A logic for nonincreasing PTIME computation

Ulrich Berger, Martin Hofmann, Helmut Schwichtenberg

1 Introduction

We wish to solve the following equation:

$$\frac{\text{Heyting Arithmetic}}{\text{Gödel's T}} = \frac{???}{\text{Calculus from M.H.'s LICS 1999 paper}}$$

In other words we seek a logic whose Π^0_2-proofs can be realised by terms in M.H.'s system and thus in particular have non-size-increasing polytime Skolem Functions.

Let us call M.H.'s system $\mathcal{C}$ (in honour of Caseiro) for further reference.

As a first approximation we aim for a first-order logic which does not have all of $\mathcal{C}$ in its term language but merely the constructor symbols. This will allow us to stick to plain first-order logic. Higher-order constructions are accommodated indirectly through the use of complex formulas in inductions.

2 Definition of the logic

We start from a usual first order signature.

It should contain in particular constructor symbols for the desired datatypes with their usual arities. For example, to accommodate unary natural numbers, booleans, and lists we would include constants $0, \mathsf{nil}, \mathsf{tt}, \mathsf{ff}$, unary function symbol S, binary function symbol cons.

The predicate symbols include a propositional constant $\diamond$, and for each sort A a unary "existence predicate" E_A.

The sorts are built up in the usual way from the data type constructors we consider. E.g. in the example case above the sorts would be given by

$$A ::= \mathsf{N} \mid \mathsf{B} \mid \mathsf{L}(A)$$

Notice that at this point we do not allow lists of pairs or functions. Maybe at a subsequent stage of development we will allow this extra generality.

The formulas are built up in the usual way using the connectives $\rightarrow, \otimes, \wedge, \vee$, and the quantifiers $\exists, \forall$.

A sequent is an expression of the form $\Phi \Longrightarrow \varphi$ where Φ is a *multiset* of formulas and φ is a formula.

Semantic validity of sequents is as usual with the understanding that the two conjunctions $\otimes, \wedge$ are identified. The intended interpretation of the signature is such that the domain of individuals D allows for an injective interpretation of the constructor symbols. The existence predicates are interpreted as the subset of the domain inductively generated by the constructors, e.g. E_{N} would consist of numerals $\mathsf{S}^n(0)$.

If A is a sort we will write $[\![A]\!]$ for $\{x \in D \mid E_A(x)\}$. This is consistent with the set-theoretic semantics of types given in M.H.'s LICS paper when restricted to sorts.

The propositional constant $\diamond$ is interpreted as truth.

3 Proof system

The proof rules are as follows:

$$\frac{\varphi \in \Phi \qquad \text{or } \varphi \text{ is an axiom}}{\Phi \Longrightarrow \varphi} \tag{Ax}$$

$$\frac{\Phi, \varphi \Longrightarrow \psi}{\Phi \Longrightarrow \varphi \to \psi} \tag{Arr-I}$$

$$\frac{\Phi \Longrightarrow \varphi \to \psi \qquad \Psi \Longrightarrow \varphi}{\Phi \cup \Psi \Longrightarrow \psi} \tag{Arr-E}$$

$$\frac{\Phi \Longrightarrow \varphi \qquad x \text{ not free in } \Phi}{\Phi \Longrightarrow \forall x.\varphi} \tag{All-I}$$

$$\frac{\Phi \Longrightarrow \forall x.\varphi}{\Phi \Longrightarrow \varphi[t/x]} \tag{All-E}$$

$$\frac{\emptyset \Longrightarrow \varphi(0) \qquad \emptyset \Longrightarrow \forall x.\Diamond \to \varphi(x) \to \varphi(\mathsf{S}(x))}{\Phi \Longrightarrow \forall x.E_{\mathsf{N}}(x) \to \varphi(x)} \tag{Ind-N}$$

$$\frac{\emptyset \Longrightarrow \varphi(\mathsf{nil}) \qquad \emptyset \Longrightarrow \forall x.\forall a.\Diamond \to E_A(a) \to \varphi(x) \to \varphi(\mathsf{cons}(a,x))}{\Phi \Longrightarrow \forall x.E_{\mathsf{L}(A)}(x) \to \varphi(x)} \tag{Ind-L}$$

Similar induction rules have to be added if further type formers are desired.

