Polynomial-time Algorithms from Ineffective Proofs

Paulo Oliva
BRICS[*], *University of Aarhus, Denmark*
pbo@brics.dk

Abstract

We present a constructive procedure for extracting polynomial-time realizers from ineffective proofs of Π^0_2-theorems in feasible analysis. By ineffective proof we mean a proof which involves the non-computational principle weak König's lemma WKL*, and by feasible analysis we mean Cook and Urquhart's system* CPV^ω *plus quantifier-free choice* QF-AC*. We shall also discuss the relation between the system* $\mathsf{CPV}^\omega + \mathsf{QF\text{-}AC}$ *and Ferreira's base theory for feasible analysis* BTFA*, for which Π^0_2-conservation of* WKL *has been non-constructively proven. This paper treats the case of weak König's lemma for trees defined by Π^0_1-formulas. Illustrating the applicability of* $\mathsf{CPV}^\omega + \mathsf{QF\text{-}AC}$ *extended with this form of weak König's lemma, we indicate how to formalize the proof of the Heine/Borel covering lemma in this system. The main techniques used in the paper are Gödel's functional interpretation and a novel form of binary bar recursion.*

1. Introduction

With the aim of capturing the notion of *feasibly constructive proof*, Stephen Cook [4] introduced in 1975 the equational system of arithmetic PV (polynomially verifiable) whose definable terms are polynomial-time computable. Later, Samuel Buss [2] developed the subsystem of classical arithmetic S^1_2 and showed that the provably recursive functions of his system are polynomial-time computable. Buss [3] also defined an intuitionistic version of S^1_2, called IS^1_2, and an intricate variant of Kleene realizability to prove that every Π^0_2-theorem of IS^1_2 has a polynomial-time computable realizer. Having as one of the motivations to simplify Buss' proof, Cook and Urquhart [5] defined systems both extending PV to higher types, obtaining PV^ω, and extending PV with intuitionistic and classical logic, obtaining IPV and CPV. A combination of those two extensions gives the systems IPV^ω and CPV^ω. Those systems have the same property of IS^1_2 that the provably recursive functions are polynomial-time computable. Cook and Urquhart then developed variants of Kreisel's modified realizability and Gödel's functional interpretation for the system IPV^ω. The latter via negative translation applies also to CPV^ω. Given a proof of a Π^0_2-theorem of IPV^ω or CPV^ω, these interpretations provide a simple procedure for extracting from this proof a polynomial-time algorithm realizing the theorem.

The main contribution of the present paper is to extend Cook and Urquhart's functional interpretation, via negative translation, of CPV^ω to include quantifier-free choice QF-AC and the non-computational principle weak König's lemma WKL (for Π^0_1-definable trees). The interpretation uses a novel form of binary bar recursion. We also show that the type one terms of the system IPV^ω extended with this new form of bar recursion are polynomial-time computable. This gives a procedure for extracting polynomial-time realizers from proofs involving weak König's lemma of Π^0_2-theorems over the basic theory $\mathsf{CPV}^\omega + \mathsf{QF\text{-}AC}$.

Weak König's lemma states that every infinite binary branching tree has an infinite path. This principle relies on the existence of non-computable functions, in the sense that it does not hold in a model where all functions are recursive. As it is well-known in Reverse Mathematics [20], WKL is actually equivalent[1] over RCA_0 to numerous mathematical principles, such as the existence and the attainment of the infimum by a continuous function on a closed interval and the Heine/Borel covering lemma. Nevertheless, Friedman showed (in an unpublished paper) that the Π^0_2-theorems of $\mathsf{RCA}_0 + \mathsf{WKL}$ are precisely the same as those of PRA, in the sense that if $\forall x \exists y A(x,y)$, for A quantifier-free, is provable in $\mathsf{RCA}_0 + \mathsf{WKL}$ then there exists a *primitive recursive program* h such that, $\mathsf{PRA} \vdash A(x, hx)$. In another words, $\mathsf{RCA}_0 + \mathsf{WKL}$ is Π^0_2-conservative over PRA.

Friedman's original proof of this Π^0_2-conservation result is based on non-constructive model-theoretic arguments. Therefore, it does not provide a procedure for extracting

* Basic Research in Computer Science, funded by the Danish National Research Foundation.

[1] The subsystem of second order arithmetic RCA_0 was first defined in [8], it contains the usual axioms for successor, addition and multiplication; induction restricted to Σ^0_1-formulas and comprehension for recursively defined sets.

the primitive recursive program h from a given proof of a Π^0_2-theorem in $\mathsf{RCA}_0 + \mathsf{WKL}$. Friedman's result was later extended by Harrington, who proved (also in an unpublished paper) Π^1_1-conservation of $\mathsf{RCA}_0 + \mathsf{WKL}$ over RCA_0. The first effective version of Friedman's result was given by Sieg [19] using cut-elimination, a Herbrand analysis and a simple form of Howard's majorizability for primitive recursive terms. In [12], a combination of Gödel's functional interpretation with Howard's hereditary majorizability for functionals in all finite types is developed to extract uniform bounds for $\forall\exists$-theorems in analysis from proofs based on various analytical principles including WKL. In particular, [12] yields effective forms of extensions of Friedman's WKL-conservation result to higher types (cf. also [1], Theorem 7.1.1).

In 1985, Sieg [18] proposed the problem of finding mathematically significant subsystems of analysis whose class of provably recursive functions consists only of *computationally feasible* ones. Fernando Ferreira took up the challenge and in [6] defined the system BTFA (Base Theory for Feasible Analysis) whose provably recursive functions are precisely the polynomial-time computable functions[2]. As done by Harrington for RCA_0, Ferreira then showed that by adding WKL (for bounded formulas Σ^b_∞) to BTFA one does not get any new Π^1_1-theorems. This shows a nice correspondence with respect to WKL between the system RCA_0, on the level of primitive recursion, and BTFA, on the level of polynomial-time. This correspondence can be expressed informally as

$$\frac{\mathsf{RCA}_0}{\mathsf{RCA}_0 + \mathsf{WKL}} \sim \frac{\mathsf{BTFA}}{\mathsf{BTFA} + \Sigma^b_\infty\text{-}\mathsf{WKL}}.$$

The congruity between the two sides of the equation goes even further. Ferreira's proof of Π^1_1-conservation, as the fore-mentioned Friedman's proof, is also based on nonconstructive model-theoretic arguments and does not give a procedure for extracting, from a proof

$$\mathsf{BTFA} + \Sigma^b_\infty\text{-}\mathsf{WKL} \vdash \forall x \exists y A(x,y),$$

where A is quantifier-free, a *polynomial-time function* h such that $A(x, hx)$ holds, for all x. We present here an effective procedure for extracting polynomial-time realizers from proofs of Π^0_2-theorems involving WKL in feasible analysis (here meaning $\mathsf{CPV}^\omega + \mathsf{QF\text{-}AC}$). It is important to note, however, that Ferreira proved conservation of WKL for trees defined by formulas of the kind $\forall z T(w, z)$, T being a *bounded formula*. This paper treats the case where T is a *quantifier-free formula*.

[2] Kohlenbach [13] also developed a subsystem of analysis (including WKL) whose Π^0_2-theorems have polynomial *bounds*, i.e. if $\forall x \exists y A(x,y)$, A quantifier-free, is a theorem of the system, then there exists effectively a polynomial $p(x)$ such that $\forall x \exists y \leq p(x) A(x,y)$.

