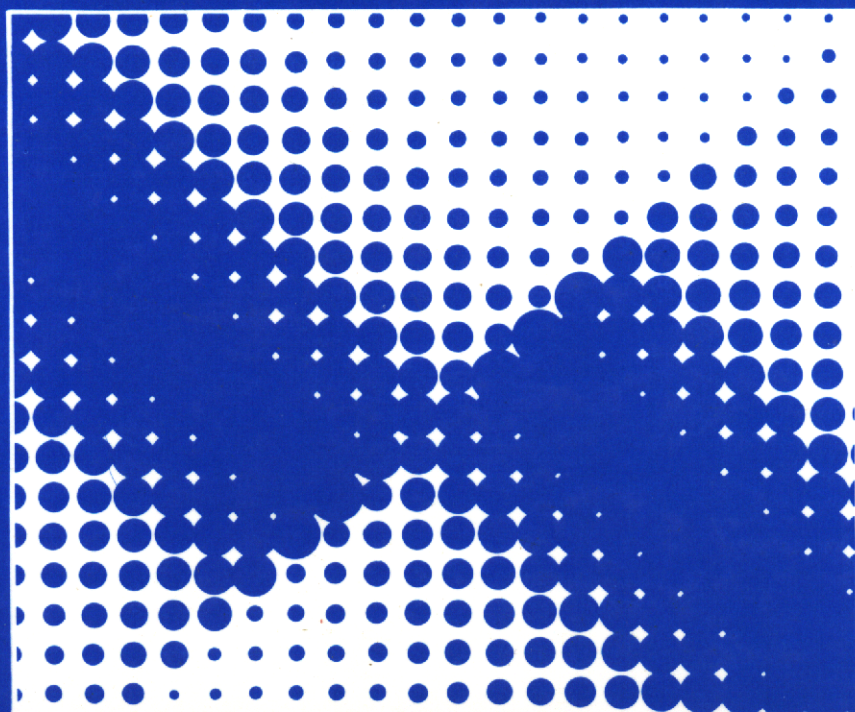


# **Lucid, the Dataflow Programming Language**

**William W. Wadge  
Edward A. Ashcroft**





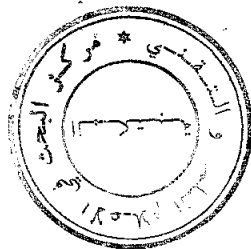
---

**LUCID,  
THE DATAFLOW  
PROGRAMMING LANGUAGE**

---

# BIBLIOTHEQUE DU CERIST

*This is volume 22 in A.P.I.C. Studies in Data Processing*  
*General Editors: Fraser Duncan and M. J. R. Shave*  
*A complete list of titles in this series appears at the end of this volume.*

C  
1883/1

---

# **LUCID, THE DATAFLOW PROGRAMMING LANGUAGE**

---

William W. Wadge

Department of Computer Science  
University of Victoria  
Victoria, British Columbia  
Canada

Edward A. Ashcroft

SRI International  
Menlo Park, California  
United States of America

1985



ACADEMIC PRESS

(Harcourt Brace Jovanovich, Publishers)

London Orlando San Diego New York  
Toronto Montreal Sydney Tokyo

COPYRIGHT © 1985, BY ACADEMIC PRESS INC. (LONDON) LTD.  
ALL RIGHTS RESERVED.

NO PART OF THIS PUBLICATION MAY BE REPRODUCED OR  
TRANSMITTED IN ANY FORM OR BY ANY MEANS, ELECTRONIC  
OR MECHANICAL, INCLUDING PHOTOCOPY, RECORDING, OR  
ANY INFORMATION STORAGE AND RETRIEVAL SYSTEM, WITHOUT  
PERMISSION IN WRITING FROM THE PUBLISHER.

ACADEMIC PRESS INC. (LONDON) LTD.  
24-28 Oval Road  
LONDON NW1 7DX

*United States Edition published by*  
ACADEMIC PRESS, INC.  
Orlando, Florida 32887

**British Library Cataloguing in Publication Data**

Wadge, William W.

Lucid, the dataflow programming language.—  
(APIC studies in data processing)

1. Programming languages (Electronic computers)
2. Electronic digital computers—Programming

I. Title II. Ashcroft, Edward A. III. Series  
001.64'24 QA76.7

**Library of Congress Cataloging in Publication Data**

Wadge, William W.

