Dies ist gerechtfertigt, da sie sowohl in der HA

[7] Was ohne Verletzung der Variablenbedingungen möglich ist, indem wir die gebundenen Variablen in der Herleitung zu den freien Variablen von $\exists x.P(x)$ disjunkt halten, was durch geeignete Umbenennung immer möglich ist.

wie auch in der HA_ω aus den restlichen logischen Operationen folgendermaßen definiert werden kann

$$A \vee B \equiv \exists n : \iota.\, (n = 0 \to A) \wedge (n \neq 0 \to B)$$

Dies ist dem strengen Sinn zu verstehen, daß für die so definierte Disjunktion die übliche zugehörige Introduktions- und Eliminationsregel aus den restlichen Regeln abgeleitet werden kann (Übung!).

Definition 2.1 *Induktiv assoziieren wir wie folgt jeder Formel A (der HA_ω) einen Gödel T Typ der sogenannten* potentiellen Realisatoren *:*

$\mathbf{tp}(P) \equiv \upsilon$ *für atomare Formeln P*

$\mathbf{tp}(\bot) \equiv \upsilon$

$\mathbf{tp}(A \wedge B) \equiv \mathbf{tp}(A) \times \mathbf{tp}(B)$

$\mathbf{tp}(A \to B) \equiv \mathbf{tp}(A) \to \mathbf{tp}(B)$

$\mathbf{tp}(\forall x{:}\sigma.A(x)) \equiv \sigma \to \mathbf{tp}(A)$

$\mathbf{tp}(\exists x{:}\sigma.A(x)) \equiv \sigma \times \mathbf{tp}(A)$.

Man beachte, daß die Definition von $\mathbf{tp}(A)$ rein von der logischen Struktur von A abhängt und nicht von der Struktur der atomaren Formeln.
Ebenfalls durch Induktion über den Aufbau von A definieren wir ein Prädikat $(-)\mathbf{mr}A$ auf den potentiellen Realisatoren von A, das sie sogenannten aktualen Realisatoren *aus den potentiellen aussondert*

$u\ \mathbf{mr}\ P \equiv P$ *für atomare Formeln P*

$u\ \mathbf{mr}\ \bot \equiv \bot$

$u\ \mathbf{mr}\ A \wedge B \equiv (\pi_1(u)\ \mathbf{mr}\ A) \wedge (\pi_2(u)\ \mathbf{mr}\ B)$

$u\ \mathbf{mr}\ A \to B \equiv \forall v : \mathbf{tp}(A).\, (v\ \mathbf{mr}\ A) \to (u(v)\ \mathbf{mr}\ B)$

$u\ \mathbf{mr}\ \forall x{:}\sigma.\, A(x) \equiv \forall x{:}\sigma.\, u(x)\ \mathbf{mr}\ A(x)$

$u\ \mathbf{mr}\ \exists x{:}\sigma.\, A(x) \equiv \pi_2(u)\ \mathbf{mr}\ A(\pi_1(u))$. $\Diamond$

Als nächstes zeigen wir, daß es für jede in HA_ω herleitbare Formel einen sie realisierenden Gödel T Term gibt, wobei diese Eigenschaft in HA_ω auch herleitbar ist.

Satz 2.2 *Wenn* $\Gamma \vdash A$ *in* HA_ω *herleitbar ist, dann gibt es einen Gödel T Term* t *vom Typ* $\mathbf{tp}(A)$*, sodaß*

$$\vec{u}\,\mathbf{mr}\,\Gamma \vdash t\,\mathbf{mr}\,A$$

in HA_ω *herleitbar ist, wobei die Variablen von* $\vec{u}$ *frisch sind, d.h. weder in* Γ *noch in* A *frei vorkommen. Dabei steht* $\vec{u}\,\mathbf{mr}\,\Gamma$ *abkürzend für* $u_1\,\mathbf{mr}\,A_1, \ldots, u_n\,\mathbf{mr}\,A_n$*, falls* $\Gamma \equiv A_1, \ldots, A_n$*.*

