

# Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

177

---

## Programming Languages and Their Definition

H. Bekič (1936–1982)

Selected Papers edited by C. B. Jones

---



Springer-Verlag  
Berlin Heidelberg New York Tokyo 1984

**Editorial Board**

D. Barstow W. Brauer P. Brinch Hansen D. Gries D. Luckham  
C. Moler A. Pnueli G. Seegmüller J. Stoer N. Wirth

**Editor**

C.B. Jones

Department of Computer Science, The University  
Manchester M13 9PL, Great Britain

CR Subject Classifications (1982): 0.3, D3.1, F3

ISBN 3-540-13378-X Springer-Verlag Berlin Heidelberg New York Tokyo  
ISBN 0-387-13378-X Springer-Verlag New York Heidelberg Berlin Tokyo

Library of Congress Cataloging in Publication Data. Bekić, H. (Hans), 1936–1982. Programming languages and their definition. (Lecture notes in computer science; 177) 1. Programming languages (Electronic computers) I. Jones, C. B. (Cliff B.), 1944–. II. Title. III. Series. QA76.7.B45 1984 001.64'24 84-14094  
ISBN 0-387-13378-X (U.S.)

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically those of translation, reprinting, re-use of illustrations, broadcasting, reproduction by photocopying machine or similar means, and storage in data banks. Under § 54 of the German Copyright Law where copies are made for other than private use, a fee is payable to "Verwertungsgesellschaft Wort", Munich.

© by Springer-Verlag Berlin Heidelberg 1984  
Printed in Germany

Printing and binding: Beltz Offsetdruck, Hemsbach/Bergstr.  
2145/3140-54321

**for:**

**Sophie, Johannes, Edith, Wolfgang,  
Hildegard, Ludwig, Rudolf and Alban**

## CONTENTS

Introduction . . . . .	vii
List of Papers and Main Scientific Talks . . . . .	ix

### Selected Papers

#### Languages

Note on a Test Example for ALGOL 60 Compilers by D.E.Knuth . . . . .	1
The Assignment to a Type Procedure Identifier in ALGOL 60 . . . . .	2
The Meaning of Names in PL/I . . . . .	4

#### Language Definition

Defining a Language in its Own Terms . . . . .	17
Note on Some Problems Concerning the PL/I Manual and Its Rewriting	26
Definable Operations in General Algebras, and the Theory of Automata and Flowcharts . . . . .	30
Formalization of Storage Properties . . . . .	56
On the Formal Definition of Programming Languages . . . . .	86
A Formal Definition of a PL/I Subset (Selected parts) . . . . .	107
Mathematical Semantics and Compiler Correctness . . . . .	156

#### Parallelism

Towards a Mathematical Theory of Processes . . . . .	168
Comments on [Opus 46] . . . . .	207
The Semantics of Parallel Processing . . . . .	215
Nondeterministic Functions and the Semantics of CSP . . . . .	230
Nondeterministic Programs: An Example . . . . .	240
A Model of Nondeterminism: Indexed Sets and Their Equivalence . . . . .	248

## INTRODUCTION

The IBM Vienna Laboratory has made a significant contribution to the work on the semantic description of computer systems. Both the operational semantics descriptions ("VDL") and the later work on denotational semantics ("Meta-IV", "VDM") contain interesting scientific ideas. Partly because of the large scale of the applications tackled, much of the material is difficult to access.

Hans Bekic was one of the key members of the "Vienna Lab". His tragic death in a mountain accident in October 1982 left unpublished an important body of research. (The editorial notes below contain further details of his scientific career; a biographical note has been written by Professors Kuich and Zemanek).

Hans' computer research can be considered under three headings. His work on programming languages took place partly within IFIP WG 2.1 of which he was a member from 1965-1971. He also had an influence on the development of PL/I. Hans was a mathematician and so moved naturally to research on formal language description. Most of his scientific career was spent on this work. Hans was largely responsible for the move by the Vienna Lab from operational to denotational semantics. He was a member of IFIP WG 2.2 from 1969 until his death. Work on the description of parallelism occupied the last years of Hans' life. This was a difficult period in which the laboratory was employed on practical programming tasks: Hans pursued his scientific work in his "spare time".

This book contains a selection of Hans Bekic's papers. Publication of his specific contribution presents special problems. Much of his work was unpublished and even, in some cases, existed only as hand written manuscripts. Sometimes Hans postponed publication because he considered a piece of work to be flawed; in other cases - it must be conceded - his habit of leaving things to the last minute resulted in his missing a deadline for publication. In spite of this, Hans' work has been widely circulated and has influenced others. His untimely death means that he cannot complete the work in the way he would have wished.

After discussions with his family, friends and colleagues, it was decided that it would be valuable to publish a selection of Hans' papers. To alleviate any fear that he would not have wished them to be published in this form, his own reservations - where known - have been included. A list of all known writings and main talks on computer topics is given below (book reviews etc. are not listed); the most important papers are reproduced.

I am one of many people who regard their contact with Hans as having been immensely stimulating and enjoyable. He was also a good friend. I hope that this volume will make some of his scientific work known to a wider community.

#### Editorial Notes

Because of Hans' interest in music, the items below have been given "Opus" numbers. They have been arranged in order of earliest reference to give coherence to the development - subsequent publication is shown as such. The list contains the title and source of every known "paper" with comments (written by CBJ) in italic fount. These comments attempt to set the background and to record any reservations which HB was known to have had about a piece of work.

Citation research used "Compumath" and "Science Citation" indices. Many of the source documents used the ISO data format (yyyy-mm-dd) - this has been preserved.

The decision was made to retype all of the papers. Apart from the effort and delay this entailed, it has clearly increased the risk of error. It is hoped that the uniformity and style of presentation justify this step.

