Generalizing Shape Analysis with Gradual Types (Artifact)

Zeina Migeed
University of California, Los Angeles (UCLA), CA, USA
Meta Platforms Inc., USA

Jens Palsberg
University of California, Los Angeles (UCLA), CA, USA

Abstract

This is the artifact for the conference article below.

2012 ACM Subject Classification Software and its engineering → Software notations and tools; Software and its engineering → General programming languages

Keywords and phrases Gradual Types, Type Theory

Digital Object Identifier 10.4230/DARTS.10.2.14

Related Article Zeina Migeed, James Reed, Jason Ansel, and Jens Palsberg, "Generalizing Shape Analysis with Gradual Types", in 38th European Conference on Object-Oriented Programming (ECOOP 2024), LIPIcs, Vol. 313, pp. 29:1–29:28, 2024.
`https://doi.org/10.4230/LIPIcs.ECOOP.2024.29`

Related Conference 38th European Conference on Object-Oriented Programming (ECOOP 2024), September 16–20, 2024, Vienna, Austria

Evaluation Policy The artifact has been evaluated as described in the ECOOP 2024 Call for Artifacts and the ACM Artifact Review and Badging Policy.

1 Scope

This artifact demonstrates our techniques for shape reasoning on a suite of 15 PyTorch benchmarks. The artifact shows that we can verify the claims made by our paper. Specifically, The data in Figures 9, 10 and 11 are supported by the artifact.

The problems in the paper correspond to the artifact components as follows

- Solving the migration problem as a constraint satisfiability problem, as described in section 4
- Using the constraint satisfiability problem as a basis for control-flow elimination, as described in section 5
- Testing our algorithms on three tracers, as described in section 6

The purpose of this artifact is to create a core tool, which can reason about the shapes of tensor programs in a generalized way. The generality comes from the fact that our tool can take additional custom constraints as input, to answer questions about the migration space, which are beyond the three key questions that we have listed in our paper.

2 Content

In the home directory of our artifact, there are three directories:
`ECOOP_main_artifact`, `torchdynamo`, and `transformers`.
The test cases for figure 9 can be found and run at:
`ECOOP_main_artifact/test/fx/test_z3_gradual_types.py`
The test cases for figures 10 can be found and run at:
`transformers/tests/gradual_types/test_gradual_types.py`

© Zeina Migeed and Jens Palsberg;
licensed under Creative Commons License CC-BY 4.0

DAGSTUHL ARTIFACTS SERIES
Dagstuhl Artifacts Series
Schloss Dagstuhl – Leibniz-Zentrum für Informatik,
Dagstuhl Publishing, Germany

Artifacts Available ECOOP V1.1
Artifacts Evaluated ECOOP Reusable V1.1

The test cases for figure 11 can be found at:
`torchdynamo/torchdynamo/torch_dynamo_benchmarks`.
There are 5 files, one for each benchmark. Each file can be run separately in Python

3 Getting the artifact

The artifact endorsed by the Artifact Evaluation Committee is available free of charge on the Dagstuhl Research Online Publication Server (DROPS). In addition, the artifact is also available at: `https://github.com/users/ecoop2024/packages/container/package/ecoop_artifact`.

4 Tested platforms

We ran our artifact on a Mac Book Pro with an 8-Core CPU, 14-Core GPU and 512GB DRAM

5 MD5 sum of the artifact

560e9e3bfc86c7b557515c734ea28218

6 Size of the artifact

7,952,189,182 Bytes