Towards formalizing and extending differential programming using tangent categories

Geoff Cruttwell , Jonathan Gallagher , and Ben MacAdam *

July 1, 2019

In this paper, we put in Part I an extended abstract for work on extending differential programming using the techniques of synthetic differential geometry and tangent categories. In Part II we summarize results to give evidence of the work described in Part I.

Part I
Extended Abstract

This paper gives an interpretation of a simple differential programming language (over finite dimensional $\mathcal{R}$ vector spaces), into a setting derived from synthetic differential geometry (SDG). The main theorem of this paper is Theorem 5.6, where we establish that there is always an interpretation of our simple differential programming language into a category of partial maps of a well-adapted smooth topos that preserves the derivative and all the control structures of the differential programming language. This shows that manifolds and internal homs (for the space of total smooth functions) can be consistently added to differential programming languages. To establish this theorem, we need to combine and extend various pieces of categorical structure related to categorical models of differentiation and partial maps.

In section 1, we introduce the simple differential programming language, which was based on Plotkin's differential programming language, and discuss the places where using partial maps is necessary. In section 2, we introduce the categorical framework we use for partial maps: join restriction categories; in this section we prove Lemma 2.2 which is crucial to this paper – it will allow us to use join restriction structure in the category we build from SDG. In section 3 we introduce the categorical framework we use for categories of smooth, partial maps, called differential restriction categories. In this section, we prove Proposition 3.2 which shows that differential join restriction categories are sufficient for giving a sound interpretation of differential programming that preserves the

*This research was supported in part by NSERC, AARMS, and PIMS.

interpretation of control structures (loops and branching mechanisms) as well as the derivative. We also introduce restriction tangent categories which generalize differential restriction categories, and show how to extend the interpretation of our differential programming language by showing that any tangent and join restriction preserving functor will preserve the intepretation of the language (Proposition 3.3). In section 4, we turn our attention to well-adapted models of SDG, by considering the partial map category of a well-adapted smooth topos, and prove Theorem 4.4. This theorem turns out to be a bit disappointing. While we can extend differential programming into the topos $\mathcal{E}$, we find that the derivative may not be preserved: that is, when we restrict to the microlinear objects, we will not in general have a tangent restriction category, and so we cannot apply Proposition 3.3. In section 5, we refine the partial map category of the smooth topos by considering a more refined notion of partiality, where partial maps are not defined on arbitrary subobjects, but rather formal étale subobjects. Such partial map categories of smooth toposes do not appear to have been studied. We show that we can obtain a join restriction tangent category from this setting (Lemma 5.2), and that we have a situation where Proposition 3.3 can be applied, and this is proven in Theorem 5.6. Thus we have given an intepretation of differential programming into a setting constructed from SDG. Moreover, we now have manifolds as objects, as well as internal homs of smooth total functions. In some models this implies that we have sequence spaces without any finiteness constraints, and thus we can model recurrent neural networks as well.

Differential programming is an emergent programming paradigm that makes use of a derivative that can be applied to any part of a program which represents a smooth function $\mathcal{R}^n \to \mathcal{R}^m$. The derivative can be used to optimize parameters of a program and facilitates deep learning [2] [1]. This generalizes the use of the derivative to optimize and train a neural network as a subroutine, as it allows training parameters of any subroutine. Typically differential programming makes use of automatic differentiation, where the derivative of a function is computed alongside the function, as there are often efficiency increases in doing so.

The methods being used for automatic differentiation are well established, and recently have drawn attention from the programming languages community, who are seeking to understand how a differentiation operator should behave in a programming language. One such investigation was exposited in MFPS 2018 by Plotkin [38], whose work this paper builds on. Plotkin isolated the features of differential programming and defined a simple programming language with `if-then-else` and `while` control constructions and a differential operator baked right into the syntax [1]. To obtain a correctness result, Plotkin provided an interpretation of his language into the category of functions $\mathcal{R}^n \to \mathcal{R}^m$ that are smooth when restricted to an open subset (denoted $\mathsf{Smooth_P}$). Using this interpretation, and an assumption used frequently in the automatic differential programming community that the guards of the control structures are continuous

[1] Plotkin used the reverse derivative, where this paper will use the forward derivative. The reverse derivative may be transformed into the forward derivative by transposition.

[3] [29], Plotkin proved that the tranformations of programs that push the derivative into the control structures `if-the-else` and `while` are sound. Plotkin also pointed out that there is a desire to extend differential programming languages with features that make manifolds into datatypes and that allow function spaces as domains (i.e. allows higher-order functions).

This paper sets out the following path towards understanding how to add manifolds as datatypes and allows for higher-order functions. We first identify the categorical structures used in Plotkin's interpretation into in $\mathsf{Smooth_P}$. Once we have identified these structures, we will construct a a functor $\mathsf{Smooth_P} \to \mathcal{E}$ where $\mathcal{E}$ is a category that has our desired features. By identifying the categorical structure used in $\mathsf{Smooth_P}$ for Plotkin's interpretation, we can then ensure that our functors preserves all the structure needed so that the model in $\mathsf{Smooth_P}$ carries to $\mathcal{E}$. Then we will build such a category $\mathcal{E}$ and a functor $\mathsf{Smooth_P} \to \mathcal{E}$.

First we will identify categorical structure that may be used to interpret `if` and `while`, and for this we will use join restriction structure. Partiality plays a crucial role to the interpertation of `if` and `while`. While loops may fail to terminate, and hence can provide an immediate source of partiality. The assumption of continuity imposed on guards forces another source of partiality: as guards are valued in $\{\mathtt{T}, \mathtt{F}\}$ with the discrete topology, for a guard to be continuous is to require that the preimages of `T` and `F` be disjoint open sets; thus the guard must be partial or trivial.

The categorical structure we will use for `if` and `while` is join restriction structure. Restriction categories were introduced in [11] as abstract categories of partial maps. Categories of partial maps play a major role in topos theory; indeed, a topos can be defined as a Cartesian closed category with finite limits and a partial map classifier. In Set a partial function $A \xrightarrow{f} B$ is a total function on some subset $A' \subseteq A$. More generally, in any topos a partial map $A \xrightarrow{f} B$ is a span $A \xleftarrow{m} A' \xrightarrow{f} B$ where m is a monic (up to equivalence), and composition is pullback. It was then realized that the restriction to a topos and all monics is not necessary; a category of partial maps makes sense with respect to any class of monics $\mathcal{M}$ that is closed to composition and pullback (called dominions [39] and domain structures [36]). Such categories of partial maps are always partial order enriched (the ordering being essentially, more defined), and the isolation of the order for understanding partiality was used in [17]. However the order used by [17] is not enough to ensure that there is an underlying category of partial maps. The categories using partial orders to formalize partiality were a generalization of an earlier attempt using [37] called dominical categories. Dominical categories realized the domain of definition of a map $A \xrightarrow{f} B$ as an idempotent $A \xrightarrow{\overline{f}} A$ that is a partial identity and defined iff f is. Restriction categories directly axiomatize the behaviour of these idempotents that represent the domain of a map [11]. Moreover, restriction categories allow for a completeness theorem with respect to categories of partial maps; every restriction category where restriction idempotents split is precisely a category of partial maps, and hence every restriction category fully and faithfully embeds (by the idempotent splitting) into a category of partial maps.

It is no surprise that restriction categories are connected to logic, due to their ties to topos theory. Every restriction categories comes equipped with a fibration into meet semilattices. An extension of this small fragment of logic to distributive meet-join lattices is given by join restriction categories [25] and [14]. Join restriction categories were used to model iteration of stack machines in [15] and [10], and the approach we give to modelling `while` and `if` is based on the more general approach of modelling iteration with join restriction structure. Thus, we use join restriction categories as a model because they capture exactly the notion of a partial map category, and they have the added benefit of providing a convenient means to express `if` and `while`.

After repackaging the `if` and `while` control structures, we will then introduce the categorical structure used to interpret the derivative, called differential restriction categories [6]. Differential restriction categories are the partial map counterpart to Cartesian differential categories [5]. The story behind using differential categories to model programming languages has precedent.

The differential λ-calculus is a an extension to the λ-calculus introduced by Ehrhard and Regnier [22] to give a syntactic characterization of the additional features that the model of the λ-calculus into Köthe sequence spaces [20] and finiteness modules [21] has. The new feature that the differential λ-calculus introduced was a formal derivative of λ-terms.

The models of the differential λ-calculus came as coKleisli categories of models of linear logic. In [6], Blute et al introduced *monoidal differential categories* to address what additional features that a model of linear logic is required to have in order to interpret the differential λ-calculus. Most importantly, one needs an additively enriched monoidal category, and the linear exponential modality $(!, \delta, \epsilon)$ needs to have a deriving transformation. In this setting, coKleisli maps have a derivative:

$$\frac{!A \xrightarrow{f} B}{A \otimes !A \xrightarrow[D_\otimes[f]]{} B}$$

Which is viewed as

$$!A \xrightarrow{D_\otimes[f]} (A \multimap B)$$

so that the derivative of f at a point is the (best) linear approximation. If the starting category is monoidal closed, then its coKleisli category is a model of the differential λ-calculus.

The characterization of models was then expanded in [5] by Blute et al who introduced *Cartesian differential categories* to characterize the categories that arise as coKleisli categories of monoidal differential categories. In a Cartesian differential category, every map has a derivative:

$$\frac{A \xrightarrow{f} B}{A \times A \xrightarrow[D[f]]{} B}$$

that is linear in its first argument, satisfies the chain rule, and a few other equational properties of the derivative from differential calculus. These capture the category of smooth maps $\mathcal{R}^n \to \mathcal{R}^m$. They also capture the derivative coming from any Fermat theory [19] and the generalization of Fermat theories to Fermat modules as in [4]. Thus, the Fréchet derivative on Fréchet vector spaces gives a Cartesian differential category. The category of convenient vector spaces and smooth maps with the convenient derivative is also an example [7], and further this example is closed, and a model of the differential λ-calculus. More generally the Kock-Lawvere vector spaces of any model of synthetic differential geometry is a Cartesian differential category (see e.g. [12]), and they were proved to be a model of the differential λ-calculus in [23].

Differential restriction categories extend the axiomatization of Cartesian differential categories to the setting of partial maps [6]. The category $\mathsf{Smooth_P}$ is a differential restriction category. The category of rational functions over a ring is a differential restriction category.

Recently, Cartesian differential categories have been directly applied to aspects of differential programming. For example [41] showed how to obtain a Cartesian differential category structure on a category of sequences and what are called causal computations. Another paper in the same spirit uses differentiation for causal computations and relates the structure directly to recurrent neural networks [40]; this paper conjectures that the differential used is a differential restriction category. Plotkin also mentioned that differential restriction categories may provide a semantics for his language [38].

The interpretation of a differential programming language into $\mathsf{Smooth_P}$ is done in a way that the derivative in the language is interpreted into the differential restriction structure. Thus, we will show that the interpretation of a simple differential programming language into $\mathsf{Smooth_P}$ can be repackaged in a way that only the differential join restriction structure on $\mathsf{Smooth_P}$ is used.

To do this, we will introduce restriction categories, restriction categories with partial products, and join restriction categories. We will show that `if-then-else` and `while` may interpreted using join restriction categories. Then we will introduce differential restriction structure, and show that derivatives may interpreted into the differential restriction structure on $\mathsf{Smooth_P}$.

Our goal is to begin an exploration of extending differential programming languages by extending the interpretation of Plotkin's language into a setting with manifolds and function spaces, as Plotkin had indicated that both of these features would be highly desirable. By understanding how the additional features of manifolds and function spaces interact with derivatives, we may be able to provide insight into the subtleties involved with extending differential programming languages with these features.

For the current paper, we will focus on internal homs of smooth partial maps. For manifolds, a construction of [24] has given a general construction of manifolds for any join restriction category. For a differential join restriction category, it was proved that the manifolds formed from a differential join restriction category always yields a related structure called a join restriction tangent category, and there is an embedding of the starting category into the category of manifolds that preserves derivatives (actually the more general result was proved that one can form manifolds out of any join restriction tangent category) [12].

The structure and behaviour of hom-objects between smooth spaces has been studied extensively in sythetic differential geometry [30]. In synthetic differential geometry (SDG) one makes use of a topos with additional features that give it a *representable tangent bundle functor* $[D, \_] : \mathcal{E} \to \mathcal{E}$. The tangent spaces of this tangent functor do not always form internal modules, and are not guaranteed to have tangent vector addition for example. One restricts to a full subcategory called the microlinear spaces of $\mathcal{E}$. The category of microlinear spaces is closed to limits and exponents, and when $\mathcal{E}$ is a Grothendieck topos, microlinear spaces are a locally presentable, cartesian closed category. In what is called a well-adapted model, $\mathcal{E}$ admits the category of smooth manifolds as a full subcategory of microlinear spaces in a way that transverse limits are preserved by the inclusion [18], and the construction of manifolds is preserved [31].

