Semantics of higher-order incremental computation

Mario Alvarez-Picallo

UNIVERSITY OF OXFORD

Acknowledgements

This research is funded by the EPSRC and the Oxford Computer Science department, and supervised by professors Luke Ong and Samson Abramsky.

Part of this research was performed while I was working for Semmle Ltd. and in close collaboration with the team there, in particular Michael Peyton-Jones and Alex Eyers-Taylor.

Outline

What is incremental computation?

- We want to compute the value of some (expensive) function f on some input x
- The value of x changes over time, taking a sequence of values $x_1, x_2, \ldots$
- We want to update the value of $f(x)$ as x_i changes...
- ...but it may be very expensive to compute $f(x_i)$ from scratch!

What is incremental computation?

- Interpret the x_i as applying successive "updates" Δx_i to an initial value x_1:

$$x_2 = x_1 + \Delta x_1$$
$$x_3 = x_2 + \Delta x_2 \dots$$

- Find Δy_i such that:

$$f(x_2) = f(x_1 + \Delta x_1) = f(x_1) + \Delta y_1$$
$$f(x_3) = f(x_2 + \Delta x_2) = f(x_2) + \Delta y_2 \dots$$

- (Optional) Make sure computing $f(x_i) + \Delta y_i$ is cheaper than computing $f(x_{i+1})$

How to find the Δy_i? Many approaches!

The compositional approach to incremental computation

A general approach is to seek a function f' of the base state x_i and the update Δx_i that allows us to compute the Δy_i.

Most approaches make some restrictions on the kind of function f' is, most commonly it's required that it depends only on Δx_i.

Ideally, we know g' for all the primitives of our language.

Then we only need a way to obtain $(g \circ f)'$ from f' and g'!

Cai and Giarrusso's approach

In a recent paper, Cai, Giarrusso et al.[1] introduce a theoretical framework for making these ideas precise.

They introduce the notion of *change structure*, which provides a semantics for a type X whose elemnts can be updated with "changes" of type ΔX.

Their work also introduces compositional ways of obtaining incrementalized versions ("derivatives") of functions compositionally, as a source-to-source transformation.

1. Cai, Giarrusso et al, PLDI 2016

Change structures in detail

Change structures

A **change structure** is a tuple $(A, \Delta A, \oplus, \ominus)$ such that:

- A is a set
- For every $a \in A$, ΔA_a is a set
- $\oplus$ is a (dependent) function of type $(x : A) \to \Delta A_a \to A$
- $\ominus$ is a (dependent) function of type $A \to (x : A) \to \Delta A_x$
- $x \oplus (y \ominus x) = y$

Intuition: ΔA_a is the type of changes that are applicable to some value a.

Change structures, continued

Derivatives

Given a function $f : A \to B$ and change structures on A and B, a **derivative** for f is a function $f' : (a : A) \to (da : \Delta A_x) \to \Delta B_{f(a)}$ such that:

- $f \ (a \oplus da) = (f \ a) \oplus (f' \ a \ da)$

Note that every function is differentiable, because $\lambda \ x \ dx.f \ (x \oplus dx) \ominus (f \ x)$ is a derivative.

Given some λ-term $\Gamma \vdash t : B$, Giarrusso and Cai provide a source-to-source transformation *Derive* to find a derivative.

Problems in paradise

Some problems with Cai and Giarrusso's approach.

Every function is differentiable!

Requires dependent types!

- More complicated theory (can't use simply-typed λ-calculus)
- In practice: either limiting or just ignored...

Always working with sets, not adequate for domains!

- $\perp \oplus (a \ominus \perp) = a$

Derive is not first-class

In some cases, not clear what $\ominus$ should be, e.g. lists!

Outline

An alternative to change structures

Difference algebra

A **difference algebra** $\overline{A}$ is a tuple $(A, \Delta A, \oplus, +, 0)$ such that:

- $(\Delta A, +, 0)$ is a monoid
- $\oplus : A \to \Delta A \to A$ is an action of ΔA on A, i.e.:
  1. $a \oplus 0 = a$
  2. $a \oplus (da + db) = (a \oplus da) \oplus db$

Fulfills same purpose as a change structure, but! No dependent types, no $\ominus$.

An alternative to change structures

Derivatives between difference algebras

Given a function $f : A \to B$ and difference algebras on A and B, a **derivative** for f is a function $f' : A \to \Delta A \to \Delta B$ such that:

- $f\ (a \oplus da) = (f\ \ a) \oplus (f'\ \ a\ da)$

Same as with change structures, minus the dependent types!

Note that *all functions are no longer differentiable!*

The chain rule

Theorem

Given $f : A \to B, g : B \to C$ differentiable maps with derivatives f', g', then the following is a derivative for $g \circ f$:

$$(g \circ f)'(a)(da) = g'(f\ a, f'\ a\ da)$$

This is in fact the same as the chain rule from multivariate calculus! If $D_a f$ is the differential of f at a, then:

$$D_a(g \circ f) = D_{f(a)} g \circ D_a f$$

But unlike in calculus, here derivatives may not be unique!

