CHAPTER 15

# Systems for arithmetic

## 1. Arithmetic in all finite types

Definition 1.1. The finite types are defined by induction as follows: 0 is a finite type, and if $\sigma$ and $\tau$ are finite types, then so are $\sigma \to \tau$ and $\sigma \times \tau$.

The system $\mathsf{HA}^\omega$ is formulated in multi-sorted intuitionistic logic, with the sorts being the finite types. There will be infinitely many variables of each sort. In addition, there will be constants:

(1) for each pair of types $\sigma, \tau$ a combinator $\mathbf{k}^{\sigma,\tau}$ of sort $\sigma \to (\tau \to \sigma)$.
(2) for each triple of types $\rho, \sigma, \tau$ a combinator $\mathbf{s}^{\rho,\sigma,\tau}$ of type $(\rho \to (\sigma \to \tau)) \to ((\rho \to \sigma) \to (\rho \to \tau))$.
(3) for each pair of types $\rho, \sigma$ combinators $\mathbf{p}^{\rho,\sigma}, \mathbf{p}_0^{\rho,\sigma}, \mathbf{p}_1^{\rho,\sigma}$ of types $\rho \to (\sigma \to \rho \times \sigma)$, $\rho \times \sigma \to \rho$ and $\rho \times \sigma \to \sigma$, respectively.
(4) a constant 0 of type 0 and a constant $S$ of type $0 \to 0$.
(5) for each type $\sigma$ a combinator $\mathbf{R}^\sigma$ ("the recursor") of type $\sigma \to ((0 \to (\sigma \to \sigma)) \to (0 \to \sigma))$.

Definition 1.2. The terms of a certain type are defined inductively as follows:

- each variable or constant of type $\sigma$ will be a term of type $\sigma$.
- if $f$ is a term of type $\sigma \to \tau$ and $x$ is a term of type $\sigma$, then $f(x)$ is a term of type $\tau$.

The convention is that an expression like $fxyz$ has to be read as $(((fx)y)z)$. It will also sometimes be written as $f(x, y, z)$.

Definition 1.3. The formulas are defined inductively as follows:

- $\perp$ is a formula and if $s$ and $t$ are terms of the same type $\sigma$, then $s =_\sigma t$ is a formula.
- if $\varphi$ and $\psi$ are formulas, then so are $\varphi \wedge \psi, \varphi \vee \psi, \varphi \to \psi$.
- if $x$ is a variable of type $\sigma$ and $\varphi$ is a formula, then $\exists x^\sigma\, \varphi$ and $\forall x^\sigma\, \varphi$ are formulas.

The axioms and rules of $\mathsf{HA}^\omega$ are:

(i) All the axioms and rules of intuitionistic logic (say in Hilbert-style).
(ii) Rules for equality at all types:

$$x =_\sigma x, \qquad x =_\sigma y \to y =_\sigma x, \qquad x =_\sigma y \wedge y =_\sigma z \to x =_\sigma z,$$

$$f =_{\sigma\to\tau} f' \wedge x =_\sigma x' \to fx =_\tau f'x'.$$

(iii) The successor axioms:

$$\neg S(x) =_0 0, \qquad S(x) =_0 S(y) \to x =_0 y$$

(iv) For any formula $\varphi$ in the language of $\mathsf{HA}^\omega$, the induction axiom:

$$\varphi(0) \to \big(\forall x^0\,(\,\varphi(x) \to \varphi(Sx)\,) \to \forall x^0\,\varphi(x)\,\big).$$

(v) The axioms for the combinators:

$$\begin{array}{rcl} \mathbf{k}xy & = & x \\ \mathbf{s}xyz & = & xz(yz) \\ \mathbf{p_0}(\mathbf{p}xy) & = & x \\ \mathbf{p_1}(\mathbf{p}xy) & = & y \\ \mathbf{p}(\mathbf{p}_0x)(\mathbf{p}_1x) & = & x \end{array}$$

as well as for the recursor:

$$\begin{array}{rcl} \mathbf{R}xy0 & = & x \\ \mathbf{R}xy(Sn) & = & yn(\mathbf{R}xyn) \end{array}$$

