



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

Z.315

LENGUAJE HOMBRE-MÁQUINA

**ESPECIFICACIÓN DE LA SINTAXIS DEL
LENGUAJE DE ENTRADA
(INSTRUCCIÓN)**

Recomendación UIT-T Z.315

(Extracto del *Libro Azul*)

NOTAS

1 La Recomendación UIT-T Z.315 se publicó en el fascículo X.7 del Libro Azul. Este fichero es un extracto del Libro Azul. Aunque la presentación y disposición del texto son ligeramente diferentes de la versión del Libro Azul, el contenido del fichero es idéntico a la citada versión y los derechos de autor siguen siendo los mismos (Véase a continuación).

2 Por razones de concisión, el término «Administración» se utiliza en la presente Recomendación para designar a una administración de telecomunicaciones y a una empresa de explotación reconocida.

© UIT 1988, 1993

Reservados todos los derechos. No podrá reproducirse o utilizarse la presente Recomendación ni parte de la misma de cualquier forma ni por cualquier procedimiento, electrónico o mecánico, comprendidas la fotocopia y la grabación en micropelícula, sin autorización escrita de la UIT.

ESPECIFICACIÓN DE LA SINTAXIS DEL LENGUAJE DE ENTRADA (INSTRUCCIÓN)

1 Consideraciones generales

A continuación se describen los elementos del lenguaje de entrada. Los diagramas sintácticos del lenguaje de entrada se indican en el § 4, en subdivisiones cuyos números se corresponden con los del § 2. Cuando en la salida se utilizan elementos de entrada, se hace referencia a la Recomendación Z.316, donde se especifica la sintaxis del lenguaje de salida. Los aspectos relativos al procedimiento se tratan en la Recomendación Z.317. Conviene tener en cuenta que ciertas áreas de la sintaxis permiten aplicar opciones que pueden originar una contradicción de sintaxis. La adopción de estas opciones deberá decidirse en consonancia con el sistema considerado.

2 Estructura de instrucción

2.1 Instrucción

Una instrucción comienza por el código de instrucción, que define la función que ha de realizar el sistema. Si se requiere más información, el código de instrucción puede ir seguido de una parte paramétrica de la que está separada por un : (dos puntos). La parte paramétrica consta de uno o más bloques de parámetros (véanse los § 2.3 y 2.9.1). Una instrucción va siempre completada por un carácter de ejecución (véase la Recomendación Z.317).

2.2 Código de instrucción

El código de instrucción puede estar constituido hasta por tres identificadores separados por un - (guión) (por ejemplo, área funcional - tipo de objeto - acción). Cuando los códigos de instrucción adopten la forma de abreviaturas mnemotécnicas sencillas, se recomienda que consten del mismo número de caracteres.

2.3 Bloque de parámetros

Un bloque de parámetros contiene la información necesaria para ejecutar la función especificada en el código de instrucción. La información contenida en un bloque de parámetros se expresa en forma de cierto número de parámetros propios de la instrucción. Si los parámetros incluidos son más de uno, irán separados por una , (coma). Todos los parámetros de cualquier bloque serán de la misma clase, es decir, parámetros definidos por la posición o parámetros definidos por el nombre.

2.4 Parámetros

Un parámetro identifica y contiene un elemento de información, y puede ser definido por el nombre o definido por la posición. Pueden omitirse los parámetros sin interés de acuerdo con los § 2.4.1 y 2.4.2.

2.4.1 Parámetro definido por la posición

Un parámetro definido por la posición consiste en un valor de parámetro que puede ir precedido de un nombre de parámetro del cual está separado por un = (signo igual). Los parámetros deben ir en un orden determinado dentro del bloque de parámetros. Cuando no deba darse un valor de parámetro, se omitirá éste, dejando el separador apropiado o el indicador adecuado utilizado para terminar una instrucción. De esta forma queda indicada la posición del parámetro en el bloque de parámetros. La omisión de un parámetro puede implicar que ha de aplicarse el valor por defecto. El valor por defecto puede indicarse también introduciendo un valor de parámetro asignado a tal fin.

