Inclusión de Consultas Conjuntivas bajo la Semántica de Bolsas

Nieves R. Brisaboa
Departamento de Computación
Universidad de A Coruña
15071 A Coruña, España
brisaboa@udc.es

Índice

1 INTRODUCCIÓN. **7**

1.1 Presentación del problema. 11

1.2 Estudios previos en semántica de bolsas. 13

1.3 Enfoque de este trabajo 14

2 DEFINICIÓN DE CONSULTAS CONJUNTIVAS. **17**

2.1 Introducción. 17

2.2 Las consultas conjuntivas en SQL. 17

2.3 Las consultas conjuntivas en notación lógica. 21

2.4 Equivalencia entre la notación lógica y la notación SQL para consultas conjuntivas. 25

3 EQUIVALENCIA E INCLUSIÓN DE LAS CONSULTAS CONJUNTIVAS BAJO EL MARCO SEMÁNTICO DE CONJUNTOS. **29**

3.1 Introducción. 29

3.2 Optimización de consultas conjuntivas. 30

3.2.1 Equivalencia. 30

3.2.2 Inclusión de consultas conjuntivas. 31

3.3 Comprobación de la inclusión de consultas conjuntivas en el marco semántico de conjuntos. 32

3.3.1 Aplicación de Inclusión e Inclusión de Consultas. . . . 32

4 LAS CONSULTAS CONJUNTIVAS EN EL MARCO SEMÁNTICO DE LAS BOLSAS. 37

4.1 Introducción. 37

4.2 Descripción del marco semántico de Bolsas. 38

4.2.1 Multiplicidad. 38

4.2.2 Subbolsas. 39

4.2.3 Unión de bolsas. 40

4.2.4 Bases de Datos Isomórficas. 41

4.3 Semántica de las Consultas Conjuntivas bajo el marco de la teoría de bolsas. 41

4.3.1 Semántica de las Consultas Conjuntivas expresadas en notación SQL, bajo el marco semántico de bolsas. . . 41

4.3.2 Las Consultas Conjuntivas expresadas en notación lógica, bajo el marco semántico de bolsas. 44

5 B-INCLUSIÓN DE CONSULTAS: RESULTADOS DE CHAUDHURI Y VARDI. 51

5.1 Introducción. 51

5.2 Condiciones necesarias para la b-inclusión de consultas conjuntivas. 52

5.3 Condición suficiente para la b-inclusión de consultas conjuntivas. 56

5.4 Otros resultados sobre b-inclusión. 60

6 BASES DE DATOS Y SISTEMAS DE ETIQUETADO. 63

6.1 Introducción. 63

6.2 Definiciones y Notación. 63

6.3 Sistemas de Etiquetado. 64

6.4 Definición de Bases de Datos respecto a un sistema de etiquetado $\mathcal{L}$. 65

6.5 Bases de Datos y Bases de Datos **LS2**. 67

6.5.1 Unión de bases de datos respecto a **LS1** y **LS2**. 68

6.5.2 Bases de datos isomórficas. 69

6.5.3 Relación de Subbolsa entre Bases de Datos. 69

6.6 Semántica de las Consultas Conjuntivas sobre bases de datos LS1 y LS2. 71

6.6.1 Aplicación de una consulta conjuntiva a una base de datos LS1 o LS2. 75

7 BASES DE DATOS CANÓNICAS. 77

7.1 Introducción. 77

7.2 Ejemplo de construcción del Conjunto de Bases de Datos Canónicas de una consulta conjuntiva Q dada. $\mathcal{CBDC}(Q)$. . 79

7.3 Cálculo del número de Bases de Datos Canónicas construidas a partir de una consulta Q dada. 82

7.4 **Definición formal de Bases de Datos Canónicas.** 86

7.4.1 Definición de $db(Q)$. 86

7.4.2 Q-aplicaciones θ_i. 87

7.4.3 Aplicación de una Q-aplicación sobre un átomo de $db(Q)$. 87

7.4.4 Aplicación de una Q-aplicación sobre $db(Q)$. 87

7.4.5 Q-aplicaciones isomórficas. 88

7.5 Algoritmo para construir $\mathcal{CBDC}(Q)$. 89

8 COMPROBACIÓN DE LA B-INCLUSIÓN. 93

8.1 Introducción. 93

8.2 Lema y definiciones previas. 93

8.2.1 Definición de θ_d y t_d. 96

8.3 Demostración del Teorema Principal. 97

8.4 Procedimiento para demostrar la b-inclusión de consultas usando $\mathcal{CBDC}(Q)$. 99

8.5 Ejemplos de comprobación de la b-inclusión. 100

8.6 Comparación con procedimientos anteriores. 104

9 COMPARACIÓN DE LOS POLINOMIOS QUE EXPRESAN LAS MULTIPLICIDADES. 105

9.1 Comparación de polinomios de una variable. 105

9.1.1 Grupo I. 107

9.1.2 Grupo II. 107

9.1.3 Grupo III. 110

9.1.4 Grupo IV. 111

9.2 Procedimiento de comparación de polinomios de una variable $p(x)$ y $q(x)$. 112

9.3 Comparación de polinomios de dos variables. 114

10 EJEMPLO COMPLETO DE COMPROBACIÓN

DE LA B-INCLUSIÓN DE DOS CONSULTAS. **117**

10.1 Introducción. 117

10.2 Ejemplo de comprobación de la b-inclusión. 117

10.3 Construcción del conjunto de bases de datos canónicas $\mathcal{CBDC}(Q)$. 119

10.4 Aplicación de Q y Q' sobre cada base de datos canónica $d_i \in \mathcal{CBDC}(Q)$. 120

10.5 Comparación de los polinomios $|t_{d_i}|_{Q(d_i)}$ y $|t_{d_i}|_{Q'(d_i)}$ 122

11 CONCLUSIONES. **127**

12 LÍNEAS DE TRABAJO FUTURAS. **129**

1

INTRODUCCIÓN.

Las técnicas de optimización de consultas se basan en transformar una consulta dada en otra que sea semánticamente equivalente pero mas "barata" de evaluar. Por ello, poder decidir si dos consultas son equivalentes o no, es fundamental para la optimización. Dos consultas Q y Q' son equivalentes si al aplicar ambas sobre cualquier base de datos $\mathcal{D}$ los resultados que obtienen, que denotaremos $Q(\mathcal{D})$ y $Q'(\mathcal{D})$ respectivamente, son iguales[1]. Es decir:

$$Q \equiv_s Q' \Longleftrightarrow Q(\mathcal{D}) =_s Q'(\mathcal{D}) \quad \forall \mathcal{D}$$

En general la equivalencia de consultas es indecidible por lo que la investigación se ha centrado en el estudio de la equivalencia de tipos de consultas concretos. El tipo de consulta que ha atraido más investigación ha sido el de las consultas conjuntivas por ser el más habitual en bases de datos relacionales. Así, el problema de la optimización de consultas conjuntivas se ha estudiado extensamente y se entiende actualmente muy bien. Un estudio extenso de este tipo de consultas y de los problemas asociados a su optimización puede encontrarse en [28].

[1] Utilizaremos la letra s (de *sets*) a lo largo de este trabajo como subíndice de las relaciones de igualdad $=_s$, equivalencia $\equiv_s$, inclusión de conjuntos $\subseteq_s$ e inclusión de consultas $\leq_s$ para expresar que dichas relaciones se circunscriben al marco semántico de conjuntos. Del mismo modo se utilizará la letra b (de *bags o bolsas*), como subíndice de esas mismas relaciones, cuando se den en el marco semántico de *bolsas* o *multisets*. Más adelante, en esta tesis, se definen formalmente dichas relaciones y su notación, pero se ha optado por utilizar ya los subíndices en este capítulo de introducción para darle mayor unidad a la notación del trabajo.

En una primera aproximación se podría decir que una consulta conjuntiva es aquella formada por operaciones de proyección selección y join, es decir, es una consulta típica de las que se pueden realizar con SQL (sin usar la operación de Unión).

Por otro lado, es también posible estudiar las consultas conjuntivas bajo la aproximación de la programación lógica [24]. La lógica matemática es el modelo de datos subyacente a las bases de datos deductivas. Un lenguaje de manipulación y consulta de estas bases de datos es Datalog, una adaptación de Prolog para adecuarlo al trabajo con bases de datos [28, 24]. Por tanto, las consultas conjuntivas pueden estudiarse también cuando se expresan en notación Datalog. Como se muestra más adelante las notaciones SQL y Datalog son equivalentes para la expresión de consultas conjuntivas.

En la aproximación de la programación lógica, es decir en Datalog, una *consulta conjuntiva* es una regla Datalog *segura* y *no recursiva* [28]

Se ha estudiado mucho sobre el modo de comprobar la equivalencia de consultas conjuntivas, y sobre los problemas de optimización de las mismas [7, 1, 2]. En estas investigaciones se asume el postulado teórico básico de las bases de datos relacionales de que las relaciones son *conjuntos* de tuplas.

