CHAPTER 18

Normalisation

Closed terms of type 0 in the language of HA^ω can be rather complicated. Still, the idea is that they are just numbers. So a natural question is: can one find for any closed term of type 0 a natural number n such that $\mathsf{HA}^\omega \vdash t = S^n 0$? The answer is yes, and we will prove this in the first section. We explore its proof-theoretic consequences in the second section.

1. Normalisation

If one looks at the axioms for the combinators in HA^ω, it is quite natural to read them in a directed way. For example, the equation $\mathbf{k}xy = x$ is naturally read as saying that: one can rewrite $\mathbf{k}xy$ to x (rather than saying that one can always replace x by $\mathbf{k}xy$). In general, if t is an expression on the left of the following table and t' is an expression in the same row on the right, then the natural thing is to rewrite t as t', and not vice versa.

t	t'
$\mathbf{k}t_1t_2$	t_1
$\mathbf{s}t_1t_2t_3$	$t_1t_3(t_2t_3)$
$\mathbf{p}_i(\mathbf{p}t_0t_1)$	t_i
$\mathbf{p}(\mathbf{p}_0t_1)(\mathbf{p}_1t_1)$	t_1
$\mathbf{R}t_1t_20$	t_1
$\mathbf{R}t_1t_2(St_3)$	$t_2t_3(\mathbf{R}t_1t_2t_3)$

What we explore in this section is what happens if one starts from any term in the language of HA^ω and one starts rewriting it using the axioms for the combinators only in the "natural direction".

Warning: From now on we will only look at *closed* terms. So in the sequel of this section term will mean closed term.

Definition 1.1. An expression on the left of the above table is called a *redex*. If t is redex and t' is the corresponding expression on the right of the table, then we will say that t *converts to* t' and we will write $t \text{ conv } t'$.

Definition 1.2. The *reduction relation* $\succeq$ is inductively defined by:

$$\begin{aligned}
&t \succeq t\\
&t \text{ conv } t' \Rightarrow t \succeq t'\\
&t \succeq t' \Rightarrow t''t \succeq t''t'\\
&t \succeq t' \Rightarrow tt'' \succeq t't''\\
&t \succeq t', t' \succeq t'' \Rightarrow t \succeq t''
\end{aligned}$$

If $t \succeq t'$ we shall say that t *reduces* to t'. We write $t \succ_1 t'$ if t' is obtained from t by converting a single redex in t. Observe that $\succeq$ is the transitive and reflexive closure of $\succ_1$.

Proposition 1.3. *If* $t \succeq t'$*, then* $\mathsf{HA}^\omega \vdash t = t'$.

Proof. This is immediate from the inductive definition of $t \succeq t'$. □

Definition 1.4. A term t is in *normal form*, if t does not contain a redex.

Proposition 1.5. *An expression is in normal form iff it has one of the following forms:*

$$0, St_1, \mathbf{k}, \mathbf{k}t_1, \mathbf{s}, \mathbf{s}t_1, \mathbf{s}t_1t_2,$$
$$\mathbf{R}, \mathbf{R}t_1, \mathbf{R}t_1t_2, \mathbf{R}t_1t_2\hat{t}t_3\dots t_n$$
$$\mathbf{p}_i, \mathbf{p}_it_0\dots t_n, \mathbf{p}, \mathbf{p}t_1, \mathbf{p}t't''$$

where:

- $t_1, \dots, t_n$ *are in normal form,*
- $\hat{t}$ *is in normal form and not of the form 0 or* Ss *for some term* s,
- t_0 *is in normal form and not of the form* $\mathbf{p}s_1s_2$, *and*
- t' *and* t'' *are in normal form and not simultaneously of the form* $\mathbf{p}_0s$ *and* $\mathbf{p}_1s$.

Proof. Easy. □

Proposition 1.6.
(i) *If t is a closed term in normal form of type 0, then t is a numeral (that is, an expression of the form S^n0 for some n).*
(ii) *If t is a closed term in normal form of type $\sigma \times \tau$, then t is of the form $\mathbf{p}t_1t_2$ for suitable t_1, t_2.*

Proof. Most of the expressions that were listed in the previous proposition necessarily have type $\sigma \to \tau$ for certain σ and τ. Those which do not (necessarily) have this type are:

$$0, St_1, \mathbf{R}t_1t_2\hat{t}t_3\dots t_n, \mathbf{p}_it_0\dots t_n, \mathbf{p}t't''.$$

Now one shows (i) and (ii) by simultaneous induction on the length of t. (Incidentally, this shows that terms $\hat{t}$ and t_0 as in the previous proposition do not really exist!) □

In view of Proposition 1.3 and Proposition 1.6 it now suffices to show that for any term t of type 0 there is a normal form t' such that $t \succeq t'$. In other words, we want to show that terms of type 0 are *normalisable.*

Definition 1.7. A term t is *normalisable* if there is a term t' in normal form such that $t \succeq t'$.

In order to show this we use a *computability predicate.* This method was first employed by Tait and we will do the same here.

