## §1. Logic with dependent sorts

First, we describe the kinds of structure which the assertions of logic with dependent sorts are about.

It is well-known from categorical logic that the similarity types that are graphs (having sorts the objects, and unary sorted operation symbols only) are sufficient for all purposes. The simplest consideration here replaces a relation-symbol sorted as $R \subset A_1 \times \ldots \times A_n$ by a new sort $R$, and operations $R \xrightarrow{p_i} A_i$, $i=1, \ldots, n$. Our first move is to restrict attention to *one-way graphs*; in fact, more conveniently, to *one-way categories*.

The concept of one-way category is due to F. W. Lawvere [L]. In [M1], I reproduce Lawvere's observation to the effect that categories of finite sketches obtained by the repeated use of the second construction of [M1] starting from $\mathtt{Set}$ are exactly the ones of the form $\mathtt{Set}^{\boldsymbol{C}}$, with $\boldsymbol{C}$ a finite one-way category.

A one-way category is one in which all endomorphisms are trivial (identities). In a skeletal one-way category, for any objects $A$ and $B$, it is not possible that there are *proper* (non-identity) arrows in both direction $A \to B$ and $B \to A$. As a consequence, there are no cycles (positive-length paths $A_0 \to A_1 \to \ldots \to A_n$ of proper arrows with $A_0 = A_n$).

We are mainly interested in finite, skeletal, one-way categories. However, for certain purposes, we need to relax the finiteness condition.

A category $\boldsymbol{C}$ has *finite fan-out* (I owe this concept to Jim Otto) if for every object $A$, there are altogether finitely many arrows with domain $A$; the set $\bigsqcup\{\boldsymbol{C}(A, C) : C \in \mathtt{Ob}(\boldsymbol{C})\}$ is finite. A *simple* category is one which is one-way, skeletal, and has finite fan-out.

A simple category is reverse-well-founded; in other words, it satisfies the ascending chain condition: there are no infinite paths $A_0 \to A_1 \to \ldots \to A_n \to A_{n+1} \to \ldots$ ($n<\omega$) consisting of proper arrows. (Namely, any such would have to have the objects $A_n$ pairwise distinct, by the above, and that would mean, *a fortiori*, infinitely many arrows out of $A_0$.)

If $\boldsymbol{L}$ is a simple category, the set $\mathtt{Ob}(\boldsymbol{L})$ of objects is partitioned as in

$$\mathrm{Ob}(\mathbf{L}) = \dot{\bigcup_{i<\ell}} \mathbf{L}_i$$

into non-empty *levels* $\mathbf{L}_i$, for $i<\ell$, $\ell$ the *height* of $\mathbf{L}$, $\ell\leq\omega$, such that $\mathbf{L}_0$ consist of the objects $A$ for which there is no proper arrow with domain $A$, and such that, for $i>0$, $\mathbf{L}_i$ consists of those objects $A$ for which all proper arrows $A\to B$ have $B\in\mathbf{L}_{<i} = \dot{\bigcup_{j<i}} \mathbf{L}_j$, and there is at least one arrow $A\to B$ with $B\in\mathbf{L}_{i-1}$. (If $A\in\mathrm{Ob}(\mathbf{L})$, and for all proper $f\colon A\to B$, $B\in \dot{\bigcup_{i<\omega}} \mathbf{L}_i$, then $A\in \dot{\bigcup_{i<\omega}} \mathbf{L}_i$; in fact $A\in\mathbf{L}_i$ for some $i$ not greater than the maximum of the levels of the codomains of the finitely many proper arrows with domain $A$ plus one. Therefore, if $A\in\mathrm{Ob}(\mathbf{L}) - \dot{\bigcup_{i<\omega}} \mathbf{L}_i$, then there is a proper $A\to B$ with $B\in\mathrm{Ob}(\mathbf{L}) - \dot{\bigcup_{i<\omega}} \mathbf{L}_i$, and thus there is an infinite proper path out of $A$.) All proper arrows go from a level to a lower level. Of course, the height of a finite simple category is finite.

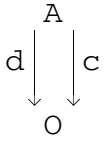
A *maximal* object in a simple category is one which is not the codomain of a proper arrow. Every object of the maximal level (if any) is maximal, but not necessarily conversely.

By a *vocabulary for logic with dependent sorts*, or *DS vocabulary*, or even DSV, we mean a simple category given with a distinguished, but otherwise arbitrary (possibly empty) set of maximal objects. The distinguished maximal objects of the DSV are its *relation symbols* (or *relations*); the rest of its objects are its *kinds*. We write $\mathrm{Rel}(\mathbf{L})$ and $\mathrm{Kind}(\mathbf{L})$ for the sets of relations and of kinds of $\mathbf{L}$, respectively.

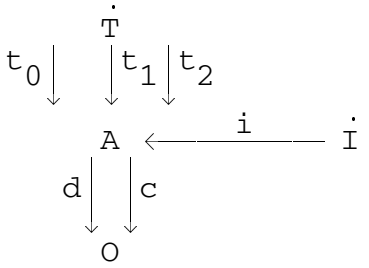
DS vocabularies are our similarity types for structures for logic with dependent sorts; concomitantly, they figure as vocabularies for the syntax of logic with dependent sorts. Unlike in multisorted logic, the arrows of a DSV do not enter the syntax of FOLDS as operation-symbols; the role of the arrows in a DSV and their composition will serve to determine the "dependence structure" of the variables.

Here are some examples for DSV's .

$\mathbf{L}_{\mathrm{graph}}$ :



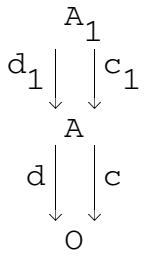
$\mathbf{L}_{\mathtt{cat}}$ :




$dt_1=ct_0$ , $dt_2=ct_1$ ,
$dt_2=dt_0$ , $di=ci$ .

relations: I , T

$\mathbf{L}_{2\text{-}\mathtt{graph}}$ :




$dd_1=dc_1$ , $cd_1=cc_1$ .

Only non-identity arrows are shown. The proper arrows are those shown and their composites, among which we have the equalities shown, and no more. *E.g.*, there are three distinct arrows $\mathtt{T}\to\mathtt{O}$ . $\mathbf{L}_{\mathtt{graph}}$ and $\mathbf{L}_{2\text{-}\mathtt{graph}}$ have no relations. The dots in $\mathbf{L}_{\mathtt{cat}}$ signify that I and T are relations.

For a DSV $\boldsymbol{L}$ , and an object $A$ in it, we write $A|\boldsymbol{L}$ for the set of *proper* arrows with domain $A$ (the notation resembles the notation $A{\downarrow}\boldsymbol{L}$ for the comma category). For an arrow $p$ , $\mathtt{K}_p$ denotes its codomain.

