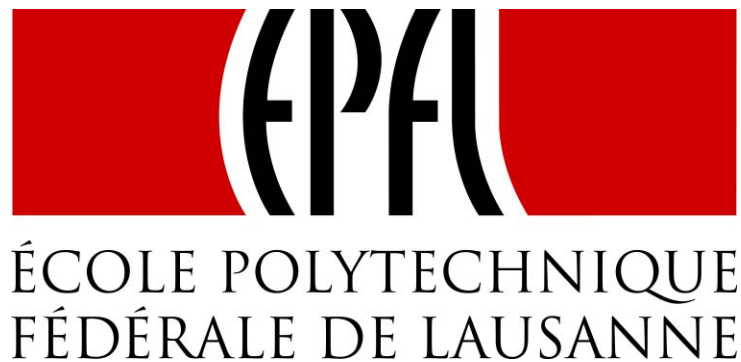


# New lms frontend and staged computation graphs<sup>1</sup>

**Ruben Fiszel**

December 5, 2016



<sup>1</sup>Thanks to my beloved parents, my awesome supervisor Nada Amin, Prof. Martin Odersky, the lms master and author Tiark Rompf, and the delite folks Kevin James Brown and David Koeplinger.

## **Abstract**

In this report, we explore staging, in particular the LMS framework and the development of its new frontend whose aim is ease the writing of staged dsl through, among others, shadowing of types. We also explore the usage of this new frontend for a particular case study: Staged computation graphs.

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>LMS</b>	<b>3</b>
	Staging . . . . .	3
	Why staging . . . . .	3
	Exp tree . . . . .	3
	Deep reuse of embedding language order . . . . .	3
	Frontend/Backend ? . . . . .	3
	lms . . . . .	3
	library author . . . . .	3
	delite . . . . .	3
	user . . . . .	3
<b>3</b>	<b>The new frontend</b>	<b>4</b>
	Lift . . . . .	4
	Typeclass . . . . .	4
	Typeclass overloading . . . . .	4
	Primitives types and collections . . . . .	4
<b>4</b>	<b>Computation Graph</b>	<b>5</b>
	Graph . . . . .	5
	Cycle check . . . . .	5
	Arithmetic . . . . .	5
	Benchmark . . . . .	5
	DerivableGraph . . . . .	5
	Backpropagation . . . . .	5
	MatrixGraph . . . . .	5
	Dimensions check . . . . .	5
<b>5</b>	<b>Conclusion</b>	<b>6</b>

# Introduction

bla bla bla

# LMS

Staging

Why staging

Exp tree

Deep reuse of embedding language order

Frontend/Backend ?

lms

library author

delite

user

# The new frontend

Lift

Typeclass

Typeclass overloading

Primitives types and collections

# Computation Graph

Graph

Cycle check

Arithmetic

Benchmark

DerivableGraph

Backpropagation

MatrixGraph

Dimensions check

# Conclusion

blablabla