Chapter 11

Second-order logic

{11}

11.1 Intuitionistic second-order logic

There exists a close connection between intuitionistic second-order logic and the so-called *polymorphic lambda calculus* $\boldsymbol{\lambda 2}$. $\boldsymbol{\lambda 2}$ is an extension of $\boldsymbol{\lambda}_{\rightarrow}$ permitting abstraction over type variables, and under the formulas-as-types paradigm it may be regarded as isomorphic to the natural deduction system for intuitionistic second-order propositional logic $\rightarrow\forall^2\mathbf{Nip}^2$, that is to say, $\rightarrow\mathbf{Nmp}$ extended with quantification over propositions. $\boldsymbol{\lambda 2}$ is an important component of various type systems studied in computer science.

In this chapter we show how strong normalization for intuitionistic second-order predicate logic may be reduced to normalization for $\mathbf{Nmp}^2$, and we present a proof of strong normalization for $\rightarrow\forall^2\mathbf{Nip}^2$ ($= \boldsymbol{\lambda 2}$). Next we show that intuitionistic second-order arithmetic $\mathbf{HA}^2$ (formal intuitionistic analysis) can be represented in intuitionistic second-order logic, and show that every recursive function which is provably total in $\mathbf{HA}^2$ can be represented by a term of $\boldsymbol{\lambda 2}$. ¿From the formal deduction establishing totality of the function being considered, one can read off an algorithm, encoded by a term of $\boldsymbol{\lambda 2}$, for computing the function.

11.1.1. *Description of* $\mathbf{Ni}^2$

To the first-order language, second-order quantifiers $\forall X^n, \exists X^n$ are added. If we wish to distinguish the symbols for second-order quantification from those for first-order quantification, we write $\forall^2, \exists^2$. The additional quantifier rules are given by:

$$\frac{A}{\forall Y^n A[X^n/Y^n]}\,\forall^2\mathrm{I} \qquad\qquad \frac{\forall X^n A}{A[X^n/\lambda x_1 \ldots x_n.B]}\,\forall^2\mathrm{E}$$

$$\frac{A[X^n/\lambda x_1 \ldots x_n.B]}{\exists X^n A}\,\exists^2\mathrm{I} \qquad\qquad \frac{\exists X^n A \qquad \begin{matrix} A^n[X^n/Y^n] \\ \mathcal{D} \\ C \end{matrix}}{C}\,\exists^2\mathrm{E}$$

where

- in $\forall^2$I A does not depend on open hypotheses containing X^n free, and Y^n is free for X^n in A;
- $A[X^n/\lambda x_1 \ldots x_n.B]$ is obtained from A by replacing each occurrence of a subformula $X^n t_1 \ldots t_n$ in A by $B[x_1, \ldots, x_n/t_1, \ldots, t_n]$;
- C in the premise of $\exists^2$E does not depend on open assumptions containing Y^n free except the assumptions in $[A[X^n/Y^n]]$.

These rules are the same as in 6.6.3, except that the restriction on B has been dropped.

11.1.2. *Restriction to the language with* $\to, \forall, \forall^2$ {*restr*}

There is a good deal of redundancy in the operators of second-order logic, since we can define $\wedge, \vee, \bot, \exists, \exists^2$ from $\to, \forall, \forall^2$ as follows. In the definitions below X^0 is not free in A, B.

$$
\begin{array}{lcl}
\bot & := & (\forall X^0)X^0, \\
A \wedge B & := & \forall X^0((A \to (B \to X)) \to X), \\
A \vee B & := & \forall X^0((A \to X) \to ((B \to X) \to X)), \\
\exists y A & := & \forall X^0(\forall y(A \to X) \to X), \\
\exists Y^n A & := & \forall X^0(\forall Y^n(A \to X) \to X)).
\end{array}
$$

Under these definitions, the usual introduction and elimination rules for the defined operators become derived rules in the fragment based on $\to, \forall, \forall^2$. For example, the following deductions show that $\wedge$I, $\wedge \mathrm{E_L}$ are derivable.

$$
\dfrac{\dfrac{\dfrac{\dfrac{A\to(B\to X^0)^x \quad A}{B \to X^0} \qquad B}{X^0}}{(A\to(B\to X^0))\to X^0}\,x}{\forall X^0((A\to(B\to X))\to X)}
\qquad
\dfrac{\dfrac{\forall X^0((A\to(B\to X))\to X)}{(A\to(B\to A))\to A} \qquad \dfrac{\dfrac{A^x}{B\to A}}{A\to(B\to A)}\,x}{A}
$$

and the proof of $\wedge \mathrm{E_R}$ is similar to the proof of $\wedge \mathrm{E_L}$. Note that the distinction between minimal and intuitionistic logic disappears in second-order logic.

Henceforth we shall assume $\mathbf{I}^2$ to be formulated with primitives $\to, \forall, \forall^2$ only, unless expressly indicated otherwise.

11.1.2A. ♠ Derive the rules for the other defined operators.

11.1.3. *Normal deductions and normalization for* $\mathbf{Ni}^2$

We can formulate notions of contraction, reduction and normal form in the same way as for **Ni**. In particular, to the detour contractions of the $\rightarrow\forall$-fragment of **Ni** we add $\forall^2$-detour contractions:

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D}\\ A\end{array}}{\forall Y^n A[x^n/y^n]}}{A[X^n/\lambda\vec{x}.B]} \qquad \text{cont} \qquad \begin{array}{c}\mathcal{D}[X^n/\lambda\vec{x}.B]\\ A[X^n/\lambda\vec{x}.B]\end{array}$$

Since we restrict attention to the $\rightarrow\forall\forall^2$-language, there is no need for considering permutative contractions and immediate simplifications. But it is of interest to note that the detour contractions for the operators $\wedge, \vee, \bot, \exists, \exists^2$ correspond under the definitions to transformations of prooftrees which result from $\rightarrow\forall\forall^2$-detour contractions. On the other hand, permutative contractions correspond after translation under the definitions to transformations not generated by the $\rightarrow\forall\forall^2$-detour contractions.

{*SNred*}

11.1.4. Proposition. *Strong normalization w.r.t. detour contractions of* $\mathbf{Ni}^2$ *in the full language is reducible to strong normalization w.r.t. detour contractions for the* $\rightarrow\forall\forall^2$*-fragment of* $\mathbf{Ni}^2$.

11.1.4A. ♠ Prove the preceding proposition in detail.

11.1.4B. ♠ Give an example of a permutation contraction in the full second-order language which does not translate into a series of contractions relative to $\rightarrow, \forall$ and $\forall^2$.

There is no meaningful subformula property for second-order logic. Any $A[X^n/\lambda\vec{x}.B]$ ought to count as a subformula of $\forall X^n A$, but the logical complexity of the subformula may be very much larger than the complexity of $\forall X^n A$. This is also the reason why it does not seem to be worthwhile to strive for normalization relative to the full language, including permutative contractions.

Nevertheless, some useful conclusions can be drawn from the fact that a derivation can be brought into normal form.

For one thing, a normal derivation of a first-order formula (i.e. a formula without $\forall^2$) does not contain second-order quantifiers. Another example is given by the following (cf. 6.2.7D):

11.1.5. Proposition. *A normal derivation without open assumptions ends with an introduction.*

Proof. Let $\mathcal{D}$ be a normal derivation without open assumptions. Assume that $\mathcal{D}$ ends with an elimination rule. Follow a main branch (defined as in

6.2.5) starting from the conclusion. We pass through eliminations only; at the top we find a formula which cannot be an open assumption (there are none, by hypothesis), but also cannot be discharged by $\to$I since there are no introductions below this formula; contradiction. ⊠

11.1.6. PROPOSITION. *Let $\mathcal{D}$ be a normal derivation in* $\mathbf{Ni}^2$ *of* $A \vee B$ *without open hypotheses, i.e.* $\mathcal{D}$ *derives* $\forall X^0((A \to X) \to (B \to X) \to X)$. *Then* $\mathbf{Ni}^2 \vdash A$ *or* $\mathbf{Ni}^2 \vdash B$.

PROOF. **First proof.** There is an immediate proof from the preceding proposition, if we consider normal deduction w.r.t. the full language, so that $\vee$ appears as a primitive.

However, it is instructive to see how we have to argue if $\vee$ is defined and we are reasoning about the $\to\forall\forall^2$-fragment.

Second proof. Let us follow the main branch (defined as in 6.2.5) of $\mathcal{D}$, going upwards from the conclusion.

