An arithmetic for non-size-increasing polynomial-time computation

Klaus Aehlig *

Abstract

An arithmetical system is presented in such a way that from every proof a realizing term can be extracted that is definable in a certain affine linear typed variant of Gödel's T and therefore is a non-size-increasing polynomial time computable function with time complexity $O(X^n)$ if the proof contains at most n nested uses of induction.

1 Introduction

In [5] Hofmann presented a restriction of Gödel's T allowing iteration in all types with the property that all definable functions are non-size-increasing polynomial-time computable. Gödel's T [4] is related to Heyting Arithmetic via realizability interpretation. If, for example, $\forall x^\tau \exists^* y^\rho A$ with A quantifier free can be proved, then there exists a term of type $\tau \to \rho$ in Gödel's T calculating the y for a given x. For more details on realizability interpretations see [8, III§4].

We suggest a restriction of Heyting Arithmetic with the property that every formula can be realized by a term in Hofmann's restriction of Gödel's T. As in this system every term is polynomial time computable and the degree of the polynomial is bound by the number of nested $\mathbf{N}$–Eliminations [1], the degree of the polynomial describing the time complexity for the realizing term will be bound by the number of nested inductions. The system is based on natural deduction and the realizability interpretation is s straightforward extension of the standard one. Moreover the correctness of the realizability statement can be proved within the proposed system.

Similar systems have been recently proposed for the class of *all* polynomial time computable functions [2, 7]. Although capturing a larger class of functions these systems have the disadvantage of being rather complex [2] or indirectly defined [7] and moreover, these systems do not lead to obvious polynomial bounds.

*Universität München, Mathematisches Institut, Theresienstraße 39, D-80333 München, Germany; Tel +49 89 2394 4495; Fax +49 89 2805 248; aehlig@rz.mathematik.uni-muenchen.de

2 Types and terms

As basic term system we use an affine linearly typed restriction of Gödel's T in the formulation of [1]. As this system is described elsewhere [5, 1] we just recall the definitions.

Types are built from the base types $\mathbf{N}$ (to be interpreted as binary coded natural numbers), bool, and a special type $\diamond$ that can be used to ensure free variables in terms. For example the type $\diamond \multimap \mathbf{N} \multimap \mathbf{N}$ of the successor functions together with the linear typing discipline will make sure that the usual length measure for natural numbers is bound by the number of free variables. As the term system is based on the intuition that length is expressed by the number of free variables, the interesting terms are not only closed terms, but more importantly "almost closed" terms, i.e. terms with free variables of type $\diamond$ only.

Definition 2.1 (Finite linear types). The set of linear types is defined inductively as

$$\tau, \rho ::= \ \diamond \mid \mathbf{N} \mid \mathrm{bool} \mid \tau \multimap \rho \mid \tau \otimes \rho \mid \tau \times \rho \mid \mathbf{L}(\tau)$$

$\mathbf{L}(\tau)$ denotes the list type. $\tau \otimes \rho$ and $\tau \times \rho$ can both be interpreted as ordered pairs; however the meaning of linearity for those types is different: From a tensor-product $\tau \otimes \rho$ both components can be used once, whereas in the case of an ordinary pair, the pair itself can be used only once, i.e. one has to choose which component of the pair is to be used. As can be seen from 2.7, $\otimes$ corresponds to $\multimap$, whereas $\times$ corresponds to bool.

Terms are built from variables $x, y, z, \ldots$ and constructor symbols c. Each variable has a type and it is assumed that there are infinitely many variables of every type. The notation x^τ should express that the variable x has type τ.

Definition 2.2 (Terms). The set of terms is inductively defined:

$$t, s ::= \ x^\tau \mid c \mid \lambda x^\tau\, t \mid \langle t, s\rangle \mid ts \mid \{t\} \mid \{t, s\}$$

These terms should be seen as our "raw syntax"; only correctly typed terms (in the sense of definition 2.4) represent meaningful functions.

Terms of the form $\{\cdot\}$ and $\{\cdot,\cdot\}$ appear only as argument in an $\mathbf{N}$ or $\mathbf{L}(\tau)$ elimination where they mark the step terms in an iteration. This allows one to see the time complexity of a term at first glance: The constant coefficient of the polynomial defining the run time is simply the number of symbols outside any braces, the coefficient for X is number of symbols within one pair of braces and so on. For more details see [1].

The notations $t\vec{t}$ and $\lambda \vec{x}^{\vec{\tau}}\, t$ are defined as usual, so $\lambda x^\tau, y^\rho\, t$ is an abbreviation for $\lambda x^\tau (\lambda y^\rho\, t)$. Terms that only differ in the naming of bound variables are identified.

Definition 2.3 (Constructor symbols). The constructor symbols and their types are

$\mathbf{t}$	bool
$\mathbf{f}$	bool
$\mathbf{1}$	$\mathbf{N}$
$\mathbf{S}_0$	$\Diamond \multimap \mathbf{N} \multimap \mathbf{N}$
$\mathbf{S}_1$	$\Diamond \multimap \mathbf{N} \multimap \mathbf{N}$
$\mathbf{nil}_\tau$	$\mathbf{L}(\tau)$
$\mathbf{cons}_\tau$	$\Diamond \multimap \tau \multimap \mathbf{L}(\tau) \multimap \mathbf{L}(\tau)$
$\otimes_{\tau,\rho}$	$\tau \multimap \rho \multimap \tau\otimes\rho$

$\mathbf{1}$ is to be interpreted as the natural number one and the successor functions $\mathbf{S}_i$ as the usual binary successor functions, i.e. $x \mapsto 2\cdot x + i$.

