# Strong Normalization for the Parameter-Free Polymorphic Lambda Calculus Based on the Ω-Rule

**Ryota Akiyoshi[1] and Kazushige Terui[2]**

1 **WIAS, Waseda University, Tokyo, Japan**
georg.logic@gmail.com
2 **RIMS, Kyoto University, Kyoto, Japan**
terui@kurims.kyoto-u.ac.jp

## Abstract

Following Aehlig [3], we consider a hierarchy $\mathbf{F}^p = \{\mathbf{F}^p_n\}_{n\in\mathbb{N}}$ of parameter-free subsystems of System $\mathbf{F}$, where each $\mathbf{F}^p_n$ corresponds to $\mathbf{ID}_n$, the theory of $n$-times iterated inductive definitions (thus our $\mathbf{F}^p_n$ corresponds to the $n+1$th system of [3]). We here present two proofs of strong normalization for $\mathbf{F}^p_n$, which are directly formalizable with inductive definitions. The first one, based on the Joachimski-Matthes method, can be fully formalized in $\mathbf{ID}_{n+1}$. This provides a tight upper bound on the complexity of the normalization theorem for System $\mathbf{F}^p_n$. The second one, based on the Gödel-Tait method, can be locally formalized in $\mathbf{ID}_n$. This provides a direct proof to the known result that the representable functions in $\mathbf{F}^p_n$ are provably total in $\mathbf{ID}_n$. In both cases, Buchholz' Ω-rule plays a central role.

**1998 ACM Subject Classification** F4.1 Mathematical Logic

**Keywords and phrases** Polymorphic Lambda Calculus, Strong Normalization, Computability Predicate, Infinitary Proof Theory

**Digital Object Identifier** 10.4230/LIPIcs.FSCD.2016.5

## 1 Introduction

It is well known that the second-order predicate calculus admits cut-elimination as shown by Tait using a model theoretic method, which implies the consistency of the second-order Peano arithmetic **PA2** with full comprehension. Neither his proof nor its variants, however, are considered an ultimate solution to *Takeuti's conjecture* (cut-elimination for higher order logics)[1] from the viewpoint of traditional proof theory, since they do not fully elucidate the nature of impredicativity involved in second-order arithmetic. As it is quite hard to give a proof-theoretic analysis of **PA2** directly, people in proof theory have been working on its subsystems, such as $\Pi^1_1$-$\mathbf{CA}_0$, the second-order arithmetic with $\Pi^1_1$-comprehension, and $\mathbf{ID}_n$, the theory of $n$-times iterated inductive definitions[2]. An early important achievement

[1] Precisely speaking, Takeuti's conjecture asks for a "finitistic" proof of cut-elimination for higher order logics, where his finitistic stand point is indeed a considerable extension of Hilbert's original one. As to Takeuti's philosophical position, we refer to [16].

[2] After Gentzen's monumental cut-elimination theorem for **PA** in 1930's, Takeuti proved cut-elimination for $\Pi^1_1$-$\mathbf{CA}_0 + \mathbf{BI}$ (bar induction) in 1967 [15], and Feferman, Buchholz, Pohlers, and Sieg subsequently investigated theories of inductive definitions and $\Pi^1_1$-$\mathbf{CA}_0$ in 1970's [8]. For these developments of proof theory, we refer to Feferman's [10].


1st International Conference on Formal Structures for Computation and Deduction (FSCD 2016).
Editors: Delia Kesner and Brigitte Pientka; Article No. 5; pp. 5:1–5:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

is $\Pi^1_1$-$\mathbf{CA}_0 = \mathbf{ID}_{<\omega} = \bigcup_{n\in\mathbb{N}} \mathbf{ID}_n$, which provides a *reduction* of an impredicative theory $\Pi^1_1$-$\mathbf{CA}_0$ to a "predicative" one $\mathbf{ID}_{<\omega}$[3].

Translated into lambda calculus, **PA2** corresponds to System **F** and cut-elimination corresponds to normalization. As is well known, strong normalization for System **F** was proved by an extremely powerful and elegant technique known as the *Tait-Girard method* or the *reducibility candidates* argument [11]. However, the same question persists: does it really elucidate impredicativity? In fact, the proof by reducibility candidates does not give any reduction, but just defers the foundational issue to third-order arithmetic. Since the whole System **F** is too powerful, a realistic approach is to begin with its subsystems for which one can prove normalization in a "predicative" way.

Following Altenkirch, Coquand [5] and Aehlig [3], we consider the *parameter-free* fragment $\mathbf{F}^p$ of System **F** as well as its subsystems $\mathbf{F}^p_n$ so that $\mathbf{F}^p = \bigcup_{n\in\mathbb{N}} \mathbf{F}^p_n$ holds[4] (Section 2). Given a system $L$ of lambda calculus and a system $A$ of arithmetic, let us write $L \propto A$ if the representable functions in $L$ coincide with the provably total functions in $A$. Then their main results can be stated as follows:

$$\mathbf{F}^p_0 \propto \mathbf{PA}\ [5], \qquad \mathbf{F}^p_n \propto \mathbf{ID}_n \text{ and } \mathbf{F}^p \propto \Pi^1_1\text{-}\mathbf{CA}_0\ [3].$$

As usual, one direction of the correspondence is established by a forgetful translation of arithmetical derivations into lambda terms. A little bit delicate is the other direction, which is shown by *locally* formalizing a normalization proof. By "local" we mean a term-wise formalization of the statement "$M\mathsf{n}$ is normalizable for every Church numeral $\mathsf{n}$" for each fixed term $M : \boldsymbol{N} \Rightarrow \boldsymbol{N}$.

While the above is definitely a great achievement, it is not completely satisfactory since they only prove *weak* normalization for terms of specific type $\boldsymbol{N}$, and only provides a *local* formalization. Also, the argument in [3] is *indirect* as it passes through intermediate systems of second-order Heyting arithmetic proposed in [2].

The purpose of this paper is to improve the current situation by showing:

1. A proof of *strong* normalization for $\mathbf{F}^p_n$, which is *fully* formalizable in $\mathbf{ID}_{n+1}$ (Section 3).
2. Another proof of strong normalization for $\mathbf{F}^p_n$, which is locally but *directly* formalizable in $\mathbf{ID}_n$ (Section 4).

The first one is based on what we call the *Joachimski-Matthes method* (JM method), which is pioneered by [18] and established by [12] as methodology. It fits the "predicative" spirit of inductive definitions very well, and leads to a sharp upper bound on the complexity of the normalization theorem for $\mathbf{F}^p_n$ (as we know $\mathbf{ID}_n \nvdash \mathsf{SN}(\mathbf{F}^p_n)$, our result $\mathbf{ID}_{n+1} \vdash \mathsf{SN}(\mathbf{F}^p_n)$ is best possible).

The second proof is based on the standard computability argument, which we call the *Gödel-Tait method*[5]. It is particularly suitable for local formalization. Combined with the

---

[3] Here we use term "predicative" to refer to a system without circular definitions. That is, predicativity in the sense of Martin-Löf's type theory, *not* in the sense of Feferman's ordinal analysis. Indeed, the proof-theoretic ordinals of both systems are far beyond $\Gamma_0$, the limit of predicative ordinals. Our usage also conforms to [1], in which strong normalization is proved in a "predicative" way for lambda calculus with interleaving inductive data types.

[4] Our system $\mathbf{F}^p_n$ corresponds to the $n+1$th system $\mathbf{F}^{\times}_{n+1}$ of [3]. We keep using our notation to have a better correspondence with systems of arithmetic.

[5] As is well known, Tait introduced his computability argument in [13]. However, Troelstra pointed out in [17] that Gödel already suggested a similar idea in his Princeton notes, and Tait himself admitted that Gödel knew essentially the same argument at that time [14, p.115]. Hence, it is not unfair to call it the *Gödel*-Tait method.

JM method, it provides a direct proof to the results in [5] and [3] that the representable functions in System $\mathbf{F}^p_n$ are provably total in $\mathbf{ID}_n$ (recall that $\mathbf{ID}_0 = \mathbf{PA}$).

Apart from the technical results themselves, this paper exhibits two apparently orthogonal methods for strong normalization in a comparable way. Both are very rare proofs of normalization for an impredicative system which do not rely on reducibility candidates in any sense. Candidates are replaced by the $\Omega$-*rule* of Buchholz [6, 9, 7], a well-known technique in proof theory (see below for an intuition). It has been used for ordinal analyses of the theories of iterated inductive definitions and iterations of $\Pi^1_1$-$\mathbf{CA}_0$, where an essential ingredient is a *partial* cut-elimination theorem for arithmetical sequents. It is recently extended to a *complete* cut-elimination theorem for arbitrary sequents by the first author and Mints [4]. One of our real motivations in this work is to bring this important technique to the realm of lambda calculus, where we do not yet find any explicit use of it[6].

**An intuition of the $\Omega$-rule.** Let us conclude the introduction by giving an intuition of the $\Omega$-rule. While it was originally introduced in the context of arithmetic, its basic idea can be explained in terms of the (standard) sequent calculus for second-order propositional intuitionistic logic. In what follows, we assume that (i) any second-order formula $\forall\alpha.A(\alpha)$ has a quantifier-free body $A(\alpha)$, and (ii) $\mathsf{Fv}(A(\alpha)) \subseteq \{\alpha\}$ (*parameter-free*). This corresponds to the restriction imposed by [5].

