Evaluación automática de fórmulas del sistema K1*

Dora Sánchez García

La posibilidad de computar ciertos aspectos de la lógica tiene ya una larga historia cuyo comienzo puede situarse en los trabajos de Newell-Shaw-Simon (1957), y de Hao Wang (1960). Nos gustaría subrayar la importancia de este tipo de trabajos con unas palabras del profesor Quine que nos parecen sumamente significativas: «(...) la programación llega a ser la prueba del formalismo. Estar totalmente formalizado es estar programado. (...) Un programa tiene que ser el análisis último de un formalismo, debido a la falta de intuición por parte de la máquina»[1].

Por supuesto que el programa que presentamos no constituye el análisis último del sistema K1, pero esperamos que pueda constituir una aportación, por modesta que sea, en este sentido.

Como es'sabido, el sistema K1 del profesor Kalinowski se sirre del método de las tablas de verdad como procedimiento algorítmico para determinar si una expresión bien formada cualquiera es o no deducible. Este método es obviamente computable, y vamos a presentar un programa en lenguaje BASIC que lo computa.

Antes de pasar a la exposición del programa conviene que hagamos algunas observaciones.

Salvo la traducción de los distintos símbolos que componen una fórmula a ñumeros, que es obviamente necesaria, no hemos necesitado introducir ninguna modificación en la estructura de las fórmulas para su tratamiento automático.

---

* El programa que presentamos aquí constituye una versión simplificada de un programa más amplio que incluye además de las rutinas que componen el que aquí exponemos, una rutina para la comprobación de si la fórmula que se pretende probar está bien formada o no (en esta versión más simple sólo se pueden introducir como datos fórmulas bien formadas), y otra rutina para la eliminación de negadores múltiples en caso de que los hubiera. Asimismo las subrutinas de funtores deónticos y de juntores diádicos están elaboradas en esta versión de forma a que estén lo más próximas posible a la intuición. Nos han parecido oportunas todas estas simplificaciones dado el carácter introductorio de las jornadas.

1. *Aspectos de la filosofía de W. V. Quine*, «Teorema», número monográfico, 1975, pp. 149-150. Estas palabras pertenecen a la respuesta que el profesor Quine dió al profesor Beneyto con ocasión de la presentación por parte de este último de un programa que computa el método del análisis veritativo-funcional de Quine. El conocimiento de este programa ha sido sin duda importante a la hora de concebir el aquí expuesto.

El hecho de que el sistema K1 utilice la notación polaca favorece el que se pueda tratar una fórmula como una lista ordenada de números, ya que no tenemos ningún signo de puntuación que ayude a determinar el alcance de las constantes lógicas.

El vocabulario que utilizamos es el del sistema K1, con la salvedad de que se sustituye «α» por «@» impuesta naturalmente por los recursos de impresión del ordenador.

La traducción numérica de los juntores diádicos es la siguiente:

C	(«implicación»)	= 4
A	(«disyunción inclusiva»)	= 5
K	(«conjunción»)	= 6
D	(«disyunción exclusiva»)	= 7
E	(«coimplicación»)	= 8

El negador, tanto nominal como proposicional, se traduce por un 3. La traducción de «α» es 2.

Los funtores deónticos se traducen como sigue:

L	(«deber no hacer»)	=11
S	(«deber hacer»)	=14
P	(«tener el derecho de hacer»)	=13
W	(«tener el derecho de no hacer»)	=15
M	(«poder hacer y no hacer»)	=12
V	(«no tener el derecho de no hacer o no tener el derecho de hacer»)	=16

Tomemos como ejemplo la fórmula consistente en la coimplicación de T.48 y T.55,

E E Mxα K NLxα NSxα EMxα K Pxα PxNα

en el programa, esta fórmula es tratada como el vector numérico:

8, 8, 12, 2, 6, 3, 11, 2, 3, 14, 2, 8, 12, 2, 6, 13, 2, 13, 3, 2

La variable nominal individual «x» se ignora durante el proceso, ya que se trata de la variable que simboliza a un sujeto concreto de acción y no se le asignan valores. No obstante, en la impresión de resultados sí que aparece esta variable.