#### Acknowledgements

Funding for travel and typing has been provided by the UK Science and Engineering Research Council (Contract SO/261/83) and IBM Austria. The typing of the editorial material was done by Julie Hibbs and of manuscripts themselves by Margaret Barringer. Library research was undertaken by Alison McCauley and translations by Prof. Prachar and Jill Jones.

The following people provided input to the editorial material: Derek Andrews, Egidio Astesiano, Jaco de Bakker, Kurt Bandat, Dines Bjorner, Andrzej Blikle, Manfred Broy, Edsger W. Dijkstra, K. Indermark, Don Knuth, W. Kuich, Peter Landin, Peter Lauer, J. Loecks, Peter Lucas, Robert Milne, Erich Neuhold, Robin Milner, John Nicholls, David Park, Brian Randell, Don Sannella, Dana Scott, Jim Thatcher, Kurt Walk, Fritz Weissenboeck, Niklaus Wirth, Heinz Zemanek.

Most of all, I should like to thank Sophie Bekic for allowing me access to Hans' papers.

Cliff B. Jones

## List of Papers and Main Scientific Talks

### [Opus 1]

Die Zahl der Symmetrieklassen der Funktionen n Logischer Variabler

H.Bekic

Internal Report, Vienna, December 1960

10 pages, 2 Refs.

### [Opus 2]

Extension of the Algorithmic Language ALGOL

V.Kudielka, P.Lucas, K.Walk, K.Bandat, H.Bekic, H.Zemanek

Mailuefterl Vienna, Final Report DA-91-591-EUC-1430, July 1961

200 pages.

*See note on this project with [Opus 10].*

### [Opus 3]

Compilation of ALGOL, Part I - Organization of the Object Program.

P.Lucas, H.Bekic

IBM Laboratory Vienna, LR 25.3.001, May 1962

48 pages, 4 Refs.

### [Opus 4]

Ueber die Ausdehnung zweier bekannter Probleme aus der additiven Zahlentheorie auf arithmetische Reihen grosser Differenz

H.Bekic

Dissertation at University of Vienna (Dept. Philosophy) 1963

67 pages, 7 Refs.

*Hans Bekic read mathematics at the University of Vienna and this was his Ph.D thesis. The title might be translated as "On the Generalisation of Two Known Problems from Additive Number Theory to Arithmetic Progressions with Large Differences".*

X

[Opus 5]

Some Aspects for a Comparison of FORTRAN and ALGOL

P.Lucas, H.Bekic

IBM Laboratory Vienna, LR 25.3.003, November 1963

23 pages, 13 Refs.

[Opus 6]

An Input-Output Proposal for ALGOL Based on FORTRAN IV Input-Output

H.Bekic

IBM Laboratory Vienna, LR 25.3.005, March 1964

12 pages, 5 Refs.

[Opus 7]

Some Comments on the ACM Committee Proposal on I/O Conventions for ALGOL 60, by

D.E.Knuth

H.Bekic

*Originally printed as a note:*

IBM Lab Vienna, LN 25.3.006, March 1964

5 pages, 3 Refs.

*Then as:*

ALGOL Bulletin No.18, October 1964

pages 16-19, 3 Refs.

[Opus 8]

Scope of Names in NPL

K.Bandat, H.Bekic

IBM Laboratory Vienna, LR 25.0.002, July 1964

19 pages, 7 Refs.

*"NPL" was the first name given to the programming language which eventually became PL/I. This paper refers to scope features which were changed even before the first compiler was built for the language.*

[Opus 9]

Block Concept for NPL

K.Bandat, H.Bekic, P.Lucas

IBM Laboratory Vienna, LR 25.0.003, July 1964



9 pages, 5 Refs.

*Here again, the discussion relates to features of the PL/I language which changed.*

[Opus 10] see page 1 below

Note on a Test Example for ALGOL 60 Compilers by D.E.Knuth

H.Bekic

*Originally printed as a note:*

IBM Laboratory Vienna, LN 25.3.009, September 1964

*Then as:*

ALGOL Bulletin No.18, October 1964

page 13, 3 Refs.

*A group led by Heinz Zemanek left the Technische Hochschule Vienna (now "Technical University") to become the IBM Science group (later "Laboratory") in 1961. This group designed and built the Mailuefterl computer. Zemanek explains the choice of name as follows:*

The name of MAILUEFTERL is derived from a joke I made when I first announced our project at the historic conference in Darmstadt in 1955. "We are not going to have a WHIRLWIND, TAIPHOOON, or HURRICANE", I said, "But we shall have a nice Viennese spring time breeze (MAILUEFTERL)", because the transistors we got free of charge were intended for hearing aids and so had a very low cutoff frequency. But my people developed circuits that finally allowed a clock frequency of 133 kHz, and that was not so low for the time.

*A project was started to build a compiler for the ALGOL 60 language and the compiler was available in 1961. The language, which had been designed by IFIP Working Group 2.1, included a block concept, recursive procedures and the possibility of procedures passed to parameters. The "ALGOL Bulletin" was the main publication for comments on the language and its implementation. Donald Knuth had earlier published the following program as a test to see whether compilers handled the new features correctly:*

```

begin real procedure A(k, x1, x2, x3, x4, x5);
    value k; integer k;
    begin real procedure B;
        begin k := k - 1;
            B := A := A(k, B, x1, x2, x3, x4)
        end;
        if k ≤ 0 then A := x4 + x5 else B
    end;
    outreal(A(10, 1, -1, -1, 1, 0))
end;

```

Hans Bekic had been working on the ALGOL compiler and this brief note shows that the Mailuefterl team had implemented recursion etc. correctly. This note is an interesting "pre-echo" of the discussion relating to PL/I (see [Opus 18]). The Mailuefterl project is described in "Central European Prehistory of Computing" by H. Zemanek, published in "A History of Computing in the Twentieth Century", (eds.) N. Metropolis, J. Howlett and Gian-Carlo Rota, Academic Press, 1980; and the ALGOL 60 compiler is discussed in [Opus 3].