A *context* is a set of variables. For two contexts Γ_1 and Γ_2 the notation Γ_1, Γ_2 stands for the union, expressing that Γ_1 and Γ_2 are disjoint. For contexts consisting of one element we also write x^τ instead of $\{x^\tau\}$.

The next definition states which terms of our raw syntax are correctly typed. $\Gamma \vdash t^\tau$ is to be read as "t is a typed term of type τ with free variables in Γ".

The typing system is based on elimination rules. Therefore, for every type different to $\Diamond$ there is a rule describing how to use terms of this type; in all these rules the elimination is written as (formal) application.

"Data types", i.e. types where it is possible to retrieve all the stored information by a single use of an object of this type, are introduced by constants. Only "abstractions" have special introduction rules. This way of introduction allows the relatively simple definition 2.8 of the term closure of reductions expressing that we may reduce within data, but not under abstractions.

Definition 2.4 ($\Gamma \vdash t^\tau$). The relation $\Gamma \vdash t^\tau$ is inductively defined:

$$\frac{}{\Gamma, x^\tau \vdash x^\tau} \quad (Var)$$

$$\frac{c \text{ of type } \tau}{\Gamma \vdash c^\tau} \quad (const)$$

$$\frac{\Gamma \cup \{x^\tau\} \vdash t^\rho}{\Gamma \vdash (\lambda x^\tau\, t)^{\tau\multimap\rho}} \quad (\multimap^+)$$

$$\frac{\Gamma_1 \vdash t^{\tau\multimap\rho} \qquad \Gamma_2 \vdash s^\tau}{\Gamma_1, \Gamma_2 \vdash (ts)^\rho} \quad (\multimap^-)$$

$$\frac{\Gamma \vdash t^\tau \qquad \Gamma \vdash s^\rho}{\Gamma \vdash \langle t, s\rangle^{\tau\times\rho}} \quad (\times^+)$$

$$\frac{\Gamma \vdash t^{\tau\times\rho}}{\Gamma \vdash (t\mathbf{t})^\tau} \quad (\times_1^-)$$

$$\frac{\Gamma \vdash t^{\tau\times\rho}}{\Gamma \vdash (t\mathbf{f})^\rho} \quad (\times_0^-)$$

$$\frac{\Gamma_1 \vdash t^{\mathrm{bool}} \qquad \Gamma_2 \vdash s^\tau \qquad \Gamma_2 \vdash r^\tau}{\Gamma_1, \Gamma_2 \vdash (t\,\langle s, r\rangle)^\tau} \quad (\mathrm{bool}^-)$$

$$\frac{\Gamma_1 \vdash t^{\tau\otimes\rho} \qquad \Gamma_2, x^\tau, y^\rho \vdash s^\sigma}{\Gamma_1, \Gamma_2 \vdash (t(\lambda x^\tau, y^\rho\, s))^\sigma} \quad (\otimes^-)$$

$$\frac{\Gamma \vdash t^{\mathbf{N}} \qquad \emptyset \vdash {s_i}^{\Diamond\multimap\tau\multimap\tau}}{\Gamma \vdash (t\,\{s_0, s_1\})^{\tau\multimap\tau}} \quad (\mathbf{N}^-)$$

$$\frac{\Gamma \vdash t^{\mathbf{L}(\tau)} \qquad \emptyset \vdash s^{\Diamond\multimap\tau\multimap\rho\multimap\rho}}{\Gamma \vdash (t\,\{s\})^{\rho\multimap\rho}} \quad (\mathbf{L}(\tau)^-)$$

It is crucial that we require the step terms in $(\mathbf{N}^-)$ and $(\mathbf{L}(\tau)^-)$ to be closed. Otherwise it would not be possible to define reduction rules in such a way that the system is closed under reduction since free variables in the step terms would be duplicated otherwise.

The notation t^τ should express that there is a Γ such that $\Gamma \vdash t^\tau$. The smallest such Γ is called the set of free variables. This is a meaningful definition; more precisely by induction on the definition of $\Gamma \vdash t^\tau$ one easily verifies

Lemma 2.5. *For each term t there is at most one type τ such that t^τ. In this case there is a smallest Γ with $\Gamma \vdash t^\tau$.*

We now define reduction rules on the terms. They are all correct with respect to the set-theoretic semantics. In order to be able to control the effects of recursion we allow reduction only if the argument is already calculated, i.e. if the argument is a numeral.

Definition 2.6 (Numerals, lists). Terms of the form $\mathbf{S}_{i_1} d_1^{\diamond}(\ldots(\mathbf{S}_{i_n} d_n^{\diamond}\mathbf{1}))$ are called *numerals*. *Lists* are terms of the form $\mathbf{cons}_\tau d_1^{\diamond} a_1^{\tau}(\ldots(\mathbf{cons}_\tau d_n^{\diamond} a_n^{\tau}\mathbf{nil}_\tau))$. (Here the $\vec{d}$ denote arbitrary terms of type $\diamond$.)

