Contemporary Mathematics
Volume 92, 1989

THE THEORY OF CONSTRUCTIONS: CATEGORICAL SEMANTICS AND TOPOS-THEORETIC MODELS

J. Martin E. Hyland and Andrew M. Pitts

ABSTRACT. A syntactically rich version of the Coquand-Huet *theory of constructions* is described as a theory of dependent types involving expressions at three different levels (Terms, Types and Orders) together with indexed sums and products of various kinds. Two extensions of the theory involving universal types are also discussed. A complete category-theoretic explanation of the meaning of the theory is built up, based upon a careful analysis of the categorical semantics of Martin-Löf's theory of dependent types. Finally, two particular models of the theory of constructions are described (modelling the two extensions of the theory mentioned above). In these models, the Orders and Types are denoted by particular kinds of Grothendieck topos, namely *algebraic toposes* (toposes of presheaves on small categories with finite limits) and *algebraic-localic toposes* (toposes of presheaves on meet semi-lattices).

CONTENTS

Introduction

The *Theory of Constructions* is the very high level functional programming language due to Coquand and Huet [CH]; it contains Girard's higher order lambda calculus [G1] and the core of Martin-Löf's theory of dependent types [M-L2] as subsystems. In this paper we describe a syntactically rich version of this theory, introduce a general class of models for it and give two particular models. Category theory comes into our work in two distinct ways which it is as well to distinguish here.

In the first place, in obtaining our general notion of model we give an explanation in terms of category theory of what constitutes a semantics for the various parts of the

1980 *Mathematics Subject Classification* (1985 *Revision*). 03B15, 18B25, 68Q55.

The first author's research was supported by grants from the SERC and the Alvey Project.

The second author is a Royal Society University Research Fellow and gratefully acknowledges the financial support of that institution.

theory of constructions. We certainly believe this is the right approach. For one thing, the descriptions of semantics (even for a simple subsystem like the second order lambda calculus) appear mathematically uncivilized when couched in non-categorical terms. What is more, the categorical concepts we use in defining the semantics have (we think) a remarkable degree of simplicity and elegance—especially in comparison with the syntactic complexities of the theory of constructions itself. For another thing, the category-theoretic perspective provides all kinds of valuable insights into the underlying *meaning* of the formal constructs of the theory. The main mathematical ideas which underlie our explanation of the semantics are those of (*locally*) *cartesian closed category*, Grothendieck's *fibrations* and Lawvere's *hyperdoctrines*. (These are part of the category theorist's inheritance from the 1960s.)

Secondly, the objects that we use to model the types (and "orders") of the language are not simply sets, or domains, or objects *in* some category—as in earlier work. They are *themselves* categories. (Hence they are objects in a 2-category, although we do not exploit this extra level of structure in this paper.) This is in line with an old idea of Lawvere that the structures of mathematical interest can not only be organized into categories, but also often *are* (usefully seen as) categories. The categories we use are certain kinds of *Grothendieck topos*. So the main mathematical ideas we use to develop our models are those of *classifying topos*, of the *internal logic* of toposes and of *relativization* to an arbitrary base topos of mathematical constructions and proofs. (These are for the most part the category theorist's inheritance from the 1970s.)

We would be the first to admit that this puts rather large demands on the understanding of those readers who are not topos-theorists. (We hope there are many.) Since there are equivalent, more concrete descriptions of our models (which we give briefly in 4.20 and 5.11), it seems worth saying something in justification of our chosen treatment. Certainly some aspects of the models can be developed in a way which is a straight forward imitation and generalization (from the order-theoretic to the fully category-theoretic) of ideas from domain theory. However, other aspects are really very much simpler from the topos-theoretic viewpoint. To give just one example, the existence of an Order of Types (see 2.11) in the models is an easy consequence of the notion of *classifying topos*. Indeed, there are delicate points involving variations on the notion of "internal category with finite limits" in a topos which it would be hard to steer through without some understanding of the internal logic. So our advice to computer scientists has to be not to try to avoid topos theory, but to try to learn it!

Ours are not the first models for the theory of constructions. First of course there is the term model, features of which we understand well in view of the strong normalization theorem [Coq1] which holds for the system. Then there is a whole class of models which are based on sufficiently complete, internal categories in realizability toposes (see [HRR, section 8). Here incidentally category theory provided the kind of insight we referred to above in showing how to extend what were known models of higher order polymorphic lambda calculus to models for the theory of constructions. Finally, there are models based on models of a system of types with "Type:Type" and fixpoint recursion (see [Cd] and the references therein); these are derived from variants of the closure operator model described by Scott in [Sc]. It is these models which we feel that Girard was most justified in criticising in [G2] for their *ad hoc* nature. We claim that our models can be seen as "natural" substitutes (i.e. relatively free of coding) for the *ad hoc* models. (In particular, it is the case that our models satisfy a version of "Type:Type" and support initial fixpoint recursion both for Operators (including Types) and for Terms; we will describe the modelling of "Type:Type" quite fully, but leave the discussion of recursion to another occasion.)

The structure of the paper is as follows:

In section 1 we give a detailed description of the theory of constructions. The system is presented in two tiers, each of which is a theory in the style of Martin-Lōf, with rules for sums and products of dependent types and also with unit type (which plays a crucial role in the theory as we formulate it and is not there merely for aesthetic reasons). The types of the top layer are called "Orders" and their elements (terms) are called "Operators", whilst those of the lower layer are "Types" and their elements "Terms". If you will, the top layer consists of the "large" types and the bottom layer of the "small" ones—although we will see through the models that the theory actually does not carry any import of size. The connection between the two layers is provided by a particular Order *Type* whose Operators are precisely the Types: in other words *Type* is the "Order of all Types". So far this amounts to a restricted version of Martin-Lōf type theory (no equality types, no finite sum types, no natural number type) with a single Martin-Lōf universe. On top of this we require not only the closure of Orders under Type-indexed products and sums, but also the closure of Types under Order-indexed products and sums. The latter property requires careful formulation (for the sum clauses) and is the aspect which makes finding models rather hard. We enrich the basic theory of constructions by allowing constants of various kinds and consider equational theories extending the basic theory. In particular, at the end of the section we describe two extensions of the basic theory, one with an "Order of all Orders" (1.11) and one where the Types and Orders are essentially the same (1.12).

In section 2 we build up a description of the categorical structures needed to model the theory of constructions: see 2.13 for a summary of what we require. It should be emphasised that the aim (which we achieve) is to give a notion of model which is truly *general*: the criterion for success is the familiar one in categorical logic—to have an equivalence between a 2-category of theories (over the basic theory of constructions) and a 2-category of suitable categorical structures. A fundamental part of this work is an account (2.2) of a general category-theoretic semantics for the pure notion of "dependent type" with no closure assumptions other than that of substitutivity. The semantics of sums and products is then considered on top of this (2.4), leading to the notion of *relatively cartesian closed category* (2.7 *et seq.*), taken from [Ta] and which we see (in Proposition 2.6) generalizes that of locally cartesian closed category. We believe that this material is of interest in its own right. It unifies previous work in this area by Seely [Se1], by

Cartmell [Ca] and more recently by Taylor [Ta]. (We believe also that it encompasses Obtułowicz's recent work [Ob], by applying an iterated version of the Grothendieck construction to one of his hierarchies of indexed categories to obtain one of our relatively cartesian closed categories. It is important for Obtułowicz that his structures are (essentially) algebraic; however, ours are models of a particular lim theory and the latter are in many respects just as well behaved as algebraic theories.)

The rest of the paper is devoted to describing our particular examples of the general structure which emerged in section 2. These examples involve properties of internal categories with finite limits in toposes (and the special case of internal meet semilattices). In order to develop some of these properties we employ the "lim theories" of M.Coste [Co1]; the necessary material is reviewed in section 3. Then in section 4 we present the model of the theory of constructions in which both Orders and Types are denoted by *algebraic toposes*—by definition these are the Grothendieck toposes which are equivalent to a category of presheaves on a small category with finite limits. The topos theory needed to verify that we have a model is nearly all standard; the exception to this comes when we prove (in 4.16) that the geometric morphisms which determine relative algebraic toposes are closed under exponentiation by algebraic toposes—our proof requires some work on "strictifying" *lex fibrations* (see 4.11 *et seq*.). Finally in section 5 we describe a model in which the Orders are still denoted by algebraic toposes, but the Types are now denoted by algebraic toposes which are also *localic*. With one exception, the material we need for this model is part of the well-developed theory of localic toposes. The exception has once again to do with closure under exponentiation. We prove a purely topos-theoretic result (Proposition 5.6) which as far as we know is new—namely that *for each exponentiable topos, exponentiation preserves geometric inclusions and localic geometric morphisms*.

Clearly this paper is only a beginning. Some extensions of the models, for example to injective toposes and continuous lattices, appear quite straightforward. Others, such as extensions to more general classes of toposes and domains, seem more problematic. Frustratingly, we are still unable to use topos-theoretic machinery (such as the theory of *atomic* toposes) to extend the Girard style models of polymorphism [G2]. There are also problems with understanding the proper rules for weak equality types (it seems that there is more than one notion), some of which are certainly interpretable in our models. Having only *weak* equality types (and *weak* finite sum types) is an inevitable consequence of a facet of our models which we have not addressed at all here, namely that they support the interpretation of recursive Terms and Types. We believe that the proper handling of these aspects of the models is part of the bigger problem, which we intend to address in further work, of understanding how the 2-categorical (in fact, bicategorical) structure of the models should be reflected in a richer syntax than the present one.

This paper emerged from the authors' joint interest in the "natural" model for polymorphism given by Girard [G2]. We owe an important stimulus to Thierry Coquand who, in the course of his work with Gunter and Winskel [CGW] on extensions of Girard's

idea to more general "Berry domains", noticed that by relaxing the definitions slightly, the same idea worked for Scott domains. Coquand has since pushed these ideas further in a concrete fashion—as has Lamarche [La], independently. A general notion of indexed products for Scott domains had been considered by Paul Taylor in his thesis [Ta], but he did not explicitly extend the notion to products over a base *category*. Thus in the air was a model for the theory of Types (but not Orders) corresponding to the localic component of the model described in section 5. Initially, the first author developed this part of the model via the notion of internal algebraic lattices in toposes. The second author took the step of formulating things entirely in terms of the relevant toposes, from which the full model (of Orders as well as of Types) emerged quite rapidly.

Finally we would like to pay tribute to the open atmosphere and spirit of co-operation created by the mathematicians (Johnstone, Moerdijk, dePaiva, Robinson, Rosolini) and computer scientists (Coquand, Gunter, Winskel) in Cambridge during 1986-87. All of them commented in various ways on early versions of the ideas presented here; and they contributed crucially to our understanding of what we were trying to model and how.

1 The theory of constructions

1.1. Introduction. In this section we are going to describe a syntactically rich version of the *theory of constructions* of Coquand and Huet [CH]. This language is a natural amalgam of Martin-Löf's predicative theory of dependent types [M-L2] and Girard's impredicative higher order lambda calculus [G1]. We do not assume familiarity with these languages. Many readers, whom we hope to interest in our semantics, will be familiar only with the syntax of the predicate calculus and simple type theory. Hence we give a fairly leisurely account of the formal syntax. We also refer the reader to Troelstra's careful account of the syntax of Martin-Löf type theories [Tr].

In the interests of clarity, the well-formed expressions of the language are divided into three levels: *Terms*, *Operators* and *Orders*. This is the practice in Edinburgh LF [HHP], which is a predicative fragment of the theory of constructions: there the corresponding terminology is *Terms*, *Types* (misleadingly, as most expressions of the middle level are not of kind "*Type*") and *Kinds*. Coquand and Huet see their theory as (an extension of) the proof theory of Church's theory of types [C]. So their terminology is *Proofs*, *Terms* (including *Propositions*) and *Types*. This interpretation is based on the idea that *Propositions* are some special *Types* (but not all *Types* are *Propositions*).

To describe the syntax, we use metavariables $K,L,M,...$ for Orders, metavariables $S,T,U,...$ for Operators and metavariables $s,t,u,...$ for Terms. *Types* are Operators of Order "*Type*" and we use metavariables $A,B,C,...$ for them. In the Coquand-Huet interpretation, their *Propositions* (≡ our *Types*) are explicitly a special case of their *Types* (≡ our *Orders*). We find it conceptually helpful to keep the syntax for Orders and Operators distinct from the syntax for Types and Terms. The similarities are such however as to make it useful to have a generic notation. We will use $P,Q,R,...$ to denote things which are either Orders or Types and correspondingly $p,q,r,\ldots$ to denote things which are either Operators or Terms.

The system we describe has Operator variables $X,Y,Z,\ldots$ (which run over suitable Orders) and Term variables $x,y,z,\ldots$ (which run over suitable Types). The notion of free variable is standard and we assume the reader can supply the definitions as the syntax is presented. We write $FV(K)$, $FV(S)$ and $FV(s)$ for the finite sets of free variables in an Order K, an Operator S and a Term s respectively. As a generic notation for variables we use $\xi,\eta,\zeta,\ldots$.

We also assume we have a suitable notion of substitution of Operators and Terms for free variables of given Orders and Types. We use the notation $E(p/\xi)$ for the result of substituting p for ξ throughout the expression E.

The basic theory and extensions of it by axioms consist of sets of *judgements* J, made in *contexts* Γ. We write these as

$$J\ [\Gamma].$$

All the variables free in a given judgement must be "declared" in the context Γ in which the judgement is made. The well-formed expressions of the three levels, the judgements involving them and the contexts (containing the variable declarations) in which the judgements are made, are all defined together by a mammoth simultaneous recursive definition. Such a definition presents problems of exposition. We explain the forms of judgement and the notion of a context outside the recursive definition, and then give the basic clauses of the definition in the style used by Martin-Löf [M-L2]. So the general form of a clause is:

$$\frac{J_1\ [\Gamma_1]\ \ldots\ldots\ J_n\ [\Gamma_n]}{J\ [\Gamma]}\ ;$$

and its meaning is that if judgements of the form $J_1\ [\Gamma_1],\ldots,J_n\ [\Gamma_n]$ are in our set, then so is $J\ [\Gamma]$. (Usually the reference to the contexts will be supressed.)

1.2. Forms of judgement. We adopt the conceptual framework of Martin-Löf type theory [M-L2]. Thus a theory will consist of judgements (*structural judgements*) of the form

$$K\in ORDER,\quad S\in K,\quad \text{or}\quad s\in A$$

and judgements (*equality judgements*) of the form

$$K=L\in ORDER,\quad S=T\in K,\quad \text{or}\quad s=t\in A,$$

all made in suitable contexts. Part of the force of the structural judgements is that the expression to the left of "$\in$" is well-formed. Indeed an expression is well-formed just by virtue of appearing to the left of "$\in$" in a structural judgement. Of course $s\in A$ presupposes that A is well-formed, that is, that we already have the judgement $A\in Type$. Similarly, $S\in K$ presupposes that K is well-formed, that is, that we already have the judgement $K\in ORDER$.

A judgement of form $K\in ORDER$ has no further force than that K is well-formed.

Similarly a judgement of form $K=L\in ORDER$ plays a relatively weak role in the theory. There are no *ORDER* variables (as there is no level of syntax above that of *ORDER*) and hence no question of "the substitution of equals for equals". In giving the formal rules of the language we will write "K" for "$K\in ORDER$" and "$K=L$" for "$K=L\in ORDER$".

However the "$\in$" has greater force in the other judgements—generically of the form $p\in P$ and $p=q\in P$. The judgement $p\in P$ states not only that p is well-formed, but also that P has a member (namely p). Thus in the Coquand-Huet interpretation, where A is a proposition, the judgement $s\in A$ presents a proof of A. It is worth stressing that while equality judgements *seem* familiar, $p=q\in P$ is not an *atomic* proposition appearing in more complicated ones in the theory of constructions. Equality judgements play a lively role in the production of new judgements by "the substitution of equals for equals" and have the force of "definitional" equality—for which see [M-L2]. In extensions of the basic system we may have the quite distinct notion of *equality* Types or Propositions: these will appear in complex expressions. In the pure theory, equality judgements are generated by "reductions". That is, there is a direction to them and they form a rewrite system. A strong normalization theorem can be proved (compare [Coq1]).

1.3. Variable declarations. All judgements are made in a context which, as we will shortly describe, is a structure on judgements of the form $X\in K$ or $x\in A$, where X and x are variables. Judgements of these forms, that is, judgements of the general form

$$\xi\in P$$

are called *variable declarations*. If the judgement J is a variable declaration, we write ξ_J for the variable to the left of "$\in$"—the *variable declared* by J—and P_J for the Order or Type to the right of "$\in$"—the *kind* of the declaration.

The need for variable declarations as an integral part of the language comes about as follows. In a calculus of dependent types we have to consider judgements of the form

$$y\in B(x)\ \ [x\in A],$$

that is, y a free variable of type $B(x)$ varying as x varies over A. Suppose now we wish to substitute $a\in A$ for x. We should obtain a free variable of type $B(a)$. It is pointless to attempt to mangle or decorate "y". After substituting, we must have $y\in B(a)$. Thus type expressions can not be provided with their own inviolate collection of variables. An expression must involve variables which have been declared to be of given type.

In our version of the theory of constructions we are taking variables $X,Y,Z,...$ which may be declared to be of given Order, and variables $x,y,z,...$ which may be declared to be of given Type. From the clauses of the recursive definition, the reader will see that the variables of either level may depend on (that is, be declared in the context of) variables of either level. (As mentioned in 1.1, Coquand and Huet regard their Propositions as special cases of their Types and so do without this distinction between the variables at the two levels.)

Once a variable has been declared it can be used to build up complex expressions which appear in further judgements. The variable declaration forms part of the context for these further judgements.

1.4. Contexts. We describe here a notion of *context* slightly more general than that used by Martin-Löf [M-L2] or Coquand and Huet [CH]. For us contexts are certain partial orders $<$ on finite sets Γ of variable declarations. We read "$J_1 < J_2$" as "J_1 is a prerequisite for J_2", or "J_1 is presupposed by J_2", or "J_1 precedes J_2".

If $\boldsymbol{\Gamma} = (\Gamma, <)$ is a poset and $J \in \Gamma$, write $\boldsymbol{\Gamma}|_J = (\Gamma|_J, <)$ for the restriction of the partial order $<$ to

$$\Gamma|_J = \downarrow(J) = \{J' \mid J' \leq J\}.$$

If $J \notin \Gamma$ write $(J{>}\boldsymbol{\Gamma}) = ((J{>}\Gamma), <)$ for the poset obtained by adding J as a greatest element above all of Γ; more generally, if $\boldsymbol{\Gamma}$ and $\boldsymbol{\Gamma}'$ are disjoint posets $(\boldsymbol{\Gamma}' > \boldsymbol{\Gamma})$ will denote their union, ordered so that everything in $\boldsymbol{\Gamma}'$ is greater than anything in $\boldsymbol{\Gamma}$. Finally we write $(J, \boldsymbol{\Gamma})$ for a poset with J as a maximal element whose removal leaves the poset $\boldsymbol{\Gamma}$.

Define $\boldsymbol{\Gamma} = (\Gamma, <)$ to be a *context* if and only if:

(*i*) the elements of Γ are all variable declarations and distinct variables are declared by distinct declarations;

(*ii*) if $J \in \Gamma$, then $J\ [\boldsymbol{\Gamma}|_J]$ (the judgement J in the context $[\boldsymbol{\Gamma}|_J]$) is a judgement of the theory.

(It will be a consequence of the definitions that—in line with the discussion in 1.1 and 1.2—if $\boldsymbol{\Gamma}$ is a context and $J \in \Gamma$, then $FV(P_J) \subseteq \{\xi_{J'} \mid J' < J\}$.)

A context in Martin-Löf's sense is a linearly ordered context in ours. Given one of our contexts $(\Gamma, <)$, any extension of $<$ to a linear order on Γ will provide a more or less equivalent context in Martin-Löf's sense. Martin-Löf's procedure is in harmony with the way in which the lambda calculus reduces functions of many variables to functions of one. As such it seems well-adapted to implementation. However, we have reasons for preferring a more liberal notion, as we now explain.

Firstly, if one imagines the clauses of the recursive definition as providing natural deduction trees leading to judgements, then the relevant variable declarations will sit at nodes of the tree and a partial order will be induced in a natural way. This suggests adopting the idea familiar from the predicate calculus that we can always regard an expression as involving extra free variables. If a judgement can be made in a context, it can always be made in a wider context. (So the notation $J\ [\boldsymbol{\Gamma}]$ corresponds here to the predicate calculus notation $\phi(\overline{x})$ for a formula whose free variables appear in the list $\overline{x}$.) Secondly, we wish to allow for the introduction of constant Orders (Order constructor symbols) with free variables. (Also we have constant Types with free variables, but as we could in principle do without them, they are not so pressing.) It is natural then to have constant Orders

$$K(\xi_1, \ldots, \xi_n)$$

which depend on *discrete* contexts

$$[\xi_1 \in P_1, \ldots, \xi_n \in P_n].$$

(Compare with the introduction of predicate symbols in the predicate calculus.) In particular there is no natural choice of total order on such a set of variable declarations.

However one regards contexts, one should think of the clauses in the recursive definition as defining the central notion—namely the set of (correct) judgements-in-contexts. When giving the clauses however, we will supress mention of the contexts as far as possible.

1.5. General rules. In this section we display those clauses of the recursive definition which do not involve particular operators on Orders and Types. We use notation from 1.1 and 1.2.

· *Reflexivity*

$$\frac{s \in A}{s = s \in A} \qquad \frac{S \in K}{S = S \in K} \qquad \frac{K}{K = K}$$

· *Symmetry*

$$\frac{s = t \in A}{t = s \in A} \qquad \frac{S = T \in K}{T = S \in K} \qquad \frac{K = L}{L = K}$$

· *Transitivity*

$$\frac{s = t \in A \quad t = u \in A}{s = u \in A} \qquad \frac{S = T \in K \quad T = U \in K}{S = U \in K} \qquad \frac{K = L \quad L = M}{K = M}$$

· *Equality*

$$\frac{s \in A \quad A = B \in Type}{s \in B} \qquad \frac{S \in K \quad K = L}{S \in L}$$

$$\frac{s = t \in A \quad A = B \in Type}{s = t \in B} \qquad \frac{S = T \in K \quad K = L}{S = T \in L}$$

We condense the next collection of rules by using our notation $p \in P$, $p = q \in P$ for judgements where P may be either an Order or a Type.

· *Substitution*

$$\frac{p \in P\ [\Gamma] \quad q \in Q\ [\Gamma', \xi \in P > \Gamma]}{q(p/\xi) \in Q(p/\xi)\ [\Gamma'(p/\xi) > \Gamma]} \qquad \frac{p = p' \in P\ [\Gamma] \quad q = q' \in Q\ [\Gamma', \xi \in P > \Gamma]}{q(p/\xi) = q'(p'/\xi) \in Q(p/\xi)\ [\Gamma'(p/\xi) > \Gamma]}$$

$$\frac{p \in P\ [\Gamma] \quad K\ [\Gamma', \xi \in P > \Gamma]}{K(p/\xi)\ [\Gamma'(p/\xi) > \Gamma]} \qquad \frac{p = p' \in P\ [\Gamma] \quad K = K'\ [\Gamma', \xi \in P > \Gamma]}{K(p/\xi) = K'(p'/\xi)\ [\Gamma'(p/\xi) > \Gamma]}$$

Finally we give the rules for the declaration of variables and extension of contexts.

· *Assumption*

$$\frac{A \in Type\ [\Gamma]}{x \in A\ [x \in A > \Gamma]} \qquad \frac{K\ [\Gamma]}{X \in K\ [X \in K > \Gamma]}$$

(under the assumption x and X respectively do not appear in Γ)

· *Weakening*

$$\frac{A \in Type\ [\Gamma] \quad J\ [\Gamma' > \Gamma]}{J\ [\Gamma', x \in A > \Gamma]} \qquad \frac{K\ [\Gamma] \quad J\ [\Gamma' > \Gamma]}{J\ [\Gamma', X \in K > \Gamma]}$$

(under the assumption x and X respectively do not appear in Γ or Γ')

Remark. The *Substitution*, *Assumption* and *Weakening clauses* give rise to a general principle of substitution which we now describe.

An *interpretation* of a context Γ in a context Δ consists of a function p associating to each judgement J in Γ an Operator or Term p_J (as appropriate) such that for each $J \in \Gamma$

$$p_J \in P_J(p_{J'}/\xi_{J'} \mid J' < J) \ [\Delta]$$

is a judgement of our theory.

General principle of substitution. *Whenever*

$$J \ [\Gamma]$$

is a judgement of our theory and p is an interpretation of Γ in Δ, then

$$J(p_{J'}/\xi_{J'} \mid J' \in \Gamma) \ [\Delta]$$

is also a judgement of our theory.

1.6. Constant Orders. There is a distinguished constant Order, $Type$, the Order of Types, and any number of other constant Orders which may be introduced at various stages of the recursive definition. These latter partly determine the signature of the theory. A constant Order has an "arity" given by Orders and Types for its (free) variables. The collection of *atomic* Orders is formed from the constant Orders by general substitution.

· *Type is an Order* $\dfrac{}{Type}$

(That is, we make the judgement $Type \in ORDER$ outright.)

· *Constant Orders* $\dfrac{}{L(\xi_1,\dots,\xi_n) \ [\Gamma]}$

(where all the variables ξ_i are declared in Γ)

Remarks.

(i) It is possible that in the *Constant Orders* clause, Γ declares more variables than appear in the list $\xi_1,\dots,\xi_n$.

(ii) Serious substitutions may make no visible difference to the main judgement. For example, if we introduce

$$L(y) \ [y \in B(x) > x \in A]$$

and a is a closed term of Type A, then by substitution we get

$$L(y) \ [y \in B(a)].$$

1.7. The structure of Orders and Operators. Clauses giving the closure properties of Orders (*formation clauses*) are naturally associated with clauses which give Operators or Terms of respectively the Orders or Types involved (*introduction and elimination clauses*); and these are naturally associated with clauses giving the fundamental equality judgements associated with the Operators or Terms (*equality clauses*). So we simultaneously give closure properties of Orders, of Operators and of Terms.

Orders are closed under "quantification" (that is, indexed sums and products) over both Types and Orders. We give first the clauses relating to "quantification" over Types.