Lucid, the dataflow programming language.  
(A.P.I.C. studies in data processing; no. )

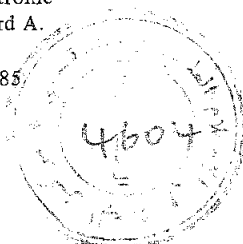
Bibliography: p.  
Includes index.

1. Lucid (Computer program language)
  2. Electronic digital computers—Programming.
- I. Ashcroft, Edward A.  
II. Title. III. Series.

QA76.73.L83W33 1985 001.64'24 84-15685  
ISBN 0-12-729650-6 (alk. paper)

PRINTED IN THE UNITED STATES OF AMERICA

85 86 87 88 9 8 7 6 5 4 3 2 1



---

# CONTENTS

---

PREFACE	ix
I INTRODUCTION	
1. What Makes Lucid Run?	3
2. Imperative Programming	7
3. Control Flow versus Dataflow	10
4. The Problems with the Imperative Approach	15
5. Some Suggested Solutions	19
6. Honest, Folks, We Was Just Funnin'	22
7. Nonprocedural Languages	24
8. Lucid, the Well-Balanced Language	28
9. Lucid—The Dataflow Programming Language?	34
II ISWIM AND ALGEBRAS	
1. Algebras	41
2. The Syntax of Iswim	44
3. The Semantics of Iswim	46
4. A Sample Program	48
5. The Operational Semantics of Iswim	49
6. Calling and Binding in Iswim	51
7. Lazy Evaluation	55
8. Infinitary Programming	57
III THE LANGUAGE LUSWIM	
1. Iteration through Sequences	63
2. Luswim: The Abstract Approach to Iteration	66
3. Iterative Programs in Luswim	68
4. Variables in PASCAL and Luswim	73
5. Continuously Operating Programs	78
6. Programs with Memory	80
7. Functions in Luswim	82
8. Continuously Operating Functions	85

9.	Filters and Recursion	87
10.	Programs and Flow Graphs	90
11.	Implementing Luswim	92
IV	LUCID: LUSWIM WITH NESTED ITERATION	
1.	The Operation <b>asa</b>	101
2.	Nesting in HyperpIswwim	103
3.	Nesting with <b>whenever</b> and <b>upon</b>	106
4.	Going beyond Luswim	106
5.	The <b>is current</b> Declaration	109
6.	Mixed Clauses	114
V	PROGRAMMING IN LUCID	
1.	Hamming's Problem, a Simple Dataflow Example	121
2.	Using Dataflow to Enumerate the Primes	124
3.	Processing Text as Streams of Characters	130
4.	A Data Processing Example	132
5.	Swap Sorting in Lucid	135
6.	Dataflow Sorting in Lucid	138
7.	Input/Output in Lucid	144
8.	Debugging Lucid Programs	149
9.	Interfacing with UNIX, or Lucid Goes Cowboy	151
10.	Interactive and Real-Time Programming	154
VI	PROGRAM TRANSFORMATION	
1.	The Basis Rule	162
2.	The Substitution Rule	165
3.	The Calling Rule	170
4.	The Renaming Rules	172
5.	The Addition and Elimination Rules	173
6.	The Amalgamation Rule	175
7.	The Import and Export Rules	177
8.	The Parameter Insertion and Deletion Rules	179
9.	Symbolic Evaluation	182
10.	The Symbolic Evaluation of Iterative Programs	187
11.	Binding and Calling Conventions	191
12.	Proving Assertions about Programs	195
13.	Verification Aiding Transformation: An Example	200
VII	BEYOND LUCID	
1.	Types and Type Checking	207
2.	Tupling, or Filters with Multiple Outputs	211
3.	Arrays, or Frozen Streams	217
4.	ILucid, Lucid with Multidimensional Time	223
5.	Lambda Lucid, Lucid with Higher-Order Functions	228

	Contents	vii
REFERENCES		235
APPENDIX: THE pLUCID PROGRAMMER'S MANUAL		239
<i>A. A. Faustini, A. A. G. Yaghi and S. G. Matthews</i>		
INDEX		305