Abstract Datatypes for Differential Programming

Benjamin MacAdam
and many others. . .

May 30, 2019

Motivation

This talk is a first stab at abstract data types for differential programming via tangent categories.

Part of a broader project in differential programming with Jonathan Gallagher, Geoff Cruttwell, Dorette Pronk.

Draws heavily from other joint projects where Enriched Sketch Theory plays a role.

- Scalar rings in a tangent category, with Jonathan Gallagher and Rory Lucyshyn-Wright
- Involution Algebroids, with Matthew Burke

Idea: Enriched sketches (and more specifically, enriched algebraic theories) should play a central role as tangent category theory develops.

What is an Abstract Data Type

Comes from software engineering - what does an object do?

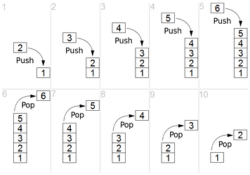

Figure: LIFO stack by Maxtremus is licensed by CC0

An Abstract Data Type (ADT) is some collection of objects, with some operations that satisfy certain equations (e.g. a stack with push and pop)

In languages like Java or Haskell you can only specify the signature (via interfaces and type classes respectively).

Outline of this talk

First, we'll discuss Barr and Wells' work on the ADT/sketch correspondence.

- We will be working in a cartesian category **Sem** (total maps).
- There's still a lot to do with *restriction theories*:
  - The restriction theory of a category is cartesian!

We discuss a "∂-ADT" for calculus.

- "Scalars in a Tangent Category" w/ Rory and Jonathan.

Enriched sketches extend Barr and Wells's correspondence.

ADTs in a CCC

Now an ADT is:

- A set of base types
- A signature of functions (which may be partially defined)
- A set of equations the functions must satisfy.

We will follow Barr and Wells and model this using Sketch Theory.

Definition (Finite Limit Sketch)

A *finite limit sketch* $\mathcal{T}$ is a small category with a chosen set of cones $\mathcal{L}$ (we will often omit "finite limit"). A model of a sketch in a functor sending cones to limits.
A *general* sketch has an set of cocones.

Some theorems about Sketches

- The category of models of a sketch is a full reflective subcategory of the presheaf category.

$$
\begin{tikzcd}
\mathrm{Mod}(\mathcal{T}) \arrow[r, hook, shift right] \arrow[r, phantom, "\perp"] & \mathrm{Psh}(\mathcal{T}) \arrow[l, shift right]
\end{tikzcd}
$$

- The category of models of a sketch in Set is *locally presentable*.
  - Complete and cocomplete
  - Every object the filtered colimit of "finitely presentable" objects.
- Every locally presentable category is the category of models of a sketch.
- Locally presentable categories have extremely nice features (adjoint functor theorems, etc)

Some Abstract Data Types

Stacks: Two base types X, S and maps

$$
\begin{tikzcd}
X \times S \arrow[rr, equal] \arrow[dr, "\mathsf{push}"'] & & X \times S \\
& S \arrow[ur, "\mathsf{pop}"'] &
\end{tikzcd}
$$

Most theories you can think of: Monoids, Categories, Graphs . . .

Isn't this supposed to be about differential programming?

Differential Programming and Tangent Categories

Theorem (Cruttwell, Gallagher, M.)

Plotkin's language from POPL2018 can be interpreted into a join tangent restriction category whose category of total maps is a coherently closed tangent category.

We'll continue with total functions - so a coherently closed tangent category.

$$T[A, -] \Rightarrow [A, T-]$$

Some problems

Now, we want to use the derivative to write machine learning algorithms. So we look at models of R-modules

- There's no reason $V \times V \cong T(V)$.
- There's no reason for there to be an R.

We can look at *differential objects* to do calculus

- Differential objects use the tangent bundle in their definition, they aren't a sketch.

Some problems

Now, we want to use the derivative to write machine learning algorithms. So we look at models of R-modules

- There's no reason $V \times V \cong T(V)$.
- There's no reason for there to be an R.

We can look at *differential objects* to do calculus

- Differential objects use the tangent bundle in their definition, they aren't a sketch.

First step: Add a universal ring object and look at its modules

This was developed in "Scalars in a Tangent Category" with Rory and Jonathan.

Linear Classifier

Under a very mild assumption, we can add a universal ring object to a tangent category.