Definition 1.8. The *computable terms* are defined by induction on type structure as follows:

(1) A term t of type 0 is computable if it is normalisable.
(2) A term t of type $\sigma \to \tau$ is computable if for any computable term t' of type σ the term tt' is computable as well.
(3) A term t of type $\sigma \times \tau$ is computable if both $\mathbf{p}_0t$ and $\mathbf{p}_1t$ are computable.

Lemma 1.9.
(i) *If s and t are computable, then so is st.*
(ii) *If $s \succeq t$ and t is computable, then s is computable as well.*

Proof. (i): If st is well-defined, then s is of type $\sigma \to \tau$ and t is of type σ for certain σ and τ. So (i) follows from what it means for an expression of arrow type to be computable.

(ii) is proved by induction on the type ρ of both s and t.

(a) If $\rho = 0$, computability means normalisability. So if $t \succeq t'$ with t' in normal form, then also $s \succeq t'$ by transitivity of $\succeq$. Hence s is normalisable as well.
(b) $\rho = \sigma \to \tau$: Let t' be a term of type σ. Our task is to show that st' is computable. But we have $st' \succeq tt'$ and that tt' is computable, so st' is computable by induction hypothesis.
(c) $\rho = \sigma \times \tau$: We have to show that $\mathbf{p}0s$ and $\mathbf{p}_1 s$ are computable. But $\mathbf{p}_0 s \succeq \mathbf{p}_0 t$ and $\mathbf{p}_1 s \succeq \mathbf{p}_1 t$, so this follows from the induction hypothesis.

□

THEOREM 1.10. *All the closed terms in* HA^ω *are computable. In particular, closed terms of type 0 are normalisable.*

PROOF. We will show that t is computable by induction on the structure of t.

(i) 0 is in normal form, hence obviously computable.
(ii) S is computable by Proposition 1.6.
(iii) To show that $\mathbf{k}$ is computable we need to show that if t_1 and t_2 are computable, then $\mathbf{k}t_1t_2$ is computable. But $\mathbf{k}t_1t_2 \succeq t_1$, so this follows from part (ii) of the previous lemma.
(iv) We leave the combinators $\mathbf{s}, \mathbf{p}_0, \mathbf{p}_1$ and $\mathbf{p}$ to the reader.
(v) For the recursor $\mathbf{R}$ we need to show that for computable t_1, t_2, t_3 the expression $\mathbf{R}t_1t_2t_3$ is computable as well. t_3 is of type 0, so from Proposition 1.6 we get that $t_3 \succeq S^n0$ for some n. So we prove the statement by induction on n:
  (a) If $n = 0$, then
  $$\mathbf{R}t_1t_2t_3 \succeq \mathbf{R}t_1t_20 \succeq t_1,$$
  so the desired statement follows from part (ii) of the previous lemma.
  (b) If $\mathbf{R}t_1t_2(S^n0)$ is computable, then so is
  $$\mathbf{R}t_1t_2t_3 \succeq \mathbf{R}t_1t_2(S^{n+1}0) \succeq t_2(S^n0)(\mathbf{R}t_1t_2(S^n0))$$
  by the previous lemma.
(vi) Induction step: We need to show that if t_1 and t_2 are computable, then so is t_1t_2. That was part (i) of the previous lemma.

□

REMARK 1.11. The previous theorem is just the tip of an iceberg. For instance, one can show that *all* terms (not just of type 0) are normalisable. Moreover, one can show that normal forms are unique: so if $t \succeq t_1$ and $t \succeq t_2$ and both t_1 and t_2 are in normal form, then t_1 and t_2 are identical expressions. One can even show that any reduction sequence $t_1 \succ_1 t_2 \succ_1 t_3 \succ_1 t_4 \succ_1 \ldots$ must necessarily terminate and therefore end with an expression in normal form (this is called "strong normalisation"). We will not prove these things here.

2. Proof-theoretic consequences

We will now explore the consequences of the Theorem 1.10 for the proof theory of HA^ω and PA^ω.

COROLLARY 2.1. *For any closed term t of type 0 there is a natural number n such that* $\mathsf{HA}^\omega \vdash t = S^n0$.

Proof. Let t be a closed term of type 0. By the previous theorem there is a closed term t' in normal form such that $t \succeq t'$. Proposition 1.6 tells us that t' is of the form $S^n 0$ for some n and Proposition 1.3 tells us that $\mathsf{HA}^\omega \vdash t = t'$. □

Corollary 2.2. (Numerical existence property for HA^ω) *If a sentence of the form $\exists x^0\, \varphi(x)$ is provable in HA^ω, then there is a numeral $S^n 0$ such that $\varphi(S^n 0)$ is provable in HA^ω as well. If φ is simple, the same holds for PA^ω.*

Proof. Follows from the term extraction theorem in combination with the previous corollary. □

Corollary 2.3. (Disjunction property for HA^ω) *If a sentence of the form $\varphi \vee \psi$ is provable in HA^ω, then either φ or ψ is provable in HA^ω.*

Proof. Remember that we treat $\varphi \vee \psi$ as an abbreviation of

$$\exists n^0 \big((n = 0 \to \varphi) \wedge (n \neq 0 \to \psi) \big).$$

So this follows from the previous corollary. □