If the final rule applied in $\mathcal{D}$ is an E-rule, then the main branch passes through eliminations only and its topmost formula must be an open assumption, which is impossible. So the final rule applied is $\forall^2$I. The immediate subdeduction $\mathcal{D}_1$ of $\mathcal{D}$ therefore terminates with $(A \to X) \to (B \to X) \to X$ and has no open assumptions. Again, the final rule of $\mathcal{D}_1$ must have been an introduction, so the premise $(B \to X) \to X$ is the conclusion of a subdeduction $\mathcal{D}_2$ from open assumptions of the form $A \to X$.

Case 1. $A \to X$ does actually occur as the top formula of the main branch. Then the first rule applied to $A \to X$ has to be an elimination rule, otherwise $A \to X$ or $\forall X(A \to X)$ would have to occur as a subformula of the conclusion in the strict sense. We see that this is impossible, if we keep in mind that X does not occur in A or B. So $\to$E is applied at the top:

$$\dfrac{A \to X \qquad \begin{array}{c}\mathcal{D}' \\ A\end{array}}{X}$$

and then introductions must follow in the main branch. $\mathcal{D}'$ may use $A \to X, B \to X$ as open assumptions. But if in $\mathcal{D}'$ we replace the X everywhere by $A \vee B$, these assumptions become derivable and we have found a deduction for A.

Case 2. If there is no formula $A \to X$ at the top of the main branch, it must be the case that the final rule of $\mathcal{D}_2$ is an introduction discharging $B \to X$ at the top of the main branch and possibly other places. The argument is similar to the preceding case, and we find a deduction of B. ⊠

11.2 $\mathbf{Ip}^2$ and $\lambda 2$

11.2.1. $\mathbf{Ni}^2$ *and polymorphic lambda calculus*

In a more or less routine fashion we can reformulate $\mathbf{Ni}^2$ as a calculus of typed terms. We shall not give here the description for the full calculus, but only for intuitionistic second-order *propositional* logic $\mathbf{Ip}^2$. In $\mathbf{Ip}^2$ only propositional variables $X^0, Y^0, Z^0, \ldots$ occur; we drop the superscript ^{0}. As in the full system, $\perp, \wedge, \vee, \exists, \exists^2$ are definable.

The new clauses describing the formation of terms for deductions in propositional $\mathbf{Ni}^2$ are given by

$$\frac{t{:}\, A}{\Lambda X.t{:}\,(\forall X)A} \qquad \frac{t{:}\,(\forall X)A}{t^{(\forall X)A}B{:}\, A[X/B]}$$

There is an obvious condition to be met in the case of $\forall^2$I: t may not contain free individual variables with a type in which X occurs free. We have added new operators of *type abstraction*, ΛX, and *type application* (application of a term to a type (= formula). The resulting calculus of typed terms is also known as the *polymorphic lambda calculus* or *system* F ([?,?]) or $\boldsymbol{\lambda}\mathbf{2}$.

Henceforth *term* will be used for first-order terms; for second-order terms we use *type* or *formula*.

The contractions for $\to$ and $\forall^2$ correspond in term notation to

$$(\lambda x^A.t^B)(s^A) \ \text{ cont } \ t[x^A/s^A]{:}\, B$$
$$(\Lambda X.t^A{:}\,(\forall X)B)A \ \text{ cont } \ t[X/A]{:}\, B[X/A]$$

Observe that we do *not* have application on the level of types: instead of having $((\forall X)A)B$ contract to $A[X/B]$, we simply identify $((\forall X)A)B$ with $A[X/B]$, so that we can in fact dispense with the notation $((\forall X)A)B$.

For this system we shall prove strong normalization and uniqueness of normal form in the next section. The remainder of this section is devoted to computational aspects of $\boldsymbol{\lambda}\mathbf{2}$, in preparation for section 11.5.

11.2.2. *Computational content*

{compcont2}

DEFINITION. Let X be a fixed propositional variable in $\boldsymbol{\lambda}_{\to}$. We put

$$\mathrm{N}'_X := X \to ((X \to X) \to X).$$

This is the *type* of *(variant) natural numbers over* X. The so-called *(variant) Church numerals* are terms of type N'_X in normal form:

$$\bar{n}'_X := \lambda x^X f^{X\to X}.f^n(x) : \mathrm{N}'_X,$$

where as before

$$f^0(x) := x; \quad f^{n+1}(x) := f(f^n(x)).$$

We shall drop the subscript X in the sequel, since it will be kept fixed. ⊠

Of course we can also define $\bar{n}'_A$ and N'_A for an arbitrary type A in the same manner.

NOTATION. In the remainder of this chapter we shall simply write $\mathrm{N}_A, \bar{n}_A$ for $\mathrm{N}'_A, \bar{n}'_A$ respectively. ⊠

11.2.2A. ♠ Show that the variant Church numerals of type X are the only terms in normal form of type $\mathrm{N} \equiv \mathrm{N}_X$.

11.2.2B. ♠ All extended polynomials are representable in the variant Church numerals. *Hint.* Take as representing terms

$$\begin{array}{lcl} F_{+} & := & \lambda x^N y^N z^X f^{X \to X}.x(yzf)f \\ F_{\times} & := & \lambda x^N y^N z^X f^{X \to X}.xz(\lambda u^X.yuf) \\ F_{\mathbf{p}_i^k} & := & \lambda x_1^N \ldots x_k^N.x_i \\ F_{\mathbf{c}_n} & := & \lambda x^N.\bar{n} \\ F_{\mathrm{sg}} & := & \lambda x^N y^X f^{X \to X}.xy(\lambda z^X.fy) \\ F_{\overline{\mathrm{sg}}} & := & \lambda x^N y^X f^{X \to X}.x(fy)(\lambda z^X.y). \end{array}$$

As we have seen before in 1.2.20, the class of representable functions becomes larger, if one considers N_A for arbitrary A, and permits representing terms where the types of the input numerals and output numerals may differ.

In $\boldsymbol{\lambda}_{\to}$ there is arbitrariness in the choice of the type X in N_X. This arbitrariness is removed in $\boldsymbol{\lambda 2}$. There we put:

11.2.3. DEFINITION.

$$\mathrm{N} := \forall X(X \to ((X \to X) \to X)),$$

$$\bar{n} := \Lambda X \lambda x^X f^{X \to X}.f^n(x).$$ ⊠

The results on representability of extended polynomials carry over to $\boldsymbol{\lambda 2}$, modulo small adaptations. But as will become clear from the exercises, many more functions are representable in $\boldsymbol{\lambda 2}$ than are representable in $\boldsymbol{\lambda}_{\to}$. In particular, the functions representable in $\boldsymbol{\lambda 2}$ are closed under recursion.

11.2.3A. ♠ Show that the following operators $\mathbf{p}, \mathbf{p}_0, \mathbf{p}_1$ may be taken as pairing with inverses for types U, V with a defined product type $U \wedge V$:

$$\begin{array}{l} \mathbf{p} := \lambda u^U v^V \Lambda X \lambda x^{U \to (V \to X)}.xuv, \\ \mathbf{p}_0 := \lambda x^{U \wedge V}.xU(\lambda y^U z^V.y), \quad \mathbf{p}_1 := \lambda x^{U \wedge V}.xV(\lambda y^U z^V.z). \end{array}$$

N.B. These terms encode the deductions exhibited in 11.1.2.

11.2.3B. ♠ Let $\mathrm{S} \equiv \lambda z^N \Lambda X \lambda x^X y^{X\to X}.y(zXxy)$, $\mathsf{It} \equiv \Lambda X.\lambda u^X f^{X\to X} z^N.zXuf$. Show that

$$\begin{aligned}\mathsf{It}\,X\,u^X f^{X\to X}\bar{0} \quad &= u,\\ \mathsf{It}\,X\,u^X f^{X\to X}(\mathrm{S}t) &= f(\mathsf{It}\,X\,u^X f^{X\to X}t).\end{aligned}$$

11.2.3C. ♠ Define with the help of pairing and the iterator of the preceding exercise a *recursor* Rec such that

$$\begin{aligned}&\mathsf{Rec}\,X\,u^X f^{X\to(N\to X)}\bar{0} = u^X,\\ &\mathsf{Rec}\,X\,u^X f^{X\to(N\to X)}(\overline{\mathrm{S}n}) = f(\mathsf{Rec}\,uf\bar{n})\bar{n}.\end{aligned}$$

11.3 Strong normalization for $\mathbf{Ni}^2$ {SN2}

We first show that strong normalization for full second-order logic can be reduced to strong normalization for propositional $\mathbf{Ni}^2$ or $\boldsymbol{\lambda 2}$.

11.3.1. PROPOSITION. *Strong normalization for* $\mathbf{Ni}^2$ *is a consequence of strong normalization for propositional* $\mathbf{Ni}^2$.

