A transitive structure $M = \langle |M|, \in \vec{A}\rangle$ is called *strongly admissible* iff, in addition to the Kripke–Platek axioms, it satisfies the Σ_1 *axiom of subsets*:

$$x \cap \{z | \varphi(z)\} \text{ is a set (for } \Sigma_1 \text{ formulae } \varphi).$$

Kripke defines the *projectum* δ_α of an admissible ordinal α to be the least δ such that $A \cap \delta \notin L_\alpha$ for some $\underline{\Sigma}_1(L_\alpha)$ set A. He shows that $\delta_\alpha = \alpha$ iff α is strongly admissible. He calls α *projectible* iff $\delta_\alpha < \alpha$. There are many projectible admissibles — e.g. $\delta_\alpha = \omega$ if α is the least admissible greater than ω. He shows that for every admissible α there is a $\underline{\Sigma}_1(L_\alpha)$ injection f_α of L_α into δ_α.

The definition of projectum of course makes sense for *any* $\alpha \geq \omega$. By refinements of Kripke's methods it can be shown that f_α exists for every $\alpha \geq \omega$ and that $\delta_\alpha < \alpha$ whenever $\alpha \geq \omega$ is not strongly admissible. We shall — essentially — prove these facts in chapter 2 (except that, for technical reasons, we shall employ a modified version of the constructible hierarchy).

1.2 Primitive Recursive Set Functions

1.2.1 PR Functions

The *primitive recursive set functions* comprise a collection of functions

$$f : V^n \to V$$

which form a natural analogue of the primitive recursive number functions in ordinary recursion theory. As with admissibility theory, their discovery arose from the attempt to generalize ordinary recursion theory. These functions are ubiquitous in set theory and have very attractive absoluteness properties. In this section we give an account of these functions and their connection with admissibility theory, though — just as in §1 — we shall suppress some proofs.

Definition 1.2.1. $f : V^n \to V$ is a *primitive recursive (pr) function* iff it is generated by successive application of the following schemata:

(i) $f(\vec{x}) = x_i$ (here $\vec{x}$ is $x_1, \ldots, x_n$)

(ii) $f(\vec{x}) = \{x_i, x_j\}$

(iii) $f(\vec{x}) = x_i \setminus x_j$

(iv) $f(\vec{x}) = g(h_1(\vec{x}), \ldots, h_m(\vec{x}))$

(v) $f(y, \vec{x}) = \bigcup_{z \in y} g(z, \vec{x})$

(vi) $f(y, \vec{x}) = g(y, \vec{x}, \langle f(z, \vec{x}) | z \in y \rangle)$

We also define:

Definition 1.2.2. $R \subset V^n$ is a *primitive recursive relation* iff there is a primitive recursive function r such that $R = \{\langle \vec{x} \rangle | r(\vec{x}) \neq \emptyset\}$.

(**Note** It is possible for a function on V to be primitive recursive as a relation but not as a function!)

We begin by developing some elementary consequences of these definitions:

Lemma 1.2.1. *If $f : V^n \to V$ is primitive recursive and $k : n \to m$, then g is primitive recursive, where*

$$g(x_0, \ldots, x_{m-1}) = f(x_{k(0)}, \ldots, x_{k(n-1)}).$$

Proof. By (i), (iv).

Lemma 1.2.2. *The following functions are primitive recursive*

(a) $f(\vec{x}) = \bigcup x_j$

(b) $f(\vec{x}) = x_i \cup x_j$

(c) $f(\vec{x}) = \{\vec{x}\}$

(d) $f(\vec{x}) = n$, *where* $n < \omega$

(e) $f(\vec{x}) = \langle \vec{x} \rangle$

Proof.