The rest of the article is organized as follows. In Sections 2 and 3 we present the systems BTFA and $\mathsf{CPV}^\omega + \mathsf{QF\text{-}AC}$, in order to discuss the relation between them. The reader with knowledge on CPV^ω can start reading from Section 4 where we introduce the new form of bar recursion, which is going to be used in the interpretation of weak König's lemma for Π^0_1-definable trees, $\Pi^0_1\text{-}\mathsf{WKL}^\omega$. In Section 4 we also prove that this new bar recursion does not give rise to any new functions when added to IPV^ω. The functional interpretation of the negative translation of $\Pi^0_1\text{-}\mathsf{WKL}^\omega$ is given in Section 5. For illustrating the applicability of this Π^0_1-form of weak König's lemma, in Section 6 we indicate how to formalize the proof of Heine/Borel covering lemma in the system $\mathsf{CPV}^\omega + \mathsf{QF\text{-}AC} + \Pi^0_1\text{-}\mathsf{WKL}^\omega$.

A functional interpretation of the negative translation of weak König's lemma, using a different form of binary bar recursion, had already been given by Howard [9]. Howard's proof, however, does not carry through to the feasible setting under consideration since it is based on exponential search. We comment further on that in Section 7.

2 Preliminaries

In the following we shall assume some basic knowledge on negative translation and functional interpretation. For a smooth introduction to functional interpretation see [1]. We shall use Kuroda's variant of negative translation which places double negations after universal quantifiers and in front of the whole formula. As shown in [16], the different variations of negative translation are over intuitionistic logic equivalent. The negative translation of a formula A will be denoted by A^N.

The *finite types* are defined inductively as follows: $\mathbb{N}$ is a finite type, and if ρ and σ are finite types then $\rho \to \sigma$ is also a finite type. We shall write $\ldots : \rho$ to denote that term $\ldots$ has type ρ.

The two feasible subsystems of analysis discussed here, BTFA and $\mathsf{CPV}^\omega + \mathsf{QF\text{-}AC}$, have two main differences. Firstly, BTFA is based on second order logic, and therefore, has variables and quantifiers for sets, whereas, the theory $\mathsf{CPV}^\omega + \mathsf{QF\text{-}AC}$ is based on the language of functionals of all finite types, and therefore, has variables for each finite type. The second main difference is that the standard model of BTFA is based on finite 0-1 sequences $\mathbb{W}$, while $\mathsf{CPV}^\omega + \mathsf{QF\text{-}AC}$ has standard model based on the natural number $\mathbb{N}$ (which we shall confuse with the basic finite type). We shall in this paper define both theories and discuss briefly the relation between them.

In a feasible setting, where the length of the representation matters, it is often useful to work with 0-1 sequences as basic elements. Therefore, when dealing with CPV^ω we shall view natural numbers as finite sequences of 0-1, via their binary expansion. Given a number x we shall denote the i-th bit of the binary expansion of x by $x(i)$. We often write $x0$ instead of $2x$, and $x1$ instead of $2x+1$. In general, given a sequence of bits $b_n, \ldots, b_0 \in \{0,1\}$ (with $b_n = 1$) we shall write $b_n \ldots b_0$ for the natural number having such binary expansion. Moreover, we write 1^n for the sequence of n bits 1 and we use $|x|$ for the length of the binary expansion of x, i.e. $\lceil \log_2(x+1) \rceil$. Although the function $|\cdot|$ is not a basic symbol in either systems BTFA or CPV^ω, it is easily definable and we shall use it freely.

In Section 2.1 (on BTFA) we shall talk about three relations on binary words: $x \subseteq y$ for x being a prefix of y; $x \trianglelefteq y$ for $|x|$ being less than or equal to $|y|$, and $x \subseteq^* y$ saying that x is a subword of y, i.e. if there exists a z such that $zx \subseteq y$. When treating CPV^ω we use $x \leq y$ for x being a number smaller than or equal to y, and $x \preceq y$ for saying that the binary expansion of x is a prefix of the binary expansion of y. In both systems only the first relation is a primitive symbol, the others are definable relations. Based on those relations, in this paper the reader shall encounter three sorts of quantifiers:

- *unbounded quantifiers*: $Qx(\ldots)$,
- *bounded quantifiers*: $Qx \trianglelefteq t(\ldots)$ in BTFA and $Qx \leq t(\ldots)$ in CPV^ω, and
- *sharply bounded quantifiers*: $Qx \subseteq t(\ldots), Qx \subseteq^* t(\ldots)$ in BTFA and $Qx \preceq t(\ldots), Qx \leq |t|(\ldots)$ in CPV^ω.

Informally, bounded quantifiers correspond to an exponential search, while sharply bounded quantifiers correspond to linear or quadratic search. A formula is Π^0_1 (resp. Π^0_2) if it is of the form $\forall x A(x)$ (resp. $\forall x \exists y A(x,y)$), where A is a quantifier-free formula. While in stronger systems, such as RCA_0, a quantifier-free formula is one not containing unbounded quantifiers, in the feasible setting a *quantifier-free* formula is one containing only sharply bounded quantifiers.

Notice that, via paring, formulas of the kind $\forall x \exists y A(x,y)$, with A being quantifier-free, are as general as when A is a Σ^0_1 formula.

2.1 The system BTFA

Ferreira's system BTFA [6] has as basis the first order theory Σ^b_1-NIA, whose standard model is the set of finite strings over $\{0,1\}$ denoted by $\mathbb{W}$. The language of Σ^b_1-NIA contains symbols ϵ, 0 and 1, function symbols $x \frown y$ for the concatenation of x with y (we usually omit $\frown$ and just write xy), $x \times y$ for the concatenation of x with itself $|y|$ times, and a binary relation symbol $\subseteq$ for string prefix.

The class of *subword quantification-formulas* (sw.q.-formulas for short) is the smallest class of formulas closed under boolean operations and subword quantification, i.e. quantification of the form $Qx \subseteq^* t(\ldots)$, where the variable x does not occur in the term t. The class of *bounded formulas* Σ^b_∞ is the smallest class of formulas containing the sw.q.-formulas and closed under boolean operations and bounded quantification, i.e. quantification of the form $Qx \trianglelefteq t(\ldots)$, where the variable x does not occur in the term t. The class of formulas of the form $\exists x \trianglelefteq t\, A$, A being a sw.q.-formula, is denoted by Σ^b_1.

Besides fourteen basic axioms governing the behaviour of the non-logical symbols, Σ^b_1-NIA contains the *induction scheme* Σ^b_1-IND

$$A(\epsilon) \wedge \forall x(A(x) \rightarrow A(x0) \wedge A(x1)) \rightarrow \forall x A(x),$$

for $A \in \Sigma^b_1$. The theory Σ^b_1-NIA is equivalent, in a sense that could be made precise, to Buss' theory S^1_2 (cf. [2]), and therefore, has the property that every Π^0_2-theorem has a polynomial-time realizer. The second order theory BTFA is obtained from Σ^b_1-NIA by adding the *bounded collection principle* Σ^b_∞-BC

$$\forall x \trianglelefteq t \exists y A(x,y) \rightarrow \exists z \forall x \trianglelefteq t \exists y \trianglelefteq z A(x,y),$$

for $A \in \Sigma^b_\infty$, and *comprehension* Δ^0_1-CA

$$\begin{cases} \forall x(\exists y A(x,y) \leftrightarrow \forall z \neg B(x,z)) \rightarrow \\ \qquad \exists \mathcal{S} \forall x(x \in \mathcal{S} \leftrightarrow \exists y A(x,y)), \end{cases}$$

for $A, B \in \Sigma^b_1$.

Lemma 2.1 ([6]) *Let A be a bounded formula. If*

$$\mathsf{BTFA} \vdash \forall x \exists y A(x,y)$$

then Σ^b_1-NIA $\vdash \forall x \exists y A(x,y)$.

