Interpreting higher computations as types with totality

by

Lill Kristiansen Dag Normann
Institute of Mathematics
The University of Oslo

1 *Introduction*

Qualitative domains were introduced by Girard [5] in order to give a model of system F, originally introduced by Girard [2]. The closed terms in system F will be interpreted as consistent objects in their corresponding domains. Girard added a notion of totality to the qualitative domains. A free type variable may be interpreted as any qualitative domain with any set of consistent sets representing the total objects. When the free variables are abstracted away, each closed type-term is interpreted as a specific domain with a specific set of total objects. Since there is no restriction on the families that may be used as the total objects, there is no structural or conceptual analysis of the notion of totality in Girard's work.
A general line of interest to both authors has been the search for a conceptual understanding of totality. The first author has carried out a program for investigation the notion of totality for qualitative domains.

In Normann [8] the notion of totality is discussed and axiomatized. It is shown that the elements of a fix-point of a strictly positive inductive definition within a reasonable semantics for type theory will satisfy the suggested axioms of totality set up. Berger [1] discusses the notion of totality for general domains and effective totality for effective domains. He uses his notion of totality for generalizing e.g. the Kreisel-Lacombe-Shoenfield theorem to domains.

The basis for this paper is an analysis of totality due to Kristiansen. Kristiansen has isolated a family of structures which will be qualitative domains with a set of total objects. The notion of totality is partly based on the conceptual discussion in [8]. An object will be total if it "answers" a fixed set of "questions", represented by *stable chains*. In addition Kristiansen's total objects and questions are effectively dense. This is

reflected in the supporting E-structures that will be described in section 2. The notion of totality used here is more restricted than that of Berger [1]. It will for instance not be the case in general that the set of maximal elements may form a set of total objects. This, and other results not used here will appear in the forthcoming Kristiansen [7].

In this paper we will construct a hierarchy of coherence-spaces (i.e. qualitative domains where consistency is 2-based) with totality, starting with the natural numbers as the base-type and using infinite dependent products for forming new types. Our main theorem is that the length of this hierarchy corresponds to recursion in the Kleene-functional ^3E. The proof is an implementation of a general proof-frame, and we expect similar results to hold for other natural ways of modelling types, e.g. using domain theory.
In Normann [9] a similar result is shown using the empty type. There is no significant analysis of totality behind the proof in [9]. The virtue of that proof is that the complexity can be shown using constructions standard in type theory.
In this paper the domains will be effectively nonempty. One main aspect of the proof in this paper is that if we restrict ourselves to a hierarchy of effectively nonempty types, we may ad a fine-structure on the types that is so rich that it enables us to carry out finer recursive constructions. We expect that these methods can be used for investigating other semantics of type-theory as well, and thereby throw some light on the various type-constructors used. We also see our result as the first developement of tools for investigating sets of high complexity in the same way that the Kleene-Kreisel continuous functionals , and the Dilators and Ptykes (Girard [3] and [4]), can be used to investigate concepts with complexity in the projective or analytic hierarchy.
Finally we think that the main theorem itself is of a nature where its proof will set a standard for the kind of fine-structure we are looking for in the semantics of type-theory.

2 *E - structures and totality*

We will assume familiarity with the notions of *qualitative domains* and *coherence spaces* as introduced in Girard [5]

2.1 *Definition*

Let X be a qualitative domain.

a) A *chain* in X will be a set C of pairwise inconsistent finite consistent sets.

b) If $x \in X$ and C is a chain in X, then x *meets* C if x contains an element of C as a subset.

If C_1 and C_2 are chains in X we may form the product $C_1 \times C_2$ by

$$C_1 \times C_2 = \{ A_1 \cup A_2 \in X \mid A_1 \in C_1 \text{ and } A_2 \in C_2 \}$$

$C_1 \times C_2$ is also a chain, and if x meets C_1 and C_2 then x also meets $C_1 \times C_2$.

2.2 *Definition*

a) Let X be a qualitative domain.
X is organized to a *partial E-structure* if it is equipped with
   i) To each $a \in |X|$ a chain $C_{X,a}$ containing $\{ a \}$.
   ii) To each finite $A \in X$ an *extension* $E_{X,A} \in X$ containing A as a subset.

   Let $X_{EXT} = \{ E_{X,A} \mid A \in_{fin} X \}$

   $X_{ETOT} = \{ x \in X \mid x \text{ meets } C_{X,a} \text{ for all } a \in |X| \}$

b) X is an *E - structure* if in addition all chains $C_{X,a}$ and all extensions meet, i.e. $X_{EXT} \subseteq X_{ETOT}$.

Remark

Using products of chains we construct chains $C_{X,A}$ containing A for all finite $A \in X$, and X_{ETOT} is not changed by adding these chains.

2.3 *Definition*

An E - structure X can be extended to a *totality-domain* if we add a set of questions $Q_X \cup \{ a \mid a \in |X| \}$ and a chain C_q to each question q, such that

$$X_{EXT} \subseteq X_{TOT} \subseteq X_{ETOT}$$

where $X_{TOT} = \{ x \in X \mid x \text{ meets all chains } C_q \text{ for } q \in Q_X \text{ and all chains } C_a \text{ for } a \in |X| \}$

We will require that Q_X is non-empty and that the chain C_q is not dummy, i.e. different from $\{ \emptyset \}$.

Our notion of totality is in accordance with the discussion in Normann [8]. The notion of totality in Berger [1] is somewhat more liberal but covers this case.

