Theoretical Computer Science 45 (1986) 159-192
North-Holland

THE SYSTEM *F* OF VARIABLE TYPES, FIFTEEN YEARS LATER

Jean-Yves GIRARD
Équipe de Logique Mathématique, UA 753 du CNRS, 75251 Paris Cedex 05, France

Communicated by M. Nivat
Received December 1985
Revised March 1986

Abstract. The semantic study of system *F* stumbles on the problem of *variable types* for which there was no convincing interpretation; we develop here a semantics based on the category-theoretic idea of *direct limit*, so that the behaviour of a variable type on *any* domain is determined by its behaviour on *finite* ones, thus getting rid of the circularity of variable types. To do so, one has first to simplify somehow the extant semantic ideas, replacing Scott domains by the simpler and more finitary *qualitative domains*. The interpretation obtained is extremely compact, as shown on simple examples. The paper also contains the definitions of a very small 'universal model' of lambda-calculus, and investigates the concept totality.

Contents

Per te, serena

Introduction

In 1970, the present author [3] introduced the idea of *variable type*, i.e., of a schema of abstraction w.r.t. types. A typical example was, for instance, to abstract

the identity function of any type σ, i.e., $\lambda x^{\sigma}.x^{\sigma}$ from the type σ, thus getting the 'universal identity' $\Lambda\alpha.\lambda x^{\alpha}.x^{\alpha}$. This universal identity has in turn the type $\Lambda\alpha.\alpha \Rightarrow \alpha$, which is the type of functions (if one can call them that way) associating to each type σ an object of type $\sigma \Rightarrow \sigma$. In fact, the formalism was quite general, since the formation of type abstraction was not limited at all. What was of course problematic was the schema of evaluation of a function of universal type $\Lambda\alpha.\sigma[\alpha]$, because it was possible to apply an object t of this type to *any* type τ, yielding $t\{\tau\}$ of type $\sigma[\tau/\alpha]$: this obviously gave circularity problems.

However, in [3] it was shown that the obvious rules of conversion for this system, called F by chance, were converging. The proof used a predicate of 'hereditary calculability', not expressible in second-order arithmetic PA_2. To do this, we were helped by Gödel's second incompleteness theorem, since we had already shown, using functional interpretation, that if the computations in F were to converge, then every provably total recursive function of PA_2 would be representable in F.

At that time, the results on F did not attract too much attention: people were more interested in the proof of the syntactic form of Takeuti's conjecture, which was contained in the same paper, and was practically the same result, from the point of view of the Curry-Howard-De Bruijn isomorphism between typed systems and natural deduction. However, F was one of the first sources of inspiration of Martin-Løf for his famous system; but, in order to handle it, he had to use the axiom "$V \in V$" which later turned out to be inconsistent, and F disappeared from the ulterior background of Martin-Løf's system. One should also mention the semantics for F of Troelstra [9], "hereditarily recursive operations of order 2" (HRO_2), which was a way of interpreting F via indices of partial recursive functions.

Later on, the system was found again by Reynolds [7], and the subject moved in the direction of computer science. The interest for computer science, according to Krivine, lies in the fact that F provides a way of computing, in which recursion (in the sense of a program calling itself) is absent, and in which one must program as one makes mathematical proofs; in fact, all kinds of current computer science data—lists, trees, pairs—have a nice description in F. So, there has been a lot of progress in the direction of how to use the system F.

Mathematically, the progress has been more limited; the papers written by Leivant, Statman etc. on the subject just reprove the original results. In fact, there has not been any mathematical progress w.r.t. the syntax of F. For the semantics, a little more has been done, namely the work of Reynolds [7] showing the impossibility of a model of F with the set-theoretic interpretation of the implicative types. Also, attempts have been made to give models by means of some kinds of Scott domains, but this kind of approach is essentially the same as Troelstra's, mentioned above: simply partial recursive functions are replaced by λ-terms, and these λ-terms are in turn interpreted in a Scott domain.

This paper uses new semantic ideas (see [6]) to develop a model for F. In [6], the present author introduced two category-theoretic semantics for λ-calculus: the quantitative and the qualitative one. F can be modelised in both, but since [6] was

mainly concerned with the quantitative case, here we have chosen to develop the qualitative framework only.

The main problem is of course the interpretation of terms of variable types, i.e., what does it mean that we have $t\{\sigma\}$ for all types σ? As mentioned above, there is an obvious circularity problem. Category theory provides an elegant way of getting rid of this circularity: we interpret a type as a qualitative domain ("qD" for short), and we want to make sense for the concept of a function associating, for any qD X, an object $t\{X\}$ of type $\sigma[X]$. Then we observe that, perhaps, t is a functor from qualitative domains to something not specified, with nice preservation properties, namely direct limits and pull-backs. Then, using a *normal form theorem* for such functors, it is possible to show that their behaviour is determined by what they do on *finite* qD's, and then we get rid of the circularity. Moreover, the term of variable type can in turn be encoded by its *trace* which is nothing but the set of possible kinds of normal forms, and so a universal type has the good taste to form a qualitative domain. The interpretations thus obtained are very small (because combinations coming from the same normal form are only counted once), and for instance, the universal identity we started with has an interpretation consisting of one point.

The interpretation of F also has an interesting by-product for λ-calculus: it is well known that λ-calculus can be modelised as soon as one can solve an equation

$$X \Rightarrow X \sim X$$

among some kind of domains, in particular qualitative ones. Unfortunately, all solutions of such an equation are more or less arbitrary, i.e., no model of λ-calculus built in this way can claim to be 'the' model. In order to get only one model, one simply has to remark that the interpretation t^*_D of t in the model depends functorially on the data (X, H, K) defining D (H and K being the isomorphism and its converse, respectively). Then, it suffices to prove that the functor has nice preservation properties, and to compute its invariants (its trace), and we get a very small interpretation t^* from which we can compute t^*_D in any D.

However, the situation is a bit more complex, since an isomorphism is not a direct limit of finite isomorphisms, and we have to somehow liberalise the requirements on H and K. The current requirement is that both $H \circ K$ and $K \circ H$ should be projectors, i.e., subobjects of the identity. For such models, the interpretation increases w.r.t. β- and η-reduction. It is even simpler to drop any requirement about H and K, and then we discover that we are just interpreting the term of F

$$\Lambda X.\lambda H^{X\Rightarrow(X\Rightarrow X)}.\lambda K^{(X\Rightarrow X)\Rightarrow X}.t^*_{X,H,K}$$

of type

$$\Lambda X.(X\Rightarrow(X\Rightarrow X))\Rightarrow(((X\Rightarrow X)\Rightarrow)\Rightarrow X).$$

This interpretation, which encodes any possible model of the λ-calculus by means of small sets of invariants, is called Λ_0. Λ_0 has no property w.r.t. conversion. But it is possible to define a subset $|\Lambda_1|$ or $|\Lambda_0|$, which corresponds exactly to those

models for which $H \circ K$ and $K \circ H$ are projectors. Restricted to $|\Lambda_1|$, the interpretation increases during the reduction process, and this restricted interpretation is enough to get the interpretation of t in any model where, for instance, H and K are reciprocal isomorphisms. Since there is no feedback from $|\Lambda_0| - |\Lambda_1|$ on $|\Lambda_1|$, we have not tried to eliminate the nonincreasing part which may be of some interest.

This model has been called the intrinsic model for obvious reasons. However, it is intrinsic only within a specified kind of interpretation: here qualitative domains. If one changes the kind of interpretation, for example quantitative domains, then the same kind of interpretation will lead to an intrinsic model of that kind.

This work is, as to the author's knowledge, perfectly original. It is essentially the transposition of methods already used by the present author in the theory of dilators, to the context of semantics of λ-calculi. However, to important notions used here have already been considered in the literature by Berry [1], namely 'stability' (i.e., the analogue for Scott domains of our Condition (ST3), and 'order' (see our Definition 1.7). Elegant ideas in this domain are not so common, and we therefore decided to use the name 'stable' for the functions used here, and to call 'order' the *Berry order.* A version of Theorem 1.4 (for Scott domains) can be found in [10].

However, the notion of a qualitative domain, basically a clean refinement of Scott domains, seems to be original.

1. Qualitative domains and λ-structures

Let us first recall the basic definitions and results concerning qualitative domains.

1.1. Definition. A *qualitative domain* (qD) is a set X such that;
(qD1) $\emptyset \in X$,
(qD2) X is closed under direct unions,[1] and
(qD3) if $a \in X$ and $b \subset a$, then $b \in X$.

We use the notation $|X|$ to denote $\{z; \{z\} \in X\}$; by (qD3), this set is also equal to $\bigcup X$; a qD X therefore appears as a subset of $\mathcal{P}(|X|)$. The basic operation that is problematic is the union of two elements of X; in general, this union need not belong to X. We say that a, b are *compatible* when their union belongs to X; by (qD3) this is equivalent to the existence of a $c \in X$ such that $a, b \subset c$.

1.2. Definition. Let X and X' be two qualitative domains; a function F from X to X' is said to be *stable* when the following conditions are true:
(ST1) $a \subset b \in X \rightarrow F(a) \subset F(b)$.
(ST2) F commutes with directed unions: $F(\bigcup_i a_i) = \bigcup_i F(a_i)$; the directed index set I must be nonvoid, since we do not require $F(\emptyset) = \emptyset$.

[1] In other words, $a \in X$ iff all its finite subsets belong to X (using (qD3)). In fact, the only infinite points which interests us in a qD are those which are recursively enumerable.

(ST3) if $a \cup b \in X$ (i.e., if a and b are not compatible), then $F(a \cap b) = F(a) \cap F(b)$.[2]

Conditions (ST1) and (ST2) are the analogues for qD's of familiar requirements in the context of Scott domains; (ST3) is the analogue for qD's of Berry's stability condition, and this is why we call our functions 'stable'. These three conditions are very natural if we view X, X' as categories, because then F appears as a functor (condition (ST1)) preserving direct limits (condition (ST2)) and pull-backs (condition (ST3)).

1.3. Theorem (Normal Form Theorem). *If F is a stable function from X to X', if $a \in X$ and $z \in F(a)$, then:*

(i) *it is possible to find $a' \subset a$, a' finite such that $z \in F(a')$, and*

(ii) *if a' is chosen minimal such that* (i) *holds, then a' is unique.*

Proof. (i) This follows because a is the direct union of its finite subsets: simply apply (ST2).

(ii) If a' is minimal and $b \subset a$ is such that $z \in F(b)$, then a' and b are compatible. So, by (ST3), $F(a' \cap b) = F(a') \cap F(b)$, thus $z \in F(a' \cap b)$ which forces $a' \subset b$; so, a' is minimum. □

1.4. Theorem (Representation Theorem). (i) *If F is a stable function from X to X', we can define the set*

$$\mathrm{Tr}(F) = \{(a, z);\ a \in X, \text{ a } \textit{finite}, z \in |X'|, z \in F(a), \textit{ and } z \notin F(a') \textit{ for all } a' \subsetneq a\}.$$

Then F is completely determined by $\mathrm{Tr}(F)$, *by means of the equation*

$$F(b) = \{z \in |X'|;\ \exists a \subset b, (a, z) \in \mathrm{Tr}(F)\}.$$

(ii) *The set of all* $\mathrm{Tr}(F)$, *when F varies through stable functions from X to X', is a* qD, *denoted $X \Rightarrow X'$.*

Proof. (i) This is just the Normal Form Theorem 1.3.

(ii) Define $X \Rightarrow X'$ to consist of all sets A such that:

(FS1) $x \in A \to x$ is a pair (a, z) with a finite in X and $z \in |X'|$,

(FS2) given any finite $b \subset X$, then $\{z \in |X'|;\ \exists a \in X, (a, z) \in A\} \in X'$, and

(FS3) if (a, z), $(a', z) \in A$ and a, a' are compatible, then $a = a'$.

It is plain that $X \Rightarrow X'$ is a qualitative domain. Moreover, every set $\mathrm{Tr}(F)$ fulfills (FS1)-(FS3). It remains to show that any $A \in X \Rightarrow X'$ is of the form $\mathrm{Tr}(F)$ for a suitable F: given $A \in X \Rightarrow X'$, define F by

$$F(b) = \{z \in |X'|;\ \exists a \subset b, (a, z) \in A\}.$$

If b is finite, then $F(b)$ belongs to X' by (FS2). Moreover, for an arbitrary b in X,

$$F(b) = \bigcup \{F(b');\ b' \subset b,\ b' \text{ finite}\},$$

[2] It is enough to state the condition for a, b finite.

which follows from (FS1). Since F is clearly increasing, the directed union of the $F(b')$'s belongs to X': hence, F maps X into X', and fulfills (ST1) and (ST2).

It remains to prove (ST3). But, if b and b' are compatible and $z \in F(b) \cap F(b')$, then we get $(a, z), (a', z) \in A$ such that $a \subset b$, $a' \subset b'$; but a and a' must be compatible, so $a = a'$ by (FS3), and $a = a' \subset b \cap b'$. Thus, we get $z \in F(b \cap b')$, i.e., $F(b) \cap F(b') \subset F(b \cap b')$, which is the nontrivial half of (ST3). Hence, F is stable.

It now remains to compute the trace of F. It is immediate that $\mathrm{Tr}(F) = A$ so we are done. □

1.5. Remarks. (i) $|X \Rightarrow X'| = X_{\mathrm{fin}} \times |X'|$. It can be convenient to use a sequential notation for elements of $|X \Rightarrow X'|$: instead of the pair (a, z) one can use $a \vdash z$, or even $x_1, \ldots, x_n \vdash z$, with $a = \{x_1, \ldots, x_n\}$; if one uses the latter notation (which should be viewed as the intuitionistic sequent "if $x_1, \ldots, x_n$, then z"), remember that there is no order between $x_1, \ldots, x_n$, so that, for instance, $x_1, x_2 \vdash z$ is the same as $x_2, x_1 \vdash z$.

