Homomorphic reverse differentiation for partial features

Fernando Lucatelli Nunes*

Utrecht University
fernandolucatellinunes@gmail.com

Reverse mode Automatic Differentiation (AD), a fundamental technique leveraging the chain rule to compute derivatives of functions implemented by programs, plays an integral role in addressing the challenges of efficiently computing derivatives of functions with high-dimensional domains while maintaining numerical stability. Its widespread application in machine learning and scientific computing underlines its significance.

Within the programming languages (PL) community, the pursuit of a simple, easily implementable, and purely functional reverse AD that can be performed at compile time through source-code transformation has been a central objective. The realization of this goal hinges on the development of an AD source-code transformation method underpinned by a well-defined denotational semantics.

Reverse mode Combinatory Homomorphic Automatic Differentiation (CHAD) [1, 3, 11], a paradigm that can be seen as an extension of Elliot's simple essence of automatic differentiation, implements reverse AD following a principled denotational semantics; namely, we define it as a uniquely defined structure-preserving functor. Following this principle, CHAD provides us with a reverse mode AD whose categorical semantics is the dual of the corresponding forward mode. The correctness proof follows by a logical relations (LR) argument [8, 2].

CHAD has been shown to apply to expressive total languages, involving sum, function and (co)inductive types, by making use of a target language with linear dependent types [3]. On the implementation side, CHAD can be made efficient (complexity-preserving) [9], and it is parallelism preserving.

A remaining semantic challenge is to apply CHAD to languages with partial features. More precisely, we are particularly interested in understanding CHAD for languages with iteration (while-loops) and term recursion, as both are common features exploited in Machine Learning and Scientific Computing.

In this talk, we will explore CHAD's categorical denotational semantics for languages with iteration [5] and term recursion, based on ongoing work [5, 6, 7]. We begin by incorporating partiality into the framework of [3], then demonstrate that we can add iteration and prove correctness through a logical relations argument [8, 2]. This argument revolves around preserving the properties of the differentiation functor within the CHAD setting.

If time permits, we will also discuss the addition of recursion. This addition raises questions about the reconciliation and coherence of domain theory, topology, linear types, and differentiation—or more broadly, sheaf theory and domain theory.

In our work [6, 7], we present two solutions: one simply typed and another dependently typed. Both solutions rely on novel concrete models for effectful dependent type theories and, more importantly, Dialectica formulas as presented in [4].

*This is joint work with Matthijs Vákár and Gordon Plotkin.

References

[1] Conal Elliott. 2018. The simple essence of automatic differentiation. Proc. ACM Program. Lang. 2, ICFP (2018), 70:1–70:29.

[2] Fernando Lucatelli Nunes and Matthijs Vákár. Automatic Differentiation for ML-family languages: correctness via logical relations. (Accepted for publication – Mathematical Structures in Computer Science).

[3] Fernando Lucatelli Nunes and Matthijs Vákár. CHAD for Expressive Total Languages. Mathematical Structures in Computer Science 33(4-5), 311-426 (2023).

[4] Fernando Lucatelli Nunes and Matthijs Vákár. Monoidal closure of Grothendieck constructions via Σ-tractable monoidal structures and Dialectica formulas. arXiv:2405.07724. 2024.

[5] Fernando Lucatelli Nunes, Matthijs Vákár and Gordon Plotkin. Homomorphic Reverse Differentiation of Iteration. LAFI 24.

[6] Fernando Lucatelli Nunes, Matthijs Vákár and Gordon Plotkin. Unraveling the iterative CHAD (In preparation).

[7] Fernando Lucatelli Nunes, Matthijs Vákár and Gordon Plotkin. Recursive CHAD (In preparation).

[8] Gordon Plotkin. Lambda-definability and logical relations. Memorandum. School of Artificial Intelligence, University of Edinburgh. 1973

[9] Tom Smeding and Matthijs Vákár. 2023. Efficient CHAD. ArXiv:2307.05738.

[10] Matthijs Vákár. 2021. Reverse AD at Higher Types: Pure, Principled and Denotationally Correct. ESOP 2021, 607–634.

[11] Matthijs Vákár and Tom Smeding. 2021. CHAD: Combinatory Homomorphic Automatic Differentiation. ACM Trans. Program. Lang. Syst. 44(3): 20:1-20:49 (2022).