2.4.2 Parámetro definido por el nombre

Un parámetro definido por el nombre consiste en un nombre de parámetro seguido de un valor de parámetro del cual está separado por un = (signo igual). Estos parámetros pueden disponerse en un orden arbitrario en el bloque de parámetros. Cuando no deba indicarse un valor de parámetro, se omiten también el nombre de parámetro y el separador = (signo igual), así como el separador , (coma) que siguen al parámetro. Esta omisión puede implicar que se desea el valor por defecto. El valor por defecto puede también indicarse especificando un valor de parámetro destinado a tal fin. Cuando un valor de parámetro implica el nombre de parámetro, pueden, omitirse éste, y el separador = (signo igual).

2.5 *Nombre de parámetro*

Un nombre de parámetro indica unívocamente el tipo y la estructura del valor de parámetro subsiguiente, y por tanto define el valor de parámetro y cómo deberá interpretarse. Se trata de un identificador, y puede ser simple o compuesto. El nombre de parámetro simple indica un solo valor de parámetro, y un nombre compuesto indica un valor de parámetro de una lista o tabla de tipos de parámetros semejantes.

2.5.1 *Nombre de parámetro simple*

Un nombre de parámetro simple consta de un identificador.

2.5.2 *Nombre de parámetro compuesto*

Un nombre de parámetro compuesto consta de uno o más identificadores y/o números índice separados por un - (guión).

2.5.2.1 *Número índice*

Un número índice consta de una o más cifras.

2.6 *Valor de parámetro*

Un valor de parámetro contiene la información necesaria para especificar el objeto (objetos) o valor (valores) adecuados, y se compone de una o más unidades de información. En caso de que no se aplique agrupación de información (véase el § 2.9), un valor de parámetro se reduce a un argumento de parámetro. El § 2.10 trata los aspectos relativos a la búsqueda en bases de datos.

2.7 *Argumento de parámetro*

Un argumento de parámetro contiene la información necesaria para especificar el objeto o valor apropiado. Es la forma que adopta un valor de parámetro cuando no se aplica agrupación de información (véase el § 2.9). Un argumento de parámetro puede ser simple o compuesto.

2.7.1 *Argumento de parámetro simple*

Un argumento de parámetro simple consta de una unidad de información.

2.7.2 *Argumento de parámetro compuesto*

Un argumento de parámetro compuesto consta de dos o más unidades de información separadas por un - (guión).

2.8 *Unidad de información*

Una unidad de información es la menor cantidad de información en el lenguaje, desde el punto de vista sintáctico. Una unidad de información puede ser un numeral, un identificador, un nombre simbólico, una cadena de texto o una expresión aritmética. Un numeral tiene siempre una base por defecto (por ejemplo, hexadecimal) que puede ser invalidada, si es necesario, introduciendo la base deseada, como se especifica en la Recomendación Z.314. No obstante, la base por defecto para un numeral de teclado no puede ser invalidada por otra base.

2.9 *Agrupación de información*

La agrupación de información se utiliza para acelerar y facilitar las operaciones de introducción. Se efectúa introduciendo conjuntos de informaciones del mismo tipo dentro de la misma instrucción.

2.9.1 *Agrupación de bloques de parámetros*

Si deben incluirse varios bloques de parámetros en una introducción, deberán ir separados por un : (dos puntos).

2.9.2 *Agrupación de argumentos de parámetro*

La agrupación de argumentos de parámetro permite conseguir la entrada de más de un argumento de parámetro en un parámetro de una instrucción.

2.9.2.1 *Agrupación de argumentos de parámetro simples*

Es posible indicar varios argumentos de parámetro simples dentro del mismo valor de parámetro separados por un & (y comercial). *Ejemplo 1:* 5&9 significa los argumentos de parámetro simples 5 y 9.

En el caso de una secuencia de argumentos simples consecutivos valor incrementar implícito = 1, los argumentos pueden indicarse escribiendo los argumentos simples inferior y superior separados por && (y comercial, y comercial)¹⁾.
Ejemplo 2: 5&&9 significa los argumentos de parámetro simples 5, 6, 7, 8 y 9.

Puede especificarse un valor incremental explícito a continuación del argumento superior separado por + (signo más, signo más). *Ejemplo 3:* 5&&9 ++ 2 significa los argumentos de parámetro simples 5, 7 y 9.

