First Order Logic with Dependent Sorts,
with Applications to Category Theory

by M. Makkai
McGill University

Preliminary version (Nov 6, 1995)

Contents:

The author's research is supported by NSERC Canada and FCAR Quebec

Introduction

1. This work introduces First-Order Logic with Dependent Sorts (FOLDS). FOLDS is inspired by Martin-Löf's Theory of Dependent Types (TDT) [M-L]; in fact, FOLDS may be regarded a proper part of TDT, similarly to ordinary first-order logic being a proper part of higher-order logic. At the same time, FOLDS is of a much simpler nature than the theory of dependent types. First of all, the expressive power of FOLDS is no more than that of ordinary first-order logic; in fact, FOLDS may be regarded as a constrained form of Multi-Sorted First-Order Logic (MSFOL). Secondly, the syntax of FOLDS is quite simple, only slightly more complicated than that of MSFOL.

In general terms, the significance of FOLDS is analogous to that of ordinary first-order logic (FOL). On the one hand, FOL has a simple and powerful semantic metatheory; on the other hand, FOL is the basis of a multitude of specific foundational theories. Correspondingly, FOLDS has a simple semantic metatheory, not essentially more complicated than that for FOL. It is one of the aims of this work to develop the basic semantic theory of FOLDS. On the other hand, I make a start on showing that FOLDS is good, and better than FOL, for the purposes of formal systems dealing with sets, categories, and more general categorical concepts.

FOLDS is very simple; for the understanding of the motivation for, and the basic mechanics of, FOLDS there is no need for any prior knowledge of the, by now, extensive literature of dependent types. I find the idea of FOLDS so simple and natural (we will also see that FOLDS is *useful*, which is another issue) that I am thoroughly surprised by the apparent fact that, in the literature, it has not so far been singled out for study. (Nevertheless, there are important pointers to FOLDS in the literature that I will point out below.) Incidentally, I decided to use the word "sort", instead of "type", in "first-order logic with dependent sorts", to emphasize the closeness of FOLDS to MSFOL, and because of the strongly-felt connotation, in phrases like "type-theory", of the word "type", that implies the presence of a higher-order structure; you would not say "multi-typed first-order logic", would you?

J. Cartmell [C] introduced a syntax of variable types for the purposes of a novel presentation of generalized algebraic theories; Cartmell's syntax was also "abstracted from ... Martin-Löf type theory". FOLDS differs in two ways from Cartmell's syntax. Firstly, in Cartmell's syntax, there are no logical operators in the usual sense; there are no propositional connectives, or quantifiers; FOLDS has them, with quantification constrained in the natural way already given in TDT. Secondly, the type-structure of FOLDS is much simpler than that of Cartmell's syntax.

Cartmell's syntax may be characterized as the result of abstracting the structure of *contexts, types, terms* and *equality* out of TDT. FOLDS has the first two of these, contexts and types (although the latter are called "sorts"), but it does not have the third, terms (except in the rudimentary form of mere variables), and it has equality in a greatly restricted form only.

The restriction on the use of equality in FOLDS is a fundamental feature. FOLDS is to be used in formulating categorical situations in which, for example, equality of objects of a category is not an admissible primitive. The absence of term-forming operators, to be interpreted as functions, is a consequence of the absence of equality; it seems to me that the notion of "function" is incoherent without equality.

It is convenient to regard FOLDS a logic without equality entirely, and deal with equality, as much as is needed of it , as extralogical primitives.

It is worth-while for the reader at this point to make a quick comparison of the way [C] formulates the theory of categories (pp. 212, 213 in [C]), and the way FOLDS formulates the same (see §1, p.11). Let me emphasize that essentially this particular instance of FOLDS have been introduced early on by G. Blanc [B], in his characterization (mathematically equivalent to P. Freyd's earlier characterization) of first-order properties of categories invariant under equivalence of categories. A. Preller [P] makes the specification of the specific instance of FOLDS clearer. The theme of invariance under equivalence is in fact the main theme for this work; see below.

