The Provably Total Functions of Second-Order Arithmetic

Lecture Notes

Colin Riba
ENS de Lyon, Université de Lyon, LIP*
colin.riba@ens-lyon.fr
http://perso.ens-lyon.fr/colin.riba/

March 16, 2021

The goal of these notes is the show a result, due to J.-Y. Girard, stating that all provably total functions of second-order Peano’s arithmetic are representable in System $\mathcal{F}$.

Similarly to the case of first-order arithmetic and System $\mathcal{T}$, the result is based on a general $\Pi^0_2$-conservativity result of classical arithmetic over intuitionistic arithmetic.

An important reference on Second Order Arithmetic and System $\mathcal{F}$ is [GLT89]. Our approach combines the developments presented in [SU06] and [Kri90].

We refer to [Sim10] for details on the (important) amount of mathematics which can be formalized in Second-Order Arithmetic.

1 Preliminaries

1.1 Second-Order Logic

We recall here the definition of minimal second-order predicate logic. We assume given a first-order language $(\mathcal{V}, \Sigma)$ consisting of an infinite set $\mathcal{V} = \{x, y, \ldots \}$ of individual variables and of a signature $\Sigma$ of function symbols on individuals, each coming with an arity $n \in \mathbb{N}$. We write $\text{Ter}(\mathcal{V}, \Sigma)$ for the set of first-order (individual) terms built with symbols from $\Sigma$ and variables from $\mathcal{V}$. We moreover assume given, for each $n \in \mathbb{N}$, an infinite set $\mathcal{X}_n = \{X^n, Y^n, Z^n, \ldots \}$ of predicate (or second-order) variables of arity $n$. The formulae are given by:

$$A, B ::= X^n(a_1, \ldots, a_n) \mid A \Rightarrow B \mid (\forall x) A \mid (\forall X^n) A$$

where $X^n \in \mathcal{X}_n$ and $a_1, \ldots, a_n \in \text{Ter}(\mathcal{V}, \Sigma)$. We often write $X$ for $X^n$.

Intuitionistic natural deduction for second-order logic (NJ2) is given by the rules of Figure 1.
\[ \begin{align*}
A \in \Delta & \quad \Delta, A \vdash B & \quad \Delta \vdash A \Rightarrow B \\
\Delta \vdash A & \quad \Delta \vdash (\forall x)A & \quad \Delta \vdash \neg \neg A \\
\Delta \vdash (\forall X^n)A & \quad \Delta \vdash (\forall X^n)A[\alpha/x] & \quad \Delta \vdash \neg \neg A \\
\end{align*} \]

Figure 1: Deduction Rules for NJ_2.

**Notation 1.1.** We assume the usual impredicative codings of the connectives \( \bot, \top, \land, \lor, (\exists x), (\exists X^n) \) and of equality (\( = \)). As usual we let \( \neg A := A \Rightarrow \bot \) and \( A \leftrightarrow B := (A \Rightarrow B) \land (B \Rightarrow A) \).

*Proof.* Exercise!

**Lemma 1.2.** In NJ_2, the following rules are equivalent:

\[ \begin{align*}
\Delta \vdash ((A \Rightarrow B) \Rightarrow A) \Rightarrow A & & \Delta \vdash \neg \neg A \Rightarrow A & & \Delta \vdash A \lor \neg A \\
\end{align*} \]

Classical natural deduction for second-order logic (NK_2) is given by the rules of NJ_2 augmented with Peirce’s law:

\[ \Delta \vdash ((A \Rightarrow B) \Rightarrow A) \Rightarrow A \]

### 1.2 Second-Order Peano’s Arithmetic

Similarly as for first-order arithmetic, we shall consider systems with a function symbol for each primitive recursive function. We consider here an axiomatization analogous to that of PA^* and HA^*.\(^1\) We thus consider the signature \( \Sigma_{PR} \) which consists of one function symbol \( f \) of arity \( n \) for each primitive recursive function \( f : \mathbb{N}^k \rightarrow \mathbb{N} \). In the following, we shall write \( 0 \) for the constant primitive recursive function \( 0 \in \mathbb{N} \) and \( S \) for the primitive recursive successor function \( n \in \mathbb{N} \mapsto n + 1 \in \mathbb{N} \).