(a) By (i), (v), Lemma 1.2.1, since $\bigcup x_j = \bigcup_{z \in x_j} z$

(b) $x_i \cup x_j = \bigcup \{x_i, x_j\}$

(c) $\{\vec{x}\} = \{x_1\} \cup \ldots \cup \{x_m\}$

(d) By in induction on n, since $0 = x \setminus x, n + 1 = n \cup \{n\}$

(e) The proof depends on the precise definition of n–tuple. We could for instance define $\langle x, y \rangle = \{\{x\}, \{x, y\}\}$ and $\langle x_1, \ldots, x_n \rangle = \langle x_1, \langle x_2, \ldots, x_n \rangle \rangle$ for $n > 2$.

If, on the other hand, we wanted each tuple to have a unique length, we could call the above defined ordered pair (x, y) and define:

$$\langle x_1, \ldots, x_n \rangle = \{(x_1, 0), \ldots, (x_n, n-1)\}.$$

QED (Lemma 1.2.2)

Lemma 1.2.3. *(a)* $\notin$ *is pr*

(b) If $f : V^n \to V, R \subset V^n$ *are primitive recursive, then so is*

$$g(\vec{x}) = \begin{cases} f(\vec{x}) \text{ if } R\vec{x} \\ \emptyset \text{ if not} \end{cases}$$

(c) $R \subset V^n$ *is primitive recursive iff its characteristic functions* χ_R *is a primitive recursive function*

(d) If $R \subset V^n$ *is primitive recursive so is* $\neg R =: V^n \setminus R$

(e) Let $f_i : V^n \to V, R_i \subset V^n$ *be* $pr(i = 1, \ldots, m)$ *where* $R_1, \ldots, R_m$ *are mutually disjoint and* $\bigcup_{i=1}^{m} R_i = V^n$*. Then* f *is pr where:*

$$f(\vec{x}) = f_i(x) \text{ when } R_i\vec{x}.$$

(f) If $Rz\vec{x}$ *is primitive recursive, so is the function*

$$f(y, \vec{x}) = y \cap \{z | Rz\vec{x}\}$$

(g) If $Rz\vec{x}$ *is primitive recursive so is* $\bigvee z \in yRz\vec{x}$

(h) If $R_i\vec{x}$ *is primitive recursive* $(i = 1, \ldots, m)$*, then so is* $\bigvee_{i=1}^{m} R_i\vec{x}$

(i) If $R_1, \ldots, R_n$ *are primitive recursive relations and* φ *is a* Σ_0 *formula, then* $\{\langle\vec{x}\rangle | \langle V, R_1, \ldots, R_n\rangle \models \varphi[\vec{x}]\}$ *is primitive recursive.*

(j) If $f(z, \vec{x})$ *is primitive recursive, then so are:*

$$g(y, \vec{x}) = \{f(z, \vec{x}) : z \in y\}$$
$$g'(y, \vec{x}) = \langle f(z, \vec{x}) : z \in y\rangle$$

(k) If $R(z, \vec{x})$ *is primitive recursive, then so is*

$$f(y, \vec{x}) = \begin{cases} \text{That } z \in y \text{ such that } Rz\vec{x} \text{ if exactly} \\ \text{one such } z \in y \text{ exists;} \\ \emptyset \text{ if not.} \end{cases}$$

Proof.

(a) $x \notin y \leftrightarrow \{x\} \setminus y \neq \emptyset$

(b) Let $R\vec{x} \leftrightarrow r(\vec{x}) \neq \emptyset$. Then $g(\vec{x}) = \bigcup_{z \in r(\vec{x})} f(\vec{x})$.

(c) $\chi_r(\vec{x}) = \begin{cases} 1 \text{ if } R\vec{x} \\ 0 \text{ if not} \end{cases}$

(d) $\chi_{\neg R}(\vec{x}) = 1 \setminus \chi_R(\vec{x})$

(e) Let $f_i'(\vec{x}) = \begin{cases} f_i(\vec{x}) \text{ if } R_i\vec{x} \\ \emptyset \text{ if not} \end{cases}$
Then $f(\vec{x}) = f_i'(\vec{x}) \cup \ldots \cup f_m'(\vec{x})$.