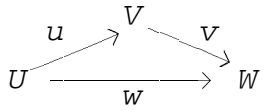
Given a DSV $\boldsymbol{L}$ , the intended structures for $\boldsymbol{L}$ , the $\boldsymbol{L}$*-structures*, are the functors $M:\boldsymbol{L}\to\mathtt{Set}$ in which for each relation $R\in\mathtt{Rel}(\boldsymbol{L})$ the following holds: the family $\langle M(p):M(R)\to M(\mathtt{K}_p)\rangle_{p\in R|\boldsymbol{L}}$ of functions, indexed by the proper arrows in $\boldsymbol{L}$ with domain $R$ , is jointly monomorphic: for $a, b\in M(R)$ , if $M(p)(a)=M(p)(b)$ for all $p\in R|\boldsymbol{L}$ , then $a=b$ . The condition means that $M(R)$ is essentially a subset of the set $\prod_{p\in R|\boldsymbol{L}} M(\mathtt{K}_p)$ , actually a subset of $M[R]$ ; here, for any $A\in\mathtt{Ob}(\boldsymbol{L})$ ,

$$M[A] \stackrel{\mathtt{def}}{=} \{\langle a_p\rangle_p \in \prod_{p\in A|\boldsymbol{L}} M(\mathtt{K}_p) : M(q)(a_p)=a_{p'} \text{, whenever } qp=p'\} \qquad (1)$$

( $M[A]$ is the limit (joint pullback) of the diagram $A{\downarrow}(\boldsymbol{L}-\{1_A\}) \xrightarrow{\Phi} \boldsymbol{L} \xrightarrow{M} \mathtt{Set}$ (with $\Phi$ the forgetful functor) mapping $(A\to K)$ to $M(K)$ ). We will usually (and without loss of generality) assume that in case $R\in\mathtt{Rel}(\boldsymbol{L})$ , the canonical monomorphism

$m^{M}_{R}: M(R) \to M[R]$ taking $a$ to $\langle (Mp)(a) \rangle_{p \in R \mid K_p}$ is an *inclusion* of sets.

We recognize that the $L_{graph}$-structures are the graphs, the $L_{2\text{-}graph}$-structures are the 2-graphs. Categories are particular $L_{cat}$-structures. If $M$ is a category, for $M$ as an $L_{cat}$-structure, $M(O)$ , $M(A)$ are the sets of objects and of arrows, $M(d)$ and $M(c)$ are the domain and codomain functions; as a consequence, $M[T]$ is the set of triangles




in $M$; by definition, $M(T)$ is the set of *commutative* triangles, a subset of $M[T]$ ; $M(I)$ is the set of identity arrows. In fact, we realize that the $L_{cat}$-structures are exactly the *category-sketches* of [M1] .

For $\mathbf{L}$ a DSV , $|\mathbf{L}|$ denotes its underlying graph. Any (small) graph $L$ can be used as a similarity type for multisorted logic; the $L$-structures are the graph-maps (diagrams) $L \to Set$ ; $\mathbf{C}$-valued $L$-structures are the diagrams $L \to \mathbf{C}$ . Multisorted first-order logic with $L$ as vocabulary uses the objects of $L$ as sorts and the arrows of $L$ as sorted unary operation symbols; we always allow equality (to be interpreted in the *standard* way) when we refer to multisorted logic. For these matters, see [MR1]. First-order logic with dependent sorts over $\mathbf{L}$ will be a *proper part* of multisorted first order logic over $|\mathbf{L}|$ .

To be sure, the $|\mathbf{L}|$ -structures are not exactly the $\mathbf{L}$-structures; the latter are those among the former that satisfy a certain set $\Sigma[\mathbf{L}]$ of axioms over $|\mathbf{L}|$ , to be described next. $\Sigma[\mathbf{L}]$ consists of the following sentences:

$$\forall x \in A.\, ( \bigwedge \{ q(p(x)) = p'(x) \,:\, p, p' \in A \mid \mathbf{L},\ q \in Arr(\mathbf{L}) ,\ qp = p' \} ,$$

one for each $A \in Ob(\mathbf{L})$ ( $= Kind(\mathbf{L}) \cup Rel(\mathbf{L})$ ) ; and

$$\forall x \in R.\, \forall y \in R.\, [ ( \bigwedge_{p \in R \mid \mathbf{L}} p(x) = p(y) ) \longrightarrow x = y ] ,$$

one for each $R \in Rel(\mathbf{L})$ .

One feature of logic with dependent sorts is that there will not be any operation symbols (explicitly) used in it; thus, the just-listed sentences are definitely not in logic with dependent sorts over $\mathbf{L}$.

Let us explain the intuition behind logic with dependent sorts for the case when the vocabulary is $\mathbf{L}_{\mathtt{cat}}$. First of all, logic with dependent sorts is a (proper) *part* of what we know as ordinary multisorted logic over $|\mathbf{L}_{\mathtt{cat}}|$, the (a) language of categories. In logic with dependent sorts over $\mathbf{L}_{\mathtt{cat}}$, we have variables ranging over $\mathtt{O}$; we can quantify these variables. However, instead of variables ranging over $\mathtt{A}$, we will have ones that range over $\mathtt{A}(U, V)$, where $U$ and $V$ are variables of sort $\mathtt{O}$. $\mathtt{A}(U, V)$ is a "dependent sort", one depending on the variables $U$ and $V$. A variable $u$ ranging over $\mathtt{A}(U, V)$ is *of* sort $\mathtt{A}(U, V)$, and we write $u\mathtt{:A}(U, V)$. Of course, we should think of $\mathtt{A}(U, V)$ as $\mathtt{hom}(U, V)$, and of $u\mathtt{:A}(U, V)$ as $u\mathtt{:}U\to V$. In terms of the semantics of $\mathtt{L}_{\mathtt{cat}}$-structures, the interpretation of $\mathtt{A}(U, V)$ in $M$ is $\{a\in M\mathtt{A:}(M\mathtt{d})(a)=(M\mathtt{c})(a)\}$. Thus, we have no variables ranging over all arrows at once; only ones ranging over arrows with a *fixed* domain and codomain.

An immediate consequence of this is that if a formula has the free variables $U$ and $V$, and also $u\mathtt{:}U\to V$ (that is, $u\mathtt{:A}(U, V)$), then forming $\forall U\varphi$ should and will be illegal; the free variable $u$ in $\forall U\varphi$ has lost its fixed reference to a domain.

In FOLDS in general, and in particular over $\mathbf{L}_{\mathtt{cat}}$, we will have a restricted use of equality only. The reason for this is our main aim, which is to formulate languages for categorical structures in which all statements are invariant under the equivalence appropriate for the kind of categorical structure at hand. Typically, equivalences does not respect equality of certain kinds of entities; in the case of categories, equality of objects, in the case of bicategories, equality of objects (0-cells) and equality of 1-cells. In FOLDS with restricted equality, we will allow "fiberwise equality" over maximal kinds; in the case of $\mathbf{L}_{\mathtt{cat}}$, this means fiberwise equality over $\mathtt{A}$. The restrictions on equality in FOLDS over $\mathbf{L}_{\mathtt{cat}}$ will correspond to the intuition that in category theory, one should not refer to equality of objects, and equality on arrows should be mentioned only with reference to arrows which have the same domain and the same codomain.

