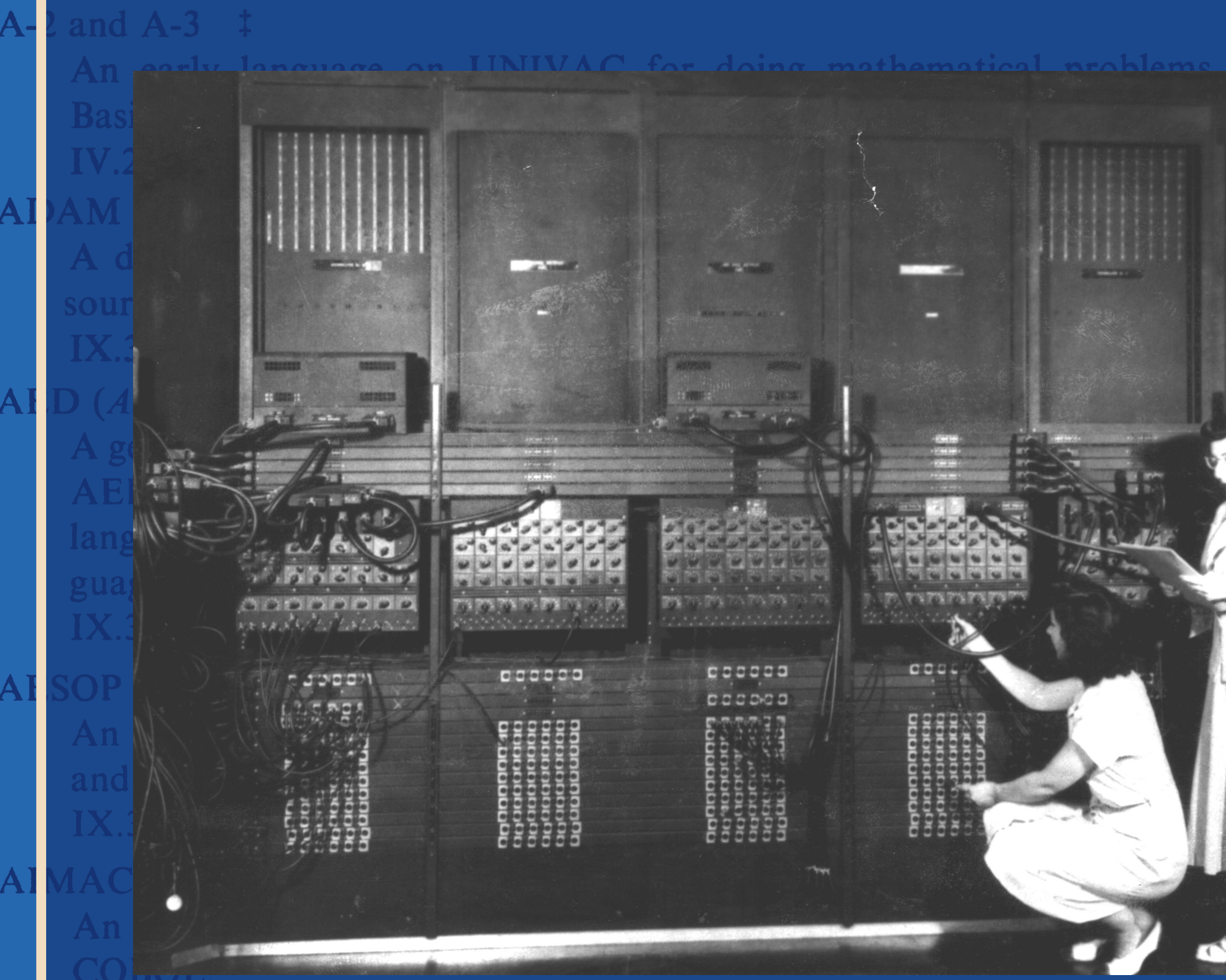# The invention of a programming language

This appendix contains a list of every language specifically listed in the outline, together with the acronym and a very brief description. The chapter (and subsection in which the language is described) is given. The numbers of the one or two books and/or references in which the language is defined are also given, along with reference to where a difference is less accessible than a slightly inferior document, both have been given. (It should be obvious that old reports are probably unavailable; they are being listed only for the sake of completeness.)