The FOLDS formulation of the theory of categories is, admittedly, longer than the Cartmell formulation. It consists in writing out the axioms of "category" in essentially the usual first-order terms, with a special regard for the typing of variables. The main points to observe are that (1) no equality on objects is used; (2) equality of arrows is used only when the arrows already are assumed to be parallel; and (3) quantification on arrows is restricted to one hom-set at a time.

The formulation in [C] is more "mathematical"; in particular, the essential algebraic nature of the concept of category is clear on it, whereas, because of the presence of the usual first-order operators that in general do not yield essentially algebraic concepts, in the FOLDS formulation the essential algebraic quality of the concept of category is obscured.

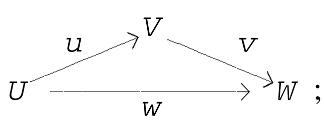
In the case of the theory of categories, the notions of *context* in the two formulations coincide; in fact, now a context is a finite diagram of objects and arrows represented by variables. Below, we will take a look at the formulations of the concept of a category with finite limits in the two frameworks, when the differences become greater.

The most obvious difference of the two formulations is that the one in FOLDS is *purely relational*, in Cartmell's syntax, purely operationaI. In FOLDS, the concepts of identity and composition are represented by relations, rather than operations as in [C]. The arity of a relation is the type of a particular *context*; the places of a relation are to be filled by variables forming a context of a given type. To give an example, in case of composition as a relation, the variables filling the places of the relation T (for (commutative) triangle) form a system consisting of variables U, V, W, u, v, w (not necessarily all distinct), related to each other by sorting data

$$U, V, W{:}O;\ u{:}A(U, V),\ v\in A(V, W),\ w\in A(U, W)$$

(O for "object", A for "arrow"),

or more pictorially,



T then *says* of this diagram that it is commutative.

A general context in the FOLDS language for categories is a finite graph of object and arrow variables, with sorting data specifying which object variable is the domain of each arrow variable, and the same for codomain (when we say "graph", we mean to imply that there is no arrow-variable without a corresponding object-variable designated as its domain, or codomain).

An immediate consequence of the absence of operations in FOLDS is the simplification of the notions of *context* and *type* (*sort*) in FOLDS with respect to the Cartmell syntax. To see the effect of this, we take the example of the theory of categories with finite limits. Although this example is not discussed in [C], it is highly relevant to the subject of [C] as acknowledged by the title of section 6: "Essentially algebraic theories and categories with finite limits".

In the Cartmell syntax, pullbacks would be introduced by the following introductory rules:

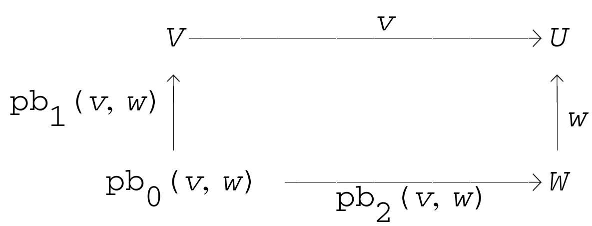
$$U,\ V,\ W \in \mathrm{Ob}\ ,\quad v \in \mathrm{Hom}(V,\ U)\ ,\quad w \in \mathrm{Hom}(W,\ U)\ :\ \mathrm{pb}_0(U,\ V,\ W,\ v,\ w) \in \mathrm{Ob}$$

(here, the "informal syntax" allows writing $\mathrm{pb}_0(v,\ w)$ in place of the longer term) ;

$$U,\ V,\ W \in \mathrm{Ob}\ ,\quad v \in \mathrm{Hom}(V,\ U)\ ,\quad w \in \mathrm{Hom}(W,\ U)\ :$$

$$\mathrm{pb}_1(v,\ w) \in \mathrm{Hom}(\mathrm{pb}_0(v,\ w),\ V)\ ,\qquad \mathrm{pb}_2(v,\ w) \in \mathrm{Hom}(\mathrm{pb}_0(v,\ w),\ W)\ .$$

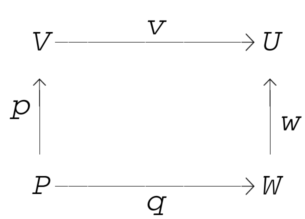
(Of course, one has in mind the pullback diagram



.)