(ii) It should be clear that condition (ST3) has to be verified for finite a's and b's only.

1.6. Examples. (i) Let 1 be the qualitative domain consisting of $\emptyset$ and $\{0\}$; then there are three stable functions from 1 to itself, namely:

$$F_1(\emptyset) = F_1(\{0\}) = \emptyset, \qquad F_2(\emptyset) = \emptyset, \quad F_2(\{0\}) = \{0\},$$

$$F_3(\emptyset) = F_3(\{0\}) = \{0\}.$$

Their respective traces are $\mathrm{Tr}(F_1) = \emptyset$, $\mathrm{Tr}(F_2) = \{0 \vdash 0\}$, and $\mathrm{Tr}(F_3) = \{\vdash 0\}$.

(ii) If X is a qD, then the identity map from X to itself is clearly stable; the Normal Form Theorem 1.3 for Id^X is as follows: if $z \in \mathrm{Id}^X(a) = a$, then z already belongs to $\mathrm{Id}^X(\{z\}) = \{z\}$, i.e., $\mathrm{Tr}(\mathrm{Id}^X) = \{z \vdash z\,;\, z \in |X|\}$.

(iii) If X and Y are qD's, one can define a map f_b from $X \Rightarrow Y$ into Y by $f_b(A) = \{z\,;\, \exists a \subset b,\ (a, z) \in A\}$, for any $b \in X$. This map is stable, and its trace consists of all pairs $(\{(a, z)\}, z)$ such that $a \in X$, a is a finite subset of b, and $z \in |Y|$. Hence, f_b can be viewed as an element of $(X \Rightarrow Y) \Rightarrow Y$; the map which associates f_b to b is stable itself: its trace consists of all tuples $(a, (\{(a, z)\}, z))$ with $a \in X$ finite, and $z \in Y$.

1.7. Definition. The *order of Berry* is defined as follows. Let F, G be stable functions from X to X'; $F \subset G$ means that

$$\forall a, b \in X \quad (a \subset b \to F(a) = F(b) \cap G(a)).^3$$

Another equivalent formulation is

$$\forall a, b \in X \quad (a, b \text{ compatible} \to F(a \cap b) = F(a) \cap G(b)).$$

[3] It suffices to consider the particular case where a and b are finite.

Remark that $F \subset G$ implies $F(a) \subset G(a)$ for all a (in the above definition, take $a = b$). But, the reverse is false; typically, the inclusion $F_2 \subset F_3$ fails (see Example 1.6(i)) while $F_2(a) \subset F_3(a)$ for all a: $F_2(\emptyset) = \emptyset$, but $F_2(\{0\}) \cap F_3(\emptyset) = \{0\}$. In terms of categories, $F \subset G$ means that there is a cartesian natural transformation from F to G.

1.8. Proposition. *$F \subset G$ iff* $\mathrm{Tr}(F) \subset \mathrm{Tr}(G)$.

Proof. (i) Assume that $F \subset G$, and let $(a, z) \in \mathrm{Tr}(F)$. Then, $z \in F(a) \subset G(a)$. In order to show that $(a, z) \in \mathrm{Tr}(G)$, we assume that $z \in G(a')$ for $a' \subset a$. Then, $F(a') = F(a) \cap G(a')$, so $z \in F(a')$, thus $a' = a$.

(ii) Assume conversely that $\mathrm{Tr}(F) \subset \mathrm{Tr}(G)$, and that $b' \subset b \in X$. If $z \in F(b) \cap G(b')$, this proves that $(a', z) \in \mathrm{Tr}(G)$, $(a, z) \in \mathrm{Tr}(F)$ for some $a' \subset b'$, $a \subset b$; but then, (a', z) and (a, z) belong to $\mathrm{Tr}(G)$, and since a and a' are compatible, $a = a'$; so, $z \in F(b')$. The reversed inclusion is immediate. □

1.9. Remarks. (i) If F and G are stable functions from X to X', then one can define a stable function $F \cap G$ from X to X' by $(F \cap G)(a) = F(a) \cap G(a)$; it is immediate that $\mathrm{Tr}(F \cap G) = \mathrm{Tr}(F) \cap \mathrm{Tr}(G)$.

(ii) If (F_i) is a family of stable functions from X to X', indexed by a nonvoid directed set I, such that $i \leqslant j \to F_i \subset F_j$, then it is possible to define another stable function $F = \bigcup_i F_i$, by means of the equation $F(a) = \bigcup_i F_i(a)$. It is immediate that $\mathrm{Tr}(F) = \bigcup_i \mathrm{Tr}(F_i)$.

(iii) Stable functions of n arguments: if $X_1, \ldots, X_n$ are qD's, then it makes sense to speak of an n-ary stable function from $X_1, \ldots, X_n$ to Y. One has just to adapt the definition, for instance, (ST3) becomes: if a_1, b_1 are compatible, ..., and if a_n, b_n are compatible, then

$$F(a_1 \cap b_1, \ldots, a_n \cap b_n) = F(a_1, \ldots, a_n) \cap F(b_1, \ldots, b_n).$$

Stable functions of n variables have exactly the same kind of behaviour as usual stable functions; there are two equivalent ways of handling them.

(1) One can define a trace for such functions: $\mathrm{Tr}(F)$ is the set of all tuples $(a_1, \ldots, a_n, z)$ such that $a_1 \in X_1, \ldots, a_n \in X_n$, $z \in |Y|$, $z \in F(a_1, \ldots, a_n)$, and $z \in F(a'_1, \ldots, a'_n)$ for $a'_1 \subset a_1, \ldots, a'_n \subset a_n \to a'_1 = a_1, \ldots, a'_n = a_n$. Then we define the qD $(X_1, \ldots, X_n \Rightarrow Y)$ to be the set of all sets $\mathrm{Tr}(F)$, etc. and prove the analogues of our results for the unary case.

(2) One can also define the product $X_1 \times \cdots \times X_n$ to consist of all sets $a_1 \times \{1\} \cup \cdots \cup a_n \times \{n\}$ for $a_1 \in X_1, \ldots, a_n \in X_n$: this is a qD, and to any n-ary stable function F from $X_1, \ldots, X_n$ to Y we can associate F' from $X_1 \times \cdots \times X_n$ to Y by

$$F'(a_1 \times \{1\} \cup \cdots \cup a_n \times \{n\}) = F(a_1, \ldots, a_n) \qquad (*)$$

and, conversely, any stable function from $X_1 \times \cdots \times X_n$ to Y induces an n-ary stable function from $X_1, \ldots, X_n$ to Y by means of $(*)$. In fact, the respective traces of F

and F' are related by the formula

$$\mathrm{Tr}(F') = \{(a_1 \times \{1\} \cup \cdots \cup a_n \times \{n\}, z); (a_1, \ldots, a_n, z) \in \mathrm{Tr}(F)\},$$

which defines an isomorphism between $(X_1 \times \cdots \times X_n \Rightarrow Y)$ and $(X_1, \ldots, X_n \Rightarrow Y)$.

In order to handle λ-calculus, we have to take care of n-ary stable functions; the crucial tool is the following.

1.10. Notations. (i) When F is a stable function from X to Y, we use the notation $\lambda a.F(a)$ for $\mathrm{Tr}(F)$. The λ-notation is used for n-ary stable functions; for example, one can use $\lambda a.F(a, a_1, \ldots, a_n)$ to denote the trace of the function $a \rightsquigarrow F(a, a_1, \ldots, a_n)$, $a_1, \ldots, a_n$ being fixed.

(ii) When $A \in X \Rightarrow Y$ and $a \in X$, then $\mathrm{Ap}(A, a)$ denotes the result of the stable function encoded by A, at the argument a:

$$\mathrm{Ap}(A, a) = \{z \in |Y|; \exists b \subset a, (b, z) \in A\}.$$

Observe that λ and Ap are reciprocal:

$$\mathrm{Ap}(\lambda a.F(a), b) = F(b), \quad \text{'beta conversion'},$$
$$\lambda a.\mathrm{Ap}(A, a) = A, \quad \text{'eta conversion'}.$$

1.11. Theorem. (i) *The transformation consisting in associating to any $(n+1)$-ary function from $X, X_1, \ldots, X_n$ to Y, the n-ary function*

$$G(a_1, \ldots, a_n) = \lambda a.F(a, a_1, \ldots, a_n)$$

is stable: this means that the induced map on traces $(X, X_1, \ldots, X_n \Rightarrow Y)$ to $(X_1, \ldots, X_n \Rightarrow (X \Rightarrow Y))$ is stable.

(ii) *If F and G are n-ary stable functions from $X_1, \ldots, X_n$ to $X \Rightarrow Y$ and X respectively, then*

$$H(a_1, \ldots, a_n) = \mathrm{Ap}(F(a_1, \ldots, a_n), G(a_1, \ldots, a_n))$$

is a stable function from $X_1, \ldots, X_n$ to Y; moreover, the transformation constructing H from F and G is stable.

Proof. This theorem is more or less immediate. Both points contain two distinct results: first, the result of the transformation is stable, and second the transformation in turn is stable. We shall content ourselves with the expression of the action of these two operations on traces, without justification:

(i) $\mathrm{Tr}(G) = \{(a_1, \ldots, a_n, (a, z)); (a, a_1, \ldots, a_n, z) \in \mathrm{Tr}(F)\}$,

(ii) $\mathrm{Tr}(H) = \{(a_1, \ldots, a_n, z)$ such that one can find $(a'_1, \ldots, a'_n, (a, z)) \in \mathrm{Tr}(F)$ and $(a^1_1, \ldots, a^1_n, x^1), \ldots, (a^p_1, \ldots, a^p_n, x^p) \in \mathrm{Tr}(G)$, such that $a = \{x^1, \ldots, x^p\}$ and $a_i = a'_i \cup a^1_i \cup \cdots \cup a^p_i$ for $i = 1, 2, \ldots, n\}$. □

1.12. Definition. A *λ-structure* $D=(X, H, K)$ consists of:
(i) a qualitative domain X,
(ii) a stable function H from X to $X \Rightarrow X$, and
(iii) a stable function K from $X \Rightarrow X$ to X.

If $t=t[x_1, \ldots, x_n]$ is a term of λ-calculus ($x_1, \ldots, x_n$ include all free variables of t), then one defines an n-ary stable function t^*_D from x^n to X: $a_1, \ldots, a_n \mapsto t^*_D[x_1, \ldots, x_n]$ by the following inductive clauses:
- if t is x_i, then $t^*_D[a_1, \ldots, a_n]=a_i$,
- if t is $\lambda x.u[x, x_1, \ldots, x_n]$, then $t^*_D[a_1, \ldots, a_n]=K(\lambda a.u^*_D[a, a_1, \ldots, a_n])$,
- if t is $u[x_1, \ldots, x_n](v[x_1, \ldots, x_n])$, then $t^*_D[a_1, \ldots, a_n]=\mathrm{Ap}(H(u^*_D[a_1, \ldots, a_n]), v^*_D[a_1, \ldots, a_n])$.

1.13. Proposition. (i) *Let* $t[x, x_1, \ldots, x_n]$ *and* $u[x_1, \ldots, x_n]$ *be λ-terms; then* $(t[u/x])^*_D=t^*_D[u^*_D/a]$ *with obvious notations for the substitution of a term for a variable, or of a function for an argument.*
(ii) *Assume that* $t^*_D \subset t'^*_D$, *and* $u^*_D \subset u'^*_D$; *then,*

$$(\lambda x.t)^*_D \subset (\lambda x.t')^*_D, \qquad (t(u))^*_D \subset (t'(u'))^*_D.$$

The proof of this proposition is more or less immediate.

1.14. Proposition. (i) *If* $H \circ K \subset \mathrm{Id}^{X \Rightarrow X}$, *then* $((\lambda x.t)(u))^*_D \subset t[u/x]^*_D$.
(ii) *If* $K \circ H \subset \mathrm{Id}^X$, *then* $(\lambda x.t(x))^*_D \subset t^*_D$ (x *not free in* t).

Proof. (i) and (ii) are practically immediate; for instance, assume that $H \circ K \subset \mathrm{Id}^{X \Rightarrow X}$, which is by far the most interesting hypothesis: then, given $a_1 \subset b_1, \ldots, a_n \subset b_n$, all in X, we get, with $v=(\lambda x.t)(u)$,

$$\begin{aligned} v^*_D[a_1, \ldots, a_n] &= \mathrm{Ap}(H(K(\lambda a.t^*_D[a, a_1, \ldots, a_n])), u^*_D[a_1, \ldots, a_n]) \\ &= \mathrm{Ap}(\lambda a.t^*_D[a, a_1, \ldots, a_n] \\ &\qquad \cap H(K(\lambda a.t^*_D[a, b_1, \ldots, b_n])), u^*_D[a_1, \ldots, a_n]) \\ &= \mathrm{Ap}(\lambda a.t^*_D[a, a_1, \ldots, a_n], u^*_D[a_1, \ldots, a_n]) \cap v^*_D[b_1, \ldots, b_n] \\ &= (t[u/x])^*_D[a_1, \ldots, a_n] \cap v^*_D[b_1, \ldots, b_n]. \qquad \square \end{aligned}$$

1.15. Examples of λ-structures. (i) The most straightforward example consists of λ-structures (X, H, K) for which H and K are reciprocal isomorphisms: $H \circ K = \mathrm{Id}^{X \Rightarrow X}$ and $K \circ H = \mathrm{Id}^X$. In such structures, two terms which are interconvertible by means of beta- and eta-conversions must have the same interpretation.

(ii) The models mentioned in (i) are extensional, i.e., they interpret η-conversion by the identity. There are reasons to consider nonextensional models, and if we drop the assumption $K \circ H = \mathrm{Id}^X$ in (i), we get nonextensional models, which interpret β-conversion by the identity. Moreover, by choosing a model in which $K \circ H \subset \mathrm{Id}^X$, we can make η-conversion increasing: $t^*_D \supset (\lambda x.t(x))^*_D$ when x is not free in t.