«α» simboliza el nombre de la acción y toma los valores 1*, 1/2* y 0* (bondad, indiferencia y maldad, respectivamente). Si se antepone a «α» el negador «N» se crea el nombre de la acción opuesta. La matriz sería la siguiente:

α	$N\alpha$
1*	0*
1/2*	1/2*
0*	1*

Con el fin de conciliar la trivalencia de la acción con la bivalencia de las normas que no son más que verdaderas o falsas, se construyen las siguientes matrices:

α	$Sx\alpha$	$Lx\alpha$	$Px\alpha$	$Wx\alpha$	$Mx\alpha$	$Vx\alpha$
1*	1	0	1	0	0	1
1/2*	0	0	1	1	1	0
0*	0	1	0	1	0	1

Cualquier expresión bien formada en el sistema K1 puede probarse mediante un método cero-uno. Para el único axioma de K1 la prueba sería:

A1 $\quad CNPxN\alpha\, Px\alpha$

$$CNPxN\alpha\, Px\alpha = CNPxN1^*Px1^* = CNPx0^*\, Px1^* = CN01 = C11 = 1.$$

$$CNPxN\alpha\, Px\alpha = CNPxN1/2^*\, Px1/2^* = CNPx1/2^*\, Px1/2^* = CN11 = C01 = 1.$$

$$CNPxN\alpha\, Px\alpha = CNPxN0^*\, Px0^* = CNPx1^*\, Px0^* = CN10 = C00 = 1.$$

De esta forma la prueba de una expresión está formada por distintos vectores en los que van desapareciendo los distintos elementos que componen las subfórmulas y son sustituidos por el resultado de su evaluación. Esta es la idea que subyace a nuestro programa.

I. Proceso de computación

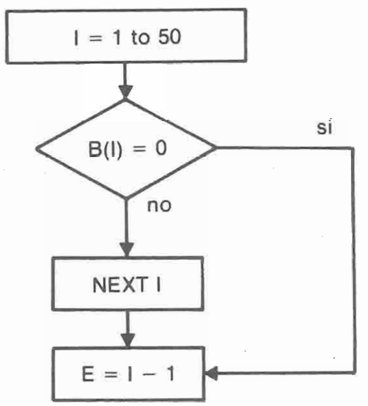
La lista ordenada de números que traduce la fórmula que queremos probar es introducida en un vector (B) mediante un enunciado INPUT. Antes de pasar a la asignación de valores y a la determinación del valor de verdad de las subfórmulas, reproducimos en otro vector (A) la fórmula a probar. Todas las operaciones se llevan a cabo en el vector A que vamos destruyendo conforme vamos calculando el valor de verdad de las subfórmulas, de forma que al final tenemos en la primera posición el valor de verdad de la fórmula para un determinado valor de α. Como tenemos la fórmula íntegra en el vector B, volvemos a reproducirla en el vector A y proseguimos el análisis con el siguiente valor de α.

Siguiendo con el ejemplo anterior, el vector A quedaría de la siguiente forma antes de la asignación de valores:

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22

8	8	12	2	6	3	11	2	3	14	2	8	12	2	6	13	2	1·	3	3	2	O	O

Los vectores están dimensionados de forma que el número de posiciones es superior al número de elementos que componen la fórmula, esto posibilita el que podamos utilizar el siguiente recurso para contarlos:



Establecemos un bucle que recorre el vector de izquierda a derecha, cuando encontramos un 0 esto nos indica que nos encontramos en la primera posición vacía del vector (en nuestro ejemplo la casilla 21) y que por tanto en la posición anterior se encuentra el último elemento de la fórmula, nos salimos del bucle y hacemos la variable E igual al valor de I menos 1.

Antes de pasar a la asignación de valores se establece el valor de α mediante una variable («Z») que toma inicialmente el valor 0. Posteriormente veremos cómo cambiamos este valor.

(Es importante hacer notar que durante el proceso iremos «borrando» aquellas casillas que no vamos a seguir necesitando, para ello les asignamos «− 1». Hemos elegido este valor porque no tenemos ningún otro valor negativo en el vector y esto nos facilita su manejo).

I.1. *Asignación de valores*

Se establece un bucle que va desde 1 hasta E mediante la variable I. Si A(I) es distinto de 2 se pasa al siguiente valor de I. Si A(I) es igual a 2 se hace

A(I) = Z, y se mira en la casilla anterior si hay *un negador* (3), en caso negativo pasamos al siguiente valor de I. En caso afirmativo lo «borramos» (hacemos A(I − 1) = − 1), y si Z = 1 hacemos A(I) = 0, si Z = 0, hacemos A(I) = 1. No consideramos la posibilidad de que Z sea igual a 0.5, ya que este valor no cambia con el negador. En cualquier caso siempre «borramos» el negador.

