The invention of a programming language

Cultures of Programming — Part 3

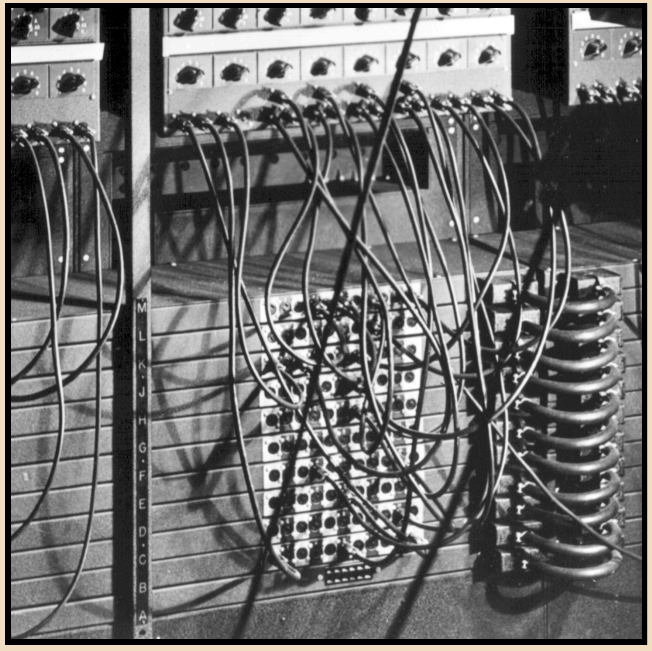
Tomas Petricek, Charles University, Prague

tomas@tomasp.net
https://tomasp.net
@tomasp.net



How did we get

From plugging cables...
Programming the ENIAC



...to languages?
Algol language definition

3.1. VARIABLES

3.1.1. Syntax

⟨variable identifier⟩ ::= ⟨identifier⟩
⟨simple variable⟩ ::= ⟨variable identifier⟩
⟨subscript expression⟩ ::= ⟨arithmetic expression⟩
⟨subscript list⟩ ::= ⟨subscript expression⟩|⟨subscript list⟩,
⟨subscript expression⟩
⟨array identifier⟩ ::= ⟨identifier⟩
⟨subscripted variable⟩ ::= ⟨array identifier⟩[⟨subscript list⟩]
⟨variable⟩ ::= ⟨simple variable⟩|⟨subscripted variable⟩

3.1.2. Examples

epsilon
detA
a17
Q[7,2]
x[sin(n×pi/2),Q[3,n,4]]

