# Proof Theory

Helmut Schwichtenberg

Notes for a lecture course, Sommersemester 2006.
Mathematisches Institut der Ludwig-Maximilians-Universität,
Theresienstraße 39, D-80333 München, Germany.
November 10, 2006.

# Contents

# Introduction

The goal of this course is to study computable functionals, in the context of a minimal logical system allowing to do proofs about them.

We develop a natural deduction system for minimal logic in the language based on implication $\rightarrow$, conjunction $\wedge$, disjunction $\vee$ and the quantifiers $\forall$ and $\exists$. We present a general notion of a model suitable for minimal logic, called Beth-structures. For such models we prove a soundness and a completeness theorem; both proofs are constructive.

Intuitionistic logic can be embedded into minimal logic, and the soundness and completeness proofs carry over, again with constructive proofs.

Classical logic can be embedded into minimal logic as well. For classical logic a different notion of a structure and of validity in such structures is appropriate. The soundness theorem for classical models is easy. We will derive the completeness theorem for classical logic as a consequence of the one for minimal logic. However, this proof will not be constructive any more. We will need the law of excluded middle and the classical axiom of dependent choice.

We also present a different completeness proof for classical logic. It consists in simultaneously searching for a derivation and a counterexample. Special attention is given to the principles used in this proof which go beyond minimal logic. We show that a test for infinity of a binary decidable tree suffices, and in fact is equivalent to the completeness theorem.

CHAPTER 1

# Logic

The main subject of Mathematical Logic is mathematical proof. In this introductory chapter we deal with the basics of formalizing such proofs. The system we pick for the representation of proofs is Gentzen's natural deduction, from (1934). Our reasons for this choice are twofold. First, as the name says this is a *natural* notion of formal proof, which means that the way proofs are represented corresponds very much to the way a careful mathematician writing out all details of an argument would go anyway. Second, formal proofs in natural deduction are closely related (via the so-called Curry-Howard correspondence) to terms in typed lambda calculus. This provides us not only with a compact notation for logical derivations (which otherwise tend to become somewhat unmanagable tree-like structures), but also opens up a route to applying the computational techniques which underpin lambda calculus.

Apart from classical logic we will also deal with more constructive logics: minimal and intuitionistic logic. This will reveal some interesting aspects of proofs, e.g., that it is possible und useful to distinguish beween existential proofs that actually construct witnessing objects, and others that don't.

An essential point for Mathematical Logic is to fix a formal language to be used. We take implication $\to$ and the universal quantifier $\forall$ as basic. Then the logic rules correspond to lambda calculus. The additional connectives $\bot$, $\exists$, $\vee$ and $\wedge$ are defined via axiom schemes. These axiom schemes will later be seen as special cases of introduction and elimination rules for inductive definitions.

## 1.1. Formal Languages

**1.1.1. Terms and formulas.** Let a countable infinite set $\{ v_i \mid i \in \mathbb{N} \}$ of *variables* be given; they will be denoted by $x, y, z$. A first order language $\mathcal{L}$ then is determined by its *signature*, which is to mean the following.

- For every natural number $n \geq 0$ a (possible empty) set of $n$-ary *relation symbols* (also called *predicate symbols*). 0-ary relation symbols are called *propositional symbols*. $\bot$ (read "falsum") is required as

a fixed propositional symbol. The language will *not*, unless stated otherwise, contain $=$ as a primitive.
- For every natural number $n \geq 0$ a (possible empty) set of $n$-ary *function symbols*. 0-ary function symbols are called *constants*.

We assume that all these sets of variables, relation and function symbols are disjoint.

$\mathcal{L}$-*terms* are inductively defined as follows.

- Every variable is an $\mathcal{L}$-term.
- Every constant of $\mathcal{L}$ is an $\mathcal{L}$-term.
- If $t_1, \ldots, t_n$ are $\mathcal{L}$-terms and $f$ is an $n$-ary function symbol of $\mathcal{L}$ with $n \geq 1$, then $f(t_1, \ldots, t_n)$ is an $\mathcal{L}$-term.

From $\mathcal{L}$-terms one constructs $\mathcal{L}$-*prime formulas*, also called *atomic formulas* of $\mathcal{L}$: If $t_1, \ldots, t_n$ are terms and $R$ is an $n$-ary relation symbol of $\mathcal{L}$, then $R(t_1, \ldots, t_n)$ is an $\mathcal{L}$-prime formula.

$\mathcal{L}$-*formulas* are inductively defined from $\mathcal{L}$-prime formulas by

- Every $\mathcal{L}$-prime formula is an $\mathcal{L}$-formula.
- If $A$ and $B$ are $\mathcal{L}$-formulas, then so are $(A \to B)$ ("if $A$, then $B$"), $(A \wedge B)$ ("$A$ and $B$") and $(A \vee B)$ ("$A$ or $B$").
- If $A$ is an $\mathcal{L}$-formula and $x$ is a variable, then $\forall_x A$ ("for all $x$, $A$ holds") and $\exists_x A$ ("there is an $x$ such that $A$") are $\mathcal{L}$-formulas.

Negation, classical disjunction, and the classical existential quantifier are defined by

$$\begin{aligned} \neg A &:= A \to \bot, \\ A \mathbin{\tilde{\vee}} B &:= \neg A \to \neg B \to \bot, \\ \tilde{\exists} x A &:= \neg \forall_x \neg A. \end{aligned}$$

Usually we fix a language $\mathcal{L}$, and speak of terms and formulas instead of $\mathcal{L}$-terms and $\mathcal{L}$-formulas. We use

| | |
|---|---|
| $r, s, t$ | for terms, |
| $x, y, z$ | for variables, |
| $c$ | for constants, |
| $P, Q, R$ | for relation symbols, |
| $f, g, h$ | for function symbols, |
| $A, B, C, D$ | for formulas. |

DEFINITION. The *depth* $\mathrm{dp}(A)$ of a formula $A$ is the maximum length of a branch in its construction tree. In other words, we define recursively $\mathrm{dp}(P) = 0$ for atomic $P$, $\mathrm{dp}(A \circ B) = \max(\mathrm{dp}(A), \mathrm{dp}(B)) + 1$ for binary operators $\circ$, $\mathrm{dp}(\circ A) = \mathrm{dp}(A) + 1$ for unary operators $\circ$.

The *size* or *length* $|A|$ of a formula $A$ is the number of occurrences of logical symbols and atomic formulas (parentheses not counted) in $A$: $|P| = 1$ for $P$ atomic, $|A \circ B| = |A| + |B| + 1$ for binary operators $\circ$, $|\circ A| = |A| + 1$ for unary operators $\circ$.

One can show easily that $|A| + 1 \leq 2^{\mathrm{dp}(A)+1}$.

NOTATION (Saving on parentheses). In writing formulas we save on parentheses by assuming that $\forall, \exists, \neg$ bind more strongly than $\wedge, \vee$, and that in turn $\wedge, \vee$ bind more strongly than $\rightarrow, \leftrightarrow$ (where $A \leftrightarrow B$ abbreviates $(A \rightarrow B) \wedge (B \rightarrow A)$). Outermost parentheses are also usually dropped. Thus $A \wedge \neg B \rightarrow C$ is read as $((A \wedge (\neg B)) \rightarrow C)$. In the case of iterated implications we sometimes use the short notation

$$A_1 \rightarrow A_2 \rightarrow \ldots A_{n-1} \rightarrow A_n \quad \text{for} \quad A_1 \rightarrow (A_2 \rightarrow \ldots (A_{n-1} \rightarrow A_n) \ldots).$$

To save parentheses in quantified formulas, we might use a mild form of the *dot notation*: a dot immediately after $\forall_x$ or $\exists_x$ makes the scope of that quantifier as large as possible, given the parentheses around. So $\forall_x . A \rightarrow B$ means $\forall_x (A \rightarrow B)$, not $(\forall_x A) \rightarrow B$.

We also save on parentheses by writing, e.g., $Rxyz$, $Rt_0t_1t_2$ instead of $R(x, y, z)$, $R(t_0, t_1, t_2)$, where $R$ is some predicate symbol. Similarly for a unary function symbol with a (typographically) simple argument, so $fx$ for $f(x)$, etc. In this case no confusion will arise. But readability requires that we write in full $R(fx, gy, hz)$, instead of $Rfxgyhz$.

Binary function and relation symbols are usually written in *infix notation*, e.g., $x + y$ instead of $+(x, y)$, and $x < y$ instead of $<(x, y)$. We write $t \neq s$ for $\neg(t = s)$ and $t \not< s$ for $\neg(t < s)$.

**1.1.2. Substitution, free and bound variables.** Expressions $\mathcal{E}, \mathcal{E}'$ which differ only in the names of bound variables will be regarded as identical. This is sometimes expressed by saying that $\mathcal{E}$ and $\mathcal{E}'$ are $\alpha$-equivalent. In other words, we are only interested in expressions "modulo renaming of bound variables". There are methods of finding unique representatives for such expressions, for example the namefree terms of de Bruijn (1972). For the human reader such representations are less convenient, so we shall stick to the use of bound variables.

In the definition of "substitution of expression $\mathcal{E}'$ for variable $x$ in expression $\mathcal{E}$", either one requires that *no* variable free in $\mathcal{E}'$ becomes bound by a variable-binding operator in $\mathcal{E}$, when the free occurrences of $x$ are replaced by $\mathcal{E}'$ (also expressed by saying that there must be no "clashes of variables"), "$\mathcal{E}'$ *is free for* $x$ *in* $\mathcal{E}$", or the substitution operation is taken to involve a systematic renaming operation for the bound variables, avoiding clashes. Having stated that we are only interested in expressions modulo

renaming bound variables, we can without loss of generality assume that substitution is always possible.

Also, it is never a real restriction to assume that distinct quantifier occurrences are followed by distinct variables, and that the sets of bound and free variables of a formula are disjoint.

NOTATION. "FV" is used for the (set of) free variables of an expression; so $\mathrm{FV}(t)$ is the set of variables free in the term $t$, $\mathrm{FV}(A)$ the set of variables free in formula $A$ etc.

$\mathcal{E}[x := t]$ denotes the result of substituting the term $t$ for the variable $x$ in the expression $\mathcal{E}$. Similarly, $\mathcal{E}[\vec{x} := \vec{t}]$ is the result of *simultaneously* substituting the terms $\vec{t} = t_1, \ldots, t_n$ for the variables $\vec{x} = x_1, \ldots, x_n$, respectively.

Locally we shall adopt the following convention. In an argument, once a formula has been introduced as $A(x)$, i.e., $A$ with a designated variable $x$, we write $A(t)$ for $A[x := t]$, and similarly with more variables. □

**1.1.3. Subformulas.** Unless stated otherwise, the notion of *subformula* we use will be that of a subformula in the sense of Gentzen.

DEFINITION. (Gentzen) subformulas of $A$ are defined by

(a) $A$ is a subformula of $A$;
(b) if $B \circ C$ is a subformula of $A$ then so are $B$, $C$, for $\circ = \rightarrow, \wedge, \vee$;
(c) if $\forall_x B$ or $\exists_x B$ is a subformula of $A$, then so is $B[x := t]$, for all $t$ free for $x$ in $B$.

If we replace the third clause by:

(c)′ if $\forall_x B$ or $\exists_x B$ is a subformula of $A$ then so is $B$,

we obtain the notion of *literal* subformula.

DEFINITION. The notions of *positive, negative, strictly positive* subformula are defined in a similar style:

(a) $A$ is a positive and a stricly positive subformula of itself;
(b) if $B \wedge C$ or $B \vee C$ is a positive [negative, strictly positive] subformula of $A$, then so are $B$, $C$;
(c) if $\forall_x B$ or $\exists_x B$ is a positive [negative, strictly positive] subformula of $A$, then so is $B[x := t]$;
(d) if $B \rightarrow C$ is a positive [negative] subformula of $A$, then $B$ is a negative [positive] subformula of $A$, and $C$ is a positive [negative] subformula of $A$;
(e) if $B \rightarrow C$ is a strictly positive subformula of $A$, then so is $C$.

A strictly positive subformula of $A$ is also called a *strictly positive part (s.p.p.)* of $A$. Note that the set of subformulas of $A$ is the union of the

positive and negative subformulas of $A$. *Literal* positive, negative, strictly positive subformulas may be defined in the obvious way by restricting the clause for quantifiers.

EXAMPLE. $(P \to Q) \to R \wedge \forall_x R'(x)$ has as s.p.p.'s the whole formula, $R \wedge \forall_x R'(x)$, $R$, $\forall_x R'(x)$, $R'(t)$. The positive subformulas are the s.p.p.'s and in addition $P$; the negative subformulas are $P \to Q$, $Q$.

DEFINITION. *Harrop formulas* (in the literature also called *Rasiowa-Harrop* formulas) are formulas for which no s.p.p. is a disjunction or an existential formula.

These formulas will play an important role later on.

## 1.2. Natural Deduction

We introduce Gentzen's system of natural deduction. To allow a direct correspondence with the lambda calculus, we restrict the rules used to those for the logical connective $\to$ and the universal quantifier $\forall$. The rules come in pairs: we have an introduction and an elimination rule for each of these. The other logical connectives are introduced by means of axiom schemes: this is done for conjunction $\wedge$, disjunction $\vee$ and the existential quantifier $\exists$. The resulting system is called *minimal logic*; it has been introduced by Johansson (1937). Notice that no negation is present.

If we then go on and require the *ex-falso-quodlibet* scheme for the nullary propositional symbol $\bot$ ("falsum"), we can embed *intuitionistic logic* . To obtain classical logic, we add as an axiom scheme the principle of *indirect proof*, also called *stability*. However, to obtain classical logic it suffices to restrict to the language based on $\to$, $\forall$, $\bot$ and $\wedge$; we can introduce classical disjunction $\tilde{\vee}$ and the classical existential quantifier $\tilde{\exists}$ via their (classical) definitions above. For these the usual introduction and elimination properties can then be derived.

**1.2.1. Examples of derivations.** Let us start with some examples for natural proofs. Assume that a first order language $\mathcal{L}$ is given. For simplicity we only consider here proofs in pure logic, i.e., without assumptions (axioms) on the functions and relations used.

$$(A \to B \to C) \to (A \to B) \to A \to C. \tag{1.1}$$

Assume $A \to B \to C$. To show: $(A \to B) \to A \to C$. So assume $A \to B$. To show: $A \to C$. So finally assume $A$. To show: $C$. We have $A$, by the last assumption. Hence also $B \to C$, by the first assumption, and $B$, using the next to last assumption. From $B \to C$ and $B$ we obtain $C$, as required. □

$$\forall_x(A \to B) \to A \to \forall_x B, \quad \text{if } x \notin \mathrm{FV}(A). \tag{1.2}$$

Assume $\forall_x(A \to B)$. To show: $A \to \forall_x B$. So assume $A$. To show: $\forall_x B$. Let $x$ be arbitrary; note that we have not made any assumptions on $x$. To show: $B$. We have $A \to B$, by the first assumption. Hence also $B$, by the second assumption. □

$$(A \to \forall_x B) \to \forall_x(A \to B), \quad \text{if } x \notin \mathrm{FV}(A). \tag{1.3}$$

Assume $A \to \forall_x B$. To show: $\forall_x(A \to B)$. Let $x$ be arbitrary; note that we have not made any assumptions on $x$. To show: $A \to B$. So assume $A$. To show: $B$. We have $\forall_x B$, by the first and second assumption. Hence also $B$. □

A characteristic feature of these proofs is that assumptions are introduced and eliminated again. At any point in time during the proof the free or "open" assumptions are known, but as the proof progresses, free assumptions may become cancelled or "closed" because of the implies-introduction rule.

We now reserve the word *proof* for the informal level; a formal representation of a proof will be called a *derivation*.

An intuitive way to communicate derivations is to view them as labelled trees. The labels of the inner nodes are the formulas derived at those points, and the labels of the leaves are formulas or terms. The labels of the nodes immediately above a node $\nu$ are the *premises* of the rule application, the formula at node $\nu$ is its *conclusion*. At the root of the tree we have the conclusion of the whole derivation. In natural deduction systems one works with *assumptions* affixed to some leaves of the tree; they can be *open* or else *closed*.

Any of these assumptions carries a *marker*. As markers we use *assumption variables* $\square_0, \square_1, \ldots$, denoted by $u, v, w, u_0, u_1, \ldots$. The (previous) variables will now often be called *object variables*, to distinguish them from assumption variables. If at a later stage (i.e., at a node below an assumption) the dependency on this assumption is removed, we record this by writing down the assumption variable. Since the same assumption can be used many times (this was the case in example (1.1)), the assumption marked with $u$ (and communicated by $u\colon A$) may appear many times. However, we insist that distinct assumption formulas must have distinct markers.

An inner node of the tree is understood as the result of passing form premises to a *conclusion*, as described by a given *rule*. The label of the node then contains in addition to the conclusion also the name of the rule. In some cases the rule binds or closes an assumption variable $u$ (and hence removes the dependency of all assumptions $u\colon A$ marked with that $u$). An application of the $\forall$-introduction rule similarly binds an object variable $x$ (and hence removes the dependency on $x$). In both cases the bound assumption or object variable is added to the label of the node.

**1.2.2. Introduction and elimination rules for $\to$ and $\forall$.** We now formulate the rules of natural deduction. First we have an assumption rule, that allows an arbitrary formula $A$ to be put down, together with a marker $u$:

$$u\colon A \quad \text{Assumption}$$

The other rules of natural deduction split into introduction rules (I-rules for short) and elimination rules (E-rules) for the logical connectives $\to$ and $\forall$. For implication $\to$ there is an introduction rule $\to^+ u$ and an elimination rule $\to^-$, also called *modus ponens*. The left premise $A \to B$ in $\to^-$ is called *major premise* (or *main* premise), and the right premise $A$ *minor premise* (or *side* premise). Note that with an application of the $\to^+ u$-rule *all* assumptions above it marked with $u\colon A$ are cancelled.

$$\frac{\begin{array}{c}[u\colon A]\\ \mid M\\ B\end{array}}{A \to B}\to^+ u \qquad \frac{\begin{array}{cc}\mid M & \mid N\\ A \to B & A\end{array}}{B}\to^-$$

For the universal quantifier $\forall$ there is an introduction rule $\forall^+ x$ and an elimination rule $\forall^-$, whose right premise is the term $r$ to be substituted. The rule $\forall^+ x$ is subject to the following *(Eigen-) variable condition*: The derivation $M$ of the premise $A$ should not contain any open assumption with $x$ as a free variable.

$$\frac{\begin{array}{c}\mid M\\ A\end{array}}{\forall_x A}\forall^+ x \qquad \frac{\begin{array}{cc}\mid M & \\ \forall_x A & r\end{array}}{A[x := r]}\forall^-$$

We now give derivations for the example formulas (1.1) – (1.3). Since in many cases the rule used is determined by the formula on the node, we suppress in such cases the name of the rule,

$$\dfrac{\dfrac{\dfrac{\dfrac{u\colon A \to B \to C \qquad w\colon A}{B \to C} \qquad \dfrac{v\colon A \to B \qquad w\colon A}{B}}{C}}{A \to C}\to^+ w}{(A \to B) \to A \to C}\to^+ v}{(A \to B \to C) \to (A \to B) \to A \to C}\to^+ u \tag{1.1}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{u\colon \forall_x(A \to B) \qquad x}{A \to B} \qquad v\colon A}{B}}{\forall_x B}\forall^+ x}{A \to \forall_x B}\to^+ v}{\forall_x(A \to B) \to A \to \forall_x B}\to^+ u \tag{1.2}$$

Note here that the variable condition is satisfied: $x$ is not free in $A$ (and also not free in $\forall_x(A \to B)$).

$$\dfrac{\dfrac{\dfrac{\dfrac{u\colon A \to \forall_x B \qquad v\colon A}{\forall_x B} \qquad x}{B}}{A \to B}\to^+ v}{\forall_x(A \to B)}\forall^+ x \Big/ (A \to \forall_x B) \to \forall_x(A \to B) \to^+ u \tag{1.3}$$

Here too the variable condition is satisfied: $x$ is not free in $A$.

**1.2.3. Axiom schemes for disjunction, conjunction, existence and falsity.** We follow the usual practice of considering all free variables in an axiom as universally quantified outside.

*Disjunction.* The introduction axioms are

$$\vee_0^+\colon A \to A \vee B$$
$$\vee_1^+\colon B \to A \vee B$$

and the elimination axiom is

$$\vee^-\colon (A \to C) \to (B \to C) \to A \vee B \to C.$$

*Conjunction.* The introduction axiom is

$$\wedge^+\colon A \to B \to A \wedge B$$

and the elimination axiom is

$$\wedge^-\colon (A \to B \to C) \to A \wedge B \to C.$$

*Existential Quantifier.* The introduction axiom is

$$\exists^+\colon A \to \exists_x A$$

and the elimination axiom is

$$\exists^-\colon \forall_x(A \to B) \to \exists_x A \to B \quad (x \text{ not free in } B).$$

*Falsity.* This example is somewhat extreme, since there is no introduction axiom; the elimination axiom is

$$\bot^-\colon \bot \to A.$$

In the literature this axiom is frequently called "ex-falso-quodlibet", written Efq. It clearly is derivable from its instances $\bot \to R\vec{x}$, for every relation symbol $R$.

*Equality.* The introduction axiom is

$$\mathrm{Eq}^+ \colon \mathrm{Eq}(x, x)$$

and the elimination axiom is

$$\mathrm{Eq}^- \colon \forall_x R(x, x) \to \mathrm{Eq}(x, y) \to R(x, y).$$

It is an easy exercise to show that the usual equality axioms can be derived.

All these axioms can be seen as special cases of a general scheme, that of an *inductively defined predicate*, which is defined by some introduction rules and one elimination rule. Later we will study this kind of definition in full generality. $\mathrm{Eq}(x, y)$ is a binary such predicate, $\bot$ is a nullary one, and $A \vee B$ another nullary one which however depends on the two parameter predicates $A$ and $B$.

The desire to follow this general pattern is also the reason that we have chosen our rather strange $\wedge^-$-axiom, instead of the more obvious $A \wedge B \to A$ and $A \wedge B \to B$ (which clearly are equivalent).

### 1.2.4. Minimal, intuitionistic and classical logic.

Definition ($\vdash$, $\vdash_i$). Consider $\to\forall\bot\vee\wedge\exists$-formulas.

(a) $A$ is called *derivable* (in *minimal logic*), written $\vdash A$, if there is a derivation of $A$ without free assumptions, from the axioms of Sec.1.2.3 using the rules from Sec.1.2.2, but *without using the ex-falso-quodlibet axiom, i.e., the elimination axiom $\bot^-$ for $\bot$*. A formula $B$ is called derivable from assumptions $A_1, \dots, A_n$, if there is a derivation (without $\bot^-$) of $B$ with free assumptions among $A_1, \dots, A_n$. Let $\Gamma$ be a (finite or infinite) set of formulas. We write $\Gamma \vdash B$ if the formula $B$ is derivable from finitely many assumptions $A_1, \dots, A_n \in \Gamma$.

(b) Let $\mathrm{Efq} := \{\, \forall\vec{x}(\bot \to R\vec{x}) \mid R \text{ relation symbol distinct from } \bot \,\}$. $A$ is called *derivable* in *intuitionistic logic*, written $\vdash_i A$, if in addition axioms from Efq are allowed. $\Gamma \vdash_i B$ is defined similarly.

We obtain *classical logic* by adding, for every relation symbol $R$ distinct from $\bot$, the *principle of indirect proof* expressed as the so-called "stability axiom" ($\mathrm{Stab}_R$): $\neg\neg R\vec{x} \to R\vec{x}$. Let

$$\mathrm{Stab} := \{\, \forall\vec{x}(\neg\neg R\vec{x} \to R\vec{x}) \mid R \text{ relation symbol distinct from } \bot \,\}.$$

For classical logic there is no need to use the full set of logical connectives: classical disjunction as well as the classical existential quantifier are defined, by $A \mathbin{\tilde{\vee}} B := \neg A \to \neg B \to \bot$ and $\tilde{\exists} x A := \neg\forall_x \neg A$. Moreover, when dealing with derivability we can even get rid of conjunction; this can be seen from the following lemma:

LEMMA (Elimination of $\wedge$). *For each formula $A$ built with the connectives $\to, \wedge, \forall$ there are formulas $A_1, \dots, A_n$ without $\wedge$ such that $\vdash A \leftrightarrow \bigwedge_{i=1}^{n} A_i$.*

PROOF. Induction on $A$. *Case* $R\vec{t}$. Take $n = 1$ and $A_1 := R\vec{t}$. *Case* $A \wedge B$. By IH (induction hypothesis), we have $A_1, \dots, A_n$ and $B_1, \dots, B_m$. Take $A_1, \dots, A_n, B_1, \dots, B_m$. *Case* $A \to B$. By IH, we have $A_1, \dots, A_n$ and $B_1, \dots, B_m$. For the sake of notational simplicity assume $n = 2$ and $m = 3$. Then

$$\begin{aligned}\vdash\ & (A_1 \wedge A_2 \to B_1 \wedge B_2 \wedge B_3)\\ & \leftrightarrow (A_1 \to A_2 \to B_1) \wedge (A_1 \to A_2 \to B_2) \wedge (A_1 \to A_2 \to B_3).\end{aligned}$$

*Case* $\forall_x A$. By IH for $A$, we have $A_1, \dots, A_n$. Take $\forall_x A_1, \dots, \forall_x A_n$, for

$$\vdash \forall_x \bigwedge_{i=1}^{n} A_i \leftrightarrow \bigwedge_{i=1}^{n} \forall_x A_i. \qquad \square$$

However, for the rest of this section we keep $\wedge$ in the language. The reason is that the notions introduced and the results obtained are slightly more general this way.

DEFINITION ($\vdash_c$). Consider $\to\forall\wedge\bot$-formulas. We call the formula $A$ *classically derivable* and write $\vdash_c A$ if there is a derivation of $A$ using $\wedge^{\pm}$-axioms and stability axioms from Stab. Similarly we define classical derivability from $\Gamma$ and write $\Gamma \vdash_c A$.

THEOREM (Stability, or Principle of Indirect Proof). *For every $\to\forall\wedge\bot$-formula $A$,*

$$\vdash_c \neg\neg A \to A.$$

PROOF. Induction on $A$. For simplicity, in the derivation to be constructed we leave out applications of $\to^+$ at the end. *Case* $R\vec{t}$ with $R$ distinct from $\bot$. Use $\mathrm{Stab}_R$. *Case* $\bot$. Observe that $\neg\neg\bot \to \bot = ((\bot \to \bot) \to \bot) \to \bot$. The desired derivation is

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

*Case* $A \to B$. Use $\vdash (\neg\neg B \to B) \to \neg\neg(A \to B) \to A \to B$; a derivation is

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

*Case* $\forall_x A$. Clearly it suffices to show $\vdash (\neg\neg A \to A) \to \neg\neg\forall_x A \to A$; a derivation is

$$\dfrac{\dfrac{u\colon \neg\neg A \to A \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{u_2\colon \forall_x A \qquad x}{A}}{\bot}\to^+ u_2}{\neg\forall_x A}}{\bot}\to^+ u_1}{\neg\neg A}}{A}}{v\colon \neg\neg\forall_x A}$$

The case $A \wedge B$ is left to the reader. □

Remark. The argument given proves a more general proposition (cf. Troelstra (1973, 1.10.8)): if $A$ is a Harrop formula (defined in Sec.1.1.3) constructed from decidable or doubly negated prime formulas, then $\vdash \neg\neg A \to A$.

Notice that $\vdash_c \bot \to A$, for stability is stronger:

$$\dfrac{\dfrac{\neg\neg A \to A \qquad \mid M_{\mathrm{Stab}}}{A} \qquad \dfrac{u\colon \bot}{\neg\neg A}\to^+ v^{\neg A}}{\bot \to A}\to^+ u$$

where $M_{\mathrm{Stab}}$ is the (classical) derivation of stability.

Notice also that even for the $\to \bot$-fragment the inclusion of minimal logic in intuitionistic logic, and of the latter in classical logic are proper. Examples are

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

Non-derivability can be proved by means of countermodels, using a semantic characterization of derivability; this will be done later. $\vdash_i \bot \to P$ is obvious, and the Peirce formula $((P \to Q) \to P) \to P$ can be derived in minimal logic from $\bot \to Q$ and $\neg\neg P \to P$, hence is derivable in classical logic.

**1.2.5. Negative translation.** We embed classical logic into minimal logic, via the so-called negative (or Gödel-Gentzen) translation. A formula $A$ is called *negative*, if every atomic formula of $A$ distinct from $\bot$ occurs negated, and $A$ does not contain $\vee$, $\exists$.

Lemma. *For negative* $A$, $\vdash \neg\neg A \to A$.

Proof. This follows from the proof of the stability theorem, using $\vdash \neg\neg\neg R\vec{t} \to \neg R\vec{t}$. □

Again we consider $\to\forall\wedge\bot$-formulas only.

DEFINITION (Negative translation $^g$ of Gödel-Gentzen).

$$\begin{aligned}
(R\vec{t})^g &:= \neg\neg R\vec{t} \quad (R \text{ distinct from } \bot),\\
\bot^g &:= \bot,\\
(A \wedge B)^g &:= A^g \wedge B^g,\\
(A \to B)^g &:= A^g \to B^g,\\
(\forall_x A)^g &:= \forall_x A^g.
\end{aligned}$$

THEOREM. *For all $\to\forall\wedge\bot$-formulas $A$,*
(a) $\vdash_c A \leftrightarrow A^g$,
(b) $\Gamma \vdash_c A$ *if and only if* $\Gamma^g \vdash A^g$, *where* $\Gamma^g := \{\, B^g \mid B \in \Gamma \,\}$.

PROOF. (a). The claim follows from the fact that $\vdash_c$ is compatible with equivalence. (b). $\Leftarrow$. Obvious $\Rightarrow$. By induction on the classical derivation. For a stability assumption $\neg\neg R\vec{t} \to R\vec{t}$ we have $(\neg\neg R\vec{t} \to R\vec{t})^g = \neg\neg\neg\neg R\vec{t} \to \neg\neg R\vec{t}$, and this is easily derivable. *Case* $\to^+$. Assume

$$\dfrac{\begin{array}{c}[u\colon A]\\ \mathcal{D}\\ B\end{array}}{A \to B}\to^+ u$$

Then we have by IH

$$\begin{array}{c} u\colon A^g\\ \mathcal{D}^g\\ B^g\end{array} \quad \text{hence} \quad \dfrac{\begin{array}{c}[u\colon A^g]\\ \mathcal{D}^g\\ B^g\end{array}}{A^g \to B^g}\to^+ u$$

*Case* $\to^-$. Assume

$$\dfrac{\begin{array}{cc}\mathcal{D}_0 & \mathcal{D}_1\\ A \to B & A\end{array}}{B}$$

Then we have by IH

$$\begin{array}{cc}\mathcal{D}_0^g & \mathcal{D}_1^g\\ A^g \to B^g & A^g\end{array} \quad \text{hence} \quad \dfrac{\begin{array}{cc}\mathcal{D}_0^g & \mathcal{D}_1^g\\ A^g \to B^g & A^g\end{array}}{B^g}$$

The other cases are treated similarly. □

COROLLARY (Embedding of classical logic). *For negative $A$, $\vdash_c A$ if and only if $\vdash A$.*

PROOF. By the theorem we have $\vdash_c A$ if and only if $\vdash A^g$. Since $A$ is negative, every atom distinct from $\bot$ in $A$ must occur negated, and hence in $A^g$ it must appear in threefold negated form (as $\neg\neg\neg R\vec{t}$). The claim follows from $\vdash \neg\neg\neg R\vec{t} \leftrightarrow \neg R\vec{t}$. □

Since every formula is classically equivalent to a negative formula, we have achieved an embedding of classical logic into minimal logic.

Note that $\not\vdash \neg\neg P \to P$ (as we shall show later). The corollary therefore does not hold for all formulas $A$.

**1.2.6. Formulas implying their negative translation.** We introduce a further observation (due to Leivant; see Troelstra and van Dalen (1988, Ch.2, Sec.3)) which will be useful for program extraction from classical proofs. There it will be necessary to actually transform a given classical derivation $\vdash_c A$ into a minimal logic derivation $\vdash A^g$. In particular, for every assumption constant $C$ used in the given derivation we have to provide a derivation of $C^g$. Now for some formulas $S$ – the so-called *spreading* formulas – this is immediate, for we can derive $S \to S^g$, and hence can use the original assumption constant.

First notice that our formulas may contain *predicate variables* denoted by $X$, which are place holders for comprehension terms, i.e., formulas with distinguished variables. We use the obvious notation $A[X := \{\, \vec{x} \mid B \,\}]$ or shortly $A[\{\, \vec{x} \mid B \,\}]$ or even $A[B]$ for substitution for predicate variables. Clearly the Gödel-Gentzen translation of $X\vec{t}$ is $\neg\neg X\vec{t}$.

Notice also that an assumption constant may be viewed as consisting of an uninstantiated formula (e.g., $X0 \to \forall_n(Xn \to X(n+1)) \to \forall_n Xn$ for induction) together with a substitution of comprehension terms for predicate variables (e.g., $X \mapsto \{\, n \mid n < n+1 \,\}$). Then in order to immediately obtain a derivation of $C^g$ for $C$ an assumption constant it suffices to know that its *uninstantiated* formula $S$ is spreading, for then we generally have $\vdash S[\vec{A}^g] \to S[\vec{A}]^g$ (see the theorem below) and hence can use the same assumption constant with a different substitution.

We define *spreading* formulas $S$, *wiping* formulas $W$ and *isolating* formulas $I$ inductively.

$$\begin{aligned} S &::= \bot \mid R\vec{t} \mid X\vec{t} \mid S \wedge S \mid I \to S \mid \forall_x S, \\ W &::= \bot \mid X\vec{t} \mid W \wedge W \mid S \to W \mid \forall_x W, \\ I &::= R\vec{t} \mid W \mid I \wedge I. \end{aligned}$$

Let $\mathcal{S}$ ($\mathcal{W}$,$\mathcal{I}$) be the class of spreading (wiping, isolating) formulas.

THEOREM.

$$\begin{aligned} &\vdash S[\vec{A}^g] \to S[\vec{A}]^g && \textit{for every spreading formula } S, \\ &\vdash W[\vec{A}]^g \to W[\vec{A}^g] && \textit{for every wiping formula } W, \\ &\vdash I[\vec{A}]^g \to \neg\neg I[\vec{A}^g] && \textit{for every isolating formula } I. \end{aligned}$$

*We assume here that all occurrences of predicate variables are substituted.*

PROOF. By induction on the simultaneous generation of $\mathcal{S}$, $\mathcal{W}$ and $\mathcal{I}$. We write $S^g$ for $S[\vec{A}]^g$ and $S$ for $S[\vec{A}^g]$, and similarly for $W$ and $I$.

*Case* $\bot \in \mathcal{S}$. We must show $\vdash \bot \to \bot^g$. Take $\lambda u^{\bot} u$.

*Case* $R\vec{t} \in \mathcal{S}$. We must show $\vdash R\vec{t} \to \neg\neg R\vec{t}$. Take $\lambda u^{R\vec{t}} \lambda v^{\neg R\vec{t}}.vu$.

*Case* $X\vec{t} \in \mathcal{S}$, with $X$ substituted by $\{\, \vec{x} \mid A \,\}$. We must show $\vdash A^g[\vec{t}] \to A^g[\vec{t}]$, which is trivial.

*Case* $S_1 \wedge S_2 \in \mathcal{S}$. We must show $\vdash S_1 \wedge S_2 \to S_1^g \wedge S_2^g$. Take

$$\dfrac{\dfrac{\dfrac{\text{IH}}{S_1 \to S_1^g} \quad \dfrac{u\colon S_1 \wedge S_2}{S_1}}{S_1^g} \qquad \dfrac{\dfrac{\text{IH}}{S_2 \to S_2^g} \quad \dfrac{u\colon S_1 \wedge S_2}{S_2}}{S_2^g}}{S_1^g \wedge S_2^g}$$

*Case* $I \to S \in \mathcal{S}$. We must show $\vdash (I \to S) \to I^g \to S^g$. Recall that $\vdash \neg\neg S^g \to S^g$ by the Stability Lemma, because $S^g$ is negative. Take

$$\dfrac{\dfrac{\text{Stab}}{\neg\neg S^g \to S^g} \quad \dfrac{\dfrac{\bot}{\neg\neg S^g} \to^+ w_1}{S^g}}{S^g}$$

where the derivation of $\neg\neg S^g$ is

$$\dfrac{\dfrac{\dfrac{\text{IH}}{I^g \to \neg\neg I} \quad v\colon I^g}{\neg\neg I} \quad \dfrac{\dfrac{\bot}{\neg I} \to^+ w_2}{\neg\neg S^g}}{\bot}$$

and the derivation of $\neg I$ is

$$\dfrac{\dfrac{\dfrac{\text{IH}}{S \to S^g} \quad \dfrac{u\colon I \to S \qquad w_2\colon I}{S}}{S^g} \quad w_1\colon \neg S^g}{\bot}$$

*Case* $\forall_x S \in \mathcal{S}$. We must show $\vdash \forall_x S \to \forall_x S^g$. Take

$$\dfrac{\dfrac{\text{IH}}{S \to S^g} \quad \dfrac{u\colon \forall_x S \qquad x}{S}}{S^g}$$

*Case* $\bot \in \mathcal{W}$. We must show $\vdash \bot^g \to \bot$. Take $\lambda u^{\bot} u$.

*Case* $X\vec{t} \in \mathcal{W}$, with $X$ substituted by $\{\, \vec{x} \mid A \,\}$. We must show $\vdash A^g[\vec{t}] \to A^g[\vec{t}]$, which is trivial.

*Case* $W_1 \wedge W_2 \in \mathcal{W}$. We must show $\vdash W_1^g \wedge W_2^g \to W_1 \wedge W_2$. Take

$$\dfrac{\dfrac{\dfrac{\text{IH}}{W_1^g \to W_1} \quad \dfrac{u\colon W_1^g \wedge W_2^g}{W_1^g}}{W_1} \qquad \dfrac{\dfrac{\text{IH}}{W_2^g \to W_2} \quad \dfrac{u\colon W_1^g \wedge W_2^g}{W_2^g}}{W_2}}{W_1 \wedge W_2}$$

*Case* $S \to W \in \mathcal{W}$. We must show $\vdash (S^g \to W^g) \to S \to W$. Take

$$\dfrac{\dfrac{\text{IH}}{W^g \to W} \quad \dfrac{u\colon S^g \to W^g \qquad \dfrac{\dfrac{\text{IH}}{S \to S^g} \quad v\colon S}{S^g}}{W^g}}{W}$$

*Case* $\forall_x W \in \mathcal{W}$. We must show $\vdash \forall_x W^g \to \forall_x W$. Take

$$\dfrac{\dfrac{\text{IH}}{W^g \to W} \qquad \dfrac{u\colon \forall_x W^g}{W^g}\, x}{W}$$

*Case* $R\vec{t} \in \mathcal{I}$. We must show $\vdash \neg\neg R\vec{t} \to \neg\neg R\vec{t}$, which is trivial.

*Case* $W \in \mathcal{I}$. We must show $\vdash W^g \to \neg\neg W$, which trivially follows from the IH $\vdash W^g \to W$. Take

$$\dfrac{v\colon \neg W \qquad \dfrac{\dfrac{\text{IH}}{W^g \to W} \qquad u\colon W^g}{W}}{\bot}$$

*Case* $I_1 \wedge I_2 \in \mathcal{I}$. We must show $\vdash I_1^g \wedge I_2^g \to \neg\neg(I_1 \wedge I_2)$. Take

$$\dfrac{\dfrac{\dfrac{\text{IH}}{I_2^g \to \neg\neg I_2} \qquad \dfrac{I_1^g \wedge I_2^g}{I_2^g}}{\neg\neg I_2} \qquad \dfrac{\dfrac{\dfrac{\text{IH}}{I_1^g \to \neg\neg I_1} \qquad \dfrac{I_1^g \wedge I_2^g}{I_1^g}}{\neg\neg I_1} \qquad \dfrac{\dfrac{\neg(I_1 \wedge I_2) \qquad \dfrac{I_1 \quad I_2}{I_1 \wedge I_2}}{\bot}}{\neg I_1}}{\bot}}{\bot}$$

□

## 1.3. Normalization

We show that every derivation can be brought into a normal form. A derivation in normal form does not make "detours", or more precisely, it cannot occur that an elimination rule immediately follows an introduction rule. The shape of derivations in normal form will be analyzed. In particular, we will prove the subformula property, which says that every formula in a normal derivation is a subformula of the goal formula or else of an assumption. Moreover, we also consider "long" normal forms.

**1.3.1. Conversion.** A conversion eliminates a detour in a derivation, i.e., an elimination immediately following an introduction. We consider the following conversions:

*→-conversion.*

$$\dfrac{\dfrac{\begin{array}{c}[u\colon A]\\ \mid M\\ B\end{array}}{A \to B}\,{\to^+}u \qquad \begin{array}{c}\mid N\\ A\end{array}}{B}\,{\to^-} \quad \mapsto \quad \begin{array}{c}\mid N\\ A\\ \mid M\\ B\end{array}$$

$\forall$-*conversion.*

$$\dfrac{\dfrac{\begin{matrix}|\,M\\ A\end{matrix}}{\forall_x A}\,\forall^+x \qquad r}{A[x:=r]}\,\forall^- \quad\mapsto\quad \begin{matrix}|\,M'\\ A[x:=r]\end{matrix}$$

**1.3.2. Derivations as terms.** It will be convenient to write derivations as terms, where the derived formula is viewed as the type of the term. This representation is known under the name *Curry-Howard correspondence*.

We give an inductive definition of derivation terms in Table 1 on page 19, where for clarity we have written the corresponding derivations to the left. For the universal quantifier $\forall$ there is an introduction rule $\forall^+x$ and an elimination rule $\forall^-$, whose right premise is the term $r$ to be substituted. The rule $\forall^+x$ is subject to the following *(Eigen-) variable condition*: The derivation term $M$ of the premise $A$ should not contain any open assumption with $x$ as a free variable.

**1.3.3. Reduction, normal form.** Every derivation term carries a formula as its type. However, we shall usually leave these formulas implicit and write derivation terms without them.

Notice that every derivation term can be written uniquely in one of the forms

$$u\vec{M} \mid \lambda v M \mid (\lambda v M)N\vec{L},$$

where $u$ is an assumption variable or assumption constant, $v$ is an assumption variable or object variable, and $M$, $N$, $L$ are derivation terms or object terms.

Here the final form is not normal: $(\lambda v M)N\vec{L}$ is called *$\beta$-redex* (for "reducible expression"). The *conversion rule* is

$$(\lambda v M)N \mapsto_\beta M[v := N].$$

Notice that in a substitution $M[v := N]$ with $M$ a derivation term and $v$ an object variable, one also needs to substitute in the formulas of $M$.

The *closure* of the conversion relation $\mapsto_\beta$ is defined by

- If $M \mapsto_\beta M'$, then $M \to M'$.
- If $M \to M'$, then also $MN \to M'N$, $NM \to NM'$, $\lambda v M \to \lambda v M'$ (*inner reductions*).

So $M \to N$ means that $M$ *reduces in one step to* $N$, i.e., $N$ is obtained from $M$ by replacement of (an occurrence of) a redex $M'$ of $M$ by a conversum $M''$ of $M'$, i.e., by a single conversion. The relation $\to^+$ ("*properly reduces to*") is the transitive closure of $\to$ and $\to^*$ ("*reduces to*") is the reflexive and transitive closure of $\to$. The relation $\to^*$ is said to be the notion

| derivation | term |
|---|---|
| $u\colon A$ | $u^A$ |
| $\dfrac{\begin{array}{c}[u\colon A]\\ \mid M\\ B\end{array}}{A \to B} \to^+ u$ | $(\lambda u^A M^B)^{A\to B}$ |
| $\dfrac{\begin{array}{cc}\mid M & \mid N\\ A \to B & A\end{array}}{B} \to^-$ | $(M^{A\to B} N^A)^B$ |
| $\dfrac{\begin{array}{c}\mid M\\ A\end{array}}{\forall_x A} \forall^+ x$ (with var.cond.) | $(\lambda x M^A)^{\forall_x A}$ (with var.cond.) |
| $\dfrac{\begin{array}{cc}\mid M & \\ \forall_x A & r\end{array}}{A[x := r]} \forall^-$ | $(M^{\forall_x A} r)^{A[x:=r]}$ |

TABLE 1. Derivation terms for $\to$ and $\forall$

of reduction *generated* by $\mapsto$. $\leftarrow$, $\leftarrow^+$, $\leftarrow^*$ are the relations converse to $\to, \to^+, \to^*$, respectively.

A term $M$ is *in normal form*, or $M$ is *normal*, if $M$ does not contain a redex. $M$ *has a normal form* if there is a normal $N$ such that $M \to^* N$.

A *reduction sequence* is a (finite or infinite) sequence $M_0 \to M_1 \to M_2 \ldots$ such that $M_i \to M_{i+1}$, for all $i$.

Finite reduction sequences are partially ordered under the initial part relation; the collection of finite reduction sequences starting from a term $M$ forms a tree, the *reduction tree* of $M$. The branches of this tree may be identified with the collection of all infinite and all terminating finite reduction sequences.

A term is *strongly normalizing* if its reduction tree is finite.

EXAMPLE.

$$\begin{aligned}(\lambda x\lambda y\lambda z.xz(yz))(\lambda u\lambda v\, u)(\lambda u'\lambda v'\, u') &\to\\ (\lambda y\lambda z.(\lambda u\lambda v\, u)z(yz))(\lambda u'\lambda v'\, u') &\to\\ (\lambda y\lambda z.(\lambda v\, z)(yz))(\lambda u'\lambda v'\, u') &\to\\ (\lambda y\lambda z\, z)(\lambda u'\lambda v'\, u') &\to \lambda z\, z.\end{aligned}$$

LEMMA (Substitutivity of $\to$). (a) *If* $M \to M'$, *then* $M[v := N] \to M'[v := N]$.
(b) *If* $N \to N'$, *then* $M[v := N] \to^* M[v := N']$.

PROOF. (a) is proved by induction on $M \to M'$; (b) by induction on $M$. Notice that the reason for $\to^*$ in (b) is the fact that $v$ may have many occurrences in $M$. □

**1.3.4. Strong normalization.** We show that every term is strongly normalizing.

To this end, define by recursion on $k$ a relation $\mathrm{sn}(M, k)$ between terms $M$ and natural numbers $k$ with the intention that $k$ is an upper bound on the number of reduction steps up to normal form.

$$\begin{aligned}\mathrm{sn}(M, 0) &:\Longleftrightarrow M \text{ is in normal form,}\\ \mathrm{sn}(M, k+1) &:\Longleftrightarrow \mathrm{sn}(M', k) \text{ for all } M' \text{ such that } M \to M'.\end{aligned}$$

Clearly a term is strongly normalizing if there is a $k$ such that $\mathrm{sn}(M, k)$. We first prove some closure properties of the relation sn.

LEMMA (Properties of sn). (a) *If* $\mathrm{sn}(M, k)$, *then* $\mathrm{sn}(M, k+1)$.
(b) *If* $\mathrm{sn}(MN, k)$, *then* $\mathrm{sn}(M, k)$.
(c) *If* $\mathrm{sn}(M_i, k_i)$ *for* $i = 1 \dots n$, *then* $\mathrm{sn}(uM_1 \dots M_n, k_1 + \dots + k_n)$.
(d) *If* $\mathrm{sn}(M, k)$, *then* $\mathrm{sn}(\lambda v M, k)$.
(e) *If* $\mathrm{sn}(M[v := N]\vec{L}, k)$ *and* $\mathrm{sn}(N, l)$, *then* $\mathrm{sn}((\lambda v M)N\vec{L}, k + l + 1)$.

PROOF. (a). Induction on $k$. Assume $\mathrm{sn}(M, k)$. We show $\mathrm{sn}(M, k+1)$. So let $M'$ with $M \to M'$ be given; because of $\mathrm{sn}(M, k)$ we must have $k > 0$. We have to show $\mathrm{sn}(M', k)$. Because of $\mathrm{sn}(M, k)$ we have $\mathrm{sn}(M', k-1)$, hence by IH $\mathrm{sn}(M', k)$.

(b). Induction on $k$. Assume $\mathrm{sn}(MN, k)$. We show $\mathrm{sn}(M, k)$. In case $k = 0$ the term $MN$ is normal, hence also $M$ is normal and therefore $\mathrm{sn}(M, 0)$. So let $k > 0$ and $M \to M'$; we have to show $\mathrm{sn}(M', k-1)$. From $M \to M'$ we have $MN \to M'N$. Because of $\mathrm{sn}(MN, k)$ we have by definition $\mathrm{sn}(M'N, k-1)$, hence $\mathrm{sn}(M', k-1)$ by IH.

(c). Assume $\mathrm{sn}(M_i, k_i)$ for $i = 1 \dots n$. We show $\mathrm{sn}(uM_1 \dots M_n, k)$ with $k := k_1 + \cdots + k_n$. Again we employ induction on $k$. In case $k = 0$ all $M_i$ are normal, hence also $uM_1 \dots M_n$. So let $k > 0$ and $uM_1 \dots M_n \to M'$. Then $M' = uM_1 \dots M_i' \dots M_n$ with $M_i \to M_i'$; We have to show $\mathrm{sn}(uM_1 \dots M_i' \dots M_n, k-1)$. Because of $M_i \to M_i'$ and $\mathrm{sn}(M_i, k_i)$ we have $k_i > 0$ and $\mathrm{sn}(M_i', k_i - 1)$, hence $\mathrm{sn}(uM_1 \dots M_i' \dots M_n, k-1)$ by IH.

(d). Assume $\mathrm{sn}(M, k)$. We have to show $\mathrm{sn}(\lambda vM, k)$. Use induction on $k$. In case $k = 0$ $M$ is normal, hence $\lambda vM$ is normal, hence $\mathrm{sn}(\lambda vM, 0)$. So let $k > 0$ and $\lambda vM \to L$. Then $L$ has the form $\lambda vM'$ with $M \to M'$. So $\mathrm{sn}(M', k-1)$ by definition, hence $\mathrm{sn}(\lambda vM', k)$ by IH.

(e). Assume $\mathrm{sn}(M[v := N]\vec{L}, k)$ and $\mathrm{sn}(N, l)$. We need to show that $\mathrm{sn}((\lambda vM)N\vec{L}, k + l + 1)$. We use induction on $k + l$. In case $k + l = 0$ the term $N$ and $M[v := N]\vec{L}$ are normal, hence also $M$ and all $L_i$. So there is exactly one term $K$ such that $(\lambda vM)N\vec{L} \to K$, namely $M[v := N]\vec{L}$, and this $K$ is normal. Now let $k + l > 0$ and $(\lambda vM)N\vec{L} \to K$. We have to show $\mathrm{sn}(K, k + l)$.

*Case* $K = M[v := N]\vec{L}$, i.e., we have a head conversion. From $\mathrm{sn}(M[v := N]\vec{L}, k)$ we obtain $\mathrm{sn}(M[v := N]\vec{L}, k + l)$ by (a).

*Case* $K = (\lambda vM')N\vec{L}$ with $M \to M'$. Then we have $M[v := N]\vec{L} \to M'[v := N]\vec{L}$. Now $\mathrm{sn}(M[v := N]\vec{L}, k)$ implies $k > 0$ and $\mathrm{sn}(M'[v := N]\vec{L}, k-1)$. The IH yields $\mathrm{sn}((\lambda vM')N\vec{L}, k - 1 + l + 1)$.

*Case* $K = (\lambda vM)N'\vec{L}$ with $N \to N'$. Now $\mathrm{sn}(N, l)$ implies $l > 0$ and $\mathrm{sn}(N', l-1)$. The IH yields $\mathrm{sn}((\lambda vM)N'\vec{L}, k + l - 1 + 1)$, since $\mathrm{sn}(M[v := N']\vec{L}, k)$ by (a),

*Case* $K = (\lambda vM)N\vec{L}'$ with $L_i \to L_i'$ for some $i$ and $L_j = L_j'$ for $j \neq i$. Then we have $M[v := N]\vec{L} \to M[v := N]\vec{L}'$. Now $\mathrm{sn}(M[v := N]\vec{L}, k)$ implies $k > 0$ and $\mathrm{sn}(M[v := N]\vec{L}', k-1)$. The IH yields $\mathrm{sn}((\lambda vM)N\vec{L}', k - 1 + l + 1)$. □

The essential idea of the strong normalization proof is to view the last three closure properties of sn from the preceding lemma without the information on the bounds as an inductive definition of a new set SN:

$$\frac{\vec{M} \in \mathrm{SN}}{u\vec{M} \in \mathrm{SN}}\ (\mathrm{Var}) \qquad \frac{M \in \mathrm{SN}}{\lambda vM \in \mathrm{SN}}\ (\lambda) \qquad \frac{M[v := N]\vec{L} \in \mathrm{SN} \qquad N \in \mathrm{SN}}{(\lambda vM)N\vec{L} \in \mathrm{SN}}\ (\beta)$$

Corollary. *For every term* $M \in \mathrm{SN}$ *there is a* $k \in \mathbb{N}$ *such that* $\mathrm{sn}(M, k)$. *Hence every term* $M \in \mathrm{SN}$ *is strongly normalizing*

Proof. By induction on $M \in \mathrm{SN}$, using the previous lemma. □

In what follows we shall show that *every* term is in SN and hence is strongly normalizing. Given the definition of SN we only have to show that SN is closed under application. In order to prove this we must prove simultaneously the closure of SN under substitution.