We will thus use well-adapted models of SDG to accomplish our goal of extending the interpretation of the differential programming language into a setting with function spaces and manifolds. However, Plotkin's language uses partial maps; thus, we must investigate the partial map category of a smooth topos. The study of the partial map category of a smooth topos appears to be new, and potentially a bit surprising. In a well-adapted topos, SMan arises as a full subcategory. However, when one moves to the partial map category, SMan admits only a faithful embedding. This is because SDG admits new subobjects of $\mathscr{R}$, and this means there are more partial maps from $\mathscr{R} \to \mathscr{R}$ in the partial map category of $\mathcal{E}$ than there were in SMan. We will nonetheless give a faithful restriction functor that preserves derivatives from $\mathsf{Smooth}_\mathsf{P}$ into the partial map category between microlinear spaces of a well-adapted model.

To facilitate the above move we will introduce the notion of cartesian join restriction tangent category. Every differential join restriction category is an instance of one of these. Then we will show that the partial map category between microlinear spaces of a smooth topos is one. Finally, we will prove the existence of a faithful functor carrying $\mathsf{Smooth}_\mathsf{P}$ into the partial map category of the microlinear spaces of $\mathcal{E}$, in a way that preserves cartesian restriction tangent structure, and hence it follows that the differential restriction structure is also preserved. To make this precise we will recall the notion of a restriction tangent category and some basic facts about them proved in [12]. The partial map category of the topos is proved to be a partial cartesian closed category, using the fact that toposes have partial map classifiers. We introduce a condition for when the partial map category of microlinear spaces is a partial cartesian closed category.

However, in SDG, one is often not concerned with arbitrary subobjects, but rather *formally étale* subobjects. If $U \subseteq \mathcal{R}^n$ is an open embedding in $\mathsf{Smooth}_\mathsf{P}$, then the inclusion into a well-adapted model is a formally étale subobject of $\mathscr{R}^n$. The slogan is that the formally étale subobjects are the "well behaved" subobjects. To make this precise we will use the partial map category with respect to étale monics. We will see that unlike the case for all monics, the partial map category of microlinear spaces with respect to étale monics is a full subcategory of the partial map category of the starting topos. We will show that there is a faithful embedding of $\mathsf{Smooth}_\mathsf{P}$ into the étale subobject partial category; however, we still do not get a full embedding. To ensure that étale monics are closed to taking joins we have to include an additional axiom from SDG: the amazing right adjoint axiom. The embedding of $\mathsf{Smooth}_\mathsf{P}$ into this partial map category also has the property that it preserves the construction of manifolds. However, function spaces become more subtle. In this setting, we do have spaces that correspond to total smooth functions, for example total smooth maps $[\mathbb{N}, \mathscr{R}]$, but having function spaces of partial smooth maps is equivalent to asking for $\mathcal{E}$ to have an étale subobject classifier. We hope to better understand étale subobjects classifiers in the future, and leave this for future work.

There are settings other than SDG that extend the category of smooth manifolds to have function spaces. One such category of generalized smooth spaces is the category of diffeological spaces. Baez and Hoffnung proved that diffeological spaces are a category of concrete sheaves on the concrete site of SMan. Kammar, Staton, and Vakar proposed an interpretation using diffeological spaces and polynomial functors [28], and thus making use of the locally cartesian closed structure of diffeological spaces. There is also a notion of tangent space, and hence tangent bundle for diffeological spaces [8][26]. However, the tangent space, like in SDG, does not always have addition [33]. Defining a notion of microlinearity for diffeological spaces seems possible, but may require additional work to get off the ground, so for the present work we start with SDG. However, Kammar, Staton and Vakar made use of a category of ω-CPOs as a category of models of an essentially algebraic theory, and we hope to consider this further in future work, as this part does not rely on locally cartesian closed structure.

Part II
Results

1 Plotkin's Language

In this section we describe the syntax for a language for differential programming in the spirit of Plotkin's MFPS 2018 talk [38]. This language has a single generating type, the reals $\mathcal{R}$, and is closed under product types $\mathcal{R}^n$, and importantly we have if-then-else and while-loops. We will parametrize the language by a set of function symbols Σ so that we can add function symbols as needed (e.g. $\sin, e^x$,etc).

The language $\mathcal{L}$ is described by the following grammar

$$\begin{aligned}
\mathcal{L} &:= \mathsf{FunDef}^* \\
\mathsf{FunDef} &:= \mathsf{Ident}\,(\,\mathsf{Ident}^*\,)\ :=\ \mathsf{Term} \\
M &:= x \in \mathsf{Var} \mid r \in \mathcal{R} \mid \sum_{i=1}^{n} M \mid f(M, \cdots, M)\ f \in \Sigma \\
&\mid (M, \cdots, M) \mid \mathtt{let}(x_1, \ldots, x_n) =\ M\, \mathtt{in} M \\
&\mid \mathtt{if}\ M\ \mathtt{then}\ M \text{ else } M \mid \mathtt{while}\ M \,.\, M \\
&\mid \frac{\partial M}{\partial(x_1, \ldots, x_n)}(M) \cdot M
\end{aligned}$$

Subject to the typing rules for a functional language and typing rules for `if-then-else`, `while`, and derivatives in Table 1. We will not develop an operational semantics here.

We do however need to point out the difference in syntax we use for `if-then-else` and `while` regarding the guards. The `if-then-else` construct in differential programming can be nasty if not used carefully: a differentiable function may be deconstructed using `if-then-else` in way that the pieces have poorly behaved differentials. For example:

$$f(x) = x^2$$

maybe decomposed as

$$f(x) = \begin{cases} x^2 & x \neq 0 \\ x & x = 0 \end{cases}$$

but the derivative of f cannot be recreated by the derivatives of the pieces.

Usually $b(x)$ is seen to take values in $\{T, F\}$, but allowing arbitrary logical connectives over $\mathcal{R}$ to build such an element is what can lead to problems. One way around this is to limit logical connectives to one use of a $<$, as done in [29]. A more generous solution is to require that $b : \mathcal{R}^n \longrightarrow \{T, F\}$ be a continuous function when $\{T, F\}$ is given the discrete topology [3]. This is equivalent to asking that both $b^{-1}(T)$ and $b^{-1}(F)$ be open sets in $\mathcal{R}^n$. As b is a function it also means that these subdomains be disjoint. However, this then is equivalent to asking that b be a union of b_1, b_2 where $b_1, b_2 : \mathcal{R}^n \longrightarrow \{T, F\}$ have disjoint domains and $b_1(x)$ is either undefined or equals T and $b_2(x)$ is either undefined or equals F. The only place that predicates are used in this language is for guards on `if` and `while` statements, so their use is to determine open subobjects. This can be performed equally well by asking for a union of two partial continuous functions $b_1', b_2' : \mathcal{R}^n \longrightarrow 1$ where $b_1'(x) = \star$ iff $b_1(x)$ is defined and equals T and $b_2'(x) = \star$ iff $b_2(x)$ is defined hence equals F.

We write the kind of union we have in mind as $b_1' \oplus b_2'$. For example

$$\mathtt{if}(x > 1) \oplus (x < -1)\,\mathtt{then}\, f(x)\,\mathtt{else}\, g(x)$$

We are motivated towards this syntax because in a restriction category a terminal object may only have one total map into it, but many partial maps.

$$\frac{\Gamma \vdash b_1, b_2 : 1 \quad \Gamma \vdash f : A \quad \Gamma \vdash g : A}{\Gamma \vdash \texttt{if}\, b_1 \oplus b_2\, \texttt{then}\, f\, \texttt{else}\, g : A} \qquad \frac{p : A \vdash f : A \quad p \vdash b_1, b_2 : 1}{p \vdash \texttt{while}\, b_1 \oplus b_2 \,.\, f : A}$$

$$\frac{\Gamma, p : A \vdash f : B \quad \Gamma \vdash a : A \quad \Gamma \vdash v : A}{\Gamma \vdash \frac{\partial f}{\partial p}(a) \cdot v : B}$$

Table 1: Typing rules for $\mathcal{L}$

Indeed, maps into the terminal object are subobjects. Operationally, this may be implemented by performing $b_1'(x)$ and $b_2'(x)$ at the same time, since at most one of them wil be defined, and then choosing an action based on which one (if either) returns an answer.

2 Join Restriction Categories

In this section we introduce join restriction categories. Join restriction categories provide sufficient structure for interpreting the if-then-else statement and while-loops as done in $\mathcal{L}$. We will prove a lemma that will allow us to put a join restriction structure on $\mathsf{Par}(\mathsf{Microl}(\mathcal{E})), \mathcal{M})$ where $\mathcal{M}$ is the collection formal étale monics in the category of microlinear spaces of a model of SDG. This will be used to give a faithful model of $\mathcal{L}$ in the partial map category of the "good" objects in a model of SDG.

Definition 2.1. *A category $\mathbb{X}$ has* **restriction structure** *[11] there is an operation on maps $\mathbb{X}(A, B) \xrightarrow{\overline{(\_)}} \mathbb{X}(A, A)$ such that*

[R.1] $\overline{f}\, f = f$;

[R.2] $\overline{f}\, \overline{g} = \overline{g}\, \overline{f}$;

[R.3] $\overline{f}\, \overline{g} = \overline{\overline{f}\, g}$;

[R.4] $h\overline{f} = \overline{hf}\, h$.

Restriction categories model categories of partial maps: the restriction $\overline{f}$ of f is an idempotent that picks out the domain of definition of a map f. The link to categories of partial maps is as follows: a restriction category is **split** when for every $e = \overline{e}$, there is a section retraction pair $sr = 1$ and $rs = e$ that splits e. There is a 2-equivalence of categories between split restriction categories and partial map categories [11].

Every restriction category is partial order enriched. The **restriction order** on maps $f \leq g$ is defined precisely when $\overline{f}\, g = f$. Intuitively, this says that g restricted to the domain of f is f, but g may be more defined.

Compatibility provides another relation on maps in a restriction category. We write $f \smile g$ when $\overline{f}\, g = \overline{g}\, f$, and say that f, g are **compatible**. Intuitively this says that whenever both f and g are defined, they are equal. This relation is not an equivalence relation; however, it is reflexive and symmetric.

Suppose $\{f_i\}_{i\in I}$ is a family of pairwise compatible maps. Then the **join** of the f_i is a map $\bigvee_{i\in I} f_i$ such that

- $f_j \leq \bigvee_i f_i$ for all j;
- if each $f_j \leq v$ then $\bigvee_i f_i \leq v$;
- $h(\bigvee_i f_i)k = \bigvee_i (hf_ik)$.

We often write the join of two maps as $f \vee g$ and the join of the empty set as $\emptyset$. Note that $\emptyset$ is an absorbing element, and is the bottom of the restriction ordering; it behaves like the nowhere defined map.

A functor $\mathbb{X} \xrightarrow{F} \mathbb{Y}$ of the underlying categories of restriction categories is a **restriction functor** when $F(\overline{g}) = \overline{F(g)}$. A restriction functor between join restriction categories is a **join restriction functor** when $F(\bigvee_i g_i) = \bigvee_i F(g_i)$.

Lemma 2.2. *Let $\mathbb{X}$ be a restriction category and $\mathbb{Y}$ a join restriction category, and*

$$\mathbb{X} \xrightarrow{J} \mathbb{Y}$$

a full and faithful restriction functor. Then $\mathbb{X}$ is a join restriction category and J preserves joins.

Proof. Suppose $\{f_i\}_i$ is a family of pairwise compatible maps $A \xrightarrow{f_j} B$ in $\mathbb{X}$. We will show that the join exists. Note that any restriction functor preserves compatibility as suppose $h \smile k$ then

$$\overline{Jh}\,Jk = J\overline{h}\,Jk = J(\overline{h}\,k) = J(\overline{k}\,h) = \overline{Jk}\,Jh.$$

Thus $\{J(f_i)\}_i$ is a family of pairwise compatible maps in $\mathbb{Y}$.

Next, compute the join in $\mathbb{Y}$, $\bigvee_i J(f_i)$. As this map is in the same homset as each $Jf_j : JA \longrightarrow JB$, by fullness there is a map $k \in \mathbb{X}(A,B)$ such that $Jk = \bigvee_i J(f_i)$. We will show that k is the join of the f_j.