Each individual \( a \) with free variables among \( x_1, \ldots, x_k \) can be interpreted as a (primitive recursive) function \( \mathbb{N}^k \rightarrow \mathbb{N} \). In particular, each closed individual \( a \) can be interpreted as a natural number \( \llbracket a \rrbracket \in \mathbb{N} \).

**Notation 1.3.** For \( n \in \mathbb{N} \), we \( \overline{n} := \mathbb{S}^n0 \) (so that \( \llbracket n \rrbracket = n \)).

The axioms of PA_2 and HA_2 are given in Figure 2, where

\[ \text{Nat}(x) := (\forall X)(X(0) \Rightarrow (\forall y)(X(y) \Rightarrow X(Sy)) \Rightarrow X(x)) \]

PA_2 corresponds to classical deduction (hence in NK_2) with these axioms while HA_2 corresponds intuitionistic deduction (in NJ_2).

\(^1\)We refer to [http://perso.ens-lyon.fr/colin.riba/teaching/pp/notes/arith.pdf](http://perso.ens-lyon.fr/colin.riba/teaching/pp/notes/arith.pdf) for the definition of HA^* and PA^*
Injectivity of the Successor:

\[ \Delta \vdash (\forall xy)(Sx = Sy \Rightarrow x = y) \]

Non-Confusion:

\[ \Delta \vdash (\forall x)(\neg (Sx = 0)) \]

Equations on Primitive Recursive Functions:

\[ \Delta \vdash (\forall x_1 \cdots (\forall x_k)(a = b)) \] (\(a, b\) with free variables among \(x_1, \ldots, x_k\))

Induction:

\[ \Delta \vdash (\forall x)\text{Nat}(x) \]

Figure 2: Axioms of PA2 and HA2.

2 The Systems PA2\(^-\) and HA2\(^-\)

In order to perform extraction in System \(\mathcal{F}\) following the Curry-Howard interpretation of second-order logic, the main difficulty is to find terms whose type corresponds to the axioms of HA2 and PA2. Injectivity of the successor poses no problem, while induction will be dealt-with in Sect. 4. For non-confusion, we follow a trick due to D. Leivant.

The idea is to weaken the axiom of non-confusion (actually the implication \((\neg) \Rightarrow \bot\)) in a way which preserves the provable \(\Pi^0_2\)-formulae. This is performed using auxiliary systems denoted PA2\(^-\) and HA2\(^-\). The axioms and deduction rules of these systems are the same as those for PA2 and HA2 respectively, but with Non-Confusion replaced by:

\[ \Delta \vdash (\forall xyz)(Sx = 0 \Rightarrow y = z) \]

Lemma 2.1. We have HA2 \(\vdash (\forall xyz)(Sx = 0 \Rightarrow y = z)\).

Lemma 2.2. We have PA2\(^-\) \(\vdash (\forall x)(\neg (Sx = 0)) \lor (\forall xy)(x = y)\).

Proof. On the one hand, in HA2\(^-\) we have \((\exists x)(Sx = 0) \vdash (\forall xy)(x = y)\). On the other hand, in NK2 we have \((\forall x)(\neg (Sx = 0)) \vdash (\exists x)(Sx = 0)\). It follows that in PA2\(^-\) we get \(\neg (\forall x)(\neg (Sx = 0)) \vdash (\forall xy)(x = y)\). As a consequence, PA2\(^-\) proves

\[ \neg ((\forall x)(\neg (Sx = 0)) \lor (\forall xy)(x = y)) \vdash (\forall xy)(x = y) \]

and thus PA2\(^-\) \(\vdash (\forall x)(\neg (Sx = 0)) \lor (\forall xy)(x = y)\). \(\square\)

We then get the promised fact that PA2\(^-\) proves the same \(\Pi^0_2\)-formulae as PA2.

Proposition 2.3. If PA2 \(\vdash (\forall x)(\exists y)(a = b)\) then PA2\(^-\) \(\vdash (\forall x)(\exists y)(a = b)\).