Charles University

Metaphors

Computers as giant electronic brains

Charles University

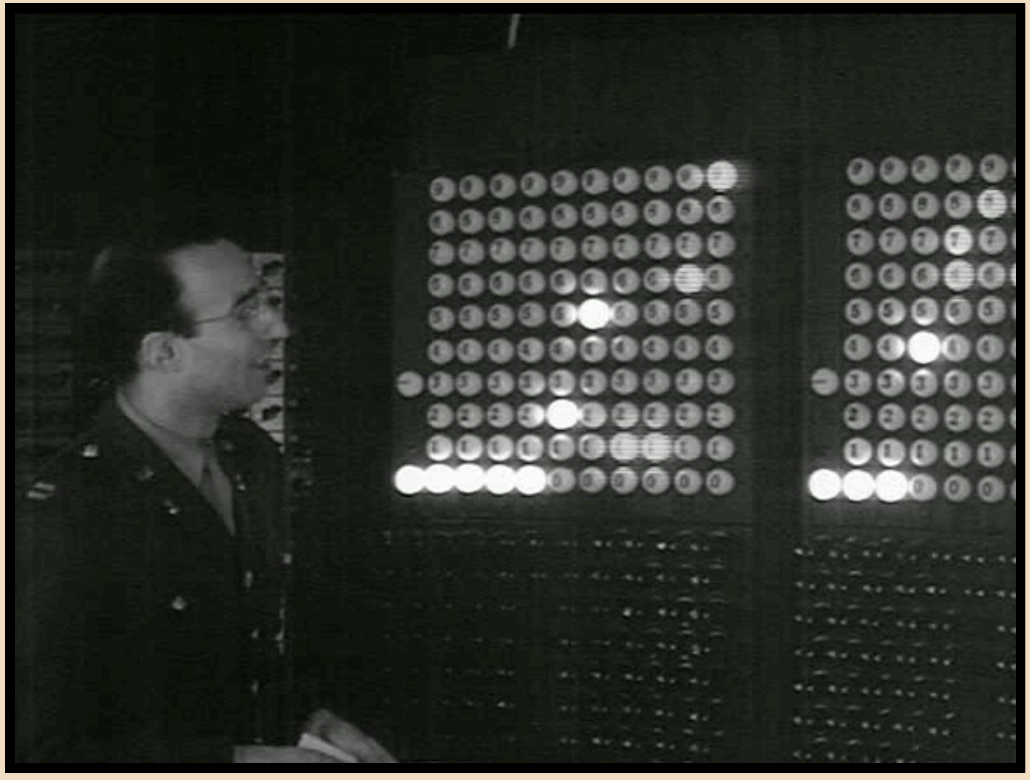


What was a computer in the 1940s?

Female computers doing numerical calculations by hand!

Patsy Simmers, Mrs. Gail Taylor, Mrs. Milly Beck, Mrs. Norma Stec

Charles University



ENIAC 1945 demonstration

Trajectory of a shell in 20 sec, not 40 hours

Labeled ping pong balls showing the "memory"

Charles University

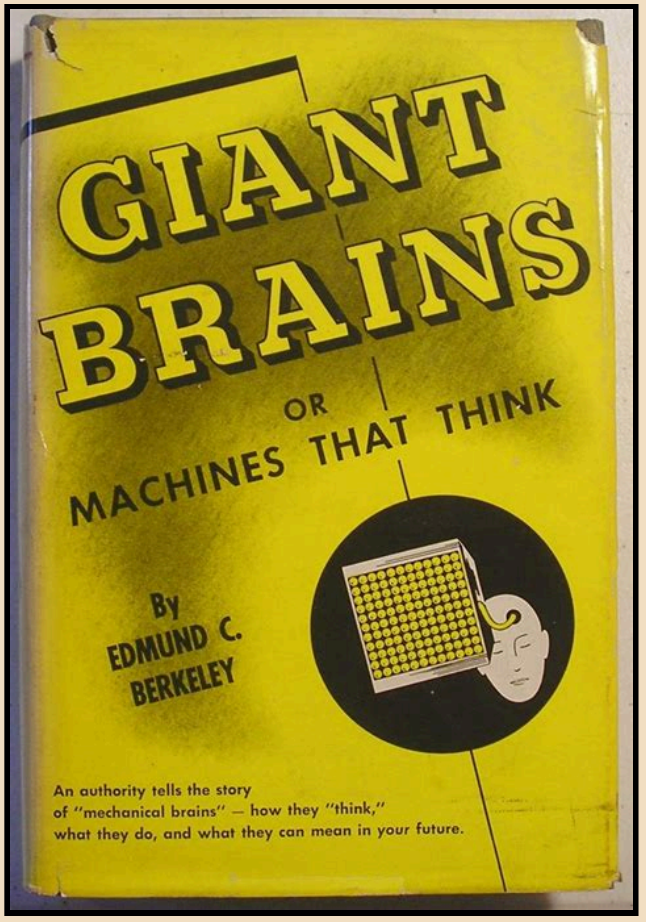
Electronic brains

Popular imagination

- Anthropomorphic metaphor
- Portrayal in media
- Berkeley's 1949 book

Cybernetics movement

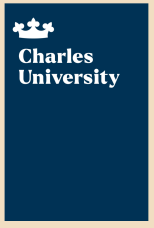
- Post-World War II movement
- Self-regulative control systems
- Electronic and biological alike



Charles University

Collaboration

Hacker programming tricks



EDSAC computer (1949)

Initial orders

- Read program from the tape
- Support for sub-routines

External orders

- Non-standard codes
- Interpretive routines

Debugging tricks

- Step-by-step execution
- Postmortem tapes



Charles University



The library of tapes on which subroutines are punched is contained in the steel cabinet shown on the left. The operator is punching a program tape on keyboard perforator. She can copy mechanically tapes taken from the library on to the tape she is preparing by placing them in the tapereader shown in the center of the photograph.

