



(19)



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 311 699**

(51) Int. Cl.:
G06F 17/30 (2006.01)

(12)

TRADUCCIÓN DE PATENTE EUROPEA

T3

(96) Número de solicitud europea: **03718905 .7**

(96) Fecha de presentación : **21.02.2003**

(97) Número de publicación de la solicitud: **1483693**

(97) Fecha de publicación de la solicitud: **08.12.2004**

(54) Título: **Representación informática de una estructura de datos arborescente y métodos de codificación/decodificación asociados.**

(30) Prioridad: **27.02.2002 FR 02 02664**

(73) Titular/es: **FRANCE TELECOM**
6, place d'Alleray
75015 Paris, FR

(45) Fecha de publicación de la mención BOPI:
16.02.2009

(72) Inventor/es: **Lassalle, Edmond**

(45) Fecha de la publicación del folleto de la patente:
16.02.2009

(74) Agente: **Justo Vázquez, Jorge Miguel de**

ES 2 311 699 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín europeo de patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre concesión de Patentes Europeas).

DESCRIPCIÓN

Representación informática de una estructura de datos arborescente y métodos de codificación/decodificación asociados.

La presente invención se refiere a una representación informática de un árbol orientado que representa la organización de un conjunto de datos, especialmente de un diccionario. La presente invención se refiere igualmente a un método de codificación de dicho árbol orientado en dicha representación informática. La presente invención se refiere también a un método de codificación de un dato perteneciente a dicho conjunto en un Índice de dicha representación informática. La presente invención se refiere finalmente a un método de decodificación que permite encontrar a partir de un Índice de dicha representación informática dicho dato correspondiente.

Previamente a la exposición del estado de la técnica, conviene definir un cierto número de términos que serán utilizados a continuación.

Se le llama gráfico orientado (denominado después simplemente gráfico) a una pareja $G=(S,A)$ donde S es un conjunto de vértices (denominados después igualmente nodos) y A es un subconjunto de $S \times S$, llamado conjunto de los arcos.

Un camino del gráfico es una serie ordenada $(s_0, s_1, \dots, s_n)$ de vértices tales que (s_{i-1}, s_i) , es un arco, para $i=1, \dots, n$, cuando $s_n=s_0$ con $n \geq 1$, el camino es llamado circuito o ciclo. Se dice que un gráfico está conectado si dos nodos cualquiera del mismo están unidos por un camino.

Un árbol está definido como un gráfico conectado sin circuito. Se puede mostrar que dos vértices cualquiera de un árbol están unidos por un camino único. Un árbol posee un vértice particular R , tal que todo vértice s distinto de R está unido a éste último por un camino. Este vértice particular se llama raíz del árbol.

Para un vértice S dado, se le llama descendiente de S a todo vértice s_d del árbol tal que exista un camino entre s y s_d . Recíprocamente, para un vértice s dado, se le llama ancestro de s a todo vértice S_a del árbol tal que exista un camino entre S_a y s . Se le llama vértice hijo de un vértice S a un descendiente s_f de s tal que $(S, s_f) \in A$. Para todo vértice s , del árbol se le llama subárbol resultante de s , al árbol de raíz s que incluye a todos los descendientes de s .

Finalmente se le llama hoja a todo vértice del árbol que no tenga descendiente.

Numerosos métodos de tratamiento de la información hacen referencia a una representación de los datos según una estructura arborescente (o árbol), especialmente los métodos de clasificación, de comprensión o de almacenamiento de la información.

El documento Chang H: "Bubble structure and operation to facilitate free traversal", IBM Technical disclosure Bulletin, vol-26, no.9, divulga un material burbuja y un algoritmo que exigen al menos dos tablas para representar un árbol en una memoria.

Según el tipo de aplicación considerado, los datos pueden ser cadenas de caracteres, series de fonemas, formas de ondas, patrones de luminancia/crominancia etc.

Sin pérdida de generalidad, se considera a continuación que los datos están constituidos por cadenas de entidades elementales o caracteres (por ejemplo letras, símbolos, cifras, signos alfanuméricos). El conjunto de estos posibles caracteres constituye un alfabeto. Se supone que este alfabeto está provisto de una relación de orden total, llamado orden alfabético.

En numerosas aplicaciones de tipo "motores de búsqueda", "búsqueda en un directorio", "búsqueda en un diccionario" etc., un volumen muy importante de datos debe poder ser almacenado y accedido, lo cual impone severas restricciones en condiciones de explotación en tiempo real, especialmente para los accesos en línea.

Cada dato debe poder ser accesible rápidamente sin necesidad de grandes recursos de cálculo. También, para reducir los tiempos de acceso, los volúmenes importantes de datos deben residir en la memoria central. A fin de no aumentar desmesuradamente el tamaño de esta memoria, frecuentemente es necesario efectuar una compresión previa de los datos. Ventajosamente, los datos deben poder ser accedidas sin ser descomprimidas, lo que todavía penalizaría los tiempos de acceso.

En los tipos de aplicación considerados mas arriba, se avizora un tratamiento asimétrico de los datos: la fase de compresión de los datos puede incluir un tratamiento relativamente largo y complejo aunque lo que permite su acceso y recuperación debe ser simple y rápida. Los datos pueden ser de esta forma almacenados en la memoria bajo una forma comprimida y fijada, siendo realizada su actualización "off-line" antes de que sean puestos en línea.

Existe una estructura de organización de los datos que se presta particularmente bien para la compresión: aquella del árbol definido más arriba. Se encuentra especialmente en los diccionarios o los directorios. Un diccionario en el sentido común de la palabra es un fichero de datos (llamados igualmente entradas), cada dato estando constituido por una cadena de caracteres alfabéticos, estando organizados estos últimos según una estructura arborescente.

En la práctica, en una representación informática, cada dato del diccionario está asociado a un Índice. La búsqueda de una cadena de caracteres (o palabra) en el diccionario conlleva a identificar el Índice de la palabra correspondiente. De esta forma un texto puede ser representado por una serie de Índices, más adaptado al tratamiento informático que la representación inicial.

Diferentes tipos de representación fueron propuestos en el estado de la técnica y especialmente:

- en forma de tabla de dicotomía, utilizando una compresión de Ziv-Lempel

- en forma de tabla de fragmentos

- en forma de árbol lexical

Estos diferentes tipos de representación conducen a desempeños equivalentes en caso de acceso perfecto. Se le llama acceso perfecto a un modo de acceso donde se busca la cadena de caracteres exacta en el diccionario correspondiente a la palabra a analizar, sin tener en cuenta los errores o las alteraciones.

La representación por árbol lexical supone un análisis (o “parsing”) de las cadenas de caracteres. Un ejemplo de árbol lexical es dado en la Fig. 1 por el diccionario Δ siguiente:

$\Delta = \{\text{abolish, abolition, appeal, attorney, bar, barrister, bench, case, court, crime}\}$

Se observa que en el árbol lexical, los arcos están asociados a los caracteres de las palabras del diccionario. Más precisamente, a cada arco del árbol está asociada una etiqueta, cada etiqueta teniendo como marca un carácter. El árbol lexical es la reunión de todos los caminos donde el esqueleto corresponde a una palabra del diccionario. Se le llama esqueleto de un camino a la cadena de los caracteres de las etiquetas de los arcos que constituyen ese camino. Una palabra del diccionario es también llamada “entrada” del diccionario.

Se notará que las hojas del árbol lexical han sido representadas por círculos aunque los otros vértices han sido representados por discos. La raíz del árbol ha sido indicada como R .

Se supone que el árbol está indizado, lo que quiere decir que a cada vértice está asociado un Índice. Una operación elemental en un árbol lexical es buscar a partir de una palabra dada, el Índice de la entrada del diccionario correspondiente. Esta operación necesita recorrer el árbol según los arcos etiquetados por los caracteres sucesivos que componen la palabra.

Particularmente, el algoritmo de búsqueda pone en marcha una función *Analizar Palabra* que arroja como valor el Índice (Índice-asociado(s)) de la entrada del diccionario si ésta última está presente, o, por defecto, un código de no identificación (Índice-Palabra-Desconocida). Ésta se muestra a continuación en pseudo-código C:

Función NúmeroÍndice *AnalizarPalabra* (cadena *PalabraAnálisis*, vértice *Raíz*)

Inicio

Vértice $s = \text{Raíz};$

Para cada *Character* de *PalabraAnálisis* **Hacer**

Si fin-de-palabra (*PalabraAnálisis*) y s es una hoja

Entonces regresar Índice-asociado(s);

Si *Character* corresponde a una etiqueta de un arco descendiente de s

Entonces $s = \text{descendiente-correspondiente}(s);$

Si no regresar Índice-Palabra-esconocida;

Fin

La navegación por las instrucciones $s=\text{descendiente-correspondiente}(s)$ supone que se dispone de una representación informática del árbol lexical.