Cuando sea necesario, pueden utilizarse otras combinaciones de las posibilidades mencionadas. *Ejemplo 4:* 5&&7&9 significa los argumentos de parámetro simples 5, 6, 7 y 9. *Ejemplo 5:* 5&&9 ++ 2&10 significa los argumentos de parámetro simples 5, 7, 9 y 10.

2.9.2.2 Agrupación de argumentos de parámetro compuestos

Es posible indicar varios argumentos de parámetro compuestos dentro del mismo valor de parámetro separados por un & (y comercial). *Ejemplo 1:* 5-1&6-3 significa los dos argumentos de parámetro compuestos 5-1 y 6-3.

Si en un grupo de argumentos de parámetro compuestos solamente difiere la última unidad de información, el primer argumento de parámetro compuesto se especifica completamente, en tanto que todos los argumentos compuestos subsiguientes se representan solamente por sus últimas unidades de información separadas por &- (y comercial, guión). *Ejemplo 2:* 7-1&-3 significa los dos argumentos de parámetro compuestos 7-1 y 7-3.

Si en un grupo de argumentos de parámetro compuestos, solamente difieren la última unidad de información y el grupo constituye una secuencia consecutiva (valor incremental implícito = 1), es posible indicar los argumentos escribiendo las unidades de información inferior y superior separadas por &&- (y comercial, y comercial, guión)¹⁾. *Ejemplo 3:* 7-1&&-3 significa los tres argumentos de parámetro compuestos 7-1, 7-2 y 7-3. *Ejemplo 4:* 7-1&-3&&-5 significa los cuatro argumentos de parámetro compuestos 7-1, 7-3, 7-4 y 7-5.

Puede especificarse un valor incremental explícito a continuación de la unidad de información superior separado por ++ (signo más, signo más).

Cuando sea necesario, puede también aplicarse cualquier combinación de las posibilidades mencionadas. *Ejemplo 5:* 5-1&&-3&8-2&-5&-6 significa los seis argumentos de parámetro compuestos 5-1, 5-2, 5-3, 8-2, 8-5 y 8-6. *Ejemplo 6:* 5-1 &&-7 ++ 2&8-1&-3 significa los seis argumentos de parámetro compuestos 5-1, 5-3, 5-5, 5-7, 8-1 y 8-3.

2.10 Búsquedas en bases de datos

Las búsquedas en bases de datos se expresan en términos de información de proyección y selección. La información de proyección puede representarse por un parámetro. Su nombre identifica la función de proyección. Su grupo de argumento(s) de parámetro identifica el (los) campo(s) apropiado(s) de los registros de datos a visualizar. La información de selección puede representarse por un parámetro en el que el nombre identifica la función de selección y el valor identifica un (o un grupo de) argumento(s) de selección. Un argumento de selección incluye una o más condiciones que deben cumplirse. Una condición se especifica por un identificador y un (o un grupo de) argumento(s) de parámetro separados por un operador relacional. El identificador especifica el nombre del campo del registro a seleccionar. La omisión de la información de selección implica que la búsqueda no es condicional.

Los nombres “proyección” y “selección” se han elegido, solamente para el ejemplo. Pueden utilizarse otros nombres tales como “seleccionar” y “donde”.

Ejemplos:

búsqueda-bdx:	proyección	=	campo a,
	selección	=	(campo c = 0).

Esta instrucción pide los registros que satisfacen el criterio de selección campo c = 0 del conjunto de datos x; sin embargo, sólo es necesario visualizar el campo a de los registros seleccionados.

búsqueda-bdx:	proyección	=	campo a & campo b,
	selección	=	(campo b > 5, campo c = 1).

¹⁾ La interpretación de los separadores && (y comercial, y comercial) y &&- (y comercial, y comercial guión) no es exclusiva, existiendo otras interpretaciones. Una alternativa implicaría que no hay ningún incremento específico inherente a la sintaxis. La relación entre los valores superior e inferior de la secuencia es una relación semántica que depende de la función para la cual se especifica la secuencia.

