Introduction to Constructive Logic and Mathematics

Thomas Streicher

WS 00/01

Prerequisites

The aim of this course is to give an introduction to constructive logic and mathematics for people who have some basic acquaintance with naive set theory and the formalism of first order logic (propositional connectives and quantifiers) as usually acquired within the first year when studying mathematics or computer science. It were desirable to know the basic notions of computability theory (recursion theory) but for reasons of selfcontainedness we will give a crash course / recap of computability theory later on when we need it. Besides that one should have seen once a construction of the real numbers though its constructive variant will be introduced and studied in detail.

Contents

1 Introduction

Constructive Logic and Mathematics has always existed as a *trend* in mainstream mathematics. However, the need of developing it as a special branch of mathematics did not arise before beginning of the 20th century when mathematics became more abstract and more inconstructive due to the influence of set theory. Inconstructive methods have dominated (the presentation) of 20th century mainstream mathematics. However, during the last 30 years—mainly triggered by the growing influence of computer science—we have experienced an increasing interest in constructive mathematics typically for the following reason.
If we have proved $\forall n.\exists m.A(n,m)$ then we want to read off from this proof an *algorithmic function* f for which we can show that $\forall n.A(n,f(n))$, i.e. we want to extract an algorithm from a proof of existence. Clearly, if $A(n,m)$ is a *decidable* property of natural numbers then from the mere validity of $\forall n.\exists m.A(n,m)$ we obtain a most stupid algorithm computing an m with $A(n,m)$ for every n: search through the natural numbers until you find (the first) m with $A(n,m)$. But very often one can read off a much more intelligent algorithm from a (constructive) proof of $\forall n.\exists m.A(n,m)$.
However, if A is not decidable this is not possible anymore in the general case even if m does not depend on n. Consider for example the formula

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

of pure predicate logic where P is an unspecified predicate constant of arity 1. Classically this is a tautology as if $\forall y.P(y)$ holds then x can be chosen arbitrarily and if $\neg\forall y.P(y)$ then there exists an a with $\neg P(a)$ which we may choose for x. However, what is sort of intriguing is that this proof does not provide us with a concrete a for which we could show $P(a) \to \forall y.\,P(y)$. One easily sees that there cannot exist a term t for which $P(t) \to \forall y.\,P(y)$ is valid (in all models).
One might think that this defect has to do with the general nature of the predicate P whose nature is left absolutely unspecified. But the following example shows that we may run into a problem also for very concrete existence statements.

Theorem 1.1 *There are irrational numbers a and b such that a^b is rational.*

Proof: Suppose that $\sqrt{2}^{\sqrt{2}}$ is rational then put $a = b = \sqrt{2}$. Otherwise if $\sqrt{2}^{\sqrt{2}}$ is irrational then we may put $a = \sqrt{2}^{\sqrt{2}}$ and $b = \sqrt{2}$ which are both irrational but

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

is certainly rational. $\square$

This proof—though undoubtedly correct w.r.t. the usual standards—does not tell us how to choose a and b unless we have decided whether $\sqrt{2}^{\sqrt{2}}$ is rational. But

deciding whether $\sqrt{2}^{\sqrt{2}}$ is rational or not is a non–trivial problem whose answer is not at all implicit in the proof just given.
Another example exhibiting the "defects" of classical logic comes from theoretical computer science. Consider the predicate

$$A(n,m) \quad \equiv \quad m = 0 \leftrightarrow \{n\}(n)\downarrow$$

where $\{n\}$ stands for the partial function computed by the n^{th} Turing machine and $e\downarrow$ stands for "e is is defined" (accordingly we write $e\uparrow$ when e is undefined). Now using classical reasoning one easily proves $\forall n.\exists m.\, A(n,m)$ as if $\{n\}(n)\downarrow$ then put $m = 0$ and otherwise put $m = 1$, say. But, there cannot exist an algorithmic function f with

$$\forall n. f(n) = 0 \leftrightarrow \{n\}(n)\downarrow$$

as otherwise the halting problem were decided by f.[1]
The next example of a non–constructive existence proof is taken from basic classical analysis where one of the first theorems is the following.

Theorem 1.2 *For every* bounded *sequence* $(x_n)_{n\in\mathbb{N}}$ *of real numbers there exists a point of accumulation.*

Proof: Let $[a,b]$ be a rational interval containing all x_n (which exists as by assumption the sequence $(x_n)_{n\in\mathbb{N}}$ is bounded). A point of accumulation is given by the nesting of intervals $[a_n, b_n]_{n\in\mathbb{N}}$ which is "constructed" as follows: put $[a_0, b_0] := [a,b]$ and

$$[a_{n+1}, b_{n+1}] = \begin{cases} [a_n, \frac{a_n+b_n}{2}] & \text{if } \forall m.\exists k \geq m. x_k \in [a_n, \frac{a_n+b_n}{2}] \\ [\frac{a_n+b_n}{2}, b_n] & \text{otherwise .} \end{cases}$$

□

Notice that the "construction" of $[a_{n+1}, b_{n+1}]$ from $[a_n, b_n]$ cannot be performed algorithmically as one has to decide whether the interval $[a_n, \frac{a_n+b_n}{2}]$ contains infinitely many items of the sequence $(x_n)_{n\in\mathbb{N}}$ which surely cannot be achieved in finite time.

[1]It is a basic fact from recursion theory that $K = \{n \in \mathbb{N} \mid \{n\}(n)\downarrow\}$ is not decidable. Otherwise there would exist a Gödelnumber e such that

$$\{e\}(n)\downarrow \quad \text{iff} \quad \{n\}(n)\uparrow$$

leading to the contradiction

$$\{e\}(e)\downarrow \quad \text{iff} \quad \{e\}(e)\uparrow$$

when instantiating n by e (a trick called *diagonalisation*). Obviously, from the undecidability of K there follows immediately the undecidability of the halting set $H = \{\langle n,m\rangle \mid \{n\}(m)\downarrow\}$.

Notice that diagonalisation arguments are all instances of the (constructive) tautology $\neg\exists x.\forall y.\, R(x,y) \leftrightarrow \neg R(y,y)$.

Thus, apparently the use of classical reasoning allows one to prove existential statements whose witnessing object cannot be read off from the argument. Critical examination of the above examples tells us that this lack of constructivity originates from the use of the (classically correct) principles

(1) $\forall y.P(y) \vee \neg\forall y.P(y)$

(2) $\neg\forall y.P(y) \leftrightarrow \exists y.\neg P(y)$

(3) $\sqrt{2}^{\sqrt{2}}$ is either rational or irrational

(4) $\{n\}(n)\downarrow \vee \{n\}(n)\uparrow$

(5) $(\forall m.\exists k \geq m.x_k \in [a_n, \frac{a_n+b_n}{2}]) \vee \neg(\forall m.\exists k \geq m.x_k \in [a_n, \frac{a_n+b_n}{2}])$.

Propositions (1), (3), (4) and (5) are instances of the schema

PEM $\qquad A \vee \neg A$

called *Principle of Excluded Middle* characteristic for classical logic. Use of PEM may render proofs inconstructive as in general we cannot decide for a proposition A whether A or $\neg A$ holds. This is typically the case if A is of the form $\forall x.B(x)$ as one would have to examine infinitely many $B(n)$ if x ranges e.g. over the natural numbers.
As long as one is not interested in extracting algorithms from existence proofs there is nothing to complain about classical logic. Most steps even in a classical proof are constructive but sometimes there is made appeal to an oracle deciding the truth of a proposition.
In a sense it is true that constructive logic is obtained from classical logic by omitting PEM. But the question is what are the constructively valid principles of reasoning. This is not so easy to answer as there are logical principles which at first sight look different from PEM but whose proof requires PEM or is even equivalent to it. For example the logical principle (2) above is usually proved via PEM making a case analysis on $\exists y.\neg P(y)$: if $\exists y.\neg P(y)$ then $\neg\forall y.P(y)$ and if $\neg\exists y.\neg P(y)$ then $\forall y.\neg\neg P(y)$ from which it follows that $\neg\forall y.\neg P(y)$. Another logical principle familiar from classical reasoning is *reductio ad absurdum*

RAA $\qquad \neg\neg A \rightarrow A$

where in order to prove A one refutes $\neg A$. We shall see later that the principles PEM and RAA are equivalent and adding either of them to constructive logic one obtains classical logic.
Having seen that it is not so obvious to identify what are "constructively valid" principles of reasoning we will discuss this question next. Usually, classical validity is explained in terms of truth values, i.e. one explains what is the truth value of a compound formula in terms of the truth values of its constituent formulas.

This will not work for constructive validity as ordinary truth value semantics does validate PEM. Accordingly, the *constructive meaning* of propositions is explained best in terms of an alternative semantics based on an (informal) notion of “proof” instead of an (informal) notion of “truth value”. *Here “proof” should not be understood in the sense of a formal proof in some logical calculus as given for example by a derivation tree but rather as an informal basic notion (like truth in case of classical logic).* One can say that the meaning of constructive logic is best understood in terms of a *proof semantics* as opposed to the well-known *truth–value semantics* appropriate for classical logic. What we have called “proof semantics” is often called *Brouwer–Heyting–Kolmogoroff Interpretation* (or simply BHK Interpretation) after the people who brought it up.

Proof Semantics of Constructive Logic

Conjunction
A proof of $A \wedge B$ is a pair $\langle p, q \rangle$ where p is a proof of A and q is a proof of B.

Implication
A proof of $A \to B$ is a (constructive) function f mapping proofs of A to proofs of B, i.e. $f(p)$ is a proof of B whenever p is a proof of A.

Falsity
There is no proof of $\bot$ (*falsity*).

Disjunction
A proof of $A \vee B$ is either a proof of A or a proof of B where it is indicated (e.g. by a label) whether it proves A or B.

Universal Quantification
A proof of $\forall x.A(x)$ is a (constructive) function f such that $f(d)$ is a proof of $A(d)$ for all $d \in D$ where D is the domain of discourse (over which the variable x ranges).

Existential Quantification
A proof of $\exists x.A(x)$ is a pair $\langle d, p \rangle$ where $d \in D$ and p is a proof of $A(d)$ where D is the domain of discourse (over which the variable x ranges).

The clauses for implication and universal quantification may appear as “circular” and actually are. However, this is the case as well for the traditional 2–valued truth value semantics of classical logic where the logic to be explained is used on the meta–level.[2] Accordingly, the BHK interpretation must not be understood as

[2]If in classical logic one defines $A \to B$ as $\neg A \vee B$ then implication gets reduced to something more primitive (in terms of the unexplained notions of negation and disjunction). However, this

a precise mathematical definition but rather as an *informal, but intuitive explanation of meaning* just as the ordinary truth semantics for classical logic. However, as we shall see later on there are variations of the BHK interpretation which have a precise technical meaning and are most useful in the sense that they provide interesting models of constructive logic and allow us to give transparent proofs of metamathematical properties.

2 Natural Deduction

In this section we introduce a derivation calculus for constructive predicate calculus which as close as possible reflects the structure of actual mathematical proofs and, therefore, has been baptized calculus of "Natural Deduction"[3]. Of course, derivations in this calculus are much more detailed than actual mathematical arguments. It is not intended to develop constructive mathematics in a purely formal way within such a calculus but rather to use it as a mathematical model of actual constructive reasoning for which one may prove certain metamathematical properties exhibiting the nature of actual constructive reasoning.[4]
Notice that the syntax of predicate logic employed here deviates from the usual practice in one particular aspect: instead of having negation as a basic propositional connective we introduce a propositional constant $\bot$ ('falsity') for the false proposition and introduce negation via the 'macro' $\neg A \equiv A \to \bot$. It is clear that under the usual 2–valued truth semantics we have that $A \to \bot$ is true iff A is false and, therefore, this 'implementation' of negation is in accordance the usual understanding of negation in classical logic.
We suggest it as an informative exercise to explain the validity of the proof rules of the following definition in terms of the BHK interpretation.

Definition 2.1 *Sequents are expressions of the form*

$$A_1, \ldots, A_n \vdash B$$

"explanation" of implication was always considered as somewhat contrived and not (properly) reflecting the intuitive understanding of implication which seems to be brought to the point by the above clause for implication though in a somewhat "circular" way. On the other hand the BHK interpretation provides a proper reduction of disjunction and existential quantification which are the two connectives where constructive logic really deviates from classical logic.

[3]Natural Deduction was introduced by G. Gentzen back in the 30ies as a mathematical model of mathematical proofs. His aim was to prove—as strongly advocated by D. Hilbert—the consistency of arithmetic, i.e. that there is no derivation of a false proposition. Generally, the endeavour of analyzing proofs by mathematical methods is called *proof theory*. Nowadays, however, consistency proofs are not the holy grail of proof theory anymore as their foundational value is more than debatable due to Gödel's (Second) Incompletness Theorem.

[4]Of course, the same applies to formalisations of classical logic. However, as classical reasoning is more familiar to most mathematicians than constructive reasoning for studying the latter it is worthwhile to explicitly state the proof rules from the very beginning.

where the A_i and B are formulas of predicate logic. The intended meaning is that the assumptions $A_1, \ldots, A_n$ entail *conclusion B. The valid sequents of* **CPL** (Constructive Predicate Logic) *are defined inductively via the following proof rules.*

Structural Rules

$$\frac{}{\Gamma, A, \Delta \vdash A}\,(\mathrm{ax}) \qquad \frac{\Gamma, A, B, \Delta \vdash C}{\Gamma, B, A, \Delta \vdash C}\,(\mathrm{ex})$$

$$\frac{\Gamma \vdash C}{\Gamma, A \vdash C}\,(\mathrm{w}) \qquad \frac{\Gamma, A, A \vdash C}{\Gamma, A \vdash C}\,(\mathrm{c})$$

Propositional Connectives

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B}\,(\wedge I) \qquad \frac{\Gamma \vdash A_1 \wedge A_2}{\Gamma \vdash A_i}\,(\wedge E_i)$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\,(\to I) \qquad \frac{\Gamma \vdash A \to B \quad \Gamma \vdash A}{\Gamma \vdash B}\,(\to E)$$

$$\frac{\Gamma \vdash A_i}{\Gamma \vdash A_1 \vee A_2}\,(\vee I_i) \qquad \frac{\Gamma \vdash A \vee B \quad \Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma \vdash C}\,(\vee E)$$

$$\frac{\Gamma \vdash \bot}{\Gamma \vdash C}\,(\bot E)$$

Quantifiers

$$\frac{\Gamma \vdash A(x) \quad x \notin FV(\Gamma)}{\Gamma \vdash \forall x.A(x)}\,(\forall I) \qquad \frac{\Gamma \vdash \forall x.A(x)}{\Gamma \vdash A(t)}\,(\forall E)$$

$$\frac{\Gamma \vdash A(t)}{\Gamma \vdash \exists x.A(x)}\,(\exists I) \qquad \frac{\Gamma \vdash \exists x.A(x) \quad \Gamma, A(x) \vdash C \quad x \notin FV(\Gamma, C)}{\Gamma \vdash C}\,(\exists E)$$

$\diamondsuit$

Notice that there are two elimination rules $(\wedge E_1)$ and $(\wedge E_2)$ for conjunction and two introduction rules $(\vee I_1)$ and $(\vee I_2)$ for disjunction.

It is absolutely necessary to take the *variable conditions* serious in rules $(\forall I)$ and $(\exists E)$ as the following counterexamples show where faulty applications of these rules are marked by †.
That in rule $(\forall I)$ the variable x must not occur in the premiss is demonstrated by the following pseudo–derivation

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A(x)\vdash A(x)}\,(\mathrm{ax})}{A(x)\vdash \forall x.A(x)}\,(\forall I)^\dagger}{\vdash A(x)\to\forall x.A(x)}\,(\to I)}{\vdash \forall x.(A(x)\to\forall x.A(x))}\,(\forall I)}{\vdash A(t)\to\forall x.A(x)}\,(\forall E)$$

That in rule $(\exists E)$ the variable x must not occur freely in C is demonstrated by the following pseudo–derivation

$$\dfrac{\dfrac{\dfrac{}{\exists x.A(x)\vdash \exists x.A(x)}\,(\mathrm{ax}) \qquad \dfrac{}{\exists x.A(x), A(x)\vdash A(x)}\,(\mathrm{ax})}{\exists x.A(x)\vdash A(x)}\,(\exists E)^\dagger}{\exists x.A(x)\vdash \forall x.A(x)}\,(\forall I)$$

That in rule $(\exists E)$ the variable x must not occur freely in Γ is shown by the following pseudo–derivation

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma\vdash \exists x.\neg A(x)}\,(\mathrm{ax}) \qquad \dfrac{\dfrac{}{\Gamma,\neg A(x)\vdash \neg A(x)}\,(\mathrm{ax}) \quad \dfrac{}{\Gamma,\neg A(x)\vdash A(x)}\,(\mathrm{ax})}{\Gamma,\neg A(x)\vdash \bot}\,(\to E)}{A(x),\exists x.\neg A(x)\vdash \bot}\,(\exists E)^\dagger}{\exists x.\neg A(x)\vdash \neg A(x)}\,(\to I)}{\exists x.\neg A(x)\vdash \forall x.\neg A(x)}\,(\forall I)$$

where $\Gamma \equiv A(x), \exists x.\neg A(x)$.
The consideration of these pseudo–derivations should have already conveyed some feeling for how to construct formal derivations in the calculus of Natural Deduction. But, of course, in correct derivations in all applications of the rules $(\forall I)$ and $(\exists E)$ the variable conditions have to be fulfilled. We now consider quite a few examples of correct derivations in **CPL** as well as derived rules not just for getting familiar with the practice of the construction of formal proofs but also to have these constructive logical principles available for subsequent use.

The derivation

$$\dfrac{\dfrac{\dfrac{}{A, \neg A \vdash \neg A}\,(\mathrm{ax}) \qquad \dfrac{}{A, \neg A \vdash A}\,(\mathrm{ax})}{A, \neg A \vdash \bot}\,(\to E)}{A \vdash \neg\neg A}\,(\to I)$$

demonstrates that a proposition entails its double negation.
However, a negated formula $\neg A$ follows from its double negation $\neg\neg\neg A$ via the derivation

$$\dfrac{\dfrac{\dfrac{}{\neg\neg\neg A, A \vdash \neg\neg\neg A}\,(\mathrm{ax}) \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{A, \neg A \vdash \neg A}\,(\mathrm{ax}) \qquad \dfrac{}{A, \neg A \vdash A}\,(\mathrm{ax})}{A, \neg A \vdash \bot}\,(\to E)}{\neg\neg\neg A, A, \neg A \vdash \bot}\,(\mathrm{w})}{\neg\neg\neg A, A \vdash \neg\neg A}\,(\to I)}{\neg\neg\neg A, A \vdash \bot}\,(\to E)}{\neg\neg\neg A \vdash \neg A}\,(\to I)$$

and, therefore, we easily can derive

$$\neg A \leftrightarrow \neg\neg\neg A$$

from which it follows that double negation is idempotent, i.e. that

$$\neg\neg A \leftrightarrow \neg\neg\neg\neg A$$

is derivable.[5] This observation allows us to reduce multiple negations either to simple negation or double negation. Classically, we also have $\neg\neg A \to A$ (and therefore also $\neg\neg A \leftrightarrow A$) which logical principle is called *reductio ad absurdum* and distinguishes classical logic from constructive logic as we shall see later.
Next we show that

$$A \vee B \to C \dashv\vdash (A \to C) \wedge (B \to C)$$

abbreviating

$$A \vee B \to C \vdash (A \to C) \wedge (B \to C) \quad \text{and} \quad (A \to C) \wedge (B \to C) \vdash A \vee B \to C$$

as suggested by the notation.

[5]The phrase "A is derivable" is an abbreviation for "$\vdash A$ is derivable". Moreover, notice that $A \leftrightarrow B$ is a 'macro' for $(A \to B) \wedge (B \to A)$.

We have

$$\dfrac{\dfrac{\dfrac{}{A \lor B \to C, A \vdash A \lor B \to C}\,(\text{ax}) \qquad \dfrac{\dfrac{}{A \lor B \to C, A \vdash A}\,(\text{ax})}{A \lor B \to C, A \vdash A \lor B}\,(\lor I_1)}{A \lor B \to C, A \vdash C}\,(\to E)}{A \lor B \to C \vdash A \to C}\,(\to I)$$

and similarly one proves $A \lor B \to C \vdash B \to C$ form which it follows that $A \lor B \to C \vdash (A \to C) \land (B \to C)$.
Now we derive the reverse direction. We have

$$\dfrac{\dfrac{\dfrac{}{\Gamma \vdash A \lor B}\,(\text{ax}) \qquad \Gamma, A \vdash C \quad \Gamma, B \vdash C}{(A \to C) \land (B \to C), A \lor B \vdash C}\,(\lor E)}{(A \to C) \land (B \to C) \vdash A \lor B \to C}\,(\to I)$$

where Γ stands for the context $(A{\to}C) \land (B{\to}C), A \lor B$. The open assumption $\Gamma, A \vdash C$ can be derived as follows

$$\dfrac{\dfrac{\dfrac{}{\Gamma, A \vdash (A \to C) \land (A \to C)}\,(\text{ax})}{\Gamma, A \vdash A \to C}\,(\land E_1) \qquad \dfrac{}{\Gamma, A \vdash A}\,(\text{ax})}{\Gamma, A \vdash C}\,(\to E)$$

and similarly for the other open assumption $\Gamma, B \vdash C$.
Notice that from $((A \to C) \land (B \to C)) \leftrightarrow (A \lor B \to C)$ there immediately follows the 'deMorgan' law

$$\neg A \land \neg B \leftrightarrow \neg(A \lor B)$$

instantiating C by $\bot$. Further instantiating B by $\neg A$ we get

$$\neg(A \lor \neg A) \leftrightarrow (\neg A \land \neg\neg A)$$

and as one easily derives

$$\neg(\neg A \land \neg\neg A)$$

it follows (Exercise!) that

$$\neg\neg(A \lor \neg A)$$

holds constructively. Thus, by *reductio ad absurdum*[6] we get PEM. On the other hand PEM entails *reductio ad absurdum* as can be seen from the derivation

$$\dfrac{\dfrac{\dfrac{}{\Gamma \vdash A \vee \neg A}\,(\text{ax}) \quad \dfrac{}{\Gamma, A \vdash A}\,(\text{ax}) \quad \dfrac{\dfrac{}{\Gamma, \neg A \vdash \neg\neg A}\,(\text{ax}) \quad \dfrac{}{\Gamma, \neg A \vdash \neg A}\,(\text{ax})}{\Gamma, \neg A \vdash A}\,(\to E)}{A \vee \neg A, \neg\neg A \vdash A}\,(\vee E)}{A \vee \neg A \vdash \neg\neg A \to A}\,(\to I)$$

where $\Gamma \equiv A \vee \neg A, \neg\neg A$.
Next we show that

$$\exists x.A(x) \to B \dashv\vdash \forall x.(A(x) \to B)$$

provided x is not free in B. The direction from left to right is given by the derivation

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\exists x.A(x) \to B, A(x) \vdash \exists x.A(x) \to B}\,(\text{ax}) \quad \dfrac{\dfrac{}{\exists x.A(x) \to B, A(x) \vdash A(x)}\,(\text{ax})}{\exists x.A(x) \to B, A(x) \vdash \exists x.A(x)}\,(\exists I)}{\exists x.A(x) \to B, A(x) \vdash B}\,(\to E)}{\exists x.A(x) \to B \vdash A(x) \to B}\,(\to I)}{\exists x.A(x) \to B \vdash \forall x.(A(x) \to B)}\,(\forall I)$$

and the reverse direction is given by the derivation

$$\dfrac{\dfrac{\dfrac{}{\Gamma \vdash \exists x.A(x)}\,(\text{ax}) \quad \dfrac{\dfrac{\dfrac{}{\Gamma, A(x) \vdash \forall x.(A(x) \to B)}\,(\text{ax})}{\Gamma, A(x) \vdash A(x) \to B}\,(\forall E) \quad \dfrac{}{\Gamma, A(x) \vdash A(x)}\,(\text{ax})}{\Gamma, A(x) \vdash B}\,(\to E)}{\forall x.(A(x) \to B), \exists x.A(x) \vdash B}\,(\exists E)}{\forall x.(A(x) \to B) \vdash \exists x.A(x) \to B}\,(\to I)$$

where $\Gamma \equiv \forall x.(A(x) \to B), \exists x.A(x)$. Instantiating B by $\bot$ we get the 'deMorgan' law

$$\neg\exists x.A(x) \leftrightarrow \forall x.\neg A(x) \quad .$$

Notice, however, that as we shall see later the dual deMorgan law

$$\neg\forall x.A(x) \leftrightarrow \exists x.\neg A(x)$$

[6]But notice that the schema $\neg\neg A \to A$ has to be instantiated by $A \vee \neg A$! Actually, as we will see later on for a particular proposition it may hold that $\neg\neg A \to A$ but *not* $A \vee \neg A$.

is not constructively valid.
We conclude this section by exhibiting some *derived rules.*
First of all we have the most useful *cut rule*

$$\frac{\Gamma \vdash A \qquad \Gamma, A \vdash B}{\Gamma \vdash B}\,(\text{cut})$$

which allows us to remove unnecessary assumptions from a sequent. Its correctness is immediate from the derivation

$$\frac{\dfrac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\,(\to I) \qquad \Gamma \vdash A}{\Gamma \vdash B}\,(\to E)$$

with open assumptions $\Gamma \vdash A$ and $\Gamma, A \vdash B$.
By definition for every connective of CPC there is a rule for its introduction on the right of the turnstile. The following derived rules allow one to introdcue compound formulas on the left of the turnstile

$$\frac{\Gamma, A, B \vdash C}{\Gamma, A \wedge B \vdash C}\,(\wedge L) \qquad\qquad \frac{\Gamma, A \vdash C \qquad \Gamma, B \vdash C}{\Gamma, A \vee B \vdash C}\,(\vee L)$$

$$\frac{\Gamma \vdash A \qquad \Gamma, B \vdash C}{\Gamma, A \to B \vdash C}\,(\to L) \qquad\qquad \frac{}{\Gamma, \bot \vdash C}\,(\bot L)$$

$$\frac{\Gamma, A(t) \vdash C}{\Gamma, \forall x.A(x) \vdash C}\,(\forall L) \qquad\qquad \frac{\Gamma, A(x) \vdash C \qquad x \notin FV(\Gamma, C)}{\Gamma, \exists x.A(x) \vdash C}\,(\exists L)$$

We leave the verification of their correctness to the diligent reader.
Notice that rule $(\wedge L)$ can be inverted, i.e. we have

$$\Gamma, A, B \vdash C \qquad \text{iff} \qquad \Gamma, A \wedge B \vdash C$$

and, therefore,

$$A_1, \ldots, A_n \vdash C \qquad \text{iff} \qquad A_1 \wedge \ldots \wedge A_n \vdash C$$

allowing us to reduce sequents to entailment between formulas.

We leave it as an exercise to show that the equivalences

$$\begin{array}{ccc} C \vdash A \wedge B & \text{iff} & C \vdash A \quad \text{and} \quad C \vdash B \\ A \vee B \vdash C & \text{iff} & A \vdash C \quad \text{and} \quad B \vdash C \\ C \vdash A \to B & \text{iff} & C \wedge A \vdash B \\ C \vdash \forall x.A(x) & \text{iff} & C \vdash A(x) \\ \exists x.A(x) \vdash C & \text{iff} & A(x) \vdash C \end{array}$$

where in the last two clauses $x \notin FV(C)$ together with the rules

$$\frac{}{A \vdash A} \qquad \frac{}{\bot \vdash A} \qquad \frac{A \vdash B \qquad B \vdash C}{A \vdash C} \qquad \frac{A \vdash B}{A[t/x] \vdash B[t/x]}$$

provide an alternative axiomatization of CPC. This axiomatization of entailment between formulas will turn out as most useful when considering semantics.

3 A Hilbert Style System

Though the calculus of Natural Deduction most closely reflects the structure of actual mathematical proofs (when spelt out in detail) for metamathematical purposes it is often convenient to use a Hilbert style system which inductively defines the set of all formulas A such that $\vdash A$ can be derived by Natural Deduction. The rules of such an inductive definition are of the form

$$A_1, \ldots, A_n \Rightarrow B$$

which means that B is provable provided $A_1, \ldots, A_n$ are all provable. If $n = 0$ then we simply write A instead of $\Rightarrow A$.