Languages considered as specialized (and discussed in Chapter IX) have been listed. Although they may be of quite general technical interest, they should nevertheless still be of some interest.

Read any of the interpretations and conclusions about a particular language should be with caution—see Sammet [SA68], which is similar (but not identical) to this Appendix. That article contains the names of individuals or organizations to contact for information. Those names have not here been included here because they are valid as of the spring of 1968 but not necessarily much beyond that.

The idea that a programming language is a standalone object independent of a specific computer came about from the meeting of three cultures of programming.

Mathematical culture

By the early 1950s, the central metaphor for thinking about programming became translation. Proponents of the mathematical culture turned to formal logic and linguistics and adopted formal grammars as a method for specifying the syntax of programming languages.

Hacker culture

Hackers have been inventing tricks to make programming easier since the ENIAC computer (1945). Early automatic coding systems A-0 and Short Code translated a limited form of symbolic notation to machine code.

The hacker culture provided the necessary implementation tricks, the managerial culture had a practical need for the idea, and the mathematical culture provided the necessary formal language description tools.

Managerial culture

Computer installation managers wanted cross-machine compatibility of programs. They realized that a common or universal notation would enable the exchange of programs among and within organizations.

Rise of programming languages

Illustration from a book documenting the growth of programming languages by Jean E. Sammet, published in 1969.

COBOL (COmmon Business Oriented Language)

An English-like language suitable for business data processing problems. Developed and maintained by a committee of representatives from manufacturers and users. It has been implemented on many computers. A USASI Standard has been approved.
V.2.1   [p. 380]   p. 381.

COGO (COordinate GEOmetry) †

A specialized language for solving coordinate geometry problems.
IX.2.2.2   [RS67], p. 694.

COLASL

A language for numerical mathematical problems, based on a special typewriter which permits two-dimensional (i.e., natural) input of mathematical expressions.
IV.7.2   [BQ62], p. 312.

COLINGO (Compile On LINe and GO) †

A formalized English-like query system for command and control applications.
IX.3.2.2   [SZ65], p. 702.

COMIT

The first significant string-handling and pattern-matching language.
VI.6   [MT61], p. 468; [MT61a], p. 468.

Commercial Translator ‡

An English-like language for doing business data processing, supplanted by COBOL.
V.1   [IB60a], p. 378.

FORTRAN (FORmula TRANslator)

The first language to be used widely for solving numerical problems. Originally developed by IBM on the 704, it has existed in many versions.

A language with a syntax similar in spirit to that of COBOL, but with some facilities from ALGOL added.
V.2.5   [GZ61], p. 379.

GPL (Generalized Programming Language)

A relatively new attempt at defining a general language, also conceived as a teaching vehicle.
IV.4.5.7   [GW00], p. 305.

GPSS (General Purpose Systems Simulator) †

A language for discrete simulation problems based on a block diagram approach.

ICES (Integrated Civil Engineering System)

