

Índice general

Índice de figuras

Índice de cuadros

Capítulo 3

El Lenguaje de Programación Pascal

3.1. Introducción

*A programming language called Pascal is described which was developed on the basis of Algol 60. Compared to Algol 60, its range of applicability is considerably increased due to a variety of data structuring facilities. In view of its intended usage both as convenient basis to teach programming and as an efficient tool to write large programs, emphasis was placed on keeping the number of fundamental concepts reasonably small, on a simple and systematic language structure, and on efficient implementability. A one-pass compiler has been constructed for the CDC 6000 computer family; it is expressed entirely in terms of Pascal itself.*¹ [?]

El Lenguaje de Programación Pascal fue creado por el profesor Niklaus Wirth² a finales de la década de los sesenta del siglo XX. En 1970 fue finalmente publicado, fijando dos objetivos en su diseño arquitectónico:

- i. Crear un **lenguaje claro y natural orientado a la enseñanza** de los fundamentos de la programación de computadores. Por ello se estructuran los módulos como funciones y procedimientos.
- ii. Definir un lenguaje que **permita realizar programas lo más eficientes posibles**. El tipado de datos es explícito.

Pascal recibe su nombre en honor al matemático francés Blaise Pascal (ver Anexo ??).

3.2. Influencias del Lenguaje Pascal

3.2.1. Fortran (The IBM Mathematical Formula Translating System)

Fortran, inicialmente conocido como **FORTTRAN** es el acrónimo de *The IBM Mathematical Formula Translating System*.

Fortan se trata del primer lenguaje de alto nivel. Es multipropósito y se basa en el paradigma de la programación estructurada.

Su origen tiene que ver con la necesidad de crear aplicaciones científicas de manera más sencilla y lógica para el entendimiento humano.

The FORTRAN language is intended to be capable of expressing any problem of numerical computation. In particular, it deals easily with problems containing large sets of formulae and

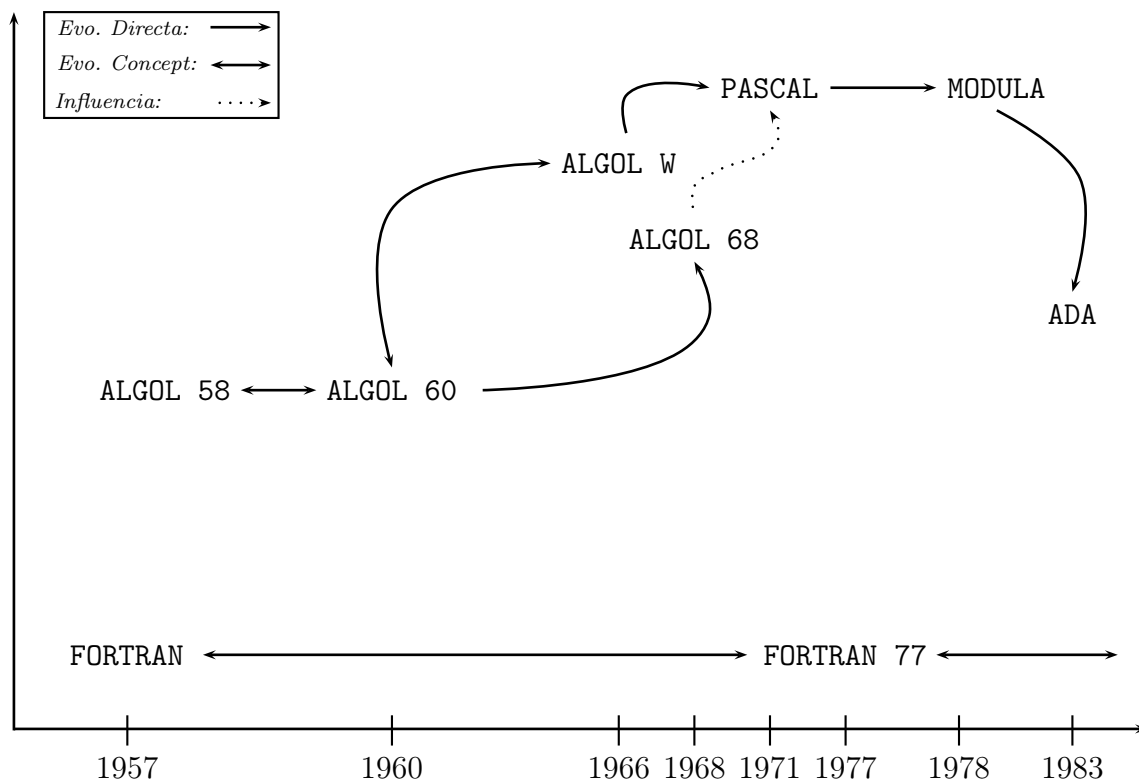


Figura 3.1: Relaciones entre los primeros lenguajes de programación.