Sub-routine library

Open sub-routines

Copied at each point

Closed sub-routines

Using control transfer

Charles University

Program coding

Before a language

- **Brief Code** - John Mauchly (BINAC, 1949)
- **A-0** - Grace Hopper (UNIVAC, 1951-2)
- **Autocode** - Alick Glennie (Manchester, 1952)

Charles University

```
Step 1  a   =  (   b   +  c   )  /  b   * c   (Original equation)
Step 2  X3  =  (   X1  +  Y1  )  /  X1  * Y1  (New variable names)
Step 3  X3  03  09  X1 07 Y1 02 04 X1    Y1  (Replace operators)

Line 1  07Y10204X1Y1      (Packed into UNIVAC I words right
Line 2  0000X30309X1      to left and top to bottom.)
```

No.	SHORT CODE	Label	Replaces
1	ZZ X0 S2 S1 S0 39		READ S0, S1, S2, X0
2	X0 03 X0 07 S1	1	X0 = X0 + S1
3	S1 03 S1 07 S0		S1 = S1 + S0
4	S1 S2 41		TEST 1, S2, S1
5	X0 57		PRINT X0
6	ZZ 08		STOP

Brief Code / Short Code

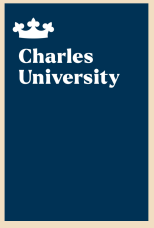
Step 2 to Step 3 translation done automatically by an editor

Not exactly a language yet...

Charles University

Collaboration

Mathematical theories



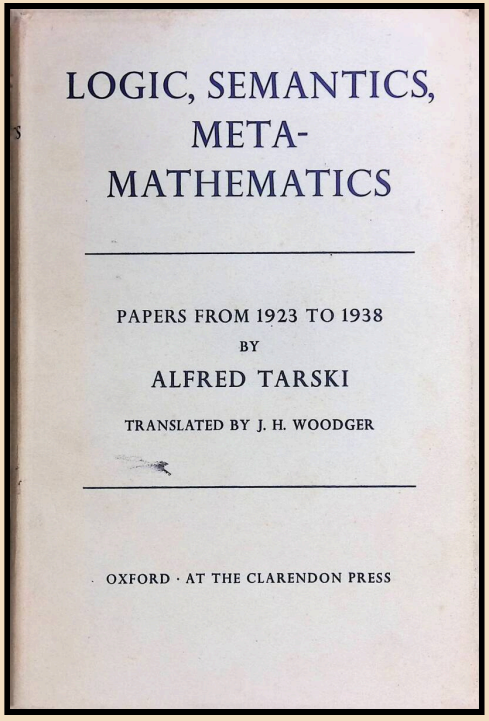
Formal language

Language models in logic

- Tarski, Carnap, early 20th century
- Language starts with a set of symbols

Language models in linguistics

- Formal structural linguistics
- Ongoing research on machine translation
- Chomsky's grammars (late 1950s)