In the feasible setting of second order arithmetic $\mathsf{WKL}(T)$ is formulated as

$$\mathsf{Tree}_\infty(T) \rightarrow \exists \mathcal{S}(\mathsf{Path}_\infty(\mathcal{S}) \wedge \forall w(w \in \mathcal{S} \rightarrow T(w)),$$

where $\mathcal{S}$ is a set variable, $\mathsf{Tree}_\infty(T)$ is defined as

$$\begin{cases} \forall w, v(T(w) \wedge v \subseteq w \rightarrow T(v)) \wedge \\ \forall y \exists w(|w| = |y| \wedge T(w)), \end{cases}$$

and $\mathsf{Path}_\infty(\mathcal{S})$ as

$$\mathsf{Tree}_\infty(w \in \mathcal{S}) \wedge \forall x, y \in \mathcal{S}(x \subseteq y \vee y \subseteq x).$$

If Φ is a class of formulas, we shall denote by Φ-WKL the principle $\mathsf{WKL}(T)$ for T restricted to the class Φ.

Using non-constructive model-theoretic arguments, Ferreira showed that BTFA extended with Σ^b_∞-WKL has the same $\forall\exists\Sigma^b_\infty$-theorems as Σ^b_1-NIA.

Theorem 2.2 ([6]) *Let A be a bounded formula. If*

$$\mathsf{BTFA} + \Sigma^b_\infty\text{-}\mathsf{WKL} \vdash \forall x \exists y A(x,y)$$

then Σ^b_1-NIA $\vdash \forall x \exists y A(x,y)$.

As a corollary, one obtains that the provably recursive functions of BTFA + Σ^b_∞-WKL are polynomial-time computable.

Corollary 2.3 *Let A be quantifier-free. If*

$$\mathsf{BTFA} + \Sigma^b_\infty\text{-}\mathsf{WKL} \vdash \forall x \exists y A(x,y)$$

then there exists a polynomial-time computable function h such that $A(x,hx)$ holds, for all x.

The main result of this paper is an effective version of Corollary 2.3 for the system $\mathsf{CPV}^\omega + \mathsf{QF\text{-}AC} + \Pi^0_1\text{-}\mathsf{WKL}^\omega$. In the following section we present the system CPV^ω + QF-AC and we explain how it relates to BTFA.

3 The system $\mathsf{CPV}^\omega + \mathsf{QF\text{-}AC}$

The system CPV^ω [5] builds on the equational calculus PV^ω. The language of PV^ω contains a single constant symbol 0, for the number zero. The function symbols of PV^ω, with their intended interpretation, are

- $s_0(x), s_1(x)$ extends x to the right with the bit 0 and 1, respectively, i.e. $s_0(x) = 2x$ and $s_1(x) = 2x+1$;
- $\mathsf{Parity}(x)$ returns 0 if the rightmost bit of x is 0;
- $\lfloor \frac{1}{2}x \rfloor$ chops off the rightmost bit of x;
- $\mathsf{Chop}(x,y)$ chops off $|y|$ bits from the right of x;
- $\mathsf{Pad}(x,y)$ appends $|y|$ zero bits to the right of x;
- $\mathsf{Smash}(x,y)$ returns the bit '1' followed by $|x|$ times $|y|$ zeros.
- $\mathsf{Cond}(x,y,z)$ returns y if x is zero and z otherwise.

PV^ω has infinitely many variables for each finite type. Unless stated otherwise, the variables x,y,z and w shall have type $\mathbb{N}$. PV^ω has also a recursor $\mathcal{R}$ of type

$$\mathbb{N} \to (\mathbb{N} \to \mathbb{N} \to \mathbb{N}) \to (\mathbb{N} \to \mathbb{N}) \to \mathbb{N} \to \mathbb{N}.$$

The terms of PV^ω are formed out of variables and function symbols as usually done in the typed λ-calculus. PV^ω contains only the predicate symbol $=$ for the basic type $\mathbb{N}$. The formulas of PV^ω consists of all equations $s = u$, where s and u are terms of type $\mathbb{N}$. The axioms of PV^ω are the defining equations for the function symbols listed above, the axiom for *higher type limited recursion on notation* HTLRN

$$\mathcal{R}(x,h,g,y) = \begin{cases} x & \text{if } y = 0 \\ g(y) & \text{if } |t| > |g(y)| \\ t & \text{otherwise,} \end{cases}$$

where t abbreviates $h(y, \mathcal{R}(x,h,g,\lfloor \frac{1}{2}y \rfloor))$, and further axioms for normalising λ-terms. Moreover, PV^ω has four rules $R1^\omega$–$R4^\omega$ governing the behaviour of the equality predicate and a rule for induction on notation (for further details see [5]).

The system IPV^ω is defined as follows. The terms of IPV^ω are those of PV^ω. The predicate symbols of IPV^ω are $=$ and $\leq$, for type $\mathbb{N}$ only. The atomic formulas are $s = u$ and $s \leq u$, where s and u are terms of type $\mathbb{N}$. The formulas of IPV^ω are built out of atomic formulas via logical connectives and quantifiers for each finite type. The logical axioms of IPV^ω are the usual ones for many-sorted intuitionistic predicate logic. The non-logical axioms of IPV^ω consist of all the theorems of PV^ω plus[3]

- $x \leq y \leftrightarrow \mathsf{Lessequ}(x,y) = 0$,
- $x = s_0\lfloor \frac{1}{2}x \rfloor \vee x = s_1 \lfloor \frac{1}{2}x \rfloor$,
- $\mathsf{Cond}(x,a,b) = c \leftrightarrow$
$$(x = 0 \wedge a = c) \vee (\neg(x=0) \wedge b = c),$$

and the *induction axiom* $\mathsf{PIND}^\omega(A)$

$$(A(0) \wedge \forall x(A(\lfloor \frac{1}{2}x \rfloor) \to A(x))) \to \forall x A(x),$$

where A is of the form $\exists y \leq t\,(s = u)$ and all the free-variables of t have type $\mathbb{N}$. At this point we note that in IPV^ω, for each quantifier-free formulas $A(x)$ one can build a term s such that $\mathsf{IPV}^\omega \vdash A(x) \leftrightarrow sx = 0$.

The system CPV^ω is obtained from IPV^ω by adding all instances of the law of excluded middle $A \vee \neg A$.

In the following we shall make use of two further logical principles, namely, the *scheme of quantifier-free choice*

$$\mathsf{QF\text{-}AC} \quad : \quad \forall x \exists y A(x,y) \to \exists h \forall x A(x,hx),$$

and *Markov's principle*

$$\mathsf{MP}^\omega \quad : \quad \neg\neg\exists x A(x) \to \exists x A(x),$$

where in both cases A is a quantifier-free formula, and in the case of Markov's principle the variable x can be of arbitrary type. We shall use Markov's principle in connection with the negative translation of the system CPV^ω + QF-AC.

As shown in [5], the system CPV^ω contains a set of *feasible coding functions*. Therefore, one can for instance replace a sequence of quantifiers of the same kind by a singe quantifier. For simplicity, we shall state results without making it explicit that tuples of quantifiers are allowed.

The next lemma is a simple extension of the negative translation of CPV^ω in $\mathsf{IPV}^\omega + \mathsf{MP}^\omega$, given in [5] (Lemma 10.3), to include quantifier-free choice.

[3] $\mathsf{Lessequ}(x,y)$ is a definable function of PV^ω which represents the characteristic function of the inequality predicate.

Lemma 3.1 *The theory* $\mathsf{CPV}^\omega + \mathsf{QF\text{-}AC}$ *has a negative translation in* $\mathsf{IPV}^\omega + \mathsf{MP}^\omega + \mathsf{QF\text{-}AC}$.

Since the functional interpretation of MP^ω and $\mathsf{QF\text{-}AC}$ are trivial, we obtain the following extension of Theorem 10.4 of [5].