(iii) The next step is to liberalise the requirement $H \circ K = \mathrm{Id}$. The obvious choice is $H \circ K \subset \mathrm{Id}^{X \Rightarrow X}$, which has a lot of finite solutions.

In particular, the β-conversion is increasing:

$$(\lambda x.t[x])(u)^*_D \subset t[u/x]^*_D.$$

In practice, the class of λ-structures corresponding to

$$H \circ K \subset \mathrm{Id}, \qquad K \circ H \subset \mathrm{Id}$$

has nice features, because it can be shown (see point (ii) of Section 4.1) that such λ-structures can be approximated by finite λ-structures of the same class.

Also observe that it makes sense to speak of the interpretation

$$t^\beta_D = \bigcup \{u^*_D; t =/ u\} \quad [4]$$

of the Böhm tree of t in such structures, because, by the Church-Rosser property, $\{u : t =/ u\}$ is directed, so t^β_D is a direct union in X.

(iv) The absolute liberalisation, no questions asked on (X, H, K), is harder to advocate. However, observe that if we consider the reduction procedure as the execution of a program, then it is important that t^* and u^* should be different when $t =/ u$; but in an increasing interpretation as considered in (iii), a cyclic λ-term would get a constant interpretation as the reduction goes on.

We close this section with a trivial remark. It is possible to choose D such that $t^*_D = u^*_D$ implies that t and u are syntactically equal (i.e., are the same, up to the names of bound variables): choose D, H and K, a, $b \in |D|$, $a \neq b$ such that:
- H viewed as a binary function from X, X to X is injective, and $a \in H(\emptyset, \emptyset)$,
- K is injective and $b \in K(\emptyset)$.

By the way, observe that this ensures a similar property for the intrinsic interpretation t^* of Section 4:

$$\text{if } t^* = u^*, \text{ then } t = u.$$

2. Semantics of variable types

2.1. Definition. Let X, Y be qualitative domains; a morphism from X to Y is an injective function f from $|X|$ to $|Y|$ such that, for all $x_1, \ldots, x_n \in |X|$, $\{x_1, \ldots, x_n\} \in X$ iff $\{f(x_1), \ldots, f(x_n)\} \in |Y|$.

We have therefore defined a category qD whose objects are qualitative domains; the set of all morphisms from X to Y is denoted $\mathrm{qD}(X, Y)$.

[4] We use the symbol =/ to denote reduction.

If $f \in \mathrm{qD}(X, Y)$, then it is possible to define two associated stable functions:
(i) f^+ from X to Y: $f^+(a) = \{f(z); z \in b\}$,
(ii) f^- from Y to X: $f^-(b) = \{z; f(z) \in b\}$.

2.2. Proposition
(i) $f^- \circ f^+ = \mathrm{Id}^X$.
(ii) $f^+ \circ f^- \subset \mathrm{Id}^Y$.

(*Terminology*: A stable function from a qD Z to itself such that $F \subset \mathrm{Id}^Z$ is called a *projector*; hence, $f^+ \circ f^-$ is a projector of Y. The trace of a projector is a set of pairs $(\{z\}, z)$, so the square of a projector is the projector itself.)

The proof of Proposition 2.2 is immediate.

2.3. Proposition. *If* $f \in \mathrm{qD}(X, X')$ *and* $g \in \mathrm{qD}(Y, Y')$, *then one can define* $f \Rightarrow g \in \mathrm{qD}(X \Rightarrow X', Y \Rightarrow Y')$ *by* $(f \Rightarrow g)(a, z) = (f^+(a), g(z))$.

With obvious abuses of notations (*we do not distinguish between a function and its trace*), *we have*

$$(f \Rightarrow g)^+(F) = g^+ \circ F \circ f^- \quad \textit{for } F \in X \Rightarrow Y,$$
$$(f \Rightarrow g)^-(G) = g^- \circ G \circ f^+ \quad \textit{for } G \in X' \Rightarrow Y'.$$

Proof. The proof is more or less immediate. Let us for instance compute $(f \Rightarrow g)^+(F)$. Its trace consists of all pairs $(f^+(a), g(z))$, when (a, z) varies through $\mathrm{Tr}(F)$; so,

$$\begin{aligned}(f \Rightarrow g)^+(F)(b) &= \{g(z); \exists a, (a, z) \in \mathrm{Tr}(F) \text{ and } f^+(a) \subset b\} \\ &= \{g(z); \exists a, (a, z) \subset \mathrm{Tr}(F) \text{ and } a \subset f^-(b)\} \\ &= g^+(F(f^-(b))). \qquad \square\end{aligned}$$

2.4. Theorem. "$\Rightarrow$" *is a functor from* qD$\times$qD *to* qD *preserving direct limits and pull-backs.*

Proof. X is a *subdomain* of Y when $|X| \subset |Y|$ and $X = Y \cap \mathcal{P}(|X|)$. In other words, X is a subdomain of Y when $|X| \subset |Y|$ and the inclusion map from $|X|$ into $|Y|$ is a morphism. It is convenient to translate questions of limits in the category of domains in terms of the subdomain relation:

(i) Assume that $(X_i)_{i \in I}$ is a family of qualitative domains, indexed by a nonvoid directed set I, and such that $i \leqslant j \to X_i$ subdomain of X_j. Then it is possible to define a qD $\bar{\bigcup} X_i$ as follows: $|\bar{\bigcup} X_i| = \bigcup |X_i|$; $a \subset |\bar{\bigcup} X_i|$ is an element of $\bar{\bigcup} X_i$ iff, given any *finite* $b \subset a$, $b \in \bigcup X_i$.

It is easy to see that $\bar{\bigcup} X_i$ is a qD, that all X_i's are subdomains of $\bar{\bigcup} X_i$, and that $\bar{\bigcup} X_i$ is the smallest (w.r.t. the subdomain relation) Y such that all X_i's are subdomains of Y.

For those with some experience of categories it should be clear that preservation of direct limits just means commutation with the operator $\bar{\bigcup}$:

$$(\bar{\bigcup} X_i) \Rightarrow (\bar{\bigcup} Y_i) = \bar{\bigcup} (X_i \Rightarrow Y_i).$$

The verification is left to the reader, but observe that the directedness of I is essential.

(ii) Assume that X, Y are two subdomains of Z; then, $X \cap Y$ is again a subdomain of Z. Once more, the reader with some experience of categories will guess that preservation of pull-backs is just the property

$$(X \cap X') \Rightarrow (Y \cap Y') = (X \Rightarrow Y) \cap (X' \Rightarrow Y')$$

where X, X' are subdomains of X'', and Y, Y' are subdomains of Y''. Once more, the verification is quite obvious. □

2.5. Theorem (Normal Form Theorem for Variable Types). *Let $\mathfrak{T}$ be a functor from* qD *to itself preserving direct limits and pull-backs. Let X be a* qD *and let $x \in |\mathfrak{T}(X)|$; then there is a finite* qD X_0 *and a morphism $f \in \mathrm{qD}(X_0, X)$ and $x_0 \in |\mathfrak{T}(X_0)|$ such that:*

(i) $x = \mathfrak{T}(f)(x_0)$ (*the* ***normal form of*** *x, w.r.t. $\mathfrak{T}$ and X_0), and*

(ii) *given any* qD Y, *any $f' \in \mathrm{qD}(Y, X)$ and any $y \in |\mathfrak{T}(Y)|$ such that $x = \mathfrak{T}(f')(y)$, there is a unique $h \in qD(X_0, Y)$ such that*

$$y = \mathfrak{T}(h)(x_0) \quad \textit{and} \quad f = f'h.$$

Proof. Any qD is the direct union of its finite subdomains. Hence, equation (i) has a solution, taking X_0 to be a finite subdomain of X, f to be the inclusion map from $|X_0|$ to $|X|$. Moreover, we can assume that X_0 has been chosen minimal w.r.t. the subdomain relation. Then, we prove that (ii) holds: choose Y, f' and y such that $x = \mathfrak{T}(f')(y)$. Let X_1 be a qD and let $g \in \mathrm{qD}(X_1, X_0)$, $g' \mathrm{qD}(X_1, Y)$ be such that the diagram

$$\begin{array}{ccc} X_1 & \xrightarrow{\ g\ } & X_0 \\ {\scriptstyle g'}\downarrow & & \downarrow{\scriptstyle f} \\ Y & \xrightarrow[\ f'\ ]{} & X \end{array}$$

is cartesian, i.e., is a pull-back diagram. Without loss of generality, we can assume that X_1 is a subdomain of X_0, i.e., that g is the inclusion map between $|X_1|$ and $|X_0|$. To say that the diagram is cartesian means that $fg = f'g'$ and $\mathrm{rg}(fg) = \mathrm{rg}(f) \cap \mathrm{rg}(f')$. By preservation of pull-backs, the diagram

$$\begin{array}{ccc} \mathfrak{T}(X_1) & \xrightarrow{\ \mathfrak{T}(g)\ } & \mathfrak{T}(X_0) \\ {\scriptstyle \mathfrak{T}(g')}\downarrow & & \downarrow{\scriptstyle \mathfrak{T}(f)} \\ \mathfrak{T}(Y) & \xrightarrow[\ \mathfrak{T}(f')\ ]{} & \mathfrak{T}(X) \end{array}$$

is in turn cartesian, i.e., $\mathrm{rg}(\mathfrak{T}(gf)) = \mathrm{rg}(\mathfrak{T}(f)) \cap \mathrm{rg}(\mathfrak{T}(f'))$, and since $x \in \mathrm{rg}(\mathfrak{T}(f)) \cap \mathrm{rg}(\mathfrak{T}(f'))$, it turns out that $x \in \mathrm{rg}(\mathfrak{T}(fg))$, so, since X_0 has been chosen minimal,

this forces $X_1 = X_0$, and g to be the identity. But then, $y = \mathfrak{T}(g')(x_0)$. The condition $f = f'g'$ has only one solution in g'. □

2.6. Remarks. (i) Theorem 2.5 is the adaptation to the category qD of various normal form theorems obtained by the present author, for example for dilators, but also for normal functors [6]. Theorem 2.5 is much simpler than the corresponding result for normal functors, but more complex than the ultimate simplification of Theorem 1.4: what makes the difference between Theorems 1.4 and 2.5 is that in the former case (stability), the only morphisms are inclusions. Of course, if we have more morphisms, we get more expressions of the form $x = \mathfrak{T}(f)(y)$ which makes unicity requirements more difficult to fulfill, but in turn the functor is defined by means of a smaller set of data, i.e., a smaller trace.

(ii) A typical example to which we can apply the Normal Form Theorem 1.3 is the functor $\mathfrak{T}(X) = X \Rightarrow X$, $\mathfrak{T}(f) = f \Rightarrow f$. For instance, if $0, 1, 2 \in |X|$ and $\mathcal{P}(\{0, 1, 2\}) \subset X$, then $(\{0, 1\}, 2) \in \mathfrak{T}(X)$ and we can therefore write the normal form $(\{0, 1\}, 2) = \mathfrak{T}(f)(\{0, 1\}, 2)$ where f is the inclusion map between $3 = \{0, 1, 2\}$ and $|X|$. But, consider $f' \in \mathrm{qD}(\mathcal{P}(3), X)$ defined by $f'(0) = 1, f'(1) = 0, f'(2) = 2$; it is easy to see that

$$(\{0, 1\}, 2) = (\{1, 0\}, 2) = \mathfrak{T}(f')(\{0, 1\}, 2),$$

i.e., in the Normal Form Theorem, the function f is not uniquely determined. This corresponds to nonpreservation of kernels. In another paper we shall give an unexpected explanation of this phenomenon, but this is somewhat outside the scope of this paper.

2.7. Definition. (i) A *variable type* T is (as expected) a functor from qD to qD preserving direct limits and pull-backs.

(ii) If T is a variable type, an *object of variable type* T is a family $t = (t(X))$ indexed by all qualitative domains X, such that $t(X) \in T(X)$ for all X, and enjoying the mutilation property: for all X, Y and $f \in \mathrm{qD}(X, Y)$, $t(X) = T(f)^{-}(t(Y))$.

This definition, which is simple transposition of the author's concept of *mutilation* in proofs (see, e.g., *Proceedings Warsaw ICM 1983*) can also be rewritten using $T(f)^{+}$, but the formulation is less manageable:

$$T(f)^{+}(t(X)) = t(Y) \cap \mathrm{rg}(T(f)).$$

Variable objects and types are the obvious candidates to interpret quantification on types; to prove the adequacy of this idea, we have to represent a variable type T by a qD $\mathrm{Tr}(T)$ (the *trace* of T, obtained through the Normal Form Theorem) in such a way that the objects of variable type T will correspond to the elements of $\mathrm{Tr}(T)$.

2.8. Definition. Let T be a variable type; a *trace* of T is any set A formed of pairs (X, x) such that:

(i) X is a finite qD and $x \in |T(X)|$,

(ii) given any qD Y and any $y \in |T(Y)|$, there is a unique (X, x) in A and a morphism f (in general nonunique) such that y has the normal form $y = T(f)(x)$.

We shall use the notation $A = \|\mathrm{Tr}(T)\|$ which may seem ambiguous since there are many possible choices for A. In fact, $\|\mathrm{Tr}(T)\|$ is the set of all equivalence classes of normal forms, and we have preferred to pick up an element in each equivalence class. The way we use the trace will show that our abuse of notations is harmless.

The qD $\mathrm{Tr}(T)$ is defined as a subset of $\mathcal{P}(\|\mathrm{Tr}(T)\|)$. A subset a of $\|\mathrm{Tr}(T)\|$ belongs to $\mathrm{Tr}(T)$ exactly when the following holds: take $(X_0, x_0), \ldots, (X_n, x_n)$ in a, take a finite qD X and morphisms $f_0 \in \mathrm{qD}(X_0, X), \ldots, f_n \in \mathrm{qD}(X_n, X)$; then, $F(f_0)(x_0), \ldots, F(f_n)(x_n) \in F(X)$.