PROOF. We define a mapping ϕ from formulas and deductions of $\mathbf{Ni}^2$ to formulas and deductions of propositional $\mathbf{Ni}^2$, as follows.

$$\phi(X^n t_1 \dots t_n) := X^*$$

where X^* is some propositional variable bijectively associated with the relation variable X^n, and for compound formulas we put

$$\begin{aligned}\phi(A \to B) &:= \phi A \to \phi B,\\ \phi(\forall X^n A) &:= \forall X^* \phi A,\\ \phi(\forall x A) &:= \forall x^* \phi A, \quad x^* \text{ not free in } \phi A.\end{aligned}$$

Here x^* is a *propositional* variable associated to the individual variable x. Deductions are translated as follows. We write $\overset{\phi}{\mapsto}$ for "ϕ maps to".

• The single-node prooftree "A" is translated into the single-node prooftree "ϕA".

• For prooftrees of depth greater than one, we define inductively

$$\dfrac{\begin{matrix}\mathcal{D}\\ B\end{matrix}}{A \to B}\to\mathrm{I} \qquad \overset{\phi}{\mapsto} \qquad \dfrac{\begin{matrix}\phi\mathcal{D}\\ \phi B\end{matrix}}{\phi A \to \phi B}\to\mathrm{I}$$

$$\dfrac{\begin{matrix}\mathcal{D}\\ A \to B\end{matrix} \quad \begin{matrix}\mathcal{D}'\\ A\end{matrix}}{B}\to\mathrm{E} \qquad \overset{\phi}{\mapsto} \qquad \dfrac{\begin{matrix}\phi\mathcal{D}\\ \phi A \to \phi B\end{matrix} \quad \begin{matrix}\phi\mathcal{D}'\\ \phi A\end{matrix}}{\phi B}\to\mathrm{E}$$

$$\dfrac{\begin{array}{c}\mathcal{D}\\ A[X^n]\end{array}}{\forall Y^n A[x^n/Y^n]}\,\forall^2\mathrm{I} \quad\overset{\phi}{\mapsto}\quad \dfrac{\begin{array}{c}\phi\mathcal{D}\\ \phi(A[X^n])\end{array}}{\phi(\forall Y^* A[x^n/Y^n])}\,\forall^2\mathrm{I}$$

$$\dfrac{\begin{array}{c}\mathcal{D}\\ \forall X^n A\end{array}}{A[X^n/\lambda\vec{x}.B]}\,\forall^2\mathrm{E} \quad\overset{\phi}{\mapsto}\quad \dfrac{\begin{array}{c}\phi\mathcal{D}\\ \forall X^*\phi(A)\end{array}}{\phi(A)[X^*/\phi B]}\,\forall^2\mathrm{E}$$

$$\dfrac{\begin{array}{c}\mathcal{D}\\ A[x]\end{array}}{\forall y A[x/y]}\,\forall\mathrm{I} \quad\overset{\phi}{\mapsto}\quad \dfrac{\begin{array}{c}\phi\mathcal{D}\\ \phi(A[x])\end{array}}{\forall y^*\phi(A[x])}\,\forall^2\mathrm{I}$$

$$\dfrac{\begin{array}{c}\mathcal{D}\\ \forall x A\end{array}}{A[x/t]}\,\forall\mathrm{E} \quad\overset{\phi}{\mapsto}\quad \dfrac{\begin{array}{c}\phi\mathcal{D}\\ \forall x^*\phi A\end{array}}{\phi A}\,\forall^2\mathrm{E}$$

Note that for all *individual* variables $\vec{x}$ and terms $\vec{t}$

$$\phi A \equiv \phi(A[\vec{x}/\vec{t}]).$$

Furthermore,

$$\mathcal{D} \succ_1 \mathcal{D}' \Rightarrow \phi\mathcal{D} \succ \phi(\mathcal{D}'). \qquad \boxtimes$$

11.3.2. In contrast to the situation for $\boldsymbol{\lambda}_{\rightarrow}$, the propositional variables in $\mathbf{Ni}^2$ cannot simply be regarded as formulas of minimal complexity, since in the course of the normalization process quite complex formulas may be substituted for these variables. The notion of computability for $\boldsymbol{\lambda}\mathbf{2}$ has to reflect this; the idea is to assign "variable computability predicates" to the propositional variables. A "computability candidate" has to satisfy certain requirements. A straightforward generalization of 6.8.3 (on which the proof of strong normalization of $\boldsymbol{\lambda}\mathbf{2}$ in [**?**] is based) is obtained by defining the notion of a computability candidate as follows.

Definition. A term t is called *non-introduced* if t is not of the form $\lambda x.s$ or $\Lambda X.s$. A set of terms $\mathbf{X}$, all of the same type A, is a *computability candidate* (a *c.c.*) (*of type* A) iff

CC1 If $t \in \mathbf{X}$ then $\mathrm{SN}(t)$;

CC2 If $t \in \mathbf{X}$ and $t \succeq t'$ then $t' \in \mathbf{X}$;

CC3 If t is non-introduced, and $\forall t' \prec_1 t(t' \in \mathbf{X})$ then $t \in \mathbf{X}$. $\boxtimes$

As a corollary of CC3 we find

CC4 If $t : A$ is non-introduced and normal, then $t \in \mathbf{X}$.

Instead of this definition, obtained by transferring the properties C1–3 in 6.8.3 from computability predicates for function types to c.c.'s, we use a slightly different definition which is convenient for generalizations.

11.3.3. DEFINITION. (*$\mathcal{P}$-candidate, candidate-closed*) Let $\mathcal{P}$ be a property of terms of $\boldsymbol{\lambda 2}$, and let $\mathcal{P}_A$ be the set of terms of type A in $\mathcal{P}$.

In this section, $\vec{\varepsilon}$ will be used for a sequence of first- and second-order terms (types). We say that $\vec{\varepsilon}$ is in $\mathcal{P}$, if the first-order terms of $\vec{\varepsilon}$ are in $\mathcal{P}$. A set $\mathbf{A}$ of terms is said to be a *$\mathcal{P}$-candidate of type A* (notation $\mathbf{A}{:}\,A$) if $\mathbf{A}$ consists of terms of type A such that

$\mathcal{P}$-C1 If $t \in \mathbf{A}$, then $t \in \mathcal{P}$.

$\mathcal{P}$-C2 If $\vec{\varepsilon} \in \mathcal{P}$, then for any variable x of the appropriate type $x\vec{\varepsilon} \in \mathbf{A}$.

$\mathcal{P}$-C3 If $t[x/r]\vec{\varepsilon} \in \mathbf{A}$, $t, r, \vec{\varepsilon} \in \mathcal{P}$, then $(\lambda x.t)r\vec{\varepsilon} \in \mathbf{A}$.

$\mathcal{P}$-C4 If $t[X/B]\vec{\varepsilon} \in \mathbf{A}$, $t, \vec{\varepsilon} \in \mathcal{P}$, then $(\Lambda X.t)B\vec{\varepsilon} \in \mathbf{A}$.

The set $\mathcal{P}$ is *candidate-closed* if

Cl1 If $tx \in \mathcal{P}$, then $t \in \mathcal{P}$.

Cl2 If $tX \in \mathcal{P}$, then $t \in \mathcal{P}$.

Cl3 $\mathcal{P}_A$ is a $\mathcal{P}$-candidate.

In the sequel we drop the "$\mathcal{P}$-" and talk about *candidates* instead of $\mathcal{P}$-candidates. ⊠

11.3.4. DEFINITION. The following definition of a predicate of *strong computability* "Comp" extends the definition given before for predicate logic.