many variables, and it permits any variable to have up to three independent subscripts. However, for problems in which machine words have a logical rather than a numerical meaning it is less satisfactory, and it may fail entirely to express some such problems. Nevertheless, many logical operations not directly expressible in the FORTRAN language can be obtained by making use of provisions for incorporating library routines. ³ [?]

El primer proyecto de compilador de FORTRAN fue un Milestone que ocupaba 15KB aproximadamente. Era muy rudimentario y funcionaba con rutinas muy primitivas de los SSOO de la época, prácticamente era código ensamblador.

El compilador oficial de FORTRAN fue escrito entre 1954 y 1957 a cargo de John W. Backus y grandes programadores como: Sheldon F. Best, Harlan Herrick, Peter Sheridan, Roy Nutt, Robert Nelson, Irving Ziller, Richard Goldberg, Lois Haibt and David Sayre. La primera ejecución del compilador se realizó sobre una máquina IBM 704.

Su primeros programas fueron para control energético de reactores nucleares. Demostraba ser mucho más rápido que otras soluciones tradicionales sobre Lenguaje Ensamblador.

El Lenguaje Fortran ha sido parte de seis estandarizaciones:

- I. FORTRAN o FORTRAN 66: La característica más destacada es la separación de las fases de compilación, además de la posibilidad de enlazar con rutinas de lenguaje ensamblador.
- II. FORTRAN 77: Entre sus características destacan:
 - i. Bucles DO con variable índice de incremento y decremento.
 - ii. Bloque de secuencias: {IF...THEN...ELSE...ENDIF.}

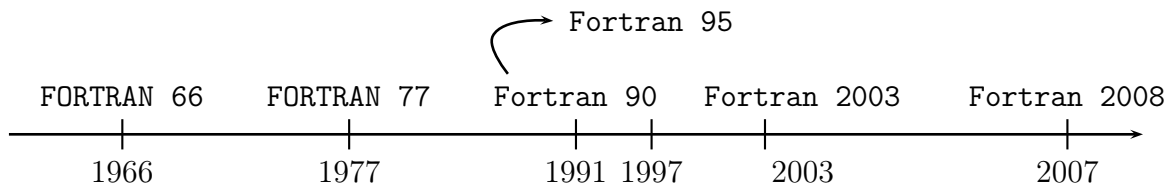


Figura 3.2: Evolución del Lenguaje Fortran.

- iii. Pruebas antes de compilación de bucles `{DO}`.
- iv. Tipo de dato `CHARACTER`.
- v. El símbolo apostrofe (') como delimitador de conjuntos de caracteres.
- vi. Final de un programa sin necesidad de usar la palabra `{STOP}`.

III. Fortran 90: Sus principales novedades son:

- i. Nuevas estructuras de flujo: `{CASE & DO WHILE}`.
- ii. Estructuras de datos tipo `RECORD`.
- iii. Mejora en el manejo de `ARRAY` (nuevos operadores).
- iv. Memoria dinámica.
- v. Sobrecarga de operadores.
- vi. Paso de argumentos por referencia.
- vii. Control de precisión y rango.
- viii. Módulos (paquetes de código).

IV. Fortran 95:

- i. Construcciones `{FORALL}`.
- ii. Procedimientos `PURE` y `ELEMENTAL`.
- iii. Mejoras en la inicialización de objetos.
- iv. Sentencia `{DO}` para tipos de datos: `REAL` y `DOUBLE PRECISION`.
- v. Sentencia `{END IF}` para terminar bloque.
- vi. Sentencia `{PAUSE}`.
- vii. Incorporación de ISO/IEC 1539-1:1997 que incluye dos tipos de módulos opcionales:
 - a. `STRINGS` dinámicos ISO/IEC 1539-2:2000.
 - b. Compilación condicional ISO/IEC 1539-3:1998.

V. Fortran 2003:

- i. Soporte de Programación Orientada a Objetos: Extensión de tipos, Polimorfismo y completo soporte para TADS (Tipos Abstractos de Datos) entre otras características.