Cuando salimos del bucle determinado por I pasamos a la rutina de impresión.

El estado del vector que constituye nuestro ejemplo en este punto sería el siguiente:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
E	E	M	O	K	N	L	O	N	S	O	E	M	0	K	P	O	P		1	O	O	O

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
8	8	12	O	6	3	11	O	3	14	O	8	12	O	6	13	O	13	−1	1	O	O	O

Cuando se vuelve a la rutina general se pasa a la rutina de funtores deónticos.

I.2. *Evaluación de las proposiciones normativas*

Como hemos visto, el valor de una proposición normativa depende por un lado del funtor deóntico, y por otro del valor de α.

Si recorremos el vector de izquierda a derecha (desde I = 1 hasta E), cuando encontremos un A(I) mayor que 10 sabemos que el primer 0, 0.5, o 1 posterior que encontremos será el argumento del funtor deóntico que se encuentra en A(I).

Retenemos el funtor de que se trata (A(I)) en la variable «F», y el valor de α en la variable «G» y pasamos a la subrutina correspondiente para calcular el valor de la proposición normativa. Este valor lo asignamos a la posición anteriormente ocupada por el funtor y que hemos conservado en la variable «H».

La posición que ocupaba α fue «borrada» tras retener el valor en la variable «G».

Como puede haber un negador precediendo al funtor deóntico, miramos si en la posición A(H − 1) hay un negador. En caso afirmativo, lo «borramos» y pasamos a la rutina del negador para cambiar el valor de la proposición que se encuentra en A(H).

Pasamos al siguiente valor de I para seguir evaluando las proposiciones normativas. Cuando salimos del bucle determinado por I imprimimos.

Como se puede observar, una vez que se ha determinado el valor de una proposición normativa, sólo queda este valor en la casilla anteriormente ocupada por el funtor deóntico, los demás elementos han sido «borrados».

La situación que presentaría el vector de nuestro ejemplo tras este proceso sería la siguiente:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
E	E	O		K		O			1		E	O		K	O		1					

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
8	8	O	−1	6	−1	O	−1	−1	1	−1	8	O	−1	6	O	−1	1	−1	−1			

I.3. *Evaluación de las subfórmulas y de la fórmula general*

Como se puede observar, cuando entramos en esta fase del proceso lo que nos queda por evaluar son fórmulas de la lógica de enunciados.

No podemos empezar la localización de los juntores de izquierda a derecha porque dentro del alcance de los juntores que nos vamos encontrando se encuentran subfórmulas cuyo valor está por determinar.

Si empezamos buscando los juntores de derecha a izquierda sabemos que cuando nos encontramos un juntor, los dos primeros valores que se encuentran a su derecha son los valores de sus argumentos.

Así pues podemos evaluar esa expresión, conservando sólo su valor en la posición anteriormente ocupada por el juntor (salvo que éste vaya precedido de un negador; más adelante explicaremos este caso), y «borrando» las casillas anteriormente ocupadas por los valores de sus argumentos.

Al final del proceso tendremos el valor de la fórmula para el correspondiente valor de α en la primera casilla, y sólo nos quedará imprimirlo.

El proceso se desarrolla como sigue:

Establecemos un bucle que va desde E hasta 1 con un paso de -1 para recorrer el vector de derecha a izquierda.

Como sólo nos quedan en el vector juntores y el valor de sus argumentos, basta con que busquemos los números mayores de 4 para localizar los juntores.

Cuando encontramos el primer juntor lo conservamos en la variable «B», y su posición la retenemos en la variable «T».

El papel de la variable «N» es indicarnos si se trata del miembro izquierdo o derecho de la fórmula, cuando N = 0 se trata del miembro izquierdo, cuando N = 1 se trata del miembro derecho.

En este momento iniciamos un bucle que va desde la casilla siguiente al juntor (T + 1) hasta la última casilla ocupada por la fórmula («E») para buscar los valores que se encuentran a su derecha.

Cuando encontramos el primero, lo guardamos en la variable «P», y el segundo en la variable «Q», en ambos casos se «borra» la casilla que ocupaban.

