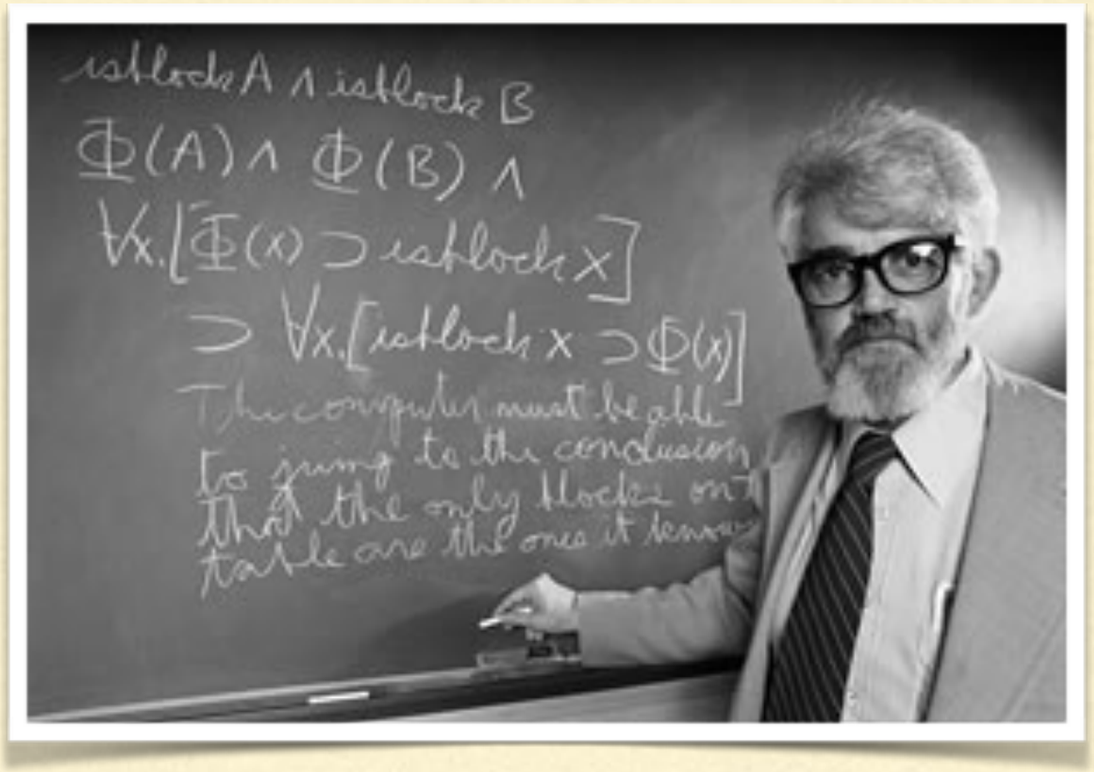
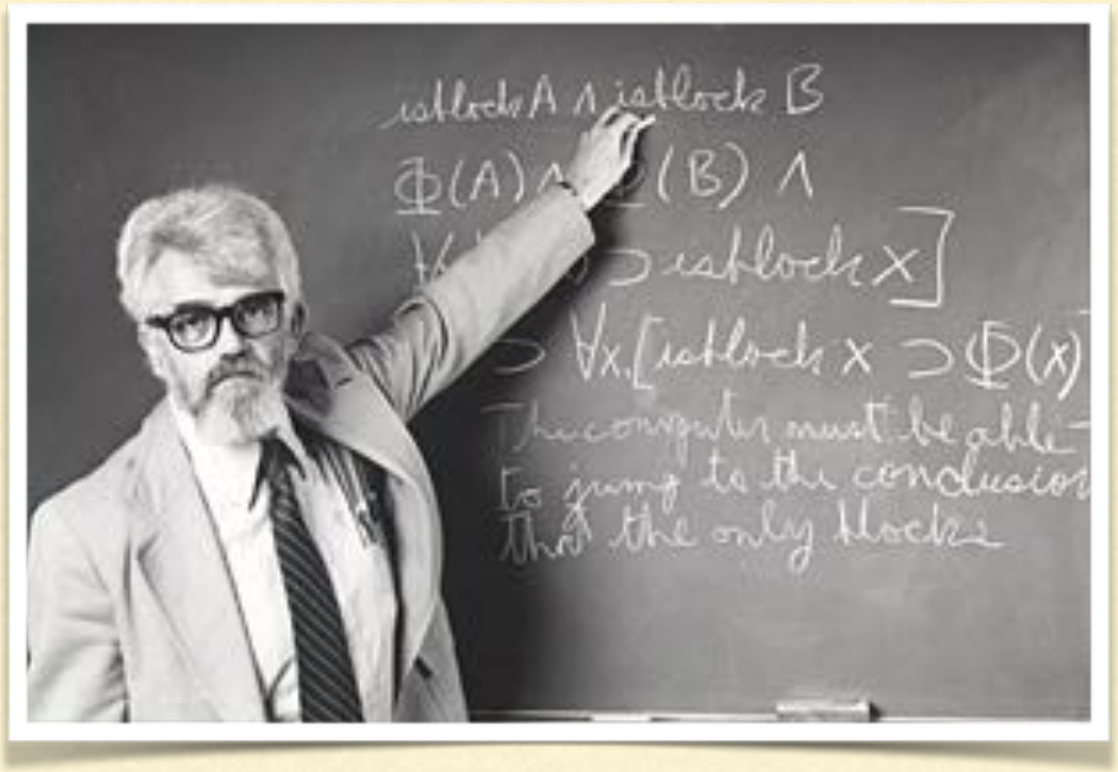