The above remarks, made for the case $\mathbf{L} = \mathbf{L}_{\mathtt{cat}}$, on how logic with dependent sorts over $\mathbf{L}$ is constrained with respect to ordinary first-order multi-sorted logic over $|\mathbf{L}|$ have natural extensions to the case of a general vocabulary $\mathbf{L}$. The constraints will be built into the general

definition of the syntax.

Before giving the general definitions, to illustrate FOLDS (first-order logic with dependent sorts), we write down the axioms for category in this logic.

$$\forall U{:}\mathtt{O}.\exists i{:}U\to U.\mathtt{I}(i)\ ;$$
$$\forall U{:}\mathtt{O}.\forall i{:}U\to U.\forall j{:}U\to U.(\mathtt{I}(i)\wedge\mathtt{I}(j)\to i{=}j)\ ;$$
$$\forall U{:}\mathtt{O}.\forall V{:}\mathtt{O}.\forall W{:}\mathtt{O}.\forall u{:}U\to V.\forall v{:}V\to W.\exists w{:}U\to W.\mathtt{T}(u,v,w)\ ;$$
$$\forall U{:}\mathtt{O}.\forall V{:}\mathtt{O}.\forall W{:}\mathtt{O}.\forall u{:}U\to V.\forall v{:}V\to W.\forall w{:}U\to W.\forall w'{:}U\to W$$
$$(\mathtt{T}(u,v,w)\wedge\mathtt{T}(u,v,w')\longrightarrow w{=}w')\ ;$$
$$\forall U{:}\mathtt{O}.\forall i{:}U\to U.\forall u{:}U\to V.\mathtt{T}(i,u,u)\ ;$$
$$\forall U{:}\mathtt{O}.\forall i{:}U\to U.\forall u{:}V\to U.\mathtt{T}(u,i,u)\ ;$$
$$\forall U{:}\mathtt{O}.\forall V{:}\mathtt{O}.\forall W{:}\mathtt{O}.\forall X{:}\mathtt{O}.$$
$$\forall u{:}U\to V.\forall v{:}V\to W.\forall w{:}U\to W.\forall x{:}W\to X.\forall y{:}V\to X.\forall z{:}U\to X$$
$$((\mathtt{T}(u,v,w)\wedge\mathtt{T}(v,x,y)\wedge\mathtt{T}(w,x,z))\longrightarrow\mathtt{T}(u,y,z))\ .$$

We have applied certain abbreviations in writing these formulas. The atomic formula $\mathtt{I}(i)$ should be really $\mathtt{I}(U,i)$ ; $U$ is also a variable in it; in fact, $i{:}U\to U$ cannot appear anywhere without $U$. Similarly, $\mathtt{T}(u,v,w)$ is really $\mathtt{T}(U,V,W,u,v,w)$ . However, the abbreviations used are systematic, and can be made into a formal feature. Also, $w{=}w'$ is an atomic formula depending on all of the variables $U$, $W$, $w$, $w'$ ; it is written, more fully, as $w{=}_{\mathtt{A}(U,W)}w'$ .

Many of the usual properties of categories, and of diagrams of objects and arrows in categories, can be expressed in FOLDS over $\mathbf{L}_{\mathtt{cat}}$ . For instance, the definition of elementary topos (with operations defined by universal properties up to isomorphism, not specified as univalued operations) can be given as a finite set of sentences in FOLDS over $\mathbf{L}_{\mathtt{cat}}$ ; the reader will find it easy to write down the axioms for elementary topos in the style of the above axioms for category. As Freyd [F] and Blanc [B] have shown, and as we will see below, this is closely related to the fact that the usual properties of categories, and of diagrams in categories, are invariant under equivalence of categories.

Let us turn to the formal specification of the syntax of logic with dependent sorts. We fix a DSV $\boldsymbol{L}$ . For a while, only the kinds in $\boldsymbol{L}$ will be used; let $\boldsymbol{K}$ be the full subcategory of $\boldsymbol{L}$ on the objects the kinds; $\boldsymbol{K}$ is a simple category, the *category of kinds* of $\boldsymbol{L}$ ; it may regarded as a DSV without relations.

Note that kinds have been assigned a level in $\boldsymbol{K}$; levels range over the natural numbers less than $k$, where $k$ is the height of $\boldsymbol{K}$. Recall that for any $K \in \boldsymbol{K}$, we use the notation $K|\boldsymbol{K}$ for the set of all *proper* arrows $p : K \to K_p$ with domain $K$. The set $K|\boldsymbol{K}$ will figure as the *arity* of the symbol $K$. In particular, the ones with empty arity are exactly the level-0 kinds.

We are going to define what sorts are, and what variables of a given sort are. These notions are relative to a given $\boldsymbol{L}$ (actually, to the category $\boldsymbol{K}$ of kinds of $\boldsymbol{L}$), which is considered fixed now.

When $X$ is anything, we write $x : X$ to mean that $x = \langle 2, X, a \rangle$ for some (any) $a$. When we have defined sorts, and $X$ is a sort, $x : X$ is to be read as "$x$ is a variable of sort $X$".

By definition, a *sort* is an entity of the form

$$\langle 1, K, \langle x_p \rangle_{p \in K|\boldsymbol{K}} \rangle$$

such that $K$ is a kind, and for each $p \in K|\boldsymbol{K}$, and for

$$X_p \stackrel{\text{def}}{=} \langle 1, \mathrm{K}_p, \langle x_{qp} \rangle_{q \in \mathrm{K}_p|\boldsymbol{K}} \rangle \ ,$$

we have $x_p : X_p$.

We will also write $K(\langle x_p \rangle_{p \in K|\boldsymbol{K}})$ for $\langle 1, K, \langle x_p \rangle_{p \in K|\boldsymbol{K}} \rangle$; thus, a sort is obtained by filling in the "$p$th" place of a kind $K$, for any $p$ in the arity $K|\boldsymbol{K}$ of $K$, by a *suitable* variable $x_p$. The sort $K(\langle x_p \rangle_{p \in K|\boldsymbol{K}})$ is said to be *of the kind* $K$.

When $X$ is a sort, and $x : X$, that is, $x = \langle 2, X, a \rangle$ for some $a$, $x$ is called a *variable of sort* $X$; $a$ is called the *parameter* of the variable $x$. Usually, the notation $x : X$ will imply that $X$ is a sort.

Note that every variable "carries" its own sort with it. This is in contrast with the practice of most of the relevant literature (see e.g. [C]), where variables are "locally" declared to be of certain definite sorts, but by themselves, they do not carry sort-information. For a sort $X = K(\langle x_p \rangle_{p \in K|\boldsymbol{K}})$, $\mathrm{Var}(X) \stackrel{\text{def}}{=} \{x_p : p \in K|\boldsymbol{K}\}$; and if $x : X$, $\mathrm{Dep}(x) \stackrel{\text{def}}{=} \mathrm{Var}(X)$; $x$ *depends on* the variables in $\mathrm{Dep}(x)$.