Definition 3.1 *The atomic formulas $\Diamond$ and $E_A(t)$ are computationally relevant, all other atomic formulas are computationally irrelevant. If a formula φ is computationally irrelevant so are $\forall x.\varphi, \exists x.\varphi$ and $\psi \to \varphi$ for arbitrary ψ. If φ and ψ are both computationally irrelevant, so is $\varphi \star \psi$ for $\star \in \{\wedge, \otimes\}$. Notice that disjunctions are always computationally relevant.*

The axioms are

- all true computationally irrelevant formulas or a suitable r.e. subset thereof.
- axioms asserting existence of the constructors,

$E_{\mathsf{N}}(0)$	E-Zero
$\Diamond \to E_{\mathsf{N}}(x) \to E_{\mathsf{N}}(\mathsf{S}(x))$	E-Succ
$E_{\mathsf{B}}(\mathsf{tt})$	E-True
$E_{\mathsf{B}}(\mathsf{ff})$	E-False
$E_{\mathsf{L}(A)}(\mathsf{nil})$	E-Nil
$\Diamond \to E_A(x) \to E_{\mathsf{L}(A)}(l) \to E_{\mathsf{L}(A)}(\mathsf{cons}(x,l))$	E-Cons

– axioms for the logical connectives:

$$\begin{array}{lr}
\varphi \to \psi \to \varphi \otimes \psi & \textsc{Tens-I} \\
(\varphi \to \psi \to \rho) \to \varphi \otimes \psi \to \rho & \textsc{Tens-E} \\
(\rho \to \varphi) \to (\rho \to \psi) \to \rho \to \varphi \wedge \psi & \textsc{Cart-I} \\
\varphi \wedge \psi \to \varphi & \textsc{Cart-E-1} \\
\varphi \wedge \psi \to \psi & \textsc{Cart-E-1} \\
\varphi \to \varphi \vee \psi & \textsc{Or-I-1} \\
\psi \to \varphi \vee \psi & \textsc{Or-I-2} \\
(\varphi \to \rho) \wedge (\psi \to \rho) \to \varphi \vee \psi \to \rho & \textsc{Or-E} \\
\forall x.\varphi \to \exists x.\varphi & \textsc{Ex-I} \\
(\forall x.\varphi \to \rho) \to (\exists x.\varphi) \to \rho & \textsc{Ex-E}
\end{array}$$

4 Remarks

The proof system is affine linear in the sense that the contraction rule is absent. For example, we can derive $\diamond, \diamond \Longrightarrow \diamond \otimes \diamond$, but we can *not* derive $\diamond \Longrightarrow \diamond \otimes \diamond$. It is, however, possible to derive $\diamond \Longrightarrow \diamond \wedge \diamond$. In Girard's terminology, $\otimes$ is multiplicative conjunction, whereas $\wedge$ is additive conjunction.

The additive conjunction is weaker than the multiplicative one in the sense that $\varphi \otimes \psi \to \varphi \wedge \psi$.

Accordingly, the axiom Or-E is stronger than it would be with $\otimes$ instead of $\wedge$.

In view of the existence axioms it is not possible to derive existence of nonzero terms without additional premises. For example, we can derive $\diamond, \diamond \Longrightarrow E_{\mathsf{N}}(\mathsf{S}(\mathsf{S}(0)))$, but we can *not* derive $\emptyset \Longrightarrow E_{\mathsf{N}}(\mathsf{S}(\mathsf{S}(0)))$.

The natural way to discharge such $\diamond$-assumptions is via the inductive case of an induction which provides access to one extra such assumption.