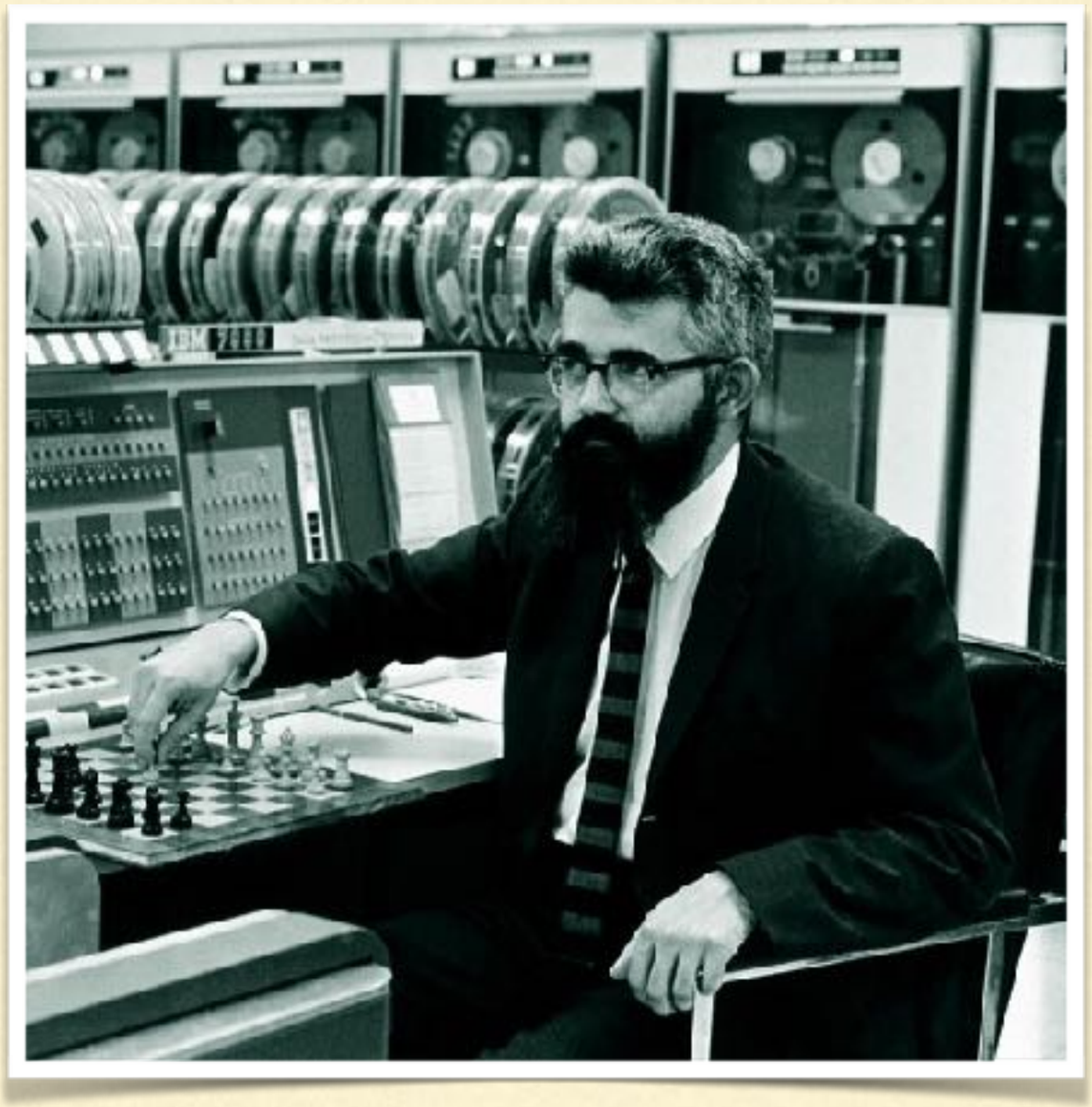
ORIGINS AND IMPACT OF FORMAL SEMANTICS

Troy Kaighin Astarte t.astarte@ncl.ac.uk

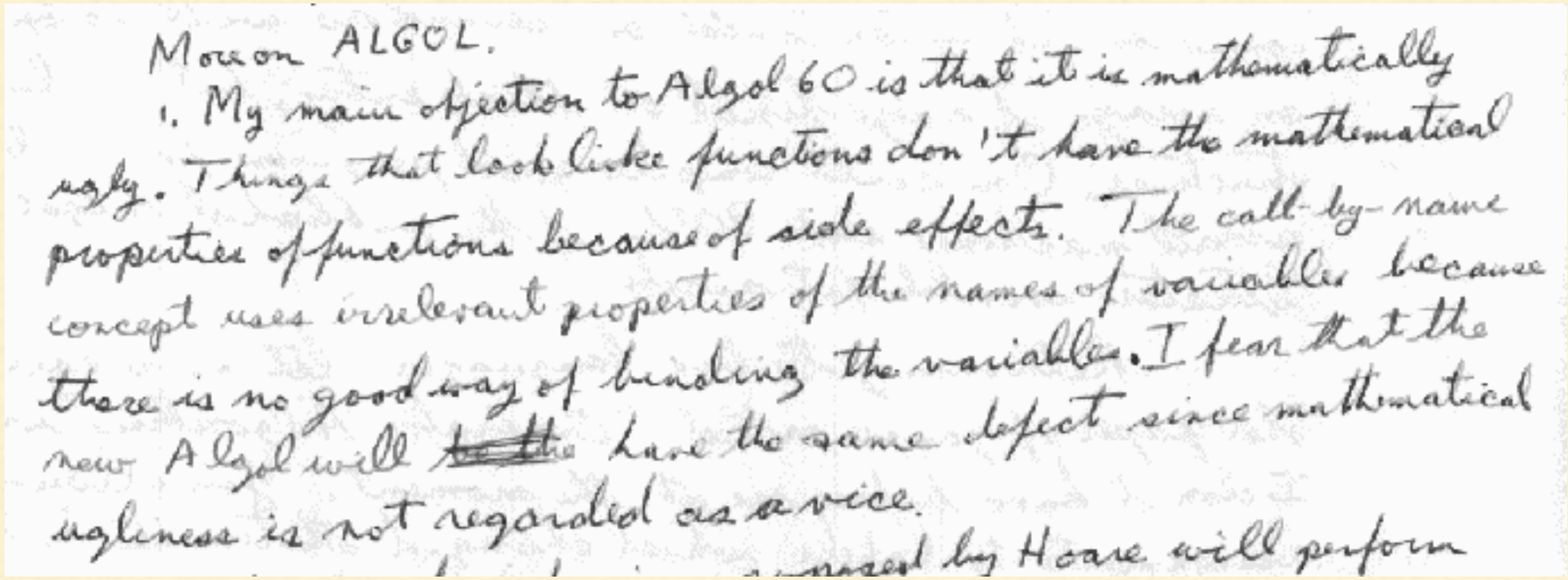
INTRODUCTION

- Programming in the1950s: machine code systems
- Computation in terms of machine—meaning fairly clear
- FORTRAN: FORmula TRANslation system for IBM 704
- ALGOL: ALGOrithmic Language (Nofre et al. 2014)
- What did this "paradigm shift" mean for computing? (Priestley 2011)
- Formalisation of programming languages

JOHN McCARTHY (1927–2011)



- Caltech mathematician; worked at Dartmouth, MIT, Stanford
- Created LISP at end of 1950s (McCarthy 1960)
- Mathematical presentation; abstraction key
- Recursive function theory basing
- *eval* function is important



Moreon ALGOL.