First we must show that $f_j \leq k$ for all j. But this follows for a more general reason from faithfulness. For any two maps s, t in the same homset in $\mathbb{X}$, $Js \leq Jt$ iff $s \leq t$. Suppose s, t are in the same homset and $Js \leq Jt$. Then $\overline{Js}\,Jt = Js$. But then $Js = \overline{Js}\,Jt = J(\overline{s}\,t)$, and by faithfulness $s = \overline{s}\,t$, hence $s \leq t$. Then for this stronger reason, as $J(f_j) \leq Jk$ we have $f_j \leq k$.

Suppose that v is such that $f_j \leq v$ for all j. But then $J(f_j) \leq J(v)$ for all j. Hence

$$Jk \leq \bigvee_i F(f_i) \leq J(v).$$

and again by the stronger reason above, $k \leq v$.

Finally, we must show the compatibility with composition.

Under our proposal, $\bigvee_i mf_in$ is the unique element that maps to $\bigvee_i J(mf_in)$. It suffices by the faithfulness of J then to show that mkn maps to this under J as well. Note that

$$\begin{aligned} J(mkn) = JmJkJn = J(m) \left(\bigvee_i J(f_i) \right) J(n) \\ = \bigvee_i (JmJ(f_i)Jn) \\ = \bigvee_i J(mf_in) \end{aligned}$$

as desired. Thus, $\bigvee_i (mf_in) = m(\bigvee_i f_i)n$. Thus $\mathbb{X}$ is a join restriction category.

We defined the join in $\mathbb{X}$ such that

$$J(\bigvee_i f_i) = J(k) = \bigvee_i J(f_i)$$

thus J preserves joins. $\square$

Restriction categories often arise as the partial map category of an $\mathcal{M}$-category.

Definition 2.3. *A* **stable system of monics** *on a category* $\mathbb{X}$ *is a class of monics* $\mathcal{M}$ *such that*

1. *M contains all isomorphisms;*
2. *M is closed to composition;*
3. *M is closed to pullbacks.*

We call a category $\mathbb{X}$ *equipped with a stable system of monics* $\mathcal{M}$ *an* $\mathcal{M}$*-category.*

Note that if $m, am \in M$ then a will itself be in m as we have the following pullback diagram

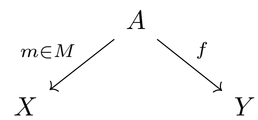
$$\begin{array}{ccc} A & \xrightarrow{a} & A' \\ \| & & \downarrow{\scriptstyle m} \\ A & \xrightarrow[am]{} & B \end{array}$$

Every M-category has an associated restriction category:

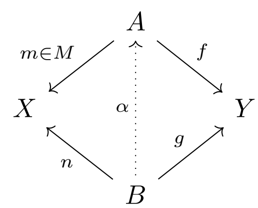
Definition 2.4. *Let* $(\mathbb{X}, M)$ *be an* M*-category. There is a restriction category* $\mathsf{Par}(\mathbb{X}, M)$ *called the* partial map category*.*

- *Objects: Those of* $\mathbb{X}$*.*

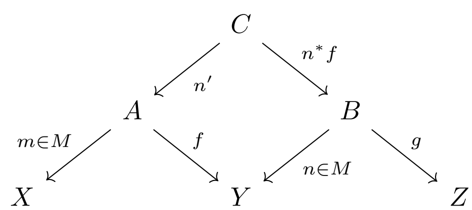
- *Morphisms: Equivalence classes of spans* $(m, f) : X \longrightarrow Y$



where $(m, f) \equiv (n, g)$ *if there is an isomorphism making the following diagram commute:*



- *Composition: Given* $(m, f) : X \longrightarrow Y, (n, g) : Y \longrightarrow Z$*, they compose via the following pullback:*



- *Restriction combinator:* $\overline{(m, f)} = (m, m)$.

Associativity follows from the fact that morphisms are only defined up to isomorphism.

Lin characterized exactly when an $\mathcal{M}$-category is such that $\mathsf{Par}(\mathbb{X}, \mathcal{M})$ is a join restriction category.

Theorem 2.5. *[35] theorem 11 An* $\mathcal{M}$ *category* $\mathbb{X}$ *has joins iff*

1. *For any family of* $\mathcal{M}$*-subobjects* $\{m_i : A_i \longrightarrow A\}$ *the colimit of its matching diagram* $\cup_i A_i$ *exists.*

2. *The induced map* $\vee_i m_i : \cup_i A_i \longrightarrow A$ *is in* $\mathcal{M}$*;*

3. *The colimit of 1. is stable under pullback.*

In a topos all colimits are stable under pullback because the pullback along a map is always a left adjoint between the slice categories. To prove the result one must show that the induced map is in $\mathcal{M}$. Johnstone shows this is the case for any coherent category [27] Theorem 1.4.3. Thus $\mathsf{Par}(\mathcal{E}, \mathsf{Monic})$ is a join restriction category.

3 Interpretation into smooth functions

In this section we will provide an interpretation of $\mathcal{L}$ into the differential join restriction category of smooth maps defined on open subsets of $\mathcal{R}^n$. In this section we will introduce differential join restriction categories, and produce an interpretation of $\mathcal{L}$ into the differential join restriction structure. We will generalize this to tangent restriction categories in the sequel.

A **differential restriction category** is a restriction category with partial products (see [13]) where every object is a total commutative monoid, where the addition on $A \times B$ is given componentwise by addition on A and B, and where there is a differentation operation:

$$\frac{A \xrightarrow{f} B}{A \times A \xrightarrow[D[f]]{} B}$$

that satisfies the following equations (see [9] for more details).

[DR.1] $D[f+g] = D[f] + D[g]$ and $D[0] = 0$;

[DR.2] $\langle g+h, k\rangle\, D[f] = \langle g, k\rangle\, D[f] + \langle h, k\rangle\, D[f]$ and $\langle 0, g\rangle\, D[f] = \overline{gf}\, 0$;

[DR.3] $D[\pi_i] = \pi_0 \pi_i$;

[DR.4] $D[\langle f, g\rangle] = \langle D[f], D[g]\rangle$;

[DR.5] $D[fg] = \langle D[f], \pi_1 f\rangle\, D[g]$;

[DR.6] $\langle \langle g, 0\rangle, \langle h, k\rangle\rangle\, D[D[f]] = \overline{h}\, \langle g, k\rangle\, D[f]$;

[DR.7] $\langle \langle a, b\rangle, \langle c, d\rangle\rangle\, D[D[f]] = \langle \langle a, c\rangle, \langle b, d\rangle\rangle\, D[D[f]]$;

[DR.8] $D[\overline{f}\,] = (1 \times \overline{f}\,)\pi_0$;

[DR.9] $\overline{D[f]} = 1 \times \overline{f}$.

The differential restriction category $\mathsf{Smooth_P}$ has objects $\mathcal{R}^n$ and arrows $(f, U) : \mathcal{R}^n \longrightarrow \mathcal{R}^m$ is an m-tuple of arrows $(f_1, \ldots, f_m)$ each from $\mathcal{R}^n \longrightarrow \mathcal{R}$ and smooth when restricted $U \longrightarrow \mathcal{R}^m$ where U is an open set of $\mathcal{R}^n$.

The category $\mathsf{Smooth_P}$ is a restriction category by

$$\overline{(f, U)}\,(v) = \begin{cases} v & v \in U \\ \uparrow & \text{else} \end{cases}$$

where $\uparrow$ means undefined, and its open set is U. Because domains are assumed to be open, compatible functions are those that agree on the intersection of their domains. Given a family of pairwise compatible maps (f_i, U_i), the join of the function is

$$(\bigvee_i f_i)(x) = \begin{cases} f_i(x) & \exists i. f_i(x) \downarrow \\ \uparrow & \text{else.} \end{cases}$$

This is well defined because of the compatibility assumption - thus this is a join restriction category.

The category $\mathsf{Smooth_P}$ has a cartesian restriction structure, where the restriction terminal is $1 = \mathcal{R}^0$. There are total smooth maps $\mathcal{R}^n \xrightarrow{\pi_i} \mathcal{R}$ and given m maps $(f_i, U_i) : \mathcal{R}^n \longrightarrow \mathcal{R}$ the map $((f_i)_{i=1}^m, \bigcap_i U_i) : \mathcal{R}^n \longrightarrow \mathcal{R}^m$ is unique such that $((f_i)_i, \bigcap_i U_i)\pi_i = (f_i, \bigcap_i U_i) = \circ_{j\neq i}\overline{(f_j, U_j)}\,(f_i, U_i)$, thus giving the category restriction products. The differential restriction structure is defined by the Jacobian:

$$\frac{\mathcal{R}^n \xrightarrow{f} \mathcal{R}^m}{\mathcal{R}^n \times \mathcal{R}^n \xrightarrow[D[f]]{} \mathcal{R}^m}$$

given by $D[f](v,p) := J(f)(p)\cdot v = \frac{\partial f(p)}{\partial p}(p)\cdot v$. The domain is $\mathcal{R}^n \times U$.

Also note that the derivatives are defined equally well "in context" where we only differentiate with repsect to some of the variables:

$$\frac{V \times \mathcal{R}^n \xrightarrow{f} \mathcal{R}^m}{V \times \mathcal{R}^n \times \mathcal{R}^n \xrightarrow[D_V[f]]{} \mathcal{R}^m}$$

Now we define an interpretation $[\![\_]\!]$ of $\mathcal{L}$ into $\mathsf{Smooth_P}$. For types we define $[\![\mathcal{R}^n]\!] := \mathcal{R}^n$. For constants r in $\mathcal{L}$ we interpret as the number r. For each function symbol $f \in \Sigma_n$ we require a smooth map $\mathcal{R}^n \xrightarrow{[\![f]\!]} \mathcal{R}$ defined on some domain U. The interpretation is extended to contexts by $[\![.]\!] := 1$ and $[\![\Gamma, p : V]\!] := [\![\Gamma]\!] \times [\![V]\!]$. We then extend this interpretation inductively to terms by the following rules.

Proj:

- $[\![x : \mathcal{R} \vdash x : \mathcal{R}]\!] := 1_{\mathcal{R}}$;
- $[\![\Gamma, x : \mathcal{R} \vdash x : \mathcal{R}]\!] := [\![\Gamma]\!] \times \mathcal{R} \xrightarrow{\pi_1} \mathcal{R}$;
- $[\![\Gamma, y : \mathcal{R} \vdash x : \mathcal{R}]\!] := [\![\Gamma]\!] \times R \xrightarrow{\pi_0} [\![\Gamma]\!] \xrightarrow{[\![\Gamma \vdash x : R]\!]} \mathcal{R}$.

Cut: We denote $\mathtt{let}\, p = t\, \mathtt{in}\, m$ by $m[t/p]$.

$$[\![\Gamma \vdash m[t/p] : V]\!] := [\![\Gamma]\!] \xrightarrow{\langle 1, [\![\Gamma \vdash t : A]\!]\rangle} [\![\Gamma]\!] \times [\![A]\!] \xrightarrow{[\![\Gamma, p : A \vdash m : V]\!]} [\![V]\!].$$

Flattening:

- $[\![\Gamma, x : 1 \vdash m : B]\!] := [\![\Gamma]\!] \times 1 \xrightarrow{\pi_0} [\![\Gamma]\!] \xrightarrow{[\![\Gamma, \vdash m : B]\!]} [\![B]\!]$;
- $[\![\Gamma, (p,q), \Gamma' : A \times B \vdash m : C]\!]$ is defined to be

$$[\![\Gamma, (p,q) : A \times B, \Gamma']\!] \simeq [\![\Gamma, p : A, q : B, \Gamma']\!] \xrightarrow{[\![\Gamma, p : A, q : B, \Gamma' \vdash m : C]\!]} [\![C]\!].$$

Tuple:

$$[\![\Gamma \vdash (f_1, \ldots, f_m) : A_1 \times \cdots \times A_m]\!] := [\![\Gamma]\!] \xrightarrow{\langle [\![\Gamma \vdash f_i : A_i]\!]\rangle_{i=1}^m} \prod_{i=1}^m [\![A_i]\!]$$

Fun: $[\![\Gamma \vdash f(t_1, \ldots, t_n) : B]\!]$ is defined to be

$$[\![\Gamma]\!] \xrightarrow{\langle [\![\Gamma \vdash t_i : A_i]\!] \rangle} \prod_{i=1}^{m} [\![A_i]\!] \xrightarrow{[\![f]\!]} [\![B]\!]$$

Sums:

$$\left[\!\!\left[\Gamma \vdash \sum_{i=1}^{m} f_i\right]\!\!\right] := \sum_{i=1}^{m} [\![\Gamma \vdash f_i]\!]$$