- *Unit clauses:*
  - *formation* $\dfrac{}{1_O}$
  - *introduction* $\dfrac{}{()_O \in 1_O}$
  - *equality* $\dfrac{T \in 1_O}{T = ()_O \in 1_O}$
- *Sum clauses:*
  - *formation* $\dfrac{K \ [x \in A, \Gamma]}{\sum x \in A.K \ [\Gamma]}$ $\qquad \dfrac{A = A' \in Type \ [\Gamma] \quad K = K' \ [x \in A, \Gamma]}{\sum x \in A.K = \sum x \in A'.K' \ [\Gamma]}$
  - *introduction* $\dfrac{s \in A \quad S \in K(s/x)}{\langle s, S \rangle \in \sum x \in A.K}$ $\qquad \dfrac{s = s' \in A \quad S = S' \in K(s/x)}{\langle s, S \rangle = \langle s', S' \rangle \in \sum x \in A.K}$
  - *elimination* $\dfrac{T \in \sum x \in A.K}{\mathrm{fst}(T) \in A}$ $\qquad \dfrac{T = T' \in \sum x \in A.K}{\mathrm{fst}(T) = \mathrm{fst}(T') \in A}$

    $\dfrac{T \in \sum x \in A.K}{\mathrm{Snd}(T) \in K(\mathrm{fst}(T)/x)}$ $\qquad \dfrac{T = T' \in \sum x \in A.K}{\mathrm{Snd}(T) = \mathrm{Snd}(T') \in K(\mathrm{fst}(T)/x)}$
  - *equality* $\dfrac{s \in A \quad S \in K(s/x)}{\mathrm{fst}(\langle s, S \rangle) = s \in A}$ $\qquad \dfrac{s \in A \quad S \in K(s/x)}{\mathrm{Snd}(\langle s, S \rangle) = S \in K(s/x)}$

    $$\dfrac{T \in \sum x \in A.K}{\langle \mathrm{fst}(T), \mathrm{Snd}(T) \rangle = T \in \sum x \in A.K}$$
- *Product clauses:*
  - *formation* $\dfrac{K \ [x \in A, \Gamma]}{\prod x \in A.K \ [\Gamma]}$ $\qquad \dfrac{A = A' \in Type \ [\Gamma] \quad K = K' \ [x \in A, \Gamma]}{\prod x \in A.K = \prod x \in A'.K' \ [\Gamma]}$
  - *introduction* $\dfrac{S \in K \ [x \in A, \Gamma]}{\lambda x \in A.S \in \prod x \in A.K \ [\Gamma]}$ $\qquad \dfrac{A = A' \in Type \ [\Gamma] \quad S = S' \in K \ [x \in A, \Gamma]}{\lambda x \in A.S = \lambda x \in A'.S' \in \prod x \in A.K \ [\Gamma]}$
  - *elimination* $\dfrac{T \in \prod x \in A.K \quad s \in A}{Ts \in K(s/x)}$ $\qquad \dfrac{T = T' \in \prod x \in A.K \quad s = s' \in A}{Ts = T's' \in K(s/x)}$
  - *equality* $\dfrac{s \in A \ [\Gamma] \quad S \in K \ [x \in A, \Gamma]}{(\lambda x \in A.S)s = S(s/x) \in K(s/x)}$ $\qquad \dfrac{T \in \prod x \in A.K}{\lambda x \in A.Tx = T \in \prod x \in A.K}$

Now we consider the analogous clauses giving the closure of Orders under "quantification" over Orders.

- *Sum clauses:*
  - *formation* $\dfrac{L \ [X \in K, \Gamma]}{\sum X \in K.L \ [\Gamma]}$ $\qquad \dfrac{L = L' \ [\Gamma] \quad K = K' \ [X \in K, \Gamma]}{\sum X \in K.L = \sum X \in K'.L' \ [\Gamma]}$
  - *introduction* $\dfrac{S \in K \quad T \in L(S/X)}{\langle S, T \rangle \in \sum X \in K.L}$ $\qquad \dfrac{S = S' \in K \quad T = T' \in L(S/X)}{\langle S, T \rangle = \langle S', T' \rangle \in \sum X \in K.L}$

· *elimination*

$$\frac{U\in\sum X\in K.L}{\mathrm{Fst}(U)\in K} \qquad \frac{U=U'\in\sum X\in K.L}{\mathrm{Fst}(U)=\mathrm{Fst}(U')\in K}$$

$$\frac{U\in\sum X\in K.L}{\mathrm{Snd}(U)\in K(\mathrm{Fst}(U)/X)} \qquad \frac{U=U'\in\sum X\in K.L}{\mathrm{Snd}(U)=\mathrm{Snd}(U')\in K(\mathrm{Fst}(U)/X)}$$

· *equality*

$$\frac{S\in K \quad T\in L(S/X)}{\mathrm{Fst}(\langle S,T\rangle)=S\in K} \qquad \frac{S\in K \quad T\in L(S/X)}{\mathrm{Snd}(\langle S,T\rangle)=T\in L(S/X)}$$

$$\frac{U\in\sum X\in K.L}{\langle\mathrm{Fst}(U),\mathrm{Snd}(U)\rangle=U\in\sum X\in K.L}$$

· *Product clauses*:

· *formation*

$$\frac{L\ [X\in K,\Gamma]}{\prod X\in K.L\ [\Gamma]} \qquad \frac{K=K'\ [\Gamma] \quad L=L'\ [X\in K,\Gamma]}{\prod X\in K.L=\prod X\in K'.L'\ [\Gamma]}$$

· *introduction*

$$\frac{T\in L\ [X\in K,\Gamma]}{\lambda X\in K.T\in\prod X\in K.L\ [\Gamma]} \qquad \frac{K=K'\ [\Gamma] \quad T=T'\in L\ [X\in K,\Gamma]}{\lambda X\in K.T=\lambda X\in K'.T'\in\prod X\in K.L\ [\Gamma]}$$

· *elimination*

$$\frac{U\in\prod X\in K.L \quad S\in K}{US\in L(S/X)} \qquad \frac{U=U'\in\prod X\in K.L \quad S=S'\in K}{US=U'S'\in L(S/X)}$$

· *equality*

$$\frac{S\in K\ [\Gamma] \quad T\in L\ [X\in K,\Gamma]}{(\lambda X\in K.T)S=T(S/X)\in L(S/X)} \qquad \frac{U\in\prod X\in K.L}{\lambda X\in K.UX=U\in\prod X\in K.L}$$

Remarks.

(*i*) The only way in which we have varied the presentation from that of Martin-Löf [M-L2] is in the elimination and equality rules for sum types. There we have followed Seely [Se1] and used constants for first and second projection rather than "elimination" constants. The latter would involve using the following rules:

$$\frac{S\in\sum x\in A.K\ [\Gamma] \quad p\in P(\langle x,X\rangle/Z)\ [X\in K,x\in A,\Gamma]}{\mathrm{E}(S,(x,X).p)\in P(S/Z)\ [\Gamma]}$$

$$\frac{S=S'\in\sum x\in A.K\ [\Gamma] \quad p=p'\in P(\langle x,X\rangle/Z)\ [X\in K,x\in A,\Gamma]}{\mathrm{E}(S,(x,X).p)=\mathrm{E}(S',(x,X).p')\in P(S/Z)\ [\Gamma]}$$

$$\frac{s\in A\ [\Gamma] \quad S\in K(s/x)\ [\Gamma] \quad p\in P(\langle x,X\rangle/Z)\ [X\in K,x\in A,\Gamma]}{\mathrm{E}(\langle s,S\rangle,(x,X).p)=p(s/x,S/X)\in P(\langle s,S\rangle/Z)\ [\Gamma]}$$

$$\frac{S\in\sum x\in A.K\ [\Gamma] \quad p\in P\ [Z\in\sum x\in A.K,\Gamma]}{\mathrm{E}(S,(x,X).p(\langle x,X\rangle/Z))=p(S/Z)\in P(S/Z)\ [\Gamma]}$$

for sums of Orders over Types and similar rules for sums of Orders over Orders. This formulation is equivalent to the one we have given as regards equalities (though not as regards reductions): in one direction we can define $\mathrm{E}(S,(x,X).p)$ to be $p(\mathrm{fst}(S)/x,\mathrm{Snd}(S)/X)$ and derive the above rules from the *Sum clauses*; and in the other direction we can define $\mathrm{fst}(S)$ to be $\mathrm{E}(S,(x,X).x)$, $\mathrm{Snd}(S)$ to be $\mathrm{E}(S,(x,X).X)$ and derive the *Sum clauses* from the above rules. (The first three of the above rules appear in [M-L2], where they are used to derive the rules involving "fst" and "Snd" *in the presence of rules for equality types.* We do not introduce equality types here because the models we are going to consider in sections 4 and 5 do not support them—or at least do so only with very weak rules.)

(*ii*) As in [M-L2], we can have notations $A\times K$ and $A\to K$ for $\sum x\in A.K$ and $\prod x\in A.K$ respectively when x is not free in K; and notations $K\times L$ and $K\to L$ for $\sum X\in K.L$ and $\prod X\in K.L$ respectively when X is not free in K. Note that for any Type A, the Order $A\times 1_O$ is "essentially equivalent" to A: up to provable equality, there is a bijective correspondence between Terms of Type A and Operators of Order $A\times 1_O$.

(*iii*) Strictly speaking, the notation $\langle s,S\rangle$, $\langle S,T\rangle$ for members of sum Orders, is not satisfactory because of the ambiguity involved in $K(s/x)$, $K(S/X)$. An unambiguous notation is preferred by computer scientists (see [MiPl] for example).

1.8. Constant Operators. We can introduce any number of constant Operators of various Orders at appropriate stages of the recursive definition. These partly determine the signature. A constant Operator has "arity" given by Orders and Types for its (free) variables. The collection of *atomic* Operators is formed from the constant Operators by general substitution.

· *Constant Operators*
$$\frac{K\ [\Gamma]}{T(\xi_1,\ldots,\xi_n)\in K\ [\Gamma]}$$
(where all the variables ξ_i are declared in Γ)

Remark. It is worth noting however that there is no real call for constant Operators *with arities*. For example, instead of introducing

$$T(x)\in K(x)\ [x\in A]$$

we can introduce

$$T\in\prod x\in A.K\ .$$

Then the elimination clause for products will enable us to recapture $T(x)$ as Tx. (Note however, that there is one problem with this procedure. If Γ declares more variables than appear in $T(\xi_1,\ldots,\xi_n)$, we will want to declare explicitly that T is constant in some arguments. We have to do this by adding appropriate equality axioms—see 1.10.)

1.9. The structure of Types and Terms. As for Orders and Operators in 1.7, so for Types and terms here we have *formation, introduction, elimination* and *equality* clauses.

Just as for Orders, Types are closed under "quantification" (indexed sums and products) over both Types and Orders: there is an important difference however in the treatment of sums indexed over Orders. We treat first the clauses relating to "quantification" over Types.

- *Unit clauses*:

  - *formation* $$\frac{}{1_T \in Type}$$

  - *introduction* $$\frac{}{()_T \in 1_T}$$

  - *equality* $$\frac{t \in 1_T}{t = ()_T \in 1_T}$$

- *Sum clauses*:

  - *formation* $$\frac{B \in Type\ [x \in A, \Gamma]}{\sum x \in A . B \in Type\ [\Gamma]} \qquad \frac{A = A' \in Type\ [\Gamma] \quad B = B' \in Type\ [x \in A, \Gamma]}{\sum x \in A . B = \sum x \in A' . B' \in Type\ [\Gamma]}$$

  - *introduction* $$\frac{s \in A \quad t \in B(s/x)}{\langle s,t \rangle \in \sum x \in A . B} \qquad \frac{s = s' \in A \quad t = t' \in B(s/x)}{\langle s,t \rangle = \langle s',t' \rangle \in \sum x \in A . B}$$

  - *elimination* $$\frac{u \in \sum x \in A . B}{\mathsf{fst}(u) \in A} \qquad \frac{u = u' \in \sum x \in A . B}{\mathsf{fst}(u) = \mathsf{fst}(u') \in A}$$
    $$\frac{u \in \sum x \in A . B}{\mathsf{snd}(u) \in B(\mathsf{fst}(u)/x)} \qquad \frac{u = u' \in \sum x \in A . B}{\mathsf{snd}(u) = \mathsf{snd}(u') \in B(\mathsf{fst}(u)/x)}$$

  - *equality* $$\frac{s \in A \quad t \in B(s/x)}{\mathsf{fst}(\langle s,t \rangle) = s \in A} \qquad \frac{s \in A \quad t \in B(s/x)}{\mathsf{snd}(\langle s,t \rangle) = t \in B(s/x)}$$
    $$\frac{u \in \sum x \in A . B}{\langle \mathsf{fst}(u), \mathsf{snd}(u) \rangle = u \in \sum x \in A . B}$$

- *Product clauses*:

  - *formation* $$\frac{B \in Type\ [x \in A, \Gamma]}{\prod x \in A . B \in Type\ [\Gamma]} \qquad \frac{A = A' \in Type\ [\Gamma] \quad B = B' \in Type\ [x \in A, \Gamma]}{\prod x \in A . B = \prod x \in A' . B' \in Type\ [\Gamma]}$$

  - *introduction* $$\frac{t \in B\ [x \in A, \Gamma]}{\lambda x \in A . t \in \prod x \in A . B\ [\Gamma]} \qquad \frac{A = A' \in Type \quad t = t' \in B\ [x \in A, \Gamma]}{\lambda x \in A . t = \lambda x \in A' . t' \in \prod x \in A . B\ [\Gamma]}$$

  - *elimination* $$\frac{u \in \prod x \in A . B \quad s \in A}{us \in B(s/x)} \qquad \frac{u = u' \in \prod x \in A . B \quad s = s' \in A}{us = u's' \in B(s/x)}$$

  - *equality* $$\frac{s \in A\ [\Gamma] \quad t \in B\ [x \in A, \Gamma]}{(\lambda x \in A . t)s = t(s/x) \in B(s/x)} \qquad \frac{u \in \prod x \in A . B}{\lambda x \in A . ux = u \in \prod x \in A . B}$$

Now we treat the clauses giving the closure of Types under "quantification" over Orders.

- *Sum clauses*:

  - *formation* $$\frac{A \in Type\ [X \in K, \Gamma]}{\sum X \in K . A \in Type\ [\Gamma]} \qquad \frac{K = K'\ [\Gamma] \quad A = A' \in Type\ [X \in K, \Gamma]}{\sum X \in K . A = \sum X \in K' . A'\ [\Gamma]}$$

· *introduction* $\dfrac{S\in K \quad s\in A(S/X)}{\langle S,s\rangle \in \sum X\in K.A}$ $\qquad \dfrac{S=S'\in K \quad s=s'\in A(S/X)}{\langle S,s\rangle = \langle S',s'\rangle \in \sum X\in K.A}$

· *elimination* $\dfrac{s\in \sum X\in K.A \ [\Gamma] \quad t\in B(\langle X,x\rangle/z)\ [x\in A, X\in K,\Gamma]}{\mathrm{E}(s,(X,x).t)\in B(s/z)\ [\Gamma]}$

$\dfrac{s=s'\in \sum X\in K.A \ [\Gamma] \quad t=t'\in B(\langle X,x\rangle/z)\ [x\in A, X\in K,\Gamma]}{\mathrm{E}(s,(X,x).t)=\mathrm{E}(s',(X,x).t')\in B(s/z)\ [\Gamma]}$

· *equality* $\dfrac{S\in K\ [\Gamma] \quad s\in A(S/X)\ [\Gamma] \quad t\in B(\langle X,x\rangle/z)\ [x\in A, X\in K,\Gamma]}{\mathrm{E}(\langle S,s\rangle,(X,x).t)=t(S/X,s/x)\in B(\langle S,s\rangle/z)\ [\Gamma]}$

$\dfrac{s\in \sum X\in K.A\ [\Gamma] \quad t\in B\ [z\in \sum X\in K.A,\Gamma]}{\mathrm{E}(s,(X,x).t(\langle X,x\rangle/z))=t(s/z)\in B(s/z)\ [\Gamma]}$

· *Product clauses*:

· *formation* $\dfrac{A\in Type\ [X\in K,\Gamma]}{\prod X\in K.A\in Type\ [\Gamma]}$ $\qquad \dfrac{K=K'\ [\Gamma] \quad A=A'\in Type\ [X\in K,\Gamma]}{\prod X\in K.A=\prod X\in K'.A'\in Type\ [\Gamma]}$

· *introduction* $\dfrac{s\in A\ [X\in K,\Gamma]}{\lambda X\in K.s\in \prod X\in K.A\ [\Gamma]}$ $\qquad \dfrac{K=K'\ [\Gamma] \quad s=s'\in A\ [X\in K,\Gamma]}{\lambda X\in K.s=\lambda X\in K'.s'\in \prod X\in K.A\ [\Gamma]}$

· *elimination* $\dfrac{t\in \prod X\in K.A \quad S\in K}{tS\in A(S/X)}$ $\qquad \dfrac{t=t'\in \prod X\in K.A \quad S=S'\in K}{tS=t'S'\in A(S/X)}$

· *equality* $\dfrac{s\in A\ [X\in K,\Gamma] \quad S\in K\ [\Gamma]}{(\lambda X\in K.s)S=s(S/X)\in A(S/X)\ [\Gamma]}$ $\qquad \dfrac{t\in \prod X\in K.A}{\lambda X\in K.tX=t\in \prod X\in K.A}$

Finally, we can introduce any number of constant Terms at appropriate stages of the recursive definition. These partly determine the signature. A constant Term has "arity" given by Orders and Types for its (free) variables. (The remark we made in 1.8 about constant Operators *with arities* applies equally well here to constant Terms.) The collection of *atomic* Terms is formed from the constant Terms by general substitution.

· *Constant Terms* $\dfrac{A\in Type\ [\Gamma]}{f(\xi_1,\ldots,\xi_n)\in A\ [\Gamma]}$

Remarks.

(*i*) In the formulation of "quantification" over Orders, the clauses for sum elimination and equality resemble those familiar from Martin-Löf's presentations (with the exception of the last equality rule—*cf*. Remark (*i*) in 1.7). But the two different levels of "types" (Types and Orders) introduce a subtle distinction at this point. We are referring to the fact that in the *elimination* and *equality clauses* for sums of Types over Orders, B is only a Type and not an Order. Consequently it is not even possible to define a first projection "Fst" as we did in Remark (*i*) of 1.7, so that the rules given are no longer equivalent to a formulation involving projections. (We will discuss in 1.11 what happens if one strengthens the rules by replacing B by an Order K.) Nevertheless, the sum rules we have given are still strictly stronger than those in

Girard's original formulation of higher order lambda calculus (and so also stronger than the equivalent formulation in [MiPl]): we will explain why when we discuss their category theoretic interpretation in 2.12.

(*ii*) Just as in the case of quantification of Orders, we write $A\times B$ and $A\to B$ for $\sum x\in A.B$ and $\prod x\in A.B$ respectively when x is not free in B; and similarly write $K\times A$ and $K\to A$ for $\sum X\in K.A$ and $\prod X\in K.A$ respectively when X is not free in B. In particular, we can associate to any Order K a Type $K\times 1_T$. Unlike the situation in 1.7 for Types (where the stronger *Sum clauses* imply that the Type A is essentially equivalent to the Order $A\times 1_O$), it is *not* the case that K is essentially equivalent to $K\times 1_T$. Rather, the process of sending K to $K\times 1_T$ sets up a reflection of Orders into Types: this will be discussed further in 2.12.

(*iii*) Occurences of the variables X,x in t are *bound* in the elimination term $\mathrm{E}(s,(X,x).t)$—that is, $FV(\mathrm{E}(s,(X,x).t))=FV(t)\setminus\{X,x\}\cup FV(s)$. The notation we use for this elimination term is Martin-Löf's; it is sometimes written as

let (X,x) **be** s **in** t

which conveys its intended meaning better, but is a less convenient notation in compound expressions.

1.10. Theories. We have now described the *basic theory* of constructions. A *theory* over this basic one is obtained essentially by adding as axioms equality judgements of the kind

$$S=T\in K \text{ or } s=t\in A$$

(but *not* of the kind "$K=L$"—see the Remark below). Thus if we have for example $s\in A\ [\Gamma]$ and $t\in A\ [\Gamma]$ in the basic theory, then

$$s=t\in A\ [\Gamma]$$

is a possible axiom. However, the recursive definition of judgements means that the situation is more complicated. By adding equality judgements as axioms, one generates new structural judgements which lead to the possibility of new equality judgements as axioms. So a theory **T** should consist of an ordinal indexed family τ_α of sets of judgements such that for each

$$p=q\in P\ [\Gamma]$$

in τ_α, the judgements

$$p\in P\ [\Gamma] \text{ and } q\in P\ [\Gamma]$$

are derivable from the basic theory of constructions plus the axioms in $\bigcup\{\tau_\beta \mid \beta<\alpha\}$.

Remark. Note that we are specifically excluding the possibility of having equality judgements of the form "$K=L\in ORDER$" in our theories. This means that the weak, definitional role played by this form of judgement in the basic theory is carried over to equational extensions. What we gain by this restriction is the possibility (outlined in 2.7 and 2.13) of a perfect correspondence between theories and instances of the kind of categorical structure to be described in the next section. Moreover, there is little loss of expressive power since the equality "$K=L$" can be simulated by an *isomorphism* $K \cong L$—by introducing Operators $S \in (K \to L)$, $T \in (L \to K)$ together with axioms saying that S and T are mutually inverse. This technique is used in the following extension of the basic theory:

1.11. Theory of constructions with "*Order* ∈ *ORDER*". We introduce a constant Order *Order* together with rules that make it a *universal* Order, or "Order of all Orders". Specifically we introduce constant Orders and constant Operators as follows:

$$\frac{}{Order} \qquad \frac{}{O(X)\ [X \in Order]} \qquad \frac{K\ [\Gamma]}{T_K(\xi) \in Order\ [\Gamma]}$$

$$\frac{K\ [\Gamma]}{I_K(\xi) \in O(T_K(\xi)) \to K\ [\Gamma]} \qquad \frac{K\ [\Gamma]}{J_K(\xi) \in K \to O(T_K(\xi))\ [\Gamma]}$$

where $\xi = \xi_1, \ldots, \xi_n$ are the variables in the maximal declarations of Γ. Then whenever $K\ [\Gamma]$ is derivable, we introduce the axioms:

$$J_K(\xi)(I_K(\xi)X) = X \in O(T_K(\xi)) \qquad [X \in O(T_K(\xi)), \Gamma]$$

and

$$Y = I_K(\xi)(J_K(\xi)Y) \in K\ [Y \in K, \Gamma].$$

Thus *Order* is an Order of "names" of Orders: $O(X)$ is the Order named by $X \in Order$ and for each Order K there is a name $T_K \in Order$ whose corresponding Order $O(T_K)$ is isomorphic to K via I_K and J_K.

Of course this extension to the theory of constructions has some very odd consequences. For one thing we can carry out Girard's Paradox in it (see [Coq2]) and hence in particular every Order possesses a closed Operator. However, the system is very far from being contradictory (in the sense of all Operators of any particular Order being provably equal). Indeed the topos-theoretic models which we present in sections 4 and 5 are both very rich models of the theory of constructions with "*Order* ∈ *ORDER*". Moreover it is possible to make an even more radical extension of the theory without entailing contradiction:

1.12. Theory of constructions with "*Type* ≃ *ORDER*". We now consider what happens if we strengthen the basic theory of constructions by replacing the Type B by an Order K in the *elimination* and *equality clauses* for sums of Types indexed over Orders in 1.9. The resulting system does extend the original one because of the correspondence of Terms of Type B with Operators of Order $B \times 1_O$ noted in Remark (*ii*) of 1.7. We call the strengthened system the *theory of constructions with "Type ≃ ORDER"* for reasons which we now explain:

First note that Remark (*ii*) of 1.9 no longer applies and we can now define

$$\mathrm{Fst}(s) = \mathrm{E}(s,(X,x).X) \text{ and } \mathrm{snd}(s) = \mathrm{E}(s,(X,x).x)$$

and derive elimination and equality rules for $\sum X{\in}K.A$ entirely analogous to those for $\sum x{\in}A.K$ in 1.7. As a consequence we get a bijective correspondence up to provable equality between Operators of Order K and Terms of Type $K\times 1_T$. In fact the collections of Types and of Orders can essentially be identified, since the operations

$$A\in Type \longmapsto A\times 1_O \in ORDER \qquad \text{and} \qquad K\in ORDER \longmapsto K\times 1_T \in Type$$

establish an equivalence between Types and Orders: A is naturally isomorphic to $(A\times 1_O)\times 1_T$ and K is naturally isomorphic to $(K\times 1_T)\times 1_O$. One consequence of this identification of Types and Orders is that there is a universal Order, namely $Order = Type$. In other words the theory of constructions with "$Type \simeq ORDER$" manages to model (in a very strong way) the theory of constructions with "$Order \in ORDER$".

Note also that the Type $(Type\times 1_T)\in Type$ acts as a "type of all types" since the Terms of Type $Type\times 1_T$ correspond bijectively up to equality to Operators of Order $Type$, that is, to Types. So the theory of constructions with "$Type \simeq ORDER$" also models a theory "$type \in Type$" of a universal Type which is like 1.11 "moved one level down" and which we do not give explicitly here.

Despite its strength, the theory of constructions with "$Type \simeq ORDER$" is not contradictory. We will see in section 4 that the algebraic toposes provide a highly non-trivial model of this theory.

2 Categorical interpretations of type theories

2.1. Introduction. Our aim in this section is to give a plausible explanation of what categorical models of the theory of constructions look like. The fundamental idea behind categorical models of simple type theories is that the objects of a category will model the types of a theory, while the morphisms will model the terms. (One of the simplest significant cases is the well-known connection between the typed lambda calculus and cartesian closed categories; this is explained very fully in the book of Lambek and Scott [LS].) When one attempts to model more complicated type theories, especially those involving *dependent types*, then this fundamental idea of objects for types and morphisms for terms needs some elaboration. Therefore we first give a description of categorical models for calculi of dependent types.

This serves to describe models for the pure theory of Orders and Operators and for the pure theory of Types and Terms. However, Orders and Types are not independent of each other. In the first place we can take sums and products of each indexed over the other; and secondly, Type is an Order. Therefore we successively add on features required to model these ways in which Orders and Types interact, finally obtaining a complete description of categorical models for the theory of constructions.

Whilst our explanation of categorical models for the theory of constructions is based upon a categorical explanation of Martin-Löf type theory, there are many other interesting fragments of the theory of constructions whose categorical models we do not consider separately. (Explaining the models for a stronger system does not entail explaining the models for a weaker one.) In particular we regret not being able to comment usefully on models for the second and higher order lambda calculus. The reader will find accounts of categorical models for these calculi in [Se2] and [Pi2].

In the final part of this section we indicate what is required to model the extensions of the basic theory of constructions considered in 1.11 and 1.12.

2.2. Dependent types. Martin-Löf type theory is a calculus of *dependent types*, so before we can consider the rules for unit, sums and products we must explain how that notion is to be modelled categorically. So let us restrict our attention to that part of the theory presented in section 1 which is concerned with the pure theory of Types and Terms, without unit, sums or products. This leaves the general rules for Types and Terms (part of 1.5), together with rules for introducing constant Types (special case of 1.6) and constant Terms (last part of 1.9). We could just as well restrict to the pure theory of Orders and Operators. To use Cartmell's terminology [Ca], this is the "generalized algebraic" fragment of Martin-Löf's theory of dependent types.