Lemma 3.2 *Let A be a quantifier-free formula. If*

$$\mathsf{CPV}^\omega + \mathsf{QF\text{-}AC} \vdash \forall x \exists y A(x, y),$$

then from this proof one can extract a closed term t of type $\mathbb{N} \to \mathbb{N}$ of IPV^ω *such that* $\mathsf{IPV}^\omega \vdash \forall x A(x, tx)$.

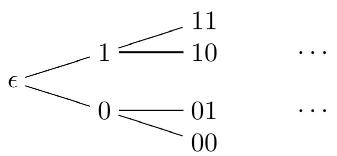
Moreover, since the terms of type $\mathbb{N} \to \mathbb{N}$ of IPV^ω denote polynomial-time computable functions, we get a procedure from extracting polynomial-time realizers from proofs of Π^0_2-theorems in $\mathsf{CPV}^\omega + \mathsf{QF\text{-}AC}$.

3.1 The system $\mathsf{CPV}^\omega + \mathsf{QF\text{-}AC} + \Pi^0_1\text{-}\mathsf{WKL}^\omega$

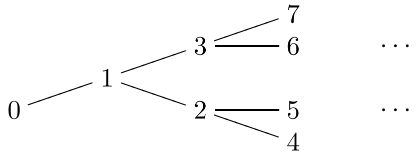
As we have mentioned, the theory BTFA has as standard model the set of finite 0-1 sequences $\mathbb{W}$. This setting is particularly convenient for working with weak König's lemma, since the prefix relation $\subseteq$ is one of the primitives of the system. The system CPV^ω, however, has the natural numbers as its standard model. Therefore, based on the bijective feasible mapping η (which assigns 0 to ϵ and positive numbers to their binary expansion) between natural number and the set of strings $1\{0,1\}^* \cup \{\epsilon\}$, we define the prefix relation $\preceq$ in CPV^ω as

$$x \preceq y :\equiv \eta(x) \subseteq \eta(y),$$

where x, y are numbers. The prefix relation $\subseteq$ in $\mathbb{W}$ is a partial order which can be depicted as



Notice that the binary words of the form $0\{0,1\}^*$ are not valid binary representation of any natural number. Therefore, under the mapping η, in $\mathbb{N}$ the prefix relation $\preceq$ gives rise to the partial order



A predicate T on numbers is said to define a *tree* if it is closed under the prefix relation $\preceq$, i.e. whenever $T(w)$ holds and $v \preceq w$ then $T(v)$ also holds. Formally

$$\mathsf{Tree}(T) :\equiv \forall w, v(T(w) \wedge v \preceq w \to T(v)).$$

A function $f : \mathbb{N} \to \mathbb{N}$ is an *infinite path* if $f(y) \in \{0,1\}$, for all y, and $f(0) = 1$, i.e.

$$\mathsf{Path}(f) :\equiv \forall y(f(y) \leq 1) \wedge f(0) = 1.$$

We require that $f(0) = 1$ since we shall view initial segments of f as numbers, and finite 0-1 sequences of the form $0\{0,1\}^*$ do not correspond to valid natural numbers.

We say that an infinite path f *belongs to a tree* T if every initial segment of f belongs to T, i.e. $\forall y T(\overline{f}y)$, where for a given path f, the function $\overline{f} : \mathbb{N} \to \mathbb{N}$ is defined as

$$\overline{f}(y) = \begin{cases} 0 & \text{if } y = 0 \\ f(0)f(1)\ldots f(1^{|y|-1}) & \text{otherwise.} \end{cases}$$

Therefore, in the feasible setting of finite types weak König's lemma for a predicate T is expressed as

$$\begin{cases} \mathsf{Tree}(T) \wedge \forall y \exists w(|w| = |y| \wedge T(w)) \to \\ \qquad \exists f(\mathsf{Path}(f) \wedge \forall y T(\overline{f}(y))). \end{cases}$$

The predicates Tree and Path can actually be omitted via the use of feasible operations (cf. [12]). The transformation

$$T^t(w) :\equiv \forall v \preceq w T(v),$$

makes an arbitrary predicate T into a tree T^t. It is easy to show that if $T(w)$ is already a tree, then $T^t(w)$ holds iff $T(w)$ holds. Moreover, via the transformation

$$f^p(y) = \begin{cases} 1 & \text{if } y = 0 \\ \mathsf{Parity}(f(y)) & \text{otherwise,} \end{cases}$$

we can make an arbitrary function $f : \mathbb{N} \to \mathbb{N}$ into an infinite path f^p. Again, if f is already a path then $f^p(y) = f(y)$, for all y. Using these transformations, weak König's lemma (in the language of higher types) $\mathsf{WKL}^\omega(T)$ can be stated as

$$\forall y \exists w(|w| = |y| \wedge T^t(w)) \to \exists f \forall y T^t(\overline{f^p}(y)). \quad (1)$$

Since the formulation of weak König's lemma changes from the setting of second order arithmetic to the setting of finite types, we use the superscript ω for the latter. The two definitions, however, can be easily shown to be equivalent, i.e. one can define feasible functionals which given the characteristic function of a path $\mathcal{S}$ produces a path f, and vice-versa.

The transformation $f \mapsto f^p$ used above actually allows for quantification over infinite paths. Therefore, in the following we take f as a meta-variable for infinite paths, and omit the transformation f^p.

In order to carry out the functional interpretation of $\mathsf{WKL}^\omega(T)$ it will be particularly convenient to treat it as an axiom (rather than an axiom schema)

$$\mathsf{WKL}^\omega : \forall g \, \mathsf{WKL}^\omega(gw = 0).$$

The Π_1^0-form of weak König's lemma is then stated as

$$\Pi_1^0\text{-}\mathsf{WKL}^\omega \ : \ \forall g\, \mathsf{WKL}^\omega(\forall z(gwz = 0)).$$

We shall use the superscript ω to differentiate between Ferreira's and our formulation of weak König's lemma.

3.2 BTFA versus $\mathsf{CPV}^\omega + \mathsf{QF\text{-}AC}$

In the system CPV^ω, using limited recursion on notation, sharply bounded quantifiers can be absorbed by quantifier-free matrices. Therefore, for A quantifier-free, the subword quantification of BTFA (which is definable in CPV^ω) can be also absorbed by two applications of recursion, since $Qx \subseteq^* tA(x)$ can be rewritten as

$$Qy \preceq tQx \preceq yA(\mathsf{Interv}(x, y, t)),$$

where the feasible function $\mathsf{Interv}(x, y, z)$ returns all the bits of z between $|x|$ and $|y|$. In this way, the sw.q.-formulas of BTFA correspond to quantifier-free formulas of CPV^ω. The predicate $\trianglelefteq$ can be easily defined using $\leq$, so that the formulas Σ_1^b of BTFA correspond to formulas of the form $\exists x \leq t(s = u)$ in CPV^ω.

Moreover, the system $\mathsf{CPV}^\omega + \mathsf{QF\text{-}AC}$ proves comprehension for Δ_1^0-formulas

$$\forall x(A(x) \leftrightarrow \neg B(x)) \rightarrow \exists h \forall x(hx = 0 \leftrightarrow A(x)),$$

where $A, B \in \Sigma_1^0$, which corresponds precisely to Δ_1^0-CA of BTFA, and the following weaker form of bounded collection

$$\forall x \preceq t \exists y A(x, y) \rightarrow \exists z \forall x \preceq t \exists y \leq z A(x, y),$$

for $A \in \Sigma_1^0$. It does not seem to prove, however, the more general Σ_∞^b-BC.

One advantage of $\mathsf{CPV}^\omega + \mathsf{QF\text{-}AC}$ over BTFA is the availability of higher order functionals. In this way one can talk about transformation between numbers (objects of type $\mathbb{N}$), real numbers and continuous functions (objects of type $\mathbb{N} \rightarrow \mathbb{N}$) in a straightforward way, as opposed to using encodings with sets.

