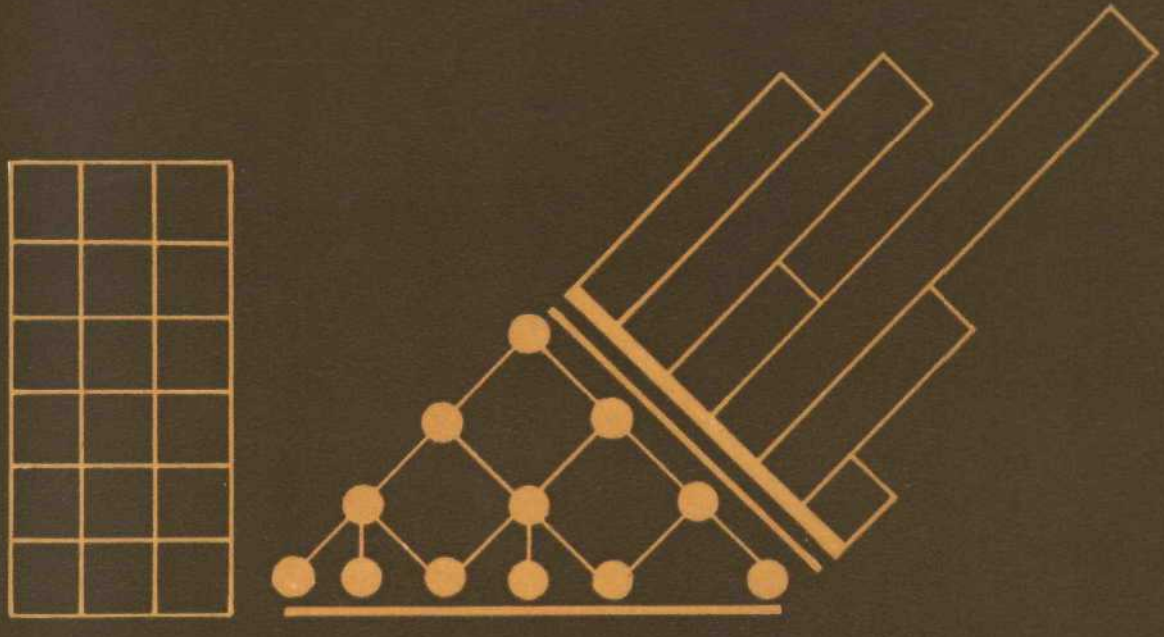
CONSTRUCCION DE UN TRADUCTOR

R. MOYA QUILES

I. RAMOS SALAVERT



Construcción de un traductor

CONSTRUCCION DE UN TRADUCTOR

por

Roberto Moya Quiles *Isidro Ramos Salavert*

PROFESORES DEL INSTITUTO DE INFORMATICA

INSTITUTO DE INFORMATICA
MADRID, 1974

Edita: Servicio de Publicaciones del Ministerio de Educación y Ciencia
Imprime: IMNASA. Menorca, 47. Madrid-9
Depósito Legal: M. 31.572-1974
ISBN: 84-369-0351-X
Printed in Spain

INDICE

Págs.

1. INTRODUCCION

Con el desarrollo de los lenguajes de programación se han impuesto cada vez más las condiciones de flexibilidad en el uso y construcción de traductores, el objeto de este trabajo es presentar algunas de las técnicas utilizadas, mediante su aplicación a la construcción de un traductor, encuadrable en el tipo de los metatraductores.

El concepto de "metatraductor", se basa en considerar un traductor independiente de la máquina, en el sentido de que tanto el lenguaje que será aceptado (fuente) como el código que se genera (objeto) se suministran junto con el programa a traducir; es decir, ambos lenguajes fuente y objeto son parámetros de entrada al metatraductor.

En principio los traductores se escribían a mano en lenguaje máquina o algún lenguaje de bajo nivel. La tendencia actual es escribirlos en lenguaje de alto nivel, intentando con ello automatizar la codificación, o en su defecto, reducir sensiblemente su tiempo, el tiempo de depuración será menor; es más, si el proceso previo fue automatizado, la depuración no será necesaria, obtendremos incluso una mayor legibilidad del traductor una vez construido.

Por tanto, necesitamos implementar lenguajes específicos para la escritura de traductores y seguir técnicas especiales, constituyentes ambos de lo que denominamos Sistemas de Escritura de Traductores (TWS, Translator Writing Systems). Cabe mencionar entre los TWS, al cual pertenecerá nuestro metatraductor, los compiladores de compiladores, los compiladores de lenguaje de producciones, los constructores de algoritmos para análisis automático, entre otros.

La estructura de los TWS consta fundamentalmente de dos componentes:

a) *Un lenguaje* L_1 *para describir la sintaxis del lenguaje* L_2 *en el cual los programas serán escritos (diremos que* L_1 *es el metalenguaje de* L_2*).*

b) *Un lenguaje en el cual se escriben las rutinas semánticas (por ejemplo: FORTRAN, ALGOL, etc.).*

Si tenemos presente que en un proceso de traducción cambia la sintaxis, permaneciendo la semántica, deducimos que la fase de análisis sintáctico es el corazón del proceso, no dejando de reconocer que el tratamiento semántico, si bien es obvio, constituye una ardua tarea de programación.

Esta tarea es automatizable si podemos hacer uso de algún mecanismo de descripción de un lenguaje en función de otro, puesto que de esta forma toda decisión tomada sobre el texto fuente, se convertirá en una decisión entre términos análogos en el texto objeto.

En nuestro traductor implementaremos este mecanismo mediante tablas.

Finalmente, cabe mencionar que los trabajos actuales sobre traductores se basan generalmente en los estudios de Knuth (principio de su "máquina sintáctica"), Foster (Programas de prueba de sintaxis), Ferguson (de quien partió la idea de un metatraductor) y Feldman y Gries (Sistemas de Escritura de Traductores).

Este trabajo se ha concebido con un propósito estrictamente didáctico. El lenguaje escogido a efectos de implementación nos sirve para ejemplificar técnicas en sí aplicables a lenguajes más complejos dada su potencialidad, y el lenguaje objeto es el utilizado por los alumnos del Instituto de Informática desde primer año. Con ello justificamos la elección de ambos insistiendo en que el esquema es mucho más potente y aplicable a lenguajes de alto nivel.

2. SOPORTE TEORICO

2.1. DEFINICIONES

2.1.1. Concepto de gramática y tipos

En la definición de un lenguaje intervienen primordialmente dos aspectos, el sintáctico (gramática) y el semántico (significado).

El primero ha sido más desarrollado, de tal suerte que podemos elegir entre varias formas descriptivas [1].

Las más utilizadas son: los autómatas abstractos y las gramáticas de Chomsky, uno de cuyos tipos se corresponde con la BNF.

Otras varias son: lenguaje natural, expresiones regulares de Kleene, diagramas de estado, máquina de Turing.

En cuanto al aspecto semántico, su desarrollo es menos espectacular, si bien en la actualidad se está afrontando formalizadamente.

Aplicando el clásico concepto de gramática, diremos que, el conjunto de reglas que describen, de forma generativa, un lenguaje de programación dando un procedimiento para producir todas las sentencias legales del mismo, se denomina la *gramática de ese lenguaje de programación.*

De otra parte y según esto, una gramática dada, nos permite determinar si una sentencia es o no válida en el lenguaje que ella define.

Una gramática (Tipo 2) consta de cuatro componentes [2]:

1) Un alfabeto A_n de caracteres no terminales (también denominados sintagmas, clases sintácticas, variables metalingüísticas), que especifican conjuntos utilizados en la construcción del lenguaje.

2) Un alfabeto A_t de caracteres terminales, que se utilizan en el lenguaje.

3) Un conjunto de reglas denominadas producciones, escritas según el formato

$$N \rightarrow C$$

siendo $N \in A_n$ y C una cadena de elementos C_i, $C_i \in (A_n \cup A_t)$; pudiendo ser la cadena vacía, que representaremos por Φ.

4) Un elemento singular (también denominado símbolo inicial, símbolo distinguido o axioma de la gramática) que simboliza todo el lenguaje y, en general, no aparecerá en ninguna cadena C.

Debemos observar que al hablar de alfabetos de caracteres terminales y no terminales nos referimos a uno o más "objetos" (un símbolo que se denota así mismo o a la clase de símbolos similares a él [3]) concatenados, es decir, un nuevo "objeto" con identidad propia. Así, por ejemplo, si consideramos los "objetos": "S", "U", "M", "A", aplicando la operación concatenación, obtendremos un nuevo "objeto": "SUMA", símbolo perteneciente a un alfabeto terminal o no terminal, según se defina en una gramática determinada.

En general [4], denotando por A_t^* el conjunto de todas las cadenas finitas formadas con elementos del alfabeto A_t, podemos decir que un lenguaje L es un subconjunto de A_t^* ($L \subset A_t^*$).

mento de descripción sintáctica muy flexible y de que los lenguajes de programación suelen tener muchos aspectos del tipo 2, han hecho que éstas sean las más utilizadas.

El metalenguaje BNF se corresponde exactamente con las gramáticas de este tipo, teniendo las siguientes particularidades:

1) Las metavariables se representan entre los metaparéntesis < >; por ejemplo: <sentencia>.

2) Los símbolos terminales se representan por ellos mismos.

3) La flecha de las reglas de producción de las gramáticas tipo 2, se sustituye por el símbolo : : =

Así: <número> : : = 26.4

4) Las distintas partes derechas de una clase sintáctica o metavariable se separan por el símbolo ¦ (significa o) y se asignan a la misma clase sintáctica. Así: <dígito binario> : : = 0 | 1.

Asociada con la BNF está la carta sintáctica, que consiste en una representación gráfica en dos dimensiones de la anterior.

La notación utilizada en las cartas sintácticas es la siguiente:

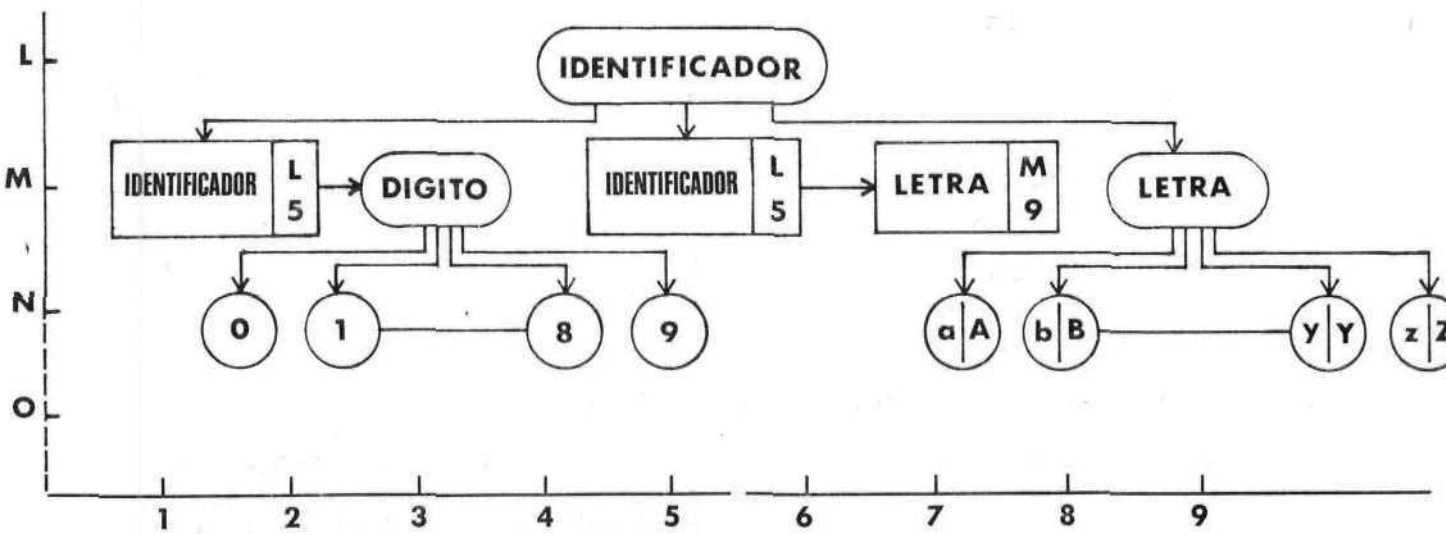
a) Asociado con cada clase sintáctica hay un óvalo, en el lugar de su definición (única) y un rectángulo en el lugar donde se usa (que contiene el nombre de la clase sintáctica y lugar donde se define).

b) Los símbolos terminales se representan tantas veces como haga falta, dentro de un círculo.

c) Cada una de las partes derechas de una regla se sitúan debajo del óvalo de definición de la parte izquierda y en una misma horizontal, sustituyéndose el símbolo : : = (para cada parte derecha) por una flecha vertical que parte del óvalo de definición y va a la representación de la parte derecha.

d) La concatenación entre los elementos de cada parte derecha viene representada por una flecha horizontal.

Un ejemplo parcial de carta sintáctica sería el siguiente, el cual presenta la definición de identificador en el ALGOL 60:



(*L*,5) y (*M*,9) son las coordenadas de los puntos de definición dentro de la carta sintáctica, de las clases sintácticas, identificador y letra, respectivamente.

En el apéndice A se presenta una carta sintáctica del ALGOL 60.

Consideramos como formas ampliadas o derivadas de la BNF las siguientes [6]:

a) Las que *permiten la factorización de los términos comunes de las distintas partes derechas* de una misma clase sintáctica, usándose para ello los metaparéntesis $[\![$ y $]\!]$ y otros.

b) La repetición completa o parcial de una parte derecha de una producción se expresa con metaexponentes (indicando el número de veces que se repite...), o puntos suspensivos.

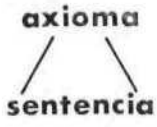
c) La definición de una metametagramática (que describe los símbolos del lenguaje usado en el nivel inferior para hablar de otro...)

Son formas que han proporcionado mayor flexibilidad en la descripción de lenguajes, pero en las que los lenguajes que describen no son siempre de tipo 2, perdiéndose toda la potencialidad de este esquema (un trabajo frecuente es el paso de una gramática ampliada a la BNF para usar los métodos ya desarrollados en esta última).

2.1.3. El problema del análisis

2.1.3.1. *Introducción*

Este problema puede definirse como: dada una sentencia y una gramática, construir el triángulo



e intentar llegar desde la base al vértice (bottom-up) o viceversa (top-down), y de izquierda a derecha, *a través de una serie de derivaciones en árbol, que formen una estructura consistente en sí misma* y que se denomina "parse" [8].

El orden en que dicho análisis se efectúe afecta a la eficiencia del proceso.

Independientemente de esto, los pasos intermedios consisten en buscar una producción que se ajuste al contexto local bajo consideración.

Puede ocurrir que desde una situación determinada, sea imposible generar un "parse" consistente, debido a haber elegido anteriormente una producción que se ajustaba a una situación local, pero no a la total, habiendo dado así un paso en falso. Deberíamos, pues, volver atrás ("backtracking") y probar de nuevo otra producción.

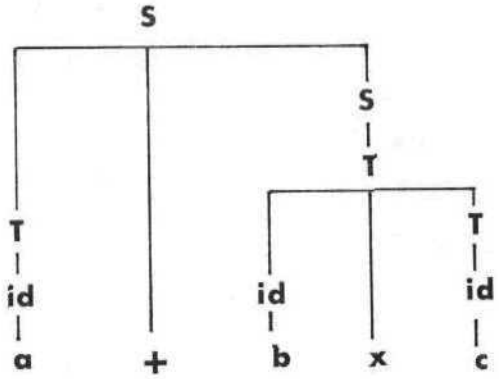
El problema de "backtracking" ha conducido a la clasificación de las gramáticas de acuerdo con el grado de contexto que se debe examinar para asegurar que la exactitud de un "parse" dado es tal que nunca pueda conducir al "backtracking". Una gramática libre de contexto se denomina de *contexto limitado (m,n)* si es siempre posible evitar "backtracking" examinando los m caracteres que preceden a la cadena ya analizados y los n caracteres que la siguen. Floyd desarrolló este concepto de gramáticas de contexto limitado (Bounded Context) [9].

Tendente a evitar el "backtracking" se han desarrollado técnicas consistentes, por ejemplo, en aceptar únicamente una producción si supera unas pruebas "a priori".

Un ejemplo de "parsing" sería el siguiente, supuesta una gramática expresada en BNF:

$$\langle S\rangle ::= \langle T\rangle \mid \langle T\rangle + \langle S\rangle$$
$$\langle T\rangle ::= \langle id\rangle \mid \langle id\rangle \times \langle T\rangle$$
$$\langle id\rangle ::= a \mid b \mid c$$

para la cadena $a + b \times c$ el triángulo de "parse" sería el siguiente:



2.1.3.2. *Análisis morfológico y sintáctico*

La distinción entre ambos procesos es más bien una tradición que una realidad a nivel de lenguajes de programación, ya que la formalización de éstos no necesita una diferenciación explícita, como sucede en la lingüística tradicional.

En el *preproceso* (análisis morfológico, lexicográfico y edición del texto fuente) se convierte la cadena de símbolos terminales que constituye el programa en otra formada por metavariables (elegidas a conveniencia) que será tomada a su vez como terminal en el proceso de análisis sintáctico.

Realmente, consiste en realizar un análisis parcial o primario que nos conduce a unos "axiomas" previos que son punto de partida de la etapa posterior. Se ha desdoblado, pues, el triángulo problema en dos áreas, a las que se les aplica la misma tecnología.

En el ejemplo presentado en 2.1.3.1. podemos señalar esta primera etapa de análisis morfológico, en la que los símbolos terminales se sustituyen por la metavariable, *<id>*, que los reduce directamente.

El objetivo principal de la fase de análisis sintáctico es tomar la cadena producida en la fase anterior y las tablas resultado del análisis lexicográfico y utilizando algún algoritmo de análisis, verificar que la misma es una cadena o conjunto de cadenas legales del lenguaje dado. Además, debe producir como salida una estructura, la cual estará lista para ser ejecutada o interpretada en la fase de generación de código objeto, pero que todavía se parece a una representación estructural de la cadena de que partimos [10].

2.1.3.3. *Analizadores morfológico y sintáctico.*

Los algoritmos encargados de la realización de cada etapa de análisis se denominan analizadores morfológico y sintáctico respectivamente.

Al primero se le pueden asignar las siguientes funciones:

a) Lectura del texto fuente.

b) Reconocimiento de los denominados "axiomas previos".

c) Construcción de las tablas de constantes (lexicográfico).

d) Construcción de la tabla de símbolos (lexicográfico).

e) Eliminación de blancos superfluos y comentarios (edición).

Al margen de la función de lectura, que puede ser independiente del preprocesador, éste no es sino un analizador sintáctico, ya que, en definitiva, utiliza la misma filosofía funcional [6].

En el diseño de un analizador sintáctico, se pueden adoptar distintas estrategias en función del método de análisis elegido.

Será un analizador ascendente "bottom-up" si el análisis parte del programa y va reconociendo subcadenas y etiquetándolas con nombres de metavariables que forman partes derechas de producciones que cuando estén completas se sustituirán por las correspondientes partes izquierdas y así sucesivamente hasta llegar al axioma de la gramática. El orden de sustitución consiste en reducir cuanto antes las subcadenas iniciales de izquierda a derecha ("parse" canónico).

El analizador descendente "top-down" seguirá el esquema contrario: partiendo del axioma de la gramática, trata de llegar en forma predictiva hasta la cadena terminal creando sucesivas formas sentenciales que cumplen ciertas condiciones "a priori".

2.1.3.4. *Semántica: Acciones semánticas.*

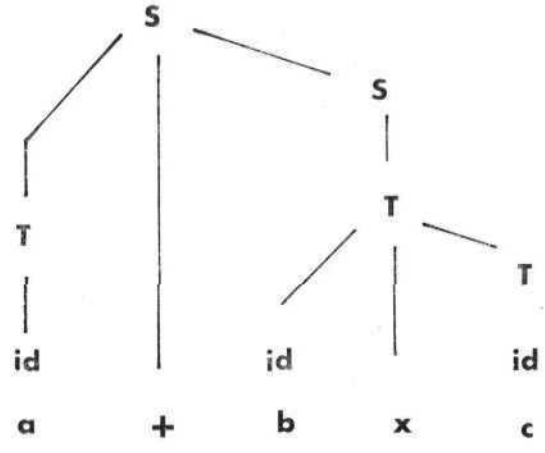
La evaluación de un programa presenta, además del aspecto sintáctico o estructural, el aspecto semántico.

Una frase (programa) de un lenguaje viene descrita por la gramática que la genera, a nivel estructural y subyacente con esta estructura, se encuentra el significado asignado.

Tomemos el ejemplo precedente (expresiones aritméticas):

$$\langle S\rangle ::= \langle T\rangle \mid \langle T\rangle + \langle S\rangle$$
$$\langle T\rangle ::= \langle id\rangle \mid \langle id\rangle \times \langle T\rangle$$
$$\langle id\rangle ::= a \mid b \mid c$$

dada la frase $a + b \times c$ obtendríamos su estructura, que será:



Además de la comprobación de que el programa está bien escrito, entendamos el significado de las sucesivas reducciones:

Reducciones del tipo $\langle id\rangle \leftarrow a \mid b \mid c$

El afirmar que un identificador puede ser a, b o c presenta dos aspectos:

1. Nos permite distinguir lo que es o no es un identificador (aspecto externo).

2. Una vez comprobado el punto 1) le asigna una categoría sintáctica con lo que nos aparece un nuevo objeto que llamaremos nombre de identificador y que da cuenta de su significado (pensemos que la letra a (o b o c) podría formar parte de una palabra clave real (por ejemplo) y no tener el mismo significado).

Un nombre de identificador se representa por una palabra de memoria (de dirección d) y su tipo t.

Cuando nosotros afirmamos

$$<id> \rightarrow a$$

decimos cómo se escribe un identificador aspecto 1) y además (acción semántica asociada) que el resultado de esta "reducción" es (d, t) (dirección y tipo) aspecto 2). Más exactamente deberíamos decir que el resultado de la compilación de a es (d, t).

Análogamente para b, c... o cualquier otra regla que nos especifique lo que es un identificador.

Reducciones del tipo $<T> \leftarrow <id>$

La necesidad de introducir reglas de este tipo viene impuesta por expresar cómodamente (de forma recursiva en nuestro caso) que un término es un producto de un número indefinido de $<id>$, no porque asignemos una categoría sintáctica superior a $<id>$. Son reglas que dan cuenta escuetamente de la estructura sin asignar un nuevo significado a $<id>$ (podríamos pensar en una acción semántica asociada vacía, para continuar con nuestro esquema inicial). Un $<id>$ es lo que es independientemente de esta nueva reducción.

Lo mismo podríamos decir de las reducciones de tipo $<S> \leftarrow <T>$.

El objeto designado por $<T>$ (identificador o no) no cambia de significado (no es un nuevo objeto).

Reducciones del tipo $<T> \leftarrow <id> \times <T>$

Aquí nuevamente sería interesante considerar los dos aspectos:

1. Estructural (o sintáctico): especificamos cómo se escribe un término (producto de un $<id>$ por un $<T>$).
2. Significado (o semántico): nos aparece un nuevo objeto designado externamente por el aspecto 1) pero con un significado nuevo:

$$<id> \times <T>$$
$$\downarrow \text{ designa}$$
$$\text{objeto}$$

El objeto designado es el resultado de la operación de multiplicación (con sus correspondientes conversiones de tipo, si necesario) y que habitualmente representaremos por una palabra de memoria (a la que accedemos mediante su dirección) y su tipo (resultado de las convenciones de compatibilidad y conversión de tipo autorizadas).

O sea, compilar $<T> \leftarrow <id> \times <T>$ significa:

1. Compilar $<id>$, lo que da como resultado (d_1, t_1).
2. Compilar $<T> \approx$ compilar $<id>$ (punto anterior) da como resultado (d_2, t_2).
3. Efectuar la acción (de forma inmediata: intérprete o diferida: compilador): $(d_1, t_1) \times (d_2, t_2)$ cuyo resultado será (d_3, t_3) (dirección y tipo del resultado).

Las acciones 1 y 2 habrán sido realizadas previamente; la acción 3 será la asociada a esta reducción y da cuenta del significado que nosotros asociamos a la acción de multiplicar, sería la acción semántica asociada con esta reducción.

Podemos pensar de forma análoga para las reducciones del tipo $<S> \leftarrow <T> + <S>$, por lo que omitimos su desarrollo.

2.2. FUNDAMENTOS DE NUESTRO TRABAJO

2.2.1. La máquina sintáctica de Knuth

2.2.1.1. *Introducción*

En el análisis sintáctico automático realizado en un computador, varios especialistas de la primera hora (Barnett, Brooker, Morris, etc.), utilizaron un método de análisis "top-down", sistema que está siendo utilizado en muchos de los compiladores actuales.

Knuth observó que los métodos de estos autores se pueden describir convenientemente en términos de un pequeño dispositivo parecido a un computador al que le dio el nombre de "Parsing Machine". En ella se centra la consideración de las propiedades sintácticas del citado método de análisis "top-down" [17].

La "Parsing Machine" (PM) es una máquina abstracta que analiza cadenas sobre un cierto alfabeto. Trabaja carácter a carácter de acuerdo con un programa. Este programa está formado por una *familia de procedimientos, llamándose entre sí recursivamente; el programa en sí mismo es uno de* estos procedimientos. Cada procedimiento se asocia a una metavariable y tratará de encontrar su tipo sintáctico particular en la entrada, dando una respuesta de verdadero o falso, según lo haya encontrado o no.

Las instrucciones tienen cuatro campos: un campo de dirección que puede estar en blanco o contener una metavariable que representa la dirección del procedimiento asociado con ella, un código de operación y dos direcciones denominadas *DC* y *DF* (dirección "cierto" y dirección "falso", respectivamente). Los procedimientos se escriben utilizando dos tipos de instrucciones correspondientes a dos formas distintas del código de operación:

Tipo 1) El código de operación es un símbolo del alfabeto terminal.

Tipo 2) El código de operación es una metavariable encerrada entre paréntesis cuadrados, que representa la dirección del procedimiento asociado a ella.

Supongamos una cadena de entrada: $S_1, S_2, \ldots S_n$ y sea S_h el carácter que está siendo analizado por la máquina en un momento dado. El efecto de aquellas instrucciones será el siguiente:

Tipo 1) *if* S_h = código operación *then* read (colocar $h = h + 1$) and *go to DC else goto DF.*

Tipo 2) *Call* al procedimiento cuya dirección indica el código de operación (recursivamente); *if si vuelve con el valor "cierto" then go to DC, else if si vuelve con el valor "falso" then go to DF.*

Cada campo *DC* o *DF* puede contener bien la dirección de una instrucción o uno de los dos símbolos especiales *C* o *F*. Si contiene *C* el procedimiento "vuelve" con el valor "cierto", si contiene una *F* el procedimiento "vuelve" con el valor "falso" y *h* es devuelto al valor que tenía cuando el procedimiento fue llamado (esto implica que el valor de *h* se guarda junto con la dirección de vuelta cuando se llama un procedimiento mediante una instrucción del tipo 2).

Si un campo *DC* o *DF* está en blanco, hará referencia a la instrucción que aparece en la línea que sigue inmediatamente.

Consideremos, por ejemplo, la siguiente gramática, expresada en BNF:

$$\langle B \rangle ::= \langle R \rangle \mid (\langle B \rangle)$$
$$\langle R \rangle ::= \langle E \rangle = \langle E \rangle$$
$$\langle E \rangle ::= a \mid b \mid \$ \langle E \rangle + \langle E \rangle \$$$

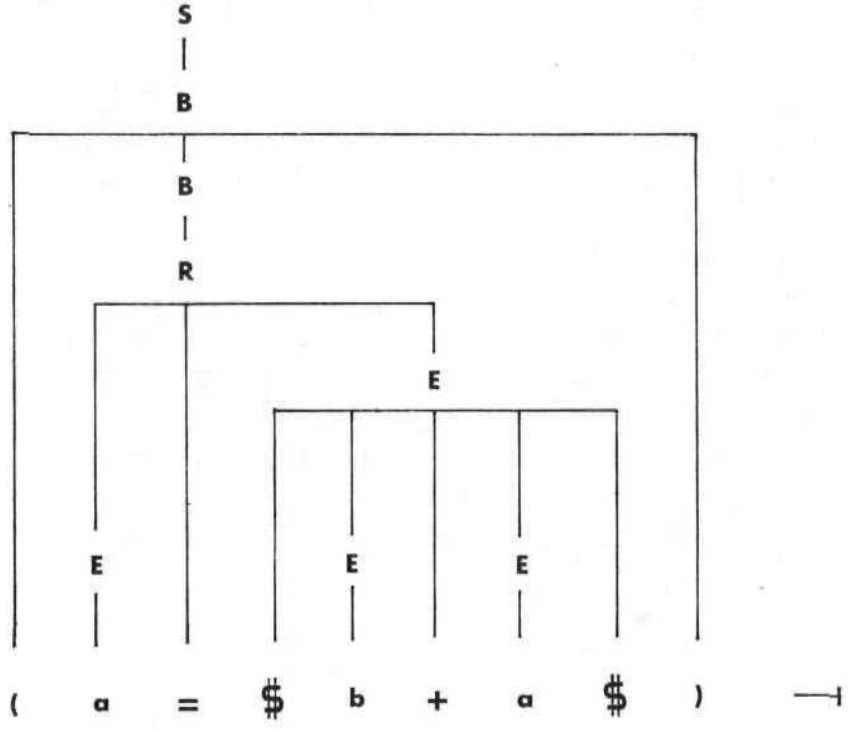
Utilizando el lenguaje descrito, podemos escribir el programa siguiente:

Dirección	Código operación	DC	DF
B	[*R*]	*C*	
	(		*F*
	[*B*]		*F*
	)	*C*	*F*
R	[*E*]		*F*
	=		*F*
	[*E*]	*C*	*F*
E	*a*	*C*	
	b	*C*	
	$		*F*
	[*E*]		*F*
	+		*F*
	[*E*]		*F*
	$	*C*	*F*
S	[*B*]		ERROR
	—|	CORRECTO	ERROR

el símbolo —| denota el fin de una sentencia.

El procedimiento *S* inicia el proceso y su última instrucción proporciona por la "dirección de cierto" el mensaje correcto.

El árbol de análisis para una cadena de entrada, por ejemplo: $(a = \$\ b + a\$)$ *sería:*



Cabe hacer notar la correspondencia entre la gramática y el programa en lenguaje para la máquina sintáctica.

El método de análisis que la máquina sintáctica de Knuth nos proporciona, requiere aplicar a la gramática cuatro reglas restrictivas, que aseguran el éxito del algoritmo, así como la eliminación del "backtracking".

Dichas condiciones son [17]:

1.ª No se permitirán producciones de la forma

$$A \overset{*}{\longrightarrow} A_a \qquad \text{donde} \qquad A \in A_n \quad a \in V^*$$

esto equivale a *no* admitir la recursividad a izquierdas, pues, efectivamente, ello cerraría el proceso en un ciclo sin fin.

2.ª Los símbolos terminales que pueden encabezar las distintas alternativas de una clase sintáctica deben formar conjuntos disjuntos. Es decir:

$$A \overset{*}{\longrightarrow} Bb \mid Cc$$

donde:

$$A, B, C \in A_n \qquad b, c \in V^*$$

no debe ocurrir que:

$$B \overset{*}{\longrightarrow} de$$

$$C \overset{*}{\longrightarrow} df$$

con:

$$d \in A_t \qquad e, f \in V^*$$

Esto implica que en todo momento el símbolo términal que estamos examinando nos señale sin ambigüedad qué alternativa de una clase sintáctica debemos escoger, sin posibilidad de error y, en consecuencia, sin "backtracking".

3.ª Si una alternativa de una clase sintáctica origina la cadena vacía los símbolos terminales que pueden seguir a la clase en la producción donde aparezca han de formar un conjunto disjunto con los terminales que pueden encabezar las distintas alternativas de dicha clase sintáctica.

Es decir, supongamos la cadena

$$\ldots u_1 \ldots u_2 \ldots u_3 u_4 u_5 \ldots$$

y sea u_3 el símbolo que se está analizando;

si existen las producciones: $u_3 \longrightarrow a\ x \mid \Phi$

$$u_4 \longrightarrow u_3\ a\ y$$

Para el supuesto de que u_3 sea el símbolo a, tendríamos conflicto, pues no podemos determinar si el elemento a pertenece a la producción de u_3 o a la de u_4 que implica que u_3 es transparente.

4.ª Ninguna clase sintáctica puede tener dos o más alternativas que conduzcan a la cadena vacía.

Esta condición se deriva de la 3.ª

Un ejemplo puede ser:

Dadas las producciones $X \longrightarrow A \mid B$

$$A \longrightarrow \Phi \mid C$$

$$B \longrightarrow \Phi \mid D$$

obviamente tendremos conflicto al decidir si es A o B.

2.2.1.2. *Relaciones binarias fundamentales definidas por una gramática de contexto libre.*

Dada una gramática (tipo 2) $G = \{ A_t, A_n, P, S \}$ podemos definir las siguientes relaciones binarias, impuestas por P sobre los elementos del conjunto finito $V = A_t \cup A_n$.

a) Dependencia simple (representada por R_s).

Diremos que $\Sigma \in A_n$ y $\alpha \in V$ cumplen:

$$\Sigma \; R_s \; \alpha$$

Sí y solamente si α aparece en una parte derecha de la regla en la que Σ es la parte izquierda, es decir:

$$\exists \; \Sigma \rightarrow \sigma_1 \; \alpha \; \sigma_2 ; \; \sigma_1, \sigma_2 \in V^*$$

Expresando la dependencia simple en forma de grafo habría una línea de unión entre Σ y α.

Si esta relación binaria la representamos mediante una matriz booleana A (n,n), siendo:

$$n = \text{cardinal } (V),$$

escribiremos un 1 en el elemento de coordenadas A (Σ, α); donde no exista relación aparecerá un cero. Las columnas de esta matriz nos proporcionan, pues, las tablas de referencias cruzadas para cada α dado.

Veamos un ejemplo de lo expuesto.