A generalized system for civil engineering, including specific languages (e.g., COGO) and some facilities for defining new languages.
IX.2.2.3   [RS65], p. ?; [RS67a], p. 694.
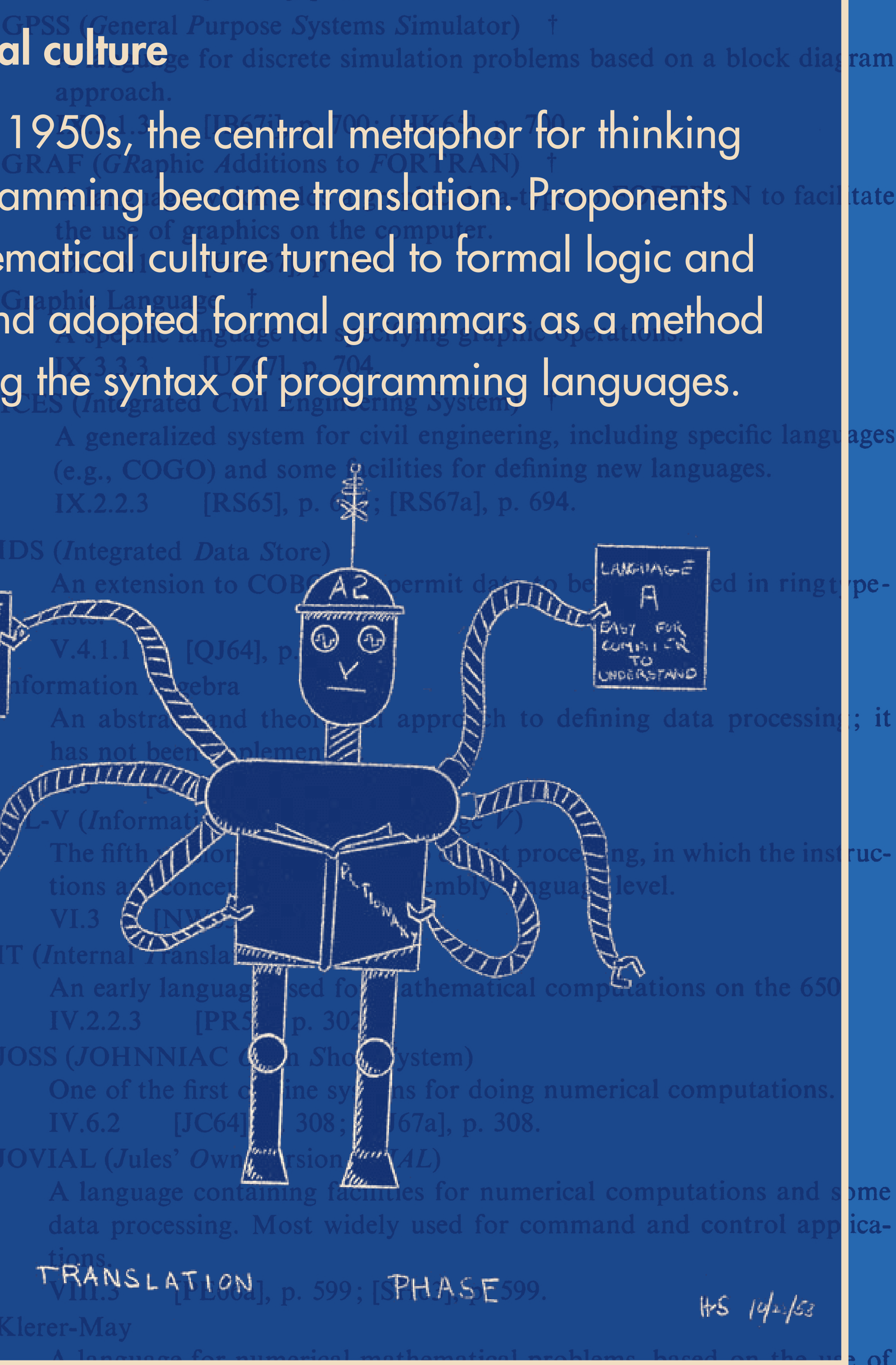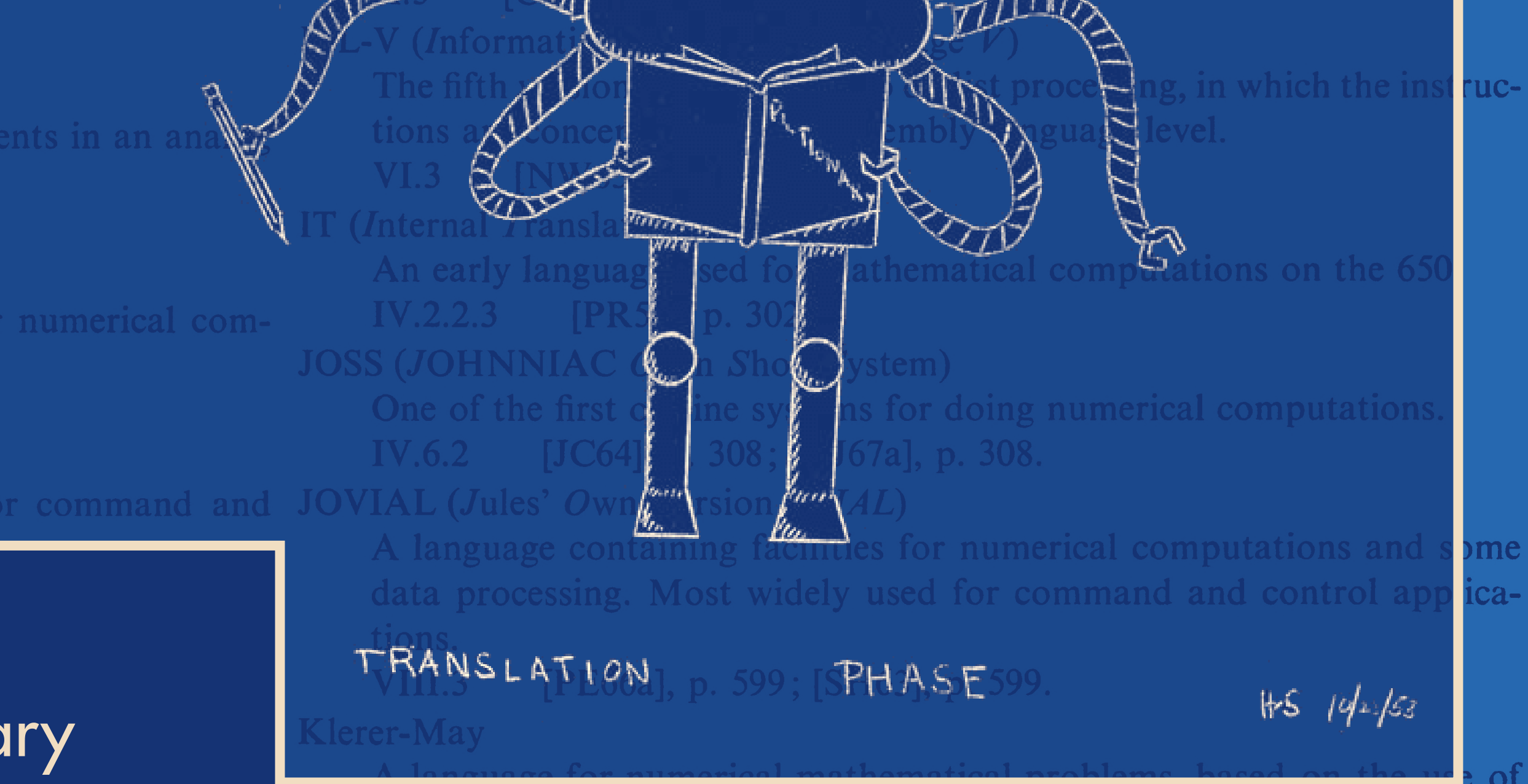
IDS (Integrated Data Store)

An extension to COBOL to permit data to be stored in ring-type structures.

JOSS (JOHNNIAC Open Shop System)

One of the first on-line systems for doing numerical computations.
IV.6.2   [JC64], p. 308; [JC67a], p. 308.

JOVIAL (Jules' Own Version of IAL)

A language containing facilities for numerical computations and some data processing. Most widely used for command and control applications.