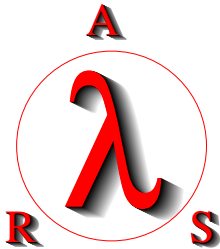


A++ and the Lambda Calculus

Georg P. Loczewski



A++ and the Lambda Calculus

Principles of Functional Programming

 tredition®
www.tredition.de

IMPRESSUM

Copyright ©2018 Georg P. Loczewski
A++ and the Lambda Calculus

The book was set by the author using the $\text{\LaTeX}$ typesetting system and was printed and bound in the Federal Republic of Germany.

1st. Edition 2018
tredition GmbH, Hamburg

ISBN
978-3-7469-3811-0 (Paperback)
978-3-7469-3809-7 (Hardcover)
978-3-7469-3810-3 (e-Book)

See also A++ *The Smallest Programming Language in the World*[29]
Printed in Germany

The author and publisher make no warranty of any kind, expressed or implied, with regard to these programs or the documentation contained in this book. The author and publisher shall not be liable in any event for incidental or consequential damages in connection with the use of these programs.

All rights reserved. No part of this book may be reproduced in any form by any electronic or mechanical means (including photocopying, recording, or information storage and retrieval) without permission in writing from the publisher and the author.

The book was set by the author using the $\text{\LaTeX}$ typesetting system and was printed and bound in the Federal Republic of Germany.

*To my wife Ursula
and my sons Thomas and Johannes
dedicated in love.*



Contents

I	The Lambda Calculus	1
1	Introduction	3
1.1	Origin	3
1.2	Definition	3
1.3	Literature	3
1.4	Syntax of Lambda Expressions	4
2	Basic Rules for Lambda Conversions	5
2.1	Notation used in Conversion Rules	5
2.1.1	Notation used to specify conversion of lambda expressions	5
2.1.2	Notation used to specify substitution	6
2.2	Alpha Conversion	6
2.2.1	Rules of Alpha Conversion	6
2.2.2	Beta Conversion	7
2.2.3	Rule of β -Conversion	7
2.2.4	β -Reduction	7
2.2.5	β -Abstraction	8
2.3	Eta Conversion	8
2.3.1	η -Reduction	8
2.3.2	η -Abstraction	9

2.4	Rules of Associativity	9
2.4.1	Rule of Associativity for Abstraction	9
2.4.2	Rule of Associativity for Application	10
2.4.3	Example for both rules:	10
2.5	Y-Combinator	10
2.5.1	Using the Y-Combinator to implement recursion	12
II	A++	15
1	Introduction to A++	17
1.1	Purpose of A++ and Origin	17
1.1.1	Purpose	17
1.1.2	Origin	17
1.1.3	ARS — <i>Generalization of the Lambda-Calculus</i>	18
1.1.4	Name of the language	19
1.2	Motivations for the development of A++	19
1.2.1	To support an alternate method of teaching the principles of programming	19
1.2.2	To provide a learning tool for exploring the fundamentals of programming	19
1.2.3	To support a method teaching powerful programming patterns applicable to most languages	20
1.2.4	To open a new view of programming for many programmers:	20
1.3	Features of A++	20
1.3.1	Programming paradigms supported:	20
1.3.2	Constitutive Principles of A++	20

1.3.3	Closure	22
1.3.4	Basic abstractions derived from ARS	22
1.3.5	Development of Applications with A++	23
1.4	Internal Architecture of A++	24
1.4.1	Overview	24
1.4.2	Internal Structure Overview (commented version)	25
1.4.3	Syntax of ARS (A++)	26
1.4.4	Definition of A++ in the form of a Tree Diagram	27
1.4.5	Commenting the definition:	27
1.4.6	Examples of A++ - Syntax	28
2	General Programming Patterns and A++	31
2.1	Closure Pattern	31
2.2	CLAM Pattern	32
2.3	List Pattern	32
2.4	Dictionary Pattern	34
2.5	Set Pattern	35
2.6	Recursion Pattern	35
2.7	Higher Order Function Pattern	36
2.8	Message Passing Pattern	36
2.8.1	Classes of objects	36
2.8.2	Instance of a class	38
2.8.3	Constructors	38
2.8.4	Creating instances of a class	39
2.8.5	Sending messages	40
2.8.6	Executing methods	41

2.8.7	Essential features of object oriented programming	43
2.8.8	Relation between classes	44
2.9	Meta Object Protocol Pattern	44
Bibliography		45

Part I

The Lambda Calculus

