

Prueba automática para las lógicas LE, K, T y S4

Antonio BENÍTEZ - Jaime SARABIA ALVAREZ-UDE
Universidad Complutense

1. INTRODUCCIÓN

Vamos a plantar en lo que sigue las líneas fundamentales de un método deductivo automático para los sistemas lógicos LE, K, T y S4. La idea fundamental del programa LOMSI (Lógica Modal Sin Imaginación) es la de simular los métodos de tablas analíticas o tablas semánticas. La extensión de los métodos de tablas analíticas a diferentes sistemas de lógicas modales y deónticas, realizada por Fitting (1983) es la base formal en que se asienta el programa que presentamos. Hemos procurado que la máquina trabajara siguiendo de cerca las operaciones de un deductor humano y que la salida del programa fuera fácilmente reconocible como una tabla analítica. LOMSI está definido en BASIC y ha sido probado —con éxito— en un IBM Personal Computer.

2. LAS LÓGICAS LE, K, T S4. CÁLCULOS

La lógica LE no es sino la lógica de enunciados bivalente standard. Esta lógica es un subconjunto de los otros sistemas que consideramos. Las teorías K, T y S4, sistemas de lógica modal, pueden caracterizarse semánticamente como sigue. Dados un conjunto M , no vacío, a cuyos miembros llamamos situaciones o mundos posibles, una relación binaria R sobre M —la relación de accesibilidad entre situaciones— y una relación binaria $\vdash M \times \text{Frm}l_a$, donde $\text{Frm}l_a$ es el conjunto de las fórmulas en una sintaxis apropiada, el tripló $[M, R, \vdash]$ es un modelo kripkeano si \vdash cumple las condiciones siguientes:

- 0) Para toda fórmula elemental A, $m \vdash A$ o no $m \vdash A$
- 1) $m \vdash \neg A$ sii no $m \vdash A$
- 2) $m \vdash (A \wedge B)$ sii $m \vdash A$ y $m \vdash B$
- 3) $m \vdash (A \vee B)$ sii $m \vdash A$ o $m \vdash B$
- 4) $m \vdash \Box A$ sii para todo $m' \in M$, mRm' -- $m' \vdash A$
- 5) $m \vdash \Diamond A$ sii para algún $m' \in M$, mRm' y $m' \vdash A$. (1)

Según las condiciones que se impongan sobre R obtendremos el tipo de modelos característicos de una u otra lógica modal. Así los modelos de la lógica K se caracterizan por el hecho de que R no está sometida a ninguna condición especial, mientras que en los de T, R es reflexiva, y en los de S4 es reflexiva y transitiva. Conocida la noción de modelo para cada teoría, podemos definir las correspondientes relaciones de consecuencia de la forma habitual y la teoría en términos de esa relación.

Fitting ha extendido la notación uniforme y el cálculo de tablas analíticas definidos por Smullyan (1968) de manera que sean aplicables a diversos sistemas modales y deónticos. En este tipo de cálculos se construye, dado un conjunto de fórmulas, un conjunto de secuencias de fórmulas que, con ciertas condiciones adicionales, constituyen lo que llamamos un árbol. La idea es que si el conjunto de fórmulas original es satisfacible al menos una de las secuencias —ramas— del árbol es satisfacible. El proceso es similar en lógica estándar y en lógica modal, salvo por una importante diferencia. De la satisfacibilidad de una fórmula precedida por un operador modal en una situación dada, no se sigue la satisfacibilidad de esa fórmula sin el operador en esa situación, al menos en ciertos casos. Posiblemente esa segunda fórmula sólo es satisfacible en una situación alternativa. Por así decir, al operar con ciertas fórmulas modales hemos de cambiar de situación. Eso se representa formalmente construyendo un árbol nuevo adaptado a la nueva situación. El problema es determinar qué nos podemos llevar a la nueva situación. Fitting introduce, junto con reglas de deducción similares a las habituales unas «reglas de modificación de rama» que permiten el paso de un árbol dado a otro nuevo indicando qué información de la contenida en el árbol antiguo es válida en el nuevo. En síntesis las reglas de deducción y de modificación de rama son las siguientes:

K:	$\frac{FA}{A} \quad (*)$		$\frac{\neg \Box A}{\neg A} \quad (**)$
	(*) Aplicando antes la primera regla de modificación de rama.		
T:			
1)	$\frac{\Box A}{A}$		$\frac{\neg \Diamond A}{\neg A}$
2)	$\frac{\Diamond A}{A} \quad (**)$		$\frac{\Box A}{A} \quad (**)$

(1) El modo de exponer la semántica kripkeana para estos sistemas de lógica modal así como las reglas de deducción para K, T y S4 están tomadas del capítulo 2 de Fitting (1983).

S4:

1)	$\frac{NA}{A}$		$\frac{-FA}{-A}$	
2)	$\frac{FA}{A} \quad (*)$		$\frac{-NA}{-A} \quad (*)$	

(*) Aplicando antes la segunda regla de modificación de rama?

Nuestro programa de cálculo automático realiza deducciones según las reglas de cada uno de estos cálculos.

3. EL PROGRAMA.

Como input general el programa toma una fórmula, con operadores modales o sin ellos, y previa indicación de la lógica en que se va a realizar la deducción, ofrece como output el árbol o tabla analítica que constituye la deducción en esa lógica para esa fórmula, así como un diagnóstico respecto a si la fórmula es satisfacible o no lo es y una indicación respecto a qué fórmulas se han eliminado en cada rama siguiendo las instrucciones de las correspondientes reglas de modificación de rama. El programa es notablemente independiente del tipo de notación empleada, aunque hemos utilizado la siguiente: operadores monádicos: \neg (no), N (necesario), P (posible). Operadores diádicos: $\&$ (y), \vee (o). Variables proposicionales: p, q, r, s, t. Paréntesis: (,). Se supone una notación prefija de los operadores monádicos e infija de los diádicos, ateniéndose a las siguientes convenciones sobre paréntesis: toda fórmula va rodeada de paréntesis, salvo las fórmulas de grado cero (variables proposicionales). Además, los operadores monádicos ligan más fuertemente que los diádicos. Como vemos, la entrada es una fórmula en su grafía habitual. El proceso posterior es el siguiente: una vez seleccionada una fórmula de una rama de la tabla analítica, se generan el (los) descendiente(s) de esa fórmula. Si es necesario, se modifica la rama correspondiente. A continuación se prolonga la rama con el descendiente así obtenido o se ramifica añadiendo descendientes distintos a la rama antigua y a la nueva. Se comprueba si la rama está cerrada y si lo está, si el árbol está cerrado. El proceso continúa con una nueva selección de fórmulas hasta que o bien todas las ramas están cerradas o bien no hay posibilidad de aplicar ninguna regla a ninguna de las ramas (con una excepción que comentaremos posteriormente). El diagrama I, en las páginas siguientes, representa esquemáticamente este proceso.

Pasamos a comentar más pormenorizadamente alguno de estos pasos.

(2) Primera regla de modificación de rama: Sustituir NA por A, sustituir $\neg PA$ por $\neg A$ y borrar todas las demás fórmulas.