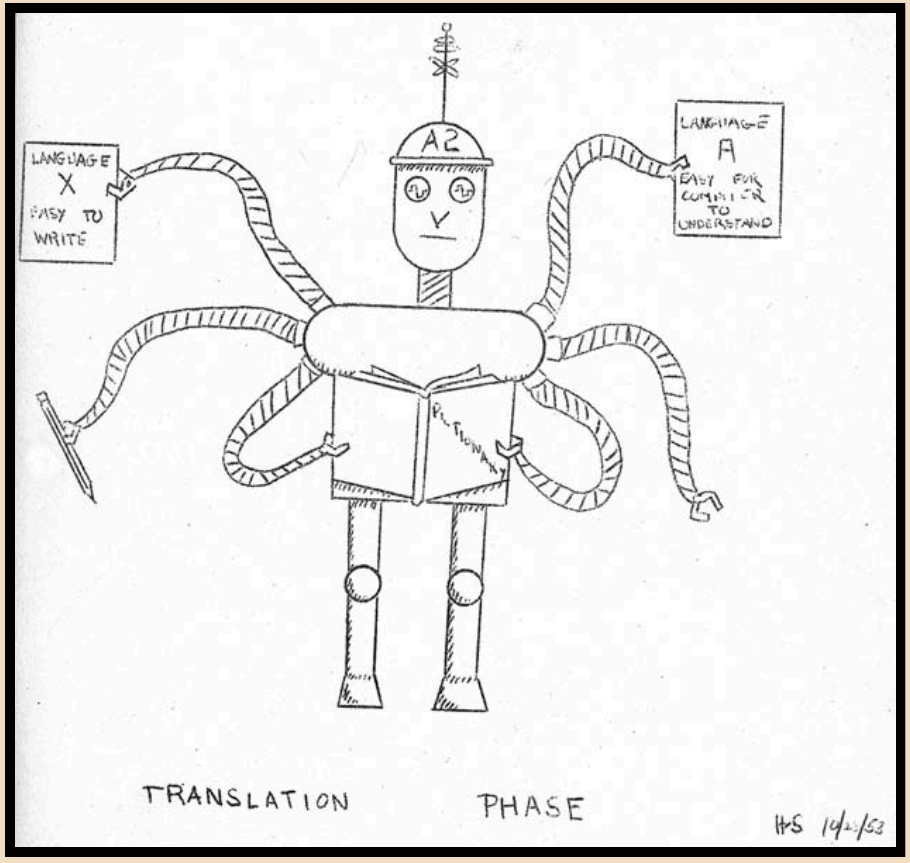

LOGIC, SEMANTICS, META-MATHEMATICS

PAPERS FROM 1923 TO 1938
BY
ALFRED TARSKI

TRANSLATED BY J. H. WOODGER

OXFORD · AT THE CLARENDON PRESS

Charles University



Translation metaphor

Hopper's illustration of the A-2 compiler (1954)

From giant electronic brains to logic and linguistics

Formal languages and grammars for description

Charles University

Collaboration

Managerial need

Charles University

Software gap

More software needed

- Industry crisis perception
- Getting the computers to do what we need!

Programming is the limit

- Creating software is hard
- Hiring programmers is hard
- Also, excluding women did not help!



Are YOU the man

to command electronic giants?

From the recent advance of electronic digital computers has emerged an exciting new job—creating instructions that enable these giant computers to perform logical operations for a variety of tasks in business, science and government.

You could be eligible for a position in computer programming. Because it is a new and dynamic field, there are no rigid qualifications. Do you enjoy algebra, geometry or other logical operations? Can you do musical composition or arrangement? Do you have an orderly mind that enjoys such games as chess, bridge or anagrams . . . finally, do you have a lively imagination?

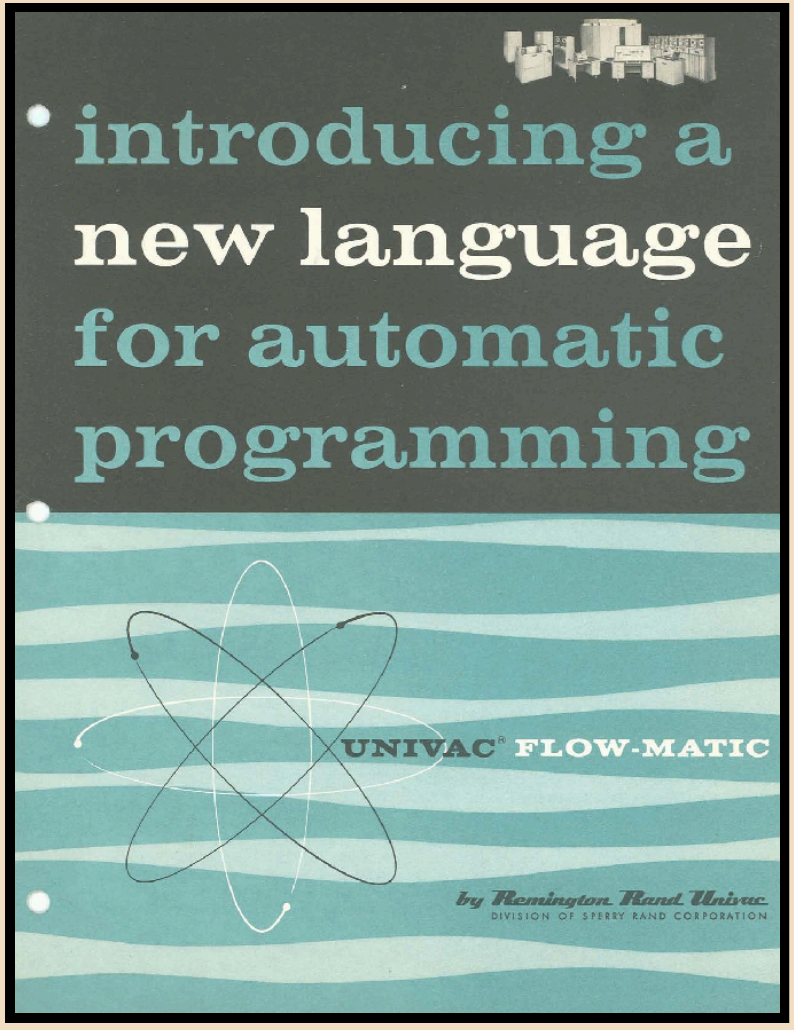
If you do, *you* can qualify. You will receive training (at full pay) and work at IBM's Engineering Laboratories—among the most modern in the world. For more information, write to: G. W. Woodsum, Dept. 203, International Business Machines Corp., Research Laboratory, Poughkeepsie, N. Y.