Cartmell [Ca], Obtułowicz [Ob] and Seely [Se1] have given accounts of categorical models for calculi of dependent types. As we will explain shortly, Cartmell's notion of "contextual category" is too rigid for our purposes; and Obtułowicz's notion (involving a hierarchy of indexed categories) is for us unnecessarily general. The notion we need can be obtained by adapting the analysis in [Se1]. We consider the following category-theoretic structure:

- A category **B** with finite products. (The product of objects I and J in **B** will be denoted $I\times J$ with projection morphisms $\pi_1 : I\times J \longrightarrow I$ and $\pi_2 : I\times J \longrightarrow J$; the terminal object in **B** will be denoted 1.)
- A collection **A** of morphisms of **B** satisfying the following condition:

  Stability. *If $a : A \longrightarrow I$ is in* **A**, *$f : J \longrightarrow I$ is in* **B** *and*

$$\begin{array}{ccc} B & \longrightarrow & A \\ {\scriptstyle b}\downarrow & \lrcorner & \downarrow{\scriptstyle a} \\ J & \xrightarrow[f]{} & I \end{array}$$

  is a pullback square in **B**, *then $b : B \longrightarrow J$ is in* **A**; *moreover there is a pullback square for each such a and f.*

Note that we are not assuming that **B** has *all* pullbacks. **A** should be regarded as the class of objects of a full subcategory, $\mathbf{A} \hookrightarrow \mathbf{B}^2$, of the arrow category $\mathbf{B}^2$. Composing with the codomain functor $cod : \mathbf{B}^2 \longrightarrow \mathbf{B}$ we obtain a functor $\mathbf{A} \longrightarrow \mathbf{B}$ which is a categorical *fibration* (*cf.* [B2]). If **B** does in fact have all pullbacks then $cod : \mathbf{B}^2 \longrightarrow \mathbf{B}$ is itself a fibration and we have a full and faithful cartesian functor over **B**:

$$\begin{array}{ccc} \mathbf{A} & \hookrightarrow & \mathbf{B^2} \\ & \searrow \quad \swarrow_{cod} & \\ & \mathbf{B} & \end{array} .$$

From this point of view **A** determines a "full subcategory of **B** as seen from **B**"—although not necessarily a "definable" full subcategory in the sense of [B2]. Even if $cod:\mathbf{B^2}\longrightarrow\mathbf{B}$ is not a fibration, it is convenient to say when **A** satisfies the **Stability** assumption above that

A *is a full subcategory of* $\mathbf{B^2}$ *over* **B**.

We need some notation related to the category-theoretic structure just described. Recall that the *slice* of **B** by an object I, denoted $\mathbf{B}/I$, is the category whose objects are morphisms in **B** with codomain I and whose morphisms are commutative triangles in **B** with vertex I. For each object I in **B**, we write

$\mathbf{A}(I)$

for the full subcategory of $\mathbf{B}/I$ whose objects are the morphisms with codomain I which lie in **A**; this is precisely the fibre over I of the fibration $\mathbf{A}\longrightarrow\mathbf{B}$. Note in particular that we can identify $\mathbf{B}/1$ with **B**, in which case $\mathbf{A}(1)$ becomes a full subcategory of **B** in the usual sense: its objects are those $I\in\mathbf{B}$ for which the unique morphism $I\longrightarrow 1$ is in **A**. Given $f:J\longrightarrow I$ in **B**, we write

$f^{\#}:\mathbf{A}(I)\longrightarrow\mathbf{A}(J)$

for the pullback functor. We are tacitly assuming that a choice of such pullback functors is given along with the information specifying **A** (which is to say, in the language of fibrations, that $\mathbf{A}\longrightarrow\mathbf{B}$ is a "cloven" fibration); but note that the way we have phrased the **Stability** condition implies that *we can compose a morphism in* **A** *with an isomorphism on either side and still remain in* **A**. We use the above notation for pullback functors rather than the more usual "f^*" only because in sections 4 and 5, f will be a geometric morphism between toposes, in which case "f^*" conventionally denotes the inverse image part of f. However, applying the functor $f^{\#}$ to an object $A\longrightarrow I$ of $\mathbf{A}(I)$, the *domain* of the resulting object of $\mathbf{A}(J)$ will as usual be denoted by $J\times_I A$; in other words our standard notation for a pullback square in **B** is:

$$\begin{array}{ccc} J\times_I A & \longrightarrow & A \\ {\scriptstyle f^{\#}(a)}\downarrow & \lrcorner & \downarrow{\scriptstyle a} \\ J & \xrightarrow[f]{} & I \end{array} .$$

We now indicate in general terms how a category **B** with finite products equipped with a full subcategory $\mathbf{A}\hookrightarrow\mathbf{B^2}$ over **B** in the above sense, serves to interpret the calculus of dependent types. The interpretation is based upon having a "structure" in **B** for the language—in other words, having a particular choice of interpretation for the constant Types and Terms: a constant Type of arity n is interpreted by a morphism in the class **A** of the form $J \longrightarrow I_1 \times \ldots \times I_n$; and a constant Term of arity n is interpreted by a morphism in **B** of the form $I_1 \times \ldots \times I_n \longrightarrow J$. (In case $n=0$, this means that a constant Type which is dependent on no variables is essentially interpreted as an object J of **B**, but one for which the unique morphism $J \longrightarrow 1$ is in **A**; and a constant Term depending upon no variables is interpreted as a global section $1 \longrightarrow J$ of an object in **B**.)

As formulated in section 1, the calculus consists of a collection of rules for deriving judgements (in contexts) about certain expressions. These judgements-in-context not only assert the typing and the equality of expressions, but also tell us which expressions are well-formed; moreover, these three kinds of assertion are built up simultaneously by mutual recursion. Therefore, it is not possible *first* to give a recursive definition of the denotations of well-formed expressions as objects and morphisms of **B** and *then* to give a recursive definition of when the various kinds of judgement are satisfied by the categorical structure. Instead one has to give a single definition by recursion on the derivation of a judgement of its satisfaction and (depending on the form of judgement) of the denotations of its constituent expressions which are used to express this satisfaction. We consider what is involved for each form of judgement in turn:

A judgement of the form $A \in Type\ [\Gamma]$ has the force that, in the context Γ, A is a well-formed Type. Its satisfaction then amounts to having built up a morphism in **A** of the form

$$[\![A]\!] \longrightarrow [\![\Gamma]\!] =_{def} \prod_{J} [\![A_J]\!]$$

where the codomain is a (finite) product indexed by the *maximal* elements J of the context Γ and A_J is the Type of the variable declared in J.

A judgement of the form $A=B \in Type\ [\Gamma]$ has the force that, in the context Γ, A and B are equal (well-formed) Types. It is satisfied if $[\![A]\!] \longrightarrow [\![\Gamma]\!]$ and $[\![B]\!] \longrightarrow [\![\Gamma]\!]$ are equal morphisms in **A**.

A judgement of the form $s \in A\ [\Gamma]$ has the force that s is a (well-formed) Term of Type A. As all relevant variables appear in the context Γ, the satisfaction of the judgement amounts to having built up a morphism

$$[\![s]\!] : [\![\Gamma]\!] \longrightarrow [\![A]\!]$$

in **B** which is a section of the **A**-morphism $[\![A]\!] \longrightarrow [\![\Gamma]\!]$ (that is, whose composition with this morphism is the identity on $[\![\Gamma]\!]$.)

Finally, a judgement of the form $s=t \in A\ [\Gamma]$ has the force that s and t are equal Terms of Type A. It is satisfied if $[\![s]\!] : [\![\Gamma]\!] \longrightarrow [\![A]\!]$ and $[\![t]\!] : [\![\Gamma]\!] \longrightarrow [\![A]\!]$ are equal morphisms in **B**.

The actual clauses of the recursive definition of satisfaction of judgements are rather straightforward for the pure theory of dependent types which we are considering at the moment: the only significant rule is that for *Substitution* and this is handled using the pullback functors f^{*}. (Contexts and variable declarations are handled by finite products and projections, as above.)

2.3. Remarks.

(*i*) The idea of using a special class of morphisms closed under (at least) pullback is not new. For example Bénabou's notion of a "catégorie calibrée" in [B1] is the above notion with the addition of the condition for **Sums** from 2.4 and a further condition (his (P3)) which makes the special class of morphisms like a class of local homeomorphisms. Our approach was inspired by the notion of "display map" in Taylor's thesis [Ta], which is essentially our notion with the addition of the **Unit** and **Sums** conditions from 2.4 and the **Display** condition from 2.7.

(*ii*) The idea of a special class of maps is also the basis for Cartmell's *contextual categories* in [Ca]. Cartmell has a canonical choice of morphisms in **A** to represent the dependent types. Up to equivalence of categories, there is not much to choose between the two approaches. A contextual category gives rise to one of our subcategories (also satisfying the **Unit**, **Sums** and **Display** conditions below) just by closing the set of canonical morphisms under isomorphism. On the other hand, one of our subcategories (satisfying **Unit**, **Sums** and **Display**) can be "unravelled" to give an equivalent contextual category with the necessary canonical choices.

(*iii*) The version of Martin-Löf type theory which Seely models has equality types and strong rules for equality as in [M-L1], as well as strong rules for sums. Equality is interpreted via equalizers in the category. It follows that *every* morphism

$$[\![s]\!] : [\![C]\!] \longrightarrow [\![A]\!]$$

in **B** interpreting a term s can be thought of as also representing an indexed family

$$B(x) \;\; [x \in A]$$

of types over A: one simply takes $B(x)$ to be

$$\Sigma y \in C \, . \, I_A(s(y), x) \;\; .$$

where I_A is the equality type for A. (See Sublemma 3.2.3.2 of [Se1].) So in this case *every* morphism of **B** should be in **A**. Without strong equality types—which we emphatically do not have in the topos-theoretic models presented in sections 4 and 5—one does not have this phenonomen. On the other hand, as we explain below, we are still able to model indexed families

$$B(x) \;\; [x \in A]$$

by morphisms $[\![B(x)]\!] \longrightarrow [\![A]\!]$ (in **A**) which also represent terms ("first projection" in this case).

2.4. Martin-Löf type theory. We take the basic rules of Martin-Löf type theory to be the *Unit*, *Sum* and *Product clauses* for "quantification" of Types over Types as given in 1.9. (In his usual presentation, Martin-Löf has the unit rules appearing amongst the rules for *finite types*. We regard them as an essential part of the system: they denote an essential part of the semantics—*cf*. 2.8.)

The theory of constructions contains two instances of these basic rules: one in 1.7 for the pure theory of Orders and one in 1.9 for the pure theory of Types. We now consider what properties of a full subcategory **A** of $\mathbf{B}^2$ over **B** (as introduced in 2.2) are needed to model the basic rules.

For the *Unit clauses*, firstly *formation* gives us an object $[\![1_T]\!]$ in **B** for which the unique morphism $[\![1_T]\!] \longrightarrow 1$ is in **A**; and *introduction* gives us a morphism $[\![\langle\rangle_T]\!]: 1 \longrightarrow [\![1_T]\!]$ in **B** which is a section of $[\![1_T]\!] \longrightarrow 1$, i.e. is a right-sided inverse for it; but finally, *equality* implies that it is actually a two-sided inverse—since we can apply the clause in the context $\Gamma = [x \in 1_T]$ (for which $[\![\Gamma]\!] = [\![1_T]\!]$), to conclude that $\pi_1 : [\![\Gamma]\!] \times [\![1_T]\!] \longrightarrow [\![\Gamma]\!]$ has a *unique* section and hence that $[\![1_T]\!] \longrightarrow 1 \longrightarrow [\![1_T]\!]$ is the identity. So we conclude not only that $[\![1_T]\!]$ is isomorphic to the terminal object of **B**, but also that **A** contains an isomorphism with codomain 1. Because of the **Stability** condition in 2.2, this is equivalent to requiring:

Unit. **A** *contains all the isomorphisms of* **B**.

Turning to the *Sum clauses*, categorically, indexed sums provide (stable) left adjoints to pullback functors. Suppose that $f : [\![B(x)]\!] \longrightarrow [\![A]\!]$ is the morphism in **A** interpreting $B(x)\ [x \in A]$. Then the left adjoint $f_!$ to $f^{\#}$ corresponds to taking

$$C(x,y)\ [y \in B(x) > x \in A]$$

to

$$\textstyle\sum y{\in}B(x)\,.\,C(x,y)\ [x \in A]\ .$$

So we need left adjoints

$$f_! : \mathbf{A}(J) \longrightarrow \mathbf{A}(I)$$

to the pullback functor $f^{\#} : \mathbf{A}(I) \longrightarrow \mathbf{A}(J)$ for all morphisms $f : J \longrightarrow I$ *which are in* **A**. However, the rules for sums give something more. We have of course the unit of the adjunction, which syntactically is essentially

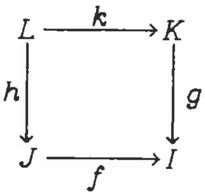
$$z \in C(x,y) \longmapsto \langle y,z \rangle \in \textstyle\sum y{\in}B(x)\,.\,C(x,y)$$

and makes the diagram

$$\begin{array}{ccc} [\![C(x,y)]\!] & \longrightarrow & [\![\sum y{\in}B(x)\,.\,C(x,y)]\!] \\ \downarrow & & \downarrow \\ [\![B(x)]\!] & \longrightarrow & [\![A]\!] \end{array}$$

commute. "Snd" provides an inverse for this map, so it is an isomorphism. Consequently the indexed sums in **A** should be the standard ones given by *composition*. (If the pullback functor $f^{\#} : \mathbf{B}/I \longrightarrow \mathbf{B}/J$ exists, then it automatically has a left adjoint given by composing with $f : J \longrightarrow I$.) Then the expected "Beck-Chevalley" condition comes for free—namely that if

(2.1)



is a pullback square in **B** with f (and hence also k) in **A**, then the canonical natural transformation $k_! \circ h^* \longrightarrow g^* \circ f_!$ is an isomorphism. So the *Sum clauses* are covered by the assumption:

Sums. **A** *is closed under composition.*

Categorically, indexed products provide (stable) right adjoints to pullback functors. Thus as we argued above, we expect to have right adjoints

$$f_{\#} : \mathbf{A}(J) \longrightarrow \mathbf{A}(I)$$

to f^* for all morphisms $f : J \longrightarrow I$ which are in **A**. We will write

$$\Pi_f(B)$$

for the domain of the object of $\mathbf{A}(I)$ resulting from the application of $f_{\#}$ to an object $B \longrightarrow J$ of $\mathbf{A}(J)$. In contrast to the case for sums, we have to give an explicit condition for stability (which is needed to model the behaviour of products under substitution):

Beck-Chevalley condition. *If (2.1) is a pullback square in* **B** *with* f *(and hence also* k*) in* **A**, *then the canonical natural transformation* $g^* \circ f_{\#} \longrightarrow k_{\#} \circ h^*$ *is an isomorphism.*

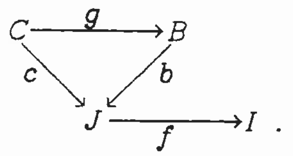
Thus the assumption needed to model indexed products is as follows:

Products. *For all* f *in* **A** , *we have a right adjoint* $f_{\#}$ *to* f^* *which satisfies the Beck-Chevalley condition for pullbacks in* **B** *(along arbitrary morphisms).*

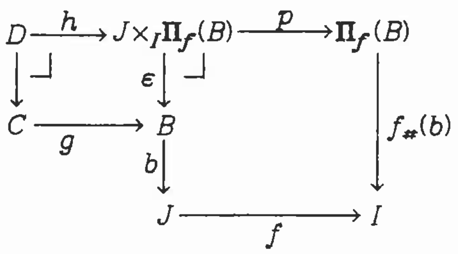
It is useful both conceptually and technically to have an equivalent formulation of this condition. First we give a preliminary result:

2.5. Lemma. *Suppose that* **A** *is a class of maps in a category* **B** *satisfying the above conditions of* **Stability**, **Unit**, **Sums** *and* **Products**. *Then the functor* $f_{\#}$ *preseves morphisms which are in* **A**.

Proof. Suppose we are given the following commutative diagram of morphisms in **A**:



We have to show that $\Pi_f(g) : \Pi_f(C) \longrightarrow \Pi_f(B)$ is in **A**. Form the pullback squares



where $\epsilon : f^{*}f_{*}(b) \longrightarrow b$ is the counit of the adjunction $f^{*} \dashv f_{*}$ at b. Note that p and h are both in **A** (because f and g are): hence we can form $p_{*}(h) : \Pi_{p}(D) \longrightarrow \Pi_{f}(B)$ in $\mathbf{A}(\Pi_{f}(B))$. We claim that $p_{*}(h)$ is isomorphic to $\Pi_{f}(g)$ in $\mathbf{B}/\Pi_{f}(B)$, which is sufficient to show that the latter is in **A**, bearing in mind our remark in 2.2 about the closure of **A** under composition with isomorphisms.

Whilst it is possible to prove this claim purely categorically, it is both easier and more illuminating to argue type-theoretically. So suppose that

	$f : J \longrightarrow I$	denotes the type	$J(i)\ [i \in I]$
	$b : B \longrightarrow J$	denotes the type	$B(i,j)\ [j \in J(i) > i \in I]$
and	$g : C \longrightarrow B$	denotes the type	$C(i,j,b)\ [b \in B(i,j) > j \in J(i) > i \in I]$.

Then $\quad f_{*}(b) : \Pi_{f}(B) \longrightarrow I \quad$ denotes $\quad \prod j \in J(i) \,.\, B(i,j)\ [i \in I]$,

and $\quad f_{*}(c) : \Pi_{f}(C) \longrightarrow I \quad$ denotes

(2.2) $\qquad \prod j \in J(i) \,.\, \sum b \in B(i,j) \,.\, C(i,j,b)\ [i \in I]$.

Moreover, $\Pi_{f}(g)$ is the morphism corresponding to

(2.3) $\qquad z \in \prod j \in J(i) \,.\, \sum b \in B(i,j) \,.\, C(i,j,b) \longmapsto \lambda j \in J(i) \,.\, \mathrm{Fst}(z(j)) \in \prod j \in J(i) \,.\, B(i,j)$.

By construction $\quad h : D \longrightarrow J \times_{I} \Pi_{f}(B) \quad$ denotes $\quad C(i,j',t(j'))\ [j' \in J(i)\,, t \in \prod j \in J(i) \,.\, B(i,j) > i \in I]$,

so that $\quad p_{*}(h) : \Pi_{p}(D) \longrightarrow \Pi_{f}(B) \quad$ denotes $\quad \prod j' \in J(i) \,.\, C(i,j',t(j'))\ [t \in \prod j \in J(i) \,.\, B(i,j) > i \in I]$

and hence the composition $f_{*}(b) \circ p_{*}(h)$ denotes

(2.4) $\qquad \sum t \in (\prod j \in J(i) \,.\, B(i,j)) \,.\, \prod j' \in J(i) \,.\, C(i,j',t(j'))\ [i \in I]$

But by Martin-Löf's form of the Axiom of Choice, the types in (2.2) and (2.4) are isomorphic over I; indeed the isomorphism is given in one direction by

(2.5) $\qquad z \in \prod j \in J(i) \,.\, \sum b \in B(i,j) \,.\, C(i,j,b) \longmapsto \langle \lambda j \in J(i) \,.\, \mathrm{Fst}(z(j))\,, \lambda j' \in J(i) \,.\, \mathrm{Snd}(z(j')) \rangle$

(*cf.* [M-L1], page 173) and in the other by

(2.6) $\qquad w \in \sum t \in (\prod j \in J(i) \,.\, B(i,j)) \,.\, \prod j' \in J(i) \,.\, C(i,j',t(j')) \longmapsto \lambda j \in J(i) \,.\, \langle (\mathrm{Fst}\,w) j\,, (\mathrm{Snd}\,w) j \rangle$.

Since the composition of (2.5) with the first projection is the map in (2.3), the interpretation of (2.5) and (2.6) give the required isomorphism between $\Pi_{f}(g)$ and $p_{*}(h)$ over $\Pi_{f}(B)$.

□

2.6. Proposition. *Suppose that* **A** *is a class of maps in* **B** *satisfying* **Stability**, **Unit** *and* **Sums**. *Then the condition* **Products** *is equivalent to the combination of the following* *conditions*:

(i) *each* $\mathbf{A}(I)$ *is cartesian closed*;

(ii) *for each* f *in* $\mathbf{B}$, f^{*} *preserves the cartesian closed structure;*

(iii) *for any* $a:A\longrightarrow I$ *in* $\mathbf{A}$, *the exponential functor* $(a\to_I -):\mathbf{A}(I)\longrightarrow\mathbf{A}(I)$ *preserves morphisms which are in* $\mathbf{A}$.

Proof. First note that if $\mathbf{A}$ satisfies the **Stability** and **Sums** conditions, then each $\mathbf{A}(I)$ has binary products given by pullbacks over I; and if $\mathbf{A}$ satisfies the **Unit** condition, then $id_I:I\longrightarrow I$ is a terminal object in $\mathbf{A}(I)$. If $\mathbf{A}$ also satisfies the **Products** condition, then we can calculate the exponential $(a\to_I b):(A\to_I B)\longrightarrow I$ of $a:A\longrightarrow I$ and $b:B\longrightarrow I$ in $\mathbf{A}(I)$ as $a_{*}(a^{*}(b)):\Pi_a(A\times_I B)\longrightarrow I$, so we have ($i$). Moreover, the Beck-Chevalley condition ensures that these exponentials are preserved under pullback, so we have (ii). Finally, since $\mathbf{A}$ satisfies the **Sums** condition, then by Lemma 2.5 $(a\to_I -)=a_{*}(a^{*}(-))$ preserves morphisms in $\mathbf{A}$ (since the pullback functor a^{*} always does).

Conversely, suppose that $\mathbf{A}$ satisfies **Stability**, **Unit**, **Sums** and conditions (i) to (iii). Given $f:J\longrightarrow I$ and $b:B\longrightarrow J$ in $\mathbf{A}$, then by the **Sums** condition fb is also in $\mathbf{A}$ and we can regard b as a morphism $fb\longrightarrow f$ in $\mathbf{A}(I)$. Then by (iii), $(f\to_I b):(f\to_I fb)\longrightarrow(f\to_I f)$ is given by a morphism which is also in $\mathbf{A}$. Hence we can form the following pullback square in $\mathbf{B}$:

$$\begin{array}{ccccc}
\Pi_f(B) & \longrightarrow & (J\to_I B) & & \\
\big\downarrow & \lrcorner & \big\downarrow{\scriptstyle (J\to_I b)} & \searrow{\scriptstyle (f\to_I fb)} & \\
I & \xrightarrow[\ulcorner\pi_2\urcorner]{} & (J\to_I J) & \xrightarrow[(f\to_I f)]{} & I
\end{array}$$

where $\ulcorner\pi_2\urcorner:id_I\longrightarrow(f\to_I f)$ in $\mathbf{A}(I)$ is the exponential transpose of the isomorphism $\pi_2:id_I\times_I f\cong f$. Thus the morphism $\Pi_f(B)\longrightarrow I$ is in $\mathbf{A}$ and a simple calculation shows that it has the correct universal property to be $f_{*}(b)$. The fact that these right adjoints to pulling back satisfy the Beck-Chevalley condition follows from this recipe for their construction together with condition (ii).

□

When $\mathbf{A}$ consists of all the morphisms in $\mathbf{B}$, then the above proposition reduces to the equivalence of two well known characterizations of locally cartesian closed categories. This justifies the following terminology, which we have borrowed from [Ta]:

2.7. Definition. A *relatively cartesian closed category* (or *rccc*, for short) is a category $\mathbf{B}$ with finite products equipped with a distinguished class of morphisms (called the *display morphisms* of $\mathbf{B}$) satisfying the conditions **Stability** of 2.2 and **Unit**, **Sums** and **Products** of 2.4. A *morphism* of rccc's is a finite product preserving functor which sends display morphisms to display morphisms, preserves pullbacks of display morphisms along arbitrary morphisms and preserves the right adjoints to pulling back display morphisms along display morphisms. We will let **RCCC** denote the 2-category whose objects are rccc's, whose morphisms are rccc morphisms and whose 2-cells are natural transformations.

The situation for rccc's is almost analogous to that in [Se1], where an equivalence is established between locally cartesian closed categories and theories (in the sense of 1.10) over Martin-Löf type theory with unit, products and strong rules for sums and equality types. Here we are considering theories without equality types. A *model* of such a theory **T** in an rccc **B** is given by an assignment of display morphisms to the constant Types and morphisms to the constant Terms in such a way that the judgements which comprise the axioms of the theory are all satisfied in the rccc; this is a sound notion since one can show that any judgement derivable from the axioms using the rules is also satisfied. Defining an appropriate notion of homomorphism of models, we get a category **Mod(T,B)** of models of **T** in **B**. If $F: \mathbf{B} \longrightarrow \mathbf{B}'$ is an rccc morphism, then it sends T-models in **B** to T-models in **B'** and gives a functor $F_*: \mathbf{Mod(T,B)} \longrightarrow \mathbf{Mod(T,B')}$; similarly each natural transformation $\phi: F \longrightarrow F'$ between rccc morphisms induces a natural transformation $\phi_*: F_* \longrightarrow F'_*$. In this way we get a 2-functor **Mod(T, −)** from **RCCC** into the 2-category of categories, **CAT**. The fundamental observation is that, in an appropriately bicategorical sense, this 2-functor is representable:

Theorem. *For each theory* T *there is an rccc* **B(T)**, *called the* classifying category *of* **T**, *and a model* M_{T} *of* T *in* **B(T)**, *called the* generic model *of* **T**, *with the property that for any rccc* **B** *the functor*

$$(-)_*(M_{\mathrm{T}}): \mathbf{RCCC(B(T),B)} \longrightarrow \mathbf{Mod(T,B)}$$

is an equivalence of categories.

□

The construction of **B(T)** is simplified by the presence of the rules for dependent sums, since as we remarked in 2.3(*iii*), it follows that we can take every morphism to be represented by a term. So we can take the objects of **B(T)** to be the closed types in **T** and the morphisms to be eqivalence classes of closed terms, under the equivalence relation of provable equality in **T**; the display morphisms are those which are isomorphic to first projections, $\mathrm{Fst}: \Sigma a \in A.B(a) \longrightarrow A$.