De manera general, es necesario disponer de una representación informática de un árbol para poder recorrerlo, utilizarlo y modificarlo fácilmente.

Según, una primera representación informática conocida, un árbol está representado por un tabla de adyacencia $M=(m_{ij})$ $i=0, \dots, n$; $j=0, \dots, n$ almacenado en la memoria con $m_{ij}=1$ si $(s_i, s_j) \in A$.

Según una representación informática mas corriente, un árbol está representado como una serie de punteros informáticos. Según una primera variante conocida ilustrada en la Fig. 2A, cada nodo está representado por un valor (o Índice) y un tabla de punteros apuntando hacia sus nodos hijos. El tamaño del tabla corresponde al número máximo de hijos (k) que puede tener un nodo del árbol (se dice en este caso el árbol "k-aire"). La Fig. 2A muestra un ejemplo de representación de un árbol 3-aire según esta variante.

La codificación de los hijos de un nodo a través de un tabla de punteros presenta el inconveniente de consumir demasiado espacio de memoria cuando el árbol contiene un pequeño número de nodos que tienen muchos hijos y otros numerosos nodos que tienen pocos hijos. Según una segunda variante conocida de representación, se remedia esta dificultad utilizando para un nodo dado un puntero hacia uno de sus nodos hijos, llamado hijo mayor, y un puntero del hijo mayor hacia una lista encadenada de sus hermanos. La Fig. 2B muestra un ejemplo de representación según esta segunda variante, para un árbol 5-aire.

Los punteros permiten modificar rápidamente la estructura del árbol pero son relativamente costosos en espacio en memoria. Además, la detección de una relación de descendencia entre nodos no es inmediata. Se supone que determine el camino que une a los dos nodos, los que, en el marco de una representación por punteros, implican importantes recursos de cálculo. Se puede reducir considerablemente la cantidad de cálculos a efectuarse memorizando el cierre transitivo del árbol, en el caso de que un nodo apunte hacia cada uno de sus descendientes. Sin embargo, ésta última opción es particularmente voraz en espacio de memoria.

El problema base de la invención es proponer una representación informática de un árbol que ocupe poco espacio de memoria, que permita recorrerlo fácilmente y modificarlo simplemente.

Ese problema es resuelto por la representación informática de un árbol representativo de la organización de un conjunto de datos, en particular de un diccionario de datos, cada dato estando asociado a un nodo particular de dicho árbol, incluyendo esta representación una tabla de valores almacenados en una memoria, dichos valores siendo representativos de los rangos de los nodos de dicho árbol ordenados según una primera relación de orden total, las direcciones a las cuales dichos valores son almacenados siendo representativos de los rangos de los nodos de dicho árbol ordenados según una segunda relación de orden total.

Ventajosamente, la primera relación de orden total es una combinación de una relación de orden de descendencia ordenando un nodo con respecto a sus descendientes y una relación de orden de primogenitura ordenando los nodos hijos de un mismo nodo.

Según un primer modo de realización, un primer nodo del árbol es inferior a un segundo nodo del árbol según dicha primera relación de orden total si el segundo nodo es un descendiente del primer nodo o si, el ancestro común del primer y segundo nodo tiene un primer hijo del cual desciende el primer nodo o confundido con éste último y un segundo hijo del cual desciende el segundo nodo o confundido con éste último, dicho primer hijo es inferior a dicho segundo hijo según la relación de orden de primogenitura.

Según un segundo modo de realización, un primer nodo del árbol es superior a un segundo nodo del árbol según dicha primera relación de orden total si el segundo nodo es un descendiente de primer nodo o si, el ancestro común del primer y segundo nodo tenga un primer hijo del cual desciende el primer nodo o confundido con éste último y un segundo hijo del cual desciende el segundo nodo o confundido con éste último, dicho primer hijo es inferior a dicho segundo hijo según la relación de orden de primogenitura.

Ventajosamente, la segunda relación de orden total es una combinación de la relación de orden inversa de dicha relación de orden de descendencia y de dicha relación de orden de primogenitura.

Según una primera variante, un primer nodo del árbol es inferior a un segundo nodo del árbol según dicha segunda relación de orden total si el primer nodo es un descendiente del segundo nodo o si, el ancestro común del primer y segundo nodo tiene un primer hijo del cual desciende el primer nodo o confundido con éste último y un segundo hijo del cual desciende el segundo nodo o confundido con éste último, dicho primer hijo es inferior a dicho segundo hijo según la relación de orden de primogenitura.

Según una segunda variante, un primer nodo del árbol es superior a un segundo nodo del árbol según dicha segunda relación de orden total si el primer nodo es un descendiente del segundo nodo o si, el ancestro común del primer y segundo nodo tenga un primer hijo del cual desciende el primer nodo o confundido con éste último y un segundo hijo del cual desciende el segundo nodo o confundido con éste último, dicho primer hijo es inferior a dicho segundo hijo según la relación de orden de primogenitura.

Si los datos son series de caracteres de un alfabeto provisto de un orden alfabético, cada arco de dicho árbol estando asociado a un carácter de al menos un dato, la relación de orden de primogenitura entre dos hijos de un mismo nodo puede estar dada por la relación de orden alfabético entre los caracteres asociados a los arcos respectivos entre dicho nodo y sus dos hijos.

La invención se refiere igualmente a un método de codificación de un árbol orientado representativo de la organización de un conjunto de datos, especialmente de un diccionario, cada dato de dicho conjunto estando asociado a un nodo particular de dicho árbol, en el cual se atribuye a cada nodo de dicho árbol un primer y segundo Índice, el primer Índice siendo representativo del rango del nodo según una primera relación de orden total ordenando los nodos de dicho árbol, el segundo Índice siendo representativo del rango del nodo según una segunda relación de orden total, la primera relación de orden total siendo una combinación de una relación de orden de descendencia ordenando un nodo con respecto a sus descendientes y una relación de orden de primogenitura ordenando los nodos hijos de un mismo nodo, la segunda relación de orden total siendo una combinación de la relación de orden inversa de dicha relación de orden de descendencia y de dicha relación de orden de primogenitura.

Ventajosamente el método de codificación incluye una referencia recurrente de una etapa de cálculo que brinda para un nodo cualquiera del árbol, el tamaño del subárbol resultante de dicho nodo.

Para un primer y segundo hijo de un mismo nodo, llamado nodo padre, adyacentes en una lista de hijos ordenada según dicha relación de orden de primogenitura, la etapa de cálculo determina el primer Índice del segundo hijo a partir del primer Índice del primer hijo y del tamaño del subárbol resultante del primer hijo, y el segundo Índice del segundo hijo a partir del segundo Índice del primer hijo y del tamaño del subárbol resultante del segundo hijo.

Dicha etapa de cálculo determina el primer Índice del hijo clasificado primero en dicha lista a partir del primer Índice de dicho nodo padre y el segundo Índice de dicho nodo padre a partir del segundo Índice del hijo clasificado último en dicha lista.

Dicha etapa de cálculo determina igualmente el tamaño del subárbol resultante de dicho nodo padre a partir de la suma de los tamaños de los subárboles resultantes de sus hijos.

Ventajosamente, dicho método de codificación opera sobre una primera representación de dicho árbol por medio de punteros en la cual, para un nodo dado, un primer tipo de puntero brinda un nodo hijo según la relación de orden de descendencia y un segundo tipo de puntero brinda la lista de sus otros hijos.

La invención es igualmente definida por un método de codificación de un dato de entrada perteneciente a un conjunto de datos organizados según una estructura de árbol orientado, especialmente de un diccionario de datos, estando formados los datos por series de caracteres de un alfabeto provisto de un orden alfabético, cada dato estando asociado a un nodo particular de dicho árbol y a cada arco de dicho árbol estando asociado un carácter, en el cual dicho árbol está representado por medio de la representación informática ya mencionada, se recorre el árbol de nodo en nodo según un camino partiendo de la raíz y se analiza dicho dato de entrada carácter por carácter, el nodo siguiente de un nodo corriente de dicho camino escogiéndose a través de los hijos de éste último, efectuándose la elección por medio de una sucesión de etapas de comparación, cada etapa de comparación comparando el carácter en curso de dicho dato de entrada y el carácter asociado al arco que une al nodo corriente con uno de sus hijos, siendo interrumpido el recorrido sólo cuando dicho dato de entrada fue enteramente analizada, brindando el método como valor codificado de dicho dato de entrada un Índice función de la dirección de la tabla de dicha representación informática representativa del último nodo de dicho camino.