Esta instrucción pide los registros que satisfacen los dos criterios de selección campo b > 5 y campo c = 1 del conjunto de datos. La visualización resultante sólo necesita presentar los campos a y b de los registros seleccionados.

búsqueda-bdz: proyección = campo a & campo b & campo d,
selección = (campo d <= 7, campo e = 0) & (campo b = P).

Esta instrucción pide al conjunto de datos z los registros que satisfacen los dos criterios campo d <= 7 y campo e = 0. También pide los registros que satisfacen el criterio campo b = P. La visualización de todos los registros seleccionados sólo necesita presentar los campos a, b y c de los registros seleccionados.

Aviso

La utilización de los caracteres , (coma) y & (y comercial) en el LHM del CCITT corresponde a los operadores Y y O en la lógica de predicados. Se hace la suposición general de que el personal normal de explotación no utiliza esta lógica. También pueden evitarse confusiones realizando las funciones de los diversos caracteres de separación del LHM del CCITT. La coma se utiliza como separador de parámetros dentro de un bloque, en el que todos los parámetros juntos desempeñan un papel al ejecutar la instrucción. La y comercial actúa como separador en el agrupamiento de información, y se utiliza para introducir una instrucción “valor 1 & valor 2”, como alternativa a la introducción de dos instrucciones, una para el “valor 1” y otra para “valor 2”.

Restricción

Para evitar expresiones sin sentido, si el argumento de parámetro del diagrama sintáctico 4.10.1.1 (condición) se utiliza en combinación con un operador relacional no simétrico, debe limitarse a numerales. Sin embargo, se permiten identificadores y nombres simbólicos si representan miembros de un conjunto ordenado.

3 Correcciones y borrado de instrucciones

Pueden hacerse correcciones borrando una entrada y presentando una nueva.

No se proponen caracteres específicos dada la diversa naturaleza de los dispositivos terminales de entrada/salida disponibles.

3.1 Borrar último carácter

Esta facilidad puede utilizarse para borrar los caracteres de entrada sucesivos hasta la última salida del sistema (véase el § 3.2).

3.2 Borrar hasta la última salida del sistema

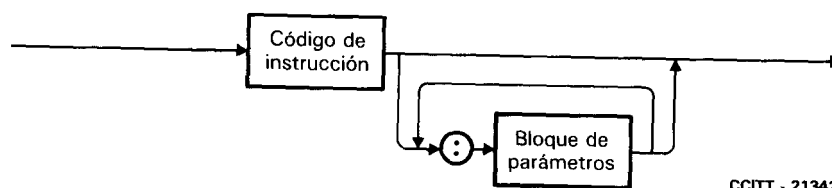
Esta facilidad borra todos los caracteres de entrada después de la última salida del sistema, que puede ser el indicador de preparado o una salida de sugerencia (véase la Recomendación Z.317).

3.3 Borrar instrucción

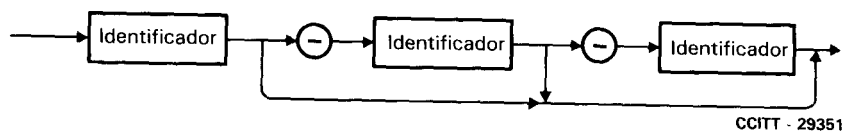
La petición borrar instrucción es transmitida por el carácter CAN (anulación). El empleo de este carácter hace que el sistema responda con un acuse de recibo que presenta la entrada después de que se haya anulado la última instrucción ejecutada. El sistema debe responder con una nueva indicación de preparado, para notificar que está en espera de un nuevo código de instrucción (véase la Recomendación Z.317).