[Opus 11] see pages 2 - 3 below

The Assignment to a Type Procedure Identifier in ALGOL 60

H.Bekic

Originally printed as a note:

IBM Laboratory Vienna, LN 25.3.010, September 1964

2 pages, 2 Refs.

Then as:

ALGOL Bulletin No. 18, October 1964

pages 14-15, 2 Refs.

This discussion of the implementation consists of technical details but the introduction is characteristic of Hans Bekic's style. It should be remembered that ALGOL 60 is not "fully typed": procedure formal parameters were not, in general, fully specified. The "w" used in the text was probably intended to be a Greek "ω".

[Opus 12] see pages 17 - 25 below

Defining a Language in its Own Terms

H.Bekic

IBM Laboratory Vienna, TN 25.3.016, 22nd December 1964

12 pages, 4 Refs.

*The Vienna Laboratory decided to apply the ideas on formal definition to PL/I - [Opus 27] includes the following comments:*

### History

This Technical Report, called "Formal Definition of PL/I" (Universal Language Document No.3: 'ULD 3'.), describes PL/I as gathered from the SRL - Manual Form C28-6571-3. It formally describes the programming language, but not a specific implementation in a specific environment.

The project was initiated by a formal proposal made by members of the Hursley Laboratory, of the Poughkeepsic Laboratory, and of the Vienna Laboratory in October 1965 /1/. This proposal was accepted by the PL/I manager and the Vienna Laboratory was committed subsequently to prepare the document by end of 1966 in the form of a technical report.

The early ideas of the Vienna group on the method are documented in the "Tentative Steps" /2/ and have been discussed with responsible and interested IBM advanced programming specialists on many occasions. There was a frequent exchange of working documents and there were regular working meetings between the Hursley language definition group, the Poughkeepsie language evaluation group, and the Vienna language definition group. K.BANDAT represented the Vienna Laboratory in the Language Resolution Board and served as liaison to the other IBM PL/I groups.

The method of definition and its application to PL/I have been presented at various occasions internally in IBM to groups involved in PL/I and at the IBM Programming Symposium in Skytop in November 1965 /3/, to SHARE representatives in Vienna in October 1966, and to language committees outside IBM /4/.

### Remark

This first version of the document contains essentially the formal description as such, with a minimum of reading aids. It certainly is not a teaching device or a kind of text-book. But some additional work will yield a considerable improvement in this direction.

We believe that formal definition is a necessary step in the development of the art of programming. This document on PL/I not only removes the ambiguities

that no informal description can avoid. It also establishes the way of discussion and asking questions in precise terms. It constitutes a ground for the future development of PL/I.

We hope that this document and its coming improved versions will be a good service for IBM and IBM's customers

H.Zemanek, Vienna, 30th December 1966.

#### References

- /1/ Unambiguous Definition of PL/I. - Memorandum in response to a Telex of Dr. M. de V.ROBERTS signed by K.BANDAT, E.F.CODD, R.A.LARNER, P.LUCAS, J.E.NICHOLLS, and K.WALK, dated 8th October 1965.
- /2/ K.BANDAT (ed.): Tentative Steps Towards a Formal Definition of PL/I. - IBM Laboratory Vienna, Technical Report TR 25.065, July 1965.
- /3/ P.LUCAS: On the Formalization of Syntax and Semantics of PL/I. - IBM Laboratory Vienna, Technical Report TR 25.060, November 1965.
- /4/ P.LUCAS, K.WALK: Formal Definition of PL/I. - Presented at ASA X3.42C - Meeting, New York, 28th July 1966.  
K.BANDAT: Formal Definition of PL/I. - Presented at ECMA TC10 Meeting, Madrid, 12th October 1966.

*The group acknowledged the influence of Elgot, Landin and McCarthy. The approach (cf. note on [Opus 27]) was "operational". This first paper by Hans Bekic on semantics shows his distinctive style. It is particularly interesting to note how section 4 indicates an awareness of an alternative, more denotational, approach.*

*It should be remembered that a conference had been held in Baden bei Wien in September 1964. This conference led to the formation of IFIP's working group on Formal Description (WG 2.2). The proceedings of this conference are published as "Formal Language Description Languages" (ed) T.B.Steel, North-Holland, 1966 and include key papers by, among others, Peter Landin and Christopher Strachey.*

## [Opus 13]

Comment on "Cleaning up ALGOL 60"

H.Bekic

*Originally printed as a note:*

IBM Laboratory Vienna, LN 25.3.018, January 1965

2 pages, 2 Refs.

*Then as:*

ALGOL Bulletin No.19, January 1965

page 58, 2 Refs.

## [Opus 14]

The Meaning of Identifiers in ALGOL and in "Generalized ALGOL"

H.Bekic

*Originally printed as a note:*

IBM Laboratory Vienna, LN 25.3.019, February 1965

2 pages, 1 Ref.

*Then as:*

ALGOL Bulletin No.20, July 1965,

pages 29-30, 1 Ref.

## [Opus 15]

Bibliography on Formal Language Description Languages

H. Bekic, F. Schwarzenberger

IBM Laboratory Vienna, TR 25.052, February 1965

34 pages.

## [Opus 16]

Mechanical Transformation Rules for the Reduction of ALGOL to a Primitive Language  
M and their Use in Defining the Compiler Function

H.Bekic

IBM Laboratory Vienna, TR 25.051, February 1965

77 pages, 4 Refs.