In Section 6 we shall illustrate how the system $\mathsf{CPV}^\omega +$ $\mathsf{QF\text{-}AC}$ can be used for practical applications by sketching the proof of Heine/Borel theorem in $\mathsf{CPV}^\omega + \mathsf{QF\text{-}AC} +$ Π_1^0-WKL^ω.

4 A simple form of (binary) bar recursion

Howard showed in [9] that a simpler form of Spector's [21] bar recursion was sufficient for giving a functional interpretation of the negative translation of weak König's lemma. Howard's proof, however, does not seem to be suitable for weak theories such as IPV^ω, since it makes essential use of exponential search (cf. Section 7). For our conservation result we shall add to the language of IPV^ω the constant (of binary bar recursion) $\mathcal{B}$ having type

$$((\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}) \rightarrow (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N} \rightarrow \mathbb{N},$$

and the axiom

$$\mathcal{B}(Y, W, z) = \begin{cases} z & \text{if } |Y\hat{w}_z| \leq |w_z| \\ & \text{or } |w_z| \neq |z| \\ \mathcal{B}(Y, W, z1) & \text{otherwise,} \end{cases} \tag{2}$$

where w_z abbreviates Wz, and for a given $w \in \mathbb{N}$, the function $\hat{w} : \mathbb{N} \rightarrow \{0, 1\}$ is defined as

$$\hat{w}(y) := \begin{cases} w(|y|) & \text{if } |y| < |w| \\ 0 & \text{otherwise.} \end{cases}$$

The function $\hat{w}$ denotes the infinite 0-1 sequence obtained by extending the binary expansion of w with 0's. In order to make sure that $\hat{w}$ always represents an infinite path (as defined in Section 3.1), we need to consider the particular case $w = 0$, since $\hat{0}(0) = 0$. Therefore, we change slightly the definition of $\hat{w}$ and set $\hat{0} = \hat{1}$. It is important to note, moreover, that for the function $\hat{w} : \mathbb{N} \rightarrow \{0, 1\}$ only the length of the argument is considered. This shall often be used since in a feasible setting functions should be computed in polynomial-time on the length of the input. In fact, in those cases it is more convenient to use the tally part of $\mathbb{N}$ instead. We abstain from that in order to keep the basic setup of CPV^ω unchanged.

The main result of this paper is based on the fact that IPV^ω is closed under the "rule version" of (2), i.e. if Ψ is a closed term of type $\mathbb{N} \rightarrow (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ and Φ a closed term of type $\mathbb{N} \rightarrow \mathbb{N} \rightarrow \mathbb{N}$ then there exists a closed term $t : \mathbb{N} \rightarrow \mathbb{N}$ such that $tx = \mathcal{B}(\Psi x, \Phi x, 0)$, for all x. In fact, even 0-1 oracles are allowed (cf. Lemma 4.4).

Remark 4.1 *Note that the functional $\mathcal{B}$ only applies the first argument Y to 0-1 functions. Therefore, if Y_1 and Y_2 coincide on all 0-1 functions then $\mathcal{B}(Y_1, W, z) = \mathcal{B}(Y_2, W, z)$, for all W and z.*

Notice that the functional $\mathcal{B}(Y, W, z)$ can also be viewed as the unbounded search

$$\min y \approx z1^n \, (|Y\hat{w}_y| \leq |w_y| \vee |w_y| \neq |y|), \tag{3}$$

where w_y abbreviates Wy and $y \approx z1^n$ means that y has the same binary expansion as z followed by a finite number of ones. The functional $\mathcal{B}$ has a flavour of bar recursion since the sequences w_z, on the "hat transformation" of which the functional Y is applied, gets longer and longer as the recursion progresses.

For justifying this new form of binary bar recursion (2) (or, equivalently, for bounding the search (3)) we can, for instance, assume *boundedness of functionals of type* $(\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$ *on 0-1 functions*

$$\mathsf{BND} \ : \ \forall Y \exists u \forall \alpha(\forall v(\alpha(v) \leq 1) \rightarrow |Y\alpha| \leq |u|),$$

which is a consequence of uniform continuity for functionals $Y : (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ on the Cantor space. The idea is that, since $|z|$ keeps increasing in the recursion (2), either $|w_z| \neq |z|$ for some z, or the length of w_z also increases, and eventually reaches the length of the bound u. The condition $|Y\hat{w}_z| \leq |w_z|$ is then satisfied. We shall need BND in the verification of our interpretation of weak König's lemma (cf. Theorem 5.1).

For the rest of this section the variable x should be viewed as a sequence of variables of type $\mathbb{N}$ and α as a sequence of variables of type $\mathbb{N} \to \mathbb{N}$.

Lemma 4.2 ([10], Lemma 5.4) *For any closed term Ψ of type $\mathbb{N} \to (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ of IPV^ω there exist constants c_1 and c_2 such that for any x and 0-1 functions α we have $|\Psi x\alpha| \leq |x|^{c_1} + c_2$.*

Using Lemma 4.2 one can show that IPV^ω is closed under the "rule version" of (2).

Lemma 4.3 *Let Ψ be a closed term of type $\mathbb{N} \to (\mathbb{N} \to \mathbb{N}) \to (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ and Φ a closed term of type $\mathbb{N} \to (\mathbb{N} \to \mathbb{N}) \to \mathbb{N} \to \mathbb{N}$ of IPV^ω. Then, there exists a closed term $t : \mathbb{N} \to (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ such that for all x and for all 0-1 functions α, $tx\alpha = \mathcal{B}(\Psi x\alpha, \Phi x\alpha, 0)$.*

Proof. Let Ψ and Φ be fixed. We shall define t by limited recursion on notation. Let c_1 and c_2 be such that (cf. Lemma 4.2) for all x and 0-1 valued functions α and β, $|\Psi x\alpha\beta| \leq |x|^{c_1} + c_2$. For a given x, let d_x denote the number having binary expansion $1^{|x|^{c_1}+c_2}$, then $|d_x| = |x|^{c_1} + c_2$. We then define two functions

$$h_{x,\alpha}(y, z) := \begin{cases} v & \text{if } |\Psi x\alpha\hat{w}_v| \leq |w_v| \text{ or } |w_v| \neq v \\ z & \text{otherwise,} \end{cases}$$

where v abbreviates $\mathsf{Chop}(d_x, y)$ and w_v abbreviates $\Phi x\alpha v$; and $g_x(y) := d_x$, i.e. g_x is a constant function with value d_x. Finally, we define $tx\alpha := \mathcal{R}(0, h_{x,\alpha}, g_x, d_x)$. $\Box$

The following lemma shows that arbitrary terms of type $\mathbb{N} \to (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ (on 0-1 functions) of $\mathcal{L}(\mathsf{IPV}^\omega) + \{\mathcal{B}\}$ denote polynomial-time computable functions with boolean oracles.