DATA PROCESSING
ELECTRIC TYPEWRITERS
TIME EQUIPMENT
MILITARY PRODUCTS

IBM

INTERNATIONAL BUSINESS MACHINES CORPORATION

Charles University



Automatic coding

User groups and computer installation managers

Solve the software gap!

Common universal notations
Cross-machine compatibility
Allow program exchange
Teaching at universities

Charles University

UNIQUE SAVINGS
of the
UNIVAC
FLOW-MATIC
SYSTEM



1

Virtually Eliminates Your Coding Load

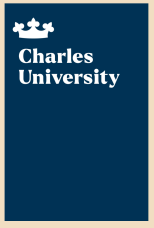
Your skilled programmers are freed from clerical drudgery to do more creative work. FLOW-MATIC shifts emphasis of the programming effort from detailed coding to problem definition and systems analysis. Slashes drastically the time required to program new or altered UNIVAC applications.

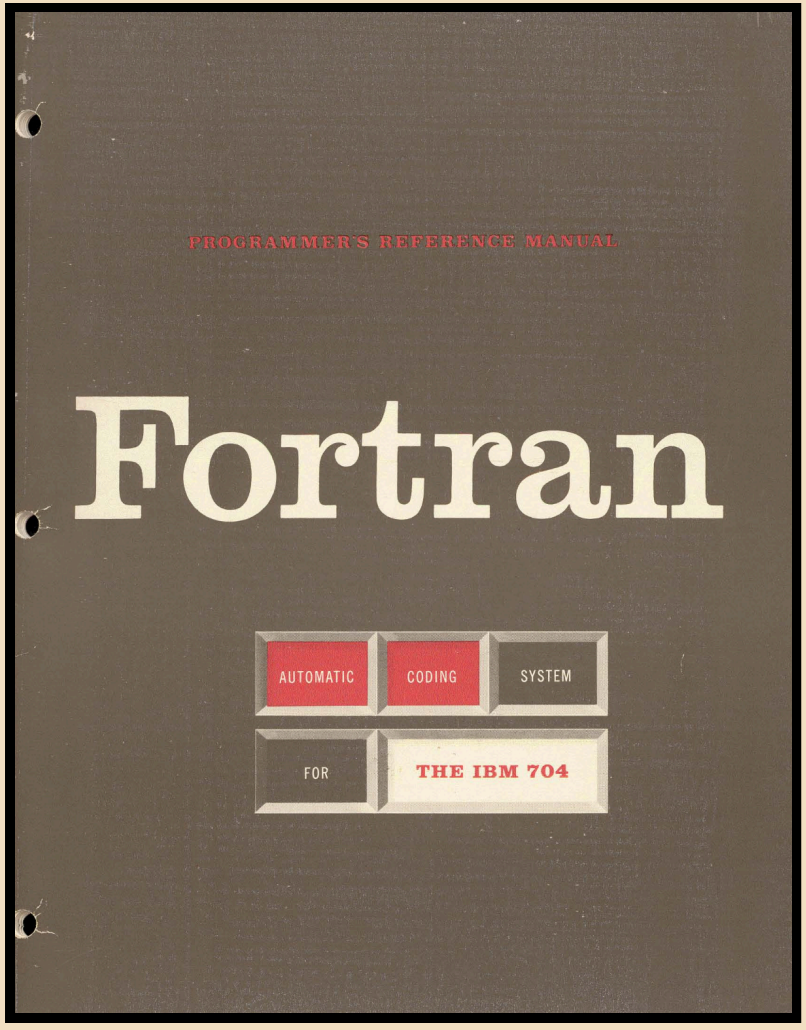
Would you guess the publication year (1957)?