In $\mathsf{HA}^\omega$ we cannot prove the following *extensionality axiom*:

$$\forall f^{\sigma\to\tau}, g^{\sigma\to\tau}\big(\,(\forall x^\sigma\, fx =_\tau gx) \to f =_{\sigma\to\tau} g\big).$$

The result of adding this axiom to $\mathsf{HA}^\omega$ will be denoted by $\mathsf{E\text{-}HA}^\omega$.

To both $\mathsf{HA}^\omega$ and $\mathsf{E\text{-}HA}^\omega$ we can add classical logic (in the form of the Law of Excluded Middle $\varphi \vee \neg\varphi$ or Double Negation Elimination $\neg\neg\varphi \to \varphi$): the resulting systems will be denoted by $\mathsf{PA}^\omega$ and $\mathsf{E\text{-}PA}^\omega$, respectively.

Some remarks about these systems:

(1) All these systems satisfy the deduction theorem, so one can freely use natural deduction to prove things in these systems.

(2) For any formula $\varphi(x)$ in the language of $\mathsf{HA}^\omega$ we have

$$\mathsf{HA}^\omega \vdash x =_\sigma y \wedge \varphi(x) \to \varphi(y).$$

Indeed, this is quite easy to prove by induction on the structure of $\varphi$. And from this it follows that the same is provable in all other systems, because $\mathsf{HA}^\omega$ is a subsystem of all of them.

(3) It is sometimes convenient to regard disjunction as a defined connective in $\mathsf{HA}^\omega$. The reason for this is that $\varphi \vee \psi$ is provably equivalent to

$$\exists n^0\,\big[\,(n = 0 \to \varphi) \wedge (\neg n = 0 \to \psi)\,\big]$$

in $\mathsf{HA}^\omega$. Moreover, if we regard $\varphi \vee \psi$ as an abbreviation for this expression, the logical axioms for disjunction can be proved on the basis of the other axioms of $\mathsf{HA}^\omega$.

## 2. Lambda abstraction

In working with $\mathsf{HA}^\omega$ lambda notation is essential. The following propositions explains and justifies its use:

PROPOSITION 2.1. *For any variable $x$ and term $t$ in the language of $\mathsf{HA}^\omega$ there is another term denoted by $\lambda x.t$ such that*

$$\mathsf{HA}^\omega \vdash (\lambda x.t)t' = t[t'/x].$$

PROOF. We define $\lambda x.t$ by induction on the complexity of $t$.

(i) If $t$ is just $x$, then $\lambda x.t := \mathbf{skk}$.
(ii) If $t$ consists just of a variable $y$ distinct from $x$ or $t$ is a constant, then $\lambda x.t := \mathbf{k}t$.
(iii) If $t = t_0 t_1$, then $\lambda x.t := \mathbf{s}(\lambda x.t_0)(\lambda x.t_1)$.

□

One can repeat $\lambda$-abstraction as in $\lambda x.\lambda y.t$; but instead of $\lambda x.\lambda y.t$ we will usually write $\lambda xy.t$.

Using this proposition one can define the usual arithmetical operations. For example, there is a closed term **plus** of type $0 \to (0 \to 0)$ such that:

$$\begin{array}{rcl} \mathbf{plus}\, m\, 0 & = & m \\ \mathbf{plus}\, m\, Sn & = & S(\mathbf{plus}\, m\, n) \end{array}$$

From these equations the standard properties of addition (like associativity and commutativity) can be derived. Indeed, one can define **plus** as:

$$\mathbf{plus} := \lambda mn.\mathbf{R}m(\lambda xy.Sy)n$$

(please check!). Exercise: also define multiplication and exponentiation and derive the standard "high school identities".