La invención también es definida por un método de decodificación de un Índice representativo de un dato perteneciente a un conjunto de datos organizados según una estructura de árbol orientado, en particular de un diccionario de datos, los datos siendo formados por series de caracteres de un alfabeto provisto de un orden alfabético, cada dato estando asociado a un nodo particular de dicho árbol y a cada arco de dicho árbol estando asociado un carácter, en el cual dicho árbol está representado por medio de la representación informática ya mencionada, se recorre el árbol según un camino que parte desde la raíz, el nodo siguiente de un nodo corriente de dicho camino escogiéndose a través de los hijos de éste último, efectuándose la elección por medio de una sucesión de etapas de comparación, cada etapa de comparación comparando dicho Índice a un Índice representativo de uno de dichos hijos en dicha representación informática, brindando el método como dato decodificado la cadena de caracteres asociados a los arcos que forman dicho camino.

Las características de la invención mencionadas más arriba, de igual forma que otras, aparecerán mas claramente tras la lectura de la descripción siguiente de ciertos modelos de realización, dicha descripción hecha en relación con los dibujos adjuntos, entre los cuales:

La Fig. 1 representa un árbol lexical;

La Fig. 2A muestra una primera representación informática de un árbol con ayuda de punteros;

La Fig. 2B muestra una segunda representación informática de un árbol con ayuda de punteros;

La Fig. 3A ilustra con un ejemplo un método de codificación de árbol según un primer modo de realización de la invención;

La Fig. 3B muestra una primera variante de representación informática del árbol de la Fig. 3A;

La Fig. 3C muestra una segunda variante de representación informática del árbol de la Fig. 3A;

La Fig. 4 ilustra una porción de árbol antes de la indización por rango prefijo y rango posfijo.

La idea en la base de la invención es crear una nueva representación informática de un árbol a partir de una nueva relación de orden total traduciendo las relaciones de dependencia entre nodos.

La relación de dependencia entre nodos induce una relación de orden parcial sobre el conjunto de nodos del árbol. Por ejemplo, si se convenía para dos nodos s_1 y s_2 del árbol que: $s_1 > s_2$ si y sólo si s_2 es un descendiente de s_1 , se dispone de una relación de orden. Sin embargo este orden no es más que un orden parcial ya que todos los nodos del árbol no pueden ser comparados de esta forma (por ejemplo los hijos de un mismo nodo).

Se puede construir una relación de orden total sobre los nodos de un árbol si se sabe ordenar todos los hijos de un mismo nodo. El orden según el cual se ordenan los hijos de un mismo nodo será convencionalmente llamado orden de primogenitura. Para un árbol lexical, donde las etiquetas de los arcos contienen los caracteres alfabéticos, se puede conveniar que dos hijos s_1 y s_2 de un mismo nodo S satisfacen a la relación $s_1 > s_2$ si el carácter de la etiqueta asociada al arco (S, s_1) es precedido por aquel de la etiqueta asociada al arco (S, s_2) . De otro modo el orden alfabético de las marcas de las etiquetas induce un orden de primogenitura sobre los nodos hijos de un mismo nodo.

La combinación de la relación de orden parcial de descendencia (señalada a continuación $\overset{>}{D}$ o $\overset{<}{D}$) y de la relación de orden de primogenitura (señalada a continuación $\overset{>}{P}$ o $\overset{<}{P}$) permite obtener una relación de orden total sobre el conjunto de los nodos. Esta combinación puede ser realizada de varias maneras diferentes:

- relación de orden prefijo (señalada convencionalmente $\overset{<}{pref}$):

$$a \overset{<}{pref} b \text{ si } a \overset{<}{D} b \text{ o } a' \overset{<}{P} b'$$

donde a' y b' son los hijos del ancestro común de a y b , tales que a desciende o es confundida con a' y b desciende o es confundida con b' .

Dicho de otra forma, el nodo a es inferior al nodo b en el sentido del orden prefijo si b es un descendiente de a o a' es un hermano mayor de b' .

- relación de orden prefijo inverso (señalada convencionalmente $\overset{<}{fresp}$):

$$b \overset{<}{fresp} a \text{ si } a \overset{<}{D} b \text{ o } a' \overset{<}{P} b'$$

- relación de orden posfijo inverso (señalada convencionalmente $\overset{<}{post}$):

$$a \overset{<}{post} b \text{ si } a \overset{<}{D} b \text{ o } a' \overset{<}{P} b'$$

dicho de otra forma el nodo a es inferior al nodo b en el sentido del orden posfijo si a es un descendiente de b o a' es un hermano mayor de b' .

- relación de orden posfijo inverso (señalada convencionalmente $\overset{<}{tsop}$):

$$b \overset{<}{tsop} a \text{ si } a \overset{<}{D} b \text{ o } a' \overset{<}{P} b'$$

ES 2 311 699 T3

Siendo dado que dos nodos cualquiera de un árbol son, ya sean descendientes el uno del otro, ya sea descendientes de un ancestro común, las relaciones de orden definidas más arriba son relaciones de orden total.

Las relaciones de orden $\overset{<}{pref}$ (o $\overset{<}{frep}$) y $\overset{<}{post}$ (o $\overset{<}{tsop}$) permiten pues ordenar enteramente el conjunto S de los nodos del árbol. Dicho de otra forma, a cada una de las relaciones de orden se le puede asociar una función “rango” de S en $[0, n]$, por ejemplo:

RangoPrefijo: $S \rightarrow [0, n]$

tal que $s_1 \overset{<}{pref} s_2$ si y sólo si $RangoPrefijo(s_1) < RangoPrefijo(s_2)$

RangoPosfijo: $S \rightarrow [0, n]$

tal que $s_1 \overset{<}{post} s_2$ si y sólo si $RangoPosfijo(s_1) < RangoPosfijo(s_2)$.

$RangoPrefijo$ y $RangoPosfijo$ son morfismos de conjuntos ordenados. De la misma manera se puede definir las funciones rango $RangoPrefijoInverso$ y $RangoPosfijoInverso$ a partir de relaciones de orden $\overset{<}{frep}$ y $\overset{<}{tsop}$:

RangoPrefijoInverso: $S \rightarrow [0, n]$

tal que $s_1 \overset{<}{frep} s_2$ si $RangoPrefijoInverso(s_1) < RangoPrefijoInverso(s_2)$

RangoPosfijoInverso: $S \rightarrow [0, n]$

tal que $s_1 \overset{<}{tsop} s_2$ si y sólo si $RangoPosfijoInverso(s_1) < RangoPosfijoInverso(s_2)$.

Según un primer modo de realización, se utiliza, para construir la representación informática, una biyección T de $[0, n]$ en $[0, n]$ definida por:

$T: [0, n] \rightarrow [0, n]$

$T = RangoPosfijo$ o $RangoPrefijo^{-1}$.

Según una variante de este primer modo de realización, se utiliza la biyección T^1 .

De la misma manera, se podrá utilizar las biyecciones formadas por composición: $RangoPosfijoInverso$ o $RangoPrefijo^{-1}$, $RangoPosfijo$ o $RangoPrefijoInverso^{-1}$ ó $RangoPosfijoInverso$ o $RangoPrefijoInverso^{-1}$ en otros modos de realización de la invención o bien, en las variantes de éstos, las inversas de sus biyecciones.

Con objetivo de simplificar, limitaremos la exposición de la invención a la utilización de las biyecciones T y T^1 , a fin de que las otras biyecciones puedan ser igualmente utilizadas.

Dicha biyección T puede ser representada de manera informática bajo la forma de un primer tabla de valores almacenados en memoria, el rango posfijo de un nodo siendo almacenado en una dirección representativa del rango prefijo de ese nodo.

Del mismo modo, la biyección T^1 puede ser representada de manera informática bajo la forma de un segundo tabla de valores almacenados en la memoria, el rango posfijo de un nodo siendo almacenado en una dirección representativa del rango prefijo de ese nodo.

Un ejemplo hará comprender mejor el interés y la utilización de esas biyecciones.