Segunda regla de modificación de rama: Borrar todas las fórmulas excepto las de forma NA y $\neg PA$.

3.1. Entrada de la fórmula a analizar y selección de fórmulas

El programa comienza leyendo los símbolos en ficheros secuenciales previamente definidos (líneas 10-180 del programa). En la realización concreta del programa hemos trabajado con las cuatro categorías signicas a que antes hemos aludido, correspondiéndoles un fichero a cada una. La grafía concreta de los signos es irrelevante en este punto. Multiplicando el número de categorías signicas se podría haber hecho el programa totalmente independiente del grafismo, pero no lo hemos considerado necesario, una vez lograda la realización en principio de esa idea. Los signos de cada categoría se incluyen en una lista *ad hoc*.

Dos vectores tridimensionales cumplen una función esencial en el programa. De uno de ellos, FRMLA\$(x, y, z), hablaremos más por extenso al tratar del analizador de fórmulas. El segundo, ARBOL\$(w, r, i), es la representación en la máquina de la tabla analítica. La variable r representa el número de rama de la tabla analítica e i representa el puesto de una fórmula en la rama en cuestión. w, entre 0 y 2, es un indicador que remite a datos relevantes de la fórmula contenida en ARBOL\$(0, r, i). La utilización del 0 como valor de las variables r e i permite incluir información sobre el árbol, o sobre ramas enteras. ARBOL\$ incluye la información siguiente:

```
ARBOL$(0,r,0) : la rama r está abierta/cerrada
ARBOL$(0,r,i) : la fórmula de índice o nivel i de la rama r
ARBOL$(1,r,0) : en su caso, la rama r ramifica a partir de la
                 rama r' en el nivel i'
ARBOL$(1,r,i) : la fórmula de nivel i de la rama r está usada/
                 no lo está / está tachada (por efecto de una
                 regla de modificación de rama) / no lo está.
ARBOL$(2,r,0) : longitud de la rama r.
ARBOL$(2,r,n) : conectiva de la fórmula en rama r y nivel i
                 y lugar de su aparición.
```

La fórmula de origen pasa directamente al mecanismo que hemos llamado analizador y que estudiaremos más abajo. Nos interesa ahora dar una explicación mínima del proceso de selección de fórmulas a analizar. Para que el programa sea efectivo hay que contar con un mecanismo que seleccione al menos una vez cada una de las fórmulas de cada rama de la tabla analítica y que lo haga de manera eficaz. Si bien en LE el mecanismo de selección no repercute más que en la extensión de la deducción, en los sistemas modales es vital que en cada ocasión se seleccione la fórmula adecuada, so pena de que haya fallos en el diagnóstico de la satisfacibilidad de las fórmulas. Así, por ejemplo, si en una deducción en T aparecen las fórmulas

$$\neg N(p \& q)$$

$$(Np \& Nq)$$

y se opera en T con la primera fórmula antes que con la segunda, la regla de modificación de rama elimina la segunda fórmula, impidiendo que la máquina genere una rama cerrada, con el consiguiente error de diagnóstico. El mecanismo que hemos desarrollado (1000-2000) para seleccionar fórmulas opera en los siguientes pasos: a) selecciona la primera a rama que no esté cerrada y que sea posterior en el árbol a la que ha estado considerando, si la hay. Si no, selecciona la primera rama. b) Selecciona la última fórmula de entre las de menor tipo de esa rama que no haya sido usada, que no esté tachada y que no sea de tipo 0. Los tipos son: 0, variables proposicionales negadas y no negadas; 1, dobles negaciones; 2 y 3, fórmulas conjuntivas precedidas o no de negación, respectivamente; 4 y 5, fórmulas disyuntivas precedidas o no de negación; 6 y 7, fórmulas de estructura \neg NA y PA; 8 y 9, fórmulas de la forma \neg PA y NA. A la vez el mecanismo controla si las ramas están cerradas y si no es posible aplicar alguna regla de deducción sobre ellas. Si ocurre uno de estos casos en todas las ramas, se da por terminado el proceso y se pasa a la representación en pantalla del árbol. Esta parte del programa queda representada en el diagrama II. Hay dos salidas posibles del mecanismo que consideramos. O bien el programa decide acabar la deducción porque todas las ramas están cerradas o no se puede aplicar ninguna regla de deducción más —pasando entonces al dispositivo impresor— o bien se ha seleccionado una fórmula y el programa continúa en el subsistema siguiente, de construcción de descendientes, que pasamos a analizar.

3.2. Construcción de descendientes y ampliación del árbol

El proceso corresponde a las líneas 2000-2410 del listado del programa. Dada la fórmula seleccionada B se construyen su(s) descendiente(s) siguiendo las reglas características del cálculo de árboles. En general, los descendientes de B son subfórmulas de B precedidas, en ciertos casos, de una negación. En este punto hay que atender, dado el tipo de cálculos que consideramos, a las reglas de modificación de rama y a que, en el caso de K, ciertas fórmulas —NA y 9 PA— no generan descendientes en el árbol. Una vez construidos el o los descendientes de B, se envían al analizador para posteriormente ser incorporados al vector ARBOL\$ previa una asignación de tipo. El añadir los descendientes de B a ARBOL\$ supone, naturalmente, tener en cuenta si la regla de deducción empleada sobre B produce ramificación o no. Según sea el caso, los descendientes de B se incorporan a ARBOL\$ siguiendo uno u otro (extensión simple de rama o ramificación) de los submecanismos de la rutina de extensión de rama. En cada extensión de una rama por medio de las rutinas de extensión se comprueba si la nueva rama está cerrada. Una vez realizado todo este proceso, tras comprobar si el árbol en su conjunto está cerrado, se pasa a una nueva selección de fórmula. El diagrama de flujo de esta parte del

programa (diagrama III) puede informar sobre la concepción general del mismo.

3.3. El analizador de fórmulas

La entrada de esta subrutina (líneas 3000-3610) es una fórmula A y la salida un análisis de las apariciones de subfórmulas de esa fórmula con indicación de su conectiva, si la hubiere, así como los lugares de la fórmula donde comienza y termina la subfórmula. El proceso es recursivo y podríamos definirlo como sigue:

- a) Las fórmulas de grado 0 en A son subfórmulas de A.
- b) D es subfórmula de A si hay una(s) subfórmula(s) B (B y C) y un operador monádico op1 (o un operador diádico op2) tales que op1-(B) es D [(B-op2-C) es D] y A es de la forma —D—;

donde '—' indica la concatenación y '—' son contextos posiblemente vacíos. El primer paso de la subrutina es leer letra por letra la expresión y atribuir categorías a cada uno de los signos. A continuación se produce la búsqueda de subfórmulas situándolas en el vector FRMLA\$. En primer lugar se sitúan en ese vector las fórmulas de grado cero. Después, y dado un conjunto de apariciones de las subfórmulas se estudia si cumplen la condición b) antedicha. El proceso se repite un número de veces igual al grado de la fórmula. La última subfórmula obtenida debe coincidir con la fórmula A. Si esto no ocurre la máquina emite un aviso de error indicando que la expresión no es una fórmula. El vector FRMLA\$ (n, y, z) almacena los datos obtenidos del modo siguiente:

```
FRMLA$(n,0,0) : la subfórmula n en el vector
FRMLA$(n,0,1) : comienzo de la subfórmula n en la fórmula dada
FRMLA$(n,1,0) : fin de la subfórmula n en la fórmula dada
FRMLA$(n,1,1) : conectiva de la subfórmula n / lugar de la
                  conectiva.
```

