

Building Compilers in Pharo

G. Polito, P. Tesone, S. Ducasse

August 23, 2024

Copyright 2017 by G. Polito, P. Tesone, S. Ducasse.

The contents of this book are protected under the Creative Commons Attribution-ShareAlike 3.0 Unported license.

You are **free**:

- to **Share**: to copy, distribute and transmit the work,
- to **Remix**: to adapt the work,

Under the following conditions:

Attribution. You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

Share Alike. If you alter, transform, or build upon this work, you may distribute the resulting work only under the same, similar or a compatible license.

For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to this web page:

<http://creativecommons.org/licenses/by-sa/3.0/>

Any of the above conditions can be waived if you get permission from the copyright holder. Nothing in this license impairs or restricts the author's moral rights.



Your fair dealing and other rights are in no way affected by the above. This is a human-readable summary of the Legal Code (the full license):

<http://creativecommons.org/licenses/by-sa/3.0/legalcode>

Contents

Illustrations	iii
I A Stack Machine	
1 A stack based language (and the stack)	3
2 Compiling AST to stack code	5
3 A stack based interpreter	7
4 Calling conventions, stack management (FP, SP...)	9
5 Control flow (if)	11
5.1 Basic Block	11
6 Loops: Back Jump	13
II A Register Machine	
7 Assembly Overview (maybe our own simple assembly?)	17
8 Compiling AST to register code - first version	19
9 Compiling Stack Code to Register code	21
10 Calling convention revisited	23
III Static Single Assignment	
11 SSA	27
12 Building SSA from AST	29

13	Building SSA from stack code	31
IV Code Transformations		
14	Fixed Point algorithms	35
15	Optimization a	37
16	Optimization b	39
17	Register Allocation	41
	Bibliography	43

Illustrations

Part I

A Stack Machine

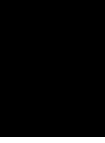


A stack based language (and the stack)

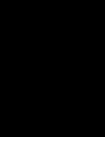


Compiling AST to stack code

CHAPTER 3

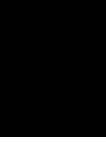


A stack based interpreter



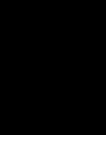
Calling conventions, stack management (FP, SP...)

CHAPTER 5



Control flow (if)

5.1 Basic Block



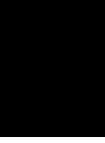
Loops: Back Jump

Part II

A Register Machine



Assembly Overview (maybe our own simple assembly?)



Compiling AST to register code - first version



Compiling Stack Code to Register code



Calling convention revisited

Part III

Static Single Assignment

CHAPTER

11



SSA

CHAPTER

12



Building SSA from AST



Building SSA from stack code

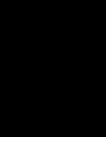
Part IV

Code Transformations



Fixed Point algorithms

CHAPTER 15



Optimization a

CHAPTER

16



Optimization b

CHAPTER

17



Register Allocation

Bibliography