La Fig. 3A ilustra un árbol donde los nodos han sido indizados por los rangos prefijo (en negrita y subrayado) y los rangos posfijo (en cursiva). La relación de orden de primogenitura, por ejemplo inducida por un orden alfabético

sobre las marcas de las etiquetas en el caso de un árbol lexical, ha sido representada convencionalmente en el sentido creciente de izquierda hacia derecha. A cada nodo s está de esta forma asociado una pareja:

$$(RangoPrefijo(s), RangoPosfijo(s))$$

Estas parejas son ventajosamente almacenadas en una tabla por medio de la biyección T (Fig. 3B) o T^1 (Fig. 3C). En Fig. 3B los valores de rango posfijo han sido almacenados en las direcciones dadas por los valores de rango prefijo correspondientes. A la inversa, en la Fig. 3C los valores de rango prefijo han sido almacenados en las direcciones dadas por los valores de rango posfijo correspondientes.

Una primera ventaja de la representación informática del árbol según la invención es que no ocupa un espacio de memoria del tamaño del árbol ($n+1$) al contrario de una representación clásica por punteros (Figs. 2A y 2B) que necesita al menos el doble de espacio de memoria.

Una segunda ventaja esencial de esta representación informática es que permite detectar muy simplemente una relación de dependencia entre dos nodos del árbol. En efecto, para determinar si un nodo s_2 es dependiente de un nodo s_1 , basta con comparar $RangoPrefijo(s_1)$ a $RangoPrefijo(s_2)$ de una parte, y $RangoPosfijo(s_1)$ a $RangoPosfijo(s_2)$ de otra parte:

s_2 es dependiente de s_1 si y sólo si se tiene:

$$RangoPrefijo(s_2) > RangoPrefijo(s_1) \quad y \quad RangoPosfijo(s_2) < RangoPosfijo(s_1)$$

De esta forma, en el ejemplo de la Fig. 3A, se verifica que el nodo representado por la pareja ($RangoPrefijo, RangoPosfijo$) = (5, 1) es bien dependiente de aquel representado por la pareja ($RangoPrefijo, RangoPosfijo$) = (1, 5) pero no de aquel representado por la pareja ($RangoPrefijo, RangoPosfijo$) = (22, 21).

De la misma manera, gracias a la tabla de la Fig. 3B, es muy fácil determinar los descendientes o los ancestros de un nodo dado. Por ejemplo, para determinar la lista de los descendientes del nodo (8, 12), basta con barrer la tabla en el sentido de las direcciones crecientes a partir de la dirección 8 y de buscar entre los datos almacenados aquellos que son inferiores al posfijo 12 (aquí 6, 10, 11, 7, 8, 9). Estos valores indican el rango posfijo de los descendientes del nodo en cuestión. Para determinar la lista de los ancestros del nodo (8, 12), basta con barrer la tabla en el sentido de las direcciones decrecientes a partir de la dirección 8 y de buscar entre los datos almacenados aquellos que son inferiores al posfijo 12 (aquí 19, 22). Estos valores indican los rangos posfijo de los ancestros del nodo en cuestión.

Se procede de manera dual a partir de la tabla de la Fig. 3C. Tomando el ejemplo anterior, basta, para determinar la lista de los descendientes, con borrar la tabla en el sentido de las direcciones decrecientes a partir de la dirección 12 y de buscar entre los datos almacenados aquellos que son superiores al prefijo 8 (aquí 14, 10, 13, 12, 11, 9). Esos valores indican los rangos prefijo de los descendientes del nodo en cuestión. Del mismo modo, para determinar la lista de los ancestros de (8, 12), basta con barrer la tabla en el sentido de las direcciones crecientes a partir de la dirección 12 y de buscar entre los datos almacenados aquellos que son inferiores al prefijo 8 (aquí 7, 0).

Una tercera ventaja de la representación informática según la invención es que permite un fácil recorrido del árbol, si es de la raíz hacia las hojas, recorrido que se efectúa por ejemplo cuando se analiza una cadena de caracteres (palabra) con la ayuda de un árbol lexical, si es de las hojas hacia la raíz, recorrido que se efectúa por ejemplo cuando se genera una cadena de caracteres a partir del Índice de un nodo.

El recorrido del árbol de la raíz hacia las hojas supone que se sepa determinar los hijos de un nodo dado. Como se verá, la tabla de la Fig. 3B (o aquella de la Fig. 3C) permite encontrarlos rápidamente.

Supongamos que el algoritmo de navegación busca los hijos del nodo (12, 8) y consideremos la tabla de la Fig. 3B. La tabla es barrida en el sentido de las direcciones crecientes a partir de la dirección 8. Como anteriormente, se busca los datos de la tabla inferiores a 12. Cuando se retiene un dato x inferior a 12, no se consideran los datos siguientes que son inferiores a x . Dicho de otra forma, se continua barriendo la tabla hasta que se encuentre de nuevo un dato x' superior a x (pero siempre inferior al valor inicial 12). Se itera el proceso hasta el fin de la tabla. De esta forma en el presente ejemplo, se encuentra ante todo el valor 6 que está retenido ($6 < 12$) después el valor 10 que está igualmente retenido ($6 < 10 < 12$). Los valores siguientes 7, 8, 9, no son retenidos ya que, si son bien inferiores a 12, no son superiores al último valor retenido 10. El valor 11 es retenido a continuación ($10 < 11 < 12$) pero no los valores siguientes ya que son superiores a 12.

Se procede de manera dual a partir de la tabla de la Fig. 3C. Tomando el ejemplo anterior, se barre la tabla en el sentido de las direcciones decrecientes a partir de la dirección 12. Como anteriormente, se buscan los datos almacenados superiores al prefijo 8. Cuando se encuentra un dato x superior a 8, no se consideran más los datos siguientes que son superiores a x . Dicho de otra forma, se barre la tabla hasta que se encuentra de nuevo un dato y

inferior a $\underline{x}$ (pero siempre superior al valor inicial $\underline{8}$). Se itera el proceso hasta que se alcanza el principio de la tabla. De esta forma en el presente ejemplo, se encuentra ante todo el valor 14 que está retenido (>8) y después el valor 10 que está retenido igualmente ($8 < 10 < 14$). Los valores siguientes $\underline{11}$, $\underline{12}$, $\underline{13}$ no están retenidos ya que, si son bien superiores a $\underline{8}$, no son inferiores al último valor retenido $\underline{10}$. El valor $\underline{9}$ es retenido a continuación ($8 < 9 < 10$) pero no los valores siguientes ya que son superiores a $\underline{8}$.

Esta manera de determinar los hijos de un nodo dado puede convenir para los árboles de débil tamaño. Sin embargo, como se verá más adelante, para los árboles de gran tamaño, es más rápido calcular directamente los rangos prefijo/posfijo de dichos hijos.

Se notará que, si se había empleado, en lugar de tablas construidos a partir de biyecciones T y T^l , tablas construidos respectivamente a partir de biyecciones $RangoPosfijoInverso$ o $RangoPrefijo^{-1}$, $RangoPosfijo$ o $RangoPoefijoInverso^{-1}$ o $RangoPosfijoInverso$ o $RangoPoefijoInverso^{-1}$ o bien a partir de las inversas de sus biyecciones, se hubiera podido determinar de manera similar los hijos de un nodo dado, al precio eventual de un cambio de sentido de barrido y/o de cambio de sentido de las desigualdades.

De la misma forma, el recorrido de un árbol desde sus hojas hacia la raíz supone que se sepa determinar el padre de un nodo dado. Suponiendo que el algoritmo de navegación busca el padre de un nodo (12, 8) y considerando ante todo el tabla de la Fig. 3B. Se barre el tabla en el sentido de las direcciones decrecientes a partir de la dirección $\underline{8}$. El primer dato encontrado superior a 12 da el indicio posfijo del padre del nodo en cuestión (aquí 19).

Entendido esto, se procede de manera dual a partir de la tabla de la Fig. 3C. En ese caso se barre la tabla en el sentido de las direcciones crecientes a partir de la dirección 12. El primer dato encontrado inferior a $\underline{8}$ da el indicio prefijo del padre del nodo en cuestión (aquí 7).

Para transformar un árbol cualquiera en su representación informática, operación llamada codificación del árbol, se debe ante todo proceder a una indización de los nodos. Para codificar el árbol bajo la forma de una representación informática según la invención, se debe indizar los nodos por medio de las funciones $RangoPrefijo$ y $RangoPosfijo$ (u otras funciones equivalentes vistas más arriba). Sin pérdida de generalidad, limitaremos la exposición del método de indización según la invención a dos funciones precitadas.

El método de indización opera sobre una representación informática clásica del árbol por medio de punteros como, por ejemplo, aquella ilustrada en la Fig. 2B. Esta representación informática clásica será obtenida de manera conocida a partir de un fichero de las entradas del diccionario. La cadena vertical de los punteros corresponde a una relación prefija de las entradas. La cadena horizontal de los hermanos de un mismo nodo se hace según el orden de primogenitura, tal que aquel hereda una clasificación de las marcas de las etiquetas.