Theorem 3.1 *The set of all formulas A of predicate logic for which the sequent $\vdash A$ is derivable in the calculus of Natural Deduction is defined inductively by the following rules*

(L1) $A \to A$

(L2) $A \ , \ A \to B \Rightarrow B$

(L3) $A \to B \ , \ B \to C \Rightarrow A \to C$

(L4) $A \wedge B \to A \ , \ A \wedge B \to B$

(L5) $C \to A \ , \ C \to B \Rightarrow C \to A \wedge B$

(L6) $A \to A \lor B \ , \ B \to A \lor B$

(L7) $A \to C \ , \ B \to C \Rightarrow A \lor B \to C$

(L8) $A \land B \to C \Rightarrow A \to B \to C$

(L9) $A \to B \to C \Rightarrow A \land B \to C$

(L10) $\bot \to A$

(L11) $B \to A(x) \Rightarrow B \to \forall x.A(x) \quad (x \notin FV(B))$

(L12) $\forall x.A \to A(t)$

(L13) $A(t) \to \exists x.A$

(L14) $A(x) \to B \Rightarrow \exists x.A(x) \to B \quad (x \notin FV(B))$.

Proof: One easily shows that if A can be derived via the rules (L1)–(L12) then $\vdash A$ can be proved by Natural Deduction.
For the reverse direction one shows that if $A_1, \dots, A_n \vdash B$ can be derived in the calculus of natural deduction then the formula $A_1 \to \dots \to A_n \to B$ is derivable via the rules (L1)–(L14). □

4 Truth–Value Semantics of Constructive Logic

Although proof semantics is conceptually more adequate as an explanation of the constructive meaning of propositional connectives and quantifiers we will introduce in this section a truth–value semantics as an alternative. We will mainly use it for obtaining *independence* results, i.e. for showing that classical tautologies (as e.g. $A \lor \neg A$ or $\neg\neg A \to A$) cannot be proved constructively. It will turn out that semantic proofs of underivability will be fairly simple due to their 'algebraic' nature whereas direct purely syntactical proofs would be very cumbersome as they require a careful analysis of derivations as mathematical objects.
The idea of truth–value semantics is to assign every formula a truth value, i.e. an element of some structured set P of propositions.[7] The distinguishing structure on propositions is given by a partial order $\leq$ thought of as entailment. In the light of the discussion at the end of Section 2 conjunction/disjunction will be interpreted as (binary) supremum/infimum and falsity will be interpreted as the least element of the partial order $(P, \leq)$. Implication will turn out as another binary operation which, however, is also fully determined by $\leq$. Such a kind of poset (= partially ordered set) is axiomatized as follows.

[7] We have the tendency to understand the word 'proposition' in a semantical sense. A formula or a sentences *denotes* a proposition. But we will not be consequent in this respect.

Definition 4.1 *A* Heyting algebra *or* Heyting lattice *is a poset $(A, \leq)$ with finite infima and suprema such that for all $a, b \in A$ there exists $a \to b \in A$ with*

$$c \leq a \to b \qquad \textit{iff} \quad c \wedge a \leq b$$

for all $c \in A$. Observe that $a \to b$ is uniquely determined by this property and, therefore, gives rise to a binary operation $\to$ on A called Heyting implication.

Notice that (in accordance with the subsequent definition of interpretation) we write $\wedge$ for binary infimum and $\vee$ for binary supremum. Further notice that the existence of a greatest element $\top$ in A is already ensured by $\to$ as $c \leq \bot \to \bot$ holds for all $c \in A$ (where $\bot$ stands for the least element of A).
Heyting algebras provide enough structure to interpret the propositional part of constructive logic.

Definition 4.2 *Let A be a Heyting algebra. A mapping ρ from PC, the set of propositional constants, to A is called a valuation (of propositional constants in A). Such a valuation ρ induces an interpretation of formulas of propositional logic as given by the following inductive clauses*

$$\begin{array}{lcl}
[\![p]\!]\rho & = & \rho(p) \\
[\![A \wedge B]\!]\rho & = & [\![A]\!]\rho \wedge_A [\![B]\!]\rho \\
[\![A \vee B]\!]\rho & = & [\![A]\!]\rho \vee_A [\![B]\!]\rho \\
[\![A \to B]\!]\rho & = & [\![A]\!]\rho \to_A [\![B]\!]\rho \\
[\![\bot]\!]\rho & = & \bot_A
\end{array}$$

where $\wedge_A$, $\vee_A$, $\to_A$ and $\bot_A$ stand for binary infimum, binary supremum, Heyting implication and least element in A.

The proof of the following theorem is straightforward and recommended as an instructive exercise as it reveals the 'design decisions' behind the definition of Heyting algebra.

Theorem 4.1 *If there is a derivation of the sequent $A_1, \ldots, A_n \vdash B$ in (the propositional part of) the calculus of Natural Deduction then*

$$[\![A_1]\!]\rho \wedge_A \ldots \wedge_A [\![A_1]\!]\rho \leq_A [\![B]\!]\rho$$

for every Heyting algebra A and valuation ρ in A.

This correctness theorem w.r.t. interpretation in Heyting algebras allows us to show by purely semantical means that a sequent $A_1, \ldots, A_n \vdash B$ is not derivable, namely by exhibiting appropriate A and ρ such that

$$[\![A_1]\!]\rho \wedge_A \ldots \wedge_A [\![A_1]\!]\rho \not\leq_A [\![B]\!]\rho \quad .$$

Next we discuss a wide class of Heyting algebras among which we will find the counterexamples needed for our independence results.

Example 4.1 *Let X be a topological space. We write $\mathcal{O}(X)$ for the poset of open subsets of X under subset inclusion. Then $\mathcal{O}(X)$ is a complete Heyting algebra. The poset has arbitrary joins given by set–theoretic union. Therefore, $\mathcal{O}(X)$ has also arbitrary infima (given by the interior of intersections). Thus, $\mathcal{O}(X)$ is a complete lattice, i.e. a poset with arbitrary infima and suprema. Notice that finite infima are given by ordinary set–theoretic intersections as open sets are by definition required to be closed under finite intersections. Accordingly, in $\mathcal{O}(X)$ there holds the following* infinitary distributive law

$$U \wedge \bigvee_{i\in I} V_i = \bigvee_{i\in I} U \wedge V_i$$

as

$$U \cap \bigcup_{i\in I} V_i = \bigcup_{i\in I} U \cap V_i$$

holds for sets and joins and finite meets in $\mathcal{O}(X)$ are given by unions and finite intersections, respectively. Due to this infinitary distributive law Heyting implication in $\mathcal{O}(X)$ exists and is given by

$$U \to V = \bigcup \{W \mid U \cap W \subseteq V\}$$

as it immediately follows from the infinite distributive law that $(U \to V) \cap U \subseteq V$ and $U \cap W \subseteq V$ implies $W \subseteq U \to V$.

The following particular instances will be most useful for providing counterexamples.

Example 4.2 *Let P be a poset and*

$$\mathsf{dcl}(P) := \{A \in \mathcal{P}(P) \mid y \leq x \in A \Rightarrow y \in A\}$$

be the set of downward closed subsets of P partially ordered by $\subseteq$. As $\mathcal{P}(P)$ is closed under arbitary unions and arbitrary intersections it certainly is a complete Heyting algebra (coming from a topological space). In this particular case Heyting implication is given by

$$U{\to}V = \{p \in P \mid \forall q \leq p.\, q \in U \Rightarrow q \in V\}$$

where $\downarrow\!p = \{q \in P \mid q \leq p\}$. This readily follows from the fact that $U{\to}V = \bigcup\{\downarrow\!p \in P \mid U \cap \downarrow\!p \subseteq V\}$ (Exercise!).

Having these classes of Heyting algebras at hand we can easily show that PEM and *reductio ad absurdum* are not derivable.

Theorem 4.2 *In the calculus of Natural Deduction one cannot derive*

$$\neg\neg p \to p \qquad p \vee \neg p \qquad \neg p \vee \neg q \leftrightarrow \neg(p \wedge q)$$

for propositional constants p and q.

Proof: Let A be the Heyting algebra $\mathsf{dcl}(\mathbf{2})$ where $\mathbf{2}$ is the poset $0 < 1$ (i.e. the ordinal 2). Let $u = {\downarrow}0$ for which we have

$$\neg u = \bot \qquad \text{and} \qquad \neg\neg u = \top$$

and, therefore,

$$\neg\neg u = \top > u \qquad \text{and} \qquad u \vee \neg u = u \vee \bot = u < \top \qquad .$$

Thus, PEM and *reductio ad absurdum* cannot be derived.
For refuting $\neg p \vee \neg q \leftrightarrow \neg(p \wedge q)$ consider the poset $a < 1 > b$ with a and b incomparable. Let $p = {\downarrow}a$ and $q = {\downarrow}b$. Then we have $\neg p = q$ and $\neg q = p$ and, therefore,

$$\neg p \vee \neg q = {\downarrow}\{a, b\} \neq \{a, b, 1\} = \neg\emptyset = \neg(p \wedge q) \quad .$$

$\square$

For complete Heyting algebras one may extend the interpretation of constructive propositional logic to predicate logic in the following way. One chooses a set D as universe of discourse and n-ary predicates are interpreted as functions from D^n to A. Universal and existential quantification are interpreted as (possibly) infinite joins and meets in A which exist as A is assumed to be complete. More precisely, we have

$$\begin{array}{rcl} [\![\exists x.A(x)]\!]e & = & \bigvee_{d \in D} [\![A]\!]e[d/x] \\ [\![\forall x.A(x)]\!]e & = & \bigwedge_{d \in D} [\![A]\!]e[d/x] \end{array}$$

where e is an environment sending (object) variables to elements of A. This interpretation of constructive predicate logic is quite in accordance with the usual Tarskian semantics of classical predicate logic. The difference just is that the 2 element Heyting algebra is replaced by arbitrary complete Heyting algebras.
Though Heyting–valued models of CPC are an interesting topic[8] we refrain from investigating them in greater detail.

Intermezzo on Kripke Semantics

It might be illuminating to give a more concrete version of the interpretation of propositional logic in cHa's of the form $\mathsf{dcl}(P)$ commonly known as Kripke semantics. The key idea of Kripke semantics is to think of the poset P as a set of 'possible worlds' and of $v \leq u$ as "v is a possible development of u". The meaning of a proposition then is considered as a set of possible worlds "closed under all possible developments", i.e. as a downward closed subset of P.

[8]The adequate setting for Heyting–valued semantics is the theory of *sheaves* over a topological space or a cHa (complete Heyting algebra). Categories of sheaves are the basic examples of so–called *toposes* which constitute a further even more general notion of model for higher order constructive logic.

Let ρ be some valuation of propositional constants in $\mathsf{dcl}(P)$. For $u \in P$ we write $u \Vdash A$ for $u \in [\![A]\!]\rho$. Usually, $u \Vdash A$ is read as "u forces A". Obviously, for the 'forcing' relation $\Vdash$ it holds that

$u \Vdash p$	iff	$u \in \rho(p)$
$u \Vdash \bot$		never
$u \Vdash A \wedge B$	iff	$u \Vdash A$ and $u \Vdash B$
$u \Vdash A \vee B$	iff	$u \Vdash A$ or $u \Vdash B$
$u \Vdash A \to B$	iff	$v \Vdash B$ forall $v \leq u$ with $v \Vdash A$

which amounts to an inductive redefinition of $\Vdash$ closer to the 'spirit' of the original Tarskian definition of truth. But notice that $u \Vdash \neg A$ iff $v \nVdash A$ for all $v \leq u$ and in general for $u \Vdash \neg A$ it is not sufficient that $u \nVdash A$!

5 Embedding Classical into Constructive Logic

At first sight one may be inclined to consider constructive logic as a properly weaker fragment of classical logic as the latter proves more theorems than the former. However, as we shall show in this section this (possible) view is quite misleading as there is a full embedding of classical logic into the fragment of constructive logic based on $\bot$, $\wedge$, $\to$ and $\forall$. The point is that the constructive meaning of $\vee$ and $\exists$ is incomparable with the classical meaning of these connectives. It will turn out that the classical disjunction of A and B can be expressed constructively as $\neg(\neg A \wedge \neg B)$ and classical existential quantification can be expressed as $\neg\forall\neg$.
Before turning to syntax let us study the relation between classical and constructive logic on the level of the truth–value semantics introduced in the previous section. Usually a model for classical propositional logic is given by a so–called Boolean algebra which we define as follows.

Definition 5.1 *A* Boolean algebra *is a Heyting algebra B such that*

$$\neg\neg a \leq a$$

for all $a \in B$.

Equivalently one may define a Boolean algebra as a Heyting algebra B with

$$a \vee \neg a = \top$$

for all all $a \in B$ (Exercise!). Usually, Boolean algebras are defined as a certain algebraic structure satisfying a long list of equational axioms. However, we think that the equivalent formulation given in Definition 5.1 is easier to memorize and easier to work with. We suggest it as a (not too difficult) exercise to show that $\mathsf{dcl}(P)$ is a Boolean algebra if and only if all elements of P are incomparable.
The next theorem shows that within any Heyting algebra one can find a Boolean algebra.

Theorem 5.1 *Let A be a Heyting algebra and*

$$B := A_{\neg\neg} := \{\neg a \mid a \in A\}$$

be the set of $\neg\neg$–closed *elements of A. This terminology makes sense as $a \in B$ iff $\neg\neg a \leq a$. The set B of $\neg\neg$–closed elements of A satisfies the following closure properties*

(1) $\bot, \top \in B$

(2) *B is closed under $\wedge$*

(3) *$a \to b \in B$ whenever $b \in B$ (for arbitrary $a \in A$).*

Moreover, B considered as a sub–poset of A is a Boolean algebra where $b_1 \vee_B b_2 = \neg(\neg b_1 \wedge \neg b_2)$.

Proof: Let $a \in A$. Then we have $a \leq \neg\neg a$ as $a \wedge \neg a \leq \bot$ and, therefore, $\neg\neg\neg a \leq \neg a$ as $\neg\neg\neg a \wedge a \leq \neg\neg\neg a \wedge \neg\neg a \leq \bot$. Thus, it follows that $A_{\neg\neg}$ consists precisely of negations of elements of A.
As $\bot = \top \to \bot$ and $\top = \bot \to \bot$ it follows that $\top$ and $\bot$ are in $A_{\neg\neg}$.
Suppose $a, b \in A_{\neg\neg}$, i.e. $\neg\neg a = a$ and $\neg\neg b = b$. Then $a \wedge b = \neg\neg a \wedge \neg\neg b = \neg(\neg a \vee \neg b) \in A_{\neg\neg} = B$.
Let $a \in A$ and $b \in B$. Then $\neg\neg b = b$ and, therefore, $a \to b = a \to \neg\neg b = \neg(a \wedge \neg b) \in A_{\neg\neg} = B$.
It remains to check that joins in B are given by $b_1 \vee_B b_2 = \neg(\neg b_1 \wedge \neg b_2)$. Suppose that $b_1, b_2 \in B$, i.e. $b_1 = \neg\neg b_1$ and $\neg\neg b_2 = b_2$. Clearly, $\neg(\neg b_1 \wedge \neg b_2) \in A_{\neg\neg}$ as its is negated. For $i = 1, 2$ we have that $\neg b_1 \wedge \neg b_2 \leq \neg b_i$ and, therefore, $b_i = \neg\neg b_i \leq \neg(\neg b_1 \wedge \neg b_2) = b_1 \vee_B b_2$, i.e. $b_1 \vee b_2$ is an upper bound of b_1 and b_2. We now show that it is the least upper bound in B. Suppose $b_1, b_2 \leq c \in B$. Then $\neg c \leq \neg b_i$ for $i = 1, 2$ and, accordingly, $\neg c \leq \neg b_1 \wedge \neg b_2$ from which it follows that $b_1 \vee_B b_2 = \neg(\neg b_1 \wedge \neg b_2) \leq \neg\neg c = c$ as desired. $\square$

It is easy to see (by inspection of the proof we just gave) that for a cHa A the subposet $B = A_{\neg\neg}$ of double negation closed elements is a cBa. Infinite meets in B are inherited from A and $\bigvee^B_{i \in I} b_i = \neg \bigwedge_{i \in I} \neg b_i$. Notice that for a topological space X the Boolean algebra $\mathcal{O}(X)_{\neg\neg}$ is the subposet of $\mathcal{O}(X)$ on the so–called *regular open sets*, i.e. those open sets U which coincide with the interior of the closure of U ($\mathrm{int}(\mathrm{cl}(U)) = U$).
After these simple algebraic considerations we apply a construction analogous to $(-)_{\neg\neg}$ to the syntax of predicate logic.

Definition 5.2 *The* Gödel–Gentzen double negation translation $(-)^G$ *of formulas of predicate logic is defined inductively as follows*

$$\begin{array}{lcll}
\bot^G & \equiv & \bot & \\
P^G & \equiv & \neg\neg P & P \text{ atomic but different from } \bot \\
(A \wedge B)^G & \equiv & A^G \wedge B^G & \\
(A \to B)^G & \equiv & A^G \to B^G & \\
(A \vee B)^G & \equiv & \neg(\neg A^G \wedge \neg B^G) & \\
(\forall x.A)^G & \equiv & \forall x.\, A^G & \\
(\exists x.A)^G & \equiv & \neg\forall x.\neg A^G & .
\end{array}$$

The double negation translation can be described informally as follows: atomic formulas are negated twice, $\wedge$, $\to$, $\bot$ and $\forall$ are left unchanged but $\vee$ and $\exists$ are replaced by their "de Morgan dual". Obviously, the double negation translation is a syntactic analogue of Theorem 5.1 in the sense that classical connectives and quantifiers look as they have to be in $\mathcal{A}_{\neg\neg}$. Often (e.g. in constructive arithmetic) the basic predicates are decidable, i.e. satisfy PEM, and, therefore, one may define $P^G \equiv P$ for atomic formulas. However, in the general case atomic formulas different[9] from $\bot$ have to be doubly negated. We leave it as an exercise to show that the equivalence of A^G and $(A^G)^G$ can be proved constructively for all formulas A. Notice also that in general $\neg\neg A$ and A^G are not provably equivalent in constructive logic (exercise!).
The key result of this section will be the following

Theorem 5.2 *A sequent $A_1, \ldots, A_n \vdash B$ can be derived in classical predicate logic if and only if its G–translation $A_1^G, \ldots, A_n^G \vdash B^G$ can be derived in constructive predicate logic.*

Notice, however, that in general A and A^G are not provably equivalent in constructive logic (e.g. if A is an appropriate instance of PEM) though they are provably equivalent in classical logic.
But before proving Theorem 5.2 we need the following notion.

Definition 5.3 *The class of* Harrop formulas *is defined inductively by the following clauses*

- *$\bot$ is a Harrop formula*
- *$A \wedge B$ is a Harrop formula whenever A and B are Harrop formulas*

[9] We could have defined $\bot^G$ as $\neg\neg\bot$ but as $\neg\neg\bot$ is provably equivalent to $\bot$ this makes no difference!

- *$A \to B$ is a Harrop formula whenever B is a Harrop formula*
- *$\forall x.A$ is a Harrop formula whenever A is a Harrop formula.*

Notice that it follows from the first and third clause of the previous definition that *every* negated formula is a Harrop formula. Obviously, A^G is always a Harrop formula.
We next show that Harrop formulas are provably double negation closed.

Lemma 5.1 *For every Harrop formula A one can derive $\neg\neg A \vdash A$ in the calculus of Natural Deduction. Thus, for Harrop formulas A the equivalence $A \leftrightarrow \neg\neg A$ can be proved constructively.*

Proof: As for every formula A one can prove constructively that $A \vdash \neg\neg A$. It suffices to verify that $\neg\neg A \vdash A$ can be proved constructively for all Harrop formulas A.
For the first 3 clauses of Definition 5.3 the argument is analogous to the proof of Theorem 5.1.
It remains to show that $\forall x.A$ is double negation closed if A is. Suppose that $\neg\neg A \vdash A$. We have $\forall x.A \vdash A$ and, therefore, also $\neg\neg\forall x.A \vdash \neg\neg A$. As A was assumed as double negation closed we also have $\neg\neg\forall x.A \vdash A$ from which it follows by $(\forall I)$ that $\neg\neg\forall x.A \vdash \forall x.A$ as desired. □

Now we are ready to give the

Proof (of Theorem 5.2) :
First notice that the formulas A^G and A are provably equivalent in classical predicate logic as there one can derive the deMorgan laws

$$\neg(A \wedge B) \dashv\vdash \neg A \vee \neg B \qquad \text{and} \qquad \neg\forall x.A(x) \dashv\vdash \exists x.\neg A(x) \quad .$$

Thus, if $A_1^G, \ldots, A_n^G \vdash B^G$ is derivable in constructive logic then $A_1, \ldots, A_n \vdash B$ can be derived in classical logic.
Thus, it remains to prove that $A_1^G, \ldots, A_n^G \vdash B^G$ is derivable in constructive logic whenever $A_1, \ldots, A_n \vdash B$ can be derived classically. This we show by induction on derivations in classical predicate logic. The cases of structural rules as well as introduction and elimination rules for $\wedge$, $\to$ $\bot$ and $\forall$ are trivial (as in these cases the double negation rule does not change the connectives). Accordingly, we just discuss the few remaining cases.
$(\vee I_1)$ Suppose that $\Gamma \vdash A$ can be derived classically and $\Gamma^G \vdash A^G$ can be derived intuitionistically. The following derivation shows that then the sequent $\Gamma^G \vdash$

$(A \vee B)^G$ can be derived constructively, too,

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma^G, \neg A^G \wedge \neg B^G \vdash \neg A^G \wedge \neg B^G}\,(\mathrm{ax})}{\Gamma^G, \neg A^G \wedge \neg B^G \vdash \neg A^G}\,(\wedge E_1) \qquad \dfrac{\Gamma^G \vdash A^G}{\Gamma^G, \neg A^G \wedge \neg B^G \vdash A^G}\,(\mathrm{w})}{\Gamma^G, \neg A^G \wedge \neg B^G \vdash \bot}\,(\to E)}{\Gamma^G \vdash \neg(\neg A^G \wedge \neg B^G)}\,(\to I)$$

where the open assumption $\Gamma^G \vdash A^G$ is derivable by assumption.
The case of $(\vee I_2)$ is analogous.
$(\vee E)$ Suppose that the sequents

$$\Gamma \vdash A_1 \vee A_2 \qquad \Gamma, A_1 \vdash C \qquad \Gamma, A_2 \vdash C$$

are classically derivable. Then by induction hypothesis their double negation translations

$$\Gamma^G \vdash (A_1 \vee A_2)^G \qquad \Gamma^G, A_1^G \vdash C^G \qquad \Gamma^G, A_2^G \vdash C^G$$

can be derived constructively, too. Thus, we can derive constructively the sequents $\Gamma^G, \neg C^G \vdash \neg A_i^G$ for $i = 1, 2$ from which it follows that $\Gamma^G, \neg C^G \vdash \neg A_1^G \wedge \neg A_2^G$ can be derived constructively. But by assumption we have $\Gamma^G \vdash \neg(\neg A_1^G \wedge \neg A_2^G)$ (as $(A_1 \vee A_2)^G \equiv \neg(\neg A_1^G \wedge \neg A_2^G)$) and, therefore, by $(\to E)$ we get $\Gamma^G, \neg C^G \vdash \bot$. Thus, by $(\to E)$ we get $\Gamma^G \vdash \neg\neg C^G$ and, therefore, by Lemma 5.1 that $\Gamma^G \vdash C^G$.
The introduction and elimination rules for $\exists$ are analogous to those for $\vee$.
Thus, we finally have to consider the rule *reductio ad absurdum.* Suppose that $\Gamma \vdash \neg\neg A$ is classically derivable. Then by induction hypothesis the sequent $\Gamma^G \vdash (\neg\neg A)^G$ can be derived constructively. Unfolding the definition of $(-)^G$ we observe $(\neg\neg A)^G \equiv \neg\neg A^G$ and, therefore, we get that $\Gamma \vdash \neg\neg A^G$ is constructively derivable. As by Lemma 5.1 A^G is provably $\neg\neg$–closed it follows that $\Gamma^G \vdash A^G$ is constructively derivable as well. $\square$

The above theorem about double negation translation nicely extends to theories $\mathcal{T}$ (i.e. sets of sentences of predicate logic) under a certain proviso which is often satisfied, e.g. for the important case of constructive arithmetic which will be considered in the next section.

Corollary 5.1 *Let $\mathcal{T}$ be a theory, i.e. a set of sentences (closed formulas) of predicate logic, such that A^G is constructively derivable for every $A \in \mathcal{T}$.*
Then A is classically derivable from $\mathcal{T}$ iff A^G is constructively derivable from $\mathcal{T}$.

Proof: Exercise! $\square$

Summarizing we can say that though classical logic proves more formulas than constructive logic it is the case that classical logic can be embedded into the

$\wedge\!\to\!\bot\forall$–fragment of constructive logic in a full and faithful way. Not every formula of constructive predicate logic is provably equivalent to one in the $\wedge\!\to\!\bot\forall$–fragment as not every formula of constructive predicate logic is $\neg\neg$–closed. Thus, constructive logic is rather an extension of classical logic than a restriction as it contains the latter properly!
Another conclusion we may draw from Theorem 5.2 and Corollary 5.1 is that constructive logic cannot be considered as "saver" than classical logic as they are equiconsistent, i.e. one is consistent iff the other is. This fact has to be seen in sharp contrast with the point of view widely[10] adopted in the 20ies and 30ies of the 20th century where constructive logic was considered as a save(r) alternative to classical logic. The reason was that L. E. J. Brouwer forcefully pushed constructive or — as he called it — "intuitionistic" logic as a save alternative to classical logic and, in particular, modern set theory. He certainly was right w.r.t. the inherently inconstructive nature of axiomatic set theory but not w.r.t. classical logic. Although Gödel proved Theorem 5.2 already in the early 30ies it took some time until people realized that it implies the equiconsistency of constructive and classical logic.

6 Constructive Arithmetic and Analysis

After having explained the basics of constructive logic and its relation to classical logic it definitely is time to consider some (formalization of) elementary constructive mathematics. Certainly natural numbers are the basic phenomenon where mathematics starts and, therefore, we first consider *Heyting Arithmetic* (**HA**) called after A. Heyting, a disciple of Brouwer, who gave the first axiomatization of constructive logic and arithmetic.
In principle one could base an axiomatisation of constructive arithmetic on the basic operations 0, succ (for the successor operation on numbers), addition and multiplication and equality as the only predicate. However, this has the disadvantage that developing elementary number theory, e.g. defining division, the divisibility relation or the predicate "is a prime number", would involve a lot of coding. Therefore, we prefer to take all *primitive recursive* algorithms as basic constants and postulate their defining equations as axioms.[11]

[10]Though most people considered constructive logic as "saver" than classical logic the great majority did not adopt it as they thought that it is much too weak to develop a reasonable part of mathematics on its grounds. After some decades of investigations into constructive logic and mathematics since that time, however, it has turned out that most parts of (applied) analysis *can* indeed be developed in a purely constructive manner (see e.g. [BiBr]).

[11]In principle it were desirable to add function symbols for all algorithms (together with their defining equations) to the language. But as the class of algorithms which terminate for all arguments cannot be effectively enumerated this would render our syntax undecidable and this is unacceptable for a formal system (as its underlying language should be decidable and its consequences should be effectively enumerable!).

For sake of completeness we recall the definition of the basic concept of *primitive recursive function.*

Definition 6.1 *The set* PRIM ($\subseteq \bigcup_{n\in\mathbb{N}} \mathbb{N}^n {\to} \mathbb{N}$) *of* primitive recursive functions *is defined inductively as follows.*

(1) 0 *(considered as a function from* $\mathbb{N}^0{\to}\mathbb{N}$*) is in* PRIM.

(2) *The successor function* $\mathsf{succ} : \mathbb{N} \to \mathbb{N} : n \mapsto n+1$ *is in* PRIM.

(3) *For every* $n > 0$ *and* i *with* $1 \leq i \leq n$ *the projection function*

$$\mathrm{pr}^n_i : \mathbb{N}^n \to \mathbb{N} : (x_1, \ldots, x_n) \mapsto x_i$$

is in PRIM.

(4) *If* $g : \mathbb{N}^n \to \mathbb{N}$ *and* $h_i : \mathbb{N}^m \to \mathbb{N}$ *for* $i = 1, \ldots, n$ *then the function*

$$f : \mathbb{N}^m \to \mathbb{N} : \vec{x} \mapsto g(h_1(\vec{x}), \ldots, h_n(\vec{x}))$$

is in PRIM *whenever* g *and the* h_i *are in* PRIM.

(5) *If* $g : \mathbb{N}^n \to \mathbb{N}$ *and* $h : \mathbb{N}^{n+2} \to \mathbb{N}$ *are in* PRIM *then the function* $f : \mathbb{N}^{n+1} \to \mathbb{N}$ *with*

$$f(\vec{x}, 0) = g(\vec{x}) \quad \textit{and} \quad f(\vec{x}, n+1) = h(\vec{x}, n, f(\vec{x}, n))$$

is in PRIM.