The restriction $X_{EXT} \subseteq X_{TOT} \subseteq X_{ETOT}$ will in some cases rule out X_{MAX} as a legal choice for X_{TOT}. In Kristiansen [7] it will be shown that there is no E - structure on $X = (\mathbf{N} \to \mathbf{N}) \to \mathbf{N}$ with $X_{TOT} = X_{MAX}$.
The standard total elements of $(\mathbf{N} \to \mathbf{N}) \to \mathbf{N}$ will however satisfy

$$X_{EXT} \subset X_{TOT} \subset X_{ETOT} \subset X_{MAX}$$

where $\subset$ is proper inclusion. This will be used in section 4.

Remark
In a totality-domain we will call a chain *total* if it meets all $x \in X_{TOT}$.
In an E - structure we will call a chain *ext-total* if it meets all $x \in X_{EXT}$ and *E-total* if it meets all $x \in X_{ETOT}$.
By a modified chain product we may define chains

$$C_{X, \{ A_1, \ldots, A_n \}} \supseteq \{ A_1, \ldots, A_n \}$$

for all finite chains $\{ A_1, \ldots, A_n \}$ such that $C_{X, \{ A_1, \ldots, A_n \}}$ is an E-total chain.

Girard [5] uses stable functions for the interpretation of function spaces, and the stable functions are represented by their traces. We assume some familiarity with the use of stable functions, though the original definition can be recovered from definition 2.4 below, using a constant parameterisation.
We see stable functions as a way to model deterministic sequential algorithms, and we will often use such algorithms for proving that a given function is stable.

2.4 *Definition*
a) Let X and Y be two qualitative domains. X is a *subdomain* of Y if $|X| \subseteq |Y|$, and if $x \subseteq |X|$ then $x \in Y$ if and only if $x \in X$.
b) A *stable parameterisation* of a family of qualitative domains is a family $\{ Y_A \}_{A \in X}$ such that:
   i) $c(A) = | Y_A |$ is a stable function, $c : X \to \mathcal{P}(\cup \{ |Y_A| \mid A \in X \})$
   ii) If $A_1 \subseteq A_2$ then Y_{A_1} is a subdomain of Y_{A_2}
c) A *stable function* relative to this parameterisation is an $f : X \to \cup \{ Y_A \mid A \in X \}$ such that
   i) $f(A) \in Y_A$ (f gives consistent output)
   ii) $A_1 \subseteq A_2$ implies $f(A_1) \subseteq f(A_2)$ (f is monotone)
   iii) If $b \in f(A)$ there is a finite $A' \subseteq A$ with $b \in f(A')$ (f is finitely based)

iv) If $A_1 \cup A_2 \in X$ and $b \in f(A_1) \cap f(A_2)$ then $b \in f(A_1 \cap A_2)$ (f is uniquely based)

d) The qualitative domain $\prod(A \in X)Y_A$ is defined using the traces of the stable functions:

We let $|\prod(A \in X)Y_A| = \{(A, b) \mid A \in_{fin} X \text{ and } b \in |Y_A|\}$

$t = \{(A_1, b_1), \ldots, (A_n, b_n)\}$ is consistent (i.e. $t \in \prod(A \in X)Y_A$) if

i) $A_{i_1} \cup \ldots \cup A_{i_k} = A \in X \Rightarrow \{b_{i_1}, \ldots, b_{i_k}\} \in Y_A$

ii) $A_i \cup A_j \in X \wedge b_i = b_j \Rightarrow A_i = A_j$

Then we have the usual bijection between stable functions and $\prod(A \in X)Y_A$ via traces and application. For $t \in \prod(A \in X)Y_A$ the application $f(A) = \lambda A \in X.App(t, A)$ is defined by

$f(A) = \{b \mid (\exists A' \subseteq A)((A', b) \in t)\}$

The $\prod$ - construction can be extended to partial E-structures, to E-structures and to totality-domains. Here we will consider parameterisations of partial E-structures and the product of such:

2.5 *Definition*

a) Let X be a qualitative domain.
A *stable parameterisation* of a family of partial E-structures indexed by X will be a stable parameterisation of the corresponding qualitative domains such that

i) For each $b \in \cup\{|Y_A| \mid A \in X\}$, the map $C_b(A) = C_{Y_A, b}$ is stable (if $b \notin |Y_A|$ then $C_b(A)$ will be empty).

ii) For each finite, consistent $B \subseteq \cup\{|Y_A| \mid A \in X\}$, the map $E_B(A) = E_{Y_A, B}$ is stable (if $B \notin Y_A$ then $E_B(A)$ will be empty).

2.6 *Lemma*

Let X be a partial E-structure and let $\{Y_A\}_{A \in X}$ be a stable parameterisation of partial E-structures.

a) $\prod(A \in X)Y_A$ can be organized to a partial E-structure.

b) If X is an E-structure, and $Y_{A'}$ is an E-structure for all $A' \in X_{EXT}$ then $\prod(A \in X)Y_A$ is an E-structure.

c) If in addition X is a totality-domain and $Y_{A'}$ is a totality-domain for all $A' \in X_{TOT}$ then $\prod(A \in X)Y_A$ is a totality-domain.

Proof

The details of the proof will be worked out in Kristiansen [7]. Here we will define the chains and the extensions in $\Pi(A \in X)Y_A$ and give the intuition behind them. Let $Z = \Pi(A \in X)Y_A$.

For part c) $Z_{TOT} = \{ f \in Z \mid (\forall A \in X_{TOT})(f(A) \in (Y_A)_{TOT} \}$.

Let $Q_Z = \{(A , q) \mid A \in X_{TOT} \wedge q \in Q_{Y_A} \}$.

