

ECE491 Advanced compilers final report

Implementing a lazy functional language

Jonathan Lam

2022/05/12

Contents

1	Project overview	2
1.1	Motivation	2
1.2	Commentary on the tutorial	2
1.3	Implementation setup	2
2	Background	2
2.1	Definition of a programming language	2
2.2	Implementations of programming language	2
2.3	The untyped λ -calculus	2
2.4	Functional programming	2
3	The Core language	2
3.1	Syntax	2
3.2	Dynamic semantics	2
3.3	Sample programs	2
3.4	Data structures	2
3.5	Lexer	2
3.6	Parser	2
3.7	Pretty-print utility	2
4	Template instantiation (TI) evaluator	2
5	G-Machine (GM) compiler	2

1 Project overview

1.1 Motivation

1.2 Commentary on the tutorial

1.3 Implementation setup

TODO: github repo (all three, including this one)

2 Background

2.1 Definition of a programming language

TODO: syntax, semantics

TODO: notation used for dynamic semantics here: state machine?

2.2 Implementations of programming language

TODO: compilers, interpreters

2.3 The untyped λ -calculus

2.4 Functional programming

3 The Core language

3.1 Syntax

3.2 Dynamic semantics

3.3 Sample programs

3.4 Data structures

3.5 Lexer

3.6 Parser

3.7 Pretty-print utility

4 Template instantiation (TI) evaluator

5 G-Machine (GM) compiler