Recall that one of the most innovative ideas of Gentzen is to replace *axioms* with *rules:*

$$C \Rightarrow D \quad \mapsto \quad \frac{D, \Gamma \Rightarrow \Pi}{C, \Gamma \Rightarrow \Pi} \qquad\qquad D, \Gamma \Rightarrow \Pi \quad \mapsto \quad \frac{\Delta \Rightarrow D}{\Delta, \Gamma \Rightarrow \Pi}$$

so that we obtain a good cut-elimination procedure.

According to our understanding, the essence of the $\Omega$-rule lies in applying this replacement *twice* to the comprehension axiom $\forall\alpha.A(\alpha) \Rightarrow A(B)$:

$$\forall\alpha.A(\alpha) \Rightarrow A(B) \quad \mapsto \quad \frac{A(B), \Gamma \Rightarrow \Pi}{\forall\alpha.A(\alpha), \Gamma \Rightarrow \Pi}\,(\forall l) \quad \mapsto \quad \frac{\{\ \Delta, \Gamma \Rightarrow \Pi\ \}_{\Delta \Rightarrow A(B)}}{\forall\alpha.A(\alpha), \Gamma \Rightarrow \Pi}$$

where the last rule has a premise $\Delta, \Gamma \Rightarrow \Pi$ for each provable sequent $\Delta \Rightarrow A(B)$.

Unfortunately, the last rule is not useful to inductively define the set of provable sequents, since the indices of the premises themselves depend on provability. To break this circularity, we consider a two-layered setting. Let us write $\Gamma \Rightarrow^\alpha A(\alpha)$ if $\Gamma \Rightarrow A(\alpha)$ is provable and $\alpha \notin \mathsf{Fv}(\Gamma)$ (the *eigenvariable* condition). We also write $\Gamma \Rightarrow^\alpha_{\mathsf{fo}} A(\alpha)$ if furthermore $\Gamma$ and $A(\alpha)$ are quantifier-free. Since second-order intuitionistic logic is conservative over first-order one, the provability $\Gamma \Rightarrow^\alpha_{\mathsf{fo}} A(\alpha)$ can be defined without recourse to the second-order part.

We are now ready to introduce the $\Omega$-rule corresponding to $\forall\alpha.A(\alpha) \Rightarrow A(B)$:

$$\frac{\{\ \Delta, \Gamma \Rightarrow \Pi\ \}_{\Delta \Rightarrow^\alpha_{\mathsf{fo}} A(\alpha)}}{\forall\alpha.A(\alpha), \Gamma \Rightarrow \Pi}\,(\Omega)$$

This rule indeed admits a well-defined reduction step:

$$\frac{\dfrac{\Sigma \Rightarrow^\alpha A(\alpha)}{\Sigma \Rightarrow \forall\alpha.A(\alpha)}\,(\forall r) \qquad \dfrac{\{\ \Delta, \Gamma \Rightarrow \Pi\ \}_{\Delta \Rightarrow^\alpha_{\mathsf{fo}} A(\alpha)}}{\forall\alpha.A(\alpha), \Gamma \Rightarrow \Pi}\,(\Omega)}{\Sigma, \Gamma \Rightarrow \Pi}\,(cut) \quad \longrightarrow \quad \Sigma, \Gamma \Rightarrow \Pi$$

[6] Article [2] mentioned above uses the $\Omega$-rule for subsystems of second-order Heyting arithmetic, not directly for lambda calculus.

*provided that* $\Sigma$ *is quantifier-free*. Indeed, we have $\Sigma \Rightarrow^{\alpha}_{\mathsf{fo}} A(\alpha)$ by the premise of $(\forall r)$, hence $\Sigma, \Gamma \Rightarrow \Pi$ is a premise of the $\Omega$-rule.

Moreover, the standard left rule $(\forall l)$ for $\forall$, inferring $\forall\alpha.A(\alpha), \Gamma \Rightarrow \Pi$ from $A(B), \Gamma \Rightarrow \Pi$ (see above), can be simulated by the $\Omega$-rule. To see this, suppose that $\Delta \Rightarrow^{\alpha}_{\mathsf{fo}} A(\alpha)$ (an index of the $\Omega$-rule). We then obtain $\Delta \Rightarrow A(B)$ by substituting $B$ for $\alpha$, hence $\Delta, \Gamma \Rightarrow \Pi$ by the premise of $(\forall l)$. By $(\Omega)$, we conclude $\forall\alpha.A(\alpha), \Gamma \Rightarrow \Pi$.

Thus provability is preserved by replacing $(\forall l)$ with $(\Omega)$ (called *Embedding* in traditional proof theory), while cut-elimination can be proved *predicatively* (without any semantic argument), provided that the conclusion sequent is quantifier-free and the requirements (i) and (ii) above are satisfied (called *Collapsing*). This technique can be further extended to the *parameter-free* fragment of second order intuitionistic logic, which correspond to the system studied in [3].

# 2 System $\mathbf{F}^p$

## 2.1 Syntax

Given a countable set of *type variables* $\alpha, \beta, \gamma, \dots$, we define the set $\mathsf{Tp}_n$ of *types at level* $n$ for each $n \in \mathbb{N} \cup \{-1\}$ as follows:

$$A_n, B_n \quad ::= \quad \alpha \mid A_n \Rightarrow B_n \mid \forall\alpha.A_{n-1}$$

with the proviso that there is no type at level $-2$, and type $\forall\alpha.A_{n-1}$ can be formed only when $\mathsf{Fv}(A_{n-1}) \subseteq \{\alpha\}$. Here $\mathsf{Fv}(A)$ denotes the set of free type variables in $A$. That is to say, a quantified type $\forall\alpha.A$ is always *parameter-free*, so that we may treat it as a self-standing entity (like a data type). Let $\mathsf{Tp} := \bigcup_{n\in\mathbb{N}\cup\{-1\}} \mathsf{Tp}_n$.

Since there is no type at level $-2$, $\mathsf{Tp}_{-1}$ just consists of simple types. As it is unpleasant to refer to a negative integer, we write $\mathsf{simp}$ to denote the number $-1$. Thus $\mathsf{Tp}_{\mathsf{simp}} = \mathsf{Tp}_{-1}$. Types in $\mathsf{Tp}_0$ are built by arrow $\Rightarrow$ from type variables and $\forall\alpha.A$, where $A$ is a simple type over single variable $\alpha$. For instance:

| | | | | |
|---|---|---|---|---|
| $\boldsymbol{N}$ | $:=$ | $\forall\alpha.(\alpha \Rightarrow \alpha) \Rightarrow (\alpha \Rightarrow \alpha)$ | $\in \mathsf{Tp}_0$ | (natural numbers) |
| $\boldsymbol{T}$ | $:=$ | $\forall\alpha.(\alpha \Rightarrow \alpha \Rightarrow \alpha) \Rightarrow (\alpha \Rightarrow \alpha)$ | $\in \mathsf{Tp}_0$ | (binary trees) |
| $\boldsymbol{L}(\boldsymbol{N})$ | $:=$ | $\forall\alpha.(\boldsymbol{N} \Rightarrow \alpha \Rightarrow \alpha) \Rightarrow (\alpha \Rightarrow \alpha)$ | $\in \mathsf{Tp}_1$ | (lists of nat. numbers) |
| $\boldsymbol{O}$ | $:=$ | $\forall\alpha.((\boldsymbol{N} \Rightarrow \alpha) \Rightarrow \alpha) \Rightarrow (\alpha \Rightarrow \alpha) \Rightarrow (\alpha \Rightarrow \alpha)$ | $\in \mathsf{Tp}_1$ | (Brouwer ordinals) |

On the other hand, $\boldsymbol{L}(\beta) := \forall\alpha.(\beta \Rightarrow \alpha \Rightarrow \alpha) \Rightarrow (\alpha \Rightarrow \alpha)$ is not a type. Hence the polymorphic map function, whose type would be

$$\forall\beta.\forall\gamma.(\beta \Rightarrow \gamma) \Rightarrow \boldsymbol{L}(\beta) \Rightarrow \boldsymbol{L}(\gamma),$$

is not representable in our setting. A more striking example is $\forall\beta.(\boldsymbol{L}(\beta) \Rightarrow \beta) \Rightarrow \beta$, which is the type for finitely but arbitrarily branching trees. Thus *interleaving* inductive data types (cf. [1]) are out of scope and left to future work.

An important property is that
**(*)** $A, B \in \mathsf{Tp}_n$ implies $A[B/\alpha] \in \mathsf{Tp}_n$,
where $[B/\alpha]$ stands for a substitution (which is always capture-free).

