An Application of the Refined A-Translation for a Variant of the Infinite Pigeon Hole Principle

Diana Ratiu*

Ludwig-Maximilians Universität München, Fakultät für Mathematik
ratiu@math.lmu.de

Abstract. It is well-known that every Π^0_2-formula is intuitionistically provable iff it is classically provable. Berger et al. (2002) proposed an improvement of the A-Translation method which transforms a proof by contradiction of a Π^0_2-statement into its intuitionistic counterpart and Schwichtenberg (2007) refined this procedure further. In this paper we apply this mechanism to Stolzenberg's Principle (SP), stating that any infinite boolean sequence has infinite occurrences of either 0's or 1's. We aim at applying the extraction procedures further on Ramsey's Theorem, which plays a foundational role in combinatorics. Since the proof of Ramsey's Theorem is built upon a generalisation of (SP), an essential component of its computational content is represented by the extracted program presented in this paper.

Keywords: Infinite Pigeon Hole Principle, program extraction, classical proofs, refined A-Translation, realizability.

1 Introduction

In the case of intuitionistic proofs, the semantics associated to the logical operators allows for an interpretation of proofs as programs, via the Curry-Howard correspondence, but in the classical setting the computational content of proofs is not explicit. However, for the Π^0_2- formulas $\forall x \exists y G(x, y)$ with arithmetical and decidable kernel, it is known that they are equiderivable in both systems, i.e.

$$\Gamma \vdash_c \forall x \exists y G(x, y) \text{ iff } \Gamma' \vdash_i \forall x \exists y G(x, y), \tag{1.1}$$

$\vdash_c$ denoting derivability in classical, $\vdash_i$ in intuitionistic and $\vdash_m$ in minimal logic.

This suggests that one can obtain the constructive counterpart, and thus unravel the computational content, of proofs by contradiction of $\forall x \exists y G(x, y)$. Indeed, beginning as early as the 1930's with the work of Gödel, various methods have been proposed in order to transform classical proofs into constructive ones. Gödel (1958) suggested the so-called Dialectica interpretation and also,

* The author gratefully acknowledges financial support by MATHLOGAPS (MEST-CT-2004-504029), a Marie Curie Early Stage Training Site.

in parallel with Gentzen, a form of double-negation translation, which has further developed into variants. The transformation of concern to us is a hybrid of the Gödel-Gentzen negative translation and Friedman-Dragalin (also known as A-) translation as presented in Friedman (1978). More precisely, we work with a refinement of this method, as proposed in Berger et al. (2002) and further improved in Schwichtenberg (2007), such that the insertion of double negations is avoided. Due to the key role played by negations in the A-translation, the extracted term is significantly simplified.

We read formula $\forall x \exists y G(x, y)$ as "given an input x is there an algorithm which produces an output y from this input, such that the requirement G on x is met?". Our work investigates under which circumstances and with what results we can synthesise programs from proofs of such specifications, carried out in the classical setting. More precisely, we analyse the computational potential of proofs in the fragment of the intuitionistic logic which coincides with the classical logic.

In this paper we apply the refined A-Translation to a corollary of the (infinite) pigeon hole principle (IPH) stated bellow. IPH is used to prove Ramsey's Theorem, a fundamental result in the realm of combinatorics, so before we attempt at identifying the computational content of the latter, we need to retrieve the content from the classical proof of (IPH).

Lemma 1.1 (Infinite Pigeon Hole Principle). *Any infinite sequence coloured with finitely many colours has (at least) one colour occurring infinitely often.*

For simplicity reasons, we restrict our attention here to the variant of the (IPH) which regards only a two-colouring, and refer to this as the boolean case or Stolzenberg's Principle (SP), adopting the reference from Berardi (2006).

Lemma 1.2 (Stolzenberg's Principle). *Any infinite sequence over {0,1} has either infinitely many 0's or infinitely many 1's.*

One cannot expect, however, to extract a computable functional producing such an infinite subsequence from a given infinite one, because this problem is not decidable. This amounts to saying that Lemma 1.2 does not have a constructive proof. For this reason, we restrict our attention to a corollary

Corollary 1 (Finite SP). *Any infinite boolean sequence has a finite constant subsequence of a specified length.*

Since this is a Π^0_2-statement, and thus intuitionistically provable if classically provable, we are able transform its classical proof with the refined A-Translation method and realize the resulting proof. Our purpose is to argue that in this way we are able to recover the computational content, i.e., produce the algorithm computing the constant sequence from the given infinite one.

The relevance of the (IPH) is pointed out in Halbeisen (2008). On the one hand by the Axiom of Choice, (IPH) leads to

Lemma 1.3 (König's Lemma). *Every infinite, finitely branching tree contains an infinite branch.*

On the other hand, Infinite Ramsey's Theorem can be also proved by (IPH) and the case of 2 colours by (SP). Further, the finite version of Ramsey's Theorem follows by a classical argument, using König's Lemma (Halbeisen (2008)).

Theorem 1 (Infinite Ramsey's Theorem). *Let $n \in \mathtt{N} \setminus \{0\}$ and S be an infinite set*[1]*. Let colours from some finite set be associated to the n-element subsets of S. Then there exists an infinite set $M \subseteq S$, which is* homogeneous*, i.e., the n-element subsets of M all have the same colour.*

Outline. The following section introduces the system in which we are working and the transformation method which, when combined with the modified realizability, enables the extraction of computational content from classical proofs. We present in detail in Section 3 the application of this method to Corollary 1 and comment on the results. In Section 4 we overview related work and briefly compare our results with those reported in the literature. Since we are reporting on work in progress, we will conclude with an emphasis on the future perspectives.

2 Recovering the algorithmic content from classical proofs

2.1 The System as a Negative Fragment of HA^ω

We work in a fragment of Heyting's Arithmetic with higher types, based on Gödel's System T. *Types* are built from base types by the formation of function and product types

$$\rho, \sigma := \mathtt{N} \mid \mathtt{B} \mid \mathtt{L}(\rho) \mid \rho \Rightarrow \sigma \mid \rho \times \sigma$$