A lo largo del proceso deductivo se indica, con cada fórmula en el árbol, el lugar del análisis de la fórmula en FRMLA\$ de modo que se pueda rescatar la información allí contenida cuando se necesite. Las siguientes líneas son un ejemplo de la salida del analizador:

orden en FRMLA\$		comienzo	fin	operador	lugar	conect
formula:	N(¬Npvp)					
frmla 0	p	5	5	-	5	
frmla 1	p	7	7	-	7	
frmla 2	Np	4	5			N 4
frmla 3	¬Np	3	5			¬ 3
frmla 4	(¬Npvp)	2	8			∨ 6
frmla 5	N(¬Npvp)	1	8			N 1

3.4. Subrutinas de cierre, asignación de tipo y extensión

Estas subrutinas aparecen en las líneas 4000-4240, 5000-5150 y 6000-6690 del listado. En la primera de ellas dada una rama, se comparan entre sí todas sus fórmulas no tachadas con el fin de ver si hay una que sea negación de otra. Si esto ocurre, se indica en el lugar correspondiente si la rama está cerrada. Según el lugar del programa desde el que se acceda a la subrutina, este proceso se realiza en una sola rama o en todas las ramas. Si la máquina encuentra que todas las ramas están cerradas pasa al programa impresor. En otro caso vuelve al punto de envío.

En la subrutina de asignación de tipo, se localiza primero de todo la conectiva principal de la fórmula en FRMLA\$ y, si la fórmula es una negación, cuál es la conectiva principal de la fórmula negada y sobre esa base se asigna un tipo a la fórmula, según los principios a que aludimos en el punto 3.2.

La tercera de las subrutinas que comentamos tiene dos partes a las que hemos llamado a) extensión simple (6000-6080) y b) ramificación doble (6500-6690). Cuando se utiliza la segunda subrutina se ha utilizado también la primera, pero a) puede funcionar en solitario. En ambos casos la entrada es una rama y una fórmula y la salida es el resultado de o bien -a)- añadir una nueva fórmula a la rama o bien -(b)- la generación de una nueva rama que se diferencia de la dada sólo en la última fórmula. Es decir, se produce una ramificación en el árbol. En ambos casos se añade a RAMA\$ la información a que aludimos más arriba.

3.5. Rutina de modificación de rama. Representación en pantalla

La versión maquinaal de las reglas de modificación de rama que expusimos en el apartado 2 se recoge en las líneas 8000-8180 del listado. La máquina lee las fórmulas de la rama y mientras no encuentra fórmulas de la forma $\neg PA$ o NA marca esas fórmulas como tachadas. En el caso en que la fórmula tiene una de las estructuras indicadas o (S4) indica que la fórmula puede volver a usarse o (K, T) tras tacharla, añade a la rama que esté considerando el descendiente que la regla de modificación de rama indique. Esto supone, de nuevo, un análisis de esas fórmulas, asignación de tipo y uso de una de las rutinas de extensión. En el diagrama IV se indica cómo transcurre el proceso.

El aparato impresor, si bien no es esencial para la realización de la deducción, puesto que la representación interna del árbol permite llegar a un diagnóstico acerca de la satisfacibilidad de la fórmula, es importante sin embargo para el usuario humano permitiéndole, por ejemplo, descubrir algunos hechos sobre los sistemas en que trabaja la máquina o emitir hipótesis sobre la estructura de fórmulas que podrían ser leyes lógicas en cada uno de los sistemas. El paso, sin embargo, de la representación

del árbol en ARBOL\$ como un conjunto —estrictamente, una secuencia— de secuencias de fórmulas a la representación clásica de una tabla analítica ha requerido más pasos de programa de los que se hubiera pensado en principio. En dos palabras, el proceso sería primero, estudiar las cadenas de ramificaciones. Aludimos con esto a situaciones en que una rama, digamos la primera, ramifica con otra, la sexta, por ejemplo, que a su vez ramifica con otra, etc. A partir de esto, se construye una serie que indica el orden en que han estudiarse las ramas, como sigue: dada la serie de ramas con n elementos, el elemento $n+1$ se obtiene, como sigue:

- a) Considera el elemento n de la serie, sea la rama r .
- b) Si r no ramifica, define el incremento, y elimina a r de la serie.
- c) Si r ramifica, considera la última de las ramas, r' , con origen en r ; añade r' a la serie.
- d) Vuelve a a).

El incremento es la medida del corrimiento a la derecha que han de sufrir las fórmulas de ramas posteriores a r en la representación. Una vez hecho esto, se asignan lugares en la pantalla a cada una de las fórmulas de modo que partiendo de una indicación del usuario sobre la columna en que debe empezar a escribirse, las ramas y fórmulas que deben aparecer más a la derecha lo hagan correctamente. Una vez situadas las fórmulas se pasa a presentarlas en la pantalla. Finalmente, la máquina indica mediante un asterisco si la rama está cerrada, el número de la rama y además qué fórmulas están tachadas y en qué rama lo están.

4. CONCLUSIÓN

Presentamos primero algún ejemplo de los árboles generados por LOMSI. Una salida típica es, en un caso sencillo, la siguiente:

[] en 1 2	[$\neg (\neg (Np \& Nq) \vee N(p \& q))$]
[] en 1 2	[$\neg \neg (Np \& Nq)$]
[] en 1 2	[$\neg N(p \& q)$]
[] en 1 2	[$\neg (Np \& Nq)$]
	Np
	Nq
	$\neg (p \& q)$
	$\neg p$ $\neg q$
	q q
	p $*$
	$*$ 2
	1

la fórmula es insatisfacible en S4

Los símbolos [,] indican que la fórmula está tachada. Los números a la izquierda indican en qué rama o ramas lo están. La deducción de la misma fórmula en T es totalmente similar salvo que las líneas quinta y sexta tienen la forma

[Np]
[Nq]

Esto es debido, evidentemente, a la distinta regla de modificación de rama. Comparamos ahora una fórmula típicamente distintiva entre S4 y T:

[] en 1 [¬(¬Np∨NNp)]
[] en 1 [¬¬Np]
[] en 1 [¬NNp]
 Np
 ¬Np
 *
 !

La fórmula es insatisfecha en S4

[] en 1 [¬(¬Np∨NNp)]
[] en 1 [¬¬Np]
[] en 1 [¬NNp]
[] en 1 [Np]
[] en 1 [p]
[] en 1 [¬Np]
 ¬p
 !

La fórmula es satisfacible en T

La diferencia entre ambas deducciones estriba en el modo en que operan las reglas de modificación de rama. Mientras que en S4 al añadir $\neg Np$, por una aplicación de una regla sobre $\neg NNp$, no se tacha Np , no ocurre lo mismo en T. En este caso se ha tachado Np y se ha añadido $\neg Np$. Al aplicar una regla sobre esta última fórmula, se ha tachado p y el árbol queda abierto.