We now introduce terms, which are explicitly typed à la Church. We presuppose that a countable set $\mathsf{Var}$ of symbols $x, y, z, \dots$ together with a distinguished symbol $c \notin \mathsf{Var}$ is provided. A *variable* is a pair of $x \in \mathsf{Var}$ and a type $A$, written $x^A$. Likewise a *constant* is a pair $c^A$. We never use $x^A$ and $x^B$ with $A \neq B$ together in the same context. Type

$$\frac{}{x^A \in X}\ \text{(var)} \qquad \frac{}{c^A \in X}\ \text{(con)} \qquad \frac{M^B \in X}{(\lambda x^A.M)^{A\Rightarrow B} \in X}\ \text{(abs)}$$

$$\frac{M^{A\Rightarrow B} \in X \quad N^A \in X}{(MN)^B \in X}\ \text{(app)} \qquad \frac{M^A \in X \cap \mathsf{Ec}(\alpha)}{(\Lambda\alpha.M)^{\forall\alpha.A} \in X}\ \text{(Abs)} \qquad \frac{M^A \in X}{(MB)^{A[B/\alpha]} \in X}\ \text{(App)}$$

**Figure 1** Term rules: $\mathsf{Ec}(\alpha) = \{M : x^B \in \mathsf{fv}(M) \text{ implies } \alpha \notin \mathsf{Fv}(B)\}$.

annotations are often omitted when they are irrelevant. We write $M^A$ to indicate that expression $M$ has type $A$, and $\mathsf{fv}(M)$ to denote the set of free (term) variables in $M$.

The set $\mathsf{Tm}$ of *terms* is defined to be the least set closed under the *term rules* in Figure 1, where $\mathsf{Ec}(\alpha)$ is the set of terms $M$ subject to the *eigenvariable condition* with respect to $\alpha$: for any $x^B \in \mathsf{fv}(M)$, $\alpha \notin \mathsf{Fv}(B)$.

As usual, we assume that terms are identified up to $\alpha$-equivalence. The reduction relation $\to$ is defined to be the contextual closure of

$$(\lambda x^A.M)N \to M[N/x^A], \qquad (\Lambda\alpha.M)B \to M[B/\alpha],$$

where $[N/x^A]$ stands for a capture-free substitution.

This defines the *System* $\mathbf{F}^p$. For each $n \in \mathbb{N} \cup \{\mathsf{simp}\}$, the subsystem $\mathbf{F}^p_n$ is obtained by restricting types to $\mathsf{Tp}_n$ and terms to $\mathsf{Tm}_n \subseteq \mathsf{Tm}$, which is obtained by restricting the types to $\mathsf{Tp}_n$ when applying the term rules. It is a legitimate definition since $\mathsf{Tm}_n$ is closed under reduction by (*).

Below are additional terminology and notational conventions. A term is *closed* if it does not contain a free term variable $x$ (it may contain a free type variable $\alpha$). We write $\mathsf{type}(M) = A$ if $M$ is of type $A$. Symbols $T, T_0, T_1, \dots$ stand for a term or a type, and $\overline{T}$ for a list $T_1, \dots, T_n$ $(n \geq 0)$. The following convention turns out quite useful: when we write $\overline{T} \in X$, it means that all *terms* among $T_1, \dots, T_n$ belong to $X$, leaving types aside. Finally we assume that *all terms are well typed* throughout this paper. This means that we write $MN$ only when $\mathsf{type}(M) = A \Rightarrow B$ and $\mathsf{type}(N) = A$. Likewise, we write $M[N/x^A]$ only when $\mathsf{type}(N) = A$, and $MB$ only when $\mathsf{type}(M) = \forall\alpha.A$.

▶ Remark. $\mathbf{F}^p_{\mathsf{simp}}$ is nothing but the simply typed lambda calculus, while $\mathbf{F}^p_0$ exactly corresponds to the system of [5]. If the product types are added, our $\mathbf{F}^p_n$ corresponds to System $\mathrm{F}^{\times}_{n+1}$ of [3][7]. As noted in the introduction, it is known that $\mathbf{F}^p_0 \propto \mathbf{PA}$ and $\mathbf{F}^p_n \propto \mathbf{ID}_n$ for $n \in \mathbb{N}$.

## 2.2 Strongly normalizable terms

A term $M$ is *strongly normalizable* if there is $n \in \mathbb{N}$ which bounds the length of any reduction sequence $M \equiv M_0 \to M_1 \to M_2 \to \cdots$. Since $\to$ is finitely branching, it is equivalent to say that there is no infinite reduction sequence from $M$ by König's lemma. We prefer the former definition, since it is *arithmetical* (i.e., definable by a first-order formula of Peano arithmetic). Let $\mathsf{SN}$ be the set of strongly normalizable terms in $\mathsf{Tm}$.

As is well known, the set $\mathsf{SN}$ admits an alternative inductive definition (cf. [19]).

[7] The second-order definition of product type $A \times B := \forall\alpha.(A \Rightarrow B \Rightarrow \alpha) \Rightarrow \alpha$ is not quite useful in our setting, since it would raise the level by one. However, all the results in this paper can be easily extended to systems with product types. Also, $\mathbf{F}^p_n \propto \mathbf{ID}_n$ holds in absence of product types.

$$\frac{\overline{T} \in X}{x\overline{T} \in X}\ (\mathsf{vap}) \qquad \frac{\overline{T} \in X}{c\overline{T} \in X}\ (\mathsf{cap}) \qquad \frac{M[N/x^A]\overline{T} \in X \quad N \in X}{(\lambda x^A.M)N\overline{T} \in X}\ (\beta) \qquad \frac{M[B/\alpha]\overline{T} \in X}{(\Lambda\alpha.M)B\overline{T} \in X}\ (\mathsf{B})$$

**Figure 2** SN rules.

▶ **Lemma 1.** SN *coincides with the least set $X$ closed under* (vap)*,* (cap)*,* (abs)*,* (Abs)*,* ($\beta$) *and* (B) *(see Figures 1 and 2).*

Most importantly, SN is closed under ($\beta$) (as well as (B)), a fact known as the *fundamental lemma of perpetuality* [20][8].

Notice that (var) and (con) are special cases of (vap) and (cap) with $\overline{T}$ empty. Since SN also satisfies (abs) and (Abs), we could conclude Tm $\subseteq$ SN (the *strong normalization theorem for* $\mathbf{F}^p$), if SN would satisfy (app) and (App). Of course we do not know that *a priori*. Hence proofs of strong normalization usually proceed as follows:

1. Define a set $X$ which *approximates* SN.
2. Prove Tm $\subseteq X$ by showing that $X$ is closed under the term rules (*Embedding*).
3. Prove $X \subseteq$ SN (*Collapsing*).

We may then conclude Tm $\subseteq X \subseteq$ SN, the strong normalization theorem.

## 2.3 Freezing

When discussing normalization of a lambda term, it is reasonable to distinguish two kinds of variable. For instance, consider $K \equiv \Lambda\alpha.\lambda x^{A(\alpha)}.N^{\alpha\Rightarrow\alpha}$. We immediately notice that variables $\alpha$ and $x$ are never replaced by another expression during normalization, so can be treated as if they were constants. On the other hand, $K$ may contain variables for which terms/types are actually substituted. In terms of proof theory, this corresponds to the distinction between *explicit* and *implicit* formulas [16, 4]. The following operation, called *freezing*, allows us to dynamically replace explicit bound variables with constants.

Let $o$ be a distinguished type variable which we think of as constant. It is clear that for every type $A$, there is a unique list $\bar{t}$ of constants $c$ and $o$ such that $M\bar{t}$ is of atomic type for any $M^A \in$ Tm. We write $M^\circ := M\bar{t}$. For instance, $K^\circ = K\, o\, c^{A(o)}\, c^o$, which is of type $o$.

The following lemma is obvious, since SN is closed under subterms, and the reduction rules do not make a distinction between free variables and constants.

▶ **Lemma 2.** *Let $\sigma = [\bar{o}/\alpha, \bar{c}/\overline{x}]$ be a substitution which replace some free type variables with $o$ and free term variables with $c$. If $(M\sigma)^\circ \in$* SN*, then $M \in$* SN*.*

# 3 Joachimski-Matthes method

We now present the first proof of strong normalization. It is based on the *JM method* established by Joachimski and Matthes, who gave a remarkably simple proof to strong normalization for the simply typed lambda calculus and its extensions, including System $\mathbf{T}$ [12]. It has a precursor [18], and owes the inductive characterization of SN to [19].

[8] Proofs of the lemma often rely on the definition of SN as the set of terms without infinite reduction sequences. It does not matter for our purpose, however, since $\mathbf{PA}$ can be conservatively extended to $\mathbf{ACA}_0$, in which König's lemma is available. Thus $\mathbf{PA}$ proves the lemma. Also, a very careful argument based on the other definition of SN can be found in [12, p.68] (footnote 18).

$$\frac{\overline{M} \in X}{x\overline{M} \in X}\ (\mathsf{vap}^-) \qquad \frac{\overline{T^\circ} \in X}{c\overline{T} \in X}\ (\mathsf{cap}^\circ) \qquad \frac{M \in X}{\lambda x^A.M \in X}\ (\mathsf{abs})$$

$$\frac{M \in X \cap \mathsf{Ec}(\alpha)}{\Lambda\alpha.M \in X}\ (\mathsf{Abs}) \qquad \frac{M[N/x^A]\overline{T} \in X \quad N^\circ \in X}{(\lambda x^A.M)N\overline{T} \in X}\ (\beta^\circ) \qquad \frac{M[B/\alpha]\overline{T} \in X}{(\Lambda\alpha.M)B\overline{T} \in X}\ (\mathsf{B})$$

**Figure 3** JM rules.

We begin with the simply typed lambda calculus $\mathbf{F}^p_{\mathsf{simp}}$ in 3.1. This will be useful for recalling the JM method, and also as the basis for higher level systems $\mathbf{F}^p_n$ with $n \geq 0$. The latter will be dealt with in 3.2. Finally we will informally discuss formalization in the theories of inductive definitions in 3.3.