THEOREM (Properties of SN). *For all formulas $A$, derivation terms $M \in \mathrm{SN}$ and $N^A \in \mathrm{SN}$ the following holds.*

(a) $M[v := N] \in \mathrm{SN}$.
(a') $M[x := r] \in \mathrm{SN}$.
(b) *Suppose $M$ derives $A \to B$. Then $MN \in \mathrm{SN}$.*
(b') *Suppose $M$ derives $\forall_x A$. Then $Mr \in \mathrm{SN}$.*

PROOF. By course-of-values induction on $\mathrm{dp}(A)$, with a side induction on $M \in \mathrm{SN}$. Let $N^A \in \mathrm{SN}$. We distinguish cases on the form of $M$.

*Case* $u\vec{M}$ by (Var) from $\vec{M} \in \mathrm{SN}$. (a). The SIH(a) (SIH means side induction hypothesis) yields $M_i[v := N] \in \mathrm{SN}$ for all $M_i$ from $\vec{M}$. In case $u \neq v$ we immediately have $(u\vec{M})[v := N] \in \mathrm{SN}$. Otherwise we need $N\vec{M}[v := N] \in \mathrm{SN}$. But this follows by multiple applications of IH(b), since every $M_i[v := N]$ derives a subformula of $A$ with smaller depth. (a'). Similar, and simpler. (b), (b'). Use (Var) again.

*Case* $\lambda v M$ by $(\lambda)$ from $M \in \mathrm{SN}$. (a), (a'). Use $(\lambda)$ again. (b). Our goal is $(\lambda v M)N \in \mathrm{SN}$. By $(\beta)$ it suffices to show $M[v := N] \in \mathrm{SN}$ and $N \in \mathrm{SN}$. The latter holds by assumption, and the former by SIH(a). (b'). Similar, and simpler.

*Case* $(\lambda w M)K\vec{L}$ by $(\beta)$ from $M[w := K]\vec{L} \in \mathrm{SN}$ and $K \in \mathrm{SN}$. (a). The SIH(a) yields $M[v := N][w := K[v := N]]\vec{L}[v := N] \in \mathrm{SN}$ and $K[v := N] \in \mathrm{SN}$, hence $(\lambda w M[v := N])K[v := N]\vec{L}[v := N] \in \mathrm{SN}$ by $(\beta)$. (a'). Similar, and simpler. (b), (b'). Use $(\beta)$ again. □

REMARK (Arithmetical comprehension). The theorem continues to hold if we allow quantification over predicate variables, but restrict the rule $\forall^-$ to comprehension terms with quantification over object variables only. This restriction is known under the name *arithmetical comprehension*. The proof above then is by a main induction on the depth of nesting of predicate quantifiers, a first side induction on $\mathrm{dp}(A)$ (which disregards predicate quantifiers), and a second side induction on $M \in \mathrm{SN}$.

COROLLARY. *For every term we have $M \in \mathrm{SN}$; in particular every term $M$ is strongly normalizing.*

PROOF. Induction on the (first) inductive definition of derivation terms $M$. In cases $u$ and $\lambda v M$ the claim follows from the definition of SN, and in case $MN$ it follows from the preceding theorem. □

**1.3.5. The structure of normal derivations.** To analyze normal derivations, it will be useful to introduce the notion of a *track* in a proof tree, which makes sense for non-normal derivations as well.

DEFINITION. A *track* of a derivation $M$ is a sequence of f.o.'s $A_0, \ldots, A_n$ such that

(a) $A_0$ is a top f.o. in $M$ (possible discharged by an application of an $\to^-$-rule);
(b) $A_i$ for $i < n$ is not the minor premise of an instance of $\to^-$, and $A_{i+1}$ is directly below $A_i$;
(c) $A_n$ is either the minor premise of an instance of $\to^-$, or the conclusion of $M$.

The *track of order* 0, or *main track*, in a derivation is the (unique) track ending in the conclusion of the whole derivation. A *track of order* $n+1$ is a track ending in the minor premise of an $\to^-$-application, with major premise belonging to a track of order $n$.

LEMMA. *In a derivation each formula occurrence belongs to some track.*

PROOF. By induction on derivations. □

Now consider a normal derivation $M$. Since by normality an E-rule cannot have the conclusion of an I-rule as its major premise, the E-rules have to precede the I-rules in a track, so the following is obvious: a track may be divided into an E-part, say $A_0, \ldots, A_{i-1}$, a minimal formula $A_i$, and an I-part $A_{i+1}, \ldots, A_n$. In the E-part all rules are E-rules; in the I-part all rules are I-rules; $A_i$ is the conclusion of an E-rule and, if $i < n$, a premise of an I-rule. Tracks are pieces of branches of the tree with successive f.o.'s in the subformula relationship: either $A_{i+1}$ is a subformula of $A_i$ or vice versa. As a result, all formulas in a track $A_0, \ldots, A_n$ are subformulas of $A_0$ or of $A_n$; and from this, by induction on the order of tracks, we see that every formula in $M$ is a subformula either of an open assumption or of the conclusion. To summarize:

THEOREM. *In a normal derivation each formula is a subformula of either the end formula or else an assumption formula.*

PROOF. We prove this for tracks of order $n$, by induction on $n$. □

REMARK (Conservativeness of predicate quantifiers). Again the theorem continues to hold if we allow quantification over predicate variables, but restrict the rule $\forall^-$ to comprehension terms with quantification over object variables only. But notice that *every* formula with quantification over object variables only is a subformula of $\forall_P P$, so the notion of a subformula is of limited use here. However, we can conclude that the extension of the logic to predicate quantifiers is conservative over the original one.

**1.3.6. Long normal forms.** *$\eta$-conversion* is defined by

$$\lambda x.Mx \mapsto_\eta M \quad \text{if } x \notin \mathrm{FV}(M) \text{ and } M \text{ is non-introduced.}$$

It can easily be analyzed. *$\eta$-expansion* is supposed to reverse $\eta$-conversion. Unfortunately this can lead to reduction loops when combined with $\beta$-reduction:

$$MN \to_{\eta\uparrow} (\lambda x.Mx)N \to_\beta MN \qquad \text{or} \qquad \lambda xM \to_{\eta\uparrow} \lambda y.(\lambda xM)y \to_\beta \lambda xM.$$

Thus we have to prevent terms in applicative positions and abstractions from being expanded. This is achieved if we define $\to_{\eta\uparrow}$ from the conversion rule

$$M^{\rho\Rightarrow\sigma} \mapsto_{\eta\uparrow} \lambda x^\rho.Mx \qquad \text{if } M \text{ is non-introduced}$$

by means of term closure for non-applicative positions, i.e.,

- If $M \mapsto_{\eta\uparrow} M'$, then $M \to_{\eta\uparrow} M'$.
- If $M \to_{\eta\uparrow} M'$, then also $NM \to_{\eta\uparrow} NM'$, $\lambda vM \to_{\eta\uparrow} \lambda vM'$.

The following can be seen easily:

(a) $\eta\uparrow$-reduction does not create any new $\beta$-redexes.

(b) Normal forms with respect to $\to_{\eta\uparrow}$ can be characterized by the grammar

$$M ::= (x\vec{M})^\iota \mid \lambda xM \mid ((\lambda xM)N\vec{N})^\iota.$$

(c) $\beta\eta\uparrow$-normal forms (also called *long normal forms*) are obtained if we omit the last rule.

Define $\eta_\rho(M^\rho)$ (*outer $\eta$-expansion*) and its expansion height $\mu_\rho \in \mathbb{N}$ by

$$\eta_\iota(M) := M, \qquad \mu_\iota := 0,$$
$$\eta_{\rho\Rightarrow\sigma}(M) := \lambda x^\rho \eta_\sigma(M\eta_\rho(x)). \quad \mu_{\rho\Rightarrow\sigma} := 1 + \mu_\rho + \mu_\sigma.$$

Lemma.

(1.4) $M \to_\beta M' \Rightarrow \eta(M) \to_\beta \eta(M')$.

(1.5) $M^\rho \to_{\eta\uparrow}^{\mu_\rho} \eta_\rho(M)$ *if $M$ is non-introduced.*

(1.6) *If $\vec{M}, M, N$ are $\eta\uparrow$-normal, then also $\eta(x\vec{M})$ and $\eta((\lambda xM)N\vec{M})$.*

(1.7) $\eta(M)\vec{N} \to_\beta^* \eta(M\eta(\vec{N}))$.

(1.8) $\eta(\eta(M)) \to_\beta^* \eta(M)$.

Proof. (1.4) is clear. (1.5)-(1.8) are proved by induction on the type $\rho$ of $\eta_\rho$. For (1.5) we have

$$\begin{aligned} M^{\rho\Rightarrow\sigma} &\to_{\eta\uparrow} \lambda x^\rho.(Mx)^\sigma \\ &\to_{\eta\uparrow}^{\mu_\rho} \lambda x.M\eta_\rho(x) \qquad \text{by IH} \\ &\to_{\eta\uparrow}^{\mu_\sigma} \lambda x\, \eta(M\eta(x)) \quad \text{by IH.} \end{aligned}$$

For (1.6), e.g. $\eta_{\rho\Rightarrow\sigma}(x\vec{M}) = \lambda y^{\rho}\eta_{\sigma}(x\vec{M}\eta_{\rho}(y)) \in \mathrm{Nf}_{\eta\uparrow}$ by $\mathrm{IH}_{\rho}$ and $\mathrm{IH}_{\sigma}$. For (1.7) we have

$$\begin{aligned}\eta_{\rho\Rightarrow\sigma}(M)N\vec{N} &= (\lambda x\,\eta(M\eta(x)))N\vec{N}\\ &\to_{\beta} \eta(M\eta(N))\vec{N}\\ &\to_{\beta}^{*} \eta(M\eta(N)\eta(\vec{N})) \qquad \text{by IH,}\end{aligned}$$

and for (1.8)

$$\begin{aligned}\eta(\eta(M^{\rho\Rightarrow\sigma})) &= \lambda x\,\eta(\eta(M)\eta(x))\\ &= \lambda x\,\eta([\lambda y\,\eta(M\eta(y))]\eta(x))\\ &\to_{\beta} \lambda x\,\eta\eta(M\eta\eta(x))\\ &\to_{\beta}^{*} \lambda x\,\eta(M\eta(x)) \qquad\qquad \text{by IH.}\end{aligned}$$

This concludes the proof. □

Define the *η-expansion* $\exp(M)$ of $M$ by

$$\begin{aligned}\exp(x\vec{M}) &:= \eta(x\exp(\vec{M})),\\ \exp(\lambda xM) &:= \lambda x\exp(M),\\ \exp((\lambda xM)N\vec{N}) &:= \eta((\lambda x\exp(M))\exp(N)\exp(\vec{N}))\end{aligned}$$

and its expansion height $\#_{\eta}(M^{\rho}) \in \mathbb{N}$ by

$$\begin{aligned}\#_{\eta}(x\vec{M}) &:= \mu_{\rho} + \#_{\eta}(\vec{M}),\\ \#_{\eta}(\lambda xM) &:= \#_{\eta}(M),\\ \#_{\eta}((\lambda xM)N\vec{N}) &:= \mu_{\rho} + \#_{\eta}(M, N, \vec{N}).\end{aligned}$$

Here $\#_{\eta}(\vec{M})$ means $\sum_i \#_{\eta}(M_i)$.

Lemma.

(1.9) $M \to_{\eta\uparrow}^{\#_{\eta}(M)} \exp(M) \in \mathrm{Nf}_{\eta\uparrow}$.

(1.10) $M \in \mathrm{Nf}_{\eta\uparrow} \iff \exp(M) = M \iff \#_{\eta}(M) = 0$.

(1.11) $\eta(\exp(M)) \to_{\beta}^{*} \exp(M)$.

(1.12) $\exp(M)[x := \eta(x)] \to_{\beta}^{*} \exp(M)$.

(1.13) $\eta(\exp(M)\exp(\vec{N})) \to_{\beta}^{*} \exp(M\vec{N})$.

(1.14) $\exp(M)[x := \exp(N)] \to_{\beta}^{*} \exp(M[x := N])$.

(1.15) $M \to_{\eta\uparrow} M' \Rightarrow \exp(M) = \exp(M'), \quad \#_{\eta}(M) = \#_{\eta}(M') + 1$.

PROOF. (1.9). First show $\exp(M) \in \mathrm{Nf}_{\eta\uparrow}$ by induction on $M$, using (1.6). Then prove $M \to_{\eta\uparrow}^{\#_\eta(M)} \exp(M)$ also by induction on $M$, using (1.5). (1.10). Use (1.6) and the above characterization of $\mathrm{Nf}_{\eta\uparrow}$. (1.11) and (1.12) are proven by simultaneous induction on $M$. (1.11). For non-introduced terms use (1.8). For an abstraction we have

$$\begin{aligned}
\eta(\exp(\lambda x M)) &= \eta(\lambda x \exp(M)) \\
&= \lambda y\, \eta((\lambda x \exp(M))\eta(y)) \\
&\to_\beta \lambda x\, \eta(\exp(M)[x := \eta(x)]) \\
&\to_\beta^* \lambda x\, \eta(\exp(M)) && \text{by IH(1.12)} \\
&\to_\beta^* \lambda x \exp(M) && \text{by IH(1.11).}
\end{aligned}$$

The only interesting case for (1.12) is $M \equiv x\vec{M}$, where we have

$$\begin{aligned}
\exp(x\vec{M})[x := \eta(x)] &= \eta(x \exp(\vec{M}))[x := \eta(x)] \\
&= \eta(\eta(x) \exp(\vec{M})[x := \eta(x)]) \\
&\to_\beta^* \eta(\eta(x) \exp(\vec{M})) && \text{by IH(1.12)} \\
&\to_\beta^* \eta\eta(x\, \eta(\exp(\vec{M}))) && \text{by (1.7)} \\
&\to_\beta^* \eta(x\, \eta(\exp(\vec{M}))) && \text{by (1.8)} \\
&\to_\beta^* \eta(x \exp(\vec{M})) && \text{by IH(1.11)} \\
&= \exp(x\vec{M}).
\end{aligned}$$

(1.13) is shown by induction on $M$. By (1.11) we can assume that $\vec{N}$ is not empty.

$$\begin{aligned}
\eta(\exp(x\vec{M})\ \exp(\vec{N})) &= \eta(\eta(x \exp(\vec{M}))\ \exp(\vec{N})) \\
&\to_\beta^* \eta\eta(x \exp(\vec{M})\, \eta(\exp(\vec{N}))) && \text{by (1.7)} \\
&\to_\beta^* \eta\eta(x \exp(\vec{M})\ \exp(\vec{N})) && \text{by (1.11)} \\
&\to_\beta^* \eta(x \exp(\vec{M})\ \exp(\vec{N})) && \text{by (1.8)} \\
&= \exp(x\vec{M}\vec{N}).
\end{aligned}$$

and

$$\begin{aligned}
\eta(\exp(\lambda x M)\ \exp(\vec{N})) &= \eta((\lambda x \exp(M))\ \exp(\vec{N})) \\
&= \exp((\lambda x M)\, \vec{N}).
\end{aligned}$$

The case $(\lambda x M)K\vec{K}$ is similar. (1.14) is an easy induction on $M$. We only treat the case $x\vec{M}$, where one needs (1.13).

$$\begin{aligned}\exp(x\vec{M})[x := \exp(N)] &= \eta(\exp(N)\exp(\vec{M})[x := \exp(N)]) \\ &\to_\beta^* \eta(\exp(N)\exp(\vec{M}[x := N])) && \text{by IH} \\ &\to_\beta^* \exp(N\,\vec{M}[x := N]) && \text{by (1.13).}\end{aligned}$$

(1.15). For an $\eta\uparrow$-conversion $M \mapsto_{\eta\uparrow} \lambda x.Mx$, $M$ non-introduced, we treat the case $M \equiv x\vec{M}$.

$$\begin{aligned}\exp(x\vec{M}) &= \eta_{\rho\Rightarrow\sigma}(x\exp(\vec{M})) \\ &= \lambda y^\rho \eta_\sigma(x\exp(\vec{M})\eta_\rho(y)) \\ &= \exp(\lambda y.x\vec{M}y).\end{aligned}$$

and

$$\begin{aligned}\#_\eta((x\vec{M})^{\rho\Rightarrow\sigma}) &= \mu_{\rho\Rightarrow\sigma} + \#_\eta(\vec{M}) \\ &= 1 + \mu_\sigma + \#_\eta(\vec{M}) + \mu_\rho \\ &= 1 + \mu_\sigma + \#_\eta(\vec{M}, y^\rho) \\ &= 1 + \#_\eta(\lambda y^\rho.x\vec{M}y).\end{aligned}$$

The case $M \equiv (\lambda x N)K\vec{K}$ is analogous. □

LEMMA. *$\beta$-reduction is simulated on expanded terms:*

$$M \to_\beta M' \Longrightarrow \exp(M) \to_\beta^+ \exp(M').$$

*Conclude that $\to_{\beta\eta\uparrow}$ is strongly normalizing.*

PROOF. The first part is proved by induction on $M$. We only handle the interesting case of a $\beta$-conversion.

$$\begin{aligned}\exp((\lambda x M)N\vec{N}) &= \eta((\lambda x\exp(M))\exp(N)\ \exp(\vec{N})) \\ &\to_\beta \eta(\exp(M)[x := \exp(N)]\ \exp(\vec{N})) \\ &\to_\beta^* \eta(\exp(M[x := N])\exp(\vec{N})) && \text{by (1.14)} \\ &\to_\beta^* \exp(M[x := N]\vec{N}) && \text{by (1.13).}\end{aligned}$$

All other cases follow directly from the IH.

To prove strong normalization of $\to_{\beta\eta\uparrow}$, note that we can simulate any $\beta$-reduction on a term $M$ by a positive number of $\beta$-reductions on $\exp(M)$, while $\eta$-expansions leave $\exp(M)$ unchanged by (1.15). Strong normalizability of $M$ now follows by induction on the height of the $\beta$-reduction tree of $\exp(M)$ and side induction on $\#_\eta(M)$. □

## 1.4. Normalization with Permutative Conversions

We now consider $\to\forall\bot\vee\wedge\exists$-formulas. The normalization result in Sec.1.3 and in particular the subformula property does not say much in this case, since in our derivations we allow arbitrary $\wedge^{\pm}$, $\vee^{\pm}$ and $\exists^{\pm}$-axioms. The cure consists in the following. (1) In derivations in long normal form we can replace every use of an $\wedge^{\pm}, \vee^{\pm}, \exists^{\pm}$-axiom by a corresponding rule; in fact, there is almost no difference between these derivations. (2) After this replacement the need for *permutative conversions* becomes visible, if we want to keep the subformula property for normal derivations. We shall prove strong normalization, and analyse again the shape of normal derivations.

**1.4.1. Rules for $\vee$, $\wedge$ and $\exists$.** Notice that we have not given rules for the connectives $\vee$, $\wedge$ and $\exists$. There are two reasons for this omission:

- They can be covered by means of appropriate axioms as constant derivation terms, as given in Sec.1.2.3;
- For simplicity we want our derivation terms to be pure lambda terms formed just by lambda abstraction and application. This would be violated by the rules for $\vee$, $\wedge$ and $\exists$, which require additional constructs.

However – as just noted – in order to have a normalization theorem with a useful subformula property as a consequence we do need to consider rules for these connectives. So here they are:

*Disjunction.* The introduction rules are

$$\frac{\begin{array}{c}|\, M\\ A\end{array}}{A\vee B}\vee_0^+ \qquad \frac{\begin{array}{c}|\, M\\ B\end{array}}{A\vee B}\vee_1^+$$

and the elimination rule is

$$\frac{\begin{array}{ccc} & [u\colon A] & [v\colon B]\\ |\, M & |\, N & |\, K\\ A\vee B & C & C\end{array}}{C}\vee^- u, v$$

*Conjunction.* The introduction rule is

$$\frac{\begin{array}{cc}|\, M & |\, N\\ A & B\end{array}}{A\wedge B}\wedge^+$$

and the elimination rule is

$$\dfrac{\begin{array}{ccc} & [u\colon A] & [v\colon B] \\ \mid M & & \mid N \\ A \wedge B & & C \end{array}}{C}\ \wedge^{-} u, v$$

*Existential Quantifier.* The introduction rule is

$$\dfrac{\begin{array}{cc} & \mid M \\ r & A[x := r] \end{array}}{\exists_x A}\ \exists^{+}$$

and the elimination rule is

$$\dfrac{\begin{array}{ccc} & [u\colon A] & \\ \mid M & & \mid N \\ \exists_x A & & B \end{array}}{B}\ \exists^{-} x, u\ \text{(var.cond.)}$$

The rule $\exists^{-} x, u$ is subject to the following *(Eigen-) variable condition*: The derivation $N$ should not contain any open assumptions apart from $u\colon A$ whose assumption formula contains $x$ free, and moreover $B$ should not contain the variable $x$ free.

It is easy to see that for each of the connectives $\vee$, $\wedge$, $\exists$ the rules and the axioms are equivalent, in the sense that from the axioms and the premises of a rule we can derive its conclusion (of course without any $\vee, \wedge, \exists$-rules), and conversely that we can derive the axioms by means of the $\vee, \wedge, \exists$-rules. This is left as an exercise.

The left premise in each of the elimination rules $\vee^{-}$, $\wedge^{-}$ and $\exists^{-}$ is called *major premise* (or *main* premise), and each of the right premises *minor premise* (or *side* premise).

**1.4.2. Conversion.** In addition to the $\rightarrow, \forall$-conversions in Sec.1.3.1, we consider the following conversions:

$\vee$*-conversion.*

$$\dfrac{\dfrac{\begin{array}{c} \mid M \\ A \end{array}}{A \vee B}\ \vee_0^{+} \qquad \begin{array}{ccc} [u\colon A] & & [v\colon B] \\ \mid N & & \mid K \\ C & & C \end{array}}{C}\ \vee^{-} u, v \quad \mapsto \quad \begin{array}{c} \mid M \\ A \\ \mid N \\ C \end{array}$$

and

$$\dfrac{\dfrac{\begin{array}{c} \mid M \\ B \end{array}}{A \vee B}\ \vee_1^{+} \qquad \begin{array}{ccc} [u\colon A] & & [v\colon B] \\ \mid N & & \mid K \\ C & & C \end{array}}{C}\ \vee^{-} u, v \quad \mapsto \quad \begin{array}{c} \mid M \\ B \\ \mid K \\ C \end{array}$$

$\wedge$-*conversion.*

$$\dfrac{\dfrac{\begin{array}{c}|\,M\\ A\end{array}\qquad\begin{array}{c}|\,N\\ B\end{array}}{A\wedge B}\wedge^{+}\qquad\begin{array}{c}[u\colon A]\quad[v\colon B]\\ |\,K\\ C\end{array}}{C}\wedge^{-}u,v \quad\mapsto\quad \begin{array}{c}|\,M\\ A\end{array}\qquad\begin{array}{c}|\,N\\ B\end{array}\qquad\begin{array}{c}|\,K\\ C\end{array}$$

$\exists$-*conversion.*

$$\dfrac{\begin{array}{cc}&|\,M\\ r\qquad A[x:=r]\end{array}}{\exists_x A}\exists^{+}\qquad\begin{array}{c}[u\colon A]\\ |\,N\\ B\end{array}}{B}\exists^{-}x,u \quad\mapsto\quad \begin{array}{c}|\,M\\ A[x:=r]\\ |\,N'\\ B\end{array}$$

**1.4.3. Permutative conversion.** In a permutative conversion we permute an E-rule upwards over the minor premises of $\vee^{-}$, $\wedge^{-}$ or $\exists^{-}$.

$\vee$-*perm conversion.*

$$\dfrac{\begin{array}{ccc}|\,M & |\,N & |\,K\\ A\vee B & C & C\end{array}\qquad\begin{array}{c}|\,L\\ C'\end{array}}{D}\text{E-rule} \quad\mapsto$$

$$\dfrac{\begin{array}{c}|\,M\\ A\vee B\end{array}\qquad\dfrac{\begin{array}{cc}|\,N & |\,L\\ C & C'\end{array}}{D}\text{E-rule}\qquad\dfrac{\begin{array}{cc}|\,K & |\,L\\ C & C'\end{array}}{D}\text{E-rule}}{D}$$

$\wedge$-*perm conversion.*

$$\dfrac{\begin{array}{cc}|\,M & |\,N\\ A\wedge B & C\end{array}\qquad\begin{array}{c}|\,K\\ C'\end{array}}{D}\text{E-rule} \quad\mapsto$$

$$\dfrac{\begin{array}{c}|\,M\\ A\wedge B\end{array}\qquad\dfrac{\begin{array}{cc}|\,N & |\,K\\ C & C'\end{array}}{D}\text{E-rule}}{D}$$

$\exists$-*perm conversion.*

$$\dfrac{\dfrac{\exists_x A \quad \overset{|\,M}{B}}{\overset{|\,N}{D}} \quad \overset{|\,K}{C}}{D}\ \text{E-rule} \qquad \mapsto$$

$$\dfrac{\exists_x A \quad \dfrac{\overset{|\,N}{B} \quad \overset{|\,K}{C}}{D}\ \text{E-rule}}{D} \qquad \overset{|\,M}{}$$

**1.4.4. Derivations as terms.** The term representation of derivations has to be extended. The rules for $\vee$, $\wedge$ and $\exists$ with the corresponding terms are given in Table 2 on page 32.

The introduction rule $\exists^+$ has as its left premise the witnessing term $r$ to be substituted. The elimination rule $\exists^- u$ is subject to an *(Eigen-) variable condition*: The derivation term $N$ should not contain any open assumptions apart from $u\colon A$ whose assumption formula contains $x$ free, and moreover $B$ should not contain the variable $x$ free.

**1.4.5. Reduction for permutative conversions.** In this section we shall write derivation terms without formula superscripts. We usually leave implicit the extra (formula) parts of derivation constants and for instance write $\exists^+$, $\exists^-$ instead of $\exists^+_{x,A}$, $\exists^-_{x,A,B}$. So we consider derivation terms $M, N, K$ of the forms

$$\begin{aligned} &u \mid \lambda v M \mid \lambda y M \mid \vee_0^+ M \mid \vee_1^+ M \mid \langle M, N\rangle \mid \exists^+ r M \mid \\ &\quad MN \mid Mr \mid M(v_0.N_0, v_1.N_1) \mid M(v, w.N) \mid M(v.N); \end{aligned}$$

in these expressions the variables $y, v, v_0, v_1, w$ get bound.

To simplify the technicalities, we restrict our treatment to the rules for $\to$ and $\exists$. It can easily be extended to the full set of rules; some details for disjunction are given in Sec.1.4.6. So we consider

$$u \mid \lambda v M \mid \exists^+ r M \mid MN \mid M(v.N);$$

in these expressions the variable $v$ gets bound.

We reserve the letters $E, F, G$ for *eliminations*, i.e., expressions of the form $(v.N)$, and $R, S, T$ for both terms and eliminations. Using this notation we obtain a second (and clearly equivalent) inductive definition of terms:

$$\begin{aligned} &u\vec{M} \mid u\vec{M}E \mid \lambda v M \mid \exists^+ r M \mid \\ &(\lambda v M)N\vec{R} \mid \exists^+ r M(v.N)\vec{R} \mid u\vec{M}ER\vec{S}. \end{aligned}$$

| derivation | term |
|---|---|
| $\dfrac{\begin{array}{c}\mid M\\ A\end{array}}{A \vee B}\vee_0^+ \qquad \dfrac{\begin{array}{c}\mid M\\ B\end{array}}{A \vee B}\vee_1^+$ | $\left(\vee_{0,B}^+ M^A\right)^{A\vee B} \quad \left(\vee_{1,A}^+ M^B\right)^{A\vee B}$ |
| $\dfrac{\begin{array}{ccc}\mid M & [u\colon A] & [v\colon B]\\ A \vee B & \mid N & \mid K\\ & C & C\end{array}}{C}\vee^- u, v$ | $\left(M^{A\vee B}(u^A.N^C, v^B.K^C)\right)^C$ |
| $\dfrac{\begin{array}{cc}\mid M & \mid N\\ A & B\end{array}}{A \wedge B}\wedge^+$ | $\langle M^A, N^B\rangle^{A\wedge B}$ |
| $\dfrac{\begin{array}{ccc}\mid M & [u\colon A] & [v\colon B]\\ A \wedge B & \mid N & \\ & C & \end{array}}{C}\wedge^- u, v$ | $\left(M^{A\wedge B}(u^A, v^B.N^C)\right)^C$ |
| $\dfrac{\begin{array}{cc}r & \mid M\\ & A[x := r]\end{array}}{\exists_x A}\exists^+$ | $\left(\exists_{x,A}^+ r M^{A[x:=r]}\right)^{\exists_x A}$ |
| $\dfrac{\begin{array}{cc}\mid M & [u\colon A]\\ \exists_x A & \mid N\\ & B\end{array}}{B}\exists^- x, u$ (var.cond.) | $\left(M^{\exists_x A}(u^A.N^B)\right)^B$ (var.cond.) |

TABLE 2. Derivation terms for $\vee$, $\wedge$ and $\exists$

Here the final three forms are not normal: $(\lambda v M)N\vec{R}$ and $\exists^+ r M(v.N)\vec{R}$ both are *$\beta$-redexes*, and $u\vec{M}ER\vec{S}$ is a *permutative redex*. The conversion rules are

$$\begin{array}{lll} (\lambda v M)N & \mapsto_\beta M[v := N] & \beta_\to\text{-conversion,} \\ \exists^+_{x,A} r M(v.N) & \mapsto_\beta N[x := r][v := M] & \beta_\exists\text{-conversion,} \\ M(v.N)R & \mapsto_\pi M(v.NR) & \text{permutative conversion.} \end{array}$$

The *closure* of these conversions is defined by

- If $M \mapsto_\beta M'$ or $M \mapsto_\pi M'$, then $M \to M'$.
- If $M \to M'$, then also $MR \to M'R$, $NM \to NM'$, $N(v.M) \to N(v.M')$, $\lambda v M \to \lambda v M'$, $\exists^+ r M \to \exists^+ r M'$ (*inner reductions*).

We now give the rules to inductively generate a set SN:

$$\frac{\vec{M} \in \mathrm{SN}}{u\vec{M} \in \mathrm{SN}}\,(\mathrm{Var}_0) \qquad \frac{M \in \mathrm{SN}}{\lambda v M \in \mathrm{SN}}\,(\lambda) \qquad \frac{M \in \mathrm{SN}}{\exists^+ r M \in \mathrm{SN}}\,(\exists)$$

$$\frac{\vec{M}, N \in \mathrm{SN}}{u\vec{M}(v.N) \in \mathrm{SN}}\,(\mathrm{Var}) \qquad \frac{u\vec{M}(v.NR)\vec{S} \in \mathrm{SN}}{u\vec{M}(v.N)R\vec{S} \in \mathrm{SN}}\,(\mathrm{Var}_\pi)$$

$$\frac{M[v := N]\vec{R} \in \mathrm{SN} \qquad N \in \mathrm{SN}}{(\lambda v M)N\vec{R} \in \mathrm{SN}}\,(\beta_\to)$$

$$\frac{N[x := r][v := M]\vec{R} \in \mathrm{SN} \qquad M \in \mathrm{SN}}{\exists^+_{x,A} r M(v.N)\vec{R} \in \mathrm{SN}}\,(\beta_\exists)$$

where in $(\mathrm{Var}_\pi)$ we require that $v$ is not free in $R$.

Write $M{\downarrow}$ to mean that $M$ is strongly normalizing, i.e., that every reduction sequence starting from $M$ terminates. By analyzing the possible reduction steps we now show that the set $\mathrm{Wf} := \{\, M \mid M{\downarrow} \,\}$ has the closure properties of the definition of SN above, and hence $\mathrm{SN} \subseteq \mathrm{Wf}$.

Lemma. *Every term in* SN *is strongly normalizing.*

Proof. We distinguish cases according to the generation rule of SN applied last. The following rules deserve special attention.

*Case* $(\mathrm{Var}_\pi)$. We prove, as an auxiliary lemma, that

$$u\vec{M}(v.NR)\vec{S}{\downarrow} \text{ implies } u\vec{M}(v.N)R\vec{S}{\downarrow},$$

by induction on $u\vec{M}(v.NR)\vec{S}{\downarrow}$ (i.e., on the reduction tree of this term). We consider the possible reducts of $u\vec{M}(v.N)R\vec{S}$. The only interesting case is

$u\vec{M}(v.N)(v'.N')T\vec{T}$, and we have a permutative conversion of $(v'.N')$ with $T$, leading to the term $M = u\vec{M}(v.N)(v'.N'T)\vec{T}$. Now $M{\downarrow}$ follows, since

$$u\vec{M}(v.N(v'.N'))T\vec{T}$$

leads in two permutative steps to $M$, hence by assumption $M{\downarrow}$.

*Case* $(\beta_\to)$. We show that $M[v := N]\vec{R}{\downarrow}$ and $N{\downarrow}$ imply $(\lambda vM)N\vec{R}{\downarrow}$. This is done by a induction on $N{\downarrow}$, with a side induction on $M[v := N]\vec{R}{\downarrow}$. We need to consider all possible reducts of $(\lambda vM)N\vec{R}$. In case of an outer $\beta$-reduction use the assumption. If $N$ is reduced, use the IH. Reductions in $M$ and in $\vec{R}$ as well as permutative reductions within $\vec{R}$ are taken care of by the side IH.

*Case* $(\beta_\exists)$. We show that $N[x := r][v := M]\vec{R}{\downarrow}$ and $M{\downarrow}$ together imply $\exists^+ rM(v.N)\vec{R}{\downarrow}$. This is done by a threefold induction: first on $M{\downarrow}$, second on $N[x := r][v := M]\vec{R}{\downarrow}$ and third on the length of $\vec{R}$. We need to consider all possible reducts of $\exists^+ rM(v.N)\vec{R}$. In case of an outer $\beta$-reduction use the assumption. If $M$ is reduced, use the first IH. Reductions in $N$ and in $\vec{R}$ as well as permutative reductions within $\vec{R}$ are taken care of by the second IH. The only remaining case is when $\vec{R} = S\vec{S}$ and $(v.N)$ is permuted with $S$, to yield $\exists^+ rM(v.NS)\vec{S}$. Apply the third IH, since $(NS)[x := r][v := M]\vec{S} = N[x := r][v := M]S\vec{S}$. □

For later use we prove a slightly generalized form of the rule $(\mathrm{Var}_\pi)$:

PROPOSITION. *If* $M(v.NR)\vec{S} \in \mathrm{SN}$*, then* $M(v.N)R\vec{S} \in \mathrm{SN}$.

PROOF. Induction on the generation of $M(v.NR)\vec{S} \in \mathrm{SN}$. We distinguish cases according to the form of $M$.

*Case* $u\vec{T}(v.NR)\vec{S} \in \mathrm{SN}$. If $\vec{T} = \vec{M}$, use $(\mathrm{Var}_\pi)$. Otherwise we have $u\vec{M}(v'.N')\vec{R}(v.NR)\vec{S} \in \mathrm{SN}$. This must be generated by repeated applications of $(\mathrm{Var}_\pi)$ from $u\vec{M}(v'.N'\vec{R}(v.NR)\vec{S}) \in \mathrm{SN}$, and finally by (Var) from $\vec{M} \in \mathrm{SN}$ and $N'\vec{R}(v.NR)\vec{S} \in \mathrm{SN}$. The IH for the latter yields $N'\vec{R}(v.N)R\vec{S} \in \mathrm{SN}$, hence $u\vec{M}(v.N'\vec{R}(v.N)R\vec{S}) \in \mathrm{SN}$ by (Var) and finally $u\vec{M}(v.N')\vec{R}(v.N)R\vec{S} \in \mathrm{SN}$ by $(\mathrm{Var}_\pi)$.

*Case* $\exists^+ rM\vec{T}(v.NR)\vec{S} \in \mathrm{SN}$. Similarly, with $(\beta_\exists)$ instead of $(\mathrm{Var}_\pi)$. In detail: If $\vec{T}$ is empty, by $(\beta_\exists)$ this came from $(NR)[x := r][v := M]\vec{S} = N[x := r][v := M]R\vec{S} \in \mathrm{SN}$ and $M \in \mathrm{SN}$, hence $\exists^+ rM(v.N)R\vec{S} \in \mathrm{SN}$ again by $(\beta_\exists)$. Otherwise we have $\exists^+ rM(v'.N')\vec{T}(v.NR)\vec{S} \in \mathrm{SN}$. This must be generated by $(\beta_\exists)$ from $N'[x := r][v' := M]\vec{T}(v.NR)\vec{S} \in \mathrm{SN}$. The IH yields $N'[x := r][v' := M]\vec{T}(v.N)R\vec{S} \in \mathrm{SN}$, hence $\exists^+ rM(v'.N')\vec{T}(v.N)R\vec{S} \in \mathrm{SN}$ by $(\beta_\exists)$.

*Case* $(\lambda v M)N'\vec{R}(w.NR)\vec{S} \in \mathrm{SN}$. By $(\beta_\to)$ this came from $N' \in \mathrm{SN}$ and $M[v := N']\vec{R}(w.NR)\vec{S} \in \mathrm{SN}$. The IH yields $M[v := N']\vec{R}(w.N)R\vec{S} \in \mathrm{SN}$, hence $(\lambda v M)N'\vec{R}(w.N)R\vec{S} \in \mathrm{SN}$ by $(\beta_\to)$. □

In what follows we shall show that *every* term is in SN and hence is strongly normalizing. Given the definition of SN we only have to show that SN is closed under $\to^-$ and $\exists^-$. In order to prove this we must prove simultaneously the closure of SN under substitution.

THEOREM (Properties of SN). *For all formulas $A$,*

(a) *for all $M \in \mathrm{SN}$, if $M$ proves $A = A_0 \to A_1$ and $N \in \mathrm{SN}$, then $MN \in \mathrm{SN}$,*
(b) *for all $M \in \mathrm{SN}$, if $M$ proves $A = \exists_x B$ and $N \in \mathrm{SN}$, then $M(v.N) \in \mathrm{SN}$,*
(c) *for all $M \in \mathrm{SN}$, if $N^A \in \mathrm{SN}$, then $M[v := N] \in \mathrm{SN}$.*

PROOF. Induction on $\mathrm{dp}(A)$. We prove (a) and (b) before (c), and hence have (a) and (b) available for the proof of (c). More formally, by induction on $A$ we simultaneously prove that (a) holds, that (b) holds and that (a), (b) together imply (c).

(a). By induction on $M \in \mathrm{SN}$. Let $M \in \mathrm{SN}$ and assume that $M$ proves $A = A_0 \to A_1$ and $N \in \mathrm{SN}$. We distinguish cases according to how $M \in \mathrm{SN}$ was generated. For $(\mathrm{Var}_0)$, $(\mathrm{Var}_\pi)$, $(\beta_\to)$ and $(\beta_\exists)$ use the same rule again.

*Case* $u\vec{M}(v.N') \in \mathrm{SN}$ by (Var) from $\vec{M}, N' \in \mathrm{SN}$. Then $N'N \in \mathrm{SN}$ by side IH for $N'$, hence $u\vec{M}(v.N'N) \in \mathrm{SN}$ by (Var), hence $u\vec{M}(v.N')N \in \mathrm{SN}$ by $(\mathrm{Var}_\pi)$.

*Case* $(\lambda v M)^{A_0 \to A_1} \in \mathrm{SN}$ by $(\lambda)$ from $M \in \mathrm{SN}$. Use $(\beta_\to)$; for this we need to know $M[v := N] \in \mathrm{SN}$. But this follows from IH(c) for $M$, since $N$ derives $A_0$.

(b). By induction on $M \in \mathrm{SN}$. Let $M \in \mathrm{SN}$ and assume that $M$ proves $A = \exists_x B$ and $N \in \mathrm{SN}$. The goal is $M(v.N) \in \mathrm{SN}$. We distinguish cases according to how $M \in \mathrm{SN}$ was generated. For $(\mathrm{Var}_\pi)$, $(\beta_\to)$ and $(\beta_\exists)$ use the same rule again.

*Case* $u\vec{M} \in \mathrm{SN}$ by $(\mathrm{Var}_0)$ from $\vec{M} \in \mathrm{SN}$. Use (Var).

*Case* $(\exists^+ r M)^{\exists_x A} \in \mathrm{SN}$ by $(\exists)$ from $M \in \mathrm{SN}$. Use $(\beta_\exists)$; for this we need to know $N[x := r][v := M] \in \mathrm{SN}$. But this follows from IH(c) for $N[x := r]$, since $M$ derives $A[x := r]$.

*Case* $u\vec{M}(v'.N') \in \mathrm{SN}$ by (Var) from $\vec{M}, N' \in \mathrm{SN}$. Then $N'(v.N) \in \mathrm{SN}$ by side IH for $N'$, hence $u\vec{M}(v.N'(v.N)) \in \mathrm{SN}$ by (Var) and therefore $u\vec{M}(v.N')(v.N) \in \mathrm{SN}$ by $(\mathrm{Var}_\pi)$.

(c). By induction on $M \in \mathrm{SN}$. Let $N^A \in \mathrm{SN}$; the goal is $M[v := N] \in \mathrm{SN}$. We distinguish cases according to how $M \in \mathrm{SN}$ was generated. For $(\lambda)$, $(\exists)$, $(\beta_\to)$ and $(\beta_\exists)$ use the same rule again.

*Case* $u\vec{M} \in \mathrm{SN}$ by $(\mathrm{Var}_0)$ from $\vec{M} \in \mathrm{SN}$. Then $\vec{M}[v := N] \in \mathrm{SN}$ by SIH(c). If $u \neq v$, use $(\mathrm{Var}_0)$ again. If $u = v$, we must show $N\vec{M}[v := N] \in \mathrm{SN}$. Note that $N$ proves $A$; hence the claim follows from (a) and the IH.

*Case* $u\vec{M}(v'.N') \in \mathrm{SN}$ by (Var) from $\vec{M}, N' \in \mathrm{SN}$. If $u \neq v$, use (Var) again. If $u = v$, we must show $N\vec{M}[v := N](v'.N'[v := N]) \in \mathrm{SN}$. Note that $N$ proves $A$; hence in case $\vec{M}$ empty the claim follows from (b), and otherwise from (a) and the IH.

*Case* $u\vec{M}(v'.N')R\vec{S} \in \mathrm{SN}$ by $(\mathrm{Var}_\pi)$ from $u\vec{M}(v'.N'R)\vec{S} \in \mathrm{SN}$. If $u \neq v$, use $(\mathrm{Var}_\pi)$ again. If $u = v$, from the IH we obtain

$$N\vec{M}[v := N](v'.N'[v := N]R[v := N])\vec{S}[v := N] \in \mathrm{SN}$$

Now use the proposition above. □

Corollary. *Every term is strongly normalizing.*

Proof. Induction on the (first) inductive definition of terms $M$. In cases $u$, $\lambda vM$ and $\exists^+ rM$ the claim follows from the definition of SN, and in cases $MN$ and $M(v.N)$ from parts (a), (b) of the previous theorem. □

**1.4.6. Disjunction.** We describe the changes necessary to extend the result above to the language with disjunction $\vee$.

We have additional $\beta$- and permutative conversions

$$\begin{array}{lll} \vee_i^+ M(v_0.N_0, v_1.N_1) \mapsto_\beta N_i[v_i := M] & & \beta_{\vee_i}\text{-conversion,} \\ M(v_0.N_0, v_1.N_1)R \ \mapsto_\pi M(v_0.N_0R, v_1.N_1R) & & \text{permutative conversion.} \end{array}$$

The definition of SN needs to be extended by

$$\frac{M \in \mathrm{SN}}{\vee_i^+ M \in \mathrm{SN}}\,(\vee_i)$$

$$\frac{\vec{M}, N_0, N_1 \in \mathrm{SN}}{u\vec{M}(v_0.N_0, v_1.N_1) \in \mathrm{SN}}\,(\mathrm{Var}_\vee) \qquad \frac{u\vec{M}(v_0.N_0R, v_1.N_1R)\vec{S} \in \mathrm{SN}}{u\vec{M}(v_0.N_0, v_1.N_1)R\vec{S} \in \mathrm{SN}}\,(\mathrm{Var}_{\vee,\pi})$$

$$\frac{N_i[v_i := M]\vec{R} \in \mathrm{SN} \qquad N_{1-i}\vec{R} \in \mathrm{SN} \qquad M \in \mathrm{SN}}{\vee_i^+ M(v_0.N_0, v_1.N_1)\vec{R} \in \mathrm{SN}}\,(\beta_{\vee_i})$$

The former rules (Var), $(\mathrm{Var}_\pi)$ should then be renamed into $(\mathrm{Var}_\exists)$, $(\mathrm{Var}_{\exists,\pi})$.

The lemma above stating that every term in SN is strongly normalizing needs to be extended by an additional clause:

*Case* $(\beta_{\vee_i})$. We show that $N_i[v_i := M]\vec{R}\!\downarrow$, $N_{1-i}\vec{R}\!\downarrow$ and $M\!\downarrow$ together imply $\vee_i^+ M(v_0.N_0, v_1.N_1)\vec{R}\!\downarrow$. This is done by a fourfold induction: first on $M\!\downarrow$, second on $N_i[v_i := M]\vec{R}\!\downarrow$, $N_{1-i}\vec{R}\!\downarrow$, third on $N_{1-i}\vec{R}\!\downarrow$ and fourth on the length

of $\vec{R}$. We need to consider all possible reducts of $\vee_i^+ M(v_0.N_0, v_1.N_1)\vec{R}$. In case of an outer $\beta$-reduction use the assumption. If $M$ is reduced, use the first IH. Reductions in $N_i$ and in $\vec{R}$ as well as permutative reductions within $\vec{R}$ are taken care of by the second IH. Reductions in $N_{1-i}$ are taken care of by the third IH. The only remaining case is when $\vec{R} = S\vec{S}$ and $(v_0.N_0, v_1.N_1)$ is permuted with $S$, to yield $(v_0.N_0S, v_1.N_1S)$. Apply the fourth IH, since $(N_iS)[v := M]\vec{S} = N_i[v := M]S\vec{S}$.

Finally the theorem above stating properties of SN needs an additional clause:

- for all $M \in \mathrm{SN}$, if $M$ proves $A = A_0 \vee A_1$ and $N_0, N_1 \in \mathrm{SN}$, then $M(v_0.N_0, v_1.N_1) \in \mathrm{SN}$.

PROOF. The new clause is proved by induction on $M \in \mathrm{SN}$. Let $M \in \mathrm{SN}$ and assume that $M$ proves $A = A_0 \vee A_1$ and $N_0, N_1 \in \mathrm{SN}$. The goal is $M(v_0.N_0, v_1.N_1) \in \mathrm{SN}$. We distinguish cases according to how $M \in \mathrm{SN}$ was generated. For $(\mathrm{Var}_{\exists,\pi})$, $(\mathrm{Var}_{\vee,\pi})$, $(\beta_\to)$, $(\beta_\exists)$ and $(\beta_{\vee_i})$ use the same rule again.

*Case* $u\vec{M} \in \mathrm{SN}$ by $(\mathrm{Var}_0)$ from $\vec{M} \in \mathrm{SN}$. Use $(\mathrm{Var}_\vee)$.

*Case* $(\vee_i^+ M)^{A_0 \vee A_1} \in \mathrm{SN}$ by $(\vee_i)$ from $M \in \mathrm{SN}$. Use $(\beta_{\vee_i})$; for this we need to know $N_i[v_i := M] \in \mathrm{SN}$ and $N_{1-i} \in \mathrm{SN}$. The latter is assumed, and the former follows from main IH (with $N_i$) for the substitution clause of the theorem, since $M$ derives $A_i$.

*Case* $u\vec{M}(v'.N') \in \mathrm{SN}$ by $(\mathrm{Var}_\exists)$ from $\vec{M}, N' \in \mathrm{SN}$. For brevity let $E := (v_0.N_0, v_1.N_1)$. Then $N'E \in \mathrm{SN}$ by SIH for $N'$, so $u\vec{M}(v'.N'E) \in \mathrm{SN}$ by $(\mathrm{Var}_\exists)$ and therefore $u\vec{M}(v'.N')E \in \mathrm{SN}$ by $(\mathrm{Var}_{\exists,\pi})$.

*Case* $u\vec{M}(v_0'.N_0', v_1'.N_1') \in \mathrm{SN}$ by $(\mathrm{Var}_\vee)$ from $\vec{M}, N_0', N_1' \in \mathrm{SN}$. Let $E := (v_0.N_0, v_1.N_1)$. Then $N_i'E \in \mathrm{SN}$ by SIH for $N_i'$, so $u\vec{M}(v_0'.N_0'E, v_1'.N_1'E) \in \mathrm{SN}$ by $(\mathrm{Var}_\vee)$ and therefore $u\vec{M}(v_0'.N_0', v_1'.N_1')E \in \mathrm{SN}$ by $(\mathrm{Var}_{\vee,\pi})$.

Clause (c) now needs additional cases, e.g.,

*Case* $u\vec{M}(v_0.N_0, v_1.N_1) \in \mathrm{SN}$ by $(\mathrm{Var}_\vee)$ from $\vec{M}, N_0, N_1 \in \mathrm{SN}$. If $u \neq v$, use $(\mathrm{Var}_\vee)$. If $u = v$, we show $N\vec{M}[v := N](v_0.N_0[v := N], v_1.N_1[v := N]) \in \mathrm{SN}$. Note that $N$ proves $A$; hence in case $\vec{M}$ empty the claim follows from (b), and otherwise from (a) and the IH. □

**1.4.7. The structure of normal derivations.** As mentioned already, normalization aims at removing local maxima of complexity, i.e., formula occurrences which are first introduced and immediately afterwards eliminated. However, an introduced formula may be used as a minor premise of an application of $\vee^-$, $\wedge^-$ or $\exists^-$, then stay the same throughout a sequence of

applications of these rules, being eliminated at the end. This also constitutes a local maximum, which we should like to eliminate; this is what the permutative conversions are designed for.

DEFINITION. A *segment* of (*length* $n$) in a derivation $M$ is a sequence $A_1, \ldots, A_n$ of occurrences of a formula $A$ such that

(a) for $1 \leq i < n$, $A_i$ is a minor premise of an application of $\vee^-$, $\wedge^-$ or $\exists^-$, with conclusion $A_{i+1}$;
(b) $A_n$ is not a minor premise of $\vee^-$, $\wedge^-$ or $\exists^-$.
(c) $A_1$ is not the conclusion of $\vee^-$, $\wedge^-$ or $\exists^-$.

(Note: An f.o. which is neither a minor premise nor the conclusion of an application of $\vee^-$, $\wedge^-$ or $\exists^-$ always belongs to a segment of length 1.) A segment is *maximal* or a *cut (segment)* if $A_n$ is the major premise of an E-rule, and either $n > 1$, or $n = 1$ and $A_1 = A_n$ is the conclusion of an I-rule.

We shall use $\sigma, \sigma'$ for segments. We shall say that $\sigma$ is a *subformula* of $\sigma'$ if the formula $A$ in $\sigma$ is a subformula of $B$ in $\sigma'$. Clearly a derivation is normal if and only if it does not contain a maximal segment.

The argument in Sec.1.3.5 needs to be refined to also cover the rules for $\vee, \wedge, \exists$. The reason for the difficulty is that in the E-rules $\vee^-, \wedge^-, \exists^-$ the subformulas of a major premise $A \vee B$, $A \wedge B$ or $\exists_x A$ of an E-rule application do not appear in the conclusion, but among the assumptions being discharged by the application. This suggests the definition of track below.

The general notion of a track is designed to retain the subformula property in case one passes through the major premise of an application of a $\vee^-, \wedge^-, \exists^-$-rule. In a track, when arriving at an $A_i$ which is the major premise of an application of such a rule, we take for $A_{i+1}$ a hypothesis discharged by this rule.

DEFINITION. A *track* of a derivation $M$ is a sequence of f.o.'s $A_0, \ldots, A_n$ such that

(a) $A_0$ is a top f.o. in $M$ not discharged by an application of an $\vee^-, \wedge^-, \exists^-$-rule;
(b) $A_i$ for $i < n$ is not the minor premise of an instance of $\rightarrow^-$, and *either*
  (i) $A_i$ is not the major premise of an instance of a $\vee^-, \wedge^-, \exists^-$-rule and $A_{i+1}$ is directly below $A_i$, *or*
  (ii) $A_i$ is the major premise of an instance of a $\vee^-, \wedge^-, \exists^-$-rule and $A_{i+1}$ is an assumption discharged by this instance;
(c) $A_n$ is *either*
  (i) the minor premise of an instance of $\rightarrow^-$, *or*
  (ii) the conclusion of $M$, *or*

(iii) the major premise of an instance of a $\vee^-, \wedge^-, \exists^-$-rule in case there are no assumptions discharged by this instance.

LEMMA. *In a derivation each formula occurrence belongs to some track.*

PROOF. By induction on derivations. For example, suppose a derivation $K$ ends with an $\exists^-$-application:

$$\dfrac{\exists_x A \qquad \begin{matrix}[u\colon A] \\ \mid N \\ B\end{matrix}}{B}\,\exists^- x, u \qquad \mid M$$

$B$ in $N$ belongs to a track $\pi$ (IH); either this does not start in $u\colon A$, and then $\pi, B$ is a track in $K$ which ends in the conclusion; or $\pi$ starts in $u\colon A$, and then there is a track $\pi'$ in $M$ (IH) such that $\pi', \pi, C$ is a track in $K$ ending in the conclusion. The other cases are left to the reader. □

DEFINITION. A *track of order* 0, or *main track*, in a derivation is a track ending either in the conclusion of the whole derivation or in the major premise of an application of a $\vee^-$, $\wedge^-$ or $\exists^-$-rule, provided there are no assumption variables discharged by the application. A *track of order* $n+1$ is a track ending in the minor premise of an $\rightarrow^-$-application, with major premise belonging to a track of order $n$.