If-then-else:

$$[\![\Gamma \vdash \texttt{if}\, b_1 \oplus b_2\, \texttt{then}\, f\, \texttt{else}\, g]\!] := \overline{[\![\Gamma \vdash b_1]\!]}\, [\![\Gamma \vdash f]\!] \vee \overline{[\![\Gamma \vdash b_2]\!]}\, [\![\Gamma \vdash g]\!]$$

While:

$$[\![\Gamma \vdash \texttt{while}\, b_1 \oplus b_2\, .f]\!] := \bigvee_{i=0}^{\infty} \left(\left(\overline{[\![q : A \vdash b_1 : 1]\!]}\, [\![q : A \vdash f : A]\!] \right)^i \overline{[\![q : A \vdash b_2 : 1]\!]} \right)$$

Differentials: Suppose $\Gamma, p : A \vdash m : B$, and that $\Gamma \vdash a : A$ and $\Gamma \vdash v : A$. Then we interpret $\left[\!\!\left[\Gamma \vdash \frac{\partial m}{\partial p}(a) \cdot v\right]\!\!\right]$ as

$$[\![\Gamma]\!] \xrightarrow{\langle 1, [\![\Gamma \vdash v : A]\!], [\![\Gamma \vdash a : A]\!] \rangle} [\![\Gamma]\!] \times [\![A]\!] \times [\![A]\!] \xrightarrow{D^R_{[\![\Gamma]\!]} [\![\Gamma, p : A \vdash m : B]\!]} [\![B]\!]$$

Lemma 3.1. *In any differential restriction category:*

1. $\overline{\vee_i f_i} = \vee_i \overline{f_i}$
2. $\langle \vee_i f_i, g \rangle = \vee_i \langle f_i, g \rangle$
3. $\vee_i f_i + \vee_j g_j = \vee_{i,j}(f_i + g_j)$
4. *Restriction idempotents are linear: if* $e = \overline{e}$ *then*
$$D[e] = \pi_0 e.$$
5. *Any join that exists is preserved by the differential*
$$D[\vee_i f_i] = \vee_i D[f_i]$$

Proof. These may be found in [2.14,2.19,3.4,DR.8,3.21] respectively in [9]. □

The following semantic theorem shows what is needed to extend our syntax.

Proposition 3.2. *Any functor*

$$\mathsf{Smooth_P} \xrightarrow{I} \mathcal{E}$$

into a differential join restriction category $\mathcal{E}$ *that preserves restriction, partial products, joins and differential restriction structure preserves the interpretation of* $\mathcal{L}$.

Proof. As restriction and joins are preserved, the interpretation of `if-then-else` and `while` are preserved. As partial products are preserved, so is the monoid structure, hence the interpretation of tuples and sums is preserved. As the derivative is preserved by I and the differential in $\mathcal{L}$ is sent to the derivative, the interpretation of the differential is preserved. □

In the remainder of this section, we will consider a generalization of differential restriction categories called restriction tangent categories.

Tangent restriction categories are introduced in section 6 of [12]. A **tangent restriction category** is a restriction category that has a restriction preserving functor $\mathbb{X} \xrightarrow{T} \mathbb{X}$ and total natural transformations:

$$T \xrightarrow{p} 1 \qquad 1 \xrightarrow{0} T \qquad T_2 \xrightarrow{\sigma} T \qquad T \circ T \xrightarrow{c} T \circ T \qquad T \xrightarrow{l} T \circ T$$

where T_2 is the restriction pullack of p along p. This data is subject to certain coherences (see [12]). In a join restriction category $\mathbb{X}$ if $\mathbb{X}$ is also a restriction tangent category, then the tangent functor preserves the join: $T(\vee_i f_i) = \vee_i T(f_i)$ [12] Proposition 6.15.(ii). Thus, the restriction tangent functor on $\mathsf{Par}(\mathsf{Microl}(\mathcal{E}), \mathcal{M})$ preserves the joins.

Proposition 3.3. *Any functor*

$$\mathsf{Smooth_P} \xrightarrow{I} \mathcal{E}$$

into a Cartesian join restriction tangent category $\mathcal{E}$ that preserves join restriction structure and cartesian tangent restriction structure preserves the interpretation of $\mathcal{L}$.

Proof. By Proposition 3.2 it suffices to show that the image of $\mathsf{Smooth_P}$ under I is a differential restriction subcategory of $\mathcal{E}$,

In [12] was proved the differential restriction objects of a Cartesian restriction tangent category form a differential restriction category Proposition 6.18. Any functor that preserves Cartesian tangent restriction structure preserves differential restriction object structure, and moreover, the derivative between differential objects is preserved.

In $\mathsf{Smooth_P}$, as it is a differential restriction category, regarded as a restriction tangent category, every object is a differential restriction object, hence as the functor preserves Cartesian restriction tangent structure, it preserves the differential restriction structure.

Then this result follows from Proposition 3.2. □

4 First Extension to SDG

In this section we introduce a basic extension of $\mathsf{Smooth_P}$ to the partial maps of a smooth topos, and give a way to extend this to the partial maps between microlinear spaces.

We will introduce the notion of a partial cartesian closed category [36]. We will then recall that the partial map category of a smooth topos $\mathcal{E}$ is a partial cartesian closed category. We will then recall why the partial map category of a topos is a join restriction category. We will then show that the partial map category of the microlinear spaces of a smooth topos, denoted $\mathsf{Par}(\mathsf{Microl}(\mathcal{E}))$, is a cartesian restriction category. Finally, we will show that there is a faithful, Cartesian join restriction functor $\mathsf{Smooth}_\mathsf{P} \to \mathsf{Par}(\mathsf{Microl}(\mathcal{E}))$ whenever $\mathcal{E}$ is additionally well-adapted. We record a pair of definitions regarding M-categories:

- An $\mathcal{M}$-category $(\mathbb{X}, \mathcal{M})$ is classified when there is a monad M such that $\mathbb{X}(A, M(B)) \simeq \mathsf{Par}(\mathbb{X}, \mathcal{M})(A, B)$.
- An $\mathcal{M}$-category $(\mathbb{X}, \mathcal{M})$ where $\mathbb{X}$ has products is called a **partial cartesian closed category** when the functor

$$\mathbb{X} \xrightarrow{A \times \_} \mathbb{X} \to \mathsf{Par}(\mathbb{X}, \mathcal{M})$$

has a right adjoint for each A.

Observation 4.1. *If $(\mathbb{X}, \mathcal{M})$ is an $\mathcal{M}$-category where $\mathbb{X}$ is a cartesian closed category and it is classified, then it is a partial cartesian closed category.*

Proof. Note that $\mathsf{Par}(\mathbb{X}, \mathcal{M})$ is always a Cartesian restriction category whenever $\mathbb{X}$ has products. Suppose $(\mathbb{X}, \mathcal{M})$ is classified by M.

The right adjoint to the functor

$$\mathbb{X} \xrightarrow{A \times \_} \mathbb{X} \to \mathsf{Par}(\mathbb{X}, \mathcal{M})$$

is

$$\mathsf{Par}(\mathbb{X}, \mathcal{M}) \xrightarrow{[A, M(\_)]} \mathbb{X}$$

Indeed

$$\mathsf{Par}(\mathbb{X}, \mathcal{M})(A \times B, C) \simeq \mathbb{X}(A \times B, MC) \simeq \mathbb{X}(B, [A, MC]).$$

□

Recall the following result from Mulry.

Proposition 4.2 ([36])**.** *Every topos has a partial map classifier for $\mathcal{M}$ the class of all monics. Thus, the partial map category of a topos is a partial cartesian closed category.*

Next, when we take $\mathcal{M}$ to be the class of monics in a topos $\mathcal{E}$, $\mathsf{Par}(\mathbb{X}, \mathcal{M})$ is always a join restriction category.

Lemma 4.3. *Let $\mathcal{E}$ be a topos, and $\mathcal{M}$ the class of monics in $\mathcal{E}$. Then $\mathsf{Par}(\mathcal{E}, \mathcal{M})$ is a join restriction category.*

Proof. Lin's theorem 2.5 [35] gives the conditions required for joins. Conditions 1. and 3. hold: as $\mathcal{E}$ is a topos, all colimits exist, and the pullback functor is cocontinuous. Johnstone 1.4.3 [27] shows that the induced map

$$\vee_i : \cup_i A_i \longrightarrow A$$

is monic. □

However, now we have to be careful. $\mathsf{Par}(\mathsf{Microl}(\mathcal{E}), \mathcal{M})$ is not known to be join restriction category: we cannot use Lemma 2.2 because in general $\mathsf{Par}(\mathsf{Microl}(\mathcal{E}), \mathcal{M})$ is not known to be a full subcategory of $\mathsf{Par}(\mathcal{E}, \mathcal{M})$. Indeed, in general a microlinear object can have non-microlinear subobjects. We will fix this in the next section by considering a better class of monics. We still can exhibit tangent structure in this section, and obtain a faithful, restriction preserving functor from $\mathsf{Smooth_P}$; we just will not preserve the interpretation of `if` and `while` until we consider étale monics.

For the last piece of this section, we must introduce the notion of a well-adapted model of SDG, so that we can exhibit a faithful cartesian restriction tangent functor

$$\mathsf{Smooth_P} \longrightarrow \mathsf{Par}(\mathsf{Microl}(\mathcal{E}), \mathcal{M}).$$

A smooth topos $\mathcal{E}$ with specified ring of line type $\mathscr{R}$ is well adapted model [30] when

[WAM.1] There is a full and faithful inclusion $\mathsf{SMan} \xrightarrow{\iota} \mathcal{E}$;

[WAM.2] The inclusion has $\iota(\mathcal{R}) = \mathscr{R}$;

[WAM.3] For any transverse limit $\mathsf{lim}_i M_i$ in SMan, $\iota(\mathsf{lim}_i M_i) \simeq \mathsf{lim}_i \iota(M_i)$;

[WAM.4] ι takes open coverings of M to jointly epic families of maps into $\iota(M)$.

There is a fifth axiom that is ofen additionally used. **[WAM.5]** says that for every Weil algebra U, $[\mathsf{Spec}(U), \_]$ is a left adjoint [2].

Now we begin building the interpretation of $\mathsf{Smooth_P}$ into $\mathsf{Par}(\mathsf{Microl}(\mathcal{E}), \mathcal{M})$.

Theorem 4.4. *If $\mathcal{E}$ is a well-adapted model of SDG, then there is a faithful cartesian restriction functor*

$$\mathsf{Smooth_P} \xrightarrow{\iota_P} \mathsf{Par}(\mathsf{Microl}(\mathcal{E}), \mathcal{M})$$

Proof. $U \subseteq \mathcal{R}^n$ is an open embedding hence a submersion. Then the inclusion is a monic, and hence may be characterized by the fact that the pullback along itself is the domain of the monic. But since it is a submersion, this pullback is transverse in SMan and hence preserved by ι. Thus

$$\iota(U) \longrightarrow \mathscr{R}^n$$

[2] Its right adjoint is often called amazing.

is monic.

We then define the functor on objects $\iota_P(\mathcal{R}^n) := \mathscr{R}^n$. On arrows:

$$\iota_P((f,U):\mathcal{R}^n \to \mathcal{R}^m) := \begin{array}{ccccc} & & \iota(U) & & \\ & \swarrow & & \searrow^{\iota(f)} & \\ \mathscr{R}^n & & & & \mathscr{R}^m \end{array}$$

Again we use that an open inclusion $U \to \mathcal{R}^n$ is a submersion. Thus, the pullback of any map along it is transverse. As ι preserves transverse pullbacks, $\iota(f^*U) \simeq \iota(f)^*\iota(U)$. Therefore, ι_P preserves composition, and is a functor. That it is a restriction functor is immediate. It is faithful because ι is.

Products are transverse pullbacks so ι preserves products. □

Thus we have proved that $\mathsf{Smooth_P}$ may be extended by a faithful, cartesian restriction preserving functor into a cartesian restriction category that contains manifolds. Also this functor lands equally well in $\mathsf{Par}(\mathcal{E},\mathcal{M})$ which always has, additionally, partial function space objects. However, we have not shown that $\mathsf{Par}(\mathsf{Microl}(\mathcal{E}),\mathcal{M})$ retains these partial function space objects. We give a condition that characterizes when $\mathsf{Par}(\mathsf{Microl}(\mathcal{E}),\mathcal{M})$ is a partial cartesian closed category.