Se inicializa el rango prefijo de la raíz en 0 y se recorre el árbol a partir de la raíz siguiendo los punteros del hijo mayor hasta que se alcanza una hoja (la más a la izquierda según la convención de representación aquí escogida). El rango posfijo de esta hoja se inicializa en 0.

Sea ahora un nodo S del árbol teniendo como hijos $s_0, s_1, \dots, s_p$ ordenados según el orden creciente de primogenitura (quiere decir ordenadas según la cadena horizontal), como se ilustra en la Fig. 4. Se tienen las relaciones siguientes:

$$RangoPrefijo(s_0) = RangoPrefijo(S) + 1$$

$$RangoPrefijo(s_{i+1}) = RangoPrefijo(s_i) + \Gamma(s_i)$$

$$RangoPosfijo(s_{i+1}) = RangoPosfijo(s_i) + \Gamma(s_{i+1})$$

$$RangoPosfijo(S) = RangoPosfijo(s_p) + 1$$

$$\Gamma(S) = \sum_{i=0}^p \Gamma(s_i) + 1$$

donde $\Gamma(s)$ es el tamaño del subárbol resultante de s .

La indización por rango prefijo y rango posfijo puede ser efectuada en un solo paso, a partir de la raíz del árbol, utilizando la referencia recurrente de una función que arroja como valor, para un vértice s dado, el tamaño $\Gamma(s)$ del subárbol que resulta. Esta función se muestra a continuación en pseudo-código C:

```

5  Función tamaño CodificaciónArbol (vértice  $S$ , rango Prefijo,
   rango Posfijo, tabla Biyección)
10 Inicio
    TamañoSubArbolesDeHijos=0;
    Si  $S$  es una hoja Entonces {
15         Biyección [Prefijo]=Posfijo;
        Retornar 1;
20     }
    Si no para todos los descendientes  $s$  de  $S$  Hacer {
        TamañoSubArbol=CodificaciónArbol ( $s$ ,
25             Prefijo+1,
                Posfijo+TamañoSubArbolesDeHijos,
30             Biyección) ;
        Prefijo+=TamañoSubArbol;
        TamañoSubArbolesDeHijos+= TamañoSubArbol;
35         Biyección[Prefijo]= Posfijo+TamañoSubArbolesDeHijos1;
    Retornar TamañoSubArbolesDeHijos+1;
40 Fin

```

Se notará en el programa de más arriba que la variable *TamañoSubArbol* es el tamaño del subárbol resultante del nodo hijo(s) corriente y que la variable *TamañoSubArbolesDeHijos* es el valor acumulado de los tamaños de los subárboles resultantes de los nodos hijos ya explorados.

La función *CodificaciónArbol* crea directamente una representación informática bajo la forma de una tabla de Índice del tipo de la Fig. 3B, almacenado en la memoria. Una vez creada esta representación informática, la representación inicial por punteros, al hacerse inútil, es suprimida de la memoria.

A continuación, nos ubicaremos, por razones de simplificación, en el contexto de un diccionario organizado según un árbol lexical, donde cada entrada del diccionario corresponde a una hoja del árbol. Como se acaba de ver, una representación informática bajo forma de tabla del mismo tamaño que la del árbol puede ser obtenida por medio del método de codificación según la invención.

Esta representación informática es ventajosamente utilizada para buscar, a partir de una cadena de caracteres dada, el Índice de la entrada del diccionario correspondiente. Se tomará convencionalmente como Índice el rango prefijo de la hoja (alternativamente se puede escoger su rango prefijo inverso). La búsqueda del Índice hace referencia a un primer método de recorrido del árbol desde la raíz hacia las hojas según la invención.

Recíprocamente, esta representación informática es ventajosamente utilizada para generar, a partir del Índice de una entrada del diccionario, la cadena de caracteres correspondiente. La generación de la cadena de caracteres hace referencia a un segundo método de recorrido de árbol desde la raíz hacia las hojas según la invención.

Se considera en primer lugar el caso de una búsqueda del Índice de una cadena de caracteres C . El árbol es recorrido a partir de la raíz siguiendo los arcos donde las etiquetas tienen los caracteres sucesivos de C .

Ventajosamente, el primer método de recorrido de árbol desde la raíz hasta las hojas según la invención opera de la siguiente manera.

Durante la llegada al primer hijo s_0 de un nodo S , se inicializa:

$$\mathbf{RangoPrefijo}(s_0) = \mathbf{RangoPrefijo}(S) + 1$$

y se calcula el tamaño $\Gamma(s_0)$ resultante de s_0 a partir de:

$$\mathbf{\Gamma}(s_0) = \mathbf{RangoPosfijo}(s_0) - \mathbf{UltimoPosfijo}$$

donde *UltimoPosfijo* es el rango posfijo del último nodo donde se abandonó el recorrido del subárbol (en otros términos, el rango posfijo de la raíz del último subárbol expuesto en el recorrido). A modo de ilustración, si en la Fig. 3A el nodo de rango posfijo 5 no ha sido retenido porque el arco que une la raíz y ese nodo no tiene el carácter buscado, el subárbol resultante de ese nodo no es recorrido y se tiene como *UltimoPosfijo* = 5. Se continua el recorrido por el nodo del rango posfijo 19 y, de ser exitoso, por aquel de rango posfijo 12. El tamaño del subárbol resultante de ese nodo (primer hijo s_0 del nodo S de rango posfijo 19) es efectivamente 7.

Se determinan a continuación los rangos prefijos de los hijos sucesivos s_i de S por medio de las relaciones de recurrencia siguientes:

$$\mathbf{RangoPrefijo}(s_{i+1}) = \mathbf{RangoPrefijo}(s_i) + \mathbf{\Gamma}(s_i)$$

$$\mathbf{\Gamma}(s_{i+1}) = \mathbf{RangoPosfijo}(s_{i+1}) - \mathbf{RangoPosfijo}(s_i)$$

A cada hijo explorado s_i , se le prueba si el carácter en curso c de C es igual al carácter que tiene la etiqueta del arco que une S a s_i . Si ese no es el caso se prosigue con la exploración del hijo siguiente s_{i+1} y así sucesivamente hasta que se encuentre el carácter en curso o se exploren todos los hijos de S .

Cuando se llega a un nodo S dado, no se conoce *a priori* el nombre de sus hijos. Para hacer esto, ventajosamente se memoriza durante la exploración del nodo S el tamaño $\Gamma(S)$ del subárbol que resulta. A continuación, durante la exploración sucesiva de los hijos s_i , se pone al día la variable *TamañoSubArbolesDeHijos*.

$$\mathbf{TamañoSubArbolesDeHijos} = \sum_{j=0}^i \mathbf{\Gamma}(s_j)$$

Y se sabe que todos los hijos s_i de S fueron explorados cuando:

$$\mathbf{TamañoSubArbolesDeHijos} = \mathbf{\Gamma}(S) - 1$$

Si todos los hijos han sido explorados sin que se haya encontrado el carácter en curso, la cadena de carácter completa C no corresponde a una entrada del diccionario (al menos una porción de C puede formar parte). Si el carácter en curso fue encontrado por uno de los hijos s_i entonces $S=s_i$ y el ciclo de búsqueda reinicia con el carácter siguiente. El proceso es iterado hasta que se alcanza una hoja del árbol. El Índice buscado es el rango prefijo de esta hoja. Ventajosamente, a fin de poder buscar las palabras que son prefijos una de otra, como por ejemplo “bar” y “barrister” en la Fig. 1, se puede agregar, al final de cada palabra un marcador de fin de palabra, por ejemplo un carácter espacio. En este caso, todas las hojas del árbol tienen marcadores de fin de palabra.