Charles University

Collaboration

Programming language





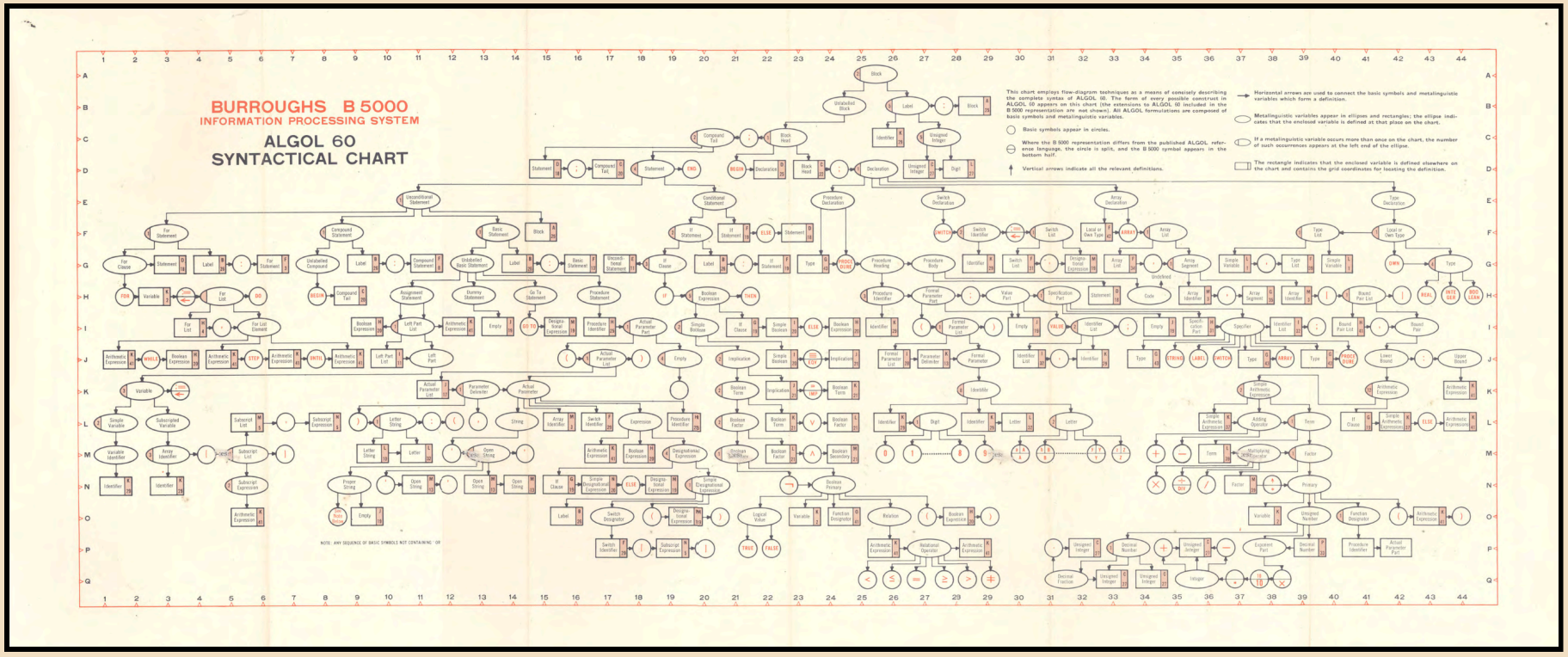
IBM FORTRAN
FORmula TRANslator

John Backus (IBM, 1957)

Translation metaphor, hacker tricks, managerial need

Not referred to as a programming language!