Pasamos a la subrutina correspondiente que evalúa el valor de la fórmula y lo asigna a la posición anteriormente ocupada por el juntor ((A(T)) antes de volver a la rutina general.

Una vez aquí, se mira si en la casilla anterior (T − 1) hay un negador en cuyo caso pasamos a la rutina del negador para cambiar el valor. Este último valor se asigna a la casilla anteriormente ocupada por el negador, y se borra la que ocupaba el juntor. La razón de hacer esto es garantizar que al final del análisis el valor de la fórmula nos quede en la primera casilla aunque exista un negador al principio de la fórmula.

Una vez resuelta la posibilidad de que un negador preceda a un juntor, se pasa al bucle I, que es el que nos permite localizar los juntores de derecha a izquierda, y se continúa el proceso. Cuando hayamos recorrido todo el vector tendremos el valor de la fórmula para el correspondiente valor de α en la primera casilla A(1). Imprimimos este resultado.

Para saber si tenemos que seguir asignando valores a α o si hemos terminado, comprobamos el valor de «Z», si éste es igual a 1, hemos terminado, si no es así aumentamos el valor de «Z» en 0.5 y volvemos al principio de la rutina general.

Los distintos estados del vector de nuestro ejemplo a lo largo de este proceso serían los siguientes:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
E	E	O		K		O			1		E	O		O								

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
8	8	O	−1	6	−1	O	−1	−1	1	−1	8	O	−1	O	−1	−1	−1	−1	−1			

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
E	E	O		K		O			1		1											

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
8	8	O	−1	6	−1	O	−1	−1	1	−1	1	−1	−1	−1	−1	−1	−1	−1	−1			

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
E	E	O		O							1											

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
8	8	O	−1	O	−1	−1	−1	−1	−1	−1	1	−1	−1	−1	−1	−1	−1	−1	−1			

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
E	1											1										

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
8	1	−1	−1	−1	−1	−1	−1	−1	−1	−1	1	−1	−1	−1	−1	−1	−1	−1	−1			