Note also that *any* parameter gives rise to a variable of a given sort; for any sort $X$, and for any $a$ whatever, $\langle 2, X, a\rangle$ is a variable of sort $X$. In the "purely syntactic" contexts, it suffices to restrict the parameters to be natural numbers (thereby ensuring a countable infinite recursive set of variables of each sort). However, for the purposes of model-theory, it is convenient to have a proper class of variables of each sort (as a consequence, we have a proper class of sorts). Let us call a variable *natural* when its parameter, as well as that of each variable it depends on, *etc.*, is a natural number.

For a variable $y$, let's write $\mathrm{X}_y$ for the sort of $y$ ($y{:}\mathrm{X}_y$), and let's use the notation

$$y{:}\mathrm{X}_y{=}\mathrm{K}_y(\langle \mathrm{x}_y, p\rangle_{p\in \mathrm{K}_y \mid \boldsymbol{L}}) \qquad (1')$$

displaying the ingredients of the sort $\mathrm{X}_y$ in dependence on $y$. Also, let's write $\mathrm{a}(y)$ for the parameter of $y$.

The first question arising concerning the definition of "sort" is whether the constituent entities $X_p$ are also sorts; the answer is "yes". Assume $X = \langle 1, K, \langle x_p\rangle_{p\in K\mid \boldsymbol{K}}\rangle$ is a sort. Applying the definition of "sort" to $X_p$, for $q\in \mathrm{K}_p \mid \boldsymbol{K}$, we want that for

$$(X_p)_q = \langle 1, \mathrm{K}_q, \langle x_{(rq)p}\rangle_{r\in \mathrm{K}_q\mid \boldsymbol{K}}\rangle \ ,$$

we have $x_{qp}{:}(X_p)_q$. But since $\mathrm{K}_q = \mathrm{K}_{qp}$ and $(rq)p = r(qp)$, we have $(X_p)_q = X_{qp}$; and $x_{qp}{:}X_{qp}$, by $X$ being a sort.

Although the definition unambiguously defines what sorts and variables are, it is not (quite) clear, for instance, that for every $K\in \boldsymbol{K}$, *there are* sorts of the kind $K$. We show that the sorts of the kind $K$ are in a bijective correspondence with families $\langle a_p\rangle_{p\in K\mid \boldsymbol{K}}$ of arbitrary entities $a_p$; the correspondence maps $X = \langle 1, K, \langle x_p\rangle_{p\in K\mid \boldsymbol{K}}\rangle$ to $\mathrm{a}(X) \stackrel{\mathrm{def}}{=} \langle \mathrm{a}(x_p)\rangle_{p\in K\mid \boldsymbol{K}}$ for which $x_p = \langle 2, X_p, a_p\rangle$ for a suitable $X_p$.

We want to prove that for any $\langle a_p\rangle_{p\in K\mid \boldsymbol{K}}$, there is a unique sort $X$ of the kind $K$ with $\mathrm{a}(X) = \langle a_p\rangle_{p\in K\mid \boldsymbol{K}}$.

Let $K\in \boldsymbol{K}$ and $\langle a_p\rangle_{p\in K\mid \boldsymbol{K}}$ be given. By recursion on the level of $K_p$, for each $p\in K\mid \boldsymbol{K}$, we define $X_p$ (a sort, as it turns out), and the variable $x_p{:}X_p$. Let $p\in K\mid \boldsymbol{K}$. We put

$X_p \stackrel{=}{\mathrm{def}} \langle 1, \mathrm{K}_p, \langle x_{qp}\rangle_{q\in \mathrm{K}_p \mid \boldsymbol{K}}\rangle$ , and $x_p \stackrel{=}{\mathrm{def}} \langle 2, X_p, a_p\rangle$ .

Since for each $q\in \mathrm{K}_p \mid \boldsymbol{K}$, $\mathrm{K}_{qp}$ is of lower level than $\mathrm{K}_p$, the entity $x_{qp}$ has been defined; thus, $X_p$ and $x_p$ are defined for $p$ as well. This defines $X_p$ and $x_p$ for all $p$.

Put $X \stackrel{=}{\mathrm{def}} \langle 1, K, \langle x_p\rangle_{p\in K \mid \boldsymbol{K}}\rangle$ . Then $X_p$ formed for $X$ as in the definition of "sort" is the same as the $X_p$ we just defined. Since $x_p : X_p$, $X$ is a sort. Clearly, $\mathrm{a}(X) = \langle a_p\rangle_{p\in K \mid \boldsymbol{K}}$.

The uniqueness of $X$ with this property is (also) easily seen.

Let us remark that for kinds $K$ of level $0$, there is exactly one sort of the kind $K$, namely $K(\emptyset)$ ; this can safely be identified with $K$ itself.

Let us consider the case $\boldsymbol{K} = \mathrm{L}_{\mathrm{graph}}$. We have the level-$0$ sort $\mathrm{O}$ ; let us use the letters $U$, $V$, $W$, ... for denoting variables of sort $\mathrm{O}$ ; $U:\mathrm{O}$, *etc*. The level-$1$ sorts are of the form $\mathrm{A}(\langle x_p\rangle)_{p\in\{\mathrm{d},\mathrm{c}\}}$ with $x_\mathrm{d}, x_\mathrm{c}:\mathrm{O}$, for which we write $\mathrm{A}(x_\mathrm{d}, x_\mathrm{c})$ . Thus, we have sorts $\mathrm{A}(U, V)$ , $\mathrm{A}(V, U)$ , $\mathrm{A}(U, U)$ , ... Let us use $u$ , $v$ , for variables of level $1$ ; we may have $u:\mathrm{A}(U, V)$ , which we paraphrase as $u: U\to V$.

In the case of $\mathrm{L}_{\mathrm{graph}}$ , the ones listed are all the sorts and variables.

For $\boldsymbol{K} = \mathrm{L}_{2\text{-}\mathrm{graph}}$ , we have the additional sorts of the kind $\mathrm{A}_1$ . Let us write $\mathrm{d}_{10}$ for the arrow $\mathrm{dd}_1=\mathrm{dc}_1$ , and $\mathrm{c}_{10}$ for $\mathrm{cd}_1=\mathrm{cc}_1$ . $\mathrm{A}_1 \mid \boldsymbol{K} = \{\mathrm{d}_{10}, \mathrm{c}_{10}, \mathrm{d}_1, \mathrm{c}_1\}$ . Writing $\mathrm{A}_1(x_{\mathrm{d}_{10}}, x_{\mathrm{c}_{10}}, x_{\mathrm{d}_1}, x_{\mathrm{c}_1})$ for $\mathrm{A}(\langle x_p\rangle_{p\in\{\mathrm{d}_{10},\mathrm{c}_{10},\mathrm{d}_1,\mathrm{c}_1\}})$ , we see that the sorts of level-$2$ are those

$$\mathrm{A}(U, V, u, v)$$

for which $U:\mathrm{O}$ , $V:\mathrm{O}$ , $u: U\to V$ and $v: U\to V$. Here, $U$ and $V$, as well as $u$ and $v$, may coincide. We would like to paraphrase $\mathrm{A}(U, V, u, v)$ as

$$\mathrm{A}\Big(U \underset{v}{\overset{u}{\rightrightarrows}} V\Big) .$$

Before we complete the definition of the syntax of logic with dependent sorts, let us discuss the semantics of sorts. To begin with an example, let us take $\mathbf{K} = \mathrm{L}_{2\text{-graph}}$. Now, we know that the intended $\mathbf{K}$-structures are the functors $\mathbf{K} \to \mathrm{Set}$. But we may take a different view. We may say that a $\mathbf{K}$-structure $M$ consists of

a set $M\mathrm{O}$,

for each $A, B \in M(\mathrm{O})$, a set $M\mathrm{A}(A, B)$,

and

for each $A, B \in M(\mathrm{O})$ and $f, g \in M\mathrm{A}(A, B)$, a set $M\mathrm{A}_1(A, B; f, g)$.