## 3.1 Simply typed case

Let us begin with defining a suitable *term domain* for each $\mathbf{F}^p_n$, in which an approximating set $X \subseteq \mathsf{SN}$ is to be defined. The set $\mathsf{Tm}_n$ is not suitable. As it will turn out, it is crucially important to include as many terms as possible, while restricting the types of free variables and whole terms. Below is the right definition.

For each $n \in \mathbb{N} \cup \{\mathsf{simp}\}$, we define a set $\mathsf{Dom}_n$ as follows:

$$\mathsf{Dom}_n := \{M \in \mathsf{Tm} : \mathsf{type}(\mathsf{fv}(M)) \subseteq \mathsf{Tp}_n,\ \mathsf{type}(M) \in \forall\mathsf{Tp}_n\},$$

where $\forall\mathsf{Tp}_n := \mathsf{Tp}_n \cup \{\forall\alpha.A : A \in \mathsf{Tp}_n\}$.

Thus all free variables of $M \in \mathsf{Dom}_{\mathsf{simp}}$ have quantifier-free types, and $\mathsf{type}(M)$ is either quantifier-free or a quantified type $\forall\alpha.A \in \mathsf{Tp}_0$. Let us emphasize that the definition of $\mathsf{Dom}_n$ is only concerned with the types of free variables and whole terms, *not* with the internal structure of terms at all. Thus $\mathsf{Tm}_n \subsetneq \mathsf{Dom}_n$.

We now define the first approximating set $X \subseteq \mathsf{SN}$, which we call the *JM predicate at level* $-1$.

▶ **Definition 3.** Let $\mathbf{JM}_{\mathsf{simp}}$ be the least set $X \subseteq \mathsf{Dom}_{\mathsf{simp}}$ closed under the rules in Figure 3, called the *JM rules*.

Compared with the rules defining $\mathsf{SN}$ (Lemma 1), rule $(\mathsf{vap})$ is restricted to $(\mathsf{vap}^-)$. This will be important in Lemma 6, where we argue by induction on the $\Rightarrow$-rank of a type. Another difference is that the freezing operator is employed in $(\mathsf{cap}^\circ)$ and $(\beta^\circ)$. This results in a very pleasant property: for any $n \in \mathbb{N}$ and any JM rule, if the conclusion term belongs to $\mathsf{Dom}_n$, so do the premise terms. For instance, look at $(\mathsf{cap}^\circ)$. Even though $c\overline{T} \in \mathsf{Dom}_n$, the type of each term $T_i$ may be quite complicated. Still, $\mathsf{type}(T_i^\circ)$ is atomic so that $T_i^\circ$ belongs to $\mathsf{Dom}_n$. Observe that the same is true of $(\mathsf{vap}^-)$, because the type of each $M_i$ in $x^A\overline{M}$ is a subtype of $A$.

▶ **Lemma 4** (Collapsing). $\mathbf{JM}_{\mathsf{simp}} \subseteq \mathsf{SN}$.

**Proof.** $\mathsf{SN}$ is closed under the JM rules by Lemma 1; notice that closure under $(\mathsf{cap}^\circ)$ and $(\beta^\circ)$ is ensured by Lemma 2. ◀

We next proceed to Embedding ($\mathsf{Tm}_{\mathsf{simp}} \subseteq \mathbf{JM}_{\mathsf{simp}}$). We already have $(\mathsf{var})$, $(\mathsf{con})$, $(\mathsf{abs})$ and $(\mathsf{Abs})$ (though redundant), while $(\mathsf{App})$ is not needed for $\mathbf{F}^p_{\mathsf{simp}}$. Hence it just remains to show that $\mathbf{JM}_{\mathsf{simp}}$ is closed under $(\mathsf{app})$.

$$\frac{L \in X}{L^\circ \in X}\ (\mathsf{frz}) \qquad \frac{L \in X \quad B \in \mathsf{Tp}_n}{L[B/\alpha] \in X}\ (\mathsf{Sub}_n) \qquad \frac{L \in X \quad K^C \in X}{LK \in X}\ (\mathsf{app}) \qquad \frac{L \in X \quad K^C \in X}{L[K/y^C] \in X}\ (\mathsf{sub})$$

**Figure 4** Additional rules.

▶ **Lemma 5.** $\mathbf{JM}_{\mathsf{simp}}$ *is closed under* (frz) *in Figure 4.*

**Proof.** We prove a more general claim: let $\sigma = [\overline{o}/\overline{\alpha}, \overline{c}/\overline{x}]$ be a substitution of $o, c$ for free type variables and term variables, and $\overline{t}$ a list of $o, c$ such that $(L\sigma)\overline{t}$ is well typed. Then $L \in \mathbf{JM}_{\mathsf{simp}}$ implies $(L\sigma)\overline{t} \in \mathbf{JM}_{\mathsf{simp}}$.

The proof proceeds by induction on the derivation of $L \in \mathbf{JM}_{\mathsf{simp}}$.

- $L \equiv xM_1 \cdots M_n$ is derived from $\overline{M} \in \mathbf{JM}_{\mathsf{simp}}$ by ($\mathsf{vap}^-$). Then $\overline{t}$ does not contain $o$, since $\mathsf{type}(L)$ is quantifier-free (as $x$ is). Suppose that $x\sigma = x$ (the case $x\sigma = c$ is similar). By the IH, we have $M_i\sigma \in \mathbf{JM}_{\mathsf{simp}}$. Hence $(L\sigma)\overline{t} \equiv x(M_1\sigma)\cdots(M_n\sigma)\overline{t} \in \mathbf{JM}_{\mathsf{simp}}$ by ($\mathsf{vap}^-$).
- $L \equiv \Lambda\alpha.M \in \mathbf{JM}_{\mathsf{simp}}$ is derived from $M \in \mathbf{JM}_{\mathsf{simp}}$ by (Abs). We may assume that $\alpha\sigma = \alpha$ and $\overline{t}$ is of the form $o, \overline{u}$, if not empty. By the IH, we have $(M\sigma[o/\alpha])\overline{u} \in \mathbf{JM}_{\mathsf{simp}}$. Hence $(L\sigma)\overline{t} \equiv (\Lambda\alpha.M\sigma)o\overline{u} \in \mathbf{JM}_{\mathsf{simp}}$ by (B).

The other cases are similar. ◀

The next lemma is the highlight of the JM method. Given a type $A$, its $\Rightarrow$-*rank* is defined as follows:

$$\mathsf{rk}(\alpha) = \mathsf{rk}(\forall\alpha.A) := 0, \qquad \mathsf{rk}(A \Rightarrow B) := \max\{\mathsf{rk}(A)+1, \mathsf{rk}(B)\}.$$

▶ **Lemma 6.** $\mathbf{JM}_{\mathsf{simp}}$ *is closed under* (app) *and* (sub) *in Figure 4.*

**Proof.** By main induction on $\mathsf{rk}(C)$ and side induction on the derivation of $L \in \mathbf{JM}_{\mathsf{simp}}$.

For (sub), we consider two cases; the first one is crucial, while the second one is a typical one, from which the other cases are easily understood.

- $L \equiv x^A\overline{M} \in \mathbf{JM}_{\mathsf{simp}}$ is derived from $\overline{M} \equiv M_1, \ldots, M_n \in \mathbf{JM}_{\mathsf{simp}}$ by ($\mathsf{vap}^-$). Suppose that $y^C \equiv x^A$ so that $L[K/y] \equiv K(M_1[K/y])\cdots(M_n[K/y])$ (the case $y \not\equiv x$ is easier). This means that $C$ is of the form $B_1 \Rightarrow \cdots B_n \Rightarrow B_0$. By the side IH (sub) we have $M_i[K/y]^{B_i} \in \mathbf{JM}_{\mathsf{simp}}$. Since $\mathsf{rk}(B_i) < \mathsf{rk}(C)$ we may apply the main IH (app) ($n$ times) to conclude that $L[K/y] \in \mathbf{JM}_{\mathsf{simp}}$.
- $L \equiv (\lambda x.M)N\overline{T} \in \mathbf{JM}_{\mathsf{simp}}$ is derived from $M[N/x]\overline{T} \in \mathbf{JM}_{\mathsf{simp}}$ and $N^\circ \in \mathbf{JM}_{\mathsf{simp}}$. Let us use a tentative notation $M' := M[K/y]$. Then $L[K/y]$ can be written as $(\lambda x.M')N'\overline{T'}$. By the side IH, we have $M'[N'/x]\overline{T'} \in \mathbf{JM}_{\mathsf{simp}}$ and $(N')^\circ \equiv (N^\circ)' \in \mathbf{JM}_{\mathsf{simp}}$, hence $L[K/y] \in \mathbf{JM}_{\mathsf{simp}}$ by ($\beta^\circ$).

For (app), we again consider two cases.