Classifying categories enable us to give a very general notion of *interpretation* of one theory in another—by defining an interpretation of **T** in **T'** to be a model of **T** in **B(T')**; similarly we can give a notion of *modification* between interpretations in terms of homomorphisms between models in the classifying category. This gives the collection of theories the structure of a bicategory (*cf.* [B3]), **MLTT**, in such a way that $\mathbf{B}(-): \mathbf{MLTT} \longrightarrow \mathbf{RCCC}$ is a fully faithful homomorphism of bicategories. The essential image of this bicategory homomorphism consists of those rccc's which are equivalent to the classifying category of some theory: this does not include every rccc for the following trivial reason. Suppose that **B** is an rccc and **A** is the class of display morphisms in **B**. If for each $n \in \mathbb{N}$ we introduce symbols for n-ary constant Types for each morphism $A \longrightarrow I_1 \times \cdots \times I_n$ in **A** and symbols for n-ary constant Terms for each morphism $I_1 \times \cdots \times I_n \longrightarrow J$ in **B**, then there is an evident theory $\mathrm{T}_{\mathbf{B}}$ whose derived judgements are just those satisfied in the rccc. The canonical model of this theory in **B** induces a full (because of Remark 2.3(*iii*)) and faithful rccc morphism $\mathbf{B}(\mathrm{T}_{\mathbf{B}}) \longrightarrow \mathbf{B}$. Its image consists of those objects of $\mathbf{B}$ which denote closed types: these are the objects B for which the unique morphism $B \longrightarrow 1$ is in $\mathbf{A}$. We therefore impose a further condition on $\mathbf{A}$, namely:

Display. *For each object B of $\mathbf{B}$, the unique morphism $B \longrightarrow 1$ is in $\mathbf{A}$.*

For the rccc's satisfying this added condition we have that $\mathbf{B}(\mathrm{T}_{\mathbf{B}}) \simeq \mathbf{B}$ and thus have:

Corollary. *The classifying category construction $\mathbf{B}(-)$ induces an equivalence of bicategories between* **MLTT** *and the 2-category of rccc's which satisfy the* **Display** *condition.*

□

Remark. The condition **Display** in the presence of **Stability**, renders redundant some of our other assumptions about $\mathbf{B}$ and $\mathbf{A}$. For one thing it implies that $\mathbf{B}$ has binary products, since these are given by pullbacks over the terminal object. It also says in particular that the unique morphism $1 \longrightarrow 1$ is in $\mathbf{A}$: but that morphism is necessarily the identity on 1 and so by **Stability**, all isomorphisms are in $\mathbf{A}$—in other words the **Unit** condition holds automatically.

2.8. Absoluteness of indexed products. In our discussion of indexed sums we justified the assumption that these sums should agree with the corresponding sums in $\mathbf{B}^2 \longrightarrow \mathbf{B}$ (i.e. be given by composition in $\mathbf{B}$). We will shortly consider a case (in 2.12) where this assumption is not justified. It is instructive to see why the corresponding question does not arise for indexed products.

Suppose that $\mathbf{B}$ has finite products and that

$$\begin{array}{ccc} \mathbf{A}_1 & \hookrightarrow & \mathbf{A}_2 \\ & \searrow \quad \swarrow & \\ & \mathbf{B} & \end{array}$$

is a full embedding of full subcategories of $\mathbf{B}^2$ over $\mathbf{B}$ which contain the terminal object (so that $\mathbf{A}_1$ and $\mathbf{A}_2$ as classes of morphisms of $\mathbf{B}$ contain the isomorphisms). Suppose also that $\mathbf{C}$ is a class of morphisms in $\mathbf{B}$ satisfying the **Stability** condition of 2.2. (Thus, although we do not wish to look at it this way, $\mathbf{C} \longrightarrow \mathbf{B}$ is also a full subcategory of $\mathbf{B}^2$ over $\mathbf{B}$.) Then we have:

Proposition. *In the above situation, if $\mathbf{A}_1$ and $\mathbf{A}_2$ both have indexed products along maps in $\mathbf{C}$ satisfying the Beck-Chevalley condition for pullbacks along arbitrary morphisms in $\mathbf{B}$, then these indexed products agree (up to isomorphism) on $\mathbf{A}_1$.*

Proof. For $f: J \longrightarrow I$ in $\mathbf{C}$, let ${}^1f_*(a) : {}^1\Pi_f(A) \longrightarrow I$ and ${}^2f_*(a) : {}^2\Pi_f(A) \longrightarrow I$ respectively denote the value at $a : A \longrightarrow J$ of the right adjoints to the pullback functors $f^* : \mathbf{A}_1(I) \longrightarrow \mathbf{A}_1(J)$ and $f^* : \mathbf{A}_2(I) \longrightarrow \mathbf{A}_2(J)$.

Suppose that $y : Y \longrightarrow J$ is in $\mathbf{A}_1(J)$. Then we have a natural comparison morphism

$$\kappa : {}^1\Pi_f(Y) \longrightarrow {}^2\Pi_f(Y)$$

such that for all $z : Z \longrightarrow I$ in $\mathbf{A}_1(I)$, composition with κ induces a bijection

$$\mathbf{B}/I(z,\ {}^1f_\#(y)) \cong \mathbf{B}/I(z,{}^2f_\#(y)).$$

Let us write $h: {}^2\Pi_f(Y) \longrightarrow I$ for ${}^2f_\#(y)$ and form the pullback square:

$$\begin{array}{ccc} L & \xrightarrow{\ q\ } & {}^2\Pi_f(Y) \\ {\scriptstyle p}\downarrow & \lrcorner & \downarrow{\scriptstyle h} \\ J & \xrightarrow[\ f\ ]{} & I \end{array} .$$

We deduce from the Beck-Chevalley condition that $h^\#(\kappa)$ is (isomorphic to) the comparison morphism

$${}^1\Pi_q(p^\# Y) \longrightarrow {}^2\Pi_q(p^\# Y) \ .$$

Hence for all $w: W \longrightarrow {}^2\Pi_f(Y)$ in $\mathbf{A}_1({}^2\Pi_f(Y))$, composition with $h^\#(\kappa)$ induces a bijection

$$\mathbf{B}/{}^2\Pi_f(Y)(w, h^\#({}^1f_\# y)) \cong \mathbf{B}/{}^2\Pi_f(Y)(w, h^\#({}^2f_\# y)),$$

that is, composition with κ induces a bijection

$$\mathbf{B}/I(hw,\ {}^1f_\#(y)) \cong \mathbf{B}/I(hw, {}^2f_\#(y)) \ .$$

Taking w to be the identity on ${}^2\Pi_f(Y)$ (which is in $\mathbf{A}_1$ by hypothesis), we deduce that there is a morphism

$$\lambda : {}^2\Pi_f(Y) \longrightarrow {}^1\Pi_f(Y)$$

such that $\kappa \circ \lambda = id$. Now pull the situation back along $k = {}^1f_\#(y) : {}^1\Pi_f(Y) \longrightarrow I$. We find similarly that for $v: V \longrightarrow {}^1\Pi_f(Y)$ in $\mathbf{A}_1({}^1\Pi_f(Y))$, composition with κ induces a bijection

$$\mathbf{B}/I(kv, {}^1f_\#(y)) \cong \mathbf{B}/I(kv, {}^2f_\#(y)) \ .$$

Taking v to be the identity, we find that both id and $\lambda \circ \kappa$ correspond to κ under this bijection, whence $\lambda \circ \kappa = id$.

□

2.9. Remarks.

(*i*) The result we have just proved is an "absoluteness" result for indexed products. The situation should be contrasted with that for indexed sums where no such absoluteness holds. (We will see an example of this in section 5, where the left adjoints (5.9) to pulling back localic-algebraic toposes differ from the left adjoints (4.9) to pulling back algebraic toposes.)

(*ii*) Constructions of a similar (right adjoint) kind—such as finite limits and stable exponentials—are also absolute by essentially the same argument.

2.10. Sums and products of Orders and Types indexed over Types. Let us start by assuming that we are modelling Orders and Operators by a relatively cartesian closed category $\mathbf{B}$ whose class of display morphisms is denoted by $\mathbf{A}$ and satisfies the **Display** condition (see 2.7 *et seq.*).

As we remarked in 1.7, the rules giving the closure of Orders under sums indexed over Types ensure that there is a bijective correspondence between Terms of Type A and Operators of Order $A\times 1_O$. It follows that we may take Types to be special Orders and model them by a distinguished class of morphisms **R** contained in **A** and also satisfying the **Stability** condition (so that we have a cartesian embedding $\mathbf{R}\hookrightarrow\mathbf{A}$ of fibrations over **B**).

We note at once that given this set up, the closure of Orders under sums and products indexed over Types becomes a special case of the closure under sums and products indexed over Orders—which we already have. In order to satisfy the *Unit, Sum* and *Product clauses* for "quantification" of Types over Types (which are the same as those for quantification of Orders over Orders) we apply the analysis of 2.3 to the class **R** itself. Combining this with the absoluteness of products proved in 2.4, we require the following conditions:

Unit'. **R** *contains all isomorphisms.*

Sums'. **R** *is closed under composition.*

Products'. *For all* $f:J\longrightarrow I$ *in* **R**, *the right adjoint* $f_*:\mathbf{A}(J)\longrightarrow\mathbf{A}(I)$ *to* f^* *restricts to a functor* $f_*:\mathbf{R}(J)\longrightarrow\mathbf{R}(I)$.

2.11. Type as an Order. To explain the special role of the Order *Type*, we need to explain the connection between A as it appears in the judgement

$$A\in Type$$

(A as an Operator of Order *Type*) and A as it appears in the judgement

$$a\in A$$

(A as a Type). To do this consider the generic case of the free Type variable X. In the usual way "$X\in Type$" will be interpreted by the identity function $id:[\![Type]\!]\longrightarrow[\![Type]\!]$. On the other hand, to interpret "$s\in X$", we must regard X as a Type (indexed over the Order *Type*) and interpret s as a morphism with that codomain.

This leads to the following situation. We have an object U of **B** which is to interpret *Type* (and so must satisfy that $U\longrightarrow 1$ is in **A**—which is automatic, since we are assuming **A** satisfies the **Display** condition); and we have a morphism $G\longrightarrow U$ in **R** interpreting the Type X indexed over *Type*. Now for any context Γ

$$Type\ [\Gamma]$$

is interpreted by the projection $U\times[\![\Gamma]\!]\longrightarrow[\![\Gamma]\!]$ and hence

$$A\in Type\ [\Gamma]$$

is interpreted by a section $[\![\Gamma]\!]\longrightarrow U\times[\![\Gamma]\!]$ of this projection—and this is equivalent to giving a morphism

$$\ulcorner A\urcorner:[\![\Gamma]\!]\longrightarrow U.$$

Then to interpret A in

$$a \in A \; [\Gamma] \; ,$$

that is, as an object $[\![A]\!] \longrightarrow [\![\Gamma]\!]$ of the category $\mathbf{R}([\![\Gamma]\!])$, we form a pullback square:

$$\begin{array}{ccc} [\![A]\!] & \longrightarrow & G \\ \downarrow & \lrcorner & \downarrow \\ [\![\Gamma]\!] & \xrightarrow[\ulcorner A \urcorner]{} & U \end{array} \; .$$

In particular the free type variable X is interpreted either as the identity on U, or as the object $G \longrightarrow U$ of $\mathbf{R}(U)$. Since every Type is obtainable from X by substitution and the latter is modelled by pullbacks, we can see that taking $Type$ as an Order amounts to the following assumption on our distinguished class $\mathbf{R}$ of morphisms in $\mathbf{B}$:

> **Generic type.** *There is a morphism $G \longrightarrow U$ in $\mathbf{R}$ which generates the class under pullback: every morphism in $\mathbf{R}$ is obtainable from $G \longrightarrow U$ by pullback along some (not necessarily unique) morphism in $\mathbf{B}$.*

2.12. Sums and products of Types indexed over Orders. We now consider what properties of

$$\begin{array}{ccccc} \mathbf{R} & \hookrightarrow & \mathbf{A} & \hookrightarrow & \mathbf{B}^2 \\ & \searrow & \downarrow & \swarrow & \\ & & \mathbf{B} & & \end{array}$$

are needed to model the "quantification" of Types over Orders. The main problem is to understand the relevant rules in 1.9 for sums. Of course we still expect indexed sums to provide a left adjoint to substitution. And as in 2.2 we can extract a morphism from the unit of the adjunction: for if we have

$$A(X,Y) \in Type \; [Y \in L(X) > X \in K]$$

then we have $x \in A(X,Y) \longmapsto (Y,x) \in \sum Y \in L(X) \,.\, A(X,Y)$, whose interpretation makes the following diagram commute:

$$\begin{array}{ccc} [\![A(X,Y)]\!] & \longrightarrow & [\![\sum Y \in L(X) \,.\, A(X,Y)]\!] \\ \downarrow & & \downarrow \\ [\![L(X)]\!] & \longrightarrow & [\![K]\!] \end{array} \; .$$

The morphism $[\![A(X,Y)]\!] \longrightarrow [\![L(X)]\!]$ is in $\mathbf{R}$ and the map $[\![L(X)]\!] \longrightarrow [\![K]\!]$ is in $\mathbf{A}$: so their composition $[\![A(X,Y)]\!] \longrightarrow [\![K]\!]$ is in $\mathbf{A}$. It follows that the left adjoint is providing a best possible factorization of morphisms in $\mathbf{A}$ through morphisms in $\mathbf{R}$.

We next have to deal explicitly with substitution: the Beck-Chevalley condition does not come for free anymore (since the left adjoints are not given simply by composition). But it is a standard result that this condition amounts to requiring that the factorization be stable under pullback.

It is tempting to think that that is all, but it is not. What we have suceeded in modelling so far are the *Sum clauses* for Types over Orders from 1.9 with the following restrictions: in the *elimination* and *equality clauses* the B that appears *does not depend on* $z\in\sum X\in K.A$. These are essentially the rules given in [MiPl], which are equivalent to the sum rules in Girard's original version of higher order typed lambda calculus—the system F_ω of [G1]. (We say "essentially" because neither Mitchell and Plotkin, nor Girard give a "second" equality rule.) What then is the added nuance in the Martin-Löf style rule we give in 1.9? It amounts to a condition on the *other* morphism in the factorization. If the morphism $f:L\longrightarrow K$ in **A** has best factorization

$$L\xrightarrow{\ s\ }\textstyle\sum X\in L.1_T\xrightarrow{\ r\ }K$$

with r in **R**, then s is *orthogonal* to **R** in the familiar categorical sense: that is, for any $g:B\longrightarrow A$ in **R** and commutative square

$$\begin{array}{ccc} L & \xrightarrow{\ s\ } & \sum X\in L.1_T \\ \downarrow & & \downarrow \\ B & \xrightarrow[\ g\ ]{} & A \end{array}$$

in **B**, there is a unique morphism $\sum X\in L.1_T\longrightarrow B$ making the diagram

$$\begin{array}{ccc} L & \xrightarrow{\ s\ } & \sum X\in L.1_T \\ \downarrow & \swarrow & \downarrow \\ B & \xrightarrow[\ g\ ]{} & A \end{array}$$

commute. It is easy to see that *any* factorization of f as a morphism orthogonal to **R** followed by a morphism in **R** is necessarily the *universal* factorization of f through a morphism in **R**. Thus the categorical version of the rule for sums of Types over Orders is as follows:

> **Big Sums.** *Any morphism* f *in* **A** *factors as* $f=r\circ s$, *where* r *is in* **R** *and* s *is orthogonal to* **R**. *Moreover this factorization is stable under pullback along arbitrary morphisms in* **B**.

The orthogonality condition in **Big Sums** remains slightly mysterious. The rest of the condition can be understood as providing a reflection of the fibration $\mathbf{A}\longrightarrow\mathbf{B}$ into the firbration $\mathbf{R}\longrightarrow\mathbf{B}$: that is, a cartesian left adjoint to the inclusion $\mathbf{R}\hookrightarrow\mathbf{A}$. In the models of sections 4 and 5 we get the orthogonality condition for free, since for these there is a finite limit preserving inclusion from **B** into a category **G** having all pullbacks (so that $cod:\mathbf{G}^2\longrightarrow\mathbf{G}$ is a fibration) and the reflection of **A** into **R** is the restriction to **A** of a reflection from $\mathbf{G}^2\longrightarrow\mathbf{G}$ into some full subfibration—for which situation orthogonality is automatic.

Turning finally to the case of products of Types indexed over Orders, in view of Proposition 2.8 it is comparatively easy to explain what is needed to interpret the relevant clauses in 1.9. We already have an interpretation of indexed products of Orders over

Orders and since we are interpreting Types as special kinds of Order, the case of products of Types over Orders is necessarily a restriction of this:

> **Big Products.** *For all* $f:J\longrightarrow I$ *in* **A**, *the right adjoint* $f_*:\mathbf{A}(J)\longrightarrow\mathbf{A}(I)$ *to* f^* *restricts to a functor* $f_*:\mathbf{R}(J)\longrightarrow\mathbf{R}(I)$.

Note that this condition directly entails the condition **Products'** of 2.10.

2.13. Summary. We have now dealt with the categorical interpretation of all parts of the syntax of the basic theory of constructions presented in section 1. Let us summarize what we require (at the same time eliminating some of the redundancies we have noted):

(*i*) A category **B** with a terminal object.

(*ii*) Distinguished collections **A** and **R** of morphisms in **B**, with **R** contained in **A**.

(*iii*) The pullback along an arbitrary morphism in **B** of a morphism in **A** (respectively **R**) exists and is again in **A** (respectively **R**).

(*iv*) **R** contains all the isomorphisms in **B**.

(*v*) **R** and **A** are both closed under composition.

(*vi*) For each $f:J\longrightarrow I$ in **A**, the pullback functor $f^*:\mathbf{A}(I)\longrightarrow\mathbf{A}(J)$ has a right adjoint, $f_*:\mathbf{A}(J)\longrightarrow\mathbf{A}(I)$. Moreover, these right adjoints satisfy the Beck-Chevalley condition for pullbacks of f along arbitrary morphisms in **B**; and they send morphisms in $\mathbf{R}(J)$ to morphisms in $\mathbf{R}(I)$.

(*vii*) Every morphism in **A** factors as the composition of a morphism orthogonal to the class **R** followed by a morphism in **R**. Moreover, this factorization is preserved under pullback along arbitrary morphisms in **B**.

(*viii*) There is a morphism $G\longrightarrow U$ in **R** from which all other morphisms in **R** can be obtained by pullback.

(*ix*) For each object B in **B**, the unique morphism from B to the terminal object is in **A**.

The equivalence between **MLTT** and **RCCC** mentioned above extends to one between theories over the basic theory of constructions (as in 1.10) and an appropriate 2-category of categorical structures satisfying (*i*) to (*ix*). It is worth pointing out that the only *structure* required to specify a model is essentially only that of a category **B** and two distinguished classes of morphisms **A** and **R** in **B**—all the other requirements amount to categorical properties of this structure. Thus if (**B**,**A**,**R**) is a model and $I:\mathbf{B}\simeq\mathbf{B}'$ is an equivalence of categories, then (**B'**,**A'**,**R'**) is also a model if we define **A'** to consist of morphisms isomorphic (in $\mathbf{B}^2$) to ones in the image of **A** under I and similarly for **R'**.

We conclude this section by considering the interpretation of the two extensions of the basic theory of constructions given in 1.11 and 1.12.

2.14. Models with "*Order* ∈ *ORDER*". The categorical explanation of the theory "*Order* ∈ *ORDER*" is similar to that given in 2.11 for the modelling of "*Type*" in the basic theory. We have to have an object V in **B** to interpret *Order* and a morphism $h:H\longrightarrow V$ in **A** to interpret $O(X)$ $[X\in Order]$. Then for any $k:K\longrightarrow I$ in **A** (interpreting

$K(\xi)$ $[\xi \in I]$ say) there is a morphism $\ulcorner K \urcorner : I \longrightarrow V$ (interpreting $T_K(\xi)$) so that $(\ulcorner K \urcorner)^{\#}(h) \cong k$ in $\mathbf{A}(I)$ (the isomorphism interpreting $I_K(\xi)$ and its inverse interpreting $J_K(\xi)$). The condition on $\mathbf{A}$ is therefore:

> ***Order* ∈ *ORDER*.** *There is a morphism* $H \longrightarrow V$ *in* $\mathbf{A}$ *from which all other morphisms in* $\mathbf{A}$ *can be obtained by pullback.*

2.15. Models with "*Type* ≃ *ORDER*". Turning to the theory "*Type* ≃ *ORDER*", our analysis of the interpretation of sum rules in this section shows that if we strengthen the rules for sums of Types indexed over Orders as indicated in 1.12, then in the categorical models we must have:

> when $f : J \longrightarrow I$ is in $\mathbf{A}$, the left adjoint $f_! : \mathbf{R}(J) \longrightarrow \mathbf{R}(I)$ to $f^{\#}$ (whose existence is guarenteed by 2.13(vii)) is given by composition with f; in other words, $f \circ r$ is in $\mathbf{R}$ whenever f is in $\mathbf{A}$ and r is in $\mathbf{R}$.

But taking $r = id$, we get that $\mathbf{A}$ is contained in $\mathbf{R}$ and therefore that *$\mathbf{A}$ and $\mathbf{R}$ are equal*. This assumption renders some of the assumptions in 2.13 redundant, and we arrive at the following requirements for a categorical model of the theory of constructions with "*Type* ≃ *ORDER*":

(i) A category $\mathbf{B}$ with a terminal object.
(ii) A distinguished collection $\mathbf{A}$ of morphisms in $\mathbf{B}$.
(iii) The pullback along an arbitrary morphism in $\mathbf{B}$ of a morphism in $\mathbf{A}$ exists and is again in $\mathbf{A}$.
(iv) $\mathbf{A}$ is closed under composition.
(v) For each $f : J \longrightarrow I$ in $\mathbf{A}$, the pullback functor $f^{\#} : \mathbf{A}(I) \longrightarrow \mathbf{A}(J)$ has a right adjoint, $f_{\#} : \mathbf{A}(J) \longrightarrow \mathbf{A}(I)$. Moreover, these right adjoints satisfy the Beck-Chevalley condition for pullbacks of f along arbitrary morphisms in $\mathbf{B}$.
(vi) There is a morphism $G \longrightarrow U$ in $\mathbf{A}$ from which all other morphisms in $\mathbf{A}$ can be obtained by pullback.
(vii) For each object B in $\mathbf{B}$, the unique morphism from B to the terminal object is in $\mathbf{A}$.

3 Lim theories

In this section we will review those parts of (more traditional) categorical logic which we will need in the next two sections to present our topos-theoretic models of the theory of constructions. In specifying these models along the general lines indicated in the previous section, we will use several constructions on *categories with finite limits*—or *lex categories* as we will call them. In order to see that these constructions can be carried out, we need a way of presenting lex categories in terms of "generators and relations". There are a number of ways in which this can be done. For example, one could use *finite projective sketches*: see [BW, 4.4]. Alternatively one could use theories over a fragment of Martin-Löf type theory, such as the *generalized algebraic theories* of [Ca] (see also [Po]). Instead we choose to use the *lim theories* of M.Coste, since their syntax and semantics are part of the familiar formalism of the first order predicate calculus. The following account stresses model-theoretic aspects; for a fuller picture, see the original work of M.Coste [Co1, Co2] on the subject.

For the purposes of this section, a *language* **L** is specified by:

- A collection of *sort symbols*, $S, S', S'', \ldots$
- A collection of *function symbols* of specified *types*. The type of such a function symbol f, is given by a non-empty list of sort symbols $S_1,\ldots,S_n,S$, and this will be indicated by writing $f: S_1\times\cdots\times S_n \longrightarrow S$. (This includes the case $n=0$, when f is more usually called a *constant symbol* of type S.)
- A collection of *relation symbols* of specified *types*. The type of such a relation symbol R, is given by a (possibly empty) list of sort symbols $S_1,\ldots,S_n$, and this will be indicated by writing $R \rightarrowtail S_1\times\cdots\times S_n$.

Starting with countably infinite sets of *variables* for each sort symbol, the *terms* of **L** and their associated *types* are defined recursively in the usual way:

- Each variable x of type S is a term of type S.
- If $t_1,\ldots,t_n$ are terms of type $S_1,\ldots,S_n$ and $f: S_1\times\cdots\times S_n \longrightarrow S$ is a function symbol, then $f(t_1,\ldots,t_n)$ is a term of type S.

(We write $t{:}S$ to indicate that a term t has type S.) We next define the *basic formulas* over **L** recursively as follows:

- If t,t' are terms of the same type, then $t=t'$ is a basic formula.
- If $t_1,\ldots,t_n$ are terms of type $S_1,\ldots,S_n$ and $R \rightarrowtail S_1\times\cdots\times S_n$ is a relation symbol, then $R(t_1,\ldots,t_n)$ is a basic formula.
- The symbol $\top$ ("true") is a basic formula.
- If ϕ and ψ are basic formulas, then so is $(\phi\wedge\psi)$.

Basic formulas of the first two kinds are usually called *atomic* formulas. Thus a basic formula is a finite (possibly empty) conjunction of atomic formulas.

We now define a *lim theory* **T** to be given by a language **L** and a set of *lim sentences* (the *axioms* of **T**): such a lim sentence is specified formally by two disjoint lists $\overline{x},\overline{y}$ of distinct variables and two basic formulas ϕ,ψ, the variables involved in the first all appearing in the list $\overline{x}$ and the variables involved in the second appearing either in $\overline{x}$ or in $\overline{y}$. The lim sentence will be written as

$$(3.1) \qquad \forall\overline{x}\,(\phi(\overline{x})\rightarrow\exists!\,\overline{y}\,\psi(\overline{x},\overline{y}))$$

to indicate that its intended meaning is: "for all $\overline{x}$ such that $\phi(\overline{x})$ holds, there exist *unique* $\overline{y}$ so that $\psi(\overline{x},\overline{y})$ holds". In the case that ψ does not depend on y, then (3.1) will be abbreviated to

$$\forall\overline{x}\,(\phi(\overline{x})\rightarrow\psi(\overline{x}))$$

and if furthermore ϕ is $\top$, then *(3.1)* will be written as

$$\forall \overline{x}\, \psi(\overline{x}).$$

Clearly every (many-sorted) algebraic theory can be regarded as a lim theory. Here are two simple examples of non-algebraic lim theories:

3.1. Example: the lim theory **cat** of *categories*.

The underlying langage of **cat** has two sorts, *Ob* and *Mor*, three function symbols

$$dom : Mor \longrightarrow Ob$$
$$cod : Mor \longrightarrow Ob$$
$$id : Ob \longrightarrow Mor,$$

and one relation symbol

$$comp \rightarrowtail Mor \times Mor \times Mor,$$

whose intended meaning is the graph of the composition partial function on morphisms in a category. Thus the axioms of **cat** are:

- $\forall x{:}Ob(\, dom(id(x))=x \wedge cod(id(x))=x\,)$
- $\forall f,g,h{:}Mor\big(\, comp(f,g,h) \rightarrow (\, dom(f)=dom(h) \wedge cod(f)=dom(g) \wedge cod(g)=cod(h)\,)\,\big)$
- $\forall f,g{:}Mor(\, cod(f)=dom(g) \rightarrow \exists! h{:}Mor\ comp(f,g,h)\,)$
- $\forall f,g,h,k,l,m,n{:}Mor(\, comp(f,g,k) \wedge comp(g,h,l) \wedge comp(k,h,m) \wedge comp(f,l,n) \rightarrow m=n\,)$
- $\forall f{:}Mor(\, comp(f,id(cod(f)),f) \wedge comp(id(dom(f)),f,f)\,)$.