In this definition, $(X_0, x_0), \ldots, (X_n, x_n)$ are not necessarily distinct; so even when a is a singleton, the condition is not always fulfilled (i.e., there are phenomena of 'self-incompatibility'), and so in general $|\mathrm{Tr}(T)|$ is strictly included in $\|\mathrm{Tr}(T)\|$ (see, for instance, Theorems 3.13, 3.14, and 3.15).

2.9. Theorem. *There is a canonical bijection between* $\mathrm{Tr}(T)$ (*where T is a variable type*) *and the set of all objects of variable type T; the bijection is as follows:*

(i) *to t of variable type T, associate*

$$\mathrm{Tr}(t) = \{(X, x) \in \|\mathrm{Tr}(T)\| ; x \in t(X)\};$$

(ii) *to $a \in \mathrm{Tr}(T)$, associate the function*

$$a\{Y\} = \{T(f)(x) ; (X, x) \in a, f \in \mathrm{qD}(X, Y)\}.$$

Proof. If t is of variable type T, then $\mathrm{Tr}(t)$ is a subset of $\|\mathrm{Tr}(T)\|$. Now, if $(X_0, x_0), \ldots, (X_n, x_n) \in \mathrm{Tr}(t)$, if $f_0 \in \mathrm{qD}(X_0, X), \ldots, f_n \in \mathrm{qD}(X_n, X)$, then $T(f_0)(x_0), \ldots, T(f_n)(x_n) \in t(X)$, because $T(f_i)^+$ maps $t(X_i)$ into $t(X)$; so, $\{T(f_0)(x_0), \ldots, T(f_n)(x_n)\} \subset t(X) \in T(X)$. Hence, $\mathrm{Tr}(t) \in \mathrm{Tr}(T)$.

Conversely, let $a \in \mathrm{Tr}(T)$, and define in general $a\{Y\}$ as explained above. The definition of $\mathrm{Tr}(T)$ implies that $a\{Y\} \in T(Y)$ when Y is finite, and, by a direct limit argument, $a\{Y\} \in T(Y)$ for all Y.

Now take Y, Z and $g \in \mathrm{qD}(Y, Z)$; then,

$$T(g)^-(a\{Z\}) = \{T(g)^{-1}T(f)(x) ; (X, x) \in a, f \in \mathrm{qD}(X, Z)\}.$$

Now, among all points of the form $T(g)^{-1}T(f)(x)$, we have of course the points $T(h)(x)$ (with $h \in \mathrm{qD}(X, Y)$): take $f = gh$. But, conversely, all points of the form $T(g)^{-1}T(f)(x)$ can be written as $T(h)(x)$ for some $h \in \mathrm{qD}(X, Y)$. Write the normal form of $T(g)^{-1}T(f)(x)$: $T(k)(x')$ for some X', some $x' \in T(X')$ and some $k \in \mathrm{qD}(X', Y)$. But then, $T(f)(x) = T(g)T(k)(x') = T(gk)(x')$. If the pair (X', x') has been chosen in $\|\mathrm{Tr}(T)\|$, as is always possible, then necessarily $X' = X$, $x' = x$, and

one can take $h = k$. Summing up, we have just established that $T(g)^{-}(a\{Z\}) = a\{Y\}$, i.e., the family $(a\{Y\})$ defines an object of variable type T. The fact that the processes (i) and (ii) are inverse is more or less immediate. □

2.10. Example. Let T be the variable type of Remark 2.6: then, if one defines $t(X) = \mathrm{Tr}(\mathrm{Id}^X) = \{(\{x\}, x); x \in |X|\}$, it is immediate that t is an object of variable type T. Moreover, $\mathrm{Tr}(t) = \{(1, (\{0\}, 0))\}$ where 1 denotes the qD $\{\emptyset, \{0\}\}$. This example shows that the uniform identity of system F has a finite interpretation!

2.11. Remark. The inclusion $\mathrm{Tr}(t) \subset \mathrm{Tr}(t')$ between objects of the same variable type T corresponds to the relation:

$$t \subset t' \text{ iff for all (finite) qD } X,\ t(X) \subset t'(X).$$

Similarly, the object t'' defined by $\mathrm{Tr}(t'') = \mathrm{Tr}(t) \cap \mathrm{Tr}(t')$, where t and t' are two objects of the same variable type T, satisfies

$$t''(X) = t(X) \cap t'(X).$$

The union of a directed family (t_i) (w.r.t. inclusion) of objects of variable type T can be defined by $t(X) = \bigcup_i t_i(X)$ and $\mathrm{Tr}(t) = \bigcup_i \mathrm{Tr}(t_i)$.

2.12. Example. Let $T(X) = (X \Rightarrow X) \Rightarrow X$, and let $t_n(X)$ be the following objects of variable T: if $F \in X \Rightarrow X$, then $t_n(X)(F) = F(F(\ldots(F(\emptyset))\ldots))$ (n times F). We leave the following verifications to the reader:

(i) t_n is a variable object of type T,

(ii) $t_n \subset t_{n+1}$ for all n.

(This can be obtained from our interpretation for system F: add a constant $\emptyset^\sigma$ of any type σ, and interpret it by $\emptyset$, as expected; then t_n is $\Lambda\alpha\lambda x^{\alpha \Rightarrow \alpha}.x(x(\ldots(x(\emptyset))\ldots))$. Clearly, $t_0^* \subset t_1^*$, from which we get $t_n^* \subset t_{n+1}^*$ for all n, etc.)

One can define the variable object fp of variable type T by $\mathrm{fp} = \bigcup_n t_n$, and it is clear that $F(\mathrm{fp}(X)(F)) = \mathrm{fp}(X)(F)$ for any X and $F \in X \Rightarrow X$.

2.13. Notation. (i) It is necessary to consider variable types in n arguments, i.e., functors from qD^n to qD preserving direct limits and pull-backs. A family $(t(X_1, \ldots, X_n))$ indexed by all n-uples of qD's $X_1, \ldots, X_n$, and such that $t(X_1, \ldots, X_n) \in T(X_1, \ldots, X_n)$ for all $X_1, \ldots, X_n$, is said to be an object of variable type T when the following holds:

$$\text{given } f_1 \in \mathrm{qD}(X_1, Y_1), \ldots, f_n \in \mathrm{qD}(X_n, Y_n),$$
$$T(f_1, \ldots, f_n)^{-}(t(Y_1, \ldots, Y_n)) = t(X_1, \ldots, X_n).$$

The ordering between objects of variable type T is, as expected:

$$t \subset t' \text{ iff for all } X_1, \ldots, X_n\text{: } t(X_1, \ldots, X_n) \subset t'(X_1, \ldots, X_n).$$

As usual, it is enough to restrict ourselves to finite $X_1, \ldots, X_n$. Intersections and unions are defined in analogy with Remark 2.11.

It could be of some interest to introduce the concept of trace of an n-ary variable type; for instance, $\|\mathrm{Tr}(T)\|$ consists of tuples $(X_1, \ldots, X_n; z)$ with $z \in T(X_1, \ldots, X_n)$, etc. The details are left to the reader.

(ii) When T is a unary variable type, one can define $\Lambda X.T(X)$ as $\mathrm{Tr}(T)$; when t is an object of variable type T, one can define $\Lambda X.t(X)$ as $\mathrm{Tr}(t) \in \Lambda X.T(X)$. The notation is used to denote the action of abstracting from one argument in the $(n+1)$-ary case: from $T(X, X_1, \ldots, X_n)$ or from $t(X, X_1, \ldots, X_n)$, construct

$$\Lambda X.T(X, T_1, \ldots, X_n) \quad \text{and} \quad \Lambda X.t(X, X_1, \ldots, X_n),$$

which denote the respective traces of the unary variable type and object obtained by fixing the values $X_1, \ldots, X_n$.

In fact, $\Lambda X.T(X, X_1, \ldots, X_n)$ is a variable type in the n arguments $X_1, \ldots, X_n$: if $f_1 \in \mathrm{qD}(X_1, Y_1), \ldots, f_n \in \mathrm{qD}(X_n, Y_n)$, consider $(X, x) \in \|\mathrm{Tr}(T(\cdot, X_1, \ldots, X_n))\|$; then, $T(X, f_1, \ldots, f_n)(x) \in T(X, Y_1, \ldots, Y_n)$. Now, $T(X, f_1, \ldots, f_n)(x)$ has a normal form $T(g, Y_1, \ldots, Y_n)(y)$ with $g \in \mathrm{qD}(Y, X)$ and $(Y, y) \in \|\mathrm{Tr}(T(\cdot, Y_1, \ldots, Y_n))\|$. We set

$$\Lambda X.T(X, f_1, \ldots, f_n)(X, x) = (Y, y).$$

It is easy to prove the existence of an isomorphism $h \in \mathrm{qD}(X, Y)$ such that $y = T(h, f_1, \ldots, f_n)(x)$ and, from this, it easily follows that $\Lambda X.T(X, \cdot, \ldots, \cdot)$ is a variable type. Now, if t is of variable type T, then $\Lambda X.t(X, \cdot, \ldots, \cdot)$ is easily seen to be of variable type $\Lambda X.T(X, \cdot, \ldots, \cdot)$.

(iii) When T is a unary variable type and $t \in \Lambda X.T(X)$ and Y is a qD, then $\mathrm{Ext}(t, Y)$ denotes the element of $T(Y)$ defined by

$$\mathrm{Ext}(t, Y) = \{T(f)(x); (X, x) \in t, f \in \mathrm{qD}(X, y)\}.$$

It is easy to see that if t is of variable type $\Lambda X.T(X, \cdot, \ldots, \cdot)$ and U is an n-ary variable type, then $\mathrm{Ext}(t(X_1, \ldots, X_n), U(X_1, \ldots, X_n))$ is of variable type V, with $V(X_1, \ldots, X_n) = T(U(X_1, \ldots, X_n), X_1, \ldots, X_n)$, etc.

Finally, observe that Ext and $\Lambda X.$ are reciprocal:

$$\mathrm{Ext}(\Lambda X.t(X), Y) = t(Y), \qquad \Lambda X.\mathrm{Ext}(t, X) = t.$$

3. The system *F*

The system F is based on the idea of variable types, which is now a very commonplace idea in computer science. The system was introduced by the present author in [3], and is defined as follows (we have reduced the formalism to the schemes corresponding to $\Rightarrow$ and Λ, and we have also slightly changed the symbols used, to conform to more current traditions).

3.1. Definition. The *types* of F are those that can be generated by the following clauses:

(i) the *type variables* α, β, γ etc. are types,

(ii) if σ and τ are types, then $\sigma \Rightarrow \tau$ is a type,

(iii) if σ is a type, and α is a type variable, then $\Lambda\alpha.\sigma$ is a type. The variable α is bound in $\Lambda\alpha.\sigma$.[5]

Examples of types are $\Lambda\alpha.(\alpha \Rightarrow \alpha)$ and $\Lambda\alpha.(\alpha \Rightarrow ((\alpha \Rightarrow \alpha) \Rightarrow \alpha))$.

3.2. Definition. We inductively define the concept of a *term of type* σ, where σ is a type of F: a term is anything that can be obtained by the following clauses:

(i) For any type σ, the *variables of type* σ, x^σ, y^σ, z^σ etc., are terms of type σ.

(ii) If t is a term of type τ and x^σ is a variable of type σ, then $\lambda x.t$ is a term of of type $\sigma \Rightarrow \tau$; the variable x is bound in $\lambda x.t$.

(iii) If t and u are terms of respective types $\sigma \to \tau$ and σ, then $\mathrm{Ap}(t, u)$ (often abbreviated as $t(u)$) is a term of type τ.

(iv) If t is a term of type σ and α is a type variable, then $\Lambda\alpha.t$ is a term of type $\Lambda\alpha.\sigma$. The construction is subject to the obvious restriction that, if a variable x of type τ occurs freely in t, then α does not occur freely in τ. The variable α is bound in $\Lambda\alpha.t$.

(v) If t is a term of type $\Lambda\alpha.\sigma$ and τ is a type, then $\mathrm{Ext}(t, \tau)$ (often abbreviated as $t\{\tau\}$) is a term of type $\sigma[\tau/\alpha]$.

3.3. Examples. (i) $\Lambda\alpha\lambda x^\alpha.x^\alpha$ is a term of type $\Lambda\alpha.\alpha \Rightarrow \alpha$.

(ii) $\Lambda\alpha.\lambda x^\alpha.\lambda y^{\alpha \Rightarrow \alpha}.y(y(\ldots y(x)\ldots))$ is a term of type $\Lambda\alpha.\alpha \Rightarrow ((\alpha \Rightarrow \alpha) \Rightarrow \alpha)$. This term denotes the integer n, where n is the number of occurrences of y after the $\lambda y.$; hence, we have terms $\bar{n}$ of type $\mathbf{int} = \Lambda\alpha.\alpha \Rightarrow ((\alpha \Rightarrow \alpha) \Rightarrow \alpha)$, and it is easy to check that the only normal (see below) closed terms of type **int** are the $\bar{n}$'s.

3.4. Definition. We define immediate reduction by

$$(\lambda x.t[x])(u) =/_i t[u/x], \qquad (\Lambda\alpha.t[\alpha])\{\tau\} =/_i t[\tau/\alpha].$$

Then we define reduction to be the smallest transitive relation containing immediate reduction, and compatible with the formation of terms, for example, if $t =/ u$, then $t\{\tau\} =/ u\{\tau\}$. The proof of the Church-Rosser property for usual λ-calculus can be adapted without problems to F.