There are further terms and rules expressing the universal property of the pullback.

Now, in FOLDS, we have two possibilities. One is simply adopting the same language of categories as before; after all, pullbacks are first-order definable in the language of categories; in fact, pullbacks are definable in FOLDS over the language of categories. Another possibility would be adopting an additional primitive relation of arity the diagram



;

we would do this if we wanted (as we may) to keep down the quantifier complexity of the axioms of the resulting theory. In either case, appropriate first-order axioms, formulated in FOLDS, are adopted.

Now, compare the notions of context and type (sort) in the Cartmell formulation, to those in the FOLDS formulations, in this example. In either of the FOLDS formulations, the notions of context and type remain the same as in the previous example of the theory of categories; in particular, contexts are finite graphs of variables. However, in the Cartmell formulation, because of the presence of terms of arbitrarily high complexity, both of the type of an object and of an arrow, contexts and types of arbitrarily high complexity will come up. In particular, the second rule above features a type with a place filled by a term which is not a variable.

This example explains the reason for the complexity of the definition the general concept of theory in Cartmell's syntax; see section 6, *loc.cit*. In particular, the definition of "type" cannot be made independent of the axioms of the theory in question; what counts as a well-formed type depends on what axioms are present. This is not at all unexpected; M. Coste's earlier syntax for essentially algebraic theories [Co] (not referred to in [C]) also had this feature. In contrast, in FOLDS, there is no such complication in the definition of "type" ("sort").

Let me point out another aspect in which FOLDS is simpler than Cartmell's syntax. In FOLDS, one never substitutes in a sort expression; in the formal system, there is a substitution rule, but it does not effect sorts. Related to this is the circumstance that the sorting of variables can be given rigidly; that is, when we say that the variable x is sort X, where the sort X may contain further variables, we mean a *formal*, once-for-all specification concerning x. In FOLDS, in contrast to Cartmell's syntax, it is impossible to have the same variable x to be declared of types X and Y unless X and Y are *literally* the same.

I consider the just-described feature of FOLDS to be of foundational importance. The view underlying FOLDS is that sort-declarations are not subject to logical manipulation; they are not propositions; one cannot negate a sort-declaration. One cannot ask *whether* x is of sort X *within* logic; the variable x being of sort X is purely notational, or conventional, matter. More pointedly, *membership in a set* is not a matter for logic; what *is* the matter for logic is whether certain elements, declared to belong to various sets, do or do not satisfy certain *predicates*. One should compare simple type theory (higher-order logic), in which typing of variables is also absolute. The difference in FOLDS is only that the type of a variable may also contain variables; however, the latter variables are uniquely determined from the variable being typed.

There is an important difference between the FOLDS-formulation and the Cartmell formulation, indicated above, of the notion of category with finite limits; in fact, the very notions formulated, not just their formulation, differ. Cartmell's syntax formalizes the notion of category with *specified* finite limits; FOLDS (in our application) formalizes the notion of category with finite limits, with the latter defined *only up to isomorphism*. Moreover, Cartmell's syntax *cannot* formalize the latter notion, for the simple reason that that notion is not an essentially algebraic one. Conversely, FOLDS, with the restriction that no equality on objects is allowed, cannot formalize the notion of category with *specified* finite limits.