La equivalencia de consultas conjuntivas se estudia analizando su mutua inclusión, es decir, se entiende que dos consultas conjuntivas son equivalentes cuando están mutuamente incluidas la una en la otra. En otras palabras, bajo el habitual marco semántico conjuntista, una consulta conjuntiva Q está incluida en otra consulta conjuntiva Q', denotado $Q \leq_s Q'$, si para cualquier base de datos $\mathcal{D}$, el resultado que obtiene la consulta conjuntiva Q cuando se aplica sobre $\mathcal{D}$, denotado $Q(\mathcal{D})$, está incluido en $Q'(\mathcal{D})$. Es decir, el conjunto de tuplas obtenido por Q al aplicar esta consulta sobre cualquier base de datos $\mathcal{D}$, es un subconjunto del conjunto de tuplas obtenido por Q' cuando se aplica sobre esa misma base de datos. Lógicamente dicha inclusión de conjuntos (sets) se denota con el símbolo habitual $\subseteq_s$. Por tanto:

$$Q \leq_s Q' \iff \forall \mathcal{D} \;\; Q(\mathcal{D}) \subseteq_s Q'(\mathcal{D})$$

Así pues, se entiende que dos consultas conjuntivas son equivalentes cuando los resultados que obtienen sobre cualquier base de datos están mutuamente incluidos (y por tanto son iguales). Es decir:

$$Q \equiv_s Q' \Longleftrightarrow \quad \forall \mathcal{D} \;\; Q(\mathcal{D}) \subseteq_s Q'(\mathcal{D}) \quad \bigwedge \;\; Q'(\mathcal{D}) \subseteq_s Q(\mathcal{D})$$

Dicho de otro modo, dos consultas conjuntivas son equivalentes cuando están mutuamente incluidas la una en la otra, siendo por tanto iguales en cuanto a los resultados que obtienen, cuando ambas se aplican sobre cualquier base de datos. Es decir:

$$Q \equiv_s Q' \Longleftrightarrow Q \leq_s Q' \;\; \bigwedge \;\; Q' \leq_s Q$$

La ventaja de estudiar la equivalencia de consultas conjuntivas vía el análisis de su mutua inclusión, es que existe un procedimiento que permite saber si dos consultas conjuntivas están incluidas una en la otra simplemente estudiando sintácticamente las reglas Datalog que las expresan. El Capítulo 14 de [28] presenta un estudio completo de este tema

El problema es que SQL, el lenguaje estandar de los gestores de bases de datos relacionales comerciales como ORACLE o INFORMIX etc., no asume tal postulado conjuntista y permite que las tuplas tengan duplicados (a no ser que se especifique expresamente lo contrario). Es decir en los gestores relacionales comerciales cada relación no se ve como un *conjunto* de tuplas sino como una *bolsa o multiset* de tuplas[2].

La utilización de un diferente marco semántico produce que los resultados de las investigaciones sobre optimización de consultas no sean aplicables en la realidad. Dichas investigaciones asumían una semántica de conjuntos en vez de la semántica de bolsas con la que los gestores de base de datos relacionales comerciales realmente trabajan.

Como se verá en el Ejemplo 1.1.1, el que dos consultas estén incluidas una en la otra bajo la semántica de conjuntos no implica que también lo

[2]De todos modos, los gestores relacionales comerciales permiten "forzar" la utilización de un marco conjuntista, si se desea, mediante la definición de claves primarias, en donde no se admitan duplicados, y la petición de que se eliminen, del resultado de las consultas, las tuplas duplicadas.

estén bajo la semántica de bolsas. Por tanto, dos consultas equivalentes en el primer marco semántico no tienen porque serlo en el segundo. Así, los estudios sobre equivalencia e inclusión de consultas conjuntivas realizados hasta el momento no son en absoluto aplicables a las bases de datos comerciales.

Bajo la semántica de la teoría de bolsas, las relaciones no son conjuntos sino bolsas de tuplas, es decir, una tupla puede tener cualquier número de duplicados. Bajo esta semántica de bolsas, una consulta conjuntiva Q está *b-incluida* en otra consulta conjuntiva Q', denotado $Q \leq_b Q'$, si para *toda* base de datos $\mathcal{D}$, la relación obtenida al aplicar Q a dicha base de datos $\mathcal{D}$, que denotamos $Q(\mathcal{D})$, es una *subbolsa* de $Q'(\mathcal{D})$, denotado $Q(\mathcal{D}) \subseteq_b Q'(\mathcal{D})$. Es decir:

$$Q \leq_b Q' \iff \quad \forall \mathcal{D} \;\; Q(\mathcal{D}) \subseteq_b Q'(\mathcal{D})$$

Actualmente se desconoce si es decidible, según la semántica de bolsas, el problema de saber si dadas dos consultas conjuntivas Q y Q' una está incluida en la otra. Es decir, no se sabe si la comprobación de la inclusión de consultas conjuntivas constituye un problema decidible [18, 8]. No se sabe si puede existir un algoritmo que responda a la pregunta de si se cumple o no que, dadas dos consultas conjuntivas Q y Q', es realmente cierto que $Q \leq_b Q'$.

Es evidentemente lo importante que resulta estudiar el problema de la inclusión de consultas conjuntivas bajo la semántica de la teoría de bolsas, debido a que, efectivamente, en las implementaciones reales de las bases de datos relacionales se permiten duplicados de tuplas. Ioannidis y Ramakrishnan presentan extensamente en [17] esta problemática.

En esta tesis se demuestra que dadas dos consultas conjuntivas Q y Q' puede averiguarse si la primera está incluida en la segunda bajo la semántica de bolsas, denotado $Q \leq_b Q'$, usando un conjunto *finito* de bases de datos canónicas construidas a partir del cuerpo de la consulta Q. Dicho conjunto de bases de datos canónicas se denota $\mathcal{CBDC}(Q)$.

Este resultado, que se cumple *siempre* para cualquier pareja de consultas conjuntivas, reduce el problema a la comparación de dos polinomios especiales, y es para dicha comparación para la que no contamos en la actualidad con un procedimiento que permita comparar todos los posibles pares de polinomios. En el capítulo 9 se presenta un procedimiento de comparación de tales polinomios, válido para un gran número de casos.

Por tanto, aplicando los resultados de esta tesis se posiblilita la resolución del problema de la inclusión de consultas conjuntivas bajo la semántica de bolsas para un número muy extenso de parejas de consultas conjuntivas.

1.1 Presentación del problema.

Se presenta a continuación un ejemplo en el que se muestra cómo, efectivamente, el que dos consultas conjuntivas estén incluidas la una en la otra según la semántica de conjuntos, no implica el que lo estén también bajo la semántica de bolsas. En este ejemplo se trata de presentar la aplicación de las consultas a la base de datos de forma intuitiva. En posteriores capítulos se describe formalmente la semántica utilizada.

Ejemplo 1.1.1 :
Sea $\mathcal{D}$ una base de datos. Supóngase que base de datos $\mathcal{D}$ tiene una única relación s de esquema $S(A_1, A_2)$.

Sean Q_1 y Q_2 las siguientes consultas conjuntivas.

Q_1:	SELECT $S_1.A_1,\ S_1.A_2$	Q_2:	SELECT $A_1,\ A_2$
	FROM $s\ S_1,\ \ s\ S_2$		FROM s
	WHERE $S_1.A_2 = S_2.A_1$		

Estas consultas conjuntivas podrían expresarse alternativamente en notación Datalog de este modo:

$$Q_1 = q(X,Y) : -s(X,Y), s(Y,Z).$$

$$Q_2 = q(X,Y) : -s(X,Y).$$

Considérese que se aplican las dos consultas sobre una base de datos $\mathcal{D}$, obteniéndose así las tablas resultado $Q_1(\mathcal{D})$ y $Q_2(\mathcal{D})$. Supóngase que $Q_1(\mathcal{D})$ obtiene una tupla (a, b), esto implica que dicha tupla está en la relación s de la base de datos y por tanto la consulta Q_2 también obtendrá dicha tupla. Como esto es verdad para cualquier tupla que la consulta Q_1 pudiera obtener sobre cualquier base de datos arbitraria, se puede concluir que por lo tanto Q_1 está incluida en Q_2 según la semántica habitual de conjuntos. Es decir $Q_1 \leq_s Q_2$.

Pero bajo la semántica de la teoría de bolsas, Q_2 no incluye a Q_1. Para comprender el motivo, piénsese en una base de datos $\mathcal{D}$ en la que la relación s tuviera dos duplicados de la tupla (a, a). Es decir:

s
a, a
a, a

Entonces, $Q_1(\mathcal{D})$ contendrá cuatro ocurrencias de la tupla (a, a), porque dicha tupla (a, a) puede ser obtenida por Q_1 de cuatro formas diferentes, y los duplicados se conservan bajo la semántica de bolsas. Pero, por otro lado, como la relación resultante de la aplicación de Q_2 es siempre una "reproducción" de la relación s, se tendrá que $Q_2(\mathcal{D})$ obtendrá solo dos duplicados de la tupla (a, a), por lo que, según la semántica de bolsas, $Q_1(\mathcal{D})$ no sería una subbolsa, no estaría incluida en $Q_2(\mathcal{D})$. $\Box$

Lógicamente, desde el momento en que existe una base de datos $\mathcal{D}$ sobre la que, al aplicar las dos consultas ocurre que Q_1 no obtiene una subbolsa de las tuplas que obtiene Q_2, ya se puede afirmar que Q_1 no está incluida en Q_2 según la semántica de bolsas. Es decir:

$$\exists \mathcal{D}, \quad Q_1(\mathcal{D}) \not\subseteq_b Q_2(\mathcal{D}) \Longrightarrow Q_1 \not\leq_b Q_2$$

Obsérvese que, cuando se trabaja con la semántica de conjuntos, lo que hay debajo del problema de la inclusión de consultas es lo *restrictivas* que son dichas consultas. En el ejemplo, Q_1 es evidentemente más restrictiva que Q_2, puesto que la segunda obtendrá siempre todas las tuplas de la relación s, mientras que la primera obtendrá tan sólo aquellas tuplas de

s que cumplieran la condición de tener en la segunda columna un valor que aparezca como valor del primer atributo de alguna tupla de s.