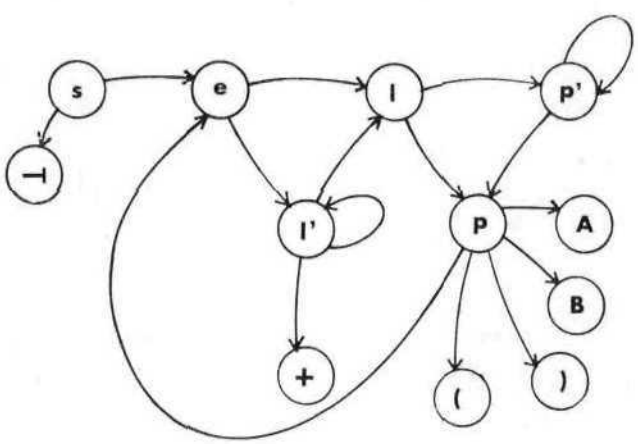
Dada la gramática $G = \{ A_t, A_n, P, S \}$
donde:

$$A_t = \{ A, B, +, (,), \dashv \}$$
$$A_n = \{ e, \; 1, \; 1', \; p, \; p', \; s \}$$

Axioma: s.
Producciones: P

$$\begin{aligned} e &::= 1 \; 1' \\ 1' &::= + \; 1 \; 1' \\ &\quad \Phi \\ 1 &::= p \; p' \\ p' &::= p \; p' \\ &\quad \Phi \\ p &::= A \\ &\quad B \\ &\quad (e) \\ s &::= e \; \dashv \end{aligned}$$

el grafo de dependencia simple sería:



que aceptaría la representación matricial:

		A_n						A_t					
		s	e	l'	l	p'	p	A	B	(	)	+	—|
A_n	s	0	1	0	0	0	0	0	0	0	0	0	1
	e	0	0	1	1	0	0	0	0	0	0	0	0
	l'	0	0	1	1	0	0	0	0	0	0	1	0
	l	0	0	0	0	1	1	0	0	0	0	0	0
	p'	0	0	0	0	1	1	0	0	0	0	0	0
	p	0	1	0	0	0	0	1	1	1	1	0	0
A_t	A	0	0	0	0	0	0	0	0	0	0	0	0
	B	0	0	0	0	0	0	0	0	0	0	0	0
	(	0	0	0	0	0	0	0	0	0	0	0	0
	)	0	0	0	0	0	0	0	0	0	0	0	0
	+	0	0	0	0	0	0	0	0	0	0	0	0
	—|	0	0	0	0	0	0	0	0	0	0	0	0

b) Dependencia a izquierdas (representada por R_I), también denominada relación Inicial.

Diremos que $\Sigma \in A_n$ y $\alpha \in V$ cumplen:

$$\Sigma \, R_I \, \alpha$$

si, y solamente si α aparece como primer símbolo (distinto de la cadena vacía) en una producción de la gramática en la que Σ es la parte izquierda.

Así pues, tendremos:

$$\exists \; \Sigma \rightarrow \Sigma_1 \, \Sigma_2 \, \ldots \, \Sigma_p \, \alpha \, \varphi$$

$$P \geq 0$$

$$\Sigma_1, \Sigma_2, \ldots, \Sigma_p \in A_n, \varphi \in V^*$$

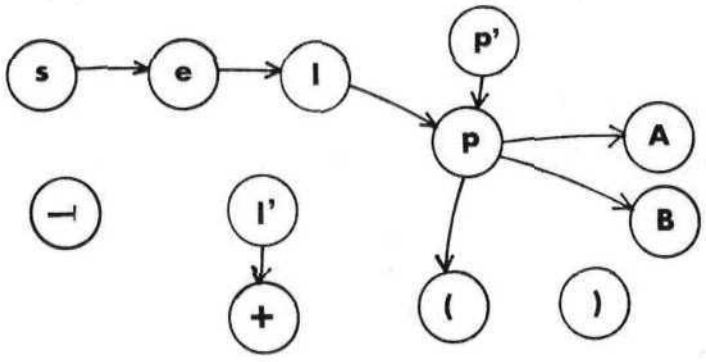
$$\Sigma_1 \overset{*}{\Rightarrow} \Phi$$

$$\vdots$$

$$\Sigma_p \overset{*}{\Rightarrow} \Phi$$

Representando esta relación en forma de la matriz booleana, A (n,n), podemos fácilmente almacenar en el ordenador el grafo de dependencia a izquierdas.

Para la gramática dada en el ejemplo anterior el grafo de dependencia a izquierdas sería:



que se puede representar matricialmente en la forma:

		A_n						A_t					
		s	e	l'	l	p'	p	A	B	(	)	+	—|
A_n	s	0	1	0	0	0	0	0	0	0	0	0	0
	e	0	0	0	1	0	0	0	0	0	0	0	0
	l'	0	0	0	0	0	0	0	0	0	0	1	0
	l	0	0	0	0	0	1	0	0	0	0	0	0
	p'	0	0	0	0	0	1	0	0	0	0	0	0
	p	0	0	0	0	0	0	1	1	1	0	0	0
A_t	A	0	0	0	0	0	0	0	0	0	0	0	0
	B	0	0	0	0	0	0	0	0	0	0	0	0
	(	0	0	0	0	0	0	0	0	0	0	0	0
	)	0	0	0	0	0	0	0	0	0	0	0	0
	+	0	0	0	0	0	0	0	0	0	0	0	0
	—|	0	0	0	0	0	0	0	0	0	0	0	0

Ahora bien, si nos interesan los terminales que pueden encabezar una clase sintáctica dada, o las clases sintácticas que pueden empezar por un terminal dado, deberemos realizar para cada elemento $A\ (i, j) = 1$ de la matriz A, una operación *OR* de las filas i y j dejando el resultado como una fila i.

Esto equivale a determinar en el grafo los caminos entre dos vértices dados, a lo que se denomina el cierre transitivo de esta relación.

Las clases sintácticas no accedidas por ningún camino de los posibles son las que reciben el nombre de símbolos inaccesibles del lenguaje. Se reconocen fácilmente en la matriz booleana por ser sus fila y columna todo ceros binarios. Esta sería una vía de automatización del proceso de limpieza de gramáticas.

Definamos a continuación la relación *siguiente* que notaremos S:
Dado $A \in V$, $B \in V$, $A\,S\,B$ si y solamente si:

$$A\,S\,B <=> \{(\exists\, c \in N,\ \exists\, \varphi \in V^*,\ \exists\, \psi \in V^*) \mid c : := \varphi\, A\, B\, \psi\}$$

Esto para el caso de una gramática sin Φ (cadena vacía); para una gramática con Φ podemos generalizar fácilmente la relación anterior en la forma

$$A\,S\,B <=> \{(\exists\, c \in N,\ \exists\, \varphi \in V^*,\ \exists\, \psi \in V^*, \exists \mu \in V^*) \mid c : := \varphi\, A\, \mu B\, \psi \quad \text{y} \quad \mu \overset{*}{>\!\!-} \Phi\}$$

Donde la relación $>\!\!-$ es la relación de producción que definimos en la forma

$$\alpha >\!\!- \beta <=> \{(\exists\, \omega \in V^*,\ \exists\, \omega' \in V^*,\ \exists\, A \in N,\ \exists\, \lambda \in V^*) \mid \alpha = \omega\, A\, \omega',\ \beta = \omega\, \mu\, \omega' \quad \text{y} \quad A : := \mu\}$$

y representamos por $\overset{*}{>\!\!-}$ su cierre transitivo.

Pasemos a considerar un algoritmo que calcule el cierre transitivo basándonos en lo que acabamos de exponer.

2.2.1.2.1. *Algoritmo de Warshall*

Warshall definió este algoritmo [21], que escribimos en lenguaje ALGOL [22]:

```
procedure    WARS    (A, N)
boolean array    A    [1 : N, 1 : N];
begin integer    i, j, k;
for i := 1    step 1 until N do
for j := 1    step 1 until N do
    if    A (j, i) = 1    then
    for k := 1 step 1 until N do
        A (j, k) := A (j, k) ∪ A (i, k);
end WARS
```

obtenemos así una nueva matriz $A^+ (n, n)$ en la que considerando las filas correspondientes a las metavariables, obtendríamos los terminales que pueden encabezarlas. Y considerando las columnas correspondientes a los terminales, las clases sintácticas que empiezan cada uno de ellos.

Aplicado al cálculo del cierre transitivo del grafo de dependencia a izquierdas visto anteriormente, obtendremos la matriz $A^+ (n, n)$:

		A_n						A_t					
		s	e	l'	l	p'	p	A	B	(	)	+	—|
A_n	s	0	1	0	1	0	1	1	1	1	0	0	0
	e	0	0	0	1	0	1	1	1	1	0	0	0
	l'	0	0	0	0	0	0	0	0	0	0	1	0
	l	0	0	0	0	0	1	1	1	1	0	0	0
	p'	0	0	0	0	0	1	1	1	1	0	0	0
	p	0	0	0	0	0	0	1	1	1	0	0	0
A_t	A	0	0	0	0	0	0	0	0	0	0	0	0
	B	0	0	0	0	0	0	0	0	0	0	0	0
	(	0	0	0	0	0	0	0	0	0	0	0	0
	)	0	0	0	0	0	0	0	0	0	0	0	0
	+	0	0	0	0	0	0	0	0	0	0	0	0
	—|	0	0	0	0	0	0	0	0	0	0	0	0

A la vista de la misma podemos construir la siguiente tabla de referencias cruzadas:

Metavariables	Terminales por las que comienza
s	*A*, *B*, (
e	*A*, *B*, (
l'	+
l	*A*, *B*, (
p'	*A*, *B*, (
p	*A*, *B*, (

La memoria ocupada por la implementación del algoritmo de Warshall es:

$$\text{cardinal } (V) \times \text{cardinal } (V) = n^2$$

Podemos pensar que si este método se aplica a una gramática de cualquiera de los lenguajes habituales, resulte prohibitiva por la memoria necesaria. Se podría ahorrar memoria representando la matriz en forma de listas circulares ortogonales [23] o vectores Iliffe, si bien así se dificultaría la implementación de este algoritmo.

Nosotros hemos desarrollado un método que obvia estas dificultades que exponemos a continuación.

2.2.1.2.2. *Aplicación de la técnica Hash al cálculo del cierre transitivo*

Sea C un conjunto finito de elementos C_j

$$C = \{ C_j ;\ j = 1, \ldots m \}$$

Si le aplicamos una función F unívoca obtendremos en general una familia A de conjuntos

$$A = \{ A_i ;\ i = 1, \ldots, n \}$$

en la que cada A_i es tal que:

$$A_i = \{ C_j \mid C_j \in A_i \longleftrightarrow F(c_j) = x_i \}$$

Cada A_i es una clase módulo F, formado por todos los elementos de C que al aplicarles F producen el mismo x_i.

La función F recibe el nombre de función de"scattering" o función Hash [24].

Y diremos que los elementos de A_i son sinónimos (también en algunos casos que "colisionan").

En el caso de que sea $n = m$, podemos decir que F es biunívoca, en este caso no tendremos colisiones.

Para determinar automáticamente los terminales por los que puede comenzar cada metavariable, construiremos primero, haciendo uso de las funciones Hash, el diccionario de antecedentes de los caracteres terminales en el grafo de dependencia a izquierdas, es decir, determinaremos qué nodos preceden a los terminales dentro del grafo.

El segundo paso será buscar en este diccionario por qué terminales comienza cada metavariable, siendo obvio este proceso.

En el primer caso, comenzaremos aplicando una función F al alfabeto terminal A_t, con lo que obtendremos una familia A de conjuntos A_i constituidos cada uno por elementos terminales, seguidamente extenderemos iterativamente los A_i, haciendo que formen parte también de cada conjunto las metavariables relacionadas mediante R_I^+ con los elementos terminales del mismo, es decir:

Si existe una relación $\Sigma\ R_I\ A_{T_i}$

donde:

$$\Sigma \in A_n,\ A_{T_i} \in A_t$$

pondremos:

$$\Sigma \in A_i,\ \forall\ A_i \mid A_{T_i} \in A_t$$

y posteriormente, si existe una relación $\varphi\ R_I\ \Sigma$
donde:

$$\varphi,\ \Sigma \in A_n$$

pondremos:

$$\varphi \in A_i,\ \forall\ A_i \mid \Sigma \in A_i$$

Esta técnica nos permite abordar gramáticas de lenguajes usuales, debido a ser suficientemente óptima en tiempo y memoria.

Siguiendo nuestro ejemplo, formaríamos el diccionario de antecedentes:

$$\begin{array}{lllll} A_1 = \{ & A & p & p' & 1\ e\ s \} \\ A_2 = \{ & B & p & p' & 1\ e\ s \} \\ A_3 = \{ & (& 1 & p' & 1\ e\ s \} \end{array}$$

$$
\begin{array}{lllll}
A_4 = \{) & & & & \} \\
A_5 = \{ + & 1' & & & \} \\
A_6 = \{ -| & & & & \} \\
\quad (1) & (2) & & (3) &
\end{array}
$$

(1) Se obtiene por aplicación de una función Hash.
(2) Primera extensión.
(3) Extensiones sucesivas.

Podemos observar, por ejemplo, que la metavariable p' empieza por los terminales A, B, (.

La implementación del método sobre UNIVAC 9300 se presenta en el apéndice B.

2.2.1.2.3. *Fórmulas lineales de dígrafos*

Un dígrafo puede ser representado por un conjunto de fórmulas lineales en notación prefix, basándose en representar un arco por un operador aplicado a los símbolos de los nodos. Tomando como símbolo del operador un asterisco (*), un arco $<a, b>$ se representará por $* ab$ [21].

Si desde un nodo emana más de un arco, una fórmula representante de todos los arcos consiste en tantos operadores como arcos, seguidos por el símbolo del nodo desde el cual parten los arcos y seguidos por los símbolos de los nodos en los cuales terminan los arcos.

Estas fórmulas lineales pueden definirse recursivamente:

1) Un símbolo de un nodo es una fórmula.
2) Si A y A' son fórmulas, entonces $* AA'$ es una fórmula.
3) Fórmulas son aquellas entidades construidas con 1 y 2.

Una gramática que describa la sintaxis de las fórmulas lineales es:

$A_t = \{ * \} \cup N$; donde $N = \{ a, b, c, \ldots, k \}$ símbolos de nodo.

$A_n = \{ K$, nodo $\}$

Símbolo distinguido K

$P : K \rightarrow$ nodo
$K \rightarrow * KK$
nodo $\rightarrow n \mid n \in N$

Consecuencia inmediata de la definición es que las fórmulas lineales pueden cambiarse para constituir otras más complejas mediante la siguiente regla de sustitución:

> Si A es la fórmula lineal de un nodo y existe otra fórmula A' en la cual el símbolo de aquel nodo aparece, sustituir la fórmula A por el símbolo en A'

Las fórmulas lineales para el grafo de dependencia a izquierdas en 2.2.1.2. son:

$$\{ * se, * el, * 1p, * ** pAB (, * p' p, * 1' +,), -| \}$$

Sustituyendo en la fórmula de s la de e y posteriormente la de 1 obtenemos:

$$\{ * s * e * 1 *** pAB (, * p' p, * 1' +,), -| \}$$

Puesto que la regla de sustitución ni crea ni destruye operadores, todos los conjuntos representativos de un dígrafo contienen el mismo número de operadores y este número es igual al de arcos.

Considerada la fórmula lineal $S_1\ S_2\ \ldots\ S_t\ S_m$ donde los S_t son símbolos de nodos u operadores, si n_t denota el número de símbolos de nodo hasta el símbolo S_t inclusive y K_t el número de operadores, se cumplirá:

$$n_t \leq K_t \quad , i = 1, 2, \ldots, m\text{-}1$$
$$n_m = K_m + 1$$

por ejemplo en la gramática:

$$A ::= A\,a \mid a$$

lo que ya habíamos expresado de una forma intuitiva suprimiendo la recursividad a izquierdas de las reglas de la gramática.

Para que el algoritmo se termine es condición necesaria y suficiente que

$$\{\nexists A \in N \mid A\, I^{+} A\}$$

El determinismo lo asegurábamos por lectura y test de un carácter adelante, veamos esto de una forma más estructurada, recordemos las siguientes relaciones:

Final F definida sobre V:

$A\,F\,B <=> (\exists \varphi \in V^{*}, \exists \psi \in V^{*} \mid A ::= \varphi B \psi$ y $\psi \overset{*}{>\!\!-} \varepsilon)$

B es final de A, sí, y sólo si: $A\,F^{*}\,B$.

Siguiente S definida sobre V:

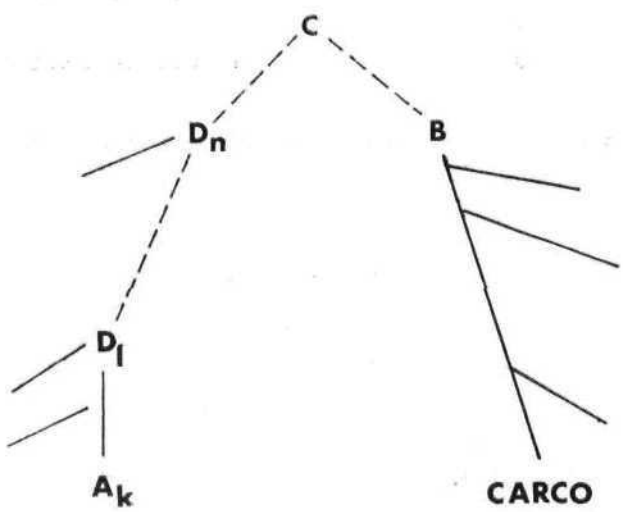
$$A\,S\,B <=> (\exists c \in N, \exists \varphi \in V^{*}, \exists \psi \in V^{*}, \mu \in V^{*} \mid c ::= \varphi A \mu B \psi \quad \text{y} \quad \mu \overset{*}{>\!\!-} \varepsilon)$$

Supongamos las reglas siguientes:

$$\begin{aligned} C &::= \ldots D_n\, B \ldots \\ D_n &::= \ldots D_{n-1} \\ D_{n-1} &::= \ldots D_{n-2} \\ D_1 &::= \ldots A_k \end{aligned} \qquad \text{y } B\,I^{*} \text{ CARCO}$$

(Siendo CARCO el carácter terminal que estamos analizando en la cadena.)

Lo que podíamos representar gráficamente por:



Que podemos representar teniendo en cuenta las relaciones definidas anteriormente por:

$$\begin{array}{lll} D_n\,F\,D_{n-1} \ldots F\,D_1\,F\,A_k & & D_n\,F^{*}\,A_k \\ D_n\,S\,B & \text{o sea} & D_n\,S\,B \\ B\,I^{*}\,\text{CARCO} & & B\,I^{*}\,\text{CARCO} \end{array}$$

De donde deducimos:

$$A_k\, F^{*-1}\, D_n, \qquad D_n\, SB,\, BI^*\ \text{CARCO}$$

o sea:

$$A_k F^{*-1}\ SI^*\ \text{CARCO}$$

A la nueva relación: $F^{*-1}\, SI^*$ la llamamos vecino: A_k vecino CARCO. Veamos un ejemplo:

$$X ::= BCa \mid BCd \mid BC \mid Bd$$
$$B ::= b\,B \mid b$$
$$C ::= a\,C\,b \mid ab$$

Analizar B = analizar b; escoger (analizar B; salir).

Tendremos la primera opción (analizar B), sí, y solamente si: $B\, I^+\, b$ como ocurre en nuestra gramática y además CARCO = b y la segunda, sí, y solamente si: B vecino CARCO; como los vecinos de B son a o d, tomemos esta segunda opción si CARCO = a o CARCO = d.

Evidentemente, para que exista determinismo se ha de cumplir:

$$\left.\begin{array}{l}\text{llamando INIC} = \{\ \forall c \mid B\, I^*\, c\ \} \\ \text{vecinos} = \{\ \forall e \mid A_k\ \textit{vecino}\ e\ \}\end{array}\right\}\ \text{INIC} \cap \text{VECINOS} = \Phi\ \text{(vacío)}$$

2.2.1.4. *Implementaciones*

La automatización de la escritura de compiladores mediante la utilización de diversos tipos de "compilador de compiladores" (por ejemplo el de Brooker y Morris), ha sido un tema de interés en estos últimos años.

M. Griffiths y M. Peltier [13], [14], combinando las ideas de M. Foster [15], [16] y D. E. Kunth [17] de su máquina sintáctica, e introduciendo algunos cambios implementaron un algoritmo para la elaboración de procesadores de lenguaje y por tanto para el desarrollo de nuevos lenguajes.

Veamos el proceso seguido en la implementación tomando un ejemplo [6]:

Supongamos la sintaxis de un bloque ALGOL de la que se ha eliminado la recursividad a izquierdas:

<BLOQUE> : : = *BEGIN* <LISTA DECLARACIONES>; <LISTA INST.> *END*

<LISTA DECLARACIONES> : : = *DECLARACION* |
DECLARACION; <LISTA DECLARACIONES>

<LISTA INST.> : : = *INSTRUCCION* |
INSTRUCCION; <LISTA INST.>

Donde los símbolos subrayados los consideramos elementos terminales. Si tenemos un bloque ALGOL correctamente construido:

BEGIN LISTA DECLARACIONES; LISTA INST. *END*

Una vez comprobado el *BEGIN* queda por comprobar LISTA DECLARACIONES; LISTA INST. *END*

Ahora bien, LISTA DECLARACIONES da dos alternativas:

DECLARACION; <LISTA INST.> *END*

Como además siempre sabemos la alternativa a tomar sin ambigüedad, podemos incluir las acciones semánticas en el proceso de análisis, desarrollándose la evaluación semántica a la vez que la sintáctica.

La gramática determinista obtenida anteriormente se puede expresar en la forma de un programa ALGOL formado por procedimientos (uno por cada metavariable);

```
procedure        BLOQUE ;
begin            COMPROBAR ("begin") ;
                 LISTA DECLARACIONES ;
                 LISTA INST. ;
                 COMPROBAR ('end')
end ;
procedure        LISTA DECLARACIONES ;
begin            COMPROBAR ('declaración') ;
                 COMPROBAR (';') ;
                 if CARCO = DECLA then LISTA DECLARACIONES
end ;
procedure        LISTA INST. ;
begin            COMPROBAR ('instrucción') ;
                 if CARCO = ';' then LISTA INST. ;
end ;
```

Hemos denotado por CARCO el carácter de la cadena terminal que está siendo analizado en un momento dado, DECLA es el (los) terminal(es) que puede(n) empezar una declaración.

Si DECLARACION e INSTRUCCION se desean considerar como elementos no terminales, tendríamos que escribir sus procedimientos respectivos, en la forma:

```
procedure        DECLARACION
begin
                 .
                 .
                 .
end;
procedure        INSTRUCCION
begin
                 .
                 .
                 .
end;
```

y sus llamadas en los procedimientos LISTA DECLARACIONES y LISTA INST. consistirían en sustituir:

COMPROBAR ('declaración') por DECLARACION y
COMPROBAR ('instrucción') por INSTRUCCION.

Cabe notar que cada metavariable se convierte en un procedimiento recursivo en el momento de su declaración y una llamada al mismo en el momento de su uso. La "comprobación" de cada

carácter terminal lleva consigo también el paso al carácter terminal siguiente. Cada carácter terminal sirve para decidir, mediante instrucciones condicionales, la alternativa a tomar.

Con la estructuración expuesta, es posible incorporar la técnica de las macroinstrucciones a la escritura de analizadores, objetivo este del trabajo de Griffiths.

El problema de la recursividad se resuelve haciendo uso de una pila (lista LIFO).

Los macros utilizados en la implementación son:

Call (metavariable)

Goto (metavariable)

Return

Fault

Decide (lista de caracteres terminales, dirección simbólica alternativa)

Check (carácter básico)

La entrada a una rutina (metavariable) se realiza mediante la macro "Call", la cual coloca la dirección de vuelta en una pila.

"Return" transfiere control a la dirección de la cabeza de la pila y posteriormente la "decapita".

"Goto" se utiliza en lugar de "Call" por ser para ciertas condiciones más eficiente, su trabajo es transferir control sin almacenar la dirección de vuelta.

"Decide" transfiere control a una dirección simbólica dependiendo de que el carácter que se está examinando, esté en la lista de caracteres terminales especificada en la línea "Decide".

Una aplicación real de todo lo expuesto anteriormente vamos a desarrollarla en los próximos capítulos, que son exposición de nuestro trabajo.

2.2.2. *Representación de las informaciones*

Entendemos por informaciones los objetos que aparecen designados en el lenguaje de programación.

En nuestro caso son de dos tipos:

- Direcciones absolutas de la máquina simulada que expresamos por: (DIRECCION).
- Constantes que nosotros escribimos: = NUMERO.

Son los dos únicos objetos que aparecen en nuestro lenguaje. Damos a continuación una serie de definiciones:

Una memoria M es un conjunto de palabras:

$$M = \{ m_1, \ldots m_n \}$$

Los posibles contenidos de estas palabras son un conjunto V de valores.

$$V = \{ v_1, \ldots v_k \}$$

Un subconjunto D del anterior lo llamamos conjunto de direcciones:

$$D \subset V \; ; D = \{ d_1, \ldots d_n \}$$

Existen una serie de funciones que nos relacionan estos conjuntos:

función de direccionamiento: $\alpha : D \rightarrow M$
función contenido: $c : M \rightarrow V$

Nosotros, en nuestro lenguaje, establecemos la siguiente representación:

$$\text{objeto del lenguaje} \xrightarrow[\text{representación}]{} \text{objeto en memoria}$$

y en concreto:

$$(\text{DIRECCION}) \xrightarrow{\text{representación 1}} d_k \in D$$

$$= \text{NUMERO} \xrightarrow{\text{representación 2}} v_k \in V$$

La representación 1 se realiza mediante un diccionario (tabla) que en nuestro caso viene dado por un cálculo d_k = cálculo [(DIRECCION)].

La representación 2 se realiza mediante un cálculo y la función contenido:

$$v_k = c \left[\text{cálculo } [= \text{NUMERO}] \right]$$

Valor al que acudimos en la forma:

$$v_k = c \left[\alpha \; [\text{cálculo'} \; [= \text{NUMERO}]] \right]$$

Donde cálculo asocia una dirección a la representación interna del número que obviamente está contenida en una palabra.

NOTA: Entendemos por palabra una zona de memoria direccionable como un todo.

3. NUESTRA IMPLEMENTACION

3.1. LENGUAJES QUE INTERVIENEN EN EL PROCESO

3.1.1. *Introducción*

Es característico de los metatraductores, la independencia con relación al lenguaje fuente y objeto; objetivo este, alcanzado por nuestra implementación.

Para la realización de un proceso de traducción, se fijaron en nuestro traductor los siguientes lenguajes:

a) Lenguaje fuente: Una modificación del lenguaje para niños 7013 [18], el lenguaje modificado por nosotros lo denominaremos 9013, y tiene las siguientes características:

Operará en una máquina teórica, que consta de un acumulador (*AC*), donde residen las habilidades aritméticas tradicionales; una memoria con 100 palabras numeradas de 00 a 99, un contador de direcciones (*CD*), el cual mantiene la dirección de la próxima instrucción a ejecutarse, un periférico lector de fichas y una impresora.

Las instrucciones, expresan en sí mismas la acción que la máquina debe realizar, presentándose seguidamente:

Entrada/Salida:

LEER EL CONTENIDO DE LA UNIDAD DE ENTRADA EN LA POSICION (XX)
ESCRIBIR EN LA UNIDAD DE SALIDA EL CONTENIDO DE LA POSICION (XX)

Manipulación de datos:

DEPOSITAR EL CONTENIDO DEL ACUMULADOR EN LA POSICION (XX)
TRASLADAR AL ACUMULADOR EL CONTENIDO DE LA POSICION (XX)
TRASLADAR AL ACUMULADOR EL NUMERO =XXXXXX

Aritméticas:

SUMAR AL ACUMULADOR EL CONTENIDO DE LA POSICION (XX)
SUMAR AL ACUMULADOR EL NUMERO =XXXXXX
RESTAR DEL ACUMULADOR EL CONTENIDO DE LA POSICION (XX)
RESTAR DEL ACUMULADOR EL NUMERO =XXXXXX
MULTIPLICAR EL ACUMULADOR POR EL CONTENIDO DE LA POSICION (XX)
MULTIPLICAR EL ACUMULADOR POR EL NUMERO =XXXXXX
DIVIDIR EL ACUMULADOR POR EL CONTENIDO DE LA POSICION (XX)

Control:

PONER EN EL CD LA DIRECCION :XX
SIEL CONTENIDO DE (XX) > (AC) PONER EN EL CD LA DIRECCION :XX
SIEL CONTENIDO DE (XX) = (AC) PONER EN EL CD LA DIRECCION :XX
SIEL CONTENIDO DE (XX) < (AC) PONER EN EL CD LA DIRECCION :XX
SIEL CONTENIDO DEL AC > 0 PONER EN EL CD LA DIRECCION :XX

HACERXX VECES LAS XX INSTRUCCIONES SIGUIENTES
PARAR
FIN

b) Lenguaje objeto: Adoptamos el lenguaje máquina del UNIVAC 9300.

c) Lenguaje usado en la implementación: El lenguaje de Macros definido en 3.2.2, junto con el lenguaje ensamblador del UNIVAC 9300.

3.1.2. Elaboración de las gramáticas

La podemos dividir en dos etapas, una primera "de limpieza", y una posterior en la que impondremos las cuatro condiciones de Kunth.

Diremos que una gramática es limpia cuando no tiene símbolos inútiles, entendiendo por tales aquellos que son parásitos inaccesibles.

Un símbolo α ($\alpha \in A_n$) es parásito de una gramática si no origina ninguna cadena terminal.

Un símbolo α ($\alpha \in V$) es inaccesible en una gramática si no existe ningún camino desde el axioma hasta ese símbolo [19] en el grafo de dependencia simple de la gramática.

En esta fase se suelen utilizar los siguientes algoritmos:

A) *Eliminación de símbolos inaccesibles*

a) Marcar el axioma de la gramática en todas las producciones donde aparezca.

b) Buscar una regla de la gramática $A \longrightarrow \varphi$, $A \in A_n$, $\varphi \in V^*$, donde A está marcado y φ no está completamente marcado.

c) Si encontramos tal regla, marcar todos los símbolos de φ, tanto en φ como en todas las partes izquierdas donde aparecen.

Ir a *b*).

d) Si no encontramos tal regla (paso *b*), la búsqueda ha terminado y los símbolos no marcados son inaccesibles.

B) *Eliminación de símbolos parásitos*

a) Marcar los símbolos terminales y $\varnothing$ (Cadena vacía) en todas las producciones de la gramática (G).

b) Buscar en la G una regla $A \longrightarrow \varphi$ ($A \in A_n$, $\varphi \in V^*$) en la que A no está marcada y φ está todo marcado.

c) Si encontramos esta regla, marcar A en todas las producciones de G.

Ir a *b*).

d) Si no encontramos tal regla, la búsqueda ha terminado y los símbolos no marcados son los parásitos.

Cabe indicar que estos algoritmos de limpieza son automatizables.

En cambio, en la fase de conversión de la gramática libre de contexto, a determinista en el sentido de Kunth, no es posible la automatización.

Usualmente se utilizan procedimientos para esta transformación, los cuales presentaremos aquí esquemáticamente.

a) Aplicación de la primera condición.

Eliminación de la recursividad a izquierdas en los casos tales como:

$$A \longrightarrow A\,\varphi$$
$$A \longrightarrow a$$

donde:

$$A \in A_n,\ \varphi \in V^*,\ a \in A_t$$

una forma obvia es sustituir la primera producción por:

$$A \rightarrow aA\varphi \mid a$$

generándose el mismo lenguaje en ambos casos.

En general las reglas recursivas son usadas para indicar la repetición de un elemento un número indefinido de veces. Por ejemplo:

<programa> : : = <instrucción> <continuación de programa>
<continuación de programa> : : = ∅ | <continuación de programa> <instrucción>

la recursividad a izquierdas la eliminamos haciendo:

<continuación de programa> : : = ∅ | <instrucción> <continuación de programa>

Teniendo mucho cuidado que no ocurra

$$\text{<instrucción>} \overset{*}{\rightarrow} \varnothing$$

ya que entonces violaríamos la primera y cuarta condición de Knuth.

b) Aplicación de la segunda condición (caracteres iniciales disjuntos).

La regla usada casi exclusivamente para obligar a la gramática es la factorización, si tenemos:

$$A \rightarrow a\varphi_1$$
$$A \rightarrow a\varphi_2$$

convertimos esta regla en:

$$A \rightarrow aX \qquad A \text{ y } X \in A_n$$
$$X \rightarrow \varphi_1 \qquad a \in A_t$$
$$X \rightarrow \varphi_2 \qquad \varphi_1 \text{ y } \varphi_2 \in V^*$$

Un caso no tan obvio sería:

$$A \rightarrow B\varphi_1 \qquad A, B, C \in A_n$$
$$A \rightarrow C\varphi_2 \qquad a \in A_t$$
$$B \rightarrow a\varphi_3$$
$$C \rightarrow a\varphi_4 \qquad \varphi_i,\ i = 1, 2, 3, 4 \in V^*$$

Lo que se aplica sistemáticamente en la elaboración de casos como éste, es "promover" a en la forma siguiente:

$$A \rightarrow aX$$
$$X \rightarrow \varphi_3\varphi_1$$
$$X \rightarrow \varphi_4\varphi_2$$

Lo que implica la creación de una nueva clase sintáctica X.

c) Aplicación de la tercera condición.

Presentemos un ejemplo:

1) <body> : : = <ecuación list> <ecuación>
2) <ecuación list> : : = ∅ | <ecuación> <ecuación list>
3) <ecuación> : : = <expresión> = <expresión>

La clase sintáctica <ecuación list> (producción 2) cuando se toma la opción cadena vacía ("se hace transparente") y va seguida por <ecuación> en <body> (producción 1) que coincide con la segunda alternativa de la producción 2, por tanto no se cumple la condición 3. Para resolverlo conviene introducir un carácter terminal, quedando entonces la gramática:

1) <body> : : = <ecuación> <ecuación list>

2) <ecuación list> : : = ∅ | *T* <ecuación> <ecuación list>

3) <ecuación> : : = <expresión> = <expresión>

(supuesto era *T* el carácter terminal a resaltar).

Con esta nueva versión de la gramática desaparece el problema. Observemos que además hemos alterado el orden en la producción 1.

d) Aplicación de la cuarta condición.

Supongamos que tenemos la siguiente gramática:

1) <Programa> : : = <Instrucción> <Cont. programa>

2) <Cont. programa> : : = ∅ | <Instrucción> <Cont. programa>

3) <Instrucción> : : = ∅ | etc. ...

Si <Instrucción> puede originar (producción 3) la cadena vacía la clase sintáctica <Cont. programa> origina de dos formas distintas la cadena vacía, violándose la cuarta condición de Knuth.

Una forma de resolverlo sería establecer un final de instrucción; por ejemplo, punto y coma: ';'

1) <Programa> : : = <Instrucción> <Cont. programa>

2) <Cont. programa> : : = ; <Nueva clase>

3) <Nueva clase> : : = ∅ | <Instrucción> <Cont. programa>

4) <Instrucción> : : = ∅ | etc. ...

La clase sintáctica <Nueva clase> (producción 3) ya no origina de dos formas distintas la cadena vacía, ya que <Cont. programa> no es nunca vacío.

Casos análogos a los expuestos se presentaron en la elaboración de nuestra gramática fuente. resolviéndose con procedimientos parecidos.