3.5. Examples. (i) $((\Lambda\alpha.(\lambda x^\alpha.x^\alpha))\{\tau\})(t) =/ t$ when t is of type τ.

(ii) If t and u are of respective types $\tau \Rightarrow \tau$ and τ, then $\bar{n}\{\tau\}(t)(u) =/ t(t(\ldots t(u)\ldots))$ (n times t).

[5] Let us advocate the choice of the symbol "Λ": it denotes both a sort of λ-abstraction (capital λ), but also a universal quantifier (although the symbol "$\forall$" is far more common).

3.6. Definition. (i) A term is said to be *normal* if no immediate reduction can be done on any of its subterms. The terms $\bar{n}$ are normal, and they are the only normal terms of type **int** which are closed: this is left as an easy exercise to the reader.

(ii) Let t be a closed term of type **int⇒int**; then t induces a partial recursive function from $\mathbb{N}$ to $\mathbb{N}$, defined as follows:

$$|t|(n) \simeq m \text{ iff } t(\bar{n}) =/ \bar{m}.$$

By the Church-Rosser property, if $t(\bar{n})$ has the *normal form* $\bar{m}$, then $\bar{m}$ is unique; the question is of course the existence of m.

3.7. Theorem (Girard [3]). (i) *In F, all terms have a normal form. In particular, the functions* $|t|$ *of Definition* 3.6(ii) *are total recursive functions.*

(ii) *The class of all functions from* $\mathbb{N}$ *to* $\mathbb{N}$ *which are of the form* $|t|$ *is exactly the class of graphs of all provably total recursive functions of second-order arithmetic* PA_2.

Proof. It would be a waste of time to reproduce here the original proofs. (For the reader who does not read French, let us mention that the proofs of these results have been often redone in the current literature of the subject, by Leivant, Statman etc.; see [7] for a bibliography). By a diagonalization argument, (ii) implies that (i) is not provable in PA_2; in fact, the method used to prove (i), 'candidats de réductibilité', uses a notion of 'calculability' which is not expressible in PA_2. However, if one restricts the schema of Definition 3.2(v) to finitely many types τ_i (and the types obtained from them by substitution), then the theorem is provable in PA_2, and this is why the functions $|t|$ are provably total in PA_2; this gives one half of (ii), by far the most difficult part. The strength of the system essentially lies in the schema of Definition 3.2(v), and we get stronger systems as we allow more types τ in this schema. (The proof that this schema preserves 'calculability', uses a comprehension axiom to define the set of all computable terms of type τ.) There is little hope of finding a direct combinatorial argument, because the method cannot be formalisable in PA_2. Up to now, the original proof (or its straightforward variants, e.g., to ensure strong normalisation) is the only method to prove (i). The remaining part of (ii) can be proved by various means: the original proof used an extension of Gödel's functional interpretation to Heyting's second-order arithmetic HA_2. Soon afterwards, in an unpublished manuscript, Per Martin-Løf gave a simpler argument involving a notion of realisability by means of terms of F. □

3.8. Remark. For the readers familiar with natural deduction, the Curry-Howard-De Bruijn isomorphism can be done for F: the types are the formulas of (intuitionistic) second-order propositional calculus, and the terms denote deductions of their types, under hypotheses which are the types of their free variables. For instance, the schemes of Definition 3.2(iv), (v) respectively correspond to

$$\frac{\begin{array}{c}\vdots\\ \sigma[\alpha]\end{array}}{\Lambda\alpha.\sigma[\alpha]} \quad \text{and} \quad \frac{\begin{array}{c}\vdots\\ \Lambda\alpha.\sigma[\alpha]\end{array}}{\sigma[\tau]}$$

which are the obvious quantifier rules for second-order propositional calculus. The reduction in F corresponds to the obvious normalisation procedure for second-order propositional calculus.

Second-order logic à la Takeuti is practically the same system, except that variables of propositions are replaced by variables for n-ary predicates, and that a 'first-order part' is added. The first-order part adds absolutely no difficulty, and this is why [3] also contains a proof of the syntactic form of Takeuti's conjecture, which is just a normalisation for second-order intuitionistic logic.

3.1. The semantics of F: Discussion

The difficulty is to interpret the universal types $\Lambda\alpha.\sigma$. For the implication, we can bear in mind the naive image of a function from σ to τ, but, for the universal type, the idea of a function assigning to any type τ an object of type $\sigma[\tau]$ is not satisfactory, because $\sigma[\tau]$ is often more complicated than $\Lambda\alpha.\sigma$. In particular, the idea of interpreting F in standard set-theoretical terms, fails; this has been shown quite recently by Reynolds [7].

Up to now, there is only one standard way of interpreting F,[6] namely to refuse to consider the schemes of Definition 3.2(iv), (v). More precisely, to any term of F, say t, associate a λ-term t^-, as follows:

$$(x^\sigma)^- = x, \qquad (\lambda x^\sigma.t) = \lambda x.t^-, \qquad (t(u))^- = t^-(u^-),$$
$$(\Lambda\alpha.t)^- = t^-, \qquad (t\{\tau\})^- = t^-.$$

It is clear that $t =/ u$ implies $t^- =/ u^-$. Since there are many models for λ-calculus, one can define the interpretation of t in such a model M as the interpretation $M(t^-)$ of t^- in M. If one defines the notion of type of M as being a subset of M enjoying ad hoc closure properties, then it is easy to interpret F:

(i) If σ and τ are interpreted by A and $B \subset M$, then $\sigma \Rightarrow \tau$ is interpreted by $A \Rightarrow B$, which is the set of all $m \in M$ which apply A into B.

(ii) If, for all A, A is interpreted by $F(A)$, then the interpretation of $\Lambda\alpha.\sigma$ is just $\bigcap\{F(A); A \text{ type of } M\}$.

Then it is easy to see that $M(t^-)$ belongs to the interpretation of the type of t.

We consider this interpretation as unsatisfactory, because:

(i) it does not interpret the types: it simply ignores them; this is just an interpretation of the underlying λ-calculus, and

(ii) the interpretation depends on something rather artificial, namely a model M for λ-calculus. We would like to have an absolute interpretation and not to be forced to restrict ourselves to a fixed list of types; for instance, the uniform identity can be evaluated on any structure of a given kind, etc.

[6] From McCracken after Scott, Bruce, Meyer, Mitchell, and Longo. A detailed inspection shows that there is little novelty in these works w.r.t. Troelstra's model called HRO_2, and which dates back to 1971.

The reader has understood that this refusal of the straightforward interpretation of F is just a way of introducing our new interpretation, which has of course all possible virtues, etc.

If we want to interpret a variable type $\Lambda\alpha.\sigma$, we stumble on the difficulty that we must consider functions which are defined on all types, including those we have not considered yet. The method already criticised was simply to make these functions constant. But, in reality, if we consider the uniform identity $\Lambda\alpha.\lambda x^\alpha.x^\alpha$, this defines a function $\alpha \rightsquigarrow \lambda x^\alpha.x^\alpha$ which is not exactly constant in α. Now, the method of Section 2 enables us to say that this function is determined by its behaviour on finite qualitative domains, i.e., on 'finite types', if we identify types with qD's in our interpretation. Then there is no longer any circularity, and the interpretation can be done. We already computed it in the case of the universal identity (Example 2.10) and we just found a structure with one point. Compare with the monster that would interpret the same thing in a model forgetting the types!

Now we give the precise definition of the interpretation; all elements have been given in Sections 1 and 2, and we have just to put things together.

3.9. Definition. By $t[\boldsymbol{\alpha}, \boldsymbol{x}]$ we mean the following:

(i) $\boldsymbol{\alpha}$ is a sequence $\alpha_1, \ldots, \alpha_n$ of type variables.

(ii) $\boldsymbol{x}$ is a sequence $x_1, \ldots, x_m$ of variables of respective types $\sigma_1, \ldots, \sigma_m$; all free variables of the σ_i's are among $\alpha_1, \ldots, \alpha_n$. We use the shorthand notation $\boldsymbol{\sigma}$ for $\sigma_1, \ldots, \sigma_n$.

(iii) t is a term of type τ. The only free type variables of t (and τ) are among $\boldsymbol{\alpha}$, and the only free type variables of t are among $\boldsymbol{x}$.

It will be convenient to individualise one of the variables, for instance, we can write $t[\alpha, \boldsymbol{\alpha}, \boldsymbol{x}]$ or $t[\boldsymbol{\alpha}, y, \boldsymbol{x}]$, etc. In order to interpret $t[\boldsymbol{\alpha}, \boldsymbol{x}]$ (of type τ), we have first to take a sequence $\boldsymbol{X} = X_1, \ldots, X_n$ of qualitative domains; we can then define (see below) qD's $\tau^*[\boldsymbol{X}]$ and $\sigma_i^*[\boldsymbol{X}]$; we use $\boldsymbol{\sigma}^*[\boldsymbol{X}]$ for the sequence $\sigma_1[\boldsymbol{X}], \ldots, \sigma_m[\boldsymbol{X}]$. In fact, $\tau^*[\boldsymbol{X}]$ and the $\boldsymbol{\sigma}^*[\boldsymbol{X}]$'s will be variable types.

Then, given objects $a_1 \in \sigma_1^*[\boldsymbol{X}], \ldots, a_m \in \sigma_m^*[\boldsymbol{X}]$ (notation: $\boldsymbol{a} \in \boldsymbol{\sigma}^*[\boldsymbol{X}]$) we define the interpretation

$$t^*[\boldsymbol{X}, \boldsymbol{a}] \in \sigma^*[\boldsymbol{X}].$$

Interpretation of a type $\tau[\alpha_1, \ldots, \alpha_n] = \tau[\boldsymbol{\alpha}]$:

(i) If $\tau[\boldsymbol{\alpha}] = \alpha_i$, then $\tau^*[\boldsymbol{X}] = X_i$; if $f_1 \in \mathrm{qD}(X_1, Y_1), \ldots, f_n \in \mathrm{qD}(X_n, Y_n)$ (notation: $\boldsymbol{f} \in \mathrm{qD}(\boldsymbol{X}, \boldsymbol{Y})$), then we define $\tau^*[\boldsymbol{f}] = f_i$.

(ii) If $\tau[\boldsymbol{\alpha}] = \sigma[\boldsymbol{\alpha}] \Rightarrow \rho[\boldsymbol{\alpha}]$, then $\tau^*[\boldsymbol{X}] = \sigma^*[\boldsymbol{X}] \Rightarrow \rho^*[\boldsymbol{X}]$; if $\boldsymbol{f} \in \mathrm{qD}(\boldsymbol{X}, \boldsymbol{Y})$, then $\tau^*[\boldsymbol{f}] = \sigma^*[\boldsymbol{f}] \Rightarrow \rho^*[\boldsymbol{f}]$.

(iii) If $\tau[\boldsymbol{\alpha}] = \Lambda\alpha.\sigma[\alpha, \boldsymbol{\alpha}]$, then $\tau^*[\boldsymbol{X}] = \Lambda X.\sigma^*[X, \boldsymbol{X}] = \mathrm{Tr}(\sigma^*[\cdot, \boldsymbol{X}])$; if $\boldsymbol{f} \in \mathrm{qD}(\boldsymbol{X}, \boldsymbol{Y})$, then $\tau^*[\boldsymbol{f}] = \Lambda X.\sigma^*[X, \boldsymbol{f}] = \mathrm{Tr}(\sigma^*[\cdot, \boldsymbol{f}])$.

The general results of Section 2 show that τ^* is a variable type in n arguments.

Interpretation of a term $t[\boldsymbol{\alpha}, \boldsymbol{x}]$ *of type* $\tau[\boldsymbol{\alpha}]$:

(i) If $t[\boldsymbol{\alpha}, \boldsymbol{x}] = x_i$, then let $t^*[\boldsymbol{X}, \boldsymbol{a}] = a_i$.

(ii) If $t[\alpha, x] = \lambda y.u[\alpha, y, x]$, then let $t^*[X, a] = \lambda b.t^*[\alpha, b, a]$.

(iii) If $t[\alpha, x] = u(\alpha, x](v[\alpha, x])$, then let $t^*[X, a] = \mathrm{Ap}(u^*[X, a], v^*[X, a])$.

(iv) If $t[\alpha, x] = \Lambda\beta.u[\beta, \alpha, x]$, then let $t^*[X, a] = \Lambda Y.u^*[Y, X, a]$; this makes sense, because σ does not depend on β.

(v) If $t[\alpha, x] = u[\alpha, x]\{\sigma[\alpha]\}$, then let $t^*[X, a] = \mathrm{Ext}(u^*[X, a], \sigma^*[X])$.

3.10. Verification. One has to verify that Definition 3.9 makes sense. We only indicate the main steps, and leave the details:

(i) One has somewhere to verify that the clauses (i)-(v) above (provided they make sense) lead to variable terms of the expected types. This is immediate, but for clause (v), for which one needs the straightforward property

$$\tau^*[\sigma^*[X], X] = (\tau[\sigma/\alpha])^*[X].].$$

(ii) More seriously, one has to verify some 'stability' requirements for the interpretation; there are two such properties:

(1) When X is fixed, then $t^*[X, \cdot\,]$ is a stable function of m arguments of types $\sigma[X]$.

(2) Let $f \in \mathrm{qD}(X, Y)$; then, for all $b \in \sigma^*[X]$,

$$\tau^*[f]^-(t^*[Y, \sigma[f]^+(b)]) = t^*[X, b].$$

The verification of (1) and (2) is an uninteresting exercise.

3.11. Theorem. *The interpretation has the following properties:*

(i) *The interpretation of the schemes of Definition* 3.2(i)-(v) *preserve inclusions: to say that* $t^* \subset u^*$, *when* t *and* u *are of the same form* $t[\alpha, x]$, $u[\alpha, x]$ *means that* $a \subset b \in \sigma[X]$ *implies* $t^*[X, a] = t^*[X, b] \cap u^*[X, a]$ (*as usual: finite* X, a, b *suffice for our purposes*). *For instance, from* $t^* \subset u^*$, *one can deduce* $(t\{\sigma\})^* \subset (u\{\sigma\})^*$.