We notice that the inductive case of an induction must be proved without assumptions. Intuitively, this is so because the inductive case is used more than once in the unfolding of an induction. A restricted access to assumptions is provided by adding them to the induction hypothesis. In particular, in this way one can derive a more generous scheme which allows the base case $\varphi(0)$ to depend on assumptions.

We also note that the only purpose of the assumptions Φ in the conclusion of induction is to give weakening as a derived rule.

5 Examples

Assume a predicate symbol $\mathsf{ADD}(x, y, z)$ which is true provided that $x, y, z \in [\![\mathsf{N}]\!]$ and $z = x + y$. Our aim is to prove

$$\forall x.E_{\mathsf{N}}(x) \to \forall y.E_{\mathsf{N}}(y) \to \exists z.E_{\mathsf{N}}(z) \wedge \mathsf{ADD}(x, y, z)$$

We put $\varphi(x) = \forall y.E_{\mathsf{N}}(y) \to \exists z.E_{\mathsf{N}}(z) \wedge \mathsf{ADD}(x, y, z)$ and establish $\forall x.E_{\mathsf{N}}(x) \to \varphi(x)$ using Ind-N. The base case $\varphi(0)$ follows from E-Zero and a computationally irrelevant axiom.

The step case would follow from

$$\forall y.\forall z.E_{\mathsf{N}}(y) \to E_{\mathsf{N}}(z) \to \diamond \to \mathsf{ADD}(x, y, z) \to \exists z'.E_{\mathsf{N}}(z') \wedge \mathrm{ADD}(\mathsf{S}(x), y, z)$$

We instantiate z' with $\mathsf{S}(z')$ and use E-Suc plus computationally irrelevant axioms.

Notice that we cannot deduce $\forall x.E_{\mathsf{N}}(x) \to \exists z.E_{\mathsf{N}}(z) \wedge \mathrm{ADD}(x, x, z)$ because we would lack one E_{N} assumption.

For a similar counterexample suppose that $\mathsf{DUP}(x, y)$ states that $x, y \in [\![\mathsf{N}]\!]$ and $y = 2x$. Then an attempt to prove $\forall x.E_{\mathsf{N}}(x) \to \exists y.E_{\mathsf{N}}(y) \wedge \mathsf{DUP}(x, y)$ will fail because we lack an additional $\diamond$-resource in the inductive case.

Now suppose that we are given a relation symbol = interpreted as equality.

We can define falsity $\bot$ by $\bot =_{\text{def}} \mathsf{tt}{=}\mathsf{ff}$ and negation by $\neg\varphi =_{\text{def}} \varphi{\to}\bot$. We can prove $\bot{\to}\varphi$ for arbitrary computationally irrelevant φ. We cannot prove $\bot{\to}\Diamond$ for instance.

By induction we can prove

$$\forall x.E_{\mathsf{N}}(x){\to}(x{=}0 \vee \neg x{=}0)) \otimes E_{\mathsf{N}}(x)$$

The base case uses E-Zero, Or-I-1, and c.i. axioms. The inductive case would follow from

$$\forall x.\Diamond{\to}E_{\mathsf{N}}(x){\to}E_{\mathsf{N}}(\mathsf{S}(x)) \otimes (\neg(\mathsf{S}(x){=}0))$$

which is direct from E-Suc.

Notice that since disjunction is computationally relevant, we cannot establish

$$\forall x, y.x{=}0 \vee \neg(x{=}0)$$

i.e. decidability of equality without existence assumption. Notice that the more natural statement

$$\forall x.E_{\mathsf{N}}(x){\to}x{=}0 \vee \neg x{=}0$$

follows from the one we've shown, but is strictly weaker since it doesn't allow us to "recycle" the information that x "exists".

We can establish

$$\forall x.E_{\mathsf{N}}(x){\to}\forall y.E_{\mathsf{N}}(y){\to}(x{=}y \vee \neg(x{=}y)) \otimes E_{\mathsf{N}}(x) \otimes E_{\mathsf{N}}(y)$$

and the same with = replaced with $\leq$.