4 Definición de la estructura del lenguaje de entrada (instrucción) mediante diagramas sintácticos

4.1 Instrucción

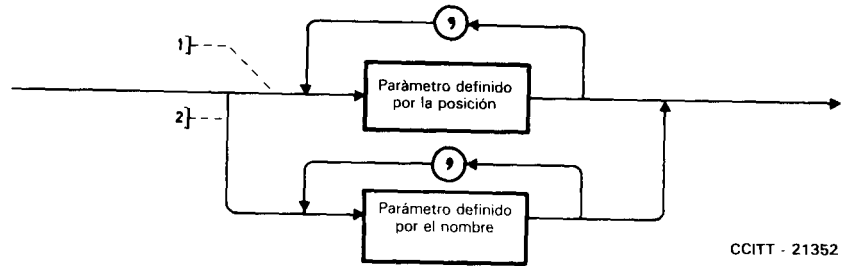


4.2 Código de instrucción



CCITT - 29351

4.3 Bloque de parámetros

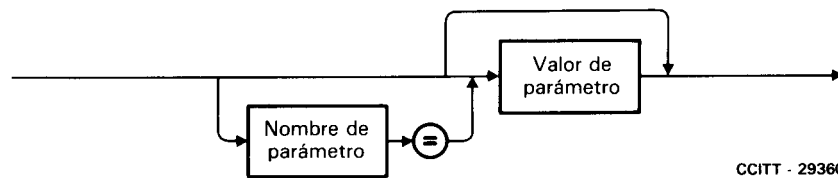


CCITT - 21352

- 1) La rama principal superior sólo es válida para bloques de parámetros definidos por la posición.
- 2) La rama principal inferior sólo es válida para bloques de parámetros definidos por el nombre de parámetro.

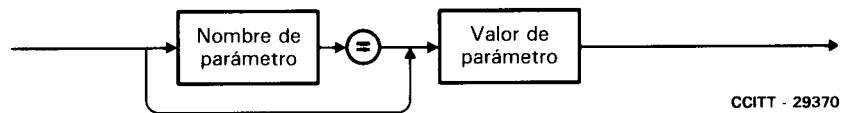
4.4 Parámetros

4.4.1 Parámetro definido por la posición



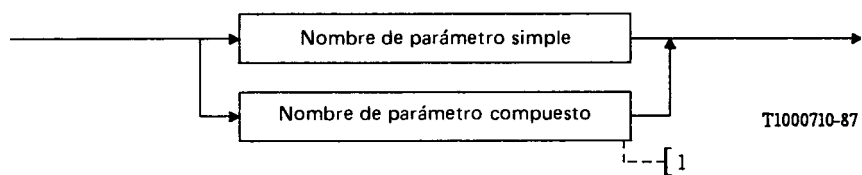
CCITT - 29360

4.4.2 Parámetro definido por el nombre



CCITT - 29370

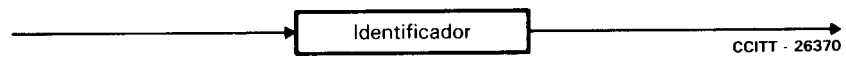
4.5 Nombre de parámetro



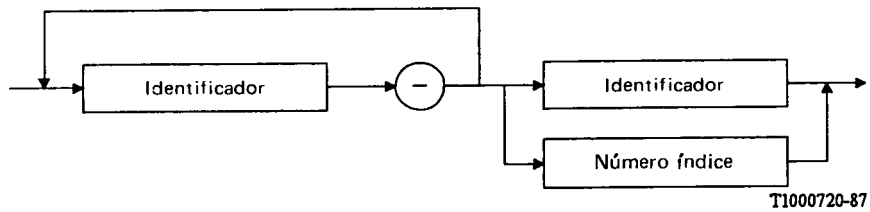
T1000710-87

- 1) Opcional.