Definition (Blute-Cockett-Seeley)

A *Scalar Unit* is a differential object with a point $1 \xrightarrow{u} R$ with the universal property that for all

$$\begin{tikzcd} V \arrow[r, "f"] \arrow[d, "\langle 1, u \rangle"'] & W \\ V \times R \arrow[ur, dotted, "\exists ! \hat{f} \text{linear in } R"'] & \end{tikzcd}$$

f (multi)-linear in $V \Rightarrow \hat{f}$ is (multi)-linear in V

Consequences of a Linear Point Classifier

- The unit object is a commutative rig R.

$$\begin{tikzcd} R \arrow[r, equal] \arrow[d, "\langle 1,u\rangle"'] & R \\ R \times R \arrow[ur, "\cdot"'] & \end{tikzcd}$$

- Every differential object is an R-module.

$$\begin{tikzcd} V \arrow[r, equal] \arrow[d, "\langle 1,u\rangle"'] & V \\ V \times R \arrow[ur, "\cdot"'] & \end{tikzcd}$$

- Every linear map preserves the R-module action (persistence).

Rewriting the lift

Every R-module has the map λ^R:

$$V \xrightarrow{\langle 1,u\rangle} V \times R \xrightarrow{0\times\lambda} T(V \times R) \xrightarrow{T(\cdot)} T(V)$$

In SDG: $v \mapsto \lambda d.vd$.

Lemma

For a differential object,λ^R satisfies the equalizer

$$V \xrightarrow{\lambda^R_V} T(V) \underset{!\xi}{\overset{p}{\rightrightarrows}} V$$

so $\lambda_V = \lambda^R_V$.

Corollary

Homogenous morphisms of differential objects are linear.

KL-Modules

Set $\nu^R := \; V \times V \xrightarrow{\lambda^R \times 0} T(V) \times T(V) \xrightarrow{T(\sigma)} T(V)$

Definition (*Kock-Lawvere R-module*)

V is a KL-module if there is an R-module map $\hat{p}$ making

$$(\nu)^{-1} = \langle \hat{p}, p_V \rangle$$

- The category of KL-modules is equivalent to the category of differential objects.
- In a locally presentable tangent category, KL-modules is a full reflective subcategory of R-modules.
- If the tangent bundle is a *group bundle*, then KL-modules are a completion of R-modules.

New Questions

The notion of a scalar unit allows one to use the simpler definition of KL-modules.

If R is a ring, KL-modules are a completion of R-modules - is there a sketch of KL-modules?

New Questions

The notion of a scalar unit allows one to use the simpler definition of KL-modules.

If R is a ring, KL-modules are a completion of R-modules - is there a sketch of KL-modules?

Move to *enriched category theory* - the tangent bundle is a weighted limit

$$T(M) = y(R[x]/x^2) \pitchfork M$$

Units in Presheaf Categories

Observation

The enriched Yoneda embedding preserves differential objects.

Theorem (Gallagher, Lucyshyn-Wright, M.)

The enriched presheaf category of a tangent category has a representable unit:

$$1 \bullet R \cong [1 \bullet y(x^2), 1 \bullet y(x^2)]$$

Observation (Dubuc and Kock)

Differential objects are sent to KL-modules by the Yoneda embedding.

Enriched Sketches

Definition

A $\mathcal{E}$ − *sketch* is a small $\mathcal{E}$-category with a set weighted limits.
A model sends cylinders to weighted limits.

Remark

- The tangent bundle is given by a weighted limit

$$T(M) = y(R[x]/x^2) \pitchfork M$$

- The category $\mathcal{E}$ is locally finitely presentable, as

$$\mathcal{E} = \mathrm{Mod}(\mathrm{Weil}_1, \mathrm{Set})$$

- Every **Set**-sketch can be made an $\mathcal{E}$-sketch where you ask that your limits are preserved by T.

KL-Mod

- Objects: Natural numbers $\mathbb{N}$.
- Homs: $[n, m] = R^{n \times m}$
- Composition: Matrix multiplication.
- Cylinders:
  - 0 as a $\mathbb{E}$-terminal object.
  - n as the n-fold $\mathbb{V}$-pullback of $1 \to 0$.
  - $2n$ for the power $R[x]/x^2 \pitchfork n$
    - $0_n : 1 \to R^{n \times 2n}$ picks out the matrix $\begin{bmatrix} 0 \\ I \end{bmatrix}$
    - $p_n : 1 \to R^{2n \times n}$ picks out $\begin{bmatrix} 0 & I \end{bmatrix}$

The map ν^R is the unique map sending $V \times V$ to $T(V)$ - a model of this sketch will induce the map $\hat{p}$ making $\langle p, \hat{p} \rangle = (\nu^R)^{-1}$. Since a KL-module must be a model of R-Mod, then the KL-property ensures that ν^R is the map mediating the limit.