Terms are defined inductively by abstraction, application, pairing and projection from typed variables (x^ρ) and constants (constructors and recursion operators)

$$\begin{aligned} s, t :=& x^\rho \mid (\lambda x^\rho t^\sigma)^{\rho \Rightarrow \sigma} \mid (s^{\rho \Rightarrow \sigma} t^\rho)^\sigma \mid \langle t^\rho, s^\sigma \rangle^{\rho \times \sigma} \mid (t^{\rho \times \sigma} 0)^\rho \mid (t^{\rho \times \sigma} 1)^\sigma \\ & \mathrm{tt}^{\mathtt{B}} \mid \mathrm{ff}^{\mathtt{B}} \mid 0^{\mathtt{N}} \mid \mathsf{S}^{\mathtt{N} \to \mathtt{N}} \mid \mathsf{nil}^{\mathtt{L}(\rho)} \mid \mathit{cons}^{\rho \Rightarrow \mathtt{L}(\rho) \Rightarrow \mathtt{L}(\rho)} \mid \\ & \mathcal{R}_{\mathtt{N}}^{\mathtt{N} \Rightarrow \sigma \Rightarrow (\mathtt{N} \Rightarrow \sigma \Rightarrow \sigma) \Rightarrow \sigma} \mid \mathcal{R}_{\mathtt{L}(\rho)}^{\mathtt{L}(\rho) \Rightarrow \sigma \Rightarrow (\rho \Rightarrow \mathtt{L}(\rho) \Rightarrow \sigma \Rightarrow \sigma) \Rightarrow \sigma} \mid \\ & \mathcal{C}_{\mathtt{N},\sigma}^{\mathtt{N} \Rightarrow \sigma \Rightarrow (\mathtt{N} \Rightarrow \sigma) \Rightarrow \sigma} \mid \mathcal{C}_{\mathtt{L},\sigma}^{\mathtt{L}(\rho) \Rightarrow \sigma \Rightarrow (\rho \Rightarrow \mathtt{L}(\rho) \Rightarrow \sigma) \Rightarrow \sigma}. \end{aligned}$$

We abbreviate $\mathrm{cons}\, t\, l$ to $t :: l$, denote the element at position $m^{\mathtt{N}}$ in the list l by l_m and the length of the list by $|l|$. In this paper we work only over $\mathtt{L}(\mathtt{N})$.

Terms are considered to be always typed, but whenever clear from the context, the types will be omitted.

We allow a special unit type[2] which we denote by ε and, by abuse of notation, take the terms of this special type to be also ε. We make the following conventions

$$\begin{gathered} \varepsilon t := \varepsilon, \qquad t\varepsilon := t, \qquad \varepsilon\varepsilon := \varepsilon \\ (\rho \Rightarrow \varepsilon) := \varepsilon, \quad (\varepsilon \Rightarrow \sigma) := \sigma, \quad (\varepsilon \Rightarrow \varepsilon) := \varepsilon. \end{gathered} \tag{ε}$$

[1] Ramsey's Theorem is most commonly stated in terms of graphs: S is taken to be an infinite complete graph, M a subgraph of S and n the number of edges

[2] The necessity of the ε type will become clear in the Section 2.3 on realizability.

A special predicate constant *atom* of arity $\mathtt{B}$ is considered, in order to express boolean valued functions in our language as decidable predicates[3]. Its logical interpretation is given by the truth axiom $\mathsf{AxT} : T$, where $T := \mathrm{atom}(\mathrm{tt})$.

$\bot$ is defined as a nullary predicate variable, since when applying the A-Translation we will need to substitute an arbitrary formula A for it.

Definition 1. *Let P be a predicate variable of arity $(\rho_1, \ldots, \rho_n)$, x a typed variable and $t_1, \ldots, t_n$ terms.* Formulas *are built from* prime formulas *by $\to$ and $\forall$*

$$A, B := atom(t^{\mathtt{B}}) \mid P(t_1^{\rho_1}, \ldots, t_n^{\rho_n}) \mid \bot \mid A \to B \mid \forall x^{\sigma} A$$

$\boldsymbol{A}$ abbreviates $A_1 \to \cdots \to A_n$ and ψ, φ will also denote formulas. To save on parenthesis, we use the "dot"-notation:

$$\forall x.\ A_1(x) \to \cdots \to A_n(x) \text{ stands for } \forall x(A_1(x) \to \cdots \to A_n(x)).$$

Formulas are arithmetical, but the predicate variables, viewed as place holders for comprehension terms, allow us to "simulate" part of the higher order logic, while remaining conservative over the first order logic.

We regard the classical (or *weak*) logical operators as abbreviations

$$\tilde{\neg} A := A \to \bot \qquad A \mathbin{\tilde{\vee}} B := \tilde{\neg} A \to \tilde{\neg} B \to \bot$$
$$\tilde{\exists} x A := (\forall x.\ A \to \bot) \to \bot \qquad A \mathbin{\tilde{\wedge}} B := \tilde{\neg} (A \to \tilde{\neg} B)$$

For the cases when we do not want a predicate variable for "falsity", we have $F := atom(\mathrm{ff})$ and, accordingly, $\neg A := A \to F$.

Since the weak "Ex falso quodlibet", $\tilde{\mathsf{efq}}_A : \bot \to A$, is problematic for the A-Translation, where we want to substitute for $\bot$ *any* formula A, we will work in the minimal setting (intuitionistic logic without $\tilde{\mathsf{efq}}_A$). However, for formulas A with no free predicate variables one can prove by induction on A that

$$\vdash_m \mathrm{F} \to A \quad (\mathsf{efq}_A) \quad \text{and} \quad \vdash_m ((A \to \mathrm{F}) \to \mathrm{F}) \to A \quad \text{(Stability)}$$

By the Curry-Howard correspondence, typed terms can be associated to the inference rules of the natural deduction system (Gentzen (1934)).

We define *proof terms* to be[4]

$$\begin{aligned} M, N :=& u^A \mid (\lambda u^A M^B)^{A \to B} \mid (M^{A \to B} N^A)^B \mid (\lambda x^{\rho} M^{A(x)})^{\forall x^{\rho} A(x)} \mid (M^{\forall x^{\rho} A(x)} t)^{A(t)} \mid \\ & \mathsf{Ind}_{n,A} : \forall m^{\mathtt{N}}.\ A(0) \to (\forall n.\ A(n) \to A(\mathsf{S}n)) \to A(m) \mid \\ & \mathsf{Ind}_{l,A} : \forall l^{\mathtt{L}(\rho)}.\ A(\mathsf{nil}) \to \forall x, l'^{\mathtt{L}(\rho)}.\ (A(l') \to A(x :: l')) \to A(l) \mid \\ & \mathsf{Cases}_{m,A} : \forall m^{\mathtt{N}}.\ A(0) \to \forall n A(\mathsf{S}n) \to A(m) \mid \\ & \mathsf{Cases}_{l,A} : \forall l^{\mathtt{L}(\rho)}.\ A(\mathsf{nil}) \to \forall n, l' A(n :: l') \to A(l) \mid \mathsf{AxT} : \mathrm{atom}(\mathrm{tt}) \end{aligned}$$

The sets of free variables $FV(M)$ and free (open) assumption variables $FA(M)$ as well as capture-free substitutions are defined inductively as usual.