Lemma 4.4 *Let $t[x, \alpha]$ be a term of $\mathcal{L}(\mathsf{IPV}^\omega) \cup \{\mathcal{B}\}$ of type $\mathbb{N}$, having as only free-variables x and α, such that (for simplicity) $\mathcal{B}$ is always applied to zero on the third argument. Then, there exists a polynomial-time computable function h (with 0-1 oracle) such that for all input x and for all 0-1 oracles α, $h(x, \alpha) = t[x, \alpha]$.*

Proof. The proof follows closely the normalisation argument given in the proof of Proposition 4.2 in [14]. In the following we say *polynomial-time computable* for *polynomial-time computable with 0-1 oracle*. We start by carrying out all possible logical reductions on the term $\lambda x, \alpha.t[x, \alpha]$. We get a term $\lambda x, \alpha.t_1[x, \alpha]$ such that $t_1[x, \alpha]$ is of the form:

- 0 or x_i (x_i in the tuple x). We are done.
- $g(t_2[x, \alpha])$, where g is either one of α or a function symbol of IPV^ω. By induction there exists a polynomial-time computable h_2 such that for all inputs x and 0-1 oracles α, $h_2(x, \alpha) = t_2[x, \alpha]$. Hence, for all inputs x and 0-1 oracles α, $h(x, \alpha) := g(h_2(x, \alpha))$ does the job.
- $\mathcal{R}(t_2[x, \alpha], t_3[x, \alpha], t_4[x, \alpha], t_5[x, \alpha])$. The terms $t_2[x, \alpha]$ and $t_5[x, \alpha]$ are again type $\mathbb{N}$, and by induction there are polynomial-time computable functions $h_2(x, \alpha)$ and $h_5(x, \alpha)$ which coincide with $t_2[x, \alpha]$ and $t_5[x, \alpha]$ on all inputs x and 0-1 oracles α. The terms $t_3[x, \alpha]$ and $t_4[x, \alpha]$ are of type $\mathbb{N} \to \mathbb{N} \to \mathbb{N}$ and $\mathbb{N} \to \mathbb{N}$ respectively. We therefore add an extra variables y and z to bring them to type $\mathbb{N}$. By induction there are polynomial-time computable functions $h_3(x, y, z, \alpha)$ and $h_4(x, y, \alpha)$ which coincide with $t_3[x, \alpha]yz$ and $t_4[x, \alpha]y$ on all inputs x, y, z and 0-1 oracles α. Then, for all inputs x and 0-1 oracles α the polynomial-time computable function

  $$h(x, \alpha) := \mathcal{R}(h_2x\alpha, \lambda y, z.h_3xyz\alpha, \lambda y.h_4xy\alpha, h_5x\alpha)$$

  does the job.
- $\mathcal{B}(\Psi_2[x, \alpha], t_3[x, \alpha], 0)$. The term $\Psi_2[x, \alpha]$ is of type $(\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$. Let β be a variables of type $\mathbb{N} \to \mathbb{N}$. By induction there exists a polynomial-time computable h_2 such that for all inputs x and 0-1 oracles α, β, $h_2(x, \alpha, \beta) = \Psi_2[x, \alpha]\beta$. The term $t_3[x, \alpha]$ is of type $\mathbb{N} \to \mathbb{N}$. Adding an extra variable to bring it to type $\mathbb{N}$ we obtain, by induction hypothesis, that there exists a polynomial-time computable h_3 such that for all inputs x, y and 0-1 oracles $\alpha, h_3(x, y, \alpha) = t_3[x, \alpha]y$. By Lemma 4.3 and Remark 4.1, there exists a polynomial-time computable h such that for all inputs x and 0-1 oracles α

  $$\begin{aligned} h(x, \alpha) &= \mathcal{B}(\lambda\beta.h_2(x, \alpha, \beta), \lambda y.h_3(x, y, \alpha), 0) \\ &= \mathcal{B}(\Psi_2[x, \alpha], t_3[x, \alpha], 0) \quad \Box \end{aligned}$$

5 Interpreting Π^0_1-WKL^ω

We shall now present the functional interpretation (via negative translation) of $\mathsf{CPV}^\omega + \mathsf{QF\text{-}AC} + \Pi^0_1\text{-}\mathsf{WKL}^\omega$ in the system IPV^ω extended with a constant symbol $\mathcal{B}$, BND and the axiom (2).

Theorem 5.1 *The theory $\mathsf{CPV}^\omega + \mathsf{QF\text{-}AC} + \Pi^0_1\text{-}\mathsf{WKL}^\omega$ has a functional interpretation (via negative translation) in $\mathsf{IPV}^\omega + \mathsf{BND} + (2)$.*

Proof. By Lemma 3.1, we just need to show that

$$\mathsf{IPV}^\omega + \mathsf{MP}^\omega + \mathsf{QF\text{-}AC} + (\forall g\, \mathsf{WKL}^\omega(\forall z(gwz = 0)))^N$$

has a functional interpretation in $\mathsf{IPV}^\omega + \mathsf{BND} + (2)$. The functional interpretations of MP^ω and QF-AC are trivial. Let $T(w, z)$ abbreviate $\forall v \preceq w(gvz = 0)$. The negative translation of $\forall g\, \mathsf{WKL}^\omega(\forall z(gwz = 0))$ gives

$$\begin{cases} \neg\neg\forall g\neg\neg(\forall y\neg\neg\exists w(|w| = |y| \wedge \forall z\neg\neg T(w, z)) \to \\ \qquad\qquad \exists f \forall y, z\neg\neg T(\overline{f}(y), z)), \end{cases}$$

which is equivalent to

$$\begin{cases} \forall g(\forall y\neg\neg\exists w(|w| = |y| \wedge \forall z T(w, z)) \to \\ \qquad\qquad \neg\neg\exists f \forall y, z T(\overline{f}(y), z)), \end{cases}$$

Since we shall give realizers independently of the tree g we henceforth omit the quantifier over g. Then

$$\begin{cases} \forall y\neg\neg\exists w(|w| = |y| \wedge \forall z T(w, z)) \to \\ \qquad\qquad \neg\neg\exists f \forall y, z T(\overline{f}(y), z), \end{cases}$$

has the functional interpretation (in three steps)

$$\begin{cases} \forall y, h\exists w(|w| = |y| \wedge T(w, hw)) \to \\ \qquad\qquad \forall Y, Z\exists f T(\overline{f}(Yf), Zf), \end{cases}$$

$$\begin{cases} \exists W \forall y, h(|Wyh| = |y| \wedge T(Wyh, h(Wyh))) \to \\ \qquad\qquad \forall Y, Z\exists f T(\overline{f}(Yf), Zf), \end{cases}$$

$$\begin{cases} \forall W, Y, Z\exists y, h, f \\ \qquad \big(|Wyh| = |y| \wedge T(Wyh, h(Wyh)) \to \\ \qquad\qquad T(\overline{f}(Yf), Zf)\big). \end{cases}$$

Uniformly in W, Y, Z we produce y, h and f satisfying

$$|Wyh| = |y| \wedge T(Wyh, h(Wyh)) \to T(\overline{f}(Yf), Zf).$$

Define $\boxed{h(w) := Z(\hat{w})}$ and let w_y abbreviate Wyh. Now, we need to produce y and f satisfying

$$|w_y| = |y| \wedge T(w_y, Z(\hat{w}_y)) \to T(\overline{f}(Yf), Zf).$$

Define $\boxed{y := \mathcal{B}(Y, \lambda y.Wyh, 0)}$. By BND one can prove that

$$|Y\hat{w}_y| \leq |w_y| \vee |w_y| \neq |y|.$$

Finally, define $\boxed{f := \hat{w}_y}$. Then, assuming $|w_y| = |y|$, we have $\overline{f}(Yf) \preceq w_y$, and

$$T(w_y, Zf) \to T(\overline{f}(Yf), Zf)$$

follows from the fact that T is a tree. □

Combined with Lemma 4.4, Theorem 5.1 gives an effective procedure from extracting polynomial-time algorithms from WKL-proofs of Π^0_2-theorems in feasible analysis.