A *main branch* of a derivation is a branch $\pi$ in the proof tree such that $\pi$ passes only through premises of I-rules and *major premises* of E-rules, and $\pi$ begins at a top node and ends in the conclusion.

REMARK. By an obvious *simplification conversion* we may remove every application of an $\vee^-$, $\wedge^-$ or $\exists^-$-rule that discharges no assumption variables. If such simplification conversion are performed, each track of order 0 in a normal derivation is a track ending in the conclusion of the whole derivation.

If we search for a main branch going upwards from the conclusion, the branch to be followed is unique as long as we do not encounter an $\wedge^+$-application.

Now let us consider normal derivations.

PROPOSITION. *Let $M$ be a normal derivation, and let $\pi = \sigma_0, \ldots, \sigma_n$ be a track in $M$. Then there is a segment $\sigma_i$ in $\pi$, the* minimum *segment or* minimum *part of the track, which separates two (possibly empty) parts of $\pi$, called the E-part (*elimination part*) and the I-part (*introduction part*) of $\pi$ such that*

(a) *for each $\sigma_j$ in the E-part one has $j < i$, $\sigma_j$ is a major premise of an E-rule, and $\sigma_{j+1}$ is a strictly positive part of $\sigma_j$, and therefore each $\sigma_j$ is a s.p.p. of $\sigma_0$;*

(b) *for each $\sigma_j$ which is the minimum segment or is in the I-part one has $i \leq j$, and if $j \neq n$, then $\sigma_j$ is a premise of an I-rule and a s.p.p. of $\sigma_{j+1}$, so each $\sigma_j$ is a s.p.p. of $\sigma_n$.*

THEOREM (Subformula property). *Let $M$ be a normal derivation where every application of an $\vee^-$, $\wedge^-$ or $\exists^-$-rule discharges at least one assumption variable. Then each formula occurring in the derivation is a subformula of either the end formula or else an assumption formula.*

PROOF. As note above, each track of order 0 in $M$ is a track ending in the conclusion of $M$. We can now prove the theorem for tracks of order $n$, by induction on $n$. □

THEOREM (Disjunction property). *If $\Gamma$ does not contain a disjunction as s.p.p. (= strictly positive part, defined in Sec.1.1.3), then, if $\Gamma \vdash A \vee B$, it follows that $\Gamma \vdash A$ or $\Gamma \vdash B$.*

PROOF. Consider a normal derivation $M$ of $A \vee B$ from assumptions $\Gamma$ not containing a disjunction as s.p.p. The conclusion $A \vee B$ is the final formula of a (main) track, whose top formula $A_0$ in $M$ must be an assumption in $\Gamma$. Since $\Gamma$ does not contain a disjunction as s.p.p., the segment $\sigma$ with the conclusion $A \vee B$ is in the I-part. Skip the final $\vee_i^+$-rule and replace the formulas in $\sigma$ by $A$ if $i = 0$, and by $B$ if $i = 1$. □

There is a similar theorem for the existential quantifier:

THEOREM (Explicit definability under hypotheses). *Let $\Gamma \vdash \exists_x A$.*

(a) *If $\Gamma$ does not contain an existential s.p.p., then there are terms $r_1$, $r_2$, ..., $r_n$ such that $\Gamma \vdash A[x := r_1] \vee \ldots \vee A[x := r_n]$.*
(b) *If $\Gamma$ neither contains a disjunctive s.p.p., nor an existential s.p.p., then there is a term $r$ such that $\Gamma \vdash A[x := r]$.*

PROOF. Consider a normal derivation $M$ of $\exists_x A$ from assumptions $\Gamma$ not containing an existential s.p.p. We use induction on the derivation, and distinguish cases on the last rule.

(a). By assumption the last rule cannot be $\exists^-$. We only consider the case $\vee^-$ and leave the others to the reader.

$$\dfrac{\begin{array}{ccc} & [u\colon B] & [v\colon C] \\ \mid M & \mid N_0 & \mid N_1 \\ B \vee C & \exists_x A & \exists_x A \end{array}}{\exists_x A} \vee^- u, v$$

By assumption again neither $B$ nor $C$ can have an existential s.p.p. Applying the IH to $N_0$ and $N_1$ we obtain

$$\dfrac{\begin{array}{ccc} & [u\colon B] & [v\colon C] \\ & \mid N_0 & \mid N_1 \\ \mid M & \dfrac{\bigvee_{i=1}^{n} A[x := r_i]}{\bigvee_{i=1}^{n+m} A[x := r_i]}\vee^+ & \dfrac{\bigvee_{i=n+1}^{n+m} A[x := r_i]}{\bigvee_{i=1}^{n+m} A[x := r_i]}\vee^+ \\ B \vee C & & \end{array}}{\bigvee_{i=1}^{n+m} A[x := r_i]}\vee^- u, v$$

(b). Similarly; by assumption the last rule can be neither $\vee^-$ nor $\exists^-$. □

Remark. For $\Gamma$ consisting of Harrop formulas both theorems above hold.

## 1.5. Soundness and Completeness for Beth Models

It is an obvious question to ask whether the logical rules we have been considering suffice, i.e., whether we have forgotten some necessary rules. To answer this question we first have to fix the *meaning* of a formula, i.e., provide a semantics. This will be done by means of Beth models. Using this concept of a model we will prove soundness and completeness for both, minimal and intuitionistic logic.

**1.5.1. Beth models.** Consider a finitely branching tree of "possible worlds". The worlds are represented as nodes in this tree. They may be thought of as possible states such that all nodes "above" a node $k$ are the ways in which $k$ may develop in the future. The worlds are increasing, that is, if an atomic formula $R\vec{s}$ true is in a world $k$, then $R\vec{s}$ is true in all future worlds $k$.

More formally, each Beth model is based on a finitely branching tree $T$. A node $k$ over a set $S$ is a finite sequence $k = \langle a_0, a_1, \ldots, a_{n-1}\rangle$ of elements of $S$; $\mathrm{lh}(k)$ is the length of $k$. We write $k \preceq k'$ if $k$ is an initial segment of $k'$. A tree on $S$ is a set of nodes closed under initial segments. A tree $T$ is finitely branching if every node in $T$ has finitely many immediate successors.

A tree $T$ is *infinite* if for every $n \in \mathbb{N}$ there is a node $k \in T$ such that $\mathrm{lh}(k) = n$. A *branch* of $T$ is a linearly ordered subtree of $T$. A *leaf* is a node without successors in $T$.

For the proof of the completeness theorem, a Beth model based on a complete binary tree (i.e., the complete tree over $\{0, 1\}$) will suffice. The nodes will be all the finite sequences of 0's and 1's, and the ordering is as above. The root is the empty sequence and $k0$ is the sequence $k$ with the element 0 added at the end; similarly for $k1$.

DEFINITION. Let $(T, \preceq)$ be a finitely branching tree. $\mathcal{B} = (D, I_0, I_1)$ is a *$\mathcal{L}$-Beth model* on $T$, where $D$ is a nonempty set, and for every $n$-ary function symbol in $\mathcal{L}$, $I_0$ assigns $f$ a map $I_0(f)\colon D^n \to D$. For every $n$-ary relation symbol $R$ in $\mathcal{L}$ and every node $k \in T$, $I_1(R, k) \subseteq D^n$ is assigned in such a way that monotonicity is preserved, that is,

$$k \preceq k' \Rightarrow I_1(R, k) \subseteq I_1(R, k').$$

If $n = 0$, then $I_1(R, k)$ is either true or false, and it follows by the monotonicity that if $k \preceq k'$ and $I_1(R, k)$ then $I_1(R, k')$. We write $R^{\mathcal{B}}(\vec{a}, k)$ for $\vec{a} \in I_1(R, k)$.

There is no special requirement set on $I_1(\bot, k)$. In minimal logic, *falsum* $\bot$ plays a role of an ordinary propositional variable.

It is obvious from the definition that any $T$ can be extended to a complete tree $\bar{T}$ without leaves, in which for every leaf $k \in T$ all sequences $k0$, $k00$, $k000, \dots$ are added to $T$. For every node $k0 \dots 0$, we then add $I_1(R, k0 \dots 0) := I_1(R, k)$.

For an assignment $\eta$, $t^{\mathcal{B}}[\eta]$ is understood in the canonical sense. The usual satisfaction relation $\mathcal{M} \models A[\eta]$ is replaced by the forcing relation in Beth models.

DEFINITION. $\mathcal{B}, k \Vdash A[\eta]$ ($\mathcal{B}$ forces $A$ at node $k$ for an assignment $\eta$) is defined inductively as follows. We write $k \Vdash A[\eta]$ when it is clear from the context what the underlying model $\mathcal{B}$ is, and $\forall_{k' \succeq_n k} A$ for $\forall_{k' \succeq k}.\mathrm{lh}(k') = \mathrm{lh}(k) + n \to A$.

$$\begin{aligned}
k \Vdash (R\vec{s})[\eta] \quad &:\Longleftrightarrow \quad \exists_n \forall_{k' \succeq_n k} R^{\mathcal{B}}(\vec{s}^{\mathcal{B}}[\eta], k').\\
k \Vdash (A \vee B)[\eta] \quad &:\Longleftrightarrow \quad \exists_n \forall_{k' \succeq_n k}.k' \Vdash A[\eta] \text{ or } k' \Vdash B[\eta].\\
k \Vdash (\exists_x A)[\eta] \quad &:\Longleftrightarrow \quad \exists_n \forall_{k' \succeq_n k} \exists_{a \in |\mathcal{B}|} k' \Vdash A[\eta^a_x].\\
k \Vdash (A \to B)[\eta] \quad &:\Longleftrightarrow \quad \forall_{k' \succeq k}.k' \Vdash A[\eta] \Rightarrow k' \Vdash B[\eta].\\
k \Vdash (A \wedge B)[\eta] \quad &:\Longleftrightarrow \quad k \Vdash A[\eta] \text{ and } k \Vdash B[\eta].\\
k \Vdash (\forall_x A)[\eta] \quad &:\Longleftrightarrow \quad \forall_{a \in |\mathcal{B}|} k \Vdash A[\eta^a_x].
\end{aligned}$$

Notice that the clauses for atoms, disjunction and existential quantifier include a concept of a "bar", in $\bar{T}$.

**1.5.2. Covering lemma.** It is easily seen (using the definition and monotonicity) that from $k \Vdash A[\eta]$ and $k \preceq k'$ we can conclude $k' \Vdash A[\eta]$. The converse is also true:

LEMMA (Covering Lemma).

$$\forall_{k' \succeq_n k} k' \Vdash A[\eta] \Rightarrow k \Vdash A[\eta].$$

PROOF. Induction on $A$. We write $k \Vdash A$ for $k \Vdash A[\eta]$.

*Case* $R\vec{s}$. Assume

$$\exists_n \forall_{k' \succeq_n k}\, k' \Vdash R\vec{s},$$

hence by definition

$$\exists_n \forall_{k' \succeq_n k} \exists_m \forall_{k'' \succeq_m k'} R^{\mathcal{B}}(\vec{s}^{\mathcal{B}}[\eta], k'')$$

Since $T$ is a finitely branching tree,

$$\exists_m \forall_{k' \succeq_m k}\, R^{\mathcal{B}}(\vec{s}^{\mathcal{B}}[\eta], k').$$

Hence $k \Vdash R\vec{s}$.

The cases $A \vee B$ and $\exists_x A$ are handled similarly.

*Case* $A \to B$. Let $k' \Vdash A \to B$ for all $k' \succeq k$ with $\mathrm{lh}(k') = \mathrm{lh}(k) + n$. We show

$$\forall_{l \succeq k}. l \Vdash A \Rightarrow l \Vdash B.$$

Let $l \succeq k$ and $l \Vdash A$. We show that $l \Vdash B$. We apply the IH to $B$ and $m := \max(\mathrm{lh}(k) + n, \mathrm{lh}(l))$. So assume $l' \succeq l$ and $\mathrm{lh}(l') = m$. It is sufficient to show $l' \Vdash B$. If $\mathrm{lh}(l') = \mathrm{lh}(l)$, then $l' = l$ and we are done. If $\mathrm{lh}(l') = \mathrm{lh}(k) + n > \mathrm{lh}(l)$, then $l'$ is an extension of $l$ as well as of $k$ and has length $\mathrm{lh}(k) + n$, and hence $l' \Vdash A \to B$ by assumption. Moreover, $l' \Vdash A$, since $l' \succeq l$ and $l \Vdash A$. It follows that $l' \Vdash B$.

The cases $A \wedge B$ and $\forall_x A$ are obvious. □

### 1.5.3. Soundness.

LEMMA (Coincidence). *Let $\mathcal{B}$ be a Beth model, $t$ a term, $A$ a formula and $\eta$, $\xi$ assignments in $|\mathcal{B}|$.*

(a) *If $\eta(x) = \xi(x)$ for all $x \in \mathrm{vars}(t)$, then $\eta(t) = \xi(t)$.*
(b) *If $\eta(x) = \xi(x)$ for all $x \in \mathrm{FV}(A)$, then $\mathcal{B}, k \Vdash A[\eta] \iff \mathcal{B}, k \Vdash A[\xi]$.*

PROOF. Induction on terms and formulas. □

LEMMA (Substitution). *Let $\mathcal{B}$ be a Beth model, $t, r$ terms, $A$ a formula and $\eta$ an assignment in $|\mathcal{B}|$. Then*

(a) $\eta(r[x := t]) = \eta_x^{\eta(t)}(r)$.
(b) $\mathcal{B}, k \Vdash A[x := t][\eta] \iff \mathcal{B}, k \Vdash A[\eta_x^{\eta(t)}]$.

PROOF. Induction on terms and formulas. □

THEOREM (Soundness). *Let $\Gamma \cup \{A\}$ be a set of formulas such that $\Gamma \vdash A$. Then, if $\mathcal{B}$ is a Beth model, $k$ a node and $\eta$ an assignment in $|\mathcal{B}|$, it follows that $\mathcal{B}, k \Vdash \Gamma[\eta]$ entails $\mathcal{B}, k \Vdash A[\eta]$.*

PROOF. Induction on derivations.

We begin with the axiom schemes $\vee_0^+$, $\vee_1^+$, $\vee^-$, $\exists^+$ and $\exists^-$. $k \Vdash C[\eta]$ is abbreviated $k \Vdash C$, when $\eta$ is known from the context.

*Case* $\vee_0^+ \colon A \to A \vee B$. We show $k \Vdash A \to A \vee B$. Assume for $k' \succeq k$ that $k' \Vdash A$. Show: $k' \Vdash A \vee B$. This follows from the definition, since $k' \Vdash A$. The case $\vee_1^+ \colon B \to A \vee B$ is symmetric.

*Case* $\vee^- \colon (A \to C) \to (B \to C) \to A \vee B \to C$. We show that $k \Vdash (A \to C) \to (B \to C) \to A \vee B \to C$. Assume for $k' \succeq k$ that $k' \Vdash A \to C$, $k' \Vdash B \to C$ and $k' \Vdash A \vee B$ (we can safely assume that $k'$ is the same for all three premises ). Show that $k' \Vdash C$. By definition, there is an $n$ s.t. for all $k'' \succeq_n k'$, $k'' \Vdash A$ or $k'' \Vdash B$. In both cases it follows that $k'' \Vdash C$, since $k' \Vdash A \to C$ and $k' \Vdash B \to C$. By the Covering Lemma, $k' \Vdash C$.

*Case* $\exists^+ \colon A \to \exists_x A$. Show that $k \Vdash (A \to \exists_x A)[\eta]$. Assume that $k' \succeq k$ and $k' \Vdash A[\eta]$. Show that $k' \Vdash (\exists_x A)[\eta]$. Since $\eta = \eta_x^{\eta(x)}$ there is an $a \in |\mathcal{B}|$ (namely $a := \eta(x)$) such that $k' \Vdash A[\eta_x^a]$. Hence, $k' \Vdash (\exists_x A)[\eta]$.

*Case* $\exists^- \colon \forall_x(A \to B) \to \exists_x A \to B$ and $x \notin \mathrm{FV}(B)$. We show that $k \Vdash (\forall_x(A \to B) \to \exists_x A \to B)[\eta]$. Assume that $k' \succeq k$ and $k' \Vdash \forall_x(A \to B)[\eta]$ and $k' \Vdash (\exists_x A)[\eta]$. We show $k' \Vdash B[\eta]$. By definition, there is an $n$ such that for all $k'' \succeq_n k'$ we have $a \in |\mathcal{B}|$ and $k'' \Vdash A[\eta_x^a]$. From $k' \Vdash \forall_x(A \to B)[\eta]$ it follows that $k'' \Vdash B[\eta_x^a]$, and since $x \notin \mathrm{FV}(B)$, from the Coincidence Lemma, $k'' \Vdash B[\eta]$. Then, finally, by the Covering Lemma $k' \Vdash B[\eta]$.

*Case* $\to^+$. Assume $k \Vdash \Gamma$. We show $k \Vdash A \to B$. Assume $k' \succeq k$ and $k' \Vdash A$. Our goal is $k' \Vdash B$. We have $k' \Vdash \Gamma \cup \{A\}$. Thus, $k' \Vdash B$ by IH.

*Case* $\to^-$. Assume $k \Vdash \Gamma$. The IH gives us $k \Vdash A \to B$ and $k \Vdash A$. Hence $k \Vdash B$.

*Case* $\forall^+$. Assume $k \Vdash \Gamma[\eta]$ and $x \notin \mathrm{FV}(\Gamma)$. We show $k \Vdash (\forall_x A)[\eta]$, i.e., $k \Vdash A[\eta_x^a]$ for an arbitrary $a \in |\mathcal{B}|$. We have

$$\begin{array}{ll} k \Vdash \Gamma[\eta_x^a] & \text{by the Coincidence Lemma, since } x \notin \mathrm{FV}(\Gamma) \\ k \Vdash A[\eta_x^a] & \text{by IH.} \end{array}$$

*Case* $\forall^-$. Let $k \Vdash \Gamma[\eta]$. We show that $k \Vdash A[x := t][\eta]$. We have

$$\begin{array}{ll} k \Vdash (\forall_x A)[\eta] & \text{by IH} \\ k \Vdash A[\eta_x^{\eta(t)}] & \text{by definition} \\ k \Vdash A[x := t][\eta] & \text{by the Substitution Lemma.} \end{array}$$

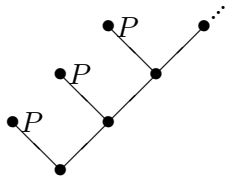
This concludes the proof. □

**1.5.4. Counter models.** With soundness at hand, it is easy to build counter models for derivations not valid in minimal or intuitionistic logic.

A *Beth model* $\mathcal{B} = (D, I_0, I_1)$ *for intuitionistic logic* is a Beth-structure in which $\bot$ is never forced, i.e., $I_1(\bot, k) = 0$ for all $k$. Then

$$\begin{aligned} k \Vdash \neg A &\iff \forall_{k' \succeq k}\, k' \nVdash A, \\ k \Vdash \neg\neg A &\iff \forall_{k' \succeq k}\, k' \nVdash \neg A \\ &\iff \forall_{k' \succeq k} \tilde{\exists}_{k'' \succeq k'}\, k'' \Vdash A. \end{aligned}$$

As an example, we show that $\nvdash_i \neg\neg P \to P$. We describe the desired Beth model by means of a diagram below. Next to every node, we write the propositions forced on that node.




Clearly this is an intuitionistic Beth model. Using the remark above, it is easily seen that

$$\langle\rangle \nVdash P, \qquad \langle\rangle \Vdash \neg\neg P.$$

Thus $\langle\rangle \nVdash \neg\neg P \to P$ and hence $\nvdash \neg\neg P \to P$. Since for every $R$ and all $k$, $k \Vdash \mathrm{Efq}_R$, it also follows that $\nvdash_i \neg\neg P \to P$. The model also shows that the Peirce formula $((P \to Q) \to P) \to P$ is invalid in intuitionistic logic.

### 1.5.5. Completeness.

THEOREM (Completeness). *Let $\Gamma \cup \{A\}$ be a set of formulas. Then the following propositions are equivalent.*

(a) $\Gamma \vdash A$.
(b) $\Gamma \Vdash A$, *i.e., for all Beth models $\mathcal{B}$, nodes $k$ and assignments $\eta$*

$$\mathcal{B}, k \Vdash \Gamma[\eta] \Rightarrow \mathcal{B}, k \Vdash A[\eta].$$

PROOF. Soundness is one direction. For the other direction we employ a technique developed by Harvey Friedman and construct a Beth model $\mathcal{B}$ (over the set $T_{01}$ of all finite 0-1-sequences $k$ ordered by the initial segment relation $k \preceq k'$) with the property that $\Gamma \vdash B$ is equivalent to $\mathcal{B}, \langle\rangle \Vdash B[\mathrm{id}]$. We can assume here that $\Gamma$ and also $A$ are closed.

In order to define $\mathcal{B}$, we will need an enumeration $A_0, A_1, A_2, \ldots$ of $\mathcal{L}$-formulas, in which every formula occurs infinitely often. We also fix an enumeration $x_0, x_1, \ldots$ of distinct variables. Write $\Gamma = \bigcup_n \Gamma_n$ with finite sets $\Gamma_n$ such that $\Gamma_n \subseteq \Gamma_{n+1}$. With every node $k \in T_{01}$, we associate a finite set $\Delta_k$ of formulas and a set $V_k$ of variables, by induction on the length of $k$.

Let $\Delta_{\langle\rangle} := \emptyset$ and $V_{\langle\rangle} := \emptyset$. Take a node $k$ such that $\mathrm{lh}(k) = n$ and suppose that $\Delta_k$, $V_k$ are already defined. Write $\Delta \vdash_n B$ to mean that there is a derivation of length $\leq n$ of $B$ from $\Delta$. We define $\Delta_{k0}$, $V_{k0}$ and $\Delta_{k1}$, $V_{k1}$ as follows:

*Case* 0. $\mathrm{FV}(A_n) \not\subseteq V_k$. Then let

$$\Delta_{k0} := \Delta_{k1} := \Delta_k \quad \text{and} \quad V_{k0} := V_{k1} := V_k.$$

*Case* 1. $\mathrm{FV}(A_n) \subseteq V_k$ and $\Gamma_n, \Delta_k \nvdash_n A_n$. Then let

$$\Delta_{k0} := \Delta_k \quad \text{and} \quad \Delta_{k1} := \Delta_k \cup \{A_n\},$$
$$V_{k0} := V_{k1} := V_k.$$

*Case* 2. $\mathrm{FV}(A_n) \subseteq V_k$ and $\Gamma_n, \Delta_k \vdash_n A_n = A'_n \vee A''_n$. Then let

$$\Delta_{k0} := \Delta_k \cup \{A_n, A'_n\} \quad \text{and} \quad \Delta_{k1} := \Delta_k \cup \{A_n, A''_n\},$$
$$V_{k0} := V_{k1} := V_k.$$

*Case* 3. $\mathrm{FV}(A_n) \subseteq V_k$ and $\Gamma_n, \Delta_k \vdash_n A_n = \exists_x A'_n(x)$. Then let

$$\Delta_{k0} := \Delta_{k1} := \Delta_k \cup \{A_n, A'_n(x_i)\} \quad \text{and} \quad V_{k0} := V_{k1} := V_k \cup \{x_i\},$$

where $x_i$ is the first variable $\notin V_k$.

*Case* 4. $\mathrm{FV}(A_n) \subseteq V_k$ and $\Gamma_n, \Delta_k \vdash_n A_n$, with $A_n$ neither a disjunction nor an existentially quantified formula. Then let

$$\Delta_{k0} := \Delta_{k1} := \Delta_k \cup \{A_n\} \quad \text{and} \quad V_{k0} := V_{k1} := V_k.$$

REMARK. (1) Because of $\vdash \exists_x \top$ and this formula is repeated infinitely often in the given enumeration, for every variable $x_i$ there is an $m$ such that $x_i \in V_k$ for all $k$ with $\mathrm{lh}(k) = m$.

(2) Obviously $\mathrm{FV}(\Delta_k) \subseteq V_k$, and $k \preceq k'$ implies that $\Delta_k \subseteq \Delta_{k'}$.

We note that

$$\forall_{k' \succeq_n k} (\Gamma, \Delta_{k'} \vdash B) \Rightarrow \Gamma, \Delta_k \vdash B, \quad \text{provided } \mathrm{FV}(B) \subseteq V_k. \tag{1.16}$$

It is sufficient to show that, for $\mathrm{FV}(B) \subseteq V_k$,

$$\Gamma, \Delta_{k0} \vdash B \quad \text{and} \quad \Gamma, \Delta_{k1} \vdash B \quad \text{imply} \quad \Gamma, \Delta_k \vdash B.$$

In cases 0, 1 and 4, this is obvious. For case 2, the claim follows immediately from the axiom scheme $\vee^-$. In case 3, we have $\mathrm{FV}(A_n) \subseteq V_k$ and $\Gamma_n, \Delta_k \vdash_n A_n = \exists_x A'_n(x)$. Assume $\Gamma, \Delta_k \cup \{A_n, A'_n(x_i)\} \vdash B$ with $x_i \notin V_k$, and $\mathrm{FV}(B) \subseteq V_k$. Then $x_i \notin \mathrm{FV}(\Delta_k \cup \{A_n, B\})$, hence $\Gamma, \Delta_k \cup \{A_n\} \vdash B$ by $\exists^-$ and therefore $\Gamma, \Delta_k \vdash B$.

Next, we show

$$\Gamma, \Delta_k \vdash B \Rightarrow \exists_n \forall_{k' \succeq_n k} (B \in \Delta_{k'}), \quad \text{provided } \mathrm{FV}(B) \subseteq V_k. \tag{1.17}$$

Choose $n \geq \mathrm{lh}(k)$ such that $B = A_n$ and $\Gamma_n, \Delta_k \vdash_n A_n$. For all $k' \succeq k$, if $\mathrm{lh}(k') = n + 1$ then $A_n \in \Delta_{k'}$ (cf. the cases 2-4).

Using the sets $\Delta_k$ we can define an $\mathcal{L}$-Beth model $\mathcal{B}$ as $(\mathrm{Ter}_\mathcal{L}, I_0, I_1)$ (where $\mathrm{Ter}_\mathcal{L}$ denotes the set of terms of $\mathcal{L}$) and the canonical $I_0(f)\vec{s} := f\vec{s}$ and

$$R^\mathcal{B}(\vec{s}, k) \;:\Longleftrightarrow\; R\vec{s} \in \Delta_k.$$

Obviously, $t^\mathcal{B}[\mathrm{id}] = t$ for all $\mathcal{L}$-terms $t$.

Write $k \Vdash B$ for $\mathcal{B}, k \Vdash B[\mathrm{id}]$. We show that

$$\Gamma, \Delta_k \vdash B \iff k \Vdash B, \quad \text{provided } \mathrm{FV}(B) \subseteq V_k. \tag{1.18}$$

The proof is by induction on $B$. *Case* $R\vec{s}$. Assume $\mathrm{FV}(R\vec{s}) \subseteq V_k$. $\Rightarrow$.

$$\begin{array}{ll}
\Gamma, \Delta_k \vdash R\vec{s} & \\
\exists_n \forall_{k' \succeq_n k}\, (R\vec{s} \in \Delta_{k'}) & \text{by (1.17)} \\
\exists_n \forall_{k' \succeq_n k}\, R^\mathcal{B}(\vec{s}, k') & \text{by definition of } \mathcal{B} \\
k \Vdash R\vec{s} & \text{by definition of } \Vdash, \text{ since } t^\mathcal{B}[\mathrm{id}] = t.
\end{array}$$

$\Leftarrow$.

$$\begin{array}{ll}
k \Vdash R\vec{s} & \\
\exists_n \forall_{k' \succeq_n k}\, R^\mathcal{B}(\vec{s}, k') & \text{by definition of } \Vdash, \text{ since } t^\mathcal{B}[\mathrm{id}] = t. \\
\exists_n \forall_{k' \succeq_n k}\, (R\vec{s} \in \Delta_{k'}) & \text{by definition of } \mathcal{B} \\
\Gamma, \Delta_k \vdash R\vec{s} & \text{by (1.16).}
\end{array}$$

*Case* $B \vee C$. Assume $\mathrm{FV}(B \vee C) \subseteq V_k$. $\Rightarrow$. Let $\Gamma, \Delta_k \vdash B \vee C$. Choose an $n \geq \mathrm{lh}(k)$ such that $\Gamma_n, \Delta_k \vdash_n A_n = B \vee C$. Then, for all $k' \succeq k$ s.t. $\mathrm{lh}(k') = n$ it follows that

$$\Delta_{k'0} = \Delta_{k'} \cup \{B \vee C, B\} \quad \text{and} \quad \Delta_{k'1} = \Delta_{k'} \cup \{B \vee C, C\},$$

and by IH

$$k'0 \Vdash B \quad \text{and} \quad k'1 \Vdash C.$$

By definition, we have $k \Vdash B \vee C$. $\Leftarrow$.

$$\begin{array}{ll}
k \Vdash B \vee C & \\
\exists_n \forall_{k' \succeq_n k}\, (k' \Vdash B \;\text{ or }\; k' \Vdash C) & \\
\exists_n \forall_{k' \succeq_n k}\, (\Gamma, \Delta_{k'} \vdash B \;\text{ or }\; \Gamma, \Delta_{k'} \vdash C) & \text{by IH} \\
\exists_n \forall_{k' \succeq_n k}\, (\Gamma, \Delta_{k'} \vdash B \vee C) & \\
\Gamma, \Delta_k \vdash B \vee C & \text{by (1.16).}
\end{array}$$

The case $B \wedge C$ is evident.

*Case* $B \to C$. Assume $\mathrm{FV}(B \to C) \subseteq V_k$. $\Rightarrow$. Let $\Gamma, \Delta_k \vdash B \to C$. We must show $k \Vdash B \to C$, i.e.,

$$\forall_{k' \succeq k}(k' \Vdash B \Rightarrow k' \Vdash C).$$

Let $k' \succeq k$ be such that $k' \Vdash B$. By IH, it follows that $\Gamma, \Delta_{k'} \vdash B$, and $\Gamma, \Delta_{k'} \vdash C$ follows by assumption. Then again by IH $k' \Vdash C$.

$\Leftarrow$. Let $k \Vdash B \to C$, i.e., $\forall_{k' \succeq k}(k' \Vdash B \Rightarrow k' \Vdash C)$. We show that $\Gamma, \Delta_k \vdash B \to C$. At this point, we apply (1.16). Choose an $n \geq \mathrm{lh}(k)$ such that $B = A_n$. Let $k' \succeq_m k$ be such that $m := n - \mathrm{lh}(k)$. We show that $\Gamma, \Delta_{k'} \vdash B \to C$.

If $\Gamma, \Delta_{k'} \vdash_n A_n$, then $k' \Vdash B$ by IH, and $k' \Vdash C$ by assumption, hence $\Gamma, \Delta_{k'} \vdash C$ again by IH and thus $\Gamma, \Delta_{k'} \vdash B \to C$.

If $\Gamma, \Delta_{k'} \nvdash_n A_n$, then by definition $\Delta_{k'1} = \Delta_{k'} \cup \{B\}$, hence $\Gamma, \Delta_{k'1} \vdash B$, and $k'1 \Vdash B$ by IH. Now $k'1 \Vdash C$ by assumption, and finally $\Gamma, \Delta_{k'1} \vdash C$ by IH. From $\Delta_{k'1} = \Delta_{k'} \cup \{B\}$, it follows that $\Gamma, \Delta_{k'} \vdash B \to C$.

*Case* $\forall_x B(x)$. Assume $\mathrm{FV}(\forall_x B(x)) \subseteq V_k$. $\Rightarrow$. Let $\Gamma, \Delta_k \vdash \forall_x B(x)$. Fix a term $t$. Then $\Gamma, \Delta_k \vdash B(t)$. Choose $n$ such that $\mathrm{FV}(B(t)) \subseteq V_{k'}$ for all $k' \succeq_n k$. Then $\forall_{k' \succeq_n k}(\Gamma, \Delta_{k'} \vdash B(t))$, hence $\forall_{k' \succeq_n k}(k' \Vdash B(t))$ by IH, hence $k \Vdash B(t)$ by the Covering Lemma. This holds for every term $t$, hence $k \Vdash \forall_x B(x)$.

$\Leftarrow$. Assume $k \Vdash \forall_x B(x)$. Pick $k' \succeq_n k$ such that $A_m = \exists_x \top$, for $m := \mathrm{lh}(k) + n$. Then at height $m$ we put some $x_i$ into the variable sets: for $k' \succeq_n k$ we have $x_i \notin V_{k'}$ but $x_i \in V_{k'j}$. Clearly $k'j \Vdash B(x_i)$, hence $\Gamma, \Delta_{k'j} \vdash B(x_i)$ by IH, hence (since at this height we consider the trivial formula $\exists_x \top$) also $\Gamma, \Delta_{k'} \vdash B(x_i)$. Since $x_i \notin V_{k'}$ we obtain $\Gamma, \Delta_{k'} \vdash \forall_x B(x)$. This holds for all $k' \succeq_n k$, hence $\Gamma, \Delta_k \vdash \forall_x B(x)$ by (1.16).

*Case* $\exists_x B(x)$. Assume $\mathrm{FV}(\exists_x B(x)) \subseteq V_k$.

$\Rightarrow$. Let $\Gamma, \Delta_k \vdash \exists_x B(x)$. Choose an $n \geq \mathrm{lh}(k)$ such that $\Gamma_n, \Delta_k \vdash_n A_n = \exists_x B(x)$. Then, for all $k' \succeq k$ such that $\mathrm{lh}(k') = n$ it follows that

$$\Delta_{k'0} = \Delta_{k'1} = \Delta_k \cup \{\exists_x B(x), B(x_i)\}$$

with $x_i \notin V_{k'}$. Hence by IH for $B(x_i)$ (applicable since $\mathrm{FV}(B(x_i)) \subseteq V_{k'j}$ for $j = 0, 1$)

$$k'0 \Vdash B(x_i) \quad \text{and} \quad k'1 \Vdash B(x_i).$$

It follows by definition that $k \Vdash \exists_x B(x)$.

$\Leftarrow$. Assume $k \Vdash \exists_x B(x)$. Then $\forall_{k' \succeq_n k} \exists_{t \in \mathrm{Ter}}(k' \Vdash B(x)[\mathrm{id}_x^t])$ for some $n$, hence $\forall_{k' \succeq_n k} \exists_{t \in \mathrm{Ter}}(k' \Vdash B(t))$. For each of the finitely many $k' \succeq_n k$ pick an $m$ such that $\forall_{k'' \succeq_m k'}(\mathrm{FV}(B(t_{k'})) \subseteq V_{k''})$. Let $m_0$ be the maximum of all these $m$. Then

$$\forall_{k'' \succeq_{m_0+n} k} \exists_{t \in \mathrm{Ter}}(k'' \Vdash B(t) \quad \text{and} \quad \mathrm{FV}(B(t)) \subseteq V_{k''}).$$

The IH for $B(t)$ yields

$$\begin{aligned}
&\forall_{k'' \succeq_{m_0+n} k} \exists_{t \in \mathrm{Ter}}(\Gamma, \Delta_{k''} \vdash B(t)) \\
&\forall_{k'' \succeq_{m_0+n} k}(\Gamma, \Delta_{k''} \vdash \exists_x B(x)) \\
&\Gamma, \Delta_k \vdash \exists_x B(x) \qquad \text{by (1.16).}
\end{aligned}$$

Now, we can finish the proof of the Completeness Theorem. We apply (b) to the Beth model $\mathcal{B}$ constructed above from $\Gamma$, the empty node $\langle\rangle$ and the assignment $\eta = \mathrm{id}$. Then $\mathcal{B}, \langle\rangle \Vdash \Gamma[\mathrm{id}]$ by (1.18), hence $\mathcal{B}, \langle\rangle \Vdash A[\mathrm{id}]$ by assumption and therefore $\Gamma \vdash A$ by (1.18) again. □

Completeness of intuitionistic logic follows as a corollary.

COROLLARY. *Let $\Gamma \cup \{A\}$ be a set of formulas. The following propositions are equivalent.*

(a) $\Gamma \vdash_i A$.
(b) $\Gamma, \mathrm{Efq} \Vdash A$, *i.e., for all Beth models $\mathcal{B}$ for the intuitionistic logic, nodes $k$ and assignments $\eta$*

$$\mathcal{B}, k \Vdash \Gamma[\eta] \Rightarrow \mathcal{B}, k \Vdash A[\eta].$$

□

## 1.6. Soundness and Completeness of the Classical Fragment

We will prove completeness by means of a technique due to Beth, Hintikka and Schütte (BHS-technique for short); it consists in simultaneously searching for a derivation and a counterexample. This proof is non-constructive: it makes use of the *principle of omniscience* (Bishop and Bridges, 1985, p.11) for the property of being a bound for the height of a binary tree; we call this principle the *infinity test* for binary trees. Using DC (dependent choice), this principle also suffices to prove König's Lemma, which is a crucial ingredient of completeness proofs with the BHS-technique.

**1.6.1. Models.** We define the notion of a model (or more accurately, $\mathcal{L}$-model), and what the value of a term and the meaning of a formula in such a model should be. The latter definition is by induction on formulas, where in the quantifier case we need a quantifier in the definition.

For the rest of this section, fix a countable formal language $\mathcal{L}$; we do not mention the dependence on $\mathcal{L}$ in the notation. Recall that by Lemma 1.2.4 it is not necessary to consider $\wedge$. So let us assume that $\mathcal{L}$ is based on $\to$, $\forall$ and $\bot$.

DEFINITION. $\mathcal{M} = (D, I)$ is a *pre-model*, if $D$ a non-empty set (the *carrier set* or the *domain* of $\mathcal{M}$) and $I$ is a map (*interpretation*) assigning to every $n$-ary function symbol $f$ of $\mathcal{L}$ a function $I(f)\colon D^n \to D$. In case $n = 0$, $I(f)$ is an element of $D$. $\mathcal{M} = (D, I_0, I_1)$ is a *model*, if $(D, I_0)$ is a pre-model and $I_1$ a map assigning to every $n$-ary relation symbol $R$ of $\mathcal{L}$ an $n$-ary relation on $D^n$. In case $n = 0$, $I_1(R)$ is either true or false; in particular we require that $I_1(\bot)$ is false.

If $\mathcal{M} = (D, I)$ or $(D, I_0, I_1)$, then we often write $|\mathcal{M}|$ for the carrier set $D$ of $\mathcal{M}$ and $f^{\mathcal{M}}$, $R^{\mathcal{M}}$ for the interpretations $I_0(f)$, $I_1(R)$ of the function and relation symbols.

An *assignment* (or variable assignment) in $D$ is a map $\eta$ assigning to every variable $x \in \mathrm{dom}(\eta)$ a value $\eta(x) \in D$. Finite assignments will be written as $[x_1 := a_1, \ldots, x_n := a_n]$ (or else as $[a_1/x_1, \ldots, a_n/x_n]$), with distinct $x_1, \ldots, x_n$. If $\eta$ is an assignment in $D$ and $a \in D$, let $\eta^a_x$ be the assignment in $D$ mapping $x$ to $a$ and coinciding with $\eta$ elsewhere, so

$$\eta^a_x(y) := \begin{cases} \eta(y), & \text{if } y \neq x \\ a, & \text{if } y = x. \end{cases}$$

Let a pre-model $\mathcal{M}$ and an assignment $\eta$ in $|\mathcal{M}|$ be given. We define a homomorphic extension of $\eta$ (denoted by $\eta$ as well) to the set Ter of terms $t$ such that $\mathrm{vars}(t) \subseteq \mathrm{dom}(\eta)$ by

$$\begin{aligned} \eta(c) &:= c^{\mathcal{M}}, \\ \eta(f(t_1, \ldots, t_n)) &:= f^{\mathcal{M}}(\eta(t_1), \ldots, \eta(t_n)). \end{aligned}$$

Observe that the extension of $\eta$ depends on $\mathcal{M}$; we sometimes write $t^{\mathcal{M}}[\eta]$ for $\eta(t)$.

Definition (Validity). For every model $\mathcal{M}$, assignment $\eta$ in $|\mathcal{M}|$ and formula $A$ such that $\mathrm{FV}(A) \subseteq \mathrm{dom}(\eta)$ we define $\mathcal{M} \models A[\eta]$ (read: $A$ is *valid* in $\mathcal{M}$ under the assignment $\eta$) by induction on $A$, with the following clauses.

$$\begin{aligned} \mathcal{M} \models R(t_1, \ldots, t_n)[\eta] &:\Longleftrightarrow R^{\mathcal{M}}(t_1^{\mathcal{M}}[\eta], \ldots, t_n^{\mathcal{M}}[\eta]), \\ \mathcal{M} \models (A \to B)[\eta] &:\Longleftrightarrow \big((\mathcal{M} \models A[\eta]) \to (\mathcal{M} \models B[\eta])\big), \\ \mathcal{M} \models (\forall_x A)[\eta] &:\Longleftrightarrow \forall_{a \in |\mathcal{M}|} \mathcal{M} \models A[\eta^a_x]. \end{aligned}$$

Since $I_1(\bot)$ is false, we have in particular $\mathcal{M} \not\models \bot[\eta]$.

If $\Gamma$ is a set of formulas, we write $\mathcal{M} \models \Gamma[\eta]$, if $\mathcal{M} \models A[\eta]$ for all $A \in \Gamma$. If $\mathcal{M} \models A[\eta]$ for all assignments $\eta$ in $|\mathcal{M}|$, we write $\mathcal{M} \models A$.

Lemma (Coincidence). *Let $\mathcal{M}$ be a model, $t$ a term, $A$ a formula and $\eta, \xi$ assignments in $|\mathcal{M}|$.*

(a) *If $\eta(x) = \xi(x)$ for all $x \in \mathrm{vars}(t)$, then $\eta(t) = \xi(t)$.*
(b) *If $\eta(x) = \xi(x)$ for all $x \in \mathrm{FV}(A)$, then $\mathcal{M} \models A[\eta]$ if and only if $\mathcal{M} \models A[\xi]$.*

Proof. Induction on terms and formulas. □

Lemma (Substitution). *Let $\mathcal{M}$ be a model, $t, r$ terms, $A$ a formula and $\eta$ an assignment in $|\mathcal{M}|$. Then*

(a) $\eta(r[x := t]) = \eta^{\eta(t)}_x(r)$.
(b) $\mathcal{M} \models A[x := t][\eta] \iff \mathcal{M} \models A[\eta^{\eta(t)}_x]$.

Proof. Induction on terms and formulas. □

A model $\mathcal{M}$ is called *stable* if $\neg\neg R^{\mathcal{M}}(\vec{a}) \to R^{\mathcal{M}}(\vec{a})$ for all relation symbols $R$ and all $\vec{a} \in |\mathcal{M}|$.

**1.6.2. Soundness.** We prove that every formula derivable in classical logic is valid in an arbitrary stable model.

THEOREM (Soundness). *Let $\Gamma \vdash_c B$, $\mathcal{M}$ a stable model and $\eta$ an assignment in $|\mathcal{M}|$. Then $\mathcal{M} \models \Gamma[\eta]$ entails $\mathcal{M} \models B[\eta]$.*

PROOF. Induction on derivations. The given derivation of $B$ from $\Gamma$ can only have finitely many free assumptions; hence we may assume $\Gamma = \{A_1, \dots, A_n\}$.

*Case* $u\colon B$. Then $B \in \Gamma$ and the claim is obvious.

*Case* $\mathrm{Stab}_R\colon \forall\vec{x}(\neg\neg R\vec{x} \to R\vec{x})$. The claim follows from our assumption that $\mathcal{M}$ is stable, i.e., $\neg\neg R^{\mathcal{M}}(\vec{a}) \to R^{\mathcal{M}}(\vec{a})$ for all $\vec{a} \in |\mathcal{M}|$. The other axioms are clearly valid.

*Case* $\to^-$. Assume $\mathcal{M} \models \Gamma[\eta]$. We must show $\mathcal{M} \models B[\eta]$. By IH, $\mathcal{M} \models (A \to B)[\eta]$ and $\mathcal{M} \models A[\eta]$. The claim follows from the definition of $\models$.

*Case* $\to^+$. Assume $\mathcal{M} \models \Gamma[\eta]$. We must show $\mathcal{M} \models (A \to B)[\eta]$. So assume in addition $\mathcal{M} \models A[\eta]$. We must show $\mathcal{M} \models B[\eta]$. By IH (with $\Gamma \cup \{A\}$ instead of $\Gamma$) this clearly holds.

*Case* $\forall^+$. Assume $\mathcal{M} \models \Gamma[\eta]$. We must show $\mathcal{M} \models A[\eta^a_x]$. We may assume that all assumptions $A_1, \dots, A_n$ actually appear in the given derivation. Since because of the variable condition for $\forall^+$ the variable $x$ does not appear free in any of the formulas $A_1, \dots, A_n$, we have by the Coincidence Lemma $\mathcal{M} \models \Gamma[\eta^a_x]$. The IH (with $\eta^a_x$ instead of $\eta$) yields $\mathcal{M} \models A[\eta^a_x]$.

*Case* $\forall^-$. Assume $\mathcal{M} \models \Gamma[\eta]$. We must show $\mathcal{M} \models A[x := t][\eta]$, i.e., by the Substitution Lemma $\mathcal{M} \models A[\eta^b_x]$ with $b := \eta(t)$. By IH, $\mathcal{M} \models (\forall_x A)[\eta]$, i.e., $\mathcal{M} \models A[\eta^a_x]$ for all $a \in |\mathcal{M}|$. With $\eta(t)$ for $a$ the claim follows. □

**1.6.3. Completeness.** Let us first introduce some relevant notions. A *node* is a finite sequence $\kappa = \langle k_0, k_1, \dots, k_{n-1}\rangle$ of natural numbers; $n$ is called the *length* $\mathrm{lh}(\kappa)$ of $\kappa$. We write $\kappa_1 \preceq \kappa_2$, if $\kappa_1$ is an initial segment $\kappa_2$. A *tree* is a set of nodes closed under the formation of initial segments. A tree $T$ is *finitely branching*, if every node $\kappa \in T$ has only finitely many immediate continuations in $T$. A tree $T$ is *infinite*, if for every $n$ there is a node $\kappa \in T$ with $\mathrm{lh}(\kappa) = n$. A *branch* in a tree $T$ is a linearly ordered subset of $T$ closed under the formation of initial segments. Every infinite branch is determined by a sequence $(k_n)_{n\in\mathbb{N}}$ of natural numbers, whose initial segments are the nodes of the branch.

LEMMA (König). *Every finitely branching infinite tree has an infinite branch.*

PROOF. We make use of the principle of the availability of an *infinity test* for binary trees, and moreover DC (dependent choice). Let $T$ be a finitely branching infinite tree. We define recursively natural numbers $k_n$ by requiring that $k_n$ is the least $k$ such that $\langle k_0, k_1, \ldots, k_{n-1}, k\rangle$ has arbitrarily large continuations in $T$. □

REMARK. Without the assumption that $T$ is finitely branching, König's Lemma is false. To see this, consider the set $T$ of all nodes $\langle n, 0, \ldots, 0\rangle$ with $n$ occurrences of 0. Clearly $T$ is infinite, but it has no infinite branch.

THEOREM (Completeness). *Let $\Gamma \cup \{A\}$ be a set of formulas. Then*

$$\Gamma \models A \Rightarrow \Gamma \vdash_c A.$$

PROOF. We use the BHS-technique and construct a universal search tree (i.e., a tree independent of $\Gamma$ and $A$). Using this universal search tree we obtain for a given $\Gamma$ and $A$ either a derivation of $A$ from $\Gamma$ in classical logic or else a counterexample, i.e., a model of $\Gamma \cup \{\neg A\}$.

In order to define this universal search tree we need at certain points new variables. Clearly we can assume that there is a countably infinite set $V$ of variables that occur neither in $\Gamma$ nor in $A$. Let AllFor be the set of universal formulas and Ter the set of terms of $\mathcal{L}$. Since $\mathcal{L}$ is assumed to be countable, we can assume that we have a fixed enumeration of AllFor $\times$ Ter; we write $(\forall_y C, s) \sqsubset (\forall_x B, t)$ if $(\forall_y C, s)$ comes before $(\forall_x B, t)$ in this enumeration. Then we can define recursively an injective map new: AllFor $\times$ Ter $\to V$ such that new$(\forall_x B, t)$ does not occur in any $(\forall_y C, s)$ such that $(\forall_y C, s) \sqsubseteq (\forall_x B, t)$.

Recall that we may assume that neither $\Gamma$ nor $A$ contains the connective $\wedge$. Let AtomFor be the set of atomic formulas and ImpFor the set of implication formulas of $\mathcal{L}$. Consider

$$I := \text{AtomFor} \cup (\text{ImpFor} \times \{0,1\}) \cup (\text{AllFor} \times \text{Ter})$$

as an index set. For every $F \in I$ and every $k \in \{0,1\}$ we define a finite set $\Delta_k(F)$ of formulas by

$$\Delta_k(R\vec{t}) := \begin{cases} \{\neg R\vec{t}\}, & \text{if } k = 0, \\ \{R\vec{t}\}, & \text{if } k = 1, \end{cases}$$

$$\Delta_k(B \to C, i) := \begin{cases} \{\neg(B \to C), B, \neg C\}, & \text{if } k = 0, \\ \{B \to C, \neg B\}, & \text{if } k = 1 \text{ and } i = 0, \\ \{B \to C, C\}, & \text{if } k = 1 \text{ and } i = 1, \end{cases}$$

$$\Delta_k(\forall_x B(x), t) := \begin{cases} \{\neg\forall_x B(x), \neg B(y)\}, & \text{if } k = 0 \text{ and } y = \text{new}(\forall_x B(x), t), \\ \{\forall_x B(x), B(t)\}, & \text{if } k = 1. \end{cases}$$

A partial map $\alpha$ from $I$ into $\{0,1\}$ will be called a *search path* or shortly *path*. For every search path $\alpha$ we define a set $\Gamma_\alpha$ of formulas by

$$\Gamma_\alpha := \bigcup_{F \in \mathrm{dom}(\alpha)} \Delta_{\alpha(F)}(F).$$

$\alpha$ is called $\Gamma, A$-*blocked* or shortly *blocked* if

- $A \in \Gamma_\alpha$ or $\neg B \in \Gamma_\alpha$ for some $B \in \Gamma$, or
- $B, \neg B \in \Gamma_\alpha$ for some $B$, or
- $\bot \in \Gamma_\alpha$.

*Positive main case.* There is an $I_0 \subseteq_{\mathrm{fin}} I$ such that all search paths $\alpha$ through $I_0$ (i.e., with $\mathrm{dom}(\alpha) = I_0$) are blocked.

We show that in this case $\Gamma \vdash_c A$. To this end let us assume that $I_0$ is ordered in such a way that behind all elements of AllFor $\times$ Ter ordered by $\sqsubset$ we list all formulas from AtomFor and (ImpFor $\times \{0,1\}$). Let $J$ range over all subsets of $I_0$ forming initial segments w.r.t. this ordering. We show that for every such $J$ and all $\alpha$ through $J$ we have $\Gamma_\alpha \cup \Gamma \vdash_c A$, by induction on the number of elements in $I_0 \setminus J$. With $J = \emptyset$ we obtain the claim $\Gamma \vdash_c A$.

In the base case we have $J = I_0$. Then by assumption $\alpha$ is blocked; hence $\Gamma_\alpha \cup \Gamma \vdash_c A$. In the step we distinguish three cases.

*Case* $\{R\vec{t}\} \dot\cup J$. Let $\alpha$ be a search path through $J$. By IH we have $\Gamma_{\alpha[k/R\vec{t}]} \cup \Gamma \vdash_c A$ for every $k \in \{0,1\}$, hence

$$\{\neg R\vec{t}\} \cup \Gamma_\alpha \cup \Gamma \vdash_c A \qquad \text{and} \qquad \{R\vec{t}\} \cup \Gamma_\alpha \cup \Gamma \vdash_c A.$$

Using $\vdash_c (R\vec{t} \to A) \to (\neg R\vec{t} \to A) \to A$ we obtain $\Gamma_\alpha \cup \Gamma \vdash_c A$.

*Case* $\{(B \to C, i)\} \dot\cup J$ . Let $\alpha$ be a search path through $J$. By IH we have $\Gamma_{\alpha[k/(B\to C,i)]} \cup \Gamma \vdash_c A$ for every $k, i \in \{0,1\}$, hence

$$\{\neg(B \to C), B, \neg C\} \cup \Gamma_\alpha \cup \Gamma \vdash_c A, \tag{1.19}$$

$$\{B \to C, \neg B\} \cup \Gamma_\alpha \cup \Gamma \vdash_c A, \tag{1.20}$$

$$\{B \to C, C\} \cup \Gamma_\alpha \cup \Gamma \vdash_c A. \tag{1.21}$$

Because of $\vdash_c \neg(B \to C) \to B$ and $\vdash_c \neg(B \to C) \to \neg C$ we obtain from (1.19)

$$\{\neg(B \to C)\} \cup \Gamma_\alpha \cup \Gamma \vdash_c A. \tag{1.22}$$

Since $\vdash_c (\neg B \to A) \to (C \to A) \to (B \to C) \to A$, from (1.20) and (1.21) we get

$$\{B \to C\} \cup \Gamma_\alpha \cup \Gamma \vdash_c A. \tag{1.23}$$

From (1.22) and (1.23) we now obtain $\Gamma_\alpha \cup \Gamma \vdash_c A$, using case distiction again.