3.1.3. Lenguaje fuente

Escribimos a continuación la gramática determinista del lenguaje 9013 que hemos utilizado en nuestro trabajo. En ella hacemos uso de una notación alternativa de la BNF; las clases sintácticas se han representado con su nombre en minúsculas, los símbolos terminales aparecen en mayúsculas, y las distintas alternativas de la parte derecha de las producciones se escriben en líneas consecutivas.

```
instruc : : = esinst
              mdar
              mdinst
              arct
              arinst
              ctinst

esinst : : = LEER    ck
             ESCRIBIR    ck

ck : : = cadena ct

mdar : : = D mdarcon

mdarcon : : = EPOSITAR    ck
              IVIDIR    ck

mdinst : : = TRASLADAR cadena contins
```

arct : : = S cu

cu : : = UMAR cadena contins
IEL ck oprela contct

contins : : = ct
cn

ct : : = ∅
(dirección)

cn : : = = número

cs : : = : dirección

arinst : : = RESTAR cadena contins
MULTIPLICAR cadena contins

ctinst : : = P cr
FIN
HACER número cadena número cadena

cr : : = ARAR
ONER cadena cs

contct : : = 0 cadena cs
(AC) cadena cs

oprela : : = <
=
>

direcc : : = dígito dígito

cadena : : = caract contcad

contcad : : = ∅
caract contcad

caract : : = A
B
C
D
.
.
.
Z
espacio

número : : = dígito contnume

contnume : : = ∅
dígito contnume

dígito : : = 0
1
2
3
.
.
.
9

3.2. NUESTRO TRADUCTOR

3.2.1. Técnica de análisis

Ya hemos expuesto anteriormente que el procedimiento de análisis sintáctico utilizado es determinista, descendente y sin vuelta atrás, lo que nos obliga a definir el lenguaje fuente mediante una gramática LL (1), siendo el paso de ésta al analizador sintáctico completamente algoritmizable como veremos en el punto 3.2.3.

Esto supone una restricción bastante considerable dado que obligamos a una elaboración previa de la gramática por motivos de deficiencia de nuestro algoritmo de construcción automática del analizador sintáctico. Creemos, y a ello encaminamos en este momento nuestro esfuerzo, que el algoritmo antedicho debería dejar mayor libertad en la escritura de la gramática (sería suficiente que fuera tipo 2 y no ambigua) y que en función de las características de ésta escogiera el procedimiento de análisis sintáctico más adecuado (descendente o ascendente con sus diversos tipos). La insistencia que se dedica en este trabajo a las relaciones impuestas por una gramática sobre sus símbolos es el punto de partida para esta ampliación del trabajo que nos permitirá una clasificación y elaboración (si necesario) de la gramática fuente para dar lugar al analizador más adecuado. Evidentemente, todo este proceso sería transparente para el usuario, que se ocuparía solamente de la descripción estructural del lenguaje y del significado que él otorga a cada una de sus producciones. Para la descripción del significado haría falta un lenguaje standard que facilitara esta tarea y cuyas bases exponemos en el punto 3.4.3.

3.2.2. Lenguaje de macro-instrucciones para el tratamiento de la fase sintáctica

El lenguaje recursivo implementado por nosotros sobre UNIVAC 9300 consta de los siguientes macros [12]:

1) Elemento de entrada-salida.

Macro LEER

Misiones:

a) Imprimir el código objeto correspondiente a la sentencia precedente, junto con la dirección en memoria donde se ubica, según el formato:

dirección código objeto

- Dirección consta de cuatro dígitos hexadecimales.
- Código objeto es variable en longitud.

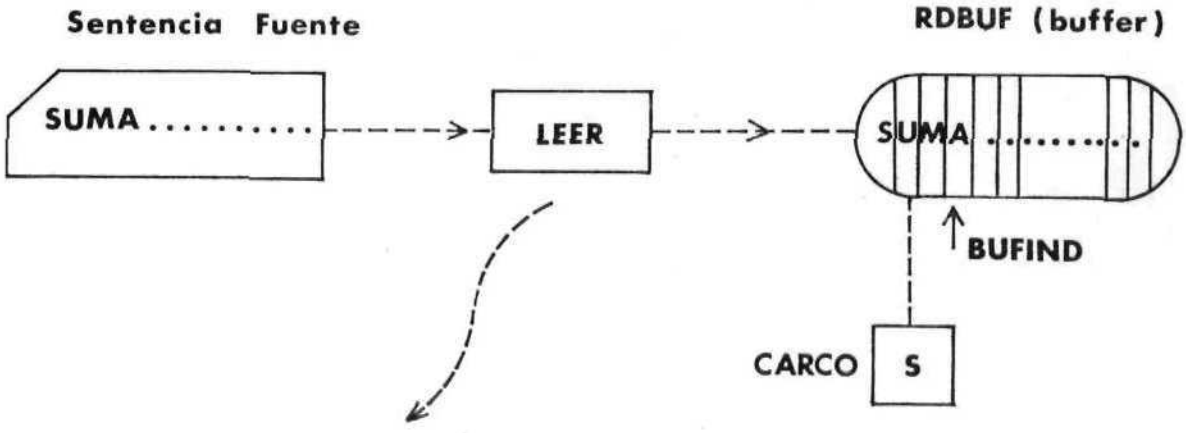
b) Leer una sentencia fuente (soportada en ficha) y colocar el primer carácter en un área denominada CARCO. Actualiza un indicador denominado BUFIND, que contiene la dirección del carácter de la sentencia que se va a analizar.

c) Imprimir la sentencia fuente leída, junto con un número de sentencia, según el formato:

secuencia sentencia fuente

- Secuencia consta de cuatro dígitos decimales.
- Sentencia fuente del lenguaje.

ESQUEMATICAMENTE:




```
2D80   D20510002850
2D86   D20510181000
2D8C   D20510002B56
2D92   D205100C1000
2D98   D20510121000
2D9E   D2051000100C
2DA4   F955100010184720FFFF
```

```
0001   TRASLADAR AL ACUMULADO...
0002   DEPOSITAR EL CONTENIDO...
0003   TRASLADAR AL ACUMULA...
0004   DEPOSITAR EL CONTENI...
0005   DEPOSITAR EL CONTEN...
0006   TRASLADAR AL ACUMU...
0007   SIEL CONTENIDO DE...
0008   SUMAR AL ACUMULAD...
```

Implementación:

```
        PROC    P,0
LEER    NAME
        MVC     ARIM,BLAN
        CP      CON,=X'0F'
        BC      8,GET
        MVC     M,=XL4'0'
        STH     12,SALR12
        STH     8,SALR15
        LH      8,DPROGFU
        SH      8,DOS
        MVC     RESER,0(8)
        MVO     M(2),RESER(1)
        MVO     M+2(2),RESER+1(1)
        NC      RESER,=X'0F0F'
        MVC     M+1(1),RESER
        MVC     M+3(1),RESER+1
        TR      M,T77
        MVC     ARIM+4(4),M
```

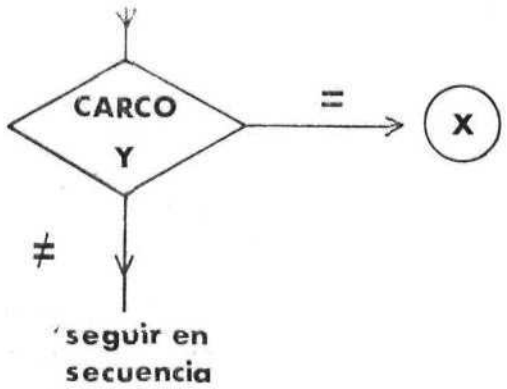
```
                LH      8,0(,8)
                LH      15,=Y(ARIM+10)
MVC             MVC     RESER,=X'0000'
                MVC     M(1),0(8)
                MVO     RESER,M(1)
                NI      M,X'0F'
                MVC     RESER+1(1),M
                MVC     0(2,15),RESER
                AH      8,=Y(1)
                AH      15,DOS
                CH      8,SALR12
                BC      4,MVO
                TR      ARIM+10(121),T77
                PUT     IMPRE,ARIM
                LH      8,SALR13
GET             GET     FICH,RDBUF
                MVC     CARCO(1),RDBUF
                MVC     BUFIND,TEMP
                AP      CON,=X'1F'
                LH      13,DPROGFU
                MVC     0(2,13),CON
                AH      13,=H'2'
                STH     12,0(,13)
                AI      DPROGFU,X'04'
                MVC     ARIM,BLAN
                MVC     ARIM+45(4),MASK
                ED      ARIM+45(4),CON
                MVC     ARIM+52(80),RDBUF
                PUT     IMPRE,ARIM
                LH      10,BASEG
                LH      11,BASE1
                B       *+130
                DC      X'999F0000'
PROGFUDI        DC      120X'00'
DPROGFU         DC      Y(PROGFUDI)
                END
```

2) Elementos de control.

Macro DECIDE X,Y

Misión: Comparar el carácter actual situado en CARCO, con el segundo argumento (*Y*). En caso de igualdad se transfiere control al primer argumento (*X*), en caso contrario se sigue en secuencia.

ESQUEMATICAMENTE:



Implementación:

```
        PROC    P,2
DECIDE  NAME
        CLI     CARCO,P(2)
        BE      P(1)
        END
```

Macro RUTINA X

Misión: Especifica que su argumento es la dirección simbólica de la próxima instrucción ejecutable.

Implementación:

```
        PROC    P,1
RUTINA  NAME
P(1)    EQU     *
        END
```

Macro ENTRAR X

Misión: Transferir control a la dirección especificada por su argumento.

Implementación:

```
        PROC    P,1
ENTRAR  NAME
        B       P(1)
        END
```

Macro VOLVER

Misión: Delimitar lógicamente una rutina, para lo que lleva incorporada una instrucción NOP (de no operación).

Implementación·

```
        PROC    P,0
VOLVER  NAME
        NOP
        END
```

Macro LLAMA X

Dado que es necesario dar una solución al mecanismo de recursividad utilizado tanto en este macro como en SALIR (descrito a continuación), se ha diseñado una pila (lista LIFO) cuya misión será guardar las direcciones de transferencia manipuladas por estos macros. Originalmente la pila consta de 50 niveles, pudiéndose ampliar fácilmente cuando sea necesario.

Misión: Almacenar en la pila diseñada la dirección de la instrucción siguiente a este macro. Transferir control a la dirección especificada por su argumento (*X*). Se prevé la situación de "overflow" de la pila, imprimiéndose en este caso el mensaje:

OVERFLOW EN LA PILA DE RECURSIVIDAD

y deteniéndose el proceso.

Implementación:

```
        PROC    P,1
LLAMA   NAME
        BAL     13,ETLLAMA
        MVC     0(2,8),*+10
        B       P(1)
        DC      Y(*+2)
        END
```

complementándose en el programa analizador con:

```
ETLLAMA AI      POINT+2,X'04'
        CLC     POINT(4),FINPILA
        BE      OVERF
        MVC     POINT1(2),POINT+2               SAVEAREA
        AI      POINT1,X'02'
        LH      8,POINT1
        B       0(,13)
OVERF   MVC     ARIM,BLAN
        MVC     ARIM+20(33),=C'OVERFLOW EN LA PILA DE RECURSIVIDAD'
        PUT     IMPRE,ARIM
        HPR     X'2222'
```

Macro SALIR

Misión: Transferir control a la última dirección de la pila y decapitarla. Se prevé la posibilidad de "underflow", imprimiéndose en este caso el mensaje:

UNDERFLOW EN LA PILA DE RECURSIVIDAD

y deteniéndose el proceso.

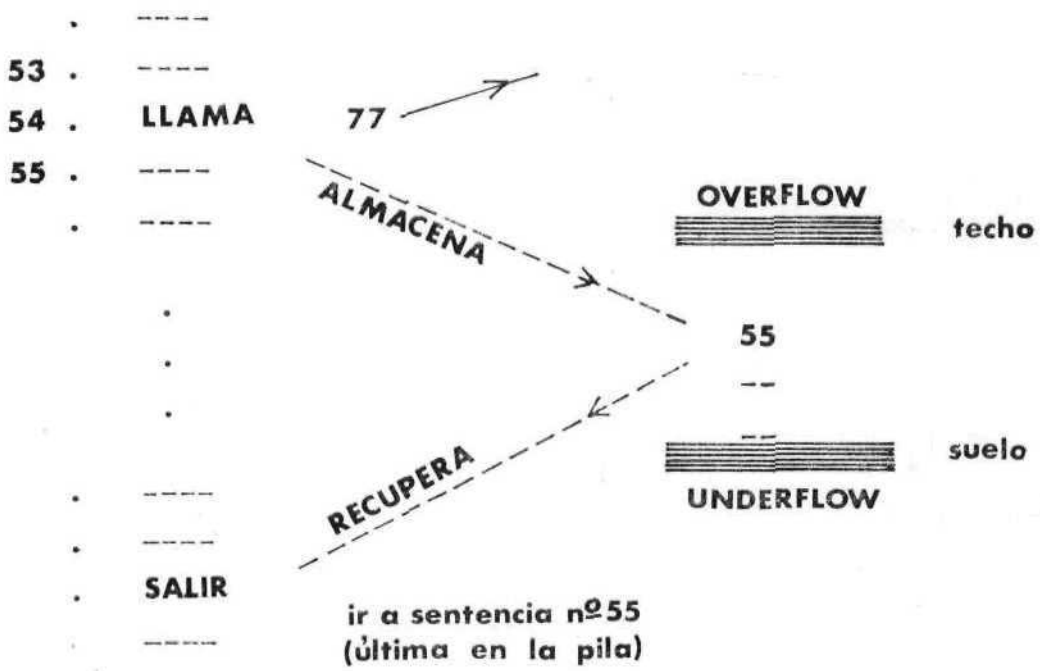
Implementación:

```
                PROC  P,0
SALIR           NAME
                B     SALIRSUB
                END
```

complementándose en el analizador con:

```
SALIRSUB  MVC   POINT2(4),POINT
          LH    8,POINT3
          SH    8,CUA
          STH   8,POINT3
          CLC   PRINC(4),POINT
          BH    UNDFL
          B     POINT2
UNDFL     MVC   ARIM,BLAN
          MVC   ARIM+20(36),=C'UNDERFLOW EN LA PILA DE RECURSIVIDAD'
          PUT   IMPRE,ARIM
          HPR   X'1111'
```

Las funciones de los elementos LLAMA y SALIR pueden representarse esquemáticamente de la manera siguiente:



3) Elemento de avance sobre la cadena terminal.

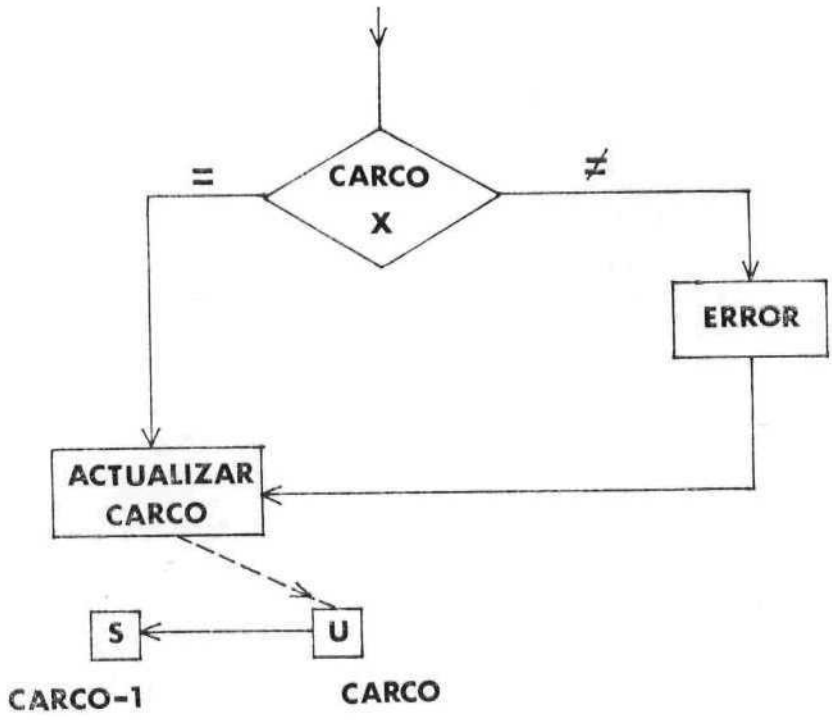
Macro TEST X

Misión: Comparar el carácter actual (en CARCO) con el argumento (X). En caso de desigualdad se imprimirá el mensaje:

CARACTER × ILEGAL EN EL CONTEXTO

donde × es el carácter almacenado en CARCO en ese momento. Si son iguales no se imprime ningún mensaje. En ambos casos se extrae el carácter siguiente en la sentencia fuente, actualizándose con él la memoria CARCO, habiendo pasado previamente a CARCO-1 su contenido anterior.

ESQUEMATICAMENTE:



Implementación:

```
        PROC    P,1
TEST    NAME
        MVC     ETTEST1,*+10
        BAL     13,ETTEST1
        CLI     CARCO,P(1)
        END
```

complementándose en el analizador con:

```
ETTEST1 CLI     CARCO,*—*
        BE      *+8
        BAL     9,FALSO
        AI      BUFIND,X'01'
        MVC     CARCO—1(1),CARCO
        LH      8,BUFIND
        MVC     CARCO,0(8)
        B       4(,13)
FALSO   MVC     ARIM,BLAN
        MVC     ARIM+20(8),=C'CARACTER'
        MVC     ARIM+30(1),CARCO
```

```
MVC     ARIM+32(21),=C'ILEGAL EN EL CONTEXTO'
PUT     IMPRE,ARIM
B       0(,9)
```

3.2.3. Algoritmo de construcción automática de un analizador

Definidos así estos macros, podemos escribir un algoritmo que construya automáticamente un analizador para una gramática dada (del tipo LL (1).

Para cada producción de la gramática se procede de la manera siguiente:

1) Parte izquierda: se genera el macro RUTINA, cuyo argumento será dicha parte izquierda.
2) Parte derecha:

a) No hay alternativas.

Para cada elemento de la cadena se genera:

a.1) Si es terminal y es el último de la cadena:

TEST elemento
SALIR

a.2) Si es terminal, pero *no* es el último:

TEST elemento

a.3) Si es clase sintáctica y es el último elemento de la cadena:

ENTRAR elemento

a.4) Si es clase sintáctica, pero *no* es el último elemento:

LLAMA elemento

b) Hay alternativas.

Se genera *para cada alternativa j* un:

DECIDE etiqueta *j*, terminal *i*

para cada terminal *i* que pueda comenzar la alternativa *j*.

Seguidos de:

etiqueta *j* EQU *

Y de la generación correspondiente a su cadena, según *a*), *para cada altenativa j*.

c) Al fin de la generación de la parte derecha se genera:

VOLVER

Pongamos un ejemplo para fijar ideas.

Sea la gramática:

```
sent  ::= id = exp
exp   ::= id + id
id    ::= A
          B
          C
```

un analizador para esta gramática sería:

```
                RUTINA      SENT
                LLAMA       ID
                TEST        =
                ENTRAR      EXP
                VOLVER
                RUTINA      EXP
                LLAMA       ID
                TEST        +
                ENTRAR      ID
                VOLVER
                RUTINA      ID
                DECIDE      L1, A
                DECIDE      L2, B
                DECIDE      L3, C
L1              EQU         *
                TEST        A
                SALIR
L2              EQU         *
                TEST        B
                SALIR
L3              EQU         *
                TEST        C
                SALIR
                VOLVER
```

En el caso de alternativas en la parte derecha de una producción, tenemos que decidir (DECIDE) sobre los caracteres terminales por los que puede comenzar cada alternativa; esto implica que para las alternativas que comienzan con una metavariable habrá que decidir sobre cada uno de los caracteres terminales que de forma transitiva dependan a izquierdas de dicha metavariable; en suma, deberemos tener resuelto el cierre transitivo del grafo de dependencia a izquierdas. Este paso previo puede ser obtenido por uno cualquiera de los métodos expuestos en 2.2.1.2.1. y 2.2.1.2.2.

Una implementación del algoritmo de construcción automática de un analizador, realizada en un UNIVAC 9300, se presenta en el Apéndice D.

3.2.4. Cálculo del cierre transitivo del grafo de dependencia a izquierdas de la gramática de nuestro lenguaje fuente

3.2.4.1. *Aplicación del algoritmo de Warshall*

La determinación de los elementos terminales por los que comienza cada metavariable fue abordada en nuestro trabajo mediante las dos técnicas mencionadas en 2.2.1.2.1. y 2.2.1.2.2., respectivamente.

La aplicación del algoritmo de Warshall se hizo manualmente, incluyéndose seguidamente la matriz del grafo de dependencia a izquierdas, una vez calculado el cierre transitivo.

Hemos considerado, por claridad, a las metavariables "caract" y "dígito" como elementos terminales; se distinguen así mismo los terminales *A, L, E, D, T, S, R, M, P, H, F, I*, (, =, :, 0, O, U, <, >, por ser los que aparecen en las relaciones de dependencia a izquierdas.

Matriz del grafo:

	instruc	esinst	ck	mdar	mdarcon	mdinst	arct	cu	contins	ct	cn	cs	arinst	ctinst	cr	contct	direcc	oprela	contcad	cadena	número	contnume
instruc	0	1	0	1	0	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
esinst	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ck	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
mdar	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mdarcon	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mdinst	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
arct	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
cu	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
contins	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ct	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
cn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
cs	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
arinst	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ctinst	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
cr	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
contct	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
direcc	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
oprela	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
contcad	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
cadena	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
número	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
contnume	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	A	L	E	D	T	S	R	M	P	H	F	I	(	=	:	O	0	U	DIGITO	CARACT	>	<
A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
L	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
E	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
=	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
O	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
U	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
DIGITO	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
CARACT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
>	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
<	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

3.2.4.2. *Cálculo automatizado*

Como mencionamos en 2.2.1.2.2., la implementación de la técnica Hash constituye el Apéndice B, la ejecución del programa para la gramática fuente da como resultado las siguientes tablas:

a) por cada metavariable, los caracteres terminales y no terminales que la comienzan;

b) metavariables que comienzan por cada terminal;

c) terminales por los que comienza cada clase;

constituyentes del Apéndice E.

3.3. CONSTRUCCION DEL ANALIZADOR PARA LA GRAMATICA FUENTE

La aplicación del algoritmo presentado en 3.2.3. fue realizada en nuestro trabajo paralelamente mediante la utilización del programa del Apéndice D y la construcción manual, obteniéndose por ambos medios analizadores semánticamente idénticos, si bien la construcción manual permite una mejor optimización.

En el Apéndice F se presentan ambos, junto con la aplicación a unas sentencias fuente.

3.4. TRATAMIENTO SEMANTICO

3.4.1. Introducción

Expusimos ya en el párrafo 2.1.3.4. cómo introducíamos las acciones semánticas asociadas con cada reducción. El hecho de que nuestro traductor acepte como dato la gramática fuente G_F la gramática del lenguaje objeto G_O y G_F/G_O; una descripción del lenguaje fuente en el lenguaje objeto, lleva implícitamente la descripción de la semántica de cualquier programa escrito en el lenguaje fuente.

En toda instrucción del lenguaje fuente, pueden considerarse dos aspectos:

a) Hace referencia a un objeto (o varios) del lenguaje fuente: dirección, contenido de una palabra, un registro, etc.

b) Modifica los objetos realizando operaciones sobre ellos: accede al contenido, modifica éste... (podemos incluir como objeto el contador de emplazamiento, lo que daría cuenta de las instrucciones de control).

Los objetos de un lenguaje de programación son detectados en los programas fuente mediante el análisis lexicográfico de éste que establece la correspondencia:

objeto en lenguaje fuente $\longrightarrow$ objeto en lenguaje objeto

El paso de uno a otro depende de los criterios de representación escogidos.

Análogamente, las modificaciones que especifica una instrucción sobre los objetos a que hace referencia se expresan de una forma cómoda y natural en el lenguaje fuente, su representación en

el lenguaje objeto ha de tener en cuenta las características de éste (dependientes de la máquina objeto).

El ejemplo escogido del lenguaje fuente y objeto se presta a un tratamiento fácil dada la correspondencia biunívoca entre objetos y tratamientos en ambos lenguajes. En un caso no tan trivial debería hacerse un estudio más a fondo de este problema de representación en sus aspectos *a*) y *b*) que complicaría grandemente la implementacion: en definitiva, la descripción G_F/G_O

Los aspectos *a*) y *b*) han dado lugar clásicamente a los dos campos:

- Representación de informaciones (estructura de la información);
- Análisis sintáctico/generación;

tratados ampliamente en la bibliografía.

Nosotros hemos insistido en el punto G_F/G_O que realmente deberíamos ampliar a L_F/L_O (descripción de lenguajes en sus dos aspectos: sintáctico y semántico y no sólo en el sintáctico), lo que ha sido trivial dada la sencillez del tratamiento semántico.

3.4.2. Esquema general del proceso

Dado que el algoritmo de análisis utilizado es determinista, existe una correspondencia biunívoca entre sentencias y parses; esta correspondencia produce un conjunto de clases de sentencias. Dos sentencias particulares pertenecen a la misma clase si su árbol de análisis (parse) es el mismo.

Vemos pues que el analizador particiona en clases el conjunto de sentencias fuente, esto implica que tendremos tantas clases de generación de código como clases de sentencias fuente, dado que códigos fuente y objeto están relacionados por la relación *semántica invariable;* un traductor es la implementación de esta relación. Pero esto precisamente simplifica nuestra labor puesto que permite tratar a cada clase por separado.

Cuando se reconoce una clase particular se realizarán, pues, las acciones semánticas convenientes, estas acciones en nuestra implementación consisten en depositar información en dos listas denominadas GUIA y TERMGUIA.

En GUIA tendremos al final del análisis de una sentencia fuente, una secuencia de entradas en la tabla denominada GFGO, la cual relaciona cada clase de sentencias fuente con su código objeto correspondiente, y que veremos en detalle en 3.4.3; la mencionada secuencia de entradas es la implementación de la secuencia de números de reducción que el analizador encontró en el proceso.

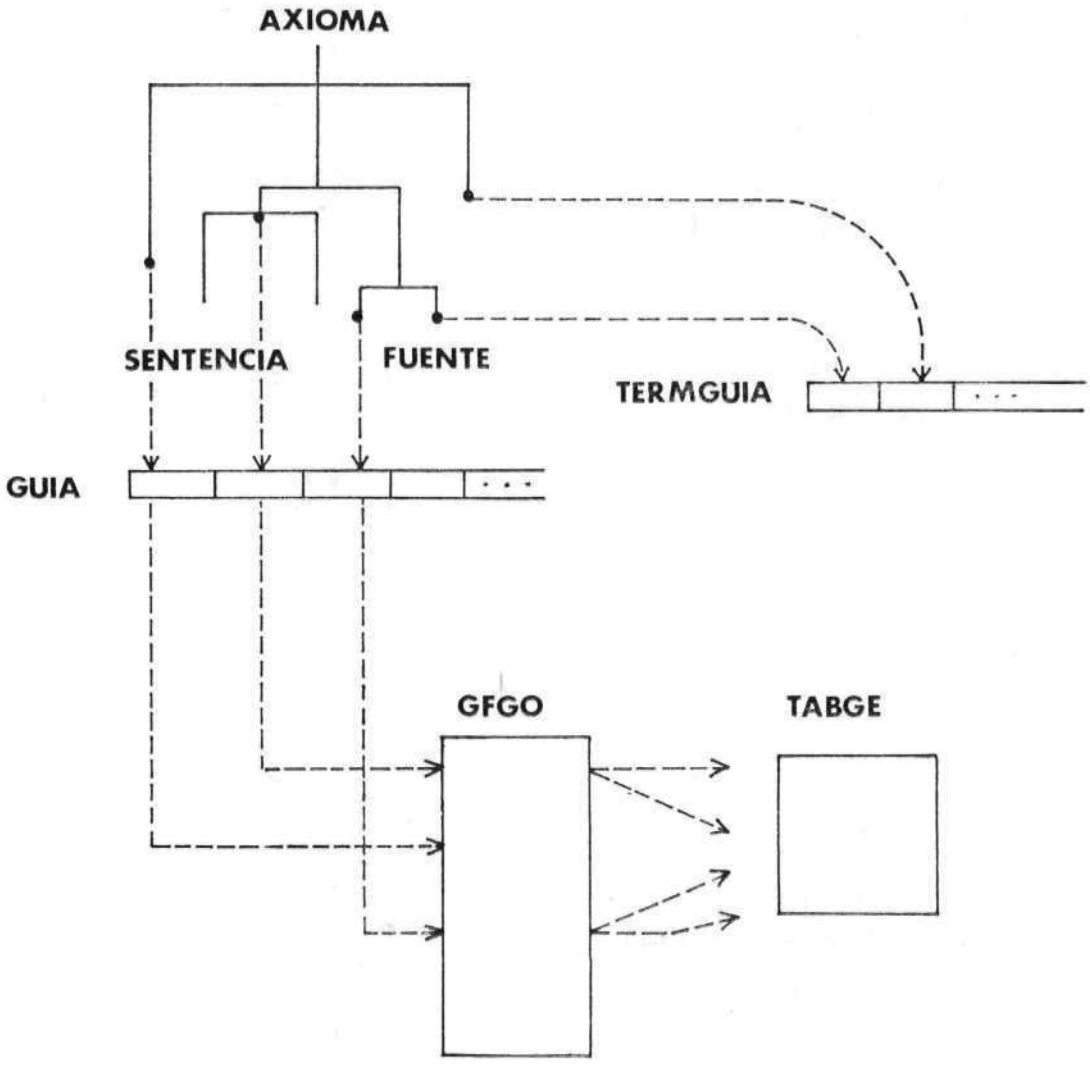
TERMGUIA contendrá los terminales particulares de la sentencia, requeridos por las rutinas semánticas. Por ejemplo, los dígitos constituyentes de la parte de dirección de una sentencia fuente; esta dirección será posteriormente elaborada para obtener el "mapping" con la dirección real de memoria principal.

Una segunda tabla, denominada TABGE, contiene todas las sentencias objeto (instrucciones) emitibles en la generación, la información contenida en esta tabla será presentada en 3.4.3.

El proceso descrito puede sintetizarse gráficamente mediante la figura que aparece en la página siguiente:

La secuencia de pasos lógicos en la ejecución del proceso es:

a) Inicialización.—Se corresponde con las líneas 93 a 96 del traductor en el Apéndice G.



b) Lectura con detección de fin de programa fuente.—En cada lectura, la macro LEER inicializa los registros del procesador utilizados para el acceso a las áreas de comunicación entre el analizador y el generador. Estos registros son:

- Registro 10 para accesos a la lista GUIA, cuya dirección de base está en el área BASEG (ver línea 70 del traductor).

- Registro 11 para accesos a TERMGUIA, cuya dirección de base se encuentra en BASE1 (ver línea 71 del traductor).

c) Análisis.

d) Generación.—El código objeto generado se deposita en la lista libre a continuación del traductor, para los accesos a la misma se utiliza la información contenida en el registro número 12 del procesador, el cual en la fase *a*) se cargó con el contenido del área DTRABG2.

e) Vuelta al paso *b*).

El compartir ambos, traductor y programa objeto, la memoria ejecutable permite que parte del "software" del traductor sea accedido por el programa generado en el momento de su ejecución.

En particular, la implementación de las sentencias LEER y ESCRIBIR (líneas 867 y 873, respectivamente, del traductor) utilizan el "software" de entrada-salida del traductor.

3.4.3. Reducciones y acciones semánticas asociadas

Empezaremos por los niveles más bajos del "parse", tomando como referencia para las reducciones el apartado 3.1.3. y para la implementación el traductor en el apéndice G.

Clase: dígito.—Deposita el terminal en TERMGUIA, para su ulterior proceso por la rutina semántica correspondiente.

Clase: cs.—Deposita en GUIA la entrada "D00C", la cual, a través del mecanismo de generación, conduce a la rutina semántica EVALCD para la evaluación de la etiqueta, línea 247 del traductor.

Clase: cn.—Deposita en GUIA la dirección "GFGO + 48" que es la novena entrada de la tabla GFGO, mediante el mecanismo de tablas presentado en 3.4.4. conduce en la generación a la rutina semántica EVALNUME para la evaluación de número. Línea 241 del traductor.

Clase: ct.—En su alternativa no vacía deposita en GUIA la dirección "GFGO + 54" que es la décima entrada de la tabla GFGO, en la generación se corresponde con la rutina semántica EVALDIRE para la evaluación de la dirección simbólica. Línea 235 del traductor.

Clase: oprela.—Las líneas 344, 349 y 354 depositan en GUIA las direcciones "GFGO + 114", "GFGO + 102" y "GFGO + 108" de entradas en la tabla GFGO, en la generación provocan la emisión de órdenes de comparación por menor, mayor e igual, respectivamente.

Clase: contct.—En su primera alternativa deposita en GUIA la dirección "GFGO + 96" (línea 328 del traductor); esta información activará la generación de la instrucción objeto que analiza si Contenido AC > 0.

La segunda alternativa realiza una acción semántica nula.

Clase esinst.—En su primera alternativa (línea 135 del traductor) se deposita la información "GFGO" en GUIA, la cual en la generación causará la emisión de las órdenes para ejecutar la lectura de una ficha aprovechando el "software" del traductor (DTFCS, en línea 648). En su segunda alternativa se procede análogamente para aprovechar el "software" DTFPR en línea 649.

Clase: ctinst.—En su segunda alternativa se cede control directamente al generador para que prepare la ejecución del programa objeto.

Dejamos al lector la elaboración o análisis de las restantes acciones semánticas asociadas a las reducciones.

3.4.4. Estructura e información de las tablas que intervienen en el proceso

Las dos tablas principales en el proceso son las denominadas GFGO y TABGE, ambas han sido ya comentadas con profundidad suficiente entre los apartados 3.4.2. y 3.4.3., seguidamente las incluimos indicando los desplazamientos a partir de la etiqueta de definición y las sentencias fuente que accederán a cada entrada de GFGO, cabe observar que la octava y décimo tercera no se utilizan.