This way of thinking of a $\mathbf{K}$-structure emphasizes that an "arrow" $f$ cannot be conceived of *before* its "domain and codomain" $A$, $B$, which have to be elements of $M\mathrm{O}$, have been given; there is a similar consideration for "2-cells". Also note that this kind of $\mathbf{K}$-structure is not literally the same as a functor $\mathbf{K} \to \mathrm{Set}$. The main difference is that, in the new version of the concept, we are not saying anything about the sets $M\mathrm{A}(A, B)$ being disjoint from each other for distinct pairs $(A, B)$. Recall the two different styles of definition of "category" (or "2-category"). The one in which arrows determine their domain and codomain is in the spirit of our notion of structure in the original sense; the other in which we talk about a function $A, B \mapsto \mathrm{hom}(A, B)$ assigning a hom-set to pairs of objects is related to the new concept.

The second version of the concept of $\mathbf{K}$-structure has the following general form. A $\mathbf{K}$-structure $M$ is given by specifying when, for $K \in \mathbf{K}$, the entities $MK(\langle a_p \rangle_{p \in K \mid \mathbf{K}})$ *are defined*, and when they are, what *sets* they are; such data are subject to the following condition:

(2) $MK(\langle a_p \rangle_{p \in K \mid \mathbf{K}})$ is defined iff for each $p \in K \mid \mathbf{K}$, $MK_p(\langle a_{qp} \rangle_{q \in \mathrm{K}_p \mid \mathbf{K}})$ is defined and $a_p \in MK_p(\langle a_{qp} \rangle_{q \in \mathrm{K}_p \mid \mathbf{K}})$.

This formulation hides the recursive character of the concept. Once it is clarified, for all $K$ of level less than $i$, when $MK(\langle a_p \rangle_{p \in K \mid \mathbf{K}})$ is defined, and if so, what set it is, then for any $K$ of level $i$, $MK(\langle a_p \rangle_{p \in K \mid \mathbf{K}})$ is defined iff for all $p \in K \mid \mathbf{K}$, $MK_p(\langle a_{qp} \rangle_{q \in \mathrm{K}_p \mid \mathbf{K}})$ is defined, and $a_p \in MK_p(\langle a_{qp} \rangle_{q \in \mathrm{K}_p \mid \mathbf{K}})$ (note that each $\mathrm{K}_p$ is of level $< i$), and *in that case*, $MK(\langle a_p \rangle_{p \in K \mid \mathbf{K}})$ is *any* set.

Any functor $M\colon \mathbf{K} \to \mathrm{Set}$ gives rise to a $\mathbf{K}$-structure in the new sense. For any $K \in \mathbf{K}$, define $M[K]$ as in (1); declare that $MK(\langle a_p \rangle_{p \in K \mid \mathbf{K}})$ is defined iff $\langle a_p \rangle_{p \in K \mid \mathbf{K}} \in M[K]$, and

in that case, put

$$MK(\langle a_p\rangle_{p\in K\mid \boldsymbol{K}}) \stackrel{\text{def}}{=} \{a\in MK: \bigwedge_{p\in K\mid \boldsymbol{K}} (Mp)(a)=a_p\}\ . \qquad (3)$$

It is clear (using that $M$ is a functor) that now, (2) is satisfied.

But conversely, "essentially all" $\boldsymbol{K}$-structures in the second sense are obtained as functors $\boldsymbol{K}\to \mathtt{Set}$ . The passage from a structure in the new sense to one in the old sense is as follows. Given a $\boldsymbol{K}$-structure $M$ in the new sense, for any $K\in\boldsymbol{K}$ , $M(K)$ is defined as the disjoint union of all *defined* sets $MK(\langle a_p\rangle_{p\in K\mid \boldsymbol{K}})$ , indexed by the tuples $\langle a_p\rangle_{p\in K\mid \boldsymbol{K}}$ , and, for $p:K\to K_p$ , $M(p)$ is given by $M(p)(\langle\langle a_p\rangle_{p\in K\mid \boldsymbol{K}}, a\rangle)=a_p$ ; this defines a functor $\boldsymbol{K}\to \mathtt{Set}$ .

Making the statement that the two notions of $\boldsymbol{K}$-structure are "essentially equivalent" precise would require defining what we mean by an isomorphism of two $\boldsymbol{K}$-structures $M$ and $N$ in the new sense, and showing that the above two passages represent an equivalence of the category of functors $\boldsymbol{K}\to \mathtt{Set}$ with natural isomorphisms as arrows on the one hand, and the category of $\boldsymbol{K}$-structures in the new sense, with isomorphisms in the new sense between them as arrows on the other. We will not go through this exercise, and return to our original concept of " $\boldsymbol{K}$-structure" (" $\boldsymbol{L}$-structure"). However, the concept of an *M-sort* as a set of the form (3) will be used.

Let us now return to the full DS vocabulary $\boldsymbol{L}$ , and define what $\boldsymbol{L}$-formulas in logic with dependent sorts are. We will have two versions: logic with dependent sorts *with* (*restricted*) *equality*, and logic with dependent sorts *without equality*. FOLDS *with unrestricted equality* also makes sense; however, it turns out to be essentially the same as full multisorted logic with equality over $|\boldsymbol{L}|$ (see Appendix C), hence, it is of no real interest.

Let us fix $\boldsymbol{L}$ .

Atomic formulas are defined very similarly to sorts. An *atomic formula* in logic with dependent sorts *without equality* is an entity of the form

$$\langle 3, R, \langle x_p\rangle_{p\in R\mid \boldsymbol{L}}\rangle$$

such that $R$ is a relation in $\mathbf{L}$, and for each $p \in R\,|\,\mathbf{L}$, and

$$X_p \stackrel{=}{\mathrm{def}} \langle 1, \mathrm{K}_p, \langle x_{qp} \rangle_{q \in \mathrm{K}_p\,|\,\mathbf{K}} \rangle \ ,$$

we have $x_p : X_p$. Under these conditions, the $X_p$ are sorts (just as with the definition of "sort"). We write $R(\langle x_p \rangle_{p \in R\,|\,\mathbf{L}})$ for $\langle 3, R, \langle x_p \rangle_{p \in R\,|\,\mathbf{L}} \rangle$.