(ii) *The interpretation of the schemes of Definition* 3.2(i)-(v) *preserve equality: this is a straightforward consequence of* (i).

(iii) $((\lambda x.t[x])(u))^* = t[u/x]^*$,

$(\lambda x.t(x))^* = t^*$ *when* x *is not free in* t,

$((\Lambda\alpha.t[\alpha])\{\sigma\})^* = t[\sigma/\alpha]^*$,

$(\Lambda\alpha.t\{\alpha\})^* = t^*$ *when* α *is not free in* t.

In other words, two terms which are (β, η)*-interconvertible have by* (ii) *and* (iii) *the same interpretation.*

The proof is left to the reader.

3.12. Remark. Our interpretation, although it uses a very pretentious formalism, is extremely effective and constructive. In particular, it can be carried out in primitive recursive arithmetic without any problem: qD can be encoded by means of its finite

elements, so we can speak of a recursive qD. Moreover, the operations used to construct qD's are primitive recursive.[7] As to elements of a qD, the good taste consists in encoding them by means of a primitive recursive enumerating function, and once more, all our operations are primitive recursive in this encoding.

It is not astonishing that our interpretation is elementary: remark that it extends to a system containing the universal fixed point operator fp of type $\Lambda\alpha.(\alpha\Rightarrow\alpha)\Rightarrow\alpha$ which does not quite lead to normal forms!

Let us end this section with the computation of the traces of simple types.

3.13. Theorem. $(\Lambda\alpha.\alpha)^* = \emptyset$.

Proof. $\|\mathrm{Tr}(\Lambda\alpha.\alpha)\|$ consists of only one point: $(1, 0)$, where 1 is the qD $\{\emptyset, \{0\}\}$. We show that this point is incompatible with itself. Consider the qD $A = \{\emptyset, \{u\}, \{v\}\}$ with $u \neq v$; there are two morphisms f_u and f_v from 1 to X; the compatiblity condition for $(1, 0)$ alone requires in particular that $\{f_u(0), f_v(0)\} \in A$, but this set is $\{u, v\} \notin A$. □

3.14. Theorem. $(\Lambda\alpha.\alpha\Rightarrow\alpha)^* = \{\emptyset, \{(1, (\{0\}, 0))\}\}$.

Proof. As usual, 1 denotes the qD $\{\emptyset, \{0\}\}$. $\|\mathrm{Tr}(\Lambda\alpha.\alpha\Rightarrow\alpha)\|$ consists of all tuples $(X, (a, z))$, where X is a finite qD, $a \in X$, and $X = a \cup \{z\}$. Take such a tuple $(X, (a, z))$, and assume that $|X|$ has at least two points. Consider the set $|Y|$ obtained by duplicating all points of $|X|$, but z: we have two function $x \rightsquigarrow x'$ and $x \rightsquigarrow x''$ from $|X|$ to $|Y|$ which disagree everywhere, but for z: $z' = z'' = z$. = We extend our functions to subsets by using the same notation: $'$ and $''$. Observe that $a' \neq a''$. A subset of Y can be written as $d = b' \cup c''$, with $b' = d \cap |X|'$, $c'' = d \cap |X|''$ and we declare d to be a member of Y exactly when $b \in X$ and $c \in X$. Then, $'$ and $''$ are morphisms from X to Y. Moreover, $a' \cup a'' \in Y$, i.e., a' and a'' are compatible. Now, the compatibility of $(X, (a, z))$ itself means in particular that $\{(a', z), (a'', z)\} \in Y \Rightarrow Y$; but this is impossible since a', a'' are comparible, but distinct.

So the only self-compatible elements of $\|\mathrm{Tr}(\Lambda\alpha.\alpha\Rightarrow\alpha)\|$ are of the form $(X, (a, z))$, with $|X| = \{z\}$ and $a \in X$. Now, $a = \emptyset$ is impossible, simply by sending X into A by f_u and f_v (as in Theorem 3.13), which would lead to the inconsistent combination $\{(\emptyset, u), (\emptyset, v)\}$ in $A \Rightarrow A$.

Then $a = \{z\}$, and, up to isomorphism, we are reduced to the solution $(1, (\{0\}, 0))$. □

3.15. Theorem. $\Lambda\alpha.(\alpha\Rightarrow(\alpha\Rightarrow\alpha))$ (*boolean type*) *contains exactly four objects*: $\emptyset$ *and the singletons of* $(1, (\{0\}, (\emptyset, 0)))$ (TRUE), *of* $(1, (\emptyset, (\{0\}, 0)))$ (FALSE) *and of* $(1, (\{0\}, (\{0\}, 0)))$ (INTER).

[7] This is an exaggeration; for the precise statement, see Appendix C.

Proof. First observe that these three singletons are pairwise incompatible: for instance, if one puts together in the domain 1, FALSE and INTER, then one gets the set $(\emptyset, (\{0\}, 0), (\{0\}, (\{0\}, 0)))$, which is not the trace of any binary stable function from 1^2 to 1.

The general form of an object of $\|\Lambda\alpha.\alpha \Rightarrow (\alpha \Rightarrow \alpha)\|$ is as follows: $(X, (a, (b, z)))$, with X a finite qD, a, $b \in X$, and $X = a \cup b \cup \{z\}$. By an imitation of the proof of Theorem 3.14 one easily shows that one can reduce to the case $X = 1$, $z = 0$, and $a \neq \emptyset$ or $b \neq \emptyset$. Then the only possibilities are TRUE, FALSE, INTER.

Now we have to show that these three points are themselves self-consistent: they obviously correspond to the following functions:

$$\text{TRUE}(Y)(a)(b) = a, \qquad \text{FALSE}(Y)(a)(b) = b,$$

$$\text{INTER}(Y)(a)(b) = a \cap b,$$

which fulfill all possible stability requirements. □

3.16. Remarks. Here we see the possible role of semantics: to suggest improvements of the syntax. For instance, F has only two closed normal terms of boolean type, namely $\Lambda\alpha.\lambda x^\alpha.\lambda y^\alpha.x^\alpha$ and $\Lambda\alpha.\lambda x^\alpha.\lambda y^\alpha.y^\alpha$. These two terms are respectively interpreted by TRUE and FALSE. But there are two other objects, $\emptyset$ and INTER. There is little to say about $\emptyset$: the possibility of adding a void object of each type could be seen even before starting the interpretation. But the object INTER is unexpected: this third truth value plays the role of the undeterminated value; its adjunction to the syntax could be considered.

Let us recall that the definition by cases IF THEN ELSE is defined in F by: $\lambda x^{\text{bool}}.\Lambda\alpha.\lambda y^\alpha.\lambda z^\alpha.x\{\sigma\}(y)(z)$, i.e., IF t THEN a ELSE b, where t is of boolean type and a, b of type σ, is $t\{\sigma\}(a)(b)$. In fact, semantically, we get for the four possible values:

$$\text{IF TRUE THEN } a \text{ ELSE } b = a,$$

$$\text{IF FALSE THEN } a \text{ ELSE } b = b,$$

$$\text{IF VOID THEN } a \text{ ELSE } b = \emptyset,$$

$$\text{IF INTER THEN } a \text{ ELSE } b = a \cap b.$$

If one defines $\text{NOT} = \lambda z^{\text{bool}}.\Lambda\alpha.\lambda x^\alpha.\lambda y^\alpha.z\{\alpha\}(y)(x)$, then

$$\text{NOT(TRUE)} = \text{FALSE}, \ \text{NOT(FALSE)} = \text{TRUE} \quad \text{and} \quad \text{NOT(INTER)} = \text{INTER}.$$

If one similarly defines the connective OR by

$$\text{OR} = \lambda z^{\text{bool}}.\lambda z'^{\text{bool}}.\Lambda\alpha.\lambda x^\alpha.\lambda y^\alpha.z\{\alpha\}(x)(z'\{\alpha\}(x)(y)),$$

then

TRUE OR TRUE = TRUE,

TRUE OR FALSE = FALSE OR TRUE = TRUE,

FALSE OR FALSE = FALSE,

TRUE OR INTER = INTER OR TRUE = TRUE,

FALSE OR INTER = INTER OR FALSE = INTER,

INTER OR INTER = INTER.

This shows that the natural three-valued connectives can be defined in F. Of course, syntactically they appear as binary connectives[8], but semantically they can be seen as three-valued ones. The importance of three-valued logic w.r.t. questions of normalisation is extreme.

3.2. *Case of* **int**

The type **int** is already more complicated. $\|\mathbf{int}\|$ is made of tuples $(X, (a, (f, z))$ (denoted $(X, a, f; z)$ for reasons of readibility) made of a finite qD X, of $a \in X$, of $f \in X \Rightarrow X$, and of $z \in |X|$ such that $|X|$ is the union of a, $\{z\}$, and of the sets $b \cup \{x\}$, when $(b, x) \in f$. It is possible to characterise $|\mathbf{int}|$, but the result obtained is not very exciting, so we prefer to look at some specific elements of $|\mathbf{int}|$, namely those belonging to the interpretation of some integer $\bar{k}$: $\bar{k}^*$ consists in all tuples $(X, a, f; z)$ in $\|\mathbf{int}\|$ such that:

(i) $z \in f^k(a)$,

(ii) if $a' \subset a$, $f' \subset f$, and $z \in f'^k(a)$, then $a = a'$ and $f = f'$.

Let us give some examples: if $|X| = \{0, \ldots, k\}$, $a = \{0\}$, $f = \{(\{i\}, i+1); i < k\}$, and $z = k$, then $f^k(a) = \{z\}$ and it is easy to see that this point belongs to $\bar{k}^*$; it does not belong to any other $\bar{k}'^*$.

Another example is the following: if $|X| = \{0, 1\}$, $a = \{0\}$, and $f = \{(\{0\}, s), (\{1\}, 1)\}$ (hence $\{0, 1\} \notin X$), then $1 \in f^k(a)$ for all $k > 0$. This point belongs to all $\overline{k+1}^*$'s, but not to $\bar{0}^*$.

Finally, take the example of $|X| = \{0, 1, \ldots, p\}$, of $a = \{0\}$, $f = \{(\{i\}, i+1); i < p\} \cup \{(\{p\}, 0)\}$, snd $z = q \leqslant p$. Then it is easy to see that $z \in f^k(a)$ exactly for $k = q, q+p+1$, $q+2p+2$, etc.; in fact, it belongs to $\bar{k}^*$ for $k = q+p+1$, $q+2p+2$, $q+3p+3$, etc., but not for $k = q$: this is because $z \in f'^q(a)$ for $f' = \{(\{i\}, i+1); i < q\}$.

In fact, it is not difficult to see that the set of all integers k such that a given point $(X, a, f; z)$ of $|\mathbf{int}|$ belongs to $\bar{k}^*$, is eventually periodic: for instance, if $N = \mathrm{card}(X)!$, then $g^n(b) = g^{n+N}(b)$ for any $n > N$, and any $b \in X$ and $g \in Y \Rightarrow Y$. Hence, if we know which one among the integers $N+1, N+2, \ldots, 2N$ is such that $(X, a, f; z)$ belongs to their interpretation, then we can find all greater solutions by shift.

[8] 'Binary' in the sense of 'manichean'!

4. The intrinsic model of λ-calculus

All we have done for the system F enables us to go back to the interpretation of λ-calculus: it is more or less immediate that the interpretation t_D^* of Definition 1.12 is a functor of the λ-structure D; more precisely, t_D^* will appear as an object of an appropriate variable type. Hence, the general results of Sections 2 and 3 will enable us to define t^* as the interpretation of the abstraction term of F (abstracted w.r.t. X, H, and K) corresponding to t_D^*. t^* encodes the value of t_D^* in any D, hence any fact about the interpretation of t in any qD can be viewed from t^*; since t^* is a universal interpretation, one can expect a deeper understanding of t from the study of t^* than from the study of any t_D^*. One can object that t^* also encodes interpretation with very bad properties; admitting that some of these interpretations have no interest at all, let us recall that t^* is a set of invariants, and that it is possible to restrict to those invariants which are thought to be noble: in t^* it is always possible to separate the wheat from the tares, which we refused to do, since the notions of wheat and tares may depend on personal taste, and particular applications.

4.1. Definition. Let α be a type variable, and let w and z be two variables of respective types $\alpha \Rightarrow (\alpha \Rightarrow \alpha)$ and $(\alpha \Rightarrow \alpha) \Rightarrow \alpha$. Then to any λ-term t we associate t° of type α, as follows:

(i) x° is the variable x^α,

(ii) $(\lambda x.t)^\circ$ is $z(\lambda x^\alpha.t^\circ)$,

(iii) $(t(u))^\circ$ is $w(t^\circ)(u)$.

4.2. Definition. Let t be a closed λ-term. Then we define t^* as the interpretation of the closed term of F

$$\Lambda\alpha.\lambda w^{\alpha\Rightarrow(\alpha\Rightarrow\alpha)}.\lambda z^{(\alpha\Rightarrow\alpha)\Rightarrow\alpha}.t^\circ[\alpha, w, z].$$

t^* is an element of the qualitative domain

$$(\Lambda\alpha.(\alpha\Rightarrow(\alpha\Rightarrow\alpha)))\Rightarrow(((\alpha\Rightarrow\alpha)\Rightarrow\alpha)\Rightarrow\alpha)^*.$$

We shall use the notation Λ_0 for this qualitative domain.

4.3. Theorem. *t^* encodes the value of t_D^* in any λ-structure (X, H, K), by*

$$t_D^* = \mathrm{Ap}(\mathrm{Ap}(\mathrm{Ext}(t^*, X), H), K), \text{ i.e, } t_D^* = t^*\{X\}(H)(K).$$

The proof of this theorem is immediate.