## [Opus 17]

Assignment to Formal Procedure Identifier, or How to Cite the Bible

H.Bekic

*Originally printed as a note:*

IBM Laboratory Vienna, LN 25.3.034, March 1965

3 pages

Then as:

ALGOL Bulletin No.20, July 1965

pages 30, 45-46, 4 Refs.

[Opus 18] see pages 4 - 16 below

The Meaning of Names in PL/I

H.Bekic

IBM Laboratory Vienna, LR 25.3.025, June 1965

23 pages, 6 Refs.

The formal definition of PL/I was produced as a series of Vienna Laboratory technical reports. The term used by the group was "Universal Language Document No.3" ("ULD"). (The index reflects the existence of the natural language description of PL/I and a semi-formal description developed in the IBM Hursley Laboratory). The more widely used term "Vienna Definition Language" ("VDL") was coined by J.A.N. Lee.

Based on the ALGOL experience (cf. note on [Opus 11]), Hans Bekic was quick to spot an inconsistency in the way recursion was described for the PL/I language. His point was not just that PL/I differed from ALGOL 60 but, as he saw it, the language designers were ignoring mathematical convention and properties. Hans Bekic spearheaded a crusade (cf. [Opus 19], [Opus 20], [Opus 21]) for a language change. Although the first version of the PL/I "F" compiler stuck to the old interpretation, this crusade resulted in a change to both language and compiler. To many PL/I programmers the examples which are cited might appear rather esoteric - even the implementors were reluctant to see the importance of the change and there was often reference to "Bekic recursion". (A letter from the chairman of the PL/I Language Control Board contains: "The first release of the F-level Compiler will indeed contain an implementation of a special kind of recursion. In the second release, however, recursion will be implemented as defined in Dr. Bekic's report".)

For obvious reasons, notes on language changes were treated as confidential when written. It now seems reasonable to publish this history: although of no great depth, Hans Bekic's contribution to this debate may have prevented a widely used programming language disseminating a strange form of recursion. It is also interesting to see Hans Bekic's preference for compact notation.

*(The PL/I term "AUTOMATIC" describes normal ALGOL-like stack variables; "STATIC" was PL/I's version of "own" variables and "CONTROLLED" had heap-like properties - among others. The topic of variable reference was subsequently studied in connection with compiler justification - see, for example "Proving Correctness of Implementation Techniques" by C.B.Jones and P.Lucas in "Symposium on Semantics of Algorithmic Languages", (ed) E.Engeler, Lecture Notes in Mathematics, No.188, Springer Verlag).*

[Opus 19]

Recursive Procedure Interpretation

H.Bekic

IBM Laboratory Vienna, LN 25.3.027, September 1965

8 pages.

[Opus 20]

Recursion

H.Bekic

LDV 1, IBM Laboratory Vienna, 22nd October 1965

4 pages, 4 Refs.

*The "LDV" file consisted of points concerning the PL/I language. They were essentially questions and proposals about the language which were sent to the Hursley Laboratory. This is the change proposal which led to the correction of the error discussed in the note on [Opus 18].*

[Opus 21]

Questions on BASED, POINTER CELL

H.Bekic

LDV 3, IBM Laboratory Vienna, 6th December 1965

2 pages.

[Opus 22]

Pointers, BASED reference class, area

H.Bekic

LDV 6, IBM Laboratory Vienna, 14th January 1966

5 pages.

[Opus 23]

Two Notes on Syntax

H.Bekic

LDV 9, IBM Laboratory, Vienna, 9th February 1966

5 pages, 3 Refs.

[Opus 24]

Correspondence between Concrete and Abstract Syntax

H.Bekic

LDV 12, IBM Laboratory Vienna, 21st February 1966

6 pages.

[Opus 25] see pages 26 - 29 below

Note on some Problems Concerning the PL/I Manual and its Rewriting

H.Bekic

26 May 1966

55 pages, 1 Ref.

*This note is self-explanatory. Although it is written in PL/I terms, it is of wider relevance. It is reprinted here to show Hans Bekic's desire for precision even when not being "formal".*

[Opus 26]

The Main Features of van Wijngaarden Report on ALGOL X

H.Bekic

IBM Laboratory Vienna, LN 25.3.042, November 1966

8 pages.

*IFIP's working group 2.1 had, after the publication of ALGOL 60, moved on to design a new language which eventually became ALGOL 68. Hans Bekic was a member of and contributor to WG 2.1. This report was written to convey the status of the work.*

[Opus 27]

Formal Definition of PL/I (Universal Language Document 3)

PL/I - Definition Group of the Vienna Laboratory

IBM Laboratory Vienna, TR 25.071, 30th December, 1966

4 cm!, 4 Refs.



*This is the first version of the "ULD" (cf. [Opus 32], [Opus 33]). Hans Bekic's contribution is shown as writing sections 4.8 Interpretation of Pseudo-Variables and 5.11 Built-in Functions, and as having read the whole (4cm!) document. The definition is definitely operational (for a description of styles of language definition see "Main Approaches to Formal Specifications" by Peter Lucas in "Formal Specification and Software Development" by D.Bjorner and C.B.Jones, Prentice-Hall International, 1982.)*

[Opus 28]

Ueber die Anzahl der Zahlen  $a^2+b^2$  in einer arithmetischen Reihe grosser Differenz.

H.Bekic

Journal fuer die reine und angewandte Mathematik, Berlin 1967

Band 226, pages 120-131, 5 Refs.

*Summary of one of the results from [Opus 4] - the title might be translated as: "On the Number of Integers of the Form  $a^2 + b^2$  in Arithmetic Progressions with Large Differences".*

[Opus 29]

Modelling ULD 3's Abstract Objects by Infinite Trees

H.Bekic

LDV 27, IBM Laboratory Vienna, April 1967

*This document has not been traced.*

[Opus 30]

A Notation for Instruction Definitions in ULD 3

H.Bekic

IBM Laboratory Vienna, LN 25.5.024, July 1967

17 pages, 3 Refs.

