ORIGINS AND IMPACT OF FORMAL SEMANTICS

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

- Programming in the 1950s: machine code systems
- Computation in terms of machine—meaning fairly clear
- FORTRAN: FORmula TRANslation system for IBM 704
- ALGOL: ALGOOrithmic 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
More on 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 have the same defect, since mathematical ugliness is not regarded as a vice.

(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
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 ✔


- Kurt Walk. Minutes of the 4th meeting of IFIP WG 2.2 on Formal Language Description Languages. September 1969.

ORIGINS AND IMPACT OF FORMAL SEMANTICS

Troy Kaighin Astarte

t.astarte@ncl.ac.uk