Proposition 4.5. *Suppose the partial map classifier M has the property that if X is microlinear, then MX is microlinear. Then $\mathsf{Par}(\mathsf{Microl}(\mathcal{E}),\mathcal{M})$ is a partial cartesian closed, join restriction category.*

Proof. $\mathsf{Microl}(\mathcal{E})$ is an exponential ideal of $\mathcal{E}$; that is, if X is microlinear, then $[A,X]$ is microlinear for any A. Since MX is microlinear whenever X is, we have that

$$[A, MX]$$

is microlinear for every object A. But then

$$\mathsf{Par}(\mathsf{Microl}(\mathcal{E}),\mathcal{M})(A\times B, X) \simeq \mathsf{Microl}(\mathcal{E})(A\times B, MX) \simeq \mathsf{Microl}(\mathcal{E})(B,[A,MX])$$

completing the proof. □

However, such a situation may be hard to obtain due to the following consequence.

Proposition 4.6. *If MX is microlinear whenever X is, then the subobject classifier is microlinear.*

Proof. Consider $M1$: as 1 is microlinear then so is $M1$.

Now,

$$\mathsf{Microl}(\mathcal{E})(A, M1) \simeq \mathsf{Par}(\mathsf{Microl}(\mathcal{E}),\mathcal{M})(A,1) \simeq \mathcal{O}(A)$$

where $\mathcal{O}(A)$ is the set of restriction idempotents on A. Restriction idempotents on A are precisely the same as subobjects of A. □

5 Second Extension to SDG

As mentioned in the introduction, in SDG, it is often desired to use étale monics, over arbitrary subobjects. In this section we form a partial map category using étale monics instead of all monics, and show that the steps taken above may be taken with one additional axiom needed: **[WAM.5]**. We will use the amazing right adjoint to prove that joins of étale subobjects are again étale subobjects. We will then refine the main theorem of the previous section, when we take étale monics as our class of monics, the join restriction category $\mathsf{Par}(\mathsf{Microl}(\mathcal{E}), \mathcal{M})$ is a restriction tangent category.

We will first introduce the notion of being étale with respect to a class of maps $\mathcal{D}$. Then in a smooth topos $\mathcal{E}$ we will define étale maps relative to a particular set of maps: the 0 elements of the objects $\mathsf{Spec}(U)$ for all Weil algebras U. After this we will provide lemmas that show the steps taken in the previous section are closed to étale maps, and we will conclude that there is an extension of $\mathsf{Smooth}_{\mathsf{P}}$ into the partial map category of microlinear spaces with respect to étale subobjects.

Let $\mathcal{E}$ be any category with products, and $\mathcal{D}$ a class of maps. A map $A \xrightarrow{f} B$ is **$\mathcal{D}$-étale** when for every $j \in \mathcal{D}$ and every square of the form:

$$\begin{array}{ccc} J \times X & \xrightarrow{h} & M \\ {\scriptstyle j \times X}\downarrow & \overset{\exists !. v}{\nearrow} & \downarrow{\scriptstyle f} \\ J' \times X & \xrightarrow[k]{} & N \end{array}$$

if the outer square commutes then there is a unique v making the two triangles commute.

When the category is cartesian closed, it is straightforward to show that by "currying" the J and J' in the above, $\mathcal{D}$-étaleness can be rephrased by asking the following diagram to be a pullback ([31] diagrams 1.2 and 1.3).

$$\begin{array}{ccc} [J', M] & \xrightarrow{[J', f]} & [J', N] \\ {\scriptstyle [j, M]}\downarrow & \lrcorner & \downarrow{\scriptstyle [f, N]} \\ [J, M] & \xrightarrow[{[J, f]}]{} & [J, N] \end{array}$$

It follows for general reasons that $\mathcal{D}$-étale maps form a right orthogonality class with respect to the generating class $j \times 1$ for $j \in \mathcal{D}$. Thus, $\mathcal{D}$-étale maps are stable under pullback and closed to composition, transfinite composition, and non-empty products. If $\mathcal{E}$ is locally presentable, for example a sheaf topos, then one can use the small object argument to generate a factorization system, but pullback stability only requires liftings. It is also easy to show that if we take the maps L that are orthogonal to every $\mathcal{D}$-étale map then the intersection of L and $\mathcal{D}$-étale is the class of isomorphisms, and again, we do not require factorizations to prove this. See Appendix A.

Let $\mathcal{E}$ be a smooth topos, and consider the pointed objects $1 \xrightarrow{0} J$ where $J = \mathsf{Spec}(U)$ for some Weil algebra U, and let $\mathcal{D}$ be the collection of these base points. Then a map is **formal étale** when it is $\mathcal{D}$-étale for this class of maps.

Lemma 5.1. *Let $\mathcal{E}$ be a representable tangent category. Then formal étale monics are a stable system of monics.*

Proof. Every isomorphism is monic, and and formal étale maps contain all the isomorphisms as per above. Also the property of being monic and formal étale is both closed under composition and stable under pullback. Thus formal étale monics are a stable system of monics. $\square$

For the remainder of this section, we will let $\mathcal{M}$ denote the class of formal étale monics.

Lemma 5.2. *Let $\mathcal{E}$ be a topos model of SDG with the amazing right adjoint* **[WAM.5]** [3]*. Then $(\mathcal{E}, \mathcal{M})$ is a geometric $\mathcal{M}$-category; that is, $\mathsf{Par}(\mathcal{E}, \mathcal{M})$ is a join restriction category.*

Proof. We wish to show that $\mathsf{Par}(\mathcal{E}, \mathcal{M})$ is a join restriction category. Since we are in a topos, we still know that 1. and 3. From 2.5 [35] we must show that the $\vee_i m_i$ is an étale monic to establish 2. holds.

Let $J = \mathsf{Spec}(U)$. Then we must show that

$$\begin{array}{ccc} [J, \cup_i A_i] & \xrightarrow{[J, \vee_i m_i]} & [J, A] \\ {\scriptstyle p}\downarrow & & \downarrow{\scriptstyle p} \\ \cup_i A_i & \xrightarrow[\vee_i m_i]{} & A \end{array}$$

is a pullback using the alternate characterization of formal étale.

Since we have assumed **[WAM.5]** has the amazing right adjoint, then we know that $[J, \_]$ is a left adjoint, hence cocontinuous. Rewrite $[J, \cup_i A_i]$ as $\cup_i [J, A_i]$ using the fact that $[J, \_]$ is cocontinuous. We also then have the map across the top can be rewritten as $\vee_i [J, m_i]$. But then we have a colimit of a diagram of pullbacks along a single map p, and pullback along p is is cocontinuous because we are in a topos, hence the colimit of the pullbacks is the pullback of the colimits. Thus, the diagram is a pullback.

This then tells us that $\vee_i m_i \in \mathcal{M}$ once all the $m_j \in \mathcal{M}$. Therefore, $\mathsf{Par}(\mathcal{E}, \mathcal{M})$ is a join restriction category. $\square$

As microlinear spaces are a complete category, and closed to limits in $\mathcal{E}$, they inherit a stable system of monics by the formal étale monics between microlinear spaces. In fact, formal étale monics give a join restriction structure on $\mathsf{Par}(\mathsf{Microl}(\mathcal{E}), \mathcal{M})$.

[3]According to Johnstone 1.4.3, we could alternatively work in a Cartesian closed, coherent category with the amazing right adjoint

Lemma 5.3. *For any smooth topos* $\mathcal{E}$*, the category* $\mathsf{Par}(\mathsf{Microl}(\mathcal{E}), \mathcal{M})$ *is a join restriction category. There is a full and faithful join restriction embedding*

$$\mathsf{Par}(\mathsf{Microl}(\mathcal{E}), \mathcal{M}) \longrightarrow \mathsf{Par}(\mathcal{E}, \mathcal{M}).$$

Proof. First we show that there is a full and faithful restriction embedding. The embedding sends a span (m, f) in $\mathsf{Par}(\mathsf{Microl}(\mathcal{E}), \mathcal{M})$ to the same span in $\mathsf{Par}(\mathcal{E}, \mathcal{M})$. Faithfulness is immediate.

For fullness, let A, B be microlinear, and suppose that there is a span $A \xleftarrow{m} A' \xrightarrow{f} B$ in $\mathsf{Par}(\mathcal{E}, \mathcal{M})$ with m an étale monic. Then ([31] Proposition 2.3) says that an etale subobject of a microlinear space must be microlinear, thus A' is microlinear. But then the span is in $\mathsf{Par}(\mathsf{Microl}(\mathcal{E}), \mathcal{M})$ hence the embedding is full.

Then use Lemma 2.2, to show that $\mathsf{Par}(\mathsf{Microl}(\mathcal{E}), \mathcal{M})$ is a join restriction category and that the embedding preserves joins. □

Note that the full and faithfulness of the embedding implies a strong feature: the embedding is a *hyperconnection* in the sense of [16]. Thus the lattices of restriction idempotents of any object in the embedding is the lattice of restriction idempotents in $\mathsf{Par}(\mathsf{Microl}(\mathcal{E}), \mathcal{M})$; thus, the local logic is the same in both categories.

Now we embed $\mathsf{Smooth}_{\mathsf{P}}$ into $\mathsf{Par}(\mathsf{Microl}(\mathcal{E}), \mathcal{M})$.

Theorem 5.4. *If* $\mathcal{E}$ *is a well-adapted model of SDG that* **[WAM.5]***, then there is a faithful join restriction functor*

$$\mathsf{Smooth}_{\mathsf{P}} \xrightarrow{\iota_p} \mathsf{Par}(\mathsf{Microl}(\mathcal{E}), \mathcal{M})$$

Proof. Previously we noted that for an open subset $U \subseteq \mathcal{R}^n$, the image under iota $\iota(U) \hookrightarrow \mathscr{R}^n$ is a monic. Kock strengthens this: ([30] Theorem 3.4) showed that for every open subset $U \subseteq \mathcal{R}^n$, the image of the inclusion under ι

$$\iota(U) \hookrightarrow \iota(\mathcal{R}^n) \simeq \iota(\mathcal{R})^n$$

is a formal étale monic. Also, $\iota(\mathcal{R})$ is the ring of line type $\mathscr{R}$ in $\mathcal{E}$

Then, the functor

$$\mathsf{Smooth}_{\mathsf{P}} \longrightarrow \mathsf{Par}(\mathsf{Microl}(\mathcal{E}), \mathcal{M})$$

is defined as it was for the case where $\mathcal{M}$ is all monics: on objects $\iota_P(\mathcal{R}^n) := \mathscr{R}^n$. On arrows:

$$\iota_P((f, U) : \mathcal{R}^n \longrightarrow \mathcal{R}^m) := \quad \mathscr{R}^n \longleftarrow \iota(U) \xrightarrow{\iota(f)} \mathscr{R}^m$$

Also, since we know that for $U \subseteq \mathcal{R}^n$ the image under the embedding is an étale monic, this is well defined, and easily verified to be a faithful restriction functor.

For join preservation, as SMan has all idempotents split, then all restriction idempotents split. Then SMan with partial maps may be given the structure of the partial maps of an $\mathcal{M}$ category, and Lin [35] show that joins are constructed from pushouts of matching diagrams. Then let $U_1, U_2 \subseteq \mathcal{R}^n$ be open. Thus the inclusions are open embeddings and their pullback is transverse. The pushout of the matching diagram:

$$\begin{array}{ccc} U_1 \cap U_2 & \longrightarrow & U_2 \\ \downarrow & \ulcorner & \downarrow \\ U_1 & \longrightarrow & U_1 \cup U_2 \end{array}$$

is the union of $U_1 \cup U_2$. Note that the maps $U_1, U_2 \to U_1 \cup U_2$ are submersions, and the intersection of U_1, U_2 in $\mathcal{R}^n$ is the intersection of U_1, U_2 in $U_1 \cup U_2$. Thus the diagram is actually a transverse pullback, and ι preserves it. But that pullback in a topos is the colimit of the matching diagram, and hence ι preserves joins.

□

Next we show that $\mathcal{M}$ is a tangent system of monics, thus we have that $\mathsf{Par}(\mathsf{Microl}(\mathcal{E}), \mathcal{M})$ is a restriction tangent category.

Proposition 5.5. *For any well adapted model $\mathcal{E}$ and $\mathcal{M}$ the formal étale monics,* $\mathsf{Par}(\mathsf{Microl}(\mathcal{E}), \mathcal{M})$ *is a restriction tangent category.*

Proof. In a representable tangent category, hence any well adapted model $\mathcal{E}$, the tangent functor is continuous, so it preserves all pullbacks. Next if m is monic, as the tangent functor preserves pullbacks $[D, m]$ is monic.

It remains to show that $[D, m]$ is formal étale.