The idea is that (A , q) is the question: What will $f(A)$ answer to question q ?

Of course Q_Z is exactly the questions we need to define Z_{TOT}, given that Q_{Y_A} is the set of questions needed for $(Y_A)_{TOT}$ for all $A \in X_{TOT}$.

So we want $C_{(A , q)}$ to meet (the trace of) f if and only if $f(A)$ meets C_q. This is obtained by the following definition:

$C_{(A , q)}$ contains all sets of the form $\{ (A_1 , b_1) , \ldots , (A_n , b_n) \}$ where A_i is a finite subset of A and $\{ b_1 , \ldots , b_n \} \in C_q$.

We use the same construction for producing the chains in the E-structure, i.e. in defining $C_{Z,(A,b)}$ when $(A , b) \in |Z|$ we use $E_{X,A}$ and $C_{Y_{E_{X,A}},b}$ as we used A and C_q above.

In order to define the extension $E_{Z,t}$ when

$$t = tr(g) = \{ (A_1 , b_1) , \ldots , (A_n , b_n) \}$$

we need C_{X,A_i} for $i = 1 , \ldots , n$.

The idea is to have $App(E_{Z,t} , A) = E_{Y_A , g(A)}$ for all $A \in X_{EXT}$ (in fact for all $A \in X_{ETOT}$).

Let $E_{Z,t}$ be the trace of the following stable function ϕ:

if for some i , $A_i \subseteq A$ and $b = b_i$ let $b \in \phi(A)$,

or if this is not the case

if A meets all chains $C_{X,A_1} , \ldots , C_{X,A_n}$ (then $g(A) = \{ b_i \mid A_i \subseteq A \}$),

then we let $b \in \phi(A)$ if $b \in E_{Y_A , g(A)}$.

This proves a).

$C_{Z,(A',b)}$ tests only $f(A)$ where $A = E_{X,A'} \in X_{EXT}$.

Under the assumptions of b), Y_A will be an E-structure for $A \in X_{EXT}$ and $f(A) = E_{Y_A g(A)} \in (Y_A)_{EXT} \subseteq (Y_A)_{ETOT}$. Thus

$(tr)(f) \in (\Pi(A \in X)Y_A)_{ETOT}$ if and only if $f(A) \in (Y_A)_{ETOT}$ for all $A \in X_{EXT}$.

If $\phi = \lambda A.App(E_{Z,t} , A)$ then $\phi(A) \in (Y_A)_{EXT}$ for all $A \in X_{ETOT}$.

So when $X_{EXT} \subseteq X_{TOT} \subseteq X_{ETOT}$ and $(Y_A)_{EXT} \subseteq (Y_A)_{TOT} \subseteq (Y_A)_{ETOT}$ for all $A \in X_{TOT}$ we obtain
$(\Pi(A \in X)Y_A)_{EXT} \subseteq (\Pi(A \in X)Y_A)_{TOT} \subseteq (\Pi(A \in X)Y_A)_{ETOT}$.
This proves the lemma, and establishes $\Pi(A \in X)Y_A$ as a totality domain.

We close this section by some constructions needed in section 3.

2.7 *Definition*
$\mathcal{N} = \{\emptyset, \{n\} \mid n \in \mathbf{N}\}$ with $|\mathcal{N}| = \mathbf{N}$, i.e. the set of natural numbers.
$E_{\mathcal{N}\emptyset} = \{0\}$, $E_{\mathcal{N}\{n\}} = \{n\}$, $C_{\mathcal{N},n} = \{\{m\} \mid m \in \mathbf{N}\}$ for $n \in \mathbf{N}$.
$Q_{\mathcal{N}}$ consists of one question q with $C_q = C_{\mathcal{N},\{n\}}$.
This establishes $\mathcal{N}$ as a totality domain.
Similarily we define $\mathcal{N}_k$ as the totality-domain representing $\{0, \ldots, k\}$.
$\mathcal{N}_k$ will inherit the E-structure and the totality from $\mathcal{N}$, so in all respects it will be a substructure of $\mathcal{N}$.

In Normann [8] embeddings between totality-domains are discussed and strictly positive inductive definitions are carried out. There are no stability-requirements and nothing like an E-structure is used in [8]. Moreover we let the empty domain start the induction.
In Kristiansen [7] we have qualitative domains with E-structure and totality. It is shown that a non-empty least fixpoint of a general positive induction can be interpreted as an E-structure with totality. We exclude the empty domain $\{\emptyset\}$ from being an E-structure with totality.

We do not need embeddings between totality-domains in this paper, only the substructure relation for qualitative domains and the interpretation of $\oplus$ and $\times$ for qualitative domains. These are discussed in detail in Girard [5], so we will be brief:

2.8 *Definition*
a) Let $X_1, \ldots, X_n$ be qualitative domains.
We define $X = X_1 \times \ldots \times X_n$ by
$$|X| = (\{1\} \times |X_1|) \cup \ldots \cup (\{n\} \times |X_n|)$$
For $x \subseteq |X|$ let $\pi_i(x) = \{a \mid (i, a) \in x\}$
Then we let $x \in X$ when $\pi_i(x) \in X_i$ for all $i = 1, \ldots, n$.
We will sometimes write $x = (x_1, \ldots, x_n)$ for $x_i = \pi_i(x)$.
b) If X_1 and X_2 are qualitative domains, we define $Y = X_1 \oplus X_2$ by

$|Y| = (\{1\} \times |X_1|) \cup (\{r\} \times |X_2|)$.