Now assume that we are given a binary relation $\leq$ such that for some sort A we can prove

$$\forall x.E_A(x){\to}\forall y.E_A(y){\to}(x{\leq}y \vee \neg(x{\leq}y)) \otimes E_A(x) \otimes E_A(y)$$

Furthermore, we assume a relation INSERT(a, l, l') stating that $a \in [\![A]\!]$ and $l, l' \in [\![\mathsf{L}(l)]\!]$ and whenever l is sorted w.r.t. $\leq$ so is l' and the members of l' are those of l together with a.

By mimicking the algorithm in the M.H. LICS paper we can prove

$$\forall l.E_{\mathsf{L}(A)}(l){\to}\Diamond{\to}\forall a.E_A(a){\to}\exists l'.E_{\mathsf{L}(A)}(l') \wedge \mathsf{INSERT}(a, l, l')$$

¿From this, we can prove that every list can be sorted.

Of course, one should not think of proofs in the logic as proofs of theorems per se; everybody knows that a list can be sorted! Rather such proof should be seen as the development of a program together with its correctness proof. The linearity restrictions ensure that such program will be executable in polynomial time.

To formalise this is the purpose of the rest of this note.

6 Realisation

We extend the calculus $\mathcal{C}$ by coproduct types $A + B$, unary natural numbers N, and unit type $\mathbf{1}$ together with the standard term formers We put $[\![A + B]\!] = [\![A]\!] + [\![B]\!], [\![\mathsf{N}]\!] = \mathbb{N}, [\![\mathbf{1}]\!] = \{0\}$. The realisation relation is extended by

$$\begin{array}{l}
e \Vdash_A x \Rightarrow TKe \Vdash_{A+B} \mathsf{inl}(x) \\
e \Vdash_B x \Rightarrow T\bar{K}e \Vdash_{A+B} \mathsf{inr}(x) \\
TKK \Vdash_{\mathsf{N}} 0 \\
e \Vdash_{\mathsf{N}} n \Rightarrow T\bar{K}e \Vdash_{\mathsf{N}} n + 1 \\
e \Vdash_{mathbf1}
\end{array}$$

To each formula φ a type $\tau(\varphi)$ is associated by the following clauses.

$$
\begin{aligned}
&\tau(\diamond) = \diamond\\
&\tau(E_A(t)) = A\\
&\tau(P(\vec{t})) = \mathbf{1} \text{ if } P \text{ is a c.i. predicate symbol}\\
&\tau(\varphi \rightarrow \psi) = \tau(\varphi) \multimap \tau(psi)\\
&\tau(\varphi \otimes \psi) = \tau(\varphi) \otimes \tau(\psi)\\
&\tau(\varphi \wedge \psi) = \tau(\varphi) \times \tau(\psi)\\
&\tau(\varphi \vee \psi) = \tau(\varphi) + \tau(\psi)\\
&\tau(\forall x.\varphi) = \tau(\varphi)\\
&\tau(\exists x.\varphi) = \tau(\varphi)
\end{aligned}
$$

Notice that when φ is computationally irrelevant then $[\![\tau(\varphi)]\!]$ is a singleton set and we can find a closed $\mathcal{C}$-term $e : \tau(\varphi)$.

Recall that we have identified the set-theoretic interpretation of sort A viewed as a $\mathcal{C}$-type with a subset of the domain of individuals D.

Let φ be a formula, η a valuation of the free variables in φ and $x \in [\![\tau(\varphi)]\!]$ an element of the *set-theoretic* semantics of $\tau(\varphi)$. If t is a term of the logic we write $\eta(t)$ for the usual first-order interpretation of t under valuation η. A (meta-level) proposition $x \ \mathbf{mr}_\eta \ \varphi$ is defined inductively by the following clauses.