Se puede definir de esta forma una función *RecorridoArbol* que arroja a partir de una cadena de caracteres *PalabraAnálisis*, el rango prefijo de la hoja alcanzada al término del recorrido. Esta función utiliza la representación informática del árbol según la invención. Su pseudo-código se muestra a continuación:

Función ÍndicePrefijo RecorridoArbol (cadena PalabraAnálisis, tabla Biyección)

Inicio

$S = \text{raíz};$

$\text{Prefijo}(S) = 0;$ // Índice prefijo de la raíz

$\text{TamañoSubArbol}(S) = \text{Biyección}[\text{Prefijo}(S)] + 1;$

$\text{UltimoPosfijo} = -1;$

Tanto que $\text{TamañoSubArbol}(S)$ diferente de 1{

$s = \text{hijo mayor de } S;$

$\text{Prefijo}(s) = \text{Prefijo}(S) + 1;$

$\text{TamañoSubArbolesDeHijos} = 0;$

Tanto que $\text{TamañoSubArbolesDeHijos} < \text{TamañoSubArbol}(S) - 1;$

Si CaracterEnCurso corresponde etiqueta de S hacia s

Entonces dirigir el recorrido hacia el hijo s haciendo

$S = s;$

Si no { $s_2 = \text{hijo siguiente de } S;$

$\text{Posfijo}(s) = \text{Biyección}[\text{Prefijo}(S)];$

$\text{TamañoSubArbol}(s) = \text{Posfijo}(s) - \text{UltimoPosfijo};$

$\text{Prefijo}(s_2) = \text{Prefijo}(s) + \text{TamañoSubArbol}(s);$

$\text{UltimoPosfijo} = \text{Posfijo}(s);$

$\text{TamañoSubArbolesDeHijos} += \text{TamañoSubArbol}(s);$

Explorar el descendiente siguiente poniendo

$s = s_2$

}

Retornar CódigoRecorridoIncompleto

}

Retornar $\text{Prefijo}(S);$

Recíprocamente, si se desea generar a partir de un Índice/(supuesto igual al rango prefijo de una hoja del árbol) la cadena de caracteres de la entrada correspondiente del diccionario, se utiliza un segundo método de recorrido según la invención. Este segundo método difiere del primero en que la selección del nodo hijo es a partir de ahora guiada por la comparación del rango prefijo de ese nodo con el Índice/buscado. Más precisamente, el hijo s_t es seleccionado desde que la relación siguiente es verificada:

$\text{RangoPrefijo}(s_{t+1}) > 1$

El Índice/es de esta forma aproximado por valores crecientes de rango prefijo de los nodos explorados.

El segundo método hace referencia al mismo cálculo iterativo de los rangos prefijos de los hijos de un nodo S dado a partir de los tamaños respectivos de los subárboles $\Gamma(s_i)$. El criterio de parada de la exploración de los hijos s_i de un nodo S dado está igualmente basado en una comparación de $\Gamma(S)$ y de la suma de las $\Gamma(s_i)$ para los hijos ya explorados.

Se puede definir una función *GeneraciónPalabra* que arroja a partir de un Índice *GuíaPrefijo* la cadena de caracteres correspondientes. Esta función utiliza igualmente la representación informática del árbol según la invención. Su pseudo-código C se muestra a continuación:

```

5  Función cadena GeneraciónPalabra (Índice GuíaPrefijo, tabla
   Biyección)
   Inicio
10      S = raíz;
      Prefijo (S) = 0; // Índice prefijo de la raíz
      TamañoSubÁrbol(S)= Biyección [Prefijo(S)] + 1;
15      UltimoPosfijo = -1;

      Tanto que TamañoSubÁrbol(S) diferente de 1{
20          s = hijo mayor de S;
          Prefijo(s) = Prefijo(S) + 1;
          TamañoSubArbolesDeHijos = 0;
25

          Tanto que TamañoSubArbolesDeHijos<TamañoSubÁrbol(S)-1{
              s2 = hijo siguiente S;
30              Posfijo(s) = Biyección [Prefijo(S)];
              TamañoSubÁrbol(s) = Posfijo(s) - UltimoPosfijo;
              Prefijo(s2) = Prefijo(s) + TamañoSubÁrbol(s);
35

          Si GuíaPrefijo < Prefijo(s2);
40          Entonces {
              Escoger el camino hacia el hijo precedente s haciendo
              S=s;
45          Poner al día CadenaGenerada
              }

          Si no {
50              UltimoPosfijo = Posfijo(s);
              TamañoSubArbolesDeHijos += TamañoSubÁrbol(s);
              explorar el descendiente siguiente poniendo
              s = s2
              }
60          }

          Retornar CódigoRecorridoIncompleto
        }
65      Retornar CadenaGenerada;

```

REIVINDICACIONES

1. Representación informática de un árbol orientado representativo de la organización de un conjunto de datos, en particular de un diccionario de datos, cada dato estando asociado a un nodo particular de dicho árbol, **caracterizada** porque un primer rango está asociado a cada nodo de dicho árbol según una primera relación de orden total y un segundo rango está asociado a cada nodo de dicho árbol según una segunda relación de orden total, la misma comprendiendo una tabla de valores almacenados en una memoria, tal que el primer rango de un nodo está representado por un valor que está almacenado en la dirección de la tabla representativa de segundo rango de ese nodo.

2. Representación informática según la reivindicación 1, **caracterizada** porque la primera relación de orden total es una combinación de una relación de orden de descendencia ordenando un nodo con respecto a sus descendientes, y de una relación de orden de primogenitura ordenando los nodos hijos de un mismo nodo.

3. Representación informática según la reivindicación 2, **caracterizada** porque un primer nodo del árbol es inferior a un segundo nodo del árbol según dicha primera relación de orden total si el segundo nodo es un descendiente del primer nodo o si, el ancestro común del primer y segundo nodo tiene un primer hijo del cual desciende el primer nodo o confundido con éste último y un segundo hijo del cual desciende el segundo nodo o confundido con éste último, dicho primer hijo es inferior a dicho segundo hijo según la relación de orden de primogenitura.

4. Representación informática según la reivindicación 2, **caracterizada** porque un primer nodo del árbol es superior a un segundo nodo del árbol según dicha primera relación de orden total si el segundo nodo es un descendiente del primer nodo o si, el ancestro común del primer y segundo nodo tiene un primer hijo del cual desciende el primer nodo o confundido con éste último y un segundo hijo del cual desciende el segundo nodo o confundido con éste último, dicho primer hijo es inferior a dicho segundo hijo según la relación de orden de primogenitura.

5. Representación informática según una de las reivindicaciones de 2 a 4, **caracterizada** porque la segunda relación de orden total es una combinación de la relación de orden inversa de dicha relación de orden de descendencia y dicha relación de orden de primogenitura.

6. Representación informática según la reivindicación 5, **caracterizada** porque un primer nodo del árbol es inferior a un segundo nodo del árbol según dicha segunda relación de orden total si el segundo nodo es un descendiente del primer nodo o si, el ancestro común del primer y segundo nodo tiene un primer hijo del cual desciende el primer nodo o confundido con éste último y un segundo hijo del cual desciende el segundo nodo o confundido con éste último, dicho primer hijo es inferior a dicho segundo hijo según la relación de orden de primogenitura.

7. Representación informática según la reivindicación 5, **caracterizada** porque un primer nodo del árbol es superior a un segundo nodo del árbol según dicha segunda relación de orden total si el segundo nodo es un descendiente del primer nodo o si, el ancestro común del primer y segundo nodo tiene un primer hijo del cual desciende el primer nodo o confundido con éste último y un segundo hijo del cual desciende el segundo nodo o confundido con éste último, dicho primer hijo es inferior a dicho segundo hijo según la relación de orden de primogenitura.

8. Representación informática según una de las reivindicaciones de 2 a 7, **caracterizada** porque, los datos son series de caracteres de un alfabeto provisto de un orden alfabético, cada arco de dicho árbol estando asociado a un carácter de al menos un dato, la relación de orden de primogenitura entre dos hijos de un mismo nodo está dada por la relación de orden alfabético de los caracteres asociados a los arcos respectivos entre dicho nodo y sus dos hijos.

9. Método de codificación de un árbol orientado representativo de la organización de un conjunto de datos, especialmente de un diccionario, cada dato de dicho conjunto estando asociado a un nodo particular de dicho árbol, **caracterizado** porque se le atribuye a cada nodo de dicho árbol un primer y segundo Índice, el primer Índice siendo representativo del rango del nodo según una primera relación de orden total que ordena los nodos de dicho árbol, el segundo Índice siendo representativo del rango del nodo según una segunda relación de orden total, la primera relación de orden total siendo una combinación de una relación de orden de descendencia que ordena un nodo con respecto a sus descendientes y de una relación de orden de primogenitura que ordena los nodos hijos de un mismo nodo, la segunda relación de orden total siendo una combinación de la relación de orden inversa de dicha relación de orden de descendencia y dicha relación de orden de primogenitura, dicho método brindando una tabla en el cual son ordenados los valores representativos de los dichos primeros Índices de los nodos de dicho árbol en direcciones representativas de dichos segundos Índices de los nodos de dicho árbol.

10. Método de codificación según la reivindicación 9, **caracterizado** porque comprende una llamada recurrente de una etapa de cálculo que brinda para un nodo cualquiera del árbol, el tamaño del subárbol resultante de dicho nodo.