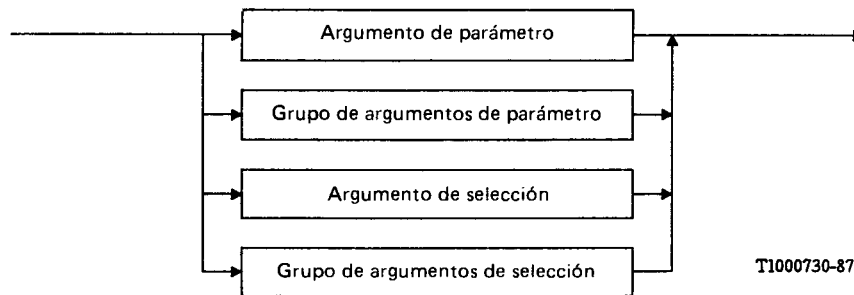
4.5.1 *Nombre de parámetro simple*



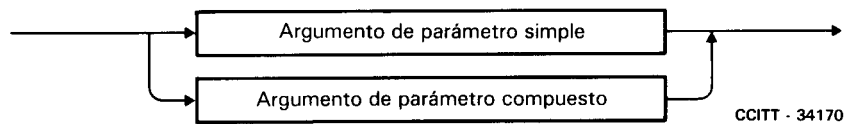
4.5.2 *Nombre de parámetro compuesto*



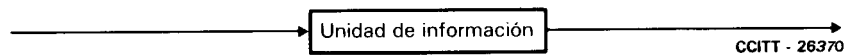
4.6 *Valor de parámetro*



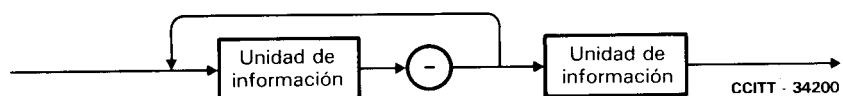
4.7 *Argumento de parámetro*



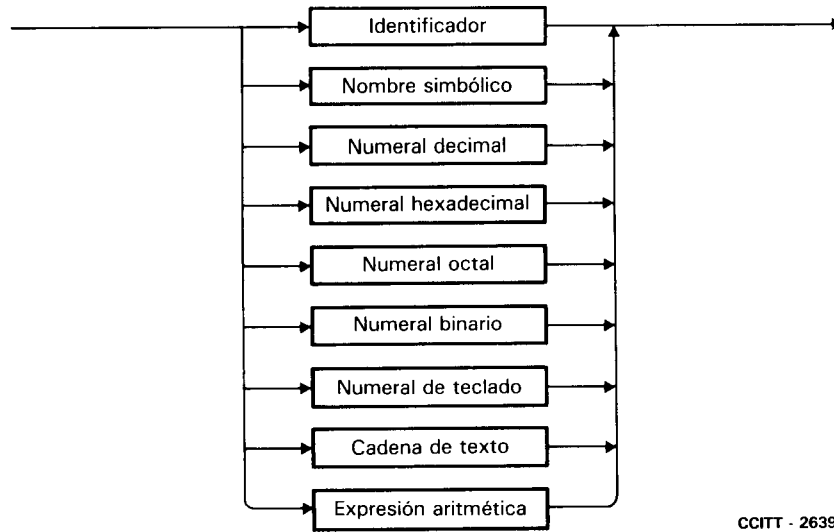
4.7.1 *Argumento de parámetro simple*



4.7.2 *Argumento de parámetro compuesto*



4.8 Unidad de información

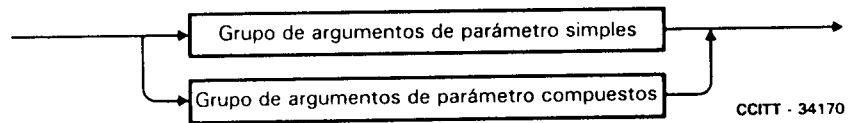


4.9 Agrupación de información

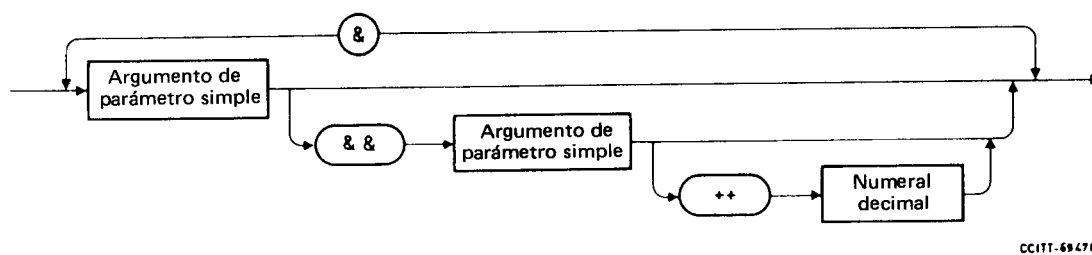
4.9.1 Grupo de bloques de parámetros

Véase el diagrama sintáctico del § 4.1.

