New Insights from Old Programs

The Structure of The First ALGOL 60 System

van den Hove d’Ertsenryck, G.M.C.J.T.G.
It is a commonplace that computer programming is hard, especially when one aims at creating a program that is correct. What kind of methods should be used to reach that goal? The subject of heated debates. Our thesis is a contribution to these discussions: To understand what computer programming is, and how it should be done, we propose to study how it is actually done — that is, to induce elements of method from actual observation. Our thesis takes the form of a detailed analysis, based on a careful reconstruction, of a particular well-crafted computer program: the first ALGOL 60 system, designed and implemented at the Mathematical Center (now CWI) by E. W. Dijkstra and J. A. Zonneveld, with the assistance of S. J. Christen and M. J. H. Römgens, on an Electrologica X1 computer. It is divided into three main chapters. Chapter I presents the two elements of the problem the Mathematical Center team was facing, namely the ALGOL 60 language and the X1 computer. Chapter II discusses the principles of its solution, explains the implementation choices made by the Mathematical Center team, and compares them to other possible choices. Chapter III presents the details of the Mathematical Center ALGOL 60 system, on an ISO C version of that system, reverse engineered from its X1 assembler source. This program is about 3000 lines long and is composed of 173 subroutines working on 57 global variables. Finally, our conclusion, in the form of 17 theses and 4 hypotheses, indicates some lessons, in particular on computer programming methods, that we believe can be drawn from the analysis of that particular computer program.

Gauthier van den Hove
New Insights from Old Programs — The Structure of The First ALGOL 60 System

Gauthier van den Hove

With A Foreword by Donald E. Knuth

NEW INSIGHTS FROM OLD PROGRAMS — THE STRUCTURE OF THE FIRST ALGOL 60 SYSTEM

CAUGHTER VAN DEN HOVE

Report on the Algorithmic Language ALGOL
Dedicated to the memory of William Turanski

INTRODUCTION

BACKGROUND

After the publication of a preliminary report on the algorithmic language ALGOL, as prepared at a conference in Zurich in 1958, much interest in the ALGOL language developed. As a result of an informal meeting held at Mainz in November 1958, about forty interested persons from several European countries held an ALGOL implementation conference in Copenhagen in February 1959. A "hardware group" was formed for working cooperatively right down to the level of the paper tape code. This conference also led to the publication by Regnecentralen, Copenhagen, of an ALGOL Bulletin, edited by Peter Naur, which served as a forum for further discussion. During the June 1959 ICIP Conference in Paris several meetings, both formal and informal ones, were held. These meetings revealed some misunderstandings as to the intent of the group which was primarily responsible for the formulation of the language, but at the same time made it clear that there exists a wide appreciation of the effort involved. As a result of the discussions it was decided to hold an international meeting in January 1960 for improving the ALGOL language and preparing a final report. At a European ALGOL Conference in Paris in November 1959 which was attended by about fifty people, seven European representatives were selected to attend the January 1960 Conference, and


2 Report on the Algorithmic Language ALGOL by the ACM Committee on Programming Languages and the GAMM Committee on Programming, edited by A. J. Perlis and K. Samelson, Numerische Mathematik Bd. 1, S. 41-60 (1959)

Copyright © 2017 by Gauthier van den Hove <ghe@fibonacci.org>. All rights reserved. This work may not be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of the author. The cover, inside covers, and pp. 1–2, 7–9, 13–15 and 249–302 of this work can however be freely distributed.
Algorithm index: A, 94; C1, 53; C2, 55; C4, 58; C6, 71; C7, 72; C8, 73; C9, 79; C10, 88; C11, 89; C12, 96; D1, 63; D2, 74; D3, 76; D4, 89; D7, 97; E1, 50; E2, 60; E3, 75; E4, 75; E8, 77; J1, 77; J2, 98; R1, 53; R2, 55; R3, 56; R4, 57; R5, 66; R6, 71; R7, 72; R8, 73; R9, 79; R10, 88; R11, 89; R13, 97; X, 61; Y1, 106; Y2, 111; Y3, 113; Y4, 115; Y5, 118; Y6, 124; Y, 128.