11. Método de codificación según la reivindicación 10, **caracterizado** porque, para un primer y segundo hijo de un mismo nodo, llamado nodo padre, adyacentes en una lista de hijos ordenados según dicha relación de orden de primogenitura, la etapa de cálculo determina el primer Índice del segundo hijo a partir del primer Índice del primer hijo y del tamaño del subárbol descendiente del primer hijo, y el segundo Índice del segundo hijo a partir del segundo Índice del primer hijo y del tamaño del subárbol del segundo hijo.

12. Método de codificación según la reivindicación 11, **caracterizado** porque dicha etapa de cálculo determina el primer Índice del hijo clasificado primero en dicha lista a partir del primer Índice de dicho nodo padre y el segundo Índice de dicho nodo padre a partir del segundo Índice del hijo clasificado último en dicha lista.

5 13. Método de codificación según una de las reivindicaciones de 10 a 12, **caracterizado** porque dicha etapa de cálculo determina el tamaño del subárbol descendiente de dicho nodo padre a partir de la suma de los tamaños de los subárboles resultantes de sus hijos.

10 14. Método de codificación según una de las reivindicaciones de 9 a 13, **caracterizado** porque opera sobre una primera representación de dicho árbol por medio de punteros en la cual, para un nodo dado, un primer tipo de puntero brinda un nodo hijo según la relación de orden de descendencia y un segundo tipo de puntero brinda la lista de sus otros hijos.

15 15. Método de codificación según una de las reivindicaciones de 9 a 13, **caracterizado** porque brinda un tabla en la cual son ordenados los valores representativos de los dichos segundos Índices de los nodos de dicho árbol en direcciones representativas de dichos primeros Índices de los nodos de dicho árbol.

16. Método de codificación de un dato de entrada perteneciente a un conjunto de datos organizadas según una estructura de árbol orientado, especialmente de un diccionario de datos, estando formadas los datos por series de caracteres de un alfabeto provisto de un orden alfabético, cada dato estando asociado a un nodo particular de dicho árbol y a cada arco de dicho árbol estando asociado un carácter, **caracterizado** porque, dicho árbol es representado por medio de la representación informática según una de las reivindicaciones de 1 a 8, se recorre el árbol de nodo en nodo según un camino que parte desde la raíz y se analiza dicho dato de entrada carácter por carácter, el nodo siguiente de un nodo corriente de dicho camino siendo escogido a través de los hijos de éste último, la elección siendo efectuada por medio de una sucesión de etapas de comparación, cada etapa de comparación comparando el carácter en curso de dicho dato de entrada y el carácter asociado al arco que une el nodo corriente con uno de sus hijos, siendo interrumpido el recorrido sólo cuando dicho dato de entrada fue enteramente analizado, el método proporcionando como valor codificado de dicho dato de entrada un Índice función de la dirección de la tabla de dicha representación informática representativa del último nodo de dicho camino.

30 17. Método de codificación según la reivindicación 16, **caracterizado** porque dicho Índice es igual a la dirección representativa del último nodo de dicho camino.

35 18. Método de codificación según la reivindicación 16, **caracterizado** porque dicho Índice es igual al valor almacenado en dicha tabla en la dirección representativa del último nodo de dicho camino.

19. Método de codificación según una de las reivindicaciones de 16 a 18, **caracterizado** porque los hijos sucesivos del nodo corriente son determinados a partir de sus respectivas direcciones representativas en la tabla de dicha representación informática, obteniéndose la dirección representativa del hijo siguiente a un hijo corriente a partir de la dirección representativa del hijo corriente y del tamaño del subárbol resultante del hijo corriente.

40 20. Método de codificación según la reivindicación 19, **caracterizado** porque el tamaño del subárbol resultante del hijo corriente es obtenido a partir del valor almacenado en dicha tabla en la dirección representativa del hijo corriente y del valor almacenado en dicha tabla en la dirección representativa del hijo precedente.

45 21. Método de decodificación de un Índice representativo de un dato perteneciente a un conjunto de datos organizados según una estructura de árbol orientado, en particular de un diccionario de datos, formándose los datos por series de caracteres de un alfabeto provisto de un orden alfabético, cada dato estando asociada a un nodo particular de dicho árbol y a cada arco de dicho árbol estando asociado un carácter, **caracterizado** porque, dicho árbol es representado por medio de la representación informática según una de las reivindicaciones de 1 a 8, el árbol siendo recorrido según un camino que parte desde la raíz, el nodo siguiente de un nodo corriente de dicho camino siendo escogido a través de los hijos de éste último, efectuándose la elección por medio de una sucesión de etapas de comparación, cada etapa de comparación comparando dicho Índice a un Índice representativo de uno de dichos hijos en dicha representación informática, el método proporcionando como dato decodificado la cadena de caracteres asociada a los arcos que forman dicho camino.

22. Método de codificación según la reivindicación 21, **caracterizado** porque dicho Índice representativo es una dirección en dicha tabla de la representación informática.

60 23. Método de codificación según la reivindicación 21, **caracterizado** porque dicho Índice representativo es un valor almacenado en dicha tabla de la representación informática.

24. Método de codificación según una de las reivindicaciones 21 a 23, **caracterizado** porque los hijos sucesivos del nodo corriente son determinados a partir de sus respectivas direcciones representativas en la tabla de dicha representación informática, obteniéndose la dirección representativa del hijo siguiente a un hijo corriente a partir de la dirección representativa del hijo corriente y del tamaño del subárbol resultante del hijo corriente.

ES 2 311 699 T3

25. Método de codificación según la reivindicación 24, **caracterizado** porque el tamaño del subárbol resultante del hijo corriente es obtenido a partir del valor almacenado en dicha tabla en la dirección representativa del hijo corriente y del valor almacenado en dicha tabla en la dirección representativa del hijo precedente.