Let A be a formula with $\mathrm{FV}(A) \subset \vec{X}$, and let $\vec{B} \equiv B_1, \ldots, B_n$ be a sequence of formulas of the same length, and let $\vec{\mathbf{B}} \equiv \mathbf{B}_1, \ldots, \mathbf{B}_n$ be a sequence of candidates with $\mathbf{B}_i{:}\,B_i$. We define $\mathrm{Comp}_A[\vec{X}/\vec{B}]$ (*computability under assignment* of $\vec{\mathbf{B}}$ to $\vec{X}$) as follows:

(i) $\mathrm{Comp}_{X_i}[\vec{X}/\vec{\mathbf{B}}] := \mathbf{B}_i$,

(ii) $\mathrm{Comp}_{B\to C}[\vec{X}/\vec{\mathbf{B}}] := \mathrm{Comp}_B[\vec{X}/\vec{\mathbf{B}}] \to \mathrm{Comp}_C[\vec{X}/\vec{\mathbf{B}}]$,

(iii) $\mathrm{Comp}_{(\forall Y)C}[\vec{X}/\vec{\mathbf{B}}] := \{t{:}\,\forall D \forall \mathbf{D}{:}\,D(tD \in \mathrm{Comp}_C[\vec{X}, Y/\vec{\mathbf{B}}, \mathbf{D}])\}$
(D ranging over types). ⊠

NOTATION. In order to save on notation, we shall use in the remainder of this section a standard abbreviation: we write Comp^*_D for $\mathrm{Comp}_D[\vec{X}/\vec{\mathbf{B}}]$, where $\mathbf{B}_i{:}\,B_i$ are candidates ($1 \le i \le m$), fixed in each proof. So $\mathrm{Comp}^*_D[Y/\mathbf{C}]$ stands for $\mathrm{Comp}_D[\vec{X}, Y/\vec{\mathbf{B}}, \mathbf{C}]$ etc. ⊠

{*compcand*}

11.3.5. LEMMA. Comp^*_C *is a candidate of type* C.

PROOF. We have to show $\mathcal{P}$-C1–4 with Comp for $\mathcal{P}$, i.e.
(i) If $r \in \mathrm{Comp}^*_A$ then $r \in \mathcal{P}$.
(ii) If $\vec{\varepsilon} \in \mathcal{P}$ then $x\vec{\varepsilon} \in \mathrm{Comp}^*_A$.
(iii) If $r, s, \vec{\varepsilon} \in \mathcal{P}$, $r[x/s]\vec{\varepsilon} \in \mathrm{Comp}^*_A$ then $(\lambda x.r)s\vec{\varepsilon} \in \mathrm{Comp}^*_A$.
(iv) If $r, \vec{\varepsilon} \in \mathcal{P}$, $r[Y/C]\vec{\varepsilon} \in \mathrm{Comp}^*_A$ then $(\Lambda Y.r)C\vec{\varepsilon} \in \mathrm{Comp}^*_A$.

(i) and (ii) are proved simultaneously by induction on the depth of A. For (i) we assume $r \in \mathrm{Comp}^*_A$, for (ii) $\vec{\varepsilon} \in \mathcal{P}$.
Case (i)1. $A \equiv X_i$. Then the premise says that $r \in \mathbf{B}_i$, hence $r \in \mathcal{P}$, since $\mathbf{B}_i$ satisfies $\mathcal{P}$-C1.
Case (ii)1. By $\mathcal{P}$-C2 for $\mathbf{B}_i$ we find $x\vec{\varepsilon} \in \mathbf{B}_i$, and hence $x\vec{\varepsilon} \in \mathrm{Comp}^*_{X_i}$.
Case (i)2. $A \equiv A_1 \to A_2$. We have to show $r \in \mathcal{P}$. Since Cl 1 for $\mathcal{P}$ holds, it suffices to show $rx^{A_1} \in \mathcal{P}$. By (ii) for A_1 (IH) it follows that $x \in \mathrm{Comp}^*_{A_1}$, hence $rx \in \mathrm{Comp}^*_{A_2}$; with (i) for A_2 (IH) we find $rx \in \mathcal{P}$.
Case (ii)2. We have to show $x\vec{\varepsilon} \in \mathrm{Comp}^*_{A_1 \to A_2}$. Assume $s \in \mathrm{Comp}^*_{A_1}$; we then have to show $x\vec{\varepsilon}s \in \mathrm{Comp}^*_{A_2}$. By (i) for A_1 (IH) we find $s \in \mathcal{P}$, and by (ii) for A_2 we find, since $\vec{\varepsilon}, s \in \mathcal{P}$, that $x\vec{\varepsilon}s \in \mathrm{Comp}^*_{A_2}$ (using (ii) for A_2).
Case (i)3. $A \equiv (\forall Y)A_1$. We must show $r \in \mathcal{P}$. By Cl 2 it suffices to show that $rY \in \mathcal{P}$. Assume $\mathbf{C}{:}\, Y$ to be some candidate (e.g. $\mathcal{P}_Y$); then $rY \in \mathrm{Comp}_{A_1}[Y/\mathbf{C}]$. With (i) for A_1 (IH) we find $rY \in \mathcal{P}$.
Case (ii)3. We have to show that $x\vec{\varepsilon} \in \mathrm{Comp}^*_{(\forall Y)A_1}$. So let $\mathbf{C}{:}\, C$ be a candidate, then we must show that $x\vec{\varepsilon}C \in \mathrm{Comp}^*_{A_1}[y/\mathbf{C}]$. But this is a consequence of (ii) for A_1, since $x\vec{\varepsilon}C \in \mathcal{P}$.

(iii) is proved by induction on the depth of A. We assume $r, s, \vec{\varepsilon} \in \mathcal{P}$, $r[x/s]\vec{\varepsilon} \in \mathrm{Comp}^*_A$.
Case (iii)1. $A \equiv X_i$. We have to show $(\lambda x.r)s\vec{\varepsilon} \in \mathbf{B}_i$. But $r, s, \vec{\varepsilon} \in \mathcal{P}$, so this follows from $\mathcal{P}$-C3.
Case (iii)2. $A \equiv A_1 \to A_2$. Let $t \in \mathrm{Comp}^*_{A_1}$. Then we must show $(\lambda x.r)s\vec{\varepsilon}t \in \mathrm{Comp}^*_{A_2}$. By property (i) $t \in \mathcal{P}$; by IH for A_1 it suffices to show that $r[x/s]\vec{\varepsilon}t \in \mathrm{Comp}^*_{A_2}$, which holds by definition of Comp.
Case (iii)3. $A \equiv (\forall Y)A_1$. Let $\mathbf{C}{:}\, C$ be a candidate. We must show $(\lambda x.r)s\vec{\varepsilon}C \in \mathrm{Comp}^*_{A_1[Y/C]}[Y/\mathbf{C}]$. By the IH for A_1 it suffices to show that $r[x/s]\vec{\varepsilon}C \in \mathrm{Comp}^*_{A_1[Y/C]}[Y/\mathbf{C}]$, which holds by definition.

(iv) is again proved by induction on the depth of A.
Cases (iv)1 and (iv)2. Left to the reader.
Case (iv)3. $A \equiv (\forall Z)A_1$. Let $r\vec{\varepsilon} \in \mathcal{P}$, $r[Y/C]\vec{\varepsilon} \in \mathrm{Comp}^*_{(\forall Z)A_1}$. We have to show $(\Lambda Z.r)C\vec{\varepsilon} \in \mathrm{Comp}^*_{(\forall Z)A_1}$. Let $\mathbf{D}{:}\, D$ be a candidate. We must show $(\Lambda Y.r)C\vec{\varepsilon}D \in \mathrm{Comp}^*_{A_1}[Z/\mathbf{D}]$. By the IH it suffices to show $r[Y/C]\vec{\varepsilon}D \in \mathrm{Comp}^*_{A_1}[Z/\mathbf{D}]$, which is trivial. ⊠

11.3.5A. ♠ Supply the proofs of the missing cases for (iv) in the proof of the preceding lemma.

11.3.6. LEMMA. *Let $\mathcal{P}$ be candidate-closed, and let $\vec{\mathbf{B}}: \vec{B}$ be a sequence of candidates as before. Then*

$$\mathrm{Comp}_{A[Y/C]}[\vec{X}/\vec{\mathbf{B}}] = \mathrm{Comp}_A[\vec{X}, Y/\vec{\mathbf{B}}, \mathrm{Comp}_C[\vec{X}/\vec{\mathbf{B}}]].$$

(The left side is well-defined by the preceding lemma.)

PROOF. By induction on the complexity of A. Let $A \equiv \forall Z.B$. Then

$$\begin{aligned}
&\mathrm{Comp}^*_{(\forall ZB)[Y/C]} := \\
&\{t : \forall D \forall \mathbf{D}{:}D(tD \in \mathrm{Comp}^*_{B[Y/C]}[Z/\mathbf{D}])\} = \\
&\{t : \forall D \forall \mathbf{D}{:}D(tD \in \mathrm{Comp}^*_{B}[Y, Z/\mathrm{Comp}^*_C, \mathbf{D}])\} = \\
&\mathrm{Comp}^*_A[Y/\mathrm{Comp}^*_C].
\end{aligned}$$

We leave the other cases to the reader. ⊠

11.3.6A. ♠ Complete the proof of the preceding lemma.