If $y \in |Y|$ we let $y \in Y$ if for some y_1 in Y_1 we have that $y = \{1\} \times y_1$ or if for some y_2 in Y_2 we have that $y = \{r\} \times y_2$.

We will sometimes write $r(a)$ for (r, a) and $r(x)$ for $\{r\} \times x$, etc.

In Kristiansen [7] it will be shown how these constructions can be extended to E-structures and to totality-domains.

3 *Codes for syntactic forms*

In this section we will define a hierarchy of E-structures with totality with the natural numbers $\mathcal{N}$ at the base and using products of stable parameterisations for obtaining new structures.
We will let all structures be substructures (as domains) of one universal domain D, and we will represent the various elements of the hierarchy by consistent sets s in a domain S of "syntactic forms".

3.1 *Definition*

a) Let $D_0 = \emptyset$, $D_{k+1} = \mathcal{N}_k \oplus (D_k \to D_k)$, and let D be the limit-structure.

b) Let $S_0 = \mathcal{N}_0 \oplus \emptyset$, $S_{k+1} = \mathcal{N}_0 \oplus (\mathcal{N}_0 \times S_k \times (D_k \to S_k))$, and let S be the limit.

Remark

D will be a qualitative domain satisfying

$$D = \mathcal{N} \oplus (D \to D)$$

while S will be a qualitative domain satisfying

$$S = \mathcal{N}_0 \oplus (\mathcal{N}_0 \times S \times (D \to S))$$

If $s \in S$, $s \neq \emptyset$ and $s \neq r(\emptyset, s_1, p)$ we have

i) $s = 1(\{0\})$

or

ii) $s = r(\{0\}, s_1, p\}$

The idea is that in case i) s will represent $\mathcal{N}$ while in case ii), if s_1 represents X and $p(A)$ represents Y_A for $A \in X$ then s represents $\Pi(A \in X)Y_A$.

Following this idea we will define the *interpretation* $I(s)$ for $s \in S$ as qualitative domains such that $\{I(s)\}_{s \in S}$ is a stable parameterisation of

qualitative domains. Also, each $I(s)$ will be given a partial E-structure, and we will show that $\{I(s)\}_{s\in S}$ also is a stable parameterisation of partial E-structures.
In case i) $\emptyset \in \mathcal{N}$ and $E_{s,\emptyset} = \{0\}$, while in case ii) $\emptyset \in \prod(A \in X)Y_A$ and $E_{s,\emptyset} = (tr)(\lambda A \in X.E_{Y_A,\emptyset})$
In order to make $E_{s,\emptyset}$ stable in s it is convenient to use $r(\{0\},\emptyset,\emptyset)$ as the unique basis for going to the right.

We will use the following conventions: d will be an element in $|D|$ while δ will be a finite element in D.
s will be an arbitrary element of S while σ will be a finite element of S.
p will be an element of $D \to S$ while π will be a finite element of $D \to S$.

3.2 *Definition*
We define $I(s)$ for $s \in S$ by defining $I(\sigma)$ for finite $\sigma \in S$ essentially by induction on the least k for which $\sigma \in S_k$.
For $\sigma = \emptyset$, let $|I(\sigma)| = \emptyset$.
For $\sigma = l(\{0\})$, let $I(\sigma) = \mathcal{N} \oplus \emptyset$
If $\sigma = r(\{0\},\sigma_1,\pi) \in S_{k+1}$, then $\sigma_1 \in S_k$ and $\pi \in D_k \to S_k$, and we let $d = r(\delta_1, d_2) \in |I(\sigma)|$ if and only if $\delta_1 \in I(\sigma_1)$ and $d_2 \in |I(\pi(\delta_1))|$.
Consistency is inherited from D, i.e. consistency of traces.

3.3 *Lemma*
The function $c(s) = |I(s)|$ is a stable function from S to $\mathcal{P}(|D|)$.

Proof
By induction on k we can prove this for $\sigma \in S_k$. The details are left for the reader.

We see that $I(s)$ is the minimal solution to the equation system

* $I(l(\{0\})) = \mathcal{N} \oplus \emptyset$
** $I(r(\{0\},s,p)) = \emptyset \oplus \prod(x \in I(s))\, I(p(s))$

and lemma 3.3 ensures that this minimal solution exists.

Likewise we can use the construction of lemma 2.6 to organize each $I(s)$ to a partial E-structure. By induction on k we can prove that for $\sigma \in S_k$ this is a stable parameterisation, and thus $\{I(s)\}_{s\in S}$ can be considered as

a stable parameterisation of partial E-structures. We thus have

3.4 *Theorem*
There is a minimal solution $\{ I(s) \}_{s \in S}$ as a stable parameterisation of partial E-structures such that the equations * and ** hold.

We will now define our hierarchy of types, $\{ I(s) \}_{s \in S_{wf}}$:

3.5 *Definition*
Simultanously we define $S_{wf} \subseteq S$ and $(I(s))_{TOT}$ for $s \in S_{wf}$ as the least solution of

i) $1(\{ 0 \}) \} \in S_{wf}$ with $(I(s))_{TOT}$ as $\{ 1(\{ n \}) \mid n \in \mathbf{N} \}$

ii) $r(\{ 0 \} , s , p \} \in S_{wf}$ if $s \in S_{wf}$ and for all $x \in (I(s))_{TOT}$ we have that $p(x) \in S_{wf}$.
$(I(r(\{ 0 \} , s , p \}))_{TOT}$ will be $r(\{ x \mid x$ is the trace of a stable function mapping an element $y \in (I(s))_{TOT}$ onto an element of $(I(p(y)))_{TOT} \}$

Using lemma 2.6 c) , lemma 3.4 and induction we see that when $s \in S_{wf}$ then $I(s)$ is a totality-domain, and thus S_{wf} is well defined.