*Case* $\{(\forall_x B(x), t)\} \dot\cup J$ with $J \subseteq \mathrm{AllFor} \times \mathrm{Ter}$ and $(\forall_x B(x), t) \sqsupset (\forall_y C(y), s)$ for all $(\forall_y C(y), s) \in J$. Let $\alpha$ be a search path through $J$. By IH we have $\Gamma_{\alpha[k/(\forall_x B(x),t)]} \cup \Gamma \vdash_c A$ for every $k \in \{0, 1\}$, hence

$$\{\neg\forall_x B(x), \neg B(y)\} \cup \Gamma_\alpha \cup \Gamma \vdash_c A,$$
$$\{\forall_x B(x), B(t)\} \cup \Gamma_\alpha \cup \Gamma \vdash_c A,$$

where $y$ does not occur in $\forall_x B(x)$, $\Gamma_\alpha$, $\Gamma$ and $A$. Using

$$\vdash_c (\forall_x B(x) \to A) \to \forall_y(\neg\forall_x B(x) \to \neg B(y) \to A) \to A$$

for $y \notin \mathrm{FV}(\forall_x B(x), A)$ we obtain $\Gamma_\alpha \cup \Gamma \vdash_c A$, again using case distiction.

*Negative main case.* For every $I_0 \subseteq_{\mathrm{fin}} I$ there is a non-blocked search path $\alpha$ through $I_0$.

We show that in this case there is a non-blocked search path $\alpha$ through all of $I$. From $\alpha$ we will construct a counterexample to the hypothesis of the Completeness Theorem.

Since $\mathcal{L}$ was assumed to be countable, $I$ is countable as well and can be assumed to be given in the form $\{ F_n \mid n \in \mathbb{N} \}$ with $F_n \neq F_m$ for $n \neq m$. Let

$$T := \{ \kappa \mid \exists_{\alpha \text{ non-blocked}} \forall_{n<\mathrm{lh}(\kappa)} . F_n \in \mathrm{dom}(\alpha) \wedge \kappa(n) = \alpha(F_n) \}.$$

Clearly $T$ is a tree, i.e., closed against the formation of initial segments Moreover, $T$ is finitely branching, since we always have $\alpha(F) \in \{0, 1\}$. $T$ is infinite because of the assumption in the negative main case. By König's Lemma there is an infinite branch in $T$, determined say by $(k_n)_{n\in\mathbb{N}}$. Let $\alpha(F_n) := k_n$. Then $\alpha$ is a non-blocked search path through all of $I$.

Using this non-blocked search path $\alpha$ and the set of formulas

$$\Gamma_\alpha = \bigcup_{F\in I} \Delta_{\alpha(F)}(F)$$

determined by it, we can now construct the required counterexample. First we collect some properties of $\Gamma_\alpha$ that follow immediately from its definition.

(a) For every formula $B$, either $B \in \Gamma_\alpha$ or $\neg B \in \Gamma_\alpha$.
(b) $\bot \notin \Gamma_\alpha$.
(c) $A \notin \Gamma_\alpha$, and $\Gamma \subseteq \Gamma_\alpha$.
(d) If $\neg(B \to C) \in \Gamma_\alpha$, then $B \in \Gamma_\alpha$ and $\neg C \in \Gamma_\alpha$.
(e) If $B \to C \in \Gamma_\alpha$, then $\neg B \in \Gamma_\alpha$ or $C \in \Gamma_\alpha$.
(f) If $\neg\forall_x B(x) \in \Gamma_\alpha$, then $\neg B(y) \in \Gamma_\alpha$ for some $y$.
(g) If $\forall_x B(x) \in \Gamma_\alpha$, then $B(t) \in \Gamma_\alpha$ for all terms $t$.

Define aa model $\mathcal{M}$ by

$$\mathcal{M} := (\mathrm{Ter}, I_0, I_1),$$

where

$$I_0(f)(t_1,\dots,t_n) := f(t_1,\dots,t_n),$$
$$R^{\mathcal{M}} := \{\, (t_1,\dots,t_n) \mid R(t_1,\dots,t_n) \in \Gamma_\alpha \,\}.$$

Write $\mathcal{M} \models B$ for $\mathcal{M} \models B[\mathrm{id}]$. We show

$$\mathcal{M} \models B \iff B \in \Gamma_\alpha. \tag{1.24}$$

The proof is by induction on the number of logical connectives $\to, \forall$ in $B$. Notice that for our canonical definition of $I_0(f)$ we have $\overline{\mathrm{id}}(t) = t$ for all terms $t$.

*Case* $R\vec{t}$. $\mathcal{M} \models R\vec{t}$ by definition means $R\vec{t} \in \Gamma_\alpha$.

*Case* $\bot$. By (b) we have $\bot \notin \Gamma_\alpha$.

*Case* $B \to C$. $\Rightarrow$. Assume $\mathcal{M} \models B \to C$. We must show $B \to C \in \Gamma_\alpha$. By (a) it suffices to know $\neg(B \to C) \notin \Gamma_\alpha$. So assume $\neg(B \to C) \in \Gamma_\alpha$. Then by (d) $B \in \Gamma_\alpha$ and $\neg C \in \Gamma_\alpha$, hence by (a) $C \notin \Gamma_\alpha$. By IH it follows that $\mathcal{M} \models B$ und $\mathcal{M} \not\models C$, contradicting the assumption $\mathcal{M} \models B \to C$.

$\Leftarrow$. Assume $B \to C \in \Gamma_\alpha$. We must show $\mathcal{M} \models B \to C$. So let $\mathcal{M} \models B$. We must show $\mathcal{M} \models C$. By IH we have $B \in \Gamma_\alpha$, hence by (e) and (a) $C \in \Gamma_\alpha$, hence by IH $\mathcal{M} \models C$.

*Case* $\forall_x B(x)$. $\Rightarrow$. Assume $\mathcal{M} \models \forall_x B(x)$. We must show $\forall_x B(x) \in \Gamma_\alpha$. By (a) it suffices to know $\neg\forall_x B(x) \notin \Gamma_\alpha$. So assume $\neg\forall_x B(x) \in \Gamma_\alpha$. Then by (f) $\neg B(y) \in \Gamma_\alpha$ for some $y$, hence by (a) and the IH $\mathcal{M} \not\models B(y)$, hence by the Substitution Lemma $\mathcal{M} \not\models B(x)[\mathrm{id}^y_x]$, contradicting the assumption $\mathcal{M} \models \forall_x B(x)$.

$\Leftarrow$. Assume $\forall_x B(x) \in \Gamma_\alpha$. We must show $\mathcal{M} \models \forall_x B(x)$. So let $t$ be an arbitrary term. We must show $\mathcal{M} \models B(x)[\mathrm{id}^t_x]$, hence $\mathcal{M} \models B(t)$ by the Substitution Lemma. But from (g) we obtain $B(t) \in \Gamma_\alpha$, hence by IH for $B(t)$ also $\mathcal{M} \models B(t)$.

Now we can finish the proof of the Completeness Theorem. Because of (c) we have $A \notin \Gamma_\alpha$ and $\Gamma \subseteq \Gamma_\alpha$, hence $\mathcal{M} \not\models A$ and $\mathcal{M} \models B$ for all $B \in \Gamma$, i.e., $\mathcal{M}$ and the assignment id form a counterexample to the hypothesis $\Gamma \models A$ of the Completeness Theorem. □

## 1.7. Notes

The proof of the existence of normal forms w.r.t permutative conversions is originally due to Prawitz (1965). We have adapted a method developed by Joachimski and Matthes (2003), which in turn is based on van Raamsdonk and Severi (1995).

The remark in Sec.1.3.4 concerning arithmetical comprehension is essentially due to Takeuti (1978); it has been extended by Troelstra (1973).

The constructive completeness proof of minimal logic w.r.t. Beth models in Sec. 1.5 is due to Friedman (1975). An exposition of (a version of) this proof, together with an extensive discussion of the history of completeness proofs for minimal and intuitionistic logic, can be found in (Troelstra and van Dalen, 1988).

Loeb (2005) shows that the completeness theorem of classical propositional calculus is equivalent to the Fan Theorem.

Kolmogorov (1925) provided the first translation of classical propositional logic into minimal logic, by inserting double negations everywhere. Gödel (1932) and Gentzen (independently, about the same time, but unpublished) rediscovered this translation, in a somewhat simplified form. For implication, Gödel had $(A \to B)^T := \neg(A^T \wedge \neg B^T)$, whereas Gentzen had $(A \to B)^T := A^T \to B^T$. So Gödel translated into the $\wedge, \neg$-language, whereas Gentzen had the $\to, \neg$-language instead. Both Gödel and Gentzen extended the translation to first order logic. The fact that classical logic can be embedded into intuitionistic logic came as a surprise at the time. Gödel and Bernays (see Hilbert and Bernays (1968)) since distinguished between "finitary" and "intuitionistic" reasoning.

Harrop formulas are called Rasiowa-Harrop formulas in Troelstra and van Dalen (1988), for they were considered by Rasiowa before Harrop came across this notion. However, we continue to use the more common name here.

CHAPTER 2

# Computation with Partial Continuous Functionals

The logic considered up to now is very general, and for instance does not allow to speak of natural numbers. We therefore introduce inductive types (or free algebras) as base domains; they are given by constructors. We also allow function spaces, with the inductive types as base types. We specialize our minimal logic to a simply typed one, where the variables are typed, and the formation of terms is adapted. We add induction axioms, to express the minimality of the inductive types. Every inductive type comes with a recursion operator, which has certain conversion or definitional equality rules associated with it. We prove that every term (possibly with free variables) can be converted into normal (or canonical) form.

We describe a constructive theory of computable functionals, based on the partial continuous functionals as their intendend domain. Such a task had been started by Scott (1969), under the well-known abbreviation LCF. However, the prime example of such a theory, the type theory of Martin-Löf (1984), in its present form deals with total (structural recursive) functionals only. An early attempt of Martin-Löf (1983) to give a domain theoretic interpretation of his type theory has not even been published, probably because it was felt that a more general approach – such as formal topology, see Coquand et al. (2003) – would be more appropriate.

Here we try to make a fresh start, and do full justice to the fundamental notion of computability in finite types, with the partial continuous functionals as underlying domains. The total ones then appear as a dense subset (Kreisel, 1959; Ershov, 1972; Berger, 1993b; Stoltenberg-Hansen et al., 1994; Schwichtenberg, 1996; Kristiansen and Normann, 1997), and seem to be best treated in this way.

*Computable functionals and logic.* Types are built from base types by the formation of function types, $\rho \Rightarrow \sigma$. As domains for the base types we choose non-flat (cf. Fig. 2 on page 66) and possibly infinitary free algebras, given by their constructors. The main reason for taking non-flat base domains is that we want the constructors to be injective and with disjoint ranges.

The naive model of such a finitely typed theory is the full set theoretic hierarchy of functionals of finite types. However, this immediately leads

to higher cardinalities, and does not lend itself well for a theory of computability. A more appropriate semantics for typed languages has its roots in (Kreisel, 1959) (which used formal neighborhoods) and (Kleene, 1959). This line of research was taken up and developed in a mathematically more satisfactory way by Scott (1970) and Ershov (1974). Today this theory is usually presented in the context of abstract domain theory (see Stoltenberg-Hansen et al. (1994); Abramsky and Jung (1994)); it is based on classical logic.

The present work can be seen as an attempt to develop a constructive theory of formal neighborhoods for continuous functionals, in a direct and intuitive style. The task is to replace abstract domain theory by a more concrete and (in case of finitary free algebras) finitary theory of representations. As a framework we use Scott's information systems (see Scott (1982); Larsen and Winskel (1991); Stoltenberg-Hansen et al. (1994)). It turns out that we only need to deal with "atomic" and "coherent" information systems (abbreviated acis), which simplifies matters considerably. In this setup the basic notion is that of a "token", or unit of information. The elements of the domain appear as abstract or "ideal" entitites: possibly infinite sets of tokens, which are "consistent" and "deductively closed".

*Total functionals.* One reason to be interested in total functionals is that for base types, that is free algebras, we can prove properties of total objects by structural induction. This is also true for the more general class of *structure-total* objects, where the arguments at parameter positions in constructor terms need not be total. An example is a list whose length is determined, but whose elements need not be total.

We show that the standard way to single out the total functionals from the partial ones works with non-flat base domains as well, and that Berger's proof (1993b) of Kreisel's (1959) density theorem can be adapted.

*Terms and their denotational and operational semantics.* Since we have introduced domains via concrete representations, it is easy to define the computable functionals, simply as recursively enumerable ideals (= sets of tokens). However, this way to deal with computability is too general for concrete applications. In practice, one wants to define computable functionals by recursion equations. We show that and how computation rules (see Berger et al. (2003); Berger (2005)) can be used to achieve this task. The meaning $[\![\lambda\vec{x}\, M]\!]$ of a term $M$ (with free variables in $\vec{x}$) involving constants $D$ defined by computation rules will be an inductively defined set of tokens $(\vec{U}, b)$, of the type of $\lambda\vec{x}\, M$.

So we extend the term language of Plotkin's PCF (1977), by constants defined via "computation rules". One instance of such rules is the definition of the fixed point operators $\mathcal{Y}_\rho$ of type $(\rho \Rightarrow \rho) \Rightarrow \rho$, by $\mathcal{Y}_\rho f = f(\mathcal{Y}_\rho f)$.

Another instance is the structural recursion operator $\mathcal{R}^\tau_{\mathrm{nat}}$, defined by

$$\mathcal{R}^\tau_{\mathrm{nat}}(f,g,0) = f, \qquad \mathcal{R}^\tau_{\mathrm{nat}}(f,g,Sn) = g(n,\mathcal{R}^\tau_{\mathrm{nat}}(f,g,n)).$$

Operationally, the term language provides some natural conversion rules to "simplify" terms: $\beta$, $\eta$, and – for every defined constant $D$ – the defining equations $D\vec{P} \mapsto M$ with non-overlapping constructor patterns $\vec{P}$; the equivalence generated by these conversions is called operational semantics. We show that the (denotational) values are preserved under conversions, including computation rules.

*Computational adequacy.* Clearly we want to know that the conversions mentioned above give rise to a "computationally adequate" operational semantics: If $[\![M]\!] = k$, then the conversion rules suffice to actually reduce $M$ to the numeral $k$. We show that this holds true in our somewhat extended setting as well, with computation rules and non-flat base domains.

*Structural recursion.* An important example of computation rules are those of the (Gödel) structural recursion operators. We prove their totality, by showing that the rules are strongly normalizing. A predicative proof of this fact has been given by Abel and Altenkirch (2000), based on Aczel's notion of a set-based relation. Our proof is predicative as well, but – being based on an extension of Tait's method of strong computability predicates – more along the standard line of such proofs. Moreover, it extends the result to the present setting.

*Related work.* The development of constructive theories of computable functionals of finite type began with Gödel's (1958). There the emphasis was on particular computable functionals, the structural (or primitive) recursive ones. In contrast to what was done later by Kreisel, Kleene, Scott and Ershov, the domains for these functionals were not constructed explicitly, but rather considered as described axiomatically by the theory.

Denotational semantics for PCF-like languages is well-developed, and usually (as in Plotkin's (1977)) done in a domain-theoretic setting. The study of the semantics of non-overlapping higher type recursion equations - called here computation rules - has been initiated in Berger et al. (2003), again in a domain-theoretic setting. Recently Berger (2005) he has introduced a "strict" variant of this domain-theoretic semantics, and used it to prove strong normalization of extensions of Gödel's $\mathsf{T}$ by different versions of bar recursion. Information systems have been conceived by Scott (1982), as an intuitive approach to domains for denotational semantics. The idea to consider atomic information systems is due to Ulrich Berger (unpublished work); coherent information systems have been introduced by Plotkin (1978, p.210). Taking up Kreisel's (1959) idea of neighborhood systems, Martin-Löf developed in unpublished (but somewhat distributed) notes (1983) a domain theoretic interpretation of his type theory. The intersection type

discipline of Barendregt, Coppo, and Dezani-Ciancaglini (1983) can be seen as a different style of presenting the idea of a neighborhood system. The desire to have a more general framework for these ideas has lead Martin-Löf, Sambin and others to develop a formal topology; cf. Coquand, Sambin, Smith, and Valentini (2003).

It seems likely that the method in (Kristiansen and Normann, 1997, Section 3.5) (which is based on an idea of Ulrich Berger) can be used to prove density in the present case, but this would require some substantial rewriting.

The first proof of an adequacy theorem (not under this name) is due to Plotkin (1977, Theorem 3.1); Plotkin's proof is by induction on the types, and uses a computability predicate. A similar result in a type-theoretic setting is in Martin-Löf's notes (1983, Second Theorem). Adequacy theorems have been proved in many contexts, by Abramsky (1991); Amadio and Curien (1998); Barendregt et al. (1983); Martin-Löf (1983). Coquand and Spiwack (2005) – building on the work of Martin-Löf (1983) and Berger (2005) – observed that the adequacy result even holds for untyped languages, hence also for dependently typed ones.

The problem of proving strong normalization for extensions of typed $\lambda$-calculi by higher order rewrite rules has been studied extensively in the literature: Tait (1971); Girard (1971); Troelstra (1973); Blanqui et al. (1999); Abel and Altenkirch (2000); Berger (2005). Most of these proofs use impredicative methods (e.g., by reducing the problem to strong normalization of second order propositional logic, called system $F$ by Girard (1971)). Our definition of the strong computability predicates and also the proof are related to Zucker's (1973) proof of strong normalization of his term system for recursion on the first three number or tree classes. However, Zucker uses a combinatory term system and defines strong computability for closed terms only. Following some ideas in an unpublished note of Berger, Benl (in his diploma thesis (1998)) transferred this proof to terms in simply typed $\lambda$-calculus, possibly involving free variables. Here it is adapted to the present context.

*Organization of the chapter.* In Sec.2.1 atomic coherent information systems are defined, and used as a concrete representation of the relevant domains, based on non-flat and possibly infinitary free algebras. Sec.2.5 deals with total and structure-total ideals; it is shown that the density theorem holds. Sec.2.3 introduces the term language, extending Plotkin's PCF by defined constants and computation rules. The denotational and operational semantics is defined, the former by an inductive definition of a relation $(\vec{U}, b) \in [\![\lambda\vec{x}\, M]\!]$, the latter by conversions which include the computation rules. We prove preservation of values under conversions. Sec.2.4 contains

the proof of the adequacy theorem. The structural recursion operators are taken up in Sec.1.3, as an example of computation rules defining total objects. The chapter concludes in Sec.2.6 with remarks on an implementation of some of its ideas, in the Minlog proof assistant `www.minlog-system.de` under development in Munich.

## 2.1. Partial Continuous Functionals

Information systems have been introduced by Scott (1982), as an intuitive approach to deal constructively with ideal, infinite objects in function spaces, by means of their finite approximations. One works with atomic units of information, called *tokens*, and a notion of *consistency* for finite sets of tokens. Finally there is an *entailment* relation, between consistent finite sets of tokens and single tokens. The *ideals* (or *objects*) of an information system are defined to be the consistent and deductively closed sets of tokens; we write $|\boldsymbol{A}|$ for the set of ideals of $\boldsymbol{A}$. One shows easily that $|\boldsymbol{A}|$ is a domain w.r.t. the inclusion relation. Conversely, every domain with countable basis can be represented as the set of all ideals of an appropriate information system (Larsen and Winskel, 1991).

Here we take Scott's notion of an information system as a basis to introduce the partial continuous functionals. Call an information system *atomic* if the entailment relation $U \vdash b$ is given by $\exists_{a\in U}\{a\} \vdash b$ and hence determined by a transitive relation on $A$ (namely $\{a\} \vdash b$, written $a \geq b$). Call it *coherent* (Plotkin, 1978, p.210) when a finite set $U$ of tokens is consistent if and only if every two-element subset of it is. We will show below that if $\boldsymbol{B}$ is atomic (coherent), then so is the "function space" $\boldsymbol{A} \to \boldsymbol{B}$. Since our algebras will be given by atomic coherent information systems, this is the only kind of information systems we will have to deal with.

**2.1.1. Types.** A free algebra is given by its *constructors*, for instance zero and successor for the natural numbers. We want to treat other data types as well, like lists and binary trees. When dealing with inductively defined sets, it will also be useful to explicitly refer to the generation tree. Such trees are quite often infinitely branching, and hence we allow infinitary free algebras.

The freeness of the constructors is expressed by requiring that their ranges are disjoint and that they are injective. To allow for partiality – which is mandatory when we want to deal with computable objects –, we have to embed our algebras into domains. Both requirements together imply that we need "lazy domains".

Our type system is defined by two type forming operations: arrow types $\rho \Rightarrow \sigma$ and the formation of inductively generated types $\mu\vec{\alpha}\,\vec{\kappa}$, where $\vec{\alpha} = (\alpha_j)_{j=1,\dots,N}$ is a list of distinct "type variables", and $\vec{\kappa} = (\kappa_i)_{i=1,\dots,k}$ is a list

of "constructor types", whose argument types contain $\alpha_1, \ldots, \alpha_N$ in strictly positive positions only.

For instance, $\mu\alpha(\alpha, \alpha \Rightarrow \alpha)$ is the type of natural numbers; here the list $(\alpha, \alpha \Rightarrow \alpha)$ stands for two generation principles: $\alpha$ for "there is a natural number" (the 0), and $\alpha \Rightarrow \alpha$ for "for every natural number there is a next one" (its successor).

DEFINITION. Let $\vec{\alpha} = (\alpha_j)_{j=1,\ldots,N}$ be a list of distinct type variables. *Types* $\rho, \sigma, \tau, \mu \in \mathrm{Ty}$ and *constructor types* $\kappa \in \mathrm{KT}(\vec{\alpha})$ are defined inductively by

$$\frac{\vec{\rho}, \vec{\sigma}_1, \ldots, \vec{\sigma}_n \in \mathrm{Ty}}{\vec{\rho} \Rightarrow (\vec{\sigma}_1 \Rightarrow \alpha_{j_1}) \Rightarrow \ldots \Rightarrow (\vec{\sigma}_n \Rightarrow \alpha_{j_n}) \Rightarrow \alpha_j \in \mathrm{KT}(\vec{\alpha})} \quad (n \geq 0)$$

$$\frac{\kappa_1, \ldots, \kappa_n \in \mathrm{KT}(\vec{\alpha})}{(\mu\vec{\alpha}\,(\kappa_1, \ldots, \kappa_n))_j \in \mathrm{Ty}} \quad (n \geq 1) \qquad \frac{\rho, \sigma \in \mathrm{Ty}}{\rho \Rightarrow \sigma \in \mathrm{Ty}}$$

Here $\vec{\rho} \Rightarrow \sigma$ means $\rho_1 \Rightarrow \ldots \Rightarrow \rho_m \Rightarrow \sigma$, associated to the right. We reserve $\mu$ for types of the form $(\mu\vec{\alpha}\,(\kappa_1, \ldots, \kappa_k))_j$. The *parameter types* of $\mu$ are the members of all $\vec{\rho}$ appearing in its constructor types $\kappa_1, \ldots, \kappa_k$.

EXAMPLES.

| | | |
|---|---|---|
| unit | $:= \mu\alpha\,\alpha$, | unit |
| boole | $:= \mu\alpha\,(\alpha, \alpha)$, | booleans |
| nat | $:= \mu\alpha\,(\alpha, \alpha \Rightarrow \alpha)$, | natural numbers |
| list($\rho$) | $:= \mu\alpha\,(\alpha, \rho \Rightarrow \alpha \Rightarrow \alpha)$, | lists |
| $\rho \otimes \sigma$ | $:= \mu\alpha\,(\rho \Rightarrow \sigma \Rightarrow \alpha)$, | (tensor) product |
| $\rho + \sigma$ | $:= \mu\alpha\,(\rho \Rightarrow \alpha, \sigma \Rightarrow \alpha)$, | sum |
| (tree, tlist) | $:= \mu(\alpha, \beta)\,(\mathrm{nat} \Rightarrow \alpha, \beta \Rightarrow \alpha, \beta, \alpha \Rightarrow \beta \Rightarrow \beta)$, | |
| bin | $:= \mu\alpha\,(\alpha, \alpha \Rightarrow \alpha \Rightarrow \alpha)$, | binary trees |
| $\mathcal{O}$ | $:= \mu\alpha\,(\alpha, \alpha \Rightarrow \alpha, (\mathrm{nat} \Rightarrow \alpha) \Rightarrow \alpha)$, | ordinals |
| $\mathcal{T}_0$ | $:= \mathrm{nat}$, | |
| $\mathcal{T}_{n+1}$ | $:= \mu\alpha\,(\alpha, (\mathcal{T}_n \Rightarrow \alpha) \Rightarrow \alpha)$. | trees |

Notice that there are many equivalent ways to define these types. For instance, we could take unit + unit to be the type of booleans, and list(unit) to be the type of natural numbers.

A type is called *finitary* if it is a $\mu$-type with all its parameter types $\vec{\rho}$ finitary, and in all its constructor types

$$\vec{\rho} \Rightarrow (\vec{\sigma}_1 \Rightarrow \alpha_{j_1}) \Rightarrow \ldots \Rightarrow (\vec{\sigma}_n \Rightarrow \alpha_{j_n}) \Rightarrow \alpha_j \tag{2.1}$$

the $\vec{\sigma}_1, \dots, \vec{\sigma}_n$ are all empty. In the examples above unit, boole, nat, tree, tlist and bin are all finitary, whereas $\mathcal{O}$ and $\mathcal{T}_{n+1}$ are not. list($\rho$), $\rho \otimes \sigma$ and $\rho + \sigma$ are finitary provided their parameter types are. An argument position in a type is called *finitary* if it is occupied by a finitary type.

### 2.1.2. Atomic coherent information systems.

DEFINITION. An *atomic coherent information system* (abbreviated *acis*) is a triple $(A, \mathrm{Con}, \geq)$ with $A$ a countable set (the *tokens*, denoted $a, b, \dots$), Con a nonempty set of finite subsets of $A$ (the *consistent* sets or *formal neighborhoods*, denoted $U, V, \dots$), and $\geq$ a transitive and reflexive relation on $A$ (the *entailment relation*) which satisfy

(a) $\emptyset \in \mathrm{Con}$, and $\{a\} \in \mathrm{Con}$ for every $a \in A$,
(b) $U \in \mathrm{Con}$ if and only if every two-element subset of $U$ is in Con, and
(c) if $\{a, b\} \in \mathrm{Con}$ and $b \geq c$, then $\{a, c\} \in \mathrm{Con}$.

We write $U \geq a$ for $\exists_{b \in U} b \geq a$, and $U \geq V$ for $\forall_{a \in V} U \geq a$. – Every acis is an information system in the sense of Scott (1982); this follows from

LEMMA. *Let* $\boldsymbol{A} = (A, \mathrm{Con}, \geq)$ *be an acis.* $U \geq V_1, V_2$ *implies* $V_1 \cup V_2 \in \mathrm{Con}$.

PROOF. Let $b_1 \in V_1$, $b_2 \in V_2$. Then we have $a_1, a_2 \in U$ such that $a_i \geq b_i$. From $\{a_1, a_2\} \in \mathrm{Con}$ we obtain $\{a_1, b_2\} \in \mathrm{Con}$ by (c), hence $\{b_1, b_2\} \in \mathrm{Con}$ again by (c). □

DEFINITION. Let $\boldsymbol{A} = (A, \mathrm{Con}_A, \geq_A)$ and $\boldsymbol{B} = (B, \mathrm{Con}_B, \geq_B)$ be acis's. Define $\boldsymbol{A} \to \boldsymbol{B} = (C, \mathrm{Con}, \geq)$ by

$$C := \mathrm{Con}_A \times B,$$

$$\{(U_1, b_1), \dots (U_n, b_n)\} \in \mathrm{Con} :\leftrightarrow \forall_{i,j}\big(U_i \cup U_j \in \mathrm{Con}_A \to \{b_i, b_j\} \in \mathrm{Con}_B\big),$$

$$(U, b) \geq (V, c) :\leftrightarrow V \geq_A U \wedge b \geq_B c.$$

LEMMA. *Let* $\boldsymbol{A} = (A, \mathrm{Con}_A, \geq_A)$ *and* $\boldsymbol{B} = (B, \mathrm{Con}_B, \geq_B)$ *be acis's. Then* $\boldsymbol{A} \to \boldsymbol{B}$ *is an acis again.*

PROOF. Clearly $\geq$ is transitive and reflexive, and the conditions (a) and (b) of an acis hold; it remains to check (c). So let $\{(U_1, b_1), (U_2, b_2)\} \in \mathrm{Con}$ and $(U_2, b_2) \geq (V, c)$, hence $V \geq U_2$ and $b_2 \geq c$. We must show $\{(U_1, b_1), (V, c)\} \in \mathrm{Con}$. So assume $U_1 \cup V \in \mathrm{Con}$; we must show $\{b_1, c\} \in \mathrm{Con}$. Now $U_1 \cup V \in \mathrm{Con}$ and $V \geq U_2$ by the previous lemma imply $U_1 \cup U_2 \in \mathrm{Con}$. But then $\{b_1, b_2\} \in \mathrm{Con}$, hence $\{b_1, c\} \in \mathrm{Con}$ by (c). □

Scott (1982) introduced the notion of an *approximable map* from $\boldsymbol{A}$ to $\boldsymbol{B}$. Such a map is given by a relation $r$ between $\mathrm{Con}_A$ and $B$, where $r(U, b)$ intuitively means that whenever we are given the information $U \in \mathrm{Con}_A$ on the argument, then we know that at least the token $b$ appears in the value.

DEFINITION (Approximable map). Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be acis's. A relation $r \subseteq \mathrm{Con}_A \times B$ is an *approximable map* from $\boldsymbol{A}$ to $\boldsymbol{B}$ (written $r\colon \boldsymbol{A} \to \boldsymbol{B}$) if and only if

(a) if $r(U, b_1)$ and $r(U, b_2)$, then $\{b_1, b_2\} \in \mathrm{Con}_B$, and
(b) if $r(U, b)$, $V \geq_A U$ and $b \geq_B c$, then $r(V, c)$.

Call a (possibly infinite) set $x$ of tokens *consistent* if $U \in \mathrm{Con}$ for every finite subset $U \subseteq x$, and *deductively closed* if $\forall_{a \in x} \forall_{b \leq a} b \in x$. The *ideals* (or *objects*) of an information system are defined to be the consistent and deductively closed sets of tokens; we write $|\boldsymbol{A}|$ for the set of ideals of $\boldsymbol{A}$.

THEOREM. *Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be acis's. The ideals of $\boldsymbol{A} \to \boldsymbol{B}$ are exactly the approximable maps from $\boldsymbol{A}$ to $\boldsymbol{B}$.*

PROOF. We show that $r \in |\boldsymbol{A} \to \boldsymbol{B}|$ satisfies the axioms for approximable maps. (a). Let $r(U, b_1)$ and $r(U, b_2)$. Then $\{b_1, b_2\} \in \mathrm{Con}_B$ by the consistency of $r$. (b). Let $r(U, b)$, $V \geq_A U$ and $b \geq_B c$. Then $(U, b) \geq (V, c)$ by definition, hence $r(V, c)$ by the deductive closure of $r$.

For the other direction suppose $r\colon \boldsymbol{A} \to \boldsymbol{B}$ is an approximable map. We must show that $r \in |\boldsymbol{A} \to \boldsymbol{B}|$. Consistency of $r$: Suppose $r(U_1, b_1)$, $r(U_2, b_2)$ and $U = U_1 \cup U_2 \in \mathrm{Con}_A$. We must show that $\{b_1, b_2\} \in \mathrm{Con}_B$. Now by definition of approximable maps, from $r(U_i, b_i)$ and $U \geq_A U_i$ we obtain $r(U, b_i)$, and hence $\{b_1, b_2\} \in \mathrm{Con}_B$. Deductive closure of $r$: Suppose $r(U, b)$ and $(U, b) \geq (V, c)$, i.e., $V \geq_A U \wedge b \geq_B c$. Then $r(V, c)$ by definition of approximable maps. □

The set $|\boldsymbol{A}|$ of ideals for $\boldsymbol{A}$ carries a natural topology (the Scott topology), which has the cones $\tilde{U} := \{\, z \mid z \supseteq U \,\}$ generated by the formal neighborhoods $U$ as basis. The continuous maps $f\colon |\boldsymbol{A}| \to |\boldsymbol{B}|$ and the ideals $r \in |\boldsymbol{A} \to \boldsymbol{B}|$ are in a bijective correspondence. With any $r \in |\boldsymbol{A} \to \boldsymbol{B}|$ we can associate a continuous $|r|\colon |\boldsymbol{A}| \to |\boldsymbol{B}|$:

$$|r|(z) := \{\, b \in B \mid r(U, b) \text{ for some } U \subseteq z \,\},$$

and with any continuous $f\colon |\boldsymbol{A}| \to |\boldsymbol{B}|$ we can associate $\hat{f} \in |\boldsymbol{A} \to \boldsymbol{B}|$:

$$\hat{f}(U, b) \quad :\Longleftrightarrow \quad b \in f(\overline{U}).$$

These assignments are inverse to each other, i.e., $f = |\hat{f}|$ and $r = \widehat{|r|}$. – We will usually write $r(z)$ for $|r|(z)$, and similarly $f(U, b)$ for $\hat{f}(U, b)$. It will be clear from the context where the mods and hats should be inserted.

**2.1.3. Algebras with approximations.** We can now define the acis $\boldsymbol{C}_{\mu_j}$ of an algebra $\mu_j$, given by constructors $\mathrm{C}_i$.

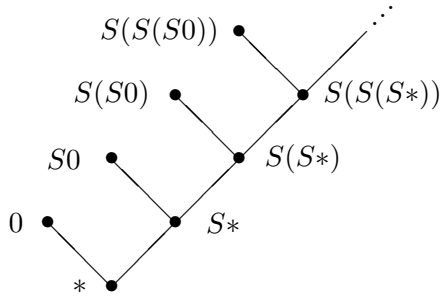



FIGURE 1. Tokens and entailment for nat

- The *tokens* are all type correct constructor expressions with an outermost $C_i$, such that at any finitary argument position we have either a special symbol – written $*$ –, which carries no information or else a token, and at any other argument position we have a formal neighborhood of the appropriate type. – By an *extended token* $a^*$ we mean a token or $*$, and $a^* \geq b^*$ means that $b^*$ is $*$, or both are tokens and the entailment relation holds.
- Two tokens are in the *entailment* relation $\geq$ if they start with the same constructor, and for every argument position the arguments located there are either extended tokens $a^*$, $b^*$ such that $a^* \geq b^*$, or formal neighborhoods $U$, $V$ such that $U \geq V$, as defined above (notice that this is an inductive definition).
- A finite set of tokens is *consistent* if every two-element subset is; two tokens are consistent if both start with the same constructor and have consistent extended tokens resp. formal neighborhoods at corresponding argument positions.

For example, the (extended) tokens for the algebra nat are as shown in Fig. 1 on page 65. A token $a$ entails another one $b$ if and only if there is a path from $a$ (up) to $b$ (down). In this case (and similarly for every finitary algebra) a finite set $U$ of tokens is consistent if and only if it has an upper bound. Every constructor C generates

$$r_{\mathrm{C}} := \{\, (\vec{U}, C\vec{b^*}) \mid \vec{U} \geq \vec{b^*} \,\},$$

with $b_i^*$ extended tokens or formal neighborhoods. The continuous map $|r_{\mathrm{C}}|$ is defined by

$$|r_{\mathrm{C}}|(\vec{z}) := \{\, b \mid (\vec{U}, b) \in r_{\mathrm{C}} \text{ for some } \vec{U} \subseteq \vec{z} \,\}.$$

Hence the (continuous maps corresponding to) constructors are injective and their ranges are disjoint, which is what we wanted to achieve.

The ideals $x$ for $\mu$ are – as for any information system – the consistent and deductively closed sets of tokens. Clearly all tokens in $x$ begin with the

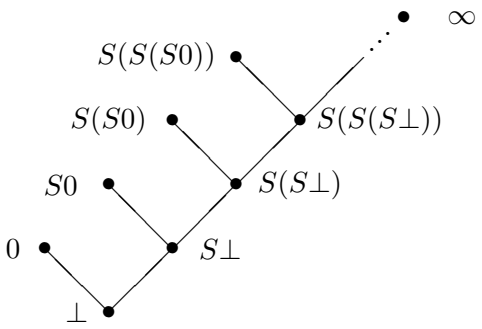



FIGURE 2. Ideals and inclusion for nat, i.e., its domain

same constructor. For instance, $\{S(S0), S(S*), S*\}$, $\{S(S*), S*\}$, $\{0\}$ are ideals for nat, but also the infinite set $\{\, S^n* \mid n > 0 \,\}$. The ideals for nat and their inclusion relation are pictured in Fig. 2 on page 66. Here we have denoted the ideals $\emptyset$, $\{0\}$, $\{\, S^n* \mid n > 0 \,\}$ by $\bot$, 0, $\infty$, respectively, and any other ideal by applications of (the continuous map corresponding to) the constructor $S$ to 0 or $\bot$. The ambiguous notation – $S$ denotes a symbol in constructor expressions and also the continuous map $|r_S|$ – should not lead to confusion.

Let $\boldsymbol{C}_{\rho\Rightarrow\sigma} := \boldsymbol{C}_\rho \to \boldsymbol{C}_\sigma$. The ideals $x \in |\boldsymbol{C}_\rho|$ are called *partial continuous functionals* of type $\rho$.

## 2.2. Structural Recursion

The inductive structure of the types $\vec{\mu} = \mu\vec{\alpha}\,\vec{\kappa}$ corresponds to two sorts of constants: with the *constructors* $\mathrm{C}_i^{\vec{\mu}} \colon \kappa_i[\vec{\mu}]$ we can construct elements of a type $\mu_j$, and with the *recursion operators* $\mathcal{R}_{\mu_j}^{\vec{\mu},\vec{\tau}}$ we can construct mappings from $\mu_j$ to $\tau_j$ by recursion on the structure of $\vec{\mu}$. In the present section we take a syntactical point of view: the recursion operators are introduced as constants, together with their conversion rules. It is proved (by a predicative method) that every reduction sequence terminates.

**2.2.1. Recursion operators.** In order to define the type of the recursion operators w.r.t. $\vec{\mu} = \mu\vec{\alpha}\,\vec{\kappa}$ and result types $\vec{\tau}$, we first define for

$$\kappa_i = \vec{\rho} \Rightarrow (\vec{\sigma}_1 \Rightarrow \alpha_{j_1}) \Rightarrow \ldots \Rightarrow (\vec{\sigma}_n \Rightarrow \alpha_{j_n}) \Rightarrow \alpha_j \in \mathrm{KT}(\vec{\alpha})$$

the *step type*

$$\begin{aligned}\delta_i^{\vec{\mu},\vec{\tau}} := \vec{\rho} \Rightarrow (\vec{\sigma}_1 \Rightarrow \mu_{j_1}) \Rightarrow \ldots \Rightarrow (\vec{\sigma}_n \Rightarrow \mu_{j_n}) \Rightarrow \\ (\vec{\sigma}_1 \Rightarrow \tau_{j_1}) \Rightarrow \ldots \Rightarrow (\vec{\sigma}_n \Rightarrow \tau_{j_n}) \Rightarrow \tau_j.\end{aligned}$$

Here $\vec{\rho}, (\vec{\sigma}_1 \Rightarrow \mu_{j_1}), \ldots, (\vec{\sigma}_n \Rightarrow \mu_{j_n})$ correspond to the *components* of the object of type $\mu_j$ under consideration, and $(\vec{\sigma}_1 \Rightarrow \tau_{j_1}), \ldots, (\vec{\sigma}_n \Rightarrow \tau_{j_n})$ to the previously defined values. The recursion operator $\mathcal{R}^{\vec{\mu},\vec{\tau}}_{\mu_j}$ has type

$$\mathcal{R}^{\vec{\mu},\vec{\tau}}_{\mu_j} \colon \delta^{\vec{\mu},\vec{\tau}}_1 \Rightarrow \ldots \Rightarrow \delta^{\vec{\mu},\vec{\tau}}_k \Rightarrow \mu_j \Rightarrow \tau_j$$

(recall that $k$ is the total number of constructors for all types $\mu_1, \ldots, \mu_N$).

We will often write $\mathcal{R}^{\vec{\mu},\vec{\tau}}_j$ for $\mathcal{R}^{\vec{\mu},\vec{\tau}}_{\mu_j}$, and omit the upper indices $\vec{\mu}, \vec{\tau}$ when they are clear from the context. In case of a non-simultaneous free algebra, i.e., of type $\mu\alpha\,\kappa$, for $\mathcal{R}^{\mu,\tau}_{\mu}$ we write $\mathcal{R}^{\tau}_{\mu}$.

### 2.2.2. Examples.

$$\begin{aligned}
&\mathsf{tt}^{\mathrm{boole}} := \mathrm{C}^{\mathrm{boole}}_1, \quad \mathsf{ff}^{\mathrm{boole}} := \mathrm{C}^{\mathrm{boole}}_2,\\
&\mathcal{R}^{\tau}_{\mathrm{boole}} \colon \tau \Rightarrow \tau \Rightarrow \mathrm{boole} \Rightarrow \tau,\\
&0^{\mathrm{nat}} := \mathrm{C}^{\mathrm{nat}}_1, \quad \mathrm{S}^{\mathrm{nat}\Rightarrow\mathrm{nat}} := \mathrm{C}^{\mathrm{nat}}_2,\\
&\mathcal{R}^{\tau}_{\mathrm{nat}} \colon \tau \Rightarrow (\mathrm{nat} \Rightarrow \tau \Rightarrow \tau) \Rightarrow \mathrm{nat} \Rightarrow \tau,\\
&\mathrm{nil}^{\mathrm{list}(\alpha)} := \mathrm{C}^{\mathrm{list}(\alpha)}_1, \quad \mathrm{cons}^{\alpha\Rightarrow\mathrm{list}(\alpha)\Rightarrow\mathrm{list}(\alpha)} := \mathrm{C}^{\mathrm{list}(\alpha)}_2,\\
&\mathcal{R}^{\tau}_{\mathrm{list}(\alpha)} \colon \tau \Rightarrow (\alpha \Rightarrow \mathrm{list}(\alpha) \Rightarrow \tau \Rightarrow \tau) \Rightarrow \mathrm{list}(\alpha) \Rightarrow \tau,\\
&\big(\mathrm{inl}_{\rho\sigma}\big)^{\rho\Rightarrow\rho+\sigma} := \mathrm{C}^{\rho+\sigma}_1,\\
&\big(\mathrm{inr}_{\rho\sigma}\big)^{\sigma\Rightarrow\rho+\sigma} := \mathrm{C}^{\rho+\sigma}_2,\\
&\mathcal{R}^{\tau}_{\rho+\sigma} \colon (\rho \Rightarrow \tau) \Rightarrow (\sigma \Rightarrow \tau) \Rightarrow \rho + \sigma \Rightarrow \tau,\\
&\big(\otimes^{+}_{\rho\sigma}\big)^{\rho\Rightarrow\sigma\Rightarrow\rho\otimes\sigma} := \mathrm{C}^{\rho\otimes\sigma}_1,\\
&\mathcal{R}^{\tau}_{\rho\otimes\sigma} \colon (\rho \Rightarrow \sigma \Rightarrow \tau) \Rightarrow \rho\otimes\sigma \Rightarrow \tau.
\end{aligned}$$

*Terms* are inductively defined from typed variables $x^{\rho}$ and the constants, that is, constructors $\mathrm{C}^{\vec{\mu}}_i$ and recursion operators $\mathcal{R}^{\vec{\mu},\vec{\tau}}_{\mu_j}$, by abstraction $\lambda x^{\rho} M^{\sigma}$ and application $M^{\rho\Rightarrow\sigma} N^{\rho}$. One can see easily that for instance the following functions can be "expressed" by means of terms involving recursion operators: *existence* $E_{\mathrm{nat}} \colon \mathrm{nat} \Rightarrow \mathrm{boole}$ and $E_{\mathrm{list}(\alpha)} \colon \mathrm{nat} \Rightarrow \mathrm{boole}$, and *equality* $= \colon \mathrm{nat} \Rightarrow \mathrm{nat} \Rightarrow \mathrm{boole}$.

$$\begin{aligned}
&E_{\mathrm{nat}}(0) := \mathsf{tt}, && (0 = 0) := \mathsf{tt},\\
&E_{\mathrm{nat}}(\mathrm{S}(n)) := E_{\mathrm{nat}}(n); && (0 = \mathrm{S}(n)) := \mathsf{ff},\\
&E_{\mathrm{list}(\alpha)}(\mathrm{nil}) := \mathsf{tt}, && (\mathrm{S}(m) = 0) := \mathsf{ff},\\
&E_{\mathrm{list}(\alpha)}(\mathrm{cons}(x, l)) := E_{\mathrm{list}(\alpha)}(l); && (\mathrm{S}(m) = \mathrm{S}(n)) := \ (n = m).
\end{aligned}$$

**2.2.3. Conversion.** To define the conversion relation, it will be helpful to use the following notation. Let $\vec{\mu} = \mu\vec{\alpha}\,\vec{\kappa}$ and

$$\kappa_i = \rho_1 \Rightarrow \ldots \Rightarrow \rho_m \Rightarrow (\vec{\sigma}_1 \Rightarrow \alpha_{j_1}) \Rightarrow \ldots \Rightarrow (\vec{\sigma}_n \Rightarrow \alpha_{j_n}) \Rightarrow \alpha_j \in \mathrm{KT}(\vec{\alpha}),$$

and consider $\mathrm{C}_i^{\vec{\mu}}\vec{N}$. Then we write $\vec{N}^P = N_1^P, \ldots, N_m^P$ for the *parameter arguments* $N_1^{\rho_1}, \ldots, N_m^{\rho_m}$ and $\vec{N}^R = N_1^R, \ldots, N_n^R$ for the *recursive arguments* $N_{m+1}^{\vec{\sigma}_1 \Rightarrow \mu_{j_1}}, \ldots, N_{m+n}^{\vec{\sigma}_n \Rightarrow \mu_{j_n}}$, and $n^R$ for the number $n$ of recursive arguments.

We define a *conversion relation* $\mapsto_\rho$ between terms of type $\rho$ by

$$(\lambda x M)N \mapsto M[x := N], \tag{2.2}$$

$$\lambda x.Mx \mapsto M \quad \text{if } x \notin \mathrm{FV}(M) \ (M \text{ not an abstraction}), \tag{2.3}$$

$$(\mathcal{R}_j\vec{M})^{\mu_j \Rightarrow \tau_j}(\mathrm{C}_i^{\vec{\mu}}\vec{N}) \mapsto M_i\vec{N}\big((\mathcal{R}_{j_1}\vec{M}) \circ N_1^R\big) \ldots \big((\mathcal{R}_{j_n}\vec{M}) \circ N_n^R\big). \tag{2.4}$$

Here we have written $\mathcal{R}_j$ for $\mathcal{R}_{\mu_j}^{\vec{\mu},\vec{\tau}}$.

The *one step reduction relation* $\to$ can now be defined as follows. $M \to N$ if $N$ is obtained from $M$ by replacing a subterm $M'$ in $M$ by $N'$, where $M' \mapsto N'$. The reduction relations $\to^+$ and $\to^*$ are the transitive and the reflexive transitive closure of $\to$, respectively. For $\vec{M} = M_1, \ldots, M_n$ we write $\vec{M} \to \vec{M}'$ if $M_i \to M_i'$ for some $i \in \{1, \ldots, n\}$ and $M_j = M_j'$ for all $i \neq j \in \{1, \ldots, n\}$. A term $M$ is *normal* (or in *normal form*) if there is no term $N$ such that $M \to N$.

Clearly normal closed terms are of the form $\mathrm{C}_i^{\vec{\mu}}\vec{N}$.

**2.2.4. Strong normalization.**

Definition. The set SN of *strongly normalizing* terms is inductively defined by

$$\forall_{N;M \to N} N \in \mathrm{SN} \to M \in \mathrm{SN}.$$

Note that with $M$ clearly every subterm of $M$ is strongly normalizing.

Definition. We define *strong computability predicates* $\mathrm{SC}^\rho$ by induction on $\rho$.

*Case* $\mu_j = (\mu\vec{\alpha}\,\vec{\kappa})_j$. Then $M \in \mathrm{SC}^{\mu_j}$ if

$$\forall_{N;M \to N} N \in \mathrm{SC}, \text{ and} \tag{2.5}$$

$$M = \mathrm{C}_i^{\vec{\mu}}\vec{N} \to \vec{N}^P \in \mathrm{SC} \wedge \bigmmand_{p=1}^{n^R} \forall_{\vec{K} \in \mathrm{SC}}\, N_p^R\vec{K} \in \mathrm{SC}^{\mu_{j_p}}. \tag{2.6}$$

*Case* $\rho \Rightarrow \sigma$.

$$\mathrm{SC}^{\rho \Rightarrow \sigma} := \{\, M \mid \forall_{N \in \mathrm{SC}^\rho} MN \in \mathrm{SC}^\sigma \,\}.$$

The reference to $\vec{N}^P \in \mathrm{SC}$ and $\vec{K} \in \mathrm{SC}$ in (2.6) is legal, because the types $\vec{\rho}, \vec{\sigma}_i$ of $\vec{N}, \vec{K}$ must have been generated *before* $\mu_j$. Note also that by (2.6) $\mathrm{C}_i^{\vec{\mu}} \vec{N} \in \mathrm{SC}$ implies $\vec{N} \in \mathrm{SC}$.

We now set up a sequence of lemmata leading to a proof that every term is strongly normalizing.

Lemma (Closure of SC under reduction). *If $M \in \mathrm{SC}^\rho$ and $M \to M'$, then $M' \in \mathrm{SC}^\rho$.*

Proof. Induction on $\rho$. *Case* $\mu$. By (2.5). *Case* $\rho \Rightarrow \sigma$. Assume $M \in \mathrm{SC}^{\rho \Rightarrow \sigma}$ and $M \to M'$; we must show $M' \in \mathrm{SC}$. So let $N \in \mathrm{SC}^\rho$; we must show $M'N \in \mathrm{SC}^\sigma$. But this follows from $MN \to M'N$ and $MN \in \mathrm{SC}^\rho$ by IH on $\sigma$. □

Lemma (Closure of SC under variable application).

$$\forall_{\vec{M} \in \mathrm{SN}}(\vec{M} \in \mathrm{SC} \to (x\vec{M})^\mu \in \mathrm{SC}).$$

Proof. Induction on $\vec{M} \in \mathrm{SN}$. Assume $\vec{M} \in \mathrm{SN}$ and $\vec{M} \in \mathrm{SC}$; we must show $(x\vec{M})^\mu \in \mathrm{SC}$. So assume $x\vec{M} \to N$; we must show $N \in \mathrm{SC}$. Now by the form of the conversion rules $N$ must be of the form $x\vec{M}'$ with $\vec{M} \to \vec{M}'$. But $\vec{M}' \in \mathrm{SC}$ by closure of SC under reduction, hence $x\vec{M}' \in \mathrm{SC}$ by IH for $\vec{M}'$. □

Lemma. (a) $\mathrm{SC}^\rho \subseteq \mathrm{SN}$,
(b) $x \in \mathrm{SC}^\rho$.

Proof. By simultaneous induction on $\rho$. *Case* $\mu_j = (\mu \vec{\alpha}\, \vec{\kappa})_j$. (a). We show that $M \in \mathrm{SC}^{\mu_j}$ implies $M \in \mathrm{SN}$ by (side) induction on $M \in \mathrm{SC}^{\mu_j}$. So assume $M \in \mathrm{SC}^{\mu_j}$; we must show $M \in \mathrm{SN}$. But for every $N$ with $M \to N$ we have $N \in \mathrm{SC}$ by (2.5), hence $N \in \mathrm{SN}$ by the SIH. (b). $x \in \mathrm{SC}^{\mu_j}$ holds trivially.

*Case* $\rho \Rightarrow \sigma$. (a). Assume $M \in \mathrm{SC}^{\rho \Rightarrow \sigma}$; we must show $M \in \mathrm{SN}$. By IH(b) for $\rho$ we have $x \in \mathrm{SC}^\rho$, hence $Mx \in \mathrm{SC}^\sigma$, hence $Mx \in \mathrm{SN}$ by IH(a) for $\sigma$. But $Mx \in \mathrm{SN}$ clearly implies $M \in \mathrm{SN}$. (b). Let $\vec{M} \in \mathrm{SC}^{\vec{\rho}}$ with $\rho_1 = \rho$; we must show $x\vec{M} \in \mathrm{SC}^\mu$. But this follows from the closure of SC under variable application, using IH(a) for $\vec{\rho}$. □

It follows that each constructor is strongly computable:

Corollary. $\vec{N} \in \mathrm{SC} \to \mathrm{C}_i^{\vec{\mu}} \vec{N} \in \mathrm{SC}$, *i.e.*, $\mathrm{C}_i^{\vec{\mu}} \in \mathrm{SC}$.

Proof. First show $\forall_{\vec{N} \in \mathrm{SN}}(\vec{N} \in \mathrm{SC} \to \mathrm{C}_i^{\vec{\mu}} \vec{N} \in \mathrm{SC})$ by induction on $\vec{N} \in \mathrm{SN}$ as we proved closure of SC under variable application, and then use $\mathrm{SC}^\rho \subseteq \mathrm{SN}$. □

LEMMA. $\forall_{M,N,\vec{N}\in\mathrm{SN}}(M[x:=N]\vec{N}\in \mathrm{SC}^{\mu}\to(\lambda xM)N\vec{N}\in\mathrm{SC}^{\mu})$.