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1																						

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1			

LISTADO DEL PROGRAMA

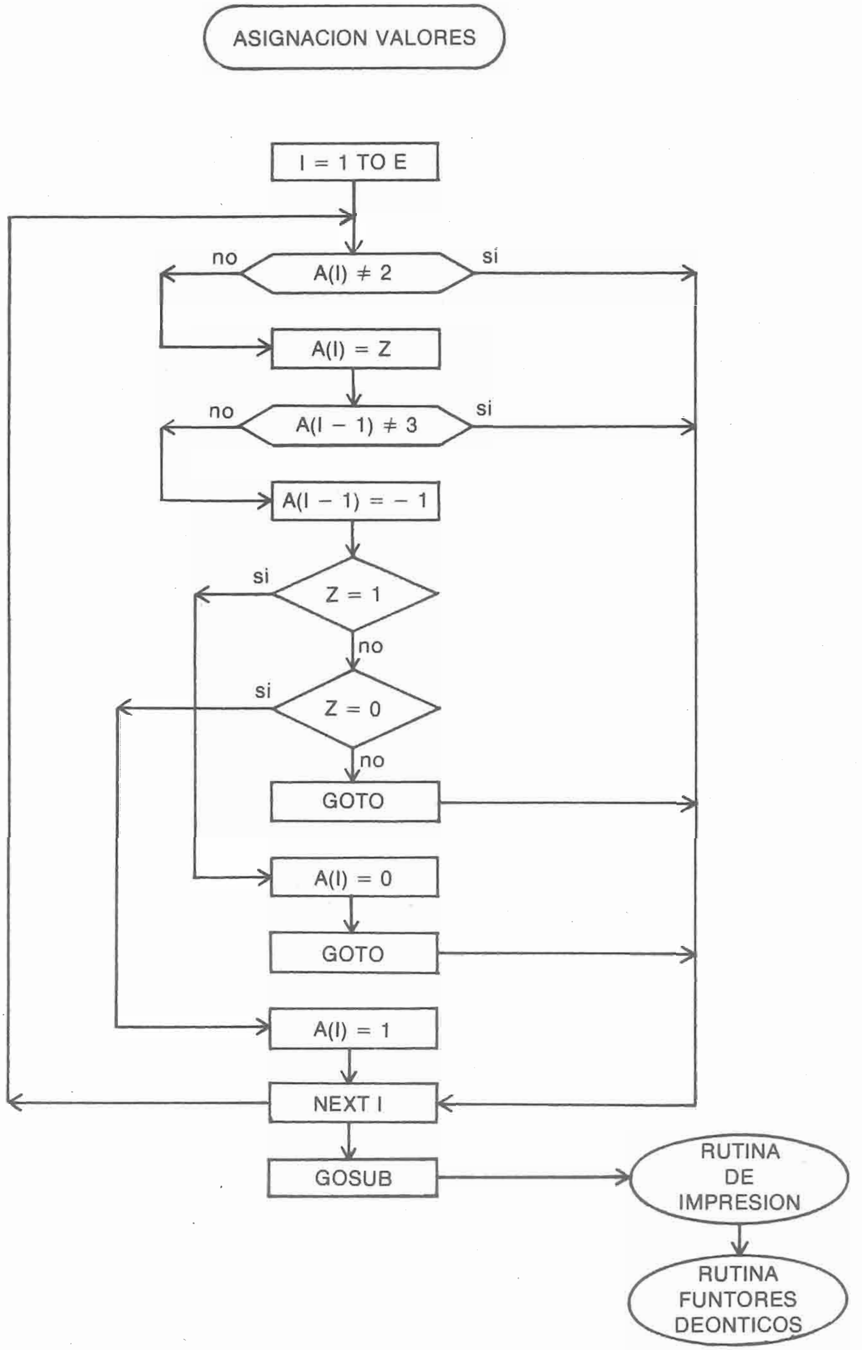
```
10   DIM B(50)
15   DIM A(50)
25   REM LECTURA DEL ESQUEMA
30   PRINT 'CUAL ES LA FORMULA?'
35   MAT INPUT B
40   FOR I = 1 TO 50
42   IF B(I) = 0 THEN 48
45   NEXT I
48   E = I - 1
50   LET Z = 0
55   MAT A = B
66   PRINT 'FORMULA A PROBAR:';
67   GOSUB 5000
70   REM ASIGNACION DE VALORES
75   FOR I = 1 TO E
80   IF A(I)<>2 THEN 130
85   LET A(I) = Z
90   IF A(I - 1)<>3 THEN 130
95   LET A(I - 1) = - 1
100  IF Z = 1 THEN 115
105  IF Z = 0 THEN 125
110  GOTO 130
115  LET A(I) = 0
120  GOTO 130
125  LET A(I) = 1
130  NEXT I
133  PRINT ''
135  GOSUB 5000
150  REM RUTINA FUNTORES DEONTICOS
155  FOR I = 1 TO E
160  IF A(I)<10 THEN 180
165  LET F = A(I)
170  LET H = I
175  GOTO 220
180  IF A(I) = 0 THEN 190
181  IF A(I) = 0.5 THEN 190
182  IF A(I) = 1 THEN 190
185  GOTO 220
190  LET G = A(I)
195  LET A(I) = - 1
200  GOSUB 3000
205  IF A(H - 1)<>3 THEN 220
210  LET A(H - 1) = - 1
215  GOSUB 3200
220  NEXT I
223  PRINT ''
225  GOSUB 5000
300  REM RUTINA JUNTORES DIADICOS
305  FOR I = E TO 1 STEP -1
310  IF A(I) < 4 THEN 391
315  LET B = A(I)
320  LET T = I
325  LET N = 0
330  FOR J = T + 1 TO E
```