11.3.7. THEOREM. *Let $t[x_1{:}A_1, \dots, x_n{:}A_n]{:}C$, and assume that the free second-order variables of $\{B_1, A_1, \dots, A_n\}$ are contained in $\{X_1, \dots, X_n\}$.*

For $1 \le i \le m$, $1 \le j \le n$ let $\mathbf{B}_i{:}B_i$ be candidates, $s_j{:}A_j[\vec{X}/\vec{B}]$, $s_j \in \mathrm{Comp}_{A_j}[\vec{X}/\vec{\mathbf{B}}]$. Then

$$t[\vec{X}/\vec{B}][\vec{x}/\vec{s}] \in \mathrm{Comp}_C[\vec{X}/\vec{\mathbf{B}}].$$

PROOF. Let us write t^* for $t[\vec{X}/\vec{B}][\vec{x}/\vec{s}]$. We prove the statement of the theorem by induction on t.

Case 1. $t \equiv x_i^{A_i}$. Immediate, since $s_i \in \mathrm{Comp}^*_{A_j}$ by assumption.

Case 2. $t \equiv rs$. By IH, r^*, s^* are strongly computable, hence so is $r^* s^*$.

Case 3. $t \equiv r^{(\forall Z)A}C{:}\,A[Z/C]$. By IH $t^* \in \mathrm{Comp}^*_{(\forall Z)A}$. By the definition of Comp^* and the fact that

$$t^*(C[\vec{X}/\vec{B}]) \equiv ((tC)[\vec{X}/\vec{B}])^* \in \mathrm{Comp}^*_{(\forall Z)A}[Z/\mathrm{Comp}^*_{C[\vec{X}/\vec{B}]}] = \mathrm{Comp}^*_{A[Z/C]}.$$

Case 4. $t \equiv \lambda z^{C_1}.r^{C_2}$. We have to show that $\lambda z^{C_1[\vec{X}/\vec{B}]}.r^* \in \mathrm{Comp}^*_{C_1 \to C_2}$. So assume $s \in \mathrm{Comp}^*_{C_1}$. Then we have to show $(\lambda z.r^*)s' \in \mathrm{Comp}^*_{C_2}$. Computability predicates are candidates (by lemma 11.3.5), so $s \in \mathcal{P}$, and by IH $r^* \in \mathcal{P}$. Hence by $\mathcal{P}$-C3 it suffices to show that

$$r^*[x/s'] = r[\vec{X}/\vec{B}][z^{C_1[\vec{X}/\vec{B}]}/s'][\vec{x}/\vec{s}] \in \mathrm{Comp}^*_{C_2[\vec{X}/\vec{B}]},$$

which holds by the IH.

Case 5. $t \equiv \lambda Z.r$. t^* is of the form $\lambda Z.r^*$. We have to show $t^* \in \mathrm{Comp}^*_{(\forall Z)A_1}$. So let $\mathbf{C}{:}C$. Then we have to show $(\lambda Z.r^*)C \in \mathrm{Comp}^*_{A_1[Z/C]}$. By IH $r^* \in \mathrm{Comp}^*_{A_1}[Z/\mathbf{C}]$, so $r^* \in \mathcal{P}$ by $\mathcal{P}$-C3, hence by $\mathcal{P}$-C4 we need only to show $r^*[Z/C] \in \mathrm{Comp}^*_{A_1}[Z/\mathbf{C}]$, which follows by IH. ⊠

COROLLARY. *If $\mathcal{P}$ is candidate-closed, all terms of $\boldsymbol{\lambda}\mathbf{2}$ belong to $\mathcal{P}$; hence, a fortiori, since strong normalizability is candidate-closed, all terms are strongly normalizable.* ⊠

11.3.8. *An auxiliary system*

In section 11.5 we should like to map deductions and formulas of second-order arithmetic to terms and types of $\boldsymbol{\lambda}\mathbf{2}$, in such a way that the notion of reducibility is preserved, normal forms go to normal forms, and second-order propositions go to the corresponding type. For this it is necessary that $(\forall X)X$ is mapped to an inhabited type of $\boldsymbol{\lambda}\mathbf{2}$; but since the type $(\forall X)X$ is uninhabited in $\boldsymbol{\lambda}\mathbf{2}$, we introduce the auxiliary system $\boldsymbol{\lambda}\mathbf{2}\boldsymbol{\Omega}$. In the next section we shall show that in $\boldsymbol{\lambda}\mathbf{2}\boldsymbol{\Omega}$ precisely the same recursive functions are representable as in $\boldsymbol{\lambda}\mathbf{2}$; this is proved by a suitable encoding of $\boldsymbol{\lambda}\mathbf{2}\boldsymbol{\Omega}$ into $\boldsymbol{\lambda}\mathbf{2}$.

DEFINITION. Let $\boldsymbol{\lambda}\mathbf{2}\boldsymbol{\Omega}$ be obtained from $\boldsymbol{\lambda}\mathbf{2}$ by adding a constant $\Omega : \forall X.X$. There are no contraction rules involving Ω. ⊠

THEOREM.

(i) *The terms in $\boldsymbol{\lambda}\mathbf{2}\boldsymbol{\Omega}$ are strongly normalizable.*

(ii) *Normal forms in $\boldsymbol{\lambda}\mathbf{2}$ and $\boldsymbol{\lambda}\mathbf{2}\boldsymbol{\Omega}$ are unique.*

(iii) *If we include conjunction as a type-forming operator, strong normalization and uniqueness of normal form remain valid for $\boldsymbol{\lambda}\mathbf{2}$ and $\boldsymbol{\lambda}\mathbf{2}\boldsymbol{\Omega}$.*

PROOF. (i) As for $\boldsymbol{\lambda}\mathbf{2}$, adding some cases where necessary.

(ii) By extending the proof of weak confluence in 1.2.10. The rest of the proof is left to the reader. ⊠

11.3.8A. ♠ Supply the missing details in the proof of the preceding theorem.

11.4 Encoding of $\lambda 2\Omega$ into $\lambda 2$

11.4.1. PROPOSITION. *Let $t : \mathrm{N} \to \mathrm{N}$ be a [closed] term in $\boldsymbol{\lambda}\mathbf{2}\boldsymbol{\Omega}$. Then there is a [closed] term t' in $\boldsymbol{\lambda}\mathbf{2}$ such that $t\bar{n} \succeq \bar{m} \Rightarrow t'\bar{n} \succeq \bar{m}$.*

PROOF. For the proof we define (1) an encoding $^\circ$ of types and terms of $\boldsymbol{\lambda}\mathbf{2}\boldsymbol{\Omega}$ into $\boldsymbol{\lambda}\mathbf{2}$, for which it can be proved that if $t \succ_1 t'$, then $t^\circ \succeq (t')^\circ$, and (2) maps W, C such that $\mathsf{W}\bar{n} \succeq \bar{n}^\circ$, $\mathsf{C}\bar{n}^\circ \succeq \bar{n}$. ¿From this the t' in the statement of the theorem is obtained as $\lambda z^{\mathrm{N}}.\mathsf{C}(t^\circ(\mathsf{W}z))$. Once the right definitions for $^\circ$, W, C have been given, the proofs become straightforward inductive arguments, which are left almost entirely to the reader.

We define the encoding map $^\circ$ on types by

$$\begin{array}{ll} X^\circ & := X, \\ (A \to B)^\circ & := A^\circ \to B^\circ \\ (\forall X.A)^\circ & := \forall X(X \to A^\circ). \end{array}$$

It follows that

$$(A[X/B])^\circ \equiv A^\circ[X/B^\circ].$$

In order to define the encoding of terms, we assume that there is associated to each type variable X a "fresh" individual variable x of type X, not occurring free in the terms to be encoded. The correspondence between the X and the associated x^X is assumed to be one-to-one. Below, in the definition of the encoding, we shall use the tacit convention that the type variables X_i, X, Y have x_i, x, y associated to them.