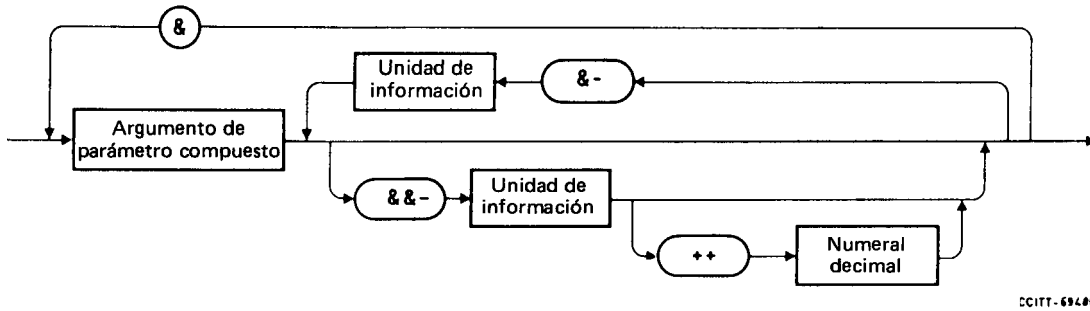
4.9.2 Grupo de argumentos de parámetro



4.9.2.1 Grupo de argumentos de parámetro simples



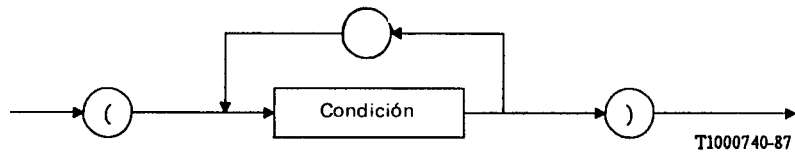
4.9.2.2 Grupo de argumentos de parámetro compuestos



CCITT-69400

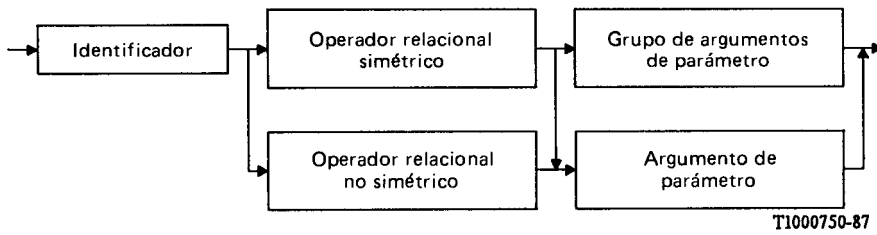
4.10 Búsquedas en bases de datos

4.10.1 Argumento de selección



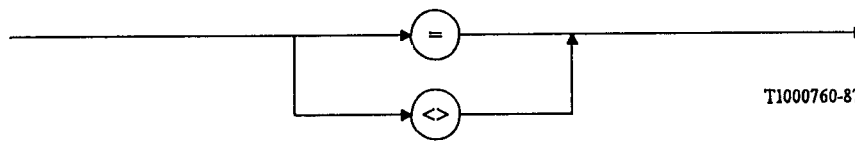
T1000740-87

4.10.1.1 Condición



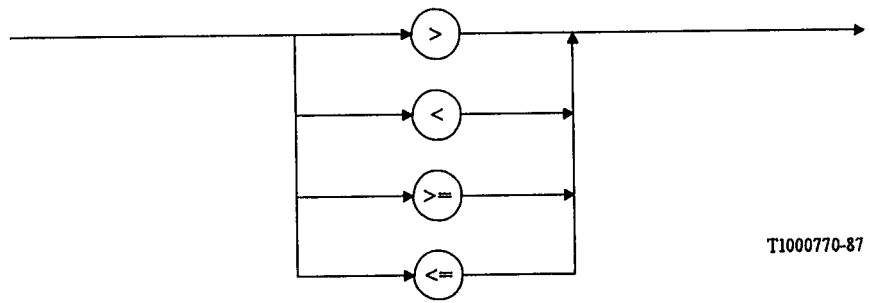
T1000750-87

4.10.1.2 Operador relacional simétrico



T1000760-87

4.10.1.3 Operador relacional no simétrico



4.10.2 Grupo de argumentos de selección