Definition 2.7 (Conversions). $\mapsto$ is defined as:

$$
\begin{array}{l}
(\lambda x^\tau\, t)s \mapsto t[s/x] \\
\langle t, s\rangle\, \mathbf{t} \mapsto t \\
\langle t, s\rangle\, \mathbf{f} \mapsto s \\
\mathbf{t}\, \langle t, s\rangle \mapsto t \\
\mathbf{f}\, \langle t, s\rangle \mapsto s \\
\otimes_{\tau,\rho} ts(\lambda x^\tau, y^\rho\, r) \mapsto r[t, s/x, y] \\
\mathbf{1}\, \{t_0, t_1\}\, s \mapsto s \\
\mathbf{S}_i d^\diamond \nu\, \{t_0, t_1\}\, s \mapsto t_i d^\diamond(\nu\, \{t_0, t_1\}\, s) \quad \nu \text{ a numeral} \\
\mathbf{nil}_\tau\, \{t\}\, s \mapsto s \\
\mathbf{cons}_\tau d^\diamond a\ell\, \{t\}\, s \mapsto t d^\diamond a(\ell\, \{t\}\, s) \quad \ell \text{ a list}
\end{array}
$$

Definition 2.8 ($t \to t'$). The term closure $t \to t'$ is inductively defined:

$$
\frac{t \mapsto t'}{t \to t'} \qquad \frac{t \to t'}{ts \to t's} \qquad \frac{s \to s'}{ts \to ts'}
$$

It has been shown [5], that every function definable by a closed term is non-size-increasing and polynomial time computable. Moreover [1], every normal almost closed term (i.e. with free variables of type $\diamond$ only) is a numeral and a normal form can be found in polynomial many reduction steps with the degree of the polynomial being the depth of nesting of $\{\cdot\}$s.

3 Formulae

We assume a fixed set of predicate symbols $P, Q, \ldots$ of fixed arrity. When writing $P(\vec{t})$ we implicitly assume correct length and typing of the $\vec{t}$. Moreover we assume a special predicate symbol $\approx_\tau$ for every type τ. We sometimes abbreviate $\approx_\tau (t, s)$ by $t \approx_\tau s$ or even by $t \approx s$.

Definition 3.1 (Formulae). The set of formulae is defined inductively.

$$A, B ::= \ P(\vec{t}) \mid A \wedge B \mid A \to B \mid \exists^* xA \mid \forall xA \mid \forall^* A$$

In the following we write $A, B, C, \ldots$ for formulae.

The conjunction is the "weak" one corresponding to the pair, i.e. $A \to B \to (A \wedge B \to C) \to C$ can be proven for any A, B and C but not $A \wedge B \to (A \to B \to C) \to C$.

The strong all-quantifier $\forall^*$ corresponds to the "underlined universal quantification" in [6] or the $\{\forall\}$ in [3] and marks all-quantifications with no computational content, i.e. a proof of $\forall^* x.A$ is of such a form that the realizing term does not depend on x.

4 Equality

We use two devices to handle equality: first, on the meta level we identify terms with the same $\mapsto$ normal form with $\mapsto$ being extended by allowing the reductions $\mathbf{S}_i d^\diamond \nu\, \{t_0, t_1\}\, s \mapsto t_i d^\diamond(\nu\, \{t_0, t_1\}\, s)$ and $\mathbf{cons}_\tau d^\diamond a\ell\, \{t\}\, s \mapsto t d^\diamond a(\ell\, \{t\}\, s)$ even if ν is not a numeral or ℓ is not a list. As there is at most on $\mapsto$ reduct, this identification is decidable.

Moreover we add the axiom schemes $s \approx t \to A[s/x] \to A[t/x]$ denoted by the constants $\boxminus_{A,x,s,t}$ and $t \approx t$ denoted by $\boxminus_t$.

5 Extracted types and Harrop formulae

For each formula we define the "computational content" $[\![A]\!]$, the type of a potential realizer or $\star$ if the formula does not have computational content. Formulas without computational content are also called Harrop formulae, or *irrelevant*, non-Harrop formulas are also called *relevant*.

Definition 5.1. The computational content $[\![A]\!]$ of a formula A is defined by induction on A:

$$
\begin{array}{lcl}
[\![P(\vec{t})]\!] & = & \star \\
[\![\exists^* x^\tau A]\!] & = & \begin{cases} \tau & [\![A]\!] = \star \\ \tau \otimes [\![A]\!] & \text{otherwise} \end{cases} \\
[\![\forall x^\tau A]\!] & = & \begin{cases} \star & [\![A]\!] = \star \\ \tau \multimap [\![A]\!] & \text{otherwise} \end{cases} \\
[\![\forall^* x^\tau A]\!] & = & [\![A]\!] \\
[\![A_0 \wedge A_1]\!] & = & \begin{cases} [\![A_i]\!] & [\![A_{1-i}]\!] = \star \\ [\![A_0]\!] \times [\![A_1]\!] & \text{otherwise} \end{cases} \\
[\![A \to B]\!] & = & \begin{cases} [\![B]\!] & [\![A]\!] = \star \\ \star & [\![B]\!] = \star \\ [\![A]\!] \multimap [\![B]\!] & \text{otherwise} \end{cases}
\end{array}
$$

6 Proof terms and proofs

Proof terms are intended to denote proofs in natural deduction style. However, as for the (object) terms we first define "raw terms" and a typing calculus. This is done in such a way that the denoted proofs are uniquely defined, as will show lemma 6.3.

Proof terms $N, M, L, \dots$ are built up from ordinary terms t, Axioms $\square$ ("black boxes for the user") and assumption variables $u, v, w, \dots$. Each of those assumption variables has a formula as type (in the sense of the Curry-Howard correspondence). For each formula there are infinitely many variables of this type. We write u^A to emphasize that the variable u has the type A.

Definition 6.1 (Proof terms).

$$\begin{array}{l} N, M ::= \ \square \mid u^A \mid \lambda u^A N \mid \lambda x^\tau N \mid \lambda^* x^\tau N \mid \langle N, M \rangle \\ \qquad\quad \mid NM \mid Nt \mid tN \mid \{N\} \mid \{N, M\} \end{array}$$

Generalized contexts (as opposed to contexts for the typing of terms) are sets of object and assumption variables. We denote generalized contexts by $\Psi, \Phi, \dots$. We write Ψ, Φ for the union expressing that Ψ and Φ are disjoint.