For A with $\mathrm{FV}(A) = \{X_1, \dots, X_p\}$, we define a term $\tau_A \colon A^\circ$ with free variables $x_1 \colon X_1, \dots, x_p \colon X_p, X_1, \dots, X_p$ as follows:

$$\begin{array}{lll} \tau_{X_i} & := & x_i \colon X_i \\ \tau_{B \to C} & := & \lambda y \colon B^\circ . \tau_C \ (y \notin \mathrm{FV}(\tau_C)) \\ \tau_{\forall X.B} & := & \Lambda X \lambda x^X . \tau_B. \end{array}$$

For closed A, $\tau_A : A^\circ$ is closed. For example,

$$\tau_{(\forall X)X} \equiv \Lambda X \lambda x^X . x \colon \forall X(X \to X).$$

We now extend $^\circ$ to terms, and associate to each $t \in A$, with free variables $X_1, \dots, X_p$, $y_1 \colon A_1, \dots, y_q \colon A_q$ a term $t^\circ \colon A^\circ$ with free variables $X_1, \dots, X_p$, $x_1 \colon X_1, \dots, x_p \colon X_p$, $y_1 \colon A_1^\circ, \dots, y_q \colon A_q^\circ$, as follows:

$$\begin{array}{llll} \text{if } A \equiv A_i & \text{then } (y_i \colon A_i)^\circ & := & y_i \colon A^\circ; \\ \text{if } A \equiv B \to C & \text{then } (\lambda y \colon B.t \colon C)^\circ & := & \lambda y \colon B^\circ . t^\circ \colon C^\circ; \\ & \phantom{\text{then }} (t \colon B \to A)(s \colon B)^\circ & := & (t^\circ \colon B^\circ \to A^\circ)(s^\circ \colon B^\circ); \\ \text{if } A \equiv (\forall X)B & \text{then } (\Lambda X.t \colon \forall X.B)^\circ & := & \Lambda X \lambda x \colon X.t^\circ \colon B^\circ; \\ \text{if } A \equiv B[X/C] & \text{then } ((t \colon (\forall X)B)C)^\circ & := & (t^\circ \colon ((\forall X)B)^\circ)C^\circ \tau_C; \\ \text{if } A \equiv (\forall X)X & \text{then } \Omega^\circ & := & \tau_{\forall X.X} \equiv \Lambda X \lambda x \colon X.x. \end{array}$$

Then one verifies commutativity with substitution:

(i) $(t[y \colon A/s \colon A])^\circ \equiv t^\circ[y \colon A^\circ/s^\circ \colon A^\circ]$,
(ii) $\tau_{A[X/B]} \equiv \tau_A[X/B^\circ][x \colon B^\circ/\tau_B \colon B^\circ]$, and
(iii) $(t[X/B])^\circ \equiv t^\circ[X/B^\circ][x \colon B^\circ/\tau_B \colon B^\circ]$

by induction on t, A and t respectively, and uses this in verifying

$$\text{if } t \succ_1 t' \text{ then } t^\circ \succeq (t')^\circ.$$

For example, we have as one step in the induction for (ii):

$$\begin{aligned}
&\tau_{(\forall Y)C[X/B]} \equiv \Lambda Y \lambda y^Y.\tau_{C[X/B]} \equiv \\
&\Lambda Y \lambda y^Y.(\tau_C[X/B^\circ][x{:}\, B^\circ/\tau_B{:}\, B^\circ]) \equiv \\
&(\Lambda Y \lambda y^Y.(\tau_C))[X/B^\circ][x{:}\, B^\circ/\tau_B{:}\, B^\circ] \equiv \\
&\tau_{(\forall Y)C}[X/B^\circ][x{:}\, B^\circ/\tau_B{:}\, B^\circ].
\end{aligned}$$

Further, we note that

$$\begin{aligned}
\mathrm{N}^\circ \;\;&\equiv \forall X(X \to (X \to ((X \to X) \to X))) \\
(\bar{n})^\circ &\equiv \Lambda X \lambda x^X y^X z^{X\to X}.z^n y.
\end{aligned}$$

We can define operators W ("weaken") and C ("contract") such that

$$\mathsf{W}\bar{n} \succeq \bar{n}^\circ, \quad \mathsf{C}\bar{n}^\circ \succeq \bar{n};$$

simply take

$$\begin{aligned}
\mathsf{W} &:= \lambda u^{\mathrm{N}} \Lambda X \lambda x^X y^X z^{X\to X}.(uXxz), \\
\mathsf{C} &:= \lambda u^{\mathrm{N}^\circ} \Lambda X \lambda y^X z^{X\to X}.uXyyz.
\end{aligned}$$

For the term t' of the proposition we may now take

$$t' := \lambda z^{\mathrm{N}}.\mathsf{C}(t^\circ(\mathsf{W}z)). \qquad \boxtimes$$

11.4.1A. ♠ Fill in the missing details of this proof.

11.5 Provably recursive functions of $\mathbf{HA}^2$ {repprov}

11.5.1. DEFINITION. *Intuitionistic second-order arithmetic* $\mathbf{HA}^2$ (alternative notation **HAS**) is obtained taking the language of pure second-order logic, with a single individual constant 0 (*zero*), a single function constant S (*successor*), and a binary predicate symbol = for equality between individuals, with the axioms and rules of intuitionistic second-order logic, and the following axioms for equality and successor:

$$\begin{aligned}
&\forall x(x = x), \\
&\forall X^1 \forall xy(x = y \wedge Xx \to Xy), \\
&\forall xy(\mathrm{S}x = \mathrm{S}y \to x = y), \\
&\forall x(\neg \mathrm{S}x = 0), \text{ i.e. } \forall x(\mathrm{S}x = 0 \to (\forall X^0)X),
\end{aligned}$$

and the induction axiom:

$$\forall X^1(X0 \wedge \forall x(Xx \to X(\mathrm{S}x)) \to \forall y Xy).$$

Note that the second equality axiom implies

$$\begin{array}{l}\forall xy(x=y\to y=x);\\ \forall xyz(x=y\wedge x=z\to y=z);\\ \forall xy(x=y\to \mathrm{S}x=\mathrm{S}y).\end{array}$$

(For the first, take $Xz\equiv(z=x)$, then $x=y\wedge Xx\to Xy$ yields $x=y\wedge x=x\to y=x$, and since $x=x$ holds, $x=y\to y=x$; for the second take $Xx\equiv(x=z)$; for the third, take $Xz\equiv(\mathrm{S}x=\mathrm{S}z)$). ⊠

11.5.2. DEFINITION. A subsystem $\mathbf{HA}^{2*}$ based on $\mathbf{I}^2$ and equivalent to $\mathbf{HA}^2$ is the following. The language consists of $0,\mathrm{S},=$ as before, but we define a predicate $\mathbb{N}$ by

$$x\in\mathbb{N}\equiv\mathbb{N}x:=\forall X^1(X0\wedge\forall x(Xx\to X(\mathrm{S}x))\to Xx).$$

As axioms we include

$$\begin{array}{l}\forall x(x=x),\ \forall xyz(x=y\to(x=z\to y=z)),\\ \forall xy(\mathrm{S}x=\mathrm{S}y\to x=y),\ \forall xy(x=y\to\mathrm{S}x=\mathrm{S}y),\\ \forall x(\mathrm{S}x=0\to(\forall X^0)X).\end{array}$$

The *standard model* of $\mathbf{HA}^{2*}$ (and of $\mathbf{HA}^2$) has $\mathbb{N}$ and the powersets of $\mathbb{N}^k$ as domains of individuals and k-place relations respectively; 0, $\mathrm{S},=$ get their usual interpretation. ⊠

11.5.3. DEFINITION. $X^1\in\mathsf{Ext}:=\forall\vec{x}{\in}\mathbb{N}\forall\vec{y}{\in}\mathbb{N}(\vec{x}=\vec{y}\wedge X\vec{x}\to X\vec{y})$, where $\vec{x}\in\mathbb{N}$ stands for $x_1\in\mathbb{N},\ldots,x_n\in\mathbb{N}$ etc. ⊠

So "$X\in\mathsf{Ext}$" means that X behaves as an "extensional predicate" w.r.t. $\mathbb{N}$, or, perhaps more appropriately, satisfies *replacement* w.r.t. elements of $\mathbb{N}$. Note that $\mathbb{N}\in\mathsf{Ext}$.

11.5.4. DEFINITION. (*Interpretation of* $\mathbf{HA}^2$ *in* $\mathbf{HA}^{2*}$) Formulas of $\mathbf{HA}^2$ can now be interpreted in $\mathbf{HA}^{2*}$, by relativization of all individual variables to $\mathbb{N}$, and all relation variables to Ext; so $\forall x$, $\exists x$ go to $\forall x{\in}\mathbb{N}$, $\exists x{\in}\mathbb{N}$, and $\forall X^n,\exists X^n$ to $\forall X^n{\in}\mathsf{Ext},\exists X^n{\in}\mathsf{Ext}$; with respect to the other logical operators the embedding is a homomorphism. ⊠

¿From now on we take as a matter of convenience the language based on $\to,\wedge,\forall,\forall^2,\exists$; from the results in 11.1.4 we know that we have strong normalization w.r.t. detour contractions for this language.

11.5.4A. ♠ Verify that the embedding of the definition (call it ψ) indeed satisfies for all closed A of $\mathbf{HA}^2$: $\mathbf{HA}^2\vdash A\Rightarrow\mathbf{HA}^{2*}\vdash\psi(A)$.

