

UN IDIOMA PARA COMPUTADORES: ALGOL (I) (*)

Por ALFONSO RECUERO FORNIER

Ingeniero de Caminos, Canales y Puertos

A la hora de preparar un programa para un ordenador hemos de expresarnos en un lenguaje que comprendamos, tanto nosotros como el ordenador. Nuestro lenguaje es demasiado confuso para el ordenador y el del ordenador demasiado conciso para nosotros. Hemos de llegar a una solución de compromiso, esto es, un idioma que nos resulte comprensible a nosotros, y que mediante el traductor adecuado lo sea también para el ordenador. Esto da origen a los llamados lenguajes simbólicos, de los que existe gran número, adecuado cada uno a cierto tipo de aplicaciones y cierta gama de computadores. Algol es un idioma adecuado para aplicaciones matemáticas y técnicas, diseñado por sucesivas comisiones de expertos, que han ido produciendo las distintas versiones, tres principales hasta el momento, de este idioma. Vamos a tratar en la presente serie de artículos de la versión Algol 60, versión actualmente en uso, de gran sencillez de comprensión en sus niveles elementales y de enorme potencia. En este primer artículo vamos a referirnos a los primeros conceptos; tales como símbolos básicos, nombres y tipos de variables, instrucciones aritméticas, etc. Presuponemos en el lector cierta familiaridad con la programación, en la que no vamos a entrar. Una vez tratada la suficiente materia, se incluirán algunos ejemplos de programas de interés.

INTRODUCCION

A la hora de ponernos en comunicación con un computador debemos decidir en qué idioma vamos a conversar con él. Nosotros tenemos nuestro idioma, adecuado a nuestra constitución, y de igual modo el computador posee su propio idioma, igualmente adecuado a su "fisiología".

El tratar de comunicarnos en uno de estos dos idiomas, pura y simplemente, comporta gran dificultad. Para nosotros sería comodísimo contarle nuestro problema en nuestras propias palabras, pero dada la multitud de formas que podemos emplear para decir la misma cosa y las cosas que damos como sobreentendidas al hablar, el intérprete que se lo tradujese al computador, cuyo idioma es extremadamente conciso y preciso, tendría enormes dificultades. De igual modo, a nosotros nos es sumamente costoso expresar nuestras ideas en el escueto y para nosotros casi inabordable idioma del computador, idioma que en su más pura expresión suele ser una larga tira de 0 y 1 cada palabra.

Debemos, pues, llegar a una solución de compromiso, algo que podamos nosotros comprender y expresar claramente, en general, perdiendo simplemente florituras, y que el intérprete pueda traducir fácilmente al idioma del computador.

Esta solución intermedia puede ser muy próxima al idioma de la máquina, como ocurre con idiomas de tipo Assembler, Autocode, etc., con lo que se obtendrán traductores muy simples y eficaces, pero también una mayor dificultad por nuestra parte para expresarnos; o bien puede ser muy próxima a nuestro idioma, adaptando a nuestras necesidades.

Así Cobol es adecuado para temas comerciales, y Fortran y Algol lo son para temas científicos y técnicos.

El primer intento de elaboración del Algol data de 1958. Una comisión de expertos reunidos en Zurich elaboró un documento básico que dio origen a la primera

(*) Se admiten comentarios sobre el presente artículo, que pueden remitirse a la Redacción de esta Revista, hasta el 31 de octubre de 1970.

versión de Algol: el llamado Algol 58. Posteriormente se introdujeron diversas modificaciones que dieron lugar a un nuevo documento base, que se elaboró en 1960 en París, que generó el Algol 60, que ha ido sufriendo ligeras modificaciones que no lo alteran en forma esencial.

Esta versión es la que se viene utilizando hasta hoy. La mayoría de los computadores para usos científicos del mercado admiten este idioma, si bien su empleo no es todo lo amplio que desearon sus creadores en lo que a programación se refiere, sobre todo en España, donde su uso es limitadísimo. Sin embargo, es un idioma que se presta muy bien a las comunicaciones científicas, y en tal campo su uso es amplio. Así las revistas especializadas lo usan con gran frecuencia para la publicación de procesos y algoritmos de cálculo. Tiene la ventaja de ser un idioma casi coloquial (en inglés) y con gran concisión de conceptos, por lo que sus expresiones son de fácil seguimiento y comprensión. Permite así plasmar muy exactamente nuestros esquemas mentales.

Pero la evolución de los computadores es constante y las enormes posibilidades de la tercera generación han desbordado el Algol 60. Ya desde su aparición se ha ido perfeccionando y evolucionando hacia el Algol X. X es actualmente 68. Recientemente ha aparecido el documento base de Algol 68, elaborado finalmente en Amsterdam y adaptado a las nuevas posibilidades de los computadores. Hasta el momento sólo ha aparecido el citado documento. Aún habrá de pasar algún tiempo hasta que aparezcan traductores, y algo más hasta que se use.

En tanto, y aun después, Algol 60 se seguirá usando, puesto que existen actualmente traductores eficaces y gran número de máquinas que no podrán usar Algol 68.