(f) $f(y, \vec{x}) = \bigcup_{z \in y} h(z, \vec{x})$, where:

$$h(z, \vec{x}) = \begin{cases} \{z\} \text{ if } Rz\vec{x} \\ \emptyset \text{ if not} \end{cases}$$

(g) Let $Py\vec{x} \leftrightarrow: \bigvee z \in yRz\vec{x}$. Then $\chi_P(\vec{x}) = \bigcup_{z \in y} \chi_R(z, \vec{x})$.

(h) Let $P\vec{x} \leftrightarrow \bigvee_{i=1}^{m} R_i\vec{x}$. Then

$$X_P(\vec{x}) = X_{R_1} \cup \ldots \cup X_{R_n}(\vec{x}).$$

(i) is immediate by (d), (g), (h)

(j) $g(y, \vec{x}) = \bigcup_{z \in y} \{f(z, \vec{x})\}, g'(y, \vec{x}) = \bigcup_{z \in y} \{\langle f(z, \vec{x}), z \rangle\}$

(k) $R'zu\vec{x} \leftrightarrow: (z \in u \wedge Rz\vec{x} \wedge \bigwedge z' \in u(z \neq z' \to \neg Rz'\vec{x}))$ is primitive recursive by (i). But then:

$$f(y, \vec{x}) = \bigcup (y \cap \{z | R'zy\vec{x}\})$$

QED (Lemma 1.2.3)

Lemma 1.2.4. *Each of the functions listed in §1 Lemma 1.1.12 is primitive recursive.*

The proof is left to the reader.

Note Up until now we have only made use of the schemata (i) – (v). This will be important later. The functions and relations obtainable from (i) – (v) alone are called *rudimentary* and will play a significant role in fine structure theory. We shall use the fact that Lemmas 1.2.1 – 1.2.3 hold with "rudimentary" in place of "primitive recursive".

Using the recursion schema (vi) we then get:

Lemma 1.2.5. *The functions $TC(x), rn(x)$ are primitive recursive.*

The proof is the same as before (§1 Corollary 1.1.14).

Definition 1.2.3. $f : \mathrm{On}^n \times V^m \to V$ is primitive recursive iff f' is primitive recursive, where

$$f'(\vec{y}, \vec{x}) = \begin{cases} f(\vec{y}, \vec{x}) \text{ if } y_1, \ldots, y_n \in \mathrm{On} \\ \emptyset \text{ if not} \end{cases}$$

As before:

Lemma 1.2.6. *The ordinal function $\alpha + 1, \alpha + \beta, \alpha \cdot \beta, \alpha^\beta, \ldots$ are primitive recursive.*

Definition 1.2.4. Let $f : V^{n+1} \to V$.

$f^\alpha (\alpha \in \mathrm{On})$ is defined by:

$$\begin{aligned} &f^0(y, \vec{x}) = y \\ &f^{\alpha+1}(y, \vec{x}) = f(f^\alpha(y, \vec{x}), \vec{x}) \\ &f^\lambda(y, \vec{x}) = \bigcup_{r<\lambda} f^r(y, \vec{x}) \text{ for limit } \lambda. \end{aligned}$$

Then:

Lemma 1.2.7. *If f is primitive recursive, so is $g(\alpha, y, \vec{x}) = f^\alpha(y, \vec{x})$.*

There is a strengthening of the recursion schema (vi) which is analogous to §1 Lemma 1.1.16. We first define:

Definition 1.2.5. Let $h : V \to V$ be primitive recursive. h is *manageable* iff there is a primitive recursive $\sigma : V \to \mathrm{On}$ such that

$$x \in h(y) \to \sigma(x) < \sigma(y).$$

(Hence the relation $x \in h(y)$ is well founded.)