It is possible to recapture the full expressive power, and more, of Cartmell's syntax within the framework of FOLDS. This will essentially be shown in Appendix C, when discussing "global equality". However, FOLDS with global equality captures more than Cartmell's syntax; because of this, it fails to represent that syntax faithfully. Thus, Cartmell's syntax is not rendered superfluous, or redundant, in any sense by what we do here. There is a similar situation with Coste's syntax for essentially algebraic theories mentioned above. Coste's syntax is one using the unique existential quantifier; it can be easily subsumed under *the simpler* regular logic which uses the ordinary existential quantifier. The point of Coste's syntax, and of Cartmell's, is that they capture *exactly* the essentially algebraic doctrine. In addition, I want to stress the great practical value of Cartmell's syntax. It is, in my opinion, the most practical specification language for structures such as (possibly) higher dimensional categories, with (possible) additional structure.

In this work, I present two ways of introducing FOLDS, which, however, are ultimately equivalent; one in §1, the other in Appendix A. The one in Appendix A is the more direct one. It starts with a simultaneous inductive definition of the concepts of kind, context, sort and variable, together with some other auxiliary concepts. Kinds are the heads (names) of sorts; each sort is obtained by *appropriately* filling out the places of a kind by variables. After defining the syntax in a global manner, one isolates specific *vocabularies*, or similarity types, for the purposes of formulating specific *theories* in FOLDS.

On the other hand, the treatment in §1 starts with the idea of a vocabulary for FOLDS (DSV). It is interesting that the data for a DSV can be naturally and succinctly captured by a, usually finite, *one-way category*. One-way categories were isolated by F. W. Lawvere in [L]; a category is one-way if its endomorphism monoids are trivial; in the skeletal case, this means that there are no non-trivial circuits of arrows. Subsequently, Lawvere observed that one-way categories are intimately related to the sketch-based syntax of [M1]. Their appearance in this paper is related to their role in [M1], although this fact is not worked out here. The DSV as a one-way category has objects the kinds and the relation-symbols; the latter are "top"-level objects in the category; the arrows between kinds represent the dependencies built into the syntax.

The formulation of FOLDS based on one-way categories is simpler than the "direct" approach. In fact, it can be put into a succinct algebraic form, in the form of certain hyperdoctrine-type structures. We will exploit this possibility for the presentation of the Gödel and the Kripke completeness theorems for FOLDS.

2. Let me indicate the foundational motivation behind this work.

P. Benaceraff, in a well-known paper [Ben] entitled "What numbers could not be", expressed a criticism of the set-theoretical reconstruction of mathematical concepts such as that of "natural number". Benaceraff's point is that any one set-theoretical definition of "natural number" gives rise to truths, such as "17 has exactly seventeen members", that become false under an alternative, but equally legitimate set-theoretical definition of "natural number" (his illustration compares the von Neumann definition ($0=\emptyset$, $n+1=n\cup\{n\}$) and the Zermelo definition ($0=\emptyset$, $n+1=\{n\}$). Thus, the set-theoretical reconstruction of mathematics is inevitably cluttered with irrelevant and arbitrary truths.

The way out of this requires a language of mathematics in which one talks about the system $(\mathbb{N}, 0, S)$ of the natural numbers in such a way *that any property of* $(\mathbb{N},0,S)$ *that can be expressed in the language is necessarily invariant under isomorphism* of structures of the form $(A;\ a\in A,\ f:A\to A)$. We quickly realize, as did Benaceraff, that in such a language, we cannot allow an equality predicate relating things belonging to various sets; we may contemplate equality $a=_A a'$ of elements a, a' of a *fixed*, but arbitrary, set A only. As a consequence, we cannot allow an equality predicate whose arguments are sets; for if A and B are sets, $A=B$ should imply that $\forall a\in A.\exists b\in B.a=b$, but the last use of the equality predicate is not restricted to elements of a fixed set!

Doing mathematics under such restrictions is not as absurd as it may sound first. In fact, considering sets to be objects of a category, with functions as arrows, and using the FOLDS language of category theory mentioned above, one may do, specifically in the Lawvere-Tierney theory of elementary toposes with a natural numbers object, a significantly large part of mathematics, without violating the said exclusions, and in fact, fully observing the above-italicized requirement.

