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

Robert Kelly¹, Barak A. Pearlmutter¹, Jeffrey Mark Siskind²
¹Dept of Computer Science, Maynooth University ²School of Electrical and Computer Engineering, Purdue University

Key Ideas
- 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)

\[ \Delta R = R[\epsilon] \]
uncurry
\[ \Delta R = R[\epsilon]/\epsilon^2 \]
uncurry
\[ \Delta R = R[\epsilon]/\epsilon^2 \]
uncurry

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

ILC has Properties Resembling Calculus
\[ D(\lambda x. f \,(g \,x)) = (\lambda x \,x' \cdot D \,(f \,(g \,x \,x')) \]
\[ D \,(f \circ g) \,x = D \,(f \,(g \,x)) \circ D \,g \,x \]

Steps 1 in More Detail (Power Series)
- Consider only change sets on base type $R$
- Take change sets on $R$ to be zero-constant-term power series in $\epsilon$
- This is a valid change set because $R$ addition is associative
- Augment $\lambda$-Calculus with terms representing power series:
  \[ \langle \psi \rangle \ ::= 0 \mid \epsilon \ast \langle \psi \rangle \]
  \[ \langle \psi \rangle \ ::= R \mid \langle \psi \rangle + \langle \psi \rangle \]

Steps 3 Commutes with Steps 1 and 2 Above

Step 3 in More Detail (Uncurrying and Bundling)
\[ D : (\alpha_1 \to \alpha_2 \to \cdots \to \alpha_n \to \beta) \]
\[ \to (\alpha_1 \to \Delta \alpha_1 \to \alpha_2 \to \Delta \alpha_2 \to \cdots \to \alpha_n \to \Delta \alpha_n \to \Delta \beta) \]
\[ D : (\alpha_1 \to \alpha_2 \to \cdots \to \alpha_n \to \beta) \to (F \alpha_1 \to F \alpha_2 \to \cdots \to F \alpha_n \to F \beta) \]

Differences from Related Work
- Framework for machine-verifiable proofs of correctness and efficiency
- The Simply Typed $\lambda$-Calculus of Forward Automatic Differentiation (Manzuy, 2012) has confluence issues, and conflates numeric basis functions which operate on $R$ 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

Reducing ILC to Forward AD in Three Steps
1. Take the change sets of $R$ to be power series over $R$
2. Truncate these power series to dual numbers
3. Uncurry and bundle

\[ f(x)(x') \leftrightarrow f(x,x') \leftrightarrow f(x,x') \]
and return primal
\[ D : (\alpha \to \beta) \to (\alpha \to \Delta \alpha \to \Delta \beta) \]
\[ \cong D : (\alpha \to \beta) \to (\alpha \times \Delta \alpha \to (\beta \times \Delta \beta)) \]

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