[3] For instance, $r^{\mathtt{N}} < t^{\mathtt{N}}$ is formulated as $\mathrm{atom}(<^{\mathtt{N} \Rightarrow \mathtt{N} \Rightarrow \mathtt{B}}(r, s))$.

[4] The variable condition is imposed on the term $\lambda x M^A$ corresponding to the rule $\forall^+ x$: The derivation term M should not contain any open assumption with x as a free variable.

2.2 Refined A-Translation

A-Translation is a combination of the Gödel-Gentzen negative translation (or double-negation translation) and the A-translation. The former associates to each formula ψ its negative translation ψ^g, which is classically equivalent to ψ, but weaker than it intuitionistic logic[5]. We have

$$\Gamma \vdash_c \psi \text{ iff } \Gamma^g \vdash_i \psi^g$$

Friedman (1978) defines the "A-translation" ϕ^A as the result of simultaneously replacing in ϕ each atomic subformula φ by $(\varphi \vee A)$. We call this Friedman's "trick", since the known A-Translation is a combination of this remark and the double negation translation. Let Γ^A be Γ with each formula ϕ replaced by ϕ^A.

$$\text{If } \Gamma \vdash_i \psi \text{ then } \Gamma^A \vdash_i \psi \vee A,$$

and further, if ψ is $\bot$, since $\bot \vee A \equiv A$,

$$\text{if } \Gamma \vdash_i \bot \text{ then } \Gamma^A \vdash_i A.$$

If we now combine the two translations and take A to be $\forall x \exists y G(x, y)$, then

$$\text{from } \Gamma \vdash_c \forall x \tilde{\exists} y.\ G(x, y) \text{ we can infer } \Gamma^A \vdash_i \forall x \exists y G(x, y).$$

However, the above mentioned method is not efficient in practice, since all $\bot$'s inserted by the double negation will A-translate to $\forall x \exists y G(x, y)$, which is computationally relevant. We present in what follows the refinement from Berger et al. (2002), which identifies the cases where the double negation is not necessary.

Inspired from Seisenberger (2003), we restrict "decidable formulas" to

Definition 2 (Decidable formulas). *A formula is said to be* decidable *if it is built from atomic formulas,* $\mathrm{atom}(t)$*, only by propositional connectives and quantifiers which are either boolean or range over finite sets of naturals.*

Lemma 2.1 (Case Distinction). *Let B be an arbitrary formula and C a decidable formula. Then we have*

$$\vdash (\neg C \to B) \to (C \to B) \to B \qquad \text{(Cases)}$$

The proof, by structural induction on C, is sketched in Berger et al. (2002).

Definition 3. *If a formula "ends" with $\bot$, it is said to be* relevant *and otherwise it is called* irrelevant*. That is, C is a relevant formulas iff*

$$C := \bot \mid B \to C \mid \forall x C$$

[5] For reasons of brevity, we omit this definition, but refer the reader to Gödel (1933); Gentzen (1934).

Let P range over atomic formulas. Goal formulas G *and* definite formulas D *are defined inductively to be*

$$\begin{array}{l} G := P \mid \bot \mid D \to G, \textit{ provided } D \textit{ relevant } \bigvee D \textit{ quantifier-free} \\ \qquad\quad \mid \forall x G, \quad \textit{provided } G \textit{ irrelevant } \bigvee \forall x G \textit{ decidable}, \end{array}$$

$$\begin{array}{l} D := P \mid \bot \mid G \to D, \textit{ provided } D \textit{ relevant } \bigvee G \textit{ irrelevant} \\ \qquad\quad \mid \forall x D. \end{array}$$

These families of formulas are chosen such that Friedman's "trick" is applied only to $\bot$ and such that the insertion of double negations is avoided. Furthermore, by substituting $\bot$ by F whenever possible - and such situations are identified by the following lemma - we can confine the substitution even further.

Lemma 2.2. *Let A^F denote $A[\bot := F]$. For definite formulas D and goal formulas G we have from $F \to \bot$ derivations in intuitionistic arithmetic of*

$$D^F \to D, \tag{2.1}$$

$$G \to (G^F \to \bot) \to \bot. \tag{2.2}$$

A more detailed version can be found in Schwichtenberg (2007) and the proof, by simultaneous induction, follows the line of its variant from Berger et al. (2002). In order to show (2.2) we need (Cases), so G has to be a decidable formula, which justifies some of the restrictions from Definition 3. The formula (2.2) can be extended (see Schwichtenberg (2007)) to $\boldsymbol{G} \to (\boldsymbol{G}^F \to \bot) \to \bot$.

Having kept $\bot$ only in the relevant positions, we use Friedman's trick to substitute for it $A := \exists y G(x, y)$. Putting the pieces of the puzzle together,

Lemma 2.3. *Consider that we are given*

$$\vdash_c \forall x.\ \boldsymbol{D} \to \boldsymbol{H} \to \tilde{\exists} y\ G(x, y), \tag{2.3}$$

where G is a goal, $\boldsymbol{D}$ are definite and $\boldsymbol{H}$ are arbitrary formulas. Then

$$\vdash_i \forall x.\ \boldsymbol{D}^F \to \boldsymbol{H}[\bot := \exists y\ \boldsymbol{G}^F(x, y)] \to \exists y\ \boldsymbol{G}^F(x, y). \tag{2.4}$$

Proof (Sketch). (2.3) is transformed according to Lemma 2.2 to

$$\vdash_i (F \to \bot) \to \boldsymbol{D}^F \to \boldsymbol{H} \to \forall x\ (\forall y.\ \boldsymbol{G}^F(x, y) \to \bot) \to \bot.$$

If we now substitute in this latter $\bot$ by $\exists y\ \boldsymbol{G}^F(x, y)$, since

$$\vdash_i F \to \exists y\ \boldsymbol{G}^F(x, y) \text{ and } \vdash_i \forall y.\ \boldsymbol{G}^F(x, y) \to \exists y\ \boldsymbol{G}^F(x, y),$$

we obtain (2.4).

2.3 Realizers

We assign to every formula a computational type $\tau(A)$. If the proof M of A has no computational content we call A is computationally *irrelevant* and mark this by assigning to A the type ε. We let $[\![M]\!] := \epsilon$, with $[\![M]\!]$ denoting the program extracted from the proof of A. A is computationally *relevant* when $\tau(A) \neq \varepsilon$.

