

# RECUPERACION DEDUCTIVA DE DATOS

*Dr. Rogelio Silverio Castro, Lic. María Elena Martínez Busto, Lic. Antonio Toledo Dorrego\**

## INTRODUCCION

El surgimiento de nuevas aplicaciones en áreas tan diversas como los sistemas de manufactura, multimedia, CAD - CAM, etc., ha traído como consecuencia que los sistemas de bases de datos orientados a registros sean insuficientes para la modelación y recuperación de información en tales sistemas.

Los sistemas de Bases de Datos Deductivas, llamados también Base de Conocimientos, generalizan los sistemas tradicionales de bases de datos, particularmente los relacionales.

Una base de datos deductiva está compuesta de dos tipos de información: los hechos y las reglas. Los hechos representan la información almacenada explícitamente y constituyen la parte extensional de la base de datos; mientras que las reglas deductivas permiten derivar información a partir del conjunto de hechos y constituyen la parte intencional de la base de datos.

Una de las variantes más utilizadas para crear las facilidades deductivas en un sistema de gestión de base de datos es la creación y acoplamiento de un módulo que permite realizar las consultas deductivas.

Debido a que la diferencia entre un programa lógico y el conjunto de sentencias de una Base de Datos Deductiva es desde el punto de vista cuantitativo, o sea proporción entre

hechos y reglas, es por lo que en las BDD se utilizan los avances alcanzados en la programación lógica y por ello en este trabajo se utilizó para la creación de la base de conocimientos un dialecto del Prolog implementado en Borland Delphi 2.0.

## DESARROLLO

### ***Generalidades sobre Bases de Datos Deductivas***

Las investigaciones sobre Bases de Datos Deductivas (BDAS) comenzaron justamente con el libro de Gallaire y Minker (1978) «Logic and Database», aunque apenas a mediados de los años 80, las investigaciones en este campo, comenzaron a enfocarse sobre los resultados de la optimización, lo cual es esencial si se desea que tales sistemas se conviertan en una perspectiva realista. Estos sistemas se basan en la lógica de predicados como un modelo de datos, frecuentemente siguiendo un estilo Prolog, utilizando cláusulas de Horn o reglas IF - THEN para definir los predicados o relaciones.

El enfoque deductivo en los sistemas de bases de datos es motivado por el surgimiento de nuevas aplicaciones de las tecnologías de bases de datos; las primeras generaciones de sistemas de bases de datos surgieron como una respuesta a una serie de importantes necesidades, orientadas a registros tales como sistemas de reservaciones de aerolíneas o bancarias. Muchas de las aplicaciones que surgen no

---

\* Docentes investigadores de la Universidad Central de Las Villas, Cuba.

pueden ser satisfechas por los sistemas primarios, entre las que se encuentran el diseño y las aplicaciones de ingeniería concurrente, integración de base de datos heterogéneas e independientes y aplicaciones que involucran datos de multimedia. Con frecuencia los datos de tales aplicaciones tienen una estructura más compleja que la que se encuentra en los sistemas orientados a registros, con lo que surge la necesidad de resolver aplicaciones más complejas con niveles múltiples de abstracción. Podemos citar algunas de las características que invariablemente sugieren la utilización de Sistemas Deductivos; son capaces de manipular gran volumen de información en un tiempo de acceso y de respuesta pequeño, además de utilizar la diferenciación de datos, quedando claro que en un sistema orientado a registros sería realmente imposible dar respuesta a estas nuevas aplicaciones que requieren de una captura, almacenamiento y manipulación óptimas.

Los sistemas de bases de datos deductivas llamados también bases de conocimientos, generalizan los sistemas tradicionales de bases de datos, particularmente los sistemas relacionales. Una base de datos deductiva se compone de dos tipos de información: los hechos y las reglas. Los hechos representan la información almacenada explícitamente y constituyen la parte extensional de la base de datos; las reglas deductivas permiten derivar información a partir del conjunto de hechos y constituyen la parte intensional de la base de datos.

Mientras los sistemas tradicionales eran construidos sobre expresiones lógicas de primer orden como lenguaje de consulta, los sistemas de bases de datos deductivas usan colecciones de expresiones de primer orden que son usualmente escritas como cláusulas

de Horn o reglas IF - THEN y tienen una semántica basada en puntos fijos, o en ideas más complicadas para colecciones no ambiguas o reglas sin un único punto fijo mínimo, tales como estratificación (Van Gelder -1986-, Apt, Blair y Schiipf -1991-, Naqvi -1986-) o semántica mejor fundada (Van Gelder, Ross y Schlpf -1991-, Ross -1991-).