The scheme of defining f from g and h as in (5) is commonly called the schema of *definition by primitive recursion.* Most number theoretic functions (of elementary number theory and far beyond) are primitive recursive. It is obvious that addition, multiplication and exponentiation are primitive recursive. Just to give a simple example (which is needed subsequently) the *predecessor function* pred with $\mathrm{pred}(0) = 0$ and $\mathrm{pred}(n+1) = n$ is primitive recursive as its defining equations can be rewritten as

$$\mathrm{pred}(0) = 0 \qquad \mathrm{pred}(n+1) = \mathrm{pr}^2_1(n, \mathrm{pred}(n)) \quad .$$

Notice further that basic predicates on numbers such as equality, $\leq$ etc. have primitive recursive characteristic functions.[12]
Now we are ready to define Heyting arithmetic.

[12]We often claim that a function or predicate is primitive recursive without proving it in detail. This is common practice in computability theory. To get confidence have a look at any book on recursion theory or the first part of my course *LogikII: Berechenbarkeit und Unvollständigkeit formaler Systeme.* For people used to imperative programming in FORTRAN, PASCAL, C, Java or whatever language of this kind it may be informative to know that a function is primitive recursive iff it can be implemented using only **for**–loops and no **while**–loops (and, of course, no arbitrary jumps).

Definition 6.2 *The language of* Heyting arithmetic (**HA**) *is given by function symbols for every (definition of a) primitive recursive function and binary equality predicate* $=$.
Besides the rules of constructive predicate calculus the usual equality axioms

$$t = t \qquad A(t) \wedge t = s \to A(s)$$

(called reflexivity *and* replacement*, respectively) the axioms of* **HA** *are*

(1) *the defining equations for the primitive recursive functions*

(2) $\neg\, 0 = \mathsf{succ}(x)$

(3) *the induction axiom* $A(0) \wedge (\forall x.(A(x) \to A(\mathsf{succ}(x)))) \to \forall x.A(x)$ *for every predicate* $A(x)$ definable in the language of **HA**.

The system **PRA** *of* Primitive Recursive Arithmetic *is obtained from* **HA** *by restricting induction to quantifier–free predicates* $A(x)$.

Note that classical arithmetic, often called Peano arithmetic (**PA**), is obtained from **HA** by adding the axiom schema of *reductio ad absurdum*, i.e. strengthening constructive logic to classical logic.
It turns out that most theorems of elementary number theory can be proved already in **HA**. Decidability of equality, i.e. $\forall x, y.\, x = y \vee \neg x = y$ holds in **PA** for purely logical reasons. It holds in **HA** as well but there it needs a mathematical argument.

Lemma 6.1 *In* **HA** *one can prove decidability of equality. Therefore, equality is* $\neg\neg$*–closed, i.e.* $\forall x, y.\ \neg\neg\, x{=}y \to x{=}y$.

Proof: Instead of giving a formal derivation in **HA** we give a detailed, but informal argument from which a derivation in **HA** could[13] be constructed.
We prove $\forall x, y.\ x{=}y \vee \neg\, x{=}y$ by induction on x.
We leave it as an exercise to prove $\forall y.\ 0{=}y \vee \neg\, 0{=}y$ by by induction on y (in the induction step the hypothesis isn't used at all!).
Assume as induction hypothesis that $\forall y.\, x{=}y \vee \neg\, x{=}y$ holds for an arbitrary, but fixed x. We show by induction on y that $\forall y.\, \mathsf{succ}(x){=}y \vee \neg\, \mathsf{succ}(x){=}y$. For $y{=}0$ we clearly have $\neg\, \mathsf{succ}(x){=}y$ as $\neg\, 0{=}\mathsf{succ}(x)$ is an axiom of **HA**. Assume as induction hypothesis that $\mathsf{succ}(x){=}y \vee \neg\mathsf{succ}(x){=}y$. We have to show that $\mathsf{succ}(x){=}\mathsf{succ}(y) \vee \neg\, \mathsf{succ}(x){=}\mathsf{succ}(y)$. Due to the first induction hypothesis we know that $x{=}y \vee \neg\, x{=}y$. If $x{=}y$ then $\mathsf{succ}(x) = \mathsf{succ}(y)$. Suppose $\neg\, x{=}y$. We show that under this assumption $\neg\, \mathsf{succ}(x){=}\mathsf{succ}(y)$: if $\mathsf{succ}(x){=}\mathsf{succ}(y)$ then

$$x{=}\mathrm{pred}(\mathsf{succ}(x)){=}\mathrm{pred}(\mathsf{succ}(y)){=}y$$

[13]It would be overly laborious to construct a formal derivation without machine assistance. See my course on *Constructive Type Theory* for an introduction to such systems.

and, therefore, it follows that $\bot$ as $\neg\, x{=}y$. Thus, we have $\forall y.\, \mathsf{succ}(x){=}y \vee \neg\, \mathsf{succ}(x){=}y$ as desired which also completes the outer induction proof. $\square$

Our next goal is to identify a class of formulas for which **PA** is *conservative* over **HA** which means that every formula in this class which is provable in **PA** can already be proved in **HA**.

Lemma 6.2 *If A is derivable in* **PA** *then its double negation translation A^G is derivable in* **HA**.

Proof: By Corollary 5.1 it suffices to show that double negation translations of **HA** axioms can be derived in **HA**, too. This can be seen as follows.
By the previous Lemma 6.1 an equation $t{=}s$ is provably equivalent to $(t{=}s)^G \equiv \neg\neg\, t{=}s$. The axiom $\neg\, 0{=}\mathsf{succ}(x)$ is double negation closed as it is itself a negation. The double negation of an induction axiom

$$A(0) \wedge (\forall x.(A(x) \to A(\mathsf{succ}(x)))) \to \forall x.A(x)$$

is

$$A^G(0) \wedge (\forall x.(A^G(x) \to A^G(\mathsf{succ}(x)))) \to \forall x.A^G(x)$$

which itself is an instance of the induction schema and, therefore, derivable in **HA**. $\square$

Alas, the previous lemma does not help us to show that if for a quantifier–free formula A the formula $\exists x.A(x)$ can be derived in **PA** then the same formula can be derived in **HA**. The reason is that from $(\exists x.A(x))^G \equiv \neg\forall x.\neg A(x)$ we cannot derive constructively that $\exists x.A(x)$. Nevertheless, it is the case that **PA** is conservative over **HA** w.r.t. existential quantifications of quantifier–free formulas. But for this purpose we need the so–called *Friedman translation* defined next.

Definition 6.3 *Let R be a fresh propositional constant (predicate of arity 0). Then the* Friedman *or R–*translation *$(-)^R$ is defined inductively by the following clauses*

$\bot^R$	$\equiv$	R	
P^R	$\equiv$	$\neg_R\neg_R P$	P atomic but different from $\bot$
$(A \wedge B)^R$	$\equiv$	$A^R \wedge B^R$	
$(A {\to} B)^R$	$\equiv$	$A^R \to B^R$	
$(A \vee B)^R$	$\equiv$	$\neg_R(\neg_R A^R \wedge \neg_R B^R)$	
$(\forall x.A)^R$	$\equiv$	$\forall x.\, A^R$	
$(\exists x.A)^R$	$\equiv$	$\neg_R\forall x.\neg_R A^R$	

where $\neg_R A$ stands for $A \to R$. $\Diamond$

Now one can show analogously to the proof of Theorem 5.2 that

Lemma 6.3 *The sequent $\Gamma^R \vdash A^R$ is derivable in constructive predicate logic without $(\bot)$ whenever $\Gamma \vdash A$ is derivable in classical predicate logic.*
Therefore, A^R is derivable in **PA** *whenever A is derivable in* **HA** *as the R–translations of* **HA** *axioms can be derived in* **HA**.

Proof: Exercise! □

Now using Lemma 6.3 we can easily prove the conservativity of **PA** over **HA** w.r.t. existential quantifications of quantifier–free formulas.

Theorem 6.1 *Let A be a quantifier–free formula. If $\exists\vec{x}.A$ can be derived in* **PA** *then it can be derived in* **HA**, *too.*

Proof: We just consider the case where $\vec{x}$ consists of one single variable x and leave the (easy) generalisation as an exercise.
Suppose that **PA** proves $\exists x.A$. Then by Lemma 6.3 the formula $(\exists x.A)^R$ can be derived in **HA**. As $(\exists x.A)^R \equiv (\forall x.\, \neg_R\neg_R\neg_R A(x)) \to R$ is **HA**–provably equivalent to $(\forall x.\, A(x) \to R) \to R$ we conclude that the latter formula is derivable in **HA**. As this proof does not use any assumptions about R it follows that **HA** proves[14] also $(\forall x.\, A(x) \to \exists x.A(x)) \to \exists x.A(x)$. As the premiss of this implication can be derived already in constructive predicate logic it follows that **HA** derives $\exists x.A(x)$ as desired. □

The idea of replacing R in $(\exists x.A(x))^R$ by $\exists x.A(x)$ itself is often called the *Friedman trick.*
As Theorem 6.1 applies to open formulas of the form $\exists\vec{y}.\, A(\vec{x},\vec{y})$ with A quantifier–free it follows that $\forall\vec{x}.\exists\vec{y}.\, A(\vec{x},\vec{y})$ can be derived already in **HA** whenever it can be derived in **PA** (Exercise!). As most theorems of elementary number theory are of this form constructive and classical arithmetic are not too different.[15] Notice, however, that it is not clear whether the proof of Fermat's Last Theorem by A. Wiles (and R. Taylor) can be formalized in **PA** and, therefore, also in **HA**.
But in general **PA** is *not conservative* over **HA** for formulas of a logical structure more complicated than $\forall\exists$–statements. For example the *least number principle*

$$\text{LNP} \qquad (\exists x.\, A(x)) \to \exists x.\, (A(x) \wedge \forall y.\, A(y) \to x \le y)$$

familiar from classical mathematics does not hold anymore in the constructive context as we show in the next Lemma.

[14]Notice, however, that in the proof of $(\forall x.\, A(x) \to R) \to R$ we have to rename all bound variables in such a way that they are distinct from $\mathrm{FV}(A) \setminus \{x\}$ in order to avoid violations of the variable conditions of $(\forall I)$ and $(\exists E)$ when replacing R by $\exists x.A(x)$.

[15]As opposed to elementary analysis (discussed a bit later on) where already the basic predicates $=$ and $<$ on real numbers are not decidable and, therefore, in general for real numbers x and y the trichotomy law $x < y \vee x = y \vee y < x$ is not constructively valid anymore!

Lemma 6.4 *The scheme* LNP *entails* PEM. *As* PEM *cannot be derived in* **HA** *it follows that* LNP *cannot be derived in* **HA**.

Proof: Let B be a closed formula of **HA**. Consider the instantiation of LNP by the predicate

$$A(x) \;\equiv\; x{=}1 \vee (x{=}0 \wedge B)$$

which entails

$$(\dagger) \qquad \exists x.\, (A(x) \wedge \forall y.\, A(y) \to x \leq y)$$

by LNP as $A(1)$ obviously holds. As $A(x)$ implies $x{=}0 \vee x{=}1$ we can derive from ($\dagger$) that $B \vee \neg B$ as if $x{=}0$ then B and if $x{=}1$ then $\forall y.\, (y{=}1 \vee (y = 0 \wedge B) \to 1 \leq y)$ which entails $\neg B$ when instantiating y by 0. $\square$

Notice that classically LNP is equivalent to the induction principle (exercise!) which, however, constructively is not the case as induction holds in **HA** whereas LNP does not.

Next we consider the formalism **EL** of Elementary Analysis. Among other aspects it differs from **HA** in the respect that it is 2–sorted.

Definition 6.4 *The system* **EL** *of* Elementary Analysis *is the extension of* **HA** *defined as follows.*
Besides the type of natural numbers ranged over by variables $x, y, z \ldots$ there is also a type of unary functions on natural numbers ranged over by variables $\alpha, \beta, \gamma \ldots$. The type of natural numbers is also called type 0 *and the type of unary functions on numbers is also called* type 1.
We have the following term formation rules in addition to those of **HA**.

(1) *Every variable α of type 1 is a term of type 1.*

(2) *If φ is a term of type 1 and t is a term of type 0 then $\mathrm{Ap}(\varphi, t)$ is a term of type 0 for which we usually write $\varphi(t)$.*

(3) *If t is a term of type 0 and x is a variable of type 0 then $\lambda x.t$ is a term of type 1.*

(4) *If t is a term of type 0 and φ is a term of type 1 then $\mathsf{R}(t, \varphi)$ is a term of type 1.*

Let $\langle -, - \rangle$ be some primitive recursive coding of pairs of natural numbers with primitive recursive projection functions pr_0 and pr_1 satisfying $\mathsf{pr}_i(\langle n_0, n_1 \rangle) = n_i$ for $i = 0, 1$. We write $\varphi(t_1, t_2)$ as a shorthand for $\varphi(\langle t_1, t_2 \rangle)$.
Besides the axioms of **HA** *with induction principle*

$$A(0) \wedge (\forall x.(A(x) \to A(\mathsf{succ}(x)))) \to \forall x.A(x)$$

extended to all predicates $A(x)$ definable in the language of **EL** *we postulate in addition the following axioms*

(β) $\qquad (\lambda x.t)(s) = t[s/x]$

(R0) $\qquad \mathsf{R}(t,\varphi)(0) = t$

(R1) $\qquad \mathsf{R}(t,\varphi)(\mathsf{succ}(t')) = \varphi(t', \mathsf{R}(t,\varphi)(t'))$

(AC–QF) $\qquad \forall x.\exists y.A(x,y) \to \exists \alpha.\forall x.A(x,\alpha(x))$

where $A(x,y)$ is quantifier-free

which allow one to define functions explicitely via λ*–abstraction and by primitive recursion via* R *relative to variables of function type* 1. $\Diamond$

If unary functions on numbers are all assumed to be recursive then this extension can be avoided as one may refer to number–theoretic functions via their Gödel numbers (cf. the next section on Basic Recursion Theory in **HA**). This situation is typical for *Constructive Recursive Arithmetic* (CRA) which postulates *Church's Thesis* in the (strong) from that all number–theoretic functions are recursive. However, in *Brouwerian Intuitionistic Mathematics* (BIM) – which we study later on – the notion of *choice sequence* is essential where a choice sequence α is thought of as an *arbitrary* sequence

$$\alpha(0), \alpha(1), \ldots, \alpha(n), \ldots$$

of natural numbers typically not given by a law, i.e. an algorithm for computing the items of this sequence. Brouwer's motivation for considering choice sequences was (1) to make the intuitionistic continuum comprehensive enough by avoiding the restriction to computable real numbers and (2) to retain principles like "every continuous real–valued function on $[0,1]$ is bounded" which fails when restricting to computable reals.

7 Constructive Real Numbers

The theory of rational numbers can be developed constructively quite as in the classical case because one can effectively code rational numbers by natural numbers in a way that the basic functions and predicates on the rational numbers are primitive recursive. In particular, the equality predicate $=$ and the predicate $<$ on $\mathbb{Q}$ are decidable and the *law of trichotomy*

$$\forall p, q \in \mathbb{Q}.\; p < q \vee p = q \vee q < p$$

is valid also constructively.

Classically there are two ways of introducing the real numbers: either as equivalence classes of Cauchy sequences of rational numbers or as Dedekind cuts. Constructively, these two constructions give different results unless[16] one postulates the *axiom of choice for numbers* (also known as *number choice*)

$$\mathrm{AC}_{00} \qquad \forall n.\exists m.A(n,m) \to \exists \alpha.\forall n.A(n,\alpha(n))$$

where n and m range over natural numbers. As AC_{00} is most natural from a constructive point of view we just deal with Cauchy reals. For a detailed discussion of the relation between Cauchy and Dedekind reals in absence of number choice see for example vol.1 of [TvD].

Definition 7.1 *A* fundamental sequence *is a sequence $r = (r_n)_{n\in\mathbb{N}}$ of rational numbers for which there exists a sequence β of natural numbers, called its* (Cauchy-)modulus*, such that*

$$|r_{\beta(n)+m} - r_{\beta(n)+m'}| < 2^{-n}$$

for all $n, m, m' \in \mathbb{N}$. $\Diamond$

Obviously, under assumption of number choice a sequence r of rational numbers is a fundamental sequence if and only if

$$\forall k.\exists n.\forall m, m'.\, |r_{n+m} - r_{n+m'}| < 2^{-k}$$

as by AC_{00} this condition implies the existence of a Cauchy modulus. Thus, under assumption of number choice a fundamental sequence is nothing else but a Cauchy sequence of rational numbers in the usual sense.

From now on we postulate AC_{00} *for the rest of this section.*

Definition 7.2 *Let r and s be fundamental sequences.*
We say that r and *s* are equal *(notation $r \approx s$) iff*

$$\forall k.\exists n.\forall m.\, |r_{n+m} - s_{n+m}| < 2^{-k} \quad .$$

We say that r is strictly below *s (notation $r < s$) iff*

$$\exists k, n.\forall m.\; s_{n+m} - r_{n+m} > 2^{-k} \quad .$$

We write $r \le s$ as an abbreviation for $\neg(s < r)$ for which we say that "r is not strictly above s".
We say that r and s are apart *(notation $r\#s$) iff $r < s \vee s < r$.* $\Diamond$

[16]In general Dedekind reals are more general than Cauchy reals. For example in sheaf toposes over a connected topological space X the Cauchy and the Dedekind reals look as follows. The Cauchy reals R_c are given by the constant functor from $\mathcal{O}(X)^{\mathsf{op}}$ to Set sending all open subsets U of X to the set $\mathbb{R}$. The Dedekind reals are given by the sheaf $R_d : \mathcal{O}(X)^{\mathsf{op}} \to \mathsf{Set}$ sending $U \in \mathcal{O}(X)$ to the set of continuous functions from U to $\mathbb{R}$ and whose morphism part is given by restriction.

The following disjunction–free characterisation of apartness will turn out as useful.

Lemma 7.1 *Fundamental sequences r and s are apart iff*

$$\exists k, n. \forall m. \, |r_{n+m} - s_{n+m}| > 2^{-k} \quad .$$

Proof: Obviously, the condition is necessary.
For the reverse direction assume that there exist natural numbers k and n such that $|r_{n+m} - s_{n+m}| > 2^{-k}$ for all $m \in \mathbb{N}$. As both r and s are fundamental sequences there exists a natural number ℓ with $\ell \geq n$ such that $|r_\ell - r_{\ell+i}| < 2^{-k-1}$ and $|s_\ell - s_{\ell+i}| < 2^{-k-1}$ for all i.
Now by the law of trichotomy for rational numbers it holds that either $r_\ell < s_\ell$ or $s_\ell < r_\ell$ because $r_\ell = s_\ell$ is impossible as $|r_\ell - s_\ell| > 2^{-k}$.
Assume that $r_\ell < s_\ell$. We show that for all i it holds that $r_{\ell+i} < s_{\ell+i}$ and hence $s_{\ell+i} - r_{\ell+i} > 2^{-k-1}$ from which it follows that $r < s$. Suppose $\neg(r_{\ell+i} < s_{\ell+i})$, i.e. $s_{\ell+i} \leq r_{\ell+i}$. Then $r_{\ell+i} - s_{\ell+i} > 2^{-k}$ and, therefore,

$$\begin{aligned} r_{\ell+i} &> s_{\ell+i} + 2^{-k} = s_{\ell+i} + 2 \cdot 2^{-k-1} > s_{\ell+i} + |s_\ell - s_{\ell+i}| + |r_{\ell+i} - r_\ell| \\ &\geq s_{\ell+i} + s_\ell - s_{\ell+i} + r_{\ell+i} - r_\ell = r_{\ell+i} + s_\ell - r_\ell \end{aligned}$$

which entails that $0 > s_\ell - r_\ell$ and hence $r_\ell > s_\ell$ in contradiction to our assumption $r_\ell < s_\ell$.
If $s_\ell < r_\ell$ it follows by an analogous argument that $s < r$. $\square$

We suggest it as a straightforward exercise to show that $\approx$ is a congruence relation w.r.t. $<$ (which means that $(r \approx r' \wedge s \approx s') \wedge r < s \to r' < s'$) and hence also w.r.t. the relation $\leq$ and $\#$ as they are defined in terms of $<$. Thus, we may factorize the collection of fundamental sequence by the equivalence relation $\approx$ and define the predicates $<$ and $\#$ on the quotient $\mathbb{R}$ as follows: $x < y$ iff $r < s$ for some fundamental sequences $r \in x$ and $y \in y$ and similarly for $\#$. Notice that $x = y$ iff $r \approx s$ for some fundamental sequences $r \in x$ and $y \in y$.
An alternative to factorization of fundamental sequences by the relation $\approx$ one could take the approach taken by Bishop in [BiBr]. He considers collections of objects as *presets* and defines a *set* as a preset together with a symmetric and transitive relation on this preset. From this point of view the set of reals is given by the preset of functions from $\mathbb{N}$ to $\mathbb{Q}$ and r and s are considered as equal iff both r and s are fundamental sequences and $r \approx s$ in the sense of Definition 7.2. Of course, when defining functions and predicates on such a set one has to show that they respect the equality relation which is part of the set.[17]

[17]Such an approach is also taken (implicitly) in programming when implementing an abstract data type by a concrete data type. One has to specify which concrete objects represent abstract objects and when concrete objects represent the same abstract object.

Next we study a few properties of the apartness relation which are often considered an axiomatization of a general notion of apartness (put to use when studying e.g. constructive algebra).

Lemma 7.2 *For all real numbers x and y it holds that*

(AP1)	$\neg x\#y \leftrightarrow x = y$
(AP2)	$x\#y \rightarrow y\#x$
(AP3)	$x\#y \rightarrow \forall z.\, x\#z \vee y\#z.$

Proof: We first prove (AP1). If $x = y$ then $\neg x\#y$ as obviously $\neg x\#x$ and $=$ is a congruence w.r.t. the apartness relation.
For the reverse direction assume that $\neg x\#y$. Pick some fundamental sequences r and s out of x and y, respectively. Then we have

$$(1) \qquad \forall k, n. \neg\forall m.\, |r_{n+m} - s_{n+m}| > 2^{-k}$$

due to the assumption $\neg x\#y$. Let α and β be Cauchy moduli for the fundamental sequences r and s, respectively, i.e.

$$(2) \qquad \forall k, n, m.\, |r_{\alpha(k)+n} - r_{\alpha(k)+m}| < 2^{-k}$$
$$(3) \qquad \forall k, n, m.\, |s_{\beta(k)+n} - s_{\beta(k)+m}| < 2^{-k}.$$

Let γ be the function from $\mathbb{N}$ to $\mathbb{N}$ defined as $\max(\alpha(\mathrm{k}+2), \beta(\mathrm{k}+2))$. We show that $\forall k, n.\, |r_{\gamma(k)+n} - s_{\gamma(k)+n}| < 2^{-k}$ from which it follows that $r \approx s$ and hence $x = y$.
First notice that

$$(4) \qquad \forall k. \neg\forall m.\, |r_{\gamma(k)+m} - s_{\gamma(k)+m}| > 2^{-k-2}$$

due to (1). Let k be an arbitrary natural number. Suppose that $|r_{\gamma(k)+n} - s_{\gamma(k)+n}| \geq 2^{-k}$ for some n. Then for arbitrary m we have

$$\begin{aligned} &|r_{\gamma(k)+m} - s_{\gamma(k)+m}| \\ &\geq |r_{\gamma(k)+n} - s_{\gamma(k)+n}| - |r_{\gamma(k)+m} - r_{\gamma(k)+n}| - |s_{\gamma(k)+m} - s_{\gamma(k)+n}| \\ &\geq 2^{-k} - 2 \cdot 2^{-k} = 2^{-k} - 2^{-k-1} = 2^{-k-1} \end{aligned}$$

in contradiction to (4). Thus, we conclude $\forall k, n.\, |r_{\gamma(k)+n} - s_{\gamma(k)+n}| < 2^{-k}$.
(AP2) is trivial and the proof of (AP3) is left as an exercise. □

Thus, apartness appears as more basic than equality on reals as $\neg\, x{=}y$ is equivalent to $\neg\neg\, x\#y$ but in general[18] not to $x\#y$.

[18]One can show (exercise!) that $\forall x, y.\, x\#y \leftrightarrow \neg x{=}y$ is equivalent to the so–called *Markov's Principle* claiming that for decidable predicates $A(x)$ it holds that $\neg\neg\,\exists x.A(x) \rightarrow \exists x.A(x)$.

Moreover, another consequence of (AP1) is that

$$x{=}y \leftrightarrow \neg(x{<}y \vee y{<}x) \leftrightarrow x{\leq}y \wedge y{\leq}x$$

which gives some justification for the definition of $\leq$.
A further useful consequence of Lemma 7.2 is the following.

Corollary 7.1 *Equality on reals is $\neg\neg$–closed, i.e. $\neg\neg x = y \to x = y$ for all real numbers x and y.*

Proof: Immediate from (AP1) as it says that $x{=}y$ is equivalent to the negated proposition $\neg x \# y$. □

Next we show that decidability of equality on $\mathbb{R}$ entails $\forall n.A(n) \vee \neg\forall n.A(n)$ for all decidable predicates A on $\mathbb{N}$.

Lemma 7.3 *Let A be a decidable predicate on natural numbers. Then the function r^A from $\mathbb{N}$ to $\mathbb{Q}$ defined as*

$$r^A_n = \begin{cases} 2^{-n} & \text{if } \forall k \leq n.\, A(k) \\ 2^{-k} & \text{if } k \leq n \wedge \neg A(k) \wedge \forall k' < k.\, A(k') \end{cases}$$

is a fundamental sequence. We write x^A for the real number containg r^A. It holds that

$$x^A{=}0 \leftrightarrow \forall n.\, A(n) \qquad \textit{and} \qquad \neg\, x^A{=}0 \leftrightarrow \neg\, \forall n.\, A(n) \quad .$$

Proof: Let A be some decidable predicate. Obviously, r^A is a fundamental sequence as $\forall n, m.\, |r^A_n - r^A_{n+m}| < 2^{-n}$.
Next we show that $x^A{=}0 \leftrightarrow \forall n.A(n)$. The implication from right to left is obvious. For the converse direction assume that $r^A = 0$, i.e.

$$(1) \qquad \forall k.\exists n.\forall m.\, |r^A_{n+m}| < 2^{-k} \quad .$$

Suppose that $\neg A(n)$ holds for some n. Then for $\forall m{>}n.\, |r^A_m| \geq 2^{-n}$ contradicting (1). Thus, it follows that $\neg\neg A(n)$ and, therefore, also $A(n)$ as $A(n)$ is $\neg\neg$–closed. Thus, we have shown that $\forall n.A(n)$.
From the equivalence $x^A{=}0 \leftrightarrow \forall n.\, A(n)$ it follows immediately by propositional logic that $\neg\, x^A{=}0 \leftrightarrow \neg\, \forall n.\, A(n)$. □

Theorem 7.1 *Under the assumption of decidability of equality for real numbers* (i.e. $\forall x, y \in \mathbb{R}.\, x{=}y \vee \neg\, x{=}y$) *one can prove*

$$\forall x.\, A(x) \vee \neg\, \forall x.\, A(x)$$

for every primitive recursive predicate $A(x)$ on natural numbers.

Then for any formal system $\mathcal{S}$ which proves only true[19] *formulas of the form $\neg\forall x.A(x)$ with $A(x)$ primitive recursive it holds that if $\mathcal{S}$ proves the decidability of equality on real numbers there is a provable disjunction $A \vee \neg A$ such that $\mathcal{S}$ proves neither A nor $\neg A$.*

Proof: Let $A(x)$ be some primitive recursive predicate on natural numbers. From decidability of equality on $\mathbb{R}$ it follows that $x^A{=}0 \vee \neg\, x^A{=}0$ where x^A is defined as the equivalence class of the fundamental sequence r^A defined in the previous Lemma 7.3. Thus, it follows that $\forall x.\, A(x) \vee \neg\, \forall x.\, A(x)$ as by Lemma 7.3 $x^A{=}0$ is provably equivalent to $\forall x.\, A(x)$.
But now take for $A(x)$ some primitive recursive predicate for which $\forall x.\, A(x)$ is true but not derivable. Such a predicate exists for every formal system as its set of consequences is recursively enumerable by definition. But as $\neg\, \forall x.\, A(x)$ is not true it cannot be derived in our formal system $\mathcal{S}$ as by assumption $\mathcal{S}$ derives only true sentences of this form. $\square$

This sort of argument was introduced by Brouwer and called by him "method of weak counterexamples". He refuted constructively dubious principles by showing that they entail $A \vee \neg A$ for a proposition A yet undecided (e.g. Goldbach's conjecture or formerly Fermat's Last Theorem before it was proved by Wiles). As Brouwer developed this sort of argument before Gödel's Incompleteness Theorems and the advent of recursion theory he had to make use of "yet undecided propositions". Today we know by Gödel that for any formal system such propositions A exists and are of the form $\forall n.P(n)$ for some primitive recursive predicate[20].
From Theorem 7.1 it follows immediately that the apartness relation is not decidable either as $x\#y \vee \neg\, x\#y$ implies $x{=}y \vee \neg\, x{=}y$ by (AP1) of Lemma 7.2. This failure of decidability of apartness is also "empirically" true in the following sense. If real numbers x and y are given by nestings of intervals then one may *observe* in finite time that x and y are different by *observing* that *eventually* the intervals of x and y get disjoint. However, that they *always overlap* cannot be observed in finite time! Moreover, notice that decidability of apartness, i.e.

$$\forall x, y \in \mathbb{R}.\, x\#y \vee \neg\, x\#y$$

is equivalent to the "trichotomy law"

$$\forall x, y \in \mathbb{R}.\, x{<}y \vee y{<}x \vee x{=}y$$

for reals as $x\#y$ is equivalent to $x{<}y \vee y{<}x$ by definition and $x{=}y$ is equivalent to $\neg x\#y$ by (AP1) of Lemma 7.2.