Since we do not a priori know what comprehension term will be substituted for the predicate variable P, we assign it some type variable α_P. With this and the conventions we made in (ε), we consider the following typing rules

$$\begin{aligned}\tau(P(\boldsymbol{r})) &:= \begin{cases}\varepsilon & \text{if } P \text{ does not have content}\\ \alpha_P & \text{otherwise,}\end{cases}\\ \tau(A \to B) &:= \tau(A) \Rightarrow \tau(B),\\ \tau(\forall x^\rho A) &:= \rho \Rightarrow \tau(A).\end{aligned}$$

Let $r\,\mathbf{mr}\,A$ denote (Kreisel's) *modified realizability*, which reads "the term r realizes the formula A" and is defined by

$$\begin{aligned}r\,\mathbf{mr}\,P(\boldsymbol{s}^{\boldsymbol{\sigma}}) &:= \begin{cases}P^{\mathbf{r}}(r,\boldsymbol{s}) & \text{if } P \text{ is a predicate variable with assigned } \alpha_P\\ P(\boldsymbol{s}) & \text{if } P \text{ is a predicate constant}\end{cases}\\ r\,\mathbf{mr}\,\forall x A &:= \forall x\; rx\,\mathbf{mr}\,A\\ r\,\mathbf{mr}\,A \to B &:= \forall x.\; x\,\mathbf{mr}\,A \;\to\; rx\,\mathbf{mr}\,B\end{aligned}$$

Here $P^{\mathbf{r}}$ is a new predicate variable of arity $(\tau(r),\boldsymbol{\sigma})$ associated to P of arity $\boldsymbol{\sigma}$.

The extracted term of a derivation is obtained inductively

$$\begin{aligned}[\![u^A]\!] &:= x_u^{\tau(A)} \quad (x_u^{\tau(A)} \text{ the object variable associated with } u^A),\\ [\![(\lambda u^A M^B)^{A\to B}]\!] &:= \lambda x_u^{\tau(A)} [\![M]\!]^{\tau(B)},\\ [\![M^{A\to B} N^A]\!] &:= [\![M]\!]^{\tau(A\to B)} [\![N]\!]^{\tau(A)},\\ [\![(\lambda x^\rho M^A)^{\forall x A}]\!] &:= \lambda x^\rho [\![M]\!]^{\tau(A)},\\ [\![M^{\forall x A} r]\!] &:= [\![M]\!]^{\tau(\forall x A)} r.\end{aligned}$$

In particular, for our induction schemes, when $\sigma := \tau(A) \neq \varepsilon$, we have

$$[\![\mathsf{Ind}_{n,A}]\!] := \mathcal{R}_{\mathtt{N}}^{\mathtt{N}\Rightarrow\sigma\Rightarrow(\mathtt{N}\Rightarrow\sigma\Rightarrow\sigma)\Rightarrow\sigma} \qquad [\![\mathsf{Ind}_{L,A}]\!] := \mathcal{R}_{\mathtt{L}(\rho)}^{\mathtt{L}(\rho)\Rightarrow\sigma\Rightarrow(\rho\Rightarrow\mathtt{L}(\rho)\Rightarrow\sigma\Rightarrow\sigma)\Rightarrow\sigma},$$

with the following associated conversion rules

$$\begin{aligned}\mathcal{R}_{\mathtt{N}}(0,f,g) &= f, \qquad \mathcal{R}_{\mathtt{N}}(Sn,f,g) = g(n,\mathcal{R}_{\mathtt{N}}(n,f,g))\\ \mathcal{R}_{\mathtt{L}}(\mathsf{nil},f,g) &= f, \qquad \mathcal{R}_{\mathtt{L}}(n::l,f,g) = g(n,l,\mathcal{R}_{\mathtt{L}}(l,f,g)).\end{aligned}$$

For "Cases"

$$[\![\mathsf{Cases}_{n,A}]\!] := \mathcal{C}_{\mathtt{N},\sigma}^{\mathtt{N}\Rightarrow\sigma\Rightarrow(\mathtt{N}\Rightarrow\sigma)\Rightarrow\sigma} \qquad [\![\mathsf{Cases}_{l,A}]\!] := \mathcal{C}_{\mathtt{L},\sigma}^{\mathtt{L}(\rho)\Rightarrow\sigma\Rightarrow(\rho\Rightarrow\mathtt{L}(\rho)\Rightarrow\sigma)\Rightarrow\sigma},$$

we associate the "if ... then ... else" construct

$$\begin{aligned}\mathcal{C}_{\mathtt{N},\sigma}(n,f,g) &= \mathsf{if}\ (n=0)\ \mathsf{then}\ f\ \mathsf{else}\ g\,(n-1),\\ \mathcal{C}_{\mathtt{L},\sigma}(l,f,g) &= \mathsf{if}\ (l=\mathsf{nil})\ \mathsf{then}\ f\ \mathsf{else}\ (g\ l_0\ \mathrm{cdr}\ l),\end{aligned}$$

with (cdr l) denoting the list l without its head element l_0.

Theorem 2 (Soundness). *(Berger et al. (2002)) Let M be a derivation of B. Then there is a derivation of $[\![M]\!]\,\mathbf{mr}\,B$ from the assumptions*

$$\{x_u^{\tau(C)}\,\mathbf{mr}\,C | u^C \in FA(M)\}.$$

To determine the realizer for (2.4), let us first assume that we have terms s and t realizing H and D, respectively, and the proofs

$$\vdash D \to H \to s\,\mathbf{mr}\,H[\bot/\exists y G^F(y)], \qquad \vdash D \to H \to t\,\mathbf{mr}\,D^F.$$

Let M be the derivation of (2.4) and $[\![M]\!]$ its extracted program. By Theorem 2

$$\vdash H \to D \to [\![M]\!]s\,t\,\mathbf{mr}\,\exists y G^F(y)$$

If we extend the definition of modified realizability by

$$r\,\mathbf{mr}\,\exists x A(x) := r0\,\mathbf{mr}\,A(r1), \text{ where } \tau(\exists x^\rho A) := \rho \times \tau(A)$$

we obtain

$$H \to D \to [\![M]\!]s\,t\,0\,\mathbf{mr}\,G^F([\![M]\!]s\,t\,1). \tag{2.5}$$

3 Case Study - Infinite Tape

Recall Lemma 1.2 (SP): Any infinite recursive sequence over {0,1} has either infinitely many 0's or infinitely many 1's.