Por el contrario, debajo de la inclusión de consultas bajo la semántica de bolsas, lo que se encuentra es una combinación de dos problemas: el de lo *restrictivas* que sean las consultas y el de *cuantos duplicados* puedan obtener de cada tupla.

Es evidente que, bajo la semántica de bolsas, la consulta Q_2 será capaz de obtener todas las tuplas de la relación s, mientras que Q_1 sólo obtendrá aquellas tuplas de s que cumplieran la condición que les impone; pero la relación Q_1 será capaz de obtener, en algunos casos, *más duplicados* de alguna tupla de s, como se muestra en el ejemplo. Por tanto, bajo la semántica de bolsas ninguna de las dos consultas del ejemplo está incluida en la otra.

1.2 Estudios previos en semántica de bolsas.

Las consultas conjuntivas bajo la semántica de bolsas se abordaron por primera vez en 1982 en un artículo de Dayal, Goodman y Katz [12], en donde esbozaban un algebra relacional extendida que permitía controlar la eliminación de tuplas duplicadas. Más tarde, en 1986, Klausner estudió el problema de la equivalencia y la inclusión de consultas conjuntivas bajo la semántica de bolsas [19]. Recientemente Ioannidis y Ramakrishnan [18] trabajaron sobre este mismo problema. Pero fueron Chaudhuri y Vardi [8] quienes más avanzaron en el estudio de la inclusión de consultas conjuntivas bajo el marco semántico de bolsas, que en esta tesis denominaremos *b-inclusión*.

Recientemente se ha estudiado cómo se ven afectados los lenguajes de consulta a bases de datos relacionales, en su capacidad expresiva y en su complejidad, cuando se trabaja sobre el marco semántico de bolsas [15]. Por otro lado, en [21] se presenta un estudio del significado de los duplicados en el modelo de datos relacional.

1.3 Enfoque de este trabajo

En esta tesis se trata de esclarecer las dificultades implícitas en el problema de la inclusión de consultas conjuntivas bajo la semántica de la teoría de bolsas. Se demuestra que, bajo la semántica de bolsas, se puede estudiar la inclusión de una consulta conjuntiva Q en una consulta conjuntiva Q', para cualquier par de consultas conjuntivas Q y Q' usando un conjunto reducido y *finito* de Bases de Datos Canónicas d, generadas a partir del cuerpo de la consulta Q. Dicho conjunto de bases de datos lo denotaremos ($\mathcal{CBDC}(Q)$).

En particular, en esta tesis se demuestra que:

$$Q \leq_b Q' \Longleftrightarrow \forall d \ \in \mathcal{CBDC}(Q) \ \ (\ Q(d) \subseteq_b Q'(d) \ )$$

Obsérvese, que este resultado permite reducir la comprobación de la inclusión de consultas conjuntivas en el marco semántico de bolsas, a su comprobación en un número finito (y reducido) de bases de datos canónicas que pueden construirse algorítmicamente a partir del cuerpo de la consulta Q presuntamente incluida en la otra consulta Q'.

De este modo, desde una definición de la inclusión de consultas que requiere que se dé la relación de subbolsa entre los resultados de Q y Q' aplicadas sobre cualquiera de las *infinitas posibles* bases de datos $\mathcal{D}$, se pasa a una definición en donde sólo se pide que se dé la relación de subbolsa entre los resultados de dichas consultas cuando se aplican al conjunto (finito y relativamente pequenõ) de bases de datos canónicas $d \in \mathcal{CBDC}(Q)$. Lógicamente el teorema principal de esta tesis garantiza que, si el resultado de Q es una subbolsa del de Q' cuando ambas consultas se aplican sobre cualquiera de las bases de datos canónicas $d \in \mathcal{CBDC}(Q)$, entonces el resultado que obtenga Q será una subbolsa del de Q' cuando ambas consultas se apliquen sobre cualquier base de datos $\mathcal{D}$. Es decir se demuestra que:

$$\forall d \in \mathcal{CBDC}(Q), \qquad Q(d) \subseteq_b Q'(d) \quad \Longleftrightarrow \qquad \forall \mathcal{D} \quad Q(\mathcal{D}) \subseteq_b Q'(\mathcal{D})$$

y por tanto que:

$$Q \leq_b Q' \iff \forall d \in \mathcal{CBDC}(Q), \qquad Q(d) \subseteq_b Q'(d)$$

Las tuplas de las bases de datos $d \in \mathcal{CDBS}(Q)$ tienen multiplicidades simbólicas[3], debido a lo cual comprobar si $Q(d)$ está incluida en $Q'(d)$, implica comprobar si

$$Poli(Q(d)) \leq Poli(Q'(d)) \quad sobre \quad \mathcal{Z}^+$$

Donde $\mathcal{Z}^+$ denota los enteros positivos y $Poli(Q(d))$ y $Poli(Q'(d))$ son dos polinomios que representan las multiplicidades de las tuplas en $Q(d)$ y $Q'(d)$. Dichos polinomios se obtienen a partir de las multiplicidades símbolicas de las tuplas en d.

Actualmente desconocemos si el problema de comparar tales polinomios es decidible. Dicho problema parece estar relacionado con el décimo problema de Hilbert. Para una explicación detallada de dicho problema, ver [11].

Resultados parciales de este trabajo de investigación han sido ya publicados [4, 5, 6]. En estas publicaciones, especialmente en la que se cita en tercer lugar, se perfilan ya las ideas fundamentales de esta tesis.

En los capítuplos segundo y tercero se presentan las consultas conjuntivas tanto en notación lógica como en notación SQL bajo el habitual marco semántico de conjuntos y la problemática asociada a su inclusión.

En el capítupo cuarto se introduce el marco semántico de bolsas, y en el cinco se presentan los resultados que, sobre inclusión de consultas en este marco, obtuvieron otros autores y que constituyen el punto de partida de este trabajo.

[3]En la página 64 de [8], Chaudhuri y Vardi *mencionan* que pretenden demostrar en un próximo artículo, que un resultado similar al nuestro se cumple *algunas veces*; pero aún no escribieron tal artículo. También es de destacar que nuestro resultado se cumple *siempre*.

Del capítulo sexto al octavo se describe lo que constituye la aportación principal de esta tesis. En el sexto se presenta la notación y algunas definiciones no estandares. En el séptimo se describen las Bases de Datos Canónicas y el algoritmo que permite construir el Conjunto de Bases de Datos Canónicas para una consulta Q dada, que denotamos $\mathcal{CBDC}(Q)$. En el capítulo octavo se presenta el teorema principal y el procedimiento de comprobación de la inclusión de consultas bajo el marco semántico de bolsas.

En el capítulo noveno se presenta un procedimiento capaz de comparar los polinomios que expresan las multiplicidades símbolicas obtenidas por el procedimiento descrito en el capítulo anterior.

Por último, en el capítulo décimo se presenta un ejemplo completo de comprobación de la inclusión de consultas bajo el marco semántico de bolsas. Las consultas conjuntivas que se utilizan en dicho ejemplo no podrían haber sido comparadas con procedimientos anteriores al descrito en esta tesis.

Finalmente, en el capítulo undécimo se presentan las conclusiones.

2

DEFINICIÓN DE CONSULTAS CONJUNTIVAS.

2.1 Introducción.

Las consultas conjuntivas han sido extensamente estudiadas por lo que se conocen actualmente muy bien. Ullman en [27, 28] realiza una amplia exposición de sus características y propiedades. A continuación se realiza una breve presentación de este tipo de consultas incidiendo especialmente en su semántica en el marco de conjuntos bajo el que se han estudiado habitualmente. Se presenta, así mismo, una notación alternativa y equivalente a la usual notación SQL para expresar consultas conjuntivas. Dicha notación es la usada en Datalog, un lenguaje de programación lógica desarrollado para trabajar en bases de datos deductivas. Se demuestra, además, la equivalencia de las dos notaciones con respecto a las consultas conjuntivas. La notación lógica tiene algunas ventajas por lo que será la que se emplee a lo largo de este trabajo.

2.2 Las consultas conjuntivas en SQL.

En álgebra relacional las consultas conjuntivas son aquellas que se forman usando tan sólo operaciones de selección[1] proyección y join. Por ello en

[1]Los predicados válidos para la selección son aquellos formados por conjunciones de predicados en los que un atributo se iguala a una constante o a otro atributo, tal y como

ocasiones se denominan consultas SPJ.

Una consulta conjuntiva en SQL es una consulta de la siguiente forma:

SELECT [DISTINCT] < lista de nombres de atributos >
FROM < lista de nombres de relaciones >
WHERE < lista de igualdades entre nombre de atributos o entre atributos y constantes >

Puesto que SQL no asume por defecto el marco semántico de conjuntos, es necesario usar *distinct* para forzar la eliminación de duplicados.

Ejemplo 2.2.1 :
Usando las relaciones del clásico ejemplo de Date[10]:

Suministrador(NumSum, NomSum, Ciudad, Status)
Piezas (NumPieza, NomPieza, Color, Ciudad)
Pedido (NumPieza, NumSum, Cantidad)

La consulta conjuntiva que obtendría todas parejas de número de piezas (NumPieza) y número de suministrador (NumSum), de aquellas piezas y suministradores con igual valor en el atributo *Ciudad*, sería:

SELECT *DISTINCT* NumSum, NumPieza
FROM Suministradores S, Piezas P
WHERE S.Ciudad=P.Ciudad

□

La semántica operacional de las consultas conjuntivas en SQL, podría definirse siguiendo estos pasos [10]:

---

ocurre en SQL. Podría decirse que las consultas conjuntivas son las típicas de SQL

- Se realiza el producto cartesiano de las relaciones que aparecen en la lista de relaciones.

- Se aplican a la tabla resultante las condiciones de selección expresadas en la lista de igualdades.