Proof. Assume that PA2 proves \((\forall x)(\exists y)(a = b)\). It follows that PA2\(^-\) proves \(\neg (Sx = 0) \vdash (\forall x)(\exists y)(a = b)\). Moreover, in NJ2, we have \((\forall xy)(x = y) \vdash (\forall x)(\exists y)(a = b)\). It then follows from Lem. 2.2 that PA2\(^-\) proves \((\forall x)(\exists y)(a = b)\). \(\square\)
Proposition 2.3 gives the following relative consistency result for \( \text{PA}^2 \) w.r.t. \( \text{PA}^2 \).

**Corollary 2.4.** If \( \text{PA}^2 \vdash \bot \) then \( \text{PA}^2 \vdash (\forall x)(x \equiv y) \).

**Proof.** If \( \text{PA}^2 \vdash \bot \) then \( \text{PA}^2 \vdash (\exists y)(S^y \equiv 0) \). By Prop. 2.3 we get \( \text{PA}^2 \vdash (\exists y)(S^y \equiv 0) \), and thus \( \text{PA}^2 \vdash (\forall xz)(x \equiv z) \). \( \square \)

### 3 Friedman’s Translation

Friedman’s translation for second-order arithmetic is as follows. Let \( R \) be any formula. Define, for \( A \) a formula:

\[ \neg_R A := A \Rightarrow R \]

Now, define \( A^{-\overline{R}} \) by induction on \( A \) as follows:

\[
\begin{align*}
(X(a))^{-\overline{R}} &:= \neg_R X(a) \\
(\forall x)A^{-\overline{R}} &:= (\forall x)A^{-\overline{R}} \\
((\forall^n X)A)^{-\overline{R}} &:= (\forall^n X)A^{-\overline{R}}
\end{align*}
\]

**Lemma 3.1.** For all formula \( A \), \( \text{NJ}^2 \vdash \neg_R \neg_A \Rightarrow A^{-\overline{R}} \).

**Proof.** Exercise! \( \square \)

**Lemma 3.2.** For all formulae \( A, B \), if \( X \notin \text{FV}(R) \) then \( \text{NJ}^2 \vdash (A[B[x]/X])^{-\overline{R}} \Leftrightarrow A^{-\overline{R}}[B^{-\overline{R}}[x]/X] \).

**Proof.** Exercise! \( \square \)

**Lemma 3.3.** For all formula \( A \), \( \text{NJ}^2 \vdash R \Rightarrow A^{-\overline{R}} \).

**Proof.** Exercise! \( \square \)

**Lemma 3.4.** We have \( \text{NJ}^2 \vdash (((A \Rightarrow B) \Rightarrow A) \Rightarrow A)^{-\overline{R}} \).

**Proof.** Exercise! \( \square \)

We now have everything at hand to prove the correctness of Friedman’s translation for \( \text{NK}^2 \).

**Proposition 3.5.** If \( \text{NK}^2 \vdash A \) then \( \text{HA}^2 \vdash A^{-\overline{R}} \).

**Proof.** Exercise! \( \square \)

We now turn to the correctness of Friedman’s translation for \( \text{PA}^2 \). We rely on the following to handle the axioms of \( \text{PA}^2 \) involving equality.

**Lemma 3.6.** In \( \text{NJ}^2 \), we have

\[
\begin{align*}
(1) \ (a \equiv b)^{-\overline{R}} \Rightarrow \neg_{\overline{R}} \neg(a \equiv b) \\
(2) \ \neg_{\overline{R}} \neg(a \equiv b) \Rightarrow (a \equiv b)^{-\overline{R}}
\end{align*}
\]

**Proof.** Exercise! \( \square \)

**Theorem 3.7.** If \( \text{PA}^2 \vdash A \) then \( \text{HA}^2 \vdash A^{-\overline{R}} \).

**Proof.** Exercise! \( \square \)

**Theorem 3.8.** If \( \text{PA}^2 \vdash (\forall x)(\exists y)(a \equiv b) \) then \( \text{HA}^2 \vdash (\forall x)(\exists y)(a \equiv b) \).