The most intuitive way to represent the infinite (boolean) sequence is by $f^{\mathtt{N}\to\{0,1\}}$, i.e. the sequence which has the *implicit* property that $\forall k\,\tilde{\exists} b^{\{0,1\}}\,f\,k = b$. However, since we have in mind a generalisation to more colours, we choose to encode the sequence as $f^{\mathtt{N}\to\mathtt{N}}$ with the property that $\forall k.\ f\,k < 2$

Since we do not allow $\tilde{\neg}\,(\,\tilde{\neg}\,A) \to A$ (weak stability), we make the double negation explicit in the assumption

$$Inf_f := \forall k.\ \tilde{\neg}\,(f\,k < 2) \to \bot.$$

Definition 4. *We say that f has an* infinite monochromatic subsequence *iff*

$$\infty_f(r) := \forall n \tilde{\exists} k.\ n \le k \,\tilde{\wedge}\, f(k) = r. \text{ for some } r \in \{0,1\}$$

(SP) can be now expressed as

$$\forall f.\ Inf_f \to (\forall r.\ \infty_f(r) \to \bot) \to \bot \tag{SP}$$

Proof (of (SP)). Assume by contradiction that neither 0, nor 1 occur infinitely often ($\tilde{\neg}\,\infty_f(0)$ and $\tilde{\neg}\,\infty_f(1)$). Take n and m to be the last occurrences of 0 and 1, respectively, and use Inf_f on the maximum of m and n, denoted by $m \sqcup n$. Since $f\,(m \sqcup n) < 2$, the value at this point must be either 0 or 1, which comes in contradiction with one of our two initial claims and thus proves (SP).

In what follows, we give the term associated with this proof. Consider the following lemmas being already proved

$$L_i : \forall n_1, n_2.\ n_i \le (n_1 \sqcup n_2) \qquad L_{<0} : \forall n.\ n < 1 \to n = 0$$
$$L_{CaseDist} : \forall n_1, n_2.\ n_1 < \mathsf{S} n_2 \to (n_1 < n_2 \to \bot) \to (n_1 = n_2 \to \bot) \to \bot.$$

Then M_{SP} proves (SP), where

$$\begin{aligned}
M_{SP} :=& \lambda f \lambda v^{Inf_f} \lambda w^{(\forall r.\ \infty_f(r) \to \bot)}.\ (w\, 0\, M_{\infty_f(0)})^{\bot} \\
M_{\infty_f(0)} :=& \lambda n \lambda w_0^{\forall k. n \le k \to f\, k = 0 \to \bot}.\ (w\, 1\, M_{\infty_f(1)})^{\bot} \\
M_{\infty_f(1)} :=& \lambda m \lambda w_1^{\forall k. m \le k \to f\, k = 1 \to \bot}.\ (v^{\forall k.\ (f\, k < 2 \to \bot) \to \bot}\ (m \sqcup n) \lambda u_1^{f\,(m \sqcup n) < 2}.\ M_{CD}^{\bot})^{\bot} \\
M_{CD} :=& L_{CaseDist}\ (f\ (m \sqcup n))\, 1\, u_1\, M_{Case1}^{f\,(m \sqcup n) < 1 \to \bot}\, M_{Case2}^{f\,(m \sqcup n) = 1 \to \bot} \\
M_{Case1} :=& \lambda u_2^{f\,(m \sqcup n) < 1}.\ (w_0\ (m \sqcup n)\ (L_2\, m\, n)^{n < (m \sqcup n)} (L_{<0}\, u_2)^{f\,(m \sqcup n) = 0})^{\bot} \\
M_{Case2} :=& w_1\ (m \sqcup n)\ (L_1\, m\, n)^{m < (m \sqcup n)}
\end{aligned}$$

Since (SP) is not constructive, we restrict our attention to the Π^0_2-statement Corollary 1 (Finite SP, see Section 1). We break its proof in two small steps and first show an auxiliary lemma, which states that the sequence of length n can be obtained from the infinite monochromatic sequence. This trivial fact is an essential component of the proof, as its computational content constitutes the algorithm for selecting of indices from the monochromatic subsequence.

Let l be the list that we expect to output with the extracted algorithm, when provided f and n. The list l should have the following properties

- has the specified length, $|l| = n$
- is not constant, $\forall m.\ \mathsf{S} m < n \to l_m < l_{\mathsf{S} m}$ (denote this by $G_1(l, n)$)
- f has colour $r \in \{0, 1\}$ at each point from the list, $\forall m.\ m < n \to f l_m = r$ (denote this by $G_2(f, l, n, r)$)

Cumulating in $\tilde{\neg}\, G_f(n, r)$ the specifications for l,

$$G_f(n, r) := \forall l.\ |l| = n \to G_1(l, n) \to G_2(f, l, n, r) \to \bot,$$

the auxiliary lemma can be expressed as

$$\forall f, r.\ \infty_f(r) \to \forall n.\ G_f(n, r) \to \bot \qquad (\mathrm{SP}Aux)$$

Before presenting the proof term for SPAux, we give an intuition on the proof

*Proof (*SPAux*).* Fix arbitrary f and r and assume $\infty_f(r)$. We show $\tilde{\neg}\, G_f(n, r)$ by induction on n.

Base case If $n = 0$ then we take the empty list and use "Ex falso quodlibet".

Step case By case distinction on n we identify if the list provided by (IH) is non-empty.

Case n=0 We populate the empty list from the (IH) with the element l_0 obtained from $\infty_f(r)$ on $n = 0$. With $(l_0 :)$ the first conjunct of $Gc(f, 1, r)$ follows trivially and the other two by "Ex falso quodlibet".

Case S(n) Since $n \neq 0$, the list provided by the (IH) is not empty, so it is $l :: n_1$. With respect to $\mathsf{S}n_1$, we know from $\infty_f(r)$ that there exists a next element, n_2, which is fresh (thus complying with the second requirement on the list) and has the value r in the sequence f (with this the third clause of Gc is fulfilled). Thus, we have the list $l :: n_1 :: n_2$.

We take $K(f, r, n) := \forall k.\ n \leq k \to fk = r \to \bot$ (so $\infty_f(r) := \forall n.\ \tilde{\neg}\, K(f, r, n)$). Using

$$L_4 : \forall n_1, n_2.\ \mathsf{S}n_1 \leq n_2 \to n_1 < n_2$$
$$L_{Compat}(n_1, P) : \forall n_2.\ n_1 = n_2 \to P \to P(n_2 := n_1)$$