Corollary 5.2 *Let A be a quantifier-free formula. From a proof of $\forall x\exists y A(x, y)$ in the system $\mathsf{CPV}^\omega + \mathsf{QF\text{-}AC} + \Pi^0_1\text{-}\mathsf{WKL}^\omega$ one can extract a polynomial-time computable function h such that for all x, $A(x, hx)$ is true.*

Proof. Via negative translation and functional interpretation one can extract a term t of $\mathcal{L}(\mathsf{IPV}^\omega) + \mathcal{B}$ such that $\mathsf{IPV}^\omega + \mathsf{BND} + (2) \vdash \forall x A(x, tx)$. By Lemma 4.4 the recursor $\mathcal{B}$ can be eliminated from t, giving rise to a term h of $\mathcal{L}(\mathsf{IPV}^\omega)$ such that $\mathsf{IPV}^\omega + \mathsf{BND} + (2) \vdash \forall x A(x, hx)$. Scarpellini's [17] type structure of all continuous set-theoretical functionals $\mathcal{C}$ is a model of $\mathsf{IPV}^\omega + \mathsf{BND} + (2)$. Therefore, since $\mathcal{C}$ coincides with the full type structure in the types zero and one, the conclusion of the corollary follows. □

We have used Lemma 4.4 in the above proof in order to verify that the term t denotes a polynomial-time computable function. Notice, however, that in the present context the binary bar recursion $\mathcal{B}$ has the same computational behaviour as the limited recursion on notation by which it gets replaced. Therefore, for practical applications, when one is only interested in obtaining a polynomial-time computable realizer for the given Π^0_2-theorem, we can directly use the program t, avoiding the normalization procedure used in Lemma 4.4.

We can strengthen Corollary 5.2, by noticing that Lemma 4.4 holds even for terms t which have 0-1 oracle variables.

Corollary 5.3 *Let A be a quantifier-free formula. From a proof of $\forall\alpha\forall x\exists y A(\alpha, x, y)$ in the system $\mathsf{CPV}^\omega + \mathsf{QF\text{-}AC} + \Pi^0_1\text{-}\mathsf{WKL}^\omega$ one can extract a polynomial-time computable function (with 0-1 oracle) h such that for all 0-1 oracles α and input x, $A(\alpha, x, h\alpha x)$ holds.*

Notice that, in order to turn the above results into *conservation results*, one needs furthermore to show that $\mathsf{IPV}^\omega + \mathsf{BND} + (2)$ is Π^0_2-conservative over IPV^ω.

6 The Heine/Borel covering lemma

In this section we indicate how to formalize in $\mathsf{CPV}^\omega + \mathsf{QF\text{-}AC} + \Pi^0_1\text{-}\mathsf{WKL}^\omega$ the proof of the Heine/Borel covering lemma. Our formalization follows closely the ones given in [7] (Theorem 1) and [20] (Lemma IV.1.1).

In the system CPV^ω we shall represent the *tally part* of $\mathbb{N}$ (to be used as unary numbers) as $\mathbb{T}$. Those are natural numbers having binary expansion in the form 1^n. In the following we use variable i, j and n to range over $\mathbb{T}$, and p, q to range over $\mathbb{Q}$. *Real numbers* $\mathbb{R}$ are represented via functions $\psi : \mathbb{T} \to \mathbb{Q}$ satisfying

$$\forall i, j(i \leq_{\mathbb{T}} j \to \mathsf{Abs}(\psi(i) -_{\mathbb{Q}} \psi(j)) \leq 2^{-i}),$$

where $\mathsf{Abs}(q)$ returns the absolute value of a rational number. A real number ψ_1 is said to be *smaller* than ψ_2, written $\psi_1 <_{\mathbb{R}} \psi_2$, if

$$\exists i(\psi_1(i +_{\mathbb{T}} 1) + 2^{-i} <_{\mathbb{Q}} \psi_2(i +_{\mathbb{T}} 1)).$$

The Heine/Borel covering lemma says that if a sequence of open sets $(\psi_i^L, \psi_i^R)_{i \in \mathbb{T}}$ covers the unit interval $[0, 1]$, then an initial segment of the sequence already covers $[0, 1]$.

Theorem 6.1 *The following is provable in* $\mathsf{CPV}^\omega + \mathsf{QF\text{-}AC} + \Pi_1^0\text{-}\mathsf{WKL}^\omega$. *Given two sequences of real numbers* $(\psi_i^L)_{i \in \mathbb{T}}$ *and* $(\psi_i^R)_{i \in \mathbb{T}}$*, if*

$$\forall \psi \in [0, 1] \exists i(\psi_i^L <_{\mathbb{R}} \psi <_{\mathbb{R}} \psi_i^R)$$

then $\exists n \forall \psi \in [0, 1] \exists i \leq n(\psi_i^L <_{\mathbb{R}} \psi <_{\mathbb{R}} \psi_i^R)$.

Proof. For each positive number $w \in \mathbb{N}$ (having binary expansion $b_m b_{m-1} \ldots b_0$ with $b_m = 1$) we define two rational numbers, written for simplicity in radix notation,

$$p_w := 0.b_{m-1} \ldots b_0,$$

$$q_w := 0.b_{m-1} \ldots b_0 + 2^{-m}.$$

For completeness we set $p_0 := p_1$ and $q_0 := q_1$. That is, for each positive number m we have partitioned the unit interval $[0, 1]$ into 2^m subintervals of length 2^{-m}. Let

$$T(w) :\equiv \neg \exists i(i \leq |w| \wedge \psi_i^L <_{\mathbb{R}} p_w <_{\mathbb{R}} q_w <_{\mathbb{R}} \psi_i^R).$$

It is easy to show that $T(w)$ defines a tree, i.e. if $T(w)$ holds and $v \preceq w$ then $T(v)$ also holds. Moreover, notice that $T(w)$ is Π_1^0. Assuming that

$$(*) \ \forall \psi \in [0, 1] \exists i(\psi_i^L <_{\mathbb{R}} \psi <_{\mathbb{R}} \psi_i^R)$$

we claim that T has no infinite path. For the sake of contradiction, assume f is an infinite path in T. Define then the real number ψ as (in radix notation)

$$\psi(n) := 0.f(1) \ldots f(n -_{\mathbb{T}} 1).$$

Note that for all $j \in \mathbb{T}$, $p_{\overline{f}j} \leq \psi \leq q_{\overline{f}j}$, and moreover, as j increases both $p_{\overline{f}j}$ and $q_{\overline{f}j}$ converge to ψ. For such ψ, let i be as in assumption $(*)$, i.e. $\psi_i^L <_{\mathbb{R}} \psi <_{\mathbb{R}} \psi_i^R$. Let n be so large that $i \leq n$ and $\psi_i^L <_{\mathbb{R}} p_{\overline{f}n} <_{\mathbb{R}} q_{\overline{f}n} <_{\mathbb{R}} \psi_i^R$. Then $\neg T(\overline{f}n)$, which proves the claim. By weak König's lemma it follows that T is finite. Let $n \in \mathbb{T}$ be such that

$$\forall w(T(w) \rightarrow |w| < n).$$

Therefore

$$\forall w(|w| = n \rightarrow \exists i \leq n(\psi_i^L <_{\mathbb{R}} p_w <_{\mathbb{R}} q_w <_{\mathbb{R}} \psi_i^R)),$$

which implies

$$\exists n \forall \psi \in [0, 1] \exists i \leq n(\psi_i^L <_{\mathbb{R}} \psi <_{\mathbb{R}} \psi_i^R). \quad \square$$

7 Related results and open problems

As mentioned above, Howard [9] gave a functional interpretation (of the negative translation of) WKL using a different form of binary bar recursion, namely

$$\mathcal{B}^H(Y, z) = \begin{cases} 0 & \text{if } Y\hat{z} \leq |z| \\ t & \text{otherwise,} \end{cases}$$

where $t = 1 + \max\{\mathcal{B}^H(Y, z0), \mathcal{B}^H(Y, z1)\}$. Note that $\mathcal{B}^H(Y, 0)$ can also be defined in terms of unbounded search as

$$\min n \forall w(|w| = n \rightarrow \exists z \preceq w(Y\hat{z} \leq |z|)). \tag{4}$$

As opposed to the matrix of the search (3), which is associated to $\mathcal{B}$, the above unbounded search apparently requires at each step of the computation an exponential search. This reflects, as we shall see, the fact that Howard interpretation of WKL does not take bounded quantifier into consideration.