0686	288C	2982	*LEER*	GFGO		DC	Y(TABGE+126)
0687	288E	0000				DC	Y(0)
0688	2890	0000				DC	Y(0)
0689	2892	298A	*ESCRIBIR*	+	6	DC	Y(TABGE+134)
0690	2894	0000				DC	Y(0)
0691	2896	0000				DC	Y(0)
0692	2898	2912	*DEPOSITAR*	+	12	DC	Y(TABGE+14)
0693	289A	0000				DC	Y(0)
0694	289C	0000				DC	Y(0)
0695	289E	2992	*DIVIDIR*	+	18	DC	Y(TABGE+142)
0696	28A0	0000				DC	Y(0)
0697	28A2	0000				DC	Y(0)
0698	28A4	2922	*TRASLADAR*	+	24	DC	Y(TABGE+30)
0699	28A6	0000				DC	Y(0)
0700	28A8	0000				DC	Y(0)
0701	28AA	292A	*SUMAR*	+	30	DC	Y(TABGE+38)
0702	28AC	0000				DC	Y(0)
0703	28AE	0000				DC	Y(0)
0704	28B0	2932	*SIEL*	+	36	DC	Y(TABGE+46)
0705	28B2	0000				DC	Y(0)
0706	28B4	0000				DC	Y(0)
0707	28B6	D000				DC	X'D000'
0708	28B8	0000				DC	Y(0)
0709	28BA	0000				DC	Y(0)
0710	28BC	D010	*= XXXXXX*	+	48	DC	X'D010'
0711	28BE	0000				DC	Y(0)
0712	28C0	0000				DC	Y(0)
0713	28C2	D008	*(XX)*	+	54	DC	X'D008'
0714	28C4	0000				DC	Y(0)

0715	28C6	0000				DC	Y(0)
0716	28C8	293A	*RESTAR*	+	60	DC	Y(TABGE+54)
0717	28CA	0000				DC	Y(0)
0718	28CC	0000				DC	Y(0)
0719	28CE	29A6	*MULTIPLICAR*	+	66	DC	Y(TABGE+162)
0720	28D0	0000				DC	Y(0)
0721	28D2	0000				DC	Y(0)
0722	28D4	295A				DC	Y(TABGE+86)
0723	28D6	0000				DC	Y(0)
0724	28D8	0000				DC	Y(0)
0725	28DA	295F	*HACER*	+	78	DC	Y(TABGE+91)
0726	28DC	0000				DC	Y(0)
0727	28DE	0000				DC	Y(0)
0728	28E0	2954	*PARAR*	+	84	DC	Y(TABGE+80)
0729	28E2	0000				DC	Y(0)
0730	28E4	0000				DC	Y(0)
0731	28E6	295A	*PONER*	+	90	DC	Y(TABGE+86)
0732	28E8	0000				DC	Y(0)
0733	28EA	0000				DC	Y(0)
0734	28EC	2974	*(AC)>0*	+	96	DC	Y(TABGE+112)
0735	28EE	0000				DC	Y(0)
0736	28F0	0000				DC	Y(0)
0737	28F2	2973	*(XX)>(AC)*	+	102	DC	Y(TABGE+111)
0738	28F4	0000				DC	Y(0)
0739	28F6	0000				DC	Y(0)
0740	28F8	2978	*(XX)=(AC)*	+	108	DC	Y(TABGE+116)
0741	28FA	0000				DC	Y(0)
0742	28FC	0000				DC	Y(0)
0743	28FE	297D	*(XX)<(AC)*	+	114	DC	Y(TABGE+121)
0744	2900	0000				DC	Y(0)
0745	2902	0000				DC	Y(0)

0746	2904	00000000000000	TABGE		DC	X'00000000000000'
0747	290B	00000000000000	+	7	DC	X'00000000000000'
0748	2912	D205FFFF10006D6D	+	14	DC	X'D205FFFF10006D6D6D'
	291A	6D				
0749	291B	FD551000FFFF6D	+	23	DC	X'FD551000FFFF6D'
0750	2922	D2051000FFFF6D6D	+	30	DC	X'D2051000FFFF6D6D'
0751	292A	FA551000FFFF6D6D	+	38	DC	X'FA551000FFFF6D6D'
0752	2932	F9551000FFFF6D6D	+	46	DC	X'F9551000FFFF6D6D'
0753	293A	FB551000FFFF6D6D	+	54	DC	X'FB551000FFFF6D6D'
0754	2942	FC551000FFFF6D6D	+	62	DC	X'FC551000FFFF6D6D'
0755	294A	02066D6D			DC	X'02066D6D'
0756	294E	45C055006D6D	+	74	DC	X'45C055006D6D'
0757	2954	A90007776D6D	+	80	DC	X'A90007776D6D'
0758	295A	47F0FFFF6D6D	+	86	DC	X'47F0FFFF6D6D'
0759	2960	45B060006D6D	+	92	DC	X'45B060006D6D'
0760	2966	45B060506D6D	+	98	DC	X'45B060506D6D'
0761	296C	45B061006D6D6D	+	104	DC	X'45B061006D6D6D'
0762	2973	4740FFFF6D	+	111	DC	X'4740FFFF6D'
0763	2978	4780FFFF6D	+	116	DC	X'4780FFFF6D'
0764	297D	4720FFFF6D	+	121	DC	X'4720FFFF6D'
0765	2982	45802A56	+	126	BAL	8,TSUB
0766	2986	FFFF6D6D			DC	X'FFFF6D6D'
0767	298A	45802A5A	+	134	BAL	8,TSUB+4
0768	298E	FFFF6D6D			DC	X'FFFF6D6D'
0769	2992	F8B52D501000FDB5	+	142	DC	X'F8B52D501000FDB52D50FFFFF85510002D506D6D'
	299A	2D50FFFFF8551000				
	29A2	2D506D6D				
0770	29A6	F8B52D501000FCB5	+	162	DC	X'F8B52D501000FCB52D50FFFFF85510002D566D6D'
	29AE	2D50FFFFF8551000				

El traductor asocia a cada sentencia fuente un número de secuencia, éste número es su etiqueta o dirección simbólica, para resolver ulteriores referencias entre sentencias es preciso conocer la dirección física de máquina correspondiente a cada etiqueta, en nuestra implementación optamos por una tabla para mantener dicha correspondencia.

La definición de la tabla se realiza en la macro LEER (3.2.2) de la siguiente forma:

	ETIQUETA	DIRECCION
constante de principio	999F	0000
PROGFUDI	•	•
	•	•
	capacidad para 60 sentencias	
	•	•

Finalmente, en la generación se diseñó una tabla denominada TSUB, que contiene todas las direcciones de principio de las rutinas utilizadas en la fase de generación. El mecanismo del generador al detectar en GUIA o en las entradas de GFGO información del tipo "DXXX" interpreta que debe ceder control a la dirección que indica, ésta es:

Desplazamiento XXX sobre registro de base D (registro 13)

previamente a esta cesión de control, el contenido del registro 13 es inicializado a la dirección de base de TSUB contenida en BASETSUB; la parte XXX se convierte así en el desplazamiento sobre TSUB para obtener la entrada correcta y en consecuencia activar la rutina deseada.

3.4.5. Rutinas semánticas utilizadas

La misión de estas rutinas es establecer la correspondencia entre los *objetos* utilizados en el programa y los *recursos* del "hardware" del computador.

En nuestro traductor fue necesario implementar las siguientes:

- Evaluación de la parte :XX
- Evaluación de la parte =XXXXXX
- Evaluación de la parte (XX)

En la línea 784 del traductor (Apéndice G) comienza la rutina EVALCD, para la evaluación de la parte ":XX" que aparece en las cadenas:

... PONER EN EL CD LA DIRECCION :XX

su función la realiza mediante la consulta en la tabla de etiquetas presentada en 3.4.4. Su dato de entrada "XX" lo espera tener en TERMGUIA+2.

En ciertas situaciones XX referencia a una entrada aún por definir (referencias hacia adelante), en este caso la rutina almacena la pareja de terminales XX en una pila denominada PILA1 y cuya profundidad es de 5.

El último paso antes de la ejecución (línea 873 del traductor), es verificar mediante consulta al "pointer" PONPILA1 si la pila tiene información, en cuyo caso se activa cíclicamente EVALCD hasta vaciar la pila.

La propia rutina coloca el resultado en la instrucción correspondiente; para esta tarea se basa en que en el área TRABG2 que es donde se está generando el programa objeto, las direcciones que están por satisfacer se caracterizan por una cadena de unos binarios (en hexadecimal FFFF); de otra parte, el valor calculado por la rutina se corresponde siempre con la última instrucción generada.

En la situación antes de la ejecución, esta correspondencia se mantiene, por lo que si consideramos TRABG2 como una pila de direcciones por satisfacer, se corresponderá con las terminales en PILA1.

La rutina EVALNUME (línea 822 del traductor), convierte los literales numéricos de la forma =XXXXXX al formato utilizado por la máquina, el resultado de la conversión se deposita en el "pool de literales" (línea 830 del traductor), esta rutina utiliza parte de la EVALCD para colocar en la instrucción generada la dirección del operando en la "pool".

La rutina EVALDIRE (línea 834 del traductor) realiza el cálculo (ver 2.2.2) de la dirección física que corresponde a la especificada en las sentencias fuente.

En la implementación se fijó como dirección del acumulador la dirección 1000_{16}, de la máquina, las palabras están en direcciones consecutivas a partir del acumulador (dirección de comienzo 1006_{16}), cada una está implementada por un vector de memoria de 6 bytes, el cálculo para determinar la dirección de la base de su vector es:

$$4102 + XX \cdot 6 \qquad ; 4102 = 1006_{16})$$

4. REFERENCIAS

Las más directamente relacionadas con el trabajo se han distinguido con un asterisco.

[1] *Specification Languages for Mechanical Languages and Their Processors. A Baker's Dozen*, por SAUL GORN, CACM, diciembre 1961.

[2] *Programming Languages, Information structures and machine organization*, por PETER WEGNER, Mc. Graw-Hill, 1968.

[3] *A sintax-Oriented Translator*, por ZILAHY INGERMAN, Academic Press, 1966.

[4] *Translator Writing Systems*, por J. FELDMAN y GRIES, CACM, febrero 1968.

[5] *Programming Languages*, por J. J. DUBY, Course nº 211, 1970.

*[6] *Construcción de un traductor para un lenguaje especialmente orientado a problemas físicos*, por I. RAMOS SALAVERT, Tesis Doctoral, 1971.

[7] *Formal Semantics for Computer languages and its Applications in a Compiler-Compiler*, por J. FELDMAN, CACM, enero 1966.

[8] *Automatic Syntactic Analysis*, por J. M. FOSTER, Mc. Donald Computer Monograph, series 1970.

[9] *Bounded Context Syntactic Analysis*, por R. W. FLOYD, CACM, febrero 1964.

[10] *Compiling Tecniques*, por F. R. A. HOPGOOD, Computer Monograph, series Stanley Gill, 1969.

[11] *The Anatomy of a Compiler*, por JOHN A. N. LEE.

*[12] *Construcción de un Metaensamblador*, por R. MOYA, Tesina de Licenciatura, 1972.

[13] *A macro-generable language for the 360 computers*, por M. GRIFFITHS y M. PELTIER, Computer Bulletin, volumen 13, núm. 11, noviembre 1969.

*[14] *Grammar Transformation as an aid to compiler production*, por M. GRIFFITHS y M. PELTIER, Centre Scientifique IBM France, estudio núm. FF2-0057-0, mayo 1968.

[15] *A note on Foster's Syntax Improving Device*, por P. M. WOODWARD, R. R. E. Memorandum 2352, diciembre 1966.

[16] *A Sintax Improving Program*, por J. M. FOSTER, R. R. E. Memorandum 2389, julio 1967.

*[17] *Top-Down Syntax Analysis*, por D. E. KNUTH, International Summer School on Computer programming Copenhagen, agosto 1967.

*[18] *A basic language oriented to secondary schools*, por I. FERNÁNDEZ-FLÓREZ, E. CAMARERO, I. RAMOS, IFIP Congress, agosto 1971, Booklet TA-7.

[19] *Analyse syntaxique et compilation*, por BOUSSARD, curso 1968-69.

[20] *Compilation*, por C. PAIR, Escuela de verano del AFCET, Neuchatel, 1972.

*[21] *Data structures theory and practice*, por A. T. BERZTISS, computer science and applied mathematics, Academic Press, 1971.

*[22] *Cierre transitivo de una relación binaria entre los elementos de un conjunto finito: Teorema de Warshall*, por L. GÓMEZ BUENO e I. RAMOS SALAVERT, nota interna Instituto de Informática, Madrid, 1971.

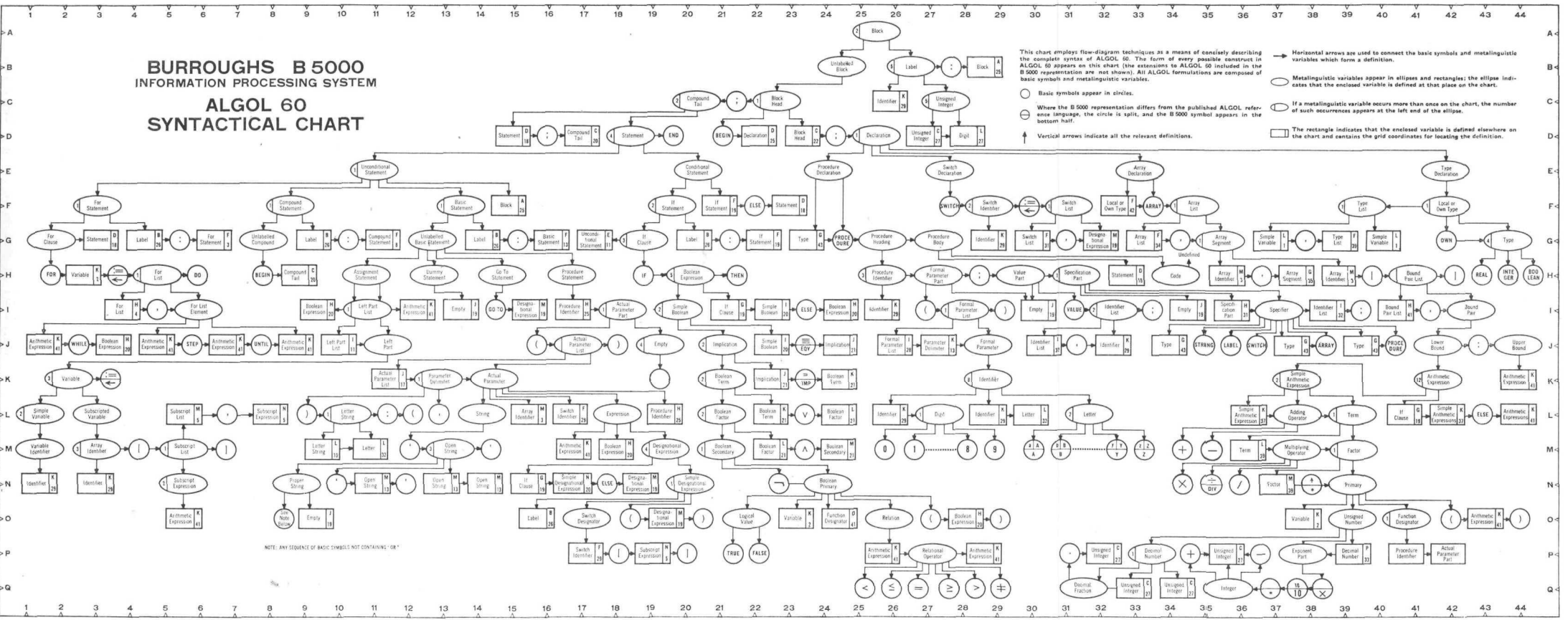
[23] *The art of computer programming*, por D. E. KNUTH, Addison Wesley, 1968.

[24] *Technologie de la programmation*, por VINCENT TIXIER, Ecole Nationale Superieure de L'Aeronautique, mayo 1969.

APENDICE A

Carta sintáctica del ALGOL 60
Tomada de comunications del
ACM. Volumen 4 - Núm. 9 - Septiembre 1961

BURROUGHS B 5000
INFORMATION PROCESSING SYSTEM
ALGOL 60
SYNTACTICAL CHART



APENDICE B

Implementación de la técnica Hash al cálculo del cierre transitivo

Descripción

El dato de entrada al programa es la gramática del lenguaje fuente escrita en notación BNF; con propósito de reducir el tiempo de ejecución hemos introducido particularidades en la notación. Estas son:

- Preceder a las cadenas de caracteres terminales por el carácter blanco.
- Sustituir en las producciones con alternativa única el símbolo ":: =" por el ":!=".
- Sustituir en el caso de varias alternativas en una producción, el símbolo "|" por el "!" en la última alternativa.
- Yuxtaponer los símbolos "|—¡" como indicador de que sigue una ficha de continuación.

El proceso se puede descomponer lógicamente en cinco pasos etiquetados desde P1 hasta P5.

En el primer paso la gramática del lenguaje fuente es almacenada en una tabla ubicada en la lista libre a continuación del programa, los accesos a esta tabla utilizan el registro 8 del procesador. Las fichas son leídas sobre el área denominada ROBUF y aquí se utiliza para los accesos el registro 9 del procesador. En esta fase se imprimen las metavariables junto con sus elementos terminales y no terminales que están relacionados a izquierdas. El área "buffer" de impresión se denomina AUX y para su acceso se utilizó el registro 10 del procesador. Con propósitos de marca de fin de información se utilizó el carácter "—¡" de código EBCDIC "5F".

En el paso P2 se construye la estructura de tabla Hash (vector y entradas), para su ubicación se hacen peticiones de espacio de la lista libre.

La función de aleatorización elegida fue una operación:

REPRESENTACION INTERNA DEL TERMINAL .AND. $3F_{16}$)

(línea 76 del programa fuente).

Los accesos utilizan los siguientes registros:

- Registro 9 para la tabla creada en paso P1.
- Registro 8 para peticiones a la lista libre.
- Registro 10 para posicionamiento dentro de las entradas de la tabla.
- Registro 11 para accesos al vector Hash.

Una vez creada la estructura se insertan en las entradas correspondientes los primeros nombres de metavariables de los que dependen inmediatamente los terminales en el vector Hash.

En el paso P3, mediante iteracciones, se completan las entradas con los nombres de metavariables que dependen inmediatamente de clases que ya han sido incluidas, resultando al fin *por cada terminal las metavariables* con las que está relacionado.

En el paso P4 se imprime el resultado del paso anterior, con fines de control sobre el programa.

En P5 se elabora el resultado de P3 para obtener la misma información pero organizada en la forma: *por cada metavariable los terminales* con los que está relacionado, finalmente se imprime y perfora en fichas el resultado.

Los accesos utilizan los siguientes registros:

- Registro 8 para el vector Hash.
- Registro 9 para la tabla creada en P1.
- Registro 10 para el área AUX.
- Registro 11 para las entradas en la tabla Hash.

Codificación en el lenguaje
Ensamblador del UNIVAC 9300

```
0001                                        PRINT   ON,NOGEN
0002    13A8                        P1      START   X'13A8'
0003                                        USING   *,0,1,2,3,4
0004    13A8    45E0188C            EMPE    OPEN    FICH
0005    13AC    45E019C2                    OPEN    IMPR
0006    13B0    48C01DCA                    LH      12,ULTM
0007    13B4    925FC000                    MVI     0(12),C'—|'
0008    13B8    D283179B179A                MVC     AUX,BL
0009    13BE    48801DCA                    LH      8,ULTM
0010    13C2    A9005555                    HPR     X'5555'
0011    13C6    45E018941746        A1      GET     FICH,ROBUF
0012    13CC    48901798                    LH      9,PBUF
0013    13D0    957A1750                    CLI     ROBUF + 10,C':'
0014    13D4    47701408                    BNE     A2
0015    13D8    48A01820                    LH      10,DAUX
0016    13DC    45E019D2179B                PUT     IMPR,AUX
0017    13E2    D283179B179A                MVC     AUX,BL
0018    13E8    925F8000                    MVI     0(8),X'5F'
0019    13EC    AA80182E                    AH      8,UNO
0020    13F0    D20980009000                MVC     0(10,8),0(9)
0021    13F6    AA801830                    AH      8,DIEZ
0022    13FA    D209A0009000                MVC     0(10,10),0(9)
0023    1400    AAA01DC6                    AH      10,TREC
0024    1404    AA901DC6                    AH      9,TREC
0025    1408    955F9000            A2      CLI     0(9),X'5F'
0026    140C    478013C6                    BE      A1
0027    1410    954C9000                    CLI     0(9),C'<'
0028    1414    47801468                    BE      A4
0029    1418    D20180009000                MVC     0(2,8),0(9)
0030    141E    AA801836                    AH      8,DOS
0031    1422    D201A0009000                MVC     0(2,10),0(9)
0032    1428    AAA01DC8                    AH      10,CIN
0033    142C    49A01826                    CH      10,DUX1
0034    1430    47401444                    BC      4,A5
0035    1434    45E019D2179B                PUT     IMPR,AUX
0036    143A    48A01824                    LH      10,DAOX
0037    143E    D283179B179A                MVC     AUX,BL
0038    1444    AA90182E            A5      AH      9,UNO
0039    1448    955F9000                    CLI     0(9),X'5F'
0040    144C    478013C6                    BE      A1
0041    1450    954F9000                    CLI     0(9),X'4F'
0042    1454    47801460                    BE      A6
0043    1458    955A9000                    CLI     0(9),X'5A'
```

```
0044  145C  47701444                BNE   A5
0045  1460  AA90182E          A6    AH    9,UNO
0046  1464  47F01408                B     A2
0047  1468  D20980009000      A4    MVC   0(10,8),0(9)
0048  146E  AA801830                AH    8,DIEZ
0049  1472  D209A0009000            MVC   0(10,10),0(9)
0050  1478  AAA01DC6                AH    10,TREC
0051  147C  49A01826                CH    10,DUX1
0052  1480  47401444                BC    4,A5
0053  1484  45E019D2179B            PUT   IMPR,AUX
0054  148A  48A01824                LH    10,DAOX
0055  148E  D283179B179A            MVC   AUX,BL
0056  1494  47F01444                B     A5
0057  1498  45E019D2179B      P2    PUT   IMPR,AUX
0058  149E  92008000                MVI   0(8),X'00'
0059  14A2  AA80182E                AH    8,UNO
0060  14A6  48901DCA                LH    9,ULTM
0061  14AA  925F17A5                MVI   AUX + 10,X'5F'
0062  14AE  925F8000                MVI   0(8),X'5F'
0063  14B2  D2BE80018000            MVC   1(191,8),0(8)
0064  14B8  4080182A                STH   8,BASE
0065  14BC  AA801822                AH    8,CIEN92
0066  14C0  955F9000          B1    CLI   0(9),X'5F'
0067  14C4  477014D2                BNE   B2
0068  14C8  AA90182E                AH    9,UNO
0069  14CC  D209179B9000            MVC   AUX(10),0(9)
0070  14D2  AA901830          B2    AH    9,DIEZ
0071  14D6  95009000          B3    CLI   0(9),X'00'
0072  14DA  47801568                BE    P3
0073  14DE  95409000                CLI   0(9),C' '
0074  14E2  477014C0                BNE   B1
0075  14E6  D20118329000            MVC   TEMP,0(9)
0076  14EC  943F1833                NI    TEMP + 1,X'3F'
0077  14F0  92001832                MVI   TEMP,X'00'
0078  14F4  48B01832                LH    11,TEMP
0079  14F8  AAB01832                AH    11,TEMP
0080  14FC  AAB01832                AH    11,TEMP
0081  1500  AAB0182A                AH    11,BASE
0082  1504  955FB000                CLI   0(11),X'5F'
0083  1508  4770155A                BNE   B4
0084  150C  D200B0009001            MVC   0(1,11),1(9)
0085  1512  4080182C                STH   8,SOPOR
0086  1516  D201B001182C            MVC   1(2,11),SOPOR
0087  151C  AA801834                AH    8,OCHE
```

```
0088  1520  48A0182C          LH    10,SOPOR
0089  1524  9240A001          MVI   1(10),C' '
0090  1528  D24DA002A001      MVC   2(78,10),1(10)
0091  152E  925FA000          MVI   0(10),X'5F'
0092  1532  AA901836    B5    AH    9,DOS
0093  1536  D201182CB001      MVC   SOPOR,1(11)
0094  153C  48A0182C          LH    10,SOPOR
0095  1540  955FA000    B6    CLI   0(10),X'5F'
0096  1544  47801550          BE    B7
0097  1548  AAA01830          AH    10,DIEZ
0098  154C  47F01540          B     B6
0099  1550  D20AA000179B  B7    MVC   0(11,10),AUX
0100  1556  47F014D6          B     B3
0101  155A  D500B0009001  B4    CLC   0(1,11),1(9)
0102  1560  47801532          BE    B5
0103  1564  A9007EC8          HPR   X'7EC8'
0104  1568  48B0182A    P3    LH    11,BASE
0105  156C  AAB01822          AH    11,CIEN92
0106  1570  40B01798          STH   11,PBUF
0107  1574  40B01832          STH   11,TEMP
0108  1578  48B0182A          LH    11,BASE
0109  157C  955FB000    D21   CLI   0(11),X'5F'
0110  1580  477015A0          BNE   D20
0111  1584  AAB01838    D22   AH    11,TRES
0112  1588  49B01798          CH    11,PBUF
0113  158C  4740157C          BL    D21
0114  1590  48B01832          LH    11,TEMP
0115  1594  AAB01834          AH    11,OCHE
0116  1598  40B01832          STH   11,TEMP
0117  159C  47F015B4          B     D3
0118  15A0  D50118328001  D20   CLC   TEMP,1(11)
0119  15A6  47201584          BH    D22
0120  15AA  D2011832B001      MVC   TEMP,1(11)
0121  15B0  47F01584          B     D22
0122  15B4  48901DCA    D3    LH    9,ULTM
0123  15B8  9200183A          MVI   INDI,X'00'
0124  15BC  955F9000    D4    CLI   0(9),X'5F'
0125  15C0  477015D2          BNE   D5
0126  15C4  AA90182E          AH    9,UNO
0127  15C8  D209179B9000      MVC   AUX(10),0(9)
0128  15CE  AA901830          AH    9,DIEZ
0129  15D2  95009000    D5    CLI   0(9),X'00'
0130  15D6  4780162A          BE    D11
0131  15DA  954C9000          CLI   0(9),X'4C'
```

```
0132  15DE  478015EA          BE    D8
0133  15E2  AA901836          AH    9,DOS
0134  15E6  47F015BC          B     D4
0135  15EA  48A01798    DB    LH    10,PBUF
0136  15EE  9201183A          MVI   INDI,X'01'
0137  15F2  D5089001A001 D6    CLC   1(9,9),1(10)
0138  15F8  47801610          BE    D7
0139  15FC  AAA01830          AH    10,DIEZ
0140  1600  49A01832          CH    10,TEMP
0141  1604  474015F2          BL    D6
0142  1608  AA901830          AH    9,DIEZ
0143  160C  47F015BC          B     D4
0144  1610  AAA01830    D7    AH    10,DIEZ
0145  1614  955FA000          CLI   0(10),X'5F'
0146  1618  47701610          BNE   D7
0147  161C  92409000          MVI   0(9),X'40'
0148  1620  D20AA000179B       MVC   0(11,10),AUX
0149  1626  47F015F2          B     D6
0150  162A  9501183A    D11   CLI   INDI,X'01'
0151  162E  478015B4          BE    D3
0152  1632  9240179B    P4    MVI   AUX,C' '
0153  1636  D2B2179C179B       MVC   AUX + 1(131),AUX
0154  163C  4890182A          LH    9,BASE
0155  1640  AA901822          AH    9,CIEN92
0156  1644  40901828          STH   9,FINHAS
0157  1648  4890182A          LH    9,BASE
0158  164C  D201182C9001 C1    MVC   SOPOR,1(9)
0159  1652  955F9001          CLI   1(9),X'5F'
0160  1656  4780167A          BE    SUM
0161  165A  48A0182C          LH    10,SOPOR
0162  165E  D20017A09000       MVC   AUX + 5(1),0(9)
0163  1664  D24F17A5A000       MVC   AUX + 10(80),0(10)
0164  166A  45E019D2179B       PUT   IMPR,AUX
0165  1670  9240179B          MVI   AUX,C' '
0166  1674  D2821790179B       MVC   AUX + 1(131),AUX
0167  167A  AA901838    SUM   AH    9,TRES
0168  167E  49901828          CH    9,FINHAS
0169  1682  4740164C          BL    C1
0170  1686  45E01890          CLOSE FICH
0171  168A  45E01C06          OPEN  PERF
0172  168E  48901DCA    P5    LH    9,ULTM
0173  1692  955F9000    F1    CLI   0(9),X'5F'
0174  1696  478016B6          BE    F6
0175  169A  95009000          CLI   0(9),X'00'
```

```
0176  169E  477016AE                BNE     F7
0177  16A2  45E019C6                CLOSE   IMPR
0178  16A6  45E01C0A                CLOSE   PERF
0179  16AA  A9001111                HPR     X'1111'
0180  16AE  AA90182E        F7      AH      9,UNO
0181  16B2  47F01692                B       F1
0182  16B6  AA90182E        F6      AH      9,UNO
0183  16BA  48A01820                LH      10,DAUX
0184  16BE  D209A0009000            MVC     0(10,10),0(9)
0185  16C4  AA901830                AH      9,DIEZ
0186  16C8  AAA01830                AH      10,DIEZ
0187  16CC  9240A000                MVI     0(10),X'40'
0188  16D0  AAA0182E                AH      10,UNO
0189  16D4  48B0182A                LH      11,BASE
0190  16D8  955FB000        F2      CLI     0(11),X'5F'
0191  16DC  4770170C                BNE     F3
0192  16E0  AAB01838                AH      11,TRES
0193  16E4  49B01828                CH      11,FINHAS
0194  16E8  474016D8                BL      F2
0195  16EC  D24F1746179B            MVC     ROBUF,AUX
0196  16F2  45E019D2179B            PUT     IMPR,AUX
0197  16F8  45E01C161746            PUT     PERF,ROBUF
0198  16FE  9240179B                MVI     AUX,X'40'
0199  1702  D282179C179B            MVC     AUX + 1(131),AUX
0200  1708  47F01692                B       F1
0201  170C  D201182CB001    F3      MVC     SOPOR,1(11)
0202  1712  4880182C                LH      8,SOPOR
0203  1716  D509179B8000    F4      CLC     AUX(10),0(8)
0204  171C  47801734                BE      F8
0205  1720  AA801830                AH      8,DIEZ
0206  1724  9555F8000               CLI     0(8),X'5F'
0207  1728  47701716                BNE     F4
0208  172C  AAB01838        F5      AH      11,TRES
0209  1730  47F016D8                B       F2
0210  1734  D200A000B000    F6      MVC     0(1,10),0(11)
0211  173A  AAA0182E                AH      10,UNO
0212  173E  47F0172C                B       F5
0213  1742  A9001111                HPR     X'1111'
0214  1746                ROBUF     DS      CL80
0215  1796  5F                      DC      X'5F'
0216  1798  1746          PBUF      DC      Y(ROBUF)
0217  179A  40            BL        DC      C' '
0218  179B                AUX       DS      CL132
0219  1820  179B          DAUX      DC      Y(AUX)
```

```
0220          1822   00C0          CIEN92   DC     Y(192)
0221          1824   17A8          DAOX     DC     Y(AUX + 13)
0222          1826   181F          DUX1     DC     Y(AUX + 132)
0223          1828   18EA          FINHAS   DC     Y(BASE + 192)
0224          182A   0000          BASE     DC     Y(0)
0225          182C   0000          SOPOR    DC     Y(0)
0226          182E   0001          UNO      DC     Y(1)
0227          1830   000A          DIEZ     DC     Y(10)
0228          1832   0000          TEMP     DC     Y(0)
0229          1834   0050          OCHE     DC     Y(80)
0230          1836   0002          DOS      DC     Y(02)
0231          1838   0003          TRES     DC     Y(03)
0232          183A   40            INDI     DC     C' '
0233          183B                 LUIS     DS     CL80
0234                               FICH     DTFCS  EOFA = P2
0235                               IMPR     DTFPR  DEVA = 10,PROV = YES
0236                               PERF     DTFRP  DEVA = 9,MODE = TRANS,OTBL = TBPU,OUAR = LUIS
        T     1D3A   A6FC0002      C?20     AI     2,—4    SET PROC PSC FOR REJECT RETURN  TS003110
        T     1D52   A6FF0108               AI     264,—1  ADJUST E?AS FOR XIOF COMPLETION TS003180
0237          1DC6   000D          TREC     DC     Y(13)
0238          1DC8   0005          CIN      DC     Y(05)
0239          1DCA   2008          ULTM     DC     Y(8200)
0240          1DCC   0000000013A8           END    EMPE
```

APENDICE C

Implementaciones para el tratamiento automático de las fórmulas lineales de un dígrafo