[19] where $\neg\forall x.A(x)$ is understood as "true" iff $\neg A(n_0)$ holds for some natural number n_0

[20] take for example the predicate "n is not a code of a derivation of $\perp$ in the formal system under consideration"

We further notice that $\forall x, y{\in}\mathbb{R}.\ x{\leq}y \leftrightarrow (x{=}y{\vee}x{<}y)$ does not hold constructively as it would entail decidability of apartness because we have $0{\leq}|x{-}y|$, $0{=}|x{-}y| \to x{=}y$ and $0{<}|x{-}y| \to x\#y$ for all $x, y \in \mathbb{R}$.
All these consideration show that constructive analysis is much more different from classical analysis than is constructive arithmetic from classical arithmetic. Of course, much much more can be said about real numbers and real analysis from a constructive point of view. But this would go beyond the scope of our aims which are mainly logical in character. For a deeper investigation of constructive analysis we refer the reader to [TvD] and in particular [BiBr].

8 Basic Recursion Theory in HA

We will *not* give a leisurely introduction to recursion theory[21]. Instead we recall some basic facts and introduce some notation.

Definition 8.1 *The* partial recursive functions *are the subset of* $\bigcup_{k\in\mathbb{N}} \mathbb{N}^k \rightharpoonup \mathbb{N}$ *(where* $A \rightharpoonup B$ *stands for the set of all* partial *functions from* A *to* B*) defined inductively by the clauses of Definition 6.1 together with the clause*

(6) *If* $f : \mathbb{N}^{k+1} \rightharpoonup \mathbb{N}$ *is partial recursive then the function* $\mu(f) : \mathbb{N}^k \rightharpoonup \mathbb{N}$ *defined as*
$$\mu(f)(\vec{x}) \simeq \begin{cases} n & \text{if } f(\vec{x}, n) = 0 \text{ and } \forall m < n.\, f(\vec{x}, m) > 0 \\ \uparrow & \textit{otherwise} \end{cases}$$
is partial recursive, too. ◊

The most important fact about the unary partial recursive functions is that they can be Gödelized in the following most pleasant way.

Theorem 8.1 *There is a surjective map* ϕ *from* $\mathbb{N}$ *to the unary partial recursive functions satisfying the following conditions.*

(1) *The function*
$$u : \mathbb{N}^2 \rightharpoonup \mathbb{N} : (e, n) \mapsto \phi_e(n)$$
is partial recursive.

(2) *For every* $k \in \mathbb{N}$ *and* $k{+}1$*–ary partial recursive function* f *there is a* k*–ary primitive recursive function* h *such that*
$$\phi_{h(\vec{n})}(m) \simeq f(\vec{n}, m)$$
for all $\vec{n} \in \mathbb{N}^k$ *and* $m \in \mathbb{N}$.

[21]For that purpose see for example my lecture notes on the course "Logik II : Berechenbarkeitstheorie und Unvollständigkeit formales Systeme" or e.g. N. Cutland's excellent textbook "Computability" (Cambridge University Press, 1980)

Moreover, there is a ternary primitive recursive function T and a unary primitive recursive function U such that

$$\phi_n(m) = U(\mu k.\, T(n,m,k))$$

where T is called Kleene's T–predicate *and U is called the* result extraction function. *Moreover, for the predicate T it holds that $T(n,m,k) \wedge T(n,m,k') \to k = k'$.*

Proof: For details see e.g. [TvD].
We just mention the idea behind T and U. The intuitive reading of $T(n,m,k)$ is that k is a code for a (successful) computation of the algorithm with number n applied to argument m and $U(k)$ is the result of this computation. As for given n and m there exists at most one (code of a) computation "single–valuedness" of T is obvious. $\square$

Convention 8.1 *For reasons of tradition we write $\{n\}$ instead of ϕ_n. Whether $\{n\}$ means the n–th partial recursive function or the singleton set containing n will always be clear from the context as e.g. in $\{n\}(m)$ where $\{n\}$ means the partial function as it is applied to an argument. If $A(x)$ is a predicate then we write $A(\{n\}(m))$ as an abbreviation for $\exists k.T(n,m,k) \wedge A(U(k))$ which is equivalent to $\exists k.\, T(n,m,k) \wedge \forall k.T(n,m,k) \to A(U(k))$ as k is determined uniquely by n and m. The partial operation $\{.\}(.)$ is called* Kleene application *and will be used freely for building terms.*
Let e be an expression describing a partial recursive function in the free variables of e. Then by Theorem 8.1(2) there exists a primitive recursive term $\Lambda x.e$ with $\{\Lambda x.e\}(n) \simeq e[n/x]$ for all $n \in \mathbb{N}$.

As shown in [TvD] all these constructions and their relevant properties can be expressed and verified in **HA**.

We now discuss a few basic recursion–theoretic notions and facts which will be used later.

Definition 8.2 *A total recursive function is a partial recursive function which is total, i.e. defined for all arguments.*
Let $A \subseteq \mathbb{N}$. A is called recursively enumerable[22] (r.e.) *iff there is a unary partial recursive function f such that $n \in A$ iff $f(n)\downarrow$ and A is called* decidable *iff there is a unary total recursive function f such that $n \in A$ iff $f(n) = 0$.* $\diamondsuit$

Obviously, every decidable set is also recursively enumerable but the reverse inclusion does not hold.

[22]This terminology may be surprising at first sight but it isn't as one can show that a set A of natural numbers is r.e. iff A is empty or there exists a total recursive function f with $A = \{f(n) \mid n \in \mathbb{N}\}$.

Theorem 8.2 *The set $K := \{n \in \mathbb{N} \mid \{n\}(n)\downarrow\}$ is recursively enumerable but not decidable.*

Proof: If K were decidable then $\mathbb{N} \setminus K = \{n \in \mathbb{N} \mid \{n\}(n)\uparrow\}$ were recursively enumerable, i.e. there were an $e \in \mathbb{N}$ with

$$\{e\}(n)\downarrow \quad \Leftrightarrow \quad \{n\}(n)\uparrow$$

but then (putting $n = e$) it would hold that

$$\{e\}(e)\downarrow \quad \Leftrightarrow \quad \{e\}(e)\uparrow$$

which clearly is impossible. $\square$

Consequently, the *halting set* $H := \{\langle n, m\rangle \mid \{n\}(m)\downarrow\}$ is not decidable as otherwise K were decidable in contradiction to Theorem 8.2.
Notice that $n \notin K$ can be expressed in **HA** by the formula $\forall k. \neg T(\underline{n}, \underline{n}, k)$. Thus, no formal system can prove all true formulas of the form $\forall k. \neg T(\underline{n}, \underline{n}, k)$ as otherwise K were decidable.
An important consequence of this observation is that *no axiomatic system whose set of theorems is recursively enumerable can derive all true* Π_1*-sentences* where Π_1-sentences are the sentences of the form $\forall n.A(n)$ for a (primitive) recursive predicate $A(x)$.

Theorem 8.3 *Let $A_i = \{n \in \mathbb{N} \mid \{n\}(n){=}i\}$ for $i = 0, 1$. Then there is no total recursive function f with $f(n) = i$ whenever $n \in A_i$ for $i = 0, 1$.*

Proof: If there were such a recursive f then there would exist a total recursive g with $g[\mathbb{N}] \subseteq \{0, 1\}$ and

$$n \in A_0 \Rightarrow g(n) = 1 \qquad \text{and} \qquad n \in A_1 \Rightarrow g(n) = 0$$

for all $n \in \mathbb{N}$. Let $g = \{e\}$. Then $\{e\}(e) \in \{0, 1\}$ and, therefore, $e \in A_0 \cup A_1$. But this is impossible as if $e \in A_0$ then $0 = \{e\}(e) = g(e) = 1$ and if $e \in A_1$ then $1 = \{e\}(e) = g(e) = 0$. $\square$

One also says that A_0 and A_1 are *recursively inseparable* as there does not exist a recursive set P such that $A_0 \subseteq P$ and $A_1 \subseteq \mathbb{N} \setminus P$.

We conclude this section by fixing some notation concerning the primitive recursive coding of finite sequences of natural numbers by natural numbers. Such an encoding can be obtained via the coding of pairs $\langle -, - \rangle$ and its projections pr_0 and pr_1 in the following way: the natural number n is the (unique) code of the sequence

$$\mathsf{pr}_0(\mathsf{pr}_1(n)), \ldots, \mathsf{pr}_0(\mathsf{pr}_1^{\mathsf{pr}_0(n)-1}(n)), \mathsf{pr}_1^{\mathsf{pr}_0(n)}(n) \quad .$$

We write $\langle n_0, \ldots, n_{k-1} \rangle$ for the code of the sequence $n_0, \ldots, n_{k-1} \in \mathbb{N}^*$. Moreover, there exists a primitive recursive concatenation function $*$ satisfying

$$\langle s \rangle * \langle t \rangle = \langle s, t \rangle$$

for all $s, t \in \mathbb{N}^*$. A primitive recursive length function ℓh with $\ell h(\langle n_0, \ldots, n_{k-1} \rangle) = k$ is given by first projection. For $n = \langle m_0, \ldots, m_{k-1} \rangle$ and $i \in \mathbb{N}$ we define

$$n_i = \begin{cases} m_i & \text{if } i < k \\ 0 & \text{otherwise} \end{cases}$$

which mapping is primitive recursive.
We write $\langle s \rangle \preceq \langle t \rangle$ iff s is a prefix of t and $\langle s \rangle \prec \langle t \rangle$ iff s is a proper prefix of t. Obviously, $\preceq$ and $\prec$ are primitive recursive predicates on codes of sequences. Furthermore, for a function α from $\mathbb{N}$ to $\mathbb{N}$ we write $\overline{\alpha}(n)$ for (the code of) the finite sequence $\langle \alpha(0), \ldots, \alpha(n{-}1) \rangle$. This operation is primitive recursive in α.

9 Kleene's Number Realizability

Kleene' number realizability from 1945 can be considered as a mathematically precise version of the BHK interpretation of constructive logic. The idea is to consider the meaning of a proposition A as the set of "proofs" or "realizers" of A which are coded by natural numbers. Accordingly, the meaning of a proposition A is the set of natural numbers which realize A. We use $n\,\mathbf{rn}\,A$ as notation for "n realizes A".
Just as the truth–value of a compound statement depends functionally on the truth–values of its constituent parts the condition that a number realizes a compound statement can be formulated in terms of the realizability conditions for its constituent parts:

- n realizes $t = s$ iff $t = s$
- n realizes $A \wedge B$ iff $\mathsf{pr}_0(n)$ realizes A and $\mathsf{pr}_1(n)$ realizes B
- n realizes $A \to B$ iff for every m realizing A the computation $\{n\}(m)$ terminates and its result realizes B
- n realizes $A \vee B$ iff $\mathsf{pr}_0(n) = 0$ and $\mathsf{pr}_1(n)$ realizes A or $\mathsf{pr}_0(n) \neq 0$ and $\mathsf{pr}_1(n)$ realizes B
- n realizes $\forall x.A(x)$ iff for all numbers m the computation $\{n\}(m)$ terminates and its result realizes $A(m)$
- n realizes $\exists x.A(x)$ iff $\mathsf{pr}_1(n)$ realizes $A(\mathsf{pr}_0(n))$.

Notice that in arithmetic disjunction of A and B can be expressed in terms of existential quantification as

$$\exists n.\,(n{=}0{\to}A){\wedge}(n{\neq}0{\to}B)$$

for arbitrary propositions A and B. Then e realizes $\exists n.\,(n{=}0{\to}A){\wedge}(n{\neq}0{\to}B)$ iff $\mathsf{pr}_1(e)$ realizes $(\mathsf{pr}_0(e){=}0{\to}A){\wedge}(\mathsf{pr}_0(e){\neq}0{\to}B)$, i.e. $\{\mathsf{pr}_0(\mathsf{pr}_1(e))\}(n)$ realizes A for all n if $\mathsf{pr}_0(e){=}0$ and $\{\mathsf{pr}_1(\mathsf{pr}_1(e))\}(n)$ realizes B for all n if $\mathsf{pr}_0(e){\neq}0$. Thus, $\exists n.\,(n{=}0{\to}A){\wedge}(n{\neq}0{\to}B)$ is realized by $\langle 0, \{\mathsf{pr}_0(\mathsf{pr}_1(e))\}(0)\rangle$ if $\mathsf{pr}_0(e){=}0$ and by $\langle 1, \{\mathsf{pr}_1(\mathsf{pr}_1(e))\}(0)\rangle$ otherwise. On the other hand from a realizer for $A{\vee}B$ one easily can construct a realizer for $\exists n.\,(n{=}0{\to}A){\wedge}(n{\neq}0{\to}B)$ (exercise!).
Let $P(x)$ be a primitive recursive predicate. Then e realizes $\forall n.P(n)$ iff $\{e\}$ is a total recursive function and $\forall n.P(n)$ is true and e realizes $\exists n.P(n)$ iff $\mathsf{pr}_1(e)$ realizes $P(\mathsf{pr}_0(e))$, i.e. iff $P(\mathsf{pr}_0(e))$ is true. From these considerations it follows that there cannot exist a realizer e for the classically valid proposition

$$\forall n.\,(\exists k.T(n,n,k)) \vee (\forall k.\neg T(n,n,k))$$

as $\Lambda n.\mathsf{pr}_0(\{e\}(n))$ would decide the set $K = \{n{\in}\mathbb{N} \mid \exists k.T(n,n,k)\}$. Accordingly, every number realizes the classically wrong proposition

$$\neg\forall n.\,(\exists k.T(n,n,k)) \vee (\forall k.\neg T(n,n,k))$$

Thus, realizability provides us with a semantics for constructive arithmetic which is perfectly understandable from a purely classical point of view and under which classically wrong propositions get valid, i.e. realizable. Moreover, realizability semantics is consistent[23] as for example by definition the proposition $\bot$ is not realizable.
Although there are classically valid formulas which are not realizable one easily shows that for propositions of the form $\forall n.P(n)$ and $\exists n.P(n)$ with P a primitive recursive predicate it holds that they are realizable iff they are true (exercise!). (Such propositions are called Π_1 and Σ_1, respectively.) But realizability and classical validity coincide as well for Π_2 and Σ_2 propositions, i.e. propositions of the form $\forall n.\exists m.R(n,m)$ and $\exists n.\forall m.R(n,m)$, respectively, where R is a primitive recursive predicate.
Notice that quite generally a number e realizes $\forall n.\exists m.A(n,m)$ iff $\mathsf{pr}_1(\{e\}(n))$ realizes $A(n, \mathsf{pr}_0(\{e\}(n)))$ for all numbers n. In case that truth and realizability coincide for all instances of A we then get that the total recursive function f with $f(n) := \mathsf{pr}_0(\{e\}(n))$ satisfies $\forall n.A(n, f(n))$.

Next we consider *formalized number realizability* where to every formula A of **HA** one associates a new formula $n\,\mathbf{rn}\,A$ in **HA** with $n \notin FV(A)$ which expresses in the language of **HA** which n realize A.

[23]Consistent means that not all formulas are valid. This entails and is actually equivalent to the maybe more well–known requirement that for no proposition A and $\neg A$ are valid.

Definition 9.1 *The realizability relation n **rn** A is defined by induction on the structure of A via the following clauses*

$$\begin{array}{lcl}
n\,\mathbf{rn}\,P & \equiv & P \qquad \textit{where } P \textit{ is primitive} \\
n\,\mathbf{rn}\,A \wedge B & \equiv & \mathsf{pr}_0(n)\,\mathbf{rn}\,A \;\wedge\; \mathsf{pr}_1(n)\,\mathbf{rn}\,B \\
n\,\mathbf{rn}\,A \to B & \equiv & \forall m.\, m\,\mathbf{rn}\,A \to \{n\}(m)\,\mathbf{rn}\,B \\
n\,\mathbf{rn}\,A \vee B & \equiv & (\mathsf{pr}_0(n) = 0 \to \mathsf{pr}_1(n)\,\mathbf{rn}\,A) \wedge (\mathsf{pr}_0(n) \neq 0 \to \mathsf{pr}_1(n)\,\mathbf{rn}\,B) \\
n\,\mathbf{rn}\,\forall x.A(x) & \equiv & \forall m.\{n\}(m)\,\mathbf{rn}\,A(m) \\
n\,\mathbf{rn}\,\exists x.A(x) & \equiv & \mathsf{pr}_1(n)\,\mathbf{rn}\,A(\mathsf{pr}_0(n)) \quad .
\end{array}$$

$\Diamond$

Notice that when expanding the defining clauses for implication and universal quantification according to Convention 8.1 we get

$$\begin{array}{lcl}
n\,\mathbf{rn}\,A \to B & \equiv & \forall m.\, m\,\mathbf{rn}\,A \to \exists k.\, T(n,m,k) \wedge U(k)\,\mathbf{rn}\,B \\
n\,\mathbf{rn}\,\forall x.A(x) & \equiv & \forall m.\exists k.\, T(n,m,k) \wedge U(k)\,\mathbf{rn}\,A(m)
\end{array}$$

which are more precise but also less readable.

Next we show that for formulas provable in **HA** one may find a realizer by recursion on the structure of derivations.

Theorem 9.1 (Soundness of Number Realizability)
If a closed formula A can be derived in **HA** *then there is a term e possibly containing Kleene application for which $e\,\mathbf{rn}\,A$ can be derived in* **HA**.

Proof: As we want to prove soundness by induction on the structure of derivations in **HA** we have to generalise the claim of the Theorem as follows: whenever $A_1, \ldots, A_n \vdash A$ is derivable in **HA** then there is a term e such that **HA** proves

$$u_1\,\mathbf{rn}\,A_1 \wedge \ldots \wedge u_n\,\mathbf{rn}\,A_n \vdash e\,\mathbf{rn}\,A$$

where the variables u_i are fresh and e is built from the usual primitive recursive operations and the partial application operation $\{.\}(.)$ according to Convention 8.1 and $FV(e) \subseteq FV(A_1, \ldots, A_n, A) \cup \{u_1, \ldots, u_n\}$.
For the sake of readability we often write $\vec{u}\,\mathbf{rn}\,\Gamma$ when $\Gamma \equiv A_1, \ldots, A_n$ instead of $u_1\,\mathbf{rn}\,A_1 \wedge \ldots \wedge u_n\,\mathbf{rn}\,A_n$.
It is simple to show that the generalised claim holds for the structural rules (ax), (ex), (w) and (c) as primitive recursive functions contain all projections and are closed under permutation of arguments, addition of dummy arguments and identification of arguments.
($\wedge I$) If **HA** proves $\vec{u}\,\mathbf{rn}\,\Gamma \vdash e_1\,\mathbf{rn}\,A_1$ and $\vec{u}\,\mathbf{rn}\,\Gamma \vdash e_2\,\mathbf{rn}\,A_2$ then **HA** proves $\vec{u}\,\mathbf{rn}\,\Gamma \vdash \langle e_1, e_2\rangle\,\mathbf{rn}\,A_1 \wedge A_2$.

$(\wedge E)$ If **HA** proves $\vec{u}\,\mathbf{rn}\,\Gamma \vdash e\,\mathbf{rn}\,A_0{\wedge}A_1$ then **HA** proves $\vec{u}\,\mathbf{rn}\,\Gamma \vdash \mathsf{pr}_i(e)\,\mathbf{rn}\,A_i$ for $i = 0, 1$.
$({\to}I)$ If **HA** proves $\vec{u}, v\,\mathbf{rn}\,\Gamma, A \vdash e\,\mathbf{rn}\,B$ then **HA** proves $\vec{u}\,\mathbf{rn}\,\Gamma \vdash \Lambda v.e\,\mathbf{rn}\,A{\to}B$.
$({\to}E)$ If **HA** proves $\vec{u}\,\mathbf{rn}\,\Gamma \vdash e_1\,\mathbf{rn}\,A{\to}B$ and $\vec{u}\,\mathbf{rn}\,\Gamma \vdash e_2\,\mathbf{rn}\,A$ then **HA** proves $\vec{u}\,\mathbf{rn}\,\Gamma \vdash \{e_1\}(e_2)\,\mathbf{rn}\,B$.
$(\bot E)$ Suppose that **HA** proves $\vec{u}\,\mathbf{rn}\,\Gamma \vdash e\,\mathbf{rn}\,\bot$. Then **HA** proves $\vec{u}\,\mathbf{rn}\,\Gamma \vdash \bot$ because $e\,\mathbf{rn}\,\bot$ is provably equivalent to $\bot$. Thus **HA** proves also $\vec{u}\,\mathbf{rn}\,\Gamma \vdash 0\,\mathbf{rn}\,A$.
$(\forall I)$ Suppose that **HA** proves $\vec{u}\,\mathbf{rn}\,\Gamma \vdash e\,\mathbf{rn}\,A(x)$ where $x \notin FV(\Gamma)$. Then **HA** proves $\vec{u}\,\mathbf{rn}\,\Gamma \vdash \Lambda x.e\,\mathbf{rn}\,\forall x.A(x)$.
$(\forall E)$ If **HA** proves $\vec{u}\,\mathbf{rn}\,\Gamma \vdash e\,\mathbf{rn}\,\forall x.A(x)$ then **HA** proves $\vec{u}\,\mathbf{rn}\,\Gamma \vdash \{e\}(t)\,\mathbf{rn}\,A(t)$.
$(\exists I)$ If **HA** proves $\vec{u}\,\mathbf{rn}\,\Gamma \vdash e\,\mathbf{rn}\,A(t)$ then **HA** proves $\vec{u}\,\mathbf{rn}\,\Gamma \vdash \langle t, e\rangle\,\mathbf{rn}\,\exists x.A(x)$.
$(\exists E)$ Suppose that **HA** proves $\vec{u}\,\mathbf{rn}\,\Gamma \vdash e_1\,\mathbf{rn}\,\exists x.A(x)$ and $\vec{u}, u\,\mathbf{rn}\,\Gamma, A(x) \vdash e_2\,\mathbf{rn}\,B$ where $x \notin FV(B)$. Then **HA** proves $\vec{u}\,\mathbf{rn}\,\Gamma \vdash e_2[\mathsf{pr}_0(e_1), \mathsf{pr}_1(e_1)/x, u]\,\mathbf{rn}\,B$.
$(\vee I)$ and $(\vee E)$ are left as exercises.
It remains to check that the axioms of **HA** are realized. This is trivial for the equations as these are realized by any number (e.g. 0). The axiom $\neg\,\mathsf{succ}(x) = 0$ is realized e.g. by $\Lambda n.0$.
Next we consider instances of the induction scheme. First of all notice that there exists[24] a number r such that

$$\{\{r\}(\langle e_0, e_1\rangle)\}(0) = e_0 \qquad \{\{r\}(\langle e_0, e_1\rangle)\}(k{+}1) \simeq \{\{e_1\}(k)\}(\{\{r\}(\langle e_0, e_1\rangle)\}(k))$$

holds for all numbers e_0, e_1 and k and these properties can be verified in **HA**. Now, for a predicate $A(x)$ with free variables $\vec{z}$ besides x one can prove in **HA** that

$$r\,\mathbf{rn}\,A(0) \wedge (\forall x.(A(x) \to A(\mathsf{succ}(x)))) \to \forall x.A(x)$$

i.e. that r realizes $A(0) \wedge (\forall x.(A(x) \to A(\mathsf{succ}(x)))) \to \forall x.A(x)$. $\square$

Now one might hope that for every formula A one can prove in **HA** the equivalence $A \leftrightarrow \exists x.x\,\mathbf{rn}\,A$ or at least that[25] $\mathbf{HA} \vdash A$ iff $\mathbf{HA} \vdash \exists x.x\,\mathbf{rn}\,A$. Alas, this hope is in vain as for

$$\mathrm{CT}_0 \qquad\qquad (\forall x.\exists y.A(x, y)) \to \exists e.\forall x.A(x, \{e\}(x))$$

we have that $\mathbf{HA} \vdash \exists x.\,x\,\mathbf{rn}\,\mathrm{CT}_0$. But, obviously, CT_0 cannot be proved in **HA** as CT_0 cannot be proved in **PA** as for some instance of CT_0 its negation can be proved in **PA** (Exercise!). However, for an Extended Church's Thesis ECT_0 defined subsequently we can achieve our goal, namely to prove that

[24]This is a typical argument by appeal to Church's Thesis. One can easily exhibit an algorithm for the primitive recursion operator R in any programming language whatsoever and, therefore, this algorithm has a Gödel number, say r.

[25]We employ the notation $\mathbf{HA} \vdash A$ for the meta–mathematical statement that **HA** proves the sequent $\vdash A$

Theorem 9.2 (Characterisation of Number Realizability)
For all formulas A of $\mathbf{HA}$ *it holds that*

(1) $\mathbf{HA} + \mathrm{ECT}_0 \vdash A \leftrightarrow \exists x.\, x \,\mathbf{rn}\, A$

(2) $\mathbf{HA} + \mathrm{ECT}_0 \vdash A \Longleftrightarrow \mathbf{HA} \vdash \exists x.\, x \,\mathbf{rn}\, A.$

In order to formulate ECT_0 we have to introduce the following notion.

Definition 9.2 *The* almost negative *or* almost $\exists$–free *formulas are those which can be built from primitive formulas and formulas of the form $\exists x.\, t = s$ by $\wedge$, $\to$ and $\forall$.* $\Diamond$

Now we can formulate the *Extended Church's Thesis*

$$\mathrm{ECT}_0 \quad (\forall x.\, A(x) \to \exists y. B(x,y)) \to \exists e. \forall x.\, A(x) \to \exists z.\, T(e,x,z) \wedge B(x, U(z))$$

for *almost negative* A. However, for proving Theorem 9.2 we have to establish some useful properties of almost negative formulas before.
By inspection of the defining clauses for number realizability (Def. 9.1) it is evident that for all formulas A the formula $x \,\mathbf{rn}\, A$ is equivalent to an almost negative formula (by eliminating all occurrences of $\{n\}(m)$ as described in Convention 8.1.
Next we show that almost negative formulas A are equivalent to $\exists x.\, x \,\mathbf{rn}\, A$ and that this equivalence can be proved in $\mathbf{HA}$.