Una observación final. El sistema deductivo que estamos empleando no es un método de decisión para S4. Debido a la estructura de la regla de modificación de rama puede ocurrir que se genere un árbol infinito en el que siempre haya una fórmula por analizar. Ante una fórmula de este tipo el autómata LOMSI sigue funcionando en tanto se lo permita la limitación de memoria de la máquina calculadora. Este hecho se debe a que el sistema deductivo que empleamos para S4 trata las modalidades N y $\neg P$ de modo muy análogo al tratamiento de los cuantores en lógica estándar. Debido a la transitividad de la relación de accesibilidad en S4, la cuantificación sobre mundos posibles en S4 no está restringida y la función que el sistema constituye no es recursiva. Sin embargo, dado que existe una prueba de suficiencia para el sistema deductivo que utilizamos sabemos que el conjunto de las fórmulas insatisfactibles, y con él, el de las leyes lógicas en S4 si es recursivo. Un ejemplo de fórmula que genera un bucle infinito sería $N(\neg Npvp)$. Damos a continuación las primeras líneas del árbol generado por LOMSI.

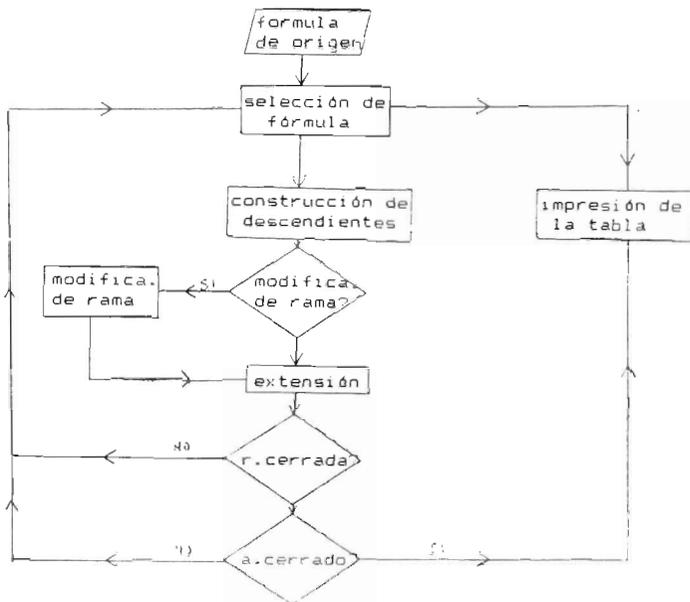
[] en 1	3	4	5	6	N(¬Npvp)		
[] en 1	3	4	5	6	[(¬Npvp)]		
[] en 1	3	4	5	6	[¬Np]		
[] en 1		4	5	6	[¬p]		2
[] en 1		4	5	6	[(¬Npvp)]		
[] en 1			5	6	[¬Np]		p
[] en 1			5	6	[¬p]		*
[] en 1			5	6	[(¬Npvp)]		3
[] en 1				6	[¬Np]		p
[] en 1				6	[¬p]		*
[] en 1				6	[(¬Npvp)]		4
[] en 1					[¬Np]	p	
[] en 1					[¬p]	*	
[] en 1					[(¬Npvp)]	5	
[] en 1					[¬Np]	p	
[] en 1					¬p	*	
[] en 1					(¬Npvp)	6	

Incluimos a continuación los diagramas a que hemos aludido así como un listado del programa.

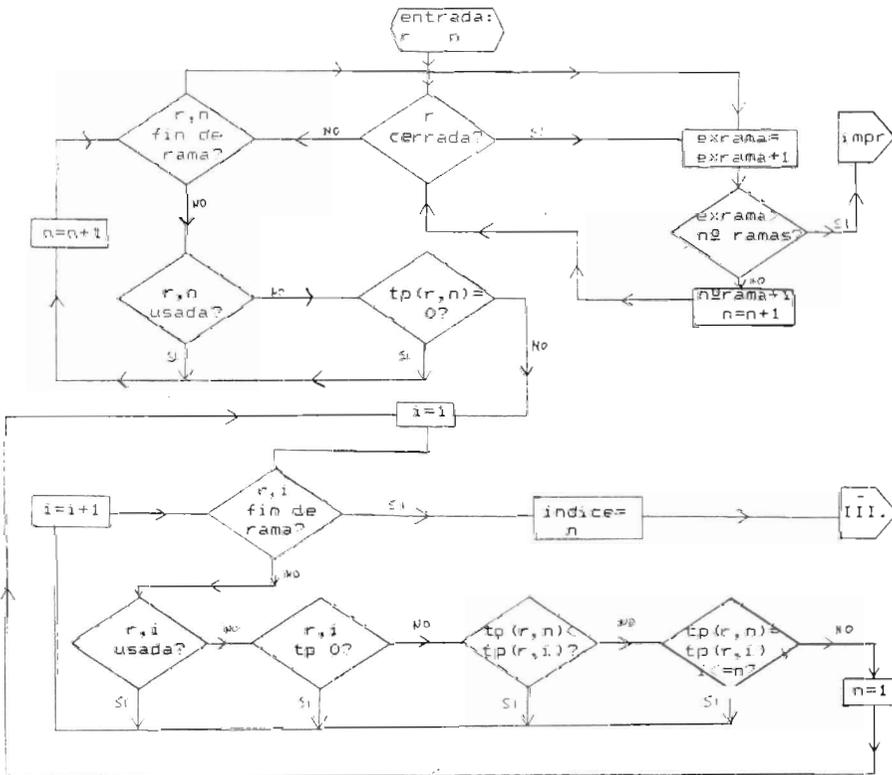
En los diagramas que siguen empleamos las siguientes indicaciones:

- imp.: impresora.
- r, r': ramas.
- n, i, in: lugares en una rama.
- tp (r, n): tipo de la fórmula en árbol\$ (0, r, n).
- r, i usada, tachada: la fórmula correspondiente está usada o tachada.
- extensión, ramificación, análisis, etc.: llamadas a las subrutinas correspondientes.
- sistema: sistema de cálculo.
- ct1, ct2: El primer, primer y segundo, signos de una fórmula.
- pre. fórmula\$: proceso previo a la definición de un descendiente.
- fórmula\$: proceso de definición de un descendiente, previo al análisis del mismo.
- índice y rama son variables usadas por LOMSI como marcadores del número de ramas cerradas o analizadas y de la profundidad de la fórmula en que se opera en cada momento, respectivamente.

