

Mina 1.0

A Hybrid-Partial Evaluator for Scala

<http://www.amanj.me/projects/mina>

Amanj Sherwany

<http://www.amanj.me>

Draft of 20th May 2013

Contents

1	Introduction	2
1.1	What is Mina?	2
1.2	Partial Evaluation	2
2	How to use Mina	2
3	Use cases	2
4	Credits	2

1 Introduction

1.1 What is Mina?

Mina is a Partial Evaluator for Scala. It is a port of Civet for Java, which is an implementation of Hybrid-Partial Evaluator paper (<http://www.cs.utexas.edu/~wcook/Civet/>). “Hybrid partial evaluation (HPE) is a pragmatic approach to partial evaluation that borrows ideas from both online and offline partial evaluation. HPE performs offline-style specialization using an online approach without static binding time analysis. The goal of HPE is to provide a practical and predictable level of optimization for programmers, with an implementation strategy that fits well within existing compilers or interpreters. HPE requires the programmer to specify where partial evaluation should be applied. It provides no termination guarantee and reports errors in situations that violate simple binding time rules, or have incorrect use of side effects in compile-time code.”¹

Mina is a Scala compiler plugin, runs after the **patmat** phase, and partially evaluates Scala source code to a more efficient code, which has exactly the same meaning as the original one.

The plugin name is mina and it is read like */mi:na:/*, which is a Kurdish name of Forget-me-not flower <http://en.wikipedia.org/wiki/Forget-me-not>.

1.2 Partial Evaluation

“In computing, partial evaluation is a technique for several different types of program optimization by specialization. The most straightforward application is to produce new programs which run faster than the originals while being guaranteed to behave in the same way.

A computer program, *prog*, is seen as a mapping of input data into output data:

$$prog : I_{static} \times I_{dynamic} \rightarrow O$$

I_{static} the static data, is the part of the input data known at compile time. The partial evaluator transforms $\langle prog, I_{static} \rangle$ into $prog^* : I_{dynamic} \rightarrow O$ by precompiling all static input at compile time. $prog^*$ is called the “residual program” and should run more efficiently than the original program. The act of partial evaluation is said to “residualize” *prog* to $prog^*$.”²

2 How to use Mina

3 Use cases

4 Credits

¹from Civet <http://www.cs.utexas.edu/~wcook/Civet/>

²from Wikipedia http://en.wikipedia.org/wiki/Partial_evaluation