\[ \]
\[
\Delta \vdash (\forall xy)(Sx \doteq Sy \Rightarrow x \doteq y) \quad \Delta \vdash (\forall x y z)(Sx \doteq 0 \Rightarrow y \doteq z)
\]

\[
\vdash [a] = [b] \quad (a, b \text{ with free variables among } x_1, \ldots, x_k)
\]

\[
\Delta \vdash (\forall x_1) \cdots (\forall x_k)(a \doteq b) \quad (f \in \Sigma_{PR} \text{ of arity } k)
\]

\[
\Delta \vdash (\forall x_1) \cdots (\forall x_k)(\text{Nat}(x_1) \Rightarrow \cdots \Rightarrow \text{Nat}(x_k) \Rightarrow \text{Nat}(f(x_1, \ldots, x_k)))
\]

**Figure 3:** Axioms of PA\(^2\)\({}_0\) and HA\(2\)\(_0\).

**Proof.** Exercise! □

**Corollary 3.9.** If PA\(2\) \(\vdash (\forall x)(\exists y)(a \doteq b)\) then HA\(2\)\(_-\) \(\vdash (\forall x)(\exists y)(a \doteq b)\).

**Proof.** By Thm. 3.8 and Prop. 2.3. □

**Corollary 3.10.** If PA\(2\) \(\vdash \perp\) then HA\(2\)\(_-\) \(\vdash (\forall xy)(x \doteq y)\).

**Proof.** Reasoning similarly as in Cor. 2.4 we get that if PA\(2\) \(\vdash \perp\) then PA\(2\) \(\vdash (\exists x)(Sx \doteq 0)\). Hence HA\(2\)\(_-\) \(\vdash (\exists x)(Sx \doteq 0)\) by Thm. 3.8 and we get HA\(2\)\(_-\) \(\vdash (\forall xy)(x \doteq y)\). □

### 4 Elimination of Induction

We now discuss the second-order induction axiom \((\forall x)\text{Nat}(x)\). This axiom is not realizable by a term of System \(\mathcal{F}\). In order to handle it, the usual trick is to relativize all individual quantifications to \text{Nat}(-), and then to work in systems without induction.

We thus consider the systems PA\(2\)\(_-\) and HA\(2\)\(_0\), which are obtained from respectively PA\(2\)\(_-\) and HA\(2\)\(_0\) by removing the induction axiom. But in the absence of induction, relativizing all individual quantifications to \text{Nat}(-) requires to be able the prove \text{Nat}(a) for each term \(a\) (assuming \text{Nat} on the free variables of \(a\)). The axioms of PA\(2\)\(_-\) and HA\(2\)\(_0\) are given in Figure 3.

**Lemma 4.1.** Consider an individual term \(a \in \text{Ter}(\mathcal{V}, \Sigma_{PR})\) with free variables among \(x_1, \ldots, x_k\). In HA\(2\)\(_0\) we have \(\text{Nat}(x_1), \ldots, \text{Nat}(x_k) \vdash \text{Nat}(a)\).

**Proof.** By induction on \(a\). □

The **relativization** of a formula \(A\) to the predicate \text{Nat}(-) is the formula \(A^{\text{Nat}}\) defined by induction on \(A\) as follows:

\[
(X(a))^{\text{Nat}} := X(a) \\
((\forall x)A)^{\text{Nat}} := (\forall x)(\text{Nat}(x) \Rightarrow A^{\text{Nat}}) \\
(A \Rightarrow B)^{\text{Nat}} := A^{\text{Nat}} \Rightarrow B^{\text{Nat}} \\
((\forall X^n)A)^{\text{Nat}} := (\forall X^n)A^{\text{Nat}}
\]

**Lemma 4.2.** We have HA\(2\)\(_0\) \(\vdash ((\forall x)\text{Nat}(x))^{\text{Nat}}\).

**Proof.** We are done if we show that

\[
\text{HA}\(2\)\(_0\) \vdash \text{Nat}(x) \Rightarrow X(0) \Rightarrow (\forall y)(\text{Nat}(y) \Rightarrow X(y) \Rightarrow X(Sy)) \Rightarrow X(x)
\]