One may contemplate a comprehensive language of abstract mathematics, with the property that in it, only "relevant", that is, *suitably invariant*, predicates can be expressed. In the case of properties of sets, "suitably invariant" means "invariant under isomorphism (bijection)". In the contemplated foundational framework, sets are singled out among arbitrary totalities by the quality of a set that an equality predicate on its elements as arguments is present as part of the "structure" of the set. The totality of all sets is not a set, since there is no equality predicate on sets as arguments.

But then, what kind of structure does the totality of all sets form? Answer: a *category*. The isomorphisms will be particular arrows. We quickly realize that, to do set-theory, we need more general arrows than isomorphisms. In category theory, equality of *parallel* arrows is fundamental; we stipulate that the arrows from a fixed object to another fixed object form a *set*. We find that there are other categories, such as that of groups and homomorphisms, which in many ways are similar to that of sets and functions. For instance, we do not want to have equality of groups as a primitive. Categories appear as *generalizations* of sets; every set is a category, a discrete category. There is, in general, no such thing as the "underlying *set* of the objects of a category", not because of size considerations, but rather because, in general, there is no equality predicate whose arguments are the objects of the category.

We find that the idea of an *isomorphism* of categories, let alone equality of categories, is incoherent; it is obvious that the notion of an isomorphism of two categories must involve reference to equality of objects in each of the categories. This entails that a totality of categories cannot be, in general, a category; in any category, the notion of isomorphism is well-defined. For totalities of categories, we must have a new type of structure, some kind of 2-dimensional category.

However, in our quest for the "perfectly invariant" language we quickly get into conflict with standard category theory. The trouble is that we must conclude that the notion of functor, surely a mainstay of the subject, is not acceptable. The problem with it is that it implicitly refers to equality of objects in the codomain category, in the requirement that its value at any given object in the domain category be *uniquely* determined. Is there a way out of this?

In an old paper ([Kel]), G. M. Kelly described a common situation one finds oneself when one wants to define a functor. It appears that all data are there to define the functor, still, it is not possible to canonically single out the value of the functor at an argument-object; one needs to make an arbitrary choice of a value, while it is also clear that it is immaterial what choice one makes. Frequently, the choice cannot be made without the Axiom of Choice. Kelly described in precise terms what the data are like *before* one makes the arbitrary choices. Relatively recently, without knowing about Kelly's paper, I also went through a similar consideration, and made a formal definition of the notion of anafunctor (a term suggested by D. Pavlovic), anticipated by Kelly some thirty years ago (he did not give a name to the concept). (Related ideas occurred to R. Paré some time ago.) I have found that one can live, quite well actually, with anafunctors, without converting them into functors by making non-canonical choices. There is a basic category theory that, in its main outline, does not deviate too much from the standard one, and which uses anafunctors in place of functors; this theory gets by to a large extent without the Axiom of Choice. The beginnings of anafunctor theory is presented in [M2].

Let me emphasize that the work in [M2] is done in a traditional set-theoretic framework. The "perfectly invariant" foundation is not yet available for use; the mathematical work in [M2] is intended to help formulate such a foundation.

I envisage a foundational set-up, a *universe* of abstract concepts, in which we have sets, functions, categories and *anafunctors* as specific distinct kinds of entities. It is clear that we cannot stop here. We will have *natural transformations* of anafunctors. But the totality of all categories, anafunctors and natural transformations of the latter will form a new kind of entity, an *anabicategory*. This differs from a bicategory in that each composition operation of 1-cells, one for each triple of objects (0-cells), instead of being a functor, is an anafunctor. [M2] treats the afore-mentioned concepts.