Suppose $A \xrightarrow{m} B$ is étale, then by the alternate characterization of étale, the following square is a pullback.

$$\begin{array}{ccc} [D, A] & \xrightarrow{[D,m]} & [D, B] \\ {\scriptstyle p}\downarrow & \lrcorner & \downarrow{\scriptstyle p} \\ A & \xrightarrow[m]{} & B \end{array}$$

We will first show that the following is a pullback.

$$\begin{array}{ccc} [D, [D, A]] & \xrightarrow{[D,[D,m]]} & [D, [D, B]] \\ {\scriptstyle p}\downarrow & & \downarrow{\scriptstyle p} \\ [D, A] & \xrightarrow[{[D,m]}]{} & [D, B] \end{array}$$

Note, the following square is a pullback as $[D, \_]$ is continuous.

$$\begin{array}{ccc} [D,[D,A]] & \xrightarrow{[D,[D,m]]} & [D,B] \\ {\scriptstyle [D,p]}\downarrow & & \downarrow{\scriptstyle [D,p]} \\ [D,A] & \xrightarrow[{[D,m]}]{} & [D,B] \end{array}$$

But the above pullback square can be factored using the canonical flip involution c as:

$$\begin{array}{ccccccc} [D,[D,A]] & \xrightarrow{c} & [D,[D,A]] & \xrightarrow{[D,[D,m]]} & [D,[D,B]] & \xrightarrow{c} & [D,[D,B]] \\ {\scriptstyle [D,p]}\downarrow & & {\scriptstyle p}\downarrow & & \downarrow{\scriptstyle p} & & \downarrow{\scriptstyle [D,p]} \\ [D,A] & = & [D,A] & \xrightarrow[{[D,m]}]{} & [D,B] & = & [D,B] \end{array}$$

A straightforward proof shows that as the outer square is a pullback, and c is an isomorphism, that the middle square is a pullback. But then, by the alternate criteria for formal étale, $[D, m]$ is formal étale.

Then by Theorem B.2., we have that $\mathsf{Par}(\mathsf{Microl}(\mathcal{E}), \mathcal{M})$ is a tangent restriction category. □

We now extend Theorem 5.4 to show that the tangent structure is lifted.

Theorem 5.6. *If $\mathcal{E}$ is a well-adapted model of SDG that satisfies the amazing right adjoint, then there is a faithful and conservative, tangent and join restriction functor*

$$\mathsf{Smooth}_{\mathsf{P}} \xrightarrow{\iota_P} \mathsf{Par}(\mathsf{Microl}(\mathcal{E}), \mathcal{M}).$$

Proof. The proof that functor is a faithful, cartesian restriction functor is given in Theorem 5.4. That the functor reflects isomorphisms, suppose a partial map $\mathscr{R}^n \supset \iota(U) \xrightarrow{f} \mathscr{R}^m$ is invertible, then the partial map is in fact total, and we have $\mathscr{R}^n = \iota(U)$, and the partial map is just the total map $\mathscr{R}^n \xrightarrow{\iota(f)} \mathscr{R}^m$; note that the inverse of f is also total, hence there is some total map $\mathscr{R}^m \xrightarrow{g} \mathscr{R}^n$ that inverts $\iota(f)$ But since $\mathscr{E}$ is a well-adapted model, ι is full and faithful on total maps (i.e. on $\mathscr{E}$), thus there is some g_0 such that $\iota(g_0) = g$ and hence g_0 inverts f.

For the preservation of the tangent bundle: In the proof of [30] Theorem 4.1, Kock shows that ι commutes with the tangent bundle on open subsets of $\mathcal{R}^n$. That is, if $U \subseteq \mathcal{R}^n$ then $\iota(TU) \simeq T(\iota U) \equiv [D, \iota U]$. But $\iota(TU) \simeq \iota(\mathcal{R}^n \times U) \simeq \mathscr{R}^n \times \iota(U)$. Finally [30] Theorem 3.3, we have that ι commutes with the derivative: $\iota(D[f]) = D[\iota(f)]$, and more generally $\iota(T(f)) = [D, \iota(f)]$. Thus the image of the $T(f)$ under ι_P is

$$\begin{array}{ccccc} & & \mathscr{R}^n \times \iota(U) & & \\ & \swarrow & & \searrow{\scriptstyle \iota(T(f))} & \\ \mathscr{R}^n \times \mathscr{R}^n & & & & \mathscr{R}^m \times \mathscr{R}^m \end{array}$$

which is the same (by the above) as

$$\begin{array}{ccc} & [D,\iota(U)] & \\ \swarrow & & \searrow^{[D,\iota(f)]} \\ \mathscr{R}^n \times \mathscr{R}^n & & \mathscr{R}^m \times \mathscr{R}^m \end{array}$$

Note the above are in the same equivalence class as partial maps; thus

$$\iota_P(T(f)) = T(\iota_P(f)).$$

Thus ι_P preserves restriction tangent structure.
The join preservation is also given by Theorem 5.4. □

For the partial cartesian closure of partial maps between mircolinear spaces with respect to all monics, we gave a condition for when it is an exponential ideal of the partial map category of the starting topos. The argument we made depended on having a partial map classifier in any topos. When we move to formal étale maps, the étale subobject classifier in the topos is not currently known to be microlinear.

We will show that if one has an étale partial map classifier M and has the property that for any microlinear X that MX is microlinear, then partial maps between microlinear spaces have a partial cartesian closed structure.

Proposition 5.7. *Suppose the partial map classifier satisfies the property that MX is microlinear whenever X. Then $\mathsf{Par}(\mathsf{Microl}(\mathcal{E}), \mathcal{M})$ is a partial cartesian closed, join restriction tangent category.*

Proof. The same as before. □

As before, we get a notion of étale subobject classifier. However, as étale subobjects are more well behaved, it seems more likely an étale subobject classifier will be microlinear.

Proposition 5.8. *If the partial map classifier M has the property that MX is microlinear whenever X, then there is an étale subobject classifier Ω^E and Ω^E is microlinear.*

Proof. As before, consider $M1$. As 1 is microlinear so is $M1$. Next note

$$\mathsf{Microl}(\mathcal{E})(A, M1) \simeq \mathsf{Par}(\mathsf{Microl}(\mathcal{E}), \mathcal{M})(A, 1) \simeq \mathcal{O}(A)$$

where $\mathcal{O}(A)$ is the restriction idempotents on A. But restriction idempotents on A are the precisely étale subobjects of A. □

We should remark that the embedding $\mathsf{Smooth}_\mathsf{P} \to \mathsf{Par}(\mathsf{Microl}(\mathcal{E}), \mathcal{M})$ is never full. There are new, non-classically constructible objects in $\mathcal{E}$.

Observation 5.9. *The embedding*

$$\mathsf{Smooth}_{\mathsf{P}} \xrightarrow{\iota_P} \mathsf{Par}(\mathsf{Microl}(\mathcal{E}), \mathcal{M})$$

is not full.

Proof. There is an object

$$D_\infty := \{d \in \mathcal{R} \,|\, d^k = 0 \,\text{for some}\ k\}$$

The object D_∞ is a subgroup of $(\mathcal{R}, +, 0)$; that is, it is closed under addition and subtraction. Then note that the map

$$D_\infty \times \mathcal{R} \xrightarrow{\alpha} [D, D_\infty]$$

defined by

$$\alpha(a, b) := \lambda d.db + a$$

is invertible. First note that it is well defined. As $a \in D_\infty$ by assumption and $db \in D_\infty$ becasue $(db)^2 = 0$ and D_∞ is closed to addition, α is well defined. Given a map $f : D \longrightarrow D_\infty$ extend the codomain to $\mathcal{R}$: $D \xrightarrow{f} D_\infty \hookrightarrow \mathcal{R}$. Then there is a unique $b \in \mathcal{R}$ such that $f(d) = f(0) + db \in D_\infty$; in particular $f(0) \in D_\infty$. Then the inverse to α is $\alpha^{-1}(f) = (f(0), b)$. Thus $[D, D_\infty] \simeq D_\infty \times \mathcal{R}$. But then,

$$\begin{array}{ccc}
D_\infty \times \mathcal{R} & \hookrightarrow & \mathcal{R} \times \mathcal{R} \\
\simeq \downarrow & & \downarrow \simeq \\
[D, D_\infty] & \hookrightarrow & [D, \mathcal{R}] \\
p \downarrow & & \downarrow p \\
D_\infty & \hookrightarrow & \mathcal{R}
\end{array}
\qquad (\text{outer curved arrows } \pi_0 \text{ on both sides})$$

is a pullback square, hence D_∞ is an étale subobject of $\mathcal{R}$. This means that the partial map

$$\begin{array}{ccc}
 & D_\infty & \\
\swarrow & & \searrow \\
\mathcal{R} & & \mathcal{R}
\end{array}$$

is in $\mathsf{Par}(\mathsf{Microl}(\mathcal{E}), \mathcal{M})$ but is not the image of the embedding. □

6 Future work

First, we would like to clean up and finish this current work, and consider more deeply the use of ω-CPOs discussed by [28].

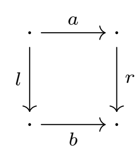
One aspect of this paper we find particularly interesting is the subtleties involved in finding spaces of smooth partial functions between microlinear spaces of models of SDG. We have all spaces of total smooth functions between microlinear spaces, such as $[\mathscr{R}, \mathscr{R}]$ and $[M, N]$ for any manifolds M, N, but we do not currently have a model with function spaces of smooth partial maps. If we use all monics, to have a partial cartesian closed category on microlinear spaces seems to require that the subobject classifier be microlinear, and this is quite a surprising requirement. It is not known whether this is possible. The ability to use étale subobject classifiers might shed some light onto the subtleties involved in defining spaces of smooth partial functions. We also find this a fascinating opportunity for continued research because étale subobject classifiers do not seem to have been studied.

This subtlety might not be too surprising. Partial cartesian closed categories are much closer to toposes than ordinary cartesian closed categories due to partial map classifiers providing a subobject classifier, and thus in some sense is a much stronger requirement.

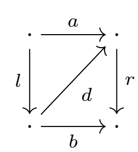
A Lifting Systems

This section was inspired by the notion of weak factorization system [4]. For our purposes, we do not need the ability factor maps; we show that certain properties like pullback stability follow from the weaker notion of having classes of maps that admit solutions to lifting problems against each other.

Let $\mathbb{X}$ be a category. A **lifting problem** is a square:



We say that l has a **left lifting problem** and r has a **right lifting problem**. A **solution** is a map d such that



commutes. We write $l \perp r$, and we say that l is perpendicular to r.

We also write $L \perp R$ if L and R are classes of maps such that every l is perpendicular to every map in R. We write $L_\perp = \{r \in \mathbb{X}_1 \mid \forall l \in L.\, l \perp r\}$, and we say $L_\perp$ is maximally right perpendicular to L. We write ${}_\perp R = \{l \in \mathbb{X}_1 \mid \forall r \in R.\, l \perp r\}$, and say that ${}_\perp R$ is maximally left perpendicular to R.

Definition A.1. *An* **antitone Galois connections** *between posets A, B is two maps $A \xrightarrow{f} B$ and $B \xrightarrow{g} A$ such that*

[4] For an introduction to weak factorization systems, see https://pdfs.semanticscholar.org/e73e/6224fb406575d3f30c2ebcd3cbbf10b13541.pdf.

1. *Antitone:* $a < a'$ *implies* $f(a') < f(a)$ *and* $b < b'$ *implies* $g(b') < g(b)$.
2. *Galois Inv:* $a < g(f(a))$ *and* $b < f(g(b))$.

It is a standard result that, regarding the posets as categories, a Galois connection induces an adjoint equivalence on the subsets $f(A)$ and $g(B)$.

Lemma A.2. *If* $(f, g) : A \rightleftarrows B$ *is a Galois connection, then* $f(A) \simeq g(B)$

Proof. We know $f(a) \leq f(g(f(a)))$ by definition. We also know $a \leq g(f(a))$ by definition, and since f is antitone by definition, we have $f(g(f(a))) \leq f(a)$. Thus, $f(a) = f(g(f(a)))$. Similarly, $g(b) = g(f(g(b)))$. ☐

Lemma A.3. ${}_\perp(\_)$ *and* $(\_)_\perp$ *form an antitone Galois connection (connexion) on the class of maps in a category.*

Proof. First, suppose that A, A' are classes of maps with $A \subseteq A'$. Then $A'_\perp \subseteq A_\perp$, since let $f \in A'_\perp$, then $l \perp f$ for every $l \in A'$, and since $A \subseteq A'$, $l \perp f$ for every $l \in A$. Similarly, ${}_\perp(\_)$ is order reversing.

Next, we have that $A \subseteq ({}_\perp A)_\perp$. Left $f \in A$. Then we must show $l \perp f$ for every map $l \in {}_\perp A$. But the maps in ${}_\perp A$ are chosen to be the ones that are left perpendicular to everything in A, including f. Thus, $f \in ({}_\perp A)_\perp$.