Charles University



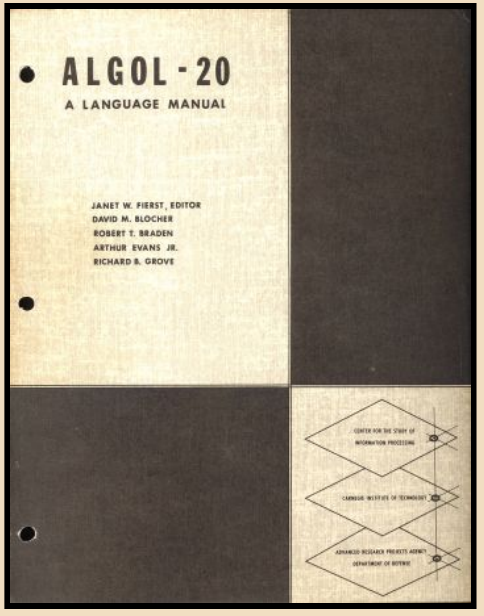
International Algebraic Language (IAL, 1958)

Charles University

Algol - Algorithmic Language

Scientific notation

- Extension of mathematical language
- Description of algorithms in publication
- Translatable into machine programs



Independent of a computer

A formal mathematical, linguistic object, not just a messy adjunct to the technical business of getting a computer to run.

Charles University

Programming language

Remarkable new idea



Hacker culture

Implementation tricks, sub-routines, external codes



Managerial culture

Needs to simplify programming and universal codes



Mathematical culture

Formal description using language of logic

Charles University

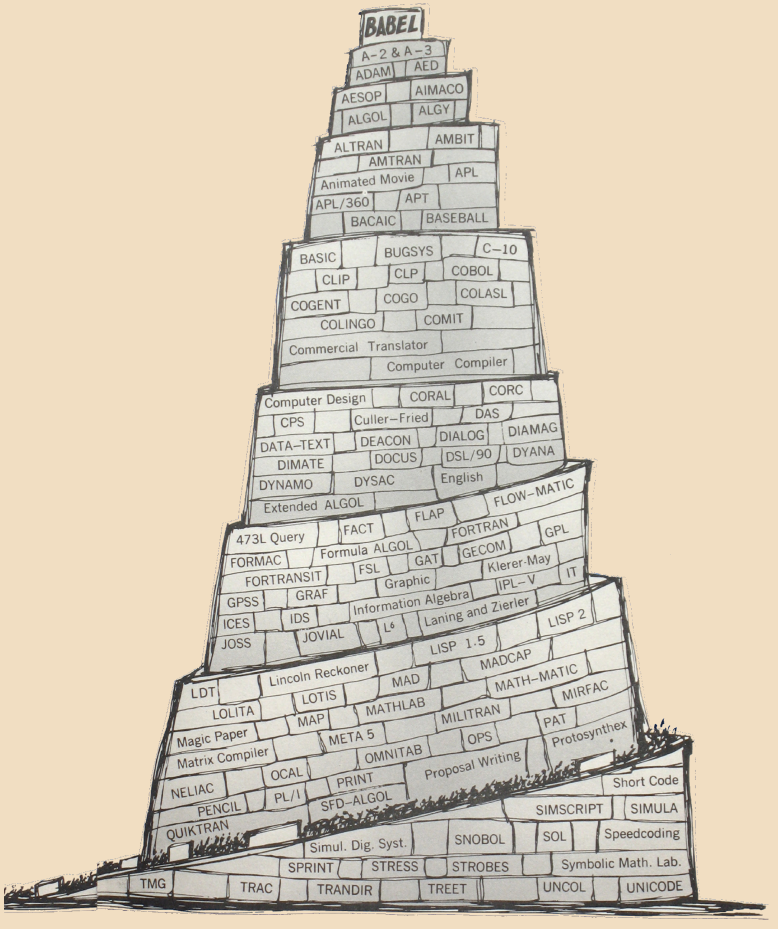
Limits of collaboration

Meeting of cultures but...

- **Algol** for mathematical theory
- **COBOL** for managerial business use
- **LISP** for symbolic AI and hackers

Boundary object

- Shared understanding
- Exchange of ideas between cultures
- Failure of universal languages



Cultures of Programming



Home → Continue to Part 4

Charles University