Lemma 1.2.8. *Let h be manageable. Let $g : V^{n+2} \to V$ be primitive recursive. Then $f : V^{n+1} \to V$ is primitive recursive, where:*

$$f(y, \vec{x}) = g(y, \vec{x}, \langle f(z, \vec{x}) | z \in h(y) \rangle).$$

Proof. Let σ be as in the above definition. Let $|x| = \mathrm{lub}\{|y| | y \in h(x)\}$ be the rank of x in the relation $y \in h(x)$. Then $|x| \leq \sigma(x)$. Set:

$$\Theta(z, \vec{x}, u) = \bigcup \{ \langle g(y, \vec{x}, z \restriction h(y)), y \rangle | y \in u \wedge h(y) \subset \mathrm{dom}(z) \}.$$

By induction on α, if u is h–closed (i.e. $x \in u \to h(x) \subset u$), then:

$$\Theta^\alpha(\emptyset, \vec{x}, u) = \langle f(y, \vec{x}) | y \in u \wedge |y| < \alpha \rangle$$

Set $\tilde{h}(v) = v \cup \bigcup_{z \in v} h(z)$. Then $\tilde{h}^\alpha(\{y\})$ is h–closed for $\alpha \geq |y|$. Hence:

$$f(y, \vec{x}) = \Theta^{\sigma(y)+1}(\emptyset, \vec{x}, \tilde{h}^{\sigma(y)}(\{y\}))(y).$$

QED (Lemma 1.2.8)

Corresponding to §1 Lemma 1.1.17 we have:

Lemma 1.2.9. *Let $u \in H_\omega$. The constant function $f(x) = u$ is primitive recursive.*

Proof: By $\in$–induction on u. QED

As we shall see, the constant function $f(x) = \omega$ is not primitive recursive, so the analogue of §1 Lemma 1.1.18 fails. We say that f is primitive recursive *in the parameters* $p_1, \ldots, p_m H$:

$$f(\vec{x}) = g(\vec{x}, \vec{p}), \text{ where } g \text{ is primitive recursive.}$$

In place of §1 Lemma 1.1.19 we get:

Lemma 1.2.10. *The class* Fin *and the function $f(x) = \mathbb{P}_\omega(x)$ are primitive recursive in the parameter ω.*

Proof: Let f be primitive recursive such that $f(0, x) = \{\emptyset\} \cup \{\{z\} | z \in x\}$, $f(n + 1, x) = \{u \cup v | \langle u, v \rangle \in f(n, x)^2\}$. Then $\mathbb{P}_\omega(x) = \bigcup_{n \in \omega} f(n, x)$. But then:

$$x \in \text{Fin} \leftrightarrow \bigvee n \in \omega \bigvee g \in \bigcup_{n<\omega} \mathbb{P}^n_\omega(x \times \omega) g : n \leftrightarrow x.$$

QED

Corollary 1.2.11. *The constant function $f(x) = H_\omega$ is primitive recursive in the parameter ω.*

Proof: $H_\omega = \bigcup_{n<\omega} \mathbb{P}^n_\omega(\emptyset)$. QED

Corresponding to Lemma 1.1.21 of §1 we have:

Lemma 1.2.12. *The function* Def(u) *is primitive recursive in the parameter ω.*

The proof involves carrying out the proof of §1 Lemma 1.1.21 (which we also omitted) while ensuring that the relevant classes and functions are primitive recursive. We give not further details here (though filling in the details can be an arduous task). A fuller account can be found in [PR] or [AS].

Hence:

Corollary 1.2.13. *The function $f(\alpha) = L_\alpha$ is primitive recursive in ω.*

Similarly:

Lemma 1.2.14. *The function $f(\alpha, x) = L_\alpha(x)$ is primitive recursive in ω.*

Lemma 1.2.15. *Let $A \subset V$ be primitive recursive in the parameter p. Then $f(\alpha) = L_\alpha^A$ is primitive recursive in p.*

One can generalize the notion primitive recursive to *primitive recursive in the class $A \subset V$* (or *in the classes $A_1, \ldots, A_n \subset V$*).