3.2. Example: the lim theory **lex** of *categories with finite limits*.

We axiomatize the property of having finite limits via those of having a terminal object and pullbacks. Thus **lex** is obtained from **cat** by adding a constant symbol $\top$ of type *Ob*, a new relation symbol $pb \rightarrowtail Mor \times Mor \times Mor \times Mor$ and new axioms:

- $\forall x{:}Ob\, \exists! f{:}Mor\,(\, dom(f)=x \wedge cod(f)=\top\,)$
- $\forall f,g,h,k{:}Mor\big(\, pb(f,g,h,k) \rightarrow \exists! l{:}Mor(comp(f,g,l) \wedge comp(h,k,l))\,\big)$
- $\forall g,k{:}Mor(cod(g)=cod(k) \rightarrow \exists! f,h{:}Mor\ pb(f,g,h,k)\,)$
- $\forall f,f',g,h,h',k,l{:}Mor\big(pb(f,g,h,k) \wedge comp(f',g,l) \wedge comp(h',k,l) \rightarrow$
  $\exists! m{:}Mor(comp(m,f,f') \wedge comp(m,h,h'))\,\big)$.

3.3. Structures and satisfaction. Given a set-valued *structure* for a language $\mathbf{L}$ (i.e. an assignment of sets for the sorts, functions for the function symbols and relations for the relation symbols, all satisfying the evident typing requirements), the intended meaning of the lim sentence *(3.1)* mentioned above amounts to an informal definition of the notion of *satisfaction* of a lim sentence in a structure. In giving the formal definitions of "structure for a language" and "satisfaction of a lim sentence by a structure" one only needs to use set theoretic operations of a very simple kind: in fact formation of finite limits in the category of sets and functions is all that is needed. Accordingly these definitions can be given for an arbitrary category with finite limits and as such, are a fragment of the categorical interpretation of first order logic given by Makkai and Reyes in [MR, Chapter 2, section 3]. We recall from there the following definitions:

If **L** is a language and **C** is a category with finite limits, then an **L**-*structure* M in **C** assigns

- to each sort symbol S, an object $M(S)$ in **C**,
- to each function symbol $f{:}S_1\times\cdots S_n\longrightarrow S$, a morphism $M(f){:}M(S_1)\times\cdots\times M(S_n)\longrightarrow M(S)$ in **C**,
- to each relation symbol $R\rightarrowtail S_1\times\cdots\times S_n$, a subobject $M(R)\rightarrowtail M(S_1)\times\cdots\times M(S_n)$ in **C**.

If $\overline{x}=x_1,\ldots,x_n$ is a finite list of distinct variables with x_i of type S_i say, and if t is a **L**-term whose variables all occur in $\overline{x}$, then a morphism $[\![t(\overline{x})]\!]{:}M(S_1)\times\cdots\times M(S_n)\longrightarrow M(S)$ in **C** is defined by structural induction, as follows:

- If t is x_i, then $[\![t(\overline{x})]\!]=\pi_i$, the i^{th} product projection.
- If t is $f(t_1,\ldots,t_m)$, then $[\![t(\overline{x})]\!]=M(f)\circ\langle[\![t_1(\overline{x})]\!],\ldots,[\![t_m(\overline{x})]\!]\rangle$ (where $\langle[\![t_1(\overline{x})]\!],\ldots,[\![t_m(\overline{x})]\!]\rangle$ is the unique morphism whose composition with each π_j is $[\![t_j(\overline{x})]\!]$).

Similarly, if ϕ is a basic **L**-formula involving variables from the list $\overline{x}$, then a subobject $[\![\phi(\overline{x})]\!]\rightarrowtail M(S_1)\times\cdots\times M(S_n)$ in **C** is defined by structural induction, as follows:

- If ϕ is $t{=}t'$, then $[\![\phi(\overline{x})]\!]$ is the equalizer of the pair of morphisms $[\![t(\overline{x})]\!],[\![t'(\overline{x})]\!]$.
- If ϕ is $R(t_1,\ldots,t_m)$, then $[\![\phi(\overline{x})]\!]$ is the pullback of the subobject $M(R)$ along the morphism $\langle[\![t_1(\overline{x})]\!],\ldots,[\![t_m(\overline{x})]\!]\rangle$.
- If ϕ is $\top$, then $[\![\phi(\overline{x})]\!]$ is the *greatest* subobject of $M(S_1)\times\cdots\times M(S_n)$, i.e. that given by the identity morphism for $M(S_1)\times\cdots\times M(S_n)$.
- If ϕ is $\psi\wedge\theta$, then $[\![\phi(\overline{x})]\!]$ is the *meet* of the subobjects $[\![\psi(x)]\!]$ and $[\![\theta(x)]\!]$, i.e. is given by forming a pullback from monomorphisms representing these subobjects.

Now given a lim sentence as in (*3.1*), define the objects X,Y of **C** to be $M(S_1)\times\cdots\times M(S_n)$ and $M(S_1')\times\cdots\times M(S_m')$ repectively (where S_i is the type of x_i and S_j' the type of y_j); suppose also that the subobject $[\![\psi(\overline{x},\overline{y})]\!]$ is represented by the monomorphism $\langle a,b\rangle{:}[\![\psi(\overline{x},\overline{y})]\!]\rightarrowtail X\times Y$. Then we say that the **L**-structure M *satisfies* the sentence (*3.1*), and write

$$M\vDash\forall\overline{x}(\phi(\overline{x})\rightarrow\exists!\overline{y}\,\psi(\overline{x},\overline{y})),$$

if on forming the pullback square

$$\begin{array}{ccc} \cdot & \rightarrowtail & [\![\psi(\overline{x},\overline{y})]\!] \\ {\scriptstyle a'}\downarrow & \lrcorner & \downarrow{\scriptstyle a} \\ [\![\phi(\overline{x})]\!] & \rightarrowtail & X \end{array},$$

in **C**, the morphism a' is an isomorphism.

3.4. Definition. If **T** is a lim theory with underlying language **L**, a *model* of **T** in a category **C** with finite limits is an **L**-structure in **C** which satisfies all the axioms of **T**. Let **T(C)** denote the category whose objects are such models and whose morphisms are *homomorphisms* of **L**-structures in **C**. Such a homomorphism $h{:}M\longrightarrow N$ is specified by a family $h_S{:}M(S)\longrightarrow N(S)$ of morphisms in **C** indexed by the sort symbols S of **L** such that: for each function symbol $f{:}S_1\times\cdots\times S_n\longrightarrow S$ in **L** the square

$$\begin{array}{ccc} M(S_1)\times\cdots\times M(S_n) & \xrightarrow{M(f)} & M(S) \\ {\scriptstyle h_{S_1}\times\cdots\times h_{S_n}}\downarrow & & \downarrow{\scriptstyle h_S} \\ N(S_1)\times\cdots\times N(S_n) & \xrightarrow[N(f)]{} & N(S) \end{array}$$

in **C** commutes; and for each relation symbol $R\rightarrowtail S_1\times\cdots\times S_n$ in **L** there is a commutative square in **C** of the form

$$\begin{array}{ccc} M(R) & \rightarrowtail & M(S_1)\times\cdots\times M(S_n) \\ \downarrow & & \downarrow{\scriptstyle h_{S_1}\times\cdots\times h_{S_n}} \\ N(R) & \rightarrowtail & N(S_1)\times\cdots\times N(S_n)\ . \end{array}$$

Composition and identities in **T**(**C**) are given componentwise from those in **C**.

Let us apply this definition to the two examples of lim theories given above. In the case of 3.1 and when **C** = **Set** the category of sets, evidently **cat**(**Set**) is just the category **Cat** of small categories and functors; and in general, **cat**(**C**) is the category of *internal categories and functors* in **C**, as defined in [J1, Chapter 2] for example. In the case of 3.2 the situation is more subtle. The objects of **lex**(**Set**) are small categories equipped with operations specifying terminal object and pullbacks; and the morphisms are functors which exactly preserve these operations. As is well known, the operation of taking a finite limit of any particular shape can be given as a derived operation from terminal object and pullbacks: thus we will refer to the objects of **lex**(**Set**) simply as *small categories with finite limits*, or *small lex categories*. But whilst a morphism in **lex**(**Set**) is necessarily a *lex functor* (that is, one which *preserves finite limits* in the usual sense of sending finite limit cones to finite limit cones), the converse is not generally the case since a lex functor need not preserve the given operations for terminal object and pullbacks up to equality. We will call the morphisms in **lex**(**Set**) *strict lex functors*.

3.5. Classifying categories. Consideration of the models of a lim theory **T** in *all* lex categories rather than just in the category of sets opens up the possibility of constructing a most general, or *generic* model of **T**. First note that models of **T** in different lex categories can be compared by transporting them along lex functors: given lex categories **C**, **D** and a model M of **T** in **C**, then for any lex functor $F{:}\mathbf{C}\longrightarrow\mathbf{D}$ there is a model $F(M)$ of **T** in **D**, obtained by applying F to the objects, morphisms and subobjects which comprise the structure M. (That $F(M)$ is a structure follows just from the fact that F preserves finite products and monomorphisms; to see that it is a **T**-model one needs that the satisfaction of lim sentences is preserved by F and this follows precisely from the preservation of finite limits.) Similarly, if $\phi{:}F\longrightarrow G$ is a natural transformation between lex functors, we get a homomorphism $\phi(M){:}F(M)\longrightarrow G(M)$ of **T**–models in **D** whose component at a sort symbol S is $\phi_{M(S)}$. In this way one obtains a functor

$$(-)(M) : \mathbf{LEX}(\mathbf{C},\mathbf{D}) \longrightarrow \mathbf{T}(\mathbf{D})$$

from the category of lex functors and natural transformations from **C** to **D** into the category of **T**-models in **D**. We can now state the fundamental result linking lim theories and lex categories:

Theorem. *For each lim theory* **T** *there is a small lex category* $\mathbf{C_T}$ *(called the* classifying category *of* **T***) and a model* $G_\mathbf{T}$ *of* **T** *in* $\mathbf{C_T}$ *(called the* generic model *of* **T***) such that for any lex category* **D** *the functor* $(-)(G_\mathbf{T}):\mathbf{LEX}(\mathbf{C_T},\mathbf{D})\longrightarrow\mathbf{T}(\mathbf{D})$ *is an equivalence of categories.*

□

Note that the above property of $\mathbf{C_T}$ and $G_\mathbf{T}$ uniquely determines the former up to equivalence of categories and the latter up to isomorphism of **T**-models. There are at least two ways of constructing the classifying category. The more elementary way is in terms of the syntax of **T**: see [Co1, 2.3]. The second way is model theoretic and depends upon the fact that **T**(**Set**) is a (typical) *locally finitely presentable* category in the sense of Gabriel and Ulmer [GU] (see also [MP]). Thus the full subcategory $\mathbf{T}(\mathbf{Set})_{fp} \hookrightarrow \mathbf{T}(\mathbf{Set})$ of finitely presentable set-valued **T**-models is equivalent to a small category and its opposite category is equivalent to the classifying category:

$$\mathbf{C_T} \simeq (\mathbf{T}(\mathbf{Set})_{fp})^{op}.$$

Any small lex category **D** can be presented, up to equivalence, as the classifying category of some lim theory. First define the *internal language* **L** of **D** to have a sort symbol $\ulcorner X \urcorner$ for each object X of **D**, a function symbol $\ulcorner f \urcorner : \ulcorner X_1 \urcorner \times \cdots \times \ulcorner X_n \urcorner \longrightarrow \ulcorner X \urcorner$ for each morphism $f:X_1\times\cdots\times X_n \longrightarrow X$ and a relation symbol $\ulcorner R \urcorner : \rightarrowtail \ulcorner X_1 \urcorner \times \cdots \times \ulcorner X_n \urcorner$ for each subobject $R:\rightarrowtail X_1\times\cdots\times X_n$. There is an evident **L**-structure M in **D** given by erasing "$\ulcorner$ $\urcorner$". Let **T** be the lim theory with underlying language **L** and whose axioms are all those lim sentences of **L** which are satisfied by M. Then M is by definition a **T**-model in **D**; hence by the universal property of the classifying category of **T**, there is a lex functor $F:\mathbf{C_T}\longrightarrow\mathbf{D}$ and an isomorphism $M \cong F(G_\mathbf{T})$ in **T**(**D**). Finally one can prove that the functor F is necessarily an equivalence, so that $\mathbf{D}\simeq\mathbf{C_T}$, as required.

The construction of classifying categories of lim theories provides a useful way of constructing lex categories with specified properties. We illustrate this with two examples which we will need later:

3.6.Example: *copower* of a lex category by a category.

Let **Cat** denote the 2-category of small categories, functors and natural transformations; and let **Lex** denote the 2-category of small lex categories, lex functors and natural transformations. Given **C** in **Cat** and **D** in **Lex**, we wish to construct a small lex category $\mathbf{C}\cdot\mathbf{D}$, called the *copower* of **D** by the category **C**, with the property that there is a natural equivalence:

$$\mathbf{Cat}(\mathbf{C},\mathbf{Lex}(\mathbf{D}, -)) \simeq \mathbf{Lex}(\mathbf{C}\cdot\mathbf{D}, -).$$

This amounts to giving a functor $(-)\cdot(-):\mathbf{C}\times\mathbf{D}\longrightarrow\mathbf{C}\cdot\mathbf{D}$ with the properties:

(*i*) $(-)\cdot(-)$ is *lex in its second variable*, i.e. for each $X\in\mathbf{C}$, $X\cdot(-):\mathbf{D}\longrightarrow\mathbf{C}\cdot\mathbf{D}$ is a lex functor;

(*ii*) if $B(-,-):\mathbf{C}\times\mathbf{D}\longrightarrow\mathbf{E}$ is any functor into a lex category which is lex in its second variable, then there is a lex functor $\overline{B}:\mathbf{C}\cdot\mathbf{D}\longrightarrow\mathbf{E}$, unique up to unique isomorphism, and a natural isomorphism $\overline{B}((-)\cdot(-))\cong B(-,-)$.

Therefore to construct $\mathbf{C}\cdot\mathbf{D}$ we define a language $\mathbf{L}$ as follows:

For each $X\in\mathbf{C}$ and $Y\in\mathbf{D}$ take a sort symbol $X\cdot Y$. For each $f:X\longrightarrow X'$ in $\mathbf{C}$ and $g:Y\longrightarrow Y'$ in $\mathbf{D}$ take a function symbol $f\cdot g:X\cdot Y\longrightarrow X'\cdot Y'$. There are no relation symbols.

And over this language we take the lim theory $\mathbf{T}$ with the following axioms:

(*i*) For each $X\in\mathbf{C}$ and $Y\in\mathbf{D}$, the axiom $\forall z:X\cdot Y(\, id_X\cdot id_Y(z)=z\,)$.

(*ii*) For $f:X\longrightarrow X'$, $f':X'\longrightarrow X''$ in $\mathbf{C}$ and $g:Y\longrightarrow Y'$, $g':Y'\longrightarrow Y''$ in $\mathbf{D}$, the axiom $\forall z:X\cdot Y(\, f'f\cdot g'g(z)=f'\cdot g'(f\cdot g(z))\,)$.

(*iii*) For each $X\in\mathbf{C}$, the axiom $\exists ! z:X\cdot 1(\, z=z\,)$ (where 1 denotes the terminal object in $\mathbf{D}$).

(*iv*) For each $X\in\mathbf{C}$ and each pullback square

$$\begin{array}{ccc} P & \xrightarrow{p'} & Y' \\ {\scriptstyle p}\downarrow & \lrcorner & \downarrow{\scriptstyle g'} \\ Y & \xrightarrow[g]{} & B \end{array}$$

in $\mathbf{D}$, the axiom

$$\forall z:X\cdot Y, z':X\cdot Y'\Big(id_X\cdot g(z)=id_X\cdot g'(z') \rightarrow \exists ! u:X\cdot P(id_X\cdot p(u)=z\wedge id_X\cdot p'(u)=z')\Big).$$

Clearly an $\mathbf{L}$-structure in a lex category $\mathbf{E}$ satisfying the lim sentences of types (*i*) and (*ii*) is precisely a functor $\mathbf{C}\times\mathbf{D}\longrightarrow\mathbf{E}$; and this functor is lex in its second variable if and only if the structure satisfies the lim sentences of types (*iii*) ("preserves terminal object in its second variable") and (*iv*)("preserves pullbacks in its second variable"). Note also that a homomorphism of $\mathbf{L}$-structures is precisely a natural transformation between the corresponding functors; and the functor transporting $\mathbf{T}$-models along a lex functor $F:\mathbf{E}\longrightarrow\mathbf{E}'$ is given by composing the corresponding functors with F. Thus $\mathbf{T}(-)\cong\mathbf{Cat}(\mathbf{C},\mathbf{Lex}(\mathbf{D},-))$ and hence the copower $\mathbf{C}\cdot\mathbf{D}$ in $\mathbf{Lex}$ is given by the classifying category $\mathbf{C}_{\mathbf{T}}$.

3.7. Example: *oplax colimits* of lex categories.

Let $\mathbf{C}$ be a small category. A *pseudofunctor* $\mathbf{D}:\mathbf{C}^{op}\longrightarrow\mathbf{Lex}$ is specified by the following information:

- for each object U of $\mathbf{C}$, a small lex category $\mathbf{D}(U)$,
- for each morphism $\alpha:U\longrightarrow V$ in $\mathbf{C}$, a lex functor $\alpha^*:\mathbf{D}(V)\longrightarrow\mathbf{D}(U)$,
- for each $U\in\mathbf{C}$, a natural isomorphism $\iota_U:id_{\mathbf{D}(U)}\cong(id_U)^*$,
- for each composable pair of morphisms $\alpha:U\longrightarrow V,\beta:V\longrightarrow W$ in $\mathbf{C}$, a natural

isomorphism $\kappa_{\alpha,\beta}:\alpha^*\circ\beta^*\cong(\beta\circ\alpha)^*$,

satisfying the coherence conditions that the diagrams

$$\begin{array}{ccc} \alpha^*\beta^*\gamma^* & \xrightarrow{\alpha^*\kappa_{\beta,\gamma}} & \alpha^*(\gamma\beta)^* \\ {\scriptstyle \kappa_{\alpha,\beta}\gamma^*}\downarrow & & \downarrow{\scriptstyle \kappa_{\alpha,\gamma\beta}} \\ (\beta\alpha)^*\gamma^* & \xrightarrow[\kappa_{\beta\alpha,\gamma}]{} & (\gamma\beta\alpha)^* \end{array} \quad \text{and} \quad \begin{array}{ccccc} id_{D(U)}\alpha^* & = & \alpha^* & = & \alpha^* id_{D(V)} \\ {\scriptstyle \iota_U\alpha^*}\downarrow & & \downarrow{\scriptstyle id} & & \downarrow{\scriptstyle \alpha^*\iota_V} \\ (id_U)^*\alpha^* & \xrightarrow[\kappa_{id,\alpha}]{} & \alpha^* & \xleftarrow[\kappa_{\alpha,id}]{} & \alpha^*(id_V)^* \end{array}$$

commute.

If $\mathbf{E}\in\mathbf{Lex}$, then an *oplax cone M under* $\mathbf{D}:\mathbf{C}^{op}\longrightarrow\mathbf{Lex}$ *with vertex* $\mathbf{E}$ is specified by:

- for each $U\in\mathbf{C}$, a lex functor $M_U:\mathbf{D}(U)\longrightarrow\mathbf{E}$,
- for each $\alpha:U\longrightarrow V$ in $\mathbf{C}$, a natural transformation $M_\alpha:M_V\longrightarrow M_U\circ\alpha^*$,

satisfying the coherence conditions

$$M_{\beta\alpha}=(M_U\kappa_{\alpha,\beta})\circ(M_\alpha\beta^*)\circ M_\beta \text{ and } M_{id_U}=M_U\iota_U\,.$$

Then the *oplax colimit* of the pseudofunctor $\mathbf{D}:\mathbf{C}^{op}\longrightarrow\mathbf{Lex}$, denoted $\underrightarrow{oplax}_{\mathbf{C}^{op}}\mathbf{D}$, is the small lex category which is the vertex of a universal oplax cone I under $\mathbf{D}$—that is, I should have the property that for any other oplax cone M with vertex $\mathbf{E}$ there exists a lex functor $\overline{M}:\underrightarrow{oplax}_{\mathbf{C}^{op}}\mathbf{D}\longrightarrow\mathbf{E}$ unique up to unique isomorphism with the property that there are natural isomorphisms $\mu_U:\overline{M}\circ I_U\cong M_U$ $(U\in\mathbf{C})$ satisfying

$$M_\alpha\circ\mu_V=(\mu_U\alpha^*)\circ(\overline{M}I_\alpha) \qquad (\alpha:U\longrightarrow V \text{ in } \mathbf{C}).$$

One can construct the oplax colimit $\underrightarrow{oplax}_{\mathbf{C}^{op}}\mathbf{D}$ as the classifying category of a suitable lim theory $\mathbf{T}$. The underlying language of $\mathbf{T}$ has

- sort symbols $U\cdot X$ for each $U\in\mathbf{C}$ and $X\in\mathbf{D}(U)$
- function symbols $\alpha\cdot f:V\cdot Y\longrightarrow U\cdot X$ for each $\alpha:U\longrightarrow V$ in $\mathbf{C}$ and $f:\alpha^*Y\longrightarrow X$ in $\mathbf{D}(U)$

and no relation symbols. The axioms of $\mathbf{T}$ are:

(*i*) For each sort symbol $U\cdot X$, the axiom $\forall z{:}U\cdot X(\,id\cdot\iota^{-1}(z)=z)$.

(*ii*) For $\alpha:U\longrightarrow V$, $\beta:V\longrightarrow W$ in $\mathbf{C}$, $f:\alpha^*Y\longrightarrow X$ in $\mathbf{D}(V)$ and $g:\beta^*Z\longrightarrow Y$ in $\mathbf{D}(W)$, the axiom $\forall z{:}W\cdot Z(\beta\alpha\cdot(f\circ\alpha^*g\circ\kappa^{-1})(z)=\alpha\cdot f(\beta\cdot g(z)))$.

(*iii*) For each $U\in\mathbf{C}$ and pullback square

$$\begin{array}{ccc} P & \xrightarrow{p'} & Y' \\ {\scriptstyle p}\downarrow & \lrcorner & \downarrow{\scriptstyle f'} \\ Y & \xrightarrow[f]{} & X \end{array}$$

in $\mathbf{D}(U)$, the axiom

$$\forall z{:}U\cdot Y, z'{:}U\cdot Y'\left(id\cdot f\iota^{-1}(z)=id\cdot f'\iota^{-1}(z')\rightarrow\exists!u{:}U\cdot P(id\cdot p\iota^{-1}(u)=z\wedge id\cdot p'\iota^{-1}(u)=z')\right).$$

If G denotes the generic model of **T** in its classifying category $\underrightarrow{oplax}_{\mathbf{C}^{op}}\mathbf{D}$, then the colimiting oplax cone cone I is given by

$$I_U(X) = G(U\cdot X)\ ,\ I_U(f) = G(id_U\cdot f\iota^{-1})\ \text{and}\ (I_\alpha)_X = G(\alpha\cdot id_X).$$

The proof that this works is a similar, but more complicated version of the argument in 3.6. Indeed the copower $\mathbf{C}^{op}\cdot\mathbf{D}$ is a special case of the oplax colimit construction, since we can take $\mathbf{C}^{op}\cdot\mathbf{D} = \underrightarrow{oplax}_{\mathbf{C}^{op}}\mathbf{D}$ where $\mathbf{D}\in\mathbf{Lex}$ is regarded as the constant pseudofunctor $\mathbf{C}^{op}\longrightarrow\mathbf{Lex}$ with value **D**.

We will see in 4.20(*ii*) that for one form of the model of the theory of constructions developed in section 4, the constant Orders K are denoted by small lex categories $[\![K]\!]\in\mathbf{Lex}$ and that more generally Orders L dependent on $X\in K$ are denoted by pseudofunctors $[\![L]\!]:[\![K]\!]^{op}\longrightarrow\mathbf{Lex}$. Then the lex category denoting the product $\prod X\in K.L$ is in fact obtained by taking the oplax colimit of $[\![L]\!]:[\![K]\!]^{op}\longrightarrow\mathbf{Lex}$; hence in particular the lex category denoting $K\rightarrow K'$ is given by the copower $[\![K]\!]^{op}\cdot[\![K']\!]$.

4 Algebraic toposes

In this section we will describe the first of our topos-theoretic models of the theory of constructions. For what follows, the basic reference is Johnstone's book [J1], especially chapters 2 and 4. We will denote the 2-category of Grothendieck toposes, geometric morphisms and natural transformations (between inverse image functors) by **GTOP**. Given a Grothendieck topos **E**, **GTOP(E)** will denote the (pseudo)slice 2-category whose objects are Grothendieck **E**-toposes, f:**F**⟶**E**, whose morphisms are triangles in **GTOP** commuting up to a given isomorphism and whose 2-cells are 2-cells in **GTOP** compatible with the given isomorphisms. In the case **E**=**Set**, since **Set** is terminal in **GTOP** we can identify **GTOP(Set)** with **GTOP**; however, even in the general case we will often confuse a Grothendieck **E**-topos with its domain topos **F** when the particular geometric morphism f:**F**⟶**E** is clear from the context.

GTOP(E) is the same as the 2-category **BTOP/E** of [J1, Chapter 4]. Just as in that reference, we will loosely refer to certain contructions in **GTOP(E)** as (finite) *limits* even though they are actually *bilimits*, i.e. given by universal properties which involve equivalences rather than isomorphisms of hom-categories. Thus for example, we said in the previous paragraph that **Set** is terminal in **GTOP**, meaning that for any **E**∈**GTOP** the category of geometric morphisms **GTOP(E,Set)** is *equivalent* to **1** (the one object, one morphism category) rather than isomorphic to it.

If **E**∈**GTOP** and **C** is an internal category in **E**, then the **E**-*topos of (internal) presheaves on* **C** will be denoted $[\mathbf{C}^{op},\mathbf{E}]$. Thus the objects of $[\mathbf{C}^{op},\mathbf{E}]$ are essentially the discrete fibrations over **C** in **E**: see [J1, 2.15]. Lifting the name from Johnstone's paper [J2], we make the following definition:

4.1. Definition. An **E**-topos f:**F**⟶**E** is *algebraic* if it is equivalent in **GTOP(E)** to $[\mathbf{C}^{op},\mathbf{E}]$ for some internal lex category **C**, i.e. for some model in **E** of the lim theory **lex** of 3.2.

ATOP(E) will denote the full sub-2-category of **GTOP**(E) whose objects are the algebraic E-toposes. (And in case E=**Set**, we speak simply of *algebraic toposes* and write **ATOP** for the corresponding full sub-2-category of **GTOP**.)