- $L \equiv \lambda x.M \in \mathbf{JM}_{\mathsf{simp}}$ is derived from $M \in \mathbf{JM}_{\mathsf{simp}}$ by (abs). We have $M[K/x] \in \mathbf{JM}_{\mathsf{simp}}$ by the side IH (sub), hence $(\lambda x.M)K \in \mathbf{JM}_{\mathsf{simp}}$ by ($\beta^\circ$), noting that $K^\circ \in \mathbf{JM}_{\mathsf{simp}}$ follows from $K \in \mathbf{JM}_{\mathsf{simp}}$ by the previous lemma.
- $L \equiv x\overline{M} \in \mathbf{JM}_{\mathsf{simp}}$ is derived from $\overline{M} \in \mathbf{JM}_{\mathsf{simp}}$ by ($\mathsf{vap}^-$). We may add a new premise $K \in \mathbf{JM}_{\mathsf{simp}}$ to obtain $x\overline{M}K \in \mathbf{JM}_{\mathsf{simp}}$. ◀

Since $\mathbf{JM}_{\mathsf{simp}}$ satisfies (var), (con), (abs) and (app), we have $\mathsf{Tm}_{\mathsf{simp}} \subseteq \mathbf{JM}_{\mathsf{simp}} \subseteq \mathsf{SN}$ (Embedding). This completes the proof.

▶ **Theorem 7.** $\mathbf{F}^p_{\mathsf{simp}}$ *admits strong normalization.*

$$\frac{M^{\forall\alpha.A} \in X \quad \{\ K[B/\alpha]\overline{T} \in X\ \}_{K^A \in \mathbf{JM}_{n-1} \cap \mathsf{Ec}(\alpha)}}{MB\overline{T} \in X}\ (\Omega_n) \qquad \frac{M^{\forall\alpha.A} \in X \quad B \in \mathsf{Tp}_n}{MB \in X}\ (\mathsf{App}_n)$$

**Figure 5** $(\Omega_n)$ and $(\mathsf{App}_n)$.

## 3.2 Inductive cases

Now a crucial question is how to extend $\mathbf{JM}_{\mathsf{simp}}$ so that it also accommodates $(\mathsf{App})$. Extending $(\mathsf{vap}^-)$ to $(\mathsf{vap})$ would totally spoil the fine-tuned structure of the JM method, as the use of induction on $\mathsf{rk}(C)$ in the proof of Lemma 6 would not work anymore. We will instead adopt a brilliant idea due to Buchholz: the $\Omega$*-rule*.

▶ **Definition 8.** Let $\mathbf{JM}_0$ be the least set $X \subseteq \mathsf{Dom}_0$ closed under the JM rules (Figure 3) and $(\Omega_0)$ (Figure 5). More generally, $\mathbf{JM}_n$ $(n \in \mathbb{N})$ is defined to be the least set $X \subseteq \mathsf{Dom}_n$ closed under the JM rules and $(\Omega_0), \dots, (\Omega_n)$. $\mathbf{JM}_n$ is called the *JM predicate at level* $n$.

Rule $(\Omega_0)$ has a premise $K[B/\alpha]\overline{T} \in X$ for each $K \in \mathbf{JM}_{\mathsf{simp}} \cap \mathsf{Ec}(\alpha)$. Thus it depends on the set $\mathbf{JM}_{\mathsf{simp}}$, which has been already defined. In general, $\mathbf{JM}_n$ is obtained from $\mathbf{JM}_{n-1}$ by extending the term domain to $\mathsf{Dom}_n$ and by adding a new rule $(\Omega_n)$, which depends on $\mathbf{JM}_{n-1}$. Hence we have $\mathbf{JM}_{n-1} \subseteq \mathbf{JM}_n$ by definition. Notice that $B$ is an *arbitrary* type in $\mathsf{Tp}$; it is condition $K \in \mathsf{Ec}(\alpha)$ that ensures that $K[B/\alpha]\overline{T}$ belongs to $\mathsf{Dom}_n$ as far as $MB\overline{T}$ does.

▶ **Lemma 9.** $\mathbf{JM}_n$ *is closed under* $(\mathsf{App}_n)$ *(Figure 5).*

**Proof.** Suppose that $M^{\forall\alpha.A} \in \mathbf{JM}_n$ and $B \in \mathsf{Tp}_n$. For each $K^A \in \mathbf{JM}_{n-1} \cap \mathsf{Ec}(\alpha)$ we have $K \in \mathbf{JM}_n$ and so $K[B/\alpha] \in \mathbf{JM}_n$ by Lemma 10 below. Hence we obtain $MB \in \mathbf{JM}_n$ by $(\Omega_n)$. ◀

▶ Remark. The $\Omega$-rule is often called an *impredicative* cut. In the current situation, it can be thought of as a *meta-cut* on derivations, rather than a redex occurring in a term $M$. Imagine that rule $(\mathsf{Abs})$ is sort of an "introduction rule" in natural deduction. Then $(\Omega_n)$ provides a matching "elimination rule" with a notion of "reduction":

$$\frac{\dfrac{N^A \in \mathbf{JM}_n}{\Lambda\alpha.N \in \mathbf{JM}_n}\ (\mathsf{Abs}) \qquad \left\{\ \begin{matrix} \vdots\ \pi_K \\ K[B/\alpha] \in \mathbf{JM}_n \end{matrix}\ \right\}_{K^A \in \mathbf{JM}_{n-1} \cap \mathsf{Ec}(\alpha)}}{(\Lambda\alpha.N)B \in \mathbf{JM}_n}\ (\Omega_n) \quad \Longrightarrow \quad \frac{\begin{matrix} \vdots\ \pi_N \\ N[B/\alpha] \in \mathbf{JM}_n \end{matrix}}{(\Lambda\alpha.N)B \in \mathbf{JM}_n}\ (\mathsf{B})$$

which is triggered by showing $N^A \in \mathbf{JM}_{n-1}$.

The $\Omega$-rule was first introduced by Buchholz [6] to give ordinal analyses of iterated inductive definitions. His main theorem called *Collapsing* amounts to a partial cut-elimination theorem for derivations of arithmetical sequents. Later it is extended to a complete cut-elimination theorem for the $\Omega$-rule by the first author and Mints [4]. In these developments, it is always a crucial issue how to define or extend the "domain" of the $\Omega$-rule. A technical contribution of this paper is that we have managed to include strongly normalizable terms in the domain, in contrast to the "proof theoretic" domains which consist of normal (cut-free) derivations.

Coming back to the formal argument, it is not hard to extend $(\mathsf{frz})$, $(\mathsf{app})$ and $(\mathsf{sub})$ (Figure 4) to $\mathbf{JM}_n$. We also consider a new rule $(\mathsf{Sub}_n)$.

▶ **Lemma 10.** *For every* $n \in \mathbb{N} \cup \{\mathsf{simp}\}$, $\mathbf{JM}_n$ *is closed under* $(\mathsf{frz})$, $(\mathsf{Sub}_n)$, $(\mathsf{app})$ *and* $(\mathsf{sub})$.

Hence $\mathbf{JM}_n$ satisfies all of (var), (con), (abs), (Abs), (app) and ($\mathsf{App}_n$). We therefore conclude:

▶ **Lemma 11** (Embedding). *For every $n \in \mathbb{N} \cup \{\mathsf{simp}\}$, $\mathsf{Tm}_n \subseteq \mathbf{JM}_n$.*

Let us now move on to the Collapsing part. We first need an inversion lemma for ($\mathsf{cap}^\circ$).

▶ **Lemma 12.** *If $c\overline{T} \in \mathbf{JM}_n$, then $\overline{T^\circ} \in \mathbf{JM}_n$.*

**Proof.** By induction on the derivation. It is obvious if $c\overline{T} \in \mathbf{JM}_n$ is derived by ($\mathsf{cap}^\circ$). Otherwise, it is derived by ($\Omega_m$) ($m \leq n$):

$$\frac{(c\overline{T_1})^{\forall\alpha.A} \in \mathbf{JM}_n \quad \{\ K[B/\alpha]\overline{T_2} \in \mathbf{JM}_n\ \}_{K^A \in \mathbf{JM}_{m-1} \cap \mathsf{Ec}(\alpha)}}{c\overline{T_1}B\overline{T_2} \in \mathbf{JM}_n}$$

Let $K := c^A$ to obtain $c^{A[B/\alpha]}\overline{T_2} \in \mathbf{JM}_n$. By the IH (twice), we have $\overline{T_1^\circ}, \overline{T_2^\circ} \in \mathbf{JM}_n$. ◀

The next lemma lies at the heart of the $\Omega$-rule technique. It describes a "meta-cut elimination procedure" to eliminate ($\Omega_{n+1}$) from a derivation in $\mathbf{JM}_{n+1}$.