PROOF. By induction on $M,N,\vec{N}\in\mathrm{SN}$. Let $M,N,\vec{N}\in\mathrm{SN}$ and assume $M[x:=N]\vec{N}\in\mathrm{SC}$; we must show $(\lambda xM)N\vec{N}\in\mathrm{SC}$. Assume $(\lambda xM)N\vec{N}\to K$; we must show $K\in\mathrm{SC}$. *Case* $K=(\lambda xM')N'\vec{N}'$ with $M,N,\vec{N}\to M',N',\vec{N}'$. Then $M[x:=N]\vec{N}\to^{*}M'[x:=N']\vec{N}'$, hence by (2.5) from our assumption $M[x:=N]\vec{N}\in\mathrm{SC}$ we can infer $M'[x:=N']\vec{N}'\in\mathrm{SC}$, therefore $(\lambda xM')N'\vec{N}'\in\mathrm{SC}$ by IH. *Case* $K=M[x:=N]\vec{N}$. Then $K\in\mathrm{SC}$ by assumption. □

By induction on $\rho$ (using $\mathrm{SC}^{\rho}\subseteq\mathrm{SN}$) it follows that this property holds for arbitrary types $\rho$ as well:

$$\forall_{M,N,\vec{N}\in\mathrm{SN}}(M[x:=N]\vec{N}\in\mathrm{SC}^{\rho}\to(\lambda xM)N\vec{N}\in\mathrm{SC}^{\rho}). \tag{2.7}$$

LEMMA. $\forall_{N\in\mathrm{SC}^{\mu_j}}\forall_{\vec{M},\vec{L}\in\mathrm{SN}}(\vec{M},\vec{L}\in\mathrm{SC}\to\mathcal{R}_j\vec{M}N\vec{L}\in\mathrm{SC}^{\mu})$.

PROOF. By main induction on $N\in\mathrm{SC}^{\mu_j}$, and side induction on $\vec{M},\vec{L}\in\mathrm{SN}$. Assume

$$\mathcal{R}_j\vec{M}N\vec{L}\to L.$$

We must show $L\in\mathrm{SC}$.

*Case* 1. $\mathcal{R}_j\vec{M}'N\vec{L}'\in\mathrm{SC}$ by the SIH.

*Case* 2. $\mathcal{R}_j\vec{M}N'\vec{L}\in\mathrm{SC}$ by the main IH.

*Case* 3. $N=\mathrm{C}_i^{\vec{\mu}}\vec{N}$ and

$$L=M_i\vec{N}\big((\mathcal{R}_j\vec{M})\circ N_1^R\big)\dots\big((\mathcal{R}_j\vec{M})\circ N_n^R\big)\vec{L}.$$

$\vec{M},\vec{L}\in\mathrm{SC}$ by assumption. $\vec{N}\in\mathrm{SC}$ follows from $N=\mathrm{C}_i^{\vec{\mu}}\vec{N}\in\mathrm{SC}$ by (2.6). Note that for all recursive arguments $N_p^R$ of $N$ and all strongly computable $\vec{K}$ by (2.6) we have the IH for $N_p^R\vec{K}$ available. It remains to show $(\mathcal{R}_j\vec{M})\circ N_p^R=\lambda\vec{x}_p.\mathcal{R}_j\vec{M}(N_p^R\vec{x}_p)\in\mathrm{SC}$. So let $\vec{K},\vec{Q}\in\mathrm{SC}$ be given. We must show $(\lambda\vec{x}_p.\mathcal{R}_j\vec{M}(N_p^R\vec{x}_p))\vec{K}\vec{Q}\in\mathrm{SC}$. By IH for $N_p^R\vec{K}$ we have $\mathcal{R}_j\vec{M}(N_p^R\vec{K})\vec{Q}\in\mathrm{SC}$, since $\vec{K},\vec{Q}\in\mathrm{SN}$ because of $\mathrm{SC}^{\rho}\subseteq\mathrm{SN}$. Now (2.7) yields the claim. □

So in particular $\mathcal{R}_j\in\mathrm{SC}$.

DEFINITION. A substitution $\xi$ is *strongly computable*, if $\xi(x)\in\mathrm{SC}$ for all variables $x$. A term $M$ is *strongly computable under substitution*, if $M\xi\in\mathrm{SC}$ for all strongly computable substitutions $\xi$.

THEOREM. *Every term is strongly computable under substitution.*

PROOF. Induction on the term $M$. *Case* $x$. $x\xi \in \mathrm{SC}$, since $\xi$ is strongly computable. The *cases* $\mathrm{C}_i^{\vec{\mu}}$ and $\mathcal{R}_j$ have been treated above. *Case* $MN$. By IH $M\xi, N\xi \in \mathrm{SC}$, hence $(MN)\xi = (M\xi)(N\xi) \in \mathrm{SC}$. *Case* $\lambda x M$. Let $\xi$ be a strongly computable substitution; we must show $(\lambda x M)\xi = \lambda x M\xi_x^x \in \mathrm{SC}$. So let $N \in \mathrm{SC}$; we must show $(\lambda x M\xi_x^x)N \in \mathrm{SC}$. By IH $M\xi_x^N \in \mathrm{SC}$, hence $(\lambda x M\xi_x^x)N \in \mathrm{SC}$ by (2.7). □

It follows that every term is strongly normalizing.

## 2.3. Terms; Denotational and Operational Semantics

For every type $\rho$, we have defined what a partial continuous functional of type $\rho$ is: an ideal consisting of tokens at this type. These tokens or rather the formal neighborhoods formed from them are syntactic in nature; they are reminiscent to Kreisel's "formal neighborhoods" (Kreisel, 1959; Martin-Löf, 1983; Coquand and Spiwack, 2005). However – in contrast to Martin-Löf (1983) – we do not have to deal separately with a notion of consistency for formal neighborhoods: this concept is built into information systems.

Let us now turn our attention to a formal (functional programming) language, in the style of Plotkin's PCF (1977), and see how we can provide a denotational semantics (that is, a "meaning") for the terms of this language. A closed term $M$ of type $\rho$ will denote a partial continuous functional of this type, that is, a consistent and deductively closed set of tokens of type $\rho$. We will define this set inductively.

It will turn out that these sets are recursively enumerable. In this sense every closed term $M$ of type $\rho$ denotes a computable partial continuous functional of type $\rho$. However, it is not a good idea to *define* a computable functional in this way, by providing a recursive enumeration of its tokens. We rather want to be able to use recursion equations for such definitions. Therefore we extend the term language by constants $D$ defined by certain "computation rules", as in (Berger et al., 2003; Berger, 2005). Our semantics will cover these as well.

There are some natural questions one can have for such a term language:

(1) Preservation of values under conversion (as in (Martin-Löf, 1983, First Theorem)). Here we need to include applications of computation rules.
(2) An adequacy theorem (cf. (Plotkin, 1977, Theorem 3.1) or (Martin-Löf, 1983, Second Theorem)), which in our setting says that whenever a closed term has a proper token in the ideal it denotes, then it evaluates to a constructor term entailing this token.

Propertie (1) will be proved in the present section, and (2) in Sec.2.4.

Coquand and Spiwack (2005) observed that the types play only a somewhat minor role in this setup. It suffices to know the *arity* (a natural number) of the constants (constructors and defined constants), to guide the definitions. An interesting consequence is that one can use this approach for dependently typed languages as well, for instance, the terms of Martin-Löf's type theory.

**2.3.1. Terms.** Terms are built from (typed) variables and (typed) constants (constructors C or defined constants $D$, see below) by (type-correct) application and abstraction:

$$M, N ::= x^\rho \mid \mathrm{C}^\rho \mid D^\rho \mid (\lambda x^\rho\, M^\sigma)^{\rho\Rightarrow\sigma} \mid (M^{\rho\Rightarrow\sigma} N^\rho)^\sigma.$$

Every defined constant comes with a *system of computation rules*, consisting of finitely many equations $D\vec{P}_i = M_i$ $(i = 1, \dots, n)$ with constructor patterns $\vec{P}_i$, such that $\vec{P}_i$ and $\vec{P}_j$ $(i \neq j)$ are non-unifiable. *Constructor patterns* are lists of applicative terms with distinct variables, defined inductively as follows (we write $\vec{P}(\vec{x}\,)$ to indicate all variables in $\vec{P}$; notice that $x$ can be a variable for a formal neighborhood, and that all expressions must be type-correct):

- $x(x)$ is a constructor pattern.
- If $C$ is a constructor and $\vec{P}(\vec{x}\,)$ a constructor pattern, then $(C\vec{P})(\vec{x}\,)$ is a constructor pattern.
- If $\vec{P}(\vec{x}\,)$ and $Q(\vec{y}\,)$ are constructor patterns whose variables $\vec{x}$ and $\vec{y}$ are disjoint, then $(\vec{P}, Q)(\vec{x}, \vec{y}\,)$ is a constructor pattern.

**2.3.2. Ideals as meaning of terms.** How can we use computation rules to define an ideal $z$ in a function space? The general idea is to inductively define the set of tokens $(U, b)$ that make up $z$. However, since arbitrary terms are allowed on the right, we need to define the value $[\![\lambda\vec{x}\, M]\!]$, where $M$ is a term with free variables among $\vec{x}$. Since this value is a token set, we can define inductively the relation $(\vec{U}, b) \in [\![\lambda\vec{x}\, M]\!]$.

We use the following notation. $(\vec{U}, b)$ means $(U_1, \dots (U_n, b) \dots)$, and at argument positions of constructors we use $b^*$ for extended tokens as well as for formal neighborhoods. $(\vec{U}, V) \subseteq [\![\lambda\vec{x}\, M]\!]$ means $(\vec{U}, b) \in [\![\lambda\vec{x}\, M]\!]$, for all (finitely many) $b \in V$. For a constructor $C$, let

$$C(V) := \begin{cases} \{C*\} & \text{if } V = \emptyset, \\ \{\, Ca \mid a \in V \,\} & \text{otherwise.} \end{cases}$$

Definition (Inductive, of $(\vec{U}, b) \in [\![\lambda\vec{x}\, M]\!]$).

$$\frac{U_i \geq b}{(\vec{U}, b) \in [\![\lambda\vec{x}\, x_i]\!]}(V), \qquad \frac{(\vec{U}, V) \subseteq [\![\lambda\vec{x}\, N]\!] \qquad (\vec{U}, V, c) \in [\![\lambda\vec{x}\, M]\!]}{(\vec{U}, c) \in [\![\lambda\vec{x}.MN]\!]}(A).$$

For every constructor $C$ and defined constant $D$ we have

$$\frac{\vec{V} \geq \vec{b^*}}{(\vec{U}, \vec{V}, C\vec{b^*}) \in [\![\lambda\vec{x}\, C]\!]}(C), \qquad \frac{(\vec{U}, \vec{V}, b) \in [\![\lambda\vec{x}, \vec{y}\, M]\!]}{(\vec{U}, \vec{P}(\vec{V}), b) \in [\![\lambda\vec{x}\; D]\!]}(D),$$

with one such rule $(D)$ for every computation rule $D\vec{P}(\vec{y}) = M$.

Here are some simple consequences of this definition. First we show a useful property of constructors:

LEMMA. *$(\vec{U}, b) \in [\![\lambda\vec{x}.C\vec{N}]\!]$ if and only if there are $\vec{c^*} \geq \vec{b^*}$ such that $b = C\vec{b^*}$ and $(\vec{U}, c_i) \in [\![\lambda\vec{x}\, N_i]\!]$ $(i = 1, \ldots, n)$ for the tokens $c_i$ among $\vec{c^*}$.*

PROOF. We may assume that $\vec{b^*}, \vec{c^*}$ are tokens $\vec{b}, \vec{c}$. Let $(\vec{U}, c_i) \in [\![\lambda\vec{x}\, N_i]\!]$ $(c_i \geq b_i,\ i = 1, \ldots, n)$. For $j = 0, \ldots, n$ we show $(\vec{U}, \{c_{j+1}\}, \ldots, \{c_n\}, C\vec{b}) \in [\![\lambda\vec{x}.CN_1 \ldots N_j]\!]$. In case $j = 0$ use $(C)$:

$$\frac{\{c_1\} \geq b_1 \quad \ldots \quad \{c_n\} \geq b_n}{(\vec{U}, \{c_1\}, \ldots, \{c_n\}, C\vec{b}) \in [\![\lambda\vec{x}\, C]\!]}.$$

In the step from $j - 1$ to $j$ use $(A)$:

$$\frac{(\vec{U}, c_j) \in [\![\lambda\vec{x}\, N_j]\!] \qquad (\vec{U}, \{c_j\}, \ldots, \{c_n\}, C\vec{b}) \in [\![\lambda\vec{x}.CN_1 \ldots N_{j-1}]\!]}{(\vec{U}, \{c_{j+1}\}, \ldots, \{c_n\}, C\vec{b}) \in [\![\lambda\vec{x}.CN_1 \ldots N_j]\!]}.$$

For $j = n$ the claim follows. – For the other direction, observe that only $(A)$ could have been applied. Hence the argument can be read backwards. □

Using the fact that the left hand sides of computation rules are non-unifiable we can prove:

LEMMA. *$[\![\lambda\vec{x}\, M]\!]$ is an ideal, i.e., consistent and deductively closed.*

PROOF. Induction on $(\vec{U}, b) \in [\![\lambda\vec{x}\, M]\!]$.

(1) Consistency. *Case* $(V)$. Assume $(\vec{U_1}, b_1), (\vec{U_2}, b_2) \in [\![\lambda\vec{x}\, x_i]\!]$, and that $\vec{U_1}$ and $\vec{U_2}$ are pairwise consistent. We must show $\{b_1, b_2\} \in \mathrm{Con}$. By (V), $U_{1i} \geq b_1$ and $U_{2i} \geq b_2$. Now $\{b_1, b_2\} \in \mathrm{Con}$ follows from $U_{1i} \cup U_{2i} \in \mathrm{Con}$.

*Case* $(A)$. Let $(\vec{U_1}, c_1), (\vec{U_2}, c_2) \in [\![\lambda\vec{x}.MN]\!]$, with $\vec{U_1}$ and $\vec{U_2}$ pairwise consistent. We show $\{c_1, c_2\} \in \mathrm{Con}$. By $(A)$, $(\vec{U_1}, V_1), (\vec{U_2}, V_2) \subseteq [\![\lambda\vec{x}\, N]\!]$, so by IH $V_1 \cup V_2 \in \mathrm{Con}$. Similarly, again by $(A)$, $(\vec{U_1}, V_1, c_1), (\vec{U_2}, V_2, c_2) \in [\![\lambda\vec{x}\, M]\!]$, hence $\{c_1, c_2\} \in \mathrm{Con}$ by IH.

*Case* $(C)$. Assume $(\vec{U_1}, \vec{V_1}, C\vec{b^*}_1), (\vec{U_2}, \vec{V_2}, C\vec{b^*}_2) \in [\![\lambda\vec{x}\, C]\!]$, and that $\vec{U_1}, \vec{V_1}$ and $\vec{U_2}, \vec{V_2}$ are pairwise consistent. We show $\{C\vec{b^*}_1, C\vec{b^*}_2\} \in \mathrm{Con}$. By (C), $\vec{V_i} \geq \vec{b^*}_i$ $(i = 1, 2)$. From the pairwise consistency of $\vec{V_1}$ and $\vec{V_2}$ we obtain the pairwise consistency of $\vec{b^*}_1$ and $\vec{b^*}_2$. Hence $\{C\vec{b^*}_1, C\vec{b^*}_2\} \in \mathrm{Con}$.

*Case* $(D)$. Let $(\vec{U}_i, \vec{P}_i(\vec{V}_i), b_i) \in [\![\lambda\vec{x}\, D]\!]$ $(i = 1, 2)$, and assume that $\vec{U}_1, \vec{P}_1(\vec{V}_1)$ and $\vec{U}_2, \vec{P}_2(\vec{V}_2)$ are pairwise consistent. From the fact that the left hand sides of computation rules are non-unifiable we can infer $\vec{P}_1 = \vec{P}_2$, and that $\vec{V}_1$ and $\vec{V}_2$ are pairwise consistent. Then $\{b_1, b_2\} \in \mathrm{Con}$ by IH.

(2) Closure under $\geq$. *Case* $(V)$. Assume $\vec{V} \geq \vec{U}$ and $b \geq c$. We must show $(\vec{V}, c) \in [\![\lambda\vec{x}\, x_i]\!]$. By (V) it suffices to show $V_i \geq c$. But this follows from $V_i \geq U_i \geq b \geq c$.

*Case* $(A)$. The IH clearly suffices here.

*Case* $(C)$. Assume $\vec{U}_1 \geq \vec{U}$, $\vec{V}_1 \geq \vec{V}$ and $C\vec{b^*} \geq C\vec{c^*}$. We must show $(\vec{U}_1, \vec{V}_1, C\vec{c^*}) \in [\![\lambda\vec{x}\, C]\!]$. By (C) it suffices to show $\vec{V}_1 \geq \vec{c^*}$. But this follows from $\vec{V}_1 \geq \vec{V} \geq \vec{b^*} \geq \vec{c^*}$.

*Case* $(D)$. Assume $\vec{U}_1 \geq \vec{U}$, $\vec{Z} \geq \vec{P}(\vec{V})$ and $b \geq b_1$. Notice that $\vec{Z} \geq \vec{P}(\vec{V})$ implies $\vec{Z} = \vec{P}(\vec{V}_1)$ with $\vec{V}_1 \geq \vec{V}$, so we must show $(\vec{U}_1, \vec{P}(\vec{V}_1), b_1) \in [\![\lambda\vec{x}\, D]\!]$. By IH we have $(\vec{U}_1, \vec{V}_1, b_1) \in [\![\lambda\vec{x}, \vec{y}\, M]\!]$. Now use (D). □

**2.3.3. Preservation of values.** We now prove that our definition above of the meaning of a term is reasonable in the sense that an application of the standard ($\beta$- and $\eta$-) conversions and also of a computation rule does not change the meaning of a term. For the $\beta$-conversion part of this proof it is helpful to first introduce a more standard notation, which involves variable environments.

Definition. Assume that all free variables in $M$ are among $\vec{x}$. Let $[\![M]\!]^{\vec{U}}_{\vec{x}} := \{\, b \mid (\vec{U}, b) \in [\![\lambda\vec{x}\, M]\!] \,\}$ and $[\![M]\!]^{\vec{u}}_{\vec{x}} := \bigcup_{\vec{U} \subseteq \vec{u}} [\![M]\!]^{\vec{U}}_{\vec{x}}$.

We have a useful monotonicity property, which follows from the deductive closure of $[\![\lambda\vec{x}\, M]\!]$.

Lemma. (a) *If* $\vec{V} \geq \vec{U}$, $b \geq c$ *and* $b \in [\![M]\!]^{\vec{U}}_{\vec{x}}$, *then* $c \in [\![M]\!]^{\vec{V}}_{\vec{x}}$.
(b) *If* $\vec{v} \supseteq \vec{u}$, $b \geq c$ *and* $b \in [\![M]\!]^{\vec{u}}_{\vec{x}}$, *then* $c \in [\![M]\!]^{\vec{v}}_{\vec{x}}$.

Proof. (a). By the deductive closure of $[\![\lambda\vec{x}\, M]\!]$, $\vec{V} \geq \vec{U}$, $b \geq c$ and $(\vec{U}, b) \in [\![\lambda\vec{x}\, M]\!]$ together imply $(\vec{V}, c) \in [\![\lambda\vec{x}\, M]\!]$. (b) follows from (a). □

Lemma. (a) $[\![x_i]\!]^{\vec{u}}_{\vec{x}} = u_i$.
(b) $[\![\lambda y\, M]\!]^{\vec{u}}_{\vec{x}} = \{\, (V, b) \mid b \in [\![M]\!]^{\vec{u}, V}_{\vec{x}, y} \,\}$.
(c) $[\![MN]\!]^{\vec{u}}_{\vec{x}} = [\![M]\!]^{\vec{u}}_{\vec{x}} [\![N]\!]^{\vec{u}}_{\vec{x}}$.

Proof. (b). It suffices to prove this with $\vec{U}$ for $\vec{u}$. But $(V, b) \in [\![\lambda y\, M]\!]^{\vec{U}}_{\vec{x}}$ and $b \in [\![M]\!]^{\vec{U}, V}_{\vec{x}, y}$ are both equivalent to $(\vec{U}, V, b) \in [\![\lambda\vec{x}, y\, M]\!]$.

(c).

$$c \in [\![M]\!]^{\vec{u}}_{\vec{x}} [\![N]\!]^{\vec{u}}_{\vec{x}}$$

$$\begin{aligned}
&\leftrightarrow \exists_{V\subseteq[\![N]\!]^{\vec{u}}_{\vec{x}}}(V,c)\in[\![M]\!]^{\vec{u}}_{\vec{x}} \quad \text{(application in acis's)}\\
&\leftrightarrow \exists_{V\subseteq[\![N]\!]^{\vec{u}}_{\vec{x}}}\exists_{\vec{U}\subseteq\vec{u}}(V,c)\in[\![M]\!]^{\vec{U}}_{\vec{x}}\\
&\leftrightarrow \exists_{\vec{U}_1\subseteq\vec{u}}\exists_{V\subseteq[\![N]\!]^{\vec{U}_1}_{\vec{x}}}\exists_{\vec{U}\subseteq\vec{u}}(V,c)\in[\![M]\!]^{\vec{U}}_{\vec{x}}\\
&\leftrightarrow^{(*)} \exists_{\vec{U}\subseteq\vec{u}}\exists_{V\subseteq[\![N]\!]^{\vec{U}}_{\vec{x}}}(V,c)\in[\![M]\!]^{\vec{U}}_{\vec{x}}\\
&\leftrightarrow \exists_{\vec{U}\subseteq\vec{u}}\exists_V.(\vec{U},V)\subseteq[\![\lambda\vec{x}\,N]\!]\wedge(\vec{U},V,c)\in[\![\lambda\vec{x}\,M]\!]\\
&\leftrightarrow \exists_{\vec{U}\subseteq\vec{u}}(\vec{U},c)\in[\![\lambda\vec{x}.MN]\!] \quad \text{(by } (A))\\
&\leftrightarrow \exists_{\vec{U}\subseteq\vec{u}}\,c\in[\![MN]\!]^{\vec{U}}_{\vec{x}}\\
&\leftrightarrow c\in[\![MN]\!]^{\vec{u}}_{\vec{x}}.
\end{aligned}$$

Here is the proof of the equivalence marked $(*)$. The upwards direction is obvious. For the downwards direction we use monotonicity. Assume $\vec{U}_1\subseteq\vec{u}$, $V\subseteq[\![N]\!]^{\vec{U}_1}_{\vec{x}}$, $\vec{U}\subseteq\vec{u}$ and $(V,c)\in[\![M]\!]^{\vec{U}}_{\vec{x}}$. Let $\vec{U}_2:=\vec{U}_1\cup\vec{U}\subseteq\vec{u}$. Then by monotonicity $V\subseteq[\![N]\!]^{\vec{U}_2}_{\vec{x}}$ and $(V,c)\in[\![M]\!]^{\vec{U}_2}_{\vec{x}}$. □

Corollary. $[\![\lambda y\,M]\!]^{\vec{u}}_{\vec{x}}v=[\![M]\!]^{\vec{u},v}_{\vec{x},y}$.

Proof.

$$\begin{aligned}
b\in[\![\lambda y\,M]\!]^{\vec{u}}_{\vec{x}}v &\leftrightarrow \exists_{V\subseteq v}(V,b)\in[\![\lambda y\,M]\!]^{\vec{u}}_{\vec{x}} \quad \text{(application in acis's)}\\
&\leftrightarrow \exists_{V\subseteq v}\,b\in[\![M]\!]^{\vec{u},V}_{\vec{x},y} \quad \text{(by the lemma)}\\
&\leftrightarrow b\in[\![M]\!]^{\vec{u},v}_{\vec{x},y}.
\end{aligned}$$

□

Lemma (Substitution). $[\![M]\!]^{\vec{u},[\![N]\!]^{\vec{u}}_{\vec{x}}}_{\vec{x},z}=[\![M[z:=N]]\!]^{\vec{u}}_{\vec{x}}$.

Proof. *Case* $\lambda y\,M$. For readability we leave out $\vec{x}$ and $\vec{u}$.

$$\begin{aligned}
[\![\lambda y\,M]\!]^{[\![N]\!]}_{z} &= \{\,(V,b)\mid b\in[\![M]\!]^{[\![N]\!],V}_{z,y}\,\}\\
&= \{\,(V,b)\mid b\in[\![M[z:=N]]\!]^{V}_{y}\,\} \quad \text{(by IH)}\\
&= [\![\lambda y.M[z:=N]]\!] \quad \text{(by the lemma)}\\
&= [\![(\lambda y\,M)[z:=N]]\!].
\end{aligned}$$

The other cases are easy. □

Lemma. $[\![(\lambda y\,M)N]\!]^{\vec{u}}_{\vec{x}}=[\![M[y:=N]]\!]^{\vec{u}}_{\vec{x}}$.

Proof. For readability we leave out $\vec{x}$ and $\vec{u}$. By the last two lemmata and the corollary, $[\![(\lambda y\,M)N]\!]=[\![\lambda y\,M]\!][\![N]\!]=[\![M]\!]^{[\![N]\!]}_{y}=[\![M[y:=N]]\!]$. □

Lemma. $[\![\lambda y.My]\!]^{\vec{u}}_{\vec{x}}=[\![M]\!]^{\vec{u}}_{\vec{x}}$, *if* $y\notin\mathrm{FV}(M)$.

PROOF. For readability we leave out $\vec{x}$ and $\vec{u}$.

$$\begin{aligned}(V,b) \in [\![\lambda y.My]\!] &\leftrightarrow b \in [\![My]\!]^V_y \\ &\leftrightarrow b \in [\![M]\!]\overline{V} \\ &\leftrightarrow \exists_{U \subseteq \overline{V}}\,(U,b) \in [\![M]\!] \quad \text{(application in acis's)} \\ &\leftrightarrow (V,b) \in [\![M]\!],\end{aligned}$$

where in the last step we have used monotonicity. □

To prove preservation of values under computation rules, the following observation will be needed (it removes the need for "(generalized) predecessor functions" of Berger et al. (2003); Berger (2005)):

LEMMA.

$$(\vec{U},\vec{V},b) \in [\![\lambda\vec{x},\vec{y}.M[z := \mathrm{C}\vec{y}]]\!] \leftrightarrow (\vec{U},\mathrm{C}\vec{V},b) \in [\![\lambda\vec{x},z\,M]\!]. \tag{2.8}$$

PROOF. Induction on $(\vec{U},\vec{V},b) \in [\![\lambda\vec{x},\vec{y}.M[z := \mathrm{C}\vec{y}]]\!]$, and cases on the form of $M$. *Case* $MN$.

$$\begin{aligned}&(\vec{U},\vec{V},c) \in [\![\lambda\vec{x},\vec{y}.M[z := \mathrm{C}\vec{y}]N[z := \mathrm{C}\vec{y}]]\!] \\ &\leftrightarrow \exists_Z.(\vec{U},\vec{V},Z) \subseteq [\![\lambda\vec{x},\vec{y}\,N[z := \mathrm{C}\vec{y}]]\!] \wedge (\vec{U},\vec{V},Z,c) \in [\![\lambda\vec{x},\vec{y}\,M[z := \mathrm{C}\vec{y}]]\!] \\ &\leftrightarrow \exists_Z.(\vec{U},\mathrm{C}\vec{V},Z) \subseteq [\![\lambda\vec{x},z\,N]\!] \wedge (\vec{U},\mathrm{C}\vec{V},Z,c) \in [\![\lambda\vec{x},z\,M]\!] \quad \text{(by IH)} \\ &\leftrightarrow (\vec{U},\mathrm{C}\vec{V},c) \in [\![\lambda\vec{x},z.MN]\!] \quad \text{(by } (A)\text{)}.\end{aligned}$$

*Case* $z$.

$$\begin{aligned}(\vec{U},\vec{V},c) \in [\![\lambda\vec{x},\vec{y}.\mathrm{C}\vec{y}]\!] = [\![\lambda\vec{x}\,\mathrm{C}]\!] &\leftrightarrow \exists_{\vec{b^*}}.\vec{V} \geq \vec{b^*} \wedge \mathrm{C}\vec{b^*} = c \\ &\leftrightarrow \mathrm{C}\vec{V} \geq c \\ &\leftrightarrow (\vec{U},\mathrm{C}\vec{V},c) \in [\![\lambda\vec{x},z\,z]\!].\end{aligned}$$

In all other cases both sides are clearly equivalent. □

We can now prove preservation of values under computation rules:

LEMMA. *For every computation rule* $D\vec{P}(\vec{y}) = M$ *of a defined constant* $D$, $[\![\lambda\vec{y}.D\vec{P}(\vec{y})]\!] = [\![\lambda\vec{y}\,M]\!]$.

PROOF. The following are equivalent:

$$\begin{aligned}&(\vec{V},b) \in [\![\lambda\vec{y}.D\vec{P}(\vec{y})]\!] \\ &(\vec{P}(\vec{V}),b) \in [\![D]\!] = [\![\lambda\vec{z}.D\vec{z}]\!] \quad \text{by (2.8)} \\ &(\vec{V},b) \in [\![\lambda\vec{y}\,M]\!],\end{aligned}$$

where the last step is by definition. □

**2.3.4. Examples.** We consider the doubling function $D\colon \mathrm{nat} \Rightarrow \mathrm{nat}$, addition $+\colon \mathrm{nat} \Rightarrow \mathrm{nat} \Rightarrow \mathrm{nat}$ and the fixed point operators $\mathcal{Y}_\rho$. Structural recursion could be treated as well.

*Doubling.* $D\colon \mathrm{nat} \Rightarrow \mathrm{nat}$ is defined by the computation rules

$$D0 = 0, \qquad D(Sn) = S(S(Dn)).$$

One can show easily that all tokens

$$(\{0\}, 0), \qquad (\{S^{n+1}0\}, S^{2n+2}0), \qquad (\{S^{n+1}*\}, S^{2n+2}*)$$

are in $[\![D]\!]$, and that any token $(V, c) \in [\![D]\!]$ is entailed by one of these.

*Addition.* $+\colon \mathrm{nat} \Rightarrow \mathrm{nat} \Rightarrow \mathrm{nat}$ is defined by the computation rules

$$n + 0 = n, \qquad n + Sm = S(n + m).$$

As above one shows that all tokens

$$(\{0\}, 0), \qquad (\{S^{n+1}0\}, S^{n+1}0), \qquad (\{S^{n+1}*\}, S^{n+1}*)$$

are in $[\![\lambda m.0 + m]\!]$, and that any token $(V, c) \in [\![\lambda m.0 + m]\!]$ is entailed by one of these. So we can conclude that $[\![\lambda m.0 + m]\!] = [\![\lambda m\, m]\!]$. This is of interest, because it allows us to replace $0 + M$ by $M$ for an *arbitrary* (not necessarily total) term $M$ without affecting the values.

*Fixed points.* The computation rule $\mathcal{Y}_\rho f = f(\mathcal{Y}_\rho f)$ defines the fixed point operator $\mathcal{Y}_\rho$ of type $(\rho \Rightarrow \rho) \Rightarrow \rho$.

## 2.4. Adequacy

The adequacy theorem of Plotkin (1977, Theorem 3.1) says that whenever the value of a closed term $M$ is a numeral, then $M$ head-reduces to this numeral. So in this sense the (denotational) semantics is (computationally) "adequate". Plotkin's proof is by induction on the types, and uses a computability predicate. We prove an adequacy theorem in our setting, for arbitrary computation rules.

**2.4.1. Operational semantics.** Recall that a token of a base type $\mu$ is a constructor expression (possibly involving $*$) whose outermost constructor is for $\mu$. We use $B$ to denote both, constructors $C$ and defined constants $D$.

Definition ($M \succ_1 N$, $M$ head-reduces to $N$).

$$(\lambda x\, M)N \succ_1 M[x := N], \qquad \frac{M \succ_1 M'}{MN \succ_1 M'N},$$

$$D\vec{P}(\vec{N}\,) \succ_1 M[\vec{y} := \vec{N}] \quad \text{for } D\vec{P}(\vec{y}) = M \text{ a computation rule},$$

$$\frac{M \succ_1 M'}{Ba_1 \dots a_n M \succ_1 Ba_1 \dots a_n M'} \quad \text{for } n < \mathrm{ar}(B).$$

$\succeq$ denotes the reflexive transitive closure of $\succ_1$.

Clearly for every term $M$ there is at most one $M'$ such that $M \succ_1 M'$; call $M$ *normal* if there is no such $M'$.

We define an "operational interpretation" (Martin-Löf, 1983) of formal neighborhoods $U$. To this end we define a notion $M \in [a]$, for $M$ closed, by induction on the type of the token $a$, and write $M \in [U]$ for $\forall_{a \in U} M \in [a]$.

DEFINITION. (a) For $a$ of base type $\mu$, $M \in [a]$ if and only if $\exists_{N \geq a} M \succeq N$.

(b) $M \in [(U, b)]$ if and only if $M \succeq \lambda x\, M'$ or $M \succeq B\vec{M}$ with length of $\vec{M}$ less than $\mathrm{ar}(B)$, and $\forall_{N \in [U]} MN \in [b]$.

LEMMA. *If $M \succeq N$, $N \in [V]$ and $V \geq U$, then $M \in [U]$.*

PROOF. Let $a \in U$. We show $M \in [a]$, i.e., $\exists_{K \geq a} M \succeq K$. Because of $V \geq U$ we have $c \in V$ such that $c \geq a$. Because of $N \in [c]$ we have a term $K$ such that $N \succeq K \geq c$. Hence $M \succeq N \succeq K \geq c \geq a$. □

THEOREM (Adequacy). *If $(\vec{U}, b) \in [\![\lambda \vec{x}\, M]\!]$, then $\lambda \vec{x}\, M \in [(\vec{U}', b')]$ for some $(\vec{U}', b') \geq (\vec{U}, b)$.*

PROOF. By induction on the rules defining $(\vec{U}, b) \in [\![\lambda \vec{x}\, M]\!]$, and cases on the form of $M$.

*Case $x_i$.*

$$\frac{U_i \geq b}{(\vec{U}, b) \in [\![\lambda \vec{x}\, x_i]\!]}(V).$$

We need $(\vec{U}', b') \geq (\vec{U}, b)$ such that $\lambda \vec{x}\, x_i \in [(\vec{U}', b')]$, i.e., $\forall_{\vec{K} \in [\vec{U}']} K_i \in [b']$. Take $\vec{U}' = \vec{U}$, $b' = b$. Let $K_i \in [U_i]$. Then by definition $K_i \in [b']$.

*Case $MN$.*

$$\frac{(\vec{U}, V) \subseteq [\![\lambda \vec{x}\, N]\!] \qquad (\vec{U}, V, c) \in [\![\lambda \vec{x}\, M]\!]}{(\vec{U}, c) \in [\![\lambda \vec{x}.MN]\!]}(A).$$

We need to find some $(\vec{U}', c') \geq (\vec{U}, c)$ such that $\lambda \vec{x}.MN \in [(\vec{U}', c')]$, i.e., $\forall_{\vec{K} \in [\vec{U}']} (MN)[\vec{x} := \vec{K}] \in [c']$.

By IH, for all $b \in V$ we have some $(\vec{U}_1, b') \geq (\vec{U}, b)$ such that $\lambda \vec{x}\, N \in [(\vec{U}_1, b')]$, i.e., $\forall_{\vec{K} \in [\vec{U}_1]} N[\vec{x} := \vec{K}] \in [b']$. Recall that $(\vec{U}_1, b') \geq (\vec{U}, b)$ means $\vec{U} \geq \vec{U}_1$ and $b' \geq b$. Hence we can pick the same $U_1$ for all $b \in V$, and

$$\forall_{\vec{K} \in [\vec{U}_1]} N[\vec{x} := \vec{K}] \in [V].$$

Also, by IH we have $(\vec{U}_2, V', c') \geq (\vec{U}, V, c)$ such that $\lambda \vec{x}\, M \in [(\vec{U}_2, V', c')]$, i.e.,

$$\forall_{\vec{K} \in [\vec{U}_2]} M[\vec{x} := \vec{K}] \in [(V', c')].$$

Recall that $(\vec{U}_2, V', c') \geq (\vec{U}, V, c)$ means $\vec{U} \geq \vec{U}_2$, $V \geq V'$ and $c' \geq c$.

Let $\vec{U}' := \vec{U}_1 \cup \vec{U}_2$ (component wise union), and fix $\vec{K} \in [\vec{U}']$. Clearly $\vec{K} \in [\vec{U}_1]$ and $\vec{K} \in [\vec{U}_2]$. From $M[\vec{x} := \vec{K}] \in (V', c')$ we know that $M[\vec{x} := \vec{K}] \succeq \lambda x\, M'$ or $M \succeq B\vec{M}$ with length of $\vec{M}$ less than $\mathrm{ar}(B)$, and also $\forall_{L \in [V']} . M[\vec{x} := \vec{K}]L \in [c']$.

Since $N[\vec{x} := \vec{K}] \in [V]$ and hence $\in [V']$ we obtain $(MN)[\vec{x} := \vec{K}] \in [c']$, as required.

*Case D*.

$$\frac{(\vec{U}, \vec{V}, b) \in [\![\lambda \vec{x}, \vec{y}\, M]\!]}{(\vec{U}, \vec{P}(\vec{V}), b) \in [\![\lambda \vec{x}\, D]\!]}(D),$$

with $D\vec{P}(\vec{y}) = M$ be a computation rule. We need $(\vec{U}', \vec{Z}, b') \geq (\vec{U}, \vec{P}(\vec{V}), b)$ such that $\lambda \vec{x}\, D \in [(\vec{U}', \vec{Z}, b')]$. Recall that $(\vec{U}', \vec{Z}, b') \geq (\vec{U}, \vec{P}(\vec{V}), b)$ means $\vec{U} \geq \vec{U}'$, $\vec{P}(\vec{V}) \geq \vec{Z}$ and $b' \geq b$.

By IH we have $(\vec{U}', \vec{V}', b') \geq (\vec{U}, \vec{V}, b)$ such that $\lambda \vec{x}, \vec{y}\, M \in [(\vec{U}', \vec{V}', b')]$, i.e.,

$$\forall_{\vec{K} \in [\vec{U}']} \forall_{\vec{N} \in [\vec{V}']}\, M[\vec{y} := \vec{N}] \in [b'].$$

Recall that $(\vec{U}', \vec{V}', b') \geq (\vec{U}, \vec{V}, b)$ means $\vec{U} \geq \vec{U}'$, $\vec{V} \geq \vec{V}'$ and $b' \geq b$.

Pick the required $\vec{U}', b'$ as the ones provided by the IH, and $\vec{Z} := \vec{P}(\vec{V}')$. We must show $\lambda \vec{x}\, D \in [(\vec{U}', \vec{P}(\vec{V}'), b')]$, i.e.,

$$\forall_{\vec{K} \in [\vec{U}']} \forall_{\vec{L} \in [\vec{P}(\vec{V}')]}\, D\vec{L} \in [b'].$$

Now fix $\vec{K} \in [\vec{U}']$ and $\vec{L} \in [\vec{P}(\vec{V}')]$. Then $\vec{L} \succeq \vec{P}(\vec{N})$ with $\vec{N} \in [\vec{V}']$. From $D\vec{P}(\vec{N}) \succ_1 M[\vec{y} := \vec{N}]$ and $M[\vec{y} := \vec{N}] \in [b']$ the claim follows.

*Case C*.

$$\frac{\vec{V} \geq \vec{b^*}}{(\vec{U}, \vec{V}, C\vec{b^*}) \in [\![\lambda \vec{x}\, C]\!]}(C).$$

We need $(\vec{U}', \vec{V}', b') \geq (\vec{U}, \vec{V}, C\vec{b^*})$ such that $\lambda \vec{x}\, C \in [(\vec{U}', \vec{V}', b')]$. Recall that $(\vec{U}', \vec{V}', b') \geq (\vec{U}, \vec{V}, C\vec{b^*})$ means $\vec{U} \geq \vec{U}'$, $\vec{V} \geq \vec{V}'$ and $b' \geq C\vec{b^*}$.

Pick $\vec{U}' := \vec{U}$, $\vec{V}' := \vec{V}$, $b' := C\vec{b^*}$. We must show $\lambda \vec{x}\, C \in [(\vec{U}, \vec{V}, C\vec{b^*})]$, i.e.,

$$\forall_{\vec{K} \in [\vec{U}]} \forall_{\vec{L} \in [\vec{V}]}\, C\vec{L} \in [C\vec{b^*}].$$

This follows from $\vec{V} \geq \vec{b^*}$. □

## 2.5. Total Functionals

Total ideals are important because one can prove their properties by (structural) induction. We also introduce the concept of *structure-total* ideals, first for a free algebra $\mu$ and then for arbitrary types. They are more general, because ideals at parameter positions need not be total, but

still allow to argue by induction. An example of the latter notion are lists whose structure (number of cons's) is known, but whose elements may be partial. This is of interest, because for such "structure-total" objects an obvious induction principle holds.

Kreisel (1959) states the important density theorem, which says that any finite functional can be extended to a total one. Proofs of various versions of the density theorem have been given by Ershov (1972), Berger (1993b), Stoltenberg-Hansen et al. (1994), Schwichtenberg (1996) and Kristiansen and Normann (1997). Here we give a proof for the practically important case where the base domains are not just the flat domain of natural numbers, but non-flat and possibly parametrized free algebras.

**2.5.1. Total and structure-total ideals.** It is well-known how one can single out the total functionals from the partial ones.

DEFINITION. The *total* ideals of type $\rho$ are defined inductively.

- *Case* $\mu$. For an algebra $\mu$, the total ideals $x$ are those of the form $\mathrm{C}\vec{z}$ with C a constructor of $\mu$ and $\vec{z}$ total (C denotes the continuous function $|r_{\mathrm{C}}|$).
- *Case* $\rho \Rightarrow \sigma$. An ideal $r$ of type $\rho \Rightarrow \sigma$ is total if and only if for all total $z$ of type $\rho$, the result $|r|(z)$ of applying $r$ to $z$ is total.

The *structure-total* ideals are defined similarly; the difference is that in case $\mu$ the ideals at parameter positions of C need not be total. – We write $x \in G_\rho$ to mean that $x$ is a total ideal of type $\rho$.

For instance, for nat the ideals 0, $S0$, $S(S0)$ etc. in Fig. 2 on page 66 are total, but $\bot$, $S\bot$, $S(S\bot)$, $\ldots$, $\infty$ are not. For list($\rho$), precisely all ideals of the form $\mathrm{cons}(x_1, \ldots \mathrm{cons}(x_n, \mathrm{nil}) \ldots)$ are structure-total. The total ones are those where in addition all list elements $x_1, \ldots, x_n$ are total.

For non-flat base domains it is easy to see that there are maximal but not total ideals: $\infty$ is an example for nat. This is less easy for flat base domains; a counterexample has been given by Ershov (1974); a more perspicious one (at type (nat $\Rightarrow$ nat) $\Rightarrow$ nat) is in (Stoltenberg-Hansen et al., 1994).

Conversely, the total continuous functionals need not be maximal ideals in $\boldsymbol{C}_\rho$: A counterexample is $\{\, (S^n 0, 0) \mid n \in \mathrm{nat} \,\}$, which clearly is a total object of type nat $\Rightarrow$ nat representing the constant function with value 0. However, addition of the pair $(\emptyset, 0)$ yields a different total object of type nat $\Rightarrow$ nat. However, it is easy to show both functionals are "equivalent" in the sense that they have the same behaviour on total arguments.

**2.5.2. Equality for total functionals.**

DEFINITION. An *equivalence* $\sim_\rho$ between total ideals $x_1, x_2 \in G_\rho$ is defined inductively.

- *Case* $\mu$. For an algebra $\mu$, two total ideals $x_1, x_2$ are equivalent if both are of the form $\mathrm{C}\vec{z}_1$ with the same constructor C of $\mu$, and we have $z_{1j} \sim_\tau z_{2j}$ for all $j$, where $\tau$ is a predecessor type (one of $\vec{\rho}, \vec{\sigma}_1, \ldots, \vec{\sigma}_n$) from the inductive clause for $\mu$.
- *Case* $\rho \Rightarrow \sigma$. Two ideals $r_1, r_2$ of type $\rho \Rightarrow \sigma$ are equivalent if and only if $\forall_{x \in G_\rho} f(x) \sim_\sigma g(x)$.

Clearly $\sim_\rho$ is an equivalence relation. Similarly, one can define an equivalence relation $\approx_\rho$ between structure-total ideals $x_1, x_2$.

We obviously want to know that $\sim_\rho$ (and similarly $\approx_\rho$) is compatible with application; we only treat $\sim_\rho$ here. The nontrivial part of this argument is to show that $x \sim_\rho y$ implies $f(x) \sim_\sigma f(y)$. First we need some lemmata. Recall that our partial continuous functionals are ideals (i.e., certain sets of tokens) in the information systems $\boldsymbol{C}_\rho$.

LEMMA. *If* $f \in G_\rho$, $g \in |\boldsymbol{C}_\rho|$ *and* $f \subseteq g$, *then* $g \in G_\rho$.

PROOF. By induction on $\rho$. For base types $\mu$ the claim easily follows from the IH. $\rho \Rightarrow \sigma$: Assume $f \in G_{\rho\Rightarrow\sigma}$ and $f \subseteq g$. We must show $g \in G_{\rho\Rightarrow\sigma}$. So let $x \in G_\rho$. We have to show $g(x) \in G_\sigma$. But $g(x) \supseteq f(x) \in G_\sigma$, so the claim follows by IH. □

LEMMA.

$$(f_1 \cap f_2)(x) = f_1(x) \cap f_2(x), \text{ for } f_1, f_2 \in |\boldsymbol{C}_{\rho\Rightarrow\sigma}| \text{ and } x \in |\boldsymbol{C}_\rho|. \tag{2.9}$$

PROOF. By the definition of $|r|$,

$$\begin{aligned}
&|f_1 \cap f_2|(x)\\
&= \{\, b \in C_\sigma \mid \exists_{U \subseteq x}\, (U, b) \in f_1 \cap f_2 \,\}\\
&= \{\, b \in C_\sigma \mid \exists_{U_1 \subseteq x}\, (U_1, b) \in f_1 \,\} \cap \{\, b \in C_\sigma \mid \exists_{U_2 \subseteq x}\, (U_2, b) \in f_2 \,\}\\
&= |f_1|(x) \cap |f_2|(x).
\end{aligned}$$

The part $\subseteq$ of the middle equality is obvious. For $\supseteq$, let $U_i \subseteq x$ with $(U_i, b) \in f_i$ be given. Choose $U = U_1 \cup U_2$. Then clearly $(U, b) \in f_i$ (as $(U_i, b) \geq (U, b)$ and $f_i$ is deductively closed). □

LEMMA. $f \sim_\rho g$ *if and only if* $f \cap g \in G_\rho$, *for* $f, g \in G_\rho$.

PROOF. By induction on $\rho$. For base types $\mu$ the claim easily follows from the IH. $\rho \Rightarrow \sigma$:

$$\begin{aligned}
f \sim_{\rho\Rightarrow\sigma} g &\iff \forall_{x \in G_\rho} f(x) \sim_\sigma g(x)\\
&\iff \forall_{x \in G_\rho} f(x) \cap g(x) \in G_\sigma \quad \text{by IH}\\
&\iff \forall_{x \in G_\rho} (f \cap g)x \in G_\sigma \quad \text{by (2.9)}\\
&\iff f \cap g \in G_{\rho\Rightarrow\sigma}.
\end{aligned}$$

This completes the proof. □

THEOREM. *$x \sim_\rho y$ implies $f(x) \sim_\sigma f(y)$, for $x, y \in G_\rho$ and $f \in G_{\rho\Rightarrow\sigma}$.*

PROOF. Since $x \sim_\rho y$ we have $x \cap y \in G_\rho$ by the last lemma. Now $f(x), f(y) \supseteq f(x \cap y)$ and hence $f(x) \cap f(y) \in G_\sigma$. But this implies $f(x) \sim_\sigma f(y)$ again by the last lemma. □

**2.5.3. Dense and separating sets.** We now prove the density theorem, which says that any finitely generated functional (i.e., any $\overline{U}$ with $U \in \mathrm{Con}_\rho$) can be extended to a total functional.

However, we need some assumptions on the base types for this theorem to hold. Otherwise, density might fail for the trivial reason that there are no total ideals at all (e.g., in $\mu\alpha\,(\alpha \to \alpha)$). A type $\mu\alpha_1, \ldots, \alpha_N(\kappa_1, \ldots, \kappa_n)$ is said to *have total ideals* if for every $j$ $(1 \le j \le N)$ there is a constructor type $\kappa_{i_j}$ of form (2.1) with $j_1, \ldots, j_n < j$. Then clearly for every $j$ we have a total ideal of type $\alpha_j$; call it $z_j$. Moreover, we assume that all base types are finitary. Then their total ideals are finite and maximal, which will be used in the proof.

THEOREM (Density). *Assume that all base types are finitary and have total ideals. Then for any $U \in \mathrm{Con}_\rho$ we can find an $x \in G_\rho$ such that $U \subseteq x$.*

PROOF. Call a type $\rho$ *dense* if $\forall_{U\in\mathrm{Con}_\rho}\exists_{x\in G_\rho} U \subseteq x$, and *separating* if

$$\forall_{U_1,U_2\in\mathrm{Con}_\rho}\big(U_1 \cup U_2 \notin \mathrm{Con}_\rho \Rightarrow \exists_{\vec{z}\in G}\,\mathrm{InCon}(\overline{U_1}(\vec{z}) \cup \overline{U_2}(\vec{z}))\big).$$

Here $\vec{z} \in G$ means that $\vec{z}$ is a sequence of total $z_i$ such that $U_j\vec{z}$ is of a base type $\mu$. We prove by simultaneous induction on $\rho$ that any type $\rho$ is dense and separating. This extended claim is needed for the inductive argument.

For base types $\mu$ both claims are easy: the fact that $\mu$ is separating is obvious, and density for $\mu$ can be inferred from the IH, as follows. For simplicity of notation assume that $\mu$ is non-simultaneously defined. Let $U \in \mathrm{Con}_\mu$. Then (since $\mu$ is finitary) $\exists_b\forall_{a\in U}\, b \ge a$. In the token $b$, replace every constructor symbol by its corresponding continuous function, every token at a parameter argument position by a total ideal of its type (which exists by IH), and every $*$ at a type-$\mu$-position by the total ideal $z$ of type $\mu$ (which exists by assumption). The result is the required total ideal.

$\rho \Rightarrow \sigma$ is separating: This will follow from the inductive hypotheses that $\rho$ is dense and $\sigma$ is separating. So let $W, W' \in \mathrm{Con}_{\rho\Rightarrow\sigma}$ such that $W \cup W' \notin \mathrm{Con}_{\rho\Rightarrow\sigma}$. Then there are $(U, a) \in W$ and $(U', a') \in W'$ such that $U \cup U' \in \mathrm{Con}_\rho$ but $\{a, a'\} \notin \mathrm{Con}_\sigma$. Since $\rho$ is dense, we have a $z \in G_\rho$ such that $U \cup U' \subseteq z$. Hence $a \in \overline{W}(z)$ and $a' \in \overline{W'}(z)$. Now since $\sigma$ is separating there are $\vec{z} \in G$ such that

$$\mathrm{InCon}(\overline{\{a\}}(\vec{z}) \cup \overline{\{a'\}}(\vec{z})),$$

hence also

$$\mathrm{InCon}(\overline{W}(z,\vec{z}) \cup \overline{W'}(z,\vec{z})).$$

This concludes the proof that $\rho \Rightarrow \sigma$ is separating.

$\rho \Rightarrow \sigma$ is dense: This will follow from the inductive hypotheses that $\rho$ is separating and $\sigma$ is dense. So fix $W = \{\, (U_i, a_i) \mid i \in I \,\} \in \mathrm{Con}_{\rho\Rightarrow\sigma}$. Consider $i, j$ such that $\{a_i, a_j\} \notin \mathrm{Con}_\sigma$. Then $U_i \cup U_j \notin \mathrm{Con}_\rho$. Since $\rho$ is separating, there are $\vec{z}_{ij} \in G$ and $k_{ij}, l_{ij} \in G_\mu$ such that with $k_{ij} := \overline{U_i}(\vec{z}_{ij})$ and $l_{ij} := \overline{U_j}(\vec{z}_{ij})$

$$\mathrm{InCon}(k_{ij} \cup l_{ij}).$$

We clearly may assume that $\vec{z}_{ij} = \vec{z}_{ji}$ and $(k_{ij}, l_{ij}) = (l_{ji}, k_{ji})$.

Now define for any $U \in \mathrm{Con}_\rho$ a set $I_U$ of indices $i \in I$ such that "$U$ behaves as $U_i$ with respect to the $\vec{z}_{ij}$". More precisely, let

$$I_U := \{\, i \in I \mid \forall_j(\{a_i, a_j\} \notin \mathrm{Con}_\sigma \to \overline{U}(\vec{z}_{ij}) = k_{ij}) \,\}.$$

We first show that

$$\{\, a_i \mid i \in I_U \,\} \in \mathrm{Con}_\sigma. \tag{2.10}$$

It suffices to show that $\{a_i, a_j\} \in \mathrm{Con}_\sigma$ for all $i, j \in I_U$. So let $i, j \in I_U$ and assume $\{a_i, a_j\} \notin \mathrm{Con}_\sigma$. Then $\overline{U}(\vec{z}_{ij}) = k_{ij}$ as $i \in I_U$ and $\overline{U}(\vec{z}_{ji}) = k_{ji}$ as $j \in I_U$, and because of $\vec{z}_{ij} = \vec{z}_{ji}$ and $\mathrm{InCon}(k_{ij} \cup k_{ji})$ (recall $l_{ij} = k_{ji}$) we could conclude that $\overline{U}(\vec{z}_{ij})$ would be inconsistent. This contradiction proves $\{a_i, a_j\} \in \mathrm{Con}_\sigma$ and hence (2.10).