The notion of a proof is expressed by the relation $\Psi \,;\, \Phi \vdash N : A$. This is to be read as "N denotes a proof of A in the *normal* (generalized) context Ψ and the *essential* (generalized) context Φ". The essential context controls the variables free in the realizing terms. It should be noted that the normal context is defined as usual, i.e. when ignoring the essential context every rule is correct with respect to the normal Heyting Arithmetic. In particular, every formula provable in this restricted arithmetic is also provable in normal intuitionistic arithmetic and therefore true.

Definition 6.2. The relation $\Psi \,;\, \Phi \vdash N : A$ is inductively defined:

$$\frac{}{\emptyset \,;\, \emptyset \vdash \square^A : A} \quad (\mathit{axiom})$$

$$\frac{}{u^A \,;\, u^A \vdash u^A : A} \quad (\mathit{assumption})$$

$$\frac{\Psi, u^A \,;\, \Phi, u^A \vdash N : B}{\Psi \,;\, \Phi \vdash \lambda u^A.N : A \to B} \quad (\to^+)$$

$$\frac{\begin{array}{l}\Psi \,;\, \Phi_1 \vdash N : A \to B \\ \Psi \,;\, \Phi_2 \vdash M : A\end{array}}{\Psi \,;\, \Phi_1, \Phi_2 \vdash NM : B} \quad (\to^-)$$

$$\frac{\Psi, x^\tau \,;\, \Phi, x^\tau \vdash N : A \qquad \mathtt{Vars!}}{\Psi \,;\, \Phi \vdash \lambda x^\tau.N : \forall x^\tau A} \quad (\forall^+)$$

$$\frac{\begin{array}{l}\Psi \,;\, \Phi \vdash N : \forall x^\tau \\ \Gamma \vdash t^\tau \qquad \Gamma \subset \Psi\end{array}}{\Psi \,;\, \Phi, \Gamma \vdash Nt : A[t/x]} \quad (\forall^-)$$

$$\frac{\begin{array}{l}\Psi, x^\tau \,;\, \Phi \vdash N : A \\ x \notin \Phi \qquad \mathtt{Vars!}\end{array}}{\Psi \,;\, \Phi \vdash \lambda^* x^\tau.N : \forall^* x^\tau A} \quad (\forall^{*+})$$

$$\frac{\begin{array}{l}\Psi \,;\, \Phi \vdash N : \forall^* x^\tau \\ \Gamma \vdash t^\tau \qquad \Gamma \subset \Psi\end{array}}{\Psi \,;\, \Phi \vdash Nt : A[t/x]} \quad (\forall^{*-})$$

$$\frac{\Psi \,;\, \Phi \vdash N : A \qquad \Psi \,;\, \Phi \vdash M : B}{\Psi \,;\, \Phi \vdash \langle N, M \rangle : A \wedge B} \quad (\wedge^+)$$

$$\frac{\Psi \,;\, \Phi \vdash N : A \wedge B}{\Psi \,;\, \Phi \vdash N\mathbf{t} : A} \quad (\wedge_0^-)$$

$$\frac{\Psi \,;\, \Phi \vdash N : A \wedge B}{\Psi \,;\, \Phi \vdash N\mathbf{f} : B} \quad (\wedge_1^-)$$

Here `Vars!` is the usual condition on free variables, i.e. x must not be free in the type of any element of Ψ.

The next group of rules consists of $(\exists^{*-})$, $(\otimes - \mathit{Ind})$, $(\mathrm{bool} - \mathit{Ind})$, $(\mathbf{L}(\tau) - \mathit{Ind})$ and $(\mathbf{N} - \mathit{Ind})$.

$$\frac{\begin{array}{l}\Psi \,;\, \Phi_1 \vdash N : \exists^* x^\tau A \\ \Psi, x^\tau, u^A \,;\, \Phi_2, x^\tau, u^A \vdash M : B \\ \mathtt{Vars!}\end{array}}{\Psi \,;\, \Phi_1, \Phi_2 \vdash N(\lambda x^\tau, u^A\, M) : B}$$

$$\frac{\begin{array}{l}\Psi, x_1^\tau, x_2^\rho, u^{t \approx \otimes_{\tau,\rho} x_1^\tau x_2^\rho} \,;\, \Phi, x_1^\tau, x_2^\rho \vdash N : A[\otimes_{\tau,\rho} x_1 x_2 / x] \\ \Gamma \vdash t^{\tau \otimes \rho} \qquad \Gamma \subset \Psi \qquad \mathtt{Vars!}\end{array}}{\Psi \,;\, \Phi, \Gamma \vdash t(\lambda x_1, x_2, u\, N) : A[t/x]}$$

$$\frac{\begin{array}{l}\Psi \,;\, \Phi \vdash N : t \approx \mathbf{t} \to B \\ \Psi \,;\, \Phi \vdash M : t \approx \mathbf{f} \to B \\ \Gamma \vdash t^{\mathrm{bool}} \qquad \Gamma \subset \Psi\end{array}}{\Psi \,;\, \Phi, \Gamma \vdash t\,\langle N, M \rangle : B}$$

$$\frac{\begin{array}{l}\Psi \,;\, \emptyset \vdash N : \forall^* x^{\mathbf{L}(\tau)} \forall y^\diamond \forall z^\tau.A \to A[\mathbf{cons}_\tau yzx/x] \\ \Gamma \vdash t^{\mathbf{L}(\tau)} \qquad \Gamma \subset \Psi\end{array}}{\Psi \,;\, \Gamma \vdash t\,\{N\} : A[\mathbf{nil}_\tau/x] \to A[t/x]}$$

$$\frac{\begin{array}{l}\Psi \,;\, \emptyset \vdash N_i : \forall^* x^{\mathbf{N}} \forall y^\diamond.A \to A[\mathbf{S}_i yx/x] \\ \Gamma \vdash t^{\mathbf{N}} \qquad \Gamma \subset \Psi\end{array}}{\Psi \,;\, \Gamma \vdash t\,\{N_1, N_2\} : A[0/x] \to A[t/x]}$$