The concepts of anafunctor and anabicategory mentioned above are "non-radical" revisions of established notions of category theory. As Kelly explained in [Kel], using a global version of the Axiom of Choice, anafunctors can be "converted" into functors. Technically, this amounts to saying that, under an appropriate Axiom of Choice, every anafunctor is isomorphic to a functor (this makes sense since a functor is canonically an anafunctor; "anafunctor" is a generalization of "functor"). Thus, under the full force of the usual set-theoretic foundations, anafunctors are of no importance. (Let me mention in this context that the global Axiom of Choice we have in mind is in fact *meaningless* in the Invariant Foundation, since it talks about a function with values which are sets, the very idea of which is inexpressible because of the lack of equality on sets. In fact, Kelly already in *loc.cit.* considered the global type of choice involved here more suspect than ordinary choice.)

The universe of the Invariant Foundation is not clearly defined as yet. It should contain ana-n-categories for all natural n's; the totality of ana-n-categories, with their morphisms, etc., will form an ana-$n+1$-category. The task of formulating these concepts is closely related with the task of defining the general notion of "weak n-category", mentioned in [BD].

3. In the previous subsection, I gave an incomplete outline of the universe of the Invariant Foundation. The contribution of the present work is to the language of that foundation. The proposal is to use FOLDS as the basic language.

For any vocabulary $\mathcal{L}$ for FOLDS, taken (for convenience) completely without equality, I introduce the notion of $\mathcal{L}$-*equivalence* of $\mathcal{L}$-structures; this is the replacement for the notion of isomorphism for ordinary kinds of structure. An $\mathcal{L}$-structure M is at the same time an ordinary structure for an ordinary language $|\mathcal{L}|$; the properties of M expressible in FOLDS are particular ordinary first-order properties of M as an $|\mathcal{L}|$ -structure, but not vice versa. It turns out (General Invariance Theorem, GIT) that the first-order properties that are invariant under $\mathcal{L}$-equivalence are precisely the ones that are expressible in FOLDS over $\mathcal{L}$. This indicates that $\mathcal{L}$-equivalence is the right notion of "isomorphism" for structures for FOLDS.

As was mentioned above, anafunctors are a generalization of functors. But, upon closer look, we see that the requirements of the "logic of (generalized) equality" impose an additional condition on anafunctors. Whereas an anafunctor determines its value at a given argument up to isomorphism, meaning that any two possible values are isomorphic, in the case of a *saturated anafunctor*, the value is determined also *no more* than up to isomorphism, meaning that any object isomorphic to a possible value is also one. (The precise definition also relates to the given isomorphism between a possible value and a new object.) The requirement of saturation is an extension of the principle of substitutability of equal for equal, transferred to isomorphism from equality. Now, it turns out that every anafunctor, in particular every functor, has a canonically defined *saturation*, a parallel saturated anafunctor, to which it is isomorphic. The right notion of "functor" is "saturated anafunctor".

On the one hand, we have traditional types of categorical structures, examples which are (1) categories, (2) diagrams of categories, functors and natural transformations, and (3) bicategories, *etc.*. We have notions of equivalence for each of these kinds; e.g., the one for bicategories is usually called "biequivalence".

On the other hand, we have anaversions of each of the above kinds of structure. In particular, we have a canonical saturation of any structure of each of the above kinds; in case of the first (category), the saturation is identical to the original. Each kind of anastructure has a vocabulary $\mathcal{L}$ for FOLDS as its similarity type; as a result, we have the notion of $\mathcal{L}$-equivalence for these anastructures. The chief point of the work here is that the concept of equivalence for traditional structures of a given kind, and the concept of $\mathcal{L}$-equivalence for their saturations correspond to each other. E.g., two bicategories are biequivalent iff their saturations are $\boldsymbol{L}$-equivalent, where $\boldsymbol{L}$ is the FOLDS vocabulary for anabicategories.

The saturation $\mathcal{A}^{\#}$ of a categorical structure (e.g., bicategory) $\mathcal{A}$ is quite simply defined in terms of $\mathcal{A}$; in particular, the definition is a first-order interpretation. As a result, any first-order property, and in particular, any FOLDS property, of $\mathcal{A}^{\#}$ is also, by a direct translation, a first-order property of $\mathcal{A}$. Hence, it is meaningful to ask of a first-order property P of $\mathcal{A}$ whether it is expressible as a FOLDS property of $\mathcal{A}^{\#}$. We have the conclusion that this holds iff P is invariant under equivalence of the appropriate kind. E.g., a first-order property of a variable bicategory $\mathcal{A}$ is invariant under biequivalence iff it is expressible in FOLDS as a property of the saturation of $\mathcal{A}$. This theorem is a result of a combination of the relation of the two kinds of equivalence mentioned above, and an appropriate generalization of the GIT.