- Se realiza una proyección sobre la relación obtenida según los atributos expresados en la lista de nombres de atributos. Lógicamente esta proyeccción como operación del álgebra relacional eliminará los duplicados. Pero en realidad en SQL no los elimina a no ser que se use *distinct*.

A continuación se presenta un ejemplo detallado del modo de operación.

Ejemplo 2.2.2 :
Supóngase que la relación llamada $preferidas$, cuyo esquema es $Preferidas(Asig_1, Asig_2)$, expresa la primera y la segunda asignatura preferida por alguna persona.

$preferidas$

$Asig_1$	$Asig_2$
Matemáticas	Física
Matemáticas	Filosofía
Geografía	Historia

La relación $profesorado$, de esquema $Profesorado(Asig, Nom)$, indica qué materia enseña cada persona:

$profesorado$

$Asig$	Nom
Matemáticas	María
Filosofía	Carmen

La consulta que extraería de la base de datos la lista de personas que enseñan alguna asignatura que haya sido elegida en primer lugar por alguien, ateniéndose al marco semántico conjuntista, sería:

SELECT DISTINCT Nom
FROM $profesorado$, $preferida$
WHERE $profesorado.Asig = preferida.Asig_1$

Siguiendo paso a paso el proceso de evaluación, descrito previamente, se tiene:

1.- Producto Cartesiano: El resultado de realizar la operación de producto cartesiano entre las relaciones $profesorado$ y $preferidas$ se muestra en la siguiente tabla:

$profesorado * preferidas$

$profesorado.Asig$	$profesorado.Nom$	$preferidas.Asig_1$	$preferidas.Asig_2$
Matemáticas	María	Matemáticas	Física
Matemáticas	María	Matemáticas	Filosofía
Matemáticas	María	Geografía	Historia
Filosofía	Carmen	Matemáticas	Física
Filosofía	Carmen	Matemáticas	Filosofía
Filosofía	Carmen	Geografía	Historia

2.- Selección de las tuplas que cumplen la condición de join: En este caso dichas tuplas son:

$profesorado.Asig$	$profesorado.Nom$	$preferidas.Asig_1$	$preferidas.Asig_2$
Matemáticas	María	Matemáticas	Física
Matemáticas	María	Matemáticas	Filosofía

3.- Proyección sobre los atributos pedidos: Al realizar dicha proyección sobre la columna Nom -por efecto de *distint* que fuerza la semántica conjuntista- se eliminan los valores duplicados, por lo que el resultado final de la consulta tiene una única tupla:

$profesorado.Nom$
María

□

2.3 Las consultas conjuntivas en notación lógica.

Bajo esta notación, una consulta conjuntiva es un programa Datalog con exactamente una regla y ésta debe ser además no recursiva. Una consulta conjuntiva puede verse como una regla Datalog o como un programa Datalog. Dicho programa Datalog se puede *siempre* expresar, como se demostrará más adelante, mediante una sentencia SQL que implemente las operaciones de proyección selección y join necesarias.

Según la sintaxis lógica típica de Datalog una consulta conjuntiva Q es una regla de la forma:

$$Q = q(\vec{X}) : -p_1(\vec{X}_1), ..., p_n(\vec{X}_n)$$

Donde $\vec{X}$ y $\vec{X}_i$ son tuplas de variables (y quiza también constantes) y donde los p_i son nombres de predicados que representan los nombres de las relaciones de la base de datos.

$q(\vec{X})$ es la *cabeza* de la regla y $p_1(\vec{X}_1), ..., p_n(\vec{X}_n)$ es su *cuerpo*. Cada $p_i(\vec{X}_i)$ del cuerpo de la regla se denomina *átomo*. Los p_i representan los nombres de los diferentes predicados. Obsérvese que las igualdades entre variables que se imponen en la consulta para implementar operaciones de join, no son explícitas, sino que se expresan por las multiples posibles ocurrencias de la misma variable.

En el contexto de las bases de datos deductivas, una base de datos es un conjunto de *hechos* cada uno de los cuales es un predicado con constantes como argumentos[2]. Se considera a la base de datos formada por dos tipos de hechos que configuran dos diferentes componentes de la misma:

La Base de Datos Extensional EDB: Está formada por los llamados *hechos base* que son aquellos hechos que forman la extensión concreta

[2]Estrictamente hablando los hechos de las bases de datos deductivas pueden tener argumentos no constantes, pero como en este trabajo no es necesario contemplar este caso puesto que estamos en realidad trabajando sobre el modelo de bases de datos de los gestores relacionales comerciales, mantendremos esa definición por simplicidad

de la base de datos en un momento determinado. Los hechos base están definidos sobre *predicados extensionales*

La Base de Datos Intensional IDB: Está formada por los llamados *hechos derivados* que son aquellos que se extraen de la base de datos mediante la ejecución de una consulta. Corresponden a lo que en las bases de datos relacionales son las vistas. Los predicados sobre los que están definidos los hechos base se denominan *predicados intensionales*

Ejemplo 2.3.1 :
Usando el mismo ejemplo de antes, la consulta expresada con notación Datalog, que obtendría las parejas de número de suministrador y número de pieza de suministradores y piezas con igual valor en el atributo ciudad sería:

q(NumSum,NumPieza):-
s(NumSum,NomSum, **Ciudad**, Status), p(NumPieza,Nompieza, Color, **Ciudad**)

Obsérvese que la variable $Ciudad$ aparece dos veces con el mismo nombre, una vez en cada átomo del cuerpo de la consulta. De este modo se indica la operación de join que debe realizarse entre las dos relaciones.

Por otro lado, la base de datos extensional equivalente a la base de datos relacional del Ejemplo 2.2.2 estaría formada por todos aquellos hechos base de predicado $preferidas$ y $profesorado$. Es decir:

$$
\begin{aligned}
\mathcal{D} \quad = \{ \quad & preferidas(\text{Matemáticas, Física}),\\
& preferidas(\text{Matemáticas, Filosofía}),\\
& preferidas(\text{Geografía, Historia}),\\
\\
& profesorado(\text{Matemáticas, María})\\
& profesorado(\text{Filosofía, Carmen}) \ \}
\end{aligned}
$$

Obsérvese que el que existan dos nombres de predicado en los hechos base indica la existencia de lo que en bases de datos relacionales serían dos diferentes relaciones.

Por otro lado, la base de datos intensional estará formada por todos aquellos hechos que la consulta Q sea capaz de derivar cuando se aplique sobre la base de datos. Los hechos derivados tendrán de nombre de predicado q. $\square$

La semántica de una consulta conjuntiva expresada en notación Datalog se define en términos de *aplicaciones de asignación*. Una *aplicación de asignación* θ es una aplicación entre los símbolos de la consulta y los de la base de datos. Dicha aplicación hace corresponder a cada símbolo de la consulta Q un dato de la base de datos $\mathcal{D}$. Además dicha aplicación debe ser la *identidad* cuando se aplica a los nombres de predicado del cuerpo de la consulta, es decir, a cada nombre de predicado del cuerpo de la consulta le debe hacer corresponder un predicado de la base de datos extensional de igual nombre.

Dicho de otro modo, para que la aplicación de asignación θ sea posible es necesario que por cada nombre de predicado en el cuerpo de la consulta Q, exista en la base de datos un predicado, correspondiente a algún hecho base, con igual nombre.

Así, una aplicación de asignación θ de una consulta conjuntiva Q a una base de datos $\mathcal{D}$ produce que a cada una de las variables de Q se les asigne el valor de una constante en $\mathcal{D}$ y a cada nombre de predicado p_i en el cuerpo de Q se le asigne un nombre de predicado correspondiente a un hecho base de $\mathcal{D}$ con igual nombre al del predicado en cuestión. De este modo a cada átomo del cuerpo de Q se le asigna un hecho base de $\mathcal{D}$. Es decir, si se tiene la consulta conjuntiva Q:

$$Q = q(\vec{Y}) : -p_1(\vec{Y_1}), ..., p_n(\vec{Y_n})$$

Donde $p_i(\vec{Y_i})$ es un átomo del cuerpo de Q y $X_1...X_k$ son las variables de $\vec{Y_i}$. Si θ es una aplicación de asignación de Q a $\mathcal{D}$, entonces se tiene que:

$\theta(X_i)$ - Es el valor del dato de $\mathcal{D}$ que θ le asigna a la variable X_i. Es decir, es el dato que se le asigna a la variable X_i mediante la aplicación de asignación θ

$\theta(p_i(\vec{Y_i}))$ - Es el hecho base de $\mathcal{D}$ que se le asigna al átomo $p_i(\vec{Y_i})$ de Q mediante la aplicación de asignación θ. Es decir, $\theta(p_i(\vec{Y_i}))$ denota el hecho base $p_i(\theta(X_1)...\theta(X_k))$ de $\mathcal{D}$

$\theta(q(\vec{Y})$ - Es el hecho derivado t obtenido por la consulta Q mediante la aplicación de asignación θ.

t - Denota cualquier hecho derivado, obtenido por la consulta Q al aplicarla sobre cualquier base de datos $\mathcal{D}$, mediante cualquier aplicación de asignación θ.

Ejemplo 2.3.2 :
Se presenta exactamente el mismo ejemplo que se utilizó para explicar la semántica de las consultas conjuntivas en notación SQL, (Ejemplo 2.2.2) pero ahora se utilizará para explicar la semántica de las consultas expresadas en Datalog.