Since (2.10) holds and $\sigma$ is dense by IH, we can find $y_{I_U} \in G_\sigma$ such that $a_i \in y_{I_U}$ for all $i \in I_U$. Define $r \subseteq \mathrm{Con}_\rho \times C_\sigma$ by

$$r(U,a) \iff \begin{cases} a \in y_{I_U}, & \text{if } \overline{U}(\vec{z}_{ij}) \text{ is finite and maximal for all } \vec{z}_{ij}; \\ \exists_{i\in I_U}\, a_i \geq_\sigma a, & \text{otherwise.} \end{cases}$$

We will show that $r \in G_{\rho\Rightarrow\sigma}$ and $W \subseteq r$.

For $W \subseteq r$ we have to show $r(U_i, a_i)$ for all $i \in I$. But this holds, since clearly $i \in I_{U_i}$ and also $a_i \in y_{I_{U_i}}$.

We now show that $r$ is an approximable map, i.e., that $r \in |\boldsymbol{C}_{\rho\Rightarrow\sigma}|$. To prove this we have to verify the defining properties of approximable maps.

(a). $r(U, b_1)$ and $r(U, b_2)$ implies $\{b_1, b_2\} \in \mathrm{Con}_\sigma$. If $\overline{U}(\vec{z}_{ij})$ is finite and maximal for all $\vec{z}_{ij}$, the claim follows from the consistency of $y_{I_U}$. If not, the claim follows from the general properties of acis's.

(b). $r(U, b)$, $V \geq_A U$ and $b \geq_B c$ implies $r(V, c)$. First assume that $\overline{U}(\vec{z}_{ij})$ is finite and maximal for all $\vec{z}_{ij}$. Then also $\overline{V}(\vec{z}_{ij})$ is maximal for all $\vec{z}_{ij}$. From $r(U, b)$ we get $b \in y_{I_U}$. We have to show that $c \in y_{I_V}$. But since $\overline{U}(\vec{z}_{ij})$ and $\overline{V}(\vec{z}_{ij})$ are maximal for all $\vec{z}_{ij}$ and $V \geq_\rho U$, they must have the same values on the $\vec{z}_{ij}$, hence $I_U = I_V$, so $y_{I_U} = y_{I_V}$ and therefore $c \in y_{I_V}$ by deductive closure. Now assume the contrary. From $r(U, b)$ we get $a_i \geq_\sigma b$

for some $i \in I_U$. From $V \geq_\rho U$ we can conclude $I_U \subseteq I_V$, by the definition of $I_U$. Hence $i \in I_V$, and also $b \in y_{I_V}$ (since $a_i \in y_{I_U}$ for all $i \in I_U$, and $y_{I_V}$ is deductively closed). Therefore $r(V, b)$ and hence $r(V, c)$.

This concludes the proof that $r$ is an approximable map. It remains to prove $r \in G_{\rho\Rightarrow\sigma}$. So let $x \in G_\rho$. We must show

$$|r|(x) = \{\, a \in C_\sigma \mid \exists_{U\subseteq x}\, r(U, a) \,\} \in G_\sigma.$$

Now $x(\vec{z}_{ij})$ is total for all $i, j$, hence by our assumption on base types finite and maximal. So there is some $U_{ij} \subseteq x$ such that $\overline{U_{ij}}(\vec{z}_{ij}) = x(\vec{z}_{ij})$. Let $U \subseteq x$ be the union of all the $U_{ij}$. Then by definition $r(U, a)$ for all $a \in y_{I_U}$. Therefore $y_{I_U} \subseteq |r|(x)$ and hence $|r|(x) \in G_\sigma$. □

**2.5.4. Applications of the density theorem.** As an application of the density theorem we prove a choice principle for total continuous functionals.

Theorem (Choice principle for total functionals). *There is an ideal* $\Gamma \in |\boldsymbol{C}_{(\rho\Rightarrow\sigma\Rightarrow\mathrm{boole})\Rightarrow\rho\Rightarrow\sigma}|$ *such that for any* $F \in G_{\rho\Rightarrow\sigma\Rightarrow\mathrm{boole}}$ *satisfying*

$$\forall_{x\in G_\rho}\exists_{y\in G_\sigma}\, F(x, y) = \mathtt{tt}$$

*we have* $\Gamma(F) \in G_{\rho\Rightarrow\sigma}$ *and*

$$\forall_{x\in G_\rho}\, F(x, \Gamma(F, x)) = \mathtt{tt}.$$

Proof. Let $V_0, V_1, V_2, \ldots$ be an enumeration of $\mathrm{Con}_\sigma$. By the density theorem we can find $y_n \in G_\sigma$ such that $V_n \subseteq y_n$. Define a relation $r \subseteq \mathrm{Con}_{\rho\Rightarrow\sigma\Rightarrow\mathrm{boole}} \times C_{\rho\Rightarrow\sigma}$ by

$$r(W, U, a) \iff \exists_m \forall_{i<m}\, (\overline{W}(\overline{U}, y_i) = \mathtt{ff} \wedge \overline{W}(\overline{U}, y_m) = \mathtt{tt} \wedge a \in y_m).$$

We first show that $\Gamma := r$ is an approximable map. To prove this we have to verify the clauses of the definition of approximable maps.

(a). $r(W, U_1, a_1)$ and $r(W, U_2, a_2)$ imply $\{(U_1, a_1), (U_2, a_2)\} \in \mathrm{Con}_{\rho\Rightarrow\sigma}$. Assume the premise and $U := U_1 \cup U_2 \in \mathrm{Con}_\rho$. We show $\{a_1, a_2\} \in \mathrm{Con}_\sigma$. The numbers $m_i$ in the definition of $r(W, U_i, a_i)$ are the same, $= m$ say. Hence $a_1, a_2 \in y_m$, and the claim follows from the consistency of $y_m$.

(b). $r(W', U, a)$, $W \geq W'$ and $(U, a) \geq (V, b)$ implies $r(W, V, b)$. Then $V \geq U$ and $a \geq b$. The claim follows from the definition of $r$, using the deductive closure of $y_m$. The $m$ from $r(W', U, a)$ can be used for $r(W, U, a)$.

We finally show that for all $F \in G_{\rho\Rightarrow\sigma\Rightarrow\mathrm{boole}}$ satisfying

$$\forall_{x\in G_\rho}\exists_{y\in G_\sigma}\, F(x, y) = \mathtt{tt}$$

and all $x \in G_\rho$ we have $rFx \in G_\sigma$ and $F(x, \Gamma(F, x)) = \mathtt{tt}$. So let $F$ and $x$ with these properties be given. By assumption there is a $y \in G_\sigma$ such that $F(x, y) = \mathtt{tt}$. Hence by the definition of application there is a $V_n \in \mathrm{Con}_\sigma$

such that $F(x, \overline{V_n}) = \mathtt{tt}$. Since $V_n \subseteq y_n$ we also have $F(x, y_n) = \mathtt{tt}$. Clearly we may assume here that $n$ is minimal with this property, i.e., that

$$F(x, y_0) = \mathtt{ff}, \dots, F(x, y_{n-1}) = \mathtt{ff}.$$

We show that $\Gamma(F, x) \supseteq y_n$; this suffices because the extension of a total ideals is total. Recall that

$$\Gamma(F) = \{\, (U, a) \in \mathrm{Con}_\rho \times C_\sigma \mid \exists_{W \subseteq F}\, r(W, U, a) \,\}$$

and

$$\begin{aligned}\Gamma(F, x) &= \{\, a \in C_\sigma \mid \exists_{U \subseteq x}\, (U, a) \in \Gamma(F) \,\} \\ &= \{\, a \in C_\sigma \mid \exists_{U \subseteq x} \exists_{W \subseteq F}\, r(W, U, a) \,\}.\end{aligned}$$

Let $a \in y_n$. By the choice of $n$ we get $U \subseteq x$ and $W \subseteq F$ such that

$$\forall_{i<n}\, \overline{W}(\overline{U}, y_i) = \mathtt{ff} \quad \text{and} \quad \overline{W}(\overline{U}, y_n) = \mathtt{tt}.$$

Therefore $r(W, U, a)$ and hence $a \in \Gamma(F, x)$. □

From the proofs of both theorems it can be seen easily that the functionals constructed are in fact computable. More precisely we have:

THEOREM (Effective density theorem). *For any $U \in \mathrm{Con}_\rho$ we can find a computable $x \in G_\rho$ such that $U \subseteq x$.*

PROOF. By inspection of the proof of the density theorem. To see that $r$ (in the proof that $\rho \Rightarrow \sigma$ is dense) is $\Sigma^0_1$-definable one needs that $\exists_{i \in I_U}\, a_i \geq a$ implies $a \in y_{I_U}$ for all $U$ and $a$, since by definition $a_i \in y_{I_U}$ for all $i \in I_U$. Hence

$r(U, a) \iff$

$\exists_{i \in I_U}\, a_i \geq a$ or ($a \in y_{I_U}$ and $\overline{U}(\vec{z}_{ij})$ is finite and maximal for all $\vec{z}_{ij}$).

Moreover, if $\overline{U}(\vec{z}_{ij})$ is finite and maximal for all $\vec{z}_{ij}$, one can actually compute $I_U$ (and not only an enumeration procedure for $I_U$). □

THEOREM (Effective choice principle). *There is a computable $\Gamma$ of type $(\rho \Rightarrow \sigma \Rightarrow \mathrm{boole}) \Rightarrow \rho \Rightarrow \sigma$ such that for any $F \in G_{\rho \Rightarrow \sigma \Rightarrow \mathrm{boole}}$ satisfying*

$$\forall_{x \in G_\rho} \exists_{y \in G_\sigma}\, F(x, y) = \mathtt{tt}$$

*we have $\Gamma(F) \in G_{\rho \Rightarrow \sigma}$ and*

$$\forall_{x \in G_\rho}\, F(x, \Gamma(F, x)) = \mathtt{tt}.$$

PROOF. Immediate from the proof of the choice principle for total continuous functionals. □

The effective choice principle generalizes the simple fact that whenever we know the truth of $\forall_{x\in\mathbb{N}}\exists_{y\in\mathbb{N}} P(x,y)$ with $P(x,y)$ decidable, then given $x$ we can just search for a $y$ such that $P(x,y)$ holds; the truth of $\forall_{x\in\mathbb{N}}\exists_{y\in\mathbb{N}} P(x,y)$ guarantees termination of the search.

## 2.6. Implementation

This "logic for computable functionals" is the basis for the Minlog proof assistant `www.minlog-system.de`, under development in Munich. It treats partial functionals as first class citizens: variables range over all partial continuous functionals of a given type. Since these functionals are viewed as sets of tokens, we in fact quantify over sets, so we have a second order theory. However, the existence axioms – here in the form of which terms are allowed in $\forall$-elimination – are weak in the sense that these terms involve quantifiers over functionals, so our theory remains predicative.

In contrast to Martin-Löf (1984), formulas and types are kept separate. This makes it possible to avoid dependent types, which simplifies the theory considerably. More importantly, by separating the logic rules from type theory one avoids the well-known difficulty: when propositions are viewed as types and types as domains, then – as every domain is inhabited by its bottom element – every proposition would have a proof.

Types are built from base types (non-flat and possibly infinitary free algebras, with type parameters) by forming function spaces; this suffices for our intended mathematical applications. For more metamathematical subjects one may also add universe formation processes, as in (Berger, Berghofer, Letouzey, and Schwichtenberg, 2006). Decidable predicates are viewed as boolean valued functions (and hence the rewrite mechanism described above applies to them), and inductive definitions are the common way to introduce undecidable predicates. In addition to free type variables also free predicate variables are allowed. They are viewed as placeholders for formulas (or more precisely, comprehension terms, that is formulas with some variables abstracted). However, in comprehension terms quantification over predicate variables is not allowed, since this would form a glaring impredicativity: we then would define a predicate (by the comprehension term) with reference to the totality of all predicates, to which the one to be defined belongs. A central application domain for the Minlog proof assistant is program extraction from constructive – and classical (Berger et al., 2002) – proofs. This is done by means of a realizability interpretation, which requires – when the formula to be realized is given by an inductively defined predicate – a (possibly non-finitary) free algebra as domain of the realizers.

Computable functionals are defined by "computation rules", as described in (Berger et al., 2003; Berger, 2005); these rules are added to the standard

conversion rules of typed $\lambda$-calculus. To simplify equational reasoning, the system identifies terms with the same normal form. Then it clearly is desirable to use other equations as rewrite rules as well; for instance, we not only want to rewrite $M + 0$ into $M$ (which is an instance of a computation rule), but also $0 + M$ into $M$. To justify this we need to prove $0 + \hat{m} = \hat{m}$, where $\hat{m}$ ranges over *all* (possibly partial) objects of type nat. The standard way to prove such equations is of course induction. However, induction is only valid for total objects (or – for types with parameters – "structure-total" objects; hence cannot be used for equations involving partial variables. Here the approach developed in the present chapter helps: one can prove the equality of the *values* of the two terms, by showing that both contain the same tokens, and then use reflection to conclude that the terms must be equal. The present chapter aims at preparing the ground for such proofs.

## 2.7. Notes

The material in the present chapter is taken from (Schwichtenberg, 2006).

CHAPTER 3

# Proof Interpretations

The Brouwer-Heyting-Kolmogorov interpretation (BHK-interpretation for short) of intuitionistic (and minimal) logic explains what it means to prove a logically compound statement in terms of what it means to prove its components; the explanations use the notions of *construction* and *constructive proof* as unexplained primitive notions. For prime formulas the notion of proof is supposed to be given. The clauses of the BHK-interpretation are:

- $p$ proves $A \wedge B$ if and only if $p$ is a pair $\langle p_0, p_1 \rangle$ and $p_0$ proves $A$, $p_1$ proves $B$;
- $p$ proves $A \to B$ if and only if $p$ is a construction transforming any proof $q$ of $A$ into a proof $p(q)$ of $B$;
- $\bot$ is a proposition without proof.
- $p$ proves $\forall_{x \in D} A(x)$ if and only if $p$ is a construction such that for all $d \in D$, $p(d)$ proves $A(d)$,
- $p$ proves $\exists_{x \in D} A(x)$ if and only if $p$ is of the form $\langle d, q \rangle$ with $d$ an element of $D$, and $q$ a proof of $A(d)$.

The problem with the BHK-interpretation clearly is its reliance on the unexplained notions of construction and constructive proof. Gödel has been concerned with this problem for more than 30 years. In 1941, Gödel gave a lecture at Yale university with the title "In what sense is intuitionistic logic constructive?". According to Kreisel, Gödel "wanted to establish that intuitionistic proof of existential theorems provide explicit realizers" (Gödel, 1990, p.219). Gödel published his "Dialectica interpretation" in (1958), and revised this work over and over again; its state in 1972 has been published in (Gödel, 1990). Troelstra, in his introductory note to the latter two papers writes in (Gödel, 1990, p.220/221):

> Gödel argues that, since the finististic methods considered are not sufficient to carry out Hilbert's program, one has to admit at least some abstract notions in a consistency proof; ... However, Gödel did not want to go as far as admitting Heyting's abstract notion of constructive proof; hence he tried to replace the notion of constructive proof by something more definite, less abstract (that is, more

> nearly finitistic), his principal candidate being a notion of "computable functional of finite type" which is to be accepted as sufficiently well understood to justify the axioms and rules of his system $\mathsf{T}$, an essentially logic-free theory of functionals of finite type.

We intend to explicate the notion of a computable functional of finite type as an ideal in an acis, as explained in Ch.2. However, already Gödel noted in (1990) that his proof interpretation is largely independent of a precise definition of computable functionals; one only needs to know that certain basis functionals are computable (including primitive recursion operators in finite types), and that they are closed under composition.

Building on Gödel's work (1958), we assign to every formula $A$ a new one $\exists_x A_1(x)$ with $A_1(x)$ $\exists$-free. Then from a derivation $M\colon A$ we want to extract a "realizing" term $r$ such that $A_1(r)$. The intention here is that its meaning should in some sense be related to the meaning of the original formula $A$. However, Gödel explicitely states in (1958, p.286) that his Dialectica interpretation is *not* the one intended by BHK-interpretation.

Special to our treatment of proof interpretations is the following.

- It is based on natural deduction (not on a Hilbert-style calculus preferred by Gödel (1958)), which is formulated as a system of *proof terms* with *assumption variables*.
- Following Oliva (2006), we bring out some similarities between the (modified) realizability interpretation mr and the Dialectica interpretation.
- Apart from decidable prime formulas, we also allow inductively defined ones, with quantifiers admitted in the clauses (examples are Tait's computability predicates and the levels of the hyperarithmetic hierarchy).
- We will exploit the possibility to substitute for logical falsity in minimal logic derivations. This idea is known under the label "$A$-translation" (Dragalin, 1979; Friedman, 1975) and its refinements (Berger, Buchholz, and Schwichtenberg, 2002).
- We notationally distinguish between the constructive existential quantifier $\exists$ and the classical one $\tilde{\exists}$. Then there is no need for a "negative" translation, and we can view "classical" arithmetic as the $\exists\vee$-free fragment of intuitionistic arithmetic.
- Practical considerations dictate that one should only extract realizers from formal proofs *relative* to some axioms or lemmata, which may or may not have computational content. One can even go one step further and give up the aim to produce exact realizers and look for "majorants" instead (in the sense of Howard's (1973)); this is

often sufficient for applications. Then more axioms can be admitted (since we only need majorants, not exact realizers). This line of research has been initiated by Kohlenbach in the 1990s, under the name of "monotone" Dialectica and realizability interpretation. It has found many applications in approximation and fixed point theory.

- We propose to treat the (clearly necessary) optimizations of the extracted realizers on the proof level already, by allowing "non-computational" quantifiers of Berger (1993a) and introducing a "let-construct" (avoiding multiple computations of the same term) by means of an "identity lemma".
- An implementation of the realizability as well as the Dialectica interpretation (in the Minlog proof assistant) makes it possible to experiment with extraction of realizers. We discuss some case studies, including Tait's proof of normalization for the simply typed lambda-calculus, and a proof of Dickson's lemma (based on the minimum principle).

## 3.1. Arithmetic in Finite Types

We define Heyting arithmetic $\mathsf{HA}$ and its extension $\mathsf{HA}^\omega$ to a finitely typed language. Its terms have been introduced above, in Sec.2.2.

Recall that we have a decidable equality $=_\mu\colon \mu \Rightarrow \mu \Rightarrow \mathrm{boole}$, for finitary base types $\mu$. Every every *atomic formula* has the form $\mathrm{atom}(r^{\mathrm{boole}})$, i.e., is built from a boolean term $r^{\mathrm{boole}}$. In particular, there is no need for (logical) falsity $\bot$, since we can take the atomic formula $F := \mathrm{atom}(\mathsf{ff})$ – called *arithmetical falsity* – built from the boolean constant $\mathsf{ff}$ instead.

The *formulas* of $\mathsf{HA}^\omega$ are built from atomic ones by the connectives $\to$, $\forall$, $\wedge$ and $\exists$. We define *negation* $\neg A$ by $A \to F$.

**3.1.1. Logic, induction.** We use natural deduction rules: $\to^+$, $\to^-$, $\forall^+$ and $\forall^-$. The logical axioms are $\wedge^+$, $\wedge^-$, $\exists^+$ and $\exists^-$, and the *truth axiom* $\mathrm{Ax}_{\mathsf{tt}}\colon \mathrm{atom}(\mathsf{tt})$.

The general form of *induction* over simultaneous free algebras $\vec\mu = \mu\vec\alpha\,\vec\kappa$, with goal formulas $\forall_{x_j^{\mu_j}} A_j(x_j)$ is as follows. For the constructor type

$$\kappa_i = \vec\rho \Rightarrow (\vec\sigma_1 \Rightarrow \alpha_{j_1}) \Rightarrow \ldots \Rightarrow (\vec\sigma_n \Rightarrow \alpha_{j_n}) \Rightarrow \alpha_j \in \mathrm{KT}(\vec\alpha)$$

we have the *step formula*

$$\begin{aligned} D_i := \forall_{y_1^{\rho_1},\ldots,y_m^{\rho_m},y_{m+1}^{\vec\sigma_1\Rightarrow\mu_{j_1}},\ldots,y_{m+n}^{\vec\sigma_n\Rightarrow\mu_{j_n}}} \\ \big(\forall_{\vec x^{\vec\sigma_1}} A_{j_1}(y_{m+1}\vec x) \to \cdots \to \forall_{\vec x^{\vec\sigma_n}} A_{j_n}(y_{m+n}\vec x) \to A_j(\mathrm{C}_i^{\vec\mu}(\vec y))\big). \end{aligned} \tag{3.1}$$

Here $\vec{y} = y_1^{\rho_1}, \dots, y_m^{\rho_m}, y_{m+1}^{\vec{\sigma}_1 \Rightarrow \mu_{j_1}}, \dots, y_{m+n}^{\vec{\sigma}_n \Rightarrow \mu_{j_n}}$ are the *components* of the object $\mathrm{C}_i^{\vec{\mu}}(\vec{y})$ of type $\mu_j$ under consideration, and

$$\forall_{\vec{x}^{\vec{\sigma}_1}} A_{j_1}(y_{m+1}\vec{x}), \dots, \forall_{\vec{x}^{\vec{\sigma}_n}} A_{j_n}(y_{m+n}\vec{x})$$

are the hypotheses available when proving the induction step. The induction axiom $\mathrm{Ind}_{\mu_j}^{\vec{x},\vec{A}}$ or shortly $\mathrm{Ind}_j$ then proves the universal closure of the formula

$$D_1 \to \cdots \to D_k \to \forall_{x_j^{\mu_j}} A_j(x_j).$$

We will often write $\mathrm{Ind}_j^{\vec{x},\vec{A}}$ for $\mathrm{Ind}_{\mu_j}^{\vec{x},\vec{A}}$, and omit the upper indices $\vec{x}, \vec{A}$ when they are clear from the context. In case of a non-simultaneous free algebra, i.e., of type $\mu\alpha\kappa$, for $\mathrm{Ind}_\mu^{x,A}$ we normally write $\mathrm{Ind}_{x,A}$.

Examples.

$$\begin{aligned}
&\mathrm{Ind}_{p,A}\colon A(\mathtt{tt}) \to A(\mathtt{ff}) \to \forall_{p^{\mathrm{boole}}} A(p),\\
&\mathrm{Ind}_{n,A}\colon A(0) \to \forall_n(A(n) \to A(\mathrm{S}n)) \to \ \forall_{n^{\mathrm{nat}}} A(n),\\
&\mathrm{Ind}_{l,A}\colon A(\mathrm{nil}) \to \forall_{x,l}(A(l) \to A(\mathrm{cons}(x,l))) \to \forall_{l^{\mathrm{list}(\alpha)}} A(l),\\
&\mathrm{Ind}_{x,A}\colon \forall_{y_1} A(\mathrm{inl}(y_1)) \to \forall_{y_2} A(\mathrm{inr}(y_2)) \to \forall_{x^{\rho_1+\rho_2}} A(x).
\end{aligned}$$

To express that every object of a pair type is in fact a pair we require a *pair elimination axiom*

$$\forall_{y^\rho, z^\sigma} A(\langle y, z\rangle) \to \forall_{x^{\rho\times\sigma}} A(x).$$

Let $\mathsf{HA}^\omega$ be the theory based on the axioms above including the induction axioms, and $\mathsf{ML}^\omega$ be the (many-sorted) minimal logic, where the induction axioms are left out.

**3.1.2. Equality.** Clearly we need the *compatibility axioms*

$$x_1 =_\mu x_2 \to A(x_1) \to A(x_2).$$

We define *pointwise equality* $=_\rho$, by induction on the type. $x_1 =_\mu x_2$ is already defined, and

$$\begin{aligned}
(x_1 =_{\rho\Rightarrow\sigma} x_2) &:= \forall_y(x_1 y =_\sigma x_2 y),\\
(x_1 =_{\rho\times\sigma} x_2) &:= (x_1 0 =_\rho x_2 0) \wedge (x_1 1 =_\rho x_2 1).
\end{aligned}$$

Later we will consider some more equality notions: *extensional equality* $=_\rho^e$, *hereditary extensional equality* $\approx_\rho$, and *Leibniz equality*, where the latter is defined inductively, by the introduction axiom

$$\mathrm{Eq}^+\colon \forall_x \mathrm{Eq}(x,x)$$

and the elimination axiom

$$\mathrm{Eq}^-\colon \forall_{x,y}\big(\forall_x A(x,x) \to \mathrm{Eq}(x,y) \to A(x,y)\big).$$

Notice that Leibniz equality introduces additional atomic formulas, which are not any more given by boolean terms. For types of level $\leq 1$, pointwise and extensional equality will coincide.

**3.1.3. Extensionality.** The *extensionality axioms* are

$$y_1 =_\rho y_2 \to xy_1 =_\sigma xy_2$$

(recall that $=_\tau$ denotes pointwise equality). We write $\mathsf{E\text{-}HA}^\omega$ when the extensionality axioms are present.

In Troelstra (1973), Howard proved that already the first non trivial instance of the extensionality scheme

$$y_1 =_1 y_2 \to xy_1 =_{\mathrm{nat}} xy_2$$

does not have a Dialectica realizer. In fact, he introduced the majorizing relation as a tool to prove this result. This is in contrast to the realizability interpretation, where extensionality axioms are unproblematic, since they are $\exists$-free.

It is customary to try to alleviate the difficulty of not being able to use extensionality when formalizing mathematical arguments (when an application of the Dialectica interpretation is envisaged) by adding a so-called *weak extensionality rule*

$$\frac{A_0 \to r =_\rho s}{A_0 \to t(r) =_\sigma t(s)} \quad (A_0 \text{ quantifier-free})$$

to the formal system considered. This "rule" is special in the sense that its premise must have been derived *without open assumptions*. – Since the conclusion is (equivalent to) a purely universal formula, adding the weak extensionality rule does not change the behaviour of the formal system w.r.t. the Dialectica interpretation.

We write $\mathsf{WE\text{-}HA}^\omega$ when the weak extensionality rule is present, but not the extensionality axioms.

**3.1.4. Axioms of choice and independence of premise.** We will also consider some more axiom schemes. The *axiom of choice* ($\mathrm{AC}_{\rho,\sigma}$) is the scheme

$$\forall_{x^\rho}\exists_{y^\sigma} A(x,y) \to \exists_{f^{\rho\Rightarrow\sigma}}\forall_{x^\rho} A(x, f(x)).$$

(AC) is the collection of all ($\mathrm{AC}_{\rho,\sigma}$). By *independence of premise* ($\mathrm{IP}^\omega_{\exists\text{-free}}$) we mean the scheme

$$(A \to \exists_{x^\rho} B) \to \exists_{x^\rho}(A \to B) \quad \text{with } A \text{ } \exists\text{-free and } x \notin \mathrm{FV}(A).$$

## 3.2. Realizability Interpretation

**3.2.1. The type of a realizer.** We assign to every formula $A$ an object $\tau(A)$ (a type or the symbol $\varepsilon$). $\tau(A)$ is intended to be the type of the program to be extracted from a proof of $A$. In case $\tau(A) = \varepsilon$ proofs of $A$ have no computational content; such formulas $A$ are called *Harrop formulas*.

$$
\begin{aligned}
\tau(P(\vec{s})) &:= \varepsilon,\\
\tau(\exists_{x^\rho} A) &:= \begin{cases} \rho & \text{if } \tau(A) = \varepsilon \\ \rho \times \tau(A) & \text{otherwise,} \end{cases}\\
\tau(\forall_{x^\rho} A) &:= \begin{cases} \varepsilon & \text{if } \tau(A) = \varepsilon \\ \rho \Rightarrow \tau(A) & \text{otherwise,} \end{cases}\\
\tau(A \to B) &:= \begin{cases} \tau(B) & \text{if } \tau(A) = \varepsilon \\ \varepsilon & \text{if } \tau(B) = \varepsilon \\ \tau(A) \Rightarrow \tau(B) & \text{otherwise,} \end{cases}\\
\tau(A \wedge B) &:= \begin{cases} \tau(B) & \text{if } \tau(A) = \varepsilon \\ \tau(A) & \text{if } \tau(B) = \varepsilon \\ \tau(A) \times \tau(B) & \text{otherwise.} \end{cases}
\end{aligned}
$$

**3.2.2. Extracted terms.** We now define the extracted term $[\![M]\!]$, for a derivation $M$ using axioms $\exists^\pm$, $\wedge^\pm$, induction axioms, (AC) and $(\mathrm{IP}^\omega_{\exists\text{-free}})$ and moreover some $\exists$-free axioms.

Assume first that $M$ derives a formula $A$ with $\tau(A) \neq \varepsilon$. Then its *extracted term* $[\![M]\!]$ of type $\tau(A)$ is

$$
\begin{aligned}
[\![u^A]\!] &:= x_u^{\tau(A)} \quad (x_u^{\tau(A)} \text{ uniquely associated with } u^A),\\
[\![\lambda u^A M]\!] &:= \begin{cases} [\![M]\!] & \text{if } \tau(A) = \varepsilon \\ \lambda x_u^{\tau(A)} [\![M]\!] & \text{otherwise,} \end{cases}\\
[\![M^{A\to B} N]\!] &:= \begin{cases} [\![M]\!] & \text{if } \tau(A) = \varepsilon \\ [\![M]\!][\![N]\!] & \text{otherwise,} \end{cases}\\
[\![(\lambda x^\rho M)^{\forall_x A}]\!] &:= \lambda x^\rho [\![M]\!],\\
[\![M^{\forall_x A} t]\!] &:= [\![M]\!] t.
\end{aligned}
$$

We also need extracted terms for the axioms mentioned above; these will be defined below. For derivations $M^A$ where $\tau(A) = \varepsilon$ (i.e., $A$ is a Harrop formula) we define $[\![M]\!] := \varepsilon$ ($\varepsilon$ some new symbol).

For the axioms

$$\exists^+_{x,A}\colon \forall_{x^\rho}(A \to \exists_{x^\rho} A)$$
$$\exists^-_{x,A,B}\colon \exists_{x^\rho} A \to \forall_{x^\rho}(A \to B) \to B \quad (x \notin \mathrm{FV}(B))$$

we set

$$[\![\exists^+_{x^\rho,A}]\!] := \begin{cases} \lambda x^\rho x & \text{if } \tau(A) = \varepsilon \\ \lambda x^\rho \lambda y^{\tau(A)} \langle x, y\rangle & \text{otherwise,} \end{cases}$$

$$[\![\exists^-_{x^\rho,A,B}]\!] := \begin{cases} \lambda x^\rho \lambda f^{\rho \Rightarrow \tau(B)}.fx & \text{if } \tau(A) = \varepsilon \\ \lambda z^{\rho \times \tau(A)} \lambda f^{\rho \Rightarrow \tau(A) \Rightarrow \tau(B)}.f(z0)(z1) & \text{otherwise.} \end{cases}$$

For the axioms

$$\wedge^+\colon A \to B \to A \wedge B$$
$$\wedge^-\colon (A \to B \to C) \to A \wedge B \to C$$

we set

$$[\![\wedge^+]\!] := \begin{cases} \lambda x^{\tau(A)} x & \text{if } \tau(B) = \varepsilon \\ \lambda y^{\tau(B)} y & \text{if } \tau(A) = \varepsilon \\ \lambda x^{\tau(A)} \lambda y^{\tau(B)} \langle x, y\rangle & \text{otherwise,} \end{cases}$$

$$[\![\wedge^-]\!] := \begin{cases} \lambda z^{\tau(C)} z & \text{if } \tau(A) = \varepsilon,\ \tau(B) = \varepsilon \\ \lambda f^{\tau(A) \Rightarrow \tau(C)} \lambda y^{\tau(B)}.fy & \text{if } \tau(A) = \varepsilon,\ \tau(B) \neq \varepsilon \\ \lambda f^{\tau(A) \Rightarrow \tau(C)} \lambda x^{\tau(A)}.fx & \text{if } \tau(A) \neq \varepsilon,\ \tau(B) = \varepsilon \\ \lambda f^{\tau(A) \Rightarrow \tau(B) \Rightarrow \tau(C)} \lambda z^{\tau(A) \times \tau(B)}.f(z0)(z1) & \text{if } \tau(A) \neq \varepsilon,\ \tau(B) \neq \varepsilon. \end{cases}$$

The extracted term $[\![\mathrm{Ind}_j]\!]$ of an induction axiom is defined to be the recursion operator $\mathcal{R}^{\vec{\mu},\vec{\tau}}_{\mu_j}$. Here $\vec{\mu},\vec{\tau}$ list only the types $\mu_j, \tau_j$ with $\tau_j := \tau(A_j) \neq \varepsilon$, i.e., the recursion operator is simplified accordingly.

Example. For the induction scheme

$$\mathrm{Ind}_{n,A}\colon A(0) \to \forall_n(A(n) \to A(n+1)) \to \forall_n A(n)$$

we have

$$[\![\mathrm{Ind}_{n,A}]\!] := \mathcal{R}^\tau_{\mathrm{nat}}\colon \tau \Rightarrow (\mathrm{nat} \Rightarrow \tau \Rightarrow \tau) \Rightarrow \mathrm{nat} \Rightarrow \tau,$$

where $\tau := \tau(A) \neq \varepsilon$.

As extracted terms of (AC) and $(\mathrm{IP}^\omega_{\exists\text{-free}})$ we can take identities of the appropriate types.

**3.2.3. Realizability.** We define the notion of *(modified) realizability*. The term "modified" is used for historical reasons, to distinguish this form of realizability from the (earlier) Kleene-style realizability. More precisely, we define formulas $r$ mr $A$, where $A$ is a formula and $r$ is either a term of type $\tau(A)$ if the latter is a type, or the symbol $\varepsilon$ if $\tau(A) = \varepsilon$.

$$
\begin{aligned}
r \text{ mr } P(\vec{s}) &:= P(\vec{s}),\\
r \text{ mr } (\exists_x A(x)) &:= \begin{cases} \varepsilon \text{ mr } A(r) & \text{if } \tau(A) = \varepsilon \\ r1 \text{ mr } A(r0) & \text{otherwise,} \end{cases}\\
r \text{ mr } (\forall_x A) &:= \begin{cases} \forall_x \varepsilon \text{ mr } A & \text{if } \tau(A) = \varepsilon \\ \forall_x rx \text{ mr } A & \text{otherwise,} \end{cases}\\
r \text{ mr } (A \to B) &:= \begin{cases} \varepsilon \text{ mr } A \ \to\ r \text{ mr } B & \text{if } \tau(A) = \varepsilon \\ \forall_x(x \text{ mr } A \ \to\ \varepsilon \text{ mr } B) & \text{if } \tau(A) \neq \varepsilon = \tau(B) \\ \forall_x(x \text{ mr } A \ \to\ rx \text{ mr } B) & \text{otherwise,} \end{cases}\\
r \text{ mr } (A \wedge B) &:= \begin{cases} (\varepsilon \text{ mr } A) \wedge (r \text{ mr } B) & \text{if } \tau(A) = \varepsilon \\ (r \text{ mr } A) \wedge (\varepsilon \text{ mr } B) & \text{if } \tau(B) = \varepsilon \\ (r0 \text{ mr } A) \wedge (r1 \text{ mr } B) & \text{otherwise.} \end{cases}
\end{aligned}
$$

Formulas which do not contain the existence quantifier $\exists$ play a special role in this context; we call them $\exists$-*free* (or *invariant*); in the literature such formulas are also called "negative". Their crucial property is that for an $\exists$-free formula $A$ we have $\varepsilon$ mr $A = A$. Notice also that every formula $r$ mr $A$ is $\exists$-free.

**3.2.4. Soundness.**

THEOREM. *Let $M\colon A$ be a derivation in $\mathsf{HA}^\omega + \mathrm{AC} + \mathrm{IP}^\omega_{\exists\text{-free}} + \mathrm{Ax}_{\exists\text{-free}}$ from assumptions $u_i\colon C_i$ $(i = 1, \dots, n)$. Then we can find a derivation $\mu(M)$ in $\mathsf{HA}^\omega + \mathrm{Ax}_{\exists\text{-free}}$ of $[\![M]\!]$ mr $A$ from assumptions $\bar{u}_i\colon x_{u_i}$ mr $C_i$.*

PROOF. Induction on $M$.
*Case* $u\colon A$. Then $\bar{u}\colon x_u$ mr $A$. Let $\mu(u) := \bar{u}$.
*Case* $c\colon A$, $c$ an axiom. These cases can be treated easily.
*Case* $\lambda u^A M^B$. We must find a derivation $\mu(\lambda u M)$ of

$$[\![\lambda u M]\!] \text{ mr } (A \to B).$$

*Subcase* $\tau(A) = \varepsilon$. Then $[\![\lambda u M]\!] = [\![M]\!]$, hence

$$[\![\lambda u M]\!] \text{ mr } (A \to B) \ \ = \ \ \varepsilon \text{ mr } A \ \to\ [\![M]\!] \text{ mr } B.$$

By IH we can define $\mu(\lambda u M) := \lambda \bar{u} \mu(M)$ with $\bar{u}\colon \varepsilon$ mr $A$.

*Subcase* $\tau(A) \neq \varepsilon = \tau(B)$. Then $[\![\lambda u M]\!] = \varepsilon$ and

$$[\![\lambda u M]\!] \text{ mr } (A \to B) \;=\; \forall_x(x \text{ mr } A \;\to\; \varepsilon \text{ mr } B),$$

and by IH we can define $\mu(\lambda u M) := \lambda x_u \lambda \bar{u} \mu(M)$ with $\bar{u}\colon x_u$ mr $A$.

*Subcase* $\tau(A) \neq \varepsilon \neq \tau(B)$. Then

$$[\![\lambda u M]\!] \text{ mr } (A \to B) \;=\; \forall_x(x \text{ mr } A \;\to\; [\![\lambda u M]\!] x \text{ mr } B).$$

Because of $[\![\lambda u M]\!] = \lambda x_u [\![M]\!]$ and since we identify terms with the same $\beta$-normal form, again by IH we can define $\mu(\lambda u M) := \lambda x_u \lambda \bar{u} \mu(M)$.

*Case* $M^{A \to B} N^A$. We must find a derivation $\mu(MN)$ of $[\![MN]\!]$ mr $B$.

*Subcase* $\tau(A) = \varepsilon$. Then $[\![MN]\!] = [\![M]\!]$. By IH we have derivations $\mu(M)$ of

$$[\![M]\!] \text{ mr } (A \to B) \;=\; \varepsilon \text{ mr } A \;\to\; [\![M]\!] \text{ mr } B$$

and $\mu(N)$ of $\varepsilon$ mr $A$; hence we can define $\mu(MN) := \mu(M)\mu(N)$.

*Subcase* $\tau(A) \neq \varepsilon = \tau(B)$. Then $[\![MN]\!] = \varepsilon$. By IH we have derivations $\mu(M)$ of

$$[\![M]\!] \text{ mr } (A \to B) \;=\; \forall_x(x \text{ mr } A \;\to\; \varepsilon \text{ mr } B)$$

and $\mu(N)$ of $[\![N]\!]$ mr $A$; hence we can define $\mu(MN) := \mu(M)[\![N]\!]\mu(N)$.

*Subcase* $\tau(A) \neq \varepsilon \neq \tau(B)$. Then $[\![MN]\!] = [\![M]\!][\![N]\!]$. By IH we have derivations $\mu(M)$ of

$$[\![M]\!] \text{ mr } (A \to B) \;=\; \forall_x(x \text{ mr } A \;\to\; [\![M]\!] x \text{ mr } B)$$

and $\mu(N)$ of $[\![N]\!]$ mr $A$; hence we can define $\mu(MN) := \mu(M)[\![N]\!]\mu(N)$.

*Case* $\lambda z M^A$. We must find a derivation $\mu(\lambda z M)$ of $[\![\lambda z M]\!]$ mr $\forall_z A$. By definition $[\![\lambda z M]\!] = \lambda z [\![M]\!]$.

*Subcase* $\tau(A) = \varepsilon$. Then

$$\lambda z [\![M]\!] \text{ mr } \forall_z A \;=\; \forall_z(\varepsilon \text{ mr } A)$$

and by IH we can define $\mu(\lambda z M) := \lambda z \mu(M)$. The variable condition is satisfied, since $\lambda z M^A$ is a derivation term, and hence $z$ does not occur free in any assumption variable $u\colon B$ free in $M^A$, hence also does not occur free in the free assumption $\bar{u}\colon x_u$ mr $B$.

*Subcase* $\tau(A) \neq \varepsilon$. Then

$$\lambda z [\![M]\!] \text{ mr } \forall_z A \;=\; \forall_z(\lambda z [\![M]\!]) z \text{ mr } A.$$

Since we identify terms with the same $\beta$-normal form, by IH we again can define $\mu(\lambda z M) := \lambda z \mu(M)$. As before one can see that the variable condition is satisfied.

*Case* $M^{\forall_z A(z)} t$. We must find a derivation $\mu(Mt)$ of $[\![Mt]\!]$ mr $A(t)$. By definition we have $[\![Mt]\!] = [\![M]\!]t$.

*Subcase* $\tau(A) = \varepsilon$. By IH we have a derivation of

$$[\![M]\!] \text{ mr } \forall_z A(z) \;=\; \forall_z(\varepsilon \text{ mr } A(z))$$

hence we can define $\mu(Mt) := \mu(M)t$.

*Subcase* $\tau(A) \neq \varepsilon$. By IH we have a derivation of

$$[\![M]\!] \text{ mr } \forall_z A(z) \;=\; \forall_z([\![M]\!]z \text{ mr } A(z)),$$

hence we again can define $\mu(Mt) := \mu(M)t$. □

REMARK. If $A$ is $\exists$-free, then $\varepsilon$ mr $A = A$. Hence for $\forall_{x^\rho}\exists_{y^\sigma} A(x,y)$ with $\exists$-free $A(x,y)$ we have $\tau(\forall_x\exists_y A(x,y)) = \rho \Rightarrow \sigma$ and

$$t \text{ mr } \forall_x\exists_y A(x,y) = \forall_x A(x,tx).$$

Since for every instance $B$ of (AC) or $(\mathrm{IP}^\omega_{\exists\text{-free}})$ one easily derives $[\![B]\!]$ mr $B$ in $\mathsf{HA}^\omega$, as a corollary to the Soundness Theorem we immediately obtain the following. Let $M\colon \forall_x\exists_y A(x,y)$ be a closed derivation in $\mathsf{HA}^\omega + \mathrm{AC} + \mathrm{IP}^\omega_{\exists\text{-free}} + \mathrm{Ax}_{\exists\text{-free}}$, where $A(x,y)$ is $\exists$-free. Then

$$\mathsf{HA}^\omega + \mathrm{Ax}_{\exists\text{-free}} \vdash \forall_x A(x, [\![M]\!](x)).$$

**3.2.5. Characterization.** We now consider the question under what conditions a formula $A$ and its modified realizability interpretation $\exists_x x$ mr $A$ are equivalent. It has been proven by Troelstra (1973, 3.4.8) that

THEOREM (Characterization).

$$\mathsf{HA}^\omega + \mathrm{AC} + \mathrm{IP}^\omega_{\exists\text{-free}} \vdash A \leftrightarrow \exists_x x \text{ mr } A.$$

*The direction "$\leftarrow$" can be proved in* $\mathsf{ML}^\omega$ *alone, provided the formulas considered only have "outer-$\exists$-premises", that is, premises of the form* $\exists_x A_1$ *with* $A_1$ $\exists$*-free.*

PROOF. Induction on $A$; we only treat the case $A \to B$ with $\tau(A) \neq \varepsilon$ and $\tau(B) \neq \varepsilon$.

$$\begin{aligned}
(A \to\ B) &\leftrightarrow (\exists_x x \text{ mr } A \to \exists_y y \text{ mr } B) && \text{by IH}\\
&\leftrightarrow \forall_x(x \text{ mr } A \to \exists_y y \text{ mr } B) && \text{by } \mathsf{ML}^\omega\\
&\leftrightarrow \forall_x\exists_y(x \text{ mr } A \to y \text{ mr } B) && \text{by } (\mathrm{IP}^\omega_{\exists\text{-free}})\\
&\leftrightarrow \exists_f\forall_x(x \text{ mr } A \to f(x) \text{ mr } B) && \text{by (AC)}\\
&\leftrightarrow \exists_f f \text{ mr } (A \to B).
\end{aligned}$$

Now assume that $A$ has only outer-$\exists$-premises. First notice that for a formula of the form $\exists_x A_1$ with $A_1$ $\exists$-free we have $\exists_x(x \text{ mr } \exists_x A_1) = \exists_x A_1$. We obtain

$$\begin{aligned}
\exists_f f \text{ mr } (A \to B) &\to \exists_f\forall_x(x \text{ mr } A \to f(x) \text{ mr } B)\\
&\to \forall_x\exists_y(x \text{ mr } A \to y \text{ mr } B)\\
&\to \forall_x(x \text{ mr } A \to \exists_y y \text{ mr } B)\\
&\to (\exists_x x \text{ mr } A \to \exists_y y \text{ mr } B)
\end{aligned}$$

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

by the remark above and the IH. □

**3.2.6. Extraction.** Using the Characterization Theorem, we can extend the remark above to arbitrary formulas $\forall_x \exists_y A(x,y)$.

THEOREM (Extraction). *Assume*

$$\mathsf{HA}^\omega + \mathrm{AC} + \mathrm{IP}^\omega_{\exists\text{-free}} + \mathrm{Ax}_{\exists\text{-free}} \vdash \forall_x \exists_y A(x,y)$$

*with $A(x,y)$ an arbitrary formula with at most the displayed variables free. Then we can find a closed $\mathsf{HA}^\omega$-term $t$ such that*

$$\mathsf{HA}^\omega + \mathrm{AC} + \mathrm{IP}^\omega_{\exists\text{-free}} + \mathrm{Ax}_{\exists\text{-free}} \vdash \forall_x A(x, tx).$$

*In fact,*

$$t = \begin{cases} \lambda x.[\![M]\!]x0 & \text{if } \tau(A(x,y)) \neq \varepsilon \\ [\![M]\!] & \text{otherwise.} \end{cases}$$

PROOF. We assume $\tau(A(x,y)) \neq \varepsilon$; otherwise the proof is even easier. $\mathsf{HA}^\omega + \mathrm{Ax}_{\exists\text{-free}}$ proves

$$\begin{aligned} &[\![M]\!] \text{ mr } \forall_x \exists_y A(x,y) && \text{by the Soundness Theorem} \\ &\forall_x([\![M]\!]x \text{ mr } \exists_y A(x,y)) \\ &\forall_x([\![M]\!]x1 \text{ mr } A(x,[\![M]\!]x0)). \end{aligned}$$

Hence $\mathsf{HA}^\omega + \mathrm{AC} + \mathrm{IP}^\omega_{\exists\text{-free}} + \mathrm{Ax}_{\exists\text{-free}} \vdash \forall_x A(x,[\![M]\!]x0)$ by the Characterization Theorem. □

## 3.3. Majorization and the Realizability Interpretation

Clearly one should consider extraction of realizers from formal proofs *relative* to some axioms or lemmata. One can even go one step further, give up the aim to produce exact realizers and look for "majorants" instead (in the sense of Howard's (1973)); this is often sufficient for applications. This line of research has been initiated by Kohlenbach in the 1990s, under the name of "monotone" realizability interpretation. One can then conveniently deal with additional assumptions $\mathrm{Ax}_{\forall\exists_\leq\exists\text{-free}}$ of the form

$$\forall_{x^\rho} \exists_{y \leq_\sigma rx} A_1(x,y) \qquad (A_1 \ \exists\text{-free}),$$

with $r$ a closed term of type $\rho \Rightarrow \sigma$. We need to consider strenghtened versions $\mathrm{Ax}'_{\forall\exists_\leq\exists\text{-free}}$ of these assumptions as well:

$$\exists_{Y \leq_{\rho\Rightarrow\sigma} r} \forall_{x^\rho} A_1(x, Yx).$$

Notice that with (AC) one can prove the strenghtened version from the original one.

**3.3.1. Majorization.** We assume here that all base types are finitary, and that $\geq_\mu$ is a given reflexive and transitive relation on the total ideals of base type $\mu$ such that

- for every $y \in G_\mu$ there are only finitely many $x \in G_\mu$ with $y \geq x$;
- there is a max-operation on $G_\mu$ such that

$$\max(x,y) \geq x, y,$$
$$z \geq x \to z \geq y \to z \geq \max(x,y).$$

$x$ is called *hereditarily majorizable* if there is an $x^*$ such that $x^* \operatorname{maj} x$.

We extend $\geq_\mu$ to higher types, in a *pointwise* fashion (as we did for $=_\mu$ above)

$$(x_1 \geq_{\rho\Rightarrow\sigma} x_2) := \forall_y(x_1 y \geq_\sigma x_2 y).$$

Following Howard (1973), we define a relation $x^* \operatorname{maj}_\rho x$ ($x^*$ *hereditarily majorizes* $x$) for $x^*, x \in G_\rho$, by induction on the type $\rho$:

$$(x^* \operatorname{maj}_\mu x) \quad := (x^* \geq_\mu x),$$
$$(x^* \operatorname{maj}_{\rho\Rightarrow\sigma} x) := \forall_{y^*,y}(y^* \operatorname{maj}_\rho y \to x^* y^* \operatorname{maj}_\sigma xy).$$

Lemma.

(a) $\vdash x^* =_\rho \tilde{x}^* \to x =_\rho \tilde{x} \to x^* \operatorname{maj}_\rho x \to \tilde{x}^* \operatorname{maj}_\rho \tilde{x}$.
(b) $\vdash x^* \operatorname{maj}_\rho x \to x \geq_\rho \tilde{x} \to x^* \operatorname{maj}_\rho \tilde{x}$.

Proof. Induction on $\rho$. We argue informally, and only treat (b). *Case* $\rho \Rightarrow \sigma$. Assume $y^* \operatorname{maj}_\rho y$. Then $x^* y^* \operatorname{maj}_\sigma xy$ and $xy \geq_\sigma \tilde{x}y$, hence by IH $x^* y^* \operatorname{maj}_\sigma \tilde{x}y$. □

**3.3.2. Majorization of closed $\mathsf{HA}^\omega$-terms.** Let 1 denote the type nat $\Rightarrow$ nat. Clearly, for every monotone function $D$ of type 1 we have $D \operatorname{maj} D$. Moreover, $\mathcal{R}^\tau_\mu$ is hereditarily majorizable:

Lemma (Majorization). (a) *Define* $M\colon (\mu \Rightarrow \tau) \Rightarrow \mu \Rightarrow \tau$ *with* $\tau = \vec{\rho} \Rightarrow \mu'$ *by*

$$Mfn\vec{x} := \max_{i\leq n} fi\vec{x}.$$

*Then* $\mathsf{HA}^\omega \vdash \forall_n \bar{f}n \operatorname{maj} fn \to M\bar{f} \operatorname{maj} f$.
(b) $\mathsf{HA}^\omega \vdash f^*, g^* \operatorname{maj} f, g \to \mathcal{R}_\mu f^* g^* n \operatorname{maj} \mathcal{R}_\mu fgn$.
(c) *Define* $\mathcal{R}^*_\mu fg := M(\mathcal{R}_\mu fg)$. *Then* $\mathsf{HA}^\omega \vdash \mathcal{R}^*_\mu \operatorname{maj} \mathcal{R}_\mu$.

Proof. We argue informally.
(a) Let $n^* \geq n$ and $\vec{x}^* \operatorname{maj} \vec{x}$; we must show $M\bar{f}n^*\vec{x}^* \geq fn\vec{x}$.

$$M\bar{f}n^*\vec{x}^* = \max_{i\leq n^*} \bar{f}i\vec{x}^* \geq \bar{f}n\vec{x}^* \geq fn\vec{x}.$$

(b) Induction on $n$; for simplicity we assume $\mu = \mathrm{nat}$. For 0 the claim is obvious, and in the step we have by IH $\mathcal{R}f^*g^*(\mathrm{S}n) =_{\mathrm{def}} g^*n(\mathcal{R}f^*g^*n)$ maj $gn(\mathcal{R}fgn) =_{\mathrm{def}} \mathcal{R}fg(\mathrm{S}n)$, where $=_{\mathrm{def}}$ is definitional equality.

(c) Let $f^*, g^*$ maj $f, g$. We must show $M(\mathcal{R}f^*g^*)$ maj $\mathcal{R}fg$. By (a) it suffices to prove $\forall_n \mathcal{R}f^*g^*n$ maj $\mathcal{R}fgn$. But this holds by (b). □

THEOREM. *Let $r(\vec{x})$ be a $\mathsf{HA}^\omega$-term with free variables among $\vec{x}$. Assume that $\mathsf{HA}^\omega \vdash c^*$ maj $c$ for all constants $c$ in $r$. Let $r^*$ be $r$ with all constants $c$ replaced by $c^*$. Then $\mathsf{HA}^\omega \vdash \vec{x}^*$ maj $\vec{x} \to r^*(\vec{x}^*)$ maj $r(\vec{x})$.*

PROOF. Induction on $r$. *Case* $\lambda y\, r(y, \vec{x})$. We argue informally. Assume $\vec{x}^*$ maj $\vec{x}$. We must show $y^*$ maj $y \to (\lambda y\, r^*(y, \vec{x}^*))y^*$ maj $(\lambda y\, r(y, \vec{x}))y$. So assume $y^*$ maj $y$. Then by IH $r^*(y^*, \vec{x}^*)$ maj $r(y, \vec{x})$, which is our claim. □

Hence every closed term $r$ of $\mathsf{HA}^\omega$ is hereditarily majorizable. In fact, we have constructed a closed term $r^*$ of $\mathsf{HA}^\omega$ such that $r^*$ maj $r$.

