Hascoët, Laurent; **Araya-Polo, Mauricio**

The adjoint data-flow analyses: formalization, properties, and applications. (English)

Zbl 1270.65089

Bücker, Martin (ed.) et al., Automatic differentiation: Applications, theory, and implementations. Selected papers based on the presentation at the 4th international conference on automatic differentiation (AD), Chicago, IL, USA, July 20–23, 2004. Berlin: Springer (ISBN 3-540-28403-6/pbk). Lecture Notes in Computational Science and Engineering 50, 135-146 (2006).

Summary: Automatic Differentiation (AD) is a program transformation that yields derivatives. Building efficient derivative programs requires complex and specific static analysis algorithms to reduce run time and memory usage. Focusing on the reverse mode of AD, which computes adjoint programs, we specify jointly the central static analyses that are required to generate an efficient adjoint code. We use a set-based formalization from classical data-flow analysis to specify adjoint liveness, adjoint write, and to be recorded analyses, and their mutual influences, taking into account the specific structure of adjoint programs. We give illustrations on examples taken from real numerical programs, that we differentiate with our AD tool TAPENADE, which implements these analyses.

For the entire collection see [Zbl 1084.65002].

MSC:

65Y99 Computer aspects of numerical algorithms
54D25 "P-minimal" and "P-closed" spaces

Keywords:

adjoint code; adjoint algorithm; data-flow analysis; reverse mode; TAPENADE

Software:

TAPENADE