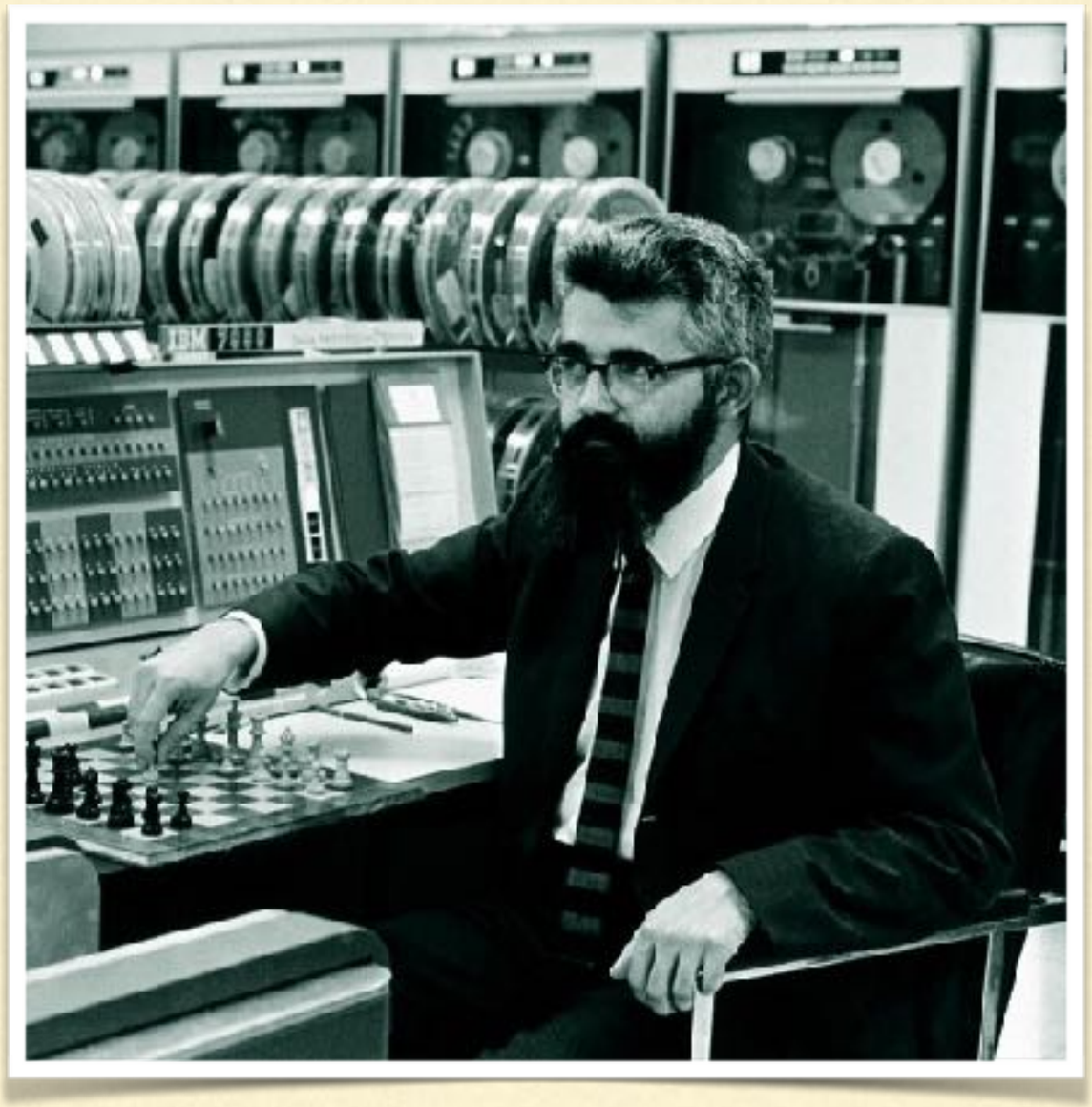
1. My main objection to Algol 60 is that it is mathematically ugly. Things that look like functions don't have the mathematical properties of functions because of side effects. The call-by-name concept uses irrelevant properties of the names of variables because there is no good way of binding the variables. I fear that the new Algol will ~~be the~~ have the same defect since mathematical ugliness is not regarded as a vice.

ugliness is not regarded as a vice.

proposed by Hoare will perform

(McCarthy 1965)

JOHN McCARTHY (1927–2011)



- Caltech mathematician; worked at Dartmouth, MIT, Stanford
- Created LISP at end of 1950s (McCarthy 1960)
- Mathematical presentation; abstraction key
- *eval* function is important
- Recursive function theory basing

McCARTHY'S AIMS

- Four main stated goals:
  - Mathematise theory of computation
  - Situate within theory of recursive functions
  - Allow proving of program correctness
  - Correctness of compilers