*On his working version of this report, there is an undated note which states:*

*"Once this metalanguage L has been introduced, it would seem better to map PL/I progs into expressions of that language, instead of writing an interpreter in that language".*

XX

[Opus 31]

PL/I Arithmetic

H.Bekic

Note to G.W. Bonsall of 16 April 1968

14 pages.

[Opus 32]

Abstract Syntax and Interpretation of PL/I

K.Walk, K.Alber, K.Bandat, H.Bekic, G.Chroust, V.Kudielka, P.Oliva, G.Zeisel

IBM Laboratory Vienna, ULD Version 2, TR 25.082, 28 June 1968

590 pages, 8 Refs.

*This document represents the core of the second version of the formal description of PL/I (which was printed as 6 reports). Hans Bekic's contribution is shown as writing chapter 8 "Data, Operations and Conversions" and revising chapter 11 "Built-in Functions". Partly because of his QMC visit to work with Peter Landin, Hans Bekic is not shown as a contributor to the widely known third version of the ULD: clearly his earlier input did have an influence.*

[Opus 33]

Informal Introduction to the Abstract Syntax and Interpretation of PL/I

P.Lucas, K.Alber, K.Bandat, H.Bekic, P.Oliva, K.Walk, G.Zeisel

IBM Laboratory Vienna, ULD Version 2, TR 25.083, 28 June 1968

205 pages, 8 Refs.

*This document provides a (much-needed) introduction to the remaining (3cm of) reports.*

[Opus 34]

The Description of Programming Languages

H.Bekic

Fortnightly seminars given at Queen Mary College, London

November 1968 onwards.

*No information has been traced on these talks*

[Opus 35]

Strong Logical Connectives and ALGOL 68

H.Bekic

IFIP W.G. 2.2 Bulletin No. 2, 25 August 1969.

[Opus 36] see pages 30 - 55 below

Definable Operations in General Algebras, and the Theory of Automata and Flowcharts

H.Bekic

IBM Laboratory Vienna, 8th December 1969

29 pages, 15 Refs.

*The importance of fixed points in reasoning about recursive functions and iterative programs is now recognized. It would appear that the idea had (at least) three independent discoverers: Dana Scott, David Park and Hans Bekic. The references are "A Theory of Programs: an outline of joint work by J.W. de Bakker and Dana Scott" (these notes were presented in August 1969 at the Vienna Laboratory - Hans Bekic was at this time still in the UK so no discussion took place until the WG 2.2 meeting) and "Fixpoint Induction and Proofs of Program Properties", D.M.R.Park, pp. 57-78 of "Machine Intelligence 5" (ed) B.Meltzer and D.Michie, Edinburgh Univ. Press, 1970.*

*This work was mainly done during Hans Bekic's time with Peter Landin at QMC (November 3 1968 - November 2 1969). He presented his work at the 1979 AI conference (September 14 - 20 1969). Characteristically, he was too late submitting his paper and it was not published. (There was some subsequent correspondence about publication in the journal "Artificial Intelligence" but the material did not really fit the interests of its readership.) In spite of receiving no formal publication, this is a widely cited paper (e.g. Indermark, Goguen, Harel, Lescarme, Scott and Wagner). Hans Bekic attended the fourth meeting (Colchester September 1969) of IFIP WG 2.2 as an observer and presented his material on fixed points.*

*The report which is reprinted here describes the results in terms of automata theory. This makes it less readable than it might otherwise have been, but clearly the firm mathematical base appealed to Hans Bekic. Landin's influence may have been responsible for the algebraic flavour of the paper. The case of simultaneous recursion is covered.*

*Although this paper is often referenced, it has never been formally published. Hans Bekic appears to have considered that his work was subsumed by "A Concrete*

*Approach to Abstract Recursive Definitions* by M.Wand, AI Lab, MIT, January 1972. (Jaco de Bakker and Andrzej Blikle have also pointed out that a related reference is Leszczylowski, J., "A Theorem on Resolving Equations in the Space of Languages", *Bull. Acad. Pol. Sci., Ser. Sci. Math. Astr. Phys.*, 19, pp 967-970, 1971.)

[Opus 37]

Universelle Algebra und die Theorie der Programmierung

H.Bekic

Seminar series in Vienna Laboratory, from 1970-01-26.

Only Hans Bekic's preparatory notes have been found.

[Opus 38] see pages 56 - 85 below

Formalization of Storage Properties

H.Bekic, K.Walk

Originally printed as a note:

IBM Laboratory Vienna, LR 25.5.034, July 1970,

43 pages, 7 Refs.

Then in:

Symposium on Semantics of Algorithmic Languages, Edited by E. Engeler,

Lecture Notes in Mathematics, Vol.188,

Springer-Verlag, 1971,

pages, 28-61.

The definition of PL/I presented a number of problems concerned with storage mapping. Concepts like sub-locations were necessary because of structured types (arrays etc.); the property of storage being "connected" was relevant to overlay defining; various changes occurred on assignment which could affect the available locations. A different set of features were present in ALGOL 68, flexible locations being especially relevant.

The ULD had taken an axiomatic approach. Here, a more constructive view is presented. Even so, Hans Bekic commented later "The storage model is a little more abstract, in fact better, than the one used in TR 139" (here [Opus 55]). See also [Opus 50]).

An attempt was made to persuade Hans Bekic to revise this paper for "Formal Specification and Software Development" D.Bjorner and C.B.Jones, Prentice-Hall International 1982. When he decided he could not complete the work in time, Hans

Bekic wrote (to Jones 1981-12-20):

"Let me say briefly what I was trying to do.