Implementación de la estrategia de análisis basada en el parse canónico
(FORTRAN V del UNIVAC 1100)

```
@FOR,IS
FOR 010A-01/15/74-20:22:52

     MAIN PROGRAM

     STORAGE USED: CODE (1) 000317; DATA (0) 004263; BLANK COMMON (2) 000000


     EXTERNAL REFERENCES (BLOCK, NAME)

     0003    NINTR$
     0004    NRDU$
     0005    NI01$
     0006    NI02$
     0007    NWDU$
     0010    NSTOP$


     STORAGE ASSIGNMENT (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0000    004201 1F      0001    000204 100L    0001    000004 107G    0001    000212 110L    0001    000012 114G
     0001    000014 120G    0001    000220 120L    0001    000042 131G    0001    000030 15L     0001    000116 164G
     0000    004170 2F      0001    000050 20L     0001    000145 206G    0001    000162 222G    0001    000240 254G
     0001    000244 260L    0001    000246 262G    0001    000256 280L    0001    000261 285L    0000    004236 3F
     0001    000273 300G    0001    000301 305G    0001    000070 34L     0000    004250 4F      0001    000076 40L
     0001    000105 45L     0000    004167 5F      0001    000122 50L     0001    000124 55L     0000    004216 6F
     0001    000130 60L     0001    000134 65L     0000    004224 7F      0001    000002 8L      0001    000155 92L
     0001    000172 96L     0001    000313 999L    0000 I  004162 I       0000 I  004165 IC      0000 I  004163 J
     0000 I  004164 K       0000 I  003504 KF      0000 I  004161 NAS     0000 I  004160 NBL     0000 I  004166 NN
     0000 I  000000 NODOS   0000 I  000074 NPRES


     00100          1*      C       ESTE PROGRAMA HALLA EL DICCIONARIO DE PRECEDENTES                          GRAFO001
     00100          2*      C       DE UN GRAFO, A PARTIR DE SU FORMULA LINEAL.                                GRAFO002
     00100          3*      C                                                                                  GRAFO003
     00100          4*      C                                                                                  GRAFO004
     00100          5*      C       OBSERVACIONES-LIMITACIONES                                                 GRAFO005
     00100          6*      C           — SE ASUME QUE LA CADENA DE ENTRADA ES UNA FORMULA LINEAL.             GRAFO006
     00100          7*      C           — PUEDE HABER HASTA 60 SIMBOLOS DE NODO DISTINTOS.                     GRAFO007
```

```
00100      8*    C          — UN NODO NO DEBE TENER MAS DE 30 PRECEDENTES.                GRAFO008
00100      9*    C          — LA LONGITUD DE LA CADENA DE ENTRADA NO EXCEDERA             GRAFO009
00100     10*    C              DE 300 CARACTERES.                                        GRAFO010
00100     11*    C                                                                        GRAFO011
00100     12*    C                                                                        GRAFO012
00101     13*         DIMENSION NODOS (60), NPRES (60,30), KF (300)                       GRAFO013
00103     14*         DATA  NBL/'  '/,NAS/'*'/                                            GRAFO014
00103     15*    C                                                                        GRAFO015
00103     16*    C                                                                        GRAFO016
00103     17*    C         INICIALIZACION DE LAS MATRICES:                                GRAFO017
00103     18*    C               KF     — ALMACEN DE LA CADENA DE ENTRADA               GRAFO018
00103     19*    C               NODOS— NODOS QUE TIENEN ALGUN PRECEDENTE               GRAFO019
00103     20*    C               NPRES — PRECEDENTES DE LOS NODOS                       GRAFO020
00103     21*    C                                                                        GRAFO021
00103     22*    C                                                                        GRAFO022
00106     23*          8 DO  9  I = 1,300                                                 GRAFO023
00111     24*          9 KF(I) = NBL                                                      GRAFO024
00113     25*            DO 10 J = 1,60                                                   GRAFO025
00116     26*            NODOS(J) = 0                                                     GRAFO026
00117     27*            DO  10  K = 1,30                                                 GRAFO027
00122     28*           10 NPRES(J,K) = NBL                                               GRAFO028
00122     29*    C                                                                        GRAFO029
00122     30*    C                                                                        GRAFO030
00122     31*    C         LECTURA DE LA CADENA DE ENTRADA Y DETERMINACION DE SU FINAL POR GRAFO031
00122     32*    C         EL PRIMER CARACTER BLANCO, LA VARIABLE IC CONTENDRA AL         GRAFO032
00122     33*    C         FINAL DE ESTE PROCESO EL NUMERO DE CARACTERES DE LA CADENA     GRAFO033
00122     34*    C                                                                        GRAFO034
00122     35*    C                                                                        GRAFO035
00125     36*            K = 1                                                            GRAFO036
00126     37*            IC = 80                                                          GRAFO037
00127     38*           15 READ(5,5,END = 999)    (KF(I),I = K.IC)                      GRAFO038
00135     39*            5 FORMAT(80A1)                                                  GRAFO039
00136     40*            I = K                                                           GRAFO040
00136     41*    C                                                                        GRAFO041
00136     42*    C         CICLO  EXPLORACION FICHA PARA DETERMINAR SI ES LA ULTIMA       GRAFO042
00136     43*    C         PORTADORA DE LA FORMULA                                        GRAFO043
00136     44*    C                                                                        GRAFO044
00137     45*           20 IF(I—IC)       , , 40                                         GRAFO045
00142     46*              IF(I—300)      , , 34                                         GRAFO046
00145     47*              IF(KF(I)—NBL)    ,45,                                         GRAFO047
00150     48*              I = I + 1                                                       GRAFO048
00151     49*              GO  TO  20                                                      GRAFO049
00151     50*    C                 MENSAJE DE ERROR POR CADENA DEMASIADO LARGA           GRAFO051
00152     51*           34 WRITE  (6,2)                                                  GRAFO053
00154     52*            2 FORMAT(1H1, 'ERROR FORMULA CON MAS DE 300 CARACTERES')        GRAFO054
```

```
00155    53*          GO TO 8                                                                    GRAFO055
00155    54*    C                                                                                GRAFO056
00155    55*    C                    FICHA TOTALMENTE LLENA                                      GRAFO057
00155    56*    C                                                                                GRAFO058
00156    57*       40 K = K + IC                                                                 GRAFO059
00157    58*          IC = IC + 80                                                               GRAFO060
00160    59*          GO TO 15                                                                   GRAFO061
00160    60*    C                                                                                GRAFO062
00160    61*    C          SE ENCONTRO EL FIN DE LA CADENA COLOCAR IC A SU VALOR                 GRAFO063
00160    62*    C                                                                                GRAFO064
00161    63*       45 IC = I—1                                                                   GRAFO065
00161    64*    C                                                                                GRAFO066
00161    65*    C          IMPRESION DE LA CADENA DE ENTRADA EN CABECERA DE PAGINA               GRAFO067
00161    66*    C                                                                                GRAFO068
00162    67*          WRITE(6,1)   (KF(I), I = 1,300)                                            GRAFO069
00170    68*        1 FORMAT(1H1, '   FORMULA DADA        ,100A1/1H    ,16 (1H—)/1H    ,20X,100A1 GRAFO070
00170    69*         1/1H0,20X,100A1)                                                          GRAFO071
00170    70*    C                                                                                GRAFO072
00170    71*    C                                                                                GRAFO073
00170    72*    C                              A L G O R I T M O                                 GRAFO074
00170    73*    C                                                                                GRAFO075
00170    74*    C                                                                                GRAFO076
00171    75*       50 I = 1                                                                      GRAFO077
00172    76*       55 IF(KF(I)—NAS)    ,65,                                                      GRAFO078
00175    77*       60 I = I + 1                                                                  GRAFO079
00176    78*          GO TO 55                                                                   GRAFO080
00177    79*       65 IF(KF(I + 1)—NAS)   ,60,                                                   GRAFO081
00202    80*          IF(KF(I + 2)—NAS)   ,60,                                                   GRAFO082
00202    81*    C                                                                                GRAFO083
00202    82*    C                    ASIGNACION DE PRECEDENTES                                   GRAFO084
00202    83*    C                    KF(I +1)   ES   PRECEDENTE DE   KF(I + 2)                   GRAFO085
00202    84*    C                                                                                GRAFO086
00205    85*          DO 90    J = 1,60                                                          GRAFO087
00210    86*          IF(NODOS(J))   ,92,                                                        GRAFO088
00213    87*          IF(NODOS(J)—KF(I + 2))   90,92,90                                          GRAFO089
00216    88*       90 CONTINUE                                                                   GRAFO090
00220    89*          GO TO   100                                                                GRAFO091
00221    90*       92 DO  94  K = 1,30                                                           GRAFO092
00224    91*          IF(NPRES(J,K)—NBL)   94,96,94                                              GRAFO093
00227    92*       94 CONTINUE                                                                   GRAFO094
00231    93*          GO TO  110                                                                 GRAFO095
00232    94*       96 NODOS(J) = KF(I +2)                                                        GRAFO096
00233    95*          NPRES(J,K) = KF(I + 1)                                                     GRAFO097
00234    96*          GO TO  120                                                                 GRAFO098
00235    97*      100 WRITE(6,6)                                                                 GRAFO099
```

```
00237      98*        6 FORMAT(1H0, 'ERROR MAS DE 60 NODOS')                                  GRAFO100
00240      99*          GO TO 8                                                               GRAFO101
00241     100*      110 WRITE(6,7)                                                            GRAFO102
00243     101*        7 FORMAT(1H0, 'ERROR ALGUN NODO TIENE MAS DE 30 PRECEDENTES')           GRAFO103
00244     102*          GO TO 8                                                               GRAFO104
00245     103*      120 IF(IC—3)    ,260,                                                     GRAFO105
00250     104*          KF(I) = KF(I + 1)                                                     GRAFO106
00251     105*          I = I + 1                                                             GRAFO107
00252     106*          IC = IC—2                                                             GRAFO108
00253     107*          DO 130   K = I,IC                                                     GRAFO109
00256     108*      130 KF(K) = KF(K + 2)                                                     GRAFO110
00260     109*          GO TO 50                                                              GRAFO111
00260     110*    C                                                                           GRAFO112
00260     111*    C              IMPRESION DEL DICCIONARIO DE PRECEDENTES                     GRAFO113
00260     112*    C                                                                           GRAFO114
00261     113*      260 DO 270 J = 1,60                                                       GRAFO115
00264     114*          IF(NODOS(J))    ,280,                                               GRAFO116
00267     115*      270 CONTINUE                                                            GRAFO117
00271     116*          NN = 60                                                             GRAFO118
00272     117*          GO TO 285                                                           GRAFO119
00273     118*      280 NN = J—1                                                            GRAFO120
00274     119*      285 WRITE(6,3)                                                          GRAFO121
00276     220*        3 FORMAT(1H0, 'NODOS', 5X, 'PRECEDENTES'/1H ,5(1H—), 5X,11(1H—))      GRAFO122
00277     121*          DO 290 J = 1,NN                                                     GRAFO123
00302     122*      290 WRITE(6,4) NODOS(J), (NPRES(J,K), K = 1,30)                         GRAFO124
00312     123*        4 FORMAT(1H0, 2X, A1, 7X, 30(A1, 1X))                                 GRAFO125
00313     124*          GO TO 8                                                             GRAFO126
00314     125*      999 STOP                                                                GRAFO127
00315     126*          END                                                                 GRAFO128

        END OF COMPILATION:          NO DIAGNOSTICS

@XQT
MAP  0023-01/15—20:23
```

FORMULA DADA ******A*BCD***JKLE*KCL*EA

NODOS	PRECEDENTES
C	B K
B	A
D	A
K	J A
L	J A
E	J A
J	A
A	E

FORMULA DADA *S*E*L***PAB(

NODOS	PRECEDENTES
A	P
B	P
(	P
P	L
L	E
E	S

Implementación de la estrategia de análisis
de derecha a izquierda
(SNOBOL del UNIVAC 1100)

```
@LGED.SNO,IS TPF$.KFORM2
  SNOBOL4 (VERSION 2.0) OCT. 7. 1968
  BELL TELEPHONE LABORATORIES, INCORPORATED


000000
000001          *       NAME KFORM2
000002          *
000003          *       DATE 17.01.1974
000004          *
000005          *       ESTE PROGRAMA ACEPTA UN GRAFO CODIFICADO COMO UNA K-FORMULA
000006          *       COMPLEJA, OBTENIENDO POR EL METODO DE LA PILA LAS K-FORMULAS
000007          *       BASE QUE LE CONSTITUYEN Y LOS DICCIONARIOS DE PRECEDENTES Y
000008          *       DE SIGUIENTES DE CADA NODO.
000009          *       SI LA K-FORMULA DE ENTRADA NO FUESE VALIDA SE IMPRIMIRA UN
000010          *       MENSAJE DE ERROR.
000011          *
000012          *
000013          *
000014          *
000015                  DEFINE('INVERTIR (CADENA)CARACTER') :(FIN.INVE)                 1
000016          INVERTIR CADENA LEN(1) . CARACTER = :F(RETURN)                          2
000017                  INVERTIR = CARACTER INVERTIR :(INVERTIR)                        3
000018          FIN.INVE DEFINE('DUPLICAR(CARACTER,VECES) ' ) :(FIN.DUPL)               4
000019          DUPLICAR VECES = NE(VECES) VECES — 1 :F(RETURN)                         5
000020                  DUPLICAR = DUPLICAR CARACTER :(DUPLICAR)                        6
000021          FIN.DUL ASTERISCOS = DUPLICAR('*',126)                                  7
000022                  BLANCOS = DUPLICAR('  ',60)                                     8
000023                  PREC = LEN(1) . PRECEDENTE                                      9
000024                  SIG = LEN(1) . SIGUIENTE                                        10
000025                  K.FORM = PREC SIG                                               11
000026                  KF = 'K-FORMULA'                                                12
000027                  KF.BAS = KF 'BASE    *'                                         13
000028                  KF.ENT = KF 'DE ENTRADA'                                        14
000029                  MENS.ERROR = '?????????????' KF.ENT 'ERRONEA'                   15
000030                  BLANCOS LEN(34) . BLANCO34                                      16
000031                  CABECERA = 'NODO DICCIONARIO DE PRECEDENTES'                    17
000032          +                       BLANCO 34 'DICCIONARIO DE SIGUIENTES'           17
000033                  BLANCO34 =                                                      18
000034          LEERFICH OUTPUT =                                                       19
000035                  OUTPUT = ASTERISCOS                                             20
000036                  FICHA  = TRIM(INPUT) :F(END)                                    21
000037                  OUTPUT =                                                        22
```

```
000038                      PILA =                                                          23
000039                      OUTPUT = KF.ENT  FICHA                                          24
000040                      OUTPUT =                                                        25
000041                      FICHA = INVERTIR(FICHA)                                         26
000042      OBT.PREC        FICHA  PREC = :F(BUSC.ERR)                                      27
000043                      PILA = DIFFER (PRECEDENTE),'*') PRECEDENTE  PILA :S(OBT.PREC)   28
000044                      PILA  K.FORM = PRECEDENTE                                       29
000045                      OUTPUT = KF.BAS  PRECEDENTE  SIGUIENTE                          30
000046                      NODOS  PRECEDENTE :S(BUSC.SIG)                                  31
000047                      NODOS = NODOS  PRECEDENTE                                       32
000048      BUSC.SIG        NODOS  SIGUIENTE :S(SALTO)                                      33
000049                      NODOS = NODOS  SIGUIENTE                                        34
000050      SALTO           $(PRECEDENTE 'S') = $(PRECEDENTE 'S') SIGUIENTE '  '            35
000051                      $(SIGUIENTE 'P') = $(SIGUIENTE 'P')  PRECEDENTE '  ' :(OBT.PREC) 36
000052      BUSC.ERR        OUTPUT =                                                        37
000053                      NE(SIZE(PILA),1)    :S(ERROR)                                   38
000054                      PILA '*' :(ERROR)                                               39
000055                      OUTPUT = CABECERA                                               40
000056                      OUTPUT =                                                        41
000057      BUS.NODO        NODOS  PREC = :F(LEERFICH)                                      42
000058                      BLANCOS  LEN(60 - SIZE($(PRECEDENTE 'P'))  .  RELL              43
000059                      OUTPUT = '  ' PRECEDENTE '    ' $(PRECEDENTE 'P') RELL          44
000060      +                          $(PRECEDENTE 'S')                                    44
000061                      $(PRECEDENTE 'P') =                                             45
000062                      $(PRECEDENTE 'S') = :(BUS.NODO)                                 46
000063      ERROR           OUTPUT = MENS.ERROR                                             47
000064      BORR.DIC        NODOS  PREC = :F(LEERFICH)                                      48
000065                      $(PRECEDENTE 'P') =                                             49
000066                      $(PRECEDENTE 'S') = :(BORR.DIC)                                 50
000067      END                                                                             51

NO ERRORS DETECTED DURING COMPILATION
```

****

K-FORMULA DE ENTRADA *AA

K-FORMULA BASE *AA

NODO	DICCIONARIO DE PRECEDENTES	DICCIONARIO DE SIGUIENTES
A	A	A

****

K-FORMULA DE ENTRADA ***AA***BBA***CCABC

K-FORMULA BASE *CC
K-FORMULA BASE *CA
K-FORMULA BASE *CB
K-FORMULA BASE *BB
K-FORMULA BASE *BA
K-FORMULA BASE *BC
K-FORMULA BASE *AA
K-FORMULA BASE *AB
K-FORMULA BASE *AC

NODO	DICCIONARIO DE PRECEDENTES	DICCIONARIO DE SIGUIENTES
C	C B A	C A B
A	C B A	A B C
B	C B A	B A C

****

K-FORMULA DE ENTRADA ******ABC*D**BCDABC*CD*BADACBC****B

K-FORMULA BASE **C
K-FORMULA BASE **C
K-FORMULA BASE **C
K-FORMULA BASE **C
K-FORMULA BASE *BA
K-FORMULA BASE *CD
K-FORMULA BASE *BC
K-FORMULA BASE *BD
K-FORMULA BASE *DB

K-FORMULA BASE ∗AB
K-FORMULA BASE ∗AC
K-FORMULA BASE ∗AD
K-FORMULA BASE ∗AA
K-FORMULA BASE ∗AB
K-FORMULA BASE ∗AC

???????????????K-FORMULA DE ENTRADA ERRONEA

****

K-FORMULA DE ENTRADA ∗∗∗AAB∗B∗∗CB∗∗BDDC
K-FORMULA BASE ∗BD
K-FORMULA BASE ∗BD
K-FORMULA BASE ∗CB
K-FORMULA BASE ∗CB
K-FORMULA BASE ∗BC
K-FORMULA BASE ∗AA
K-FORMULA BASE ∗AB
K-FORMULA BASE ∗AB

???????????????K-FORMULA DE ENTRADA ERRONEA

****

K-FORMULA DE ENTRADA ∗∗∗∗AA∗∗∗∗BAB∗∗∗∗CABC∗∗∗∗DABCDDCD

K-FORMULA BASE ∗DA
K-FORMULA BASE ∗DB
K-FORMULA BASE ∗DC
K-FORMULA BASE ∗DD
K-FORMULA BASE ∗CA
K-FORMULA BASE ∗CB
K-FORMULA BASE ∗CC
K-FORMULA BASE ∗CD
K-FORMULA BASE ∗BA
K-FORMULA BASE ∗BB
K-FORMULA BASE ∗BC
K-FORMULA BASE ∗BD
K-FORMULA BASE ∗AA
K-FORMULA BASE ∗AB
K-FORMULA BASE ∗AC
K-FORMULA BASE ∗AD

NODO	DICCIONARIO DE PRECEDENTES	DICCIONARIO DE SIGUIENTES
D	D C B A	A B C D
A	D C B A	A B C D
B	D C B A	A B C D
C	D C B A	A B C D

****

K-FORMULA DE ENTRADA **AA**BAB

K-FORMULA BASE *BA
K-FORMULA BASE *BB
K-FORMULA BASE *AA
K-FORMULA BASE *AB

NODO	DICCIONARIO DE PRECEDENTES	DICCIONARIO DE SIGUIENTES
B	B A	A B
A	B A	A B

****

K-FORMULA DE ENTRADA ***AA***BAB***CABCC

K-FORMULA BASE *CA
K-FORMULA BASE *CB
K-FORMULA BASE *CC
K-FORMULA BASE *BA
K-FORMULA BASE *BB
K-FORMULA BASE *BC
K-FORMULA BASE *AA
K-FORMULA BASE *AB
K-FORMULA BASE *AC

NODO	DICCIONARIO DE PRECEDENTES	DICCIONARIO DE SIGUIENTES
C	C B A	A B C
A	C B A	A B C
B	C B A	A B C

****

100

K-FORMULA DE ENTRADA ***AA*B**CB **DBDD

K-FORMULA BASE *BD
K-FORMULA BASE *DD
K-FORMULA BASE *CB
K-FORMULA BASE *CD
K-FORMULA BASE *BC
K-FORMULA BASE *AA
K-FORMULA BASE *AB
K-FORMULA BASE *AD

NODO	DICCIONARIO DE PRECEDENTES	DICCIONARIO DE SIGUIENTES
D	D C A	B D
B	D C A	C
C	B	B D
A	A	A B D

****

APENDICE D

Programa de construcción automática de un analizador

Codificación en el lenguaje
Ensamblador del UNIVAC 9300

```
0001                                        PRINT     ON,NOGEN
0002 E   0000                       CADUA   START
0003                                *********************************************************************
0004                                *PROGRAMA DE CREACION AUTOMATICA DE UN ANALIZADOR PARA UNA GRAMATICA DADA*
0005                                *                    ESCRITA EN LA BACUS NORMAL FORM.                 *
0006                                *********************************************************************
0007                                        USING     *,0
0008                                *ALADRO  DTFRP     DEVA = 9,MODE = TRANS,OTBL =TBPU,OUAR = BROCA1
0009                                LECTORA  DTFCS     EOFA = SWITCH
0010                                IMPRESO  DTFPR     DEVA = 10,PROV = YES
0011     037A    45E00000           INICIAR  OPEN      LECTORA
0012     037E    45E00136                    OPEN      IMPRESO
0013                                *           OPEN      TALADRO
0014     0382    47F003C0                    B         EMPEZAR
0015                                *
0016     0386    45E00008075C       LEER1    GET       LECTORA, FICHA 1
0017     038C    47F0A000                    B         0(,10)
0018                                *
0019     0390    45E0000807AC       LEER2    GET       LECTORA, FICHA 2
0020     0396    47F0A000                    B         0(,10)
0021                                *
0022     039A    45E0014607FD       IMPRIMO  PUT       IMPRESO, PAPEL
0023     03A0    D28307FD07FC                MVC       PAPEL, PAPEL-1
0024     03A6    47F0A000                    B         0(,10)
0025                                *
0026                                *ERFORO  PUT       TALADRO, BROCA
0027                                *     PARA PERFORAR QUITAR LA ETIQUETA A LA FICHA SIGUIENTE
0028     03AA    D24F06BB06BA       PERFORO  MVC       BROCA, BROCA-1
0029     03B0    47F0A000                    B         0(,10)
0030                                *
0031     03B4    45E00004           FIN      CLOSE     LECTORA
0032     03B8    45E0013A                    CLOSE     IMPRESO
0033                                *           CLOSE     TALADRO
0034     03BC    A9001237                    HPR       X'1237'
0035                                **
0036                                *** CARGAR LA TABLA EN MEMORIA
0037                                **
0038     03C0    489008E2           EMPEZAR  LH        9,DTABLA
0039     03C4    9601075B                    OI        SW,X'01'          PONER SW EN ON
0040     03C8    45A00386           SIGUE    BAL       10, LEER1         LEER FICHA1
0041     03CC    957A0766                    CLI       FICHA1+10,X'7A'   MIRAR SI COLUM. 11 ES DOSPUNTOS
0042     03D0    478003E2                    BE        ALMACEN
0043     03D4    D24F9000075C                MVC       0(80,9), FICHA1   METER FICHA1 EN TABLA
```

```
0044  03DA  AA9008E4                AH      9, = H'80'          SUMAR AL REG. 9 OCHENTA
0045  03DE  47F003C8      B         SIGUE
0046                      **
0047                      *** LECTURA DE LA GRAMATICA Y PROCESO
0048                      **
0049  03E2  40900884      ALMACEN   STH     9,DIRECMAX
0050  03E6  AB9008E4                SH      9, = H'80'
0051  03EA  40900882                STH     9,DIRECMIN
0052  03EE  48B00884      GRAMATIC  LH      11,DIRECMAX
0053  03F2  D24FB000075C            MVC     0(80,11), FICHA1
0054  03F8  AAB008E4                AH      11, = H'80'
0055  03FC  45A00390      LEER      BAL     10, LEER2
0056  0400  955C07AC                CLI     FICHA2,X'5C'        MIRAR SI ES FICHA DE CONT.
0057  0404  4770041A                BNE     BLANQUEO
0058  0408  D24EB00007AD            MVC     0(79,11), FICHA2+1
0059  040E  AAB008E6                AH      11, = H'79'
0060  0412  47F003FC                B       LEER
0061  0416  9400075B      SWITCH    NI      SW,X'00'            PONER SW EN OFF
0062  041A  D202B00008F2  BLANQUEO  MVC     0(3,11), = C'  '    PONER TRES BLANCOS AL FINAL
0063  0420  D24F06BB06BA            MVC     BROCA, BROCA-1      PONER A BLANCOS BROCA
0064  0426  D28307FD07FC            MVC     PAPEL, PAPEL-1      BORRAR SALIDA DE IMPRESORA
0065  042C  D20506C408F5            MVC     BROCA+9(6), = C'RUTINA'
0066  0432  D20706CB075D            MVC     BROCA+16(8), FICHA1+1
0067  0438  D24F07FD06BB            MVC     PAPEL(80), BROCA
0068  043E  45A003AA                BAL     10,PERFORO          IR A PERFORAR
0069  0442  45A0039A                BAL     10,IMPRIMO
0070                      **
0071                      *** COMPROBAR SI HAY ALTERNATIVAS
0072                      **
0073  0446  48B00884                LH      11,DIRECMAX
0074  044A  AAB008E8                AH      11, = H'13'
0075  044E  957A0767                CLI     FICHA1+11,X'7A'     MIRAR SI HAY ALTERNATIVAS
0076  0452  4780045E                BE      SIALTER             SI, IR A MANEJO ALTERNATIVAS
0077  0456  458005D8                BAL     8,NOALTER           IR A NO ALTERNATIVAS
0078  045A  47F00584                B       MOVER
0079  045E  D20208860889  SIALTER   MVC     LABELESK(3),LABELESX  GUARDAR EL VALOR ACTUAL DE X
0080  0464  9540B000      COMPARAR  CLI     0(11),C' '          MIRAR SI ES TERMINAL
0081  0468  4780058E                BE      TERMINAL
0082                      **
0083                      *** TRATAMIENTO DE CLASE SINTACTICA EN TABLA
0084                      **
0085  046C  48C008E2                LH      12,DTABLA
0086  0470  D509B000C000  BUSCAR    CLC     0(10,11),0(12)      COMPARAR FICHA CON TABLA
0087  0476  4780049E                BE      SUMAR
0088  047A  49C00882                CH      12,DIRECMIN         COMPARAR REG. IND. CON ULT. DIREC.
```

```
0089  047E  47400496              BL    AUMENTAR
0090                              MVC   PAPEL(61), = C'PRODUCCION   DE LA GRAMATICA NO ENCO*
0091  0482  D23C07FD08FB                NTRADA EN LA TABLA'
0092  0488  D2070808B001            MVC   PAPEL + 11(8),1(11)
0093  048E  45A0039A                BAL   10,IMPRIMO
0094  0492  47F003B4                B     FIN
0095  0496  AAC008E4   AUMENTAR     AH    12, = H'80'          SUMAR 80 AL REGISTRO INDICE
0096  049A  47F00470                B     BUSCAR
0097                    **
0098                    *** CLASE SINTACTICA ENCONTRADA EN TABLA
0099                    **
0100  049E  AAC008EA    SUMAR       AH    12, = H'11'          SUMAR 11 AL REGISTRO INDICE
0101  04A2  FA2008890938            AP    LABELESX, = P'1'
0102  04A8  954AC000    DECIDES     CLI   0(12),X'4A'
0103  04AC  477004B4                BNE   SUMALABE
0104  04B0  47F004EE                B     CLASESIG             IGNORAR TRATAMIENTO
0105  04B4  D21206C40939  SUMALABE  MVC   BROCA+9(19), = C'DECIDE      , = C' '  '' '
0106  04BA  D20006D5C000            MVC   BROCA+26(1),0(12)
0107  04C0  D20506CB088C            MVC   BROCA+16(6),MASCARA1 EDITAR ETIQUETAS
0108  04C6  DE0506CB0889            ED    BROCA+16(6),LABELESX
0109  04CC  D20006CB0912            MVC   BROCA+16(1), = C'L'
0110  04D2  D24F07FD06BB  LLEVAR     MVC   PAPEL(80), BROCA
0111  04D8  45A003AA                BAL   10,PERFORO
0112  04DC  45A0039A                BAL   10,IMPRIMO
0113  04E0  AAC008EC                AH    12, = H'1'           SUMAR UNO A REGISTRO INDICE
0114  04E4  D501C00008F2            CLC   0(2,12), = C' '      MIRAR SI FINAL DE CARACTERES
0115  04EA  477004A8                BNE   DECIDES              SI NO, VOLVER A PERFORAR
0116  04EE  AAB008EE    CLASESIG    AH    11, = H'10'          TOMAR LA CLASE SIGUIENTE
0117                    **
0118                    *** EXPLORAR HASTA LA SIGUIENTE CLASE SINTACTICA O TRES BLANCOS
0119                    **
0120  04F2  D502B00008F2  FINAL      CLC   0(3,11), = C'   '    MIRAR SI ES FINAL
0121  04F8  47800534                BE    CARGAR
0122  04FC  954FB000                CLI   0(11),X'4F'          MIRAR SI ES BARRA
0123  0500  4780052C                BE    AA
0124  0504  955AB000                CLI   0(11),X'5A'          MIRAR SI ES ADMIRACION
0125  0508  4780052C                BE    AA
0126  050C  954CB000                CLI   0(11),X'4C'          MIRAR SI ES 'MENOR QUE'
0127  0510  47700524                BNE   BB
0128  0514  956EB009                CLI   9(11),X'6E'          MIRAR SI ES 'MAYOR QUE'
0129  0518  47700524                BNE   BB
0130  051C  AAB008EE                AH    11, = H'10'
0131  0520  47F004F2                B     FINAL
0132  0524  AAB008EC    BB          AH    11, = H'1'           SUMAR UNO AL POINTER
```

```
0133  0528  47F004F2                B       FINAL
0134  052C  AAB008EC  AA            AH      11,=H'1'          SUMAR UNO AL POINTER
0135  0530  47F00464                B       COMPARAR
0136                  **
0137                  *** PERFORAR FICHAS EQUS Y LLAMADA A NO ALTERNATIVAS
0138                  **
0139  0534  48B00884  CARGAR        LH      11,DIRECMAX
0140  0538  AAB008E8                AH      11,=H'13'
0141  053C  FA2008860938            AP      LABELESK,=P'1'
0142  0542  D21006BB094C  EQUS      MVC     BROCA(17),=C'     EQU   *   '
0143  0548  D20506BB088C            MVC     BROCA(6),MASCARA1 EDITAR ETIQUETAS
0144  054E  DE0506BB0886            ED      BROCA(6),LABELESK
0145  0554  D20006BB0912            MVC     BROCA(1),=C'L'
0146  055A  D24F07FD06BB            MVC     PAPEL(80),BROCA
0147  0560  45A003AA                BAL     10,PERFORO
0148  0564  45A0039A                BAL     10,IMPRIMO
0149  0568  458005D8                BAL     8,NOALTER
0150  056C  FA2008860938            AP      LABELESK,=P'1'    SUMAR UNO A CONTADOR AUXILIAR
0151  0572  D50208860889            CLC     LABELESK(3),LABELESX COMPARAR CONTADORES
0152  0578  47D00542                BNH     EQUS
0153  057C  9101075B                TM      SW,X'01'          PROBAR SI ULTIMA FICH. PROCESADA
0154  0580  478003B4                BZ      FIN
0155  0584  D24F075C07AC  MOVER     MVC     FICHA1(80),FICHA2 LLEVAR FICHA 2 A FICHA 1
0156  058A  47F003EE                B       GRAMATIC
0157                  **
0158                  *** TRATAMIENTO DE TERMINALES
0159                  **
0160  058E  AAB008EC  TERMINAL      AH      11,=H'1'
0161  0592  954AB000                CLI     0(11),X'4A'
0162  0596  477005A2                BNE     SUMAUNO
0163  059A  4580069E                BAL     8,SALIR
0164  059E  47F004F2                B       FINAL
0165  05A2  FA2008890938  SUMAUNO    AP      LABELESX,=P'1'
0166  05A8  D21206C40939            MVC     BROCA+9(19),=C'DECIDE     ,=C''  ' ' '
0167  05AE  D20006D5B000            MVC     BROCA+26(1),0(11)
0168  05B4  D20506CB088C            MVC     BROCA+16(6),MASCARA1 EDITAR ETIQUETAS
0169  05BA  DE0506CB0889            ED      BROCA+16(6),LABELESX PERFORAR DECIDES
0170  05C0  D20006CB0912            MVC     BROCA+16(1),=C'L'
0171  05C6  D24F07FD06BB            MVC     PAPEL(80),BROCA
0172  05CC  45A003AA                BAL     10,PERFORO
0173  05D0  45A0039A                BAL     10,IMPRIMO
0174  05D4  47F004F2                B       FINAL
0175                  **
0176                  *** RUTINA DE TRATAMIENTO DE NO ALTERNATIVAS
```

```
0177                          **
0178   05D8   9540B000       NOALTER  CLI    0(11),C' '              MIRAR SI ES TERMINAL
0179   05DC   4780064A                BE     TERMI
0180   05E0   AAB008EE       SUMO     AH     11, = H'10'             SUMAR DIEZ AL POINTER
0181   05E4   D502B00008F2             CLC    0(3,11), = C' '         COMPARAR CON TRES BLANCOS
0182   05EA   47800624                BE     ENTRAR
0183   05EE   954FB000                CLI    0(11),X'4F'             COMPARAR CON BARRA
0184   05F2   47800624                BE     ENTRAR
0185   05F6   955AB000                CLI    0(11),X'5A'             CONPARAR CON ADMIRACION
0186   05FA   47800624                BE     ENTRAR
0187   05FE   ABB008F0                SH     11, = H'9'              RESTAR NUEVE AL POINTER
0188   0602   D20506C4095D            MVC    BROCA+9(6), = C'LLAMA'
0189   0608   D20706CBB000            MVC    BROCA+16(8),0(11)
0190   060E   D24F07FD06BB            MVC    PAPEL(80), BROCA
0191   0614   45A003AA                BAL    10,PERFORO
0192   0618   45A0039A                BAL    10,IMPRIMO
0193   061C   AAB008F0                AH     11, = H'9'              SUMAR NUEVE AL POINTER
0194   0620   47F005D8                B      NOALTER
0195   0624   D20606C40963   ENTRAR   MVC    BROCA+9(7), = C'ENTRAR'
0196   062A   ABB008F0                SH     11, = H'9'              RESTAR NUEVE AL POINTER
0197   062E   D20706CBB000            MVC    BROCA+16(8),0(11)       PERFORAR ENTRAR
0198   0634   D24F07FD06BB            MVC    PAPEL(80), BROCA
0199   063A   45A003AA                BAL    10,PERFORO
0200   063E   45A0039A                BAL    10,IMPRIMO
0201   0642   AAB008EE                AH     11, = H'10'             SUMAR DIEZ AL POINTER
0202   0646   47F08000                B      0(,8)                   DEVOLVER CONTROL AL PROGRAMA
0203   064A   AAB008EC       TERMI    AH     11, = H'1'              SUMAR UNO AL POINTER
0204   064E   D502B00008F2             CLC    0(3,11) = C'  '         MIRAR SI TRES BLANCOS
0205   0654   4780069E                BE     SALIR
0206   0658   955AB000                CLI    0(11),X'5A'             MIRAR SI ES ADMIRACION
0207   065C   4780069E                BE     SALIR
0208   0660   954FB000                CLI    0(11),X'4F'             MIRAR SI ES BARRA
0209   0664   4780069E                BE     SALIR
0210   0668   954AB000                CLI    0(11),X'4A'             MIRAR SI CARACTER DE VACIO
0211   066C   4780069E                BE     SALIR
0212   0670   954CB000                CLI    0(11),X'4C'             MIRAR SI SIGNO 'MENOR QUE'
0213   0674   47700680                BNE    TEST
0214   0678   956EB009                CLI    9(11),X'6E'             MIRAR SI SIGNO 'MAYOR QUE'
0215   067C   478005E0                BE     SUMO
0216   0680   D20A06C4096A   TEST     MVC    BROCA+9(11), = C'TEST     = C'  ' ' '
0217   0686   D20006CDB000            MVC    BROCA+18(1),0(11)
0218   068C   D24F07FD06BB            MVC    PAPEL(80), BROCA
0219   0692   45A003AA                BAL    10,PERFORO
0220   0696   45A0039A                BAL    10,IMPRIMO
```

```
0221  069A  47F0064A              B       TERMI
0222  069E  D20406C40976  SALIR   MVC     BROCA+9(5), = C'SALIR'
0223  06A4  D24F07FD06BB          MVC     PAPEL(80), BROCA
0224  06AA  45A003AA              BAL     10,PERFORO
0225  06AE  45A0039A              BAL     10,IMPRIMO
0226  06B2  AAB008EC              AH      11, = H'1'
0227  06B6  47F08000              B       0(,8)
0228                      **
0229                      *** DEFINICION DE CONSTANTES Y AREAS DE ALMACENAMIENTO
0230                      **
0231  06BA  40                    DC      CL1' '
0232  06BB                BROCA   DS      CL80
0233  070B                BROCA1  DS      CL80
0234  075B                SW      DS      CL1
0235  075C                FICHA1  DS      CL80
0236  07AC                FICHA2  DS      CL80
0237  07FC  40                    DC      CL1' '
0238  07FD                PAPEL   DS      CL132
0239  0882                DIRECMIN DS     H
0240  0884                DIRECMAX DS     H
0241  0886  00000C        LABELESK DC     PL3'0'
0242  0889  00000C        LABELESX DC     PL3'0'
0243  088C  F02020202020  MASCARA1 DC     X'F02020202020'
0244  0892                FICHA   DS      CL80
0245  08E2  0A7F          DTABLA  DC      Y(TABLA+260)
0246  08E4                        LTORG
      08E4  0050
      08E6  004F
      08E8  000D
      08EA  000B
      08EC  0001
      08EE  000A
      08F0  0009
      08F2  404040D9E4E3C9D5
      08FA  C1D7D9D6C4E4C3C3
      0902  C9D6D54040404040
      090A  4040404040C4C540
      0912  D3C140C7D9C1D4C1
      091A  E3C9C3C140D5D640
      0922  C5D5C3D6D5E3D9C1
      092A  C4C140C5D540D3C1
      0932  40E3C1C2D3C11CC4
      093A  C5C3C9C4C5404040
      0942  404040406B7EC37D
```

```
                094A    407D404040404040
                0952    404040C5D8E44040
                095A    405C40D3D3C1D4C1
                0962    40C5D5E3D9C1D940
                096A    E3C5E2E340407EC3
                0972    7D407D40E2C1D3C9
                097A    D9
0247            097B                    TABLA   DS      CL1
0248            097C    00000000037A            END     INICIAR
```