the proof term $M_{\mathrm{SP}Aux}$ is

$$\begin{aligned}
M_{\mathrm{SP}Aux} &:= \lambda f \lambda r \lambda w^{\infty_f(r)} \lambda n.\ \mathsf{Ind}_{n, \neg G_f(n,r)}\, n\, M_{base}^{\tilde{\neg}\, G(0,r)} M_{step}^{\forall n.\ \tilde{\neg}\, G_f(n,r) \to \tilde{\neg}\, G(\mathsf{S}n, r)} \\
M_{base} &:= \lambda u_0^{G(0,r)}.\ u_0\ \mathsf{nil}\ \mathsf{AxT}^{|\mathsf{nil}|=0}\ (\lambda m.\ \mathsf{efq}_{\mathsf{nil}_{\mathsf{S}m} < \mathsf{nil}_m})\ (\lambda m.\ \mathsf{efq}_{f(\mathsf{nil}_m)=r}) \\
M_{step} &:= \lambda n.\ \mathsf{Cases}_{n,\, \tilde{\neg}\, G_f(n,r) \to \tilde{\neg}\, G(\mathsf{S}n, r)}\, n\, M_{n=0} M_{\mathsf{S}n} \\
M_{n=0} &:= \lambda u_1^{G(0,r) \to \bot} \lambda u_2^{G(1,r)}.\ u_1 (\lambda l\, \lambda u_{1,1}^{|\mathsf{nil}|=0} \lambda u_{1,2}^{G_1(\mathsf{nil}, 0)} \lambda u_{1,3}^{G_2(f, \mathsf{nil}, 0, r)}.\ w\, 0\, M_{K(f,r,0)}) \\
M_{K(f,r,0)} &:= \lambda k \lambda v_1^{0 \leq k} \lambda w_1^{fk=r}.\ u_2\, (k{:})\, \mathsf{AxT}^{|k:|=1}\ (\lambda m.\ \mathsf{efq}_{(k:)_{\mathsf{S}m} < (k:)_m})^{G_1((k:), 1)} \\
&\qquad (\lambda m \lambda u^{m<1}.\ (L_{Compat}(m, f(k{:})_1 = r) 1 (L_{<0} u)^{m=1} w_1)^{f(k:)_m = r})^{G_2(f, (k:), 1, r)} \\
M_{\mathsf{S}n} &:= \lambda n \lambda u_1^{G(\mathsf{S}n, r) \to \bot} \lambda u_2^{G(\mathsf{S}(\mathsf{S}n), r)}. \\
&\qquad u_1 (\lambda l_1.\ \mathsf{Cases}_{l_1, G(\mathsf{S}n, r)\, l_1}\ l_1\ \mathsf{efq}^{G(\mathsf{S}n, r)\, \mathsf{nil}}\ \lambda n_1 \lambda l_2.\ M_{n_1 :: l_2}^{G(\mathsf{S}n, r)\, (n_1 :: l_2)})^{G(\mathsf{S}n, r)} \\
M_{n_1 :: l_2} &:= \lambda u_{1,1}^{|n_1 :: l_2| = \mathsf{S}n} \lambda u_{1,2}^{G_1(n_1 :: l_2, \mathsf{S}n)} \lambda u_{1,3}^{G_2(f, n_1 :: l_2, \mathsf{S}n, r)}.\ (w^{\infty_f(r)}\ (\mathsf{S}n_1)\, M_{K(f, r, \mathsf{S}n_1)})^{\bot} \\
M_{K(f,r,\mathsf{S}n_1)} &:= \lambda k_1 \lambda v_1^{\mathsf{S}n_1 \leq k_1} \lambda w_2^{fk_1 = r}.\ (u_2\, (k_1 :: n_1 :: l_2)\ u_{1,1}\ M_{G_1} M_{G_2})^{\bot} \\
\text{where } G_1 &:= G_1((k_1 :: n_1 :: l_2), \mathsf{S}(\mathsf{S}n)) \text{ and } G_2 := G_2(f, (k_1 :: n_1 :: l_2), \mathsf{S}(\mathsf{S}n), r) \\
M_{G_1} &:= \lambda m.\ \mathsf{Cases}_{m, G_1}\, m\, (\lambda u^{1 < \mathsf{S}(\mathsf{S}n)}.\ L_4\, (k_1 :: n_1 :: l_2)_1\, (k_1 :: n_1 :: l_2)_0\, v_1) \\
&\qquad (\lambda m \lambda v_2^{\mathsf{S}(\mathsf{S}m) < \mathsf{S}(\mathsf{S}n)}.\ (u_{1,2}^{|n_1 :: l_2| = \mathsf{S}n}\, m\, v_2)^{(n_1 :: l_2)_{\mathsf{S}m} < (n_1 :: l_2)_m}) \\
M_{G_2} &:= \lambda m.\ \mathsf{Cases}_{m, G_2}\, m\, (\lambda u^{0 < \mathsf{S}(\mathsf{S}n)}\ w_2)(\lambda m \lambda v_3^{\mathsf{S}m < \mathsf{S}(\mathsf{S}n)}.\ (u_{1,3}\, m\, v_3))
\end{aligned}$$

We are now able to prove (Finite SP). We give first the intuition on the proof.

Proof (Finite SP). Let n be arbitrary, but fixed. Suppose we have an infinite sequence f over $\{0, 1\}$. By (SP) (Lemma 1.2), this means that we either have a constant "1" or a constant "0" infinite subsequence. In either case, by SPAux, there exists a finite subsequence of length n.

This apparently simple proof is relevant to our investigation since it is but a first step, yet important, towards analysing the computational potential of Ramsey's Theorem. In what follows, we present the proof formally.

We are not interested in the colour of the extracted subsequence, so we take $G_2'(f, l, n) := \forall m.\ \mathsf{S}m < n \to fl_m = fl_{\mathsf{S}m}$ an alter the goal to

$$G'(n) := \forall l.\ |l| = n \to G_1(l, n) \to G_2'(f, l, n) \to \bot,$$

Thus, the proof term for the Corollary 1 is $M_{cor}^{\forall f.\ Inf_f \to \forall n.\ G'(n,r) \to \bot}$

$$M_{Cor} := \lambda f \lambda v^{Inf_f} \lambda n \lambda u^{G'(n)}.\ M_{SP}\ f\ v\ (\lambda r \lambda w^{\infty_f(r)}.\ M_{\mathrm{SP}Aux}\ f\ r\ w\ n\ M_{G_f(n,r)})$$
$$M_{G_f(n,r)} := \lambda l \lambda u_1^{|l|=n} \lambda u_2^{G_1(l,n)} \lambda u_3^{G_2(f,l,n,r)}.\ (u\ l\ u_1\ u_2\ M_{G_2'(f,l,n)})^{\bot}$$
$$M_{G_2'(f,l,n)} := \lambda m \lambda u_4^{\mathsf{S}m<n}.\ (L_{=Trans}\ (fl_m)\ r\ (fl_{\mathsf{S}m})(u_3\ m\ (L_1\ m\ n\ u_4)^{m<n})^{fl_m=r}$$
$$(L_2\ r\ (fl_{\mathsf{S}m})\ (u_3\ \mathsf{S}m\ u_4)^{fl_{\mathsf{S}m}=r})^{r=fl_{\mathsf{S}m}})^{fl_m=fl_{\mathsf{S}m}},$$

where we have used the lemmas

$$L_{=Trans} := \forall n_1, n_2, n_3.\ n_1 = n_2 \to n_2 = n_3 \to n_1 = n_3$$
$$L_1 := \forall n_1, n_2.\ \mathsf{S}n_1 < n_2 \to n_1 < n_2 \qquad L_2 := \forall n_1, n_2.\ n_1 = n_2 \to n_2 = n_1$$