But in HA\(2\)\(_0\), \text{Nat}(x) entails

\[
(\text{Nat}(0) \land X(0)) \Rightarrow (\forall y)((\text{Nat}(y) \land X(y)) \Rightarrow (\text{Nat}(Sy) \land X(Sy))) \Rightarrow (\text{Nat}(x) \land X(x))
\]
and we are done since HA2_0 ⊢ Nat(0) and, in HA2_0,

$$\text{Nat}(y) \Rightarrow X(y) \Rightarrow X(Sy) \vdash (\text{Nat}(y) \land X(y)) \Rightarrow (\text{Nat}(Sy) \land X(Sy))$$

(using that HA2_0 ⊢ Nat(x) ⇒ Nat(Sx)).

**Proposition 4.3.** For a closed formula A, if HA2_0 ⊢ A, then HA2_0 ⊢ A_Nat.

**Proof.** Let A_1, ..., A_n and A with free individual variables among \(\vec{x} = x_1, ..., x_k\). We show that if HA2_0 proves

$$A_1, ..., A_n \vdash A$$

then HA2_0 proves

$$\text{Nat}(x_1), ..., \text{Nat}(x_k), A_1^{\text{Nat}}, ..., A_n^{\text{Nat}} \vdash A^{\text{Nat}}$$

The proof is by induction on derivations.

**Axioms of HA2_0.** The cases of

$$\Delta \vdash (\forall xy)(Sx \equiv Sy \Rightarrow x \equiv y)$$

$$\Delta \vdash (\forall x_1) \cdots (\forall x_k)(a \equiv b)$$

$$\Delta \vdash (\forall x)(Sx \equiv 0 \Rightarrow y \equiv z)$$

are trivial. Induction is handled by Lem. 4.2.

**Cases of**

$$\Delta, A \vdash B \quad \Delta \vdash A \Rightarrow B \quad \Delta \vdash A \Rightarrow B$$

$$\Delta \vdash (\forall X^m)A \quad (X^m \notin \text{FV}(\Delta)) \quad \Delta \vdash A \Rightarrow B$$

$$\Delta \vdash A \Rightarrow B \quad \Delta \vdash (\forall X^m)A \quad \Delta \vdash [B[x_1, ..., x_m] / X]$$

Trivial since \((\_)^{\text{Nat}}\) commutes over \(\Rightarrow\) and \((\forall X^m)\).

**Case of**

$$A_1, ..., A_n \vdash A \quad \Delta \vdash (\forall x)A \quad (x \notin \text{FV}(A_1, ..., A_n))$$

By induction hypothesis HA2_0 proves

$$\text{Nat}(x_1), ..., \text{Nat}(x_k), A_1^{\text{Nat}}, ..., A_n^{\text{Nat}} \vdash A^{\text{Nat}}$$

and thus

$$\text{Nat}(x_1), ..., \text{Nat}(x_k), A_1^{\text{Nat}}, ..., A_n^{\text{Nat}} \vdash \text{Nat}(x) \Rightarrow A^{\text{Nat}}$$

and we are done since x is not free in A_1, ..., A_n.

**Case of**

$$A_1, ..., A_n \vdash (\forall x)A \quad A_1, ..., A_n \vdash A[a/x]$$

Let \(\vec{x} = x_1, ..., x_k\) such that A_1, ..., A_n, \((\forall x)A\) and a have free variables among \(\vec{x}\). By induction hypothesis, HA2_0 proves

$$\text{Nat}(x_1), ..., \text{Nat}(x_k), A_1^{\text{Nat}}, ..., A_n^{\text{Nat}} \vdash (\forall x)(\text{Nat}(x) \Rightarrow A^{\text{Nat}})$$
\[
\begin{align*}
(x : \tau) &\in \Gamma \\
\Gamma \vdash x : \tau &
\quad 
\Gamma, x : \sigma \vdash t : \tau \\
\Gamma \vdash \lambda x. t : \sigma \to \tau &
\quad 
\Gamma \vdash t : \sigma \to \tau \\
\Gamma \vdash u : \sigma &
\quad 
\Gamma \vdash tu : \tau
\end{align*}
\]
\[
\begin{align*}
\Gamma \vdash t : \tau &
\quad 
\Gamma \vdash t : (\forall X) \tau \\
(\forall X \notin \text{FV}(\Gamma)) &
\quad 
\Gamma \vdash t : (\forall X) \tau [\sigma / X]
\end{align*}
\]

Figure 4: Typing Rules for Curry-style System \( F \).