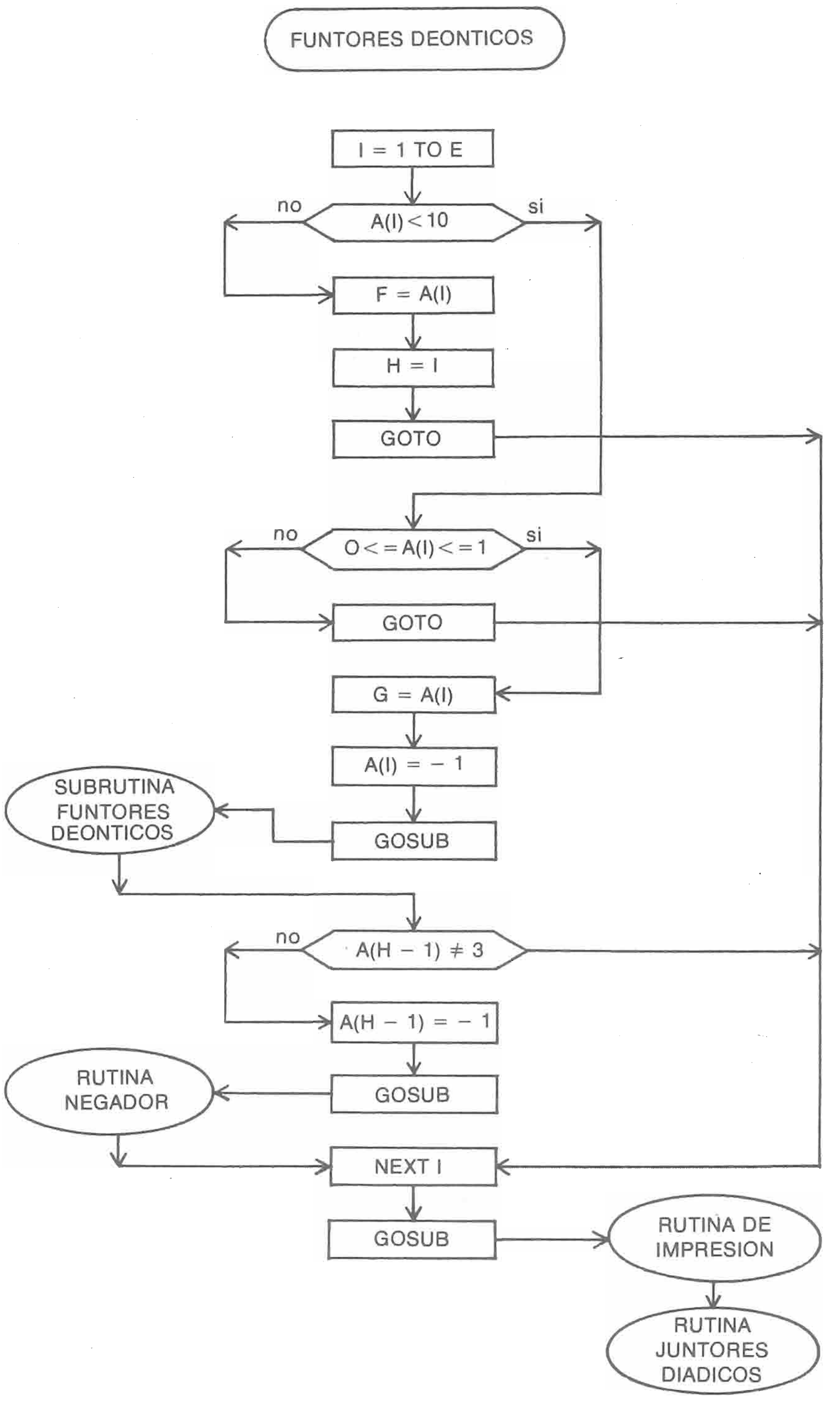
```
335   IF A(J) = -1 THEN 385
340   IF N = 1 THEN 365
345   LET P = A(J)
350   LET A(J) = -1
355   LET N = N + 1
360   GOTO 385
365   LET Q = A(J)
370   LET A(J) = -1
375   GOSUB 4000
380   GOTO 386
385   NEXT J
386   IF A(T - 1)<>3 THEN 391
387   LET H = T
388   GOSUB 3200
389   LET A(T - 1) = A(T)
390   LET A(T) = -1
391   NEXT I
394   PRINT ''
395   PRINT 'PARA @ =';Z;'FORM =';A(1)
400   IF Z = 1 THEN 5500
405   LET Z = Z + 0.5
410   GOTO 55
3000  REM SUBRUTINA FUNTORES DEONTICOS
3010  IF F<>11 THEN 3040
3015  IF G = 0 THEN 3025
3020  LET A(H) = 0
3022  RETURN
3025  LET A(H) = 1
3030  RETURN
3040  IF F<>12 THEN 3070
3045  IF G = 0.5 THEN 3060
3050  LET A(H) = 0
3055  RETURN
3060  LET A(H) = 1
3065  RETURN
3070  IF F<>13 THEN 3100
3075  IF G = 0 THEN 3090
3080  LET A(H) = 1
3085  RETURN
3090  LET A(H) = 0
3095  RETURN
3100  IF F<>14 THEN 3130
3105  IF G = 1 THEN 3120
3110  LET A(H) = 0
3115  RETURN
3120  LET A(H) = 1
3125  RETURN
3130  IF F<>15 THEN 3160
3135  IF G = 1 THEN 3150
3140  LET A(H) = 1
3145  RETURN
3150  LET A(H) = 0
3155  RETURN
3160  IF G = 0.5 THEN 3175
3165  LET A(H) = 1
3170  RETURN
```