Starting as in LNM 188, I would introduce values and ranges, but as domains:  $V$  is a domain, each  $R$  is a subdomain (Same  $\cup$ , not necessarily same  $\cap$ ), built from elementary ranges by a finite number of  $\cap$ 's and  $\cup$ 's:

composite  $R$ :  $R = \cap \langle R_i \mid i \in I \rangle$

flexible  $R$ :  $R = \cup \langle R^{(i)} \mid i \in I \rangle$

Locations have ranges and are elem/composite/flexible accordingly. There is a bijection  $l \mapsto \langle l_i \mid i \in I, \text{ the } l_i \text{ independent} \rangle$  for composite  $l$ 's, and a mapping  $l \mapsto \langle l^{(i)} \mid i \in I \rangle$  for flexible  $l$ 's. Independence is defined axiomatically, with a view to express "having no parts in common". Given  $f_0: L_0 \rightarrow V$ ,  $f_0(l) \in \text{rg}(l)$ ,  $L_0$  independent, we can "close"  $f_0$  w.r.t.

$(l, V) \mapsto \langle (l_i, V_i) \mid i \in I \rangle$  for composite  $l$ ,

and  $(l, (i, v^{(i)})) \mapsto (l^{(i)}, v^{(i)})$  for flexible  $l$ . Again, given an independently-based, closed  $f: L \rightarrow V$  and  $g: M \rightarrow l'$  with  $M$  independent,  $M \subseteq L$ , we can define  $f + g$  in terms of a similar closure operator. (For  $g = (l, v)$ , i.e.  $M = \{l\}$ , this is assignment). Of course, these closure operations make heavy existence/uniqueness assumptions, and I have gone some way towards proving those (where LNM 188 had at most plausibility arguments) - there are still problems of notation/presentation.

Now the problems. first, locations "of length zero" (e.g.  $A \text{ CHRAR}(C)$ , or  $\text{SUBSTR}(B, \text{length}(B) + 1, 0)$  in FL/I): completely ignored in LNM 188, ignored or wrong in TR 139. Second, and more important: the model is quite implicit, to certify that it is sound and "works", I would have to apply it to define - at least the storage related part of - a nontrivial language - which I cannot and will not do in the near future. Sure, the book will define Pascal; but only little of the storage model will be used, and in different notation, presumably different presentation of locations, etc. Thirdly, when I originally "invented" the model, its scope as envisaged then was quite small: I wanted to give a simple and coherent presentation of some apparently complicated features of Algol 68. If, more than ten years later, no sufficiently different or new application of the model have been worked out, rewriting of LNM 188 can hardly be justified, even if more precise etc. Now I think there ought to be new applications: data base, ADA's variant parts. Unfortunately, I don't have time to work out those applications now, quickly - so maybe all the book should do

is to refer in an appropriate place, like Foundations or your Modelling Concepts of PLS, to the existing ideas in LNM 188 and TR 139; someone starting to formulate a (next) language can then apply them, modifying/extending them as needed."

[Opus 39]

States Belong to Names, not to Multiple Values

H.Bekic

Note of 29 July 1970

2 pages

*A brief note on ALGOL 68.*

[Opus 40]

ALGOL Status Report

H.Bekic, V.Kudielka

IBM Laboratory Vienna, LR 25.2.021, October 1970

8 pages, 2 Refs.

[Opus 41]

Assemblersprachen

H.Bekic

Vorlesungen TH Wien, 1970-71.

*In spite of the modest theme of this course ("Assembler Languages"), Hans Bekic managed to bring in ideas on program proofs.*

[Opus 42] *see pages 86 - 106 below*

On the Formal Definition of Programming Languages

H.Bekic

*Originally printed as a note:*

IBM Laboratory Vienna, November 1970

30 pages, 20 Refs.

Then in:

Proceedings of the International Computing Symposium 21-22 May 1970 of the German ACM Chapter in Bonn (Ed. Wolf. D. Itzfeldt),  
pp 297-315 Gesellschaft fuer Mathematik und Datenverarbeitung, Birlinghoven,

November 1973.

In late 1972 the Vienna group was charged with a project which concerned the implementation of PL/I. This was seen (by some) as an opportunity to show that a formal definition of a language could be used as a basis for the design of a compiler. The question was which definition? Attempts in the late 1960's to use ULD definitions had shown that many of the operational features complicated proofs in a gratuitous way. The joint ECMA/ANSI committee were working on a new standard document. This had a formally described state but the state transitions were described in natural language. But, worst of all, the basis was again operational and in many respects the state was unnecessarily baroque (cf. treatment of END - the final standard document is "Programming Language PL/I", American National Standard ANSI X3.53-1976). Somewhat to the dismay of management, the decision was made to begin the compiler project by writing a new definition! The aim was to produce a denotational definition of PL/I (see [Opus 55]).

Section 4 of this paper can be seen as the logical continuation of [Opus 12]. Hans Bekic rather modestly writes:

If you have compared Vienna I (ULD) to Vienna II (TR 25.139) you will have noticed the shift from operational to denotational, and I feel somewhat responsible for that shift.

He was certainly most insistent that we stay within the denotational style.

[Opus 43]

Rapport d'Evaluation ALGOL 68

J.C.Boussard, J.J.Duby (Eds.), J.Andre, H.Bekic, M.Berthaud, S.Brehinier, M.Griffiths, Ph.Jorrand, C.H.A.Koster, M.Nicolas, J.C.Paillard, C.Pair, D.Peccoud, M.Sintzoff, P.Wodon

Revue Francaise d'Informatique et de Recherche operationnelle

RIRO 5e annee, B-1, 1971

pp 15-106, 124 Refs.