Similarly, $B \subseteq {}_\perp(B_\perp)$. ☐

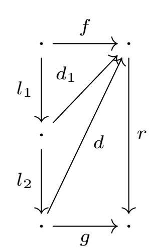
It follows from the general result about Galois connections, that ${}_\perp A = {}_\perp(({}_\perp A)_\perp)$ and $A_\perp = ({}_\perp(A_\perp))_\perp$. Thus, for any two classes of maps (L, R) with $L = {}_\perp R$ and $R = L_\perp$, we have immediately that $L = {}_\perp(L_\perp)$ and $R = ({}_\perp R)_\perp$. Then we can express (L, R) as

$$(L, R) = ({}_\perp R, L_\perp) = ({}_\perp R, ({}_\perp R)_\perp) = ({}_\perp(L_\perp), L_\perp)$$

Definition A.4. *A* **lifting system** *in a category* $\mathbb{X}$ *is two classes of maps* L, R *such that* $L = {}_\perp R$ *and* $R = L_\perp$.

Lemma A.5. *In a lifting system,* L, R *are closed to composition.*

Proof. For L: suppose $l_1, l_2 \in L$. To show that $l_1 l_2 \in L$ it suffices to show that there is a solution to any lifting problem against an R map. Then obtain a problem:

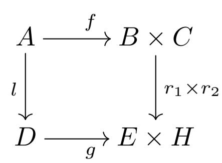


Then since $l_1 \in L$ we have a d_1 such that $l_1 d_1 = f$ and $d_1 r = l_2 g$. Then we have a lifting problem for l_2 so there is a d such that $dr = g$ and $l_2 d = d_1$. Then $l_1 l_2 d = l_1 d_1 = f$ hence $l_1 l_2 \in L$. ☐

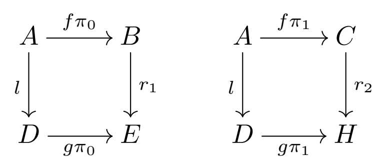
Lemma A.6. *If (L, R) is a lifting system, then*

1. *The nonempty product of R maps is an R map;*
2. *The nonempty coproduct of L maps is an L map.*

Proof. Let $r_1, r_2 \in R$. Obtain a lifting problem for $r_1 \times r_2$:



Then this lifting problem devolves into two lifting problems:

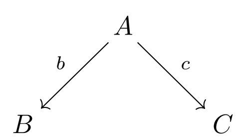


The first then has a solution d_1 and the second has a solution d_2. Then, $\langle d_1, d_2 \rangle$ is a solution to the original problem.

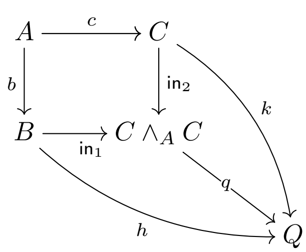
The result for coproducts is dual □

Now for a categorical interlude. Two maps f, g are called **jointly epic** if $gh = gk$ and $fh = hk$ implies $h = k$.

Given a pushout diagram (a span):



A **weak pushout** is an object $B \wedge_A C$ together with injections $B \xrightarrow{\mathsf{in}_1} B \wedge_A C$ $C \xrightarrow{\mathsf{in}_2} B \wedge_A C$ that makes the square commute, and such that for any other commuting square, there is a q such that the following commutes (note q is not unique).



The dual notion of weak pullback similarly drops the uniqueness condition for pullbacks.

Lemma A.7.

1. *A pushout is precisely a weak pushout with joint epic injections.*
2. *A pullback is precisely a weak pullback with joint monic projections.*

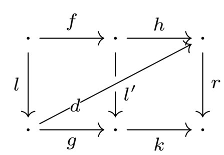
Proof. The proof is immediate. □

Now back to lifting systems.

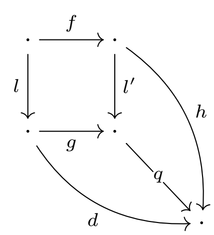
Lemma A.8. *Let (L, R) be a lifting system. Then,*

1. *The pushout of an L map is an L map.*
2. *The pullback of an R map is an R map.*

Proof. We prove for pushouts. The pullback case is dual. Let the first square be a pushout of an L map and the second a lifting problem.



Then the big square is a lifting problem, and since $l \in L$, there is a solution d such that $ld = fh$ and $dr = gk$. Rearrange this to



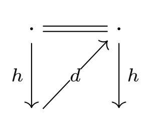
So that there is a q such that $l'q = h$ and $gq = d$ To show that q solves our lifting problem, it now suffices to show that $qr = k$. But now,

$$l'qr = hr = l'k \qquad gqr = dr = gk$$

so that by joint epicness of l', g we have $qr = k$. □

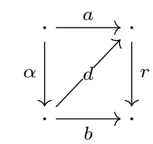
Lemma A.9. *If (L, R) is a lifting system in $\mathbb{X}$, then $L \cap R$ is precisely the class of isomorphisms in $\mathbb{X}$.*

Proof. Suppose $h \in L \cap R$. Then, we can solve the problem of h against itself



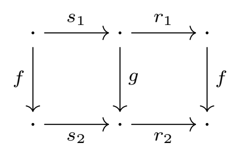
Thus h is an isomorphism.

Suppose α is an isomorphism. Let $r \in R$, and obtain a problem (i.e. the outer square)



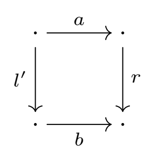
Then, define a solution $d := \alpha^{-1}a$. Then $\alpha d = \alpha\alpha^{-1}a = a$ and $dr = \alpha^{-1}ar = \alpha^{-1}\alpha b = b$. Thus $\alpha \in L$. One can similarly show that $\alpha \in R$. □

A map f is a **retract** of a g when it is a retract in the arrow category: i.e. there is a square

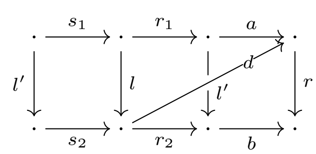


that commutes and $s_i r_i = 1$.

In a lifting system, L and R are closed to retracts. Suppose $l \in L$ and l' is a retract of l. Obtain a lifting propblem for l':

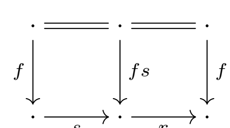


Then note, that the outside of all squares commute in the following. And we obtain a lift d such that $ld = r_1 a$ and $dr = r_2 g$.



But then $s_2 dr = s_2 r_2 b = b$ and $l' s_2 d = s_1 ld = s_1 r_1 a = a$, so that $s_2 d$ is a solution for l''s problem, hence $l' \in L = {}_{\perp}R$. A dual argument shows that R is closed to retraction.

Again suppose (L, R) is a lifting system. Suppose $sr = 1$ and $fs \in L$. Then we have



Then, since L is closed under retraction we have $f \in L$. Dually, if $sr = 1$ and $rg \in R$ then $g \in R$.

We summarize this in the following

Remark A.10. *If (L, R) is a lifting system, then L, R are closed to retraction. Further, if $sr = 1$ and $fs \in L$ then $f \in L$, and if $rg \in R$ then $g \in R$.*

The following is a standard categorical lemma:

Lemma A.11. *To given an adjunction $F \vdash G : \mathbb{X} \longrightarrow \mathbb{Y}$*

$$\frac{A \longrightarrow G(B)}{F(A) \longrightarrow B}$$

is to have two combinators

$$\frac{F(A) \xrightarrow{f} B}{A \xrightarrow{f^\flat} G(B)} \qquad \frac{A \xrightarrow{g} G(B)}{F(A) \xrightarrow{g^\sharp} B}$$

such that $(g^\sharp)^\flat = g$ and $(f^\flat)^\sharp = f$, and such that

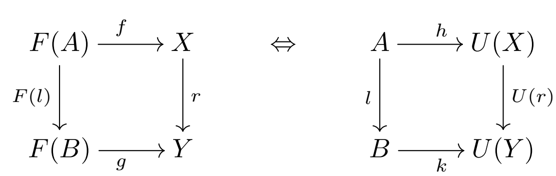
$$(F(h)fk)^\flat = hf^\flat G(k) \qquad \text{and} \qquad (k'gG(h'))^\sharp = F(k')g^\sharp h'$$

Lemma A.12. *Let $\mathbb{X}, \mathbb{Y}$ be categories and suppose $F : \mathbb{X} \longrightarrow \mathbb{Y}$, $U : \mathbb{Y} \longrightarrow \mathbb{X}$ are adjoint functors:*

$$\frac{A \longrightarrow U(B)}{F(A) \longrightarrow B}$$

Then $l \perp U(r)$ if and only if $F(l) \perp r$.

Proof. The adjunction immediately tells us that lifting problems for $F(l)$ and r bijectively correspond to lifting problems for l and $U(r)$:



I.e. if we have the first diagram, the second diagram is given by taking $h = f^\flat$ and $k = g^\flat$. Conversely, given the second diagram we have the first where $f = h^\sharp$ and $g = k^\sharp$.

Now, suppose the first diagram has a solution d_1. Then we have

$$(ld_1^\flat)^\sharp = F(l)(d_1^\flat)^\sharp = h^\sharp$$

Since $\sharp$ is injective, we have

$$ld_1^\flat = h$$

Similarly, we have

$$(d_1^\flat U(r))^\sharp = (d_1^\flat)^\sharp = d_1 r = g = k^\sharp$$

And again from $\sharp$ being injective, weh ave $d_1^\flat U(r) = k$ Hence, $d_1^\flat$ is a solution for the second problem.

Similarly, if the second problem has a solution, then the first does too. $\square$

Corollary A.13. *Let (L_1, R_1) be a lifting system on $\mathbb{X}$ and (L_2, R_2) be a lifting system on $\mathbb{Y}$, and let $F \dashv U : \mathbb{X} \longrightarrow \mathbb{Y}$ be an adjunction. Then, F sends L_1 maps to L_2 if and only if U sends R_2 maps to R_1.*

B The Partial Map Category of a Tangent Category

In this appendix we give an exposition of the partial map category of a tangent category.

We first describe the conditions needed to obtain a restriction tangent structure on a category of partial maps. When $\mathcal{M}$ is a tangent display system (i.e. the tangent functor preserves pullbacks along display maps) consisting of monics, and $A \xrightarrow{m} B$ is a monic, then Tm is a monic. This is because if we pullback Tm along Tm we get T applied to the pullback of m along m, whose domain is TA hence Tm is a monic. However, it does not follow immediately that $Tm \in \mathcal{M}$. A further complication is that natural transformations between endofunctors on a category do not yield natural transformations between the corresponding endofunctors on the partial map category; the naturality square involving maps in $\mathcal{M}$ must be a pullback (see section 3.2 of [11]).

Definition B.1. *Let $\mathbb{X}$ be a tangent category. We say a stable class of monics $\mathcal{M}$ is a* **tangent stable system of monics** *if*

1. *If $m \in \mathcal{M}$ then $Tm \in \mathcal{M}$.*
2. *T preserves pullbacks of maps along M-maps.*
3. *If $m \in \mathcal{M}$ then the naturality square:*

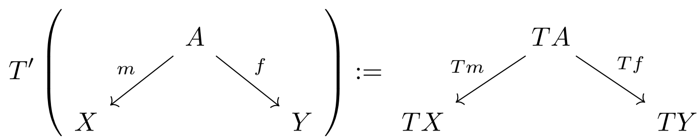
$$\begin{array}{ccc} TA & \xrightarrow{Tm} & TB \\ {\scriptstyle p}\downarrow & & \downarrow{\scriptstyle p} \\ A & \xrightarrow[m]{} & B \end{array}$$

   is a pullback.

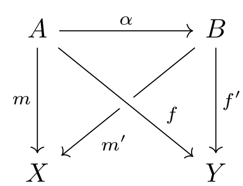
One might have expected that we axiomatize the naturality squares involving maps in $\mathcal{M}$ to all be pullbacks. This is however implied as indicated by the following theorem.

Theorem B.2. *If $\mathbb{X}$ is a tangent category and $\mathcal{M}$ a tangent stable system of monics, then $\mathsf{Par}(\mathbb{X}, \mathcal{M})$ is a restriction tangent category. If $\mathbb{X}$ is a Cartesian tangent category, then $\mathsf{Par}(\mathbb{X}, \mathcal{M})$ is a Cartesian restriction category.*

Proof. We will sketch the proof.



As maps in $\mathsf{Par}(\mathbb{X}, \mathcal{M})$ are equivalence classes, we need to show this definition is well defined. If $(m, f) \sim (m', f')$ then there is an isomorphism α between the domains of m and m' such that both triangles in the following commute:



As functors preserve isomorphisms $T(\alpha)$ then witnesses $T(m, f) \sim T(m', f')$.

