Evolving the Incremental $\lambda$-Calculus into a Model of Forward AD

Robert Kelly$^1$ Barak A. Pearlmutter$^1$ Jeffrey Mark Siskind$^2$

$^1$Dept of Computer Science, Maynooth University  $^2$School of Electrical and Computer Engineering, Purdue University

Key Idea

- Formal transformations resembling derivatives common in CS
  - Derivatives of regular expressions (Brzozowski, 1964)
  - Derivatives of types (McBride, 2001; Abbott et al., 2004)
  - Incremental $\lambda$-Calculus (ILC; Cai et al., 2014)

Incremental $\lambda$-Calculus

- $\lambda$ Calculus formalises function definition and application
- ILC adds $\Delta$ to model incremental computation
- $\Delta$ maps a function $f : B \to B$
  - which alters a database $B$
  - to an update function $\Delta f : B \to \Delta B \to \Delta B$
  - where $\Delta B$ is the type of changes to $B$
- Mechanically verifiable proofs of various properties

ILC has Properties Resembling Calculus

$$\Delta (\lambda x . f (g x)) = (\lambda x' . \Delta f (g x) (\Delta g x'))$$

Differences Between ILC and Forward AD

- Propagates changes rather than tangents
  - Changes are elements of change sets
  - Changes are finite, not infinitesimal
  - "Differences" ($\Delta$), not "differentials" ($\delta$)
  - Changes are separate arguments, tangents are bundled with primitives
- ILC wishes to partially evaluate the primal away
- Forward AD does not

Reducing ILC to Forward AD in Three Steps

1. Take the change sets of $\mathbb{R}$ to be power series over $\mathbb{R}$
2. Truncate these power series to dual numbers
3. Uncurry and bundle $f(x)(x') \leadsto f(x,x') \leadsto f(x,x')$

Step 3 Commutes with Steps 1 and 2 Above

Incremental $\lambda$-Calculus  $\Delta R = R[\varepsilon]$  Power Series  $\Delta R = R[\varepsilon]/\varepsilon^2$  Dual Numbers

Bundled ILC  $\Delta R = R[\varepsilon]$  Higher-Order Forward AD  $\Delta R = R[\varepsilon]/\varepsilon^2$  Forward AD

Steps 1 in More Detail (Power Series)

- Consider only change sets on base type $\mathbb{R}$
- Take change sets on $\mathbb{R}$ to be zero-constant-term power series in $\varepsilon$
- This is a valid change set because $\mathbb{R}$ addition is associative
- Augment $\lambda$-Calculus with terms representing power series:
  - $\delta$(ps) := $0$  $\varepsilon * (\delta$p)
  - $\delta$(ps) := $R  [ R + (\delta$(ps))
  - $\Delta R \triangleq \delta$(ps)
- Add mechanism to extract power series coefficients:
  - $\text{coeff } 0 (\lambda \varepsilon . r) \leadsto r$
  - $\epsilon \notin \text{FV}(r)$
  - $\text{coeff } 0 (\lambda \varepsilon . r + \varepsilon * e) \leadsto r$
  - $\epsilon \notin \text{FV}(r)$
  - $\text{coeff } 0 (\lambda \varepsilon . \varepsilon * e) \leadsto 0$
  - $\varepsilon \notin \text{FV}(r)$
  - $\text{coeff } i (\lambda \varepsilon . r + \varepsilon * e) \leadsto \text{coeff } (i - 1) (\lambda \varepsilon . e)$
  - $i > 0 \land \epsilon \notin \text{FV}(r)$
  - $\text{coeff } i (\lambda \varepsilon . \varepsilon * e) \leadsto \text{coeff } (i - 1) (\lambda \varepsilon . e)$
  - $i > 0$
- Examples:
  - $\text{coeff } 2 (\lambda \varepsilon . (0.1 + \varepsilon * (0.2 + \varepsilon * (0.3 + \varepsilon * (0.4 + \varepsilon * (0.5 + \ldots))))) \leadsto 0.3$
  - $\text{diff } f x \equiv \text{coeff } 1 (\lambda \varepsilon . (\Delta f x (\varepsilon * 1)))$

Generalised power series: $\text{coeff}$ distributes over constructors and post-operations over $\lambda$ expressions

$$\text{coeff } i (\lambda \varepsilon . \text{Constructor } e_1 \cdots e_n) \leadsto \text{Constructor } (\text{coeff } i (\lambda \varepsilon . e_1)) \cdots (\text{coeff } i (\lambda \varepsilon . e_n))$$

$$\text{coeff } i (\lambda \varepsilon . (\lambda x . e)) \leadsto (\lambda \varepsilon . \text{coeff } i (\lambda \varepsilon . e)) x \neq \varepsilon$$

Step 2 in More Detail (Truncate Power Series to Dual Numbers)

laziness

Step 3 in More Detail (Uncurrying and Bundling)

$\mathcal{D} : (\alpha_1 \to \alpha_2 \to \cdots \to \alpha_n \to \beta)$

$\to (\alpha_1 \to \Delta \alpha_1 \to \alpha_2 \to \Delta \alpha_2 \to \cdots \to \alpha_n \to \Delta \alpha_n \to \Delta \beta)$

$\mathcal{D} : (\alpha_1 \to \alpha_2 \to \cdots \to \alpha_n \to \beta) \equiv (F \alpha_1 \to F \alpha_2 \to \cdots \to F \alpha_n \to F \beta)$

$F \alpha$ isomorphic to $\alpha \times \Delta \alpha$

$F (\alpha \to \beta) \equiv F \alpha \to F \beta$

$\mathcal{D} : \alpha \to F \alpha$

$\mathcal{D} (f \circ g) \leadsto \mathcal{D} f \circ \mathcal{D} g$

Differences from Related Work

- Framework for machine-verified proofs of correctness and efficiency
- The Simply Typed $\lambda$-Calculus of Forward Automatic Differentiation (Manzyuk, 2012) has confluence issues, and conflates numeric basis functions which operate on $\mathbb{R}$ with those lifted to Dual numbers
- The Differential $\lambda$-Calculus (Ehrhard and Regnier, 2003) does not guarantee complexity and does not segregate levels of differentiation

Take-Home Message

- The PL. Theory community has developed methods for proving that nonstandard interpretations preserve axioms
- Some of these methods have been automated
- AD is a nonstandard interpretation
- These methods can be used to prove the correctness of AD
- We are constructing a machine-verified proof

Bibliography

C. McBride. The derivative of a regular type is its type of one-hole coinduction, 2001.