Evolving the Incremental λ Calculus into a Model of Forward AD

Maynooth University National University of Ireland Maynooth

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

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 λ-Calculus (ILC; Cai et al., 2014)

Incremental λ-Calculus

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

ILC has Properties Resembling Calculus

$$\mathcal{D}\ (\lambda\ x\ .\ f\ (g\ x)) = (\lambda\ x\ x'\ .\ \mathcal{D}\ f\ (g\ x)\ (\mathcal{D}\ g\ x\ x'))$$

$$\mathcal{D}\ (f \circ g)\ x = \mathcal{D}\ f\ (g\ x) \circ \mathcal{D}\ g\ x$$

Differences Between ILC and Forward AD

- Propagates *changes* rather than *tangents*
  - Changes are elements of *change sets*
  - Changes are finite, not infinitesimal
  - "Differences" (Δ), not "differentials" (∂)
- Changes are separate arguments, tangents are bundled with primals
  - ILC wishes to partial 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

$$f(x)(x') \rightsquigarrow f(x, x') \rightsquigarrow f(\langle x, x' \rangle)$$

and also return primal

$$\mathcal{D} : (\alpha \to \beta) \to (\alpha \to \Delta\alpha \to \Delta\beta)$$
$$\rightsquigarrow \hat{\mathcal{D}} : (\alpha \to \beta) \to ((\alpha \times \Delta\alpha) \to (\beta \times \Delta\beta))$$

Step 3 Commutes with Steps 1 and 2 Above

$$\begin{tikzcd}
\text{Incremental } \lambda\text{-Calculus} \arrow[r, "\Delta\mathbb{R} = \mathbb{R}[\varepsilon]"] \arrow[d, "\text{uncurry}"] & \text{Power Series} \arrow[r, "\Delta\mathbb{R} = \mathbb{R}[\varepsilon]/\varepsilon^2"] \arrow[d, "\text{uncurry}"] & \text{Dual Numbers} \arrow[d, "\text{uncurry}"] \\
\text{Bundled ILC} \arrow[r, "\Delta\mathbb{R} = \mathbb{R}[\varepsilon]"] & \text{Higher-Order Forward AD} \arrow[r, "\Delta\mathbb{R} = \mathbb{R}[\varepsilon]/\varepsilon^2"] & \text{Forward AD}
\end{tikzcd}$$

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 ε
- This is a valid change set because $\mathbb{R}$ addition is associative
- Augment λ-Calculus with terms representing power series:

$$\langle \mathbf{zps}_\varepsilon \rangle ::= 0 \mid \varepsilon * \langle \mathbf{ps}_\varepsilon \rangle$$
$$\langle \mathbf{ps}_\varepsilon \rangle ::= \mathbb{R} \mid \mathbb{R} + \langle \mathbf{zps}_\varepsilon \rangle$$

$$\Delta\mathbb{R} \triangleq \langle \mathbf{zps}_\varepsilon \rangle$$

- Add mechanism to extract power series coefficients:

$$\begin{aligned}
\mathbf{coeff}\ 0\ (\lambda\varepsilon\ .\ r) &\rightsquigarrow r & \varepsilon \notin \mathrm{FV}(r) \\
\mathbf{coeff}\ 0\ (\lambda\varepsilon\ .\ r + \varepsilon * e) &\rightsquigarrow r & \varepsilon \notin \mathrm{FV}(r) \\
\mathbf{coeff}\ 0\ (\lambda\varepsilon\ .\ \varepsilon * e) &\rightsquigarrow 0 & \\
\mathbf{coeff}\ i\ (\lambda\varepsilon\ .\ r + \varepsilon * e) &\rightsquigarrow \mathbf{coeff}\ (i-1)\ (\lambda\varepsilon\ .\ e) & i > 0 \land \varepsilon \notin \mathrm{FV}(r) \\
\mathbf{coeff}\ i\ (\lambda\varepsilon\ .\ \varepsilon * e) &\rightsquigarrow \mathbf{coeff}\ (i-1)\ (\lambda\varepsilon\ .\ e) & i > 0
\end{aligned}$$

- Examples:

$$\mathbf{coeff}\ 2\ (\lambda\varepsilon\ .\ 0.1 + \varepsilon * (0.2 + \varepsilon * (0.3 + \varepsilon * (0.4 + \varepsilon * (0.5 + \cdots))))) \rightsquigarrow 0.3$$

$$\mathbf{diff}\ f\ x \triangleq \mathbf{coeff}\ 1\ (\lambda\varepsilon\ .\ (\mathcal{D}\ f\ x\ (\varepsilon * 1)))$$

- Generalised power series: **coeff** distributes over constructors and post-composes over λ expressions

$$\mathbf{coeff}\ i\ (\lambda\varepsilon\ .\ \mathbf{Constructor}\ e_1\ \cdots\ e_n)$$
$$\rightsquigarrow \mathbf{Constructor}\ (\mathbf{coeff}\ i\ (\lambda\varepsilon\ .\ e_1))\ \cdots\ (\mathbf{coeff}\ i\ (\lambda\varepsilon\ .\ e_n))$$

$$\mathbf{coeff}\ i\ (\lambda\varepsilon\ .\ (\lambda x\ .\ e)) \rightsquigarrow (\lambda x\ .\ \mathbf{coeff}\ i\ (\lambda\varepsilon\ .\ e)) \qquad x \neq \varepsilon$$

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

laziness

Step 3 in More Detail (Uncurrying and Bundling)

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

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

- $F\alpha$ isomorphic to $\alpha \times \Delta\alpha$
- $F(\alpha \to \beta) \triangleq F\alpha \to F\beta$
- $\hat{\mathcal{D}} : \alpha \to F\alpha$
- $\hat{\mathcal{D}}\ (f \circ g) \rightsquigarrow \hat{\mathcal{D}}\ f \circ \hat{\mathcal{D}}\ g$

Differences from Related Work

- Framework for machine-verified proofs of correctness and efficiency
- The Simply Typed λ-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 λ-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 use to prove the correctness of AD
- We are constructing a machine-verified proof

Bibliography

M. Abbott, N. Ghani, T. Altenkirch, and C. Mcbride. ∂ for data: Differentiating data structures. *Fundamenta Informaticae*, 65(1-2):1–28, 2004.

J. A. Brzozowski. Derivatives of regular expressions. *JACM*, 11(4):481, 1964.

Y. Cai, P. G. Giarrusso, T. Rendel, and K. Ostermann. A theory of changes for higher-order languages: Incrementalizing λ-calculi by static differentiation. In *POPL*, pages 145–55, 2014.

T. Ehrhard and L. Regnier. The differential lambda-calculus. *TCS*, 309(1-3):1–41, 2003.

O. Manzyuk. A simply typed λ-calculus of forward automatic differentiation. In *Mathematical Foundations of Programming Semantics*, pages 259–73, 2012.

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

This work was supported, in part, by Science Foundation Ireland grant 09/IN.1/I2637 and by NSF grant 1522954-IIS. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.