Lemma 9.1 *For almost negative formulas A it holds that*

(1) $\mathbf{HA} \vdash (\exists x.\, x \,\mathbf{rn}\, A) \to A$ *and*

(2) *there is a term ψ_A with* $\mathbf{HA} \vdash A \to \psi_A \,\mathbf{rn}\, A.$

and, therefore, that $\mathbf{HA} \vdash A \leftrightarrow \exists x.\, x \,\mathbf{rn}\, A.$

Proof: We prove (1) and (2) simultaneously by induction on the structure of almost negative formulas.
For primitive formulas $t{=}s$ we have that $\exists x.\, x \,\mathbf{rn}\, t{=}s$ equals $\exists x.\, t{=}s$ which is equivalent to $t{=}s$ as x is not free in $t{=}s$. Thus, (1) holds for $t{=}s$. Claim (2) holds for $t{=}s$ by putting $\psi_{t=s} \equiv 0$.
For formulas of the form $\exists x.\, t{=}s$ we have that

$$x \,\mathbf{rn}\, \exists x.\, t{=}s \;\equiv\; \mathsf{pr}_1(x) \,\mathbf{rn}\, t{=}s[\mathsf{pr}_0(x)/x]$$

and, therefore, one easily proves $x \,\mathbf{rn}\, \exists x.\, t{=}s \;\to\; \exists x.\, t{=}s$. For claim (2) one puts $\psi_{\exists x.t=s} \equiv \langle \mu x.t{=}s, 0\rangle$ where $\mu x.t{=}s$ is the (Gödel number of an) algorithm searching for the least x satisfying the decidable condition $t{=}s$. Obviously, $\mu x.t{=}s$ terminates if $\exists x. x \,\mathbf{rn}\, t{=}s$ and, therefore, $\mathbf{HA}$ proves that $\exists x.\, t{=}s \;\to$

$0\,\mathbf{rn}\,t{=}s[\mu x.t{=}s/x]$. But as $0\,\mathbf{rn}\,t{=}s[\mu x.t{=}s/x]$ is easily seen to be equivalent to $\langle \mu x.t{=}s, 0\rangle\,\mathbf{rn}\,\exists x.\,t{=}s$ it follows that $\mathbf{HA} \vdash \exists x.\,t{=}s \to \psi_{\exists x.t=s}\,\mathbf{rn}\,\exists x.\,t{=}s$.
Suppose as induction hypothesis that the almost negative formulas A and B satisfy the claims (1) and (2).
Then claim (1) holds for $A \wedge B$ as $y\,\mathbf{rn}\,A \to A$ and $z\,\mathbf{rn}\,B \to B$ hold by induction hypothesis and, therefore, also $(\mathsf{pr}_0(x)\,\mathbf{rn}\,A \wedge \mathsf{pr}_1(x)\,\mathbf{rn}\,B) \to A{\wedge}B$, i.e. $x\,\mathbf{rn}\,A{\wedge}B \to A{\wedge}B$. Claim (2) for $A \wedge B$ follows readily by putting $\psi_{A\wedge B} \equiv \langle\psi_A, \psi_B\rangle$.
Now we show (1) for $A{\to}B$. Suppose $x\,\mathbf{rn}\,A{\to}B$, i.e. $\forall y.\, y\,\mathbf{rn}\,A \to \{x\}(y)\,\mathbf{rn}\,B$. As by induction hypothesis $A \to \psi_A\,\mathbf{rn}\,A$ we get that $A \to \{x\}(\psi_A)\,\mathbf{rn}\,B$ and as $z\,\mathbf{rn}\,B \to B$ by induction hypothesis for B it follows that $A{\to}B$. As this argument can be formalised in $\mathbf{HA}$ it follows that $\mathbf{HA} \vdash x\,\mathbf{rn}\,A{\to}B \to A{\to}B$ and we have established claim (1) for $A{\to}B$. Claim (2) for $A{\to}B$ follows by putting $\psi_{A\to B} \equiv \Lambda x.\psi_B$ using that by induction hypothesis we have $x\,\mathbf{rn}\,A \to A$ and $B \to \psi_B\,\mathbf{rn}\,B$.
We leave the case of the universal quantifier as an exercise.
As (2) entails in particular that $\mathbf{HA} \vdash A \to \exists x.\,x\,\mathbf{rn}\,A$ for almost negative A it follows from (1) and (2) that $\mathbf{HA} \vdash A \leftrightarrow \exists x.\,x\,\mathbf{rn}\,A$ for almost negative A. $\square$

The following *idempotency* of formalized realizability appears as a corollary.

Corollary 9.1 *For every formula A in the language of* $\mathbf{HA}$ *it holds that* $\mathbf{HA} \vdash \exists x.\,x\,\mathbf{rn}\,A \leftrightarrow \exists x.\,x\,\mathbf{rn}\,(\exists x.\,x\,\mathbf{rn}\,A)$.

Proof: Straightforward exercise using Lemma 9.1 and that $x\,\mathbf{rn}\,A$ is provably equivalent to an almost negative formula. $\square$

Using Lemma 9.1 one can now show that

Lemma 9.2 *For every instance A of* ECT_0 *we have* $\mathbf{HA} \vdash \exists e.\,e\,\mathbf{rn}\,A$.

Proof: Let A be almost negative. Suppose that $e\,\mathbf{rn}\,\forall x(A(x) \to \exists y.B(x,y))$, i.e. that

$$\forall x, n.(n\,\mathbf{rn}\,A(x) \to \exists z.\,T(\{e\}(x), n, z) \wedge U(z)\,\mathbf{rn}\,\exists y.B(x,y))$$

Substituting ψ_A for n we get

$$\forall x.(\psi_A\,\mathbf{rn}\,A(x) \to \exists z.\,T(\{e\}(x), \psi_A, z) \wedge U(z)\,\mathbf{rn}\,\exists y.B(x,y))$$

As A is almost negative from Lemma 9.1 we get $n\,\mathbf{rn}\,A(x) \to \psi_A\,\mathbf{rn}\,A(x)$ and, therefore, we have

$$\forall x, n.(n\,\mathbf{rn}\,A(x) \to \exists z.\,T(\{e\}(x), \psi_A, z) \wedge U(z)\,\mathbf{rn}\,\exists y.B(x,y))$$

i.e.

$$\forall x, n.(n\,\mathbf{rn}\,A(x) \to \exists z.\,T(\{e\}(x), \psi_A, z) \wedge \mathsf{pr}_1(U(z))\,\mathbf{rn}\,B(x, \mathsf{pr}_0(U(z))))$$

Let $t_1[e] \equiv \Lambda x.\mathsf{pr}_0(\{\{e\}(x)\}(\psi_A))$. As

$$\forall x(A(x) \to \exists z.\, T(t_1[e], x, z) \wedge B(x, U(z)))$$

is realized by $t_2[e] \equiv \Lambda x.\Lambda n.\langle \mu z.T(t_1[e], x, z), \langle 0, \mathsf{pr}_1(\{\{e\}(x)\}(\psi_A))\rangle\rangle$ we finally get that $\Lambda e.\langle t_1[e], t_2[e]\rangle$ realizes

$$\forall x.(A(x) {\to} \exists y.B(x, y)) \to \exists e.\forall x.(A(x) {\to} \exists z.\, T(e, x, z) {\wedge} B(x, U(z)))$$

as desired.
As the whole argument can be formalized within **HA** the claim follows. $\square$

The assumption that A is almost negative has been used for making the choice of y with $B(x, y)$ independent from the realizer of the premiss A. Actually, adding the unrestricted[26] scheme

$$\mathrm{ECT}_0^* \qquad (\forall x.\, A \to \exists y.B(x, y)) \to \exists e.\forall x.\, A \to \exists z.T(e, x, z) {\wedge} B(x, U(z))$$

to **HA** is inconsistent as can be seen when instantiating A by $\exists z.\, T(x, x, z) \vee \neg\exists z.\, T(x, x, z)$ and $B(x, y)$ by $(y{=}0 \wedge \exists z.T(x, x, z)) \vee (y{=}1 \wedge \neg\exists z.\, T(x, x, z))$ (*cf.* the Remark on p.197 of [Tr73]).[27]

Now we are ready to give the

Proof (of Theorem 9.2):
(1) We show that $\mathbf{HA} + \mathrm{ECT}_0 \vdash A \leftrightarrow \exists x.\, x\, \mathbf{rn}\, A$ by induction on the structure of formulas A in **HA**.
Condition (1) is obvious for atomic formulas.
($\wedge$) Obviously, $\exists x.\, x\, \mathbf{rn}\, A {\wedge} B \leftrightarrow \exists x.x\, \mathbf{rn}\, A \wedge \exists x.x\, \mathbf{rn}\, B$ is provable in **HA**. Thus, as by induction hpothesis $\mathbf{HA} + \mathrm{ECT}_0 \vdash A \leftrightarrow \exists x.\, x\, \mathbf{rn}\, A$ and $\mathbf{HA} + \mathrm{ECT}_0 \vdash B \leftrightarrow \exists x.\, x\, \mathbf{rn}\, B$ it follows that $\mathbf{HA} + \mathrm{ECT}_0 \vdash A {\wedge} B \leftrightarrow \exists x.x\, \mathbf{rn}\, A {\wedge} B$.
($\to$) By induction hypothesis A and B satisfy (1). Therefore, $A {\to} B$ is equivalent to $\forall x.\, x\, \mathbf{rn}\, A \to \exists y.\, y\, \mathbf{rn}\, B$ which by ECT_0 (as $x\, \mathbf{rn}\, A$ is almost negative) is equivalent to $\exists z.\forall x.\, x\, \mathbf{rn}\, A \to \{z\}(x)\, \mathbf{rn}\, B$, i.e. $\exists z.\, z\, \mathbf{rn}\, A {\to} B$.

[26] i.e. there are no restrictions on the syntactic form of A

[27] For this choice of A and B the premiss of ECT_0^* is obviously provable in **HA**. Thus, by ECT_0^* it follows that $\exists e.\forall x.\, A(x) \to \{e\}(x){\downarrow} \wedge B(x, \{e\}(x))$. As $\neg\neg A(x)$ is provable in **HA** it follows from ECT_0^* that $\exists e.\forall x.\neg\neg(\{e\}(x){\downarrow} \wedge B(x, \{e\}(x))$, i.e. more explicitly that

$$(1) \qquad \forall x.\, \neg\neg\big(\{e\}(x){\downarrow} \wedge \big((\{e\}(x){=}0 \wedge \{x\}(x){\downarrow}) \vee (\{e\}(x){=}1 \wedge \neg\{x\}(x){\downarrow})\big)\big)$$

for some e. Let e_0 be a Gödel number of an algorithm such that $\{e_0\}(x){\downarrow}$ iff $\{e\}(x){=}1$. Now instantiating (1) by e_0 it follows that

$$(2) \qquad \neg\neg\big(\{e\}(e_0){\downarrow} \wedge \big((\{e\}(e_0){=}0 \wedge \{e_0\}(e_0){\downarrow}) \vee (\{e\}(e_0){=}1 \wedge \neg\{e_0\}(e_0){\downarrow})\big)\big)$$

which, however, is contradictory as due to the nature of e_0 if $\{e\}(e_0){=}0$ then $\neg\{e_0\}(e_0){\downarrow}$ and if $\{e\}(e_0){=}1$ then $\{e_0\}(e_0){\downarrow}$.

($\forall$) By induction hypothesis $A(y)$ satisfies (1). Therefore, $\forall y.A(y)$ is equivalent to $\forall y.\exists x.\, x \,\mathbf{rn}\, A(y)$ which by ECT_0 is equivalent to $\exists z.\forall y.\, \{z\}(y) \,\mathbf{rn}\, A(y)$, i.e. $\exists z.\, z \,\mathbf{rn}\, \forall y.A(y)$.
($\exists$) Assume as induction hypothesis that $\mathbf{HA} + \mathrm{ECT}_0 \vdash A(x) \leftrightarrow \exists z.\, z \,\mathbf{rn}\, A(x)$. By definition $x \,\mathbf{rn}\, \exists x.\, A(x) \equiv \mathsf{pr}_1(x) \,\mathbf{rn}\, A(\mathsf{pr}_0(x))$. Thus, we have $\mathbf{HA} + \mathrm{ECT}_0 \vdash x \,\mathbf{rn}\, \exists x.\, A(x) \to A(\mathsf{pr}_0(x))$ as it follows from the induction hypothesis (by substituting $\mathsf{pr}_0(x)$ for x) that $\mathbf{HA} + \mathrm{ECT}_0 \vdash \mathsf{pr}_1(x) \,\mathbf{rn}\, A(\mathsf{pr}_0(x)) \to A(\mathsf{pr}_0(x))$. But from $\mathbf{HA} + \mathrm{ECT}_0 \vdash x \,\mathbf{rn}\, \exists x.\, A(x) \to A(\mathsf{pr}_0(x))$ it follows immediately that $\mathbf{HA} + \mathrm{ECT}_0 \vdash x \,\mathbf{rn}\, \exists x.\, A(x) \to \exists x.\, A(x)$ and, therefore, also that $\mathbf{HA} + \mathrm{ECT}_0 \vdash \exists x.\, x \,\mathbf{rn}\, \exists x.\, A(x) \to \exists x.\, A(x)$.
On the other hand by induction hypothesis we have $\mathbf{HA} + \mathrm{ECT}_0 \vdash A(x) \to \exists z.\, z \,\mathbf{rn}\, A(x)$. As $\mathbf{HA} \vdash z \,\mathbf{rn}\, A(x) \to \langle x, z\rangle \,\mathbf{rn}\, \exists x.A(x)$ and, therefore, also $\mathbf{HA} \vdash z \,\mathbf{rn}\, A(x) \to \exists x.\, x \,\mathbf{rn}\, \exists x.A(x)$ it follows that $\mathbf{HA} \vdash \exists z.\, z \,\mathbf{rn}\, A(x) \to \exists x.\, x \,\mathbf{rn}\, \exists x.A(x)$. Thus, $\mathbf{HA} + \mathrm{ECT}_0 \vdash A(x) \to \exists x.\, x \,\mathbf{rn}\, \exists x.A(x)$ from which it readily follows that $\mathbf{HA} + \mathrm{ECT}_0 \vdash \exists x.A(x) \to \exists x.\, x \,\mathbf{rn}\, \exists x.A(x)$.
($\vee$) This case is redundant as disjunction can be expressed in terms of the other connectives and quantifiers.

(2) Suppose that $\mathbf{HA} \vdash \exists e.\, e \,\mathbf{rn}\, A$. Then even more $\mathbf{HA} + \mathrm{ECT}_0 \vdash \exists e.\, e \,\mathbf{rn}\, A$ from which it follows by the already established claim (1) that $\mathbf{HA} + \mathrm{ECT}_0 \vdash A$. Suppose that $\mathbf{HA} + \mathrm{ECT}_0 \vdash A$. Then $\mathbf{HA} \vdash B_1 \wedge \ldots \wedge B_n \to A$ where the B_i are instances of ECT_0. By Theorem 9.1 we have $\mathbf{HA} \vdash \exists e.\, e \,\mathbf{rn}\, (B_1 \wedge \ldots \wedge B_n \to A)$ from which it follows that $\mathbf{HA} \vdash \exists e.\, e \,\mathbf{rn}\, A$ as for the B_i we have $\mathbf{HA} \vdash \exists e.\, e \,\mathbf{rn}\, B_i$ by Lemma 9.2. $\square$

In the next subsection we consider a variant of Kleene's number realizability where realizability of A entails A. This will be used to establish the *Disjunction and Existence Property of* **HA** which after all was our main motivation for constructive logic in section 1.

Disjunction and Existence Property via rnt–Realizability

It may be considered as a defect of Kleene's number realizability that in general one cannot prove $(\exists x.\, x \,\mathbf{rn}\, A) \to A$ in **HA**. Even worse for particular instances A of CT_0 one can prove $\exists x.\, x \,\mathbf{rn}\, A$ in **HA** but not A itself. This 'defect' can be remedied by introducing a notion of *number realizability combined with truth* (denoted by $n \,\mathbf{rnt}\, A$). The defining clauses for number realizability combined with truth will be like those for number realizability with a single, but crucial exception: $n \,\mathbf{rnt}\, A {\to} B$ iff the algorithm with Gödel number n sends **rnt**–realizers of A to **rnt**–realizers of B *and* A implies B. This little ammendment will render it a triviality to show that $n \,\mathbf{rnt}\, A$ implies A provably in **HA**.

Definition 9.3 (**rnt**–realizability)
The realizability relation n **rnt** *A is defined by induction on the structure of A*

via the following clauses

$$\begin{array}{lcl}
n\,\mathbf{rnt}\,P & \equiv & P \qquad \textit{where } P \textit{ is primitive} \\
n\,\mathbf{rnt}\,A \wedge B & \equiv & \mathsf{pr}_0(n)\,\mathbf{rnt}\,A \ \wedge\ \mathsf{pr}_1(n)\,\mathbf{rnt}\,B \\
n\,\mathbf{rnt}\,A \to B & \equiv & (\forall m.\, m\,\mathbf{rnt}\,A \to \{n\}(m)\,\mathbf{rnt}\,B) \wedge (A \to B) \\
n\,\mathbf{rnt}\,A \vee B & \equiv & (\mathsf{pr}_0(n) = 0 \to \mathsf{pr}_1(n)\,\mathbf{rnt}\,A) \wedge (\mathsf{pr}_0(n) \neq 0 \to \mathsf{pr}_1(n)\,\mathbf{rnt}\,B) \\
n\,\mathbf{rnt}\,\forall x.A(x) & \equiv & \forall m.\{n\}(m)\,\mathbf{rnt}\,A(m) \\
n\,\mathbf{rnt}\,\exists x.A(x) & \equiv & \mathsf{pr}_1(n)\,\mathbf{rnt}\,A(\mathsf{pr}_0(n)) \quad . \qquad \Diamond
\end{array}$$

Now we show that **rnt**–realizability entails truth and based on that a soundness theorem for **rnt**-realizability.

Theorem 9.3 *For all formulas A in the language of* **HA** *it holds that*

(1) $\mathbf{HA} \vdash (\exists x.\, x\,\mathbf{rnt}\,A) \to A$

(2) *If* $\mathbf{HA} \vdash A$ *then there is a number e with* $\mathbf{HA} \vdash \{e\}(\langle\vec{x}\rangle)\,\mathbf{rnt}\,A$ *where $\vec{x}$ contains all free variables of A.*

Proof: Claim (1) is established by straightforward induction on the structure of A. The case of implication (which prevents an anlogous result for **rn**–realizability to hold) is trivial for **rnt**–realizability as

$$(\forall m.\, m\,\mathbf{rnt}\,A \to \{n\}(m)\,\mathbf{rnt}\,B) \wedge (A \to B) \to A \to B$$

is a logical triviality.
The proof of claim (2) is quite similar to the proof of the Correctnes Theorem 9.1 for **rn**–realizability. □

As a consequence of Theorem 9.3 we can establish for **HA** the Disjunction and Existence Property which in the introduction we advocated as *the* criterion for constructivity.

Theorem 9.4 (Disjunction and Existence Property)

(1) *If* $\mathbf{HA} \vdash A \vee B$ *with A and B closed then* $\mathbf{HA} \vdash A$ *or* $\mathbf{HA} \vdash B$

(2) *If* $\mathbf{HA} \vdash \exists x.A(x)$ *and $\exists x.A(x)$ is closed then there exists a number n such that* $\mathbf{HA} \vdash A(\underline{n})$.

Proof: First notice that from Theorem 9.3(2) it follows that for every closed formula A with $\mathbf{HA} \vdash A$ there exists a number n with $\mathbf{HA} \vdash \underline{n}\,\mathbf{rnt}\,A$.
Suppose $\mathbf{HA} \vdash A \vee B$ for closed formulas A and B. Then for some n we have $\mathbf{HA} \vdash \underline{n}\,\mathbf{rnt}\,A \vee B$. If $\mathsf{pr}_0(n)=0$ then $\mathbf{HA} \vdash \mathsf{pr}_1(\underline{n})\,\mathbf{rnt}\,A$ from which it follows by Theorem 9.3(1) that $\mathbf{HA} \vdash A$. Analogously, if $\mathsf{pr}_0(n)>0$ one shows that $\mathbf{HA} \vdash B$. The proof of the second claim is analogous and left to the reader (exercise!). □

10 Markov's Principle

There is a logical principle called *Markov's Principle* which is somewhat dubious from the point of view of the BHK interpretation but nevertheless valid from the point of view of Kleene's number realizability. Markov's Principle

$$\mathrm{MP} \qquad (\forall x.\, A(x) \vee \neg A(x)) \to \neg\neg\exists x.A(x) \to \exists x.A(x)$$

says that for *decidable* predicates $A(x)$ the existence statement $\exists x.A(x)$ is $\neg\neg$–closed.
The realizability of MP can be seen as follows. One easily sees that **HA** proves the realizability of

$$(\forall x.\, A(x) \vee \neg A(x)) \to \exists e.\forall x.\, [\{e\}(x)\downarrow \wedge (A(x) \leftrightarrow \{e\}(x)=0)]$$

and, therefore, also that

$$(\forall x.\, A(x) \vee \neg A(x)) \to \exists e.\forall x.\, [A(x) \leftrightarrow \exists y.T(e,x,y)] \quad .$$

Thus, for showing that MP is realizable it suffices (exercise!) to show that

$$\mathrm{MP}_0 \qquad \neg\neg\exists z.T(x,y,z) \to \exists z.T(x,y,z)$$

is realizable. But this is the case as a realizer for MP_0 can be found easily using unbounded search by taking for example $\Lambda u.\Lambda v.\langle \mu z.\, T(\mathsf{pr}_0(u), \mathsf{pr}_1(u), z), 0\rangle$. This works as $\neg\neg\exists z.T(n,m,z)$ has a realizer if and only if there is a k with $T(n,m,k)$ and this (unique) k can be found by stupid search via μ as $T(n,m,\_)$ is a (primitive recursively) decidable predicate.
Notice that often instead of the somewhat peculiar axiom MP_0 one postulates the schema

$$\mathrm{MP}_{\mathrm{PR}} \qquad \neg\neg\exists x.A(x) \to \exists x.A(x) \qquad (A \text{ primitive recursive})$$

which under assumption of CT_0 is equivalent to MP (by considerations similar to the ones above).
The pragmatic relevance of MP in CRM (Constructive Recursive Mathematics) axiomatized by **HA** + ECT_0 + MP is that one uses the instance MP_0 of MP for showing the termination of computations by contradiction ("if a computation cannot diverge then it must terminate"). This was precisely Markov's motivation for introducing the principle called after him !
We conclude this section by observing that in presence of MP Kleene's number realizability can be axiomatized using the schema

$$\mathrm{ECT}_0' \qquad (\forall x.(\neg A(x) \to \exists y.B(x,y))) \to \exists e.\forall x.(\neg A(x) \to B(x,\{e\}(x)))$$

without any restriction on $A(x)$. In presence of MP every almost negative formula is provably equivalent to a negative one, i.e. a formula without $\exists$ and $\vee$, since for primitive recursive $P(x)$ the formula $\exists x.P(x)$ is equivalent by MP to $\neg\neg\exists x.P(x)$ which in turn is provably equivalent to the negative formula $\neg\forall x.\neg P(x)$. Using this observation one can show easily using Theorem 9.2 (exercise!) that under $\mathbf{HA} + \mathrm{MP}$ the principles ECT_0 and ECT_0' are provably equivalent from which it follows that

Theorem 10.1 *For all formulas A of* $\mathbf{HA}$ *it holds that*

(1) $\mathbf{HA} + \mathrm{MP} + \mathrm{ECT}_0' \vdash A \leftrightarrow \exists x.\, x\,\mathbf{rn}\,A$

(2) $\mathbf{HA} + \mathrm{MP} + \mathrm{ECT}_0' \vdash A \Longleftrightarrow \mathbf{HA} + \mathrm{MP} \vdash \exists x.\, x\,\mathbf{rn}\,A.$

This axiomatization of realizability may be considered as simpler than the one of Theorem 9.2 as ECT_0' is simpler than ECT_0 in the sense that it avoids the somewhat clumsy syntactic notion 'almost negative'.

11 Kleene's Function Realizability

For the language **EL** one can define a notion of *function realizability*, i.e. specify for every proposition A a predicate on functions saying which α realize A (notation $\alpha\,\mathbf{rf}\,A$). Function realizability was introduced originally in [KV] for the purpose of giving an mathematical model of a (formalization of) Brouwerian Intuitionism.
In order to formulate the realizability conditions for implication and universal quantification (over functions) we need to specify in which sense functions can be applied to functions giving as result a number or a function:

$$\begin{array}{lcl} \alpha(\beta) = x & \Leftrightarrow & \exists n.\, \alpha(\overline{\beta}(n)) = x + 1 \wedge \forall m < n.\, \alpha(\overline{\beta}(m)) = 0 \\ \alpha|\beta = \gamma & \Leftrightarrow & \forall x.\, \alpha(\langle x\rangle * \beta) = \gamma(x) \quad . \end{array}$$

The operational intuition behind the definition of $\alpha(\beta)$ is that one 'scan's through the list $\alpha(0), \alpha(1), \ldots, \alpha(n), \ldots$ until one has found the first (code of a) prefix $\overline{\beta}(n)$ with $\alpha(\overline{\beta}(n)) > 0$ and the result of the computation will be $\alpha(\overline{\beta}(n)) - 1$. Of course, $\alpha(\beta)$ will be defined for all β if and only if $\forall\beta.\exists n.\alpha(\overline{\beta}(n)) > 0$. Obviously, this condition is also necessary and sufficient for $\alpha|\beta$ being defined for all β.
Notice that the functionals from $\mathbb{N}^{\mathbb{N}}$ to $\mathbb{N}$ induced by some α are precisely(!) the *continuous* functionals when $\mathbb{N}$ is endowed with the discrete toplogy and $\mathbb{N}^{\mathbb{N}}$ is endowed with the so–called *Baire topology* whose open sets are arbitray unions of sets of the form

$$U_s := \{\alpha \in \mathbb{N}^{\mathbb{N}} \mid \exists n.\, \overline{\alpha}(n) = s\}$$

for codes s of finite sequence of natural numbers. Thus, a functional $F : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ is continuous iff for all α there exists a natural number n such that $F(\beta) = F(\alpha)$

for all β with $\alpha(i) = \beta(i)$ for all $i < n$. For ease of notation we write $\mathcal{B}$ for $\mathbb{N}^{\mathbb{N}}$ endowed with the Baire topology.
Notice that $\mathcal{B}$ and $\mathcal{B} \times \mathcal{B}$ are isomorphic as topological spaces by sending α to $(\lambda n.\mathsf{pr}_0(\alpha(n)), \lambda n.\mathsf{pr}_1(\alpha(n)))$. The inverse is given by sending the pair (α, β) to $\langle \alpha, \beta\rangle := \lambda n.\langle \alpha(n), \beta(n)\rangle \in \mathcal{B}$. Accordingly, we have for all $k \in \mathbb{N}$ that $\mathcal{B} \cong \mathcal{B}^k$ where the isomorphism is given by argumentwise coding of k–tuples

$$\langle \alpha_1, \dots, \alpha_k\rangle := \lambda n.\langle \alpha_1(n), \dots, \alpha_k(n)\rangle \quad .$$

We write $\alpha(\beta_1, \dots, \beta_k)$ and $\alpha|(\beta_1, \dots, \beta_k)$ for $\alpha(\langle\beta_1, \dots, \beta_k\rangle)$ and $\alpha|(\langle\beta_1, \dots, \beta_k\rangle)$, respectively.
On $\mathcal{B}$ one may develop a certain amount of "recursion theory" which suffices for studying function realizability.

Theorem 11.1 *There are functions $\upsilon, \sigma, \rho \in \mathcal{B}$ such that for all $\alpha, \beta, \gamma \in \mathcal{B}$*

(1) $\upsilon|(\alpha, \beta) \simeq \alpha|\beta$

(2) $(\sigma|(\alpha, \beta))|\gamma \simeq \alpha|(\beta, \gamma)$

(3) $\rho|(\alpha, \beta, \gamma) \simeq \alpha$ *if* $\gamma(0) = 0$ *and*
$\rho|(\alpha, \beta, \gamma) = \alpha|(\gamma, \rho|(\alpha, \beta, \gamma'))$ *if* $\gamma'(0) + 1 = \gamma(0)$ *and* $\gamma'(n + 1) = \gamma(n + 1)$ *for all* $n \in \mathbb{N}$.

Proof: A straightforward, but lengthy and tedious programming exercise which doesn't give new insight. For details see Ch. 3 Sect. 7 of [TvD]. □

Obviously, υ is a universal function, σ provides functional abstraction on the level of codes in Baire space and ρ provides a primitive recursor.

Now we have assembled enough machinery for defining the notion of function realizability à la Kleene.