Next, the naturality squares are proved by showing that each of the naturality squares involving a tangent structural transformation $(p, l, c, 0, +)$ and an $\mathcal{M}$ map is a pullback. This actually follows from the fact that p has this property and then applying the pullback pasting lemma (by pasting p appropriately). Finally, we then apply Proposition 3.2 of [11] to show that the natural transformations lift to $\mathsf{Par}(\mathbb{X}, \mathcal{M})$.

The required diagrams are pullbacks is relatively straightforward if long, and follows from the proof that the pullbacks involve total maps. □

C Very brief introduction to SDG

Let $\mathcal{E}$ be topos with a ring $\mathscr{R}$. We consider $\mathsf{Weil}_{\mathscr{R}}$, the category of Weil algebras over $\mathscr{R}$. There is a functor

$$D(\_) : \mathscr{R}\text{-}\mathsf{Weil}^{\mathrm{op}} \longrightarrow \mathcal{E}$$

Suppose a Weil algebra is presented as $\mathscr{R}[x_1, \ldots, x_n]/I$ and we choose a finite set of polynomials $p_1, \ldots, p_k$ that generate I; this is possible as Weil algebras are finitely generated. Then define the *spectrum* [5] by:

$$D(U) := \{(a_1, \ldots, a_n) \in \mathscr{R}^n \mid \forall k. p_k(a_1, \ldots, a_n) = 0\}$$

Sometimes $D(\_)$ goes by the name Spec:

$$\mathsf{Spec}(U) := D(U)$$

[5] We used $\mathsf{Spec}(\_)$ in the above writeup.

For example

$$D(W^1) = D(\mathscr{R}[x]/(x^2)) := \{d \in \mathscr{R} \mid d^2 = 0\}$$

Given a homomorphism of Weil algebras $U \xrightarrow{f} V$ where U has n generators and V has m generators, let $(b_1, \ldots, b_m) \in D(V)$. For each generator x_i of U, $f(x_i)$ is a polynomial (of V). Let $a_i := f(x_i)(b_1, \ldots, b_m)$.

$$D(f)(b_1, \ldots, b_m) := (a_1, \ldots, a_n)$$

This is well defined. A map $U \xrightarrow{f} V$ on presentations is a map $\mathscr{R}[x_1, \ldots, x_n]/I_U \xrightarrow{f} \mathscr{R}[x_1, \ldots, x_m]/I_V$. This must arise from a map $\hat{f} : \mathscr{R}[x_1, \ldots, x_n] \to \mathscr{R}[x_1, \ldots, x_m]/I_V$ that sends $p \in I_U$ to $0 \mod I_V$. In turn this means that $\hat{f}(p) \in I_V$. Let $(b_1, \ldots, b_m) \in D(V)$. In particular, $\hat{f}(p)(b_1, \ldots, b_m) = 0$. But

$$\begin{aligned}
&\hat{f}(p)(b_1, \ldots, b_m) \\
&= \hat{f}(\sum_\alpha c_\alpha x^\alpha)(b_1, \ldots, b_m) \\
&= \sum_\alpha c_\alpha (fx_1)^{\alpha_1}(b_1, \ldots, b_m) \cdots (fx_n)^{\alpha_n}(b_1, \ldots, b_m) \\
&= \sum_\alpha c_\alpha (f(x_1)(b_1, \ldots, b_m))^{\alpha_1} \cdots (f(x_n)(b_1, \ldots, b_m))^{\alpha_n} \\
&= \sum_\alpha c_\alpha a_1^{\alpha_1} \cdots a_n^{\alpha^n} = p(a_1, \ldots, a_n)
\end{aligned}$$

Then define an action

$$\mathcal{E} \times \mathscr{R}\text{-}\mathsf{Weil} \xrightarrow{\otimes_\infty} \mathcal{E}$$

Where $X \otimes_\infty U := [D(U), X]$ on objects and $f \otimes_\infty g := [D(g), f]$ on arrows. This is clearly a bifunctor as $D(\_)$ is a functor. It is also a monoidal action: note that $D(U \otimes V) \simeq D(U) \times D(V) \in \mathcal{E}$ (see [34] *3* in 2.1.2).

The **microlinear spaces** of $\mathcal{E}$ are those objects M for which $M \otimes_\infty (\mathsf{lim}_i U_i) \simeq \mathsf{lim}_i (M \otimes_\infty U_i)$ for every connected limit $\mathsf{lim}_i U_i$ of weil algebras.

Then

Proposition C.1. *For any topos $\mathcal{E}$ and any ring $\mathscr{R}$, $\mathsf{Microl}(\mathcal{E})$ is a coherently closed tangent category with all limits, and the tangent bundle is defined by $TM \equiv [D(R[x]/(x^2)), M]$.*

However, this is not satisfying. The image of D need not be microlinear, nor does $\mathscr{R}$ need to be microlinear, and thus the most basic space for assembling differential geometry may not be microlinear!

A topos $\mathcal{E}$ is called **smooth** (with respect to $\mathscr{R}$) when the following canonical map

$$W \xrightarrow{\alpha} \mathscr{R} \otimes_\infty W := [D(W), \mathscr{R}]$$

is an isomorphism for each Weil algebra W. If W is presented as $W \simeq \mathscr{R}[x_1,\ldots,x_n]/I$ then α arises by considering $\hat{\alpha} : \mathscr{R}[x_1,\ldots,x_n] \to [D(W),\mathscr{R}]$ which sends a polynomial p to the function which evaluates p on $D(W)$. This map passes to the quotient, since any $p \in I$ is 0 on any element of $D(W)$.

A consequence of the smoothness assumption is:

$$W_1 \otimes W_2 \simeq \mathscr{R} \otimes_\infty (W_1 \otimes W_2) \simeq (\mathscr{R} \otimes_\infty W_1) \otimes_\infty W_2 \simeq W_1 \otimes_\infty W_2$$

The following is then immediate

Proposition C.2. *The microlinear spaces of a smooth topos $\mathcal{E}$ are a cartesian closed, representable tangent category with all limits where $\mathscr{R}$ is microlinear and for each Weil algebra W, $D(W)$ is microlinear.*

In $\mathsf{Microl}(\mathcal{E})$ the differential objects are precisely the Euclidean $\mathscr{R}$-vector space; that is, vector spaces that satisfy the Kock-Lawvere axiom: $[D(W_1),V] \simeq V \times V$. One can formulate the notion of manifold modelled on Euclidean $\mathscr{R}$-vector spaces. Kock showed the following in [32]

Proposition C.3. *Synthetic manifolds are microlinear.*

References

[1] Pytorch. https://pytorch.org/.

[2] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[3] Thomas Beck and Herbert Fischer. The if-problem in automatic differentiation. *Journal of Computational and Applied Mathematics*, 50(1):119 – 131, 1994.

[4] W. Bertram, H. Glöckner, and K.H. Neeb. Differential calculus over general base fields and rings. *Expositiones Mathematicae*, 22(3):213 – 282, 2004.

[5] R. Blute, R. Cockett, and R. Seely. Cartesian Differential Categories. *Theory and Applications of Categories*, 22:622–672, 2009.

[6] R.F. Blute, J.R.B. Cockett, and R.A.G. Seely. Differential categories. *Mathematical structures in computer science*, 16(6):1049–1083, 2006.

[7] Richard Blute, Thomas Ehrhard, and Christine Tasson. A convenient differential category. *CoRR*, abs/1006.3140, 2010.

[8] J.D. Christensen and E. Wu. Tangent spaces and tangent bundles for diffeological spaces. *Cahiers de Topologie et Geométrie Différentielle Catégoriques*, 2016.

[9] J.R.B. Cockett, G.S.H. Cruttwell, and J.D. Gallagher. Differential restriction categories. *Theory and applications of categories*, 25(21):537–613, 2011.

[10] J.R.B. Cockett, P.J.W. Hofstra, and P. Hrubeš. Total maps of turing categories. *Electronic Notes in Theoretical Computer Science*, 308:129–146, 2014. Proceedings of the 30th Conference on the Mathematical Foundations of Programming Semantics (MFPS XXX).

[11] J.R.B. Cockett and Stephen Lack. Restriction categories i: categories of partial maps. *Theoretical Computer Science*, 270(1):223 – 259, 2002.

[12] R. Cockett and G. Cruttwell. Differential structure, tangent structure, and SDG. *Applied Categorical Structures*, 22:331–417, 2014.

[13] R. Cockett and S. Lack. Restriction categories iii: colimits, partial limits and extensivity. *Mathematical Structures in Computer Science*, 17(4):775–817, 2007.

[14] R. Cockett and E. Manes. Boolean and classical restriction categories. *Mathematical Structures in Computer Science*, 19(2):357–416, 2009.

[15] Robin Cockett, Joaquín Díaz-Boïls, Jonathan Gallagher, and Pavel Hrubeš. Timed sets, functional complexity, and computability. *Electronic Notes in Theoretical Computer Science*, 286:117–137, 2012. Proceedings of the 28th Conference on the Mathematical Foundations of Programming Semantics (MFPS XXVIII).

[16] Robin Cockett and Richard Garner. Restriction categories as enriched categories. *Theoretical Computer Science*, 523:37 – 55, 2014.

[17] P.-L. Curien and A. Obtułowicz. Partiality, cartesian closedness, and toposes. *Information and Computation*, 80(1):50 – 95, 1989.

[18] E. Dubuc. Sur les modéles de la géométrie différentielle synthétique. *Cahiers de Topologie Géométrie Différentielle Catégoriques*, 20(1):231–279, 1979.

[19] E. Dubuc and A. Kock. On 1-Form Classifiers. *Communications in Algebra*, 12(12):1471–1531, 1984.

[20] T. Ehrhard. On Köthe Sequence Spaces and Linear Logic. *Mathematical Structures in Computer Science*, 12:579–623, 2002.

[21] T. Ehrhard. Finiteness spaces. *Mathematical Structures in Computer Science*, 15:615–646, 2005.

[22] T. Ehrhard and L. Regnier. The differential lambda-calculus. *Theoretical Computer Science*, 309:1–41, 2003.

[23] J. Gallagher. *The differential lambda-calculus: syntax and semantics for differential geometry.* PhD thesis, University of Calgary, 2009.

[24] Marco Grandis. Cohesive categories and manifolds. *Annali di Matematica Pura ed Applicata*, 157(1):199–244, 1990.

[25] X. Guo. *Products, Joins, Meets, and Ranges in Restriction Categories*. PhD thesis, University of Calgary, 2012.

[26] P. Iglesias-Zemmour. 2013.

[27] P. Johnstone. *Sketches of an Elephant: a topos theory compendium.* Oxford University Press, 2002.

[28] O. Kammar, S. Staton, and M. Vákár. Diffeological spaces and denotational semantics for differential programming. MFPS 2018 Contributed Talk.

[29] R. Baker Kearfott. Chapter 1 automatic differentiation of conditional branches in an operator overloading context, 1996.

[30] A. Kock. *Synthetic differential geometry.* Cambridge University Press, 1981.

[31] A. Kock and G. E. Reyes. *Manifolds in formal differential geometry*, pages 514–533. Springer Berlin Heidelberg, Berlin, Heidelberg, 1979.

[32] Anders Kock. *Synthetic Geometry of Manifolds.* Cambridge Tracts in Mathematics. Cambridge University Press, 2009.

[33] M. Laubinger. *Differential Geometry in Cartesian Closed Categories of Smooth Spaces*. PhD thesis, Louisiana State University and Agricultural and Mechanical College, 2008.

[34] R. Lavendhomme. *Basic Concepts of Synthetic Differential Geometry*. Kluwer Texts in Mathematical Sciences. Kluwer Academic Publishers, 1996.

[35] Daniel Lin. Presheaves over a join restriction category. *Applied Categorical Structures*, 27(3):289–310, Jun 2019.

[36] Philip S. Mulry. Partial map classifiers and partial cartesian closed categories. *Theoretical Computer Science*, 136(1):109 – 123, 1994.

[37] Robert A. Di Paola and Alex Heller. Dominical categories: Recursion theory without elements. *The Journal of Symbolic Logic*, 52(3):594–635, 1987.

[38] G. Plotkin. A simple differential programming language. MFPS 2018 Keynote Address, June 2018.

[39] G. Rosolini. *Continuity and effectiveness in topoi.* PhD thesis, University of Oxford, 1986.

[40] David Sprunger and Bart Jacobs. The differential calculus of causal functions. *CoRR*, abs/1904.10611, 2019.

[41] David Sprunger and Shin-ya Katsumata. Differentiable causal computations via delayed trace. *CoRR*, abs/1903.01093, 2019.