5

10

15

20

25

30

35

40

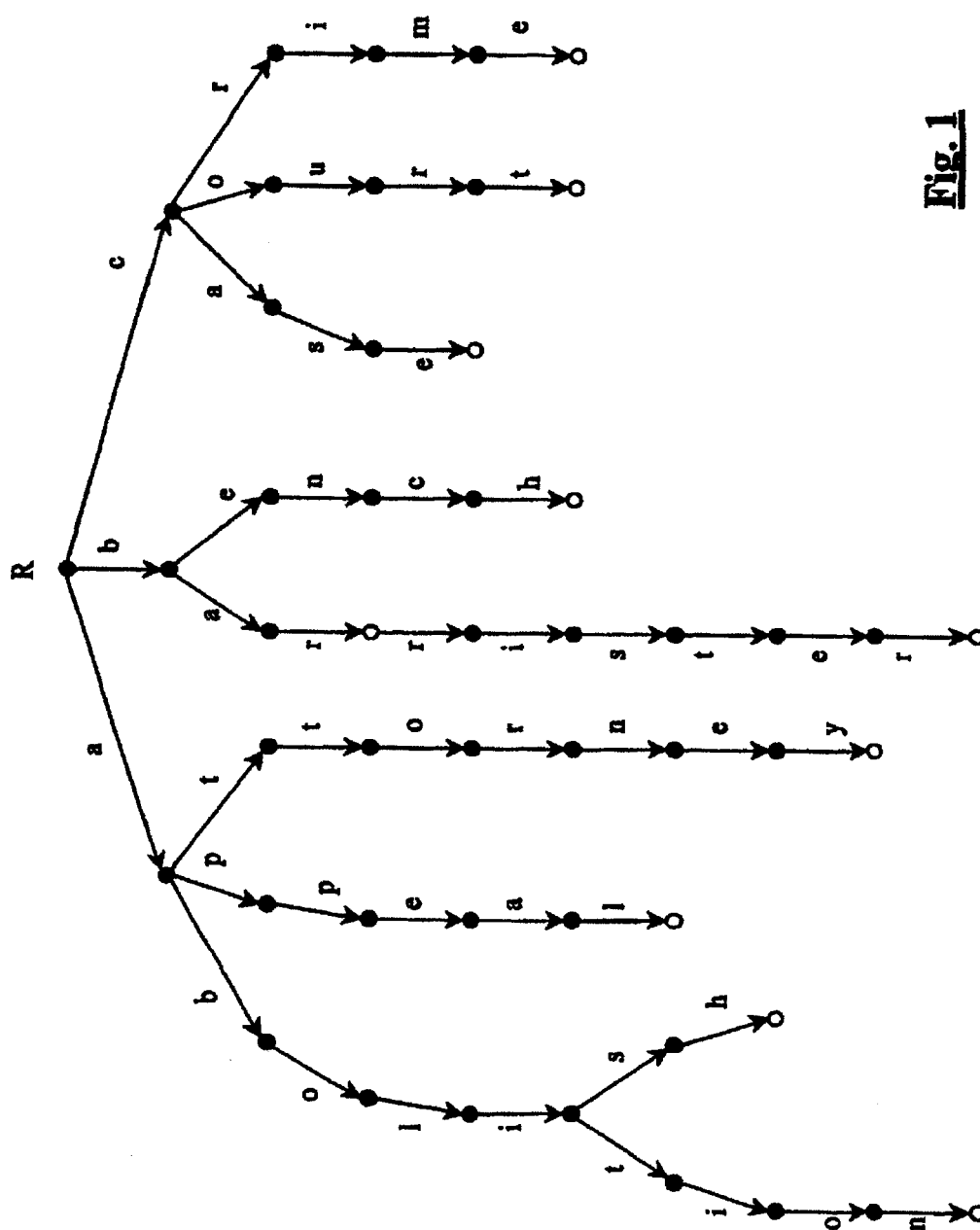
45

50

55

60

65



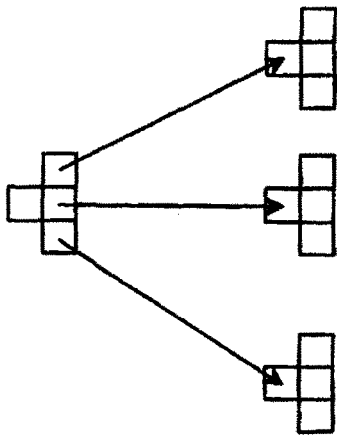


Fig. 2A

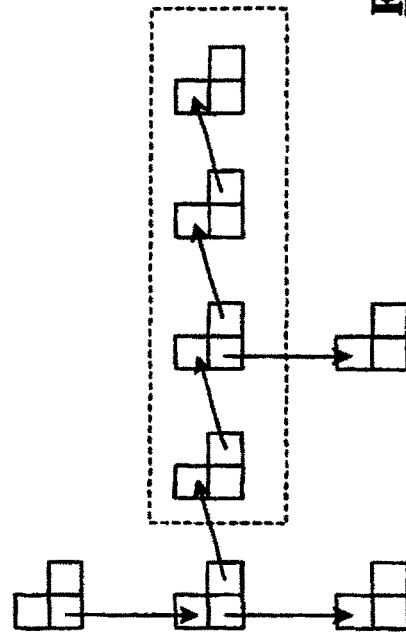
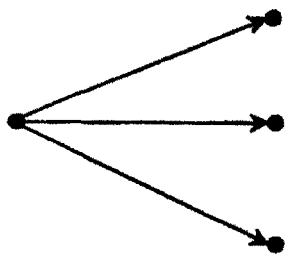
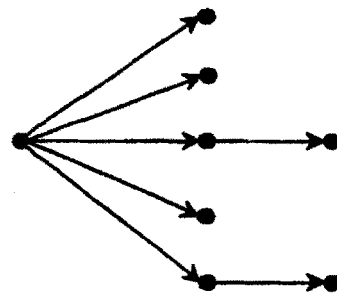


Fig. 2B



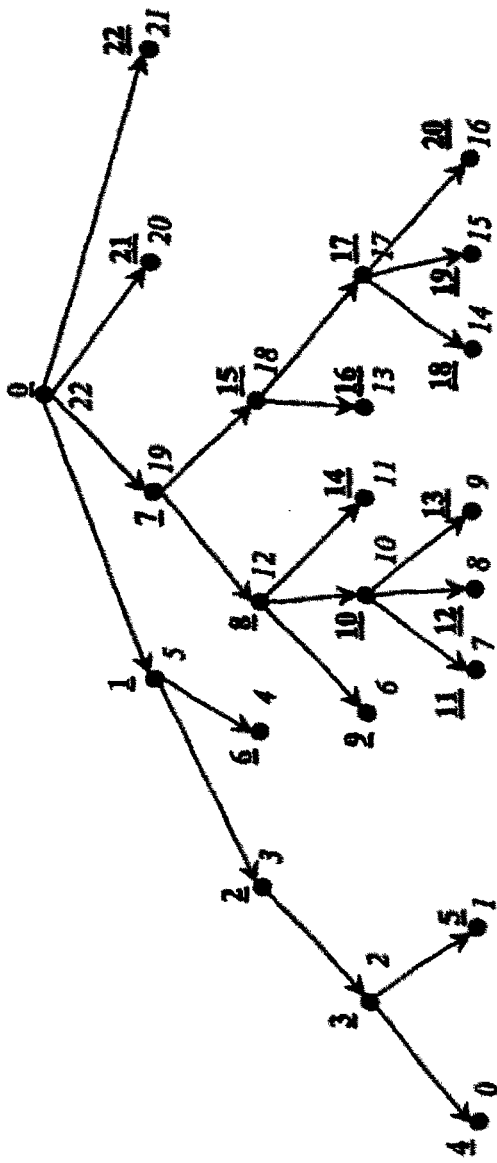


Fig. 3A

Fig. 3B

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
22	5	3	2	0	1	4	19	12	6	10	7	8	9	11	18	13	17	14	15	16	20	21

T

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
4	5	3	2	6	1	2	11	12	13	10	14	8	16	18	18	20	17	15	7	21	22	0

T^{-1}

Fig. 3C

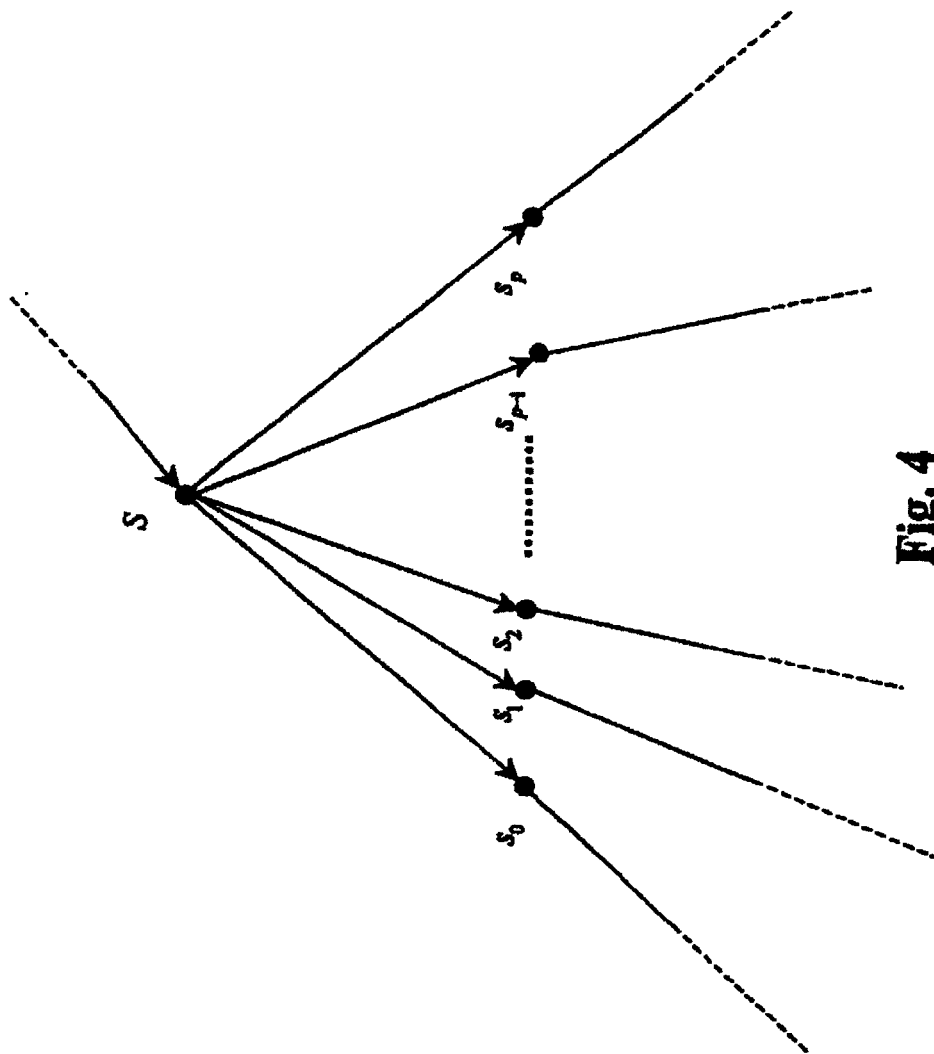


Fig. 4