Definition 11.1 *The realizability relation α **rf** A is defined by induction on the structure of A via the following clauses*

$$\begin{array}{lcl}
\alpha\,\mathbf{rf}\,P & \equiv & P \qquad \textit{where } P \textit{ is primitive} \\
\alpha\,\mathbf{rf}\,A \wedge B & \equiv & \mathsf{pr}_0(\alpha)\,\mathbf{rf}\,A \;\wedge\; \mathsf{pr}_1(\alpha)\,\mathbf{rf}\,B \\
\alpha\,\mathbf{rf}\,A \to B & \equiv & \forall\beta.\,\beta\,\mathbf{rf}\,A \to \alpha|\beta\,\mathbf{rf}\,B \\
\alpha\,\mathbf{rf}\,A \vee B & \equiv & (\mathsf{pr}_0(\alpha)(0) = 0 \to \mathsf{pr}_1(\alpha)\,\mathbf{rf}\,A) \wedge (\mathsf{pr}_0(\alpha)(0) \neq 0 \to \mathsf{pr}_1(\alpha)\,\mathbf{rf}\,B) \\
\alpha\,\mathbf{rf}\,\forall x.A(x) & \equiv & \forall x.\;\alpha|\lambda n.x\,\mathbf{rf}\,A(x) \\
\alpha\,\mathbf{rf}\,\forall\beta.A(\beta) & \equiv & \forall\beta.\;\alpha|\beta\,\mathbf{rf}\,A(\beta) \\
\alpha\,\mathbf{rf}\,\exists x.A(x) & \equiv & \mathsf{pr}_1(\alpha)\,\mathbf{rf}\,A(\mathsf{pr}_0(\alpha)(0)) \\
\alpha\,\mathbf{rf}\,\exists\beta.A(\beta) & \equiv & \mathsf{pr}_1(\alpha)\,\mathbf{rf}\,A(\mathsf{pr}_0(\alpha)) \quad .
\end{array}$$
◊

As in Section 9 for number realizability one may show that propositions derivable in **EL** are realized by an element of Baire space. When proving this soundness theorem by induction on the structure of derivations one easily observes that *the desired realizer can always most naturally be chosen as a recursive element of Baire space.*
Also analogously to section 9 (Theorem 9.2) one may show that function realizability can be axiomatized by **EL** + GC where GC is the schema of *Generalised Continuity*

$$\text{GC} \quad (\forall\alpha.\ A(\alpha) \to \exists\beta.B(\alpha,\beta)) \to \exists\gamma.\forall\alpha.\ (A(\alpha) \to B(\alpha,\gamma|\alpha))$$

for *almost negative* A. Analogous to Section 9 the almost negative formulas of **EL** are those built up from formulas of the form $\bot$, $t = s$, $\exists x.t = s$ or $\exists\alpha.t = s$ by $\wedge$, $\to$ and $\forall$.
Notice that GC entails in particular the following *Choice Principle for Baire Space*

$$\text{AC}_{11} \qquad \forall\alpha.\exists\beta.A(\alpha,\beta) \to \exists\gamma.\forall\alpha.A(\alpha,\gamma|\alpha))$$

which in turn implies

$$\text{AC}_{10} \qquad \forall\alpha.\exists n.A(\alpha,n) \to \exists\gamma.\forall\alpha.A(\alpha,\gamma(\alpha))$$

and axiom of choice for numbers

$$\text{AC}_{00} \qquad \forall n.\exists m.A(n,m) \to \exists\alpha.\forall x.A(n,\alpha(n)) \quad .$$

Notice that AC_{10} is inconsistent with Church's Thesis as one cannot read off a Gödel number of a function from an initial segment of the function (exercise!).
One can say that whereas CRM (Constructive Recursive Mathematics) is based on choice principles combined with the assumption that all functions are *computable* BIM (Brouwer's Intuitionistic Mathematics) is based on choice principles combined with the assumption that all function(al)s on Baire space are *continuous* and the principle of *Bar Induction*

$$\begin{aligned}\text{BI} \quad & (\forall\alpha.\exists n.P(\overline{\alpha}(n))) \to (\forall n.(P(n) \to \forall m.P(n{*}m))) \to (\forall n.P(n) \to Q(n)) \\ & \to (\forall n.\,(\forall m.Q(n{*}\langle m\rangle)) \to Q(n)) \to Q(\langle\rangle)\end{aligned}$$

i.e. a principle of (transfinite) induction over well–founded countably branching trees.[28]

[28]From a classical point of view the tree T consists of all (codes of) finite sequences of numbers which do *not* satisfy the predicate P. Classically, the well–foundedness of T is equivalent to the the following induction principle for T: for every predicate Q on (the nodes of) T the root $\langle\rangle$ satisfies Q provided Q holds for a node s whenever Q holds for all sons of s.
Constructively, the validity of this induction principle entails well-foundedness but the reverse implication is highly nonconstructive as we shall see later!

There are no direct applications of general Bar Induction but one of its consequences is most important for intuitionistic analysis, namely *König's Lemma* saying that a well–founded finitely branching tree is always finite. From König's Lemma (called "Fan Theorem" by Brouwer) one can show that all functions from $[0,1]$ to $\mathbb{R}$ are bounded which fails in CRM (*cf.* [TvD]).

We conclude this section by showing that

Church's Thesis and König's Lemma are Incompatible

That König's Lemma is inconsistent with Church's Thesis and, therefore, wrong in CRM follows from the existence of the so–called *Kleene tree*, an *infinite binary tree containg no computable infinite paths.*

Theorem 11.2 *There is a prefix–closed decidable subset T_K of $\{0,1\}^*$ which is infinite but for every recursive $\alpha \in \{0,1\}^{\mathbb{N}}$ there is an $n \in \mathbb{N}$ with $\overline{\alpha}(n) \notin T_K$.*

Proof: The set T_K is defined as follows

$$\langle b_0, \ldots, b_{n-1} \rangle \in T_K \quad :\Leftrightarrow \quad \forall m, k{<}n.\, T(m,m,k) \to (b_m{=}0 \leftrightarrow U(k){>}0) \quad .$$

Obviously, the set T_K is prefix–closed, decidable and contains elements of arbitrary finite length.
For the purpose of showing that T_K contains no infinite recursive path recall from Theorem 8.3 that the sets A_0 and A_1 are recursively where $A_i = \{n \in \mathbb{N} \mid \{n\}(n) = i\}$ for $i = 0, 1$.
Suppose that $\alpha \in \{0,1\}^{\mathbb{N}}$ is recursive and $\overline{\alpha}(n) \in T_K$ for all $n \in \mathbb{N}$. Then

$$\begin{array}{lll} \alpha(m) = 1 & \text{if } m \in A_0 & \text{and} \\ \alpha(m) = 0 & \text{if } m \in A_1 & \end{array}$$

from which it follows that A_0 and A_1 were recursively separable in contradiction to Theorem 8.3. Thus, T_K does not contain an infinite recursive path. $\Box$

Thus, König's Lemma fails in presence of Church's Thesis as we can quite explictely exhibit a binary well-founded tree T_K, the Kleene tree, which, nevertheless, is infinite as one easily constructs finite paths of arbitrary length.
The Kleene tree was constructed by Kleene in [KV] in order to show that number realizability does not model Brouwer's Intuitionistic Mathematics.

12 Higher Type Arithmetic

The aim of this section is to present a *higher type* extension $\mathbf{HA}^\omega$ of $\mathbf{HA}$, namely an extension which allows one to speak about (constructive) functionals of higher type. The higher types are built inductively from base type 0 via the binary type

forming operators $\times$ and $\rightarrow$. The elements of type $\sigma\times\tau$ are thought of as pairs $\langle a, b\rangle$ where a is of type σ and b is of type τ. The elements of type $\sigma\rightarrow\tau$ are thought of as (constructive) functions mapping objects of type σ to objects of type τ.
There are various motivations for such an extension. First of all when developing constructive analysis in order to speak about functions on constructive reals one needs function(al)s sending objects of type $1 := 0\rightarrow 0$ to objects of type 1 as fundamental sequences are given by functions from 0 to 0 (implicitly using an effective coding of $\mathbb{Q}$ by $\mathbb{N}$). Accordingly, for representing real–valued functionals on function spaces (as e.g. definite integrals) one needs functionals of type $((0\rightarrow 0)\rightarrow 0\rightarrow 0)\rightarrow 0\rightarrow 0$. Of course, one can speak about such objects in **HA** or **EL** by using numbers or first order functions as *codes* or *realizers* for higher type objects. However, it is often more convenient to have a *term language* available by which one may *directly* speak about higher type objects. This term language (usually called *Gödel's T* after its inventor K. Gödel) may also be considered as a (rudimentary) functional programming language for higher type programs over the natural numbers. Gödel's T will turn out as fairly expressive though not Turing complete as not all total recursive functions can be implemented within Gödel's T. However, Gödel's T will have the advantage that it allows one to define only total functionals which terminate for all arguments. The higher type features of T will turn out as necessary for expressing "realizers" of propositions which can be formulated already in **HA**. These aspects will be studied *en detail* in the sections about *Kreisel's Modified Realizability* and *Gödel's Dialectica Interpretation.*
Before presenting $\mathbf{HA}^\omega$ in detail we want to remark that $\mathbf{HA}^\omega$ is *not higher order* as it does not allow one to quantify over predicates. That it allows one to quantify over functions is expressed by the phrase "higher type". This distinction might be confusing for the novice as in current mathematical practice subsets of a set A are identified with their characteristic functions, i.e. with functions from A to $\{0, 1\}$ and, therefore, may be identified with those functions from A to $\mathbb{N}$ whose image is contained in $\{0, 1\}$ and these latter are available within the ontology of $\mathbf{HA}^\omega$ just sketched. However, not for every predicate A on type σ there has to exist a corresponding characteristic function p satisfying $A(x) \leftrightarrow p(x)=0$. Usually one argues in favour of such a function p as follows.

> "Obviously, for every $x \in \sigma$ there exists an $n \in \{0, 1\}$ such that $n=0$ if and only if $A(x)$. Of course, this n is determined uniquely by x and, therefore, the relation $R(x, n) \equiv (n=0\leftrightarrow A(x)) \wedge (n=1\leftrightarrow\neg A(x))$ is single–valued and total. Then the function p with graph R is the characteristic function for the predicate $A(x)$."

This arguments sounds quite reasonable but makes the following two assumptions

(1) $A(x)\vee\neg A(x)$ for all $x \in \sigma$

(2) identification of functions and single–valued, total relations

which in general need not hold. Assumption (1) hinges on classical logic which we typically do not presuppose in our context. Assumption (2) implicitly postulates the *Axiom of Unique Choice*

$$\forall x{:}\sigma.\exists! y{:}\tau.R(x,y) \to \exists f{:}\sigma{\to}\tau.\forall x{:}\sigma.A(x,f(x))$$

which need not hold when we assume classical logic but want to restrict functions to computable or continuous ones (as is done typically in logics for reasoning about (functional) programs as e.g. the language LCF studied in Dana Scott's seminal paper [Sc69]).

12.1 Description of $\mathbf{HA}^\omega$

The set $\mathcal{T}$ of finite type symbols is defined inductively as follows

(1) $0 \in \mathcal{T}$ (type of natural numbers)

(2) if $\sigma, \tau \in \mathcal{T}$ then $\sigma{\times}\tau \in \mathcal{T}$ (formation of product types)

(3) if $\sigma, \tau \in \mathcal{T}$ then $\sigma{\to}\tau \in \mathcal{T}$ (formation of function types).

We use σ, τ, ρ possibly decorated with primes and subscripts as metavariables for type symbols.
The language of $\mathbf{HA}^\omega$ is a many–sorted language with variables $(x^\sigma, y^\sigma, z^\sigma, \ldots)$ for every type symbol $\sigma \in \mathcal{T}$. For every $\sigma \in \mathcal{T}$ there is a binary equality predicate $=_\sigma$ on σ. For all $\sigma, \tau \in \mathcal{T}$ there is an application operation $\mathsf{App}^{\sigma,\tau}$ from $\sigma{\to}\tau$ and σ to τ. Usually, we simply write $t(s)$ or even ts instead of $\mathsf{App}^{\sigma,\tau}(t,s)$ as the types σ and τ can be read off from the terms t and s, respectively. Moreover, application is left–associative in the sense that $t_1t_2\ldots t_n$ stands for $(\ldots(t_1t_2)\ldots t_n)$. Moreover, we often write $t(s_1,\ldots,s_n)$ instead of $ts_1\ldots s_n$ for sake of readability. If t is a term of type τ then $\lambda x^\sigma.\,t$ is a term of type $\sigma{\to}\tau$. We often write $\lambda x{:}\sigma.\,t$ instead of $\lambda x^\sigma.\,t$ and simply write x instead of x^σ if the variable x occurs within the scope of the binder $\lambda x{:}\sigma$. Furthermore, there are the following constants

$$\begin{array}{lll} 0 : 0 \qquad \mathsf{succ} : 0{\to}0 & \mathsf{R}^\sigma : \sigma{\to}(0{\to}\sigma{\to}\sigma){\to}0{\to}\sigma & \\ \mathsf{p}^{\sigma,\tau} : \sigma \to \tau \to \sigma{\times}\tau & \mathsf{pr}_0^{\sigma,\tau} : \sigma{\times}\tau \to \sigma & \mathsf{pr}_1^{\sigma,\tau} : \sigma{\times}\tau \to \tau \end{array}$$

listed together with their types.
For subsequent use we notice that for every type σ there is a distinguished closed term 0_σ of type σ. The distinguished terms are defined inductively as follows: $0_0 := 0$ and $0_{\sigma\times\tau} := \mathsf{p}(0_\sigma, 0_\tau)$ and $0_{\sigma\to\tau} := \lambda x{:}\sigma.0_\tau$.
The logical basis of $\mathbf{HA}^\omega$ is many–sorted intuitionistic predicate logic with equality where the replacement schema $A(t) \wedge t{=}s \to A(s)$ is restricted to those instances where t and s do not occur within the scope of a λ–abstraction, i.e. one must not replace equals under a λ. On top of this we postulate the following axioms

$\neg\, 0 = \mathsf{succ}(x)$

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

(P) $\mathsf{pr}_0(\mathsf{p}(x,y)) = x$, $\mathsf{pr}_1(\mathsf{p}(x,y)) = y$ and $\mathsf{p}(\mathsf{pr}_0(z),\mathsf{pr}_1(z)) = z$

(A) $\mathsf{R}xy0 = x$ and $\mathsf{R}xy(\mathsf{succ}(z)) = yz(\mathsf{R}xyz)$.

together with the *induction schema*

$$A(0) \to (\forall x.A(x) \to A(\mathsf{succ}(x))) \to \forall x.A(x)$$

for all predicates $A(x)$ in the language of $\mathbf{HA}^\omega$.
Notice, however, that the term language of $\mathbf{HA}^\omega$ is not first order due to λ–abstraction. Thus, the intricacies of bound variables occur already on the level of terms and are resolved the same way as for formulas, i.e. substitution has to be defined appropriately in order to avoid capture of free variables.
In the literature one often finds strengthenings of $\mathbf{HA}^\omega$ concerning equality.
The system $\mathbf{I}{-}\mathbf{HA}^\omega$ is obtained from $\mathbf{HA}^\omega$ by postulating for all $\sigma \in \mathcal{T}$ a function symbol $\mathsf{e}^\sigma : \sigma{\to}\sigma{\to}0$ satisfying the axioms

$$\mathsf{exy} \leq 1 \qquad \text{and} \qquad \mathsf{exy} = 0 \leftrightarrow \mathsf{x{=}y}$$

for all types σ. In other words e^σ is a decision function for equality on σ.
On the other hand $\mathbf{E}{-}\mathbf{HA}^\omega$ is obtained from $\mathbf{HA}^\omega$ by postulating the following extensionality axiom

$$(\forall z{:}\sigma.\ x(z) =_\tau y(z)) \leftrightarrow x =_{\sigma\to\tau} y$$

for all $\sigma, \tau \in \mathcal{T}$. Accordingly, the system $\mathbf{E}{-}\mathbf{HA}^\omega$ is often called *extensional* higher type arithmetic. Notice, that one may formulate $\mathbf{E}{-}\mathbf{HA}^\omega$ equivalently by stipulating an equality predicate just for base type 0 and defining it at compund types via the axioms

$$\begin{array}{l} x =_{\sigma\to\tau} y \leftrightarrow \forall z{:}\sigma.\ x(z) =_\tau y(z) \\ x =_{\sigma\times\tau} y \leftrightarrow \mathsf{pr}_0(x) =_\sigma \mathsf{pr}_0(y) \wedge \mathsf{pr}_1(x) =_\tau \mathsf{pr}_1(y) \end{array}$$

Notice that $\mathbf{E}{-}\mathbf{HA}^\omega$ validates the principle

$$(\zeta) \qquad (\forall x{:}\sigma.t{=}s) \to \lambda x{:}\sigma.\, t{=}\lambda x{:}\sigma.\, s \quad .$$

The system $\mathbf{I}{-}\mathbf{HA}^\omega$ is often called *intensional* higher type arithmetic as the existence of computable functionals deciding equality of function types is in conflict[29] with extensionality.

[29]Actually, extensionality is inconsistent with decidable equality in presence of continuity or effectivity axioms.

12.2 Models of $\mathbf{HA}^\omega$

Now we will construct some classical models for $\mathbf{HA}^\omega$. First of all there exists the usual set theoretic model M with

$$M^0 = \mathbb{N} \qquad M^{\sigma\times\tau} = M^\sigma \times M^\tau \qquad M^{\sigma\to\tau} = M^\sigma \to M^\tau$$

where the constants receive their obvious interpretations. In particular $\mathsf{App}(f,a) = f(a)$, i.e. App is interpreted as ordinary function application. This model has the disadvantage that it contains a lot of non–computable objects. This defect is avoided by the following models.

Hereditary Recursive Operations

By structural recursion over $\mathcal{T}$ we define

$$\begin{array}{lll} \mathrm{HRO}_0 & := & \mathbb{N} \\ \mathrm{HRO}_{\sigma\times\tau} & := & \{e \in \mathbb{N} \mid \mathsf{pr}_0(e) \in \mathrm{HRO}_\sigma \wedge \mathsf{pr}_1(\mathrm{e}) \in \mathrm{HRO}_\tau\} \\ \mathrm{HRO}_{\sigma\to\tau} & := & \{e \in \mathbb{N} \mid \forall n \in \mathrm{HRO}_\sigma. \{\mathrm{e}\}(\mathrm{n}) \in \mathrm{HRO}_\tau\} \end{array}$$

App is interpreted as partial recursive application which appears as total due to the definition of function types. The other codes get their obvious meaning where R is interpreted by some Gödel number for the primitive recursor.
This way HRO gets a model for $\mathbf{I}{-}\mathbf{HA}^\omega$ when interpreting e as the code of some algorithm deciding equality of natural numbers. Moreover, the HRO–model validates Church's Thesis claiming that all functions of type $0{\to}0$ are recursive.

Hereditary Effective Operations

By structural recursion we define a *partial equivalence relation* $\sim_\sigma$ on $\mathbb{N}$, i.e. a symmetric and transitive but not necessarily reflexive binary relation on $\mathbb{N}$, for every $\sigma \in \mathcal{T}$ via the following clauses

$$\begin{array}{lll} x \sim_0 y & := & x{=}y \\ x \sim_{\sigma\times\tau} y & := & (\mathsf{pr}_0(x) \sim_\sigma \mathsf{pr}_0(y)) \wedge (\mathsf{pr}_1(x) \sim_\tau \mathsf{pr}_1(y)) \\ x \sim_{\sigma\to\tau} y & := & \forall z, z'.\ z \sim_\sigma z' \to \{x\}(z) \sim_\tau \{y\}(z') \end{array}$$

and we define[30]

$$\mathrm{HEO}_\sigma := \mathbb{N}/{\sim_\sigma}$$

for $\sigma \in \mathcal{T}$.
Now function application is defined as $\mathsf{App}([e],[n]) = [\{e\}(n)]$ and the constants of $\mathbf{HA}^\omega$ get their obvious meaning as in the case of HRO.

[30]If R is a partial equivalence relation on $\mathbb{N}$ then we understand $\mathbb{N}/R$ as a shorthand for $\{n{\in}\mathbb{N}|nRn\}/R$ as R is an equivalence relation in the proper sense on the carrier set $\{n{\in}\mathbb{N}|nRn\}$ of R.

Notice that HEO is a model of $\mathbf{E}{-}\mathbf{HA}^\omega$ due to the way how $\sim$ is defined for function types. But HEO does not model $\mathbf{I}{-}\mathbf{HA}^\omega$ as the relations $\sim$ are not decidable for function types.

Continuous Function Models

Notice that there are also variants of HRO and HEO based on Baire space $\mathcal{B}$ and continuous function application as introduced in Section 11. For example one may define in analogy to HRO a hierarchy ICF of *Intensional Continuous Functionals* in the following way

$$\begin{array}{lll}
\mathrm{ICF}_0 & := & \mathbb{N} \\
\mathrm{ICF}_{0\to 0} & := & \mathcal{B} \\
\mathrm{ICF}_{0\to\sigma} & := & \{\alpha \in \mathcal{B} \mid \forall x.\, (\lambda n.\alpha(\langle x\rangle * n) \in \mathrm{ICF}_\sigma)\} \qquad (\sigma \not\equiv 0) \\
\mathrm{ICF}_{\sigma\to 0} & := & \{\alpha \in \mathcal{B} \mid \forall \beta \in \mathrm{ICF}_\sigma.\, \alpha(\beta){\downarrow}\} \qquad (\sigma \not\equiv 0) \\
\mathrm{ICF}_{\sigma\to\tau} & := & \{\alpha \in \mathcal{B} \mid \forall \beta \in \mathrm{ICF}_\sigma.\, \alpha|\beta \in \mathrm{ICF}_\tau\} \qquad (\sigma, \tau \not\equiv 0)
\end{array}$$

which can be organized into a model of $\mathbf{I}{-}\mathbf{HA}^\omega$. In analogy to HEO one may define a hierarchy ECF of *Extensional Continuous Functionals* which can be organized into a model of $\mathbf{E}{-}\mathbf{HA}^\omega$.

13 Kreisel's Modified Realizability

The basic idea of Kreisel's Modified Realizability as introduced by him in 1959 is to replace Gödel numbers by terms of Gödel's system T. An obvious advantage of this change is that realizers extracted from proofs are already programs (of Gödel's T) and, therefore, much easier to grasp and handle than the somewhat akward Gödel numbers. Moreover, such a method works for arbitrary models of Gödel's T. Besides these simplifications other *a priori* unexpected benefits are that the modified realizability interpretation does not validate Markov's Principle but instead the principle of *Independence* of existence–free *Premiss*

$$\mathrm{IP}_{\mathrm{ef}} \qquad (A \to \exists x{:}\sigma.\, B(x)) \to \exists x{:}\sigma.\, (A \to B(x)) \qquad A\ \exists\text{–free},\ x \notin FV(A)$$

where a formula is $\exists$–free if it does not contain occurrences of $\exists$ and $\vee$. This together with the *axiom of choice for arbitrary types*

$$\mathrm{AC} \qquad \forall x{:}\sigma.\exists y{:}\tau.A(x,y) \to \exists z{:}\sigma{\to}\tau.\forall x{:}\sigma.A(x, z(x))$$

will give rise to a very convenient axiomatisation of modified realizability.

Now we give the precise definition of modified realizability. We consider $A \vee B$ as an abbreviation (or 'macro') for $\exists n{:}0.\ (n{=}0 \to A) \wedge (n{\neq}0 \to B)$ and, therefore, can omit the clause for disjunction.

Definition 13.1 (Modified Realizability)
With every formula A of $\mathbf{HA}^\omega$ *there is asscoiated a type* $\mathsf{tp}(A)$ *of* potential realizers *via the following clauses*

$$\begin{aligned}
\mathsf{tp}(t{=}s) &= 0\\
\mathsf{tp}(A{\wedge}B) &= \mathsf{tp}(A) \times \mathsf{tp}(B)\\
\mathsf{tp}(A{\to}B) &= \mathsf{tp}(A) \to \mathsf{tp}(B)\\
\mathsf{tp}(\forall z{:}\sigma.A) &= \sigma \to \mathsf{tp}(A)\\
\mathsf{tp}(\exists z{:}\sigma.A) &= \sigma \times \mathsf{tp}(A) \quad .
\end{aligned}$$

The modified realizability relation $x\,\mathbf{mr}\,A$ *with* x *of type* $\mathsf{tp}(A)$ *is defined by induction on the structure of* A *via the following clauses*

$$\begin{aligned}
x\,\mathbf{mr}\,t{=}s &\equiv t{=}s\\
x\,\mathbf{mr}\,A \wedge B &\equiv \mathsf{pr}_0(x)\,\mathbf{mr}\,A \ \wedge\ \mathsf{pr}_1(x)\,\mathbf{mr}\,B\\
x\,\mathbf{mr}\,A \to B &\equiv \forall y{:}\mathsf{tp}(A).\ y\,\mathbf{mr}\,A \to x(y)\,\mathbf{mr}\,B\\
x\,\mathbf{mr}\,\forall z{:}\sigma.\,A(z) &\equiv \forall z{:}\sigma.\,x(z)\,\mathbf{mr}\,A(z)\\
x\,\mathbf{mr}\,\exists z{:}\sigma.\,A(z) &\equiv \mathsf{pr}_1(x)\,\mathbf{mr}\,A(\mathsf{pr}_0(x)) \quad .
\end{aligned}$$

The predicate $x\,\mathbf{mr}\,A$ *specifies the* actual realizers *of* A *within the type* $\mathsf{tp}(A)$ *of potential realizers of* A. $\diamondsuit$

Notice that the main difference between modified realizability and Kleene's number realizability (besides the nature of the realizers) is that an actual realizer of $A{\to}B$ has to send potential realizers of A to potential realizers of B besides sending actual realizers of A to actual realizers of B.
To get familiar with the notion and for subsequent use let us consider what it means that $x\,\mathbf{mr}\,\neg A$. As $\neg A \equiv A \to \bot$ we have $x\,\mathbf{mr}\,\neg A \equiv \forall y{:}\mathsf{tp}(A).\,y\,\mathbf{mr}\,A \to x(y)\,\mathbf{mr}\,\bot \equiv \forall y{:}\mathsf{tp}(A).\,y\,\mathbf{mr}\,A \to \bot \equiv \forall y{:}\mathsf{tp}(A).\,\neg\,y\,\mathbf{mr}\,A$. Thus, $y\,\mathbf{mr}\,A$ is logically equivalent to $\neg\exists y.\,y\,\mathbf{mr}\,A$.

Next we prove a Soundness Theorem for Modified Realizability.

Theorem 13.1 (Soundness for Modified Realizability)
If $\mathbf{HA}^\omega$ *proves* A *then there is a term* t *of type* $\mathsf{tp}(A)$ *such that* $\mathbf{HA}^\omega$ *proves* $t\,\mathbf{mr}\,A$.