3.1 Combining the Refined A-Translation with the Modified Realizability

We first need to verify that we are in the conditions of Lemma 2.3. The Corollary which we have proven is of the form $\forall f, n.\ D \to (\forall l\ \boldsymbol{G} \to \bot) \to \bot$, where

$$D := Inf_f := \forall k.\ \neg(f\ k < 2) \to \bot, \quad G_2 := G_1(l,n) := \forall m.\ \mathsf{S}m < n \to l_{\mathsf{S}m} < l_m,$$
$$G_1 := |l| = n, \qquad G_3 := G_2(f,l,n,r) := \forall m.\ m < n \to fl_m = r.$$

For all $i \in \{1,2,3\}$, G_i does not contain $\bot$, so $G_i^F := G_i$. D is, however, relevant and with Lemma 2.2

$$D^F \to D. \tag{3.1}$$

Since $f\ k < 2$ is a prime formula, thus decidable, we can apply Lemma 2.1

$$L_{CaseDist}(f,k) := ((f\ k < 2 \to \mathrm{F}) \to \bot) \to (f\ k < 2 \to \bot) \to \bot.$$

and obtain (3.1) by

$$M_{D \to D^F} := \lambda f \lambda u^{D^F} \lambda k \lambda v_1^{f\ k<2 \to \bot}.\ L_{CaseDist}(f,k)\ (\lambda w^{f\ k<2 \to \mathrm{F}}.\ \mathsf{efq}^{F \to \bot}\ (u\ w))\ v_1$$

with the corresponding extracted term

$$[\![M_{D \to D^F}]\!] := \lambda f \lambda k \lambda v_1.\ [\![L_{CaseDist}(f,k)]\!]\ [\![\mathsf{efq}]\!]\ v_1$$
$$:= \lambda f \lambda k \lambda v_1\ \text{if}\ (f\ k < 2)\ \text{then}\ [\![\mathsf{efq}]\!]\ \text{else}\ v_1$$

The program extracted from the derivation M_{cor} of Corollary 1 is

$$[\![M_{cor}]\!] := \lambda f \lambda v \lambda n \lambda u.\ [\![M_{SP}]\!]\ f\ v\ (\lambda r \lambda w.\ [\![M_{\mathrm{SP}Aux}]\!]\ f\ r\ w\ n\ \lambda l.\ u\ l)$$
$$[\![M_{SP}]\!] := \lambda f \lambda v \lambda w.\ w\ 0\ [\![M_{\infty_f(0)}]\!]$$
$$[\![M_{\infty_f(0)}]\!] := \lambda n \lambda w_0.\ w\ 1\ [\![M_{\infty_f(1)}]\!]$$
$$[\![M_{\infty_f(1)}]\!] := \lambda m \lambda w_1.\ v\ (m \sqcup n)(L_{\mathsf{Cases}}\ (f\ (m \sqcup n))\ 1\ (w_0\ (m \sqcup n))(w_1\ (m \sqcup n)))$$
$$[\![M_{step}]\!] := \lambda n.\ \mathsf{if}\ (n = 0)\ [\![M_{n=0}]\!][\![M_{\mathsf{S}n}]\!]$$
$$[\![M_{n=0}]\!] := \lambda u_1.\ u_1(\lambda l.\ w\ 0\ \lambda k(k{:}))\ [\![\mathsf{efq}]\!]$$
$$[\![M_{\mathsf{S}n}]\!] := \lambda n \lambda u_1 \lambda u_2.\ u_1(\lambda l_2.\ \mathsf{if}\ (l_2 = \mathsf{nil})\ [\![\mathsf{efq}]\!]\ [\![M_{n_1 :: l}]\!])$$
$$[\![M_{n_1 :: l}]\!] := \lambda n_1 \lambda l.\ w\ (\mathsf{S}n_1)\ \lambda k_1.\ u_2\ (k_1 :: n_1 :: l)\ (\lambda m.\ m)(\lambda m.\ m)$$

Putting it all together and normalising, we obtain with (2.5) the following program, as output by the Minlog proof system (see the Minlog website)

```
lambda f,n1.n
 Rec n1
  (lambda x. x (Nil nat))
  (lambda n2,y1,x2.
    [if n2
      (y1 (lambda l3. R(T(0,0:,0:,x2,x3), T(S n3,(S n3):,(S n3::n3::l1),x2,x3))))
      (lambda n4.
        y1 (lambda l4.
             [if l4 (Nil nat)
                    (lambda n5,l8.
                      R(T(S n5,(S n5::n5::l8),(S n5):,x2,x3),
                        T(max(n3,n5),(S max(n3,n5)::n5::l8),
                                     (S max(n3,n5)::n3::l1),x2,x3)))]))])
  (lambda l. l)
```

with

```
R(T1,T2):= Rec n1 (lambda x. x (Nil nat)) Step (lambda l.l)
    Step:=(lambda k1,y2,x3.
             [if n (y2 (lambda l. T1))
                   (lambda k2.
                     y2 (lambda l.[if l (Nil nat) (lambda n3,l1. T2)]))])
```

and

```
T(m,l1,l2,xi,xj) := [if (f m < 2)
                       [if (f  m) (xi l1)
                                  (lambda m. [if m (xj l2) (lambda n0. (Nil nat))])]
                       (Nil nat)]
```

where $xi^{\mathtt{LN}\Rightarrow\mathtt{LN}}$ $yi^{(\mathtt{LN}\Rightarrow\mathtt{LN})\Rightarrow\mathtt{LN}}$ $Rec := \mathcal{R}^{\mathtt{N}\Rightarrow(\mathtt{LN}\Rightarrow\mathtt{LN})\Rightarrow\mathtt{LN}}$

As expected, all the case distinctions are reflected in the program, which quests recursively the given sequence for the indices at which the values are identical. We analyse the algorithm in terms of a few examples.

Example 1. We summarise our experiments in Table 1 bellow. f is the input sequence and n the length of the subsequence.

The first thing to observe, by comparing E.g. 3 and 4 or E.g. 7 and 8, is that there is a break of symmetry. This due to the fact that we are dependent of the choice made in the proof of whether to first search the subsequence coloured by 0 or for the one coloured by 1.