4.1. Discussion about t^*

(i) The first thing we want from t^* is that it encodes information about the most obvious λ-structures, namely those for which H and K are reciprocal isomorphisms. First observe that there are nontrivial examples of such a situation: start with a

nontrivial X_0 (i.e., $X_0 \neq \emptyset$) and form $X_1 = X_0 \Rightarrow X_0$; we can define $f_0 \in \mathrm{qD}(X_0, X_1)$ by $f_0(x_0) = (\emptyset, x_0)$. Define in general $X_{n+1} = X_n \Rightarrow X_n$, and $f_{n+1} \in \mathrm{qD}(X_{n+1}, X_{n+2})$ by $f_{n+1} = f_n \Rightarrow f_n$. Then, (X_n, f_{nm}) is a direct system of qD's, indexed by $\mathbb{N}$ with $f_{nm} = f_{m-1} f_{m-2} \ldots f_n$. Let $(X, g_n) = \lim_{\rightarrow} (X_n, f_{nm})$. Then

$$(X \Rightarrow X, g_n \Rightarrow g_n) \simeq \lim_{\rightarrow} (X_n \Rightarrow X_n, f_{nm} \Rightarrow f_{nm})$$

$$= \lim_{\rightarrow} (X_{n+1}, f_{n+1\,m+1}) = X, g_{n+1}).$$

So there are unique isomorphisms k from $X \Rightarrow X$ to X and h from X to $X \Rightarrow X$ such that $g_n \Rightarrow g_n = h g_{n+1}$ and $g_{n+1} = k(g_n \Rightarrow g_n)$. Obviously, h and k are reciprocal, and we are done, with $H = h^+$, $K = k^+$.

(ii) If (X, H, K) is a λ-structure with X nontrivial and H, K reciprocal isomorphisms, then it is not true that (X, H, K) can be approximated by means of similar λ-structures (X_n, H_n, K_n), with X_n finite. So it is more interesting to consider those λ-structures $D = (X, H, K)$ for which

$$H \circ K \subset \mathrm{Id}^{X \Rightarrow X} \quad \text{and} \quad K \circ H \subset \mathrm{Id}^X. \tag{**}$$

These λ-structures are increasing w.r.t. β- and η-conversion. Moreover, let X_i be any finite subset of X, and let g_i be the inclusion map from X_i to X; define

$$K_i = ((g_i \Rightarrow g_i) \Rightarrow g_i)^-(K) \quad \text{and} \quad H_i = (g_i \Rightarrow (g_i \Rightarrow g_i))^-(H);$$

it is immediate that

$$H_i \circ K_i \subset \mathrm{Id}^{X_i \Rightarrow X_i} \quad \text{and} \quad K_i \circ H_i \subset \mathrm{Id}^{X_i}$$

and (X, H, K) can be approximated by means of finite λ-structures still enjoying (**).

4.4. Definition. Remember that an object of $|\Lambda_0|$ consists of a tuple $(X, (H, (K, z)))$ where X is a finite qD, (X, H, K) is a λ-structure, and $z \in |X|$. We shall prefer the notation $(X, H, K; z)$. We define the subset $|\Lambda_1|$ of $|\Lambda_0|$ to consist of those tuples $(X, H, K; z)$ such that (X, H, K) fulfills condition (**) above.

4.5. Theorem. *The interpretation is increasing w.r.t. β- and η-conversion on $|\Lambda_1|$; namely, if $t =/ u$ by means of β- and η-conversions, then*

$$t^* \cap |\Lambda_1| \subset u^* \cap |\Lambda_1|.$$

Proof. Let D be an λ-structure enjoying (**); then the interpretation $\cdot^*_D$ is increasing w.r.t. β- and η-conversion. In particular, if $t =/ u$ and $(X, H, K; z) \in t^* \cap |\Lambda_1|$, then $z \in t^*_{X,H,K} \cap u^*_{X,H,K}$, so $(X, H, K; z) \in u^*$. □

4.2. Final remarks

(i) In order to separate the wheat from the tares, the restriction to $|\Lambda_1|$ is the obvious choice: $t^* \cap |\Lambda_1|$ determines the behaviour of t^*_D on a very wide class of λ-structures, namely all those satisfying (**). Moreover, the fact that the interpretation is increasing, namely that the reduction relation is not interpreted 'flatly' is a nice feature. Of course, this was already possible with the traditional approach, i.e., to choose a particular D satisfying (**); but any such D is particularly artificial, while the class of all such D's is a very nice one.

(ii) Of course, if we compute

$$t^\beta = \cup\{u^*; t =/ u\} \cap |\Lambda_1|,$$

then t^β is an element of Λ_0 which interprets the Böhm tree of t.

The open question is the relation between the equalities $t^\beta = u^\beta$ and the equality between the Böhm trees of t and u. We have not looked seriously at this question. We simply observe that $t^\beta = u^\beta$ implies t and u having the same intepretation in every λ-structure fulfilling $D \Rightarrow D \sim D$. So the basic problem is to look whether the results relating equality of Böhm trees with the model P_ω can be adapted to qualitative domains.

Appendix A. *F* and related systems

Here we consider some other systems, in particular possible strengthenings of F (for which we still get termination of the conversions). As to *strength*, there is a very crude way of measuring it, namely by the class of all number-theoretic functions representable in the system. For instance, systems like Gödel's functional of finite type T, Martin-Løf's system with universes, the language ML etc., have normalisation proofs that can be carried out in rather small subsystems of second-order arithmetic, and, in particular, the function associating to a term of any of these systems its normal form is (under a suitable coding) provably total in PA_2, i.e., representable in F. So F is definitely stronger than all these systems, which does not mean that F contains really these systems: for instance, Martin-Løf's type theory contains type schemes that are not nicely do-able in F.

A.1. The systems F_n

If we allow formation of types by allowing quantification over *connectives* of type n, then we get a system F_n, which is considerably stronger than F. F_0 is just F, so let us explain F_1: Besides the type variables, we add variables Ω_p, Ω'_p, Ω''_p, etc. for p-ary connectives. To the type schemes of F, we add:

- if Ω_p is a connective variable and $\tau_1, \ldots, \tau_p$ are types, then $\Omega_p(\tau_1, \ldots, \tau_p)$ is a type,
- if σ is a type, then $\Lambda\Omega_p.\sigma$ is a type.

The terms are formed as in F, except that we must now give rules for quantification over connectives:

(i) If t is of type σ and Ω_p is a variable of connective which does not occur freely in the type of a variable occurring freely in t, then $\Lambda\Omega_p.t$ is a term of type $\Lambda\Omega_p.\sigma$.

(ii) If t is of type $\Lambda\Omega_p.\sigma$ and T is an abstraction connective $\lambda\alpha_1 \ldots \lambda\alpha_p.\tau$, then $t\{T\}$ is a term of type $\sigma[T/\Omega_p]$.

(Abstraction connectives are defined as follows: if τ is a type and $\alpha_1, \ldots, \alpha_p$ are type variables which are pairwise distinct, then $\lambda\alpha_1 \ldots \lambda\alpha_p.\tau$ is an abstraction connective. In order to substitute an abstraction connective $T = \lambda\alpha_1 \ldots \lambda\alpha_p.\tau$ for a connective variable Ω_p in an expression (term or variable) E, we proceed as follows:
- we first make a formal substitution: replace all Ω_p's by T,
- then we replace all expressions $T(\sigma_1, \ldots, \sigma_p)$ by $\tau[\sigma_1/\alpha_1, \ldots, \sigma_p/\alpha_p]$, and then we get a legal expression of F_1.)

The additional conversion rule of F_1 is as follows:

$$\Lambda\Omega_p.t\{T\} =/ t[T/\Omega_p].$$

In [5], it was proved:

(i) the termination of the conversion process, and

(ii) the class of functions from $\mathbb{N}$ to $\mathbb{N}$ representable in F_n is the class of all provably total functions of arithmetic of order $n+2$, namely PA_{n+2}.

This shows of course that the improvement is genuine, even if the ideas are just a straightforward adaptation of those of F.

A.2. Towards inconsistency

Since it was possible to generalise F by using typed connectives, the idea was to look for a more powerful typing than the finite types. So why not typing the connectives as in system F? We do not give the details here but it was soon discovered that the system was inconsistent: a form of the Burali-Forti paradox could be derived in it. (In fact, this system—let us call it U, as in [4] where these things are explained—was nothing more than a natural deduction system corresponding to arithmetic, not of finite type, but with type levels as in F, comprehension axioms and quantification over types.)

Simultaneously, Martin-Løf (1971) proposed the first version of his type theory; the system took part of its inspiration in Heyting's semantics of proofs, in the Curry-Howard-De Bruijn isomorphism, and in F. F was translated in such a way in the first version, that U could be translated as well, so the system was inconsistent too. Martin-Løf later dropped his axiom "$V \in V$", and since that time, his systems have all been strictly 'below' F.

Recently, Coquand and Huet [2] worked out a system which may seem a bit mysterious:

(i) The system roughly speaking embodies features coming from De Bruijn's AUTOMATH, Martin-Løf's systems, and F (more precisely, the F_n's).

(ii) The syntax is a liberal version of AUTOMATH, in which one can form dependent products as in Martin-Løf's. There are three levels for expressions, one corresponding

to what we call terms, one corresponding to what we call types, and one corresponding to what we call connectives in F_n. There is no fourth level, because one would meet inconsistency. The somewhat obscure restrictions on the syntax all come from the need to 'stick' to the systems F_n, even if their system is much more flexible than the F_n's. The system is clearly stronger than all F_n's.

Up to now, this system is the strongest one ever proposed; moreover, it also takes into account ideas coming from other sources of inspiration, so, in some sense, this is the 'universal functional system'.

All attempts to strengthen this system, in particular to temper with the fourth level, should be considered very cautiously: the Tarpeian Rock is close to the Capitol.

Appendix B. Scott domains and qualitative domains

Scott has investigated, in a lot of papers, all possible equivalent ways of looking at his semantics, so-called Scott domains. In one of these papers [8], he presents his domains in a formalism which is close enough to qualitative domains so that we can see the links between the two notions.

(i) A qualitative domain can be seen as a set of atomic propositions (the points of $|X|$), together with a 'consistency' relation: $p_1, \ldots, p_n$ are consistent exactly when $\{p_1, \ldots, p_n\} \in X$. We can, for instance, form a theory $T(X)$ by taking as axioms all intuitionistic sequents $p_1, \ldots, p_n \vdash$, where $p_1, \ldots, p_n$ is a subset of $|X|$ not in X (and, for instance, minimal w.r.t. this property). Then $A \in X$ exactly when $A + T(X)$ is a consistent theory.

(ii) A Scott domain can be seen as a set of atomic propositions, together with a set of axioms of the form $p_1, \ldots, p_n \vdash$ or $p_1, \ldots, p_n \vdash q$. For technical reasons, there is a fixed bottommost point b_0, together with the axiom $\vdash b_0$. A set A of atomic statements belongs to the Scott domain defined by the set S of axioms when:

- $A + S$ is consistent,
- if $A + S \vdash q$, then $q \in A$.

In particular, Scott domains fulfill the analogue of (qD1) (under the form that a Scott domain is nonvoid) and (qD2). But (qD3) is essentially false: in current Scott domains, finite sets will not be closed under consequence.

If one forgets the purely technical b_0, then a qualititative domain is a Scott domain (take no axiom of the form $p_1, \ldots, p_n \vdash q$). The question is therefore: "do we really need all these sequents $p_1, \ldots, p_n \vdash q$, which complicate the interpretation?".

(1) It is possible that some of these sequents are needed to interpret some atomic data structures; however, these data structures must be slightly uneven, since all current ones (trees, lists etc.) can be done in F, hence within qualitative domains.

(2) In Scott's interpretation, the sequents essentially come from the interpretation of the implicative types. Of course, it is perhaps because Scott wants to take into account nonstable algorithms, such as the well-known 'parallel or'; but qD's also

work for nonstable algorithms: simply in Theorem 1.3, we get several minimal solutions. The nonstable version of $X \Rightarrow X'$ is defined as in Theorem 1.4, except that (FS3) is replaced by:

(F$\not$S3) if $(a, z), (a', z) \in A$ and $a \subset a'$, then $a = a'$.

The sequents come in reality from the order Scott puts between functions, which is the pointwise order: $F \leqslant G$ iff $F(x) \leqslant G(x)$ for all x. If one were taking the Berry order, then everything would work smoothly and, even without stability, we would stay within qD's. But if $F \leqslant G$, it is not true that $\mathrm{Tr}(F) \subset \mathrm{Tr}(G)$, so the only solution is to take, in order to represent F, all solutions of $z \in F(a)$ with a finite (because a solution (z, a) minimal w.r.t. F is no longer minimal w.r.t. G in general). Of course, one must say somewhere that if (z, a) is a solution, then (z, a') is a solution, for $a' \supset a$, and this leads to the extra axioms. However, if there are some real reasons to consider nonstable algorithms, it is hard to advocate the choice of the pointwise ordering!

(3) A type formation scheme where there is a more serious reason to introduce complications is the sum of types. In [6], we have shown how all possible schemes for this type could be interpreted by qualitative domains. However, it is true that one would like to interpret, if possible, the sum of types by something like a sum. This is impossible in qD's: if X and Y are qD's (suppose for simplicity that $|X| \cap |Y| = \emptyset$, if 0 and 1 are two elements not in $|X| \cup |Y|$, then one can consider $X + Y$, which consists of:

- the void set $\emptyset$,
- the sets $a \cup \{0\}$ for $a \in X$,
- the sets $b \cup \{1\}$ for $b \in Y$).

Condition (qD3) is violated, because, when $a \neq \emptyset$, the subset a of $a \cup \{0\}$ does not belong to $X + Y$. Then one has to add the axioms $z \vdash 0$ for $z \in |X|$ and $z' \vdash 1$ for $z' \in |Y|$. So, the usual interpretation of the sum introduces some typical features of Scott domains. However, it is easily seen that the Scott domains which are needed are those for which the closure of a finite consistent set w.r.t. consequence is finite, and so we do not introduce too much rubbish.