La base de datos $\mathcal{D}$ del Ejemplo 2.2.2 está formada por hechos base definidos sobre los predicados $preferidas$ y $profesorado$. Dicha base de datos, como se vió, puede expresarse en forma de conjunto de hechos base. Así, la base de datos estaría formada por el conjunto de hechos base corresponientes al predicado $preferidas$ y por el conjunto de hechos base correspondientes al predicado $profesorado$. El conjunto de hechos base correspondientes al predicado $preferidas$ es:

$$\begin{gathered}\{preferidas(\text{Matemáticas, Física}),\\ preferidas(\text{Matemáticas, Filosofía}),\\ preferidas(\text{Geografía, Historia})\}\end{gathered}$$

El conjunto de hechos base correspondientes al predicado $profesorado$ es:

$$\begin{gathered}\{profesorado(\text{Matemáticas, María})\\ profesorado(\text{Filosofía, Carmen})\}\end{gathered}$$

La consulta que extraería de la base de datos la lista de personas que enseñan alguna asignatura que haya sido elegida en primer lugar por alguien, sería en notación Datalog la siguiente:

$$Q = q(Nom) : -profesorado(\mathbf{ASIG}, Nom), preferidas(\mathbf{ASIG}, Asig_2)$$

Al aplicar la consulta Q a la base de datos[3], se tratarían de realizar todas las aplicaciones de asignación posibles. En este ejemplo sólo hay dos posibles aplicaciones de asignación que denotaremos θ_1 y θ_2 respectivamente. Los valores de la base de datos que se asignan a cada variable de Q mediante cada una de estas dos aplicaciones de asignación se describen en la siguiente tabla:

θ_1	θ_2
$\theta_1(ASIG)$ = Matemáticas	$\theta_2(ASIG)$ = Matemáticas
$\theta_1(Nom)$ =María	$\theta_2(Nom)$ = María
$\theta_1(Asig_2)$ = Física	$\theta_2(Asig_2)$ =Filosofía

El conjunto de hechos derivados como resultado de la aplicación de la consulta Q a la base de datos, es decir el conjunto de hechos derivados que formarán la base de datos intensional, estará formado por los valores que se le hayan asignado a las variables de la cabeza en cada aplicación de asignación. En la consulta del ejemplo la única variable de la cabeza es Nom y el valor que tanto θ_1 como θ_2 le asignan a dicha variable es *María*, por tanto el resultado de la aplicación de la consulta Q sobre la base de datos $\mathcal{D}$, formada por los hechos base descritos, es el conjunto de hechos derivados siguiente:

$$Q(\mathcal{D}) \qquad = \{\ q(\text{ María})\ \}$$

Como era de esperar el resultado de la consulta es el mismo, se exprese ésta en notación SQL o en notación Datalog. □

2.4 Equivalencia entre la notación lógica y la notación SQL para consultas conjuntivas.

Las dos formas de denotar la consultas conjuntivas son equivalentes y brevemente se bosqueja aquí el método de transformación de la sintaxis

[3]Obsérvese que la variable ASIG - escrita en negrita y mayúsculas para resaltarla - es la que soportará la operación de join

SQL a la sintaxis lógica.

Dada una consulta conjuntiva en notación SQL, la siguiente secuencia de pasos genera, en notación Datalog, una consulta conjuntiva equivalente a la consulta conjuntiva expresada en SQL de la que se parte:

- Por cada atributo de cada una de las relaciones que haya en la cláusula FROM se tendrá una variable diferente.
- Por cada relación que haya en la cláusula FROM se tendrá un predicado, en el cuerpo de la consulta, con su correspondiente conjunto de variables en el mismo orden en que estén dichos atributos en el esquema de la relación.
- Como cada atributo corresponde a una variable distinta, por cada igualdad entre atributos expresada en la cláusula WHERE existirá una igualdad entre variables tal que $X = Y$. Estas igualdades establecen una relación de equivalencia entre las variables. Se selecciona una variable como representante de cada clase de equivalencia y cada variable de la clase se reemplaza por su representante.
- Las variables del predicado de la cabeza serán aquellas que representan a los atributos expresados en la cláusula SELECT.

Denominaremos $transformada(Q)$ a la consulta conjuntiva, expresada con sintaxis lógica, resultado de aplicar sobre la consulta conjuntiva Q, expresada en SQL, los pasos de transformación antes expuestos.

Teorema 2.4.1 *Sea Q una consulta conjuntiva expresada en SQL y $transfomada(Q)$ la consulta conjuntiva resultado de aplicar sobre Q los pasos de transformación antes descritos. Entonces la aplicación Q sobre cualquier base de datos $\mathcal{D}$ produce idéntico resultado que la aplicación de $transformada(Q)$ sobre la misma base de datos.*

Demostración:
Es evidente por cómo se construye $transformada(Q)$. $\square$

Debido a que es más sencillo utilizar la sintaxis y la semántica lógica que la sintaxis SQL para expresar consultas conjuntivas y para probar los resultados obtenidos en este trabajo, se usará de aquí en adelante dicha notación.

3

EQUIVALENCIA E INCLUSIÓN DE LAS CONSULTAS CONJUNTIVAS BAJO EL MARCO SEMÁNTICO DE CONJUNTOS.

3.1 Introducción.

En este capítulo se presenta una revisión de la problemática asociada a la equivalencia de consultas conjuntivas bajo el habitual marco semántico de conjuntos. Se exponen además los resultados principales de la investigación sobre el tema, así como la formalización que se ha hecho del problema de la inclusión de consultas conjuntivas y el procedimiento para comprobar si dadas dos consultas conjuntivas Q y Q', una está o no, incluida en la otra según el marco semántico de la teoría de conjuntos.

A lo largo de este trabajo se asume que cuando se habla de una base de datos $\mathcal{D}$ se trata de una base de datos relacional, es decir, de una base de datos con tan sólo hechos base según la terminología de las bases de datos deductivas. Recuérdese que en bases de datos deductivas se diferencia la EDB o "Extensional Data Base", formada exclusivamente por *hechos base*, y la IDB "Intensional Data Base" formada por los conjuntos de *hechos derivados* al aplicar reglas o programas Datalog a la EDB

3.2 Optimización de consultas conjuntivas.

En la optimización de consultas, se trata siempre de encontrar una consulta que sea *equivalente* a la consulta que realiza el usuario pero que sea menos costosa de evaluar.

Se entiende que dos consultas conjuntivas Q_1 y Q_2 son *equivalentes* si al aplicarlas sobre cualquier base de datos $\mathcal{D}$ obtienen exactamente el mismo conjunto de tuplas[1]. Es decir:

$$Q_1 \equiv_s Q_2 \Longleftrightarrow \quad \forall \mathcal{D} \;\; Q_1(\mathcal{D}) =_s Q_2(\mathcal{D})$$

Si, por ejemplo, se tienen dos consultas equivalentes pero una de ellas tiene un número inferior de átomos, aquella con menos átomos será más "barata" de evaluar pues reducir el número de átomos significa reducir el número de operaciones de join necesarias para evaluar la consulta y, dado que la operación de join es la más "cara", eliminar operaciones de este tipo redunda inmediatamente en el "abaratamiento" del coste de evaluación de la consulta.

Es necesario, por tanto, contar con un procedimiento que permita saber cuándo dos consultas conjuntivas son equivalentes.

3.2.1 Equivalencia.

Sean Q_1 y Q_2 dos consultas conjuntivas. Se dice que $Q_1 \equiv_s Q_2$ si y sólo si para toda base de datos $\mathcal{D}$, se cumple que $Q_1(\mathcal{D}) = Q_2(\mathcal{D})$. Es decir:

$$Q_1 \equiv_s Q_2 \Longleftrightarrow \quad \forall \mathcal{D} \;\; Q_1(\mathcal{D}) =_s Q_2(\mathcal{D})$$

El procedimiento que permite averiguar si dos consultas conjuntivas son o no equivalentes se apoya en la inclusión mutua de dichas consultas. Es decir,

[1] A lo largo de todo este capítulo el marco semántico de referencia es el marco de conjuntos o *sets*, por ello se usa una s como subíndice de las relaciones de inclusión de conjuntos $\subseteq_s$, inclusión de consultas $\leq_s$, igualdad $=_s$ y equivalencia $\equiv_s$

para comprobar que para toda base de datos se cumple que $Q_1(\mathcal{D}) =_s Q_2(\mathcal{D})$, lo que se hace es estudiar si los resultados obtenidos por cada una de las dos consultas conjuntivas, cuando ambas se aplican sobre la misma base de datos $\mathcal{D}$, están uno incluido en el otro. Evidentemente:

$$\forall \mathcal{D} \quad Q_1(\mathcal{D}) =_s Q_2(\mathcal{D}) \iff Q_1(\mathcal{D}) \subseteq_s Q_2(\mathcal{D}) \quad \wedge \quad Q_2(\mathcal{D}) \subseteq_s Q_1(\mathcal{D})$$

Así, la definición operativa de la equivalencia de consultas conjuntivas es la siguiente:

Definición 3.2.1 Sean Q_1 y Q_2 dos consultas conjuntivas. Se dice que dichas consultas son equivalentes, $Q_1 \equiv Q_2$, si *para toda* base de datos $\mathcal{D}$, se cumple que $Q_1(\mathcal{D}) \subseteq_s Q_2(\mathcal{D})$ y $Q_2(\mathcal{D}) \subseteq_s Q_1(\mathcal{D})$, es decir, son equivalentes si se cumple que cada una de las consultas conjuntivas *incluye* a la otra, denotando la relación de inclusión de consultas, en este marco semántico de conjuntos, con el signo $\leq_s$:

$$Q_1 \equiv_s Q_2 \iff \quad \forall \mathcal{D} \;\; Q_1 \leq_s Q_2 \quad \wedge \quad Q_2 \leq_s Q_1$$

3.2.2 Inclusión de consultas conjuntivas.

Como se desprende del apartado anterior, el problema de averiguar si dos consultas conjuntivas son equivalentes se traslada a averiguar si están mutuamente incluidas la una en la otra. Bajo el marco semántico conjuntista, que es el que se estudia en el presente capítulo, una consulta conjuntiva Q está incluida en otra consulta conjuntiva Q', denotado $Q \leq_s Q'$ si para cualquier base de datos de entrada $\mathcal{D}$, la salida de Q cuando $\mathcal{D}$ es su entrada, denotada $Q(\mathcal{D})$, está incluida en $Q'(\mathcal{D})$.