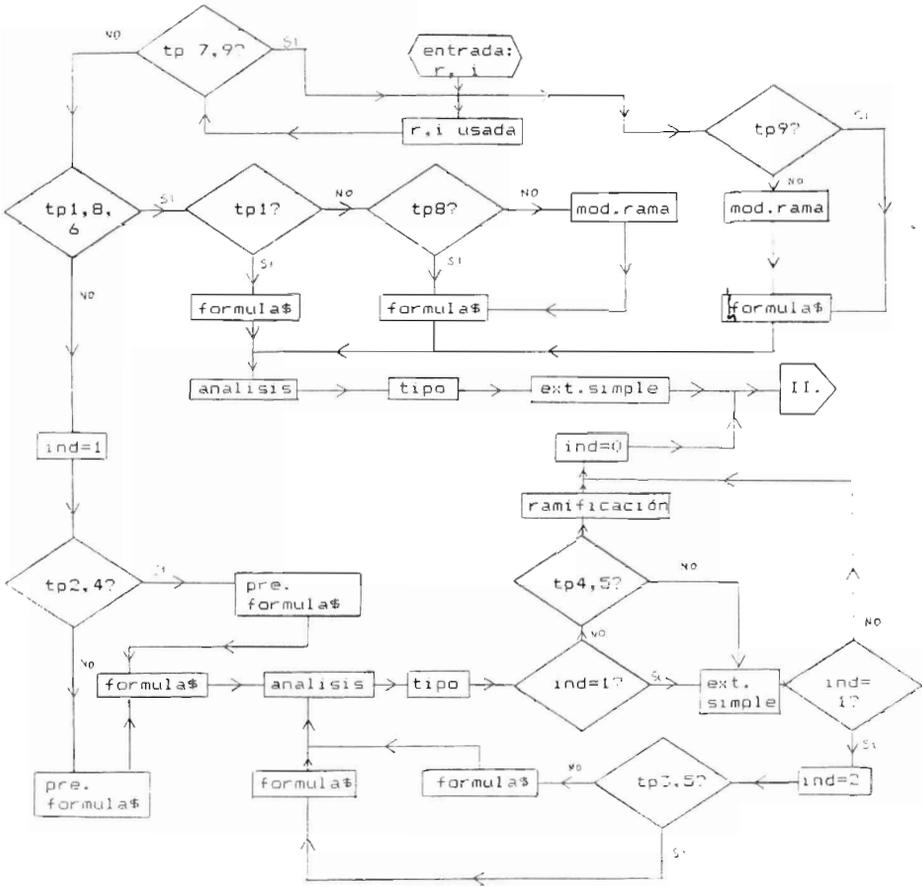
I. DIAGRAMA GENERAL



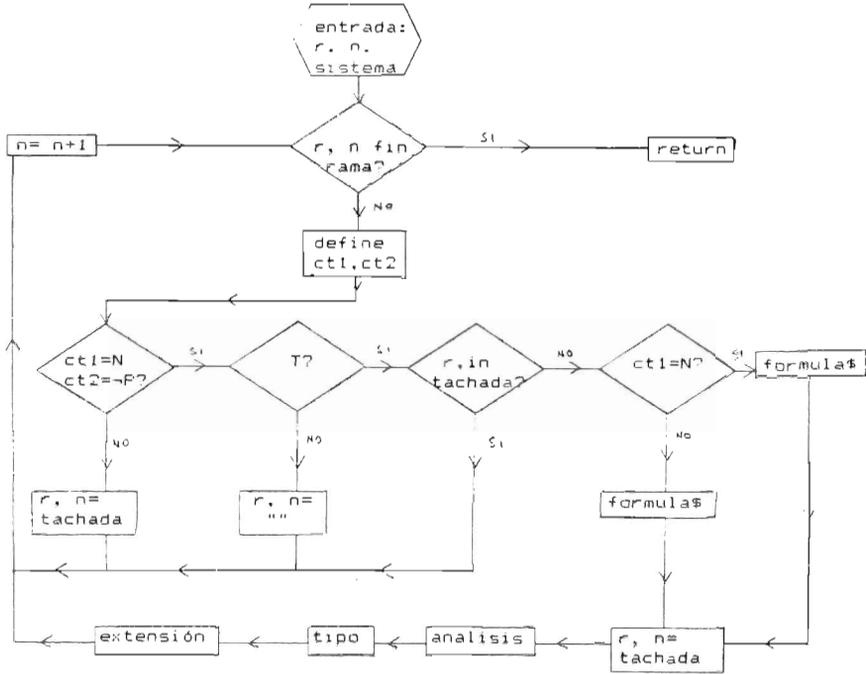
II. SELECCION DE FORMULA A ANALIZAR



III. CONSTRUCCION DE DESCENDIENTES



IV. MODIFICACION DE RAMAS



LISTADO

```

10 KEY OFF:CLS
20 INPUT "¿sistema LE, K, T o S4? ", SC$
30 INPUT "empiezo a imprimir en la columna número ", NRC
40 NPRED=4
50 PREDICADO$(1)="var"
60 PREDICADO$(2)="opmona"
70 PREDICADO$(3)="opdia"
80 PREDICADO$(4)="puntuo"
90 DIM PREDICADO1$(100):DIM PREDICADO2(100):DIM FRMLA$(250,1,1)
100 FOR MB=1 TO NPRED
110 OPEN PREDICADO$(MB) FOR INPUT AS # MB
120 IF EOF(MB) THEN 180
130 INPUT#MB,OPRED$
140 PREDICADO1$(E)=OPRED$
150 PREDICADO2(E)=MB
160 E=E+1
170 GOTO 120
180 LISTAPRED(MB)=E-LISTAPRED(MB-1)
190 NEXT MB
500 INPUT "formula: ",FORMULA$ ENTRADA DE LA FORMULA ORIGEN
510 GOSUB 3000
520 DEF FNIZDA$(A$,N)=LEFT$(A$,LEN(A$)-N)
540 DEF FNDCHA$(A$,N)=RIGHT$(A$, (LEN(A$)+1)-N)
550 DIM INBUSCA(20,50):DIM ARBOL$(2,20,50):DIM TIPO(20,50)
560 INBUSCA(1,1)=8:RAMIFICACION=1
    
```

```

570 GOSUB 5000
580 ARBOL$(0,1,0)="abierta"
590 ARBOL$(0,1,1)=FRMLA$(B,0,0)
600 ARBOL$(1,1,1)=""
605 ARBOL$(2,1,0)=STR$(2)
610 ARBOL$(2,1,1)=FRMLA$(B,1,1)
620 TIPO(1,1)=TIPO:TIPO=0:INDICE=1:RAMA=1
630 GOTO 2000
1000 RIMA=RAMA:GOSUB 4000' SELECCION DE FORMULAS A ANALIZAR
1010 EXTRAMA=0:INDICE=0
1020 NIVEL=1:RAMA=RAMA+1
1030 IF RAMA > RAMIFICACION THEN RAMA=1:EXTRAMA=0
1040 IF ARBOL$(0, RAMA, 0)="cerrada" OR ARBOL$(0, RAMA, NIVEL)="" THEN 1090
1050 IF ARBOL$(1, RAMA, NIVEL)="usada" OR ARBOL$(1, RAMA, NIVEL)="tachada"
OR TIPO(RAMA, NIVEL)=0 THEN 1070
1060 GOTO 1140
1070 NIVEL=NIVEL+1
1080 GOTO 1040
1090 EXTRAMA=EXTRAMA+1
1100 IF EXTRAMA = RAMIFICACION THEN GOTO 1120
1110 GOTO 1020
1120 GOSUB 4140
1130 GOTO 7000
1140 I=1
1150 WHILE NOT ARBOL$(0, RAMA, I)=""
1160 IF ARBOL$(1, RAMA, I)="tachada" OR ARBOL$(1, RAMA, I)="usada" THEN 1210
1170 IF TIPO(RAMA, I)=0 THEN 1210
1180 IF TIPO(RAMA, NIVEL) < TIPO(RAMA, I) THEN 1210
1190 IF TIPO(RAMA, NIVEL)=TIPO(RAMA, I) AND I<=NIVEL THEN 1210
1200 NIVEL=I: I=1:GOTO 1220
1210 I=I+1
1220 WEND
1230 INDICE=NIVEL
2000 CLS:ARBOL$(1, RAMA, INDICE)="usada" CONSTRUCCION DE SUBFORMULAS
2010 MAC=INBUSCA(RAMA, INDICE)-1
2020 IF TIPO(RAMA, INDICE)=7 OR TIPO(RAMA, INDICE)=9 THEN 2320
2030 IF TIPO(RAMA, INDICE)=1 OR TIPO(RAMA, INDICE)=8 OR
TIPO(RAMA, INDICE)=6 THEN 2260
2040 INDICADOR=1
2050 IF TIPO(RAMA, INDICE)=2 OR TIPO(RAMA, INDICE)=4 THEN 2070
2060 NLS1#=FRMLA$(MAC, 1, 1):NLS2#=FRMLA$(MAC, 0, 0):H=3:GOTO 2080
2070 NLS1#=ARBOL$(2, RAMA, INDICE):NLS2#=ARBOL$(0, RAMA, INDICE):H=2
2080 LOCCON=VAL(FNDCHA$(NLS1$, H))' Preparación
2090 FLA$(0)=FNDCHA$(NLS2$, 2)
2100 FLA$(1)=FNDCHA$(FLA$(0), 1)
2110 FLA$(2)=LEFT$(FLA$(1), LOCCON-H)
2120 IF TIPO(RAMA, INDICE)=5 OR TIPO(RAMA, INDICE)=3 THEN FLA$(3)=
FNDCHA$(FLA$(1), LOCCON-1):FORMULA#=""~"+FLA$(2):GOTO 2140
2130 FLA$(3)=FNDCHA$(FLA$(1), LOCCON):FORMULA#=FLA$(2)
2140 ERASE LETRA$, LUGAR, PRE2
2150 GR=0:F=MRBL:A=MRBL:B=MRBL:C=MRBL:GOSUB 3000
2160 GOSUB 5000
2170 IF INDICADOR=1 GOTO 2190
2180 IF TIPO(RAMA, INDICE)=4 OR TIPO(RAMA, INDICE)=5 GOTO 2250
2190 GOSUB 6000
2200 IF INDICADOR=1 THEN INDICADOR=2:GOTO 2220