Por estas razones considero que aún tiene gran interés una mayor difusión de Algol 60 en España.

Generalidades.

Algol procede de la fusión de las palabras Algorithmic Language, génesis que aclara netamente su finalidad.

Al no estar adscrito a un sistema particular debe permitir que conservando sus ideas esenciales, existan ligeras variantes que permitan a cada traductor usar la máquina de la forma más eficaz.

En el presente trabajo trataré de mantenerme en las ideas más generales, indicando en algunos casos las variantes más comúnmente usadas.

Símbolos básicos.

Algol 60 usa 116 símbolos básicos:

Números: 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Letras minúsculas: a | b | c | d | e | f | g | h | i | j | k | l | m | n
o | p | q | r | s | t | u | v | w | x | y | z

Letras mayúsculas: A | B | C | D | E | F | G | H | I | J | K | L | M | N
O | P | Q | R | S | T | U | V | X | Y | Z

Operadores aritméticos: + | - | × | / | ÷ | ↑

Operadores de relación: < | ≤ | = | ≥ | > | ≠

Operadores lógicos: ≡ | ⊃ | ∨ | ∧ | ⊔

Operadores secuenciales: go to | if | then | else | for | do
 Declaradores: o w n | Boolean | interger | real | array | switeh | procedure
 Especificadores: string | label | value
 Separadores: . | , | 10 | : | ; | = | _ | step | unty | whiles | comment
 Paréntesis: () | [] | ' ' | begin | end
 Valores lógicos: True | false

Normalmente los computadores no admiten todos estos símbolos básicos como unidades independientes, y para simplificar no se suelen usar las letras minúsculas, y los símbolos que son palabras se sustituyen por el conjunto de letras que las forman, resaltadas, por ejemplo, entre apóstrofos, o no.

Suele sustituirse el X por * y $\uparrow$ por **.

En otros casos se sustituyen algunos operadores de relación u operadores lógicos por conjuntos de letras. (Ejemplo: $\geq$ por CREQ, $\equiv$ por EQUIV.)

Nombres.

Los nombres con que se designan los identificadores están formados por una sucesión de letras y números, siendo siempre el primer carácter de una letra.

No se limita el número de caracteres que puede tener un nombre, pero la mayoría de los computadores sólo reconocen los primeros: los seis, los ochos, etc., según el computador.

No deben aparecer espacios ni caracteres distintos de letras o números.

Ejemplos válidos: alfa 1, P32am, Extravagante.

Ejemplos no válidos: 1 alfa, P(32)am, Extra-vagante.

Números.

La forma más general que puede adoptar un número en Algol consta de:

- 1.º Signo (+ o —).
- 2.º Número decimal, formado por la parte entera y la parte decimal separadas por un punto (.).
- 3.º Potencia de 10; de exponente entero positivo o negativo, exponente que se escribe tras la parte decimal, separado de ésta por un subíndice diez (₁₀).

Simplificaciones.

- a) Si el número o el exponente de 10 son positivos puede omitirse el signo.
- b) Si la parte entera del número decimal es cero puede omitirse.
- c) Si la parte decimal del número decimal es cero puede omitirse junto con el punto. Si se mantiene el punto habrá de ir seguido de, al menos, una cifra.
- d) Si el exponente de 10 es cero puede suprimirse la tercera parte.
- e) Pueden añadirse cuantos ceros no significativos se deseen al número decimal.

Ejemplos válidos.

$$+ 0.0031 = +.0031 = .0031 = 31_{10} \text{—}4 = + 3 \cdot 1_{10} \text{—}03.$$

Ejemplos no válidos.

$31_{.10} - 4 = 3.1_{10} - 03 \cdot 0.$

Tipos de variables.

Podemos usar en Algol 60 tres tipos de variables: enteras, reales y booleanas.

La forma de indicar al computador a qué tipo de éstos pertenece cada variable es mediante una declaración de tipo, que consiste en el símbolo INTEGER, REAL o BOOLEAN, respectivamente, seguido de la lista de los nombres de las variables del tipo indicado, separados éstos por comas y acabando en punto y coma.

Ejemplo: REAL X, INDICE, P72;

Variables con subíndices.

Podemos usar variables con cualquier número de índices. En tal caso, los índices ordenados y separados por comas, encerrados todos entre paréntesis cuadrados — [] — van a continuación del nombre de la matriz.

La forma de indicar al computador que se trata de una matriz es mediante una declaración de matriz:

— mediante el símbolo ARRAY, precedido del símbolo que indica el tipo de los elementos de la matriz, a menos que sean reales, en cuyo caso se puede prescindir del tipo. Tras los símbolos indicados se coloca la lista de nombres, acompañados cada uno de los límites inferior y superior de cada índice, separados estos límites por dos puntos — : —, separando los grupos de límites por comas y encerrando todo el conjunto entre paréntesis cuadrados. Los nombres irán separados por comas y al final se cierra con punto y coma.

Si los límites de los índices de varias matrices coinciden no es preciso especificarlos por separado. Basta colocar los nombres separados por comas y el paréntesis de límites a continuación del último nombre del grupo.