[Opus 44]

An Introduction to ALGOL 68

H.Bekic

Originally printed as a note:

IBM Laboratory Vienna, TR 25.118, February 1971

51 pages, 1 Ref

Then as:

Annual Review in Automatic Programming, Vol.7 part 3, 1973

pages 143-169, 5 Refs.

*Hans Bekic wrote in a letter (to Jim Thatcher):*

[this] was written because I liked the language but not its official description; it [Hans Bekic's report] emphasizes (informally) the denotational point of view.

*This is an interesting description of ALGOL 68 but, since it was published, is not reproduced here. A number of changes were made between the technical report and final versions. These improvements were largely a response to a review by Fraser Duncan.*

[Opus 45]

A Simple Algorithm for Partitions of Natural Numbers with Summands Bounded Above

H.Bekic

IBM Laboratory Vienna, LN 25.6.025, October 1971,

4 pages, 1 Ref.

[Opus 46] see pages 168 - 206 below (see also pages 207 - 214 below)

Towards a Mathematical Theory of Processes

H.Bekic

IBM Laboratory Vienna, TR 25.125, December 1971,

55 pages, 17 Refs.

*In 1970 the Vienna laboratory as a whole became involved in projects relating to parallelism. Hans Bekic's work on parallelism grew naturally out of his earlier work. PL/I had parallel features which gave rise to the control trees of the ULD. The desire to move to a denotational semantics created a serious difficulty: it was not clear how the denotational method could be applied to parallelism. Certainly, the choice of denotations as functions could not be maintained. This problem occupied the scientific part of the remaining years of Hans Bekic's life.*

*Although this report is widely cited, it would not have been correct to publish the report without Hans Bekic's own comments. Basically, after writing the technical report, he found an error ("the element-wise ordering of the powerset is not an*



ordering".) However, the paper contains one of the early attempts at solving the problem of providing a denotational semantics for parallelism; it is widely referenced; and it sets the stage for Hans Bekic's later work on the topic (cf. [Opus 60], [Opus 61] [Opus 62]). Manfred Broy points out that other people are still "discovering" this approach and not spotting the error! Robin Milner describes this work as a "pre-echo" of his own.

Also of interest is the discussion in section 6.1 of a proposed change to ALGOL 68. The text here is typed from a copy of the report on which Hans Bekic had marked a number of minor corrections. Some of his own comments on this paper have been traced and are printed after the text of the report. The paper was submitted to and accepted by IBM's "Journal of Research and Development". Two of the comments come from correspondence with the editor of that Journal. Hans Bekic eventually withdrew the paper.

[Opus 47]

Formal Semantics of Programming Languages: Theory and Applications

H.Bekic

Lecture given in Amsterdam

June 1972

Hans Bekic appears to have lectured from [Opus 46]. He afterwards wrote a note which summarized his general approach and commented on errors in his Actions paper - the latter material is reproduced below on pages 207 - 210.

[Opus 48], [Opus 49]

These two internal reports relate to a project which IBM would prefer to remain confidential. Although Hans Bekic was occupied on this work for some time, these (multi-authored) reports contain no essential scientific material.

[Opus 50]

Storage-related Concepts in PL/I

H.Bekic

Notes for the PL/I Seminar 3-5 January 1973 (for Workbook Chapter 3.1.3).

This was a (manuscript) working document which eventually led to the storage part of [Opus 55].

[Opus 51]

Axioms for 'for'

H.Bekic

Note of 15 March 1973

2 pages.

Tony Hoare's "An Axiomatic Basis for Computer Programming" had been published in 1969. It clearly evoked strong interest in the Vienna group and, after some seminars, Hans Bekic wrote this brief note.

[Opus 52]

Impl. Beispiel

H.Bekic

(undated) probably 1973/4

Manuscript 5 pages

When the PL/I implementation was being considered in later 1972, it was decided to sketch a small example of specification and design justification. This was to be the basis of many notational experiments. Because of the first schedule, it became known as the "Weihnachts Beispiel". This manuscript presents one of many attempts at the problem - in essence, it is contained in [Opus 58].

[Opus 53] - see pages 215 - 229 below

The Semantics of Parallel Processing

H.Bekic

In: Formal Aspects of Computing Science, Proceedings of the Joint IBM University of Newcastle upon Tyne Seminar 3-6 September 1974 (B.Shaw, Ed.) University of Newcastle upon Tyne Computing Laboratory 1975  
pp 105-123.

This is the transcript of a series of lectures given by Hans Bekic at the Newcastle seminar on "Formal Aspects of Computing Science". He comments in a letter (Milne 1975-12-17) that the transcripts are "just terrible" but takes the blame on himself ("missed my chance to correct ...") - it must also be recognized that the creation of such a transcript is extremely difficult.

The notes are published here since they show one possible way of correcting the error in [Opus 46]: the use of an "infinite tape of choice values" (oracle) is considered.

Edsger Dijkstra's reaction to the talks by Hans Bekic and Dana Scott at this conference sparked off a written discussion on the use of the denotational semantics. Considerably more heat than light was generated and no scientific purpose would be furthered by its publication.

[Opus 54]

Freitag der 13

H.Bekic

Friday, 13 September 1974

3 pages.

This very brief note (in German) is a humorous comment on the choice of Friday 13th September 1974 as the date for a Laboratory social gathering. Hans Bekic credited van Wijngaarden with the observation that the 13th falls (very slightly) more often on a Friday than any other day of the week.

[Opus 55] see pages 107 - 155 below

A Formal Definition of a PL/I Subset

H.Bekic, D.Bjorner, W.Henhapl, C.B.Jones, P.Lucas

IBM Laboratory Vienna, TR 25.139 (Part I and II), December 1974

201 pages.