Proof: We prove more generally that for every sequence $\Gamma \vdash A$ derivable in $\mathbf{HA}^\omega$ there is a term t of type $\mathsf{tp}(A)$ such that

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

is derivable in $\mathbf{HA}^\omega$ where $\vec{u}\,\mathbf{mr}\,\Gamma$ stands for $u_1\,\mathbf{mr}\,A_1\wedge\ldots\wedge u_n\,\mathbf{mr}\,A_n$. Moreover, the free variables of t are contained in $FV(\Gamma, A)\cup\{u_1,\ldots,u_n\}$. This generalised claim is proved by induction on the structure of derivations in $\mathbf{HA}^\omega$.
The cases of structural rules are trivial and, therefore, left to the reader.
($\wedge I$) If $\mathbf{HA}^\omega$ proves $\vec{u}\,\mathbf{mr}\,\Gamma \vdash t\,\mathbf{mr}\,A$ and $\vec{u}\,\mathbf{mr}\,\Gamma \vdash s\,\mathbf{mr}\,B$ then $\vec{u}\,\mathbf{mr}\,\Gamma \vdash \mathsf{p}(t,s)\,\mathbf{mr}\,A\wedge B$.
($\wedge R$) If $\mathbf{HA}^\omega$ proves $\vec{u}\,\mathbf{mr}\,\Gamma \vdash t\,\mathbf{mr}\,A\wedge B$ then $\mathbf{HA}^\omega$ proves $\vec{u}\,\mathbf{mr}\,\Gamma \vdash \mathsf{pr}_0(t)\,\mathbf{mr}\,A$ and $\vec{u}\,\mathbf{mr}\,\Gamma \vdash \mathsf{pr}_1(t)\,\mathbf{mr}\,B$.
($\to I$) If $\mathbf{HA}^\omega$ proves $\vec{u}\,\mathbf{mr}\,\Gamma \wedge u\,\mathbf{mr}\,A \vdash t\,\mathbf{mr}\,B$ then $\mathbf{HA}^\omega$ proves $\vec{u}\,\mathbf{mr}\,\Gamma \vdash \lambda u.t\,\mathbf{mr}\,A\to B$.
($\to E$) If $\mathbf{HA}^\omega$ proves $\vec{u}\,\mathbf{mr}\,\Gamma \vdash t\,\mathbf{mr}\,A\to B$ and $\vec{u}\,\mathbf{mr}\,\Gamma \vdash s\,\mathbf{mr}\,A$ then $\mathbf{HA}^\omega$ proves $\vec{u}\,\mathbf{mr}\,\Gamma \vdash t(s)\,\mathbf{mr}\,B$.
($\bot E$) Suppose that $\mathbf{HA}^\omega$ proves $\vec{u}\,\mathbf{mr}\,\Gamma \vdash t\,\mathbf{mr}\,\bot$. Then as $t\,\mathbf{mr}\,\bot \equiv \bot$ it follows that $\mathbf{HA}^\omega$ proves $\vec{u}\,\mathbf{mr}\,\Gamma \vdash \bot$ and, therefore, also $\vec{u}\,\mathbf{mr}\,\Gamma \vdash 0_{\mathsf{tp}(A)}\,\mathbf{mr}\,A$.
($\forall I$) If $\mathbf{HA}^\omega$ proves $\vec{u}\,\mathbf{mr}\,\Gamma \vdash t\,\mathbf{mr}\,A(x)$ and x does not occur in Γ then $\mathbf{HA}^\omega$ proves $\vec{u}\,\mathbf{mr}\,\Gamma \vdash \lambda x.t\,\mathbf{mr}\,\forall x.A(x)$.
($\forall E$) If $\mathbf{HA}^\omega$ proves $\vec{u}\,\mathbf{mr}\,\Gamma \vdash t\,\mathbf{mr}\,\forall x{:}\sigma.A(x)$ then it follows that $\mathbf{HA}^\omega$ proves $\vec{u}\,\mathbf{mr}\,\Gamma \vdash t(s)\,\mathbf{mr}\,A(s)$ for all terms s of type σ.
($\exists I$) If $\mathbf{HA}^\omega$ proves $\vec{u}\,\mathbf{mr}\,\Gamma \vdash t\,\mathbf{mr}\,A(s)$ then it follows that $\mathbf{HA}^\omega$ proves $\vec{u}\,\mathbf{mr}\,\Gamma \vdash \mathsf{p}(s,t)\,\mathbf{mr}\,\exists x.A(x)$ as $\mathsf{p}(s,t)\,\mathbf{mr}\,\exists x.A(x) \equiv \mathsf{pr}_1(\mathsf{p}(s,t))\,\mathbf{mr}\,A(\mathsf{pr}_0(\mathsf{p}(s,t)))$ and $\mathbf{HA}^\omega$ proves $\mathsf{pr}_0(\mathsf{p}(s,t)) = s$ and $\mathsf{pr}_1(\mathsf{p}(s,t)) = t$.
($\exists E$) Suppose as induction hypotheses that $\mathbf{HA}^\omega$ proves $\vec{u}\,\mathbf{mr}\,\Gamma \vdash t\,\mathbf{mr}\,\exists x.A(x)$ and $\vec{u}\,\mathbf{mr}\,\Gamma \wedge v\,\mathbf{mr}\,A(x) \vdash s\,\mathbf{mr}\,B$ where x is not free in Γ or B. Then as by definition $t\,\mathbf{mr}\,\exists x.A(x) \equiv \mathsf{pr}_1(t)\,\mathbf{mr}\,A(\mathsf{pr}_0(t))$ it follows from the first induction hypothesis that

$$(1)\qquad \vec{u}\,\mathbf{mr}\,\Gamma \vdash \mathsf{pr}_1(t)\,\mathbf{mr}\,A(\mathsf{pr}_0(t))$$

can be proved in $\mathbf{HA}^\omega$. From the second induction hypothesis it follows that

$$(2)\qquad \vec{u}\,\mathbf{mr}\,\Gamma \wedge \mathsf{pr}_1(t)\,\mathbf{mr}\,A(\mathsf{pr}_0(t)) \vdash s[\mathsf{pr}_0(t),\mathsf{pr}_1(t)/x,v]\,\mathbf{mr}\,B$$

can be proved in $\mathbf{HA}^\omega$. As (1) and (2) can be proved in $\mathbf{HA}^\omega$ it follows by purely logical reasoning that $\vec{u}\,\mathbf{mr}\,\Gamma \vdash s[\mathsf{pr}_0(t),\mathsf{pr}_1(t)/x,v]\,\mathbf{mr}\,B$ can be proved in $\mathbf{HA}^\omega$.
(Axioms) For very equational axiom $t{=}s$ of $\mathbf{HA}^\omega$ it proves $0\,\mathbf{mr}\,t{=}s$. Furthermore, $\mathbf{HA}^\omega$ proves that $\lambda x{:}0.0\,\mathbf{mr}\,\neg\,0{=}\mathsf{succ}(x)$. Let $A(0) \to (\forall x.A(x) \to A(\mathsf{succ}(x))) \to \forall x.A(x)$ be an instance of the induction schema. Then $\mathbf{HA}^\omega$ proves

$$\mathsf{R}^{\mathsf{tp}(A)}\,\mathbf{mr}\,A(0) \to (\forall x.A(x) \to A(\mathsf{succ}(x))) \to \forall x.A(x)$$

as from $u\,\mathbf{mr}\,A(0)$ and $\forall x{:}0.\forall u{:}\mathsf{tp}(A).\,u\,\mathbf{mr}\,A(x) \to v(x)(u)\,\mathbf{mr}\,A(\mathsf{succ}(x)))$ one can derive via the induction principle of $\mathbf{HA}^\omega$ that $\forall x{:}0.\ \mathsf{R}uvx\,\mathbf{mr}\,A(x)$. □

Next we are heading for an axiomatization of Modified Realizability for which we need the following notion.

Definition 13.2 *A formula A of* $\mathbf{HA}^\omega$ *is* $\exists$–free *iff it does not contain occurrences of* $\exists$ *and, therefore, also not of* $\vee$. $\Diamond$

Notice that all formulas of the form $x \,\mathbf{mr}\, A$ are $\exists$–free.

Lemma 13.1 *For every $\exists$-free formula A of* $\mathbf{HA}^\omega$

(1) $(\exists x{:}\mathsf{tp}(A).\, x \,\mathbf{mr}\, A) \to A$ *is provable in* $\mathbf{HA}^\omega$ *and*

(2) $A \to 0_{\mathsf{tp}(A)} \,\mathbf{mr}\, A$ *is provable in* $\mathbf{HA}^\omega$.

Proof: We prove (1) and (2) simultaneously by induction on the structure of A. Notice also that condition (1) for A is logically equivalent to $x \,\mathbf{mr}\, A \to A$.
If A is an atomic formula then (1) and (2) are obvious as $\exists x.\, x \,\mathbf{mr}\, A$ and $0 \,\mathbf{mr}\, A$ are provably equivalent to A itself.
($\wedge$) In $\mathbf{HA}^\omega$ one can show that $\exists x.\, x \,\mathbf{mr}\, A{\wedge}B$ is equivalent to $\exists x.\, x \,\mathbf{mr}\, A \wedge \exists x.\, x \,\mathbf{mr}\, B$. As by induction hypothesis $\mathbf{HA}^\omega$ proves $\exists x.\, x \,\mathbf{mr}\, A \to A$ and $\exists x.\, x \,\mathbf{mr}\, B \to B$ it follows that $\exists x.\, x \,\mathbf{mr}\, A{\wedge}B \to A{\wedge}B$, i.e. that $A{\wedge}B$ satisfies (1). As by induction hypothesis $\mathbf{HA}^\omega$ proves $A \to 0_{\mathsf{tp}(A)} \,\mathbf{mr}\, A$ and $B \to 0_{\mathsf{tp}(B)} \,\mathbf{mr}\, B$ it follows that $\mathbf{HA}^\omega$ proves $A{\wedge}B \to \mathsf{p}(0_{\mathsf{tp}(A)}, 0_{\mathsf{tp}(B)}) \,\mathbf{mr}\, A{\wedge}B$. As $0_{\mathsf{tp}(A\wedge B)} \equiv \mathsf{p}(0_{\mathsf{tp}(A)}, 0_{\mathsf{tp}(B)})$ we have that $\mathbf{HA}^\omega$ proves $A{\wedge}B \to 0_{\mathsf{tp}(A\wedge B)} \,\mathbf{mr}\, A{\wedge}B$, i.e. that $A{\wedge}B$ satisfies (2).
($\to$) Suppose as induction hypotheses that A and B satisfy the conditions (1) and (2). As by definition $x \,\mathbf{mr}\, A{\to}B \equiv \forall y.\, y \,\mathbf{mr}\, A \to x(y) \,\mathbf{mr}\, B$ and $\mathbf{HA}^\omega$ proves $A \to 0_{\mathsf{tp}(A)} \,\mathbf{mr}\, A$ by induction hypothesis it follows that $\mathbf{HA}^\omega$ proves also $x \,\mathbf{mr}\, A{\to}B \to A \to x(0_{\mathsf{tp}(A)}) \,\mathbf{mr}\, B$ and, therefore, also $x \,\mathbf{mr}\, A{\to}B \to A \to B$ as by induction hypothesis $\mathbf{HA}^\omega$ proves $x(0_{\mathsf{tp}(A)}) \,\mathbf{mr}\, B \to B$. Thus, in $\mathbf{HA}^\omega$ one can prove $\exists x.\, x \,\mathbf{mr}\, A{\to}B \to A{\to}B$, i.e. $A{\to}B$ satisfies (1). As by definition $0_{\mathsf{tp}(A\to B)} \equiv \lambda x{:}\mathsf{tp}(A).\, 0_{\mathsf{tp}(B)}$ $\mathbf{HA}^\omega$ proves that $0_{\mathsf{tp}(A\to B)} \,\mathbf{mr}\, A{\to}B$ is equivalent to $\forall x{:}\mathsf{tp}(A).\, (x \,\mathbf{mr}\, A \to 0_{\mathsf{tp}(B)} \,\mathbf{mr}\, B)$ and hence also to $\exists x{:}\mathsf{tp}(A).\, x \,\mathbf{mr}\, A \to 0_{\mathsf{tp}(B)} \,\mathbf{mr}\, B$. As by induction hypothesis $\mathbf{HA}^\omega$ proves $\exists x.\, x \,\mathbf{mr}\, A \to A$ and $B \to 0_{\mathsf{tp}(B)} \,\mathbf{mr}\, B$ it follows that $\mathbf{HA}^\omega$ proves $(A{\to}B) \to (\exists x{:}\mathsf{tp}(A).\, x \,\mathbf{mr}\, A) \to 0_{\mathsf{tp}(B)} \,\mathbf{mr}\, B$ hence $(A{\to}B) \to 0_{\mathsf{tp}(A\to B)} \,\mathbf{mr}\, A{\to}B$. Thus $A{\to}B$ satisfies (2) as desired.
($\forall$) Assume as induction hypothesis that $A(z)$ satisfies (1) and (2). By definition $x \,\mathbf{mr}\, \forall z{:}\sigma.A(z) \equiv \forall z{:}\sigma.\, x(z) \,\mathbf{mr}\, A(z)$. As by induction hypothesis $\mathbf{HA}^\omega$ proves $x(z) \,\mathbf{mr}\, A(z) \to A(z)$ it follows that $\mathbf{HA}^\omega$ proves $x \,\mathbf{mr}\, \forall z{:}\sigma.A(z) \to \forall z{:}\sigma.A(z)$ from which it follows immediately that $\forall z{:}\sigma.A(z)$ satisfies (1). By induction hypothesis $\mathbf{HA}^\omega$ proves $A(z) \to 0_{\mathsf{tp}(A(z))} \,\mathbf{mr}\, A(z)$ and hence also $\forall z.A(z) \to \forall z.0_{\mathsf{tp}(A(z))} \,\mathbf{mr}\, A(z)$. Thus, as by definition $0_{\forall z:\sigma.A(z)} \equiv \lambda z{:}\sigma.0_{\mathsf{tp}(A(z))}$ it follows that $\mathbf{HA}^\omega$ proves $\forall z.A(z) \to 0_{\forall z:\sigma.A(z)} \,\mathbf{mr}\, \forall z.A(z)$. Thus $\forall z.A(z)$ satisfies (2). $\Box$

Corollary 13.1 *For every formula A of* $\mathbf{HA}^\omega$

(1) $\exists x.\, x\,\mathbf{mr}\,(\exists y.\, y\,\mathbf{mr}\,A) \leftrightarrow \exists y.\, y\,\mathbf{mr}\,A$ *is provable in* $\mathbf{HA}^\omega$

(2) $A \leftrightarrow \exists x.\, x\,\mathbf{mr}\,A$ *is provable in* $\mathbf{HA}^\omega$ *iff* $\mathbf{HA}^\omega$ *proves that* A *is equivalent to an existential quantification of an* $\exists$*–free formula in* $\mathbf{HA}^\omega$*.*

Proof: (1) By definition $x\,\mathbf{mr}\,(\exists y.\, y\,\mathbf{mr}\,A) \equiv \mathsf{pr}_1(x)\,\mathbf{mr}\,(\mathsf{pr}_0(x)\,\mathbf{mr}\,A)$. As the formula $\mathsf{pr}_0(x)\,\mathbf{mr}\,A$ is $\exists$–free it follows from Lemma 13.1(1) that $\mathbf{HA}^\omega$ proves $\mathsf{pr}_1(x)\,\mathbf{mr}\,(\mathsf{pr}_0(x)\,\mathbf{mr}\,A) \to \mathsf{pr}_0(x)\,\mathbf{mr}\,A$ and, therefore, also $x\,\mathbf{mr}\,(\exists y.\, y\,\mathbf{mr}\,A) \to \exists y.\, y\,\mathbf{mr}\,A$. Thus, $\mathbf{HA}^\omega$ proves $\exists x.\, x\,\mathbf{mr}\,(\exists y.\, y\,\mathbf{mr}\,A) \to \exists y.\, y\,\mathbf{mr}\,A$. As $y\,\mathbf{mr}\,A$ is $\exists$–free it follows by Lemma 13.1(2) that $\mathbf{HA}^\omega$ proves $y\,\mathbf{mr}\,A \to 0_{y\,\mathbf{mr}\,A}\,\mathbf{mr}\,(y\,\mathbf{mr}\,A)$ and, therefore, also $y\,\mathbf{mr}\,A \to \mathsf{p}(y, 0_{y\,\mathbf{mr}\,A})\,\mathbf{mr}\,(\exists y.\, y\,\mathbf{mr}\,A)$. Thus, it follows that $\mathbf{HA}^\omega$ proves $y\,\mathbf{mr}\,A \to \exists x.\, x\,\mathbf{mr}\,(\exists y.\, y\,\mathbf{mr}\,A)$ and, therefore, also that $\mathbf{HA}^\omega$ proves $\exists y.\, y\,\mathbf{mr}\,A \to \exists x.\, x\,\mathbf{mr}\,(\exists y.\, y\,\mathbf{mr}\,A)$.
(2) The implication from left to right is immediate as $x\,\mathbf{mr}\,A$ is $\exists$–free. For the reverse direction first observe that it follows (almost) immediately from the Soundness Theorem 13.1 that if $\mathbf{HA}^\omega$ proves $A_1 \leftrightarrow A_2$ then $\mathbf{HA}^\omega$ proves also $\exists x.\, x\,\mathbf{mr}\,A_1 \leftrightarrow \exists x.\, x\,\mathbf{mr}\,A_2$. Thus, it suffices to show for $\exists$–free formulas $B(y)$ that $\mathbf{HA}^\omega$ proves $\exists y.B(y) \leftrightarrow \exists x.\, x\,\mathbf{mr}\,(\exists y.B(y))$.
From Lemma 13.1(2) it follows that $\mathbf{HA}^\omega$ proves $B(y) \to 0_{\mathsf{tp}(B(y))}\,\mathbf{mr}\,B(y)$ and, therefore, also $B(y) \to \mathsf{p}(y, 0_{\mathsf{tp}(B(y))})\,\mathbf{mr}\,\exists y.B(y)$. Thus, $\mathbf{HA}^\omega$ proves $B(y) \to \exists x.\, x\,\mathbf{mr}\,\exists y.B(y)$ and hence also $\exists y.B(y) \to \exists x.\, x\,\mathbf{mr}\,\exists y.B(y)$.
On the other hand by definition $x\,\mathbf{mr}\,\exists y.B(y) \equiv \mathsf{pr}_1(x)\,\mathbf{mr}\,B(\mathsf{pr}_0(x))$ and from Lemma 13.1(1) it follows that $\mathbf{HA}^\omega$ proves $\mathsf{pr}_1(x)\,\mathbf{mr}\,B(\mathsf{pr}_0(x)) \to B(\mathsf{pr}_0(x))$ as B is $\exists$–free by assumption. Thus, $\mathbf{HA}^\omega$ proves $\mathsf{pr}_1(x)\,\mathbf{mr}\,B(\mathsf{pr}_0(x)) \to \exists y.B(y)$ and, therefore, also $x\,\mathbf{mr}\,\exists y.B(y) \to \exists y.B(y)$. Hence $\mathbf{HA}^\omega$ proves $\exists x.\, x\,\mathbf{mr}\,\exists y.B(y) \to \exists y.B(y)$. $\square$

Theorem 13.2 (Axiomatization of Modified Realizability)
For arbitrary formulas A *of* $\mathbf{HA}^\omega$

(1) $\mathbf{HA}^\omega + \mathrm{AC} + \mathrm{IP_{ef}} \vdash \mathrm{A} \leftrightarrow \exists \mathrm{x}.\, \mathrm{x}\,\mathbf{mr}\,\mathrm{A}$

(2) $\mathbf{HA}^\omega + \mathrm{AC} + \mathrm{IP_{ef}} \vdash \mathrm{A}$ *iff* $\mathbf{HA}^\omega \vdash t\,\mathbf{mr}\,A$ *for some term* t *of type* $\mathsf{tp}(A)$*.*

Proof: Claim (1) is proved by induction on the structure of A.
For basic formulas claim (1) is obviously true.
($\wedge$) This case is trivial as already in $\mathbf{HA}^\omega$ $\exists x.\, x\,\mathbf{mr}\,A \wedge \exists x.\, x\,\mathbf{mr}\,B$ is provably equivalent to $\exists x.\, x\,\mathbf{mr}\,A{\wedge}B$.
($\to$) Assume as induction hypothesis that $\mathbf{HA}^\omega + \mathrm{AC} + \mathrm{IP_{ef}}$ proves that $A \leftrightarrow \exists y.\, y\,\mathbf{mr}\,A$ and $B \leftrightarrow \exists z.\, z\,\mathbf{mr}\,B$. Thus, by AC and $\mathrm{IP_{ef}}$ (as $x\,\mathbf{mr}\,A$ is $\exists$–free) one can show that $A{\to}B$ is equivalent to $\exists x.\forall y.\, y\,\mathbf{mr}\,A \to x(y)\,\mathbf{mr}\,B$, i.e. $\exists x.\, x\,\mathbf{mr}\,A{\to}B$.

($\forall$) Assume as induction hypothesis that $\mathbf{HA}^\omega + \mathrm{AC} + \mathrm{IP}_{\mathrm{ef}}$ proves that $A(z) \leftrightarrow \exists y.\, y \,\mathbf{mr}\, A(z)$. Thus, using AC one can show that $\mathbf{HA}^\omega + \mathrm{AC} + \mathrm{IP}_{\mathrm{ef}}$ proves the equivalence of $\forall z.\, A(z)$ and $\exists x. \forall z.\, x(z) \,\mathbf{mr}\, A(z)$, i.e. $\exists x.\, x \,\mathbf{mr}\, \forall z.\, A(z)$.
($\exists$) Suppose as induction hypothesis that (1) holds for $A(z)$. We have to show that (1) also holds for $\exists z{:}\sigma.A(z)$.
Obviously, one can derive already in $\mathbf{HA}^\omega$ that $x \,\mathbf{mr}\, A(z) \to \mathsf{p}(x,z) \,\mathbf{mr}\, \exists z{:}\sigma.A(z)$ from which it follows already in constructive predicate logic that $x \,\mathbf{mr}\, A(z) \to \exists x.\, x \,\mathbf{mr}\, \exists z{:}\sigma.A(z)$ and hence that $\exists x.\, x \,\mathbf{mr}\, A(z) \to \exists x.\, x \,\mathbf{mr}\, \exists z{:}\sigma.A(z)$. But as by assumption we have that $\mathbf{HA}^\omega + \mathrm{AC} + \mathrm{IP}_{\mathrm{ef}} \vdash \mathrm{A(z)} \to \exists \mathrm{x.\ x} \,\mathbf{mr}\, \mathrm{A(z)}$ it follows that $\mathbf{HA}^\omega + \mathrm{AC} + \mathrm{IP}_{\mathrm{ef}} \vdash \mathrm{A(z)} \to \exists \mathrm{x.\, x} \,\mathbf{mr}\, \exists \mathrm{z}{:}\sigma.\mathrm{A(z)}$ and, therefore, also $\mathbf{HA}^\omega + \mathrm{AC} + \mathrm{IP}_{\mathrm{ef}} \vdash \exists \mathrm{z.A(z)} \to \exists \mathrm{x.\, x} \,\mathbf{mr}\, \exists \mathrm{z}{:}\sigma.\mathrm{A(z)}$.
By induction hypothesis one can derive in $\mathbf{HA}^\omega + \mathrm{AC} + \mathrm{IP}_{\mathrm{ef}}$ that $y \,\mathbf{mr}\, A(z) \to A(z)$ for $y \notin FV(A)$ and, therefore, also that $y \,\mathbf{mr}\, A(z) \to \exists z.A(z)$. Substituting $\mathsf{pr}_0(x)$ and $\mathsf{pr}_1(x)$ for z and y, respectively, we get that $\mathbf{HA}^\omega + \mathrm{AC} + \mathrm{IP}_{\mathrm{ef}}$ proves $\mathsf{pr}_1(x) \,\mathbf{mr}\, A(\mathsf{pr}_0(x)) \to \exists z.A(z)$ and, therefore, also $x \,\mathbf{mr}\, \exists z.A(z) \to \exists z.A(z)$, i.e. $\exists x.\, x \,\mathbf{mr}\, \exists z.A(z) \to \exists z.A(z)$.

(2) If $\mathbf{HA}^\omega \vdash t \,\mathbf{mr}\, A$ then $\mathbf{HA}^\omega$ and, therefore, also $\mathbf{HA}^\omega + \mathrm{AC} + \mathrm{IP}_{\mathrm{ef}}$ prove $\exists x.\, x \,\mathbf{mr}\, A$. From this it follows by the already established claim (1) that $\mathbf{HA}^\omega + \mathrm{AC} + \mathrm{IP}_{\mathrm{ef}} \vdash \mathrm{A}$.
For the reverse direction first observe (exercise!) that one can construct for each instance C of AC and $\mathrm{IP}_{\mathrm{ef}}$ a term t such that $\mathbf{HA}^\omega \vdash t \,\mathbf{mr}\, C$. If $\mathbf{HA}^\omega{+}\mathrm{AC}{+}\mathrm{IP}_{\mathrm{ef}} \vdash \mathrm{A}$ then there exist instances $C_1, \ldots, C_n$ of AC or $\mathrm{IP}_{\mathrm{ef}}$ such that $\mathbf{HA}^\omega \vdash C_1 \to \ldots \to C_k \to A$. By the Soundness Theorem for Modified Realizability there is a term t with $\mathbf{HA}^\omega \vdash t \,\mathbf{mr}\, C_1 {\to} \ldots {\to} C_k {\to} A$. Let t_i be a term with $\mathbf{HA}^\omega \vdash t_i \,\mathbf{mr}\, C_i$ for $i = 1, \ldots, k$. Then obviously $\mathbf{HA}^\omega \vdash t(t_1)\ldots(t_k) \,\mathbf{mr}\, A$ as desired. $\square$

From this characterization it follows that $\mathbf{HA}^\omega + \mathrm{AC} + \mathrm{IP}_{\mathrm{ef}}$ actually proves the following strengthening of $\mathrm{IP}_{\mathrm{ef}}$

$$\mathrm{IP} \qquad (\neg A \to \exists y.B(y)) \to \exists y.(\neg A \to B(y)) \qquad y \notin FV(A)$$

where A is an *arbitrary* formula.

Corollary 13.2 *In* $\mathbf{HA}^\omega + \mathrm{AC} + \mathrm{IP}_{\mathrm{ef}}$ *one can derive all instances of* IP.

Proof: Let A and $B(y)$ be arbitrary formulas of $\mathbf{HA}^\omega$ with $y \notin FV(A)$. By Theorem 13.2 in $\mathbf{HA}^\omega + \mathrm{AC} + \mathrm{IP}_{\mathrm{ef}}$ one can prove the equivalence of A and $\exists x.\, x \,\mathbf{mr}\, A$ and hence also the equivalence of $\neg A$ and $\forall x. \neg\, x \,\mathbf{mr}\, A$. Thus, for showing that $\mathbf{HA}^\omega + \mathrm{AC} + \mathrm{IP}_{\mathrm{ef}}$ proves $(\neg A \to \exists y.B(y)) \to \exists y.(\neg A \to B(y))$ it suffices to show that $\mathbf{HA}^\omega + \mathrm{AC} + \mathrm{IP}_{\mathrm{ef}}$ proves $((\forall x. \neg\, x \,\mathbf{mr}\, A) \to \exists y.B(y)) \to \exists y.((\forall x. \neg\, x \,\mathbf{mr}\, A) \to B(y))$. But, this trivially is the case as the formula $((\forall x. \neg\, x \,\mathbf{mr}\, A) \to \exists y.B(y)) \to \exists y.((\forall x. \neg\, x \,\mathbf{mr}\, A) \to B(y))$ is an instance of $\mathrm{IP}_{\mathrm{ef}}$ because $\forall x. \neg\, x \,\mathbf{mr}\, A$ is $\exists$–free. $\square$

Next we show via modified realizability—as already announced—that in $\mathbf{HA}^\omega$ one cannot derive Markov's Principle.

Theorem 13.3 (Independence of Markov's Principle)
Relative to $\mathbf{HA}^\omega$ + AC + $\mathrm{IP}_{\mathrm{ef}}$ *Markov's Principle* $\mathrm{MP}_{\mathrm{PR}}$ *is inconsistent with Church's Thesis*

$$\mathrm{CT} \qquad \forall f{:}0{\to}0.\exists x{:}0.\forall y{:}0.\, f(y){=}\{x\}(y) \quad .$$

Thus, in $\mathbf{HA}^\omega$ *one cannot prove* $\mathrm{MP}_{\mathrm{PR}}$ *nor its modified realizability.*

Proof: From the previous Corollary 13.2 we know that $\mathbf{HA}^\omega$ + AC + $\mathrm{IP}_{\mathrm{ef}}$ proves IP. Assume that $\neg\neg\,\exists y.\,T(x,x,y) \to \exists y.T(x,x,y)$ which is an instance of $\mathrm{MP}_{\mathrm{PR}}$ as Kleene's T–predicate is primitive recursive. Then from IP it follows that $\forall x.\exists y.\,\neg\neg\,\exists y.\,T(x,x,y) \to T(x,x,y)$. Thus, by AC it follows that

$$\exists f{:}0{\to}0.\forall x{:}0.\,\neg\neg\,\exists y.\,T(x,x,y) \to T(x,x,f(x))$$

and, therefore, also

$$\exists f{:}0{\to}0.\forall x{:}0.\,\neg\,T(x,x,f(x)) \to \neg\,\exists y.\,T(x,x,y)$$

by contraposition. Thus, we have

$$\exists f{:}0{\to}0.\forall x{:}0.\,T(x,x,f(x)) \leftrightarrow \exists y.T(x,x,y)$$

from which it follows that $K{=}\{x|\exists y.T(x,x,y)\}$ is decided by the function g with $g(x){=}0$ iff $T(x,x,f(x))$. But as such a g cannot be recursive we have shown that $\mathbf{HA}^\omega$ + AC + $\mathrm{IP}_{\mathrm{ef}}$ + $\mathrm{MP}_{\mathrm{PR}}$ proves the negation of Church's Thesis.
If $\mathbf{HA}^\omega$ proves $\mathrm{MP}_{\mathrm{PR}}$ or its modified realizability then $\mathbf{HA}^\omega$ + AC + $\mathrm{IP}_{\mathrm{ef}}$ proves the modified realizability of $\mathrm{MP}_{\mathrm{PR}}$ and, therefore, also $\mathrm{MP}_{\mathrm{PR}}$ itself by Theorem 13.2(1). Thus, it follows that $\mathbf{HA}^\omega + \mathrm{AC} + \mathrm{IP}_{\mathrm{ef}} \vdash \neg\mathrm{CT}$ and, therefore, by Theorem 13.2(2) that $\mathbf{HA}^\omega \vdash \exists x.\,x\,\mathbf{mr}\,\neg\mathrm{CT}$. But as $\exists x.\,x\,\mathbf{mr}\,\neg\mathrm{CT}$ is provably equivalent to $\neg\exists x.x\,\mathbf{mr}\,\mathrm{CT}$ it follows that $\mathbf{HA}^\omega \vdash \neg\exists x.x\,\mathbf{mr}\,\mathrm{CT}$ which is impossible as $\mathbf{HA}^\omega + \mathrm{CT} \vdash \exists \mathrm{x.x}\,\mathbf{mr}\,\mathrm{CT}$ and HRO and HEO are models for $\mathbf{HA}^\omega + \mathrm{CT}$.
□

We remark that the underivability of $\mathrm{MP}_{\mathrm{PR}}$ in $\mathbf{HA}^\omega$ can be shown more directly without using Corollary 13.2. Suppose that $\mathbf{HA}^\omega$ proves $\neg\neg\,\exists y.\,T(x,x,y) \to \exists y.T(x,x,y)$. Then by the Soundness Theorem 13.1 there is a term t for which $\mathbf{HA}^\omega$ proves $t\,\mathbf{mr}\,\neg\neg\exists y.T(x,x,y){\to}\exists y.T(x,x,y)$. We leave it as an exercise to show that from this t one may construct a closed term $\tilde{t}$ for which $\mathbf{HA}^\omega$ proves $\forall x.\neg\neg\,\exists y.\,T(x,x,y) \to T(x,x,\tilde{t}(x))$ and thus also $\forall x.(\exists y.T(x,x,y) \leftrightarrow T(x,x,\tilde{t}(x))$. But this program $\tilde{t}$ would decide the halting problem which is known to be impossible.