On the other hand, by Lem. 4.1, HA2\(_0\) proves
\[
\text{Nat}(x_1), \ldots, \text{Nat}(x_k) \vdash \text{Nat}(a)
\]
and the result follows.

5 Representation in System \( F \)

5.1 Curry-style System \( F \)

The terms of Curry-style System \( F \) are pure \( \lambda \)-terms:
\[
t, u \in \Lambda := x \mid \lambda x. u \mid tu
\]
where \( x \) belongs to some countably infinite set of \( \lambda \)-variables. As usual, we identify terms modulo renaming of their bound variables (\( \alpha \)-conversion).

The relation of \( \beta \)-reduction \( \triangleright_{\beta} \) is the least relation on \( \lambda \)-terms such that:
\[
(\lambda x.t)u \triangleright_{\beta} t[u/x] \\
\lambda x. t \triangleright_{\beta} \lambda x. u \\
tv \triangleright_{\beta} uv \\
vt \triangleright_{\beta} vu
\]

The relation \( =_{\beta} \) of \( \beta \)-conversion is the least equivalence relation containing \( \triangleright_{\beta} \).

The types of System \( F \) are:
\[
\tau, \sigma := X \mid \sigma \to \tau \mid (\forall X) \tau
\]
where \( X \) belongs to a countably infinite set of type variables. Types are also identified modulo \( \alpha \)-conversion.

The typing rules of Curry-style System \( F \) are given in Figure 4.

5.2 Realizability

We will now prove that \( \Pi_2^0 \)-formulae provable in PA2 are witnessed by system \( F \) terms of the appropriate type.

We will do a proof by realizability, whose aim will be to relate proofs to typed terms. Thanks to the results of the previous questions, it is sufficient to consider proofs in HA2\(_0\). We use an erasing map \(( - )^0\) from the formulae of second-order arithmetic to the types of System \( F \). We assume given a bijection (also written \(( - )^0\)) between the second-order variables of arithmetic and the types variables of System \( F \). Given a formula \( A \), define the type \( A^0 \) by induction on \( A \) as follows:
\[
(X(\overline{a}))^0 := X^0 \\
(A \Rightarrow B)^0 := A^0 \to B^0 \\
(\forall x.A)^0 := A^0 \\
((\forall X).A)^0 := (\forall X^0).A^0
\]
Second-order variables $X^k \in \mathcal{X}_k$ will be interpreted as maps $S: \mathbb{N}^k \rightarrow \mathcal{P}(\Lambda)$ which are saturated in the sense that for all $n_1, \ldots, n_k \in \mathbb{N}$, if $u \in S(n_1, \ldots, n_k)$ and $t = \beta u$, then $t \in S(n_1, \ldots, n_k)$. Note that for any $k \in \mathbb{N}$, the constant null map $(n_1, \ldots, n_k) \mapsto \emptyset$ is saturated.

It is convenient to consider formulae with parameters among saturated maps (it would have been equivalent to have pairs $(A, \rho)$ where $\rho$ maps second-order variables $X^m \in \mathcal{X}_n$ to saturated $S: \mathbb{N}^n \rightarrow \mathcal{P}(\Lambda)$).

We shall now proceed to the definition of the realizability relation $t \VDash A$, where $A$ is a closed formula, possibly with parameters among saturated maps, and $t$ is an untyped $\lambda$-term. The relation $t \Vdash A$ is defined by induction on $A$ as follows:

- $t \Vdash S(a_1, \ldots, a_n)$ iff $t \in S([a_1], \ldots, [a_n])$
- $t \Vdash A \Rightarrow B$ iff for all $u \in \Lambda$, if $u \Vdash A$ then $tu \Vdash B$
- $t \Vdash (\forall x)A$ iff for all $n \in \mathbb{N}$, $t \Vdash A[n/x]$
- $t \Vdash (\forall X^k)A$ iff for all saturated $S: \mathbb{N}^k \rightarrow \mathcal{P}(\Lambda)$, $t \Vdash A[S/X^k]$

A formula $A$ is realized (notation $\vdash A$) if $t \Vdash A$ for some $t$.