Beweis: Man beweist die Aussage geradlinig durch Induktion über die Struktur von Herleitungen.
($\to$I) Wenn

$$\vec{u}\,\mathbf{mr}\,\Gamma, u\,\mathbf{mr}\,A \vdash t\,\mathbf{mr}\,B$$

dann

$$\vec{u}\,\mathbf{mr}\,\Gamma \vdash \lambda u : \mathbf{tp}(A).t\ \mathbf{mr}\ A \to B$$

weil

$$\vec{u}\,\mathbf{mr}\,\Gamma \vdash \forall u : \mathbf{tp}(A).\,(u\,\mathbf{mr}\,A) \to ((\lambda u : \mathbf{tp}(A).t)(u)\,\mathbf{mr}\,B)$$

da $\vdash (\lambda u : \mathbf{tp}(A).t)(u) = t$.
($\to$E) Wenn $\vec{u}\,\mathbf{mr}\,\Gamma \vdash t\,\mathbf{mr}\,A \to B$ und $\vec{u}\,\mathbf{mr}\,\Gamma \vdash s\,\mathbf{mr}\,A$, dann $\vec{u}\,\mathbf{mr}\,\Gamma \vdash t(s)\,\mathbf{mr}\,B$, weil aus $\forall u : \mathbf{tp}(A).\,(u\,\mathbf{mr}\,A) \to t(u)\,\mathbf{mr}B$ und $s\,\mathbf{mr}\,A$ unmittelbar folgt, daß $t(s)\,\mathbf{mr}\,B$.
Für die restlichen logischen Regeln ist der Nachweis ähnlich einfach. Wir betrachten bloß noch
($\exists$E) Angenommen $\vec{u}\,\mathbf{mr}\,\Gamma \vdash t\,\mathbf{mr}\,\exists x{:}\sigma.\,A(x)$ und $\vec{u}\,\mathbf{mr}\,\Gamma, u\,\mathbf{mr}\,A(x) \vdash s\,\mathbf{mr}\,B$, wobei u eine frische Variable ist. Dann gilt auch $\vec{u}\,\mathbf{mr}\,\Gamma \vdash s[\pi_1(t), \pi_2(t)/x, u]\,\mathbf{mr}\,B$, da $t\,\mathbf{mr}\,\exists x{:}\sigma.\,A(x) \equiv \pi_2(t)\,\mathbf{mr}\,A(\pi_1(t))$.
Für die nichtlogischen Axiome von HA_ω ist der Nachweis trivial, da sich die Eigenschaft ein aktualer Realisator zu sein darauf reduziert, daß die Axiome herleitbar sind, was trivial ist. Eine Ausnahme bildet natürlich das Induktionsschema. Dieses wird jedoch durch R realisiert, wie folgende Überlegung zeigt. Für eine Formel $A(x)$ gilt

$$\vdash R_{\mathbf{tp}(A)}\ \mathbf{mr}\,A(0) \to \forall x{:}\iota.(A(x) \to A(\mathrm{succ}(x))) \to \forall x{:}\iota.A(x)$$

da aus $u\,\mathbf{mr}\,A(0)$ und $\forall x{:}\iota.\forall z{:}\mathbf{tp}(A).\,(z\,\mathbf{mr}\,A(x)) \to v(x)(z)\,\mathbf{mr}\,A(\mathrm{succ}(x))$ die Aussage $\forall x{:}\iota.\,Ruvx\,\mathbf{mr}\,A(x)$ leicht mit Induktion bewiesen werden kann. □

Aus der Inspektion dieses Beweises ergibt sich unmittelbar, daß für Herleitungen in der HA_ω *Schritt für Schritt* der algorithmische oder konstruktive Gehalt in Form eines Gödel T Terms "mitgerechnet" werden kann. Somit stellt Modified Realizability in folgendem präzisen Sinn eine Methode der *Programmextraktion* bereit: aus einer Herleitung von $\forall x{:}\sigma.\exists y{:}\tau.A(x,y)$ in HA_ω lesen wir (durch eine Art "Buchhaltungsprozess") ein Gödel T Programm $t : \sigma \to \tau \times \mathbf{tp}(A)$ ab, für welches

$$\forall x{:}\sigma.\,\pi_2(t(x))\,\mathbf{mr}\,A(x, \pi_1(t(x)))$$

in der HA_ω herleitbar ist.
Falls A selbst schon quantorenfrei ist, gilt folgende stärkere Aussage.

Satz 2.3 *Wenn $\forall x.\exists y.R(x,y)$ für ein primitiv rekursives Prädikat R in der* PA *herleitbar ist, dann gibt es einen Gödel T Term $f : \iota \to \iota$, für den in der* HA_ω *bewiesen werden kann, daß $\forall x{:}\iota.R(x, f(x))$.*

Beweis: Wenn die PA die Aussage $\forall x.\exists y.R(x,y)$ beweist, so auch HA und somit auch HA_ω, da HA ein Teilsystem von HA_ω ist. Wegen Satz 2.2 gibt es in Gödel's T einen Term $t : \iota \to \iota \times \upsilon$, sodaß $\forall x{:}\iota.\, \pi_2(t(x))\, \mathbf{mr}\, R(x, \pi_1(t(x))$ in HA_ω hergeleitet werden kann. Wenn wir f definieren als $\lambda x{:}\iota.\, \pi_1(t(x))$, beweist also HA_ω, daß

$$\forall x : \iota.R(x, f(x))$$

da ja $\pi_2(t(x))\, \mathbf{mr}\, R(x, \pi_1(t(x)) \equiv R(x, \pi_1(t(x)))$, weil $R(x, \pi_1(t(x))$ atomar ist. □

Als wichtiges Korollar erhalten wir, daß PA beweisbar rekursive Funktionen in System T implementiert werden können.