FORMALISING LANGUAGES

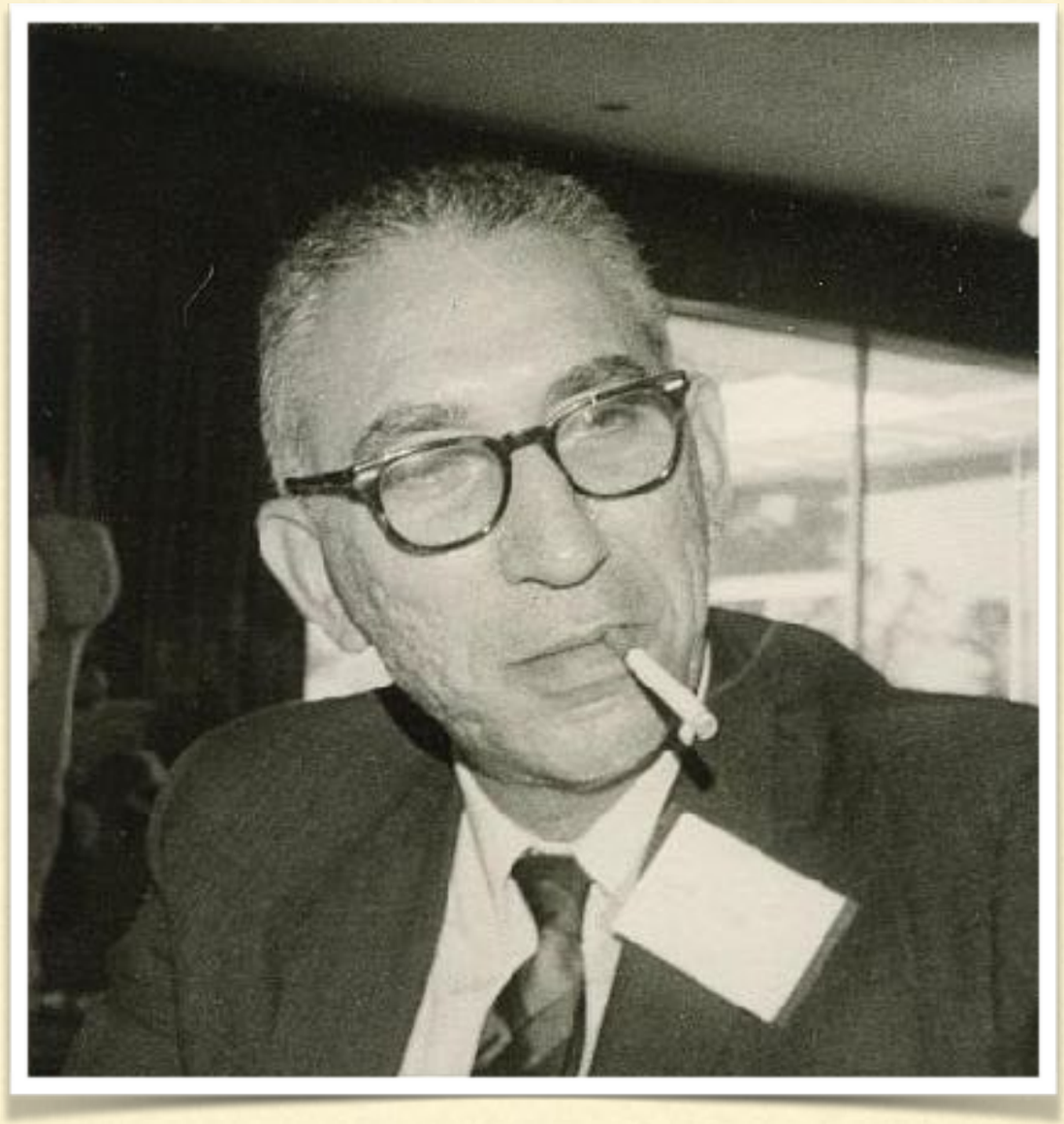
- Formalising syntax and semantics of programming languages
- Abstract ideas: structure of syntax most important
- State of computation represented simply
- Meaning of a program as a state transition
- Applied to ‘Micro-ALGOL’
- Compiler correctness: equivalence of language translation

CONTRAST: VAN WIJNGAARDEN

- ‘Founding father’ of computing in the Netherlands (Alberts 2016)
- Computation as purely symbol manipulation
- Simplify programming languages by text manipulation
- Process programs textually to create ‘list of truths’

CONTRAST: GORN

- Pushed language metaphor and was interested in standards
- Wrote on syntax definition methods but little on semantics
- Meaning in terms of a ‘background machine’ (Gorn 1963)
- Concrete and machine-focused … but idealised