11.5.5. DEFINITION. (*Normal deduction* $\mathcal{D}[n]$ of $S^n0 \in \mathbb{N}$ in $\mathbf{HA}^{2*}$) For $\mathcal{D}[0]$ we take:

$$\dfrac{\dfrac{\dfrac{0 \in X^{\,1}}{\forall y(y \in X \to \mathrm{S}y \in X) \to 0 \in X}\to\mathrm{I}}{0 \in X \to (\forall(y \in X \to \mathrm{S}y \in X) \to 0 \in X)}\to\mathrm{I},1}{0 \in \mathbb{N}}\forall^2\mathrm{I}$$

and for $\mathcal{D}[n]$ if $n > 0$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\forall y(y \in X \to \mathrm{S}y \in X)^1}{\mathrm{S}^{n-1}0 \in X \to \mathrm{S}^n0 \in X}\forall\mathrm{E} \qquad \begin{matrix}[0 \in X]^2 \\ \mathcal{D}_{n-1} \\ \mathrm{S}^{n-1}0 \in X\end{matrix}}{\mathrm{S}^n0 \in X}\to\mathrm{E}}{\forall y(y \in X \to \mathrm{S}y \in X) \to \mathrm{S}^n0 \in X}\to\mathrm{I},1}{0 \in X \to \forall y(y \in X \to \mathrm{S}y \in X) \to \mathrm{S}^n0 \in X}\to\mathrm{I},2}{\mathrm{S}^n0 \in \mathbb{N}}\forall^2\mathrm{I}$$

{standN}

where $\mathcal{D}_{n-1}$ is either $0 \in X$ (for $n = 1$) or is of the form

$$\dfrac{\dfrac{\forall y(y \in X \to \mathrm{S}y \in X)}{\mathrm{S}^{n-2}0 \in X \to \mathrm{S}^{n-1}0 \in X}\forall\mathrm{E} \qquad \begin{matrix}[0 \in X] \\ \mathcal{D}_{n-2} \\ \mathrm{S}^{n-2}0 \in X\end{matrix}}{\mathrm{S}^{n-1}0 \in X}$$

⊠

It is easy to see that $\mathcal{D}[n]$ translates under the collapsing map into $\bar{n}$ of $\boldsymbol{\lambda}\mathbf{2}$.

{*standNlemma*}

11.5.6. LEMMA. *$\mathcal{D}[n]$ is the unique normal deduction of* $\mathrm{S}^n0 \in \mathbb{N}$.

PROOF. Let $\mathcal{D}$ be a normal deduction of $\mathrm{S}^n0 \in \mathbb{N}$. $\mathcal{D}$ cannot end with an E-rule. For assume $\mathcal{D}$ to end with an E-rule; if we follow a main branch from the conclusion upwards, the branch must start in a purely first-order axiom, such as $\forall x(x = x)$, or in $\forall x(\mathrm{S}x = 0 \to (\forall X^0)X)$. ¿From a purely first-order axiom we can never arrive at the conclusion alone, passing through eliminations only. Elimination starting from $\forall x \neg \mathrm{S}x = 0$ must begin with

$$\dfrac{\dfrac{\forall x(\mathrm{S}x = 0 \to (\forall X)X)}{\mathrm{S}t = 0 \to (\forall X)X} \qquad \begin{matrix}\mathcal{D}' \\ \mathrm{S}t = 0\end{matrix}}{(\forall X)X}$$
$$\vdots$$

However, no open assumptions of $\mathcal{D}'$ can be discharged lower down; so $\mathcal{D}'$ must deduce $\mathrm{S}t = 0$ from the axioms. But in the standard model $\mathrm{S}t = 0$ is false (here we rely on the *consistency* of the system).

Therefore the deduction ends with an introduction, with subdeduction $\mathcal{D}_1$ of the premise $0 \in X \to (\forall y(y \in X \to \mathrm{S}y \in X) \to \mathrm{S}^n0 \in X)$. Again, the final step of $\mathcal{D}_1$ must be an introduction, and the premise of the conclusion of $\mathcal{D}_1$ is $\forall y(y \in X \to \mathrm{S}y \in X) \to \mathrm{S}^n0 \in X$, which is the conclusion of subdeduction $\mathcal{D}_2$ from assumptions $0 \in X$. $\mathcal{D}_2$ cannot end with an E-rule; for if it did, a main branch of $\mathcal{D}_2$ would have to terminate either in $0 \in X$ or in an axiom. The axioms are excluded as possibilities for the same reasons as before (assign $\mathbb{N}$ to the variable X in the "standard model" part of the argument). To $0 \in X$ no elimination rule is applicable. So the final step of $\mathcal{D}_2$ is an I-rule, and the immediate subdeduction $\mathcal{D}_3$ of $\mathcal{D}_2$ derives $\mathrm{S}^n0 \in X$ from assumptions $0 \in X$ and $\forall y(y \in X \to \mathrm{S}y \in X)$. The last rule of $\mathcal{D}_3$ must be an elimination rule, and the main branch must terminate in $0 \in X$ or in $\forall y(y \in X \to \mathrm{S}y \in X)$. If it terminates in $0 \in X$ and $n = 0$, we are done. Terminating in $0 \in X$ while $n \neq 0$ is impossible. So assume the main branch to start in $\forall y(y \in X \to \mathrm{S}y \in X)$:

$$\dfrac{\dfrac{\forall y(y \in X \to \mathrm{S}y \in X)}{t \in X \to \mathrm{S}t \in X} \qquad \begin{array}{c}\mathcal{D}''\\ t \in X\end{array}}{\mathrm{S}t \in X}$$
$$\vdots$$

but then $(\mathrm{S}t \in X) \equiv (\mathrm{S}^n0 \in X)$ etc. ⊠

11.5.7. DEFINITION. We define a *collapsing map* $[\![\ ]\!]$ from formulas and deductions of $\mathbf{HA}^{2*}$ to formulas and deductions of $\boldsymbol{\lambda 2\Omega}$. Let M be a fixed inhabited type of $\boldsymbol{\lambda 2\Omega}$, e.g. $M \equiv \forall X(X \to X)$, containing $T \equiv \Lambda X \lambda x.x^X$. To the relation variables X^n we let bijectively correspond propositional (type-) variables X^*. For formulas we take:

$$\begin{array}{lcl}
[\![t = s]\!] & := & M, \\
[\![Xt_1 \ldots t_n]\!] & := & X^*, \\
[\![A \to B]\!] & := & [\![A]\!] \to [\![B]\!], \\
[\![\forall x A]\!] & := & [\![A]\!], \\
[\![\exists x A]\!] & := & [\![A]\!], \\
[\![\forall X^n A]\!] & := & \forall X^* [\![A]\!].
\end{array}$$

Observe that

$$[\![A[x/t]]\!] \equiv [\![A]\!], \quad [\![t \in \mathbb{N}]\!] \equiv \mathrm{N}.$$

The definition is extended to deductions as follows. We have not before introduced a complete term calculus for $\mathbf{HA}^{2*}$, but the notations below will be self-explanatory. The definition proceeds by induction on construction of deduction terms, or what is the same, by induction on the length of deductions.

First the basis case, assumptions and axioms:

$$\begin{array}{lll}
x_i : A_i & \mapsto & x_i \colon [\![A_i]\!], \\
\forall x(x = x) & \mapsto & T \colon M, \\
\forall xyz(x = y \to (x = z \to y = z)) & \mapsto & \lambda x^M y^M . T^M \colon M \to M \to M, \\
\forall xy(\mathrm{S}x = \mathrm{S}y \to x = y) & \mapsto & \lambda y^M . T^M \colon M \to M, \\
\forall xy(x = y \to \mathrm{S}x = \mathrm{S}y) & \mapsto & \lambda y^M . T^M \colon M \to M, \\
\forall x(\mathrm{S}x = 0 \to \forall X.X) & \mapsto & \lambda z^M . \Omega \colon M \to (\forall X)X.
\end{array}$$

For the rules we put:

$$\begin{array}{lll}
\lambda x^A . t^B \colon A \to B & \mapsto & \lambda x^{[\![A]\!]} . [\![t]\!]^{[\![B]\!]} \colon [\![A \to B]\!], \\
t^{A \to B} s^A \colon B & \mapsto & [\![t]\!]^{[\![A \to B]\!]} [\![s]\!]^{[\![A]\!]} \colon [\![B]\!], \\
\lambda x^I . t \colon \forall x A & \mapsto & [\![t]\!] \colon [\![\forall x^I A]\!] (\text{note } [\![\forall x^I A]\!] \equiv [\![A[x/y]]\!]), \\
t^{\forall y^I A[x/y]} s^I \colon A[x/s] & \mapsto & [\![t]\!] \colon [\![A]\!] \ \ (\text{note } [\![A]\!] \equiv [\![\forall y A[x/y]]\!]), \\
\mathbf{p}(t_0, t_1^{A[x/t_0]}) \colon \exists x A & \mapsto & [\![t_1]\!] \colon [\![A]\!] \ \ (\text{note } [\![A[x/t_0]]\!] = [\![A]\!]), \\
\mathrm{E}^{\exists}_{y,z}(t^{\exists x A}, s(y, z^{A[x/y]})) \colon C & \mapsto & [\![s(y, z^{A[x/y]})]\!][z/[\![t]\!]^{[\![A]\!]}] \colon [\![C]\!].
\end{array}$$