The implementation project mentioned in the comments on [Opus 42] was terminated when the target machine was cancelled. It was, however, decided to salvage some of the scientifically interesting material. One part of this was the denotational description of PL/I. These reports contain a description of PL/I as in the ECMA/ANSI standard except for the Input/Output part (this was written by Walter Pachl but it was not included since the notation was not up-to-date). The undertaking was made simpler by the decision of the standardization committee to remove the (anachronistic) Tasking features of PL/I.

It would probably not be worth publishing the whole of this definition even though it is far shorter than the ULD definitions. Nor is the portion published exactly Hans Bekic's contribution. The complete index is given along with the whole of the Abstract Syntax and States part of the report. In an attempt to make these intelligible, appropriate parts of the commentary and notation are included. The discussion of exits (N4.3) and non-determinism (N4.4) are of particular interest. (Deletions are marked - Ed.)

XXX

It was felt that the use of a name for the meta-language might put a mistaken focus on irrelevant concrete syntax questions (cf. "Introduction"). The overall approach of using a formal description as the basis of a step-wise, justified, design was christened the "Vienna Development Method" - the abbreviation "VDM" was confusingly close to "VDL". The first book ("The Vienna Development Method: The Metalanguage" (eds.) D.Bjorner and C.B.Jones, Lecture Notes in Computer Science No.61, Springer-Verlag) was sometimes referred to as though it were still VDL based. This completely lost the point of the change from operational to denotational semantics. Eventually the name "Meta-IV" (!) was used in an attempt to distinguish the new metalanguage from that of the ULD.

One characteristic of the VDM metalanguage is its much more frequent use of "combinators" than the Oxford style denotational semantics. Peter Mosses acknowledges that this is a step towards his "abstract semantic algebras".

Hans Bekic commented in 1975:

TR 139 I don't send because Dines Bjorner tells me you have (and even read) it. One of our (at least my) objectives in working on it was to describe "transformations" (i.e. functions from states to states or to states x values), which are the "denotations" involved, in a "variable-free" way, i.e. without using a variable for the state, hence the "combinators" like the familiar ";" and the less familiar "let x:e;g(x)". That the family of int/eval definitions amounts to a (parametric) homomorphism is slightly blurred by some recurring uses of int-xxx with identical arguments, but N.5 points out how this can be remedied in a purely formal way. More serious is our cavalier treatment (or rather non-treatment) of  $\perp$ ,  $\subseteq$ ,  $\lim$ , but we hope to fill the gap soon. (For the most part, this only requires adaptation of existing work, the one really new point is non-determinism which TR 125 attempted to solve unsuccessfully.)

[Opus 56]

Formale Semantik

H.Bekic

Manuscript

Linz, 26 May 1975.

These notes (in German) are of a seminar given in Linz.

[Opus 57] *see pages 156 - 167 below*

# Mathematical Semantics and Compiler Correctness

H.Bekic

Presented at the Meeting of IFIP WG 2.1 in Munich, 25 August 1975

12 foils.

*The set of foils from which this contribution was typed was probably also used at the Pont-a-Mousson talk which Hans Bekic gave to WG 2.2 in September 1975. They provide a brief overview of Hans Bekic's view of compiler correctness proofs at that date.*

[Opus 58]

# Some Experiments with Using a Formal Language Definition in Compiler Development

H.Bekic, H.Izbicki, C.B.Jones, F.Weissenboeck

IBM Laboratory Vienna, LN 25.3.107, December 1975

54 pages, 7 Refs

*See comment on [Opus 52].*

[Opus 59]

# Mathematische Semantik von Programmiersprachen

H.Bekic

Vienna 1975/76

Manuscript.

*In the academic year 1975/6, Hans Bekic again taught a series of lectures at the Technical University. These notes, in German, were given to the students - broadly, they follow "Denotational Semantics: The Scott-Strachey Approach to Programming Language Theory" J. E. Stoy, MIT Press, 1977.*

[Opus 60] *see pages 230 - 239 below*

# Nondeterministic Functions and the Semantics of CSP

H.Bekic

Lecture at the Second Workshop on Semantics of Programming Languages, Bad Honnef, 17 March, 1981

Foils

Abstract in: Bull EATCS 14 (1981)

This talk (the foils of which are reproduced below) appears to be the first time that Hans Bekic spoke publicly about the use of indexed sets to model non-determinism. The introductory foil lists some problems with Power domains and gives an example which relies on unbounded non-determinacy. After developing indexed sets, the idea is applied to some problems of modelling Hoare's CSP language.

In order to understand the basic idea it is probably easier to study the note [Opus 61] below. The foils from Bad Honnef are reproduced here because of the application to CSP.

In notes made in May 1982, Hans Bekic observed that the CPO should be a domain (with an appropriate limit operation). This change is reflected in [Opus 61]. The notes also suggest the use of "diagrams" and this is pursued in [Opus 62].

[Opus 61] see pages 240 - 247 below

Nondeterministic Programs: An Example

H.Bekic

Presented in Garmisch-Partenkirchen, March 1982,

Manuscript.

This note contains the most lengthy explanation of the indexed set idea. The version here has been typed from a manuscript corrected in May 1982. The example (provided by E. Astesiano) is one which involves the fairness issue.

[Opus 62] see pages 248 - 254 below

A Model of nondeterminism: Indexed Sets and their Equivalence

H.Bekic

talk to IFIP WG 2.2, June 5-8 1982 Germany

7 foils.

The move to a category theory framework is clear from a number of manuscript notes. These working notes, however, are difficult to follow and, since they predate these foils, no attempt is made to reproduce the less organized material.

These foils were written just four months before Hans Bekic's death. As such, they must remain his last words on the semantics of parallelism.