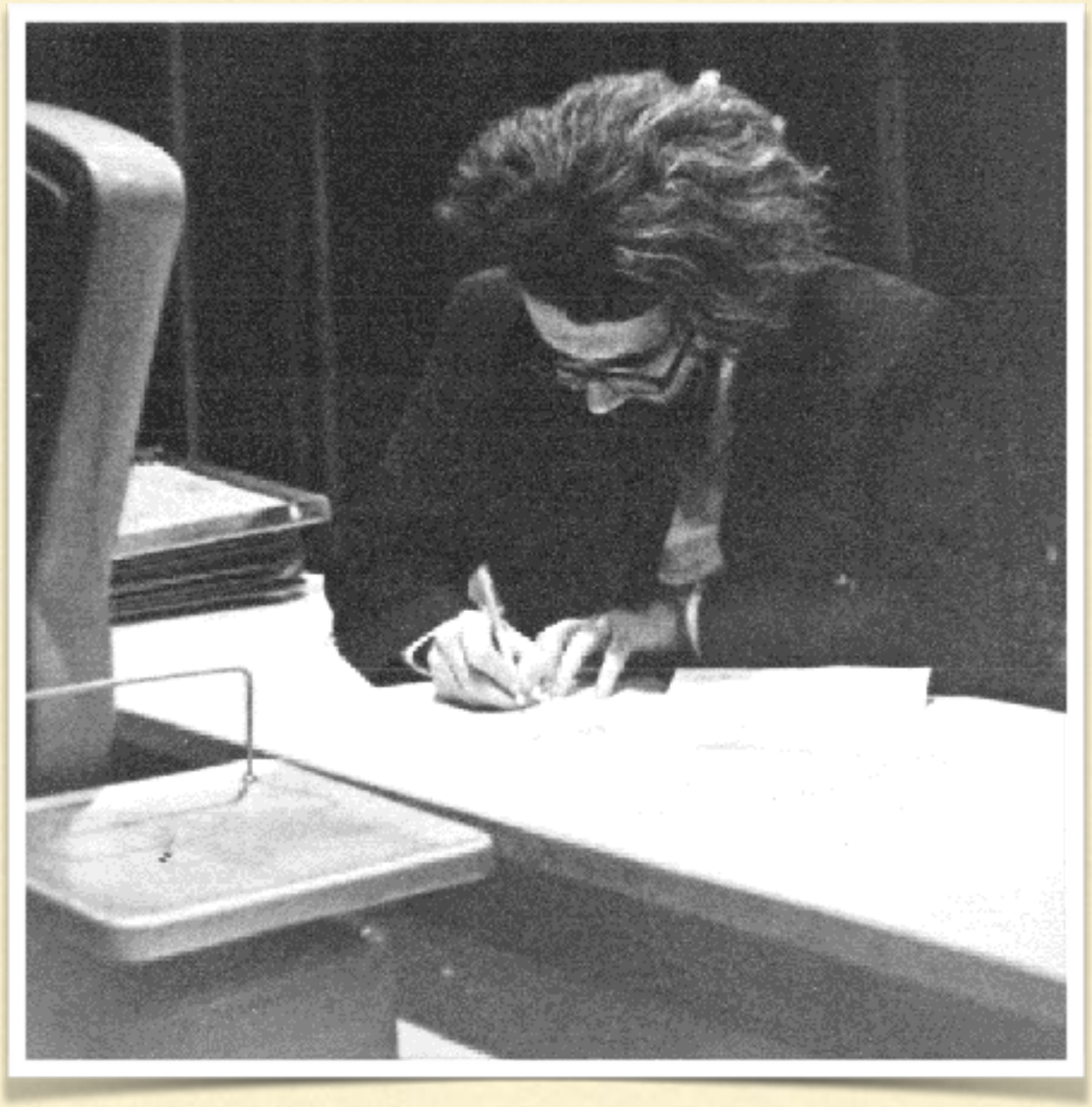
RECEPTION TO McCARTHY

- Abstraction caused confusion at first
- Micro-ALGOL too small to show real intricacies
- Praise for the descriptive approach, but not the focus on states
- … but also praise for the state!
- Does translating to another language fix the semantic problem?