For the next two cases observe that $[\![A[X/\lambda\vec{x}.C]]\!] \equiv [\![A]\!][X^*/[\![C]\!]]$.

$$\begin{array}{lll}
\Lambda X^n . t^A \colon \forall X^n A & \mapsto & \Lambda X^* . [\![t]\!]^{[\![A]\!]} \colon \forall X^* [\![A]\!], \\
t^{\forall X^n A}(\lambda \vec{x} . B) \colon A[X^n / \lambda \vec{x} . B] & \mapsto & [\![t]\!]^{\forall X^* [\![A]\!]} [\![B]\!] \colon [\![A]\!][X^* / [\![B]\!]].
\end{array}$$

For the $\wedge$-rules we need on the right hand side defined operators:

$$\begin{array}{l}
\mathbf{p}^{A,B} := \Lambda X . \lambda x^A y^B z^{A \to (B \to X)} . zxy, \\
\mathbf{p}_0^{A,B} := \lambda u^{A \wedge B} . uA(\lambda x^A y^B . x), \ \ \mathbf{p}_1^{A,B} := \lambda u^{A \wedge B} . uB(\lambda x^A y^B . y).
\end{array}$$

Then

$$\mathbf{p}(t^A, s^B) \colon A \wedge B \ \ \mapsto \ \ \mathbf{p}([\![t]\!]^{[\![A]\!]}, [\![s]\!]^{[\![B]\!]}) \colon [\![A \wedge B]\!],$$

etc. ⊠

PROPOSITION. *The collapsing map preserves reductions: if $\mathcal{D}$ reduces to $\mathcal{D}'$, then the collapse of $\mathcal{D}$ reduces to the collapse of $\mathcal{D}'$. Moreover, normal forms are mapped to normal forms.* ⊠

11.5.7A. ♠ Verify this.

{*hasrec*}

11.5.8. THEOREM. *The provably total recursive functions of* $\mathbf{HA}^2$ *are representable in* $\boldsymbol{\lambda}\mathbf{2}$.

PROOF. Let $\mathcal{D}$ be a deduction in $\mathbf{HA}^{2*}$, with conclusion,

(1) $\quad \vdash \forall x{\in}\mathbb{N}\, \exists y{\in}\mathbb{N} A(x,y)$,

or expanded

$$\vdash \forall x(x \in \mathbb{N} \to \exists y(y \in \mathbb{N} \wedge A(x,y)).$$

The collapsing map applied to $\mathcal{D}$ produces a term t such that

$$t\colon [\![\forall x(x \in \mathbb{N} \to \exists y(y \in \mathbb{N} \wedge A(x,y)))]\!] \equiv \mathrm{N} \to (\mathrm{N} \wedge [\![A]\!]).$$

Then $t^*\colon \lambda x^{\mathrm{N}}.\mathbf{p}_0(tx)$ encodes the recursive function f implicit in the proof of (1). This is seen as follows. If we specialize the deduction $\mathcal{D}$ to $x = \mathrm{S}^n 0$ we obtain a deduction $\mathcal{D}_n$

$$\dfrac{\begin{matrix}\mathcal{D}[n] \\ \mathrm{S}^n 0 \in \mathbb{N}\end{matrix} \qquad \dfrac{\begin{matrix}\mathcal{D}'_n \\ \forall x(x \in \mathbb{N} \to \exists y(y \in \mathbb{N} \wedge A(x,y)))\end{matrix}}{\mathrm{S}^n 0 \in \mathbb{N} \to \exists y(y \in \mathbb{N} \wedge A(\mathrm{S}^n 0, y))}}{\exists y(y \in \mathbb{N} \wedge A(\mathrm{S}^n 0, y))}$$

which after normalization must end with an introduction:

$$\dfrac{\begin{matrix}\mathcal{D}''_n \\ \mathrm{S}^m 0 \in \mathbb{N} \wedge A(\mathrm{S}^m 0, \mathrm{S}^n 0)\end{matrix}}{\exists y(y \in \mathbb{N} \wedge A(\mathrm{S}^n 0, y))}$$

Applying $\wedge$E to $\mathcal{D}''_n$ we find a deduction of $\mathrm{S}^m 0 \in \mathbb{N}$; the collapsing map produces a term which must be equal to $t^*\bar{n}$. By lemma 11.5.6, the deduction normalizes to $\mathcal{D}[m]$, so $t^*\bar{n}$ normalizes to $\bar{m}$. Combining this with the embedding of $\boldsymbol{\lambda}\mathbf{2}\boldsymbol{\Omega}$ into $\boldsymbol{\lambda}\mathbf{2}$, we have the desired result. Since $A(\mathrm{S}^n 0, \mathrm{S}^m 0)$ is provable in $\mathbf{HA}^2$, it is true in the standard model, so $m = f(n)$. ⊠

REMARK. As observed in 11.6.4, the $\boldsymbol{\lambda}\mathbf{2}$-representable functions are in fact *exactly* the functions provably total recursive in $\mathbf{HA}^2$.

11.5.9. *The provably total recursive functions* $\mathbf{PA}^2$

The provably total recursive functions of classical second-order arithmetic $\mathbf{PA}^2$ are in fact also provably total in $\mathbf{HA}^2$; $\mathbf{PA}^2$ is just $\mathbf{HA}^2$ with classical logic. This fact may be proved directly, but can also be obtained from the characterization of the provably recursive functions of $\mathbf{HA}^2$. In outline, the proof is as follows.

(a) In $\mathbf{HA}^2$ and $\mathbf{PA}^2$ we can conservatively extend the language and the system by adding symbols for all the primitive recursive functions. Then a provably total recursive function is given by a code number n such that $\vdash \forall x \exists y(\chi_T(\bar{n}, x, y) = 0)$, where χ_T is the characteristic function of Kleene's T-predicate.

(b) The Gödel-Gentzen negative translation $^{\mathrm{g}}$ (cf. 2.3) embeds $\mathbf{PA}^2$ into $\mathbf{HA}^2$. Hence, if $\mathbf{PA}^2 \vdash \exists y(\chi_T(\bar{n}, x, y) = 0)$, then $\mathbf{HA}^2 \vdash \neg\neg\exists y(\chi_T(\bar{n}, x, y) = 0)$ (using $\neg\forall x \neg A \leftrightarrow \neg\neg\exists x A$).

(c) By a method due to [**?**] and [**?**] we can show that $\mathbf{HA}^2$ is closed under "Markov's rule" in the form: "if $\vdash \neg\neg\exists x(t(x, \vec{y}) = 0)$ then $\vdash \exists x(t(x, \vec{y}) = 0)$" (an exposition of this result is found in [**?**, section 5.1]).

As to (a), we use the result on the conservativity of the addition for symbols for definable functions, applied to functions defined by primitive recursion (4.3.13). Therefore we need to show that in $\mathbf{HA}^2$ we can define graph predicates $H(\vec{x}, z)$ for each primitive recursive function h. The graph PRD for the predecessor function prd is given by

$$\mathrm{PRD}(x, z) := (x = 0 \wedge z = 0) \vee \exists y(x = \mathrm{S}y \wedge z = y)$$

The crucial step in the construction of these H goes as follows. If h is obtained from f and the number m (for notational simplicity we do not consider additional numerical parameters):

$$h(0) = m, \quad h(\mathrm{S}z) = f(z, h(z))$$

and we have already constructed F as the graph of f, we obtain the graph H as

$$H(z, u) := \forall X^2(A(z, X) \to X(z, u)),$$

where

$$\begin{aligned} &A(z, X^2) := \forall z' \leq z([z' = 0 \to (Xz'v \leftrightarrow v = m)] \wedge \\ &[z' \neq 0 \to (Xz'v) \leftrightarrow \exists v'z''(\mathrm{PRD}(z', z'') \wedge X(z'', v') \wedge F(z'', v', v))]). \end{aligned}$$

Then one proves by induction on z

$$\exists X^2 A(z, X), \text{ and } \forall z' \leq z \exists! u(A(z, X) \to X(z', u)).$$

11.6 Notes

For a general introduction to higher-order logic, see [**?**].