Los índices son números enteros, por lo que si su determinación es consecuencia de una expresión de resultado real, el computador tomará como valor del índice el entero más próximo al número real hallado.

Ejemplo.

```
INTEGER ARRAY, ALFA [1:6], BETA, BETA 2 [— 3 : 100, 1 : 3];  
ARRAY INDEX [1 : 10, 1 : 10, 1 : 10, 1 : 10];  
BOOLEAN ARRAY, PEP [NUM : NUM + 100];  
BETA 2 [3 X p — 1, q + 2].
```

REFERENCIAS

Cuando nos interesa destacar una instrucción podemos personalizarla asignándole una referencia. Esta referencia puede ser un número natural o un nombre, que se coloca precediendo a la instrucción de que se trata, y separada de ésta por dos puntos — : —.

Ejemplo.

```
40 : a : = a + 1;  
L31: if a = b Then goto L32;
```

Switchs.

Un switch podemos considerarlo como una matriz, de una sola fila, en la que sus elementos son referencias.

Se le indica al computador mediante una declaración de switch que está formada por el símbolo SWITCH seguido del nombre de un identificador, los símbolos dos puntos e igual — : = — y a continuación la lista de referencias, elementos del switch, separadas por comas y acabando con punto y coma.

Ejemplo.

SWITCH SS : = 11, 12, 13;

Si en algún punto se hace referencia a SS [i] el computador entenderá "la referencia que hace número i en la lista SS".

EXPRESIONES ARITMETICAS

Los signos aritméticos empleados en Algol 60 son:

+	—	X	/	÷	↑
Suma	Resta	Producto	División	División entre enteros	Elevación a potencia

Las expresiones aritméticas en Algol son a un solo nivel. Ya vimos cómo los subíndices se encerraban entre corchetes. Los exponentes se separan de la base por el símbolo.

El computador ejecuta las operaciones de una expresión matemática según la siguiente jerarquía:

- 1.º ↑
- 2.º X, /, ÷.
- 3.º +, —.

Entre operaciones de igual jerarquía se realizan de izquierda a derecha.

Si hubiese paréntesis, operará primero éstos, empezando por los más internos.

Ejemplo.

$$a + b \times c \uparrow (2 + m)/3.$$

El orden de ejecución es:

$$2 + m \quad c^{(2+m)} \quad b \cdot c^{(2+m)} \quad \frac{b \cdot c^{(2+m)}}{3};$$

$$a + \frac{b \cdot c^{(2+m)}}{3}$$

$$a \times b/c/d = (a \times b)/(c \times d) = a/c \times b/d = \frac{a \cdot b}{c \cdot d}$$

Pueden usarse cuantos paréntesis sean convenientes. Los innecesarios no dificultan el trabajo del computador, pero la omisión de uno necesario puede cambiar totalmente una expresión.

Según el tipo de operandos el resultado es:

Operandos:
ambos enteros
alguno real
alguno real
cualesquiera

Operación:
+, —, X, ÷
+, —, X
÷
↑, /

Resultado:
entero
real
carece de sentido
real

Carecen de sentido para el computador aquellas expresiones que carecen de sentido en el campo real, tales como dividir por cero o elevar un número negativo a una potencia no entera.

INSTRUCCIONES ARITMETICAS

Las instrucciones aritméticas son de la forma:

Variable: = Expresión aritmética, limitador.

En la asignación de valor a una variable se usa el signo : = (dos puntos, igual), signo al que llamaremos "toma el valor" con objeto de poner de manifiesto la asimetría de su significado.

Ante una instrucción del tipo indicado, el computador calcula el valor de la expresión aritmética, con los valores que en ese momento tienen las variables que en ella aparecen, y dicho valor calculado se lo asigna a la variable situada a la izquierda del signo "toma el valor".

Así, si la variable *a* tiene el valor 3, antes de pasar por la instrucción:

$$a := 1 + a,$$

tendrá el valor 4 una vez realizada instrucción.

Una instrucción modifica solamente el valor de la variable situada a la izquierda del signo : =.

Puesto que no hay limitación en la extensión de una expresión aritmética, debemos indicar, mediante un limitador, dónde acaba ésta. Dicho limitador es el símbolo ; (punto y coma) salvo cuando la instrucción deba ir seguida de los símbolos ELSE o END.

Si deseamos asignar el mismo valor a varias variables, podemos hacer una asignación múltiple.

Ejemplo.

```
a := b := c := 1;  
a := b := m + P/r 2;
```

Si a una variable declarada como entera se le asigna el valor que toma una expresión cuyo resultado es real, tomará dicho resultado redondeando al entero más próximo.

Salto.

El computador va realizando las instrucciones del programa de forma secuencial, a menos que le indiquemos que altere la secuencia mediante una instrucción de salto.

Se la indica mediante el símbolo GOTO y la referencia de la instrucción siguiente a realizar, y a partir de la cual seguirá secuencialmente.

Ejemplo.

```
GO TO L10;
```

La instrucción se acaba con punto y coma.