In section 4 we will investigate the length of this inductive definition. In order to do so we need some more structure on the domains $I(s)$ for $s \in S_{wf}$. Notice that if $s \in S_{wf}$ then all extensions in $I(s)$ are total and the total objects meet all the chains.

3.6 *Definition*
Assume that we have a fixed enumeration of each finite $\delta \in D$.
We define the stable functions $I^d : S \to \mathcal{P}(|D|)$ and $I^d_f : S \to \mathcal{P}(|D|^{<\omega})$ as the minimal solution to the following equations:

$I^d(1(\{ 0 \})) = | \varnothing \oplus (D \to D) |$

$I^d(r(\{ 0 \} , s , p) = | \mathcal{N} \oplus \varnothing |$
$\quad \cup \{ r(\delta , d) \mid \delta \in I^d_f(s) \}$
$\quad \cup \{ r(\delta , d) \mid \delta \subseteq |I(s)| \wedge (d \in I^d(p(E_{I(s), \delta})) \vee d \in I(p(E_{I(s), \delta})$
$\quad$ but the basis for this $(\subseteq E_{I(s), \delta})$ is not contained in $\delta \}$.

$I^d_f(s) = \{ \delta = \{ d_1 , \ldots , d_n \} \in D \mid n \geq 1$ and $\exists k \leq n (d_1 \in |I(s)| \wedge \ldots \wedge d_{k-1} \in |I(s)| \wedge d_k \in I^d(s))$.

3.7 *Lemma*

a) For all $s \in S$ $I^d(\sigma) \cap |I(s)| = \emptyset$

$I^d_f(s) \cap I(s) = \emptyset$

b) For all $s \in S_{wf}$, for all $d \in |D|$ and for all finite $\delta \in D$ we have

$d \in |I(s)|$ or $d \in I^d(s)$

$\delta \in I(s)$ or $\delta \in I^d_f(s)$

The proof is straightforward and is left for the reader.
Notice that we need 3.7 a) in order to show that I^d and I^d_f are stable functions.

The definition of I^d_f illustrates that stable functions correspond to deterministic, sequential algorithms. If we instead define I^d_f by

$$I^d_f(s) = \{\delta \in D \mid \exists d \in \delta(d \in I^d(s))\}$$

then I^d_f (and hence I^d at the next level) will <u>not</u> be stable.

We will use $|I(s)|$ and $I^d(s)$ to transform total elements x in $I(s)$ for $s \in S_{wf}$ to total functions $h_{x,s}$ from $\mathbf{N}$ to $\mathbf{N}$ in a stable way

3.9 *Definition*
Let $\{d_i\}_{i \in \mathbf{N}}$ be a fixed enumeration of $|D|$, and let $\{\delta_i\}_{i \in \mathbf{N}}$ be a fixed enumeration of the finite elements in D.
For $s \in S$ and $x \in I(s)$ let

$$h_{x,s}(\{i\}) = \begin{cases} \{0\} \text{ if } d_i \in I^d(s) \\ \{j+1\} \text{ if } d_i \in |I(s)| \text{ and } \delta_j \text{ is where } x \text{ meets} \\ C_{I(s),d_i}. \end{cases}$$

Thus $h_{x,s}(i)$ tests if d_i is in $|I(s)|$, and in case it is, find where x meets $C_{I(s),d_i}$. So, if $s \in S$ is such that $|I(s)|$ and $I^d(s)$ are complementary, we have that $h_{x,s}$ is total if and only if $x \in I(s)_{ETOT}$.

3.10 *Lemma*

a) $\lambda s \in S.\lambda x \in I(s).h_{x,s}$ is stable

b) For $s \in S_{wf}$ and $x, y \in (I(s))_{TOT}$ we have that $h_{x,s}$ is total and $x = y$ if and only if $h_{x,s} = h_{y,s}$.

c) For any $s \in S$ and $x \in I(s)$, if $h_{x,s}$ is total, then x is maximal.

The proofs are easy and left for the reader. For c) , notice that it is a general fact for E-structures X that if $x \in X_{ETOT}$ then $x \in X_{MAX}$.

4 *Simulating recursion in* ^3E

We will now prove that the hierarchy S_{wf} is as complex as recursion in ^3E. ^3E is the type-3 functional defined as follows:

$$^3E(F) = \begin{cases} 0 & \text{if } F \text{ is total with constant value } 0 \\ 1 & \text{if } F \text{ is total, but not constant } 0 \end{cases}$$

Kleene [6] defined a notion of computations $\{e\}(\psi_1, \ldots, \psi_k) \approx n$ where $\psi_1, \ldots, \psi_k$ is a sequence of total functionals of finite, pure type. We will be interested in computations with input ^3E and a sequence f from $\mathbf{N} \cup \mathbf{N}^{\mathbf{N}}$. We will omit the mentioning of ^3E in our computations. When we write

$$\{e\}\{f\) \approx n$$

we mean that $f = f_1 f_2$ and that

$$\{e\}(f_1, {}^3E, f_2) \approx n$$

according to Kleene's original definition.
We will systematically omit the Kleene-scheme S5 in our treatment, since this scheme can be reduced to the other schemes.

We will prove the following theorem

4.1 *Theorem*

Assume that $\{e\}\{f\)\downarrow$, i.e. takes a value.
Uniformly in e, f and n there is

1. An element $C(e, f, n) \in S_{wf}$
2. An object $\alpha(e, f, n) \in I(C(e, f, n))$

such that

$\alpha(e, f, n)$ is total in $I(C(e, f, n))$ if and only if $\{e\}(f) \approx n$.