$$
\begin{array}{lll}
0 \ \mathbf{mr}_\eta \ P(\vec{t}) \iff P(\vec{t}) \text{ is a true c.i. atomic formula under } \eta . 0 \ \mathbf{mr}_\eta \ \diamond & & \\
d \ \mathbf{mr}_\eta \ E_A(t) & \iff & d = \eta(t) \text{ and } d \in [\![A]\!]\\
(x, y) \ \mathbf{mr}_\eta \ \varphi \otimes \psi & \iff & x \ \mathbf{mr}_\eta \ \varphi \text{ and } y \ \mathbf{mr}_\eta \ \psi\\
(x, y) \ \mathbf{mr}_\eta \ \varphi \wedge \psi & \iff & x \ \mathbf{mr}_\eta \ \varphi \text{ and } y \ \mathbf{mr}_\eta \ \psi\\
\mathsf{inl}(x) \ \mathbf{mr}_\eta \ \varphi \vee \psi & \iff & x \ \mathbf{mr}_\eta \ \varphi\\
\mathsf{inr}(x) \ \mathbf{mr}_\eta \ \psi \vee \psi & \iff & x \ \mathbf{mr}_\eta \ \varphi\\
f \ \mathbf{mr}_\eta \ \varphi \rightarrow \psi & \iff & x \ \mathbf{mr}_\eta \ \varphi \text{ implies } f(x) \ \mathbf{mr}_\eta \ \psi\\
y \ \mathbf{mr}_\eta \ \forall x.\varphi & \iff & y \ \mathbf{mr}_{\eta[x \mapsto d]} \text{ for all } d\\
y \ \mathbf{mr}_\eta \ \exists x.\varphi & \iff & y \ \mathbf{mr}_{\eta[x \mapsto d]} \text{ for some } d
\end{array}
$$

Notice that at this stage realisability doesn't say very much: A formula φ, computationally relevant or not, is semantically true if and only if there exists $x \in [\![\tau(\varphi)]\!]$ with $x \ \mathbf{mr} \ \varphi$.

The content of the following theorem is that such x can be denoted by a $\mathcal{C}$-term provided φ is provable in our logic.

Theorem 6.1 *Whenever $\varphi_1, \ldots, \varphi_n \Longrightarrow \psi$ is derivable then there exists a $\mathcal{C}$-term*

$$h_1 \colon \tau(\varphi_1), \ldots, h_n \colon \tau(\varphi_n) \vdash e : \tau(\psi)$$

such that for each valuation η and elements $v_i \in [\![\tau(\varphi_i)]\!]$ with $v_i \ \mathbf{mr}_\eta \ \varphi_i$ it holds that

$$[\![e]\!](v_1, \ldots, v_n) \ \mathbf{mr}_\eta \ \varphi$$

Proof. Induction on derivations. A computationally irrelevant axiom φ is realised by a closed term in $\tau(\varphi)$. The axioms asserting existence of constructors are realised by the corresponding constructor terms, e.g. E-Succ is realised by $\lambda c \colon \diamond . \lambda y \colon \mathsf{N} . \mathsf{S}(c, y)$.

The axioms for the connectives are realised by the appropriate term formers,

The induction rules are realised by appropriate instances of the iterators. $\square$

Corollary 6.2 *Let $A_1, \ldots, A_n, B$ be sorts, $f : [\![A_1]\!] \times [\![A_n]\!] \to [\![B]\!]$ a (set-theoretic) function, and F a predicate symbol interpreted as its graph. If*

$$\forall \vec{x}. E_{\vec{A}}(\vec{x}) \to \exists y. E_B(y) \wedge F(\vec{x}, y)$$

is provable in the logic then f is the denotation of a $\mathcal{C}$-term.

7 Discussion

The fact that all true c.i. formulas are axioms is certainly debatable. It makes the logic a semi-formal system since provability isn't r.e. anymore. However, as emphasised in the examples section we view the logic as a formalism for developing certified resource bounded programs. We think that as far as the mere verification is concerned one should have access to as powerful methods as desired.

The realisation of the logic involving the set-theoretic semantics is somewhat non-standard. We have chosen this avenue so as to avoid having to formalise the **mr** relation in a formal system. Doing so would force us to leave the realm of first-order logic. Not impossible but somewhat messier.