In the following let $T(w)$ abbreviate $\forall v \preceq w(gv = 0)$. Consider the negative translation of WKL (over intuitionistic logic and MP^ω)[4]

$$\forall g(\forall n \neg\neg \exists w(|w| = n \wedge T(w)) \rightarrow \neg\neg \exists f \forall n T(\overline{f}n)).$$

In a system where exponential search is available, the bounded quantifier $\exists w(|w| = n \wedge \ldots)$ can be absorbed by quantifier-free matrices and functional interpretation does not witness it. Hence, in this case the functional interpretation of WKL^N asks for n and f (uniformly in g and Y) realizing

$$\forall g, Y \exists n, f(\exists w(|w| = n \wedge T(w)) \rightarrow T(\overline{f}(Yf))).$$

The functional $\mathcal{B}^H$ is used to realize n by setting $n := \mathcal{B}^H(Y, 0)$. Then, in order to produce f satisfying

$$\forall g, Y \exists f(\exists w(|w| = n \wedge T(w)) \rightarrow T(\overline{f}(Yf))),$$

one looks for a w of length n such that $T(w)$ holds. If such w is found, meaning that the premise holds, let v be the shortest prefix of w such that $Y\hat{v} \leq |v|$. We then set $f := \hat{v}$, so that (since $Yf = Y\hat{v} \leq |v| \leq n$) the conclusion also holds. If no such w exists we can safely take f to be an arbitrary path.

It is important to note that Howard's work concerns systems in which exponential search is available, i.e. bounded quantifiers can be absorbed by quantifier-free matrices. In the present context of feasible analysis, negative translation and functional interpretation need to take such quantifiers into consideration. Notice, however, that by taking the quantification over w into consideration we obtain a new

[4] In stronger settings the operation $\overline{f}$ is normally defined as

$$\overline{f}(n) = \begin{cases} 0 & \text{if } n = 0 \\ f(0) \ldots f(n-1) & \text{otherwise.} \end{cases}$$

functional as input (W in the proof of Theorem 5.1) which we could use to realize y and f in a *feasible way*, by using the simpler search (3) instead of (4).

An interesting follow-up of the present paper is to find ineffective proofs of Π^0_2-theorems which can be formalized in $\mathsf{CPV}^\omega + \mathsf{QF\text{-}AC} + \Pi^0_1\text{-}\mathsf{WKL}^\omega$, and carry out the extraction of polynomial-time algorithms (cf. [15] where, in the context of classical analysis, a proof based on WKL has been analyzed providing the first effective realizer for the theorem). It would be also interesting to investigate whether Kohlenbach's proof of WKL elimination can be translated to the feasible setting, by making a more careful treatment of bounded quantifiers.

The author has been recently informed by Avigad that Sieg's proof of WKL elimination was successfully translated to the feasible setting by Kauffmann [11], using an extension of S^1_2 with 0-1 function variables and quantifier-free choice for those functions. In fact, by making use of Parikh's lemma, Kauffmann obtains an effective WKL elimination for trees defined by arbitrary bounded formulas. This provides another effective procedure for extracting polynomial-time realizers from WKL-proofs in feasible analysis, via an initial elimination of WKL from the proof. It should be noticed, however, that our algorithm produces a polynomial-time realizer *directly* from the original WKL-proof, without having to initially go through the elimination procedure. Moreover, our approach has a finite type theory as basis and enjoys the *full modularity* of the negative translation and functional interpretation, whereas Sieg's proof, as well as Parikh's lemma, are based on cut-elimination.

Acknowledgement. I am very much indebted to Ulrich Kohlenbach for bringing the problem under consideration to my attention, for pointing me to references such as Howard's paper on bar recursion and for various discussions we have had on the results here presented. I would also like to thank Fernando Ferreira and Jeremy Avigad for their general comments on the paper.

References

[1] J. Avigad and S. Feferman. Gödel's functional ("Dialectica") interpretation. In S. R. Buss, editor, *Handbook of proof theory*, volume 137 of *Studies in Logic and the Foundations of Mathematics*, pages 337–405. Elsevier, North-Holland, Amsterdam, 1998.

[2] S. R. Buss. *Bounded arithmetic*. PhD thesis, Princeton University, Princeton, New Jersey, 1985.

[3] S. R. Buss. The polynomial hierarchy and intuitionistic bounded arithmetic. In G. Goos and J. Hartmanis, editors, *Structure in Complexity Theory*, volume 223 of *Lecture Notes in Computer Science*, pages 77–103. Springer-Verlag, 1986.

[4] S. Cook. Feasibly constructive proofs and the propositional calculus. In *Proc. 7th A.C.M. Symposium on the Theory of Computation*, pages 83–97, 1975.

[5] S. Cook and A. Urquhart. Functional interpretations of feasibly constructive arithmetic. *Annals of Pure and Applied Logic*, 63:103–200, 1993.

[6] F. Ferreira. A feasible theory for analysis. *Journal of Symbolic Logic*, 59(3):1001–1011, 1994.

[7] F. Ferreira and A. Fernandes. Basic applications of weak König's lemma in feasible analysis. Submitted.

[8] H. Friedman. Systems of second order arithmetic with restricted induction, I, II (abstracts). *Journal of Symbolic Logic*, 41:557–559, 1976.

[9] W. A. Howard. Ordinal analysis of simple cases of bar recursion. *Journal of Symbolic Logic*, 46(1):17–30, 1981.

[10] B. Kapron and S. Cook. A new characterization of type-2 feasibility. *SIAM Journal Computing*, 25(1):117–132, 96.

[11] B. Kauffmann. *Application of Proof Theory to Computational Complexity Theory: Comparison of Different Methods*. PhD thesis, Carnegie Mellon University, Department of Philosophy, 2001.

[12] U. Kohlenbach. Effective bounds from ineffective proofs in analysis: an application of functional interpretation and majorization. *Journal of Symbolic Logic*, 57:1239–1273, 1992.

[13] U. Kohlenbach. Mathematically strong subsystems of analysis with low rate of growth of provably recursive functionals. *Archive for Mathematical Logic*, 36:31–71, 1996.

[14] U. Kohlenbach. On the no-counterexample interpretation. *Journal of Symbolic Logic*, 64:1491–1511, 1999.

[15] U. Kohlenbach and P. Oliva. Proof mining in L_1-approximation. *Annals of Pure and Applied Logic*, 121:1–38, 2003.

[16] H. Luckhardt. Extensional Gödel functional interpretation – a consistency proof of classical analysis. In *Lecture Notes in Mathematics*, volume 306. Springer Verlag, 1973.

[17] B. Scarpellini. A model for bar recursion of higher types. *Compositio Mathematica*, 23:123–153, 1971.

[18] W. Sieg. Hilbert's program sixty years later. *Journal of Symbolic Logic*, 53:338–348, 1988.

[19] W. Sieg. Herbrand analyses. *Archive for Mathematical Logic*, 30:409–441, 1991.

[20] S. G. Simpson. *Subsystems of Second Order Arithmetic*. Perspectives in Mathematical Logic. Springer-Verlag, 1999.

[21] C. Spector. Provably recursive functionals of analysis: a consistency proof of analysis by an extension of principles in current intuitionistic mathematics. In F. D. E. Dekker, editor, *Recursive Function Theory: Proc. Symposia in Pure Mathematics*, volume 5, pages 1–27. American Mathematical Society, Providence, Rhode Island, 1962.