Let us analyse the behaviour of our program for the more interesting case of $n = 3$, depicted in the 4^{th} column. Suppose that we first ask for the (monochromatic) 0-subsequence. In this case, the 3 occurrences of 1 are selected only if they appear consecutively in the given sequence, which suggests that the program "stubbornly" investigates the tape to find the 3 indices at which 0 occurs.

E.g.	f	n = 2	n = 3
1	00 ... 0 ...	1::0:	2::1::0:
2	11 ... 1 ...	1::0:	2::1::0:
3	010100...	2::0:	4::2::0:
4	101011...	3::1:	6::5::4:
5	101100...	3::2:	5::4::1:
6	01101110...	2::1:	6::5::4:
7	01101101...	2::1:	6::3::0:
8	10010010 ..	2::1:	4::2::1:

Table 1. Tests

This is the case with E.g. 1,3,5, 7 and 8, where the 0's are selected, even though in E.g. 5 and 7 the 1's occur 3 times much sooner. However, in E.g. 6, the search is interrupted after finding the 3 consecutive 1's, although later a 3^{rd} 0 appears.

An important feature is that the subsequence of length 3 is not necessarily selected out of the infinite subsequence. In E.g. 7, for instance, the infinite subsequence is the one coloured by 1, yet the 0's (appearing only 3 times) are selected. It would be of interest to decouple the realization of (SP) from its corollary and only later specialise it to n. A combination of A-translation with external realizers is presented in Seisenberger (2003) discussed in the next session.

4 Related Work

The lemma that we address in this paper is not a new case study in the realm of program extraction from proofs. The novelty of this paper consists in investigating its proof purely by means of the refined A-Translation.

Seisenberger (2003) investigates a different proof, which uses the axiom of dependent choice, the emphasis being on the use of external realizers and the way the refined A-Translation can be coupled with the modified bar recursion. Naturally, this introduces new realizers for bar recursion in the extracted term, making it slightly more complicated.

One difference arising from using a different proof is that our algorithm starts the search in the natural way at the first element of the list. Due to the use of dependent choice, Seisenberger (2003) reports that the first element is skipped.

Seisenberger (2003) treats the case of subsequences of length two, so since we treat here the more general case, we refrain from a minute comparison. We only point out that the asymmetry is encountered in both works. This is due to the choice which is made already in the proof of (Lemma 1.2) as to first quest for monochromatic sequences coloured by 0 or for the 1-coloured subsequences.

The asymmetry has been pointed out also in Urban (2000), where the same proof as the one we presented here is analysed, although in its simplified version of determining subsequences of length 2. Whereas there is a similitude in the programs for the simplified case, the asymmetry reported in Urban (2000) arises

in the process of normalisation by cut-elimination, which is the method used for retrieving the computational content from classical proofs. Urban (2000) argues that there exist distinct normal forms corresponding to one proof, since in the process of cut-reduction, one has the freedom in which direction to propagate the cuts, so which reductions to apply. Such a non-deterministic choice gives rise to distinct programs, corresponding to the same proof. However, in the case of double negation or A-Translation, we need to make this choice at the proof level, because such transformations methods preserve the structure of the proof.

5 Conclusions and Future Work

We have presented in this paper a successful application of the refined A-Translation to Stolzenberg's Principle, which is a simplified version of the Infinite Pigeon Hole Principle. We have pointed out that the computational content of (IPH) is an important component of the algorithm hidden in the classical proof Ramsey's Theorem and the work reported here is but a first step.

(SP) can be generalised to the (IPH). The proof of the latter is by induction over the number of colours, the step case using the same argument as (SP). More precisely, one makes a case distinction on whether the colour $r+1$ appears infinitely often and in case it does, the claim follows by SPAux. In the other case, we take the subsequence starting at the last occurrence of $r+1$, say n', which is guaranteed to have r colours and thus gives us the monochromatic subsequence by the (IH). We do this by taking a copy of the initial sequence, with the first positions up to n', replaced by the colour at $n'+1$, i.e, $f := \lambda n.\, f(n \sqcup n')$. It would be interesting to see what is the increase in complexity of this program and what causes it as the proof level. Clearly, one reason is the outer induction on the number of colours.

On the other hand, one can choose to formulate the goal differently

$$\forall n\, \tilde{\exists}\mathbf{r}\, \tilde{\exists} l.\, |l| = n \,\tilde{\wedge}\, (\forall m.\, \mathsf{S}m < n \to l_{\mathsf{S}m} < l_m) \,\tilde{\wedge}\, (\forall m.\, m < n \to f l_m = r)$$

By this we would produce also the colour of the monochromatic sequence, which is however of no relevance. We see this as a potential case study for the uniform quantifiers presented in Berger (2005), which mark variables as computationally irrelevant.

We are currently also comparing in a joint work with Trifon Trifonov the current results for the A-Translation with those obtained by Dialectica.

Acknowledgements The author would like to address special thanks to the anonymous reviewers for their careful reading and very many useful comments.

Bibliography

S. Berardi. Some intuitionistic equivalents of classical principles for degree 2 formulas. *Annals of Pure and Applied Logic*, 139:185–200, 2006.

U. Berger. Uniform heyting arithmetic. *Ann. Pure Appl. Logic*, 133(1-3):125–148, 2005.

U. Berger, W. Buchholz, and H. Schwichtenberg. Refined program extraction from classical proofs. *Annals of Pure and Applied Logic*, 114:3–25, 2002.

H. Friedman. Classically and intuitionistically provably recursive functions. In D. Scott and G. Müller, editors, *Higher Set Theory*, volume 669 of *Lecture Notes in Mathematics*, pages 21–28. Springer Verlag, Berlin, Heidelberg, New York, 1978.

G. Gentzen. Untersuchungen über das logische Schließen. *Mathematische Zeitschrift*, 39:176–210, 405–431, 1934.

K. Gödel. Zur intuitionistischen arithmetik und zahlentheorie. *Ergebnisse eines mathematischen Kolloquiums*, 4:34–38, 1933.

K. Gödel. über eine bisher noch nicht benutzte erweiterung des finiten standpunktes. *Dialectica*, pages 12:280–287, 1958.

L. Halbeisen. Combinatorial set theory with a gentle introduction to forcing. Monograph. To appear, 2008.

H. Schwichtenberg. Content in proofs of list reversal. *Proceedings of the Marktoberdorf Summer School*, 2007.

M. Seisenberger. *On the Constructive Content of Proofs*. PhD thesis, Mathematisches Institut der Universität München, 2003. PhD thesis, Supervised by Helmut Schwichtenberg.

C. Urban. *Classical Logic and Computation*. PhD thesis, University of Cambridge, Oktober 2000.