2210 INDICADOR=0:GOTO 1000
2220 IF TIPO(RAMA, INDICE)=3 OR TIPO(RAMA, INDICE)=5 THEN
FORMULA#=""~"+FLA$(3):GOTO 2240
2230 FORMULA#=FLA$(3)
2240 GOTO 2140
2250 INDICADOR=0:GOTO 6500
2260 IF TIPO(RAMA, INDICE)=1 THEN 2310
2270 IF TIPO(RAMA, INDICE)=8 AND SC$="K" THEN 1000
2280 IF TIPO(RAMA, INDICE)=8 THEN 2300
2290 IF TIPO(RAMA, INDICE)=6 THEN GOSUB 8000
2300 FORMULA#=""~"+FNDCHA$(FRMLA$(MAC, 0, 0), 2):GOTO 2360
2310 FORMULA#=FNDCHA$(FRMLA$(MAC, 0, 0), 2):GOTO 2360
2320 IF TIPO(RAMA, INDICE)=9 AND SC$="K" THEN 1000
2330 IF TIPO(RAMA, INDICE)=9 THEN 2350
2340 GOSUB 8000

```

```

2350 FORMULA$=FRMLA$(MAC,0,0)
2360 ERASE LETRA$,LUGAR,PRED
2370 GR=0:F=MRBL:A=MRBL:B=MRBL:C=MRBL
2380 GOSUB 3000
2390 GOSUB 5000
2400 GOSUB 6000
2410 GOTO 1000
3000 LONFOR=LEN(FORMULA$)* SUBROUTINA DE ANALISIS DE FORMULAS
3010 DIM LETRA$(LONFOR+2):DIM LUGAR(LONFOR):DIM PRED(LONFOR+2)
3020 FOR MC=1 TO LONFOR
3030 LETRA$(MC)=MID$(FORMULA$,MC,1)
3040 FOR ML=0 TO E-1
3050 IF LETRA$(MC)=PREDICADO1$(ML) THEN LUGAR(MC)=MC:
PRED(MC)=PREDICADO2(ML)
3060 NEXT ML
3070 NEXT MC
3080 FOR MC=1 TO LONFOR
3090 IF PRED(MC)=1 THEN 3100 ELSE 3150
3100 FRMLA$(MRBL,0,0)=LETRA$(MC)
3110 FRMLA$(MRBL,0,1)=STR$(LUGAR(MC))
3120 FRMLA$(MRBL,1,0)=STR$(LUGAR(MC))
3130 FRMLA$(MRBL,1,1)="-"+STR$(LUGAR(MC))
3140 MRBL=MRBL+1:B=MRBL-1:C=B
3150 IF PRED(MC)=3 OR PRED(MC)=2 THEN GR=GR+1
3160 NEXT MC
3170 FOR INDUCCION=0 TO GR
3180 FOR MC=A TO B
3190 COMIENZO=VAL(FRMLA$(MC,0,1)):FIN=VAL(FRMLA$(MC,1,0))
3200 IF PRED(COMIENZO-1)=2 THEN GOTO 3460
3210 FOR MRC=F TO C
3220 COMIENZO2= VAL(FRMLA$(MRC,0,1)):FIN2=VAL(FRMLA$(MRC,1,0))
3230 IF PRED(FIN+1)=3 AND COMIENZO2=FIN+2 THEN GOTO 3250
3240 IF PRED(COMIENZO-1)=3 AND FIN2=COMIENZO-2 THEN
GOTO 3360 ELSE 3520
3250 FRMLA$(MRBL,0,0)=""+"FRMLA$(MC,0,0)+LETRA$(FIN+1)+
FRMLA$(MRC,0,0)+""
3260 FRMLA$(MRBL,0,1)=STR$(COMIENZO-1)
3270 FRMLA$(MRBL,1,0)=STR$(FIN+1)
3280 FRMLA$(MRBL,1,1)=LETRA$(FIN+1)+STR$(LUGAR(FIN+1))
3290 MRBL=MRBL+1
3300 ON ERROR GOTO 3580
3310 LONMRBL=LEN(FRMLA$(MRBL-1,0,0))
3320 MEDIO$=MID$(FORMULA$,COMIENZO-1,LONMRBL)
3330 IF NOT MEDIO$=FRMLA$(MRBL-1,0,0) THEN MRBL=MRBL-1
3340 IF MRC < A THEN GOTO 3530 ELSE MC=MC+1
3350 GOTO 3530
3360 FRMLA$(MRBL,0,0)=""+"FRMLA$(MRC,0,0)+LETRA$(FIN2+1)+
FRMLA$(MC,0,0)+""
3370 FRMLA$(MRBL,0,1)=STR$(COMIENZO2-1)
3380 FRMLA$(MRBL,1,0)=STR$(FIN+1)
3390 FRMLA$(MRBL,1,1)=LETRA$(FIN2+1)+STR$(LUGAR(COMIENZO-1))
3400 MRBL=MRBL+1
3410 LONMRBL=LEN(FRMLA$(MRBL-1,0,0))
3420 MEDIO$= MID$(FORMULA$,COMIENZO2-1,LONMRBL)
3430 IF NOT MEDIO$=FRMLA$(MRBL-1,0,0) THEN MRBL=MRBL-1:GOTO 3520
3440 IF MRC < A THEN GOTO 3530 ELSE MC=MC+1
3450 GOTO 3530
3460 FRMLA$(MRBL,0,0)= LETRA$(COMIENZO-1)+FRMLA$(MC,0,0)
3470 FRMLA$(MRBL,0,1)=STR$(COMIENZO-1)
3480 FRMLA$(MRBL,1,0)=FRMLA$(MC,1,0)
3490 FRMLA$(MRBL,1,1)=LETRA$(COMIENZO-1)+STR$(LUGAR(COMIENZO-1))
3500 MRBL=MRBL+1
3510 GOTO 3530
3520 NEXT MRC
3530 NEXT MC
3540 C=MRBL-1:A=MRBL-((MRBL-1)-B):B=MRBL-1
3550 IF FRMLA$(B,0,0)=FORMULA$ THEN RETURN
3560 NEXT INDUCCION
3570 IF NOT FRMLA$(B,0,0)= FORMULA$ THEN CLS:LOCATE 5,15:
PRINT "la expresion era ":FORMULA$:LOCATE 6,15:PRINT "y no está bien
formada":STOP