A category of difference algebras

The category **DAlg**

We define a category **DAlg** of difference algebras as follows:

- The objects of **DAlg** are difference algebras $\overline{A} = (A, \Delta A, \oplus, +, 0)$
- The morphisms in **DAlg** are differentiable functions.
- Identities and composition are as in **Set**

The chain rule guarantees that composition behaves well!

The structure of **DAlg**

DAlg is a well-behaved category, it has:

- Products (nice!)
- Coproducts (nice!)
- Exponentials (evil!)

Products in **DAlg**

Product of difference algebras

Given difference algebras $\overline{A}, \overline{B}$, their product difference algebra $\overline{A} \times \overline{B}$ is given by:

$$\begin{aligned}\overline{A} \times \overline{B} &= (A \times B, \Delta A \times \Delta B, \oplus_\times, +_\times, 0_\times) \\ (a, b) \oplus_\times (da, db) &= (a \oplus da, b \oplus db) \\ (da_1, db_1) +_\times (da_2, db_2) &= (da_1 + da_2, db_1 + db_2) \\ 0_\times &= (0, 0)\end{aligned}$$

This is, in fact, the categorical product in **DAlg**!

Coproducts in **DAlg**

Coproduct of difference algebras

Given difference algebras $\overline{A}, \overline{B}$, their coproduct difference algebra $\overline{A} + \overline{B}$ is given by:

$$\begin{aligned} \overline{A} + \overline{B} &= (A + B, \Delta A \times \Delta B, \oplus_+, +_+, 0_+) \\ a \oplus_+ (da, db) &= a \oplus da \\ b \oplus_+ (da, db) &= b \oplus db \\ (da_1, db_1) +_+ (da_2, db_2) &= (da_1 + da_2, db_1 + db_2) \\ 0_+ &= (0, 0) \end{aligned}$$

Exponentials in **DAlg**

Theorem

The category **DAlg** is equivalent to the category **PreOrd** of preorders and monotone functions.

...

...and **PreOrd** is a CCC, therefore so is **DAlg**.

Exponentials in **DAlg**

Second attempt:

Exponential difference algebras

Given difference algebras $\overline{A}, \overline{B}$, the exponential $\overline{A} \Rightarrow \overline{B}$ is given by:

$$\begin{aligned} \overline{A} \Rightarrow \overline{B} &= (A \Rightarrow B, A \Rightarrow \Delta B, \oplus_{\Rightarrow}, +_{\Rightarrow}, 0_{\Rightarrow}) \\ (f \oplus_{\Rightarrow} df)(a) &= f(a) \oplus df(a) \\ (df +_{\Rightarrow} dg)(a) &= df(a) + dg(a) \\ 0_{\Rightarrow}(a) &= 0 \end{aligned}$$

This *almost* works!

The problems: can't guarantee $f \oplus_{\Rightarrow} df$ is differentiable, **ev** isn't differentiable.

Requires higher derivatives!

Another category of difference algebras

The category $\mathbf{DAlg}_\star$

We define the category $\mathbf{DAlg}_\star$ of difference algebras as follows:

- The objects of **DAlg** are difference algebras

$$\overline{A} = (A, \Delta A, \oplus, +, 0)$$

- The morphisms in **DAlg** are pairs $\overline{f} = (f, f')$ where f is a differentiable function and f' is a derivative of it.
- The identity maps are $\mathbf{Id} = (\mathbf{Id}, \pi_2)$
- Composition is given by:

$$(g, g') \circ (f, f') = (g \circ f, g' \circ \langle f \circ \pi_1, f' \rangle)$$

Products and coproducts are like in **DAlg**, but no exponentials!

Outline

Higher-order derivatives

The problems with **DAlg** and $\mathbf{DAlg}_\star$ come from a lack of higher-order derivatives.

To get higher-order derivatives, we require:

- A difference algebra structure on the difference sets ΔA
- A suitable definition of smoothness of a function f

Higher-order derivatives, the easy way

Let $(A, \Delta A, \oplus, +, 0)$ be a difference algebra.

Higher difference algebras

There's a natural difference algebra structure on ΔA given by $(\Delta A, \Delta A, +, +, 0)$.
Whenever $+$ is commutative and $\oplus$ is differentiable w.r.t. its first argument, we will say $(A, \Delta A, \oplus, +, 0)$ is **smooth**.

Higher-order derivatives, the easy way

Smooth functions between difference algebras

Let $\overline{A}, \overline{B}$ be smooth difference algebras. We say that $f : A \to B$ is **smooth** whenever its derivative f' exists and it is a smooth map when considered as a function $f' : A \times \Delta A \to \Delta B$

Essentially, a smooth function f is one for which all higher derivatives $f', f'', \ldots$ exist (with respect to the higher difference algebras described before).

Higher-order derivatives, the easy way

We're not quite there yet. To obtain a CCC, we need to ensure functions have a unique derivative.
Fortunately, there's a very simple condition to impose on a difference algebra to ensure uniqueness of derivatives!

Thin difference algebras

A difference algebra is thin if whenever $a \oplus da = a \oplus db$, it is the case that $da = db$.