The last result for categories is due to P. Freyd [F], and G. Blanc [B]; Blanc's formulation is closer to the spirit of this work. A detailed proof is available in [FS]. The methods of the present work are entirely different from Freyd's. Restricted to the case of categories, the former give stronger results, although the additional strength that I cannot reproduce by Freyd's methods seems of minor importance. More important is the fact that Freyd's methods employ the axiom of choice, through the use of the skeleton of a category, and thus do not generalize to "constructive category theory". In Appendix E, I give a proof of the GIT for intuitionistic FOLDS. This gives rise to an intuitionistic version of the Freyd-Blanc characterization theorem for properties of categories invariant under equivalence. This does not seem to be accessible by the methods of [FS].

The main mathematical results of the present work are thus syntactic characterizations of formulas that are invariant under equivalence, in various senses of "equivalence". For the statement of these results, there is no need to understand the anaconcepts. In fact, for the case of bicategories, I organized the presentation in a way that does not refer to anabicategories explicitly, although, in this way, I missed the proof of the full strength of the main result. By contrast, in the case of diagrams of categories, functors and natural transformations, the anaconcepts are displayed.

From the foundational point of view, the results give confirmation to the idea that FOLDS employed in the context of anastructures is a suitable foundational language. I expect that the analysis started here will extend with similar results to higher dimensions. This is a concrete matter in the case of tricategories [GPS]; but I believe the case of general n-dimensional structures will soon be accessible too. I find it an interesting proposition, verified up to dimension 2 here, and conjectured to hold in all dimensions, that the appropriate notion of equivalence, "weak n-equivalence" in the terminology of [BD], has a form, namely $\mathcal{L}$-equivalence for the saturations of the structures involved, which is of a general "logical nature"; the original notion of "weak n-equivalence" looks *a priori* to be a rather involved idea.

4. Let me give an overview of the contents. I have organized the material into seven sections and five appendices, with the obvious implication as to what parts I felt to be the more important ones. §1 is the basic introduction to the syntax and semantics of FOLDS. The reader may immediately look at Appendix A, which contains the alternative, "more logical", introduction of FOLDS. §2 contains the formal systems for the classical, intuitionistic and coherent versions of FOLDS. §3 is a purely algebraic (categorical) study of "fibrations with quantification". I deal with hyperdoctrine-like structures; specifically, fibrations in which the base category has finite limits, but there is a distinguished class of arrows along which quantification is allowed. The applications to FOLDS is given in §4. I was surprised at the appropriateness of this simple idea for the purposes of FOLDS. The (Gödel, Kripke) completeness of the systems of §2 are thus seen to be a special case of something much more general.

§5 introduces the concept of $\mathcal{L}$-equivalence, the main new concept of the work, and proves, in a suitably general form, the General Invariance Theorem (GIT). Appendices B and C are elaborations on the theme of $\mathcal{L}$-equivalence. In Appendix C, I give, among others, proofs that follow the spirit of the treatment in [FS]. §§6 and 7 work out the conclusions concerning the three kinds of categorical structure we discussed above. In §6, the example of a single functor between two categories as a categorical structure is considered in some detail. In particular, fibrations are such structures. Appendix D contains some of calculations for §7.

Finally, Appendix E does two main things. One is the extension of the theory of $\mathcal{L}$-equivalence to intuitionistic logic and Kripke models. The other is ordinary Craig interpolation and Beth definability for FOLDS.

I would like to thank George Janelidze and Dusko Pavlovic for valuable conversations on the

subject of this work.