APENDICE E

Obtención automática del cierre transitivo de la relación inicial para nuestra gramática fuente

Datos de entrada al proceso

UNIVAC

HOJA DE PROGRAMACION

UNIVAC

PROGRAMA _____ PROGRAMADOR _____ FECHA _____ PAGINA ____ DE ____ PAGINAS

1 ETIQUETA 10	OPERACION 20	30 OPERANDO 40	COMENTARIOS 50 60 80
<INSTRUC>	::=	<ESINST> ! <MDAR> ! <MDINST> ! <ARCT> ! <ARINST> ! <CTINST>	
<ESINST>	::=	LEER<CK> ! ESCRIBIR<CK>	
<CK>	::=	!<CADENA><CT>	
<MDAR>	::=	!D<MDARCON>	
<MDARCON>	::=	POSITAR<CK> ! IVIDIR<CK>	
<MDINST>	::=	!TRASLADAR<CADENA><CONTINS>	
<ARCT>	::=	!S<CU>	
<CU>	::=	UMAR<CADENA><CONTINS> ! IEL<CK><OPRELA><CONTCT>	
<CONTINS>	::=	<CT> ! <CN>	
<CT>	::=	(<DIRECC>) ! ε	
<CN>	::=	!=<NUMERO>	
<CS>	::=	!:<DIRECC>	
<ARINST>	::=	RESTAR<CADENA><CONTINS> ! MULTIPLICAR<CADENA><CONTINS>	
<CTINST>	::=	P<CR> ! FIN ! HACER<NUMERO><CADENA> ! <NUMERO><CADENA>	
<CR>	::=	ARAR ! ONER<CADENA><CS>	
<CONTCT>	::=	0<CADENA><CS> ! (AC)<CADENA><CS>	
<OPRELA>	::=	< ! = ! >	
<DIRECC>	::=	!=<DIGITO><DIGITO>	
<CADENA>	::=	!=<CARACT><CONTCAD>	
<CONTCAD>	::=	=<CARACT><CONTCAD> ! ε	
<CARACT>	::=	A ! B ! C ! D ! E ! F ! G ! H ! I ! J ! K ! L ! M ! N ! O ! P ! Q ! R ! S ! T ! U ! V ! W ! Y ! Z ! X	
<NUMERO>	::=	!=<DIGITO><CONTNUME>	
<CONTNUME>	::=	=<DIGITO><CONTNUME> ! ε	
<DIGITO>	::=	0 ! 1 ! 2 ! 3 ! 4 ! 5 ! 6 ! 7 ! 8 ! 9	

Resultados obtenidos para la gramática de nuestro lenguaje fuente

<INSTRUC >	<ESINST >					<MDAR >				<MDINST >				<ARCT >				<ARINST >				<CTINST >				
<ESINST >	L		E																							
<CK >	<CADENA >																									
<MDAR >	D																									
<MDARCON >	E		I																							
<MDINST >	T																									
<ARCT >	S																									
<CU >	U		I																							
<CONTINS >	<CT		>	<CN		>																				
<CT >	(		¢																							
<CN >	=																									
<CS >	:																									
<ARINST >	R		M																							
<CTINST >	P		F		H																					
<CR >	A		O																							
<CONTCT >	0		(																							
<OPRELA >	<		=		>																					
<DIRECC >	<DIGITO >																									
<CADENA >	<CARACT >																									
<CARACT >	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W			
	Y	Z	X																							
<NUMERO >	<DIGITO >																									
<CONTNUME>	<DIGITO >																									
<DIGITO >	0	1	2	3	4	5	6	7	8	9																

	<CARACT >	<CADENA >	<CK >	⟩—┐					
A	<CR >	<CARACT >	<CADENA >	<CK >	⟩—┐				
B	<CARACT >	<CADENA >	<CK >	⟩—┐					
C	<CARACT >	<CADENA >	<CK >	⟩—┐					
D	<MDAR >	<CARACT >	<INSTRUC >	<CADENA >	<CK >	⟩—┐			
E	<ESINST >	<MDARCON >	<CARACT >	<INSTRUC >	<CADENA >	<CK >	⟩—┐		
F	<CTINST >	<CARACT >	<INSTRUC >	<CADENA >	<CK >	⟩—┐			
G	<CARACT >	<CADENA >	<CK >	⟩—┐					
H	<CTINST >	<CARACT >	<INSTRUC >	<CADENA >	<CK >	⟩—┐			
I	<MDARCON	<CU >	<CARACT >	<CADENA >	<CK >	⟩—┐			
¢	<CT	<CONTNUME >	<CONTINS >	⟩—┐					
∠	<OPRELA	⟩—┐							
(	<CT >	<CONTCT >	<CONTINS >	⟩—┐					
J	<CARACT >	<CADENA >	<CK >	⟩—┐					
K	<CARACT >	<CADENA >	<CK >	⟩—┐					
L	<ESINST >	<CARACT >	<INSTRUC >	<CADENA >	<CK >	⟩—┐			
M	<ARINST >	<CARACT >	<INSTRUC >	<CADENA >	<CK >	⟩—┐			
N	<CARACT >	<CADENA >	<CK >	⟩—┐					
O	<CR >	<CARACT >	<CADENA >	<CK >	⟩—┐				
P	<CTINST >	<CARACT >	<INSTRUC >	<CADENA >	<CK >	⟩-			
Q	<CARACT >	<CADENA >	<CK >	⟩—┐					

```
        R    <ARINST   ><CARACT   ><INSTRUC   ><CADENA   ><CK       >—┐
        S    <ARCT     ><CARACT   ><INSTRUC   ><CADENA   ><CK       >—┐
        T    <MDINST   ><CARACT   ><INSTRUC   ><CADENA   ><CK       >—┐
        U    <CU       ><CARACT   ><CADENA    ><CK       >—┐
        V    <CARACT   ><CADENA   ><CK       >—┐
        W    <CARACT   ><CADENA   ><CK       >—┐
        X    <CARACT   ><CADENA   ><CK       >—┐
        Y    <CARACT   ><CADENA   ><CK       >—┐
        Z    <CARACT   ><CADENA   ><CK       >—┐
        >    <OPRELA   >—┐
        0    <CONTCT   ><DIGITO   ><DIRECC   ><NUMERO   ><CONTNUME   >—┐
        1    <DIGITO   ><DIRECC   ><NUMERO   ><CONTNUME>—┐
        2    <DIGITO   ><DIRECC   ><NUMERO   ><CONTNUME>—┐
        3    <DIGITO   ><DIRECC   ><NUMERO   ><CONTNUME>—┐
        4    <DIGITO   ><DIRECC   ><NUMERO   ><CONTNUME>—┐
        5    <DIGITO   ><DIRECC   ><NUMERO   ><CONTNUME>—┐
        6    <DIGITO   ><DIRECC   ><NUMERO   ><CONTNUME>—┐
        7    <DIGITO   ><DIRECC   ><NUMERO   ><CONTNUME>—┐
        8    <DIGITO   ><DIRECC   ><NUMERO   ><CONTNUME>—┐
        9    <DIGITO   ><DIRECC   ><NUMERO   ><CONTNUME>—┐
        :    <CS       >—┐
        =    <CN       ><OPRELA   ><CONTINS   >—┐
<INSTRUC   >  DEFHLMPRST
<ESINST    >  EL
<CK        >   ABCDEFGHIJKLMNOPQRSTUVWXYZ
<MDAR      >  D
<MDARCON   >  EI
<MDINST    >  T
<ARCT      >  S
<CU        >  IU
<CONTINS   >  ¢( =
<CT        >  ¢(
<CN        >  =
<CS        >  :
<ARINST    >  MR
<CTINST    >  FHP
<CR        >  AO
<CONTCT    >  (0
<OPRELA    >  <> =
<DIRECC    >  0123456789
<CADENA    >   ABCDEFGHIJKLMNOPQRSTUVWXYZ
<CARACT    >   ABCDEFGHIJKLMNOPQRSTUVWXYZ
<NUMERO    >  0123456789
<CONTNUME  >  ¢0123456789
<DIGITO    >  0123456789
```

APENDICE F

Analizadores sintácticos

Construcción automatizada
Datos de entrada al proceso

<INSTRUC > DEFHLMPRST
<ESINST > EL
<CK > ABCDEFGHIJKLMNOPQRSTUVWXYZ
<MDAR > D
<MDARCON > EI
<MDINST > T
<ARCT > S
<CU > IU
<CONTINS > ¢(=
<CT > ¢(
<CN > =
<CS > :
<ARINST > MR
<CTINST > FHP
<CR > AO
<CONTCT > (O
<OPRELA > < > =
<DIRECC > 0123456789
<CADENA > ABCDEFGHIJKLMNOPQRSTUVWXYZ
<CARACT > ABCDEFGHIJKLMNOPQRSTUVWXYZ
<NUMERO > 0123456789
<CONTNUME > ¢0123456789
<DIGITO > 0123456789
<CONTCAD > ABCDEFGHIJKLMNOPQRSTUVWXYZ

UNIVAC

HOJA DE PROGRAMACION

UNIVAC

PROGRAMA _____ PROGRAMADOR _____ FECHA _____ PAGINA ____ DE ____ PAGINAS

1 ETIQUETA 10	OPERACION 20	OPERANDO 30 40	COMENTARIOS 50 60 80
<INSTRUC>	::=	<ESINST> | <MDAR> | <MDINST> | <ARCT> | <ARINST> ! <CTINST>	
<ESINST>	::=	LEER<CK> ! ESCRIBIR<CK>	
<CK>	::=	<CADENA> <CT>	
<MDAR>	::=	D<MDARCON>	
<MDARCON>	::=	POSITAR<CK> ! IVIDIR<CK>	
<MDINST>	::=	TRASLADAR<CADENA> <CONTINS>	
<ARCT>	::=	S<CU>	
<CU>	::=	UMAR<CADENA> <CONTINS> ! IEL<CK> <OPRELA> <CONTCT>	
<CONTINS>	::=	<CT> ! <CN>	
<CT>	::=	(<DIRECC>) ! ε	
<CN>	::=	=<NUMERO>	
<CS>	::=	:<DIRECC>	
<ARINST>	::=	RESTAR<CADENA> <CONTINS> ! MULTIPLICAR<CADENA> <CONTINS>	
<CTINST>	::=	P<CR> ! FIN ! HACER<NUMERO> <CADENA> <NUMERO> <CADENA>	
<CR>	::=	ARAR ! ONER<CADENA> <CS>	
<CONTCT>	::=	0<CADENA> <CS> ! (AC)<CADENA> <CS>	
<OPRELA>	::=	< ! = ! >	
<DIRECC>	::=	<DIGITO> <DIGITO>	
<CADENA>	::=	<CARACT> <CONTCAD>	
<CONTCAD>	::=	<CARACT> <CONTCAD> ! ε	
<CARACT>	::=	A B C D E F G H I J K L M N O P Q R S T U V *W Y Z ! X	
<NUMERO>	::=	<DIGITO> <CONTNUME>	
<CONTNUME>	::=	<DIGITO> <CONTNUME> ! ε	
<DIGITO>	::=	0 1 2 3 4 5 6 7 8 ! 9	

Analizador resultado del proceso automático

```
        RUTINA  INSTRUC
        DECIDE  L00001, = C'E'
        DECIDE  L00001, = C'L'
        DECIDE  L00002, = C'D'
        DECIDE  L00003, = C'T'
        DECIDE  L00004, = C'S'
        DECIDE  L00005, = C'M'
        DECIDE  L00005, = C'R'
        DECIDE  L00006, = C'F'
        DECIDE  L00006, = C'H'
        DECIDE  L00006, = C'P'
L00001  EQU  *
        ENTRAR  ESINST
L00002  EQU  *
        ENTRAR  MDAR
L00003  EQU  *
        ENTRAR  MDINST
L00004  EQU  *
        ENTRAR  ARCT
L00005  EQU  *
        ENTRAR  ARINST
L00006  EQU  *
        ENTRAR  CTINST
        RUTINA  ESINST
        DECIDE  L00007, = C'L'
        DECIDE  L00008, = C'E'
L00007  EQU  *
        TEST  = C'L'
        TEST  = C'E'
        TEST  = C'E'
        TEST  = C'R'
        ENTRAR  CK
L00008  EQU  *
        TEST  = C'E'
        TEST  = C'S'
        TEST  = C'C'
        TEST  = C'R'
        TEST  = C'I'
        TEST  = C'B'
        TEST  = C'I'
        TEST  = C'R'
        ENTRAR  CK
        RUTINA  CK
        LLAMA   CADENA
```

```
         ENTRAR CT
         RUTINA MDAR
         TEST = C'D'
         ENTRAR MDARCON
         RUTINA MDARCON
         DECIDE L00009, = C'E'
         DECIDE L00010, = C'I'
L00009   EQU *
         TEST = C'E'
         TEST = C'P'
         TEST = C'O'
         TEST = C'S'
         TEST = C'I'
         TEST = C'T'
         TEST = C'A'
         TEST = C'R'
         ENTRAR CK
L00010   EQU *
         TEST = C'I'
         TEST = C'V'
         TEST = C'I'
         TEST = C'D'
         TEST = C'I'
         TEST = C'R'
         ENTRAR CK
         RUTINA MDINST
         TEST = C'T'
         TEST = C'R'
         TEST = C'A'
         TEST = C'S'
         TEST = C'L'
         TEST = C'A'
         TEST = C'D'
         TEST = C'A'
         TEST = C'R'
         LLAMA CADENA
         ENTRAR CONTINS
         RUTINA ARCT
         TEST = C'S'
         ENTRAR CU
         RUTINA CU
         DECIDE L00011, = C'U'
         DECIDE L00012, = C'I'
```

```
L00011  EQU  *
        TEST  = C'U'
        TEST  = C'M'
        TEST  = C'A'
        TEST  = C'R'
        LLAMA  CADENA
        ENTRAR CONTINS
L00012  EQU  *
        TEST  = C'I'
        TEST  = C'E'
        TEST  = C'L'
        LLAMA  CK
        LLAMA  OPRELA
        ENTRAR CONTCT
        RUTINA CONTINS
        DECIDE L00014, = C'='
L00013  EQU  *
        ENTRAR CT
L00014  EQU  *
        ENTRAR CN
        RUTINA CT
        DECIDE L00015, = C'('
        SALIR
L00015  EQU  *
        TEST  = C'('
        LLAMA  DIRECC
        TEST  = C')'
        SALIR
        RUTINA  CN
        TEST  =C'='
        ENTRAR NUMERO
        RUTINA CS
        TEST  = C':'
        ENTRAR DIRECC
        RUTINA ARINST
        DECIDE L00016, = C'R'
        DECIDE L00017, = C'M'
L00016  EQU  *
        TEST  = C'R'
        TEST  = C'E'
        TEST  = C'S'
        TEST  = C'T'
        TEST  = C'A'
        TEST  = C'R'
```

```
          LLAMA  CADENA
          ENTRAR CONTINS
L00017    EQU  *
          TEST  = C'M'
          TEST  = C'U'
          TEST  = C'L'
          TEST  = C'T'
          TEST  = C'I'
          TEST  = C'P'
          TEST  = C'L'
          TEST  = C'I'
          TEST  = C'C'
          TEST  = C'A'
          TEST  = C'R'
          LLAMA  CADENA
          ENTRAR CONTINS
          RUTINA CTINST
          DECIDE L00018, = C'P'
          DECIDE L00019, = C'F'
          DECIDE L00020, = C'H'
L00018    EQU  *
          TEST  =C'P'
          ENTRAR CR
L00019    EQU  *
          TEST  = C'F'
          TEST  = C'I'
          TEST  = C'N
          SALIR
L00020    EQU  *
          TEST  = C'H'
          TEST  = C'A'
          TEST  = C'C'
          TEST  = C'E'
          TEST  = C'R'
          LLAMA  NUMERO
          LLAMA  CADENA
          LLAMA  NUMERO
          ENTRAR CADENA
          RUTINA CR
          DECIDE L00021, = C'A'
          DECIDE L00022, = C'O'
L00021    EQU  *
          TEST  = C'A'
          TEST  = C'R'
```

```
          TEST  = C'A'
          TEST  = C'R'
          SALIR
L00022    EQU   *
          TEST  = C'O'
          TEST  = C'N'
          TEST  = C'E'
          TEST  = C'R'
          LLAMA  CADENA
          ENTRAR CS
          RUTINA  CONTCT
          DECIDE  L00023, = C'0'
          DECIDE  L00024, = C'('
L00023    EQU   *
          TEST  = C'0'
          LLAMA  CADENA
          ENTRAR CS
L00024    EQU   *
          TEST  = C'('
          TEST  = C'A'
          TEST  = C'C'
          TEST  = C')'
          LLAMA  CADENA
          ENTRAR CS
          RUTINA  OPRELA
          DECIDE  L00025, = C'<'
          DECIDE  L00026, = C'='
          DECIDE  L00027, = C'>'
L00025    EQU   *
          TEST  = C'<'
          SALIR
L00026    EQU   *
          TEST  = C'='
          SALIR
L00027    EQU   *
          TEST  = C'>'
          SALIR
          RUTINA  DIRECC
          LLAMA  DIGITO
          ENTRAR DIGITO
          RUTINA  CADENA
          LLAMA CARACT
          ENTRAR CONTCAD
          RUTINA CONTCAD
```

```
          DECIDE  L00028, = C' '
          DECIDE  L00028, = C'A'
          DECIDE  L00028, = C'B'
          DECIDE  L00028, = C'C'
          DECIDE  L00028, = C'D'
          DECIDE  L00028, = C'E'
          DECIDE  L00028, = C'F'
          DECIDE  L00028, = C'G'
          DECIDE  L00028, = C'H'
          DECIDE  L00028, = C'I'
          DECIDE  L00028, = C'J'
          DECIDE  L00028, = C'K'
          DECIDE  L00028, = C'L'
          DECIDE  L00028, = C'M'
          DECIDE  L00028, = C'N'
          DECIDE  L00028, = C'O'
          DECIDE  L00028, = C'P'
          DECIDE  L00028, = C'Q'
          DECIDE  L00028, = C'R'
          DECIDE  L00028, = C'S'
          DECIDE  L00028, = C'T'
          DECIDE  L00028, = C'U'
          DECIDE  L00028, = C'V'
          DECIDE  L00028, = C'W'
          DECIDE  L00028, = C'X'
          DECIDE  L00028, = C'Y'
          DECIDE  L00028, = C'Z'
          SALIR
L00028    EQU  *
          LLAMA   CARACT
          ENTRAR  CONTCAD
          RUTINA  CARACT
          DECIDE  L00029, = C'A'
          DECIDE  L00030, = C'B'
          DECIDE  L00031, = C'C'
          DECIDE  L00032, = C'D'
          DECIDE  L00033, = C'E'
          DECIDE  L00034, = C'F'
          DECIDE  L00035, = C'G'
          DECIDE  L00036, = C'H'
          DECIDE  L00037, = C'I'
          DECIDE  L00038, = C'J'
          DECIDE  L00039, = C'K'
          DECIDE  L00040, = C'L'
```

```
            DECIDE  L00041, = C'M'
            DECIDE  L00042, = C'N'
            DECIDE  L00043, = C'O'
            DECIDE  L00044, = C'P'
            DECIDE  L00045, = C'Q'
            DECIDE  L00046, = C'R'
            DECIDE  L00047, = C'S'
            DECIDE  L00048, = C'T'
            DECIDE  L00049, = C'U'
            DECIDE  L00050, = C'V'
            DECIDE  L00051, = C'W'
            DECIDE  L00052, = C' '
            DECIDE  L00053, = C'Y'
            DECIDE  L00054, = C'Z'
            DECIDE  L00055, = C'X'
L00029      EQU  *
            TEST  = C'A'
            SALIR
L00030      EQU  *
            TEST  = C'B'
            SALIR
L00031      EQU  *
            TEST  = C'C'
            SALIR
L00032      EQU  *
            TEST  = C'D'
            SALIR
L00033      EQU  *
            TEST  = C'E'
            SALIR
L00034      EQU  *
            TEST  = C'F'
            SALIR
L00035      EQU  *
            TEST  = C'G'
            SALIR
L00036      EQU  *
            TEST  = C'H'
            SALIR
L00037      EQU  *
            TEST  = C'I'
            SALIR
L00038      EQU  *
            TEST  = C'J'
```

```
          SALIR
L00039    EQU  *
          TEST = C'K'
          SALIR
L00040    EQU  *
          TEST = C'L'
          SALIR
L00041    EQU  *
          TEST = C'M'
          SALIR
L00042    EQU  *
          TEST = C'N'
          SALIR
L00043    EQU  *
          TEST = C'O'
          SALIR
L00044    EQU  *
          TEST = C'P'
          SALIR
L00045    EQU  *
          TEST = C'Q'
          SALIR
L00046    EQU  *
          TEST = C'R'
          SALIR
L00047    EQU  *
          TEST = C'S'
          SALIR
L00048    EQU  *
          TEST = C'T'
          SALIR
L00049    EQU  *
          TEST = C'U'
          SALIR
L00050    EQU  *
          TEST = C'V'
          SALIR
L00051    EQU  *
          TEST = C'W'
          SALIR
L00052    EQU  *
          TEST = C' '
          SALIR
L00053    EQU  *
```

```
                TEST  = C'Y'
                SALIR
L00054          EQU   *
                TEST  = C'Z'
                SALIR
L00055          EQU   *
                TEST  = C'X'
                SALIR
                RUTINA  NUMERO
                LLAMA   DIGITO
                ENTRAR  CONTNUME
                RUTINA  CONTNUME
                DECIDE  L00056, = C'0'
                DECIDE  L00056, = C'1'
                DECIDE  L00056, = C'2'
                DECIDE  L00056, = C'3'
                DECIDE  L00056, = C'4'
                DECIDE  L00056, = C'5'
                DECIDE  L00056, = C'6'
                DECIDE  L00056, = C'7'
                DECIDE  L00056, = C'8'
                DECIDE  L00056, = C'9'
                SALIR
L00056          EQU   *
                LLAMA   DIGITO
                ENTRAR  CONTNUME
                RUTINA  DIGITO
                DECIDE  L00057, = C'0'
                DECIDE  L00058, = C'1'
                DECIDE  L00059, = C'2'
                DECIDE  L00060, = C'3'
                DECIDE  L00061, = C'4'
                DECIDE  L00062, = C'5'
                DECIDE  L00063, = C'6'
                DECIDE  L00064, = C'7'
                DECIDE  L00065, = C'8'
                DECIDE  L00066, = C'9'
L00057          EQU   *
                TEST  = C'0'
                SALIR
L00058          EQU   *
                TEST  = C'1'
                SALIR
L00059          EQU   *
```

```
                TEST  = C'2'
                SALIR
L00060          EQU   *
                TEST  = C'3'
                SALIR
L00061          EQU   *
                TEST  = C'4'
                SALIR
L00062          EQU   *
                TEST  = C'5'
                SALIR
L00063          EQU   *
                TEST  = C'6'
                SALIR
L00064          EQU   *
                TEST  = C'7'
                SALIR
L00065          EQU   *
                TEST  = C'8'
                SALIR
L00066          EQU   *
                TEST  = C'9'
                SALIR
```