McCARTHY'S INFLUENCE

- Peter Landin
  - Abstraction, use of functional notation
- Vienna Group
  - Abstract syntax, state, state transformation, compiler correctness
- Christopher Strachey
  - Abstraction, functions from state to state

DEPARTURE



- McCarthy published no further on the subject after 1967
- Partially a result of poor responses
- Began to get more interested in Manna's assertions

EVALUATION OF McCARTHY

- Mathematise theory of computation
- Situate within theory of recursive functions
- Allow proving of program correctness
- Correctness of compilers

EVALUATION OF McCARTHY

- Mathematise theory of computation ✔
- Situate within theory of recursive functions
- Allow proving of program correctness
- Correctness of compilers

EVALUATION OF McCARTHY

- Mathematise theory of computation ✔
- Situate within theory of recursive functions ✘
- Allow proving of program correctness
- Correctness of compilers

EVALUATION OF McCARTHY

- Mathematise theory of computation ✔
- Situate within theory of recursive functions ✘
- Allow proving of program correctness ✔
- Correctness of compilers

EVALUATION OF McCARTHY

- Mathematise theory of computation ✔
- Situate within theory of recursive functions ✘
- Allow proving of program correctness ✔
- Correctness of compilers ✔

- David Nofre, Mark Priestley, and Gerard Alberts. When technology became language: The origins of the linguistic conception of computer programming, 1950–1960. *Technology and Culture*, 55(1):40–75, January 2014.

- Mark Priestley. *A Science of Operations: Machines, Logic and the Invention of Programming*. Springer Science & Business Media, 2011.

- Alan J. Perlis. The American side of the development of ALGOL. In Richard L. Wexelblat, editor, *History of programming languages*, chapter 3, pages 75–91. Academic Press, 1981.

- Andrew Myers. Stanford's John McCarthy, seminal figure of artificial intelligence, dies at 84. *Stanford Report*, online, October 2011. URL https://news.stanford.edu/news/2011/october/john-mccarthy-obit-102511.html.

- John McCarthy. Letter to A. P. Ershov. Held in the Ershov archive (online), December 1965. URL http://ershov.iis.nsk.su/en/node/777964.

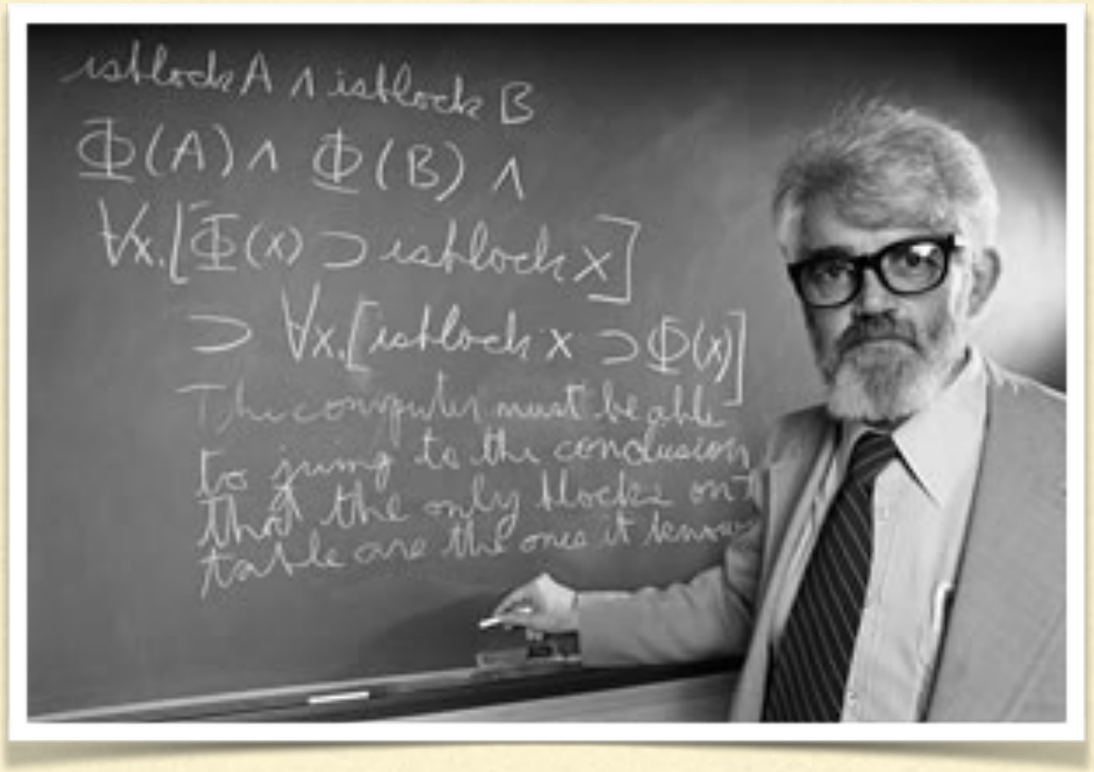
- John McCarthy. Recursive functions of symbolic expressions and their computation by machine, Part I. *Communications of the ACM*, 3(4):184–195, 1960.