4.2. Proposition. *Algebraic* E-*toposes are stable under change of base: if* f:F⟶E *is a geometric morphism between Grothendieck toposes and* **A**⟶E *is an algebraic* E-*topos, then on forming the pullback square*

$$\begin{array}{ccc} \mathbf{B} & \longrightarrow & \mathbf{A} \\ \downarrow & \cong & \downarrow \\ \mathrm{F} & \longrightarrow & \mathrm{E} \end{array}$$

in **GTOP**, *it is the case that* **B**⟶F *is an algebraic* F-*topos.*

Proof. Suppose that $\mathbf{A}\simeq[\mathbf{C}^{op},\mathrm{E}]$ with $\mathbf{C}\in\mathbf{lex}(\mathrm{E})$. The inverse image functor f^*:E⟶F certainly preserves finite limits and hence as in 3.5 applying it to **C** yields $f^*(\mathbf{C})$, a model of **lex** in F. But by [J1, Corollary 4.35] there is a pullback square in **GTOP** of the form

$$\begin{array}{ccc} [(f^*\mathbf{C})^{op},\mathrm{F}] & \longrightarrow & [\mathbf{C}^{op},\mathrm{E}] \\ \downarrow & \cong & \downarrow \\ \mathrm{F} & \underset{f}{\longrightarrow} & \mathrm{E}\,. \end{array} \qquad (4.1)$$

Hence $\mathbf{B}\simeq[(f^*\mathbf{C})^{op},\mathrm{F}]$ is an algebraic F-topos.

□

4.3. Notation. If f:F⟶E is a geometric morphism between Grothendieck toposes, then

$$f^{\#}:\mathbf{GTOP}(\mathrm{E})\longrightarrow\mathbf{GTOP}(\mathrm{F})$$

will denote the operation of change of base, i.e. of pulling back E-toposes along f. (As we remarked above, pullbacks of toposes are strictly speaking "bipullbacks" and consequently $f^{\#}$ is a bicategory homomorphism—it preserves identities and composition only up to coherent isomorphism.) By the previous proposition, we can restrict $f^{\#}$ along the full inclusions **ATOP**(-)↪**GTOP**(-) to get $f^{\#}$:**ATOP**(E)⟶**ATOP**(F).

4.4. Classifying toposes. For any small lex category **C** and Grothendieck topos E there is a natural equivalence of categories:

$$\mathbf{GTOP}(\mathrm{E},[\mathbf{C}^{op},\mathbf{Set}])\simeq\mathbf{LEX}(\mathbf{C},\mathrm{E}) \qquad (4.2)$$

where the right hand side denotes the category of lex functors and natural transformations from **C** to E. This equivalence is given in one direction by sending a geometric morphism f:E⟶$[\mathbf{C}^{op},\mathbf{Set}]$ to the lex functor obtained by restricting f^* along the Yoneda embedding $\mathbf{C}\hookrightarrow[\mathbf{C}^{op},\mathbf{Set}]$; and it is given in the other direction by sending a lex fuctor F:**C**⟶E to the geometric morphism whose inverse image part is the left Kan extension of F along the

Yoneda embedding; see [J1, Proposition 7.13].

If T is a lim theory with classifying lex category $\mathbf{C}_{\mathrm{T}}$, we can combine (*4.2*) with the equivalence of 3.5 to obtain:

$$\mathbf{GTOP}(\mathbf{E},[\mathbf{C}_{\mathrm{T}}{}^{op},\mathbf{Set}]) \simeq \mathrm{T}(\mathbf{E}).$$

Thus $[\mathbf{C}_{\mathrm{T}}{}^{op},\mathbf{Set}]$ is the *classifying topos for the lim theory* T—meaning that the category of E-points of the topos is naturally equivalent to the category of T-models in E. As we noted in 3.5, any small lex category is the classifying category of some lim theory. Thus *the algebraic toposes are precisely the toposes which classify lim theories.*

Applying the Yoneda embedding $H:\mathbf{C}_{\mathrm{T}}\hookrightarrow[\mathbf{C}_{\mathrm{T}}{}^{op},\mathbf{Set}]$ to the generic model $G_{\mathrm{T}}\in\mathrm{T}(\mathbf{C}_{\mathrm{T}})$, we obtain a T-model $U_{\mathrm{T}}=H(G_{\mathrm{T}})$ in the classifying topos *which is generic amongst models of* T *in Grothendieck toposes*. This means in particular that for any $M\in\mathrm{T}(\mathbf{E})$ there is a geometric morphism $m:\mathbf{E}\longrightarrow[\mathbf{C}_{\mathrm{T}}{}^{op},\mathbf{Set}]$ with $m^*(U_{\mathrm{T}})\cong M$ in $\mathrm{T}(\mathbf{E})$. As we mentioned in 3.5, $\mathbf{C}_{\mathrm{T}}$ is equivalent to $(\mathrm{T}(\mathbf{Set})_{fp})^{op}$, the opposite of the full subcategory of $\mathrm{T}(\mathbf{Set})$ whose objects are the finitely presentable T-models. Using this fact, we can identify U_{T} concretely: since limits are calculated pointwise in functor categories

$$\mathrm{T}[\mathbf{C}_{\mathrm{T}}{}^{op},\mathbf{Set}] \cong \mathbf{CAT}(\mathbf{C}_{\mathrm{T}}{}^{op},\mathrm{T}(\mathbf{Set})) \simeq \mathbf{CAT}(\mathrm{T}(\mathbf{Set})_{fp},\mathrm{T}(\mathbf{Set}))$$

and under this equivalence, U_{T} corresponds to the inclusion $\mathrm{T}(\mathbf{Set})_{fp}\hookrightarrow\mathrm{T}(\mathbf{Set})$.

Lim theories and small lex categories correspond via the classifying category construction; but as we saw in 3.2, the latter are themselves the models of the particular lim theory **lex** and we can apply the above considerations to this theory:

4.5. Proposition. *There is a Grothendieck topos* Υ *and an algebraic* Υ*-topos* $\Sigma\longrightarrow\Upsilon$ *with the property that for any other Grothendieck topos* E *and any algebraic* E*-topos* A⟶E *there is a pullback square in* **GTOP** *of the form:*

$$\begin{array}{ccc} \mathbf{A} & \longrightarrow & \Sigma \\ \downarrow & \cong & \downarrow \\ \mathbf{E} & \longrightarrow & \Upsilon \end{array}\ .$$

Moreover, Υ *is itself an algebraic topos.*

Proof. Define Υ to be $[\mathbf{C}_{\mathbf{lex}}{}^{op},\mathbf{Set}]$, the classifying topos of the lim theory **lex**. Then Υ is certainly an algebraic topos. The generic model $U_{\mathbf{lex}}$ is an internal lex category in Υ: define Σ to be the algebraic Υ-topos $[U_{\mathbf{lex}}{}^{op},\Upsilon]$. If E is a Grothendieck topos and A⟶E an algebraic E-topos, then $\mathbf{A}\simeq[\mathbf{C}^{op},\mathbf{E}]$ for some $\mathbf{C}\in\mathbf{lex}(\mathbf{E})$. Let $f:\mathbf{E}\longrightarrow\Upsilon$ be a geometric morphism classifying the model C, i.e. for which there is an isomorphism $\mathbf{C}\cong f^*(U_{\mathbf{lex}})$ in $\mathbf{lex}(\mathbf{E})$. Then by (*4.1*), $\mathbf{A}\simeq f^*(\Sigma)$ in $\mathbf{GTOP}(\mathbf{E})$, as required.

□

We now turn our attention to the cartesian closed structure of algebraic toposes. Recall that the *exponential* in **GTOP**(E) of two E-toposes **F**,**G** if it exists is an E-topos, denoted $F \to_E G$, together with a geometric morphism $ev:(F \to_E G) \times_E F \longrightarrow G$ inducing equivalences

$$\mathbf{GTOP}(E)(-, F \to_E G) \simeq \mathbf{GTOP}(E)((-) \times_E F, G).$$

F is called *exponentiable* if $F \to_E G$ exists for all G. Not every topos is exponentiable; we refer the interested reader to the paper of Johnstone and Joyal [JJ] for a detailed analysis of this property. Luckily for us, the situation for algebraic toposes is quite simple: [J1, Remark 7.49] implies that every algebraic topos is exponentiable. Moreover, as the next proposition shows, **ATOP** is closed in **GTOP** under exponentiation:

4.6. Proposition. *If* C *and* D *are small lex categories, then the exponential of* $[D^{op}, \mathbf{Set}]$ *by* $[C^{op}, \mathbf{Set}]$ *in* **GTOP** *is* $[(C^{op} \cdot D)^{op}, \mathbf{Set}]$, *where* $C^{op} \cdot D$ *is the copower in* **Lex** *of* D *by the category* C^{op} *(as defined in Example 3.6).*

Proof. Let E be a Grothendieck topos and let $\Delta: \mathbf{Set} \longrightarrow E$ denote the constant-sheaf functor (the inverse image part of the (essentially) unique geometric morphism from E to **Set**). Then from (4.1), the product of E and $[C^{op}, \mathbf{Set}]$ in **GTOP** is $[(\Delta C)^{op}, E]$. An internal presheaf on the constant internal category ΔC can be identified with an external presheaf valued in E: thus $E \times [C^{op}, \mathbf{Set}]$ is the functor category $\mathbf{CAT}(C^{op}, E)$. A geometric morphism out of $\mathbf{CAT}(C^{op}, E)$ is determined by its inverse image part—which is precisely a functor preserving finite limits and small colimits. Then since limits and colimits in such a functor category are calculated pointwise from E, we have

$$\mathbf{GTOP}(E \times [C^{op}, \mathbf{Set}], -) \simeq \mathbf{CAT}(C^{op}, \mathbf{GTOP}(E, -)). \qquad (4.3)$$

Combining (4.3) with (4.2) and using the universal property of the copower $C^{op} \cdot D$ (which we note from its construction in Example 3.6, is valid for lex functors valued in any lex category and not just a small one), we have:

$$\begin{aligned} \mathbf{GTOP}(E \times [C^{op}, \mathbf{Set}], [D^{op}, \mathbf{Set}]) &\simeq \mathbf{CAT}(C^{op}, \mathbf{GTOP}(E, [D^{op}, \mathbf{Set}])) \\ &\simeq \mathbf{CAT}(C^{op}, \mathbf{LEX}(D, E)) \\ &\simeq \mathbf{LEX}(C^{op} \cdot D, E) \\ &\simeq \mathbf{GTOP}(E, [(C^{op} \cdot D)^{op}, \mathbf{Set}]). \end{aligned}$$

These equivalences are natural in E and show that $[(C^{op} \cdot D)^{op}, \mathbf{Set}]$ has the correct universal property to be the exponential $[C^{op}, \mathbf{Set}] \to [D^{op}, \mathbf{Set}]$.

□

The proof of Proposition 4.6 is constructive and in a form admitting *relativization* to the category theory of any topos E with natural number object (and in particular to any Grothendieck topos), where the internal categories of E play the role of small categories and categories fibred over E play the role of large ones. (See [B2] for a general discussion of category theory relative to a base other than **Set** and [PS] for a development of aspects of this theory using indexed categories (pseudofunctors) rather than fibrations.) We therefore have:

4.7. Proposition. *For any Grothendieck topos* E, *the algebraic* E*-toposes are exponentiable objects of* **GTOP**(E) *and* **ATOP**(E) *is closed in* **GTOP**(E) *under exponentiation: for* C,D∈**lex**(E) *the exponential of* $[D^{op},E]$ *by* $[C^{op},E]$ *is* $[(C^{op}\cdot D)^{op},E]$ *where* $C^{op}\cdot D$ *is the copower of the internal lex category* D *by the internal category* C^{op}.

□

4.8. Corollary. *Each* **ATOP**(E) *is a cartesian closed bicategory and finite products and exponentials are preserved by the inclusion* **ATOP**(E)$\hookrightarrow$**GTOP**(E). *They are also preserved by the operation* $f^{\#}$:**ATOP**(E)⟶**ATOP**(F) *of pullback along a geometric morphism* f:F⟶E.

Proof. In view of the previous proposition, for the first sentence of the corollary we just have to see that **ATOP**(E) is closed in **GTOP**(E) under finite products. Proposition 4.2 implies that **ATOP**(E) is closed in **GTOP**(E) under taking binary products; indeed by [J1, Corollary 4.36], we can take the product $[C^{op},E]\times_E[D^{op},E]$ to be $[(C\times D)^{op},E]$. The terminal object of **GTOP**(E) is certainly algebraic, since $E\simeq[1^{op},E]$ where **1** is the trivial internal lex category.

For the second sentence, we just have to show that $f^{\#}$ preserves exponentials, since clearly it preserves finite limits. Since the inclusions **ATOP**(−)$\hookrightarrow$**GTOP**(−) preserve exponentials, it is sufficient to prove that $f^{\#}$:**GTOP**(E)⟶**GTOP**(F) preserves any existing exponentials. This is so because $f^{\#}$ has a left adjoint $f_!$:**GTOP**(F)⟶**GTOP**(E) ("compose with f") satisfying the condition of "Frobenius reciprocity": for G∈**GTOP**(E) and H∈**GTOP**(F), simple properties of pullbacks give that $f_!(H\times_F f^{\#}G)\simeq(f_!H)\times_E G$. Thus if the exponential $G\to_E G'$ exists in **GTOP**(E), then

$$\begin{aligned}\mathbf{GTOP}(F)(-,f^{\#}(G\to_E G')) &\simeq \mathbf{GTOP}(E)(f_!(-),G\to_E G')\\ &\simeq \mathbf{GTOP}(E)(f_!(-)\times_E G,G')\\ &\simeq \mathbf{GTOP}(E)(f_!(-\times_F f^{\#}G),G')\\ &\simeq \mathbf{GTOP}(F)(-\times_F f^{\#}G,f^{\#}G'),\end{aligned}$$

so that the exponential $(f^{\#}G)\to_F(f^{\#}G')$ exists in **GTOP**(F) and is given by $f^{\#}(G\to_E G')$.

□

In the case that f:F⟶E is algebraic, the left adjoint $f_!$ mentioned in the above proof restricts to give a left adjoint to $f^{\#}$:**ATOP**(E)⟶**ATOP**(F), as we now show:

4.9. Proposition. *If* f:F⟶E *is an algebraic* E*-topos and* g:G⟶F *is an algebraic* F*-topos, then* gf:G⟶E *is an algebraic* E*-topos.*

Proof. Consider first the special case when f is an equivalence. Then G⟶F≃E is the pullback of G⟶F along the inverse equivalence E≃F; hence the composition is algebraic by Proposition 4.2.

Now in the general case, suppose that $F\simeq[C^{op},E]$ with C∈**lex**(E). Then by the previous paragraph, $G\longrightarrow F\simeq[C^{op},E]$ is an algebraic $[C^{op},E]$-topos; hence $G\simeq[D^{op},[C^{op},E]]$ for some D∈**lex**($[C^{op},E]$). By [J1, Exercise 2.7], $[D^{op},[C^{op},E]]\simeq[(\mathit{Gr}D)^{op},E]$ where GrD∈**cat**(E) is the result of applying the *Grothendieck construction* to D∈**cat**($[C^{op},E]$). In the case $\mathbf{E}=\mathbf{Set}$, this construction can be described as follows: identifying $\mathbf{D}$ with a functor $\mathbf{C}^{op}\longrightarrow\mathbf{Cat}$, then $Gr\mathbf{D}$ is the category whose objects are pairs (U,X) with $U\in\mathbf{C}$ and $X\in\mathbf{D}(U)$, and whose morphisms $(U,X)\longrightarrow(V,Y)$ are pairs (α,f) where $\alpha:U\longrightarrow V$ in $\mathbf{C}$ and $f:X\longrightarrow\mathbf{D}(\alpha)(Y)$ in $\mathbf{D}(U)$. For a general (Grothendieck) topos $\mathbf{E}$, the construction is just that obtained by translating the above recipe for $Gr\mathbf{D}$ into the internal logic of $\mathbf{E}$. It is straightforward to show that when $\mathbf{D}\in\mathbf{cat}[\mathbf{C}^{op},\mathbf{E}]$ is actually an internal *lex* category, then so is $Gr\mathbf{D}$. Therefore $\mathbf{G}\simeq[(Gr\mathbf{D})^{op},\mathbf{E}]$ is an algebraic $\mathbf{E}$-topos.

□

4.10. Corollary. *If* $f:\mathbf{F}\longrightarrow\mathbf{E}$ *is an algebraic* $\mathbf{E}$*-topos, then the left adjoint to* $f^{*}:\mathbf{GTOP(E)}\longrightarrow\mathbf{GTOP(F)}$, *viz. the operation of composing with* f, *restricts to give a left adjoint to* $f^{*}:\mathbf{ATOP(E)}\longrightarrow\mathbf{ATOP(F)}$, *denoted* $f_{!}:\mathbf{ATOP(F)}\longrightarrow\mathbf{ATOP(E)}$.

These left adjoints satisfy a Beck-Chevalley condition *with respect to pullback squares in* **GTOP**: *if*

$$\begin{array}{ccc} \mathbf{H} & \xrightarrow{q} & \mathbf{G} \\ {\scriptstyle h}\downarrow & \cong & \downarrow{\scriptstyle g} \\ \mathbf{F} & \xrightarrow[f]{} & \mathbf{E} \end{array}$$

is a pullback with f *(and hence also* q*) algebraic, then the canonical natural transformation* $q_{!}\circ h^{*}\longrightarrow g^{*}\circ f_{!}$ *is an equivalence.*

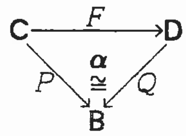
Proof. The first paragraph is a consequence of Propositition 4.9; and the second follows from the fact that the Beck-Chevalley condition holds for the left adjoints to pulling back in **GTOP**, due to the usual elementary properties of pullbacks with respect to composition.

□

So far, the results in this section have all been obtained by marshalling well known facts about presheaf toposes and classifying toposes. We are now going to show that the dual of Corollary 4.10 holds, namely that $f^{*}:\mathbf{ATOP(E)}\longrightarrow\mathbf{ATOP(F)}$ possesses a right adjoint when $f:\mathbf{F}\longrightarrow\mathbf{E}$ is an algebraic $\mathbf{E}$-topos (and that these right adjoints satisfy the Beck-Chevalley condition). The method we employ is (the bicategorical version of) that in Proposition 2.6: we show that exponentiation by an algebraic topos preserves geometric morphisms which are algebraic, and then construct the right adjoints to pulling back using exponentials. To carry out this plan we have to delve a little more deeply into the structure of internal lex categories.

4.11. Definition. A lex functor $P:\mathbf{C}\longrightarrow\mathbf{B}$ between (small) lex categories is a *lex fibration over* $\mathbf{B}$ if it possesses a right adjoint $P_{*}:\mathbf{B}\longrightarrow\mathbf{C}$ and the counit of the adjunction, $\varepsilon:P\circ P_{*}\longrightarrow 1_{\mathbf{B}}$, is an isomorphism. (Note that P_{*} is necessarily a lex functor.)

If $Q:\mathbf{D}\longrightarrow\mathbf{B}$ is another lex fibration, then a morphism



in **Lex**/**B** is *cartesian* if the natural transformation $F \circ P_* \longrightarrow Q_*$ (obtained from α by transposing across the adjunctions $P \dashv P_*$, $Q \dashv Q_*$) is an isomorphism.

4.12. Remark. A notion of (cloven) fibration can be given in any bicategory with finite (bi)limits: see Street [S]. When the bicategory is **Lex**, this notion reduces to the above, particularly simple one. (*Cf.* [RW1] and [RW2].) Because it is the *bicategorical* rather than 2-categorical notion of fibration, it is not the case that a lex fibration is the same thing as a cloven fibration of categories (in the classical sense [Gr], [B2]) all of whose fibres and pullback functors are lex. However the two concepts only differ up to equivalence in **Lex**/**B**. To see this, note that each lex fibration P:**C**$\longrightarrow$**B** determines a pseudofunctor **C**(-):$\mathbf{B}^{op}\longrightarrow$**Lex** (*cf.* 3.7), where each **C**(U) ($U \in$ **B**) is the full subcategory of **C**/$P_*(U)$ whose objects are those $x:X \longrightarrow P_*(U)$ with $P(x)$ an isomorphism; and for $\alpha:U \longrightarrow V$ in **B**, α^*:**C**(V)$\longrightarrow$**C**(U) is given by pullback in **C** along $P_*(\alpha)$. Applying the Grothendieck construction (*cf.* the proof of Proposition 4.9) to **C**(-):$\mathbf{B}^{op}\longrightarrow$**Lex**, one obtains a cloven fibration in the classical sense which is equivalent over **B** to the original functor P.

It is not hard to show that the above assignment of pseudofunctors to lex fibrations extends to give an equivalence between the 2-category of lex fibrations over **B**, cartesian lex functors and natural transformations on the one hand and the 2-category of pseudofunctors $\mathbf{B}^{op}\longrightarrow$**Lex**, pseudonatural transformations and modifications on the other. Note that if **D**:$\mathbf{B}^{op}\longrightarrow$**Lex** is a pseudofunctor, then the corresponding lex fibration P:Gr(**D**)$\longrightarrow$**B** has P equal to the projection $(U,X) \longmapsto U$, with right adjoint P_* sending $U \in$ **B** to $(U,1)$ where $1 \in$ **D**(U) is the terminal object.

4.13. Notation. If **C** is a small lex category, we will denote by $\hat{\mathbf{C}}$ the algebraic topos [$\mathbf{C}^{op}$,**Set**]. The assignment $\mathbf{C} \longmapsto \hat{\mathbf{C}}$ extends to a bicategory homomorphism $(\hat{-})$:$\mathbf{Lex}^{op}\longrightarrow$**GTOP** via the natural equivalence of (*4.2*). In particular a lex morphism F:**C**$\longrightarrow$**D** determines a geometric morphism $\hat{F}:\hat{\mathbf{D}}\longrightarrow\hat{\mathbf{C}}$ whose inverse image functor is left Kan extension along F and whose direct image functor is precomposition with F.

4.14. Proposition. *Let* **B** *be a small lex category. A Grothendieck* $\hat{\mathbf{B}}$*-topos* $\mathbf{E}\longrightarrow\hat{\mathbf{B}}$ *is algebraic iff it is equivalent to* $\hat{P_*}:\hat{\mathbf{C}}\longrightarrow\hat{\mathbf{B}}$ *for some lex fibration* P:**C**$\longrightarrow$**B**.

Proof. Suppose **E** is an algebraic $\hat{\mathbf{B}}$-topos—say $\mathbf{E} \simeq [\mathbf{D}^{op},\hat{\mathbf{B}}]$ with $\mathbf{D} \in \mathbf{lex}(\hat{\mathbf{B}})$. Then just as in the proof of Proposition 4.9, we can regard **D** as a functor $\mathbf{B}^{op}\longrightarrow$**lex**(**Set**), hence as a functor $\mathbf{B}^{op}\longrightarrow$**Lex**, apply the Grothendieck construction to it and obtain $\mathbf{E} \simeq [(Gr\mathbf{D})^{op},\mathbf{Set}]$ in **GTOP**($\hat{\mathbf{B}}$). The geometric morphism which defines $[(Gr\mathbf{D})^{op},\mathbf{Set}]$ as a topos over $[\mathbf{B}^{op},\mathbf{Set}]$ is just that whose inverse image functor is precomposition with the projection P:Gr(**D**)$\longrightarrow$**B**; but as we noted above, P is indeed a lex fibration and $(-)\cdot P:[\mathbf{B}^{op},\mathbf{Set}]\longrightarrow[(Gr\mathbf{D})^{op},\mathbf{Set}]$ is naturally isomorphic to the functor given by left Kan extension along the right adjoint P_*, i.e. to the inverse image part of the geometric morphism $\hat{P}_*$.

Conversely, given a lex fibration P:**C**⟶**B**, to show that $\hat{\mathbf{C}}$ is an algebraic $\hat{\mathbf{B}}$-topos we have to find a functor **D**:$\mathbf{B}^{op}$⟶**lex(Set)** with Gr(**D**)⟶**B** equivalent to P:**C**⟶**B** in **Lex/B**. Using the correspondence between lex fibrations over **B** and pseudofunctors $\mathbf{B}^{op}$⟶**Lex** remarked upon in 4.12, this amounts to proving:

4.15. Lemma. *Every pseudofunctor* $\mathbf{B}^{op}$⟶**Lex** *is pseudonaturally equivalent to a functor* $\mathbf{B}^{op}$⟶**Lex** *whose value at any morphism of* **B** *is a* strict *lex functor.*

Proof. The lemma can be viewed as a corollary of recent work of G.M.Kelly and J.Power showing how to turn "pseudo" structures in 2-categories into equivalent "strict" structures. Here we shall give a direct proof of this particular result.

With notation as in Remark 4.12, we can assume that the pseudofunctor is of the form **C**(-):$\mathbf{B}^{op}$⟶**Lex** for some lex fibration P:**C**⟶**B**. For each $\alpha:U\longrightarrow V$ in **B**, let $P_*\alpha$:**C**/P_*U⟶**C**/P_*V denote the functor between slice categories given by composition with $P_*\alpha$. Then because we can calculate limits in categories of presheaves pointwise from **Set**, we get that $(-)\circ(P_*\alpha)^{op}$:[(**C**/$P_*V)^{op}$,**Set**]⟶[(**C**/$P_*U)^{op}$,**Set**] is a strict lex functor. Moreover, the assignment $\alpha\mapsto(-)\circ(P_*\alpha)^{op}$ preserves identities and compositions. Thus we have a *functor*

$$\check{\mathbf{C}}(-) =_{def} [(\mathbf{C}/P_*(-))^{op},\mathbf{Set}] : \mathbf{B}^{op} \longrightarrow \mathbf{LEX}$$

whose values at morphisms are strict lex. The composition of the inclusion **C**(U)⟶**C**/$P_*(U)$ with the Yoneda embedding yields a full and faithful lex functor I_U:**C**(U)⟶$\check{\mathbf{C}}(U)$ which is pseudonatural in U (because the functors $P_*\alpha$:**C**/P_*U⟶**C**/P_*V are left adjoint to the pullback functors $(P_*\alpha)^*$:**C**/P_*V⟶**C**/P_*U). Then defining **C**'(U)⟶$\check{\mathbf{C}}(U)$ to be the least full subcategory containing all objects of the form $\check{\mathbf{C}}(\alpha)(I_V(Y))$ and closed under the given operations for terminal object and pullbacks (inherited from those for **Set**), we get a functor **C**'(-):$\mathbf{B}^{op}$⟶**Lex** whose values on morphisms are strict; and I restricts to a pseudonatural transformation **C**(-)⟶**C**'(-) whose components are not only full and faithful but also essentially surjective (since **C**(U) and I_U are lex)—and hence I yields a pseudonatural equivalence, as required.

□

Using the above lemma, we can complete the second half of the proof of Proposition 4.14: starting with a lex fibration P:**C**⟶**B**, form **C**':$\mathbf{B}^{op}$⟶**lex(Set)** as in 4.15; then since **C**(-)≃**C**'(-), the Grothendieck construction yields **C**≃Gr(**C**') in **Lex/B** and hence $\hat{\mathbf{C}}\simeq(Gr\mathbf{C}')^\wedge\simeq[(\mathbf{C}')^{op},\hat{\mathbf{B}}]$ in **GTOP**($\hat{\mathbf{B}}$) with **C**'∈**lex**($\hat{\mathbf{B}}$)—so that $\hat{\mathbf{C}}$ is indeed an algebraic $\hat{\mathbf{B}}$-topos.