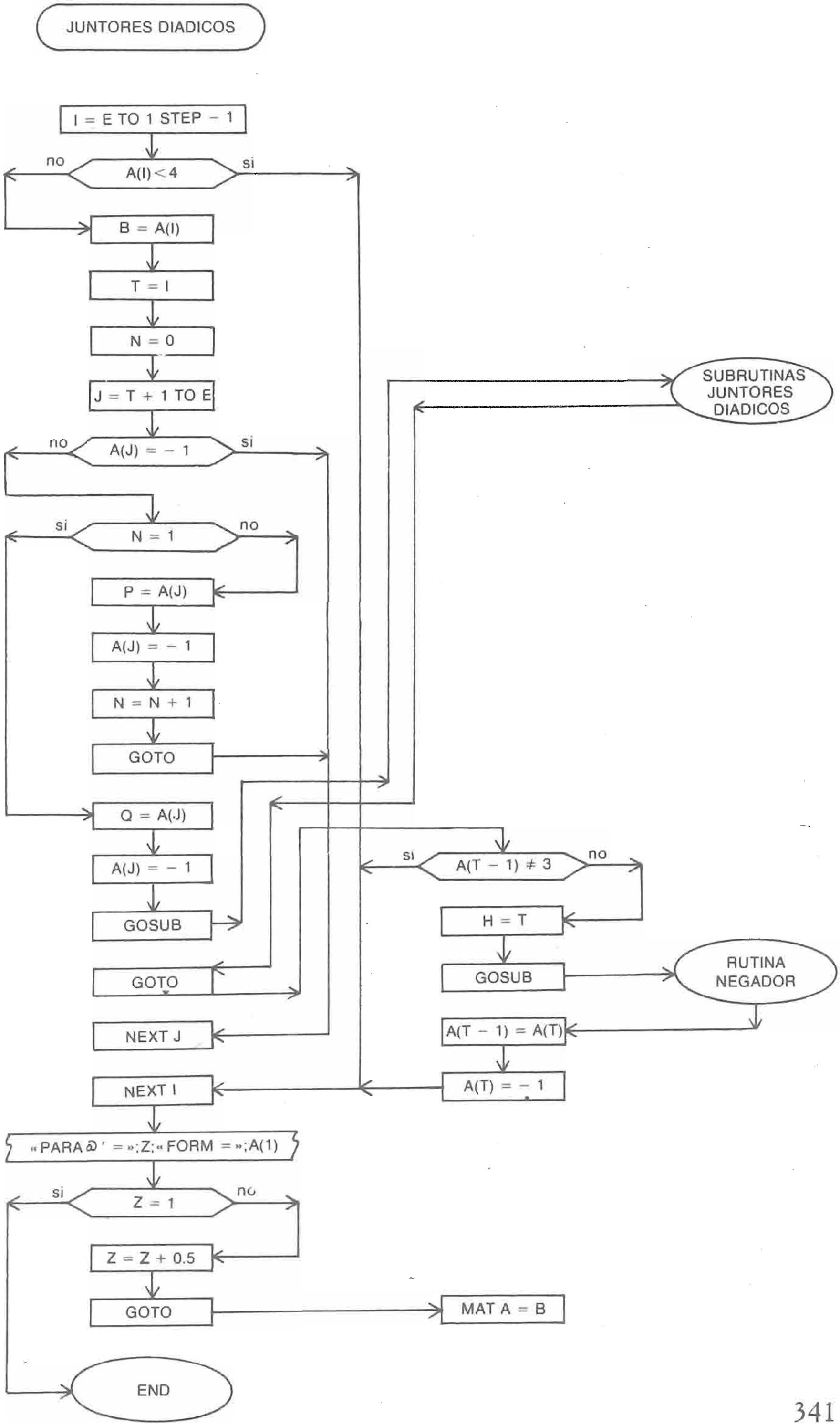
```
3175    LET A(H) = 0
3180    RETURN
3200    REM SUBRUTINA NEGADOR
3205    IF A(H) = 0 THEN 3220
3210    LET A(H) = 0
3215    RETURN
3220    LET A(H) = 1
3225    RETURN
4000    REM SUBRUTINA JUNTORES DIADICOS
4010    IF B<>4 THEN 4040
4015    IF P = 1 THEN 4030
4020    LET A(T) = 1
4025    RETURN
4030    LET A(T) = Q
4035    RETURN
4040    IF B<>5 THEN 4070
4045    IF P = 0 THEN 4060
4050    LET A(T) = 1
4055    RETURN
4060    LET A(T) = Q
4065    RETURN
4070    IF B<>6 THEN 4100
4075    IF P = 1 THEN 4090
4080    LET A(T) = 0
4085    RETURN
4090    LET A(T) = Q
4095    RETURN
4100    IF B<>7 THEN 4145
4105    IF P = 1 THEN 4120
4110    LET A(T) = 1
4115    RETURN
4120    IF Q = 0 THEN 4135
4125    LET A(T) = 0
4130    RETURN
4135    LET A(T) = 1
4140    RETURN
4145    IF P = 1 THEN 4175
4150    IF Q = 0 THEN 4165
4155    LET A(T) = 0
4160    RETURN
4165    LET A(T) = 1
4170    RETURN
4175    LET A(T) = Q
4180    RETURN
5000    REM RUTINA DE IMPRESION
5010    FOR K = 1 TO E
5015    IF A(K) = -1 THEN 5150
5020    IF A(K) = 2 THEN 5085
5025    IF A(K) = 3 THEN 5090
5030    IF A(K) = 4 THEN 5095
5035    IF A(K) = 5 THEN 5100
5040    IF A(K) = 6 THEN 5105
5045    IF A(K) = 7 THEN 5110
5050    IF A(K) = 8 THEN 5115
5055    IF A(K) = 11 THEN 5120
5060    IF A(K) = 12 THEN 5125
```

```
5065   IF A(K) = 13 THEN 5130
5070   IF A(K) = 14 THEN 5135
5075   IF A(K) = 15 THEN 5140
5080   IF A(K) = 16 THEN 5145
5082   PRINT A(K);
5083   GOTO 5150
5085   PRINT '@';
5087   GOTO 5150
5090   PRINT 'N';
5092   GOTO 5150
5095   PRINT 'C';
5097   GOTO 5150
5100   PRINT 'A';
5102   GOTO 5150
5105   PRINT 'K';
5107   GOTO 5150
5110   PRINT 'D';
5112   GOTO 5150
5115   PRINT 'E';
5117   GOTO 5150
5120   PRINT 'LX';
5122   GOTO 5150
5125   PRINT 'MX';
5127   GOTO 5150
5130   PRINT 'PX';
5132   GOTO 5150
5135   PRINT 'SX';
5137   GOTO 5150
5140   PRINT 'WX';
5142   GOTO 5150
5145   PRINT 'VX';
5150   NEXT K
5155   RETURN
5500   END
```