Korollar 2.2 *Wenn* PA $\vdash \forall x.\exists y.T(\underline{e}, x, y)$*, dann gibt es einen (geschlossenen) Term $t : \iota \to \iota$ mit $\{e\}(n) = t(n)$ für alle n.*

Beweis: Wenn PA $\vdash \forall x.\exists y.T(\underline{e}, x, y)$, dann gibt es aufgrund von Satz 2.3 einen Term $\widetilde{t} : \iota \to \iota$, sodaß $\forall x{:}\iota.\, T(\underline{e}, x, \tilde{t}(x))$ in HA_ω herleitbar ist. Offensichtlich implementiert dann $t \equiv \lambda x{:}\iota.\, U(\tilde{t}(x))$ die rekursive Funktion mit Gödelnummer e in Gödel's T. □

2.3 Termination von Gödel T Programmen in PA

Durch Induktion über den Aufbau von Typen σ definieren wir ein Prädikat T_σ auf natürlichen Zahlen in der Sprache der HA

$$T_\iota(x) \equiv x = x$$

$$T_\upsilon(x) \equiv x = 0$$

$$T_{\sigma\to\tau}(x) \equiv \forall y.\, T_\sigma(y) \to \exists z.\, T(x,y,z) \wedge T_\tau(U(z))$$

$$T_{\sigma\times\tau}(x) \equiv T_\sigma(\mathrm{pr}_1(x)) \wedge T_\tau(\mathrm{pr}_2(x)) \quad .$$

Darauf basierend definieren wir durch Induktion über σ eine Relation $\Vdash_\sigma$ zwischen natürlichen Zahlen und Objekten des Typs σ

$$x \Vdash_\iota y \equiv x = y$$

$$x \Vdash_\upsilon y \equiv x = 0$$

$$x \Vdash_{\sigma\to\tau} f \equiv \forall y:\iota.\forall u:\sigma.\ y \Vdash_\sigma u \to \exists z.T(x,y,z) \wedge U(z) \Vdash_\tau f(u)$$

$$x \Vdash_{\sigma\times\tau} u \equiv \mathrm{pr}_1(x) \Vdash_\sigma \pi_1(u) \wedge \mathrm{pr}_2(x) \Vdash_\sigma \pi_2(u) \ .$$

Satz 2.4 *Für jeden Term $x_1{:}\sigma_1, \ldots, x_n{:}\sigma_n \vdash t : \tau$ in Gödel's T gibt es eine natürliche Zahl e, sodaß*

(1) $T_{\sigma_1}(u_1) \wedge \ldots \wedge T_{\sigma_n}(u_n) \to \exists z.\, T(\underline{e}, \langle u_1, \ldots, u_n\rangle, z) \wedge T_\tau(U(z))$
in HA *und*

(2) $u_1 \Vdash_{\sigma_1} x_1 \wedge \ldots \wedge u_n \Vdash_{\sigma_n} x_n \to \exists z{:}\iota.\, T(\underline{e}, \langle u_1, \ldots, u_n\rangle, z) \wedge U(z) \Vdash_\tau t$
in HA_ω

herleitbar ist.

Beweis: Geradlinige Induktion über den Aufbau von System T Termen. □

Korollar 2.3 *Für jeden geschlossenen Term $t : \iota \to \iota$ von Gödels System T gibt es eine Gödelnummer e, sodaß $\forall x.\exists z.\, T(\underline{e}, x, z)$ in* HA *beweisbar ist und $\{e\}(n) = t[n/x]$ für alle n.*

Beweis: Folgt unmittelbar aus Satz 2.4, indem man τ gleich $\iota \to \iota$ setzt, da

$$\underline{e} \Vdash_{\iota\to\iota} t \ \equiv\ \forall x, y.\ x = y \to \exists z.\, T(\underline{e}, x, z) \wedge U(z) = t(y)$$

was äquivalent ist zu $\forall x.\exists z.\, T(\underline{e}, x, z) \wedge U(z) = t(x)$. □