We conclude this section by noticing that just as for number realizability one may define a variant of modified realizability "combining it with truth", i.e. one defines a relation $x\,\mathbf{mrt}\,A$ just as in Definition 13.1 but with the clause for implication replaced by

$$x\,\mathbf{mrt}\,A{\to}B \quad\equiv\quad (\forall y{:}\mathsf{tp}(A).\ y\,\mathbf{mrt}\,A \to x(y)\,\mathbf{mrt}\,B) \wedge (A{\to}B) \quad .$$

This modification allows one to prove in $\mathbf{HA}^\omega$ that

$$\exists x.\, x\,\mathbf{mrt}\,A \to A$$

for all propositions A. This allows one to establish the following metamathematical facts.

Theorem 13.4 (Applications of Modified Realizability combined with Truth) *Let* $\mathbf{H}$ *be one of the systems* $\mathbf{HA}^\omega$, $\mathbf{I{-}HA}^\omega$, $\mathbf{E{-}HA}^\omega$ *possibly extended by the principles* AC *and/or* $\mathrm{IP}_{\mathrm{ef}}$. *Then*

(1) $\mathbf{H}$ *is consistent relative to* $\mathbf{HA}^\omega$.

(2) *For closed* A *and* B *it follows from* $\mathbf{H} \vdash A{\vee}B$ *that* $\mathbf{H} \vdash A$ *or* $\mathbf{H} \vdash B$.

(3) *If* $\mathbf{H} \vdash \exists x.A(x)$ *then* $\mathbf{H} \vdash A(t)$ *for some term* t *with* $FV(t) \subseteq FV(\exists x.A(x))$.

(4) *If* $\mathbf{H} \vdash \forall x{:}\sigma.\exists y{:}\tau.A(x,y)$ *then* $\mathbf{H} \vdash \exists z{:}\sigma{\to}\tau.\forall x{:}\sigma.A(x,z(x))$.

(5) *If* $\mathbf{H} \vdash A{\to}\exists x.B(x)$ *then* $\mathbf{H} \vdash \exists x.(A{\to}B(x))$ *if* A *is* $\exists$*–free and* $x \notin FV(A)$.

14 Gödel's Functional Interpretation

Already in 1941 Kurt Gödel has devised a method for proof extraction by associating constructive functionals as witnesses or realizers to proofs in **HA** or $\mathbf{HA}^\omega$. It was not until 1958 that this work was published (in German[31]) in the philosophical journal *Dialectica* under the title "Über eine bisher nicht benützte Erweiterung des finiten Standpunktes". That is the reason why Gödel's functional interpretation has become known under the name "Dialectica interpretation". According to the foundationalist ideology which was still[32] very influential

[31]An English translation of this paper can be found in Gödel's Collected Works published by Oxford University Press

[32]This situation changed dramatically in the 60ies under the influence of G. Kreisel who strongly propagated the view that the significance of the proof–theoretic techniques developed for 'foundationalist' purposes does not lie in their contribution to these very "foundations". His argument was that consistency proofs for, say, arithmetic use principles more dubious than those postulated by arithmetic itself. For example Gentzen used ε_0–induction to show the consisteny of induction over natural numbers! This necessarily has to be case as by Gödel's 2nd Incompleteness Theorem no formal system containing a modicum of arithmetic can prove its own consisteny. Instead, as Kreisel emphasized, the relevance of proof–theoretic methods is that they provide techniques for explicitating or "unwinding" (as he preferred to say) the constructive contents of formal proofs like algorithms or bounds.

in the 40ies and 50ies as a consequence of the *Grundlagenkrise* of the time between the two World Wars Gödel 'sold' his functional interpretation of arithmetic as a consistency proof. But, actually, its relevance rather is that it allows one to extract from proofs programs in Gödels's T which was introduced by him for this purpose. Though Gödel's Functional Interpretation is the oldest such technique it appears as the most complex in this vein. But, despite its intriguing character and inherent (conceptual) complexity it is still of quite some interest because it simultaneously validates a strong form of Markov's Principle and a sufficiently strong version of the principle of Independence of Premiss.
The basic idea of Gödel's Functional Interpretation is to associate with every proposition A a proposition $A^D \equiv \exists x.\forall u.A_D(x,u)$ where $A_D(x,u)$ is a primitive recursive(sic!) relation between x and u of appropriate finite type. One may think of x as a 'strategy' and u as a 'counterstrategy' and of $A_D(x,u)$ as "x wins against u".
In order to simplify notation instead of x and u we consider finite lists $\mathfrak{x}$ and $\mathfrak{u}$ of variables of appropriate type.[33] We use $\mathfrak{x}, \mathfrak{y}, \mathfrak{z}, \mathfrak{u}, \mathfrak{v}, \mathfrak{w}, \mathfrak{X}, \mathfrak{Y}, \mathfrak{Z}, \mathfrak{U}, \mathfrak{V}, \mathfrak{W} \ldots$ as meta–variables for finite list of variables and $\mathfrak{s}, \mathfrak{t}$ as metavariables for finite lists of terms. If $\mathfrak{t} \equiv t_1, \ldots, t_n$ then we write $\mathfrak{t}\mathfrak{s}$ as an abbreviation for $t_1\mathfrak{s}, \ldots, t_n\mathfrak{s}$ whenever the $t_i\mathfrak{s}$ are well–formed. Moreover, if $\mathfrak{x} \equiv x_1, \ldots, x_n$ then we write $\forall\mathfrak{x}$ and $\exists\mathfrak{x}$ as shorthand for $\forall x_1. \ldots \forall x_n.$ and $\exists x_1. \ldots \exists x_n.$, respectively.
Next we consider how the implication $\exists\mathfrak{x}.\forall\mathfrak{u}.A_D(\mathfrak{x},\mathfrak{u}) \to \exists\mathfrak{y}.\forall\mathfrak{v}.B_D(\mathfrak{y},\mathfrak{v})$ can be transformed into an equivalent formula of the form $\exists\mathfrak{z}.\forall\mathfrak{w}.(A{\to}B)_D(\mathfrak{z},\mathfrak{w})$ as motivation for the subsequent definition of functional interpretation of implications. We have

$$\begin{array}{ll}
\exists\mathfrak{x}.\forall\mathfrak{u}.A_D(\mathfrak{x},\mathfrak{u}) \to \exists\mathfrak{y}.\forall\mathfrak{v}.B_D(\mathfrak{y},\mathfrak{v}) & \leftrightarrow \\
\forall\mathfrak{x}.(\forall\mathfrak{u}.A_D(\mathfrak{x},\mathfrak{u}) \to \exists\mathfrak{y}.\forall\mathfrak{v}.B_D(\mathfrak{y},\mathfrak{v})) & \leftrightarrow \\
\forall\mathfrak{x}.\exists\mathfrak{y}.(\forall\mathfrak{u}.A_D(\mathfrak{x},\mathfrak{u}) \to \forall\mathfrak{v}.B_D(\mathfrak{y},\mathfrak{v})) & \leftrightarrow \\
\forall\mathfrak{x}.\exists\mathfrak{y}.\forall\mathfrak{v}(\forall\mathfrak{u}.A_D(\mathfrak{x},\mathfrak{u}) \to B_D(\mathfrak{y},\mathfrak{v})) & \leftrightarrow \\
\forall\mathfrak{x}.\exists\mathfrak{y}.\forall\mathfrak{v}.\exists\mathfrak{u}(A_D(\mathfrak{x},\mathfrak{u}) \to B_D(\mathfrak{y},\mathfrak{v})) & \leftrightarrow \\
\exists\mathfrak{Y},\mathfrak{U}.\forall\mathfrak{x},\mathfrak{v}.(A_D(\mathfrak{x},\mathfrak{U}\mathfrak{x}\mathfrak{v}) \to B_D(\mathfrak{Y}\mathfrak{x},\mathfrak{v})) &
\end{array}$$

where the second and the fourth equivalence are valid only classically and the fifth equivalence presupposes AC. In principle classical logic would permit other ways of prenexing but the ensuing functional interpretations would not be sufficiently well–behaved w.r.t the purposes of program instruction.
As for atomic formulas P we want to put $P_D \equiv P$ we have to restrict consideration to systems like $\mathbf{I{-}HA}^\omega$ or $\mathbf{E{-}HA}^\omega$ where atomic formulas are decidable[34]. *In the following we use* $\mathbf{H}$ *as a variable ranging over* $\{\mathbf{I{-}HA}^\omega, \mathbf{E{-}HA}^\omega\}$.

[33]This notational trick could have been used in the formalization of modified realizability, too. It would have had the nice effect that for $\exists$–free formulas A we would get $\mathfrak{x}\,\mathbf{mr}\,A \equiv A$ when defining $\mathfrak{x}\,\mathbf{mr}\,P \equiv P$ for atomic P.

[34]We shall see subsequently that decidability of A_D is needed for showing that $A{\to}A{\wedge}A$ is validated by functional interpretation in the sense of Definition 14.1. But, moreover, there are

Definition 14.1 (Gödel's Functional Interpretation)
For atomic formulas P we put

$$P^D \equiv P_D \equiv P$$

and if $A^D \equiv \exists \mathfrak{x}.\forall \mathfrak{u}.\, A_D(\mathfrak{x},\mathfrak{u})$ *and* $B^D \equiv \exists \mathfrak{y}.\forall \mathfrak{v}.\, B_D(\mathfrak{y},\mathfrak{v})$ *then we put*

$(A{\wedge}B)^D \equiv \exists \mathfrak{x}\mathfrak{y}.\forall \mathfrak{u}\mathfrak{v}.\, (A{\wedge}B)_D(\mathfrak{x},\mathfrak{y},\mathfrak{u},\mathfrak{v})$ *where*
$(A{\wedge}B)_D(\mathfrak{x},\mathfrak{y},\mathfrak{u},\mathfrak{v}) \equiv A_D(\mathfrak{x},\mathfrak{u}){\wedge}B_D(\mathfrak{y},\mathfrak{v})$

$(A{\to}B)^D \equiv \exists \mathfrak{Y}\mathfrak{U}.\forall \mathfrak{x}\mathfrak{v}.\, (A{\to}B)_D(\mathfrak{Y},\mathfrak{U},\mathfrak{x},\mathfrak{v})$ *where*
$(A{\to}B)_D(\mathfrak{Y},\mathfrak{U},\mathfrak{x},\mathfrak{v}) \equiv A_D(\mathfrak{x},\mathfrak{U}\mathfrak{x}\mathfrak{v}){\to}B_D(\mathfrak{Y}\mathfrak{x},\mathfrak{v})$

$(\forall z.A(z))^D \equiv \exists \mathfrak{X}.\forall z\mathfrak{u}.\, (\forall z.A(z))_D(\mathfrak{X},z,\mathfrak{u})$ *where*
$(\forall z.A(z))_D(\mathfrak{X},z,\mathfrak{u}) \equiv A_D(z,\mathfrak{X}z,\mathfrak{u})$

$(\exists z.A(z))^D \equiv \exists z\mathfrak{x}.\forall \mathfrak{u}.\, \exists z.A(z))_D(z,\mathfrak{x},\mathfrak{u})$ *where*
$(\exists z.A(z))_D(z,\mathfrak{x},\mathfrak{u}) \equiv A_D(z,\mathfrak{x},\mathfrak{u})$.

We say that functional interpretation validates A *iff* $\mathbf{H} \vdash \exists \mathfrak{x}.\forall \mathfrak{u}.A_D(\mathfrak{x},\mathfrak{u})$. $\Diamond$

As in arithmetic $A{\vee}B$ may be considered as an abbreviation for the formula $\exists n.\,(n{=}0{\to}A){\wedge}(n{\neq}0{\to}B)$ whose functional interpretation is given by

$(A{\vee}B)^D \equiv \exists n\mathfrak{x}\mathfrak{y}.\forall \mathfrak{u}\mathfrak{v}.\, (A{\vee}B)_D(n,\mathfrak{x},\mathfrak{y},\mathfrak{u},\mathfrak{v})$ where
$(A{\vee}B)_D(n,\mathfrak{x},\mathfrak{y},\mathfrak{u},\mathfrak{v}) \equiv (n{=}0{\to}A_D(\mathfrak{x},\mathfrak{u})){\wedge}(n{\neq}0{\to}B_D(\mathfrak{y},\mathfrak{v}))$.

Next we consider how functional interpretation treats negations. As by definition $\neg A \equiv A{\to}\bot$ we have

$$(\neg A)^D \equiv \exists \mathfrak{U}.\forall \mathfrak{x}.\, \neg A_D(\mathfrak{x},\mathfrak{U}\mathfrak{x})$$

when $A^D \equiv \exists \mathfrak{x}.\forall \mathfrak{u}.A_D(\mathfrak{x},\mathfrak{u})$. Thus, as special cases we get

$$(\neg A)^D \equiv \forall \mathfrak{x}.\neg A_D(\mathfrak{x})$$

if $\mathfrak{u}$ is empty and

$$(\neg A)^D \equiv \exists \mathfrak{u}.\neg A_D(\mathfrak{u})$$

if $\mathfrak{x}$ is empty. Accordingly, we have

$$(\neg\neg\exists \mathfrak{x}.P(\mathfrak{x}))^D \equiv \exists \mathfrak{x}.\neg\neg P(\mathfrak{x})$$

for every atomic formula $P(\mathfrak{x})$ as $(\neg\exists \mathfrak{x}.P(\mathfrak{x}))^D \equiv \forall \mathfrak{x}.\neg P(\mathfrak{x})$. As a consequence we get that a rather strong form of Markov's principle is validated by functional interpretation.

formulas provable in $\mathbf{HA}^\omega$ which are not validated by functional interpretation as for example $\forall u,v{:}\,0{\to}0.\neg\forall z{:}\,0.\neg(z{=}0{\leftrightarrow}x{=}y)$. This formula is provable in $\mathbf{HA}^\omega$ as it appears as the $\neg\neg$–translation of $\forall u,v{:}\,0{\to}0.\exists z{:}\,0.(z{=}0{\leftrightarrow}x{=}y)$ which, obviously, is provable in $\mathbf{HA}^\omega$ + PEM. But if $\forall u,v{:}\,0{\to}0.\neg\forall z{:}\,0.\neg(z{=}0{\leftrightarrow}x{=}y)$ were validated by functional interpretation there would exist a closed term of type 2 deciding whether $\neg x{=}y$ or $\neg\neg x{=}y$. But this is impossible as otherwise equality of total recursive functions were decidable.

Lemma 14.1 *Functional interpretation validates*

$$\mathrm{M}' \qquad \neg\neg\exists\mathfrak{x}.P(\mathfrak{x}) \to \exists\mathfrak{x}.P(\mathfrak{x}) \qquad \textit{for } P(\mathfrak{x}) \textit{ atomic.}$$

Proof: Immediate from the fact that $(\neg\neg\exists\mathfrak{x}.P(\mathfrak{x}))^D \equiv \exists\mathfrak{x}.\neg\neg P(\mathfrak{x})$ is provably equivalent to $\exists\mathfrak{x}.P(\mathfrak{x})$ as $P(\mathfrak{x})$ is decidable and hence $\neg\neg$–closed. □

This together with the following lemma will be useful for axiomatizing Functional Interpretation.

Lemma 14.2 *Functional interpretation validates*

$$\mathrm{IP}_0' \qquad (\forall\mathfrak{x}.P(\mathfrak{x}){\to}\exists\mathfrak{y}.A(\mathfrak{y})) \to \exists\mathfrak{y}.(\forall\mathfrak{x}.P(\mathfrak{x}){\to}A(\mathfrak{y})) \qquad \textit{for } P(\mathfrak{x}) \textit{ atomic.}$$

Proof: Let $A(\mathfrak{y})^D \equiv \exists\mathfrak{u}.\forall\mathfrak{v}.A_D(\mathfrak{y},\mathfrak{u},\mathfrak{v})$. As $(\forall\mathfrak{x}.P(\mathfrak{x}))^D \equiv \forall\mathfrak{x}.P(\mathfrak{x})$ we observe that both $(\forall\mathfrak{x}.P(\mathfrak{x}){\to}\exists\mathfrak{y}.A(\mathfrak{y}))^D$ and $(\exists\mathfrak{y}.(\forall\mathfrak{x}.P(\mathfrak{x}){\to}A(\mathfrak{y})))^D$ are syntactically equal to $\exists\mathfrak{y},\mathfrak{u},\mathfrak{X}.\forall\mathfrak{v}.\,(A_D(\mathfrak{X}\mathfrak{v}){\to}B_D(\mathfrak{y},\mathfrak{u},\mathfrak{v}))$ from which it readily follows that functional interpretation validates the schema IP_0'. □

Functional interpretation is sound in the following sense.

Theorem 14.1 (Soundness of Functional Interpretation)
If $\mathbf{H} \vdash A$ *then there is a list* $\mathfrak{t}$ *of terms such that* $\mathbf{H} \vdash \forall\mathfrak{u}.A_D(\mathfrak{t},\mathfrak{u})$.

Proof: As usual we proceed by induction on the structure of derivations in **H**. Almost everything goes through without any surprises. The only exception is the verification of the correctness of the contraction rule. This requires to show that for every formula A functional interpretation validates $A{\to}A{\wedge}A$, i.e. we have to show that there are $\mathfrak{X}_1$, $\mathfrak{X}_2$ and $\mathfrak{Y}$ such that **H** proves

$$A_D(\mathfrak{x},\mathfrak{Y}\mathfrak{x}\mathfrak{u}_1\mathfrak{u}_2) \to A_D(\mathfrak{X}_1\mathfrak{x},\mathfrak{u}_1) \wedge A_D(\mathfrak{X}_2\mathfrak{x},\mathfrak{u}_2) \quad .$$

The following turns out as a good choice: put $\mathfrak{X}_1\mathfrak{x} = \mathfrak{x} = \mathfrak{X}_2\mathfrak{x}$ and

$$\mathfrak{Y}\mathfrak{u}_1\mathfrak{u}_2 = \begin{cases} \mathfrak{u}_1 & \text{if } t\mathfrak{x}\mathfrak{u}_1 \neq 0 \\ \mathfrak{u}_2 & \text{if } t\mathfrak{x}\mathfrak{u}_1 = 0 \end{cases}$$

where t is a primitive recursive term deciding A_D, i.e. $t\mathfrak{x}\mathfrak{u} = 0$ iff $A_D(\mathfrak{x},\mathfrak{u})$. □

Notice that we have used intrinsically that the atomic formulas of **H** are decidable by primitive recursive terms of the language.
Moreover, one can show that $A_D(\mathfrak{t},\mathfrak{u})$ can be proved in the quantifier–free fragment QF–**H** of **H** augmented with an induction rule. Thus, as the proof of the Soundness Theorem 14.1 is purely combinatorial in character and, therefore, can be performed in a very weak system (as e.g. **PRA**) it can be considered as a 'finitistic' reduction of the consistency of **H** to the consistency of QF–**H**. The

latter was considered by Gödel as evident due to the constructive nature of the notion of primitive recursive functionals of finite type. Thus, Gödel thought of his result as a consistency proof for intuitionistic (and, therefore, also) classical arithmetic by "essentially[35] finitistic" means.

Now we are going to axiomatize functional interpretation which this time will be easier as there is no need for considering a syntactically restricted class of formulas.

Lemma 14.3 *The system* $\mathbf{H} + \mathrm{AC} + \mathrm{IP}'_0 + \mathrm{M}'$ *proves* $A \leftrightarrow \exists \mathfrak{x}.\forall \mathfrak{u}.A_D(\mathfrak{x},\mathfrak{u})$ *for all formulas* A*.*

Proof: We proceed by induction on the structure of A. The only crucial case is implication. Suppose as induction hypothesis that $\mathbf{H} + \mathrm{AC} + \mathrm{IP}'_0 + \mathrm{M}'$ proves $A \leftrightarrow \exists \mathfrak{x}.\forall \mathfrak{u}.A_D(\mathfrak{x},\mathfrak{u})$ and $B \leftrightarrow \exists \mathfrak{y}.\forall \mathfrak{v}.A_D(\mathfrak{y},\mathfrak{v})$.
Then $A \to B$ is equivalent to $\exists \mathfrak{x}.\forall \mathfrak{u}.A_D(\mathfrak{x},\mathfrak{u}) \to \exists \mathfrak{y}.\forall \mathfrak{v}.B_D(\mathfrak{y},\mathfrak{v})$ which in turn (using IP'_0) is equivalent to $\forall \mathfrak{x}.\exists \mathfrak{y}.\forall \mathfrak{v}.\,(\forall \mathfrak{u}.A_D(\mathfrak{x},\mathfrak{u}) \to B_D(\mathfrak{y},\mathfrak{v}))$. Now if we can show that $\forall \mathfrak{u}.A_D(\mathfrak{x},\mathfrak{u}) \to B_D(\mathfrak{y},\mathfrak{v})$ is equivalent to $\exists \mathfrak{u}.(A_D(\mathfrak{x},\mathfrak{u}) \to B_D(\mathfrak{y},\mathfrak{v}))$ it follows that $A \to B$ is equivalent to $\forall \mathfrak{x}.\exists \mathfrak{y}.\forall \mathfrak{v}.\exists \mathfrak{u}.\,(A_D(\mathfrak{x},\mathfrak{u}) \to B_D(\mathfrak{y},\mathfrak{v}))$ which by AC is equivalent to $(A \to B)^D$.
The desired equivalence of $\forall \mathfrak{u}.A_D(\mathfrak{x},\mathfrak{u}) \to B_D(\mathfrak{y},\mathfrak{v})$ and $\exists \mathfrak{u}.(A_D(\mathfrak{x},\mathfrak{u}) \to B_D(\mathfrak{y},\mathfrak{v}))$ can be seen as follows

$$\begin{array}{l}
\forall \mathfrak{u}.A_D(\mathfrak{x},\mathfrak{u}) \to B_D(\mathfrak{y},\mathfrak{v}) \quad\leftrightarrow\quad B_D(\mathfrak{y},\mathfrak{v}) \lor (\neg B_D(\mathfrak{y},\mathfrak{v}) \land \neg\forall \mathfrak{u}.A_D(\mathfrak{x},\mathfrak{u})) \quad\leftrightarrow \\
\quad\leftrightarrow\quad B_D(\mathfrak{y},\mathfrak{v}) \lor (\neg B_D(\mathfrak{y},\mathfrak{v}) \land \neg\forall \mathfrak{u}.\neg\neg A_D(\mathfrak{x},\mathfrak{u})) \quad\leftrightarrow \\
\quad\leftrightarrow\quad B_D(\mathfrak{y},\mathfrak{v}) \lor (\neg B_D(\mathfrak{y},\mathfrak{v}) \land \neg\neg\exists \mathfrak{u}.\neg A_D(\mathfrak{x},\mathfrak{u})) \quad\leftrightarrow \quad \text{(by M}'\text{)} \\
\quad\leftrightarrow\quad B_D(\mathfrak{y},\mathfrak{v}) \lor (\neg B_D(\mathfrak{y},\mathfrak{v}) \land \exists \mathfrak{u}.\neg A_D(\mathfrak{x},\mathfrak{u})) \quad\leftrightarrow \\
\quad\leftrightarrow\quad \exists \mathfrak{u}.(B_D(\mathfrak{y},\mathfrak{v}) \lor (\neg B_D(\mathfrak{y},\mathfrak{v}) \land \neg A_D(\mathfrak{x},\mathfrak{u})) \quad\leftrightarrow \\
\quad\leftrightarrow\quad \exists \mathfrak{u}.(A_D(\mathfrak{x},\mathfrak{u}) \to B_D(\mathfrak{y},\mathfrak{v}))
\end{array}$$

using intrinsically the decidability of A_D and B_D. Notice that for applying M' correctly we have used (implicitly) that the quantifier–free formula $\neg A_D$ is equivalent to an atomic formula. $\square$

Now we can axiomatize Gödel's functional interpretation by AC and the principles

$$\begin{array}{ll}
\mathrm{IP}_0^\omega & (\forall \mathfrak{x}.\,A(\mathfrak{x}) \lor \neg A(\mathfrak{x})) \to (\forall \mathfrak{x}.A(\mathfrak{x}) \to \exists \mathfrak{y}.B(\mathfrak{y})) \to \exists \mathfrak{y}.(\forall \mathfrak{x}.A(\mathfrak{x}) \to B(\mathfrak{y})) \\
\mathrm{M}^\omega & (\forall \mathfrak{x}.\,A(\mathfrak{x}) \lor \neg A(\mathfrak{x})) \to \neg\neg\exists \mathfrak{x}.A(\mathfrak{x}) \to \exists \mathfrak{x}.A(\mathfrak{x})
\end{array}$$

for arbitrary formulas A and B.

[35]Hilbert's notion of 'finitistic' did not encompass the notion of primitive recursive functionals of finite type. But 'finitistic' was (and still is) an open concept and Gödel definitely thought that it should encompass the primitive recursive functionals of finite type.

Theorem 14.2 (Axiomatisation of Functional Interpretation)
For all formulas A of $\mathbf{H}$ *we have*

(1) $\mathbf{H} + \mathrm{AC} + \mathrm{IP}_0^\omega + \mathrm{M}^\omega \vdash \mathrm{A} \leftrightarrow \exists \mathfrak{x}.\forall \mathfrak{u}.\mathrm{A}_\mathrm{D}(\mathfrak{x}, \mathfrak{u})$

(2) $\mathbf{H} + \mathrm{AC} + \mathrm{IP}_0^\omega + \mathrm{M}^\omega \vdash \mathrm{A}$ *iff* $\mathbf{H} \vdash \exists \mathfrak{x}.\forall \mathfrak{u}.A_D(\mathfrak{x}, \mathfrak{u})$.

Proof: First notice that in $\mathbf{H} + \mathrm{AC}$ one easily proves the equivalence of IP_0' and IP_0^ω and of M' and M^ω as atomic predicates of $\mathbf{H}$ are decidable and in $\mathbf{H} + \mathrm{AC}$ every decidable predicate can be shown to be equivalent to an atomic predicate as AC guarantees the existence of a decision function. Thus, claim (1) follows immediately from Lemma 14.3.
The direction from right to left of (2) follows immediately from (1). One easily shows that AC is validated by functional interpretation. As we already know that w.r.t. $\mathbf{H} + \mathrm{AC}$ the principles IP_0^ω and M^ω are equivalent to IP_0' and M', respectively, it follows immediately from the Soundness Theorem 14.1 and the Lemmas 14.1 and 14.2 that IP_0^ω and M^ω are validated by functional interpretation. Thus, the direction from left to right of (2) follows from these observations by the Soundness Theorem 14.1. $\square$

We conclude this section by remarking that there exists the so–called Diller–Nahm variant of Gödel's functional interpretation which allows one to get rid of the requirement that atomic formulas are decidable and, accordingly, extends to $\mathbf{HA}^\omega$ with or without the extensionality principle.

References

[Bar] J. Barwise (ed.) *Handbook of Mathematical Logic* North Holland, 1977.

[BiBr] E. Bishop, D. Bridges *Constructive Analysis* Grundlehren der mathematischen Wissenschaften 279, Springer, 1985.

[Dum] M. Dummett *Elements of Intuitionism* Oxford University Press, 2000.

[KV] S. C. Kleene, R. Vesley *The Foundations of Intuitionistic Mathematics* North Holland, 1965.

[Sc69] D. Scott *A type theoretical alternative to ISWIM, CUCH, OWHY* Theoretical Computer Science 121, pp.411–440, 1993 (Reprint of a manuscript written at Oxford University in 1969).

[Tr73] A. Troelstra (ed.) *Metamathematical Investigations of Intuitionistic Arithmetic and Analysis* SLNM 344, Springer Verlag, 1973.

[Tr77] A. Troelstra *Aspects of Constructive Mathematics* pp. 973-1052 of [Bar].

[TvD] A. Troelstra, D. vanDalen *Constructivism in Mathematics* 2 vol.'s, North Holland, 1988.