The following are easy standard facts on the realizability relation.

**Lemma 5.1.** If $u \Vdash A$ and $t = \beta u$ then $u \Vdash A$.

**Lemma 5.2.** Let $a$ and $b$ be closed individual terms such that $J_a = J_b$. Then $t \Vdash A[a/x]$ if and only if $t \Vdash A[b/x]$.

**Lemma 5.3.** We have $t \Vdash A[B[x_1, \ldots, x_k]/X^k]$ if and only if $t \Vdash A[S/X^k]$ where $S(n_1, \ldots, n_k)$ is the set of $\lambda$-terms $u$ such that $u \Vdash B[n_1/x_1, \ldots, n_k/x_k]$.

### 5.3 Adequacy

We now turn to adequacy of realizability. We begin with NJ2.

**Proposition 5.4.** If $\text{NJ}_2 \vdash A$ for a closed formula $A$, then there is a closed $\lambda$-term $t$ of type $A^\circ$ such that $t \Vdash A$.

It is also easy to see that realizability handles well the impredicative encoding of equality.

**Lemma 5.5.** Assume that $(a \doteq b)$ is a closed formula.

1. If $[a] = [b]$, then $\lambda x.x \Vdash (a \doteq b)$.
2. If $[a] \neq [b]$, then no $\lambda$-term realizes $(a \doteq b)$.

**Proof.** Exercise! \qed

Adequacy for $\text{HA}_2^-$ requires some analysis of the realizers of $\text{Nat}$. Note that we have

$$\text{Nat}^\circ = (\forall X)(X \Rightarrow (X \Rightarrow X) \Rightarrow X)$$

Given $n \in \mathbb{N}$, the $n$-th Church’s numeral is the term $c_n := \lambda x.\lambda f.f^n x$.

Note that $\vdash c_n : \text{Nat}^\circ$.

**Lemma 5.6.** We have $c_n \Vdash \text{Nat}(n)$. 

8
Lemma 5.7. If $t \vdash \text{Nat}(n)$ then $\lambda xy. txy \triangleright^{*} c_n$ (for $x, y$ not free in $t$).

Proof. Exercise! □

Lemma 5.8. Let $f \in \Sigma_{PR}$ be a function symbol of arity $k$. The following is realized by a closed term of System $F$:

$$(\forall x_1, \ldots, x_k)(\text{Nat}(x_1) \Rightarrow \cdots \Rightarrow \text{Nat}(x_k) \Rightarrow \text{Nat}(f(x_1, \ldots, x_k)))$$

Proof. Exercise! □

Lemma 5.9. Consider an individual term $a \in \text{Ter}(V, \Sigma_{PR})$ with free variables among $x_1, \ldots, x_n$. The following is realizable by a closed term of System $F$:

$$(\forall x_1, \ldots, x_n)(\text{Nat}(x_1) \Rightarrow \cdots \Rightarrow \text{Nat}(x_n) \Rightarrow \text{Nat}(a))$$

Proof. Exercise! □

Theorem 5.10. If $HA_2^0 \vdash A$ for a closed formula $A$, then there is a closed $\lambda$-term $t$ of type $A^\circ$ such that $t \vdash A$.

Corollary 5.11. $PA_2$ does not prove $\bot$.

Proof. Exercise! □

5.4 Extraction

We can now show that the $\Pi^0_2$-formulae of $HA_2^-$ (and thus of $PA_2$) are witnessable by terms of System $F$. In this Section, we consider a fixed closed $\Pi^0_2$-formula $(\forall x) (\exists y)(a \equiv b)$.