1. (empty :: 24, 40, 77, 101, 103, 105) := |
2. (basic symbol :: 24) := (letter | digit | (logical value | (deiminer)) |
3. (letter | 16, 17) := a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
4. (digit | 16, 17) := 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
5. (logical value | 2, 51) := true | false |
6. (operator | 2) := (separator | 12) | (operator | 14) | (operator | 15) |
7. (operator | 8) := (arithmetric operator) | (relational operator) | (logical operator) |
8. (arithmetric operator | 8) := + | - | * | / | ↑ |
9. (relational operator | 20) := < | ≤ | ≥ | > | ≠ |
10. (logical operator | 8) := || | V | ∧ | ¬ |
11. (sequence operator | 8) := go to |
12. (separators | 8) := ; |
13. (bracket | 13) := ( | ) |
14. (actual parameter | 8) := (variable | label) |
15. (procedure identifier | 13) := string | label | value |
17. (unsigned integer | 17, 19, 21, 58) := (digit | 17) | (unsigned integer | 17) |
18. (integer | 19) := (unsigned integer | 17) | + | - | (unsigned integer | 17) |
19. (decimal fraction | 19) := (integer | 19) |
20. (exponent part | 22) := 10 | (integer | 19) |
21. (decimal number | 22) := (integer | 19) | (decimal fraction | 19) |
22. (unsigned number | 23, 44) := (decimal number | 21) |
23. (number | 23) := (unsigned number | 22) |
24. (proper string | 25) := (any sequence of basic symbols not containing ' or ') | (empty) |
25. (open string | 26) := (proper string | 24) | (open string | 24) |
26. (string | 26) := '' |
27. (expression | 26) := (arithmetic expression | 49) | (Boolean expression | 27) | (designational expression | 26) |
28. (variable | 26) := (identifier | 16) |
29. (simple variable | 44, 46) := (variable identifier | 26) |
30. (subscript expression | 31, 80) := (arithmetic expression | 49) |
31. (subscript list | 31, 33) := (subscript expression | 30) |
32. (array identifier | 33, 36, 94) := (identifier | 16) |
33. (subscripted variable | 34) := (array identifier | 32) |
34. (variable | 44, 51, 73, 82) := (simple variable | 29) |
35. (procedure identifier | 36, 41, 44, 100) := (identifier | 16) |
36. (actual parameter | 36) := (string | 26) | (expression | 27) | (array identifier | 32) |
37. (letter string | 37, 38) := (letter | 16) | (letter string | 37) |
38. (parameter delimiter | 39, 100) := (letter string | 37) |
39. (actual parameter list | 39, 40) := (actual parameter | 36) |
40. (actual parameter part | 41, 64) := (empty | 1) |
41. (function designator | 44, 51) := (procedure identifier | 35) |
42. (adding operator | 43) := + | - |
43. (multiplying operator | 46) := * | / | |
44. (primary | 48) := (actual number | 22) |
45. (factor | 45, 46) := (primary | 44) |
46. (term | 46) := (factor | 45) |
47. (simple arithmetic expression | 47, 49) := (term | 46) |
48. (if clause | 49, 57, 62, 78) := (empty | 1) |
49. (arithmetric expression | 50, 54, 55, 57, 75, 80, 90, 91) := (simple arithmetic expression | 47) | (if clause | 46) |
50. (relational operator | 51) := (arithmetric expression | 49) |
51. (Boolean primary | 52) := (logical value | 2, 51) |
52. (Boolean secondary | 54) := (Boolean primary | 51) |
53. (Boolean factor | 54) := (Boolean secondary | 52) |
54. (Boolean term | 54, 55) := (Boolean factor | 53) |
55. (implication | 55, 56) := (Boolean term | 54) |

Copyright © 2017 by Gauthier van den Hove <ghe@fibonacci.org>. All rights reserved. This work may not be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of the author. The cover, inside covers, and pp. 1–2, 7–9, 13–15 and 249–302 of this work can however be freely distributed.
New Insights
FROM
Old Programs

The Structure
of
The First
ALGOL 60 System
Promotiecommissie:

Promotores: prof. dr. P. Klint, Universiteit van Amsterdam
prof. dr. T. van der Storm, Rijksuniversiteit Groningen

Overige leden: prof. dr. K. R. Apt, Universiteit van Amsterdam
prof. dr. J. A. Bergstra, Universiteit van Amsterdam
prof. dr. D. J. N. van Eijck, Universiteit van Amsterdam
prof. dr. C. B. Jones, University of Newcastle upon Tyne
prof. dr. F. E. J. Kruseman Aretz, Technische Universiteit Eindhoven
prof. dr. ir. C. T. A. M. de Laat, Universiteit van Amsterdam
prof. dr. B. Randell, University of Newcastle upon Tyne

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

See http://www.fibonacci.org/GHE3 for current information about this work.

This work is complemented by an electronic companion archive. It contains all the programs discussed, and a large number of additional programs collected from the literature (see Appendix G, pp. 355 sq.).

This research was funded by the Nederlandse Organisatie voor Wetenschappelijk Onderzoek Free Competition Grant number 612.001.003, and was carried out at the Centrum Wiskunde & Informatica.

Copyright © 2017 by Gauthier van den Hove <ghe@fibonacci.org>

All rights reserved. No part of this work may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of the author.

Second improved printing, 27 June 2017