Composing Theories

One other aspect of sketches that has not been touched on: the tensor product

Definition (Tensor product of sketches)

Let $\mathcal{T}, \mathcal{S}$ be enriched sketches. Define $\mathcal{S} \otimes \mathcal{T}$ as:

- $\mathbf{Cat}(\mathcal{S} \otimes \mathcal{T}) = \mathbf{Cat}(\mathcal{S}) \times \mathbf{Cat}(\mathcal{S})$
- $\mathbf{Lim}(\mathcal{S} \times \mathcal{T}) = \mathbf{Lim}(\mathcal{S}) \times \mathbf{Ob}(\mathcal{T}) \cup \mathbf{Lim}(\mathcal{T}) \times \mathbf{Ob}(\mathcal{S})$

At least when $\mathcal{V} = \mathbf{Set}$, the following holds:

Theorem

Let $\mathcal{T}, \mathcal{S}$ be sketches, and $\mathcal{A}$ be locally presentable.

$$\mathbf{Mod}(\mathcal{T}, \mathbf{Mod}(\mathcal{S}, \mathcal{A})) \cong \mathbf{Mod}(\mathcal{S}, \mathbf{Mod}(\mathcal{T}, \mathcal{A})) \cong \mathbf{Mod}(\mathcal{A} \otimes \mathcal{S}, \mathcal{A}))$$

Differential Stacks (no, not those)

Can use the tensor product of sketchs to combine sketches.

Definition

A stack in the category of KL-modules is a *differential stack*.

$$\begin{tikzcd} X \times S \arrow[rr, equal] \arrow[dr, "\text{push}"'] & & X \times S \\ & S \arrow[ur, "\text{pop}"'] & \end{tikzcd}$$

Example

The Dubuc topos has a natural numbers object N.
Take $X = R, S = [N, R]$

- push puts your number to the start of the list, pushes everything up.
- pop takes off the first number.

Other possible data types to consider:

- **Differential Bundles**: the bundle version of differential objects.
- **Formal Submersions**: A submersion $q : F \to M$ can be characterized by the pushout:

$$\begin{tikzcd} TF \arrow[r, "\langle Tq,p\rangle"] \arrow[d, "\langle Tq,p\rangle"'] & F_q\times_p TM \arrow[d, equal] \\ F_q\times_p TM \arrow[r, equal] & F_q\times_p TM \end{tikzcd}$$

  The category of models will only be $\mathcal{E}$-*accessible*.
- **Involution Algebroids**: A generalization of *Lie Algebroids* (You'll hear more about these later).

Conclusions and Future Work

We've seen some applications of $\mathcal{E}$-sketches to Differential Programming.

What else can be done?

- Major theorem: Gabriel-Ulmer duality for tangent categories (Jonathan and Geoff are working on this).
- The category of sketches has a tensor product - can this be used to simplify Kirril MacKenzie's work on Lie theory (double Lie algebroids, LA-groupoids, etc).
- A more structural account of sector forms and symplectic mechanics - sector forms can be seen as morphisms of n-fold differential bundles.

Another direction is to develop the small object argument for a locally presentable tangent category.

References

Adamek, Jiri, Jiri Rosicky, et al. (1994). *Locally presentable and accessible categories*. Vol. 189. Cambridge University Press.

Barr, Michael and Charles Wells (1990). *Category theory for computing science*. Vol. 49. Prentice Hall New York.

Blute, R, J Robin B Cockett, and Robert AG Seely (2015). "Cartesian differential storage categories". In: *Theory and Applications of Categories* 30.18, pp. 620–686.

Cockett, J Robin B and Geoff SH Cruttwell (2014). "Differential structure, tangent structure, and SDG". In: *Applied Categorical Structures* 22.2, pp. 331–417.

Cockett, JRB and GSH Cruttwell (2017). "Connections in tangent categories". In: *Theory and Applications of Categories* 32.26, pp. 835–888.

Garner, Richard (2018). "An embedding theorem for tangent categories". In: *Advances in Mathematics* 323, pp. 668–687.

Kelly, Gregory Maxwell (1982). "Structures defined by finite limits in the enriched context, I". In: *Cahiers de topologie et géométrie différentielle catégoriques* 23.1, pp. 3–42.

Leung, Poon (2017). "Classifying tangent structures using Weil algebras". In: *Theory and Applications of Categories* 32.9, pp. 286–337.

Plotkin, G. (June 2018). "A Simple Differential Programming Language". MFPS 2019 Keynote Address.

Rosicky, Jiri (1984). "Abstract tangent functors". In: *Diagrammes* 12, JR1–JR11.