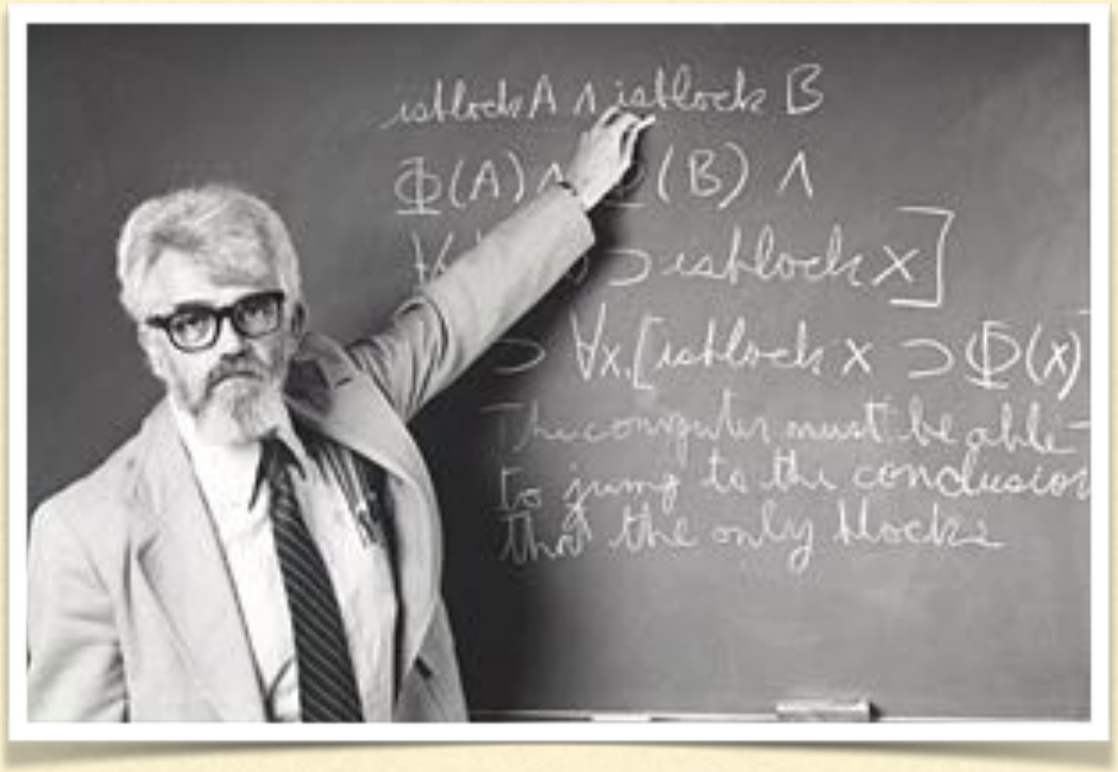
- John McCarthy. History of LISP. In Richard L. Wexelblat, editor, *History of programming languages*, pages 173–197. Academic Press, 1981.

- John McCarthy. A basis for a mathematical theory of computation. Presented at the May 9-11, 1961, Western Joint IRE-AIEE-ACM Computer Conference, pages 225–238. ACM, 1961.

- John McCarthy. Towards a mathematical science of computation. In *IFIP Congress 62*, pages 21–28, 1963.

- John McCarthy. A formal description of a subset of ALGOL. In *Formal Language Description Languages for Computer Programming*, pages 1–12, 1966.
- Adriaan van Wijngaarden. Generalized ALGOL. In *Symbolic Languages in Data Processing, Proc. Symp. Intl, Computation Center Rome*, pages 409–419, 1962.
- Adriaan van Wijngaarden. Recursive definition of syntax and semantics. In *Formal Language Description Languages for Computer Programming*, pages 13–24, 1966.
- Gerard Alberts. International informatics - Aad van Wijngaarden's 100th birthday. *ERCIM News*, 2016 (106):6–7, July 2016.
- Saul Gorn. Summary remarks (to a Working Conference on Mechanical Language Structures). *Communications of the ACM*, 7(2):133–136, February 1964.
- John McCarthy and James A. Painter. Correctness of a compiler for arithmetic expressions. *Mathematical Aspects of Computer Science*, 1. 1967
- James A. Painter. *Semantic Correctness of a Compiler for an ALGOL-like Language*. PhD. Thesis, Stanford University. 1967.
- Kurt Walk. Minutes of the 4th meeting of IFIP WG 2.2 on Formal Language Description Languages. September 1969.
- Peter J. Landin. Reminiscences. In *Program Verification and Semantics: The Early Work*, June 2001. A seminar held at the Science Museum, London.





ORIGINS AND IMPACT OF FORMAL SEMANTICS

Troy Kaighin Astarte t.astarte@ncl.ac.uk