Here `Vars!` in ($\exists^{*-}$) stands for the restrictions that in x must not be free in B nor in the type of any of the assumption variables in Ψ. `Vars!` in ($\otimes - Ind$) is the restriction that x_1, x_2 must not be free in A nor in the type of any element of Ψ.

As weakening does *not* follow from the above rules we add it explicitly. Moreover we add a rule describing the meaning of the essential context: an essential context is only needed to prove relevant formulas.

$$\frac{\Psi\ ;\ \Phi\ \vdash N\ :\ A \qquad \Psi \subset \Psi' \qquad \Phi \subset \Phi'}{\Psi'\ ;\ \Phi'\ \vdash N\ :\ A} \quad (weakening)$$

$$\frac{\Psi\ ;\ \Phi\ \vdash N\ :\ A \qquad [\![A]\!] = \star}{\Psi\ ;\ \emptyset\ \vdash N\ :\ A} \quad (pruning)$$

Besides the axioms of equality we add an axiom to introduce existence. The constants $\boxtimes_{x,A}$ are axioms of the type $\forall x.A \to \exists^* x A$.

We write N^A for N if there are Ψ and Φ such that $\Psi\ ;\ \Phi\ \vdash N\ :\ A$.

Lemma 6.3. *For every proof term N there is at most one formula A, such that N^A. In this case there are smallest Ψ and Φ with $\Psi\ ;\ \Phi\ \vdash N\ :\ A$. For those Ψ and Φ is $\Phi \subset \Psi$.*

Proof. Immediately obvious by induction on N. □

7 Definition of realizability

We now define what it means for a term t to realise a formula A. The ituition of t beeing a programm calculating examples for existential quantifiers is formalised by the formula $t\,\underline{\mathbf{mr}}\,A$. To improve readability we use the abreviations $\pi_0 = \lambda x, y\quad x$ and $\pi_1 = \lambda x, y\quad y$ for the projections. Moreover we introduce a new symbol ε that is considered to be a closed term of "type $\star$".

Definition 7.1 ($\underline{\mathbf{mr}}$). By induction on A we define a formula $t\,\underline{\mathbf{mr}}\,A$ for arbitrary $t^{[\![A]\!]}$.

$$\varepsilon\,\underline{\mathbf{mr}}\,P(\vec{t}) \quad = \quad P(\vec{t})$$

$$r\,\underline{\mathbf{mr}}\,\exists^* A \quad = \quad \begin{cases} \varepsilon\,\underline{\mathbf{mr}}\,A[r/x] & [\![A]\!] = \star \\ r\pi_1\,\underline{\mathbf{mr}}\,A[r\pi_0/x] & \text{otherwise} \end{cases}$$

$$r\,\underline{\mathbf{mr}}\,\forall x A \quad = \quad \begin{cases} \forall x.\varepsilon\,\underline{\mathbf{mr}}\,A & [\![A]\!] = \star \\ \forall x.rx\,\underline{\mathbf{mr}}\,A & \text{otherwise} \end{cases}$$

$$r\,\underline{\mathbf{mr}}\,\forall^* x A \quad = \quad \forall x.r\,\underline{\mathbf{mr}}\,A$$

$$r\,\underline{\mathbf{mr}}\,(A \to B) \quad = \quad \begin{cases} \varepsilon\,\underline{\mathbf{mr}}\,A \to r\,\underline{\mathbf{mr}}\,B & [\![A]\!] = \star \\ \forall x.x\,\underline{\mathbf{mr}}\,A \to \varepsilon\,\underline{\mathbf{mr}}\,B & [\![A]\!] \neq \star,\ [\![B]\!] = \star \\ \forall x.x\,\underline{\mathbf{mr}}\,A \to rx\,\underline{\mathbf{mr}}\,B & \text{otherwise} \end{cases}$$

$$r\,\underline{\mathbf{mr}}\,(A \wedge B) \quad = \quad \begin{cases} \varepsilon\,\underline{\mathbf{mr}}\,A \wedge r\,\underline{\mathbf{mr}}\,B & [\![A]\!] = \star \\ r\,\underline{\mathbf{mr}}\,A \wedge \varepsilon\,\underline{\mathbf{mr}}\,B & [\![B]\!] = \star \\ r\mathbf{t}\,\underline{\mathbf{mr}}\,A \wedge r\mathbf{f}\,\underline{\mathbf{mr}}\,B & \text{otherwise} \end{cases}$$

From the definition we notice that $r\,\underline{\mathbf{mr}}\,A$ does not contain an $\exists^*$. Therefore

Proposition 7.2. $[\![r\,\underline{\mathbf{mr}}\,A]\!] = \star$

For each variable u^A with $[\![A]\!] \neq \star$ we choose a unique variable x_{u^A} that is sufficiently different to all variables used so far. If $[\![A]\!] = \star$ let $x_{u^A} = \varepsilon$. Moreover let $\widetilde{u^A}$ be a sufficiently new assumption variable for $x_{u^A}\,\underline{\mathbf{mr}}\,A$.