**3.3.3. Strong majorization.** Bezem's (1985) strong majorizability relation s-maj is a slight modification of Howard's. It is a transitive relation, which Howard's is not.

$$(x^* \text{ s-maj}_\mu\ x) \quad := (x^* \geq_\mu x),$$
$$(x^* \text{ s-maj}_{\rho\Rightarrow\sigma}\ x) := \forall_{y^*,y}(y^* \text{ s-maj}_\rho\ y \to x^*y^* \text{ s-maj}_\sigma\ xy, x^*y).$$

The following is easy to see:

LEMMA. (a) $\vdash x^*$ s-maj $x \to x^*$ s-maj $x^*$,
(b) $\vdash x^{**}$ s-maj $x^* \to x^*$ s-maj $x \to x^{**}$ s-maj $x$.

PROOF. We argue informally.

(a). For $\rho = \mu$ the claim follows from the reflexivity of $\geq_\mu$. For $\rho \Rightarrow \sigma$ the claim follows from the assumption.

(b). Induction on $\rho$. For $\rho = \mu$ the claim follows from the transitivity of $\geq_\mu$. For $\rho \Rightarrow \sigma$ assume $x^{**}$ s-maj $x^*$ and $x^*$ s-maj $x$, that is,

$$\forall_{y^*,y}(y^* \text{ s-maj}_\rho\ y \to x^{**}y^* \text{ s-maj}_\sigma\ x^*y, x^{**}y),$$
$$\forall_{y^*,y}(y^* \text{ s-maj}_\rho\ y \to x^*y^* \text{ s-maj}_\sigma\ xy, x^*y).$$

We show $x^{**}$ s-maj $x$. So assume $y^*$ s-maj$_\rho$ $y$. We must show $x^{**}y^*$ s-maj$_\sigma$ $xy, x^{**}y$. The latter is already given, and for the former we use that $y^*$ s-maj $y^*$, by (a). Hence $x^{**}y^*$ s-maj$_\sigma$ $x^*y^*$ s-maj$_\sigma$ $xy$, so the claim follows by IH. □

For a closed term $r$ of $\mathsf{HA}^\omega$, the closed $\mathsf{HA}^\omega$-term $r^*$ above also satisfies $r^*$ s-maj $r$.

### 3.3.4. Soundness with majorants.

THEOREM (Soundness with majorants). *Let $M$ be a derivation in*

$$\mathsf{HA}^\omega + \mathrm{AC} + \mathrm{IP}^\omega_{\exists\text{-free}} + \mathrm{Ax}_{\forall\exists_\le\exists\text{-free}} \text{ of a formula } A$$

*from assumptions $u_i\colon C_i$ $(i = 1, \dots, n)$. Let $x_i$ of type $\tau(C_i)$ be variables for realizers of the assumptions. Let $\vec{z}$ of type $\vec{\rho}$ be the variables free in $M$. Then we can find a closed term $[\![\lambda\vec{z}, \vec{u}\, M]\!] =: T^*$ of type $\tau(C_1) \Rightarrow \dots \Rightarrow \tau(C_n) \Rightarrow \vec{\rho} \Rightarrow \tau(A)$, and a derivation $\mu(M)$ in*

$$\mathsf{HA}^\omega + \mathrm{Ax}'_{\forall\exists_\le\exists\text{-free}}$$

*of the formula*

$$\exists_T\big(T^* \text{ maj } T \wedge \forall_{\vec{x},\vec{z}}(x_1 \text{ mr } C_1 \to \dots \to x_n \text{ mr } C_n \to T\vec{x}\vec{z} \text{ mr } A)\big).$$

PROOF. Induction on $M$.

*Case $u\colon A$.* Let $x$ of type $\tau(A)$ be a variable for a realizer of the assumption $u$. We need $T^*$ such that

$$\exists_T\big(T^* \text{ maj } T \wedge \forall_{x,\vec{z}}(x \text{ mr } A \to Tx \text{ mr } A)\big).$$

We can take $Tx\vec{z} := x$, which majorizes itself.

*Case $c\colon A$, $c$ an axiom.* Consider an axiom

$$\forall_{x^\rho}\exists_{y\le_\sigma rx}A_1(x,y) \qquad (A_1 \text{ } \exists\text{-free}),$$

with $r$ a closed term of type $\rho \Rightarrow \sigma$. We have to find a majorant of some $T$ such that

$$\begin{aligned}
&T \text{ mr } \forall_{x^\rho}\exists_{y\le_\sigma rx}A_1(x,y)\\
&\forall_{x^\rho}\big(\varepsilon \text{ mr } (Tx \le_\sigma rx) \wedge \varepsilon \text{ mr } A_1(x,Tx)\big)\\
&\forall_{x^\rho}\big(Tx \le_\sigma rx \wedge A_1(x,Tx)\big),
\end{aligned}$$

where in the last step we have used that for an $\exists$-free formula $B$, $\varepsilon$ mr $B$ is the same as $B$. We now use the corresponding axiom in $\mathrm{Ax}'_{\forall\exists_\le\exists\text{-free}}$:

$$\exists_{Y\le_{\rho\Rightarrow\sigma}r}\forall_{x^\rho}A_1(x,Yx).$$

Pick this $Y$ as the desired $T$. Then as a majorant for $Y$ we can take a closed term $r^*$ majorizing $r$.

For the other axioms we have already constructed a realizer, and we can take an arbitrary majorant of it.

*Case $\lambda u^A M^B$.* By IH we have a derivation of

$$\begin{aligned}
&\exists_T\big(T^* \text{ maj } T \wedge\\
&\quad \forall_{x_1,\dots,x_n,x}(x_1 \text{ mr } C_1 \to \dots \to x_n \text{ mr } C_n \to x \text{ mr } A \to T\vec{x}x \text{ mr } B)\big).
\end{aligned}$$

But $\forall_x(x \text{ mr } A \to T\vec{x}x \text{ mr } B)$ is the same as $T\vec{x}$ mr $(A \to B)$.

*Case* $M^{A\to B}N^A$. We argue informally. By IH we have

$$\begin{aligned} T\vec{x} \text{ mr } (A\to B) = \forall_x(x \text{ mr } A \to T\vec{x}x \text{ mr } B) \quad & \text{from } x_i \text{ mr } C_i \\ S\vec{x} \text{ mr } A \quad & \text{from } x_i \text{ mr } C_i. \end{aligned}$$

Instanciating $x$ with $S\vec{x}$ gives $T\vec{x}(S\vec{x})$ mr $B$ from $x_i$ mr $C_i$. Let $R\vec{x} := T\vec{x}(S\vec{x})$; we look for a majorant $R^*$ of $R$. Let $R^*\vec{x} := T^*\vec{x}(S^*\vec{x})$. Let $\vec{x}^*$ maj $\vec{x}$. Then $S^*\vec{x}^*$ maj $S\vec{x}$; hence $R^*\vec{x}^* = T^*\vec{x}^*(S^*\vec{x}^*)$ maj $T\vec{x}(S\vec{x}) = R\vec{x}$.

*Case* $\lambda x\, M^{A(x)}$. By IH we have a derivation of $T\vec{x}x$ mr $A(x)$ from $x_i$ mr $C_i$. Since the open assumptions $C_i$ do not have $x$ free, we obtain a derivation of $\forall_x(T\vec{x}x \text{ mr } A(x))$, that is to say of $T\vec{x}$ mr $\forall_x A(x)$, again from $x_i$ mr $C_i$.

*Case* $M^{\forall_x A(x)}s$. By IH we have a derivation of $T\vec{x}$ mr $\forall_x A(x)$ from $x_i$ mr $C_i$, that is to say of $\forall_x(T\vec{x}x \text{ mr } A(x))$. Instanciating $x$ with $s$ gives a derivation of $T\vec{x}s$ mr $A(s)$ from $x_i$ mr $C_i$. Assume for simplicity that $s$ is closed. Then we can take $\tilde{T}^*\vec{x} := T^*\vec{x}s^*$. □

### 3.3.5. Extraction of uniform bounds.

THEOREM. *Let $s$ be a closed $\mathsf{HA}^\omega$-term, $A(x,y,z)$ a formula with at most the displayed variables free, and $\tau$ a type of level $\le 2$. Assume that*

$$\mathsf{HA}^\omega + \mathrm{AC} + \mathrm{IP}^\omega_{\exists\text{-free}} + \mathrm{Ax}_{\forall\exists_\le\exists\text{-free}} \vdash \forall_{x^1}\forall_{y\le_\rho sx}\exists_{z^\tau} A(x,y,z).$$

*Then we can find a closed $\mathsf{HA}^\omega$-term $t$ such that*

$$\mathsf{HA}^\omega + \mathrm{AC} + \mathrm{IP}^\omega_{\exists\text{-free}} + \mathrm{Ax}_{\forall\exists_\le\exists\text{-free}} \vdash \forall_{x^1}\forall_{y\le_\rho sx}\exists_{z\le_\tau tx} A(x,y,z).$$

PROOF. Let $H^\omega := \mathsf{HA}^\omega + \mathrm{AC} + \mathrm{IP}^\omega_{\exists\text{-free}}$. Using $\mathrm{IP}^\omega_{\exists\text{-free}}$ we obtain

$$H^\omega + \mathrm{Ax}_{\forall\exists_\le\exists\text{-free}} \vdash \forall_{x^1,y}\exists_{z^\tau}(y \le_\rho sx \to A(x,y,z)).$$

By Soundness with Majorants we have a closed term $T^*$ such that in $H^\omega + \mathrm{Ax}'_{\forall\exists_\le\exists\text{-free}}$ we can derive the existence of some $T$ with $T^*$ maj $T$ and

$$T \text{ mr } \forall_{x^1}\forall_{y\le_\rho sx}\exists_{z^\tau} A(x,y,z).$$

Unfolding the definition of mr and using the fact that $y \le_\rho sx$ is $\exists$-free we obtain (assuming $\tau(A(x,y,z)) \neq \varepsilon$)

$$H^\omega + \mathrm{Ax}'_{\forall\exists_\le\exists\text{-free}} \vdash \forall_{x^1}\forall_{y\le_\rho sx}\, Txy1 \text{ mr } A(x,y,Txy0)$$

and hence by the Characterization Theorem in Sec.3.2.5

$$H^\omega + \mathrm{Ax}'_{\forall\exists_\le\exists\text{-free}} \vdash \forall_{x^1}\forall_{y\le_\rho sx} A(x,y,Txy0).$$

Notice that using (AC) we can replace $\mathrm{Ax}'_{\forall\exists_\le\exists\text{-free}}$ by the original $\mathrm{Ax}_{\forall\exists_\le\exists\text{-free}}$.

Let $t_1 := \lambda x\lambda y.Txy0$. Pick majorizing terms $s^*, t_1^*$ for $s, t_1$. Writing $x^M$ for $Mx$ with $M$ from the Majorization Lemma in Sec.3.3.2 we have $s^*x^M \operatorname{maj}_\rho sx$, hence

$$\mathsf{HA}^\omega \vdash \forall_{x^1}\forall_{y\le_\rho sx}\, s^*x^M \operatorname{maj}_\rho y.$$

For simplicity assume $\tau = 2 := (\mathrm{nat} \Rightarrow \mathrm{nat}) \Rightarrow \mathrm{nat}$. Then

$$\mathsf{HA}^\omega \vdash \forall_{x^1}\forall_{y\le_\rho sx}\forall_f\, t_1^*x^M(s^*x^M)f^M \ge_{\mathrm{nat}} t_1xyf.$$

Hence we can take $t := \lambda x\lambda f.t_1^*x^M(s^*x^M)f^M$, because $tx \ge_2 t_1xy =_{\mathrm{def}} Txy0$. □

Corollary (Fan Rule). *Let $A(y,n)$ be a formula with at most the displayed variables free. Assume that*

$$\mathsf{HA}^\omega + \mathrm{AC} + \mathrm{IP}^\omega_{\exists\text{-free}} + \mathrm{Ax}_{\forall\exists\le\exists\text{-free}} \vdash \forall_{y^1}\exists_{n^{\mathrm{nat}}}\, A(y,n).$$

*Then*

$$\mathsf{HA}^\omega + \mathrm{AC} + \mathrm{IP}^\omega_{\exists\text{-free}} + \mathrm{Ax}_{\forall\exists\le\exists\text{-free}} \vdash \forall_{x^1}\exists_{m^{\mathrm{nat}}}\forall_{y\le_1 x}\exists_{n\le_{\mathrm{nat}} m}\, A(y,n).$$

Proof. Let $s$ be the identity in the theorem above. Take $m := tx$. □

## 3.4. Dialectica Interpretation

In his original functional interpretation of (1958), Gödel assigned to every formula $A$ a new one $\exists_{\vec{x}}\forall_{\vec{y}}A_D(\vec{x},\vec{y})$ with $A_D(\vec{x},\vec{y})$ quantifier-free. Here $\vec{x}$, $\vec{y}$ are lists of variables of finite types; the use of higher types is necessary even when the original formula $A$ was first-order. He did this in such a way that whenever a proof of $A$ say in constructive arithmetic was given, one could produce closed terms $\vec{r}$ such that the quantifier-free formula $A_D(\vec{r},\vec{y})$ is provable in $\mathsf{T}$.

In (1958) Gödel referred to a Hilbert-style proof calculus. However, since the realizers will be formed in a $\lambda$-calculus formulation of system $\mathsf{T}$, Gödel's interpretation becomes a lot more perspicious when it is done for a natural deduction calculus. Such a natural deduction based treatment of the Dialectica interpretation has been given by Jørgensen (2001) and (Hernest, 2006). However, both authors use a natural deduction system where open assumptions are viewed as formulas. Then the well-known necessity of *contractions* shows up when an application of the implication introduction rule $\to^+$ discharges two (Jørgensen, 2001) or many (Hernest, 2006) assumption formulas. Peculiar to the present appoach is that we view the natural deduction calculus as a system of *proof terms* with *assumption variables*. In this form the Curry-Howard correspondence (formulas correspond to types, and proofs to terms) is best visible, and the contractions are necessary in the (only) logical rule with two premises: modus ponens (or implication elimination $\to^-$).

**3.4.1. Positive and negative types; cleaning.** To determine the types of $x$ and $y$, we assign to every formula $A$ types $\tau^+(A)$, $\tau^-(A)$. The type $\tau^+(A)$ is intended to be the type of a (Dialectica-)realizer to be extracted from a proof of $A$, and $\tau^-(A)$ the type of a challenge for the claim that this term realizes $A$.

Rather than including amongst the types a special "nulltype" object $\varepsilon$ (indicating no computational content) and case distinctions – as we did in our treatment of the realizability interpretation –, it is more convenient here to use the unit type unit instead and so avoid case distinctions. Using some obvious isomorphisms (like $(\rho \Rightarrow \text{unit}) \cong \text{unit}$ and $(\text{unit} \Rightarrow \rho) \cong \rho$) we can later "clean" such types. Define

$$\begin{aligned}
&\tau^+(P(\vec{s})) := \text{unit}, && \tau^-(P(\vec{s})) := \text{unit},\\
&\tau^+(\forall_{x^\rho} A) := \rho \Rightarrow \tau^+(A), && \tau^-(\forall_{x^\rho} A) := \rho \times \tau^-(A),\\
&\tau^+(\exists_{x^\rho} A) := \rho \times \tau^+(A), && \tau^-(\exists_{x^\rho} A) := \tau^-(A).
\end{aligned}$$

and for implication

$$\begin{aligned}
&\tau^+(A \to B) := \big(\tau^+(A) \Rightarrow \tau^+(B)\big) \times \big(\tau^+(A) \Rightarrow \tau^-(B) \Rightarrow \tau^-(A)\big),\\
&\tau^-(A \to B) := \tau^+(A) \times \tau^-(B).
\end{aligned}$$

In case $\tau^+(A)$ ($\tau^-(A)$) is $\neq$ unit we say that $A$ has *positive (negative) computational content*.

**3.4.2. Gödel translation.** For every formula $A$ and terms $r$ of type $\tau^+(A)$ and $s$ of type $\tau^-(A)$ we define a new quantifier-free formula $|A|^r_s$ by induction on $A$.

$$\begin{aligned}
|P(\vec{s})|^r_s &:= P(\vec{s}),\\
|\forall_x A(x)|^r_s &:= |A(s0)|^{r(s0)}_{s1},\\
|\exists_x A(x)|^r_s &:= |A(r0)|^{r1}_{s},\\
|A \to B|^r_s &:= |A|^{s0}_{r1(s0)(s1)} \to |B|^{r0(s0)}_{s1}.
\end{aligned}$$

The formula $\exists_x \forall_y |A|^x_y$ is called the *Gödel translation* of $A$ and is often denoted by $A^D$. Its quantifier-free kernel $|A|^x_y$ might be called *Gödel kernel*; it is denoted by $A_D$.

For readability we sometimes write terms of a pair type in pair form. Then

$$\begin{aligned}
|\forall_z A|^x_{z,y} &:= |A|^{xz}_y,\\
|\exists_z A|^{z,x}_y &:= |A|^x_y,\\
|A \to B|^{f,g}_{x,u} &:= |A|^x_{gxu} \to |B|^{fx}_u.
\end{aligned}$$

**3.4.3. Soundness.** We now prove the soundness of the Dialectica interpretation, for our natural deduction formulation of the underlying logic. The precise formulation will involve instances ($\mathrm{IP}^\omega_\forall$) of the independence of premise scheme:

$$(\forall_x A_0 \to \exists_{y^\rho} B) \to \exists_{y^\rho}(\forall_x A_0 \to B) \quad (y \notin \mathrm{FV}(\forall_x A_0)).$$

with $A_0$ quantifier-free. Moreover, we need to consider the (constructively doubtful) *Markov principle* ($\mathrm{M}^\omega$), for higher type variables and quantifier-free formulas $A_0, B_0$:

$$(\forall_y A_0 \to B_0) \to \exists_y (A_0 \to B_0) \quad (y \notin \mathrm{FV}(B_0)).$$

THEOREM (Soundness). *Let $M$ be a derivation in*

$$\mathsf{WE\text{-}HA}^\omega + \mathrm{AC} + \mathrm{IP}^\omega_\forall + \mathrm{M}^\omega + \mathrm{Ax}_\forall \text{ of a formula } A$$

*from assumptions $u_i \colon C_i$ $(i = 1, \dots, n)$. Let $x_i$ of type $\tau^+(C_i)$ be variables for realizers of the assumptions, and $y$ be a variable of type $\tau^-(A)$ for a challenge of the goal. Then we can find terms $[\![M]\!]^+ =: t$ of type $\tau^+(A)$ with $y \notin \mathrm{FV}(t)$ and $[\![M]\!]^-_i =: r_i$ of type $\tau^-(C_i)$, and a derivation $\mu(M)$ in*

$$\mathsf{WE\text{-}HA}^\omega + \mathrm{Ax}_\forall \text{ of the formula } |A|^t_y$$

*from assumptions $\bar{u}_i \colon |C_i|^{x_i}_{r_i}$.*

PROOF. Induction on $M$.

*Case $u \colon A$.* Let $x$ of type $\tau^+(A)$ be a variable for a realizer of the assumption $u$. Define $[\![u]\!]^+ := x$ and $[\![u]\!]^- := y$.

*Case $c \colon A$, $c$ an axiom.* These cases need to be treated separately (see below).

*Case $\lambda u^A M^B$.* By IH we have a derivation of $|B|^t_z$ from $\bar{u} \colon |A|^x_r$ and $\bar{u}_i \colon |C_i|^{x_i}_{r_i}$, where $\bar{u} \colon |A|^x_r$ may be absent. Substitute $y0$ for $x$ and $y1$ for $z$. By $(\to^+)$ we obtain $|A|^{y0}_{r[x,z:=y0,y1]} \to |B|^{t[x:=y0]}_{y1}$, which is (up to $\beta$-conversion)

$$|A \to B|^{\lambda x\, t, \lambda x, z\, r}_y, \quad \text{from} \quad \bar{u}'_i \colon |C_i|^{x_i}_{r_i[x,z:=y0,y1]}.$$

Here $r$ is the canonical inhabitant of the type $\tau^-(A)$ in case $\bar{u} \colon |A|^x_r$ is absent. So we can define the required terms by (assuming that $u^A$ is $u_1$)

$$[\![\lambda u\, M]\!]^+ := (\lambda x\, [\![M]\!]^+, \lambda x, z\, [\![M]\!]^-_1),$$
$$[\![\lambda u\, M]\!]^-_i := [\![M]\!]^-_{i+1}[x, z := y0, y1].$$

*Case $M^{A\to B} N^A$.* By IH we have a derivation of

$$|A \to B|^t_x = |A|^{x0}_{t1(x0)(x1)} \to |B|^{t0(x0)}_{x1} \quad \text{from } |C_i|^{x_i}_{p_i},\ |C_k|^{x_k}_{p_k}, \text{ and of}$$
$$|A|^s_z \quad \text{from } |C_j|^{x_j}_{q_j},\ |C_k|^{x_k}_{q_k}.$$

Substituting $\langle s, y\rangle$ for $x$ in the first derivation and of $t1sy$ for $z$ in the second derivation gives

$$\begin{aligned} &|A|^{s}_{t1sy} \to |B|^{t0s}_{y} \quad \text{from } |C_i|^{x_i}_{p'_i},\ |C_k|^{x_k}_{p'_k}, \text{ and} \\ &|A|^{s}_{t1sy} \qquad\qquad\ \ \text{from } |C_j|^{x_j}_{q'_j},\ |C_k|^{x_k}_{q'_k}. \end{aligned}$$

Now we contract $|C_k|^{x_k}_{p'_k}$ and $|C_k|^{x_k}_{q'_k}$: since $|C|^{x}_{w}$ is quantifier-free, there is a boolean-valued term $r_C$ such that

$$|C|^{x}_{w} \leftrightarrow r_C w = \mathtt{tt}.$$

Hence with $r_k := \mathrm{if}(r_{C_k} p'_k, q'_k, p'_k)$ we can derive both $|C_k|^{x_k}_{p'_k}$ and $|C_k|^{x_k}_{q'_k}$ from $|C_k|^{x_k}_{r_k}$. Using $(\to^-)$ we obtain

$$|B|^{t0s}_{y} \quad \text{from } |C_i|^{x_i}_{p'_i},\ |C_j|^{x_j}_{q'_j},\ |C_k|^{x_k}_{r_k}.$$

So $[\![MN]\!]^+ := t0s$ and $[\![MN]\!]^-_i := p'_i$, $[\![MN]\!]^-_j := q'_j$, $[\![MN]\!]^-_k := r_k$.

*Case* $\lambda x\, M^{A(x)}$. By IH we have a derivation of $|A(x)|^{t}_{z}$ from $\bar{u}_i\colon |C_i|^{x_i}_{r_i}$. Substitute $y0$ for $x$ and $y1$ for $z$. We obtain $|A(y0)|^{t[x:=y0]}_{y1}$, which is (up to $\beta$-conversion)

$$|\forall_x A(x)|^{\lambda x\, t}_{y}, \quad \text{from} \quad \bar{u}'_i\colon |C_i|^{x_i}_{r_i[x,z:=y0,y1]}.$$

So we can define the required terms by

$$\begin{aligned} [\![\lambda x\, M]\!]^+ &:= \lambda x\, [\![M]\!]^+, \\ [\![\lambda x\, M]\!]^-_i &:= [\![M]\!]^-_i[x, z := y0, y1]. \end{aligned}$$

*Case* $M^{\forall_x A(x)} s$. By IH we have a derivation of $|\forall_x A(x)|^{t}_{z} = |A(z0)|^{t(z0)}_{z1}$ from $|C_i|^{x_i}_{r_i}$. Substituting $\langle s, y\rangle$ for $z$ gives

$$|A(s)|^{ts}_{y} \quad \text{from } |C_i|^{x_i}_{r_i[z:=\langle s,y\rangle]}.$$

So $[\![Ms]\!]^+ := ts$ and $[\![Ms]\!]^-_i := r_i[z := \langle s, y\rangle]$.

We treat the axioms, and show that each of them has a "Dialectica realizer", that is, a term $t$ such that $\mathsf{HA}^\omega + \mathrm{Ax}_\forall$ proves $\forall_y |A|^{t}_{y}$.

For the existence introduction and elimination axioms

$$\begin{aligned} &\forall_z(A \to \exists_z A), \\ &\forall_z(A \to B) \to \exists_z A \to B \quad (z \notin \mathrm{FV}(B)) \end{aligned}$$

this is easy: In case of the introduction axiom, by definition

$$|\forall_z(A \to \exists_z A)|^{f,g}_{z,x,w} = |A \to \exists_z A|^{fz,gz}_{x,w} = |A|^{x}_{gzxw} \to |\exists_z A|^{fzx}_{w}.$$

Define $fzx := \langle z, x\rangle$ and $gzxw := w$. Then the premise and the conlusion are identical, since by definition $|A|^{x}_{w}$ is the same as $|\exists_z A|^{z,x}_{w}$.

For the elimination axiom, again by definition

$$|\forall_z(A \to B)|^{f,g}_{z,x,w} = |A \to B|^{fz,gz}_{x,w} \qquad = |A|^{x}_{gzxw} \to |B|^{fzx}_{w},$$
$$|\exists_z A \to B|^{f,g}_{z,x,w} = |\exists_z A|^{z,x}_{gzxw} \to |B|^{fzx}_{w} = |A|^{x}_{gzxw} \to |B|^{fzx}_{w}.$$

Now generally, if $|C|^u_v = |D|^u_v$, then by definition

$$|C \to D|^{\lambda uu,\lambda u,v\, v}_{u,v} = |C|^u_v \to |D|^u_v.$$

Consider the algebra of the natural numbers, given by constructors 0 and S. The induction schema then reads

$$A(0) \to \forall_m(A(m) \to A(m+1)) \to \forall_n A(n). \tag{3.2}$$

Let $B(n) := A(0) \to \forall_m(A(m) \to A(m+1)) \to A(n)$. Clearly we can derive $B(0)$ and $B(n) \to B(n+1)$. By those parts of the proof of the Soundness Theorem that we have dealt with already, we obtain realizing terms $s$ and $t, r$ and derivations of $|B(0)|^s_y$ and of $|B(n) \to B(n+1)|^{t,r}_{x,u}$, hence of

$$|B(n)|^x_{rxu} \to |B(n+1)|^{tx}_u$$
$$\forall_y\, |B(n)|^x_y \to |B(n+1)|^{tx}_u$$
$$\forall_y\, |B(n)|^x_y \to \forall_y\, |B(n+1)|^{tx}_y.$$

So if we define $g(0) := s$ and $g(n+1) := t(g(n))$, then we have proved by induction that $\forall_y\, |B(n)|^{g(n)}_y$, hence that $\exists_g \forall_y\, |\forall_n B(n)|^g_y$. But $\forall_n B(n)$ clearly is equivalent to (3.2).

The axiom of choice (AC) and the Markov principle ($\mathrm{M}^\omega$),

$$(\forall_y A_0 \to B_0) \to \exists_y(A_0 \to B_0) \quad (y \notin \mathrm{FV}(B_0)).$$

(for quantifier-free formulas $A_0, B_0$) can be dealt with easily.

Now consider a purely universal formula $B = \forall_x A_0$, with $A_0$ quantifier-free. Then (modulo cleaning) $\tau^+(B) =$ unit, and moreover $|B|^\varepsilon_y = A_0$. Hence such axioms are interpreted by themselves. The weak extensionality rule can be dealt with in the same way. □

**3.4.4. Dialectica realizers and induction.** For an instance of the induction scheme with an existential formula $\exists_y A_0(n, y)$ one can explicitely construct Dialectica realizers.

$$\exists_y A_0(0, y) \to \forall_n\big(\exists_y A_0(n, y) \to \exists_y A_0(n+1, y)\big) \to \forall_n \exists_y A_0(n, y). \tag{3.3}$$

Its Gödel translation can be calculated as follows.

$$\exists_{y_0} A_0(0, y_0) \to \forall_{n_1}\big(\exists_{y_1} A_0(n_1, y_1) \to \exists_{y_2} A_0(n_1{+}1, y_2)\big) \to \forall_n \exists_y A_0(n, y)$$
$$\forall_{y_0}\big(A_0(0, y_0) \to \forall_{n_1,y_1}\exists_{y_2}\big(A_0(n_1, y_1) \to A_0(n_1{+}1, y_2)\big) \to \forall_n \exists_y A_0(n, y)\big)$$
$$\forall_{y_0}\big(A_0(0, y_0) \to \exists_f \forall_{n_1,y_1}\big(A_0(n_1, y_1) \to A_0(n_1{+}1, f n_1 y_1)\big) \to \forall_n \exists_y A_0(n, y)\big)$$

$$\forall_{y_0,f,n}\big(A_0(0,y_0)\to\forall_{n_1,y_1}\big(A_0(n_1,y_1)\to A_0(n_1{+}1,fn_1y_1)\big)\to\exists_y A_0(n,y)\big)$$
$$\forall_{y_0,f,n}\exists_{y,n_1,y_1}\big(A_0(0,y_0)\to\big(A_0(n_1,y_1)\to A_0(n_1{+}1,fn_1y_1)\big)\to A_0(n,y)\big).$$

Define

$$\begin{aligned}&Y(y_0,f,0):=y_0,\quad Y(y_0,f,n+1):=f(n,Y(y_0,f,n)),\\&N_1(y_0,f,n):=\min\{\,k<n\mid A_0(k,Y_k)\wedge\neg A_0(k+1,Y_{k+1})\,\},\\&Y_1(y_0,f,n):=Y_{N_1(y_0,f,n)}.\end{aligned}$$

with $Y_l:=Y(y_0,f,l)$. We prove that $Y$, $N_1$ and $Y_1$ are the required Dialectica realizers. Assume $A_0(0,Y_0)$ and $A_0(N_1,Y_1)\to A_0(N_1{+}1,fN_1Y_1)$. Then by construction there can be no $k<n$ such that $A_0(k,Y_k)\wedge\neg A_0(k{+}1,Y_{k+1})$, that is to say, $\forall_{k<n}(A_0(k,Y_k)\to A_0(k+1,Y_{k+1}))$. This implies $A(n,Y_n)$.

The explicit construction of $Y$, $N_1$ and $Y_1$ is somewhat complex, mainly because of the bounded minimum needed for $N_1$. For the practical construction of Dialectica realizers it is much easier to employ the induction *rule* rather than the induction axiom. For our present example of an existential formula $\exists_y A_0(n,y)$ this means that we start with closed derivations of

$$\exists_{y_0}A_0(0,y_0)\quad\text{and}\quad\forall_n(\exists_{y_1}A_0(n,y_1))\to\exists_{y_2}A_0(n+1,y_2)).$$

The induction rule then allows to infer $\forall_n\exists_y A_0(n,y)$. Let us see how we can construct a Dialectica realizer for it. The Gödel translation of the step formula is

$$\exists_g\forall_{n,y_1}\big(A_0(n,y_1)\to A_0(n+1,g(n,y_1))\big).$$

Let $r$, $s$ be Dialectica realizers of the base and the step formulas. Then $\mathcal{R}^{\mathrm{nat}}_{\mathrm{nat}}rs$ is a Dialectica realizer of the formula $\forall_n\exists_y A_0(n,y)$ proved by induction. To see this, we need to prove $\forall_n A_0(n,\mathcal{R}^{\mathrm{nat}}_{\mathrm{nat}}rsn)$, which is done easily by induction on $n$.

Now if in an application the proof of a goal $B$ needs an auxiliary inductive proof of $\forall_n\exists_y A_0(n,y)=:A$, then a Dialectica realizer for $B$ is obtained from the (logical) derivation of $A\to B$ as follows. Let $\langle f,g\rangle$ be a Dialectica realizer of $A\to B$, and recall that

$$|A\to B|^{f,g}_{x,u}:=|A|^x_{gxu}\to|B|^{fx}_u.$$

Then $f(\mathcal{R}^{\mathrm{nat}}_{\mathrm{nat}}rs)$ is the required Dialectica realizer of $B$.

**3.4.5. Characterization.** We now consider the question under which conditions the Gödel translation of a formula $A$ is equivalent to the formula itself.

Theorem (Characterization).

$$\mathrm{AC}+\mathrm{IP}^\omega_\forall+\mathrm{M}^\omega\vdash A\leftrightarrow\exists_x\forall_y|A|^x_y.$$

*The direction "$\leftarrow$" can be proved in* $\mathsf{ML}^\omega$ *alone, provided the formulas considered only have* $\exists\forall$*-premises, that is, premises belonging to the class of formulas built from prime formulas by* $\forall$, $\exists$ *and* $\exists_x\forall_y A_0 \to B$, *with* $A_0$ *quantifier-free.*

PROOF. Induction on $A$; we only treat one case.

$$\begin{aligned}
(A \to B) &\leftrightarrow (\exists_x\forall_y |A|^x_y \to \exists_v\forall_u |B|^v_u) && \text{by IH}\\
&\leftrightarrow \forall_x(\forall_y |A|^x_y \to \exists_v\forall_u |B|^v_u) && \text{by } \mathsf{ML}^\omega\\
&\leftrightarrow \forall_x\exists_v(\forall_y |A|^x_y \to \forall_u |B|^v_u) && \text{by } (\mathrm{IP}^\omega_\forall)\\
&\leftrightarrow \forall_x\exists_v\forall_u(\forall_y |A|^x_y \to |B|^v_u) && \text{by } \mathsf{ML}^\omega\\
&\leftrightarrow \forall_x\exists_v\forall_u\exists_y(|A|^x_y \to |B|^v_u) && \text{by } (\mathrm{M}^\omega)\\
&\leftrightarrow \exists_f\forall_x\forall_u\exists_y(|A|^x_y \to |B|^{fx}_u) && \text{by (AC)}\\
&\leftrightarrow \exists_{f,g}\forall_{x,u}(|A|^x_{gxu} \to |B|^{fx}_u) && \text{by (AC)}\\
&\leftrightarrow \exists_{f,g}\forall_{x,u}|A \to B|^{f,g}_{x,u} && \text{by definition.}
\end{aligned}$$

Now assume that $A$ has only $\exists\forall$-premises. First notice that for a formula of the form $\exists_x\forall_y A_0$ with $A_0$ quantifier-free we have $|\exists_x\forall_y A_0|^x_y = A_0$. Therefore, we can replace "$\leftrightarrow$" by "$\leftarrow$" in the argument above. □

As a consequence, we see that $\mathsf{WE\text{-}HA}^\omega + \mathrm{AC} + \mathrm{IP}^\omega_\forall + \mathrm{M}^\omega$ is conservative over $\mathsf{WE\text{-}HA}^\omega$ for formulas with $\exists\forall$-premises. This follows from the Soundness Theorem together with the observation above, as follows. Let $A$ be a formula with $\exists\forall$-premises only, and assume $\mathsf{WE\text{-}HA}^\omega + \mathrm{AC} + \mathrm{IP}^\omega_\forall + \mathrm{M}^\omega \vdash A$. Then by the Soundness Theorem $\mathsf{WE\text{-}HA}^\omega \vdash \exists_x\forall_y |A|^x_y$, and hence $\mathsf{WE\text{-}HA}^\omega \vdash A$.

**3.4.6. A unified treatment of modified realizability and the Dialectica interpretation.** Following Oliva (2006), we show that modified realizability can be treated in such a way that similarities with the Dialectica interpretation become visible. To this end, one needs to change the definitions of $\tau^+(A)$ and $\tau^-(A)$ and also of the Gödel translation $|A|^x_y$ in the implicational case, as follows.

$$\begin{aligned}
&\boldsymbol{\tau}^+(A \to B) := \boldsymbol{\tau}^+(A) \Rightarrow \boldsymbol{\tau}^+(B),\\
&\boldsymbol{\tau}^-(A \to B) := \boldsymbol{\tau}^+(A) \times \boldsymbol{\tau}^-(B),
\end{aligned} \qquad |A \to B|^f_{x,u} := \forall_y |A|^x_y \to |B|^{fx}_u.$$

Then the above definition of mr can be expressed in terms of the (new) $|A|^x_y$:

$$\vdash r \text{ mr } A \leftrightarrow \forall_y |A|^r_y.$$

This is proved by induction on $A$. For prime formulas the claim is obvious. *Case* $A \to B$, with $\boldsymbol{\tau}^+(A) \neq \varepsilon$, $\boldsymbol{\tau}^-(A) \neq \varepsilon$.

$$r \text{ mr } (A \to B) \leftrightarrow \forall_x(x \text{ mr } A \to rx \text{ mr } B) \quad \text{by definition}$$

$$\begin{aligned}
&\leftrightarrow \forall_x(\forall_y |A|^x_y \to \forall_u |B|^{rx}_u) && \text{by IH}\\
&\leftrightarrow \forall_{x,u}(\forall_y |A|^x_y \to |B|^{rx}_u) && \text{by } \mathsf{ML}^\omega\\
&= \forall_{x,u} |A \to B|^r_{x,u} && \text{by definition.}
\end{aligned}$$

The other cases are similar (even easier).

**3.4.7. Extraction.** As a consequence of the Soundness and Characterization Theorems we obtain

THEOREM (Extraction). *Assume*

$$\mathsf{WE\text{-}HA}^\omega + \mathrm{AC} + \mathrm{IP}^\omega_\forall + \mathrm{M}^\omega + \mathrm{Ax}_\forall \vdash \forall_x\big(\forall_u A_0(x,u) \to \exists_y B(x,y)\big)$$

*with $A_0$ quantifier-free, and all formulas have at most the displayed variables free. Then we can find a closed $\mathsf{HA}^\omega$-term $t$ such that*

$$\mathsf{WE\text{-}HA}^\omega + \mathrm{AC} + \mathrm{IP}^\omega_\forall + \mathrm{M}^\omega + \mathrm{Ax}_\forall \vdash \forall_x\big(\forall_u A_0(x,u) \to B(x,tx)\big).$$

PROOF. Let $C(x,y) := \forall_u A_0(x,u) \to \exists_y B(x,y)$, and recall that

$$|\forall_x \exists_y C(x,y)|^{f,g}_{x,b} = |\exists_y C(x,y)|^{fx,gx}_b = |C(x,fx)|^{gx}_b.$$

By the Soundness Theorem we obtain closed terms $t, s$ such that

$$\mathsf{WE\text{-}HA}^\omega + \mathrm{Ax}_\forall \vdash \forall_{x,b} |C(x,tx)|^{sx}_b$$

and hence

$$\mathsf{WE\text{-}HA}^\omega + \mathrm{Ax}_\forall \vdash \forall_x \exists_a \forall_b |C(x,tx)|^a_b.$$

By the Characterization Theorem we have

$$\mathrm{AC} + \mathrm{IP}^\omega_\forall + \mathrm{M}^\omega \vdash C(x,tx) \leftrightarrow \exists_a \forall_b |C(x,tx)|^a_b.$$

Therefore

$$\mathsf{WE\text{-}HA}^\omega + \mathrm{AC} + \mathrm{IP}^\omega_\forall + \mathrm{M}^\omega + \mathrm{Ax}_\forall \vdash \forall_x C(x,tx). \qquad \square$$

## 3.5. Majorization and the Dialectica Interpretation

Generally, the Dialectica interpretation has a strong tendency to produce complex extracted terms, as opposed to the realizability interpretation. This is partially due to contraction (necessary in the $\to^-$-rule). Therefore it is advisable (even more so than for the realizability interpretation) to

- consider derivations from lemmata (whose proofs are not analyzed), and
- try to simplify extracted terms by only aiming at majorants.

This has led Kohlenbach to develop his "monotone Dialectica interpretation", where one only looks for bounds of realizers rather than exact realizers. Again a Soundness Theorem can be proved (Kohlenbach, 1996), and the extraction of uniform bounds (Kohlenbach, 1998) can be achieved.

An essential point observed by Kohlenbach is that when one restricts attention to bounds rather than exact realizers, then one can conveniently deal with additional assumptions $\mathrm{Ax}_{\forall\exists_\leq\forall}$ of the form

$$\forall_{x^\rho}\exists_{y\leq_\sigma rx}\forall_{z^\tau}A_0(x,y,z) \qquad (A_0 \text{ quantifier-free}),$$

with $r$ a closed term of type $\rho\Rightarrow\sigma$. We then need to consider strenghtened versions $\mathrm{Ax}'_{\forall\exists_\leq\forall}$ of these assumptions as well:

$$\exists_{Y\leq_{\rho\Rightarrow\sigma}r}\forall_{x^\rho,z^\tau}A_0(x,Yx,z).$$

Notice that with (AC) one can prove the strenghtened version from the original one.

### 3.5.1. Soundness with majorants.

THEOREM (Soundness with majorants). *Let $M$ be a derivation in*

$$\mathsf{WE\text{-}HA}^\omega+\mathrm{AC}+\mathrm{IP}^\omega_\forall+\mathrm{M}^\omega+\mathrm{Ax}_{\forall\exists_\leq\forall}$$

*of a formula $A$ from assumptions $u_i\colon C_i$ $(i=1,\dots,n)$. Let $x_i$ of type $\tau^+(C_i)$ be variables for realizers of the assumptions, and $y$ of type $\tau^-(A)$ be a variable for a challenge of the goal. Let $\vec z$ of type $\vec\rho$ be the variables free in $M$. Then we can find closed terms $[\![\lambda\vec z,\vec u\,M]\!]^{*+}=:T^*$ of type $\tau^+(C_1)\Rightarrow\ldots\Rightarrow\tau^+(C_n)\Rightarrow\vec\rho\Rightarrow\tau^+(A)$ and $[\![\lambda\vec z,\vec u\,M]\!]^{*-}_i=:R^*_i$ of type $\tau^+(C_1)\Rightarrow\ldots\Rightarrow\tau^+(C_n)\Rightarrow\vec\rho\Rightarrow\tau^-(A)\Rightarrow\tau^-(C_i)$, and a derivation $\mu(M)$ in*

$$\mathsf{WE\text{-}HA}^\omega+\mathrm{Ax}'_{\forall\exists_\leq\forall}$$

*of the formula*

$$\exists_{T,R_1,\dots,R_n}\big(T^*\ \mathrm{maj}\ T\wedge R_1^*\ \mathrm{maj}\ R_1\wedge\cdots\wedge R_n^*\ \mathrm{maj}\ R_n\wedge \forall_{\vec x,\vec z,y}(|C_1|^{x_1}_{R_1\vec x\vec z y}\to\cdots\to|C_n|^{x_n}_{R_n\vec x\vec z y}\to|A|^{T\vec x\vec z}_y)\big).$$

PROOF. Induction on $M$.

*Case* $u\colon A$. Let $x$ of type $\tau^+(A)$ be a variable for a realizer of the assumption $u$. We need $T^*$ and $R^*$ such that

$$\exists_{T,R}\big(T^*\ \mathrm{maj}\ T\wedge R^*\ \mathrm{maj}\ R\wedge\forall_{x,y}(|A|^x_{Rxy}\to|A|^{Tx}_y)\big).$$

We can take $Tx:=x$ and $Rxy:=y$, which both majorize themselves.

*Case* $c\colon A$, $c$ an axiom. Consider an axiom

$$\forall_{x^\rho}\exists_{y\leq_\sigma rx}\forall_{z^\tau}A_0(x,y,z) \qquad (A_0 \text{ quantifier-free}),$$

with $r$ a closed term of type $\rho \Rightarrow \sigma$. We have to find a majorant of some $T$ such that

$$\begin{aligned}
&\forall_{x,z}|\forall_{x^\rho}\exists_{y\le_\sigma rx}\forall_{z^\tau}A_0(x,y,z)|^T_{x,z}\\
&\forall_{x,z}|\exists_{y\le_\sigma rx}\forall_{z^\tau}A_0(x,y,z)|^{Tx}_{z}\\
&\forall_{x,z}(Tx\le rx\wedge|\forall_{z^\tau}A_0(x,Tx,z)|_z)\\
&\forall_{x,z}(Tx\le rx\wedge A_0(x,Tx,z)).
\end{aligned}$$

We now use the corresponding axiom in $\mathrm{Ax}'_{\forall\exists_\le\forall}$:

$$\exists_{Y\le_{\rho\Rightarrow\sigma}r}\forall_{x^\rho,z^\tau}A_0(x,Yx,z).$$

Pick this $Y$ as the desired $T$. Then as a majorant for $Y$ we can take a closed term $r^*$ majorizing $r$.

For the other axioms we have already constructed a Dialectica realizer, and we can take an arbitrary majorant of it. However, we can also directly provide a majorant of some Dialectica realizer.

*Case* $\lambda u^A M^B$. By IH we have a derivation of

$$\begin{aligned}
\exists_{T,R_1,\dots,R_n,R}\big(&T^*\ \mathrm{maj}\ T\wedge R_1^*\ \mathrm{maj}\ R_1\wedge\dots\wedge R_n^*\ \mathrm{maj}\ R_n\wedge R^*\ \mathrm{maj}\ R\wedge\\
&\forall_{x_1,\dots,x_n,x,z}(|C_1|^{x_1}_{R_1x_1\dots x_nxz}\to\dots\to|C_n|^{x_n}_{R_nx_1\dots x_nxz}\to\\
&|A|^{x}_{Rx_1\dots x_nxz}\to|B|^{Tx_1\dots x_nx}_{z})\big).
\end{aligned}$$

We argue informally. Instanciating $x$ with $y0$ and $z$ with $y1$ gives

$$\begin{aligned}
\forall_{x_1,\dots,x_n,y}(&|C_1|^{x_1}_{R_1x_1\dots x_n(y0)(y1)}\to\dots\to|C_n|^{x_n}_{R_nx_1\dots x_n(y0)(y1)}\to\\
&|A|^{y0}_{Rx_1\dots x_n(y0)(y1)}\to|B|^{Tx_1\dots x_n(y0)}_{z}),
\end{aligned}$$

which is

$$\begin{aligned}
\forall_{x_1,\dots,x_n,y}(&|C_1|^{x_1}_{R_1x_1\dots x_n(y0)(y1)}\to\dots\to|C_n|^{x_n}_{R_nx_1\dots x_n(y0)(y1)}\to\\
&|A\to B|^{Tx_1\dots x_n,Rx_1\dots x_n}_{y}.
\end{aligned}$$

So we can define the required $\tilde{T}^*,\tilde{R}_i^*$ by

$$\tilde{T}^*\vec{x}:=\langle T^*\vec{x},R^*\vec{x}\,\rangle,\quad \tilde{R}_i^*\vec{x}y:=R_i^*\vec{x}(y0)(y1).$$

*Case* $M^{A\to B}N^A$. We argue informally. By IH we have

$$\begin{aligned}
|A\to B|^{T\vec{x}_i\vec{x}_k}_{x}=|A|^{x0}_{T\vec{x}_i\vec{x}_k1(x0)(x1)}\to|B|^{T\vec{x}_i\vec{x}_k0(x0)}_{x1}&\ \text{from}\ |C_i|^{x_i}_{P_i\vec{x}_i\vec{x}_kx},\ |C_k|^{x_k}_{P_k\vec{x}_i\vec{x}_kx}\\
|A|^{S\vec{x}_j\vec{x}_k}_{z}\quad&\text{from}\ |C_j|^{x_j}_{Q_j\vec{x}_j\vec{x}_kz},\ |C_k|^{x_k}_{Q_k\vec{x}_j\vec{x}_kz}.
\end{aligned}$$

Instanciating $x$ with $\langle S\vec{x}_j\vec{x}_k,y\rangle$ in the first and $z$ with $T\vec{x}_i\vec{x}_k1(S\vec{x}_j\vec{x}_k)y$ in the second derivation gives

$$|A|^{S\vec{x}_j\vec{x}_k}_{T\vec{x}_i\vec{x}_k1(S\vec{x}_j\vec{x}_k)y}\to|B|^{T\vec{x}_i\vec{x}_k0(S\vec{x}_j\vec{x}_k)}_{y}\quad\text{from}\ |C_i|^{x_i}_{p'_i},\ |C_k|^{x_k}_{p'_k},\ \text{and}$$

$$|A|^{S\vec{x}_j\vec{x}_k}_{T\vec{x}_i\vec{x}_k 1(S\vec{x}_j\vec{x}_k)y} \qquad \text{from } |C_j|^{x_j}_{q'_j},\ |C_k|^{x_k}_{q'_k},$$

with

$$\begin{aligned} p'_i &:= P_i\vec{x}_i\vec{x}_k\langle S\vec{x}_j\vec{x}_k, y\rangle, & p'_k &:= P_k\vec{x}_i\vec{x}_k\langle S\vec{x}_j\vec{x}_k, y\rangle,\\ q'_j &:= Q_j\vec{x}_j\vec{x}_k(T\vec{x}_i\vec{x}_k 1(S\vec{x}_j\vec{x}_k)y), & q'_k &:= Q_k\vec{x}_j\vec{x}_k(T\vec{x}_i\vec{x}_k 1(S\vec{x}_j\vec{x}_k)y). \end{aligned}$$

So we can take

$$\begin{aligned} \tilde{T}^*\vec{x}_i\vec{x}_j\vec{x}_k &:= T^*\vec{x}_i\vec{x}_k 0(S^*\vec{x}_j\vec{x}_k),\\ R^*_i\vec{x}_i\vec{x}_j\vec{x}_k y &:= P^*_i\vec{x}_i\vec{x}_k\langle S^*\vec{x}_j\vec{x}_k, y\rangle,\\ R^*_j\vec{x}_i\vec{x}_j\vec{x}_k y &:= Q^*_j\vec{x}_j\vec{x}_k(T^*\vec{x}_i\vec{x}_k 1(S^*\vec{x}_j\vec{x}_k)y),\\ R^*_k\vec{x}_i\vec{x}_j\vec{x}_k y &:= \max(P^*_k\vec{x}_i\vec{x}_k\langle S^*\vec{x}_j\vec{x}_k, y\rangle, Q^*_k\vec{x}_j\vec{x}_k(T^*\vec{x}_i\vec{x}_k 1(S^*\vec{x}_j\vec{x}_k)y)). \end{aligned}$$

*Case* $\lambda x\, M^{A(x)}$. By IH we have a derivation of $|A(x)|^{Tx_1\dots x_n x}_z$ from $|C_i|^{x_i}_{R_i x_1\dots x_n x z}$. Instanciating $x$ with $y0$ and $z$ with $y1$ gives $|A(y0)|^{Tx_1\dots x_n(y0)}_{y1}$, which is

$$|\forall_x A(x)|^{Tx_1\dots x_n}_y, \quad \text{from } |C_i|^{x_i}_{R_i x_1\dots x_n(y0)(y1)}.$$

So we can take

$$\begin{aligned} \tilde{T}^* x_1\dots x_n &:= T^* x_1\dots x_n,\\ \tilde{R}^*_i x_1\dots x_n y &:= R^*_i x_1\dots x_n(y0)(y1). \end{aligned}$$

*Case* $M^{\forall_x A(x)}s$. By IH we have a derivation of $|\forall_x A(x)|^{Tx_1\dots x_n}_z$, which is $|A(z0)|^{Tx_1\dots x_n(z0)}_{z1}$, from $|C_i|^{x_i}_{R_i x_1\dots x_n z}$. Instanciating $z$ with $\langle s, y\rangle$ gives

$$|A(s)|^{Tx_1\dots x_n s}_y \quad \text{from } |C_i|^{x_i}_{R_i x_1\dots x_n\langle s,y\rangle}.$$

Assume for simplicity that $s$ is closed. Then we can take

$$\begin{aligned} \tilde{T}^* x_1\dots x_n &:= T^* x_1\dots x_n s^*,\\ \tilde{R}^*_i x_1\dots x_n y &:= R^*_i x_1\dots x_n\langle s^*, y\rangle. \end{aligned}$$

□

### 3.5.2. Extraction of uniform bounds.

THEOREM (Extraction of uniform bounds). *Let $s$ be a closed $\mathsf{HA}^\omega$-term, $A(x,y,z)$ a formula with at most the displayed variables free, and $\tau$ a type of level $\leq 2$. Assume that*

$$\mathsf{WE\text{-}HA}^\omega + \mathrm{AC} + \mathrm{IP}^\omega_\forall + \mathrm{M}^\omega + \mathrm{Ax}_{\forall\exists_\leq\forall} \vdash \forall_{x^1}\forall_{y\leq_\rho sx}\exists_{z^\tau} A(x,y,z).$$

*Then we can find a closed $\mathsf{HA}^\omega$-term $t$ such that*

$$\mathsf{WE\text{-}HA}^\omega + \mathrm{AC} + \mathrm{IP}^\omega_\forall + \mathrm{M}^\omega + \mathrm{Ax}_{\forall\exists_\leq\forall} \vdash \forall_{x^1}\forall_{y\leq_\rho sx}\exists_{z\leq_\tau tx} A(x,y,z).$$

*Moreover, if $A$ contains $\forall\exists$-premises only, then the conclusion can already be derived in $\mathsf{WE\text{-}HA}^\omega + \mathrm{Ax}'_{\forall\exists_\leq\forall}$.*

PROOF. Let $P^\omega := \mathsf{WE\text{-}HA}^\omega + \mathrm{AC} + \mathrm{IP}^\omega_\forall + \mathrm{M}^\omega$. Using $\mathrm{IP}^\omega_\forall$ we obtain

$$P^\omega + \mathrm{Ax}_{\forall\exists_\leq\forall} \vdash \forall_{x^1,y}\exists_{z^\tau}(y \leq_\rho sx \to A(x,y,z)).$$

By Soundness with Majorants we have a closed term $T^*$ such that in $P^\omega + \mathrm{Ax}'_{\forall\exists_\leq\forall}$ we can derive the existence of some $T$ with $T^*$ maj $T$ and

$$\begin{aligned}
&\forall_{x^1,y,u}|\exists_{z^\tau}(y \leq_\rho sx \to A(x,y,z))|_u^{Txy}\\
&\forall_{x^1,y,u}|y \leq_\rho sx \to A(x,y,Txy0)|_u^{Txy1}\\
&\forall_{x^1,y}\exists_z\forall_u|y \leq_\rho sx \to A(x,y,Txy0)|_u^z\\
&\forall_{x^1,y}(y \leq_\rho sx \to A(x,y,Txy0));
\end{aligned}$$

in the last step we have used the Characterization Theorem in Sec.3.4.5. Notice that using (AC) we can replace $\mathrm{Ax}'_{\forall\exists_\leq\forall}$ by the original $\mathrm{Ax}_{\forall\exists_\leq\forall}$.