With our conventions, this shows that semirecursion in ^3E can be reduced to

$$\{(s, x) \mid s \in S_{wf} \text{ and } x \text{ is total in } I(s)\}.$$

The converse is not hard to prove, see Normann[9] for an analogue result.

In a way we will construct $C(e, f, n)$ and $\alpha(e, f, n)$ by recursion on the

computation $\{e\}(f)$. In the case of composition we will use a mollification-technique in order to avoid explicit use of the intermediate value, and for this we will need approximal computations. The approximations will be more at the syntactical than at the computational level, but this will be sufficient for our purposes.

4.2 *Definition*
By recursion on k we define $\{e\}_k(f)$ for all e and f.
i) $\{e\}_0(f) = 0$ for all e, f.
ii) Assume that $\{e\}_k(f)$ is defined for all e, f.
We define $\{e\}_{k+1}(f)$ following S1 - S4, S6 - S9 as follows
1. $\{e\}(x,f) = x+1 : \{e\}_{k+1}(x,f) = x+1$
2. $\{e\}(f) = q : \{e\}_{k+1}(f) = q$
3. $\{e\}(x,f) = x : \{e\}_{k+1}(x,f) = x$
4. $\{e\}(f) \approx \{e_1\}(\{e_2\}(f),f) : \{e\}_{k+1}(f) = \{e_1\}_k(\{e_2\}_k(f),f)$
6. $\{e\}(f) \approx \{e_1\}(\sigma(f)) : \{e\}_{k+1}(f) = \{e_1\}_k(\sigma(f))$
7. $\{e\}(x,f,f) = f(x) : \{e\}_{k+1}(x,f,f) = f(x)$
8. $\{e\}(f) \approx {}^3E(\lambda g.\{e_1\}(g,f)) : \{e\}_{k+1}(f) = {}^3E(\lambda g.\{e_1\}_k(g,f))$
9. $\{e\}(e_1,f) \approx \{e_1\}(f) : \{e\}_{k+1}(e_1,f) = \{e_1\}_k(f)$
Otherwise we let $\{e\}_{k+1}(f) = 0$

4.3 *Lemma*
Uniformly in the signature for f there is a primitive recursive function ρ such that
i) $\{e\}_k(f) = \{\rho(k,e)\}(f)$ for all e, f.
ii) If $k_1 \leq k_2$ then

$$\rho(k_1, \rho(k_2, e)) = \rho(k_1, e)$$
$$\rho(k_2, \rho(k_1, e)) = \rho(k_1, e)$$

The proof is trivial and is left for the reader. We will omit any reference to the signature of f, though we need it to distinguish between the otherwise-case and the rest.

The next definition represents the main construction in this paper, and the central idea of proof is laid down in this definition:

4.4 *Definition*
By induction on k we define
$C_k(e,f,n)$ and $\alpha_k(e,f,n)$
for all f and n.

We will have that $C_k(e,f,n) \in S$ and that the definition is monotone in k.
We will have that $\alpha_k(e,f,n) \in I(C_k(e,f,n))$ and that α_k is monotone in k.
Both $C_k(e,f,n)$ and $\alpha_k(e,f,n)$ are uniformly recursive in e,f,n as subsets of $|S|$ and $|D|$ respectively.

If $\{e\}(f)$ is an initial computation, let $m = \{e\}(f)$.
Let $C_k(e,f,n)$ be the element of S representing $(\mathbf{N} \to \mathbf{N}) \to \mathbf{N}$ for all k.
Let $\alpha(e,f,n)$ be the element of $I(C_k(e,f,n))$ representing
$\lambda f.\mu k((f(0) = k \wedge n = m) \vee f(k) \neq k)$, and let
$\alpha_k(e,f,n) = \alpha(e,f,n) \cap |D_k|$.

If $\{e\}(f)$ is at first sight not a computation, we let $C_k(e,f,n)$ represent $(\mathbf{N} \to \mathbf{N}) \to \mathbf{N}$ for all k.
Let $\alpha(e,f,n)$ be the element of $I(C_k(e,f,n))$ representing
$\lambda f.\mu k(f(k) \neq k)$, and let $\alpha_k(e,f,n) = \alpha(e,f,n) \cap |D_k|$.

This leaves us with the cases S4, S6, S8 and S9.
Then we let $C_0(e,f,n) = \emptyset$ and $\alpha_0(e,f,n) = \emptyset$, and we will define C_{k+1} and α_{k+1} for these cases.

S4: $\{e\}(f) \approx \{e_1\}(\{e_2\}(f),f)$.
We want $C(e,f,n)$ to represent the following object:

$$\Pi(m \in \mathbf{N})\Pi(x \in I(C(e_2,f,m))I(D(x,m)))$$

where $D(x,m) = C(e_1,m,f,n)$ if $x = \alpha(e_2,f,m)$
$D(x,m) = C(\rho(k,e_1),mf,n)$ for the least k such that
$h_{x,C(e_2,f,m)}(k) \neq h_{\alpha(e_2,f,m),C(e_2,f,m)}(k)$
when $x \neq \alpha(e_2,f,m)$.

We let $C_{k+1}(e,f,n)$ be the trace of the following function in m and x:
$r \in D_k(x,m)$ if there is a minimal $k_1 \leq k$ such that
i) $r \in C_{k_1}(e_1,m,f,n)$
or

ii) $h_{x,C_k(e_2,f,m)}(k_1) \neq h_{\alpha_k(e_2,f,m),C_k(e_2,f,m)}(k_1)$,

and in case ii), $r \in C_k(\rho(k_1,e_1),m,f,n)$.