Definition 7.3 (extracted terms). The extracted term $[\![N]\!]$ of a proof N is defined by induction of N:

$$[\![\boxtimes_{x^\tau,A}]\!] \quad = \quad \begin{cases} (\lambda x^\tau\ x) & [\![A]\!] = \star \\ \otimes_{\tau,[\![A]\!]} & \text{otherwise} \end{cases}$$

$$[\![\boxminus_{A,x,s,t}]\!] \quad = \quad \begin{cases} \varepsilon & [\![A]\!] = \star \\ \lambda x^{[\![A]\!]}\ x & \text{otherwise} \end{cases}$$

$$[\![\boxminus_t]\!] \quad = \quad \varepsilon$$

$$[\![u^A]\!] \quad = \quad x_{u^A}$$

$$[\![\lambda u^A\, N^B]\!] \quad = \quad \begin{cases} \varepsilon & [\![B]\!] = \star \\ [\![N]\!] & [\![A]\!] = \star \neq [\![B]\!] \\ \lambda x_u\ [\![N]\!] & \text{otherwise} \end{cases}$$

$$[\![N^{A \to B} M]\!] = \begin{cases} \varepsilon & [\![B]\!] = \star \\ [\![N]\!] & [\![A]\!] = \star \neq [\![B]\!] \\ [\![N]\!]\ [\![M]\!] & \text{otherwise} \end{cases}$$

$$[\![\lambda x^\tau\, N^A]\!] \quad = \quad \begin{cases} \varepsilon & [\![A]\!] = \star \\ \lambda x\ [\![N]\!] & \text{otherwise} \end{cases}$$

$$[\![N^{\forall x.A} t]\!] \quad = \quad \begin{cases} \varepsilon & [\![A]\!] = \star \\ [\![N]\!]\,t & \text{otherwise} \end{cases}$$

$$\llbracket \lambda^* x^\tau N \rrbracket = \llbracket N \rrbracket$$

$$\llbracket N^{\forall^* x.A} t \rrbracket = \llbracket N \rrbracket$$

$$\llbracket \langle N, M \rangle \rrbracket = \begin{cases} \varepsilon & \llbracket A \rrbracket = \star = \llbracket B \rrbracket \\ \llbracket N \rrbracket & \llbracket A \rrbracket \neq \star = \llbracket B \rrbracket \\ \llbracket M \rrbracket & \llbracket A \rrbracket = \star \neq \llbracket B \rrbracket \\ \langle \llbracket N \rrbracket, \llbracket M \rrbracket \rangle & \text{otherwise} \end{cases}$$

$$\llbracket N^{A \wedge B} \mathbf{t} \rrbracket = \begin{cases} \varepsilon & \llbracket A \rrbracket = \star \\ \llbracket N \rrbracket & \llbracket B \rrbracket = \star \neq \llbracket A \rrbracket \\ \llbracket N \rrbracket \, \mathbf{t} & \text{otherwise} \end{cases}$$

$$\llbracket N^{A \wedge B} \mathbf{f} \rrbracket = \begin{cases} \varepsilon & \llbracket B \rrbracket = \star \\ \llbracket N \rrbracket & \llbracket A \rrbracket = \star \neq \llbracket B \rrbracket \\ \llbracket N \rrbracket \, \mathbf{f} & \text{otherwise} \end{cases}$$

$$\llbracket N^{\exists^* x A} (\lambda x, u\, M^B) \rrbracket = \begin{cases} \varepsilon & \llbracket B \rrbracket = \star \\ (\lambda x \ \llbracket M \rrbracket) \llbracket N \rrbracket & \llbracket A \rrbracket = \star \neq \llbracket B \rrbracket \\ \llbracket N \rrbracket (\lambda x, x_u \ \llbracket M \rrbracket) & \text{otherwise} \end{cases}$$

$$\llbracket t^{\tau \otimes \rho} (\lambda x_1, x_2, u\, N^A) \rrbracket = \begin{cases} \varepsilon & \llbracket A \rrbracket = \star \\ t(\lambda x_1, x_2 \ \llbracket N \rrbracket) & \text{otherwise} \end{cases}$$

$$\llbracket t^{\mathrm{bool}} \langle N^{t \approx \mathbf{t} \to A}, M \rangle \rrbracket = \begin{cases} \varepsilon & \llbracket A \rrbracket = \star \\ t \langle \llbracket N \rrbracket, \llbracket M \rrbracket \rangle & \text{otherwise} \end{cases}$$

$$\llbracket t^{\mathbf{L}(\tau)} \{ N^B \} \rrbracket = \begin{cases} \varepsilon & \llbracket B \rrbracket = \star \\ t \{ N \} & \text{otherwise} \end{cases}$$

$$\llbracket t^{\mathbf{N}} \{ N_0^B, N_1 \} \rrbracket = \begin{cases} \varepsilon & \llbracket B \rrbracket = \star \\ t \{ \llbracket N_0 \rrbracket, \llbracket N_1 \rrbracket \} & \text{otherwise} \end{cases}$$

Remark 7.4. As can be seen from the definition the extracted term contains braces only on places where the proof does as well. Therefore we may conclude from [1] that the realizing term is a polynomial time computable function with the degree of the polynomial bound by the depth of nested inductions.

For a generalized context Ψ let $\overline{\Psi}$ be Ψ with every u^A replaced by the two variables x_{u^A} and $\widetilde{u^A}$ if $\llbracket A \rrbracket \neq \star$ and replaced only by $\widetilde{u^A}$ otherwise. Moreover let $\llbracket \vec{N} \rrbracket$ be short for $\llbracket N_1 \rrbracket, \ldots, \llbracket N_n \rrbracket$.