We define:

Definition 1.2.6. Let $A_1, \ldots, A_n \subset V$. The function $f : V^n \to V$ is *primitive recursive in* $A_1, \ldots, A_n$ iff it is obtained by successive applications of the schemata (i) – (vi) together with the schemata:

$$f(x) = \chi_{A_i}(x)(i = 1, \ldots, n).$$

A relation R is primitive recursive in $A_1, \ldots, A_n$ iff

$$R = \{\langle \vec{x} \rangle | f(\vec{x}) \neq 0\}$$

for a function f which is primitive recursive in $A_1, \ldots, A_n$.

It is obvious that all of the previous results hold with "primitive recursive in $A_1, \ldots, A_n$" in place of "primitive recursive".

By induction on the defining schemata of f we can show:

Lemma 1.2.16. *Let f be primitive recursive in $A_1, \ldots, A_n$, where each A_i is primitive recursive in $B_1, \ldots, B_m$. Then f is primitive recursive in $B_1, \ldots, B_m$.*

The proof is by induction on the defining schemata leading from $A_1, \ldots, A_n$ to f. The details are left to the reader. It is clear, however, that this proof is *uniform* in the sense that the schemata which give in f from $B_1, \ldots, B_m$ are not dependent on $B_1, \ldots, B_m$ or $A_1, \ldots, A_n$, but only on the schemata which lead from $A_1, \ldots, A_n$ to f and the schemata which led from $B_1, \ldots, B_m$ to $A_i(i = 1, \ldots, n)$.

This will be made more precise in §1.2.2

1.2.2 PR Definitions

Since primitive recursive functions are proper classes, the foregoing discussion must ostensibly be carried out in second order set theory. However, we can translate it into ZF by talking about *primitive recursive definitions*. By a primitive recursive definition we mean a finite sequence of equations of the form (i) – (vi) such that:

- The function variable on the left side does not occur in a previous equation in the sequence
- every function variable on the right side occurs previously on the left side with the same number of argument places.

We assume that the language in which we write these equation has been *arithmetized* — i.e. formulae, terms, variables etc. have been identified in a natural way with elements of ω (or at least H_ω).

Every primitive recursive definition s defines a function F_s. If $s = \langle s_0, \ldots, s_{n-1}\rangle$, then $F_s = F_s^{n-1}$, where F_s^i interprets the leftmost function variable of s_i. This is defined in a straightforward way. If e.g. s_i is "$f(y, \vec{x}) = \bigcup\limits_{z \in y} g(z, \vec{x})$" and g was leftmost in s_j, then we get

$$F^i(y, \vec{x}) = \bigcup_{z \in y} F^j(z, \vec{x}).$$

Let PD be the class of primitive recursive definitions. In order to define $\{\langle x, s\rangle | s \in PD \wedge x \in F_s\}$ in ZF we proceed as follows:

Let $s = \langle s_0, \ldots, s_{n-1}\rangle \in PD$. Let M be any admissible structure. By induction we can then define $\langle F_s^{i,M} | i < n\rangle$ where F_s^i a function on M^{n_i} (n_i being the number of argument places). By admissibility we know that F_s^i exists and is defined on all of M^{n_i}. We then set: $F_s^M = F_s^{n-1,M}$. This defines the set $\langle F_s^M | s \in PD\rangle$. If $M \subseteq M'$ and M' is also admissible, it follows by any induction on $i < n$ that $F^{i,M} = F^{i,M'} \upharpoonright M$. Hence $F_s^M \subset F_s^{M'}$. We can then set:

$$F_s = \bigcup\{F_s^M | M \text{ is admissible}\}.$$

Note that by §1, each F_s^M has a uniform Σ_1 definition φ_s which defines F_s^M over *every* admissible M. It follows that φ_s defines F_s in V. Thus we have won an important absoluteness result: Every primitive recursive function has a Σ_1 definition which is absolute in all inner models, in all generic extensions of V, and indeed, in all admissible structures $M = \langle |M|, \in \rangle$. This absoluteness phenomenon is perhaps the main reason for using the theory of primitive recursive functions in set theory. Carol Karp was the first to notice the phenomenon — and to plumb its depths. She proved results going well beyond what I have stated here, showing for instance that the canonical Σ_1 definition can be so chosen, that $F_s \restriction M$ is the function defined over M by φ_s whenever M is transitive and closed under primitive recursive function. She also improved the characterization of such M: Call an ordinal α *nice* if it is closed under each of the function:

$$f_0(\alpha, \beta) = \alpha + \beta; f_1(\alpha, \beta) = \alpha \cdot \beta, f_2(\alpha, \beta) = \alpha^\beta \ldots \text{ etc.}$$

(More precisely: $f_{i+1}(\alpha, \beta) = \tilde{f}_i^\beta(\alpha)$ for $i \geq 1$, where $\tilde{f}_i(\alpha) = f_i(\alpha, \alpha)$, $g^\beta(\alpha)$ is defined by: $g^0(\alpha) = \alpha$, $g^{\beta+1}(\alpha) = g(g^\beta(\alpha))$, $g^\lambda(\alpha) = \sup\limits_{v<\lambda} g^v(\alpha)$ for limit λ.) She showed that L_α is primitive recursively closed iff α is nice. Moreover, $L_\alpha[A_1, \ldots, A_n]$ is closed under functions primitive recursive in $A_1, \ldots, A_n$ iff α is nice.

Primitive recursiveness in classes $A_1, \ldots, A_n$ can also be discussed in terms of primitive recursive definitions. To this end we appoint new designated function variable $\dot{a}_i (i = 1, \ldots, n)$, which will be interpreted by $\chi_{A_i} (i = 1, \ldots, n)$. By a *primitive recursive definition in* $\dot{a}_1, \ldots, \dot{a}_n$ we mean a sequence of equation having either the form (i) – (vi), in which $\dot{a}_1, \ldots, \dot{a}_n$ do not appear, or the form

(*) $f(x_1, \ldots, x_p) = \dot{a}_i(x_j)(i = 1, \ldots, n, j = 1, \ldots, p)$

We impose our previous two requirements on all equations not of the form (*).

If $s = \langle s_0, \ldots, s_{n-1}\rangle$ is a pr definition in $\dot{a}_1, \ldots, \dot{a}_n$, we successively define $F_s^{i, A_1, \ldots, A_n} (i < n)$ as before, setting $F_s^{i, \vec{A}}(x_1, \ldots, x_p) = X_{A_i}(x_j)$ if s_i has the form (*). We again set $F_s^{\vec{A}} = F_s^{n-1, \vec{A}}$. The fact that $\{\langle x, s\rangle | x \in F_s^{\vec{A}}\}$ is uniformly $\langle V, \in, A_1, \ldots, A_n\rangle$ definable is shown essentially as before:

Given an admissible $M = \langle |M|, \in, a_1, \ldots, a_n\rangle$ we define $F_s^{i,M}, F_s^M = F_s^{n-1,M}$ as before, restricting to M. The existence of the total function $F_s^{i,M}$ follows as before by admissibility. Admissibility also gives a canonical Σ_1 definition φ_s such that

$$y = F_s^M(\vec{x}) \leftrightarrow M \models \varphi_s[y, \vec{x}].$$

(Thus F_s^M is uniformly Σ_1 regardless of M.) If M, M' are admissibles of the same type and $M \subseteq M'$ (i.e. M is structurally included in M'), then $F_s^M = F_s^{M'} \restriction M$. Thus we can let $F_s^{A_1, \ldots, A_n}$ be the union of all F_s^M such that $M = \langle |M|, \in, A_1 \cap |M|, \ldots, A_n \cap |M|\rangle$ is admissible. φ_s then defines $F_s^{\vec{A}}$ over

