this is a summation of key findings from my PhD research on model-based semantics (ie no axiomatic)
title is a quotation from CARH (will come back to it)

1972 Bauer at tenth anniversary IFIP talk on history of computation went back to
coins and pebbles and came up to date with full formal description of ALGOL 68
examples existed of both totally complete, fully formalised descriptions of working
PLs; other methods had deep connections to logic and mathematics
A complete formal description of syntax and semantics was even taken for granted
when in 1965 WG2.1 were working on successor to A60
now very few languages
Why a hope? Why didn’t it work out?
Programming was/is hard!

- Errors in programs, worse in compilers
- Intuitive understanding OK but serious worries about correctness (cf Software Crisis)
- Core aspect of (imperative PLs): variables and values using a state
- but increasing challenges:
  - sharing; procedures; jumps; concurrency (!)

Motivations

- Theory
  - formalising foundations of computing: develop a theory
  - combat “vague feeling of unease”

- Practice
  - correctness of compilers
  - designing programming languages
  - standardisation

The division is rather artificial: many people stated motivations for both “This [developing a CPL compiler] gives rise to a rather vague feeling of unease, and though we think we know what we mean about such things as lists, error exits from functions, and input-output, we are not altogether happy that we have really got to the bottom of the concepts involved.”
Motivations

- Theory
  - formalising foundations of computing: develop a theory
  - combat "vague feeling of unease"
- Practice
  - correctness of compilers
  - designing programming languages
  - standardisation

McC also interested in compilers, a more practical problem

And of course Jones very keen on theoretical computer science JES has spoken/written about the contributions of the PRG being
Different approaches

- Fundamental similarities (see [JA18])
- But notational differences made serious impact on usability
- Often result of different backgrounds
- But most came to semantics from language design

mathematicians vs. engineers

McC: Lisp; VAB: A60 & PL/I; Strachey: CPL; and these have big impact

this is in response to criticism that his axiomatic semantics was addressing only simple language constructs

S went on to say that assignment is hard in LC and recursion in others yet both are used easily enough in most PLs.
Organisations

- Academic: MC, PRG, Stanford...
- “highly critical and thoughtful atmosphere in which ad hoc or superficial ideas are given very short shrift”—Strachey
- Commercial: IBM
- Need for a product always a constraint
- Umbrellas: ACM, IFIP

Collaborations

- Landin/Strachey; Scott/Strachey; PRG students
- Edinburgh hub around Milner/Burstall
- VAB a group: Bekić, Jones also travelled
- one early influence a visit from Scott in 1969—traces a line back to van Wijngaarden!
- IFIP WG 2.2 a counter example

Re needing a product, I’ll show a picture of the ULD description IBM’s Language Control management were horrified by it!

Umbrellas and ways to control collab leads to next slide

Ed included Plotkin, Burstall, Moore, Gordon

DSS at VAB in Summer 1969, just after his sabbatical at MC, where he was working with dB

dB had just been breaking away from vW’s style
The semantics problem

- Does a new language give meaning?
- "Because it takes pages and pages of gobbledygook to describe how a programming language works, it's hard to prove that a given program actually does what it is supposed to. Therefore, programmers must learn not only this enormously complicated language but, to prove their programs will work, they must also learn a highly technical logical system in which to reason about them."


McCarthy: "nothing can be explained to a stone"

John McCarthy. A formal description of a subset of ALGOL 60. In Formal Language Description Languages. 1966

McCarthy’s point was that you have to have some base shared knowledge and people levelled all these accusations about Tarski’s semantics of maths logic which has been successful also, that the formalism is somehow simpler, with fewer base concepts

Too complex!

UDL-III version III

IBM Vienna's full formal definition of PL/I

and it’s printed on cigarette paper

BUT PL/I itself was a tremendously bloated language
On ALGOL 68

TURSKI: In Grenoble we decided that the proposed description method is a milestone in the development of the language.

RANDELL: A milestone or a millstone?

General laughter follows.


Or not expressive enough?

Caracciolo: A reduction to simpler questions would mean to omit the proper problem.

Scott: Only the most primitive, non-problematic things have been dealt with using this approach.

Laski: A language definition should specify as little as possible.


this tension played out a lot in WG 2.2

but perhaps the elegance–expressiveness dichotomy is found everywhere in maths, and maybe even in science generally?
Impactful elsewhere

- Defining the whole of a PL was a huge task
- So: separate problem and address instead:
  - program correctness
  - or concurrency
  - or type theory
  - or build semantics into PL (functional)
- … all influenced by classic formal semantics

as pointed out by Mahoney, formal semantics seen as a crucial part of early “formal”/“theoretical” in the European tradition
many of the big names in European CS worked on semantics: EWD, vW, Cara, Bekič, McC, CSS/DSS, JCR, PJL, RMB, Plotkin, Milner …
Would love to do some follow up work on situating within a broader history of knowledge about computing and science

Please read my thesis :-)