8 Soundness of the realizability interpretation

To simplify notation we extend the relation $\cdot \vdash \cdot$ by $\Gamma \vdash \varepsilon^\star$ for arbitrary Γ. Moreover define $\varepsilon, \Gamma = \Gamma, \varepsilon = \Gamma$.

Theorem 8.1 (Soundness of typing). $\vec{u}, \vec{x} \ ; \ \vec{v}, \vec{y} \ \vdash N : A \Longrightarrow \llbracket \vec{v} \rrbracket, \vec{y} \vdash \llbracket N \rrbracket^{\llbracket A \rrbracket}$

Proof. Induction on $\vec{u}, \vec{x} \ ; \ \vec{v}, \vec{y} \vdash N : A$ ∎

Theorem 8.2 (Soundness). *Assume $\Psi \ ; \ \Phi \ \vdash N : A$ with $\Phi \subset \Psi$. Then there is a proof term $\mu(N)$ with $\overline{\Psi} \ ; \ \emptyset \ \vdash \mu(N) : \llbracket N \rrbracket \ \underline{\mathbf{mr}}\, A$.*

Proof. Tedious but elementary induction on the definition of $\Psi \ ; \ \Phi \ \vdash N : A$, using the fact that, due to proposition 7.2 it is possible to use (*pruning*).

For example

$(\exists^{*-})$ **with $\llbracket A \rrbracket \neq \star$.** By Induction Hypothesis we have proof terms $\mu(N)$ and $\mu(M)$ with $\overline{\Psi} \ ; \ \emptyset \ \vdash \mu(N) : \llbracket N \rrbracket \pi_1 \underline{\mathbf{mr}}\, A[\llbracket N \rrbracket \pi_0 / x]$ and $\overline{\Psi}, x^\tau, x_{u^A}, \widetilde{u^A} \ ; \ \emptyset \ \vdash \mu(M) : \llbracket M \rrbracket \ \underline{\mathbf{mr}}\, B$. From $\mu(M)$, equality axioms and an assumption v of $\llbracket N \rrbracket \approx \otimes_{\tau, \llbracket A \rrbracket} x_1 x_2$ we get a proof $\mu(N)' = \boxminus_{x' \pi_1 \underline{\mathbf{mr}} A[x' \pi_0 / x], x', \llbracket N \rrbracket, \otimes_{\tau, \llbracket A \rrbracket} x_1 x_2} v \mu(N)$ with $\overline{\Psi}, v, x_1, x_2 \ ; \ \emptyset \ \vdash \mu(N)' : x_2 \underline{\mathbf{mr}}\, A[x_1/x]$. Because of the variable conditions in $(\exists^{*-})$ substitution yields $\overline{\Psi}, x_1, x_2, \widetilde{u}' \ ; \ \emptyset \ \vdash \mu(M)[x_2, x_1 / x_u, x] : \llbracket M \rrbracket [x_2, x_1 / x_u . x] \ \underline{\mathbf{mr}}\, B$ By $(\to^+)$, $(\to^-)$ and $\mu(N)$ we get $\overline{\Psi}, v, x_1, x_2 \ ; \ \emptyset \ \vdash (\lambda \widetilde{u}. \mu(M)[x_2, x_1 / x_u, x]) \mu(N)' : \underbrace{\llbracket M \rrbracket [x_2, x_1 / x_u . x]}_{\otimes_{\tau, \llbracket A \rrbracket} x_1 x_2 (\lambda x x_u . \llbracket N \rrbracket)}$ and $(\otimes - Ind)$ completes the proof.

$(\mathbf{L}(\tau) - Ind)$ **with $\llbracket A \rrbracket \neq \star$.** By induction hypothesis we have $\overline{\Psi} \ ; \ \emptyset \ \vdash \mu(N) : \forall x^{\mathbf{L}(\tau)} \forall y_1^\diamond \forall y_2^\tau \forall z^{\llbracket A \rrbracket} . z \ \underline{\mathbf{mr}}\, A \to \llbracket N \rrbracket y_1 y_2 z \ \underline{\mathbf{mr}}\, A[\mathbf{cons}_\tau y_1 y_2 x / x]$. We have to show $t \{ \llbracket N \rrbracket \} \ \underline{\mathbf{mr}} (A[\mathbf{nil}_\tau / x] \to A[t/x])$. We do this by $(\mathbf{L}(\tau) - Ind \quad t)$. So we need a proof of

$$\begin{array}{rl} & \forall^* x^{\mathbf{L}(\tau)} \forall y_1^\diamond \forall y_2^\tau . \\ & x \{ \llbracket N \rrbracket \} \ \underline{\mathbf{mr}} (A[\mathbf{nil}_\tau / x] \to A) \\ \to & \mathbf{cons}_\tau y_1 y_2 x \{ \llbracket N \rrbracket \} \ \underline{\mathbf{mr}} (A[\mathbf{nil}_\tau / x] \to \\ & \qquad A[\mathbf{cons}_\tau y_1 y_2 x / x]) \\ = & \forall^* x^{\mathbf{L}(\tau)} \forall y_1^\diamond \forall y_2^\tau . \\ & (\forall z^{\llbracket A \rrbracket} . z \ \underline{\mathbf{mr}} \ A[\mathbf{nil}_\tau / x] \to x \{ \llbracket N \rrbracket \} z \ \underline{\mathbf{mr}}\, A) \\ \to & (\forall z^{\llbracket A \rrbracket} . z \ \underline{\mathbf{mr}} \ A[\mathbf{nil}_\tau / x] \to \\ & \underbrace{\mathbf{cons}_\tau y_1 y_2 x \{ \llbracket N \rrbracket \} z}_{\llbracket N \rrbracket y_1 y_2 (x \{ \llbracket N \rrbracket \} z)} \ \underline{\mathbf{mr}}\, A[\mathbf{cons}_\tau y_1 y_2 x / x]) \end{array}$$