Proposition 5.12. Assume $HA_2^0 \vdash (\forall x) (\exists y)(a \equiv b)^{\text{Nat}}$. There is a closed $\lambda$-term $t$ of type $\text{Nat}^{\circ} \rightarrow \text{Nat}^{\circ}$ such that:

- for every $n \in \mathbb{N}$, there is an $m \in \mathbb{N}$ such that $tc_n \triangleright^{*} c_m$ and $\llbracket a[n/x, m/y] \rrbracket = \llbracket b[n/x, m/y] \rrbracket$.

Proof. Exercise! □

Corollary 5.13. If $PA_2$ proves $(\forall x)(\exists y)(a \equiv b)$, then there is a closed $\lambda$-term $t$ of type $\text{Nat}^{\circ} \rightarrow \text{Nat}^{\circ}$ such that:

- for every $n \in \mathbb{N}$, there is an $m \in \mathbb{N}$ such that $tc_n \triangleright^{*} c_m$ and $\llbracket a[n/x, m/y] \rrbracket = \llbracket b[n/x, m/y] \rrbracket$.

5.5 Concluding Remarks

Corollary 5.13 of course extends to formulae of the form $[(\forall y_1) \cdots (\forall y_k)(\exists z)(a \equiv b)]^{\text{Nat}}$. The method of realizability presented here can be extended in various ways. First, one can extend the programming language, so as to handle stronger axioms, for instance full classical logic and/or variants of the Axiom of Choice.

A variant of the technique presented here makes it possible to show that if $[(\forall y)(\exists z)A(y, z)]^{\text{Nat}}$ provable in second-order intuitionistic arithmetic, then there is a function $f: \mathbb{N} \rightarrow \mathbb{N}$ which is representable in System $F$ and such that $A^{\text{Nat}}(n, f(n))$ is true for all $n \in \mathbb{N}$.

Note that the false formula $\bot$ is not realized (take $S = \emptyset$). Hence, the set of closed formulae which are realized form a consistent theory, which is moreover closed under the deduction rules of intuitionistic logic. One then speaks of an intuitionistic theory.
Fact 5.14. If \( A \) is a closed formula, then either \( A \) or \( \neg A \) is realized, but not both.

Proof. Exercise! ⊓⊔

As a consequence of Fact 5.14, we have that \( A \lor \neg A \) is always realized for a closed formula \( A \). One may thus think that the intuitionistic theories induced by realizability are in fact classical. But this is not the case, because Fact 5.14 in general fails for open formulae, so that the universal closure of \( A \lor \neg A \) is in general not realized, in the sense that assuming \( \text{FV}(A) \subseteq \{x_1, \ldots, x_n\} \), we may have that

\[
\bigl[ (\forall x_1, \ldots, x_n) (A \lor \neg A) \bigr]^{\text{Nat}}
\]

has no realizer. In this case, by Fact 5.14, the following formula is realized:

\[
\bigl[ \neg (\forall x_1, \ldots, x_n) (A \lor \neg A) \bigr]^{\text{Nat}}
\]

Remark 5.15. Recall that \( \neg \neg (A \lor \neg A) \) and thus

\[
\bigl[ (\forall x_1, \ldots, x_n) \neg \neg (A \lor \neg A) \bigr]^{\text{Nat}}
\]

are intuitionistically provable. It is important note here that the Double Negation Shift

\[
((\forall x) \neg
A \Rightarrow \neg
(\forall x) A)^{\text{Nat}}
\]

(for second-order arithmetic) is in general not realized with pure \( \lambda \)-terms.

References


Contents

1 Preliminaries

1.1 Second-Order Logic ................................................................. 1
1.2 Second-Order Peano’s Arithmetic ........................................... 2

2 The Systems \( \text{PA}^2^- \) and \( \text{HA}^2^- \) ............................................. 3

3 Friedman’s Translation .............................................................. 4

4 Elimination of Induction ......................................................... 5
5 Representation in System $\mathcal{F}$

5.1 Curry-style System $\mathcal{F}$ ........................................ 7
5.2 Realizability ....................................................... 7
5.3 Adequacy ........................................................... 8
5.4 Extraction ........................................................... 9
5.5 Concluding Remarks ............................................... 9