ORGANIGRAMAS




FUNTORES DEONTICOS
I = 1 TO E
no
A(I) < 10
si
F = A(I)
H = I
GOTO
no
O < = A(I) < = 1
si
GOTO
G = A(I)
A(I) = − 1
SUBRUTINA FUNTORES DEONTICOS
GOSUB
no
A(H − 1) ≠ 3
A(H − 1) = − 1
RUTINA NEGADOR
GOSUB
NEXT I
GOSUB
RUTINA DE IMPRESION
RUTINA JUNTORES DIADICOS


JUNTORES DIADICOS
I = E TO 1 STEP − 1
A(I) < 4
no
si
B = A(I)
T = I
N = 0
J = T + 1 TO E
A(J) = − 1
no
si
N = 1
si
no
P = A(J)
A(J) = − 1
N = N + 1
GOTO
Q = A(J)
A(J) = − 1
GOSUB
GOTO
NEXT J
NEXT I
SUBRUTINAS
JUNTORES
DIADICOS
A(T − 1) ≠ 3
si
no
H = T
GOSUB
RUTINA
NEGADOR
A(T − 1) = A(T)
A(T) = − 1
«PARA ' = »;Z;« FORM = »;A(1)
Z = 1
si
no
Z = Z + 0.5
GOTO
MAT A = B
END