```

```

3580 LLN=ERL
3590 IF ERR=2 AND LLN=2310 THEN GOTO 3570
3600 ON ERROR GOTO 0
3610 STOP
4000 PST=1' SUBROUTINA PARA VER SI UN ARBOL O RAMA ESTAN CERRADOS
4010 WHILE NOT ARBOL$(0,RIMA,PST)="" :PST=PST+1:WEND
4020 ARBOL$(2,RIMA,0)=STR$(PST)
4030 FOR MRV=1 TO PST
4040 FOR MVR=MRV+1 TO PST
4050 IF ARBOL$(1,RIMA,MRV)="tachada" OR ARBOL$(1,RIMA,MVR)="tachada"
THEN 4070
4060 IF (ARBOL$(0,RIMA,MVR)="-"+ARBOL$(0,RIMA,MRV)) OR
(ARBOL$(0,RIMA,MRV)="-"+ARBOL$(0,RIMA,MVR)) THEN
ARBOL$(0,RIMA,0)="cerrada":GOTO 4090
4070 NEXT MVR
4080 NEXT MRV
4090 FOR RM=1 TO RAMIFICACION
4100 IF NOT ARBOL$(0,RM,0)="cerrada" THEN RETURN
4110 NEXT RM
4120 TAUTOLOGIA=1 :GOTO 7000
4130 STOP
4140 FOR RMA=1 TO RAMIFICACION
4150 IF ARBOL$(0,RMA,0)="cerrada" THEN 4230
4160 WHILE NOT ARBOL$(0,RMA,PST)="" :PST=PST+1:WEND
4170 FOR MRV=1 TO PST
4180 FOR MVR=MRV+1 TO PST
4190 IF ARBOL$(1,RMA,MRV)="tachada" OR
ARBOL$(1,RMA,MVR)="tachada" THEN 4210
4200 IF ARBOL$(0,RMA,MVR)="-"+ARBOL$(0,RMA,MRV) OR
ARBOL$(0,RMA,MRV)="-"+ARBOL$(0,RMA,MVR) THEN
ARBOL$(0,RMA,0)="cerrada":GOTO 4230
4210 NEXT MVR
4220 NEXT MRV
4230 PST=1:NEXT RMA
4240 GOTO 4090
5000 TIPO=0 ' SUBROUTINA DE ASIGNACION DE TIPO
5010 CONECTIVA$= LEFT$(FRMLA$(B,1,1),1)
5020 IF CONECTIVA$="-" THEN 5080
5030 IF CONECTIVA$="%" THEN TIPO=2:GOTO 5150
5040 IF CONECTIVA$="v" THEN TIPO=4:GOTO 5150
5050 IF CONECTIVA$="N" THEN TIPO=9:GOTO 5150
5060 IF CONECTIVA$="-" THEN TIPO=0:GOTO 5150
5070 TIPO=7:GOTO 5150
5080 CONECTIVA2$=LEFT$(FRMLA$(B-1,1,1),1)
5090 IF CONECTIVA2$="-" THEN TIPO=1:GOTO 5150
5100 IF CONECTIVA2$="v" THEN TIPO=3:GOTO 5150
5110 IF CONECTIVA2$="%" THEN TIPO=5:GOTO 5150
5120 IF CONECTIVA2$="P" THEN TIPO=8:GOTO 5150
5130 IF CONECTIVA2$="N" THEN TIPO=6:GOTO 5150
5140 IF CONECTIVA2$="-" THEN TIPO=0
5150 RETURN
6000 IND=1'SUBROUTINAS DE EXTENSION A)EXTENSION SIMPLE
6010 WHILE NOT ARBOL$(0,RAMA,IND)="" :IND=IND+1:WEND
6020 ARBOL$(0,RAMA,IND)=FRMLA$(B,0,0)
6030 ARBOL$(2,RAMA,IND)=FRMLA$(B,1,1)
6040 ARBOL$(2,RAMA,0)= STR$(IND)
6050 INBUSCA(RAMA,IND)=B:TIPO(RAMA,IND)=TIPO:TIPO=0:GOTO 6080
6060 IND=IND+1:GOTO 6020
6070 RIMA=RAMA:GOSUB 4000
6080 RETURN
6500 IND=1:X=X+1' B)RAMIFICACION
6510 RAMIFICACION=RAMIFICACION+1
6520 ARBOL$(0,RAMIFICACION,0)="abierta"
6530 WHILE NOT ARBOL$(0,RAMA,IND)=""
6540 ARBOL$(0,RAMIFICACION,IND)=ARBOL$(0,RAMA,IND)
6550 ARBOL$(1,RAMIFICACION,IND)=ARBOL$(1,RAMA,IND)
6560 ARBOL$(2,RAMIFICACION,IND)=ARBOL$(2,RAMA,IND)
6570 INBUSCA(RAMIFICACION,IND)=INBUSCA(RAMA,IND)
6580 TIPO(RAMIFICACION,IND)=TIPO(RAMA,IND)
6590 IND=IND+1
6600 WEND