In order to prove that this is stable, we must show that

$k_1 \leq k_2 \Rightarrow C_{k_1}(e,f,n) \subseteq C_{k_2}(\rho(k_2,e),f,n)$.

We will do so when the full definition of C_{k+1} is given.

We also want $\alpha(e,f,n)$ to represent the function

$\lambda m.\lambda x \in I(C(e_2,f,m)).\beta(x,m)$

where

$\beta(x,m) = \alpha(e_1,m,f,n)$ if $x = \alpha(e_2,f,m)$

$\beta(x,m) = E_{C(\rho(k,e_1),m,f,n),\alpha_k(e_1,m,f,n)}$ for the least k such that

$h_{x,C(e_2,f,m)}(k) \neq h_{\alpha(e_2,f,m),C(e_2,f,m)}(k)$

when $x \neq \alpha(e_2,f,m)$.

Let $\phi = \lambda m.\lambda x \in I(C_k(e_2,f,m))\beta_k(x,m)$

where we have the following algorithm for computing $d \in \beta_k(x,m)$:

Find a minimal $k_1 \leq k$ such that

i) $d \in \alpha_{k_1}(e_1,m,f,n)$

or

ii) $h_{x,C_k(e_2,f,m)}(k_1) \neq h_{\alpha_k(e_2,f,m),C_k(e_2,f,m)}(k_1)$.

If i) applies, we let $d \in \beta_k(x,m)$.

If ii) applies we let $d \in \beta_k(x,m)$ if

$d \in E_{C_k(\rho(k,e_1),m,f,n),\alpha_{k_1}(e_1,m,f,n)}$

We let $\alpha_{k+1}(e,f,n)$ be the part of ϕ that is in D_{k+1}.

This end case 4.

S6: $\{e\}(f) \approx \{e_1\}(\sigma(f))$

Let $C_{k+1}(e,f,n) = C_k(e_1,\sigma(f),n)$

$\alpha_{k+1}(e,f,n) = \alpha_k(e_1,\sigma(f),n)$

S8: $\{e\}(f) \approx {}^3E(\lambda g.\{e_1\}(g,f))$

We want $C(e,f,0)$ to represent $\Pi(g \in \mathcal{N} \to \mathcal{N})I(C(e_1,g,f,0))$

and $\alpha(e,f,0)$ to represent $\lambda g.\alpha(e_1,g,f,0)$.

Moreover we want $C(e,f,1)$ to represent $I(C(e,f,0)) \to \mathcal{N}$ and $\alpha(e,f,1)$ to represent $\lambda x.\mu k(h_{x,C(e,f,o)}(k) \neq h_{\alpha(e,f,0),C(e,f,0)}(k))$. We do this by letting

$C_{k+1}(e,f,0)$ represent $\Pi(g \in \mathcal{N} \to \mathcal{N})I(C_k(e_1,g,f,0)$

$\alpha_{k+1}(e,f,0)$ represent $\lambda g.\alpha_k(e_1,g,f,0)$ restricted to D_{k+1}

$C_{k+1}(e,f,1)$ represent $I(C_{k+1}(e,f,0)) \to \mathcal{N}$

$\alpha_{k+1}(e,f,1)$ represent $\lambda x.\mu k_1 \leq k(h_{x,C_{k+1}(e,f,o)}(k) \neq h_{\alpha_{k+1}(e,f,0),C_{k+1}(e,f,0)}(k))$ restricted to D_{k+1}.

If $n > 1$ we let $C_{k+1}(e,f,n)$ represent $(\mathcal{N} \to \mathcal{N}) \to \mathcal{N}$ and we let $\alpha_{k+1}(e,f,n)$represent $\lambda g.\mu k_1(g(k_1) \neq k_1)$ restricted to D_{k+1}.

S9: $\{e\}(e_1,f) \approx \{e_1\}(f)$:

Let $C_{k+1}(e,e_1,f,n) = C_k(e_1,f,n)$

$\alpha_{k+1}(e,e_1,f,n) = \alpha_k(e_1,f,n)$

This ends Definition 4.4.

4.5 *Lemma*

a) If $k_1 < k$ then for any e,f,n

$C_{k_1}(e,f,n) = C_{k_1}(\rho(k,e),f,n)$

$\alpha_{k_1}(e,f,n) = \alpha_{k_1}(\rho(k,e),f,n)$

b) For all e,f,n and k

$C_k(e,f,n) \subseteq C_k(\rho(k,e),f,n)$

$\alpha_k(e,f,n) \subseteq \alpha_k(\rho(k,e),f,n)$

Proof

The proof of a) is by induction on k_1 while the proof of b) is by induction on k. Both proofs are tedious but simple.

This lemma shows that the definition of C_k and α_k is sound. Moreover, they are recursive, so they are stable in the variables e, f, n.

We obviously have that $\alpha_k(e,f,n)$ is finite and that the definitions are monotone in k.

As indicated, we let

$$C(e,f,n) = \cup(k \in \mathbf{N})C_k(e,f,n)$$

$$\alpha(e,f,n) = \cup(k \in \mathbf{N})\alpha_k(e,f,n)$$

We complete the proof of Theorem 4.1 by showing:

4.6 *Lemma*
If $\{e\}(f)\downarrow$ then for all n
a) $C(e,f,n) \in S_{wf}$ and
$\alpha(e,f,n)$ is total in $I(C(e,f,n))$ if and only if $\{e\}(f) = n$
b) $h_{\alpha(e,f,n),C(e,f,n)}$ is total
c) $\alpha(e,f,n)$ is not an extension in $I(C(e,f,n))$, i.e. not one of the $E_{X,\delta}$ sets.