□

4.16. Corollary. *Let* **E** *be a Grothendieck topos and* f:**F**⟶**E** *an algebraic* **E**-*topos. Exponentiating by any algebraic topos* **G** *yields an algebraic* (**G**→**E**)-*topos,* ($\mathbf{G}\rightarrow f$):(**G**→**F**)⟶(**G**→**E**).

Proof. Consider first the case in which **E** is itself algebraic—say $\mathbf{E}=\hat{\mathbf{B}}$ with $\mathbf{B}\in\mathbf{Lex}$. Then by Proposition 4.14, we can take f to be $\hat{P}_*:\hat{\mathbf{C}}\longrightarrow\hat{\mathbf{B}}$ with $P:\mathbf{C}\longrightarrow\mathbf{B}$ a lex fibration. Supposing that $\mathbf{G}=\hat{\mathbf{D}}$ with $\mathbf{D}\in\mathbf{Lex}$, then we have from Proposition 4.6 that $(\mathbf{G}\to\mathbf{E})$ is $(\mathbf{D}^{op}\cdot\mathbf{B})\hat{}$ and that $(\mathbf{G}\to\mathbf{F})$ is $(\mathbf{D}^{op}\cdot\mathbf{C})\hat{}$. Moreover the calculations in the proof of that proposition imply that $(\mathbf{G}\to f)$ is the geometric morphism $(\mathbf{D}^{op}\cdot P_*)\hat{}$. But $\mathbf{D}^{op}\cdot(-):\mathbf{Lex}\longrightarrow\mathbf{Lex}$ is a homomorphism of bicategories; therefore $\mathbf{D}^{op}\cdot P_*$ is right adjoint to $\mathbf{D}^{op}\cdot P$ with counit an isomorphism, i.e. $\mathbf{D}^{op}\cdot P$ is a lex fibration. Hence by Proposition 4.14 again, $(\mathbf{G}\to f):(\mathbf{G}\to\mathbf{F})\longrightarrow(\mathbf{G}\to\mathbf{E})$ is an algebraic $(\mathbf{G}\to\mathbf{E})$-topos.

Now consider the general case in which **E** is an arbitrary Grothendieck topos. We can always find a small site of definition for **E** whose underlying category has finite limits (see [J1, Corollary 0.46]). In other words, we can find $\mathbf{B}\in\mathbf{Lex}$ and a geometric inclusion $i:\mathbf{E}\hookrightarrow\hat{\mathbf{B}}$. Since $\mathbf{F}\longrightarrow\mathbf{E}$ is algebraic there is $\mathbf{C}\in\mathbf{lex}(\mathbf{E})$ with $\mathbf{F}\simeq[\mathbf{C}^{op},\mathbf{E}]$ in $\mathbf{GTOP}(\mathbf{E})$. The direct image functor $i_*:\mathbf{E}\longrightarrow\hat{\mathbf{B}}$ is lex and so we can transport **C** along it to get $i_*(\mathbf{C})\in\mathbf{lex}(\hat{\mathbf{B}})$. Then as in *(4.1)* there is a pullback square in **GTOP** of the form:

$$\begin{array}{ccc} [(i^*i_*\mathbf{C})^{op},\mathbf{E}] & \hookrightarrow & [(i_*\mathbf{C})^{op},\hat{\mathbf{B}}] \\ \downarrow & \cong & \downarrow \\ \mathbf{E} & \underset{i}{\hookrightarrow} & \hat{\mathbf{B}}. \end{array}$$

Since i is an inclusion, $i^*i_*\mathbf{C}\cong\mathbf{C}$ in $\mathbf{lex}(\mathbf{E})$; hence there is a pullback square of the form

$$\begin{array}{ccc} \mathbf{F} & \hookrightarrow & \mathbf{H} \\ {\scriptstyle f}\downarrow & \cong & \downarrow{\scriptstyle h} \\ \mathbf{E} & \underset{i}{\hookrightarrow} & \hat{\mathbf{B}} \end{array}$$

with $\hat{\mathbf{B}}$ algebraic and $h:\mathbf{H}\longrightarrow\hat{\mathbf{B}}$ an algebraic $\hat{\mathbf{B}}$-topos. Now $(\mathbf{G}\to -):\mathbf{GTOP}\longrightarrow\mathbf{GTOP}$ preserves pullbacks (since it has a left (bi)adjoint). Applying it to the above square therefore gives that $(\mathbf{G}\to f)$ is the pullback of $(\mathbf{G}\to h)$ along $(\mathbf{G}\to i)$; but $(\mathbf{G}\to h)$ is algebraic by the special case considered above and hence by Proposition 4.2, the pullback $(\mathbf{G}\to f)$ is also algebraic.

□

As with the previous results on exponentiation, the above corollary admits of relativization from **Set** to the category theory of an arbitrary Grothendieck topos **E**:

4.17. Corollary. *If* $\mathbf{F}\longrightarrow\mathbf{E}$ *is a Grothendieck* **E**-*topos,* $\mathbf{A}\longrightarrow\mathbf{E}$ *an algebraic* **E**-*topos and* $b:\mathbf{B}\longrightarrow\mathbf{F}$ *an algebraic* **F**-*topos, then* $(\mathbf{A}\to_{\mathbf{E}}b):(\mathbf{A}\to_{\mathbf{E}}\mathbf{B})\longrightarrow(\mathbf{A}\to_{\mathbf{E}}\mathbf{F})$ *is an algebraic* $(\mathbf{A}\to_{\mathbf{E}}\mathbf{F})$-*topos.*

□

We are now in a position to prove the dual version of Corollary 4.10:

4.18. Proposition.

(*i*) *If* $f:\mathbf{F}\longrightarrow\mathbf{E}$ *is an algebraic* **E**-*topos, then* $f^{\#}:\mathbf{ATOP}(\mathbf{E})\longrightarrow\mathbf{ATOP}(\mathbf{F})$ *has a right adjoint,* *denoted* $f_{\#}$:**ATOP(F)**$\longrightarrow$**ATOP(E)**.

(*ii*) *The adjoints of* (*i*) *satisfy a Beck-Chevalley condition: given a pullback square*

$$\begin{array}{ccc} H & \xrightarrow{q} & G \\ {\scriptstyle h}\big\downarrow & \cong & \big\downarrow{\scriptstyle g} \\ F & \xrightarrow[f]{} & E \end{array}$$

in **GTOP** *with* f *(and hence* q*) algebraic, the canonical natural transformation* $g^{\#}\circ f_{\#}\longrightarrow q_{\#}\circ h^{\#}$ *is an equivalence.*

Proof. We construct the right adjoint as in Proposition 2.6. Thus given b:**B**$\longrightarrow$**F** in **ATOP(F)**, define $f_{\#}$(**B**)$\longrightarrow$**E** via the pullback

$$\begin{array}{ccc} f_{\#}B & \longrightarrow & (F\to_E B) \\ \big\downarrow & \cong & \big\downarrow{\scriptstyle (F\to_E b)} \\ E & \xrightarrow[\ulcorner 1_F \urcorner]{} & (F\to_E F) \end{array}$$

in **GTOP(E)**, where $\ulcorner 1_F \urcorner$ denotes the exponential transpose of $F\times_E E\simeq F\xrightarrow{1} F$. Combining Corollary 4.17 with Proposition 4.2, we certainly have that $f_{\#}$**B** is an algebraic **E**-topos. To see that it has the right universal property, first note that from the universal property of pullbacks, morphisms

$$\begin{array}{ccc} H & \dashrightarrow & f_{\#}B \\ & \searrow \;\cong & \big\downarrow \\ & & E \end{array}$$

in **GTOP(E)** correspond to diagrams in **GTOP** of the form

$$\begin{array}{ccc} H & \dashrightarrow & (F\to_E B) \\ \big\downarrow & \cong & \big\downarrow{\scriptstyle (F\to_E b)} \\ E & \xrightarrow[\ulcorner 1_F \urcorner]{} & (F\to_E F)\,. \end{array}$$

Transposing such a diagram across the exponential adjunction gives

$$\begin{array}{ccc} f^{\#}H & \dashrightarrow & B \\ \big\downarrow & \cong & \big\downarrow{\scriptstyle b} \\ F & \xrightarrow[1]{} & F \end{array}$$

in **GTOP(F)**. Thus **GTOP(E)**(**H**,$f_{\#}$**B**) $\simeq$ **GTOP(F)**($f^{\#}$**H**,**B**) and the equivalence is evidently natural in **H**$\longrightarrow$**E**. This proves not just (*i*), but in fact something slightly more, namely: *when* f *is algebraic then the value of the right adjoint to* $f^{\#}$:**GTOP(E)**$\longrightarrow$**GTOP(F)** *exists at any algebraic* **E**-*topos*. This also suffices to prove (*ii*), since whenever both $f_{\#}$**B** $\in$ **GTOP(E)** and $q_{\#}(h^{\#}\mathbf{B}) \in \mathbf{GTOP}(\mathbf{G})$ are defined, then elementary properties of pullbacks with respect to composition imply that $g^{*}(f_{\#}\mathbf{B})$ and $q_{\#}(h^{*}\mathbf{B})$ are canonically equivalent in **GTOP(G)**.

□

4.19. Algebraic topos model of the theory of constructions. Collecting together the results of this section, we present our first example of an instance of the categorical structure set out in section 2. In fact it is an example of the theory of constructions with "*Type* ≃ *ORDER*" (see 1.12) and so we show how to fulfil the conditions listed in 2.15:

(*i*) The category **B** is obtained from the 2-category **ATOP** (*cf* 4.1) by taking *isomorphism classes* of 1-cells. Thus **B** has for its objects the algebraic toposes and for its morphisms isomorphism classes of geometric morphisms: in other words we identify two geometric morphisms $f,g:\mathbf{F}\longrightarrow\mathbf{E}$ between algebraic toposes if they are isomorphic objects in **GTOP(F,E)**. Composition and identities in **B** are those inherited from **GTOP**. This category **B** certainly has a terminal object, namely **Set**—*cf*. 4.8.

(*ii*) The class of morphisms **A** is that determined by the geometric morphisms $f:\mathbf{F}\longrightarrow\mathbf{E}$ between algebraic toposes which make **F** an algebraic **E**-topos.

(*iii*) Note that by Proposition 4.9, if **E** is in **B** and $f:\mathbf{F}\longrightarrow\mathbf{E}$ is an algebraic **E**-topos, then **F** is also in **B**. Consequently Propositions 4.2 implies that the pullback of a morphism in **A** along an arbitrary morphism of **B** exists and is again in **A**.

(*iv*) **A** is closed under composition by Proposition 4.9.

(*v*) If $f:\mathbf{F}\longrightarrow\mathbf{E}$ is in **A**, then the pullback functor $f^{*}:\mathbf{A(E)}\longrightarrow\mathbf{A(F)}$ has a right adjoint satisfying the Beck-Chevalley condition by Proposition 4.18.

(*vi*) The topos $\mathbf{\Upsilon}$ of Proposition 4.5 is in **B** and the algebraic $\mathbf{\Upsilon}$-topos $\mathbf{\Sigma}\longrightarrow\mathbf{\Upsilon}$ of that proposition determines a morphism in **A** with the property that any other morphism in **A** can be obtained from it by pullback.

(*vii*) Finally, for each object **E** in **B**, the unique morphism $\mathbf{E}\longrightarrow\mathbf{Set}$ is in **A** because we chose **B** to consist only of *algebraic* toposes.

4.20. Equivalent descriptions of the model. We state without proofs two examples of the categorical structure in 2.15 which are both equivalent to that given in 4.19. These equivalent versions (especially the second) have advantages when it comes to making *calculations* in the model. As we remarked in 2.13, we can specify these equivalent forms of the model by giving equivalent versions of the underlying category **B** (and taking the essential image of the class **A** under the equivalence).

(i) *A version in the style of domain theory.* Instead of looking at the algebraic toposes **E** themselves, we can look at their *categories of points*, **GTOP(Set,E)**. Supposing that $\mathbf{E}\simeq[\mathbf{C}^{op},\mathbf{Set}]$ with $\mathbf{C}\in\mathbf{Lex}$, then by (*4.2*) this category of points is equivalent to **LEX(C,Set)**, which is a typical *locally finitely presentable* (*lfp*) category. (See [GU] and [MP].) If $\mathbf{F}\simeq[\mathbf{D}^{op},\mathbf{Set}]$ is another algebraic topos, then it is the case that a functor

$$\mathbf{GTOP(Set,E)}\simeq\mathbf{LEX(C,Set)}\longrightarrow\mathbf{LEX(D,Set)}\simeq\mathbf{GTOP(Set,F)}$$

is induced by composition with a geometric morphism if and only if the functor preserves filtered colimits. In this way we get *an equivalence between* **ATOP** (defined in 4.1) *and the 2-category consisting of lfp categories, functors preserving filtered colimits and natural transformations*. Using this equivalence we get an alternative description of the structure in 4.19 in terms of lfp categories and filtered colimit preserving functors. The class of display morphisms is perhaps most simply described as comprising those functors between lfp categories which preserve limits and filtered colimits and have filtered colimit preserving right adjoints with counit of the adjunction an isomorphism.

The model in this form has been studied by Coquand (see [CE, section 5]), although not in terms of the systematic framework developed in section 2. From the point of view of domain theory, it is very natural to approach the model in this way, since lfp categories directly generalize algebraic lattices and filtered colimit preserving functors generalize continuous maps beteen cpo's. Perhaps the main advantage of the topos-theoretic treatment we have given in this section is the extremely simple way in which the existence of a generic family of Orders (4.19(vi)) is demonstrated using standard properties of classifying toposes; we hope that the use of this technique will lead to the discovery of other models.

(ii) A version in the style of Scott's information systems. Instead of dealing with lfp categories, we can work directly with the (essentially) small lex categories which determine them. (Recall that an lfp category **A** is equivalent to **LEX**(**C**, **Set**) when $\mathbf{C} \simeq (\mathbf{A}_{fp})^{op}$.) Our calculations in the proof of Proposition 4.6 imply that specifying a filtered colimit preserving functor $\mathbf{LEX}(\mathbf{C},\mathbf{Set}) \longrightarrow \mathbf{LEX}(\mathbf{D},\mathbf{Set})$ is equivalent to giving a functor $\mathbf{C}^{op} \times \mathbf{D} \longrightarrow \mathbf{Set}$ which is lex in its second variable (cf. 3.6(i)): we will call such a thing a *lex module from* **C** *to* **D**. Recall that a *module* (or *profunctor* or *distributeur*) from **C** to **D** is a functor $\mathbf{C}^{op} \times \mathbf{D} \longrightarrow \mathbf{Set}$; modules can be composed and small categories, modules and natural transformations form a bicategory—see [J1, section 2.4] and [CKW] for example. Restricting the objects to be lex categories and the morphisms to be lex modules, we obtain a sub-bicategory which we call **Lexmod**. Then it is the case that **ATOP** *and* **Lexmod** *are equivalent bicategories*. This forms the basis for a second equivalent description of the model in 4.19—one that has similarities with the "information system" approach in domain theory. The (opposites of) small lex categories are to lfp categories as information systems are to domains; lex modules are to filtered colimit preserving functors as "approximable maps" are to continuous maps between domains.

With a little calculation (which we do not give here) the structures in **Lexmod** which witness the fact that it is a model of the theory of constructions with "$Type \simeq ORDER$" take a rather pleasant form. First note that the terminal object **1** in **Lexmod** is given by the trivial lex category; and the product of **C** and **D** is given by their product in **Lex**. The class of display morphisms **A** in **Lexmod** corresponding under the equivalence to that in 4.19(ii), consists of those lex modules of the form $\mathbf{C}(P(-),+) : \mathbf{E}^{op} \times \mathbf{C} \longrightarrow \mathbf{Set}$ where $P : \mathbf{E} \longrightarrow \mathbf{C}$ is a lex fibration (cf. 4.11). Thus for each **C**, these are the objects of the category **A**(**C**) (notation as in 2.2); the morphisms from $P : \mathbf{E} \longrightarrow \mathbf{C}$ to $Q : \mathbf{F} \longrightarrow \mathbf{C}$ in **A**(**C**) are (isomorphism classes of) lex modules $M:\mathbf{E}^{op}\times\mathbf{F}\longrightarrow\mathbf{Set}$ for which the composition with $\mathbf{C}(Q(-),+)$ is isomorphic to $\mathbf{C}(P(-),+)$, that is, for which we have $M(-,Q_*(+))\cong\mathbf{E}(-,P_*(+))$.

We can use the remarks in 4.12 to describe the objects of $\mathbf{A}(\mathbf{C})$ equivalently as pseudofunctors $\mathbf{C}^{op}\longrightarrow\mathbf{Lex}$. Further calculation shows that from this point of view, a morphism in $\mathbf{A}(\mathbf{C})$ between pseudofunctors $\mathbf{E}:\mathbf{C}^{op}\longrightarrow\mathbf{Lex}$ and $\mathbf{F}:\mathbf{C}^{op}\longrightarrow\mathbf{Lex}$ is specified by the following data:

- a lex module $M_U(-,+):\mathbf{E}(U)^{op}\times\mathbf{F}(U)\longrightarrow\mathbf{Set}$ for each $U\in\mathbf{C}$,
- a natural transformation $M_\alpha:M_V(-,+)\longrightarrow M_U(\alpha^*(-),\alpha^*(+))$ for each $\alpha:U\longrightarrow V$ in $\mathbf{C}$,

satisfying the coherence conditions

$$M_{id_U}=M_U(\iota^{-1},\iota)\quad\text{and}\quad M_{\beta\alpha}=M_U(\kappa,\kappa^{-1})\circ(M_\alpha)_{\beta^*\times\beta^*}\circ M_\beta$$

(where ι and κ are the canonical isomorphisms for pseudofunctors defined in 3.7).

We next describe the adjoints $M_!$ and M_* to the pullback functor $M^*:\mathbf{A}(\mathbf{C})\longrightarrow\mathbf{A}(\mathbf{D})$ for a morphism $M:\mathbf{D}\longrightarrow\mathbf{C}$ in $\mathbf{A}$. First consider the special case when $\mathbf{C}$ is the terminal object $\mathbf{1}$. We can identify $\mathbf{A}(\mathbf{1})$ with **Lexmod** and M^* with the functor $\Delta_{\mathbf{D}}:\mathbf{Lexmod}\longrightarrow\mathbf{A}(\mathbf{D})$ which sends a lex category to the constant pseodofunctor with that value and acts similarly on modules. Then it is the case that

- *the left adjoint to* $\Delta_{\mathbf{D}}$ *sends a pseudofunctor* $\mathbf{E}:\mathbf{D}^{op}\longrightarrow\mathbf{Lex}$ *to the lex category* $Gr(\mathbf{E})$ *obtained by performing the* Grothendieck construction *(cf. 4.9) on* $\mathbf{E}$;
- *the right adjoint to* $\Delta_{\mathbf{D}}$ *sends a pseudofunctor* $\mathbf{E}:\mathbf{D}^{op}\longrightarrow\mathbf{Lex}$ *to the lex category* $\underrightarrow{oplax}_{\mathbf{D}^{op}}\mathbf{E}$ *obtained by taking the* oplax colimit *of* $\mathbf{E}$ *(cf. 3.7).*

The Beck-Chevalley condition implies that we can calculate the adjoints to an arbitrary morphism in $\mathbf{A}$ *fibrewise* using the above special case. Thus if $M:\mathbf{D}\longrightarrow\mathbf{C}$ in $\mathbf{A}$ is given as $\mathbf{C}(P(-),+)$ with $P:\mathbf{D}\longrightarrow\mathbf{C}$ a lex fibration, we have:

$$(M_!\mathbf{E})(U)=Gr(\mathbf{E}|_U)$$

and

$$(M_*)(U)=\underrightarrow{oplax}_{\mathbf{D}(U)^{op}}(\mathbf{E}|_U)$$

where $\mathbf{D}(U)$ is the fibre of P over $U\in\mathbf{C}$ (*cf.* 4.12) and $\mathbf{E}|_U$ is the restriction of $\mathbf{E}$ to that fibre.

The last part of the structure of **Lexmod** which needs describing is that corresponding to $\Sigma\longrightarrow\Upsilon$ in 4.19(*vi*), namely the interpretation of *Order* and the generic family of Orders. The lex category underlying Υ is the classifying category $\mathbf{C}_{\mathbf{lex}}$ of the lim theory **lex**, which we saw in 4.4 can be described equivalently as the opposite of the category of finitely presented lex categories and strict lex functors, $(\mathbf{lex}(\mathbf{Set})_{fp})^{op}$. The generic family in $\mathbf{A}(\mathbf{C}_{\mathbf{lex}})$, regarded as a pseudofunctor $\mathbf{lex}(\mathbf{Set})_{fp}\longrightarrow\mathbf{Lex}$, is just the forgetful functor (not an inclusion because the morphisms in $\mathbf{lex}(\mathbf{Set})$ are *strict* lex).

5 Localic algebraic toposes

We now describe a second topos-theoretic model of the theory of constructions—and moreover one in which the classes **A** and **R** required in 2.13 are distinct. The underlying category and the class **A** are as in 4.19, but we restrict **R** by imposing the condition of being localic. In terms of the alternative description 4.20(i), this corresponds to modelling (families of) Orders by (continuous fibrations of) locally finitely presentable categories, but modelling (families of) Types by (continuous fibrations of) *algebraic lattices.*

A geometric morphism f:**F**⟶**E** is *localic* if the terminal object $1 \in \mathbf{E}$ is an object of generators for **F** over **E**, i.e. if for every $Y \in \mathbf{F}$ there is some $X \in \mathbf{E}$ and a diagram of the form

$$(5.1) \qquad f^*(X) \xleftarrow{\;m\;} \cdot \xrightarrow{\;e\;} Y$$

in **F** with m a monomorphism and e an epimorphism. An alternative characterization, and the one which gives rise to the name "localic", is: f:**F**⟶**E** is localic iff **F** is equivalent in **GTOP(E)** to an **E**-topos of sheaves on an internal locale of **E**. We refer the reader to [J3] and [JT] for expositions of the basic properties of localic toposes. In particular we will need to use the fact that the localic geometric morphisms form one half of a factorization system on **GTOP**. The other half is given by the *hyperconnected* morphisms—those f:**F**⟶**E** for which f^* is faithful and such that the objects in the image of f^* are closed under taking subobjects in **F**. These morphisms are orthogonal to the localic morphisms (*cf*. 2.12). An arbitrary geometric morphism f:**F**⟶**E** factors uniquely up to equivalence as $f \cong \ell \circ h$:**F**⟶**L**⟶**E**, with h:**F**⟶**L** hyperconnected and ℓ:**L**⟶**E** localic. (**L** can be taken to be the **E**-topos of sheaves on the internal locale $f_*(\Omega_\mathbf{F})$, where $\Omega_\mathbf{F}$ is the subobject classifier in **F**; more elementarily, it is equivalent to the full subcategory of **F** whose objects are those Y for which there exists a diagram of the form (*5.1*) with m mono and e epi.)

5.1. Meet semilattices. These are the models of the lim theory **msl** having a single sort Ob, a constant symbol $\top{:}Ob$, a function symbol $\wedge{:}Ob{\times}Ob \longrightarrow Ob$ and axioms

- $\forall x,y,z{:}Ob(x\wedge(y\wedge z)=(x\wedge y)\wedge z)$
- $\forall x,y{:}Ob(x\wedge y=y\wedge x)$
- $\forall x{:}Ob(x\wedge\top=x)$
- $\forall x{:}Ob(x\wedge x=x)$.

Clearly a model of **msl** is a partially ordered set (via the relation: $x\le y$ iff $x\wedge y=x$) with all finite meets (including the empty one, $\top$)—and hence can be regarded as a lex category; similarly a homomorphism of meet semilattices is in particular a strict lex functor. Thus **msl(Set)** is a full subcategory of **lex(Set)**. Indeed it is a reflective subcategory: the left adjoint to the inclusion **msl(Set)**⟶**lex(Set)** sends a lex category **C** to the meet semilattice Po**C** obtained by first forming the pre-order reflection (i.e. the set Ob**C** pre-ordered via: $X\le Y$ iff **C**(X,Y) is inhabited) and then quotienting by the associated equivalence relation

($X \equiv Y$ iff $X \leq Y$ and $Y \leq X$) to get a poset.

Relativizing the above to any Grothendieck topos **E**, the *internal meet semilatices* in **E**, **msl**(**E**), form a full reflective subcategory of **lex**(**E**), the left adjoint $Po:$ **lex**(**E**)$\longrightarrow$**msl**(**E**) being given by the internal version of the poset reflection of a category. The unit of the adjunction at **C**$\in$**lex**(**E**), the quotient functor **C**$\longrightarrow$$Po$**C**, is full and surjective on objects; hence by [J3, Proposition 3.1(i)] the geometric morphism $[\mathbf{C}^{op},\mathbf{E}]\longrightarrow[(Po\mathbf{C})^{op},\mathbf{E}]$ it induces is hyperconnected. Similarly, since Po**C** is an internal poset, the unique internal functor $Po\mathbf{C}\longrightarrow\mathbf{1}$ is faithful and hence by [J3, Proposition 3.1(ii)], the induced geometric morphism $[(Po\mathbf{C})^{op},\mathbf{E}]\longrightarrow[\mathbf{1}^{op},\mathbf{E}]\simeq\mathbf{E}$ is localic. Therefore these two geometric morphisms give the hyperconnected-localic factorization of their composition, which is the geometric morphism $[\mathbf{C}^{op},\mathbf{E}]\longrightarrow\mathbf{E}$ defining the topos of internal presheaves as an **E**-topos. If the algebraic **E**-topos $[\mathbf{C}^{op},\mathbf{E}]$ is already known to be localic, then it is equivalent to its localic factorization and so $[\mathbf{C}^{op},\mathbf{E}]\simeq[(Po\mathbf{C})^{op},\mathbf{E}]$ in **GTOP**(**E**). We have thus proved:

5.2. Proposition. *A Grothendieck* **E**-*topos* $\mathbf{F}\longrightarrow\mathbf{E}$ *is both localic and algebraic iff it is equivalent in* **GTOP**(**E**) *to* $[M^{op},\mathbf{E}]\longrightarrow\mathbf{E}$ *for some internal meet semilattice* M.

□

In view of this result we can use the material in 4.4 on classifying toposes applied to the lim theory **msl** rather than to **lex** to obtain analogues of Propositions 4.2 and 4.5:

5.3. Proposition.

(i) *Geometric morphisms which are both localic and algebraic are stable under pullback in* **GTOP**.