We now argue as in the corresponding theorem in Sec.3.3.5 (for realizability). Let $t_1 := \lambda x\lambda y.Txy0$. Pick majorizing terms $s^*, t_1^*$ for $s, t_1$. Writing $x^M$ for $Mx$ with $M$ from the Majorization Lemma in Sec.3.3.2 we have $s^*x^M$ $\mathrm{maj}_\rho$ $sx$, hence

$$\mathsf{HA}^\omega \vdash \forall_{x^1}\forall_{y\leq_\rho sx}\, s^*x^M \ \mathrm{maj}_\rho\ y.$$

For simplicity assume $\tau = 2 := (\mathrm{nat} \Rightarrow \mathrm{nat}) \Rightarrow \mathrm{nat}$. Then

$$\mathsf{HA}^\omega \vdash \forall_{x^1}\forall_{y\leq_\rho sx}\forall_f\, t_1^*x^M(s^*x^M)f^M \geq_{\mathrm{nat}} t_1xyf.$$

Hence we can take $t := \lambda x\lambda f.t_1^*x^M(s^*x^M)f^M$, for $tx \geq_2 t_1xy =_{\mathrm{def}} Txy0$.

We now show that if $A$ contains $\forall\exists$-premises only, then the conclusion can be derived in $\mathsf{WE\text{-}HA}^\omega + \mathrm{Ax}'_{\forall\exists_\leq\forall}$. To see this, notice that the used direction of the Characterization Theorem in Sec.3.4.5 in this case needs $\mathsf{ML}^\omega$ only, and that (generally) the Soundness Theorem gives derivability in $\mathsf{WE\text{-}HA}^\omega + \mathrm{Ax}'_{\forall\exists_\leq\forall}$. □

**3.5.3. The weak Lemma of König as a $\forall\exists_\leq\forall$-Axiom.** We want to show that the "weak" (that is, binary) Lemma of König WKL can be brought into the form of an axiom in $\mathrm{Ax}_{\forall\exists_\leq\forall}$. WKL says that every infinite binary tree has an infinite path. When we try to directly formalize it in our (functional) language, it does not quite have the required form, since the assumption that the given tree is infinite needs an additional $\forall$ in the premise. However, one can easily find an equivalent statement of the required form. To this end, we define the "infinite extension" of a given tree, and let WKL′ say that for every $t$, the infinite extension $I(\hat{t})$ of its "associated tree" $\hat{t}$ has an infinite path. It then is easy to see that WKL and WKL′ are equivalent.

Let us first introduce some basic definitions. Let nat be the type of natural numbers, boole the type of booleans $\mathsf{tt}$, $\mathsf{ff}$ and list(boole) the type of lists of booleans. It is convenient to write lists in reverse order, that is,

add elements at the end. We fix the types of some variables and state their intended meaning:

| | | |
|---|---|---|
| $a, b, c$ | of type list(boole) | for nodes, |
| $r, s, t$ | of type list(boole) $\Rightarrow$ boole | for decidable sets of nodes, |
| $f, g, h$ | of type nat $\Rightarrow$ boole | for paths, |
| $n, m, k, i, j$ | of type nat | for natural numbers, |
| $p, q$ | of type boole | for booleans. |

Let $\mathrm{lh}(a)$ be the *length* of $a$. Let $\bar{a}(n)$ denote the initial segment of $a$ of length $n$, if $n \leq \mathrm{lh}(a)$, and $a$ otherwise. Similarly let $\bar{f}(n)$ denote the initial segment of $f$ of length $n$, that is, the list $:f(0) :: f(1) \cdots :: f(n-1)$. Let $(a)_n$ denote the $n$-th element of $a$, if $n < \mathrm{lh}(a)$, and $\mathtt{tt}$ otherwise. $f$ is a *path in* $t$ if all its initial segments $\bar{f}(n)$ are in $t$. Call $t$ *infinite* if for every $n$ there is a node of length $n$ in $t$. Call $t$ a *tree* if it is downwards closed, i.e., $\forall_a \forall_{n \leq \mathrm{lh}(a)} (a \in t \to \bar{a}(n) \in t)$. So WKL says that

$$\begin{aligned} &\forall_t \big( \forall_a \forall_{n \leq \mathrm{lh}(a)} (a \in t \to \bar{a}(n) \in t) \to && (t \text{ is a tree}) \\ &\quad \forall_n \exists_{a \in t} \mathrm{lh}(a) = n \to && (t \text{ is infinite}) \\ &\quad \exists_f \forall_n \bar{f}(n) \in t \big) && (t \text{ has an infinite path}), \end{aligned}$$

which – because of the two premises saying that $t$ is an infinite tree – is not of the required logical form.

To obtain an equivalent formulation in the required form, we introduce some further notions.

$$\begin{aligned} &\hat{t} := \{\, a \mid \forall_{n < \mathrm{lh}(a)} \bar{a}(n) \in t \,\} && \text{the associated tree } \hat{t} \text{ for } t, \\ &b = a * \mathtt{tt}^{\mathrm{lh}(b) - \mathrm{lh}(a)} && b \text{ is the } \mathtt{tt}\text{-extension of } a, \\ &\forall_{c; \mathrm{lh}(c) = \mathrm{lh}(b)} c \notin \hat{t} && b \text{ is } t\text{-big}. \end{aligned}$$

Let $\min_{\mathrm{lex}}$ denote the minimum of a set of nodes w.r.t. the lexicographical ordering, and $\mathrm{maxlen}_{<n}(t)$ be the maximal length of all nodes of $t$ of length $< n$. Then $\mathrm{ll}_n(t)$ is the leftmost largest node in $t$ of length $< n$:

$$\begin{aligned} \mathrm{maxlen}_{<n}(t) &:= \max\{\, \mathrm{lh}(a) \mid a \in t \wedge \mathrm{lh}(a) < n \,\}, \\ \mathrm{ll}_n(t) &:= \min_{\mathrm{lex}}\{\, c \in t \mid \mathrm{lh}(c) = \mathrm{maxlen}_{<n}(t) \,\}. \end{aligned}$$

We can now define the infinite extension $I(t)$ of a tree $t$:

$$I(t) := \{\, b \mid b \in t \vee (b \text{ is } t\text{-big} \wedge b \text{ is the } \mathtt{tt}\text{-extension of } \mathrm{ll}_{\mathrm{lh}(b)}) \,\}.$$

All these notions are definable in $\mathsf{HA}^\omega$. They clearly have the following properties:

$\hat{t}$ is a tree;

if $t$ is a tree, then $\hat{t} = t$;

if $t$ is a tree, then $I(t)$ is an infinite tree extending $t$;

if $t$ is an infinite tree, then $I(t) = t$.

Then WKL is equivalent (provably in $\mathsf{HA}^\omega$) to

$$\mathrm{WKL}' := \forall_t \exists_f \forall_n \bar{f}(n) \in I(\hat{t}).$$

To see this, assume WKL, and let $t$ be arbitrary. Then $I(\hat{t})$ is an infinite tree extending $t$. By WKL applied to $I(\hat{t})$, $\exists_f \forall_n \bar{f}(n) \in I(\hat{t})$. Conversely, let $t$ be an infinite tree. Then $I(\hat{t}) = t$ and therefore $\exists_f \forall_n \bar{f}(n) \in t$.

Remark. From the results of Ishihara (1990) it is known WKL implies Brouwer's fan theorem. Moreover, a direct proof of this implication has been given by Ishihara in 2002 and published in (2006). Berger and Ishihara (2005) have shown that a weakened form WKL! of WKL, where as an additional hypothesis it is required that in an effective sense infinite paths are unique, is equivalent to Fan. One direction (WKL! implies Fan) is essentially the proof of Ishihara (2006), enhanced by the additional requirement that the tree extension to be constructed satisfies the effective uniqueness condition (as in Berger and Ishihara (2005)). The main tool of this proof is the construction of $I(\hat{t})$ described above. The other direction (Fan implies WKL!) is far less directly proved by Berger and Ishihara (2005), where the emphasis rather was to provide a fair number of equivalents to Fan, and to do the proof economically by giving a circle of implications. A direct proof of the equivalence of Fan with WKL! is in (Schwichtenberg, 2005). The latter paper also reports on a formalization in the Minlog proof assistant, and gives rather short and perspicious realizing terms (w.r.t. modified realizability) machine-extracted from each of the two directions of this proof.

**3.5.4. Application: uniform moduli of continuity.** As an application, we show that every functional of type 2 definable in $\mathsf{HA}^\omega$ has an (even uniform) modulus of continuity definable in $\mathsf{HA}^\omega$ as well, when applied to arguments $\leq_1 y$.

Consider *hereditary extensional equality*, defined as follows (cf. Troelstra (1973)):

$$\begin{aligned}(x_1 \approx_\mu x_2) &:= (x_1 =_\mu x_2),\\ (x_1 \approx_{\rho\Rightarrow\sigma} x_2) &:= \forall_{y_1,y_2}(y_1 \approx_\rho y_2 \to x_1 y_1 \approx_\sigma x_2 y_2).\end{aligned}$$

Hereditary extensional equality is compatible with pointwise equality (as defined in Sec.3.1.3):

Lemma. $\vdash x_1 =_\rho x_1' \to x_2 =_\rho x_2' \to x_1 \approx_\rho x_2 \to x_1' \approx_\rho x_2'$.

Proof. Induction on $\rho$. In the case $\rho \Rightarrow \sigma$ we can assume $y_1 \approx_\rho y_2$ and have to show $x_1' y_1 \approx_\sigma x_2' y_2$. From the assumption $x_1 \approx_{\rho\Rightarrow\sigma} x_2$ we obtain

$x_1y_1 \approx_\sigma x_2y_2$. Using $x_1y_1 =_\sigma x_1'y_1$ and $x_2y_2 =_\sigma x_2'y_2$ the IH gives us the claim. □

REMARK. By definition, $x_1 \approx_1 x_2$ is the same as $x_1 =_1 x_2$.

LEMMA. *For every* $\mathsf{HA}^\omega$*-term* $t$*,*

$$\mathsf{HA}^\omega \vdash \vec{x}_1 \approx \vec{x}_2 \to t(\vec{x}_1) \approx_\rho t(\vec{x}_2).$$

PROOF. Induction on $t$. *Case* $\lambda y\, t$. Assume $\vec{x}_1 \approx \vec{x}_2$ and $y_1 \approx_{\rho\Rightarrow\sigma} y_2$. Then $t(y_1, \vec{x}_1) \approx_\sigma t(y_2, \vec{x}_2)$ by IH, hence

$$(\lambda y\, t(y, \vec{x}_1))y_1 =_{\mathrm{def}} t(y_1, \vec{x}_1) \approx_\sigma t(y_2, \vec{x}_2) =_{\mathrm{def}} (\lambda y\, t(y, \vec{x}_2))y_2.$$

*Case* $\mathcal{R}$. Assume $f_1, g_1 \approx f_2, g_2$. We must show $\mathcal{R}f_1g_1n \approx \mathcal{R}f_2g_2n$, which can be done easily by induction on $n$. □

COROLLARY. *For every closed* $\mathsf{HA}^\omega$*-term* $t$*,* $\mathsf{HA}^\omega \vdash t \approx_\rho t$.

THEOREM. *For every closed* $\mathsf{HA}^\omega$*-term* $t$ *of type* 2*, we can find another closed* $\mathsf{HA}^\omega$*-term* $\bar{t}$ *of* $\mathsf{HA}^\omega$ *also of type* 2 *such that*

$$\mathsf{HA}^\omega \vdash \forall_{k,y}\forall_{x,x'\le_1 y}\big(\forall_{i<\bar{t}ky} xi = x'i \to \forall_{j<k} txj = tx'j\big).$$

PROOF. Because of the remark above, from $\mathsf{HA}^\omega \vdash t \approx_2 t$ we obtain

$$\begin{aligned}
&\mathsf{HA}^\omega && \vdash \forall_{x,x'}(\forall_i xi = x'i \to \forall_k\forall_{j<k} txj = tx'j),\\
&\mathsf{HA}^\omega + \mathrm{M}^\omega && \vdash \forall_{k,x,x'}\exists_i(xi = x'i \to \forall_{j<k} txj = tx'j),\\
&\mathsf{HA}^\omega + \mathrm{M}^\omega && \vdash \forall_{k,y}\forall_{x,x'\le_1 y}\exists_i(xi = x'i \to \forall_{j<k} txj = tx'j).
\end{aligned}$$

Now Sec.3.5.2 on extraction of uniform bounds gives us a closed $\mathsf{HA}^\omega$-term $\bar{t}$ such that

$$\mathsf{HA}^\omega \vdash \forall_{k,y}\forall_{x,x'\le_1 y}\exists_{i\le\bar{t}ky}(xi = x'i \to \forall_{j<k} txj = tx'j).$$ □

## 3.6. The Negative Fragment: Classical Arithmetic

When we adopt the point of view of classical logic, we understand an existential formula "there is an $x$ such that $A(x)$" as an abbreviation for "it is not true that for all $x$, $A(x)$ is false". We propose to make this distinction explicit and use both $\exists_x A$ and $\tilde{\exists}_x A$, where the latter is an abbreviation for $\neg\forall_x\neg A$. Then in a classical context we only deal with $\tilde{\exists}_x A$, and hence need to work with $\to\forall\wedge\bot$-formulas only.

Recall that in arithmetic every atomic formula has the form $\mathrm{atom}(r^{\mathrm{boole}})$, i.e., is built from a boolean term $r^{\mathrm{boole}}$. In particular, there is no need for (logical) falsity $\bot$, since we can take the atomic formula $F := \mathrm{atom}(\mathsf{ff})$ – called *arithmetical falsity* – built from the boolean constant $\mathsf{ff}$ instead. We then view negation $\neg A$ as defined by $A \to F$, and consider the arithmetical classical existential quantifier $\tilde{\exists}_x A$.

In particular, stability $\neg\neg A \to A$ holds for atomic formulas, and therefore every atomic formula is equivalent to a negated formula. Hence it suffices in classical arithmetic $\mathsf{PA}^\omega$ to work with $\to\forall\wedge$-formulas only. By what we have seen in Sec.1.2.4 and Sec.1.2.5, this implies that we have stability for all formulas.

REMARK. Notice that by the elimination of $\wedge$ in Sec.1.2.4, we can even omit conjunction $\wedge$.

**3.6.1.** IP, M **and** AC **with classical existence.** We now study what happens to the Independence of Premise axiom ($\mathrm{IP}^\omega$) and Markov's Principle ($\mathrm{M}^\omega$) – both of which involve $\exists$ – under the "negative interpretation" of the existential quantifier, that is, replacement of $\exists$ by $\tilde{\exists}$. It turns out that both become derivable.

LEMMA. (a) ($\tilde{\mathrm{IP}}^\omega$) *is derivable from* $F \to A$*:*

$$\vdash (F \to A) \to (A \to \tilde{\exists}_{x^\rho} B) \to \tilde{\exists}_{x^\rho}(A \to B) \quad (x \notin \mathrm{FV}(A)),$$

*and hence need not be assumed in* $\mathsf{PA}^\omega$*.*

(b) ($\tilde{\mathrm{M}}^\omega$) *is derivable from* $\forall_{x^\rho}(\neg\neg A \to A)$*:*

$$\vdash \forall_{x^\rho}(\neg\neg A \to A) \to (\forall_{x^\rho} A \to B) \to \tilde{\exists}_{x^\rho}(A \to B) \quad (x \notin \mathrm{FV}(B)),$$

*and hence need not be assumed in* $\mathsf{PA}^\omega$*.*

PROOF. Exercise. In fact, these proofs can easily be found by automated proof search in minimal logic. □

However, for the axiom of choice the situation is different. We show that the translation (QF-$\tilde{\mathrm{AC}}$) of the quantifier-free axiom of choice is derivable from the original (QF-AC) plus Markov's Principle ($\mathrm{M}^\omega$) for quantifier-free formulas.

LEMMA. (QF-$\tilde{\mathrm{AC}}$) *is derivable from* (QF-AC) *plus Markov's Principle* ($\mathrm{M}^\omega$) *for quantifier-free formulas.*

PROOF. We argue informally. Assume (QF-AC)

$$\forall_{x^\rho}\exists_{y^\sigma} A_0(x,y) \to \exists_{f^{\rho\Rightarrow\sigma}}\forall_{x^\rho} A_0(x, f(x))$$

with $A_0$ quantifier-free. Then

$$\begin{array}{ll}
\forall_x \tilde{\exists}_y A_0(x,y) & \\
\forall_x(\forall_y \neg A_0(x,y) \to F) & \\
\forall_x \exists_y(\neg A_0(x,y) \to F) & \text{by } (\mathrm{M}^\omega) \\
\forall_x \exists_y A_0(x,y) & \text{by stability } \neg\neg A_0 \to A_0 \\
\exists_f \forall_x A_0(x, f(x)) & \text{by (QF-AC)}
\end{array}$$

$$\tilde{\exists}_f \forall_x A_0(x, f(x)),$$

where the last step is a logical weakening. □

### 3.6.2. Extraction from classical proofs.

THEOREM. *Assume*

$$\mathsf{WE\text{-}PA}^\omega + \text{QF-}\tilde{\text{AC}} + \text{Ax}_\forall \vdash \forall_x \tilde{\exists}_y A_0(x, y),$$

*$A_0(x, y)$ a quantifier-free formula with at most the displayed variables free. Then we can find a closed $\mathsf{HA}^\omega$-term $t$ such that*

$$\mathsf{WE\text{-}HA}^\omega + \text{Ax}_\forall \vdash \forall_x A_0(x, tx).$$

PROOF. We make use of the fact proved in Sec.3.6.1 that (QF-$\tilde{\text{AC}}$) is derivable from (QF-AC) plus Markov's Principle ($\text{M}^\omega$) for quantifier-free formulas. Hence

$$\begin{aligned}
&\mathsf{WE\text{-}PA}^\omega + \text{QF-}\tilde{\text{AC}} + \text{Ax}_\forall \vdash \forall_x \tilde{\exists}_y A_0(x, y)\\
&\mathsf{WE\text{-}HA}^\omega + \text{QF-AC} + \text{M}^\omega + \text{Ax}_\forall \vdash \forall_x \tilde{\exists}_y A_0(x, y)\\
&\mathsf{WE\text{-}HA}^\omega + \text{QF-AC} + \text{M}^\omega + \text{Ax}_\forall \vdash \forall_x \exists_y A_0(x, y) \quad \text{by } (\text{M}^\omega)\\
&\mathsf{WE\text{-}HA}^\omega + \text{Ax}_\forall \vdash |\forall_x \exists_y A_0(x, y)|_x^t
\end{aligned}$$

for some closed $\mathsf{HA}^\omega$-term $t$, where in the last step we have used the Soundness Theorem. But

$$|\forall_x \exists_y A_0(x, y)|_x^t = |\exists_y A_0(x, y)|_\varepsilon^{tx} = |A_0(x, tx)|_\varepsilon^\varepsilon = A_0(x, tx). \qquad \square$$

**3.6.3. Extraction of uniform bounds from classical proofs.** As in Sec.3.5, the restriction to only look for bounds rather than exact realizers makes it possible to deal with additional assumptions $\text{Ax}_{\forall\tilde{\exists}_\leq\forall}$ of the form

$$\forall_{x^\rho} \tilde{\exists}_{y \leq_\sigma rx} \forall_{z^\tau} A_0(x, y, z) \qquad (A_0 \text{ quantifier-free}),$$

with $r$ a closed term of type $\rho \Rightarrow \sigma$. We then need to consider strenghtened versions $\text{Ax}'_{\exists_\leq\forall}$ of these assumptions as well:

$$\exists_{Y \leq_{\rho\Rightarrow\sigma} r} \forall_{x^\rho, z^\tau} A_0(x, Yx, z).$$

THEOREM (Extraction of uniform bounds from classical proofs). *Let $s$ be a closed $\mathsf{HA}^\omega$-term, $A_0(x, y, z)$ a quantifier-free formula with at most the displayed variables free, and $\tau$ a type of level $\leq 2$. Assume that*

$$\mathsf{WE\text{-}PA}^\omega + \text{QF-}\tilde{\text{AC}} + \text{Ax}_{\forall\tilde{\exists}_\leq\forall} \vdash \forall_{x^1} \forall_{y \leq_\rho sx} \tilde{\exists}_{z^\tau} A_0(x, y, z).$$

*Then we can find a closed $\mathsf{HA}^\omega$-term $t$ such that*

$$\mathsf{WE\text{-}HA}^\omega + \text{Ax}'_{\exists_\leq\forall} \vdash \forall_{x^1} \forall_{y \leq_\rho sx} \exists_{z \leq_\tau tx} A_0(x, y, z).$$

PROOF. Assume

$$\mathsf{WE\text{-}HA}^\omega + \mathrm{QF\text{-}}\tilde{\mathrm{AC}} + \mathrm{Ax}_{\forall\tilde{\exists}_{\leq}\forall} \vdash \forall_{x^1}\forall_{y\leq_\rho sx}\tilde{\exists}_{z^\tau}\, A_0(x,y,z).$$

Then clearly

$$\mathsf{WE\text{-}HA}^\omega + \mathrm{QF\text{-}}\tilde{\mathrm{AC}} + \mathrm{Ax}_{\forall\tilde{\exists}_{\leq}\forall} \vdash \forall_{x^1}\forall_y\tilde{\exists}_{\vec{u},z^\tau}(y\vec{u} \leq_0 sx\vec{u} \to A_0(x,y,z)).$$

We again make use of the fact proved in Sec.3.6.1 that (QF-$\tilde{\mathrm{AC}}$) is derivable from (QF-AC) plus Markov's Principle ($\mathrm{M}^\omega$) for quantifier-free formulas. Hence

$$\mathsf{WE\text{-}HA}^\omega + \mathrm{QF\text{-}AC} + \mathrm{M}^\omega + \mathrm{Ax}_{\forall\exists_{\leq}\forall} \vdash \forall_{x^1}\forall_y\tilde{\exists}_{\vec{u},z^\tau}(y\vec{u} \leq_0 sx\vec{u} \to A_0(x,y,z)),$$

and because of ($\mathrm{M}^\omega$) we can replace the $\tilde{\exists}$ on the right hand side by $\exists$. Hence

$$\mathsf{WE\text{-}HA}^\omega + \mathrm{QF\text{-}AC} + \mathrm{M}^\omega + \mathrm{Ax}_{\forall\exists_{\leq}\forall} \vdash \forall_{x^1}\forall_{y\leq_\rho sx}\exists_{z^\tau} A_0(x,y,z).$$

The theorem on extraction of uniform bounds in Sec.3.5.2 gives us a closed $\mathsf{HA}^\omega$-term $t$ such that

$$\mathsf{WE\text{-}HA}^\omega + \mathrm{Ax}'_{\exists_{\leq}\forall} \vdash \forall_{x^1}\forall_{y\leq_\rho sx}\exists_{z\leq_\tau tx}\, A(x,y,z). \qquad \square$$

We now derive a corollary to this theorem, which involves the so-called $\varepsilon$-weakening of the Skolem normal form of a formula

$$\forall_{a^\delta}\exists_{b\leq_\sigma ra}\forall_{c^\gamma} B_0(a,b,c),$$

namely the formula

$$\forall_c\exists_{B\leq_{\delta\Rightarrow\sigma} r}\forall_{a^\delta}\forall_{c'\leq_\gamma c} B_0(a,Ba,c').$$

COROLLARY. *Let $s$ be a closed $\mathsf{HA}^\omega$-term, $A_0(x,y,z)$ and $B_0(a,b,c)$ quantifier-free formulas with at most the displayed variables free, and $\tau,\gamma$ types of level $\leq 2$. Assume that*

$$\mathsf{WE\text{-}PA}^\omega + \mathrm{QF\text{-}}\tilde{\mathrm{AC}} \vdash \forall_{a^\delta}\tilde{\exists}_{b\leq_\sigma ra}\forall_{c^\gamma} B_0(a,b,c) \to \forall_{x^1}\forall_{y\leq_\rho sx}\tilde{\exists}_{z^\tau}\, A_0(x,y,z).$$

*Then we can find a closed $\mathsf{HA}^\omega$-term $t$ such that*

$$\mathsf{WE\text{-}HA}^\omega \vdash \forall_c\exists_{B\leq_{\delta\Rightarrow\sigma} r}\forall_{a^\delta}\forall_{c'\leq_\gamma c} B_0(a,Ba,c') \to \forall_{x^1}\forall_{y\leq_\rho sx}\exists_{z\leq_\tau tx}\, A_0(x,y,z).$$

PROOF. In $\mathsf{WE\text{-}PA}^\omega + \mathrm{QF\text{-}}\tilde{\mathrm{AC}}$ we have

$$\begin{aligned}
&\forall_a\tilde{\exists}_{b\leq ra}\forall_{c^\gamma} B_0(a,b,c) \to \forall_{x^1}\forall_{y\leq sx}\tilde{\exists}_{z^\tau}\, A_0(x,y,z)\\
&\tilde{\exists}_{B\leq r}\forall_a\forall_{c^\gamma} B_0(a,Ba,c) \to \forall_{x^1}\forall_{y\leq sx}\tilde{\exists}_{z^\tau}\, A_0(x,y,z)\\
&\forall_{x^1}\forall_{y\leq sx}\forall_{B\leq r}\tilde{\exists}_{a,c^\gamma,z^\tau}(B_0(a,Ba,c) \to A_0(x,y,z)).
\end{aligned}$$

The theorem above on extraction of uniform bounds (with $\mathrm{Ax}_{\forall\exists_{\leq}\forall}$ empty) provides us with closed terms $t,t'$ such that $\mathsf{WE\text{-}HA}^\omega$ derives

$$\forall_{x^1}\forall_{y\leq sx}\forall_{B\leq r}\exists_a\exists_{c^\gamma\leq t'x}\exists_{z^\tau\leq tx}(B_0(a,Ba,c) \to A_0(x,y,z))$$

$$\forall_{x^1}\big(\exists_{B\le r}\forall_a\forall_{c^\gamma\le t'x}B_0(a,Ba,c)\to\forall_{y\le sx}\exists_{z^\tau\le tx}A_0(x,y,z)\big)$$
$$\forall_c\exists_{B\le r}\forall_a\forall_{c'\le_\gamma c}B_0(a,Ba,c')\to\forall_{x^1}\forall_{y\le sx}\exists_{z\le_\tau tx}A_0(x,y,z),$$

where both steps are logical weakenings. □

One good reason to be interested in such results is that the $\varepsilon$-weakening of the Skolem normal form of the $\forall\exists\forall$-form $\mathrm{WKL}'$ of WKL is derivable: Recall

$$\mathrm{WKL}' := \forall_t\exists_{f^{\mathrm{nat}\Rightarrow\mathrm{boole}}}\forall_n\bar{f}(n)\in I(\hat{t}).$$

The $\varepsilon$-weakening of its Skolem normal form is

$$\forall_n\exists_F\forall_t\forall_{n'\le n}\overline{Ft}(n')\in I(\hat{t}).$$

But this is easy to derive (in $\mathsf{HA}^\omega$): Given $n$, let $F_nt$ pick from the infinite tree $I(\hat{t})$ a path of length $n$.

**3.6.4. Elimination of extensionality.** Define

$$\begin{aligned}
&E_\mu x && := \mathtt{tt},\\
&(x_1 =^e_\mu x_2) && := (x_1 =_\mu x_2),\\
&E_{\rho\Rightarrow\sigma}x && := \forall_{y_1,y_2}(y_1 =^e_\rho y_2\to xy_1 =^e_\sigma xy_2),\\
&(x_1 =^e_{\rho\Rightarrow\sigma} x_2) && := E_{\rho\Rightarrow\sigma}x_1\wedge E_{\rho\Rightarrow\sigma}x_2\wedge\forall_y(E_\rho y\to x_1y =^e_\sigma x_2y).
\end{aligned}$$

Notice that for a type $\rho$ of level $\le 1$ we always have $E_\rho x$; this follows from the compatibility axioms. This implies that $x_1 =^e_\rho x_2$ is the same as pointwise equality $x_1 =_\rho x_2$.

We now collect some general properties of these notions; they are all derivable in $\mathsf{ML}^\omega$. Clearly $=^e$ is symmetric and transitive. Moreover we have

Lemma. (a) *(Reflexivity of $=^e$ on $E$).* $E_\rho x\to x =^e x$.
(b) *(Closure of $E$ under application).* $E_{\rho\Rightarrow\sigma}x\to E_\rho y\to E_\sigma(xy)$.
(c) *(Compatibility of application with $=^e$).*
$x_1 =^e_{\rho\Rightarrow\sigma} x_2\leftrightarrow\forall_{y_1,y_2}(y_1 =^e_\rho y_2\to x_1y_1 =^e_\sigma x_2y_2)$.

Proof. (a) Induction on $\rho$. *Base.* By definition. *Step.* Assume $E_{\rho\Rightarrow\sigma}x$:

$$\forall_{y_1,y_2}(y_1 =^e_\rho y_2\to xy_1 =^e_\sigma xy_2).$$

We must show $x =^e_{\rho\Rightarrow\sigma} x$. $E_{\rho\Rightarrow\sigma}x$ is already given; it remains to show $\forall_y(E_\rho y\to xy =^e_\sigma xy)$. Let $y$ be given and assume $E_\rho y$. We must show $xy =^e_\sigma xy$. The $\mathrm{IH}_\rho$ gives $y =^e_\rho y$. But then $E_{\rho\Rightarrow\sigma}x$ implies the claim.
(b) From $E_\rho y$ we obtain $y =^e y$ by (a). Then $E_{\rho\Rightarrow\sigma}x$ gives $xy =^e xy$. Hence $E_\sigma(xy)$ by definition.

(c) $\to$. Assume $x_1 =^e_{\rho\Rightarrow\sigma} x_2$. Then by definition $E_{\rho\Rightarrow\sigma}x_i$ and

$$\forall_y(E_\rho y \to x_1 y =^e_\sigma x_2 y).$$

Assume further $y_1 =^e_\rho y_2$. We must show $x_1y_1 =^e_\sigma x_2y_2$. From $y_1 =^e_\rho y_2$ we get $E_\rho y_1$; hence $x_1y_1 =^e_\sigma x_2y_1$. But from $E_{\rho\Rightarrow\sigma}x_2$ and $y_1 =^e_\rho y_2$ we obtain $x_2y_1 =^e_\sigma x_2y_2$. Now transitivity of $=^e$ gives the claim.

$\leftarrow$. Assume $\forall_{y_1,y_2}(y_1 =^e_\rho y_2 \to x_1y_1 =^e_\sigma x_2y_2)$. We must show $x_1 =^e_{\rho\Rightarrow\sigma} x_2$. We first show $E_{\rho\Rightarrow\sigma}x_1$; for $x_2$ the argument is similar. So we must prove $\forall_{y_1,y_2}(y_1 =^e_\rho y_2 \to x_1y_1 =^e_\sigma x_1y_2)$. Let $y_1, y_2$ with $y_1 =^e_\rho y_2$ be given. Then $E_\rho y_2$ by definition, hence $y_2 =^e_\rho y_2$ by (a). Now the assumption with $y_1 =^e_\rho y_2$ gives $x_1y_1 =^e_\rho x_2y_2$, and with $y_2 =^e_\rho y_2$ gives $x_1y_2 =^e_\rho x_2y_2$, Transitivity and symmetry of $=^e$ now implies the claim $x_1y_1 =^e_\sigma x_1y_2$. We finally show $\forall_y(E_\rho y \to x_1 y =^e_\sigma x_2 y)$. Let $y$ with $E_\rho y$ be given. Then $y =^e_\rho y$ by (a) and hence $x_1y =^e_\sigma x_2y$ by our assumption. □

We show that all closed $\mathsf{HA}^\omega$-terms are in $E$. This follows from

LEMMA (Compatibility of $\mathsf{HA}^\omega$-terms with $=^e$). *Let a term $r(\vec{x})$ be given, with at most the displayed variables free. Then*

$$\vec{x}_1 =^e \vec{x}_2 \to r(\vec{x}_1) =^e r(\vec{x}_2).$$

PROOF. Induction on $r$. *Case $rs$.* By IH $r(\vec{x}_1) =^e r(\vec{x}_2)$ and $s(\vec{x}_1) =^e s(\vec{x}_2)$. Part (c) of the lemma above gives $r(\vec{x}_1)s(\vec{x}_1) =^e r(\vec{x}_2)s(\vec{x}_2)$.

*Case $\lambda xr$.* Assume $\vec{x}_1 =^e \vec{x}_2$. We must show $\lambda xr(x, \vec{x}_1) =^e \lambda xr(x, \vec{x}_2)$. By part (c) of the lemma above it suffices to show $\forall_{x_1,x_2}(x_1 =^e x_2 \to r(x_1, \vec{x}_1) =^e r(x_2, \vec{x}_2))$. So let $x_1, x_2$ with $x_1 =^e x_2$ be given. The claim then follows by the IH for $r$. □

Let $A^E$ be obtained from $A$ by relativizing all quantifiers to $E$.

LEMMA (Relativization of $E$ and $=^e$ to $E$).

$$(E_\rho x)^E \leftrightarrow E_\rho x \text{ and } (x_1 =^e_\rho x_2)^E \leftrightarrow (x_1 =^e_\rho x_2).$$

PROOF. We prove both claims simultaneously, by induction on $\rho$. *Base.* By definition. *Step.*

$$\begin{aligned}
&(E_{\rho\Rightarrow\sigma}x)^E \\
&\leftrightarrow \forall_{y_1,y_2}(E_\rho y_1 \to E_\rho y_2 \to (y_1 =^e_\rho y_2)^E \to (xy_1 =^e_\sigma xy_2)^E) \\
&\leftrightarrow \forall_{y_1,y_2}(E_\rho y_1 \to E_\rho y_2 \to y_1 =^e_\rho y_2 \to xy_1 =^e_\sigma xy_2) && \text{by IH}_{\rho,\sigma} \\
&\leftrightarrow \forall_{y_1,y_2}(y_1 =^e_\rho y_2 \to xy_1 =^e_\sigma xy_2) && \text{by definition} \\
&= E_{\rho\Rightarrow\sigma}x.
\end{aligned}$$

and similarly

$$\begin{aligned}
&(x_1 =^e_{\rho\Rightarrow\sigma} x_2)^E \\
&\leftrightarrow (Ex_1)^E \wedge (Ex_2)^E \wedge \forall_y(E_\rho y \to (E_\rho y)^E \to (x_1 y =^e_\sigma x_2 y)^E) \\
&\leftrightarrow Ex_1 \wedge Ex_2 \wedge \forall_y(E_\rho y \to x_1 y =^e_\sigma x_2 y) \\
&= (x_1 =^e_{\rho\Rightarrow\sigma} x_2).
\end{aligned}$$

Here we have used the argument above and the $\mathrm{IH}_{\rho,\sigma}$. □

We now show that relativizing pointwise equality to $E$ is the same as extensional equality, provided the objects are in $E$.

LEMMA. $E_\rho x_1 \wedge E_\rho x_2 \wedge (x_1 =_\rho x_2)^E \leftrightarrow x_1 =^e_\rho x_2$.

PROOF. Induction on $\rho$. *Base.* By definition. *Step.*

$$\begin{aligned}
&E_{\rho\Rightarrow\sigma} x_1 \wedge E_{\rho\Rightarrow\sigma} x_2 \wedge (x_1 =_{\rho\Rightarrow\sigma} x_2)^E \\
&\leftrightarrow E_{\rho\Rightarrow\sigma} x_1 \wedge E_{\rho\Rightarrow\sigma} x_2 \wedge \forall_y(E_\rho y \to (x_1 y =_\sigma x_2 y)^E) \\
&\leftrightarrow E_{\rho\Rightarrow\sigma} x_1 \wedge E_{\rho\Rightarrow\sigma} x_2 \wedge \forall_y\big(E_\rho y \to E_\sigma(x_1 y) \wedge E_\sigma(x_1 y) \wedge (x_1 y =_\sigma x_2 y)^E\big) \\
&\leftrightarrow E_{\rho\Rightarrow\sigma} x_1 \wedge E_{\rho\Rightarrow\sigma} x_2 \wedge \forall_y(E_\rho y \to x_1 y =^e_\sigma x_2 y) \\
&= (x_1 =^e_{\rho\Rightarrow\sigma} x_2),
\end{aligned}$$

where the next to last step uses the IH. □

THEOREM. $\mathsf{E\text{-}HA}^\omega \vdash A(\vec{x})$ *implies* $\mathsf{HA}^\omega \vdash E(\vec{x}) \to A^E(\vec{x})$.

PROOF. Induction on derivations. For the extensionality axiom use the lemma above, on relativizing pointwise equality to $E$. □

### 3.6.5. Extraction of uniform bounds from classical proofs with extensionality.

THEOREM. *Let $\Delta$ be a set of axioms from $\mathrm{Ax}_{\forall\exists_{\leq}\forall}$, and $\Delta_\varepsilon$ consist of their $\varepsilon$-weakenings. Assume that the types of the existential variables are all $\leq 1$ and of the final $\forall$-variables are $\leq 2$. Let $s$ be a closed $\mathsf{HA}^\omega$-term, $A_0(x,y,z)$ a quantifier-free formula with at most the displayed variables free, and $\tau$ a type of level $\leq 2$. Assume that*

$$\mathsf{E\text{-}PA}^\omega + \mathrm{QF\text{-}\tilde{AC}}^{0,1} + \tilde{\Delta} + \mathrm{W\tilde{K}L} \vdash \forall_{x^1}\forall_{y\leq_1 sx}\tilde{\exists}_{z^\tau} A_0(x,y,z).$$

*Then we can find a closed $\mathsf{HA}^\omega$-term $t$ such that*

$$\mathsf{HA}^\omega + \Delta_\varepsilon \vdash \forall_{x^1}\forall_{y\leq_1 sx}\exists_{z\leq_\tau tx} A_0(x,y,z).$$

PROOF. Notice that we can view WKL as one of the axioms $\Delta$, because the $\varepsilon$-weakening of WKL is derivable in $\mathsf{HA}^\omega$.

The first step is to apply elimination of extensionality. This gives

$$\mathsf{PA}^\omega + (\text{QF-}\tilde{\text{AC}}^{0,1})^E + \tilde{\Delta}^E \vdash \forall_{x^1}\forall_{y\leq_1 sx}\tilde{\exists}_{z^\tau;Ez}\, A_0(x,y,z).$$

For $\tilde{\Delta} = \forall_{a^\delta}\tilde{\exists}_{b\leq_\sigma ra}\forall_{c^\gamma} B_0(a,b,c)$ the restriction on the level of $\sigma$ implies

$$\tilde{\Delta}^E = \forall_{a^\delta;Ea}\tilde{\exists}_{b\leq_\sigma ra}\forall_{c^\gamma;Ec} B_0(a,b,c).$$

Hence $\tilde{\Delta} \to \tilde{\Delta}^E$. Similarly, $\text{QF-}\tilde{\text{AC}}^{0,1} \to (\text{QF-}\tilde{\text{AC}}^{0,1})^E$, and hence

$$\mathsf{PA}^\omega + \text{QF-}\tilde{\text{AC}}^{0,1} + \tilde{\Delta} \vdash \forall_{x^1}\forall_{y\leq_1 sx}\tilde{\exists}_{z^\tau}\, A_0(x,y,z).$$

By the proof of the corollary in Sec.3.6.3 (which goes through in exactly the same way if "WE-" is left out throughout) we have the claim. □

This "metatheorem" has found many applications, particularly in work of Kohlenbach. One such application – parameter independence of best $L_1$-approximation – can be found in Kohlenbach and Oliva (2003).

## 3.7. Notes

Much of the material in the present chapter is due to Troelstra (1973). More information on the BHK-interpretation and its history may be found in (Troelstra and van Dalen, 1988, 1.3, 1.5.3).

The results in Sec.3.3.2 are due to Howard (1973). The lemma in Sec.3.3.1 relating Howard's majorization relation with pointwise $\geq_\rho$ is due to Kohlenbach (1992b). The result on the Fan Rule in Sec.3.3.5 has been proved in Troelstra's (1977); the short proof given here is due to Kohlenbach (1992b). The two theorems on extraction of uniform bounds, for realizability in Sec.3.3.5 and in the Dialectica interpretation in Sec.3.5.2 are from Kohlenbach (1992b, 1998). The so-called monotone functional interpretation treated in Sec.3.5.1 and also the combination of the negative translation and monotone functional interpretation have been introduced by Kohlenbach (1996).

The fact that the weak Lemma of König WKL can be written as a $\forall\exists_{\leq}\forall$-Axiom (Sec.3.5.3) has been observed by Kohlenbach (1992a). The proof given uses ideas of Ishihara (2006).

The result in Sec.3.5.4 that every functional of type 2 definable in $\mathsf{HA}^\omega$ has an (even uniform) modulus of continuity definable in $\mathsf{HA}^\omega$ as well (when applied to arguments $\leq_1 y$) is due to Kreisel; a proof can be found in (Schwichtenberg, 1973). The present proof is from Kohlenbach (1992b).

The theorem on extraction of uniform bounds from classical proofs with extensionality in Sec.3.6.5 is due to Kohlenbach (1993); the formulation given is from (Kohlenbach, 2006).

# Bibliography


A. Abel and T. Altenkirch. A predicative strong normalization proof for a $\lambda$-calculus with interleaving inductive types. In *Types for Proofs and Programs, International Workshop, TYPES '99, Lökeberg, Sweden, June 1999*, volume 1956 of *LNCS*, pages 21–40. Springer Verlag, Berlin, Heidelberg, New York, 2000.

S. Abramsky. Domain theory in logical form. *Annals of Pure and Applied Logic*, 51:1–77, 1991.

S. Abramsky and A. Jung. Domain theory. In S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 3, pages 1–168. Clarendon Press, 1994.

R. M. Amadio and P.-L. Curien. *Domains and Lambda-Calculi*. Cambridge University Press, 1998.

H. Barendregt, M. Coppo, and M. Dezani-Ciancaglini. A filter lambda model and the completeness of type assignment. *The Journal of Symbolic Logic*, 48(4):931–940, 1983.

H. Benl. Konstruktive Interpretation induktiver Definitionen. Master's thesis, Mathematisches Institut der Universität München, 1998.

J. Berger and H. Ishihara. Brouwer's fan theorem and unique existence in constructive analysis. *Math. Log. Quart.*, 51(4):360–364, 2005.

U. Berger. Program extraction from normalization proofs. In M. Bezem and J. Groote, editors, *Typed Lambda Calculi and Applications*, volume 664 of *LNCS*, pages 91–106. Springer Verlag, Berlin, Heidelberg, New York, 1993a.

U. Berger. Total sets and objects in domain theory. *Annals of Pure and Applied Logic*, 60:91–117, 1993b.

U. Berger. Continuous semantics for strong normalization. In *Proc. CiE 2005*, volume 3526 of *LNCS*, pages 23–34, 2005.

U. Berger, S. Berghofer, P. Letouzey, and H. Schwichtenberg. Program extraction from normalization proofs. *Studia Logica*, 82:27–51, 2006.

U. Berger, W. Buchholz, and H. Schwichtenberg. Refined program extraction from classical proofs. *Annals of Pure and Applied Logic*, 114:3–25, 2002.

U. Berger, M. Eberl, and H. Schwichtenberg. Term rewriting for normalization by evaluation. *Information and Computation*, 183:19–42, 2003.

M. Bezem. Strongly majorizable functionals of finite type: a model for bar-recursion containing discontinuous functionals. *The Journal of Symbolic Logic*, 50:652–660, 1985.

E. Bishop and D. Bridges. *Constructive Analysis*, volume 279 of *Grundlehren der mathematischen Wissenschaften*. Springer, Berlin, 1985.

F. Blanqui, J.-P. Jouannaud, and M. Okada. The Calculus of Algebraic Constructions. In *RTA'99. LNCS 1631*, 1999.

T. Coquand, G. Sambin, J. Smith, and S. Valentini. Inductively generated formal topologies. *Annals of Pure and Applied Logic*, 124:71–106, 2003.

T. Coquand and A. Spiwack. Proof of normalisation using domain theory. Slides of a talk, October 2005.

N. G. de Bruijn. Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with application to the Church-Rosser theorem. *Indagationes Math.*, 34:381–392, 1972.

A. Dragalin. New kinds of realizability. In *Abstracts of the 6th International Congress of Logic, Methodology and Philosophy of Sciences*, pages 20–24, Hannover, Germany, 1979.

Y. L. Ershov. Everywhere defined continuous functionals. *Algebra i Logika*, 11(6):656–665, 1972.

Y. L. Ershov. Maximal and everywhere defined functionals. *Algebra i Logika*, 13(4):374–397, 1974.

H. Friedman. Intuitionistic completeness of Heyting's predicate calculus. *Notices Amer. Math. Soc.*, 22:A–648, 1975.

G. Gentzen. Untersuchungen über das logische Schließen. *Mathematische Zeitschrift*, 39:176–210, 405–431, 1934.

J.-Y. Girard. Une extension de l'interprétation de Gödel à l'analyse, et son application à l'élimination des coupures dans l'analyse et la théorie des types. In J. Fenstad, editor, *Proceedings of the Second Scandinavian Logic Symposium*, pages 63–92. North–Holland, Amsterdam, 1971.

K. Gödel. Zum intuitionistischen Aussagenkalkül. *Anzeiger der Akademie der Wissenschaften in Wien*, 69:65–66, 1932.

K. Gödel. Über eine bisher noch nicht benützte Erweiterung des finiten Standpunkts. *Dialectica*, 12:280–287, 1958.

K. Gödel. *Collected Works, Volume II, Publications 1938–1974*. Oxford University Press, 1990.

M.-D. Hernest. *Feasible programs from (non-constructive) proofs by the light (monotone) Dialectica interpretation*. PhD thesis, Submitted to Ecole Polytechnique Paris and LMU München, 2006.

D. Hilbert and P. Bernays. *Grundlagen der Mathematik I*, volume 40 of *Grundlehren der mathematischen Wissenschaften*. Springer–Verlag, Berlin, second edition, 1968.

W. A. Howard. Hereditarily majorizable functionals of finite type. In A. Troelstra, editor, *Mathematical Investigation of Intuitionistic Arithmetic and Analysis*, volume 344 of *Lecture Notes in Mathematics*, pages 454–461. Springer Verlag, Berlin, Heidelberg, New York, 1973.

H. Ishihara. An omniscience principle, the König lemma and the Hahn-Banach theorem. *Zeitschr. f. math. Logik und Grundlagen d. Math.*, 36: 237–240, 1990.

H. Ishihara. Weak König lemma implies Brouwer's fan theorem: a direct proof. *Notre Dame J. Formal Logic*, 47:249–252, 2006.

F. Joachimski and R. Matthes. Short proofs of normalisation for the simply-typed $\lambda$-calculus, permutative conversions and Gödel's $T$. *Archive for Mathematical Logic*, 42:59–87, 2003.

I. Johansson. Der Minimalkalkül, ein reduzierter intuitionistischer Formalismus. *Compositio Mathematica*, 4:119–136, 1937.

K. F. Jørgensen. Finite type arithmetic. Master's thesis, University of Roskilde, 2001.

S. C. Kleene. Countable functionals. In A. Heyting, editor, *Constructivity in Mathematics*, pages 81–100. North–Holland, Amsterdam, 1959.

U. Kohlenbach. Effective bounds from ineffective proofs in analysis: an application of functional interpretation and majorization. *The Journal of Symbolic Logic*, 57(4):1239–1273, 1992a.

U. Kohlenbach. Pointwise hereditary majorization and some applications. *Archive for Mathematical Logic*, 31:227–241, 1992b.

U. Kohlenbach. Effective moduli from ineffective uniqueness proofs. an unwinding of de La Vallée Poussin's proof for Chebycheff approximation. *Annals of Pure and Applied Logic*, 64:27–94, 1993.

U. Kohlenbach. Analysing proofs in analysis. In W. Hodges, M. Hyland, C. Steinhorn, and J. Truss, editors, *Logic: from Foundations to Applications. European Logic Colloquium (Keele, 1993)*, pages 225–260. Oxford University Press, 1996.

U. Kohlenbach. Relative constructivity. *The Journal of Symbolic Logic*, 63 (4):1218–1238, 1998.

U. Kohlenbach. Proof interpretations and the computational content of proofs. Draft of a book, 2006.

U. Kohlenbach and P. Oliva. Proof mining in $L_1$ approximation. *Annals of Pure and Applied Logic*, 121:1–38, 2003.

A. N. Kolmogorov. On the principle of the excluded middle (Russian). *Matematicheskij Sbornik. Akademiya Nauk SSSRi Moskovskoe Matematicheskoe Obshchestvo*, 32:646–667, 1925. Translated in J. van Heijenoort, From Frege to Gödel. A Source Book in Mathematical Logic 1879–1931, Harvard University Press, Cambridge, MA., 1967, pp. 414–437.

G. Kreisel. Interpretation of analysis by means of constructive functionals of finite types. In A. Heyting, editor, *Constructivity in Mathematics*, pages 101–128. North–Holland, Amsterdam, 1959.

L. Kristiansen and D. Normann. Total objects in inductively defined types. *Archive for Mathematical Logic*, 36(6):405–436, 1997.

K. G. Larsen and G. Winskel. Using information systems to solve recursive domain equations. *Information and Computation*, 91:232–258, 1991.

I. Loeb. Equivalents of the (Weak) Fan Theorem. *Annals of Pure and Applied Logic*, 132(1):51–66, 2005.

P. Martin-Löf. The domain interpretation of type theory. Talk at the workshop on semantics of programming languages, Chalmers University, Göteborg, August 1983.

P. Martin-Löf. *Intuitionistic Type Theory.* Bibliopolis, 1984.

P. Oliva. Unifying functional interpretations. *Notre Dame J. Formal Logic*, 47:262–290, 2006.

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

G. D. Plotkin. $\mathbf{T}^{\omega}$ as a universal domain. *Journal of Computer and System Sciences*, 17:209–236, 1978.

D. Prawitz. *Natural Deduction*, volume 3 of *Acta Universitatis Stockholmiensis. Stockholm Studies in Philosophy.* Almqvist & Wiksell, Stockholm, 1965.

H. Schwichtenberg. Einige Anwendungen von unendlichen Termen und Wertfunktionalen. Habilitationsschrift, Mathematisches Institut der Universität Münster, 1973.

H. Schwichtenberg. Density and choice for total continuous functionals. In P. Odifreddi, editor, *Kreiseliana. About and Around Georg Kreisel*, pages 335–362. A.K. Peters, Wellesley, Massachusetts, 1996.

H. Schwichtenberg. A direct proof of the equivalence between Brouwer's fan theorem and König's lemma with a uniqueness hypothesis. *Journal of Universal Computer Science*, 11(12):2086–2095, 2005. `http://www.jucs.org/jucs_11_12/a_direct_proof_of`.

H. Schwichtenberg. Recursion on the partial continuous functionals. In C. Dimitracopoulos, L. Newelski, D. Normann, and J. Steel, editors, *Logic Colloquium '05*, volume 28 of *Lecture Notes in Logic*, pages 173–201. Association for Symbolic Logic, 2006.

D. Scott. A type theoretical alternative to ISWIM, CUCH, OWHY. Published in Theoret. Comput. Sci. 121 (1993), 411–440, 1969.

D. Scott. Outline of a mathematical theory of computation. Technical Monograph PRG–2, Oxford University Computing Laboratory, 1970.

D. Scott. Domains for denotational semantics. In E. Nielsen and E. Schmidt, editors, *Automata, Languages and Programming*, volume 140 of *LNCS*, pages 577–613. Springer Verlag, Berlin, Heidelberg, New York, 1982. A corrected and expanded version of a paper prepared for ICALP'82, Aarhus, Denmark.

V. Stoltenberg-Hansen, E. Griffor, and I. Lindström. *Mathematical Theory of Domains*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1994.

W. W. Tait. Normal form theorem for bar recursive functions of finite type. In J. Fenstad, editor, *Proceedings of the Second Scandinavian Logic Symposium*, pages 353–367. North–Holland, Amsterdam, 1971.

G. Takeuti. *Two Applications of Logic to Mathematics*. Iwanami Shoten Publ. and Princeton Univ. Press, 1978.

A. S. Troelstra, editor. *Metamathematical Investigation of Intuitionistic Arithmetic and Analysis*, volume 344 of *Lecture Notes in Mathematics*. Springer Verlag, Berlin, Heidelberg, New York, 1973.

A. S. Troelstra. Some models for intuitionistic finite type arithmetic with fan functional. *The Journal of Symbolic Logic*, 42:194–202, 1977.

A. S. Troelstra and D. van Dalen. *Constructivism in Mathematics. An Introduction*, volume 121, 123 of *Studies in Logic and the Foundations of Mathematics*. North–Holland, Amsterdam, 1988.

F. van Raamsdonk and P. Severi. On normalisation. Computer Science Report CS-R9545 1995, Centrum voor Wiskunde en Informatica, 1995. Forms a part of van Raamsdonk's thesis from 1996.

J. Zucker. Iterated inductive definitions, trees and ordinals. In A. Troelstra, editor, *Mathematical Investigation of Intuitionistic Arithmetic and Analysis*, volume 344 of *Lecture Notes in Mathematics*, pages 392–453. Springer Verlag, Berlin, Heidelberg, New York, 1973.

# Index