Lemma

Let $\overline{B}$ be a difference algebra. Then the following are equivalent:

- $\overline{B}$ is thin.
- For every difference algebra $\overline{A}$ and every funciton $f : A \to B$, f has at most one derivative.

Higher-order derivatives, the easy way

The category $\mathbf{DAlg}_\Delta$

We define the category $\mathbf{DAlg}_\Delta$ of difference algebras as follows:

- The objects of $\mathbf{DAlg}_\Delta$ are smooth, thin difference algebras.
- The morphisms between $\overline{A}$ and $\overline{B}$ in $\mathbf{DAlg}_\Delta$ are smooth functions $f : A \to B$.
- Identity maps and composition are as in **Set**.

It can be shown that $\mathbf{DAlg}_\Delta$ is a CCC, with products and exponentials as described previously for **DAlg**!

Higher-order derivatives, the *hard* way

The previous category imposes too many requirements its objects. It's possible to generalize it in three ways:

- The difference algebra structure on ΔA does not need to be the one that arises from its monoid structure.
- Derivatives do not need to be unique, as long as every map carries its own derivative.
- The monoid operation $+$ does not need to be commutative, as long as it is smooth.

One would expect that it is possible to relax these three conditions and still obtain a CCC.

Higher-order derivatives, the *hard* way

Difference stacks

A **difference stack** $\hat{A}$ is a sequence $(\overline{A}_i)$ where:

- Every $\overline{A}_i$ is a difference algebra.
- $\Delta A_i = A_{i+1}$

A sequence of functions (f_i) is called a smooth map whenever $f_0 : A_0 \to B_0$ and f_{i+1} is a derivative of f_i.

This generalizes difference algebras in the first and second points!

Higher-order derivatives, the *hard* way

Projection of difference stacks

Given a difference stack $\hat{A} = (\overline{A}_i)$, its n-th projection $\Pi^n \hat{A}$ is the difference stack $(\Pi^n \hat{A})_i = (\overline{A}_{i+n})$

Smooth difference stacks

A difference stack $\hat{A}$ is **smooth** whenever every one of its structure maps $\oplus_i, +_i$ is a smooth map between the relevant difference stacks, i.e.

- $\oplus_i$ is a smooth map from $\Pi^i \hat{A} \times \Pi^{i+1} \hat{A}$ into $\Pi^i \hat{A}$
- $+_i$ is a smooth map from $\Pi^{i+1} \hat{A} \times \Pi^{i+1} \hat{A}$ into $\Pi^{i+1} \hat{A}$

This generalizes difference algebras in the third point!

Higher-order derivatives, the *hard* way

The category of smooth difference stacks

We define the category $\mathbf{DStack}_\star$ of difference stacks as follows:

- The objects of $\mathbf{DStack}_\star$ are smooth difference stacks.
- The morphisms between $\hat{A}$ and $\hat{B}$ in $\mathbf{DAlg}_\star$ are smooth maps $\hat{f} = (f_i)$.

Theorem

The category **DStack** is Cartesian closed.

Theorem (Internalization)

There is a smooth map $\partial : \hat{A} \Rightarrow \hat{B} \to \hat{A} \Rightarrow \Delta\hat{B}$ that sends every map to its first derivative.

Outline

Categorifying **DAlg**

All these constructions so far are on **Set**, but the category $\mathbf{DAlg}_\star$ can be built on top of any Cartesian category $\mathbf{C}$

What's more, $\mathbf{DAlg}_\star$ can be raised to an endofunctor on the category of Cartesian categories and product-preserving maps!

Question: does $\mathbf{DStack}_\star$ arise as a projective limit?

$$\begin{tikzcd} & & \mathbf{DStack}_\star(\mathbf{C}) \arrow[dll] \arrow[dl] \arrow[d] \arrow[dr] & \\ \mathbf{C} & \mathbf{DAlg}_\star(\mathbf{C}) \arrow[l] & \mathbf{DAlg}_\star(\mathbf{DAlg}_\star(\mathbf{C})) \arrow[l] & \dots \arrow[l] \end{tikzcd}$$

And is such a construction always a CCC whenever $\mathbf{C}$ is?

The differential geometry of difference algebras

There seems to be a connection between difference algebras and notions in differential geometry.

Derivatives, the chain rule, smoothness, etc. are suggestive names.

Can the connection be made precise?

Conjecture: the Kleisli category of the tangent bundle monad on a Cartesian differential category is a category of difference algebras.

Conjecture: certain categories of difference algebras (when $a \oplus da = a + \epsilon\ da$) where ϵ is a nilpotent homomorphism) are models of synthetic differential geometry

Change structures on domains, recursion and PCF

Change structures can be defined on domains. Is the fixpoint combinator smooth? What is its derivative?

Most of the work already done, answer is (under suitable conditions, like continuity of $\oplus$) yes!

Can this be a model of incremental PCF?

Most importantly: derivatives up-to the domain order:

$$f(x) \oplus f'(x, dx) \sqsubseteq f(x \oplus dx)$$

(with a reasonable maximality condition on $f'(x, dx)$)