```

```

6610 ARBOL$(0,RAMIFICACION,IND-1)=FRMLA$(B,0,0)
6620 ARBOL$(2,RAMIFICACION,IND-1)=FRMLA$(B,1,1)
6630 ARBOL$(1,RAMIFICACION,0)=STR$(RAMA)+"/"+STR$(IND-1)
6640 ARBOL$(1,RAMIFICACION,IND-1)=""
6650 ARBOL$(2,RAMIFICACION,0)=STR$(IND)
6660 INBUSCA(RAMIFICACION,IND-1)=0
6670 TIPO(RAMIFICACION,IND-1)=TIPO:TIPO=0
6680 RIMA=RAMIFICACION:GOSUB 4000
6690 GOTO 1000
7000 REM DISPOSITIVO IMPRESOR
7010 CLS
7020 DIM COLUMNA(20,30):DIM INCREMENTO(20)
7030 FOR MR=1 TO 20
7040 FOR FZ=1 TO 30
7050 IF NRC=0 THEN COLUMNA(MR,FZ)=25 ELSE COLUMNA(MR,FZ)=NRC
7060 NEXT FZ
7070 NEXT MR
7080 SERIE(1)=1:US=1
7090 IF US=0 THEN 7420
7100 LIMIT1=VAL(ARBOL$(2,SERIE(US),0))
7110 FOR RM=RAMIFICACION TO SERIE(US)+1 STEP-1
7120 LIMIT2=VAL(ARBOL$(2,RM,0))
7130 IF LIMIT1 >= LIMIT2 THEN LIMITE=LIMIT1 ELSE LIMITE=LIMIT2
7140 IF ARBOL$(0,0,RM)="pintada" THEN 7350
7150 SR = INSTR(ARBOL$(1,RM,0),"/")
7160 RA= VAL(LEFT$(ARBOL$(1,RM,0),SR-1))
7170 NR=VAL(RNDCHA$(ARBOL$(1,RM,0),SR+1))
7180 IF NOT RA=SERIE(US) THEN 7390
7190 LON=LEN(ARBOL$(0,SERIE(US),NR-1)):LIN=LEN(ARBOL$(0,SERIE(US),NR))
7200 IF LON>LIN THEN AUMENTO=LON ELSE AUMENTO=LIN+1
7210 IF LON > INCREMENTO(SERIE(US)) THEN INCREMENTO(SERIE(US))=AUMENTO
7220 FOR FF=1 TO LIMITE
7230 IF FF < NR THEN 7240 ELSE 7270
7240 ARBOL$(0,RM,FF)=""
7250 COLUMNA(RM,FF)=COLUMNA(SERIE(US),NR)
7260 GOTO 7300
7270 LARGO1=LEN(ARBOL$(0,RM,FF))
7280 IF LARGO1 > LARGO THEN LARGO=LARGO1
7290 COLUMNA(RM,FF)=COLUMNA(SERIE(US),NR)+INCREMENTO(SERIE(US))
7300 NEXT FF
7310 FOR BETA=US TO 1 STEP-1
7320 INCREMENTO(SERIE(BETA))=INCREMENTO(SERIE(BETA))+LARGO+2
7330 NEXT BETA
7340 LARGO=0
7350 US=US+1
7360 SERIE(US)=RM
7370 ARBOL$(0,0,RM)="pintada"
7380 GOTO 7090
7390 NEXT RM
7400 US=US-1
7410 GOTO 7090
7420 CLS:FOR RAMA=1 TO RAMIFICACION
7430 LIMIT=VAL(ARBOL$(2,RAMA,0))
7440 FOR IN=1 TO LIMIT-1
7450 IF NOT ARBOL$(1,RAMA,IN)="tachada" THEN GOTO 7590
7460 IF NOT ARBOL$(0,RAMA,IN)="" THEN 7470 ELSE 7520
7470 LOCATE IN,COLUMNA(RAMA,IN)-1:PRINT CHR$(91)
7480 LOCATE IN,COLUMNA(RAMA,IN):PRINT ARBOL$(0,RAMA,IN)
7490 LOCATE IN,COLUMNA(RAMA,IN)+LEN(ARBOL$(0,RAMA,IN))
7500 PRINT CHR$(93)
7510 GOTO 7560
7520 V=SCREEN(IN,COLUMNA(RAMA,IN)-1):V1=SCREEN(IN,COLUMNA(RAMA,IN))
7530 IF V=32 AND NOT V1=32 THEN 7540 ELSE 7560
7540 LOCATE IN,COLUMNA(RAMA,IN)-1
7550 PRINT CHR$(91)
7560 LOCATE IN,1:PRINT "[I en"
7570 LOCATE IN,4+(2*RAMA):PRINT RAMA
7580 GOTO 7610
7590 LOCATE IN,COLUMNA(RAMA,IN)
7600 PRINT ARBOL$(0,RAMA,IN):GOTO 7610
7610 NEXT IN

```

```

7620 IF ARBOL$(0,RAMA,0)="cerrada" THEN 7630 ELSE 7660
7630 LOCATE IN,COLUMNA(RAMA,IN-1):PRINT "*"
7640 LOCATE IN+1,COLUMNA(RAMA,IN-1)-1:PRINT RAMA
7650 GOTO 7680
7660 LOCATE IN,COLUMNA(RAMA,IN-1)-1
7670 PRINT RAMA
7680 NEXT RAMA
7690 IF TAUTOLOGIA THEN 7700 ELSE 7730
7700 PRINT :PRINT
7710 LOCATE ,25:PRINT "la fórmula es insatisfacible en ";SC$
7720 END
7730 PRINT :PRINT
7740 LOCATE ,25:PRINT "la fórmula es satisfacible en ";SC$
7750 END
8000 REM RUTINAS DE MODIFICACION DE RAMA
8010 LIMITA=VAL(ARBOL$(2,RAMA,0))
8020 FOR IN=1 TO LIMITA-1
8030 CNTVA1$=LEFT$(ARBOL$(0,RAMA,IN),1)
8040 CNTVA2$=LEFT$(ARBOL$(0,RAMA,IN),2)
8050 IF CNTVA1$="N" OR CNTVA2$="¬P" THEN 8090 ELSE
ARBOL$(1,RAMA,IN)="tachada"
8060 NEXT IN
8070 RETURN
8080 STOP
8090 IF SC$="S4" THEN ARBOL$(1,RAMA,IN)="" :GOTO 8060
8100 IF ARBOL$(1,RAMA,IN)="tachada" THEN 8060
8110 IF CNTVA1$="N" THEN FORMULA$=FNDCHA$(ARBOL$(0,RAMA,IN),2):GOTO 8130
8120 FORMULA$="¬"+FNDCHA$(ARBOL$(0,RAMA,IN),3)
8130 ARBOL$(1,RAMA,IN)="tachada"
8140 ERASE LETRA$,LUGAR,PRE2' envio
8150 GR=0:F=MRBL:A=MRBL:B=MRBL:C=MRBL:GOSUB 3000
8160 GOSUB 5000
8170 GOSUB 6000
8180 GOTO 8060

```

referencia bibliográfica

- FITTING (1983): Fitting, M.—Proof Methods for Modal and Intuitionistic Logics. D. Reidel. Dordrecht, 1983.
- SMULLYAN (1968): Smullyan, R.—First-Order Logic. Springer. New York, 1968.