Proof
The lemma is proved by induction on the length of the computation, which gives 8 cases S1-S4, S6-S9. To be formally correct we should first prove the lemma for computations on the form $\{\rho(k,e)\}(f)$ by induction on k, but since this will only lead to a repetition of details, we leave it for the reader.

For initial computations, a) is trivial, and in these cases C and α are carefully defined for b) and c) to hold.
The cases S6 and S9 are trivial, so we concentrate on cases S4 and S8.

S4 $\{e\}(f) = \{e_1\}(\{e_2\}(f),f)$.
Let $m_0 = \{e_2\}(f)$, $n_0 = \{e_1\}(m_0,f)$
By construction
$I(C(e,f,n)) = r(\Pi(m \in \mathcal{N})\Pi(x \in I(C(e_2,f,m)))I(D(x,m)))$
where
$D(x,m) = C(e_1,m,f,n)$ if $x = \alpha(e_2,f,m)$
$D(x,m) = C(\rho(k,e_1),m,f,n)$ for some k otherwise.
Here we use that $h_{\alpha(e_2,f,m),C(e_2,f,m)}$ is total.

If $m \neq m_0$ and x is total in $I(C(e_2,f,m))$ we cannot have that $x = \alpha(e_2,f,m)$, so $h_{x,C(e_2,f,m)} \neq h_{\alpha(e_2,f,m),C(e_2,f,m)}$, and $D(x,m) = C(\rho(k,e_1),f,m)$ for some k depending on x.
Then $\Pi(x \in I(C(e_2,f,m)))I(D(x,m)$ can be represented in S_{wf}.
Moreover $\alpha(e,f,n)(m)(x)$ is an extension and thus total whenever x is total in $I(C(e_2,f,m))$.
If $m = m_0$ we can argue as above except for $x = \alpha(e_2,f,m_0)$.

Then $D(x, m_0) = C(e_1, m_0, f, n)$, which is in S_{wf} by the induction hypothesis. This shows that $C(e, f, n) \in S_{wf}$.
Moreover, totality of $\alpha(e, f, n)$ is equivalent to totality of $\alpha(e_1, m_0, f, n)$ in $I(C(e_1, m_0, f, n))$ which again is equivalent to $n = n_0$.
This proves a) for case S4.
To prove b), recall that the extending chains in a products are based on an extension A in the parameter-space X and chains in the corresponding Y_A. Since $\alpha(e, f, n)$ is total except at most at $(m_0, \alpha(e_2, f, m_0))$, and since $\alpha(e_2, f, m_0)$ is not an extension it follows that $\alpha(e, f, n)$ will meet all extending chains, and b) follows.
To prove c) , notice that an extension in a product space will send all total elements to extensions. Since $\alpha(e_1, m_0, f, n)$ is not an extension, c) follows.

S8: $\{e\}(f) = {}^3E(\lambda g.\{e_1\}(g, f))$.
For $n \neq 0,1$ the case is trivial.
8.0: $C(e, f, 0)$ represents $\Pi(g \in \mathcal{N} \to \mathcal{N})I(C(e_1, g, f, 0))$, so $C(e, f, 0)$ will be in S_{wf} independent of the values of $\{e_1\}(g, f)$ for various g.
If $\{e_1\}(g, f) = 0$ for all g, then $\alpha(e, f, 0)$ representing $\lambda g.\alpha(e_1, g, f, 0)$ is total, otherwise not.
In any case, $\alpha(e, f, 0)$ is not an extension because $\alpha(e, f, 0)(g)$ is not an extension for any g.
Moreover $h_{\alpha(e,f,0),C(e,f,0)}$ is total because $h_{\alpha(e_1,g,f,0),C(e_1,g,f,0)}$ are total for all g.

8.1: $C(e, f, 1)$ represents $I(C(e, f, 0)) \to \mathcal{N}$ and is in S_{wf}.
We have

$\{e\}(f) = 1$
$\Leftrightarrow$ $\alpha(e, f, 0)$ is not total in $I(C(e, f, 0))$
$\Leftrightarrow$ $\alpha(e, f, 1)$ is total in $I(C(e, f, 1))$.

This shows a), and b) and c) are trivial in this case.

This ends the proof of the lemma and of Theorem 4.1.

REFERENCES

1. Berger, U.: Total sets and objects in Domain Theory, Submitted for publication.

2. Girard, J.-Y.: Une extension de l'interprétation de Gödel et à la théorie des types etc., in J. E. Fenstad (ed.): Proc. 2nd Scand. Log. Symp. , North Holland (1971)

3 Girard, J.-Y.: Π^1_2 -logic ,Part 1: Dilators, Ann. Math Logic 21, pp 75-219, 1981.

4 Girard, J.-Y.: Proof theory and logical complexity, part II, to appear at editions North Holland.

5. Girard, J.-Y.: The system F of variable types fifteen years later, TCS 45 (1986)

6. Kleene, S. C.: Recursive functionals and Quantifiers of finite types I, T.A.M.S. 91 (1959), 1 - 52.

7. Kristiansen, L.: Thesis, in preparation.

8. Normann, D.: Formalizing the notion of total information, in P. P. Petkov(ed.) Mathematical Logic, Plenum Press (1990) 67-94

9. Normann, D. : Wellfounded and non-wellfounded types of continuous functionals, Oslo Preprint Series in Mathematics No 6 (1992)