A continuación veremos un ejemplo clásico, el problema de la «misma generación, el cual ilustra el estilo tipo Prolog para la definición de los predicados o relaciones en un Sistema de bases de datos deductivas.

- (1) igualgeneración(X Y): - bandera(X,Y)
- (2) igualgeneración(X,Y): padrede(X,Xpadre)& igualgeneración(Xpadre,Ypadre)& hijode(Ypadre,Y).

En este ejemplo podemos encontrar tres relaciones predicados: *bandera*, *padrede* e *hijode*. Asumiremos que están almacenados físicamente en la base de datos o sea que son predicados de la base de datos extensional BDE; el cuarto predicado *igualgeneración* es definido implícitamente por las reglas o sea predicados de la base de datos intencional, BDI.

La regla (1) dice que los individuos X y Y son de la misma generación si ellos están relacionados por el predicado *bandera*, la regla (2) dice que X y Y son de la misma generación si podemos encontrar los individuos Xpadre y Ypadre, que sean a su vez los respectivos padres de **X** y **Y**. De esta forma **X** y **Xpadre** son relacionados por el predicado *padrede*, y de forma similar **Y** y **Ypadre** son relacionados por el predicado *hijode*; **Xpadre** y **Ypadre** son relativos a la misma generación.

La combinación de ambas reglas es un programa lógico. El menor punto fijo es la

relación para el predicado *igualgeneración* como una función de las relaciones de la base de datos extensional (las relaciones :bandera, padre e hijode).

Una característica distintiva de los sistemas relacionales y en particular de los sistemas deductivos es la habilidad de realizar preguntas ad-hoc. Estas son preguntas escritas de forma fácil y breve en un lenguaje de alto nivel.

Si unimos una relación  $R(A,B)$  con una relación  $S(B,C)$  producimos una nueva relación  $T(A,B,C)$ . Del mismo modo en lógica se puede escribir una regla como sigue:  
 $T(A,B,C) :- R(A,B) \& R(A,B)$ .

La nueva relación  $T$  puede incluso tener un nuevo tipo, ya que puede ser que nunca antes hubiéramos encontrado una relación con los atributos  $A$ ,  $B$  y  $C$ ; no obstante las operaciones del álgebra relacional están inmediatamente a la disposición para la relación  $T$ , porque ellas están definidas para cualquier esquema de relación avanzada. Es esta predefinición de operaciones lo que hace posible las preguntas ad-hoc en sistemas lógicos o relacionales. Con esto podemos construir rápidamente expresiones arbitrariamente complicadas sin tener que detenernos a crear un nuevo tipo y escribir un laborioso programa para cada operación que necesitemos.

### **Bases de Datos Deductivas. Definición intuitiva.**

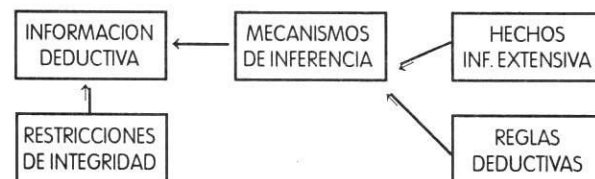
Una base de datos deductiva es una base de datos de la que se puede derivar información a partir de la que está almacenada explícitamente. Es decir, se incorpora una capacidad deductiva que permite, a partir de la información almacenada explícitamente y de unas

definiciones implícitas de información, realizar deducciones de nueva información.

Componentes de una BDD:

- Hechos: información que se almacena explícitamente.
- Reglas deductivas: permiten derivar una nueva información a partir de hechos. Estas reglas deductivas corresponden a definiciones implícitas de la información.
- Restricciones de Integridad: propiedades semánticas que la base de datos debe cumplir.

Veamos estos tres componentes en un esquema explicativo del funcionamiento de una base de datos deductiva.



### **Implementación de un Sistema de Bases de Datos Deductivas**

Se distinguen dos estrategias para la realización de un sistema de gestión de bases de datos deductivas. La primera de ellas es llamada «acoplamiento fuerte» siendo donde el sistema gestor tiene como parte de él un módulo de consultas deductivas, en cuyo caso para el usuario no hay diferencia entre el lenguaje de manipulación de datos y el lenguaje de consulta. La segunda estrategia es llamada «acoplamiento débil» y consiste en añadir al sistema de gestión de base de datos un módulo que permita realizar consultas deductivas, caso en el que el usuario distingue entre las dos interfaces.