Construcción manual

```
0049                                        PRINT   ON,NOGEN
0050          1000                  PRLLOAAA START X'1000'
0051                                        USING *, 0, 1, 2, 3, 4, 5, 6, 7
0052          1000    47000000      HOLA    NOP
0053          1004    45E01F9C              OPEN    FICH
0054          1008    45E02002              OPEN    IMPRE
0055          100C    47000000      CICLO     NOP
0056  X                                     LEER
0057                                        LLAMA INSTRUC
0058          105E    47F0100C              B CICLO
0059                                        RUTINA INSTRUC
0060                                        DECIDE E1, = C'L'
0061                                        DECIDE E1, = C'E'
0062                                        DECIDE E2, = C'D'
0063                                        DECIDE E3, = C'T'
0064                                        DECIDE E4, = C'S'
0065                                        DECIDE E5, = C'R'
0066                                        DECIDE E5, = C'M'
0067                                        DECIDE E6, = C'P'
0068                                        DECIDE E6, = C'H'
0069                                        DECIDE E6, = C'F'
0070                                        TEST  = C'T'
0071  X                                     SALIR
0072          10C4                  E1      EQU   *
0073                                        ENTRAR ESINST
0074          10C8                  E2      EQU   *
0075                                        ENTRAR  MDAR
0076          10CC                  E3      EQU   *
0077                                        ENTRAR MDINST
0078          10D0                  E4      EQU   *
0079                                        ENTRAR ARCT
0080          10D4                  E5      EQU   *
0081                                        ENTRAR ARINST
0082          10D8                  E6      EQU   *
0083                                        ENTRAR CTINST
0084  X                                     VOLVER
0085                                        RUTINA  ESINST
0086                                        DECIDE  E10, = C'E'
0087                                        TEST  = C'L'
0088                                        TEST  = C'E'
0089                                        TEST  = C'E'
0090                                        TEST  = C'R'
0091                                        ENTRAR CK
```

```
0092                    1124            E10     EQU  *
0093                                            TEST = C'E'
0094                                            TEST = C'S'
0095                                            TEST = C'C'
0096                                            TEST = C'R'
0097                                            TEST = C'I'
0098                                            TEST = C'B'
0099                                            TEST = C'I'
0100                                            TEST = C'R'
0101                                            ENTRAR CK
0102  X                                         VOLVER
0103                                            RUTINA CK
0104                                            LLAMA  CADENA
0105                                            ENTRAR CT
0106  X                                         VOLVER
0107                                            RUTINA MDAR
0108                                            TEST = C'D'
0109                                            ENTRAR MDARCON
0110  X                                         VOLVER
0111                                            RUTINA MDARCON
0112                                            DECIDE E11, = C'I'
0113                                            TEST = C'E'
0114                                            TEST = C'P'
0115                                            TEST = C'O'
0116                                            TEST = C'S'
0117                                            TEST = C'I'
0118                                            TEST = C'T'
0119                                            TEST = C'A'
0120                                            TEST = C'R'
0121                                            ENTRAR CK
0122                    1246            E11     EQU  *
0123                                            TEST = C'I'
0124                                            TEST = C'V'
0125                                            TEST = C'I'
0126                                            TEST = C'D'
0127                                            TEST = C'I'
0128                                            TEST = C'R'
0129                                            ENTRAR CK
0130  X                                         VOLVER
0131                                            RUTINA MDINST
0132                                            TEST = C'T'
0133                                            TEST = C'R'
0134                                            TEST = C'A'
0135                                            TEST = C'S'
```

```
0136                                    TEST  = C'L'
0137                                    TEST  = C'A'
0138                                    TEST  = C'D'
0139                                    TEST  = C'A'
0140                                    TEST  = C'R'
0141                                    LLAMA  CADENA
0142                                    ENTRAR CONTINS
0143  X                                 VOLVER
0144                                    RUTINA  ARCT
0145                                    TEST  = C'S'
0146                                    ENTRAR CU
0147  X                                 VOLVER
0148                                    RUTINA  CU
0149                                    DECIDE   E12, = C'I'
0150                                    TEST  = C'U'
0151                                    TEST  = C'M'
0152                                    TEST  = C'A'
0153                                    TEST  = C'R'
0154                                    LLAMA  CADENA
0155                                    ENTRAR CONTINS
0156                  13A2        E12   EQU  *
0157                                    TEST  = C'I'
0158                                    TEST  = C'E'
0159                                    TEST  = C'L'
0160                                    LLAMA  CK
0161                                    LLAMA  OPRELA
0162                                    ENTRAR CONTCT
0163  X                                 VOLVER
0164                                    RUTINA  CONTINS
0165                                    DECIDE  E13, = C'='
0166                                    ENTRAR CT
0167                  1400        E13   EQU  *
0168                                    ENTRAR CN
0169  X                                 VOLVER
0170                                    RUTINA  CT
0171                                    DECIDE E14, = C'('
0172  X                                 SALIR
0173                  1414        E14   EQU  *
0174                                    TEST  = C'('
0175                                    LLAMA  DIRECC
0176                                    TEST  = C')'
0177  X                                 SALIR
0178  X                                 VOLVER
0179                                    RUTINA  CN
```

```
0180                                        TEST  = C'='
0181                                        ENTRAR NUMERO
0182  X                                     VOLVER
0183                                        RUTINA CS
0184                                        TEST  = C':'
0185                                        ENTRAR DIRECC
0186  X                                     VOLVER
0187                                        RUTINA ARINST
0188                                        DECIDE E15, = C'M'
0189                                        TEST  = C'R'
0190                                        TEST  = C'E'
0191                                        TEST  = C'S'
0192                                        TEST  = C'T'
0193                                        TEST  = C'A'
0194                                        TEST  = C'R'
0195                                        LLAMA  CADENA
0196                                        ENTRAR CONTINS
0197                14E4          E15       EQU  *
0198                                        TEST  = C'M'
0199                                        TEST  = C'U'
0200                                        TEST  = C'L'
0201                                        TEST  = C'T'
0202                                        TEST  = C'I'
0203                                        TEST  = C'P'
0204                                        TEST  = C'L'
0205                                        TEST  = C'I'
0206                                        TEST  = C'C'
0207                                        TEST  = C'A'
0208                                        TEST  = C'R'
0209                                        LLAMA  CADENA
0210                                        ENTRAR CONTINS
0211  X                                     VOLVER
0212                                        RUTINA CTINST
0213                                        DECIDE E16, = C'H'
0214                                        DECIDE E17, = C'F'
0215                                        TEST  = C'P'
0216                                        ENTRAR CR
0217                1588          E16       EQU  *
0218                                        TEST  = C'F'
0219                                        TEST  = C'I'
0220                                        TEST  = C'N'
0221  X                                     SALIR
0222                15E6          E17       EQU  *
0223                                        TEST  = C'H'
```

```
0224                                    TEST  =C'A'
0225                                    TEST  = C'C'
0226                                    TEST  = C'E'
0227                                    TEST  = C'R'
0228                                    LLAMA  NUMERO
0229                                    LLAMA  CADENA
0230                                    LLAMA  NUMERO
0231                                    ENTRAR CADENA
0232 X                                  VOLVER
0233                                    RUTINA CR
0234                                    DECIDE E20, = C'O'
0235                                    TEST  = C'A'
0236                                    TEST  = C'R'
0237                                    TEST  = C'A'
0238                                    TEST  = C'R'
0239 X                                  SALIR
0240                    16A8        E20 EQU  *
0241                                    TEST  = C'O'
0242                                    TEST  = C'N'
0243                                    TEST  = C'E'
0244                                    TEST  = C'R'
0245                                    LLAMA  CADENA
0246                                    ENTRAR CS
0247 X                                  VOLVER
0248                                    RUTINA  CONTCT
0249                                    DECIDE  E21, = C'('
0250                                    TEST  = C'0'
0251                                    LLAMA  CADENA
0252                                    ENTRAR CS
0253                    1722        E21 EQU  *
0254                                    TEST  = C'('
0255                                    TEST  = C'A'
0256                                    TEST  = C'C'
0257                                    TEST  = C')'
0258                                    LLAMA  CADENA
0259                                    ENTRAR CS
0260 X                                  VOLVER
0261                                    RUTINA  OPRELA
0262                                    DECIDE  E22, = C'>'
0263                                    DECIDE  E23, =, C'='
0264                                    TEST  = C'<'
0265 X                                  SALIR
0266                    1794        E22 EQU  *
0267                                    TEST  = C'>'
```

```
0268 X                                      SALIR
0269                 17A6          E23      EQU  *
0270                                        TEST  = C'='
0271 X                                      SALIR
0272 X                                      VOLVER
0273                                        RUTINA  DIRECC
0274                                        LLAMA   DIGITO
0275                                        ENTRAR  DIGITO
0276 X                                      VOLVER
0277                                        RUTINA  CADENA
0278                                        LLAMA   CARACT
0279                                        ENTRAR  CONTCAD
0280 X                                      VOLVER
0281                                        RUTINA  CONTCAD
0282                                        DECIDE  E30, = C'A'
0283                                        DECIDE  E30, = C'B'
0284                                        DECIDE  E30, = C'C'
0285                                        DECIDE  E30, = C'D'
0286                                        DECIDE  E30, = C'E'
0287                                        DECIDE  E30, = C'F'
0288                                        DECIDE  E30, = C'G'
0289                                        DECIDE  E30, = C'H'
0290                                        DECIDE  E30, = C'I'
0291                                        DECIDE  E30, = C'J'
0292                                        DECIDE  E30, = C'K'
0293                                        DECIDE  E30, = C'L'
0294                                        DECIDE  E30, = C'M'
0295                                        DECIDE  E30, = C'N'
0296                                        DECIDE  E30, = C'O'
0297                                        DECIDE  E30, = C'P'
0298                                        DECIDE  E30, = C'Q'
0299                                        DECIDE  E30, = C'R'
0300                                        DECIDE  E30, = C'S'
0301                                        DECIDE  E30, = C'T'
0302                                        DECIDE  E30, = C'U'
0303                                        DECIDE  E30, = C'V'
0304                                        DECIDE  E30, = C'W'
0305                                        DECIDE  E30, = C'X'
0306                                        DECIDE  E30, = C'Y'
0307                                        DECIDE  E30, = C'Z'
0308                                        DECIDE  E30, = C' '
0309 X                                      SALIR
0310                 18C8          E30      EQU  *
0311                                        LLAMA   CARACT
```

```
0312                                        ENTRAR  CONTCAD
0313  X                                     VOLVER
0314                                        RUTINA  CARACT
0315                                        DECIDE  E50, = C'A'
0316                                        DECIDE  E51, = C'B'
0317                                        DECIDE  E52, = C'C'
0318                                        DECIDE  E53, = C'D'
0319                                        DECIDE  E54, = C'E'
0320                                        DECIDE  E55, = C'F'
0321                                        DECIDE  E56, = C'G'
0322                                        DECIDE  E57, = C'H'
0323                                        DECIDE  E58, = C'I'
0324                                        DECIDE  E59, = C'J'
0325                                        DECIDE  E60, = C'K'
0326                                        DECIDE  E61, = C'L'
0327                                        DECIDE  E62, = C'M'
0328                                        DECIDE  E63, = C'N'
0329                                        DECIDE  E64, = C'O'
0330                                        DECIDE  E65, = C'P'
0331                                        DECIDE  E66, = C'Q'
0332                                        DECIDE  E67, = C'R'
0333                                        DECIDE  E68, = C'S'
0334                                        DECIDE  E69, = C'T'
0335                                        DECIDE  E70, = C'U'
0336                                        DECIDE  E71, = C'V'
0337                                        DECIDE  E72, = C'W'
0338                                        DECIDE  E73, = C'X'
0339                                        DECIDE  E74, = C'Y'
0340                                        DECIDE  E75, = C'Z'
0341                                        TEST  = C' '
0342  X                                     SALIR
0343                19C2          E50       EQU   *
0344                                        TEST  = C'A'
0345  X                                     SALIR
0346                19D4          E51       EQU   *
0347                                        TEST  = C'B'
0348  X                                     SALIR
0349                19E6          E52       EQU   *
0350                                        TEST  = C'C'
0351  X                                     SALIR
0352                19F8          E53       EQU   *
0353                                        TEST  = C'D'
0354  X                                     SALIR
0355                1A0A          E54       EQU   *
```

```
0356                                        TEST  = C'E'
0357  X                                     SALIR
0358                 1A1C          E55      EQU   *
0359                                        TEST  = C'F'
0360  X                                     SALIR
0361                 1A2E          E56      EQU   *
0362                                        TEST  = C'G'
0363  X                                     SALIR
0364                 1A40          E57      EQU   *
0365                                        TEST  = C'H'
0366  X                                     SALIR
0367                 1A52          E58      EQU   *
0368                                        TEST  = C'I'
0369  X                                     SALIR
0370                 1A64          E59      EQU   *
0371                                        TEST  = C'J'
0372  X                                     SALIR
0373                 1A76          E60      EQU   *
0374                                        TEST  = C'K'
0375  X                                     SALIR
0376                 1A88          E61      EQU   *
0377                                        TEST  = C'L'
0378  X                                     SALIR
0379                 1A9A          E62      EQU   *
0380                                        TEST  = C'M'
0381  X                                     SALIR
0382                 1AAC          E63      EQU   *
0383                                        TEST  = C'N'
0384  X                                     SALIR
0385                 1ABE          E64      EQU   *
0386                                        TEST  = C'O'
0387  X                                     SALIR
0388                 1AD0          E65      EQU   *
0389                                        TEST  = C'P'
0390  X                                     SALIR
0391                 1AE2          E66      EQU   *
0392                                        TEST  = C'Q'
0393  X                                     SALIR
0394                 1AF4          E67      EQU   *
0395                                        TEST  = C'R'
0396  X                                     SALIR
0397                 1B06          E68      EQU   *
0398                                        TEST  = C'S'
0399  X                                     SALIR
```

```
0400            1B18        E69     EQU  *
0401                                TEST = C'T'
0402 X                              SALIR
0403            1B2A        E70     EQU  *
0404                                TEST = C'U'
0405 X                              SALIR
0406            1B3C        E71     EQU  *
0407                                TEST = C'V'
0408 X                              SALIR
0409            1B4E        E72     EQU  *
0410                                TEST = C'W'
0411 X                              SALIR
0412            1B60        E73     EQU  *
0413                                TEST = C'X'
0414 X                              SALIR
0415            1B72        E74     EQU  *
0416                                TEST = C'Y'
0417 X                              SALIR
0418            1B84        E75     EQU  *
0419                                TEST = C'Z'
0420 X                              SALIR
0421 X                              VOLVER
0422                                RUTINA  NUMERO
0423                                LLAMA   DIGITO
0424                                ENTRAR  CONTNUME
0425 X                              VOLVER
0426                                RUTINA CONTNUME
0427                                DECIDE  E40, = C'1'
0428                                DECIDE  E40, = C'2'
0429                                DECIDE  E40, = C'3'
0430                                DECIDE  E40, = C'4'
0431                                DECIDE  E40, = C'5'
0432                                DECIDE  E40, = C'6'
0433                                DECIDE  E40, = C'7'
0434                                DECIDE  E40, = C'8'
0435                                DECIDE  E40, = C'9'
0436                                DECIDE  E40, = C'0'
0437 X                              SALIR
0438            1C06        E40     EQU  *
0439                                LLAMA   DIGITO
0440                                ENTRAR  CONTNUME
0441 X                              VOLVER
0442                                RUTINA  DIGITO
0443                                DECIDE  E80, =C'0'
```

```
0444                                          DECIDE  E81, = C'1'
0445                                          DECIDE  E82, = C'2'
0446                                          DECIDE  E83, = C'3'
0447                                          DECIDE  E84, = C'4'
0448                                          DECIDE  E85, = C'5'
0449                                          DECIDE  E86, = C'6'
0450                                          DECIDE  E87, = C'7'
0451                                          DECIDE  E88, = C'8'
0452                                          TEST  = C'9'
0453  X                                       SALIR
0454                1C78                E80   EQU  *
0455                                          TEST  = C'0'
0456  X                                       SALIR
0457                1C8A                E81   EQU  *
0458                                          TEST  = C'1'
0459  X                                       SALIR
0460                1C9C                E82   EQU  *
0461                                          TEST  = C'2'
0462  X                                       SALIR
0463                1CAE                E83   EQU  *
0464                                          TEST  = C'3
0465  X                                       SALIR
0466                1CC0                E84   EQU  *
0467                                          TEST  = C'4'
0468  X                                       SALIR
0469                1CD2                E85   EQU  *
0470                                          TEST  = C'5'
0471  X                                       SALIR
0472                1CE4                E86   EQU  *
0473                                          TEST  = C'6'
0474  X                                       SALIR
0475                1CF6                E87   EQU  *
0476                                          TEST  = C'7'
0477  X                                       SALIR
0478                1D08                E88   EQU  *
0479                                          TEST  = C'8'
0480  X                                       SALIR
0481  X                                       VOLVER
0482                1D1E  45E020D6      FIN   CLOSE     IMPRE
0483                1D22  45E01FA0            CLOSE     FICH
0484                1D26  A9003333           HPR       X'3333'
0485                1D2A  D2031EB21EAC  SALIRSUB MVC    POINT2(4), POINT
0486                1D30  48801EAE          LH        8,POINT3
0487                1D34  AB801EB6        SH    8.CUA
```

```
0488  1D38  40801EAE
0489  1D3C  D5031DDC1EAC
0490  1D42  47201DB0
0491  1D46  47F01EB2
0492  1D4A  95001EB9      ETTEST1  CLI   CARCO,*—*
0493  1D4E  47801D56             BE    * + 8
0494  1D52  45901D8E               BAL   9,FALSO
0495  1D56  A6011EBA               AI    BUFIND, X'01'
0496  1D5A  D2001EB81EB9           MVC   CARCO—1(1),CARCO
0497  1D60  48801EBA               LH    8,BUFIND
0498  1D64  D2001EB98000          MVC   CARCO,0(8)
0499  1D6A  47F0D004               B     4(,13)
0500  1D6E  A6041EAE     ETLLAMA   AI    POINT + 2, X'04'
0501  1D72  D5031EAC1EA8           CLC   POINT(4),FINPILA
0502  1D78  47801DC6               BE    OVERF
0503  1D7C  D2011EB01EAE           MVC   POINT1(2),POINT + 2                    SAVEAREA
0504  1D82  A6021EB0               AI    POINT1, X'02'
0505  1D86  48801EB0               LH    8,POINT1
0506  1D8A  47F0D000               B     0(,13)
0507  1D8E  D2831F171F16  FALSO    MVC   ARIM,BLAN
0508  1D94  D2071F2B231B           MVC   ARIM + 20(8), = C'CARACTER'
0509  1D9A  D2001F351EB9           MVC   ARIM + 30(1),CARCO
0510  1DA0  D2141F372323           MVC   ARIM + 32(21), = C'ILEGAL EN EL CONTEXTO'
0511  1DA6  45E020E21F17           PUT   IMPRE,ARIM
0512  1DAC  47E09000               B     0(,9)
0513  1DB0  D2831F171F16  UNDFL    MVC   ARIM,BLAN
0514  1DB6  D2231F2B2338           MVC   ARIM + 20(36), = C'UNDERFLOW EN LA PILA DE RECURSIVIDAD'
0515  1DBC  45E020E21F17           PUT   IMPRE,ARIM
0516  1DC2  A9001111               HPR   X'1111'
0517  1DC6  D2831F171F16  OVERF    MVC   ARIM,BLAN
0518  1DCC  D2201F2B235C           MVC   ARIM + 20(33), = C'OVERFLOW EN LA PILA DE RECURSIVIDAD'
0519  1DD2  45E020E21F17           PUT   IMPRE,ARIM
0520  1DD8  A9002222               HPR   X'2222'
0521  1DDC  47F0          PRINC    DC    X'47F0'
0522  1DDE  1DDC                   DC    Y(PRINC)
0523  1DE0  47F0000047F00000 PILA  DC    50X'47F00000'
      1DE8  47F0000047F00000
      1DF0  47F0000047F00000
      1DF8  47F0000047F00000
      1E00  47F0000047F00000
      1E08  47F0000047F00000
      1E10  47F0000047F00000
      1E18  47F0000047F00000
      1E20  47F0000047F00000
```

```
                1E28  47F0000047F00000
                1E30  47F0000047F00000
                1E38  47F0000047F00000
                1E40  47F0000047F00000
                1E48  47F0000047F00000
                1E50  47F0000047F00000
                1E58  47F0000047F00000
                1E60  47F0000047F00000
                1E68  47F0000047F00000
                1E70  47F0000047F00000
                1E78  47F0000047F00000
                1E80  47F0000047F00000
                1E88  47F0000047F00000
                1E90  47F0000047F00000
                1E98  47F0000047F00000
                1EA0  47F0000047F00000
0524            1EA8  47F0      FINPILA   DC      X'47F0'
0525            1EAA  1EA8                DC      Y(FINPILA)
0526            1EAC  47F0      POINT     DC      X'47F0'
0527            1EAE  1DDC      POINT3    DC      Y(PRINC)
0528            1EB0  0000      POINT1    DC      Y(0)
0529            1EB2  0000      POINT2    DC      Y(0)
0530            1EB4  0000                DC      Y(0)
0531            1EB6  0004      CUA       DC      Y(4)
0532            1EB8  40                  DC      C' '
0533            1EB9  40        CARCO     DC      C' '
0534            1EBA  1EBC      BUFIND    DC      Y(RDBUF)
0535            1EBC  4040404040404040  RDBUF     DC      80C' '
                1EC4  4040404040404040
                1ECC  4040404040404040
                1ED4  4040404040404040
                1EDC  4040404040404040
                1EE4  4040404040404040
                1EEC  4040404040404040
                1EF4  4040404040404040
                1EFC  4040404040404040
                1F04  4040404040404040
0536            1F0C  1EBC      TEMP      DC      Y(RDBUF)
0537            1F0E  000F      CONT      DC      X'000F'
0538            1F10  F0202120  MASK      DC      X'F0202120'
0539            1F14  0002      DOS       DC      Y(2)
0540            1F16  40        BLAN      DC      C' '
0541            1F17            ARIM      DS      CL132
0542                            FICH      DTFCS   EOFA = FIN
0543                            IMPRE     DTFPR   DEVA = 10, PROV = YES
```

```
0544            2316  0000                BASEG   DC      Y(00)
0545            2318  0000                BASE1   DC      Y(00)
                231A
                231A
                231A  1FC3C1D9C1C3E3C5
                2322  D9C9D3C5C7C1D340
                232A  C5D540C5D340C3D6
                2332  D5E3C5E7E3D6E405
                233A  C4C5D9C6D3D6E640
                2342  C5D540D3C140D7C9
                234A  D3C140C4C540D9C5
                2352  C3E4D9E2C9E5C9C4
                235A  C1C4D6E5C5D9C6D3
                2362  D6E640C5D540D3C1
                236A  40D7C9D3C140C4C5
                2372  40D9C5C3E4D9E2C9
                237A  E5C9C4C1C4
0546            237F  000000001000        END     HOLA
```

Aplicación a algunas sentencias del lenguaje fuente

```
0001        LEER EL CONTENIDO DE LA UNIDAD DE ENTRADA EN LA POSICION (12)
0002        DEER EL CONTENIDO DE LA UNIDAD DE ENTRADA EN LA POSICION (12)
CARACTER    E ILEGAL EN EL CONTEXTO
CARACTER    R ILEGAL EN EL CONTEXTO
CARACTER      ILEGAL EN EL CONTEXTO
CARACTER    E ILEGAL EN EL CONTEXTO
CARACTER    L ILEGAL EN EL CONTEXTO
CARACTER      ILEGAL EN EL CONTEXTO
CARACTER    C ILEGAL EN EL CONTEXTO
0003        ESCRIBIR EN LA UNIDAD DE SALIDA EL CONTENIDO DE LA POSICION (33)
0004        DEPOSITAR EL CONTENIDO DEL ACUMULADOR EN LA POSICION (45)
0005        TRASLADAR AL ACUMULADOR EL CONTENIDO DE LA POSICION (77)
0006        TRASLADAR AL ACUMULADOR EL CONTENIDO DE LA POSICION (A7)
CARACTER    A ILEGAL EN EL CONTEXTO
0007        TRASLADAR AL ACUMULADOR EL NUMERO = 123456
0008        SUMAR AL ACUMULADOR EL CONTENIDO DE LA POSICION (11)
0009        SUMAR AL ACUMULADOR EL NUMERO = 456789
0010        RESTAR DEL ACUMULADOR EL CONTENIDO DE LA POSICION (44)
0011        RESTAR DEL ACUMULADOR EL NUMERO = 12678
0012        MULTIPLICAR AL ACUMULADOR POR EL CONTENIDO DE LA POSICION (55)
0013        MULTIPLICAR AL ACUMULADOR POR EL NUMERO = 576894
0014        DIVIDIR EL ACUMULADOR POR EL CONTENIDO DE LA POSICION (99)
0015        DIVIDIR EL ACUMULADOR POR EL CONTENIDO DE LA POSICION JJ9)
CARACTER    I ILEGAL EN EL CONTEXTO
CARACTER    D ILEGAL EN EL CONTEXTO
CARACTER      ILEGAL EN EL CONTEXTO
0016        PONER EN EL CD LA DIRECCION :77
0017        PONER: 66
0018        SIEL CONTENIDO DE (25)<(AC) PONER EN EL CD LA DIRECCION :77
0019        SIEL CONTENIDO DE (25)=(AC) PONER EN EL CD LA DIRECCION :77
0020        SIEL CONTENIDO DE (25)>(AC) PONER EN EL CD LA DIRECCION :77
0021        SIEL (27)>(AC) :56
0022        SIEL CONTENIDO DEL AC ES >0 PONER EN EL CD LA DIRECCION :86
0023        HACER 22 VECES LAS 01 INSTRUCCIONES SIGUIENTES
CARACTER      ILEGAL EN EL CONTEXTO
0024        PARAR
0025        FIN
```