$\langle V, \vec{A}\rangle$. (Here, Karp refined the construction so as to show that $F_s^{\vec{A}} \restriction M = F_s^M$ whenever $M = \langle |M|, \in, A_1 \cap |M|, \ldots, A_n \cap |M|\rangle$ is transitive and closed under function primitive recursive in $A_1, \ldots, A_n$. It can also be shown that $M = \langle |M|, \in, A_1, \ldots, A_n\rangle$ is closed under functions primitive recursive in $A_1, \ldots, A_n$ iff $|M|$ is primitive recursive closed and M is amenable, (i.e. $x \cap A_i \in M$ for all $x \in M$, $v = 1, \ldots, n$).

A full account of these results can be found in [PR] or [AS].

We can now state the uniformity involved in Lemma 2.2.19: Let $A_i \subset V$ be primitive recursive in $B_1, \ldots, B_m$ with primitive recursive def s_i in $\dot{b}_1, \ldots, \dot{b}_m$ $(i = 1, \ldots, m)$. Let f be primitive recursive in $A_1, \ldots, A_n$ with primitive recursive definition s in $\dot{a}_1, \ldots, \dot{a}_n$. Then f is primitive recursive in $B_1, \ldots, B_n$ by a primitive recursive definition s' in $\dot{b}_1, \ldots, \dot{b}_m$. s' is *uniform* in the sense that it depends only on $s_1, \ldots, s_n$ and s, not on $B_1, \ldots, B_m$. In fact, the induction on the schemata in s implicitly describes an algorithm for a function

$$s_1, \ldots, s_m, s \mapsto s'$$

with the following property: Let $B_1, \ldots, B_m$ be any classes. Let s_i define g_i from $\vec{B}(i = 1, \ldots, n)$. Set: $A_i = \{x | g_i(x) \neq 0\}$ in $i = 1, \ldots, n$. Let f be the function defined by s from $\vec{A}$. Then s' defines f from $\vec{B}$.

Note $\langle H_\omega, \in\rangle$ is an admissible structure; hence $F_s \restriction H_\omega = f_s^{H_\omega}$. This shows that the constant function ω is not primitive recursive, since $\omega \notin H_\omega$. It can be shown that $f : \omega \to \omega$ is primitive recursive in the sense of ordinary recursion theory iff

$$f^*(x) = \begin{cases} f(x) \text{ if } x \in \omega \\ 0 \text{ if not} \end{cases}$$

is primitive recursive over H_ω. Conversely, there is a primitive recursive map $\sigma : H_\omega \leftrightarrow \omega$ such that $f : H_\omega \to H_\omega$ is primitive recursive over H_ω iff $\sigma f \sigma^{-1}$ is primitive recursive in sense of ordinary recursion theory.

1.3 Ill founded ZF^- models

We now prove a lemma about arbitrary (possibly ill founded) models of ZF^- (where the language of ZF^- may contain predicates other than $\in$). Let $\mathbb{A} = \langle A, \in_{\mathbb{A}}, B_1, \ldots, B_n\rangle$ be such a model. For $X \subset A$ we of course write $\mathbb{A}|X = \langle X, \in_A \cap X^2, \ldots\rangle$. By the *well founded core* of $\mathbb{A}$ we mean the set of all $v \in \mathbb{A}$ such that $\in_{\mathbb{A}} \cap C(x)^2$ is well founded, where $C(x)$ is the closure of $\{x\}$ under $\in_{\mathbb{A}}$. Let wfc($\mathbb{A}$) be the restriction $\mathbb{A}|C$ of $\mathbb{A}$ to its well founded core C. Then wfc($\mathbb{A}$) is a well founded structure satisfying the axiom of extensionality, and is, therefore, isomorphic to a transitive