Unit object-Prolog

Para la construcción de una BDD, como su nombre y definición implican, es necesaria

una herramienta basada en la lógica para realizar las inferencias sobre la base de conocimiento; en nuestro caso utilizamos la Unit Object-Prolog, la cual implementa la mayoría de las facilidades que posee el lenguaje de programación lógica Prolog.

### Generalidades

El Prolog es un lenguaje de programación para símbolos y resuelve principalmente problemas que involucran objetos y relaciones entre ellos, la información es almacenada en forma de hechos y reglas, las cuales se representan como cláusulas de Horn. Una cláusula de Horn es una **fb1** la cual puede aparecer como :

$A_1 \wedge A_2 \wedge \dots \wedge A_n \text{ o } , \text{ hecho}$   
 $A_1 \wedge A_2 \wedge \dots \wedge A_n \square B. , \text{ regla}$   
 donde:

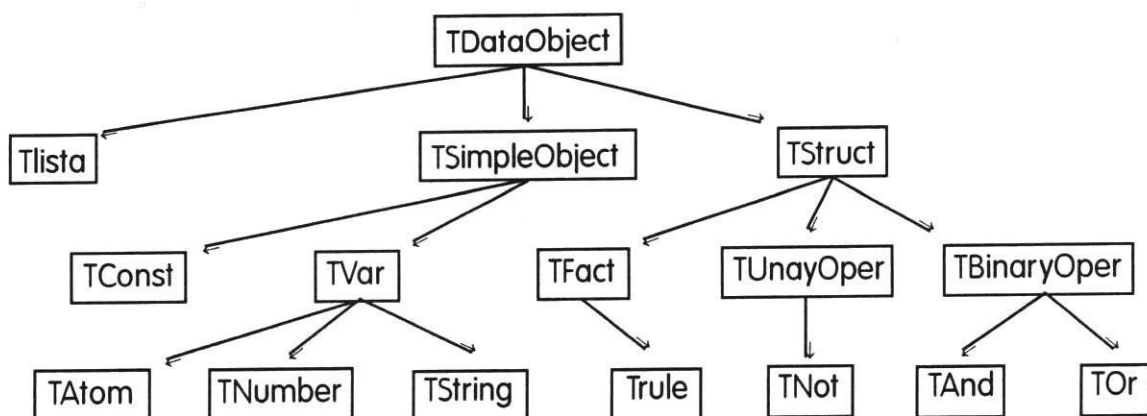
$A_i B_j$  son casos no negados de predicados en los cuales se realiza, fundamentalmente en forma recursiva, cosas que en el momento de definir logra expresar de forma clara, precisa y compacta los hechos y relaciones de un problema determinado; cuando se realizan las preguntas al Prolog, éstas son siempre una secuencia de uno o más hechos,

para responder una pregunta sólo trata de satisfacer todos los hechos. En este momento cabe hacerse la pregunta ¿qué es para Prolog satisfacer un hecho? Esto significa demostrar que el hecho es verdadero, demostrarlo siguiendo una secuencia lógica de hechos, asumiendo que las relaciones en este programa son verdaderas, es decir, demostrarlo siguiendo una secuencia lógica de hechos y reglas en un programa. Si la pregunta contiene variables, Prolog trata de encontrar los objetivos para los cuales los hechos son satisfechos, dándole instancia así a dichas variables. En caso de no poder hacerlo, automáticamente devolverá «NO».

El Object-Prolog mantuvo la filosofía central del Prolog de definir relaciones y preguntas sobre las relaciones; representar las relaciones en forma de cláusulas, las que pueden ser de tres tipos: hechos, reglas y preguntas; además se mantuvo el mismo principio de resolución.

### Estructura de los datos e implementación

El Object-Prolog posee una jerarquía de objetos en la que se contemplan la mayoría de los datos que el Prolog contiene:



En cada objeto existe una serie de métodos que permiten manipular de forma eficiente y semejante al Prolog dicha jerarquía de

datos, además de contar con otros objetos y sus respectivos componentes que hacen posible la construcción, manipulación e

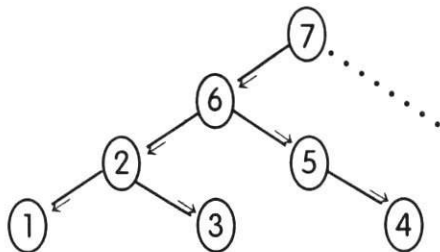
inferencia dentro de la base de conocimientos que se forma con los hechos que provienen de la base de datos y las nuevas reglas que se incorporan, unido a las restricciones de integridad. Entre estos objetos podemos citar los de manejo de listas, operadores (tanto relacionales como lógicos) y TKnowledgeBase que propiamente trabaja con la base de conocimientos proporcionando el dominio de consulta y las propias preguntas al sistema.