In logic with (restricted) equality, we also have additional atomic formulas as follows. For any *maximal* kind $K$ (maximal object of $\mathbf{K}$), sort $X = K(\langle x_p \rangle_{p \in K\,|\,\mathbf{K}})$, and variables $x$, $y$, both of sort $X$, we have that $\langle 4, X, x, y \rangle$, written as

$$x =_X y \ ,$$

is an atomic (*equality*) formula.

We define *formulas* $\varphi$ and the set $\mathrm{Var}(\varphi)$ of the *free variables of* $\varphi$ by a simultaneous induction. Any atomic formula is a formula; if $\varphi = R(\langle x_p \rangle_{p \in R\,|\,\mathbf{L}})$, $\mathrm{Var}(\varphi) = \{x_p : p \in R\,|\,\mathbf{L}\}$; if $\varphi :=: x =_X y$, $\mathrm{Var}(\varphi) = \mathrm{Var}(X) \cup \{x, y\}$.

The sentential connectives $\mathbf{t}$, $\mathbf{f}$, $\wedge$, $\vee$, $\rightarrow$, $\neg$, $\longleftrightarrow$ can be applied in an unlimited manner; $\mathrm{Var}(\ )$ for the compound formulas formed using connectives is defined in the expected way; e.g., $\mathrm{Var}(\varphi \wedge \psi) = \mathrm{Var}(\varphi) \cup \mathrm{Var}(\psi)$.

Suppose $\varphi$ is a formula, $x$ is a variable *such that there is no* $y \in \mathrm{Var}(\varphi)$ *with* $x \in \mathrm{Dep}(y)$. Then $\forall x \varphi$, $\exists x \varphi$ are (well-formed) formulas;

$$\mathrm{Var}(\forall x \varphi) \stackrel{=}{\mathrm{def}} \mathrm{Var}(\exists x \varphi) \stackrel{=}{\mathrm{def}} (\mathrm{Var}(\varphi) - \{x\}) \cup \mathrm{Dep}(x) \ .$$

All formulas are obtained as described. (Of course, we have some determinations such as $\forall x \varphi \stackrel{=}{\mathrm{def}} \langle \forall, x, \varphi \rangle$, where $\forall = 7$ (?), *etc.*)

Let us make some remark on logic with (restricted) equality. Just as in ordinary first-order logic, the syntax of logic with equality is the same as that of logic with equality, with the equality-symbol understood as another relation symbol; it is only the semantics that makes the difference.

Formally, for each maximal kind $K$, add to $\mathbf{L}$ an additional relation $\dot{\mathrm{E}}_K$, with morphisms $\mathrm{E}_K \rightrightarrows K$ (arrows $e_{K0}$, $e_{K1}$) subject to $pe_{K0}=pe_{K1}$, for all $p\in K\,|\,\mathbf{L}$; let us denote by $\mathbf{L}^{\mathrm{eq}}$ the extension of $\mathbf{L}$ by these additions. The equality formula $x=_X y$ corresponds to $\mathrm{E}_K(\langle z_r\rangle_{r\in \mathrm{E}_K\,|\,\mathbf{L}})$ where $z_{e_{K0}}=x$, $z_{e_{K1}}=y$, $z_{pe_{K0}}=z_{pe_{K1}}=x_p$. Up to the exchange of these two formulas, for each maximal $K$, the syntax of FOLDS with (restricted) equality over $\mathbf{L}$, and the syntax of FOLDS without equality over $\mathbf{L}^{\mathrm{eq}}$ coincide.

A *context* is a finite set $\mathcal{Y}$ of variables such that if $y\in\mathcal{Y}$, then $\mathrm{Dep}(y)\subset\mathcal{Y}$. It is easy to see that for any formula $\varphi$, $\mathrm{Var}(\varphi)$ is a context.

We explain the semantics of logic with dependent sorts. Let $M$ be any $\mathbf{L}$-structure. Let $\mathcal{Y}$ be a context. We define

$$M[\mathcal{Y}] \;\stackrel{\mathrm{def}}{=}\; \{\langle a_y\rangle_{y\in\mathcal{Y}} : a_y\in MK(\langle a_{x_{y,p}}\rangle_{p\in \mathrm{K}_y\,|\,\mathbf{L}}) \text{ for all } y\in\mathcal{Y}\} \quad (4)$$

(recall the notations (1') and (3)).

By recursion on the complexity of the formula $\varphi$, we define $M[\mathcal{Y}:\varphi]$, *the interpretation of* $\varphi$ *in* $M$ *in the context* $\mathcal{Y}$, whenever $\mathcal{Y}$ is a context such that $\mathrm{Var}(\varphi)\subset\mathcal{Y}$; we will have that $M[\mathcal{Y}:\varphi]\subset M[\mathcal{Y}]$. For an atomic formula $R(\langle x_p\rangle_{p\in R\,|\,\mathbf{K}})$, we stipulate, for any $\langle a_y\rangle_{y\in\mathcal{Y}}\in M[\mathcal{Y}]$,

$$\langle a_y\rangle_{y\in\mathcal{Y}}\in M[\mathcal{Y}:R(\langle x_p\rangle_{p\in R\,|\,\mathbf{K}})] \;\stackrel{\mathrm{def}}{\Longleftrightarrow}\; \langle a_{x_p}\rangle_{p\in R\,|\,\mathbf{K}}\in M(R)$$

(recall that $M(R)\subset M[R]$; clearly, $\langle a_{x_p}\rangle_{p\in R\,|\,\mathbf{K}}\in M[R]$ automatically).

In case of logic with equality,

$$\langle a_y\rangle_{y\in\mathcal{Y}}\in M[\mathcal{Y}:u=_X v] \;\stackrel{\mathrm{def}}{\Longleftrightarrow}\; a_u=a_v.$$

For the propositional connectives, the clauses are the expected ones; *e.g.*,

$$\langle a_y\rangle_{y\in\mathcal{Y}} \in M[\mathcal{Y}: \psi\wedge\theta] \iff_{\text{def}} \langle a_y\rangle_{y\in\mathcal{Y}} \in M[\mathcal{Y}:\psi] \text{ and } \langle a_y\rangle_{y\in\mathcal{Y}} \in M[\mathcal{Y}:\theta] \ .$$

Let us consider $\forall x\psi$, and a context $\mathcal{Y}$ such that $\mathtt{Var}(\forall x\psi)\subset\mathcal{Y}$. Let $x: K(\langle x_p\rangle_{p\in K} \mid \boldsymbol{L})$. First, assume that $x\notin\mathcal{Y}$; this is the case in particular when

$$\mathcal{Y}=\mathtt{Var}(\forall x\psi)=(\mathtt{Var}(\varphi)-\{x\})\cup\mathtt{Dep}(x) \ .$$