▶ **Lemma 13.** *$\mathbf{JM}_n$ satisfies ($\Omega_{n+1}$):*

$$\frac{M^{\forall\alpha.A} \in \mathbf{JM}_n \quad \{\ K[B/\alpha]\overline{T} \in \mathbf{JM}_n\ \}_{K^A \in \mathbf{JM}_n \cap \mathsf{Ec}(\alpha)}}{MB\overline{T} \in \mathbf{JM}_n}$$

**Proof.** By induction on the derivation of $M^{\forall\alpha.A} \in \mathbf{JM}_n$.

- $M \equiv \Lambda\alpha.N \in \mathbf{JM}_n$ is derived by (Abs). Then $N^A \in \mathbf{JM}_n \cap \mathsf{Ec}(\alpha)$, so let $K := N$ to obtain $N[B/\alpha]\overline{T} \in \mathbf{JM}_n$. Hence $MB\overline{T} \in \mathbf{JM}_n$ by (B).
- $M \equiv x^C\overline{N}$ is derived by ($\mathsf{vap}^-$). $C \in \mathsf{Tp}_n$ implies $\forall\alpha.A \in \mathsf{Tp}_n$, so $A \in \mathsf{Tp}_{n-1}$. Moreover, $\mathbf{JM}_{n-1} \subseteq \mathbf{JM}_n$. Hence we may apply ($\Omega_n$) to obtain the same conclusion[9].
- $M \equiv c\overline{U} \in \mathbf{JM}_n$ is derived by ($\mathsf{cap}^\circ$). Let $K := c^A$ to obtain $c^{A[B/\alpha]}\overline{T} \in \mathbf{JM}_n$. By Lemma 12, we have $\overline{U^\circ}, \overline{T^\circ} \in \mathbf{JM}_n$. Hence we obtain $MB\overline{T} \equiv c\overline{U}B\overline{T} \in \mathbf{JM}_n$ by ($\mathsf{cap}^\circ$).
- $M \equiv NC\overline{U}$ is obtained by ($\Omega_m$) with $m \leq n$:

  $$\frac{N^{\forall\beta.D} \in \mathbf{JM}_n \quad \{\ L[C/\beta]\overline{U} \in \mathbf{JM}_n\ \}_{L^D \in \mathbf{JM}_{m-1} \cap \mathsf{Ec}(\beta)}}{NC\overline{U} \in \mathbf{JM}_n}$$

  For each $L^D \in \mathbf{JM}_{m-1} \cap \mathsf{Ec}(\beta)$ we have $(L[C/\beta]\overline{U})^{\forall\alpha.A} \in \mathbf{JM}_n$. So $L[C/\beta]\overline{U}B\overline{T} \in \mathbf{JM}_n$ by the IH. Hence we obtain $MB\overline{T} \equiv NC\overline{U}B\overline{T} \in \mathbf{JM}_n$ by ($\Omega_m$).

It never happens that $M^{\forall\alpha.A} \in \mathbf{JM}_n$ is derived by (abs). The cases of ($\beta^\circ$) and (B) easily follow from the IH. ◀

The next lemma follows immediately, since $\mathbf{JM}_{n+1}$ reduces to $\mathbf{JM}_n$ by restricting the term domain to $\mathsf{Dom}_n$ and eliminating ($\Omega_{n+1}$).

▶ **Lemma 14.** *For every $n \in \mathbb{N} \cup \{\mathsf{simp}\}$, $\mathbf{JM}_{n+1} \cap \mathsf{Dom}_n = \mathbf{JM}_n$.*

As a consequence, we obtain:

▶ **Lemma 15** (Collapsing). *For every $n \in \mathbb{N} \cup \{\mathsf{simp}\}$, $\mathbf{JM}_n \subseteq \mathsf{SN}$.*

[9] Though it looks innocent, this is indeed the bottle neck of the whole argument. We restricted the term domain to $\mathsf{Dom}_n$ and introduced constants and freezing just for managing this case.

**Proof.** Given $M \in \mathbf{JM}_n$, let $N := M^\circ[\overline{c}/\overline{x}]$ be the closed term of atomic type obtained by freezing and constant substitution. Then $N \in \mathsf{Dom}_{\mathsf{simp}}$ and $N \in \mathbf{JM}_n$ by Lemma 10. Hence by Lemma 14 and Lemma 4, we obtain:

$$N \in \mathbf{JM}_n \cap \mathsf{Dom}_{\mathsf{simp}} \;=\; \mathbf{JM}_{n-1} \cap \mathsf{Dom}_{\mathsf{simp}} \;=\; \cdots \;=\; \mathbf{JM}_0 \cap \mathsf{Dom}_{\mathsf{simp}} \;=\; \mathbf{JM}_{\mathsf{simp}} \;\subseteq\; \mathsf{SN}.$$

From this, we conclude $M \in \mathsf{SN}$ by Lemma 2. ◀

▶ **Theorem 16.** *For each $n \in \mathbb{N} \cup \{\mathsf{simp}\}$, $\mathbf{F}^p_n$ admits strong normalization. Hence $\mathbf{F}^p$ admits strong normalization too.*

## 3.3 Formalization in $\mathrm{ID}_{<\omega}$

Let us recall the theories $\{\mathbf{ID}_n\}_{n\in\mathbb{N}}$ of finitely iterated inductive definitions. $\mathbf{ID}_0$ is just the standard first-order Peano arithmetic **PA**.

$\mathbf{ID}_1$ is obtained as follows. Let $A \equiv A(\boldsymbol{X}, x)$ be a first order arithmetical formula with (temporarily used) second-order variable $\boldsymbol{X}$ which occurs positively. $x$ is a first-order variable, and we suppose that $A$ does not contain any other free variables. For each such $A$, we extend the language with a new unary predicate $\boldsymbol{I}_A(x)$ together with the axioms

$$A[\boldsymbol{I}_A/\boldsymbol{X}] \subseteq \boldsymbol{I}_A, \qquad A[S/\boldsymbol{X}] \subseteq S \to \boldsymbol{I}_A \subseteq S,$$

where $S \equiv S(x, \overline{y})$ is an arbitrary formula and $B \subseteq C$ abbreviates $\forall x(B(x) \to C(x))$. Intuitively, $A$ expresses a monotone operator $\wp(\mathbb{N}) \longrightarrow \wp(\mathbb{N})$ and $\boldsymbol{I}_A$ its least fixed point. This defines $\mathbf{ID}_1$.

$\mathbf{ID}_n$ with $n > 1$ is defined similarly, except that formula $A$ can be taken from the language of $\mathbf{ID}_{n-1}$. Thus we are allowed to define a new fixed point making use of previous ones. Let $\mathbf{ID}_{<\omega} := \bigcup_{n\in\mathbb{N}} \mathbf{ID}_n$. It is known that $\mathbf{ID}_{<\omega}$ proves exactly the same arithmetical sentences as $\Pi^1_1$-$\mathbf{CA}_0$, the second-order Peano arithmetic with $\Pi^1_1$-comprehension.

Let us now discuss formalization of Theorem 16. We may assume a reasonable encoding of lambda terms as natural numbers and basic operations as primitive recursive functions (see [2]). The sets $\mathsf{SN}$ and $\mathbf{JM}_{\mathsf{simp}}$, as well as the associated induction principles, are available in **PA**, since these are defined by *finitary* rules. For instance, one can define $M \in \mathbf{JM}_{\mathsf{simp}}$ as "there exists a derivation $d$ ending with statement $M \in \mathbf{JM}_{\mathsf{simp}}$," which is arithmetical since $d$, a finite object, is encodable by a natural number.

On the other hand, the definition of $\mathbf{JM}_0$ involves $(\Omega_0)$, which is *infinitary*. It is here that inductive definitions play a role. Indeed, $\mathbf{ID}_1$ allows us to define $\mathbf{JM}_0$ quite smoothly. For $n > 0$, recall that $\mathbf{JM}_n$ involves $(\Omega_0), \ldots, (\Omega_n)$, which depend on $\mathbf{JM}_0, \ldots, \mathbf{JM}_{n-1}$ (as well as $\mathbf{JM}_{\mathsf{simp}}$, which is arithmetical and thus negligible). Hence definition of $\mathbf{JM}_n$ requires $\mathbf{ID}_{n+1}$.

Once $\mathbf{JM}_n$ has been defined, the rest of argument proceeds by induction on the derivation and some inductions on natural numbers, all of which are available in $\mathbf{ID}_{n+1}$ (see also footnote 8). We therefore conclude:

▶ **Theorem 17.** $\mathbf{ID}_{n+1}$ *proves strong normalization for $\mathbf{F}^p_n$.*

Since $\mathbf{F}^p_n \propto \mathbf{ID}_n$ [2, 3], normalization for $\mathbf{F}^p_n$ implies consistency of $\mathbf{ID}_n$ (if $\mathbf{ID}_n$ were not consistent, it would prove totality of a partial function, and $\mathbf{F}^p_n$ would represent it by a lambda term of type $\boldsymbol{N} \Rightarrow \boldsymbol{N}$, contradicting normalization). Hence by the second incompleteness theorem, $\mathbf{ID}_n$ does not prove normalization for $\mathbf{F}^p_n$. Therefore Theorem 17 is the best possible we may obtain.