Se entiende que $Q(\mathcal{D})$, está incluida en $Q'(\mathcal{D})$ cuando $Q(\mathcal{D})$ es un subconjunto de $Q'(\mathcal{D})$, denotado $Q(\mathcal{D}) \subseteq_s Q'(\mathcal{D})$. Por tanto siendo Q_1 y Q_2 dos consultas conjuntivas, se dice que $Q_1 \leq_s Q_2$ si *para toda* base de datos $\mathcal{D}$ se cumple que $Q_1(\mathcal{D}) \subseteq_s Q_2(\mathcal{D})$. Es decir:

$$Q \leq_s Q' \iff \forall \mathcal{D} \;\; Q(\mathcal{D}) \subseteq_s Q'(\mathcal{D})$$

Es decir la consulta Q_1 produce un subconjunto de las tuplas que produce la consulta Q_2 cuando ambas consultas se aplican sobre cualquier base de datos [2].

El problema de comprobar si $Q_1 \leq_s Q_2$ es NP-completo [7]

3.3 Comprobación de la inclusión de consultas conjuntivas en el marco semántico de conjuntos.

Chandra y Merlin [7] fueron los primeros en estudiar los problemas de equivalencia e inclusión de consultas conjuntivas. Posteriormente otros autores estudiaron variaciones de dichos problemas [20, 1, 2]

Ullman [28] presenta los resultados de todas estas investigaciones y detalla el procedimiento para comprobar la inclusión de cualquier pareja de consultas conjuntivas Q y Q', basándose en la posibilidad de establecer una aplicación entre los símbolos de ambas que se denomina **aplicación de inclusión**. Dicho procedimiento se apoya en el teorema que se define a continuación, el cual a su vez utiliza el concepto de **aplicación de inclusión**.

3.3.1 Aplicación de Inclusión e Inclusión de Consultas.

Una *aplicación de inclusión* del conjunto S de símbolos de la consulta conjuntiva Q_2 al conjunto T de símbolos de la consultas conjuntiva Q_1 es una aplicación τ que le asigna a cada símbolo de S un símbolo de T. Dicha aplicación τ debe cumplir algunas restricciones:

[2]Si se tratase de programas Datalog en general y no de consultas conjuntivas, el problema de probar si dos programas P_1 y P_2 están incluidos uno en el otro, denotado $P_1 \subseteq P_2$ es no decidible cuando se aplica a bases de datos con tan sólo predicados extensionales, pero si las bases de datos tienen también predicados intesionales, entonces el problema es decidible.

- La aplicación τ debe ser la identidad entre los símbolos que representan los nombres de los predicados.
- τ aplicada a una constante sólo puede devolver dicha constante.
- τ aplicada sobre una variable puede devolver una variable o una constante.

Definición 3.3.1 Sean Q_1 y Q_2 dos consultas conjuntivas tales que:

$$Q_1 : q_1(\vec{W}) : -p_1(\vec{X}_1), ..., p_l(\vec{X}_l)$$

$$Q_2 : q_2(\vec{V}) : -p_1(\vec{Z}_1), ..., p_k(\vec{Z}_k)$$

Se dice que una aplicación τ de Q_2 a Q_1 es una aplicación de inclusión si τ convierte la cabeza de Q_2 en la cabeza de Q_1, es decir $\tau(q_2(\vec{V})) = q_1(\vec{W})$, y convierte cada átomo del cuerpo de Q_2 en algún átomo del cuerpo de Q_1. No es necesario que todos los átomos del cuerpo de Q_1 queden "cubiertos", es decir, sean asignados a algún átomo de Q_2 mediante la aplicación, por lo que $\tau(Q_2)$ será como Q_1 pero quizá con algunos átomos menos. Por tanto:

$$\tau(q_2(\vec{V})) = (q_1(\vec{W}))$$

y

$$\forall i, \;\; \exists j \;\; | \;\; \tau(p_i(\vec{Z}_i)) = p_j(\vec{X}_j) \;\; 1 \leq i \leq k \qquad 1 \leq j \leq l$$

Teorema 3.3.1 *Sean Q_1 y Q_2 dos consultas conjuntivas. Entonces, $Q_1 \leq_s Q_2$ si y sólo si existe una* **aplicación de inclusión** *de Q_2 a Q_1. Es decir:*

$$Q_1 \leq_s Q_2 \iff \;\; \exists\, \tau \;\; y \;\; \tau \quad \textit{es una aplicación de inclusión} \quad \tau : Q_2 \rightarrow Q_1$$

Demostración:

Parte SI:

Asumiendo que existe la aplicación de inclusión $\tau : Q_2 \rightarrow Q_1$, considérese una base de datos extensional $\mathcal{D}$. Un hecho t será obtenido o derivado por la consulta Q_1 si y sólo si existe una aplicación de asignación ρ que hace corresponder a cada átomo del cuerpo de Q_1 un hecho base de $\mathcal{D}$. Es decir, $\rho(p_i(\vec{X}_i))$ es realmente un hecho base de $\mathcal{D}$ y $\rho(q_1(\vec{W}))$ es el hecho derivado t.

Ahora supongamos que se sustituye en Q_2 cada variable Z por $\rho(\tau(Z))$. Sabemos que $\tau(q_2(\vec{V})) = q_1(\vec{W})$ por hipótesis, luego $\rho(\tau(q_2(\vec{V}))) = \rho(q_1(\vec{W}))$ pero recuérdese que $\rho(q_1(\vec{W})) = t$ por tanto también $\rho(\tau(q_2(\vec{V}))) = t$.

Además $\tau(p_i(\vec{Z}_i)) = p_j(\vec{X}_j)$ para algún valor j por tanto $\rho(\tau(p_i(\vec{Z}_i))) = \rho(p_j(\vec{X}_j))$ pero $\rho(p_j(\vec{X}_j))$ es un hecho de $\mathcal{D}$.

Así, la aplicación consecutiva de τ y de ρ a la consulta conjuntiva Q_2 produce el hecho derivado t en la cabeza de Q_2. Puesto que t es un hecho derivado cualquiera producido por Q_1 al aplicar dicha consulta conjuntiva sobre la base de datos $\mathcal{D}$, queda claro que Q_2 aplicada sobre la misma base de datos producirá un superconjunto del conjunto de hechos derivados obtenido por la aplicación de Q_1.

Por tanto se puede concluir que sobre cualquier base de datos $\mathcal{D}$ la consulta conjuntiva Q_1 está incluida en Q_2, es decir $Q_1 \leq_s Q_2$ ya que se cumple que $Q_1(\mathcal{D}) \subseteq_s Q_2(\mathcal{D}) \forall \mathcal{D}$.

Parte SÓLO SI:

Asumiendo que $Q_1 \leq_s Q_2$ se construye una base de datos extensional concreta que denotaremos d siguiendo el procedimiento que se describe a continuación.

Se considera cada variable de Q_1 como una constante diferente de cualquier constante que pueda tener Q_1 en su cuerpo y de cualquier otra variable de Q_1.

Se define ρ de modo que para cada variable X de Q_1, $\rho(X)$ es la constante asociada a dicha variable. Por otro lado ρ es la identidad para los demás símbolos de Q_1 (es decir, para los nombre de predicado y las constantes si las hay).

De este modo d es una base de datos con exactamente un hecho base por cada átomo del cuerpo de Q_1. Así, los hechos base de la base de datos que hemos construido son:

$$d = \{\rho(p_1(\vec{X}_1)), \rho(p_2(\vec{X}_2))...\rho(p_l(\vec{X}_l))\}$$

Lógicamente, dado que ρ es la identidad entre nombres de predicado del cuerpo de la consulta y nombres de predicado de los hechos de la base de datos, se tendrá que la base de datos está formada por el siguiente conjunto de hechos:

$$d = \{(p_1(\rho(\vec{X}_1))), (p_2(\rho(\vec{X}_2)))...(p_l(\rho(\vec{X}_l)))\}$$

De este modo dado el átomo $p_i(\vec{X}_i)$ siendo $X_1, ...X_n$ las variables de $\vec{X}_i$ entonces en d habrá un hecho base de nombre de predicado p_i que tendrá como argumentos $(\rho(X_1), ...\rho(X_n))$, es decir, en d habrá un hecho base $p_i(\rho(X_1), ...\rho(X_n))$. Evidentemente si se aplica Q_1 a d se obtendrá, *al menos* el hecho derivado $\rho(q_1(\vec{W}))$.

Puesto que $Q_1 \leq_s Q_2$ es evidente que también se obtendrá el hecho derivado $\rho(q_1(\vec{W}))$ cuando se aplique Q_2 sobre d. Por lo tanto debe existir una aplicación de asignación θ aplicable a las variables de Q_2 tal que $\theta(p_i(\vec{Z}_i)$ sea un hecho de d para cada átomo del cuerpo de Q_2 y además se cumplirá que $\theta(q_2(\vec{V})) = \rho(q_1(\vec{W}))$. Del mismo modo para todo i debe existir algún j tal que $\theta(p_i(\vec{Z}_i)) = \rho(p_j(\vec{X}_j))$.