Let $\mathcal{Y}'=\mathcal{Y}\dot\cup\{x\}$; $\mathcal{Y}'$ is a context. When $\langle a_y\rangle_{y\in\mathcal{Y}}\in M[\mathcal{Y}]$ and $a\in MK(\langle a_{x_p}\rangle_{p\in K}\mid \boldsymbol{L})$, let $\langle a_y\rangle_{y\in\mathcal{Y}}(a/x)$ denote $\langle a'_y\rangle_{y\in\mathcal{Y}'}$ for which $a'_y=a_y$ for $y\in\mathcal{Y}$, and $a'_x=a$. We see that $\langle a_y\rangle_{y\in\mathcal{Y}}(a/x)\in M[\mathcal{Y}']$ as follows. Note that for $y\in\mathcal{Y}$, we have $x\notin\mathtt{Dep}(y)$ (since $x\in\mathtt{Dep}(y)$ would imply that $x\in\mathcal{Y}$); as a consequence, $a'_y\in MK_y(\langle a'_{x_{y,p}}\rangle_{p\in K_y}\mid \boldsymbol{L}$ is equivalent to $a_y\in MK_y(\langle a_{x_{y,p}}\rangle_{p\in K_y}\mid \boldsymbol{L}$ for $y\in\mathcal{Y}$; while for $y=x$, the same holds by the assumption on $a$. We define

$$\langle a_y\rangle_{y\in\mathcal{Y}}\in M[\mathcal{Y}:\forall x\psi] \iff_{\text{def}}$$

for all $a\in MK(\langle a_{x_p}\rangle_{p\in K}\mid \boldsymbol{L})$, we have $\langle a_y\rangle_{y\in\mathcal{Y}}(a/x)\in M[\mathcal{Y}':\psi]$;

and

$$\langle a_y\rangle_{y\in\mathcal{Y}}\in M[\mathcal{Y}:\exists x\psi] \iff_{\text{def}}$$

there is $a\in MK(\langle a_{x_p}\rangle_{p\in K}\mid \boldsymbol{L})$ such that $\langle a_y\rangle_{y\in\mathcal{Y}}(a/x)\in M[\mathcal{Y}':\psi]$ .

In the general case for $\mathcal{Y}\supset\mathtt{Var}(\forall x\psi)$, define

$$\langle a_y\rangle_{y\in\mathcal{Y}}\in M[\mathcal{Y}:\forall x\psi] \iff \langle a_y\rangle_{y\in\hat{\mathcal{Y}}}\in M[\hat{\mathcal{Y}}:\forall x\psi] \ ,$$
$$\langle a_y\rangle_{y\in\mathcal{Y}}\in M[\mathcal{Y}:\exists x\psi] \iff \langle a_y\rangle_{y\in\hat{\mathcal{Y}}}\in M[\hat{\mathcal{Y}}:\exists x\psi] \ ,$$

where $\hat{\mathcal{Y}}=\mathtt{Var}(\forall x\psi)=\mathtt{Var}(\exists x\psi)$. It is clear that when $x\notin\mathcal{Y}$, the second definition gives the same answer as the first one.

As usual, we also write $M \models \varphi[\langle a_Y \rangle_{Y \in \mathcal{Y}}]$ for $\langle a_Y \rangle_{Y \in \mathcal{Y}} \in M[\mathcal{Y}:\varphi]$ .

This completes the definition of the standard, Set-valued semantics of FOLDS.

Let us note that when $\varphi$ is a formula in logic with equality over $\boldsymbol{L}$, and $\hat{\varphi}$ is the corresponding formula in logic without equality over $\boldsymbol{L}^{\mathrm{eq}}$, obtained by exchanging the equality subformulas for $\mathrm{E}_K$-formulas, and $M$ is an $\boldsymbol{L}$-structure, then $M[\mathcal{Y}:\varphi] = M[\mathcal{Y}:\hat{\varphi}]$ ; in the latter instance, $M$ denotes the *standard* $\boldsymbol{L}^{\mathrm{eq}}$-structure in which each $\mathrm{E}_K$ is interpreted as true equality. In short, the semantics of logic with equality over $\boldsymbol{L}$ coincides with the semantics of logic without equality over $\boldsymbol{L}^{\mathrm{eq}}$ when the latter is restricted to standard structures.

Let us formulate a simple translation of logic with dependent sorts into ordinary multisorted logic. This amounts to a mapping $\varphi \mapsto \varphi^*$ of $\boldsymbol{L}$-formulas $\varphi$ of FOLDS to $|\boldsymbol{L}|$ -formulas of multisorted logic. Let us agree that every variable $x:X$, with $X$ a sort of kind $K$, will be regarded, in multisorted logic over $\boldsymbol{L}$, a variable of sort $K$.
The mapping $\varphi \mapsto \varphi^*$ will be so defined that the free variables of $\varphi^*$ are exactly the same as those of $\varphi$. Moreover, the essential property of the translation is that, for any $\boldsymbol{L}$-structure $M$,

$$M \models \varphi[\langle a_Y \rangle_{Y \in \mathcal{Y}}] \iff M \models \varphi^*[\langle a_Y \rangle_{Y \in \mathcal{Y}}] \ ;$$

here, in the second instance, we referred to the usual notion of truth for multisorted logic. The definition is this:

for an atomic formula $\varphi :=: R(\langle x_p \rangle_{p \in R \mid \boldsymbol{K}})$ ,

$$\varphi^* \ \mathrm{d\bar{e}f} \ \dot{R}(\langle x_p \rangle_{p \in R \mid \boldsymbol{K}}) \ \mathrm{d\bar{e}f} \ \exists y \in R. \bigwedge_{p \in R \mid \boldsymbol{L}} p(y) =_{\mathrm{K}_p} x_p \ ; \qquad (5)$$

for an equality formula $\varphi :=: x =_X y$ ,

$$\varphi^* \ \mathrm{d\bar{e}f} \ x =_K y$$

(here, $X$ is a sort of the kind $K$) ;

$(\ )^*$ commutes with propositional connectives;

$$(\forall x\varphi)^* \stackrel{\text{def}}{=} \forall x\Big(\bigwedge_{p\in K\,|\,\boldsymbol{L}} p(x)=_{K_p} x_p \longrightarrow \varphi^*\Big)\ ;$$

$$(\exists x\varphi)^* \stackrel{\text{def}}{=} \exists x\Big(\bigwedge_{p\in K\,|\,\boldsymbol{L}} p(x)=_{K_p} x_p \wedge \varphi^*\Big)$$

(in the last two clauses, $x:K(\langle x_p\rangle_{p\in K\,|\,\boldsymbol{L}})$ ).

We have a straightforward extension of the semantics of logic with dependent sorts to interpretations in categories so that the standard semantics will appear as the special when the target category is $\mathtt{Set}$ . First of all notice that the notion of $\boldsymbol{C}$-*valued* $\boldsymbol{L}$-*structure* makes sense for *any* category $\boldsymbol{C}$ ; it is that of a functor $M:\boldsymbol{L}\to\boldsymbol{C}$ such that for any $R\in\mathtt{Rel}(\boldsymbol{L})$ , the family $\langle M(p)\rangle_{p\in R\,|\,\boldsymbol{L}}$ of morphisms in $\boldsymbol{C}$ is jointly monomorphic. The use of the notation $M:\boldsymbol{L}\to\boldsymbol{C}$ will imply that $M$ is a $\boldsymbol{C}$-valued $\boldsymbol{L}$-structure. From now on, let us assume, at least, that $\boldsymbol{C}$ has finite limits.