(ii) *There is an algebraic topos* Υ' *and a localic-algebraic* Υ'*-topos* $\Sigma'\longrightarrow\Upsilon'$ *with the property that any other* $\mathbf{F}\longrightarrow\mathbf{E}$ *in* **GTOP** *which is both algebraic and localic can be obtained from* $\Sigma'\longrightarrow\Upsilon'$ *by pullback.*

Proof. For (i) we can either use the proof of 4.2 applied to internal meet semilattices or recall that localic morphisms are pullback stable—*cf.* [J3, Proposition 2.1] or [JT, Proposition VI.4].

For (ii), take $\Upsilon'=[\mathbf{C}^{op}_{\mathbf{msl}},\mathbf{Set}]$, the classifying topos of the lim theory **msl** and $\Sigma'=[U^{op}_{\mathbf{msl}},\Upsilon']$, where $U^{op}_{\mathbf{msl}}\in\mathbf{msl}(\Upsilon')$ is the generic meet semilattice—then argue as in the proof of 4.5.

□

5.4. Remark. In Proposition 4.5, the "generic" algebraic topos is defined over a topos Υ which is itself algebraic. On the other hand, in Proposition 5.3 the generic localic-algebraic topos is defined over a topos Υ' which is algebraic, but is not localic; for if it were, then $\mathbf{C}_{\mathbf{msl}}$ would have to be equivalent to $Po\mathbf{C}_{\mathbf{msl}}$ and hence be a preorder—but $\mathbf{C}_{\mathbf{msl}}$ is equivalent to the opposite of the category of finitely presented meet semilattices, which is certainly not preordered.

However, the property 5.3(ii) does not determine $\Sigma'\longrightarrow\Upsilon'$ uniquely. Perhaps there is another choice of localic-algebraic $\mathbf{\Sigma}''\longrightarrow\mathbf{\Upsilon}''$ satisfying 5.3(ii) but with $\mathbf{\Upsilon}''\longrightarrow\mathbf{Set}$ itself both algebraic and localic? In fact no such choice is possible. For if we had such, then we could find pullback squares in **GTOP** of the form

$$\begin{array}{ccc} \mathbf{\Sigma}' & \longrightarrow & \mathbf{\Sigma}'' \\ \downarrow & \lrcorner\ \cong & \downarrow \\ \mathbf{\Upsilon}' & \xrightarrow[i]{} & \mathbf{\Upsilon}'' \end{array} \qquad \text{and} \qquad \begin{array}{ccc} \mathbf{\Sigma}'' & \longrightarrow & \mathbf{\Sigma}' \\ \downarrow & \lrcorner\ \cong & \downarrow \\ \mathbf{\Upsilon}'' & \xrightarrow[r]{} & \mathbf{\Upsilon}' \end{array} .$$

Composing these two squares to give a single pullback and recalling the definition of $\mathbf{\Sigma}'$, we must have $(r\circ i)^*U_{\mathbf{msl}}\cong U_{\mathbf{msl}}$ in $\mathbf{msl}(\mathbf{\Upsilon}')$; but $\mathbf{\Upsilon}'$ is the classifying topos of meet semilattices and under the equivalence $\mathbf{msl}(\mathbf{\Upsilon}')\simeq\mathbf{GTOP}(\mathbf{\Upsilon}',\mathbf{\Upsilon}')$ this isomorphism corresponds to some isomorphism $r\circ i\cong 1_{\mathbf{\Upsilon}'}$. Therefore i and r make $\mathbf{\Upsilon}'$ a retract of $\mathbf{\Upsilon}''$ in **GTOP**—from which it follows easily that $\mathbf{\Upsilon}'$ is localic if $\mathbf{\Upsilon}''$ is. But we observed above that $\mathbf{\Upsilon}'$ is not localic; so no such $\mathbf{\Sigma}''\longrightarrow\mathbf{\Upsilon}''$ can exist.

5.5. Notation. For each Grothendieck topos **E**, let **LATOP(E)** denote the full sub-2-category of **GTOP(E)** whose objects are those $\mathbf{A}\longrightarrow\mathbf{E}$ which are both localic and algebraic. Given a geometric morphism $f:\mathbf{F}\longrightarrow\mathbf{E}$,

$$f^{\#}:\mathbf{LATOP(E)}\longrightarrow\mathbf{LATOP(F)}$$

will denote the restriction to localic-algebraic toposes of the operation of pulling back (which by Proposition 5.3(i), is well defined).

We next examine how the additional assumption of being localic effects the properties of algebraic toposes with respect to exponentiation and right adjoints to change of base. To do so, we use the following topos-theoretic result which as far as we know has not appeared elsewhere:

5.6. Proposition. *Let* $\mathbf{A}\longrightarrow\mathbf{E}$ *be an exponentiable* **E**-*topos. Then* $(\mathbf{A}\to_{\mathbf{E}} -):\mathbf{GTOP(E)}\longrightarrow\mathbf{GTOP(E)}$ *preserves localic morphisms.*

Proof. We make use of the following characterization of localic geometric morphisms, which is proved in [Pi1, Proposition 3.5] (for the case **E**=**Set**, but in a way that relativizes to arbitrary **E**):

Given $g:\mathbf{G}\longrightarrow\mathbf{F}$ in **GTOP(E)**, form the pullback of g against itself

$$\begin{array}{ccc} \mathbf{G}\times_{\mathbf{F}}\mathbf{G} & \xrightarrow{p_1} & \mathbf{G} \\ {\scriptstyle p_0}\downarrow & \lrcorner\ \overset{\pi}{\cong} & \downarrow{\scriptstyle g} \\ \mathbf{G} & \xrightarrow[g]{} & \mathbf{F} \end{array}$$

and let $d:\mathbf{G}\longrightarrow\mathbf{G}\times_{\mathbf{F}}\mathbf{G}$ be the *diagonal* geometric morphism, i.e. the morphism for which there are natural isomorphisms $\delta_i:p_i\circ d\cong 1_{\mathbf{G}}$ $(i=0,1)$ satisfying $g\delta_1\circ\pi d=g\delta_0$. Then: *$g$ is localic iff d is an inclusion.*

Since $(\mathbf{A}\to_{\mathbf{E}} -):\mathbf{GTOP(E)}\longrightarrow\mathbf{GTOP(E)}$ has a left biadjoint, it preserves the above pullback square and diagonal geometric morphism. Consequently to prove the proposition it is sufficient to prove that $(\mathbf{A}\to_{\mathbf{E}} -)$ *preserves geometric inclusions*. This is the case because inclusions can be characterized using limits in the bicategory **GTOP(E)** (and these are preserved by $(\mathbf{A}\to_{\mathbf{E}} -)$). Indeed, the geometric inclusions are precisely the *inverters* in **GTOP(E)**, i.e. those $i:\mathbf{F}\longrightarrow\mathbf{G}$ for which there are $g,h:\mathbf{G}\longrightarrow\mathbf{H}$ and $\phi:g\longrightarrow h$ with the property that for any **K**, the functor $i\circ(-):\mathbf{GTOP(E)(K,F)}\longrightarrow\mathbf{GTOP(E)(K,G)}$ is full and faithful with essential image those $k:\mathbf{K}\longrightarrow\mathbf{G}$ for which $\phi k:gk\longrightarrow hk$ is an isomorphism. We briefly indicate why this is the case, giving the argument for $\mathbf{E}=\mathbf{Set}$, but in a form admitting relativization:

First note that inverters can be constructed as sheaf subtoposes: if $(H_u \mid u\in U)$ is a small family of generators for **H** and j is the least Lawvere-Tierney topology on **G** which forces each $\phi_{H_u}:g^*(H_u)\longrightarrow h^*(H_u)$ to be iso, then $sh_j(\mathbf{G})\hookrightarrow\mathbf{G}$ is necessarily the inverter of ϕ. Conversely, if $i:\mathbf{F}\longrightarrow\mathbf{G}$ is an inclusion—say $\mathbf{F}=sh_j(\mathbf{G})$—then define **H** to be the full subcategory of the arrow category $\mathbf{G}^2$ whose objects are those $a:X\longrightarrow Y$ in **G** which are j-bidense. One can show that **H** is a Grothendieck topos. (This is just a question of exhibiting generators, since it has the right exactness properties automatically.) Then the inclusion $\mathbf{H}\hookrightarrow\mathbf{G}^2$ is the inverse image part of a geometric surjection $q:\mathbf{G}^2\longrightarrow\mathbf{H}$. The two functors and non-identity natural transformation in **Cat(1,2)** induce geometric morphisms $l_0,l_1:\mathbf{G}\longrightarrow\mathbf{G}^2$ and a natural transformation $\lambda:l_0\longrightarrow l_1$. Then the definition of **H** and the characterization [J1, 3.4] of sheaf subtoposes in terms of categories of fractions imply that $sh_j(\mathbf{G})\hookrightarrow\mathbf{G}$ is the inverter of $q\lambda:ql_0\longrightarrow ql_1$.

□

5.7. Corollary.

(*i*) *If* $a:\mathbf{A}\longrightarrow\mathbf{E}$ *is an algebraic* **E***-topos and* $b:\mathbf{B}\longrightarrow\mathbf{E}$ *a localic-algebraic* **E***-topos, then their exponential* $(\mathbf{A}\to_{\mathbf{E}}\mathbf{B})\longrightarrow\mathbf{E}$ *is localic-algebraic. So in particular each* **LATOP(E)** *is cartesian closed and for each geometric morphism* $f:\mathbf{F}\longrightarrow\mathbf{E}$*, the pullback operation* $f^{\#}:\mathbf{LATOP(E)}\longrightarrow\mathbf{LATOP(E)}$ *preserves the cartesian closed structure.*

(*ii*) *If* $f:\mathbf{F}\longrightarrow\mathbf{E}$ *is algebraic and* $b:\mathbf{B}\longrightarrow\mathbf{F}$ *is a localic-algebraic* **F***-topos, then the algebraic* **E***-topos* $f_{\#}(\mathbf{B})$ *of Proposition 4.18 is also localic. So when* f *is algebraic,* $f_{\#}$ *gives a right adjoint for* $f^{\#}:\mathbf{LATOP(E)}\longrightarrow\mathbf{LATOP(E)}$ *satisfying the Beck-Chevalley condition.*

Proof. In view of Corollary 4.8, for (*i*) we just need to see that $(\mathbf{A}\to_{\mathbf{E}}\mathbf{B})\longrightarrow\mathbf{E}$ is localic; but this morphism is the composition of the equivalence $(\mathbf{A}\to_{\mathbf{E}}\mathbf{E})\simeq\mathbf{E}$ with $(\mathbf{A}\to_{\mathbf{E}} b):(\mathbf{A}\to_{\mathbf{E}}\mathbf{B})\longrightarrow(\mathbf{A}\to_{\mathbf{E}}\mathbf{E})$, which is localic by Proposition 5.6.

For (*ii*), note that the construction of $f_{\#}(\mathbf{B})$ in 4.18 is in terms of exponentiation and pullbacks—so that the result follows from 5.6 and the fact that localic morphisms are pullback stable.

□

5.8. Remark. Using Proposition 5.2 and the formula for exponentials of algebraic toposes given in 4.7, we can rephrase 5.7(*i*) in more concrete terms: if $\mathbf{C}\in\mathbf{lex(E)}$ and $M\in\mathbf{msl(E)}$, then the copower $\mathbf{C}^{op}\cdot M\in\mathbf{lex(E)}$ is equivalent to an internal meet semilattice and hence is an internal preorder.

We now turn to the construction of left adjoints to pulling back localic-algebraic toposes. In contrast to the case for exponentials and right adjoints (for which we have the absoluteness result 2.8), the left adjoints are not the same as in the algebraic case:

5.9. Proposition. *If* f:**F**⟶**E** *is an algebraic* **E**-*topos, then* $f^{\#}$:**LATOP(E)**⟶**LATOP(F)** *possesses a left adjoint, denoted* $f_!$:**LATOP(F)**⟶**LATOP(E)**, *which satisfies the Beck-Chevalley condition* (*cf*. 4.10).

Proof. Let **LTOP(E)** denote the full sub-2-category of **GTOP(E)** whose objects are localic **E**-toposes. (The morphisms of **LTOP(E)** are also localic since **B**⟶**A** is localic when both **A**⟶**E** and **B**⟶**A**⟶**E** are.) For an arbitrary geometric morphism f:**F**⟶**E**, the hyperconnected-localic factorization of geometric morphisms mentioned at the beginning of this section provides a left adjoint $f_!$:**LTOP(F)**⟶**LTOP(E)** to the pullback operation $f^{\#}$:**LTOP(E)**⟶**LTOP(F)**. Indeed, given b:**B**⟶**F** in **LTOP(F)**, $f_!$(**B**)⟶**E** is the localic factorization of the composition $f \circ b$:**B**⟶**E**. These left adjoints satisfy the Beck-Chevalley condition for pullback squares in **GTOP** simply because both localic and hyperconnected morphisms are stable under pullback: see [JT, VI.5] and [J3, section 2].

Therefore the proposition will be proved if we can show that $f_!$:**LTOP(F)**⟶**LTOP(E)** takes algebraic toposes to algebraic toposes in the case that f is itself algebraic. In view of 4.9, this amounts to showing that the localic part of an algebraic **E**-topos is again algebraic. But we saw above that for **C** ∈ **lex**(**E**), the localic factorization of [$\mathbf{C}^{op}$,**E**]⟶**E** is [$(Po\mathbf{C})^{op}$,**E**]⟶**E**, where Po**C** ∈ **msl**(**E**) is the poset reflection of **C**.

□

5.10. Localic-algebraic model of the theory of constructions. We now organize the results of this section to give a second topos-theoretic model of the theory of constructions. It is a model of the theory "$Order \in ORDER$" of 1.11 (but not of the stronger theory "$Type \simeq ORDER$" of 1.12). So we will show how to fulfil the conditions in 2.13 and the condition in 2.14.

For (i), we take the category **B** to be just as in 4.19(i): so **B** consists of algebraic toposes and isomorphism classes of geometric morphisms.

For (ii), we take **A** as in 4.19(ii): so it is determined by those geometric morphisms f:**F**⟶**E** making **F** an algebraic **E**-topos; and for the subclass **R** we take those morphisms determined by geometric morphisms which are not only algebraic but also localic.

(iii) follows from 4.19(iii) and 5.3(i).

(iv) holds because of the fact that equivalences between toposes are trivially localic-algebraic.

(v) holds because of 4.19(iv) and the fact (easily deduced from the definition at the beginning of this section) that the composition of localic geometric morphisms is again localic.

(vi) holds because of 4.19(v) and 5.7(ii).

(vii) follows as in 5.9: the hyperconnected-localic factorization is pullback stable, hyperconnected morphisms are orthogonal to localic ones and the factorization applied to an algebraic geometric morphism yeilds a localic-algebraic morphism.

$(viii)$ is a consequence of 5.3(ii).

(ix) holds by definition of **B**.

Finally, the condition ***Order*** $\in$ ***ORDER*** of 2.14 is just 4.19(vi).

5.11. Equivalent descriptions of the model. In 4.20 we gave two equivalent forms of the category **B**. The first was in terms of locally finitely presentable categories and functors preserving filtered colimits. The second was in terms of small lex categories and lex modules. We explain briefly what the class of morphisms **R** looks like in these equivalent formulations.

Recall from 4.20(i) that a filtered colimit preserving functor $P:\mathbf{F}\longrightarrow\mathbf{E}$ between lfp categories is in the class **A** if it preserves limits (this is equivalent to its having a left adjoint) and has a right-adjoint-right-inverse which also preserves filtered colimits. In particular P is a fibration with lfp fibres; and then P is in **R** simply if its fibres are in fact pre-ordered. Note that pre-ordered lfp categories are equivalent to algebraic lattices. Thus the category **R**(**1**), whose objects model the constant Types, is equivalent to the category of algebraic lattices and continuous (i.e. directed sup preserving) maps. More generally, one can show that for **E** locally finitely presentable, **R**(**E**) is equivalent to a category whose objects are filtered colimit preserving functors from **E** into the lfp category of meet semilattices, **msl**(**Set**). The latter category is equivalent to the category of algebraic lattices and continuous maps possessing continuous right adjoints—so each object of **R**(**E**) in particular determines a functor into algebraic lattices and continuous maps; then the morphisms in **R**(**E**) are given by "lax natural" families of continuous maps. This is the form in which Coquand, Gunter and Winskel have studied the model—see [CE, section 5].

Turning to the formulation of the model in terms of lex categories and lex modules, the modules $\mathbf{D}\longrightarrow\mathbf{C}$ which are in **R** are those induced by lex fibrations $P:\mathbf{D}\longrightarrow\mathbf{C}$ whose fibres are pre-ordered and hence are equivalent to meet semi-lattices. Thus we can take the objects of **R**(**C**) to be those pseudofunctors $\mathbf{C}^{op}\longrightarrow\mathbf{Lex}$ which are in fact functors $\mathbf{C}^{op}\longrightarrow\mathbf{msl(Set)}$. Restricting the description in 4.20(ii) of the morphisms of **A**(**C**) to these objects, we find that specifying a morphism from $\mathrm{E}:\mathbf{C}^{op}\longrightarrow\mathbf{msl(Set)}$ to $\mathrm{F}:\mathbf{C}^{op}\longrightarrow\mathbf{msl(Set)}$ in **R**(**C**) amounts to giving a family of *relations* $M_U\subseteq \mathrm{E}(U)\times\mathrm{F}(U)$ $(U\in\mathbf{C})$ satisfying:

- $x'\leq x$ and $M_U(x,y)$ and $y\leq y' \Rightarrow M_U(x',y')$
- $M_U(x,\top)$
- $M_U(x,y)$ and $M_U(x,y') \Rightarrow M_U(x,y\wedge y')$
- $\alpha:U\longrightarrow V$ and $M_V(x,y) \Rightarrow M_U(\alpha^*(x),\alpha^*(y))$.

(One can use relations, since if $M:\mathbf{C}^{op}\times\mathbf{D}\longrightarrow\mathbf{Set}$ is a lex module and $\mathbf{D}$ is a meet semilattice, then $y\leq\top$ induces a monomorphism $M(X,y)\rightarrowtail M(X,\top)\cong 1$, so that each $M(X,y)$ has at most one element.) Describing the categories $\mathbf{R}(\mathbf{C})$ in this way also makes it easy to describe the reflection of $\mathbf{R}$ into $\mathbf{A}$ (i.e. 5.10(*vii*)): it is given on objects of $\mathbf{A}(\mathbf{C})$ by composing a pseudofunctor $\mathbf{C}^{op}\longrightarrow\mathbf{Lex}$ with the poset reflection functor $Po:\mathbf{Lex}\longrightarrow\mathbf{msl}(\mathbf{Set})$ of 5.1. Finally, *Type* is modelled in this setting simply by the opposite of the category of *finite* meet semilattices (since finitely presentable implies finite in this case); and the generic family of Types is modelled by the inclusion of this category into **msl(Set)**.

REFERENCES

[B1] J.Bénabou, *Théories relatives à un corpus,* C. R. Acad. Sc. Paris, Série A 281(1975) 831-834.

[B2] J.Bénabou, *Fibred categories and the foundations of naive category theory,* J. Symbolic Logic 50(1985) 10-37.

[B3] J.Bénabou, *Introduction to bicategories,* Lecture Notes in Math. No.47 (Springer-Verlag, Berlin - Heidelberg, 1967), pp1-77.

[BW] M.Barr and C.Wells, *Toposes, Triples and Theories,* Grundlehren der mathematischen Wissenschaften 278 (Springer-Verlag, New York, 1985).

[CKW] A.Carboni, S.Kasangian and R.F.C.Walters, *An axiomatics for bicategories of modules,* J. Pure Appl. Algebra 45(1987) 127-141.

[Cd] L.Cardelli, *A polymorphic λ-calculus with Type:Type,* SRC Report (DEC, Palo Alto, 1986).

[Ca] J.Cartmell, *Generalised algebraic theories and contextual categories,* Annals Pure Appl. Logic 32(1986) 209-243.

[C] A.Church, *A formulation of the simple theory of types,* J. Symbolic Logic 5(1940) 56-68.

[CE] T.Coquand and T.Ehrhard, *An equational presentation of higher order logic.* In: D.H.Pitt *et al* (eds), *Category Theory and Computer Science,* Lecture Notes in Computer Science No.283 (Springer-Verlag, Berlin - Heidelberg, 1987), pp40-56.

[CGW] T.Coquand, C.Gunter and G.Winskel, *Polymorphism and domain equations,* Computer Laboratory Technical Report No.107 (Cambridge University, 1987).

[CH] T.Coquand and G.Huet, *The Calculus of Constructions,* Information and Computation 76(1988) 95-120.

[Coq1] T.Coquand, *Une théorie des constructions,* Thèse de troisième cycle, Univ. Paris VII, 1985.

[Coq2] T.Coquand, *An analysis of Girard's paradox.* In: *Proc 1st Annual Symp. Logic in Computer Science* (IEEE Computer Society Press, Washington, 1986), pp227-236.

[Co1] M.Coste, *Localization, spectra and sheaf representation.* In: M.P.Fourman *et al* (eds), *Applications of Sheaves,* Lecture Notes in Math. No.753 (Springer-Verlag, Berlin - Heidelberg, 1979) pp212-238.

[Co2] M.Coste, *Localisation dans les catégories de modèles.* Thesis, Univ. Paris-Nord, 1977.

[G1] J.-Y.Girard, *Intérpretation fonctionelle et élimination des coupures de l'arithmétique* *d'ordre supérieur.* Thèse de doctorat d'état, Univ. Paris VII, 1972.

[G2] J.-Y.Girard, *The system F of variable types, fifteen years later,* Theoretical Computer Science 45(1986) 159-192.

[GU] P.Gabriel and F.Ulmer, *Lokal präsentierbare Kategorien,* Lecture Notes in Math. No.221 (Springer-Verlag, Berlin - Heidelberg, 1971).

[Gr] J.W.Gray, *Fibred and cofibred categories.* In: S.Eilenberg *et al* (eds), *Proc. LaJolla Conference on Categorical Algebra* (Springer-Verlag, New York, 1966) pp21-83.

[HHP] R.Harper, F.Honsell and G.Plotkin, *A framework for defining logics.* In: *Proc. 2nd Annual Symp. Logic in Computer Science,* (IEEE Computer Society Press, Washington, 1987), pp194-204.

[HRR] J.M.E.Hyland, E.P.Robinson and G.Rosolini, *The discrete objects in the effective topos,* preprint, Cambridge, 1987.

[J1] P.T.Johnstone, *Topos Theory,* L.M.S. Monographs No.10 (Academic Press, London, 1977).

[J2] P.T.Johnstone, *Injective toposes.* In: B.Banaschewski and R.-E.Hoffmann (eds), *Continuous Lattices,* Lecture Notes in Math. No.871 (Springer-Verlag, Berlin - Heidelberg, 1981), pp284-297.

[J3] P.T.Johnstone, *Factorization theorems for geometric morphisms, I,* Cahiers Top. et Géom. Diff. 22(1981) 3-17.

[JJ] P.T.Johnstone and A.Joyal, *Continuous categories and exponentiable toposes,* J. Pure Appl. Algebra 25(1982) 255-296.

[JT] A.Joyal and M.Tierney, *An extension of the Galois theory of Grothendieck,* Mem. Amer. Math. Soc. 51(1984), No.309.

[La] F.Lamarche, *A simple model of the theory of constructions.* In this volume.

[LS] J.Lambek and P.J.Scott, *Introduction to higher order categorical logic,* Cambridge Studies in Advanced Math. 7 (Cambridge University Press, 1986).

[M-L1] P.Martin-Löf, *Constructive mathematics and computer programming.* In: L.J.Cohen *et al* (eds), *Sixth International Congress for Logic Methodology and Philosophy of Science* (North-Holland, Amsterdam, 1982), pp153-175.

[M-L2] P.Martin-Löf, *Intuitionistic Type Theory. (Notes by G.Sambin of a series of lectures given in Padua, June 1980)* (Bibliopolis, Napoli, 1984).

[MiPl] J.C.Mitchell and G.D.Plotkin, *Abstract types have existential type.* In: *Proc. 12th Annual Symp. Principles of Programming Languages,* (ACM, New York, 1985) pp237-51.

[MP] M.Makkai and A.M.Pitts, *Some results on locally finitely presentable categories,* Trans Amer. Math. Soc. 299(1987) 473-496.

[MR] M.Makkai and G.E.Reyes, *First Order Categorical Logic,* Lecture Notes in Math. No.611 (Springer-Verlag, Berlin - Heidelberg, 1977).

[Ob] A.Obtułowicz, *Categorical and algebraic aspects of Martin-Löf type theory,* preprint, 1987.

[Pi1] A.M.Pitts, *Conceptual completeness for first order intuitionistic logic: an application of categorical logic,* Annals Pure Appl. Logic, *to appear.*

[Pi2] A.M.Pitts, *Polymorphism is set theoretic, constructively.* In: D.H.Pitt *et al* (eds), *Category Theory and Computer Science,* Lecture Notes in Computer Science No.283 (Springer-Verlag, Berlin - Heidelberg, 1987), pp12-39.

[Po] A.Poigné, *Tutorial on algebra categorically.* In: D.H.Pitt *et al* (eds), *Category Theory and Computer Programming,* Lecture Notes in Computer Science No.240 (Springer-Verlag, Berlin - Heidelberg, 1986) pp76-102.

[PS] R.Paré and D.Schumacher, *Abstract families and the adjoint functor theorems.* In:

P.T.Johnstone and R.Paré (eds), *Indexed Categories and their Applications*, Lecture Notes in Math. No.661 (Springer-Verlag, Berlin - New York, 1978), pp1-125.

[RW1] R.Rosebrugh and R.J.Wood, *Cofibrations in the bicategory of topoi*, J. Pure Appl. Algebra 32(1984) 71-94.

[RW2] R.Rosebrugh and R.J.Wood, *Cofibrations II: left exact right actions and composition of gamuts*, J. Pure Appl. Algebra 39(1986) 283-300.

[Se1] R.A.G.Seely, *Locally cartesian closed categories and type theory*, Math. Proc. Camb. Phil. Soc. 95(1984) 33-48.

[Se2] R.A.G.Seely, *Categorical models for higher order polymorphic lambda calculus*, J. Symbolic Logic 52(1987) 969-989.

[Sc] D.S.Scott, *Data types as lattices*, SIAM J. Computing 5(1976) 522-587.

[S] R.Street, *Fibrations in bicategories*, Cahiers Top. et Géom. Diff. 21(1980) 111-160 and *(corrections)* 28(1987) 53-56.

[Ta] P. Taylor, *Recursive domains, indexed category theory and polymorphism*, Thesis, University of Cambridge, 1986.

[Tr] A.S.Troelstra, *On the syntax of Martin-Löf's theories*, Theoretical Computer Science 51(1987) 1-26.

DEPARTMENT OF PURE MATHEMATICS,
UNIVERSITY OF CAMBRIDGE,
CAMBRIDGE CB2 1SB, ENGLAND

and

MATHEMATICS DIVISION,
UNIVERSITY OF SUSSEX,
BRIGHTON BN1 9QH, ENGLAND.