A proof for this formula is $\lambda^* x^{\mathbf{L}(\tau)} \lambda y_1^{\diamond} \lambda y_2^{\tau} \lambda u \lambda z^{[\![A]\!]} \lambda v^{z \, \underline{\mathbf{mr}} \, A}$ $\mu\,(N)\, x y_1 y_2 (x \,\{[\![N]\!]\}\, z) \quad (uvz)$, with u a new assumption variable for $\forall z. z \, \underline{\mathbf{mr}} \, A[\mathbf{nil}_\tau / x] \to x \,\{[\![N]\!]\}\, z \, \underline{\mathbf{mr}} \, A$.

Moreover it is easy to prove $\mathbf{nil}_\tau \,\{[\![A]\!]\}\, \underline{\mathbf{mr}} \,(A[\mathbf{nil}_\tau/x] \to A[\mathbf{nil}_\tau/x])$.

$\square$

9 Example: append

The predicate $\mathtt{APP}(l, l', l'')$ meaning that l'' is the concatenation of l and l' is defined as usual, i.e. let $\square_{\mathrm{app},1}$ and $\square_{\mathrm{app},2}$ be the all-closures of

$$\mathtt{APP}(\mathbf{nil}_\tau, l^{\mathbf{L}(\tau)}, l^{\mathbf{L}(\tau)})$$

$$\mathtt{APP}(l^{\mathbf{L}(\tau)}, l'^{\mathbf{L}(\tau)}, l''^{\mathbf{L}(\tau)}) \to \mathtt{APP}(\mathbf{cons}_\tau x^{\diamond} y^{\tau} l, l', \mathbf{cons}_\tau x y l'')$$

We show the termination of this logic program. That is, we give a closed proof of $\forall l'^{\mathbf{L}(\tau)}, l^{\mathbf{L}(\tau)} \exists^* l''^{\mathbf{L}(\tau)} \mathtt{APP}(l, l', l'')$.

> Let l' be given. Argue by induction on l. In the step case conclude from $\square_{\mathrm{app},2}$ and the induction hypothesis. The initial case follows from $\square_{\mathrm{app},1}$.

As this proof uses only one induction we know from remark 7.4 that the extracted program has linear time complexity. Indeed, the formalized proof is

$$\begin{aligned}\lambda l', l \quad & l\{\lambda^* l \lambda x^{\diamond}, z^{\tau}, u^{\exists^* l'' \mathtt{APP}(l,l',l'')}\; u(\lambda l'', v^{\mathtt{APP}(l,l',l'')} \\ & \boxtimes_{l'', \mathtt{APP}(l,l',l'')} (\mathbf{cons}_\tau x z l'')(\square_{\mathrm{app},2} l l' l'' x z v))\} \\ & (\boxtimes_{l'', \mathtt{APP}(\mathbf{nil}_\tau, l', l'')} l' \square_{\mathrm{app},1})\end{aligned}$$

and therefore the extracted program is

$$\begin{aligned}\lambda l', l \quad l\{\lambda x^{\diamond}, z^{\tau}, x_u^{\mathbf{L}(\tau)} \quad & (\lambda l'' \\ & (\lambda y^{\mathbf{L}(\tau)}\; y)(\mathbf{cons}_\tau x z l''))x_u\} \\ & ((\lambda y^{\mathbf{L}(\tau)}\; y) l')\end{aligned}$$

References

[1] K. Aehlig and H. Schwichtenberg. A syntactical analysis of non-size-increasing polynomial time computation. In *Proceedings of the Fifteenth IEEE Symposium on Logic in Computer Science (LICS '00)*, June 2000. To appear.

[2] S. Bellantoni and M. Hofmann. A New "Feasible" Arithmetic. Submitted, `www.dcs.ed.ac.uk/home/mxh/sjb.ps`, Jan. 2000.

[3] U. Berger. Program extraction from normalization proofs. In M. Bezem and J. Groote, editors, *Typed Lambda Calculi and Applications*, volume 664 of *Lecture Notes in Computer Science*, pages 91–106. Springer Verlag, Berlin, Heidelberg, New York, 1993. `www.mathematik.uni-muenchen.de/~berger/articles/tlca/mr.dvi.Z`.

[4] K. Gödel. Über eine bisher noch nicht benützte Erweiterung des finiten Standpunkts. *Dialectica*, 12:280–287, 1958.

[5] M. Hofmann. Linear types and non-size-increasing polynomial time computation. In *Proceedings 14'th Symposium on Logic in Computer Science (Lics'99)*, pages 464–473, 1999.

[6] J. v. d. Pol. *Termination of Higher-order Rewrite Systems.* PhD thesis, Utrecht University, 1996.

[7] H. Schwichtenberg. Feasible program from proofs. Submitted, `www.mathematik.uni-muenchen.de/~schwicht/papers/specker00/fp.ps.Z`, January 2000.

[8] A. S. Troelstra, editor. *Metamathematical Investigation of Intuitionistic Arithmetic and Analysis*, volume 344 of *Lecture Notes in Mathematics.* Springer Verlag, Berlin, Heidelberg, New York, 1973.