Let $M:\boldsymbol{L}\to\boldsymbol{C}$ . For any object $A$ of $\boldsymbol{L}$ (kind or relation), we define $M[A]$ as the limit (joint pullback) of the diagram $A\downarrow(\boldsymbol{L}-\{1_A\})\xrightarrow{\Phi}\boldsymbol{L}\xrightarrow{M}\boldsymbol{C}$ , with $\Phi$ the forgetful functor; $M\circ\Phi$ maps $p:A\to K_p$ to $M(K_p)$ . Let us write $\pi_p$ , or $\pi^M_p$ , for the limit projection $M[A]\to M(K_p)$ , and let $\pi_A=\pi^M_A:M(A)\to M[A]$ be the canonical arrow for which $\pi_p\circ\pi_A=M(p)$ . When $A$ is a relation $R$ , then $\pi_R$ is a monomorphism; we also write $\mathrm{m}^M_R$ for $\pi^M_R$ . When $A$ is a kind $K$ , then $U[K]$ and $\pi^U_p:U(K)\to U[K]$ are defined for any $U:\boldsymbol{K}\to\boldsymbol{C}$ (formally, by using the above definition for $\boldsymbol{K}$ in place of $\boldsymbol{L}$ ); of course, when $U=M\restriction\boldsymbol{K}$ , then $U[K]=M[K]$ , $\pi^U_p=\pi^M_p$ .

Continuing, let $\mathcal{Y}$ be a context; we will define $M[\mathcal{Y}]$ . We construct a graph $\langle\mathcal{Y}\rangle$ and a diagram $\Phi_{\mathcal{Y}}:\langle\mathcal{Y}\rangle\longrightarrow\boldsymbol{L}$ as follows. The objects of $\langle\mathcal{Y}\rangle$ are the elements of $\mathcal{Y}$ , $\mathrm{Ob}\langle\mathcal{Y}\rangle=\mathcal{Y}$ . The arrows of $\langle\mathcal{Y}\rangle$ are $\langle y, z; p\rangle:y\to z$ , one for each $p\in K_y\,|\,\boldsymbol{L}$ such that $z=x_{y,p}$ . $\Phi_{\mathcal{Y}}$ maps $y$ to $K_y$ , $\langle y, z; p\rangle:y\to z$ to $p:K_y\to K_p(=K_z)$ . $M[\mathcal{Y}]$ is defined as the limit of the composite $M\Phi_{\mathcal{Y}}:\langle\mathcal{Y}\rangle\longrightarrow\boldsymbol{C}$ ; let us denote the projections for this limit by $\pi_y=\pi^M_y=\pi^M_{\mathcal{Y},y}:M[\mathcal{Y}]\to M(K_y)$ $(y\in\mathcal{Y})$ .

We define $M[\mathcal{Y}:\varphi]$ as a certain subobject of $M[\mathcal{Y}]$, by recursion on the complexity of $\varphi$.

Let $\varphi$ be the atomic formula $R(\langle x_p\rangle_{p\in R\,|\,\mathbf{L}})$, and let $\mathcal{Y}$ be a context such that $\mathrm{Var}(\varphi)=\{x_p : p\in R\,|\,\mathbf{L}\}\subset\mathcal{Y}$. Let $f: M[\mathcal{Y}]\to M[R]$ be the arrow, given by the universal property of the limit defining $M[R]$, for which $\pi_p\circ f=\pi_{x_p}$ ($p\in R\,|\,\mathbf{L}$). $M[\mathcal{Y}:\varphi]$ is defined by the pullback

$$\begin{array}{ccc} M[\mathcal{Y}:\varphi] & \longrightarrow & M(R) \\ {\scriptstyle \mathrm{m}^M_{\mathcal{Y},\,\varphi}}\downarrow & \square & \downarrow{\scriptstyle \mathrm{m}^M_R} \\ M[\mathcal{Y}] & \xrightarrow[f]{} & M[R] \end{array}$$

(that is, $M[\mathcal{Y}:\varphi]$ as a subobject of $M[\mathcal{Y}]$ is represented by the monomorphism $\mathrm{m}^M_{\mathcal{Y},\,\varphi}$).

For formulas built by a propositional connective from simpler formulas, the definition is the expected one. *E.g.*,

$$M[\mathcal{Y}:\varphi\to\psi] \;=\; M[\mathcal{Y}:\varphi]\longrightarrow M[\mathcal{Y}:\psi]\ ,$$

where on the right-hand-side, reference is made to the Heyting implication $\longrightarrow$ in the subobject lattice $\mathrm{S}(M[\mathcal{Y}])$; of course, $M[\mathcal{Y}:\varphi\to\psi]$ is defined if and only if the corresponding instance of Heyting implication is defined in $\mathrm{S}(M[\mathcal{Y}])$.

Let $x\notin\mathcal{Y}$, and $\mathrm{Var}(\forall x\varphi)\subset\mathcal{Y}$. We have $f: M[\mathcal{Y}\dot\cup\{x\}]\to M[\mathcal{Y}]$ for which $\pi_Y\circ f=\pi'_Y$ ($Y\in\mathcal{Y}$; $\pi_Y=\pi^M_{\mathcal{Y},\,Y}$, $\pi'_Y=\pi^M_{\mathcal{Y}\dot\cup\{x\},\,Y}$). Let $\exists_f, \forall_f: \mathrm{S}(M[\mathcal{Y}\dot\cup\{x\}])\rightleftarrows S(M[\mathcal{Y}])$ be the partial left and right adjoints to $f^*: M[\mathcal{Y}]\to M[\mathcal{Y}\dot\cup\{x\}]$, the latter defined by pulling back along $f$. We define

$$M[\mathcal{Y}:\exists x\varphi] \;=\; \exists_f(M[\mathcal{Y}\dot\cup\{x\}:\varphi])\ ,$$

$$M[\mathcal{Y}:\forall x\varphi] \;=\; \forall_f(M[\mathcal{Y}\dot\cup\{x\}:\varphi])\ .$$

For $M[\mathcal{Y}:\exists x\varphi]$ or $M[\mathcal{Y}:\forall x\varphi]$ to be defined, it is necessary and sufficient that the corresponding instance of $\exists_f$, respectively $\forall_f$ be defined.

For the coherent part of the language (atomic formulas, **t** , **f** , $\wedge$ , $\vee$ , $\exists$ ) to be interpretable in the category, it suffices that $\mathbf{C}$ is a coherent category (see e.g. [MR1]). For the interpretation of the full language, it is suffices to have that $\mathbf{C}$ is a Heyting category (see *e.g.* [MR2]).