APENDICE G

El traductor
Ejemplos del lenguaje implementado

Traductor del lenguaje 9013

```
0001                    PROC    P,0
0002    SALIR           NAME
0003                    B       SALIRSUB
0004                    END
0005                    PROC    P,2
0006    DECIDE          NAME
0007                    CLI     CARCO,P(2)
0008                    BE      P(1)
0009                    END
0010                    PROC    P,1
0011    TEST            NAME
0012                    MVC     ETTEST1,*+10
0013                    BAL     13,ETTEST1
0014                    CLI     CARCO,P(1)
0015                    END
0016                    PROC    P,1
0017    LLAMA           NAME
0018                    BAL     13,ETLLAMA
0019                    MVC     0(2,8),*+10
0020                    B       P(1)
0021                    DC      Y(*+2)
0022                    END
0023                    PROC    P,0
0024    LEER            NAME
0025                    MVC     ARIM,BLAN
0026                    CP      CON,=X'0F'
0027                    BC      8,GET
0028                    MVC     M,=XL4'0'
0029                    STH     12,SALR12
0030                    STH     8,SALR15
0031                    LH      8,DPROGFU
0032                    SH      8,DOS
0033                    MVC     RESER,0(8)
0034                    MVO     M(2),RESER(1)
0035                    MVO     M+2(2),RESER+1(1)
0036                    NC      RESER,=X'0F0F'
0037                    MVC     M+1(1),RESER
0038                    MVC     M+3(1),RESER+1
0039                    TR      M,T77
0040                    MVC     ARIM+4(4),M
0041                    LH      8,0(,8)
0042                    LH      15,=Y(ARIM+10)
0043    MVC             MVC     RESER,=X'0000'
0044                    MVC     M(1),0(8)
```

```
0045                    MVO     RESER,M(1)
0046                    NI          M,X'0F'
0047                    MVC     RESER+1(1),M
0048                    MVC     0(2,15),RESER
0049                    AH      8, = Y(1)
0050                    AH      15,DOS
0051                    CH      8,SALR12
0052                    BC      4,MVO
0053                    TR      ARIM+10(121),T77
0054                    PUT     IMPRE,ARIM
0055                    LH      8,SALR15
0056    GET             GET     FICH,RDBUF
0057                    MVC     CARCO(1),RDBUF
0058                    MVC     BUFIND,TEMP
0059                    AP        CON, = X'1F'
0060                    LH      13,DPROGFU
0061                    MVC     0(2,13),CON
0062                    AH      13, = H'2'
0063                    STH     12,0(,13)
0064                    AI      DPROGFU,X'04'
0065                    MVC     ARIM,BLAN
0066                    MVC     ARIM+45(4),MASK
0067                    ED      ARIM+45(4),CON
0068                    MVC     ARIM+52(80),RDBUF
0069                    PUT     IMPRE,ARIM
0070                    LH      10,BASEG
0071                    LH      11,BASE1
0072                    B       *+130
0073                    DC      X'999F0000'
0074    PROGFUDI        DC      120X'00'
0075    DPROGFU         DC      Y(PROGFUDI)
0076                    END
0077                    PROC    P,1
0078    RUTINA          NAME
0079    P(1)            EQU     *
0080                    END
0081                    PROC    P,1
0082    ENTRAR          NAME
0083                    B       P(1)
0084                    END
0085                    PROC    P,0
0086    VOLVER          NAME
0087                    NOP
0088                    END
```

```
0089                                                    PRINT   ON,NOGEN
0090          1260                      PRLLOAAA        START   X'1260'
0091                                                    USING   *,0,1,2,3,4,5,6,7
0092          1260    47000000          HOLA            NOP
0093          1264    A9004444                  HPR     X'4444'
0094          1268    45E02470                          OPEN    FICH
0095          126C    45E025A6                          OPEN    IMPRE
0096          1270    48C02C68                          LH      12,DTRABG2          'DESDE EL ANALIZADOR'
0097          1274    47000000          CICLO             NOP
0098  X                                                 LEER
0099                                                    LLAMA   INSTRUC
0100          13FE    D202A0002CA6                      MVC     0(3,10), = X'6D6D6D'
0101                                                        LLAMA   GENERAD
0102          1414    47F01274                  B       CICLO
0103                                                    RUTINA  INSTRUC
0104                                                    DECIDE  E1, = C'L'
0105                                                    DECIDE  E1, = C'E'
0106                                                    DECIDE  E2, = C'D'
0107                                                    DECIDE  E3, = C'T'
0108                                                    DECIDE  E4, = C'S'
0109                                                    DECIDE  E5, = C'R'
0110                                                    DECIDE  E5, = C'M'
0111                                                    DECIDE  E6, = C'P'
0112                                                    DECIDE  E6, = C'H'
0113                                                    DECIDE  E6, = C'F'
0114                                                    TEST  = C'T'
0115  X                                                 SALIR
0116          147A                      E1              EQU   *
0117                                                    ENTRAR  ESINST
0118          147E                      E2              EQU   *
0119                                                    ENTRAR  MDAR
0120          1482                      E3              EQU   *
0121                                                    ENTRAR  MDINST
0122          1486                      E4              EQU   *
0123                                                    ENTRAR  ARCT
0124          148A                      E5              EQU   *
0125                                                    ENTRAR  ARINST
0126          148E                      E6              EQU   *
0127                                                    ENTRAR  CTINST
0128  X                                                 VOLVER
0129                                                    RUTINA  ESINST
0130                                                    DECIDE  E10, = C'E'
0131                                                    TEST  = C'L'
```

```
0132                                                    TEST  = C'E'
0133                                                    TEST  = C'E'
0134                                                    TEST  = C'R'
0135          14D6    D201A0002C70          MVC         0(2,10), = Y(GFGO)
0136          14DC    AAA02C6E                AH        10, = H'2'
0137                                                    ENTRAR CK
0138          14E4                  E10               EQU  *
0139                                                    TEST  = C'E'
0140                                                    TEST  = C'S'
0141                                                    TEST  = C'C'
0142                                                    TEST  = C'R'
0143                                                    TEST  = C'I'
0144                                                    TEST  = C'B'
0145                                                    TEST  = C'I'
0146                                                    TEST  = C'R'
0147          1554    D201A0002C72          MVC         0(2,10), = Y(GFGO+6)
0148          155A    AAA02C6E                AH        10, = H'2'
0149                                                    ENTRAR CK
0150  X                                                 VOLVER
0151                                                    RUTINA CK
0152                                                    LLAMA  CADENA
0153                                                    ENTRAR CT
0154  X                                                 VOLVER
0155                                                    RUTINA MDAR
0156                                                    TEST  = C'D'
0157                                                    ENTRAR MDARCON
0158  X                                                 VOLVER
0159                                                    RUTINA MDARCON
0160                                                    DECIDE E11, = C'I'
0161                                                    TEST  = C'E'
0162                                                    TEST  = C'P'
0163                                                    TEST  = C'O'
0164                                                    TEST  = C'S'
0165                                                    TEST  = C'I'
0166                                                    TEST  = C'T'
0167                                                    TEST  = C'A'
0168                                                    TEST  = C'R'
0169          160C    D201A0002C74          MVC         0(2,10), = Y(GFGO+12)
0170          1612    AAA02C6E                AH        10, = H'2'
0171                                                    ENTRAR CK
0172          161A                  E11               EQU  *
0173                                                    TEST  = C'I'
0174                                                    TEST  = C'V'
0175                                                    TEST  = C'I'
```

```
0176                                                    TEST  = C'D'
0177                                                    TEST  = C'I'
0178                                                    TEST  = C'R'
0179          166E      D201A0002C76            MVC   0(2,10), = Y(GFGO+18)
0180          1674      AAA02C6E                  AH  10, = H'2'
0181                                                    ENTRAR CK
0182  X                                                 VOLVER
0183                                                    RUTINA  MDINST
0184                                                    TEST  = C'T'
0185                                                    TEST  = C'R'
0186                                                    TEST  = C'A'
0187                                                    TEST  = C'S'
0188                                                    TEST  = C'L'
0189                                                    TEST  = C'A'
0190                                                    TEST  = C'D'
0191                                                    TEST  = C'A'
0192                                                    TEST  = C'R'
0193          16FE      D201A0002C78            MVC   0(2,10), = Y(GFGO+24)
0194          1704      AAA02C6E                  AH  10, = H'2'
0195                                                    LLAMA  CADENA
0196                                                    ENTRAR CONTINS
0197  X                                                 VOLVER
0198                                                    RUTINA  ARCT
0199                                                    TEST  = C'S'
0200                                                    ENTRAR CU
0201  X                                                 VOLVER
0202                                                    RUTINA  CU
0203                                                    DECIDE  E12, = C'I'
0204                                                    TEST  = C'U'
0205                                                    TEST  = CM'
0206                                                    TEST  = C'A'
0207                                                    TEST  = C'R'
0208          1776      D201A0002C7A            MVC   0(2,10), = Y(GFGO+30)
0209          177C      AAA02C6E                  AH  10, = H'2'
0210                                                    LLAMA  CADENA
0211                                                    ENTRAR CONTINS
0212          1794                      E12           EQU  *
0213                                                    TEST  = C'I'
0214                                                    TEST  = C'E'
0215                                                    TEST  = C'L'
0216          17BE      D201A0002C7C            MVC   0(2,10), = Y(GFGO+36)
0217          17C4      AAA02C6E                  AH  10, = H'2'
0218                                                    LLAMA  CK
0219                                                    LLAMA  OPRELA
```

```
0220                                                    ENTRAR CONTCT
0221  X                                                 VOLVER
0222                                                    RUTINA CONTINS
0223                                                    DECIDE E13, = C'='
0224                                                    ENTRAR CT
0225          17FC                      E13             EQU  *
0226                                                    ENTRAR CN
0227  X                                                 VOLVER
0228                                                    RUTINA CT
0229                                                    DECIDE E14, = C'('
0230  X                                                 SALIR
0231          1810                      E14             EQU  *
0232                                                    TEST  = C'('
0233                                                    LLAMA  DIRECC
0234                                                    TEST  = C')'
0235          183C  D201A0002C7E                  MVC   0(2,10), = Y(GFGO+54)
0236          1842  AAA02C6E                      AH    10, = H'2'
0237  X                                                 SALIR
0238  X                                                 VOLVER
0239                                                    RUTINA CN
0240                                                    TEST  = C'='
0241          185C  D201A0002C80                  MVC   0(2,10), = Y(GFGO+48)
0242          1862  AAA02C6E                      AH    10, = H'2'
0243                                                    ENTRAR NUMERO
0244  X                                                 VOLVER
0245                                                    RUTINA CS
0246                                                    TEST  = C':'
0247          187C  D201A0002CA9                  MVC   0(2,10), = X'D000'
0248          1882  AAA02C6E                      AH    10, = H'2'
0249                                                    ENTRAR DIRECC
0250  X                                                 VOLVER
0251                                                    RUTINA ARINST
0252                                                    DECIDE E15, = C'M'
0253                                                    TEST  = C'R'
0254                                                    TEST  = C'E'
0255                                                    TEST  = C'S'
0256                                                    TEST  = C'T'
0257                                                    TEST  = C'A'
0258                                                    TEST  = C'R'
0259          18EA  D201A0002C82                  MVC   0(2,10), = Y(GFGO+60)
0260          18F0  AAA02C6E                      AH    10, = H'2'
0261                                                    LLAMA  CADENA
0262                                                    ENTRAR CONTINS
0263          1908                      E15             EQU  *
```

```
0264                                                    TEST  = C'M'
0265                                                    TEST  = C'U'
0266                                                    TEST  = C'L'
0267                                                    TEST  = C'T'
0268                                                    TEST  = C'I'
0269                                                    TEST  = C'P'
0270                                                    TEST  = C'L'
0271                                                    TEST  = C'I'
0272                                                    TEST  = C'C'
0273                                                    TEST  = C'A'
0274                                                    TEST  = C'R'
0275          19A2      D201A0002C84            MVC   0(2,10), = Y(GFGO+66)
0276          19A8      AAA02C6E                  AH  10, = H'2'
0277                                                    LLAMA  CADENA
0278                                                    ENTRAR CONTINS
0279  X                                                 VOLVER
0280                                                    RUTINA CTINST
0281                                                    DECIDE E16, = C'H'
0282                                                    DECIDE E17, = C'F'
0283                                                    TEST  = C'P'
0284                                                    ENTRAR CR
0285          19E6                      E17           EQU  *
0286                                                    TEST  = C'F'
0287                                                    TEST  = C'I'
0288                                                    TEST  = C'N'
0289          1A10      48D02A54                      LH     13,BASETSUB
0290          1A14      47F0D014                      B      20(,13)          EJECUCION
0291  X                                                 SALIR
0292          1A1C                      E16           EQU  *
0293                                                    TEST  = C'H'
0294                                                    TEST  = C'A'
0295                                                    TEST  = C'C'
0296                                                    TEST  = C'E'
0297                                                    TEST  = C'R'
0298          1A62      D201A0002C86            MVC   0(2,10), = Y(GFGO+78)
0299          1A68      AAA02C6E                  AH  10, = H'2'
0300                                                    LLAMA  NUMERO
0301                                                    LLAMA  CADENA
0302                                                    LLAMA  NUMERO
0303                                                    ENTRAR CADENA
0304  X                                                 VOLVER
0305                                                    RUTINA CR
0306                                                    DECIDE E20, = C'O'
0307                                                    TEST  = C'A'
```

```
0308                                                    TEST  = C'R'
0309                                                    TEST  = C'A'
0310                                                    TEST  = C'R'
0311          1AE4   D201A0002C88              MVC     0(2,10), = Y(GFGO+84)
0312          1AEA   AAA02C6E                   AH     10, = H'2'
0313   X                                                SALIR
0314          1AF2                      E20            EQU   *
0315                                                    TEST  = C'O'
0316                                                    TEST  = C'N'
0317                                                    TEST  = C'E'
0318                                                    TEST  = C'R'
0319          1B2A   D201A0002C8A              MVC     0(2,10), = Y(GFGO+90)
0320          1B30   AAA02C6E                   AH     10, = H'2'
0321          1B34   AAB02C6E                   AH     11, = H'2'
0322                                                    LLAMA  CADENA
0323                                                    ENTRAR CS
0324   X                                                VOLVER
0325                                                    RUTINA CONTCT
0326                                                    DECIDE E21, = C'('
0327                                                    TEST  = C'0'
0328          1B66   D201A0002C8C              MVC     0(2,10), = Y(GFGO+96)
0329          1B6C   AAA02C6E                   AH     10, = H'2'
0330                                                    LLAMA  CADENA
0331                                                    ENTRAR CS
0332          1B84                      E21            EQU   *
0333                                                    TEST  = C'('
0334                                                    TEST  = C'A'
0335                                                    TEST  = C'C'
0336                                                    TEST  = C')'
0337                                                    LLAMA  CADENA
0338                                                    ENTRAR CS
0339   X                                                VOLVER
0340                                                    RUTINA OPRELA
0341                                                    DECIDE E22, = C'>'
0342                                                    DECIDE E23, = C'='
0343                                                    TEST  = C'<'
0344          1BF2   D201A0002C8E              MVC     0(2,10), = Y(GFGO+114)
0345          1BF8   AAA02C6E                   AH     10, = H'2'
0346   X                                                SALIR
0347          1C00                      E22            EQU   *
0348                                                    TEST  = C'>'
0349          1C0E   D201A0002C90              MVC     0(2,10), = Y(GFGO+102)
0350          1C14   AAA02C6E                   AH     10, = H'2'
0351   X                                                SALIR
```

```
0352          1C1C                          E23                 EQU  *
0353                                                              TEST  = C'='
0354          1C2A    D201A0002C92                    MVC       0(2,10), = Y(GFGO+108)
0355          1C30    AAA02C6E                          AH      10, = H'2'
0356  X                                                         SALIR
0357  X                                                         VOLVER
0358                                                            RUTINA  DIRECC
0359                                                            LLAMA   DIGITO
0360                                                            ENTRAR  DIGITO
0361  X                                                         VOLVER
0362                                                            RUTINA  CADENA
0363                                                            LLAMA   CARACT
0364                                                            ENTRAR  CONTCAD
0365  X                                                         VOLVER
0366                                                            RUTINA  CONTCAD
0367                                                            DECIDE  E30, = C'A'
0368                                                            DECIDE  E30, = C'B'
0369                                                            DECIDE  E30, = C'C'
0370                                                            DECIDE  E30, = C'D'
0371                                                            DECIDE  E30, = C'E'
0372                                                            DECIDE  E30, = C'F'
0373                                                            DECIDE  E30, = C'G'
0374                                                            DECIDE  E30, = C'H'
0375                                                            DECIDE  E30, = C'I'
0376                                                            DECIDE  E30, = C'J'
0377                                                            DECIDE  E30, = C'K'
0378                                                            DECIDE  E30, = C'L'
0379                                                            DECIDE  E30, = C'M'
0380                                                            DECIDE  E30, = C'N'
0381                                                            DECIDE  E30, = C'O'
0382                                                            DECIDE  E30, = C'P'
0383                                                            DECIDE  E30, = C'Q'
0384                                                            DECIDE  E30, = C'R'
0385                                                            DECIDE  E30, = C'S'
0386                                                            DECIDE  E30, = C'T'
0387                                                            DECIDE  E30, = C'U'
0388                                                            DECIDE  E30, = C'V'
0389                                                            DECIDE  E30, = C'W'
0390                                                            DECIDE  E30, = C'X'
0391                                                            DECIDE  E30, = C'Y'
0392                                                            DECIDE  E30, = C'Z'
0393                                                            DECIDE  E30, = C' '
0394  X                                                         SALIR
0395          1D48                          E30                 EQU  *
```

```
0396                                    LLAMA  CARACT
0397                                    ENTRAR CONTCAD
0398  X                                 VOLVER
0399                                    RUTINA CARACT
0400                                    DECIDE E50, = C'A'
0401                                    DECIDE E51, = C'B'
0402                                    DECIDE E52, = C'C'
0403                                    DECIDE E53, = C'D'
0404                                    DECIDE E54, = C'E'
0405                                    DECIDE E55, = C'F'
0406                                    DECIDE E56, = C'G'
0407                                    DECIDE E57, = C'H'
0408                                    DECIDE E58, = C'I'
0409                                    DECIDE E59, = C'J'
0410                                    DECIDE E60, = C'K'
0411                                    DECIDE E61, = C'L'
0412                                    DECIDE E62, = C'M'
0413                                    DECIDE E63, = C'N'
0414                                    DECIDE E64, = C'O'
0415                                    DECIDE E65, = C'P'
0416                                    DECIDE E66, = C'Q'
0417                                    DECIDE E67, = C'R'
0418                                    DECIDE E68, = C'S'
0419                                    DECIDE E69, = C'T'
0420                                    DECIDE E70, = C'U'
0421                                    DECIDE E71, = C'V'
0422                                    DECIDE E72, = C'W'
0423                                    DECIDE E73, = C'X'
0424                                    DECIDE E74, = C'Y'
0425                                    DECIDE E75, = C'Z'
0426                                    TEST   = C' '
0427  X                                 SALIR
0428          1E42              E50     EQU    *
0429                                    TEST   = C'A'
0430  X                                 SALIR
0431          1E54              E51     EQU    *
0432                                    TEST   = C'B'
0433  X                                 SALIR
0434          1E66              E52     EQU    *
0435                                    TEST   =C'C'
0436  X                                 SALIR
0437          1E78              E53     EQU    *
0438                                    TEST   = C'D'
0439  X                                 SALIR
```

```
0440          1E8A          E54     EQU  *
0441                                TEST = C'E'
0442  X                             SALIR
0443          1E9C          E55     EQU  *
0444                                TEST = C'F'
0445  X                             SALIR
0446          1EAE          E56     EQU  *
0447                                TEST = C'G'
0448  X                             SALIR
0449          1EC0          E57     EQU  *
0450                                TEST =C'H'
0451  X                             SALIR
0452          1ED2          E58     EQU  *
0453                                TEST = C'I'
0454  X                             SALIR
0455          1EF4          E59     EQU  *
0456                                TEST =C'J'
0457  X                             SALIR
0458          1EF6          E60     EQU  *
0459                                TEST = C'K'
0460  X                             SALIR
0461          1F08          E61     EQU  *
0462                                TEST = C'L'
0463  X                             SALIR
0464          1F1A          E62     EQU  *
0465                                TEST = C'M'
0466  X                             SALIR
0467          1F2C          E63     EQU  *
0468                                TEST = C'N'
0469  X                             SALIR
0470          1F3E          E64     EQU  *
0471                                TEST = C'O'
0472  X                             SALIR
0473          1F50          E65     EQU  *
0474                                TEST = C'P'
0475  X                             SALIR
0476          1F62          E66     EQU  *
0477                                TEST = C'Q'
0478  X                             SALIR
0479          1F74          E67     EQU  *
0480                                TEST = C'R'
0481  X                             SALIR
0482          1F86          E68     EQU  *
0483                                TEST = C'S'
```

```
0484 X                                      SALIR
0485          1F98          E69             EQU  *
0486                                        TEST = C'T'
0487 X                                      SALIR
0488          1FAA          E70             EQU  *
0489                                        TEST = C'U'
0490 X                                      SALIR
0491          1FBC          E71             EQU  *
0492                                        TEST = C'V'
0493 X                                      SALIR
0494          1FCE          E72             EQU  *
0495                                        TEST = C'W
0496 X                                      SALIR
0497          1FE0          E73             EQU  *
0498                                        TEST = C'X'
0499 X                                      SALIR
0550          1FF2          E74             EQU  *
0501                                        TEST = C'Y'
0502 X                                      SALIR
0503          204A          E75             EQU  *
0504                                        TEST = C'Z'
0505 X                                      SALIR
0506 X                                      VOLVER
0507                                        RUTINA  NUMERO
0508                                        LLAMA   DIGITO
0509                                        ENTRAR  CONTNUME
0510 X                                      VOLVER
0511                                        RUTINA  CONTNUME
0512                                        DECIDE  E40, = C'1'
0513                                        DECIDE  E40, = C'2'
0514                                        DECIDE  E40, = C'3'
0515                                        DECIDE  E40, = C'4'
0516                                        DECIDE  E40, = C'5'
0517                                        DECIDE  E40, = C'6'
0518                                        DECIDE  E40, = C'7'
0519                                        DECIDE  E40, = C'8'
0520                                        DECIDE  E40, = C'9'
0521                                        DECIDE  E40, = C'0'
0522 X                                      SALIR
0523          2086          E40             EQU  *
0524                                        LLAMA   DIGITO
0525                                        ENTRAR  CONTNUME
0526 X                                      VOLVER
0527                                        RUTINA  DIGITO
```

```
0528                                                    DECIDE  E80, = C'0'
0529                                                    DECIDE  E81, = C'1'
0530                                                    DECIDE  E82, = C'2'
0531                                                    DECIDE  E83, = C'3'
0532                                                    DECIDE  E84, = C'4'
0533                                                    DECIDE  E85, = C'5'
0534                                                    DECIDE  E86, = C'6'
0535                                                    DECIDE  E87, = C'7'
0536                                                    DECIDE  E88, = C'8'
0537                                                    TEST  = C'9'
0538          20F4    92F9B000                          MVI     0(11),X'F9'
0539          20F8    AAB02C94                  AH      11, = H'1'
0540    X                                               SALIR
0541          2100                      E80             EQU  *
0542                                                    TEST  = C'0'
0543          210E    92F0B000                          MVI     0(11),X'F0'
0544          2112    AAB02C94                  AH      11, = H'1'
0545    X                                               SALIR
0546          211A                      E81             EQU  *
0547                                                    TEST  = C'1'
0548          2128    92F1B000                          MVI     0(11),X'F1'
0549          212C    AAB02C94                  AH      11, = H'1'
0550    X                                               SALIR
0551          2134                      E82             EQU  *
0552                                                    TEST  = C'2'
0553          2142    92F2B000                          MVI     0(11),X'F2'
0554          2146    AAB02C94                  AH      11, = H'1'
0555    X                                               SALIR
0556          214E                      E83             EQU  *
0557                                                    TEST  = C'3'
0558          215C    92F3B000                          MVI     0(11),X'F3'
0559          2160    AAB02C94                  AH      11, = H'1'
0560    X                                               SALIR
0561          2168                      E84             EQU  *
0562                                                    TEST  = C'4'
0563          2176    92F4B000                          MVI     0(11),X'F4'
0564          217A    AAB02C94                  AH      11, = H'1'
0565    X                                               SALIR
0566          2182                      E85             EQU  *
0567                                                    TEST  = C'5'
0568          2190    92F5B000                          MVI     0(11),X'F5'
0569          2194    AAB02C94                  AH      11, = H'1'
0570    X                                               SALIR
0571          219C                      E86             EQU  *
```

```
0572                                          TEST  = C'6'
0573          21AA  92F6B000          MVI     0(11),X'F6'
0574          21AE  AAB02C94            AH    11, = H'1'
0575  X                                       SALIR
0576          21B6                E87         EQU  *
0577                                          TEST  = C'7'
0578          21C4  92F7B000          MVI     0(11),X'F7'
0579          21C8  AAB02C94            AH    11, = H'1'
0580  X                                       SALIR
0581          21D0                E88         EQU  *
0582                                          TEST  = C'8'
0583          21DE  92F8B000          MVI     0(11),X'F8'
0584          21E2  AAB02C94            AH    11, = H'1'
0585  X                                       SALIR
0586  X                                       VOLVER
0587          21EE  45E025AA      FIN         CLOSE IMPRE
0588          21F2  45E02474                  CLOSE FICH
0589          21F6  A9003333          HPR     X'3333'
0590          21FA  47F01260          B       HOLA
0591          21FE  D20323862380  SALIRSUB    MVC   POINT2(4),POINT
0592          2204  48802382              LH    8,POINT3
0593          2208  AB80238A              SH    8,CUA
0594          220C  40802382              STH   8,POINT3
0595          2210  D50322B02380          CLC   PRINC(4),POINT
0596          2216  47202284              BH    UNDFL
0597          221A  47F02386              B     POINT2
0598          221E  9500238D      ETTEST1 CLI   CARCO*—*
0599          2222  4780222A              BE    *+8
0600          2226  45902262              BAL   9,FALSO
0601          222A  A601238E              AI    BUFIND,X'01'
0602          222E  D200238C238D          MVC   CARCO—1(1),CARCO
0603          2234  4880238E              LH    8,BUFIND
0604          2238  D200238D8000          MVC   CARCO,0(8)
0605          223E  47F0D004              B     4(,13)
0606          2242  A6042382      ETLLAMA AI    POINT+2,X'04'
0607          2246  D5032380237C          CLC   POINT(4),FINPILA
0608          224C  4780229A              BE    OVERF
0609          2250  D20123842382          MVC   POINT1(2),POINT+2          SAVEAREA
0610          2256  A6022384              AI    POINT1,X'02'
0611          225A  48802384              LH    8,POINT1
0612          225E  47F0D000              B     0(,13)
0613          2262  D28323EB23EA  FALSO   MVC   ARIM,BLAN
0614          2268  D20723FF2CAB          MVC   ARIM+20(8), = C'CARACTER'
0615          226E  D2002409238D          MVC   ARIM+30(1),CARCO
```

```
0616  2274  D214240B2CB3            MVC   ARIM+32(21), = C'ILEGAL EN EL CONTEXTO'
0617  227A  45E025B623EB            PUT   IMPRE,ARIM
0618  2280  47F09000                B     0(,9)
0619  2284  D28323EB23EA  UNDFL     MVC   ARIM,BLAN
0620  228A  D22323FF2CC8            MVC   ARIM+20(36), = C'UNDERFLOW EN LA PILA DE RECURSIVIDAD'
0621  2290  45E025B623EB            PUT   IMPRE,ARIM
0622  2296  A9001111                HPR   X'1111'
0623  229A  D28323EB23EA  OVERF     MVC   ARIM,BLAN
0624  22A0  D22023FF2CEC            MVC   ARIM+20(33), = C'OVERFLOW EN LA PILA DE RECURSIVIDAD'
0625  22A6  45E025B623EB            PUT   IMPRE,ARIM
0626  22AC  A9002222                HPR   X'2222'
0627  22B0  47F0        PRINC       DC    X'47F0'
0628  22B2  22B0                    DC    Y(PRINC)
0629  22B4  47F0000047F00000  PILA  DC    50X'47F00000'
      22BC  47F0000047F00000
      22C4  47F0000047F00000
      22CC  47F0000047F00000
      22D4  47F0000047F00000
      22DC  47F0000047F00000
      22E4  47F0000047F00000
      22EC  47F0000047F00000
      22F4  47F0000047F00000
      22FC  47F0000047F00000
      2304  47F0000047F00000
      230C  47F0000047F00000
      2314  47F0000047F00000
      231C  47F0000047F00000
      2324  47F0000047F00000
      232C  47F0000047F00000
      2334  47F0000047F00000
      233C  47F0000047F00000
      2344  47F0000047F00000
      234C  47F0000047F00000
      2354  47F0000047F00000
      235C  47F0000047F00000
      2364  47F0000047F00000
      236C  47F0000047F00000
      2374  47F0000047F00000
0630  237C  47F0        FINPILA     DC    X'47F0'
0631  237E  237C                    DC    Y(FINPILA)
0632  2380  47F0        POINT       DC    X'47F0'
0633  2382  22B0        POINT3      DC    Y(PRINC)
0634  2384  0000        POINT1      DC    Y(0)
0635  2386  0000        POINT2      DC    Y(0)
```

```
0636  2388  0000                DC    Y(0)
0637  238A  0004          CUA   DC    Y(4)
0638  238C  40                  DC    C' '
0639  238D  40            CARCO DC    C' '
0640  238E  2390          BUFIND DC   Y(RDBUF)
0641  2390  4040404040404040 RDBUF DC  80C' '
      2398  4040404040404040
      23A0  4040404040404040
      23A8  4040404040404040
      23B0  4040404040404040
      23B8  4040404040404040
      23C0  4040404040404040
      23C8  4040404040404040
      23D0  4040404040404040
      23D8  4040404040404040
0642  23E0  2390          TEMP  DC    Y(RDBUF)
0643  23E2  000F          CON   DC    X'000F'
0644  23E4  F0202120      MASK  DC    X'F0202120'
0645  23E8  0002          DOS   DC    Y(2)
0646  23EA  40            BLAN  DC    C' '
0647  23EB                ARIM  DS    CL1 32
0648                      FICH  DTFCS EOFA = FIN
0649                      IMPRE DTFPR DEVA = 10,PROV = YES
0650  27EA  29BC          BASEG DC    Y(GUIA)
0651  27EC  48A029BA      GENERAD LH  10,DIRO
0652  27F0  D501A0002D0F  COMIENZO CLC 0(2,10), = X'D000'  COMPARAR CON RUTINA INDIRECTA EN GUIA
0653  27F6  4740280C            BL    COM
0654  27FA  48D02A54            LH    13,BASETSUB
0655  27FE  D2012806A000        MVC   *+8(2),0(10)
0656  2804  45F00000            BAL   15,*—*
0657  2808  47F02872            B     FINGENER
0658  280C  48BCA000      COM   LH    11,0(,10)          APUNTAR A LA TABLA CON R11
0659  2810  D200288A2D11        MVC   CONT, = PL1'1'     INICIALIZAR CONTADOR DE POINTER
0660  2816  D501B0002C96  BUCLE CLC   0(2,11), = Y(0000) COMPARAR CON 0000
0661  281C  47802872            BE    FINGENER           FIN DE LA GENERACION
0662  2820  D501B0002D0F        CLC   0(2,11), = X'D000' COMPARAR CON RUTINA INDIRECTA EN GFGO
0663  2826  47B02860            BNL   GENINDI            IR A GENERACION INDIRECTA
0664  282A  48D0B000            LH    13,0(,11)          R13 DONDE ESTAN LAS INSTR. PREFABRI.
0665  282E  956DD000      COMP  CLI   0(13),X'6D'        COMPARAR CON GD PARA VER SI FIN
0666  2832  47802848            BE    SUMAR              IR A INCREMENTAR POINTERS
0667  2836  D200C000D000        MVC   0(1,12),0(13)      MOVER EL CARACTER AL AREA DE SALIDA
0668  283C  AAD02C94            AH    13, = H'1'         SUMAR UNO A R13
0669  2840  AAC02C94            AH    12, = H'1'         R12 = POINTER DEL AREA DE SALIDA
0670  2844  47F0282E            B     COMP               IR A VER SI FIN
```

```
0671          2848    AAB02CC6E   SUMAR     AH      11, = H'2'          SUMAR UNO A R11
0672          284C    FA00288A2D11          AP      CONT, = P'1'        SUMAR UNO AL CONTADOR DE POINTER
0673          2852    D500288A2D12          CLC     CONT, = P'3'        COMPARARLO CON TRES
0674          2858    47202872              BH      FINGENER            IR A FIN GENERACION
0675          285C    47F02816              B       BUCLE               IR A INVESTIGAR LA SIGUIENTE DIREC.
0676          2860    48D02A54    GENINDI   LH      13,BASETSUB
0677          2864    D201286CB000          MVC     *+8(2),0(11)        PONE LA DIRECCION A LA SIGUIEN. INST.
0678          286A    45F00000              BAL     15,*-*              IR A LA CORRESPONDIENTE RUTINA
0679          286E    47F02848              B       SUMAR
0680          2872    D200C0002CA6  FINGENER MVC    0(1,12), = X'6D'    MOVER UN CARACTER COMO FIN
0681          2878    AAA02C6E              AH      10, = H'2'          SUMAR DOS AL POINTER DE GUIA
0682          287C    D502A0002CA6          CLC     0(3,10),=X'6D6D6D'  COMPARAR CON CARACTERES ESPECIALES
0683          2882    477027F0              BNE     COMIENZO            SI NO IR A GENERAR
0684  X                                     SALIR
0685          288A    0C          CONT      DC      PL1'0'
0686          288C    2982        GFGO      DC      Y(TABGE+126)
0687          288E    0000                  DC      Y(0)
0688          2890    0000                  DC      Y(0)
0689          2892    298A                  DC      Y(TABGE+134)
0690          2894    0000                  DC      Y(0)
0691          2896    0000                  DC      Y(0)
0692          2898    2912                  DC      Y(TABGE+14)
0693          289A    0000                  DC      Y(0)
0694          289C    0000                  DC      Y(0)
0695          289E    2992                  DC      Y(TABGE+142)
0696          28A0    0000                  DC      Y(0)
0697          28A2    0000                  DC      Y(0)
0698          28A4    2922                  DC      Y(TABGE+30)
0699          28A6    0000                  DC      Y(0)
0700          28A8    0000                  DC      Y(0)
0701          28AA    292A                  DC      Y(TABGE+38)
0702          28AC    0000                  DC      Y(0)
0703          28AE    0000                  DC      Y(0)
0704          28B0    2932                  DC      Y(TABGE+46)
0705          28B2    0000                  DC      Y(0)
0706          28B4    0000                  DC      Y(0)
0707          28B6    D000                  DC      X'D000'
0708          28B8    0000                  DC      Y(0)
0709          28BA    0000                  DC      Y(0)
0710          28BC    D010                  DC      X'D010'
0711          28BE    0000                  DC      Y(0)
0712          28C0    0000                  DC      Y(0)
0713          28C2    D008                  DC      X'D008'
0714          28C4    0000                  DC      Y(0)
```

```
0715  28C6  0000                                DC    Y(0)
0716  28C8  293A                                DC    Y(TABGE+54)
0717  28CA  0000                                DC    Y(0)
0718  28CC  0000                                DC    Y(0)
0719  28CE  29A6                                DC    Y(TABGE+162)
0720  28D0  0000                                DC    Y(0)
0721  28D2  0000                                DC    Y(0)
0722  28D4  295A                                DC    Y(TABGE+86)
0723  28D6  0000                                DC    Y(0)
0724  28D8  0000                                DC    Y(0)
0725  28DA  295F                                DC    Y(TABGE+91)
0726  28DC  0000                                DC    Y(0)
0727  28DE  0000                                DC    Y(0)
0728  28E0  2954                                DC    Y(TABGE+80)
0729  28E2  0000                                DC    Y(0)
0730  28E4  0000                                DC    Y(0)
0731  28E6  295A                                DC    Y(TABGE+86)
0732  28E8  0000                                DC    Y(0)
0733  28EA  0000                                DC    Y(0)
0734  28EC  2974                                DC    Y(TABGE+112)
0735  28EE  0000                                DC    Y(0)
0736  28F0  0000                                DC    Y(0)
0737  28F2  2973                                DC    Y(TABGE+111)
0738  28F4  0000                                DC    Y(0)
0739  28F6  0000                                DC    Y(0)
0740  28F8  2978                                DC    Y(TABGE+116)
0741  28FA  0000                                DC    Y(0)
0742  28FC  0000                                DC    Y(0)
0743  28FE  297D                                DC    Y(TABGE+121)
0744  2900  0000                                DC    Y(0)
0745  2902  0000                                DC    Y(0)
0746  2904  00000000000000          TABGE       DC    X'00000000000000'
0747  290B  00000000000000                      DC    X'00000000000000'
0748  2912  D205FFFF10006D6D                    DC    X'D205FFFF10006D6D6D'
      291A  6D
0749  291B  FD551000FFFF6D                      DC    X'FD551000FFFF6D'
0750  2922  D2051000FFFF6D6D                    DC    X'D2051000FFFF6D6D'
0751  292A  FA551000FFFF6D6D                    DC    X'FA551000FFFF6D6D'
0752  2932  F9551000FFFF6D6D                    DC    X'F9551000FFFF6D6D'
0753  293A  FB551000FFFF6D6D                    DC    X'FB551000FFFF6D6D'
0754  2942  FC551000FFFF6D6D                    DC    X'FC551000FFFF6D6D'
0755  294A  02C66D6D                            DC    X'02C66D6D'
0756  294E  45C055006D6D                        DC    X'45C055006D6D'
0757  2954  A90007776D6D                        DC    X'A90007776D6D'
```

```
0758          295A    47F0FFFF6D6D                DC      X'47F0FFFF6D6D'
0759          2960    45B060006D6D                DC      X'45B060006D6D'
0760          2966    45B060506D6D                DC      X'45B060506D6D'
0761          296C    45B061006D6D6D              DC      X'45B061006D6D6D'
0762          2973    4740FFFF6D                  DC      X'4740FFFF6D'
0763          2978    4780FFFF6D                  DC      X'4780FFFF6D'
0764          297D    4720FFFF6D                  DC      X'4720FFFF6D'
0765          2982    45802A56                    BAL     8,TSUB
0766          2986    FFFF6D6D                    DC      X'FFFF6D6D'
0767          298A    45802A5A                    BAL     8,TSUB+4
0768          298E    FFFF6D6D                    DC      X'FFFF6D6D'
0769          2992    F8B52D501000FDB5            DC      X'F8B52D501000FDB52D50FFFFF85510002D506D6D'
              299A    2D50FFFFF8551000
              29A2    2D506D6D
0770          29A6    F8B52D501000FCB5            DC      X'F8B52D501000FCB52D50FFFFF85510002D566D6D'
              29AE    2D50FFFFF8551000
              29B6    2D566D6D
9771          29BA    29BC        DIRO    DC      Y(GUIA)
0772          29BC                GUIA    DS      CL100
0773          2A20    2A22        BASE 1  DC      Y(TERMGUIA)
0774          2A22                TERMGUIA DS     CL50
0775          2A54    2A56        BASETSUB DC     Y(TSUB)
0776          2A56    47F02C02    TSUB    B       EVALEE          TRANSFERENCIA A LEER
0777          2A5A    47F02C20            B       EVAESCR         TRANSFERENCIA A ESCRIBIR
0778          2A5E    47F02B90            B       EVALDIRE        TRANSFERENCIA A EVALUAR (XX)
0779          2A62    47F02A76            B       EVALCD          TRANSFERENCIA A EVALUAR :XX
0780          2A66    47F02B42            B       EVALNUME        TRANSFERENCIA A EVALUAR = XXXXXX
0781          2A6A    47F02C3E            B       EJECUCI         TRANSFERENCIA A EJECUCION
0782          2A6E    47F0F000            DC      X'47F0F000'
0783          2A72    47F0F000            DC      X'47F0F000'
0784                                      *       EVALUACION DE :XX
0785          2A76    40F02AF0    EVALCD  STH     15,SALR15       SALVA R15
0786          2A7A    40C02AF2            STH     12,SALR12       SALVA R12
0787          2A7E    48F013EC            LH      15,DPROGFU
0788          2A82    F2112AF42A24        PACK    AR,TERMGUIA+2
0789          2A88    ABF02C98    AHI     SH      15, = H'4'
0790          2A8C    F9112AF4F000        CP      AR(2),0(2,15)
0791          2A92    47802ABA            BE      AHI1            ENCONTRANDO
0792          2A96    47202ADA            BH      ALMACEN         REQUIERE SEGUNDO PASO
0793          2A9A    F9112AF42D13        CP      AR(2), = X'999F'
0794          2AA0    47702A88            BNE     AHI
0795          2AA4    D28323EB23EA        MVC     ARIM,BLAN
0796          2AAA    D23C23EB2AF6        MVC     ARIM(61),MENSAGE
0797          2AB0    45E025B623EB        PUT     IMPRE,ARIM
```

```
0798  2AB6  47F02ACE                    B       EXIT1
0799  2ABA  ABC02C6E          AHI1      SH      12,=H'2'      BUSCAR      TRABG2
0800  2ABE  D501C0002D15                CLC     0(2,12),=X'FFFF'
0801  2AC4  47702ABA                    BNE     AHI1
0802  2AC8  D201C000F002                MVC     0(2,12),2(15)      ALMACENAR LO EVALUADO
0803  2ACE  48C02AF2          EXIT1     LH      12,SALR12
0804  2AD2  48F02AF0                    LH      15,SALR15
0805  2AD6  47F0F000                    B       0(,15)
0806  2ADA  48C02B34          ALMACEN   LH      12,PONPILA1
0807  2ADE  D201C0002A24                MVC     0(2,12),TERMGUIA+2
0808  2AE4  AAC02C6E                    AH      12,=H'2'
0809  2AE8  40C02B34                    STH     12,PONPILA1
0810  2AEC  47F02ACE                    B       EXIT1
0811  2AF0  0000              SALR15    DC      Y(00)
0812  2AF2  0000              SALR12    DC      Y(00)
0813  2AF4                    AR        DS      CL2
0814  2AF6  C5E2E3C1D540C4C5  MENSAGE   DC      C'ESTAN DESCABALAD'
      2AFE  E2C3C1C2C1D3C1C4
0815  2B06  D6E240D3D6E240D5            DC      C'OS LOS NUMEROS'
      2B0E  E4D4C5D9D6E240
0816  2B15  C4C540C9D5E2E340            DC      C'DE INST EN LA'
      2B1D  C5D540D3C1404040
0817  2B25  E3C1C2D3C140D7D9            DC      C'TABLA PROGFUDI'
      2B2D  D6C7C6E4C4C9
0818  2B34  2B36            PONPILA1  DC      Y(PILA1)
0819  2B36  0000000000000000  PILA1     DC      5X'0000'
      2B3E  0000
0820  2B40  2B36              DPILA1    DC      Y(PILA1)
0821                          *         EVALUACION  DE  =XXXXXX
0822  2B42  40C02AF2          EVALNUME  STH     12,SALR12
0823  2B46  40F02AF0                    STH     15,SALR15
0824  2B4A  48F02B8C                    LH      15,DPOLNUME
0825  2B4E  AAF02C9A                    AH      15,=H'6'
0826  2B52  F255F0002A22                PACK    0(6,15),TERMGUIA(6)
0827  2B58  40F02B8C                    STH     15,DPOLNUME
0828  2B5C  48F02B8E                    LH      15,DDPOLNUM
0829  2B60  47F02ABA                    B       AHI1
0830  2B64                    POLNUME   DS      CL40
0831  2B8C  2B5E              DPOLNUME  DC      Y(POLNUME—6)
0832  2B8E  2B8A              DDPOLNUM  DC      Y(DPOLNUME—2)
0833                          *         EVALUACION  DE  (XX)
0834  2B90  40F02AF0          EVALDIRE  STH     15,SALR15          SALVAR REGISTRO 15
0835  2B94  D2002BF52A22                MVC     CAM1+1(1),TERMGUIA MOVER EL PRIMER DIGITO DE (XX)
0836  2B9A  D2002BF72A23                MVC     CAM+3(1),TERMGUIA+1 MOVER EL SEGUNDO DIGITO DE (XX)
```

```
0837   2BA0   D4032BF42D17                NC     CAM1(4), = X'000F000F'  QUITAR LAS ZONAS
0838   2BA6   DC002BF52BF8                TR     CAM1+1(1),TRADU1        TRADUCCION A SU EQUIVAL BIN ARIO
0839   2BAC   48F02BF4                    LH     15,CAM1                 CARGARLO EN R15
0840   2BBC   AAF02BF6                    AH     15,CAM1+2               SUMAR LAS DOS CIFRAS
0841   2BB4   40F02BF4                    STH    15,CAM1                 *
0842   2BB8   AAF02BF4                    AH     15,CAM1                 *
0843   2BB0   40F02BF4                    STH    15,CAM1                 * MULTI EL RESULT POR 6
0844   2BC0   AAF02BF4                    AH     15,CAM1                 *
0845   2BC4   AAF02BF4                    AH     15,CAM1                 *
0846   2BC8   AAF02C9C                    AH     15,*=Y(4102)            BASE DE LAS PALABRAS 9013
0847   2BCC   40F02BF4                    STH    15,CAM1
0848   2BD0   48F02BF2                    LH     15,DCAM1
0849   2BD4   40C02AF2                    STH    12,SALR12               SALVAR REGISTRO  12
0850   2BD8   47F02ABA                    B      AHI1                    APROVECHAR LOOP FINAL DE   :XX
0851   2BDC   F0F1F2F3F4F5F6F7  T77       DC     C'0123456789ABCDEF'
       2BE4   F8F9C1C2C3C4C5C6
0852   2BEC                    RESER      DS     CL2
0853   2BEE                    M          DS     CL4
0854   2BF2   2BF2            DCAM1      DC     Y(CAM1—2)
0855   2BF4   0000            CAM1       DC     Y(0)
0856   2BF6   0000                       DC     Y(0)
0857   2BF8   00              TRADU1     DC     HL1'0'                  TABLA DE TRADUCCION
0858   2BF9   0A                         DC     HL1'10'
0859   2BFA   14                         DC     HL1'20'
0860   2BFB   1E                         DC     HL1'30'
0861   2BFC   28                         DC     HL1'40'
0862   2BFD   32                         DC     HL1'50'
0863   2BFE   3C                         DC     HL1'60'
0864   2BFF   46                         DC     HL1'70'
0865   2C00   50                         DC     HL1'80'
0866   2C01   5A                         DC     HL1'90'
0867                           *      LECTURA
0868   2C02   48C08000        EVALEE     LH     12,0(,8)
0869   2C06   45E024782390               GET    FICH,RDBUF
0870   2C0C   F255C0002390               PACK   0(6,12),RDBUF
0871   2C12   47F08002                   B      2(,8)
0872   2C16   4040404040404040          DC     10C' '
       2C1E   4040
0873                           *      ESCRIBIR
0874   2C20   48C08000        EVAESCR    LH     12,0(,8)
0875   2C24   D28323EB23EA               MVC    ARIM,BLAN
0876   2C2A   F3A523EBC000               UNPK   ARIM(11),0(6,12)
0877   2C30   96F023F5                   OI     ARIM+10,X'F0'
9878   2C34   45E025B623EB               PUT    IMPRE,ARIM
```

```
0879          2C3A    47F08002                  B       2(,8)
0880                                  *       EJECUCION
0881          2C3E    48F02B34    EJECUCI       LH      15,PONPILA1
0882          2C42    ABF02C6E                  SH      15, = H'2'
0883          2C46    40F02B34                  STH     15,PONPILA1
0884          2C4A    49F02B40                  CH      15,DPILA1
0885          2C4E    47402C60                  BL      READY           EJECUTAR CODIGO OBJETO
0886          2C52    D2012A24F000              MVC     TERMGUIA+2(2),0(15)
0887          2C58    45F02A76                  BAL     15,EVALCD
0888          2C5C    47F02C3E                  B       EJECUCI
0889          2C60    48F02C68    READY         LH      15,DTRABG2
0890          2C64    47F0F000                  B       0(,15)
0891          2C68    2D94        DTRABG2       DC      Y(*+300)
              2C6A
              2C6A
              2C6A    23F5
              2C6C    0001
              2C6E    0002
              2C70    288C
              2C72    2892
              2C74    2898
              2076    289E
              2C78    28A4
              2C7A    28AA
              2C7C    28B0
              2C7E    28C2
              2C80    28BC
              2C82    28C8
              2C84    28CE
              2C86    28DA
              2C88    28E0
              2C8A    28E6
              2C8C    28EC
              2C8E    28FE
              2C90    28F2
              2C92    28F8
              2C94    0001
              2C96    0000
              2C98    0004
              2C9A    0006
              2C9C    1006
              2C9E    0F000000000F0F1F
              2CA6    6D6D6DD00CC3C1D9
              2CAE    C1C3E3C5D9C9D3C5
```

```
                2CB6    C7C1D340C5D540C5
                2CBE    D340C3D6D5E3C5E7
                2CC6    E3D6E4D5C4C5D9C6
                2CCE    D3D6E640C5D540D3
                2CD6    C140D7C9D3C140C4
                2CDE    C540D9C5C3E4D9E2
                2CE6    C9E5C9C4C1C4D6E5
                2CEE    C5D9C6D3D6E640C5
                2CF6    D540D3C140D7C9D3
                2CFE    C140C4C540D9C5C3
                2D06    E4D9E2C9E5C9C4C1
                2D0E    C4D0001C3C999FFF
                2D16    FF000F000F
0892            2D1B    000000001260        END     HOLA
```

Programas ejemplo del lenguaje 9013

Suma de los cien primeros
números

```
2D80        D20510002B50
2D86        D20510181000
2D8C        D2051002B56
2D92        D205100C1000
2D98        D20510121000
2D9E        D2051000100C
2DA4        F955100010184720FFFF
2DAE        FA5510002B5C
2DB4        D205100C1000
2DBA        D20510001012
2DC0        FA551000100C
2DC6        D20510121000
2DCC        47F02D9E
2DD0        45802A461012
2DD6        45802A46100C
2DDC        A9000777

00000005151
00000000101
```

```
0001    TRASLADAR AL ACUMULADOR EL NUMERO = 000100
0002    DEPOSITAR EL CONTENIDO DEL ACUMULADOR EN LA POSICION (03)
0003    TRASLADAR AL ACUMULADOR EL NUMERO = 000000
0004    DEPOSITAR EL CONTENIDO DEL ACUMULADOR EN LA POSICION (01)
0005    DEPOSITAR EL CONTENIDO DEL ACUMULADOR EN LA POSICION (02)
0006    TRASLADAR AL ACUMULADOR EL CONTENIDO DE LA POSICION (01)
0007    SIEL CONTENIDO DE (03)<(AC) PONER EN EL CD LA DIRECCION :14
0008    SUMAR AL ACUMULADOR EL NUMERO = 000001
0009    DEPOSITAR EL CONTENIDO DEL ACUMULADOR EN LA POSICION (01)
0010    TRASLADAR AL ACUMULADOR EL CONTENIDO DE LA POSICION (02)
0011    SUMAR AL ACUMULADOR EL CONTENIDO DE LA POSICION (01)
0012    DEPOSITAR EL CONTENIDO DEL ACUMULADOR EN LA POSICION (02)
0013    PONER EN EL CD LA DIRECCION :06
0014    ESCRIBIR EN LA UNIDAD DE SALIDA EL CONTENIDO DE LA POSICION (02)
0015    ESCRIBIR EN LA UNIDAD DE SALIDA EL CONTENIDO DE LA POSICION (01)
0016    PARAR
0017    FIN
```

Media aritmética de números
leídos desde fichas

2D94	D20510002B64
2D9A	D20510001000
2DA0	D20510002B6A
2DA6	D20510121000
2DAC	D20510181000
2DB2	45802A56107E
2DB8	D2051000107E
2DBE	F9551000100C4780FFFF
2DC8	45802A5A107E
2DCE	FA5510001018
2DD4	D20510181000
2DDA	D20510001012
2DE0	FA5510002B70
2DE6	D20510121000
2DEC	47F02DB2
2DF0	D20510001018
2DF6	F8B52D501000FDB52D501012F85510002D50
2E08	D20510241000
2E0E	45802A5A1024
2E14	45802A5A1018
2E1A	45802A5A1012
2E20	A9000777

0001	TRASLADAR AL ACUMULADOR EL NUMERO = 999999
0002	DEPOSITAR EL CONTENIDO DEL ACUMULADOR EN LA POSICION (01)
0003	TRASLADAR AL ACUMULADOR EL NUMERO = 000000
0004	DEPOSITAR EL CONTENIDO DEL ACUMULADOR EN LA POSICION (02)
0005	DEPOSITAR EL CONTENIDO DEL ACUMULADOR EN LA POSICION (03)
0006	LEER EL CONTENIDO DE LA UNIDAD DE ENTRADA EN LA POSICION (20)
0007	TRASLADAR AL ACUMULADOR EL CONTENIDO DE LA POSICION (20)
0008	SIEL CONTENIDO DE (01) = (AC) PONER EN EL CD LA DIRECCION :16
0009	ESCRIBIR EN LA UNIDAD DE SALIDA EL CONTENIDO DE LA POSICION (20)
0010	SUMAR AL ACUMULADOR EL CONTENIDO DE LA POSICION (03)
0011	DEPOSITAR EL CONTENIDO DEL ACUMULADOR EN LA POSICION (03)
0012	TRASLADAR AL ACUMULADOR EL CONTENIDO DE LA POSICION (02)
0013	SUMAR AL ACUMULADOR EL NUMERO = 000001
0014	DEPOSITAR EL CONTENIDO DEL ACUMULADOR EN LA POSICION (02)
0015	PONER EN EL CD LA DIRECCION :06
0016	TRASLADAR AL ACUMULADOR EL CONTENIDO DE LA POSICION (03)
0017	DIVIDIR EL ACUMULADOR POR EL CONTENIDO DE LA POSICION (02)
0018	DEPOSITAR EL CONTENIDO DEL ACUMULADOR EN LA POSICION (05)
0019	ESCRIBIR EN LA UNIDAD DE SALIDA EL CONTENIDO DE LA POSICION (05)
0020	ESCRIBIR EN LA UNIDAD DE SALIDA EL CONTENIDO DE LA POSICION (03)
0021	ESCRIBIR EN LA UNIDAD DE SALIDA EL CONTENIDO DE LA POSICION (02)
0022	PARAR
0023	TRASLADAR AL ACUMULADOR EL CONTENIDO DE LA POSICION (05)

```
2E24    D20510001024
2E2A    F8B52D501000FCB52D502B76F85510002D56
2E3C    D205102A1000
2E42    45802A5A102A
2E48    A9000777

00000123456
00000234567
00000689452
00000000001
00000257469
00000145789
00000258640
00000789452
00000312353
00002498826
00000000008
00000624706
```

```
0024    MULTIPLICAR AL ACUMULADOR POR EL NUMERO = 000002
0025    DEPOSITAR EL CONTENIDO DEL ACUMULADOR EN LA POSICION (06)
0026    ESCRIBIR EN LA UNIDAD DE SALIDA EL CONTENIDO DE LA POSICION (06)
0027    PARAR
0028    FIN
```


MEC