Turning to the whole system, normalization for $\mathbf{F}^p$ cannot be proved in $\mathbf{ID}_{<\omega}$ for the same reason. Still, $\mathbf{ID}_{<\omega}$ has a proper extension $\mathbf{ID}_\omega$ which corresponds to $\Pi^1_1$-$\mathbf{CA}_0 + \mathbf{BI}$ (bar induction). We have:

▶ **Theorem 18.** $\mathbf{ID}_\omega$ *proves strong normalization for* $\mathbf{F}^p$*.*

## 4 Gödel-Tait method

The previous proof, fully based on the JM method, works almost fine. It is, however, not amenable to local formalization, so cannot be used to prove that the representable functions in $\mathbf{F}^p_n$ are provably total in $\mathbf{ID}_n$, since it involves $n+1$ times iteration of inductive definitions in an unavoidable way. Hence we are led back to a more conventional approach, which we call the *Gödel-Tait method* for the reason explained in footnote 5. It works perfectly, when combined with the JM method. In this section, we give an alternative proof of strong normalization in 4.1 and discuss local formalization in 4.2.

### 4.1 Computability predicates

Throughout this section, we fix $n \in \mathbb{N} \cup \{\mathsf{simp}\}$ and assume that the JM predicate $\mathbf{JM}_n$ at level $n$ has been defined. Our goal is to give an alternative proof to strong normalization for $\mathbf{F}^p_{n+1}$ by building a *computability predicate* on top of $\mathbf{JM}_n$. In the sequel, we write $\mathbf{JM} := \mathbf{JM}_n$ just for simplicity.

Anticipating local formalization later, we will work with a restricted set of types. Given a set $\mathcal{X} \subseteq \mathsf{Tp}_{n+1}$, let $\mathcal{X}_\downarrow$ be the least set containing $\mathcal{X}$ and satisfying the following conditions:

1. $A \Rightarrow B \in \mathcal{X}_\downarrow$ implies $A, B \in \mathcal{X}_\downarrow$.
2. $\forall\alpha.A \in \mathcal{X}_\downarrow$ and $D \in \mathcal{X}$ imply $A[D/\alpha] \in \mathcal{X}_\downarrow$.

It is clear that $\mathcal{X}_\downarrow$ is finite whenever $\mathcal{X}$ is.

Recall that $\mathbf{JM} = \mathbf{JM}_n \subseteq \mathsf{Dom}_n$. To address strong normalization for $\mathbf{F}^p_{n+1}$, we enlarge the domain $\mathsf{Dom}_n$ with terms of type $\mathcal{X}_\downarrow$. Let

$$\mathsf{Dom}(\mathcal{X}) := \{M \in \mathsf{Tm} : \mathsf{type}(\mathsf{fv}(M)) \subseteq \mathsf{Tp}_n,\ \mathsf{type}(M) \in \forall\mathsf{Tp}_n \cup \mathcal{X}_\downarrow\}.$$

In particular when $\mathcal{X}$ is finite, $\forall\mathsf{Tp}_n \cup \mathcal{X}_\downarrow$ consists of $\forall\mathsf{Tp}_n$ together with finitely many $\Rightarrow$-types in $\mathsf{Tp}_{n+1}$. Given $X, Y \subseteq \mathsf{Dom}(\mathcal{X})$, let us write

$$X \Rightarrow Y := \{M \in \mathsf{Dom}(\mathcal{X}) : \forall N \in X.\ MN \in Y\}.$$

Also, let $\mathbf{JM}(A) := \{M \in \mathbf{JM} : \mathsf{type}(M) = A\}$ for each $A \in \forall\mathsf{Tp}_n$.

Our first observation is the following:

▶ **Lemma 19.** *If* $A, B \in \mathsf{Tp}_n$*,* $\mathbf{JM}(A \Rightarrow B) = \mathbf{JM}(A) \Rightarrow \mathbf{JM}(B)$*.*

**Proof.** The inclusion $\subseteq$ is due to $(\mathsf{app})$ already established for $\mathbf{JM} = \mathbf{JM}_n$ by Lemma 10. For the other inclusion, let $M \in \mathbf{JM}(A) \Rightarrow \mathbf{JM}(B)$. Since $x \in \mathbf{JM}(A)$ by $(\mathsf{var})$, we have $Mx \in \mathbf{JM}(B)$ with $x$ a fresh variable. We can easily show that $M \in \mathbf{JM}$ by induction on the derivation of $Mx \in \mathbf{JM}$.

The only nontrivial case is when $(\lambda y.N)x \in \mathbf{JM}$ is derived from $N[x/y] \in \mathbf{JM}$ by $(\beta^\circ)$. In this case, we have $\lambda y.N \equiv \lambda x.N[x/y] \in \mathbf{JM}$ by $(\mathsf{abs})$. ◀

Notice that this is a *consequence* of the definition of JM predicates. The Gödel-Tait method works the other way round; we *define* a predicate by the above property, and then derive JM-like properties as consequences.

▶ **Definition 20.** For each $C \in \forall\mathsf{Tp}_n \cup \mathcal{X}_\downarrow$, we define a set $\mathbf{CP}(C) \subseteq \mathsf{Dom}(\mathcal{X})$ as follows.

$$\begin{array}{rcll} \mathbf{CP}(C) & := & \mathbf{JM}(C) & (C \in \forall\mathsf{Tp}_n) \\ & := & \mathbf{CP}(A) \Rightarrow \mathbf{CP}(B) & (C \equiv A \Rightarrow B \in \mathcal{X} \backslash \forall\mathsf{Tp}_n) \end{array}$$

Let $\mathbf{CP} = \mathbf{CP}(\mathcal{X}) := \bigcup_{C \in \forall\mathsf{Tp}_n \cup \mathcal{X}_\downarrow} \mathbf{CP}(C)$. This defines the *computability predicate at level* $n+1$, relative to $\mathcal{X}$.

▶ **Lemma 21.** *The set* $\mathbf{CP}$ *satisfies* (app), (Abs), $(\beta^\circ)$, (B), $(\Omega_m)$ $(m \leq n+1)$ *as well as*

$$\frac{M^C \in X}{M^\circ \in \mathbf{JM}}\ (\mathsf{sn}^\circ) \qquad \frac{\overline{T^\circ} \in \mathbf{JM}}{(c\overline{T})^C \in X}\ (\mathsf{csn}^\circ)$$

**Proof.** (app) is a consequence of the definition and Lemma 19. (Abs) is trivial. Indeed, if $\forall\alpha.A \in \forall\mathsf{Tp}_n \cup \mathcal{X}_\downarrow$, then $\forall\alpha.A \in \forall\mathsf{Tp}_n$ and $A \in \mathsf{Tp}_n$. Hence we have $\mathbf{CP}(A) = \mathbf{JM}(A)$ and $\mathbf{CP}(\forall\alpha.A) = \mathbf{JM}(\forall\alpha.A)$ so that it boils down to (Abs) for $\mathbf{JM}$.

$(\mathsf{sn}^\circ)$ and $(\mathsf{csn}^\circ)$ are simultaneously verified by induction on $\mathsf{rk}(C)$. If $C \in \forall\mathsf{Tp}_n$, it amounts to (frz) and $(\mathsf{cap}^\circ)$ for $\mathbf{JM}$. So suppose that $C \equiv A \Rightarrow B \in \mathcal{X}_\downarrow \backslash \forall\mathsf{Tp}_n$.

- $(\mathsf{sn}^\circ)$ Assume $M \in \mathbf{CP}(A \Rightarrow B)$. By the IH $(\mathsf{csn}^\circ)$ for $A$, we have $c \in \mathbf{CP}(A)$, so $Mc \in \mathbf{CP}(B)$ and thus $(Mc)^\circ \in \mathbf{JM}$ by the IH $(\mathsf{sn}^\circ)$ for $B$. That is, $M^\circ \in \mathbf{JM}$.
- $(\mathsf{csn}^\circ)$ Suppose that $\overline{T^\circ} \in \mathbf{JM}$. For any $N \in \mathbf{CP}(A)$, we have $N^\circ \in \mathbf{JM}$ by the IH $(\mathsf{sn}^\circ)$ for $A$. Hence $c\overline{T}N \in \mathbf{CP}(B)$ by the IH $(\mathsf{csn}^\circ)$ for $B$.

Finally $(\beta^\circ)$, (B) and $(\Omega_m)$ are verified by induction on $\mathsf{rk}(C)$, where $C$ is the type of the term in conclusion. It is obvious if $C \in \forall\mathsf{Tp}_n$. Otherwise, suppose that the conclusion term is $M^{A\Rightarrow B}$. To prove $M \in \mathbf{CP}(A \Rightarrow B)$, it suffices to show $MN \in \mathbf{CP}(B)$ for any $N \in \mathbf{CP}(A)$. But it follows from the IH straightforwardly, since these rules are closed under term application. ◀

As an immediate consequence of $(\mathsf{sn}^\circ)$, Lemma 15 and Lemma 2, we obtain:

▶ **Lemma 22** (Collapsing)**.** $\mathbf{CP} \subseteq \mathsf{SN}$.

We now proceed to the Embedding part. Since $\mathbf{CP}$ already satisfies (var), (con), (app) and (Abs), we only have to verify (abs) and $(\mathsf{App}_{n+1})$. Let us begin with the latter. The basic idea is to use $(\Omega_{n+1})$ as in Lemma 9, so we have to show that $\mathbf{CP}$ is closed under $(\mathsf{Sub}_{n+1})$ as far as needed.

Consider a term substitution $\sigma = [N_1/x_1^{A_1}, \dots, N_k/x_k^{A_k}]$. We say that $\sigma$ is a *cp-substitution* if $A_i \in \forall\mathsf{Tp}_n \cup \mathcal{X}_\downarrow$ and $N_i \in \mathbf{CP}(A_i)$ for every $1 \leq i \leq k$.