Se deduce por tanto que si con las variables de Q_2 se utiliza la aplicación de asignación θ y a continuación la aplicación inversa a ρ, que denotaremos ρ^{-1}, se obendrán las variables de Q_1. Es decir la aplicación de inclusión τ existe y se define como $\tau = \rho^{-1}\theta$.

Obsérvese que la aplicación ρ tiene efectivamente una aplicación inversa pues establece una correspondencia de uno a uno. ρ^{-1} asigna a cada constante de d su correspondiente variable de Q_1 reconstruyéndose así cada átomo de Q_1 $\square$

4

LAS CONSULTAS CONJUNTIVAS EN EL MARCO SEMÁNTICO DE LAS BOLSAS.

4.1 Introducción.

A pesar de que el estudio sobre la equivalencia de consultas es básico para la implementación de optimizadores en los sistemas de gestión de bases de datos, el hecho es que dichos estudios no son aplicables debido a que los gestores de bases de datos comerciales no asumen el marco teórico de la semántica de conjuntos propio del modelo relacional. Por el contrario, dichos gestores comerciales, aceptan tuplas duplicadas y no eliminan los duplicados del resultado de una consulta a no ser que el usuario lo pida explícitamente. Es decir, los sistemas de gestión de bases de datos reales utilizan el marco semántico de las bolsas.

Desgraciadamente, como veremos a continuación, las conclusiones alcanzadas por las investigaciones realizadas sobre la equivalencia de consultas conjuntivas en el marco semántico de conjuntos no son en absoluto aplicables al marco semántico de las bolsas.

En este capítulo se describe inicialmente el marco semántico de las bolsas, para después analizar la semántica de las consultas conjuntivas, ya sean expresadas con notación SQL o con notación lógica, en dicho marco.

4.2 Descripción del marco semántico de Bolsas.

4.2.1 Multiplicidad.

Cuando se consideran las relaciones como *bolsas* de tuplas (en vez de como conjuntos de tuplas), cada tupla tiene cierto número de duplicados. Otra forma de expresarlo es considerar que cada tupla tiene asociado un entero positivo que indica su **multiplicidad**. De este modo cada relación es un conjunto de tuplas cada una de las cuales tiene un número, que podemos expresar entre corchetes cuadrados, que indica su multiplicidad.

Se presenta a continuación un ejemplo paralelo al utilizado para explicar la semántica de conjuntos.

Ejemplo 4.2.1 :
Obsérvese la base de datos $\mathcal{D}$ del siguiente ejemplo. Dicha base de datos tiene una única relación denominada $preferidas$ que considerada como una bolsa de tuplas, en vez de como un conjunto de tuplas, se puede representar así:

$\mathcal{D}$

$preferidas$
Matemáticas, Física;[5]
Matemáticas, Filosofía;[1]
Geografía, Historia;[8]

□

En la relación $preferidas$ del ejemplo anterior las multiplicidades indican que hay cinco estudiantes que eligen en primer lugar Mátematicas y en segundo Física, un estudiante que prefiere en primer lugar Matemáticas y en segundo lugar Filosofía y, por último, hay ocho estudiantes que prefieren Geografia e Historia en primer y segundo lugar respectivamente.

Cuando se dice que *la multiplicidad de una tupla t en una relación es n*, con $n \geq 0$, se quiere decir que hay n ocurrencias de la tupla t en esa relación, es decir n duplicados.

Así, $r = \{..., (a_1, a_2, \ldots, a_m; [k]), ...\}$ indica que la tupla $(a_1, a_2, \ldots, a_m)$ está k veces en la relación r. Dicho de otro modo, existen k hechos $r(a_1, a_2, \ldots, a_m)$ en la base de datos extensional, lo que también podemos denotarlo así: $r(a_1, a_2, \ldots, a_m; [k])$. De este modo la base de datos extensional $\mathcal{D}$ del ejemplo también podría expresarse como una bolsa de hechos así:

$$\begin{aligned}\mathcal{D} \quad = \quad & \{preferidas(\text{Matemáticas, Física;[5]}),\\ & preferidas\ (\text{Matemáticas, Filosofía;[1]}),\\ & preferidas\ (\text{Geografía, Historia;[8]})\}\end{aligned}$$

La multiplicidad de una tupla t en una relación s, se denotará $|t|_s$. Si t no está en s, entonces $|t|_s = 0$. Del mismo modo la multiplicidad de una tupla t en una base de datos $\mathcal{D}$ se denotará $|t|_{\mathcal{D}}$.

Al igual que en [8] se define una relación como una bolsa de tuplas de una cierta aridad fija y una base de datos como una asignación de relaciones a nombres de relación.

4.2.2 Subbolsas.

Una bolsa q es una *subbolsa* de otra bolsa q', denotado $q \subseteq_b q'$, si cada elemento t de q está incluido en q' con igual o mayor multiplicidad. Es decir:

$$q \subseteq_b q' \Longleftrightarrow \forall t \in q, \quad |t|_q \leq |t|_{q'}$$

Del mismo modo se denota la relación de subconjunto entre dos conjuntos como $q \subseteq_s q'$.

Lógicamente las relaciones de igualdad entre bolsas y entre conjuntos se denotarán respectivamente $q =_b q'$ y $q =_s q'$.

Ejemplo 4.2.2 :
Considérense a continuación las dos bolsas de tuplas siguientes:

$$\begin{aligned}s \quad = \quad & \{(\text{Matemáticas, Física;[3]}),\\ & (\text{Geografía, Historia;[1]})\}\end{aligned}$$

y

$$p = \{(\text{Matemáticas, Física; [6]}), (\text{Geografía, Historia;[6]})\}$$

Evidentemente:

$$s \subseteq_b preferidas \qquad y \qquad s \subseteq_b p$$

Mientras que por otro lado

$$p \not\subseteq_b preferidas$$

Esto se debe a que la multiplicidad de la primera tupla de p es mayor que la de la correspondiente tupla de $preferidas$. Usando, sin embargo, una semántica de conjuntos donde lógicamente las tuplas de s, p y $preferidas$ no tendrían multiplicidades, se vería que:

$$s \subseteq_s p \subseteq_s preferidas$$

□

Esto se debe a que en el marco semántico de conjuntos al no existir multiplicidades no se contempla nada más que si una tupla está o no en una relación (en vez de mirar el número de duplidados que tiene).

Es evidente a partir de las definiciones, que la condición de subbolsa incluye a la de subconjunto pero no al revés. Así, si una bolsa q es subbolsa de otra bolsa q', entonces -eliminando las multiplicidades y considerando ambas bolsas como conjuntos- es también subconjunto de la misma, mientras que lo contrario no es verdad.

4.2.3 Unión de bolsas.

Siguiendo siempre a Chauhuri y Vardi en [8], se define la unión de dos bolsas q y q', denotada $q \sqcup q'$, como la bolsa formada por todos los elementos que estén en alguna de las dos bolsas de partida q y q'. La multiplicidad de cada elemento de la unión se obtiene sumando la multiplicidad que ese elemento tiene en cada una de las dos bolsas que se unen, considerando que un elemento que no aparece en una bolsa tienen multiplicidad 0. Así:

$$q \sqcup q' = \{(t;[m]) \;\; | \, t \text{ está en } q \text{ o en } q' \text{ y la multiplicidad de } t \text{ sera } m = |t|_q + |t|_{q'}\}$$

Ejemplo 4.2.3 :
Es por tanto evidente que:

$$\begin{aligned} p \sqcup preferidas \sqcup s \quad = \{ \quad & (\text{Matemáticas, Física;[14]}), \\ & (\text{Matemáticas, Filosofía;[1]}), \\ & (\text{Geografía, Historia;[15]})\} \end{aligned}$$

□

4.2.4 Bases de Datos Isomórficas.

Una *base de datos* se ha definido como una asignación de relaciones a nombres de relación. Así, dos bases de datos son isomórficas si se convierten en idénticas al hacer un renombramiento consistente de sus constantes.

4.3 Semántica de las Consultas Conjuntivas bajo el marco de la teoría de bolsas.

Bajo este marco semántico, como ocurría en el caso de los conjuntos, toda consulta conjuntiva expresada en notación SQL se puede transformar a una consulta equivalente expresada en notación lógica y viceversa. A pesar de la equivalencia de los dos tipos de notaciones, es conveniente analizar cómo operan cada una de ellas bajo este nuevo marco semántico.

4.3.1 Semántica de las Consultas Conjuntivas expresadas en notación SQL, bajo el marco semántico de bolsas.

El marco semántico de bolsas es el que SQL asume por defecto. Como ya se expuso, procesar una consulta conjuntiva sobre una base de datos supone:

1. Calcular el producto cartesiano de las relaciones implicadas.

2. Aplicar la operación de selección para extraer todas las tuplas que cumplen las condiciones impuestas en la consulta

3. Aplicar la operación de proyección para extraer los atributos pedidos de las tuplas previamente seleccionadas. Este paso ahora se realiza sin eliminar los duplicados que es lo que hace SQL por defecto cuando no se especifica *distinct*.

Ejemplo 4.3.1 :
Se desarrolla a continuación el mismo ejemplo utilizado ya para describir la semántica operacional de las consultas conjuntivas expresadas en SQL en el marco semántico de conjuntos. Simplemente, ahora, se considera que las relaciones son bolsas de tuplas, por lo que se representan en tablas en donde cada tupla tiene asociado un número que representa su multiplicidad. Se asume que la base de datos $\mathcal{D}$ está compuesta por las relaciones $preferidas$ y $profesorado$ cada una de las cuales es una bolsa de tuplas. La relación $preferidas$ es la descrita en el ejemplo 4.2.1. Por otro lado la relación $profesorado$, que indica qué materia enseña cada persona, es la bolsa de tuplas representada en la siguiente tabla:

$profesorado$
Matemáticas, María; [2]
Filosofía, Carmen ; [1]

La multiplicidad de la tupla (Matemáticas, María; [2]), indica que hay dos profesoras llamadas María que enseñan Matemáticas [1].