So, in the case of the disjunctive type, there is a clear dilemma:

- either we stay within the simple concept of qD, and the price to pay is a slight complication of the interpretation of such types,
- or we interpret it as a sum, but then we have to weaken our class of domains so that to accept some reasonable classes of Scott domains.

The interpretation of the sum developed in [6] is compatible (in case of a primitive connective sum added to F) with the interpretation given here.

Appendix C. Binary qualitative domains

In qualitative domains not all subsets are accepted. Of course, it is important to understand at which moment one actually needs some kind of incompatibility,

because this could simplify the interpretation. So let us start with qD's of the form $\mathscr{P}(|X|)$, where everything is compatible. If we form the function space $X \Rightarrow Y$, then (FS2) introduces no incompatibility, but (FS3) introduces an incompatibility between pairs. (The same would be true in the nonstable case (F$\mathscr{S}$3) sketched in Appendix B, so this has nothing to do with stability.) So, incompatible pairs exist by nature! But no scheme introduces incompatible 3-tuples, for instance. This explains the definitions below. In case somebody would later find a reason of incompatibility for 3-tuples, one can obviously replace the integer 2 by any $N \geq 2$, and get a similar concept of N-ary qD.

C.1. Definition. A qD X is *binary* when the followings holds: if $a \subset |X|$ and $a \notin X$, there are $x, y \in a$ such that $\{x, y\} \in X$.

C.2. Theorem. *Everything done so far can be done within the category* 2qD *of binary qualitative domains.*

Proof. Essentially, we have to prove that:

(i) If X and Y are binary qD's, then $X \Rightarrow Y$ is a binary qD. (FS3) is a binary condition. (FS2) is also a binary condition: if $F(b) \notin Y$, then $\{z, z'\} \notin Y$ for some $z, z' \in F(b)$, so for some $(a, z), (a', z') \in \mathrm{Tr}(F) = A$ we already have incompatibility: $\{(a, z), (a', z')\} \notin X \Rightarrow Y$.

(ii) If F is a variable type, mapping 2qD into 2qD, then we define $\mathrm{Tr}(F)$ as in Definition 2.7 (but we only consider binary qD's). It is obvious that, in the compatibility condition, it suffices to make $n = 1$, i.e., to look at pairs. □

C.3. Remark. There is a problem with general qD's, regarding the effectivity of Definition 2.7, because, when a is finite, we cannot give any bound on n (recall that $(X_0, x_0), \ldots, (X_n, n)$ are not necessarily distinct), a problem that would make $\mathrm{Tr}(F)$ uncomputable in some cases. But, if we restrict to binary (or N-ary) qD's, then we can give the bound 2 on the number of points. Now, let us remark that there are only finitely many ways to define maps $f_0 \in \mathrm{qD}(X_0, Y)$ and $f_1 \in \mathrm{qD}(X, Y)$, when Y is fixed. Also observe that one can restrict to the case where $|Y| = \mathrm{rg}(f_0) \cup \mathrm{rg}(f_1)$ and then, up to isomorphism, there are only finitely many possible Y's: this shows that the computation of $\mathrm{Tr}(F)$ is decidable in the binary case.

Appendix D. Total objects

The interpretation of F has so far been very elementary; we would like to say something that could have some relation with the fact that the conversion process eventually ends; hence, something about the functions, the objects involved, being 'total'. A first superficial impression would be to identify semantically, 'totality' with maximality, as done for instance by Scott [8]: this is wrong; moreover, there is not

even an inclusion relation between the two notions. For instance, INTER is maximal, but can hardly be claimed to be total, since IF INTER THEN TRUE ELSE FALSE = VOID. The void element of type **bool** is certainly not total, while IF... THEN... ELSE, TRUE and FALSE *are*, since they belong to F, and we are looking for a concept of 'totality' for which the definable objects of F are total, and closed under the operations of F. So, INTER cannot be total.

On the other hand, the function associating to x of type $\Lambda\alpha.(\alpha\Rightarrow\alpha)\Rightarrow\alpha$ the point $x\{\mathbf{bool}\}\ (\lambda y^{\mathbf{bool}}.y)$ which is definable in F, must be total, but its interpretation is void, so it is included in the interpretation of $\lambda x^{\mathbf{bool}}$.TRUE, whose interpretation is nonvoid.

In order to solve this question of 'totality', we simply adopt the method we already used in [3] namely to quantify over arbitrary definitions.

D.1. Definition. A *total* qD is a pair (X, X_t), where X_t is a subset of X. The elements of X_t are said to be total (w.r.t. X_t).

D.2. Definition. If $\tau[\boldsymbol{\alpha}]$ is a type of F, where $\boldsymbol{\alpha}$ lists all variables of type occurring freely in τ, and if $(\boldsymbol{X}, \boldsymbol{X}_t)$ is a sequence of total qD's of the same length as $\boldsymbol{\alpha}$, then we define a total qD $(\tau[\boldsymbol{X}, \boldsymbol{X}_t], \tau[\boldsymbol{X}, \boldsymbol{X}_t]_t)$ as follows:

(i) $\tau[\boldsymbol{X}, \boldsymbol{X}_t] = \tau * [\boldsymbol{X}]$, already defined,

(ii) if $\tau[\boldsymbol{\alpha}]$ is α^i, then $\tau[\boldsymbol{X}, \boldsymbol{X}_t]_t$ is X_t^i,

(iii) if $\tau[\boldsymbol{\alpha}]$ is $\sigma[\boldsymbol{\alpha}]\Rightarrow\rho[\boldsymbol{\alpha}]$, then $a\in\tau[\boldsymbol{X}, \boldsymbol{X}_t]_t$ iff, for any $b\in\sigma[\boldsymbol{X}, \boldsymbol{X}_t]_t$, we have $a(b)\in\rho[\boldsymbol{X}, \boldsymbol{X}_t]_t$,

(iv) if τ is $\Lambda\alpha.\sigma[\alpha, \boldsymbol{\alpha}]$, then $a\in\tau[\boldsymbol{X}, \boldsymbol{X}_t]_t$ iff, for any total qD (Y, Y_t), $a\{Y\}$ belongs to $\sigma[Y, \boldsymbol{X}, Y_t, \boldsymbol{X}_t]_t$.

D.3. Theorem. *Let $t[\boldsymbol{\alpha}, x^{\sigma}]$ be a term of type τ; let $(\boldsymbol{X}, \boldsymbol{X}_t)$ be a sequence of total* qD's *of the same length as $\boldsymbol{\alpha}$, and let $\boldsymbol{a}$ be a sequence of objects in $\sigma[\boldsymbol{X}, \boldsymbol{X}_t]_t$; then $t^*[\boldsymbol{X}, \boldsymbol{a}]\in\tau[\boldsymbol{X}, \boldsymbol{X}_t]_t$.*

Proof. The proof is practically immediate; the case of EXT requires the following lemma.

D.4. Lemma (Substitution Lemma)

$$\tau[\sigma[\boldsymbol{X}, \boldsymbol{X}_t], \boldsymbol{X}, \sigma[\boldsymbol{X}, \boldsymbol{X}_t]_t, \boldsymbol{X}_t]_t = \tau[\sigma/\alpha, \boldsymbol{X}, \boldsymbol{X}_t]_t.$$

The proof of this lemma is straightforward, but it contains a hidden use of the comprehension axiom of second-order arithmetic, namely to say that $\sigma[\boldsymbol{X}, \boldsymbol{X}_t]_t$ is a set. □

D.5. Remark. Although we start with arbitrary definitions, when σ is closed, σ_t is well defined, and we therefore have an *intrinsic* concept of a total object of type σ.

For instance, the only total objects of type **bool** are TRUE and FALSE. However, the interpretation is not extensional, in the sense that two total functions may agree on all total arguments, but be different.

D.6. Theorem. *The only total objects of type* **int** *are the integers.*

Proof. Let A be a total object of type **int**; if X is a qD, $a \in X$, and $f \in X \Rightarrow X$, then define $X_t = \{f^n(a); n \in \mathbb{N}\}$. Then (X, X_t) is a total qD, and, moreover, the object a and the map f are total (w.r.t. X_t). So, $A\{X\}(a,f) = f^k(a)$ for some k. We show that the integer k can be chosen independently of X, a, and f: first, consider $X_0 = \{\emptyset, \{0\}, \{1\}, \{2\}, \ldots\}$, $a_0 = \{0\}$, and $f_0 = \{(i, i+1); i \in \mathbb{N}\}$; we obtain $A\{X_0\}(a_0, f_0) = \{k_0\}$. Given (X, a, f), we form a new qD Y by putting together X and two disjoint isomorphic copies X_0' and X_0'' of X_0; the elements of Y will all be (disjoint) unions $b \cup c' \cup d''$ with $b \in X$, $c, d \in X_0$. Form $b_0 = a \cup a_0'$, $g_0 = f \cup f_0'$; then $b_0 \in Y$ and $g_0 \in Y \Rightarrow Y$, and $A\{Y\}(b_0, g_0) = g_0^l(b_0)$ for an appropriate l. Now considering the morphism $'$ from X_0 to Y, it is easy to show that $(A\{X\}(a_0, f_0))' \subset A\{Y\}(a_0', f_0') \subset A\{Y\}(b_0, g_0)$ and this forces $l = k_0$. For similar reasons, if $b_1 = a \cup a_0''$, $g_1 = f \cup f_0''$, then $A\{Y\}(b_1, g_1) = g_1^{k_0}(b_1)$. From this we conclude that $A\{Y\}(b_0 \cap b_1, g_0 \cap g_1) = (g_0 \cap g_i)^{k_0}(b_0 \cap b_1)$. Now, if h is the canonical morphism from X to Y, then $h^+(a) = b_0 \cap b_1$ and $(h \Rightarrow h)^+(f) = g_0 \cap g_1$, so

$$A\{Y\}(h^+(a), (h \Rightarrow h)^+(f)) = h^+(f^{k_0}(a)),$$

hence

$$A\{X\}(a,f) = h^-(A\{Y\}(h^+(a), (h \Rightarrow h)^+(f)) = h^- h^+(f^{k_0}(a)) = f^{k_0}(a).$$

Since $k = k_0$ for all X, a, f, it is clear that A is the interpretation of the integer k_0. □

D.7. Remarks. (i) Theorem D.3 is not provable in PA_2: simply, observe that Theorem D.6 is provable in PA_2, hence, from Theorem D.3, one can define, for any closed t of type **int**⇒**int** of F, a function t' from $\mathbb{N}$ to $\mathbb{N}$, by $t'(n) = m$ iff $t(\bar{n})^* = \bar{m}^*$. This function is (as we know from the normalisation theorem) equal to $|t|$ of Definition 3.6(ii). But then t' can be extensionally equal to any given provably total recursive function of PA_2, etc.

(ii) Theorem D.3 does not immediately imply the normalisation theorem, because it is not excluded a priori that $t(\bar{n})^* = \bar{m}^*$ as a result of the interpretation, without having $t(\bar{n}) =/ \bar{m}$. But, the techniques of [3] (functional interpretation) would content themselves with $t(\bar{n})^* = \bar{m}^*$, and from this we obtain that Theorem D.3 implies the 1-consistency of PA_2, which in turn implies normalisation.

(iii) A priori, the notion of totality needs third-order arithmetic because $X_t \in \mathcal{P}(\mathcal{P}(|X|))$; however, this can be lowered to second-order, simply by restricting to recursively enumerable total objects, etc.

(iv) An open question remains: if τ is a purely universal type, i.e., if τ consists of quantifiers $\Lambda\alpha_1 \ldots \Lambda\alpha_n$, followed by a quantifier-free part, does the analogue of

Theorem D.6 hold? In other words, are there total points of τ^* which are not of the form t^*?

References

[1] G. Berry, Modèles complètement adéquats et stables des λ-calculs typés, Thèse de Doctorat d'Etat, Université Paris VII, 1979.

[2] T. Coquand and G. Huet, Une théorie des constructions, in: *Proc. ASL Congress*, Orsay (North-Holland, Amsterdam, 1985) to appear.

[3] J.-Y. Girard, Une extension de l'interpretation fonctionnelle de Gödel à l'analyse et son application à l'élimination des coupures dans l'analyse et la théorie des types, in: J. F. Fenstad, ed., *Proc. 2nd Scandinavian Logic Symp.* (North-Holland, Amsterdam, 1971) 63-92.

[4] J.Y. Girard, Interprétation fonctionnelle et élimination des coupures de l'arithmétique d'ordre supérieur, Thèse d'Etat, Université Paris VII, 1972.

[5] J.-Y. Girard, Quelques résultats sur les interprétations fonctionnelles, in: Mathias and Rogers, eds., *Cambridge Summer School in Mathematical Logic*, Lecture Notes in Mathematics **337** (Springer, Berlin, 1973) 232-252.

[6] J.Y. Girard, Normal functors, power series and lambda-calculus, *Ann. Pure Appl. Logaic*, to appear.

[7] J.C. Reynolds, Polymorphism is not set-theoretic, in: *Internat. Symp. on Semantics of Data Types*, Lecture Notes in Computer Science **173** (Springer, Berlin, 1984) 145-156.

[8] D. Scott, Domains for denotational semantics, in: *Proc. ICALP'82*, Aarhus, Lecture Notes in Computer Science **140** (Springer, Berlin, 1982).

[9] A.S. Troelstra, Notes on intuitionistic second order arithmetic, in: Mathias and Rogers, eds., *Cambridge Summer School in Mathematical Logic*, Lecture Notes in Mathematics **337** (Springer, Berlin, 1973) 171-205.

[10] G. Winskel, Events in computation, Ph.D. Thesis, Edinburgh, 1981