▶ **Lemma 23.** *Let* $B \in \mathcal{X}$ *and* $\sigma$ *be a cp-substitution. Suppose that* $K^A \in \mathbf{JM}$ *satisfies:*
$(\star)$ $C[B/\alpha] \in \forall\mathsf{Tp}_n \cup \mathcal{X}_\downarrow$ *for any* $C \in \mathsf{type}(\mathsf{fv}(K)) \cup \{A\}$.
*Then* $K[B/\alpha]\sigma \in \mathbf{CP}$.

We are now ready to show:

▶ **Lemma 24.** $\mathbf{CP}$ *satisfies* $(\mathsf{App}_{n+1})$ *for* $\forall\alpha.A \in \mathcal{X}_\downarrow$ *and* $B \in \mathcal{X}$.

**Proof.** Suppose that $M^{\forall\alpha.A} \in \mathbf{CP}$. We are going to use $(\Omega_{n+1})$. So let $K^A \in \mathbf{JM}_n \cap \mathsf{Ec}(\alpha)$. Any $C \in \mathsf{type}(\mathsf{fv}(K))$ does not contain $\alpha$ as free type variable, so that $C[B/\alpha] \equiv C \in \mathsf{Tp}_n$. Also, $A[B/\alpha] \in \mathcal{X}_\downarrow$. Hence $K$ satisfies the condition of the previous lemma so that $K[B/\alpha] \in \mathbf{JM}_n$ follows. Therefore we obtain $MB \in \mathbf{CP}$ by $(\Omega_{n+1})$. ◀

Given a lambda term $M \in \mathsf{Tm}_{n+1}$, let $\mathsf{subterm}(M)$ be the set of subterms of $M$ and $\mathsf{subtype}(M)$ be defined by:

$$\mathsf{subtype}(M) := \{\mathsf{type}(N) : N \in \mathsf{subterm}(M)\} \cup \{B : NB \in \mathsf{subterm}(M)\}.$$

The next lemma, the *Basic Lemma* of logical relations, establishes Embedding. It can be proved by induction on the structure of $M$ in a completely standard way.

▶ **Lemma 25.** *Suppose that* $\mathsf{subtype}(M) \subseteq \mathcal{X}$*. Then for any cp-substitution* $\sigma$*,* $M\sigma \in \mathbf{CP} = \mathbf{CP}(\mathcal{X})$*.*

In particular, any closed term $M \in \mathsf{Tm}_{n+1}$ belongs to $\mathbf{CP} = \mathbf{CP}(\mathcal{X})$ by letting $\mathcal{X} := \mathsf{subtype}(M)$. Hence in conjunction with Lemma 22, we conclude:

▶ **Theorem 26.** *System* $\mathbf{F}^p_{n+1}$ *admits strong normalization.*

## 4.2 Local formalization in $\mathbf{ID}_{n+1}$

In contrast to the JM predicates, the computability predicate $\mathbf{CP}$ does not fit the pattern of inductive definitions. Namely, it is not defined as the least fixed point of a monotone operator, due to the use of *non-monotone* operator $\Rightarrow$. Indeed, a naive formalization would require $\Pi^1_1$-comprehension, which is too high a price to pay.

On the other hand, $\mathbf{CP}$ is easily amenable to local formulation. For simplicity, let us write $m := n + 1$. If $\mathcal{X}$ is a finite set, $\mathbf{CP} = \mathbf{CP}(\mathcal{X})$ is definable from the JM predicate $\mathbf{JM}_{m-1}$ by a single formula so that it is definable in $\mathbf{ID}_m$. By formalizing the rest of argument, we obtain:

▶ **Theorem 27.** *Let* $\mathcal{X}$ *be a finite subset of* $\mathsf{Tp}_m$*. Then* $\mathbf{ID}_m$ *proves that* $M$ *is strongly normalizable for every closed term* $M \in \mathsf{Tm}_m$ *such that* $\mathsf{subtype}(M) \subseteq \mathcal{X}$*.*

A function $f : \mathbb{N} \longrightarrow \mathbb{N}$ is *representable* in System $\mathbf{F}^p_m$ if there is a lambda term $M$ of type $\boldsymbol{N} \Rightarrow \boldsymbol{N}$ such that $M\mathsf{n} \to^* \mathsf{k}$ iff $f(n) = k$ for every $n, k \in \mathbb{N}$, where $\mathsf{n}$ is the Church numeral for $n$. By noting that $\mathsf{subtype}(\mathsf{n})$ is the same for any Church numeral $\mathsf{n}$, we conclude from the previous theorem that $M\mathsf{n}$ normalizes to a Church numeral for every $n \in \mathbb{N}$, provably in $\mathbf{ID}_m$.

▶ **Theorem 28.** *Every representable function in* $\mathbf{F}^p_m$ *is provably total in* $\mathbf{ID}_m$*.*

▶ Remark. The above theorem, together with the converse direction, is already proved by Altenkirch and Coquand [5] for $n = 0$, and by Aehlig [2, 3] for an arbitrary $n \in \mathbb{N}$. The former proof uses a Heyting-valued computability predicate, while the latter consists of two steps: article [3] locally formalizes (weak) normalization (for terms of type $\boldsymbol{N}$) in a parameter-free system $\mathbf{HA}^2_{n+1,(1)}$ of second order Heyting arithmetic, and [2] gives a proof-theoretic reduction of the latter system to $\mathbf{ID}_n$. The reduction is done by encoding (recursive) infinitary derivations involving the $\Omega$-rule into natural numbers, and then applying the computability argument. Though closely related, our proof is more direct in that it circumvents use of an intermediate system like $\mathbf{HA}^2_{n+1,(1)}$ and first-order encoding of infinitary derivations. More importantly, we prove *strong* normalization for *all* terms explicitly, in contrast to *weak* normalization for *specific* terms.

**Acknowledgments.** This work was supported by KAKENHI 16K16690 and KAKENHI 25330013.

## References


1 A. Abel and T. Altenkirch. A predicative strong normalisation proof for a lambda-calculus with interleaving inductive types. In *Proceedings of TYPE'99*, pages 1–20, 1999.

2 K. Aehlig. Induction and inductive definitions in fragments of second order arithmetic. *Journal of Symbolic Logic*, 70:1087–1107, 2005.

3 K. Aehlig. Parameter-free polymorphic types. *Annals of Pure and Applied Logic*, 156:3–12, 2008.

4 R. Akiyoshi and G. Mints. An extension of the Omega-rule. *Archive for Mathematical logic*, 55(3):593–603, 2016.

5 T. Altenkirch and T. Coquand. A finitary subsystem of the polymorphic $\lambda$-calculus. *TLCA'01 Proceedings of the 5th international conference on Typed lambda calculi and applications*, pages 22–28. Springer, 2001.

6 Wilfried Buchholz. The $\Omega_{\mu+1}$-rule. In [8], pages 188–233, 1981.

7 W. Buchholz. Explaining the Gentzen-Takeuti reduction steps. *Archive for Mathematical Logic*, 40:255–272, 2001.

8 W. Buchholz, S. Feferman, W. Pohlers, and W. Sieg (editors). *Iterated Inductive Definitions and Subsystems of Analysis: Recent Proof-Theoretical Studies*, LNM 897, Springer, 1981.

9 W. Buchholz and K. Schütte. *Proof Theory of Impredicative Subsystems of Analysis*, Bibliopolis, 1988.

10 S. Feferman. What rests on what? The proof-theoretic analysis of mathematics. In *Proceedings of the 15th International Wittgenstein Symposium*, pages 141–171, 1993.

11 J.-Y. Girard. Une Extension de l'Interpretation de Gödel á l'Analyse, et son Application á l'Élimitation des Coupures dans l'Analyse et la Théorie des Types *Proceedings of the Second Scandinavian Logic Symposium.* Amsterdam, pages 63–92, 1971.

12 F. Joachimski and R. Matthes. Short proofs of normalization for the simply-typed lambda-calculus, permutative conversions and Gödel's T. *Archive for Mathematical Logic*, 42:59–87, 2003.

13 W. Tait. Intensional interpretations of functionals of finite type I. *Journal of Symbolic Logic*, 32(2):198–212, 1967.

14 W. Tait. Gödel's unpublished papers on foundations of mathematics. *Philosophia Mathematica*, 3(9):87–126, 2001.

15 G. Takeuti. Consistency proofs of subsystems of classical analysis. *The Annals of Mathematics*, 86(2):299–348, 1967.

16 G. Takeuti. *Proof Theory (2nd. ed.)*, North-Holland, 1987.

17 A.S. Troelstra. Introductory note to *1941. In *Kurt Gödel Collected Works, volume 3*, Oxford, pages 186–189, 1995.

18 D.T. van Daalen. *The Language Theory of Automath.* PhD thesis, Technische Universiteit Eindhoven, 1980.

19 F. van Raamsdonk and P. Severi. On Normalisation. *Technical Report CS-R9545*, CWI, 1995.

20 F. van Raamsdonk, P. Severi, M.H. Sorensen and H. Xi. Perpetual reductions in $\lambda$-calculus. *Information and Computation*, 149(2): 173–225, 1999.