### ***Estrategia de búsqueda***

A partir de que existen varias estrategias para encontrar una solución, observamos dos fundamentales: el primero en profundidad y el primero a lo largo. Por su simplicidad a la hora de implementar se tomó el primero en profundidad, habiendo tomado en cuenta que era susceptible de formar ciclos. Pero existen dos métodos que lo prevén; son ellos: limitar la profundidad de búsqueda y el chequeo para evitar la repetición de nuevo, por lo que no se presentó ningún problema al implementar el Object-Prolog al seguir esta Estrategia.

Para encontrar el camino de solución, la solución (S) dado un nodo (N), para cualquier **nodo** de hecho (NH) el **algoritmo** de primero en profundidad es el que damos a continuación :

1. Si N es un nodo NH entonces S = Contenido (N) o
2. Si N1 sucesor de (N) es nodo de NH) tal que S1 es solución de N1 entonces S = Contenido (N/S1)



las consecuencias fundamentales para la práctica, de la programación de esta realización extra - lógica del intérprete son las siguientes:

1. El orden de las cláusulas es relevante.
2. El orden de los objetivos en el cuerpo de la cláusula es relevante.

### ***Descripción del sistema***

El módulo deductivo permite crear las reglas deductivas, modificar dichas reglas, adicionarles restricciones de integridad y realizarle consultas deductivas a dicha base de conocimientos. A continuación se describe brevemente cada una de las opciones del sistema.

Crear reglas: es la opción que nos permite introducir nuevas reglas al sistema, lo que permitirá con posterioridad realizar la deducción de la base de conocimiento.

Modificar reglas: en el caso de que se desee realizar cualquier cambio en las reglas ya existentes de la base de conocimiento, con esta opción se puede modificar, inclusive, hasta la propia base de datos.

Añadir restricciones: esta opción nos permite añadir restricciones a la base de conocimiento de forma tal que se mantenga la integridad del mismo.

Deducción: es la realización de la deducción propia a la base de conocimiento, es decir, escoge los elementos necesarios para realizar la consulta.

Consulta: es donde se plasma directamente la consulta que se realiza a la BDD. Pueden realizarse tantas consultas como desee o comenzar con otra base de datos.

### **Generador de reglas, de restricciones y de consultas**

A pesar de haberse diseñado una gramática para escribir las reglas, otra para las deducciones y otra para las consultas, se utilizaron las herramientas que nos brinda el Delphi 2.0 de ambiente para que el usuario no tenga que conocer a fondo el ObjectProlog

y tener que escribir completamente cada regla, restricción o consulta, sino guiarlo en este empeño. Estas facilidades permiten hacer las listas de selección, botones de comando (incluye el manejo de BD definido por el propio Delphi 2.0), cajas de edición entre otras, creándose una forma para cada generador.

## **BIBLIOGRAFÍA**

1. BERTINO, E. «Toward Deductive Database». Eight European Conference on Object-Oriented Programming. Bologna, Italy, 1994.
2. CASARNAYOR, R., J.C., «Lógica y bases de datos». Universidad Politécnica de Valencia. 1995.
3. CELMA G., M. «Bases de Datos Deductivas». Medellín, 1995.
4. GARCÍA Luciano, O. PADRÓN, R. MARINO. «Programación Lógica». Universidad de La Habana. 1992
5. KOWALSKY, Robert. «Logic for Problem Solving». New York. Elsevier, Norton Holland. 1992.
6. LOW, C: C., LU. H., OOI, B.C : HAN, J. «Efficient Access Methods in Deductive and Object-Oriented Database». On deductive and Object-Oriented Database. Second International Conference, DOOD'91. Munich, Germany. 1991.
7. PARDAENS, J., P. PEELMAN y L. YAUCA. «Semantic of Disjunctive Logic Program and Object-Oriented Database. On Second International Conference, DOOD'91. Munich, Germany. 1991.
8. ULLMAN, J. D. «A Comparison of Deductive and Object-Oriented Databases». On Second International Conference, DOOD'91. Munich, Germany, 1991.
9. ULLMAN, J. D. «Database and Knowledge - Base Sistem» Rockville, Md. Computer Science Press. 1989.
10. ULLMAN, J. D. «Principles of Database System». Copyright Computer Science Press. 1982. Second Edition.
11. WINSTON Patrick, H. «Prolog Programming for Artificial Intelligence». Cambridge, Massachusetts. 1986.