La consulta que extraería de la base de datos la lista de personas que enseñan alguna asignatura que haya sido elegida en primer lugar por alguien, sería la misma que en el caso de la semántica de conjuntos pero sin *distinct*:

[1]Tambien podría indicar que hay dos asignaturas diferentes con el mismo nombre, las dos eneñadas por una misma María o por dos Marías diferentes, pero asumiremos en el ejemplo que dos asignaturas distintas con el mismo nombre no pueden darse, mientras que dos profesoras de igual nombre sí.

SELECT $profesorado.Nom$
FROM $profesorado, preferida$
WHERE $preferida.Asig_1 = profesorado.Asig$

Siguiendo paso a paso, el proceso de evaluación descrito previamente se tiene:

1.- Producto Cartesiano: La operación de producto cartesiano entre las relaciones $profesorado$ y $preferidas$ se calcula de la forma habitual y, lógicamente, la multiplicidad de cada tupla del resultado se obtiene multiplicando las multiplicidades de las dos tuplas que se concatenan.

$profesorado * preferidas$
Matemáticas, María, Matemáticas, Física;[10]
Matemáticas, María, Matemáticas, Filosofía;[2]
Matemáticas, María, Geografía, Historia;[16]
Filosofía, Carmen, Matemáticas, Física;[5]
Filosofía, Carmen, Matemáticas, Filosofía;[1]
Filosofía, Carmen, Geografía, Historia;[8]

2.- Selección de las tuplas que cumplen la condición de join: Se eliminan del resultado anterior aquellas tuplas que no cumplen la condición de join y se obtiene el siguiente resultado:

Resultado de la selección
Matemáticas, María, Matemáticas, Física;[10]
Matemáticas, María, Matemáticas, Filosofía;[2]

3.- Proyección sobre los atributos pedidos: Aunque en la bolsa resultante del paso anterior existen dos tuplas, al hacer la proyección sobre el atributo Nom las dos tuplas tienen el mismo valor por lo que en el resultado final aparece una única tupla con la multiplicidad resultante de sumar las de las dos tuplas que se tenían. Así, el resultado de la consulta es la siguiente bolsa de tuplas:

resultado
María; [12])

La multiplicidad 12 indicaría que por doce veces aparece una profesora llamada María como persona que imparte una materia que constituye una asignatura preferida por alguien. Obsérvese que en $preferidas$ hay seis personas que eligen Matemáticas como asignatura favorita y que hay 2 Marías que enseñan Matemáticas.

□

4.3.2 Las Consultas Conjuntivas expresadas en notación lógica, bajo el marco semántico de bolsas.

Cuando se considera una base de datos $\mathcal{D}$ como una bolsa, cada hecho tiene su multiplicidad. Para calcular los hechos derivados por una consulta conjuntiva, enunciada según la notación lógica, hay que buscar las aplicaciones de asignación que pueden realizarse desde la consulta a los hechos base de la base de datos.

Asumiendo que hay l diferentes posibles aplicaciones de asignación $\theta_1, \ldots, \theta_l$ de la consulta $Q = q(\vec{X}) : -p_1(\vec{X}_1), \ldots, p_n(\vec{X}_n)$ a la base de datos $\mathcal{D}$, la multiplicidad del hecho base $theta_j(p_i(\vec{X}_i)) \in \mathcal{D}$ alcanzado por el átomo $p_i(\vec{X}_i) \in Q$ mediante la aplicación de asignación θ_j, se denotará $m_{j,i}$. Es decir:

$$m_{j,i} = |\theta_j(p_i(\vec{X}_i))|, \quad i = 1, \ldots, n \quad \text{y} \quad j = 1, \ldots, l$$

Así, la multiplicidad del hecho derivado obtenido por la consulta Q al aplicarse sobre la base de datos $\mathcal{D}$ (considerada como una bolsa), mediante la aplicación de asignación θ_j se denotará m_j y se calculará multiplicando las multiplicidades de cada hecho base de $\mathcal{D}$ asignado a algún átomo de Q mediante esa aplicación de asignación θ_j. Es decir:

$$(\theta_j(q(\vec{X})); [m_j]) \quad \text{donde} \quad m_j = m_{j,1} * m_{j,2} * \ldots * m_{j,i} * \ldots * m_{j,n}$$

Por tanto el resultado de aplicar la consulta Q sobre la base de datos $\mathcal{D}$ debido a la aplicación de asignación θ_j, denotado $Q_{\theta_j}(\mathcal{D})$ es:

$$Q_{\theta_j}(\mathcal{D}) = \{ \ (t; [m_j]) \ \} = \{ \ (\theta_j(q(\vec{X})); [m_j]) \ \}$$

donde:

$t = \theta_j(q(\vec{X}))$. Es el hecho derivado t obtenido por la consulta Q al aplicarse sobre la base de datos $\mathcal{D}$ usando la aplicación de asignación θ_j, denotado $Q_{\theta_j}(\mathcal{D})$.

$[m_j \ ]$. Es la multiplicidad del hecho t y se calcula multiplicando la multiplicidad de cada hecho base de $\mathcal{D}$ alcanzado por algún átomo de Q mediante la aplicación de asignación θ_j. Es decir: $m_j = m_{j,1} * m_{j,2} * \cdots * m_{j,n}$

Obsérvese que el mismo hecho derivado t puede ser obtenido por Q mediante diferentes aplicaciones de asignación sobre $\mathcal{D}$. Asumiendo que hay $k \leq l$ diferentes aplicaciones de asignación de Q sobre $\mathcal{D}$ que producen el mismo hecho derivado t, la multiplicidad final de dicho hecho, denotada $|t|_{Q(\mathcal{D})}$ se calcula sumando cada una de las multiplicidades $m_j \quad 1 \leq j \leq k$ que el hecho t obtiene con cada una de las diferentes aplicaciones $\theta_j \quad 1 \leq j \leq k$ que lo producen. Es decir:

$$|t|_{Q(\mathcal{D})} = \sum_{j=1}^{j=k} m_j$$

donde cada m_j es la multiplicidad del hecho derivado t obtenido por Q al aplicarse sobre $\mathcal{D}$ mediante una de las aplicaciones de asignación $\theta_j \quad 1 \leq j \leq k$ que producen dicho hecho derivado.

Así, la multiplicidad de cada hecho derivado t obtenido por la aplicación de una consulta conjuntiva Q sobre una base de datos $\mathcal{D}$, denotada $|t|_{Q(\mathcal{D})}$ se calcula en dos pasos:

1. Se multiplican las multiplicidades $m_{i,j}$ de los hechos base de $\mathcal{D}$ alcanzados por un átomo $p_i(\vec{X}_i) \in Q$ mediante una especifica aplicación de asignación θ_j

2. Se suman las multiplicidades que este hecho t obtuvo mediante cada una de las $\theta_j, \quad 1 \leq j \leq k$ aplicaciones de asignación que lo produjeron.

Consecuentemente, asumiendo que hay l aplicaciones de asignación $\theta_1, \ldots, \theta_l$, de Q sobre $\mathcal{D}$, el resultado de aplicar la consulta Q sobre la base de datos $\mathcal{D}$ denotado $Q(\mathcal{D})$ es:

$$Q(\mathcal{D}) = Q_{\theta_1}(\mathcal{D}) \sqcup \cdots \sqcup Q_{\theta_l}(\mathcal{D})$$

Se presenta a continuación la misma consulta conjuntiva que en el Ejemplo 4.3.1 ahora expresada en Datalog. Lógicamente se producirá el mismo resultado, pero este ejemplo permitirá exponer paso a paso la semántica de Datalog cuando se opera sobre bolsas.

Ejemplo 4.3.2 :
Las base de datos $\mathcal{D}$, sobre la que se va a trabajar con las dos conocidas relaciones $preferidas$ y $profesorado$ se expresa esta vez, en notación lógica, como la bolsa de hechos base de la siguiente base de datos extensional:

$$\begin{aligned} \mathcal{D} \quad = \{ \quad & preferidas(\text{ Matemáticas, Física;[5]}), \\ & preferidas(\text{ Matemáticas, Filosofía;[1]}), \\ & preferidas(\text{ Geografía, Historia;[8]}), \\ \\ & profesorado(\text{ Matemáticas, María;[2]}), \\ & profesorado\ (\text{ Filosofía, Carmen;[1] })\} \end{aligned}$$

La consulta que extraería de la base de datos la lista de personas que enseñan alguna asignatura que haya sido elegida en primer lugar por alguien, sería en notación Datalog la ya expuesta en el Ejemplo 2.3.2:

$$Q = q(Nom) : -profesorado(ASIG, Nom), preferidas(ASIG, Asig_2)$$

Al aplicar la consulta Q a la base de datos $\mathcal{D}$, se tratarían de realizar todas las aplicaciones de asignación posibles. Como en el caso de la semántica de conjuntos, esta consulta conjuntiva sólo tiene dos posibles formas de realizar aplicaciones de asignación sobre la base de datos $\mathcal{D}$, dichas aplicaciones de asignación se denotarán θ_1 y θ_2. Los valores de la base de datos que se asignan a cada variable de Q mediante cada una de estas dos aplicaciones de asignación se describen en la siguiente tabla:

θ_1	θ_2
$\theta_1(ASIG)$ = Matemáticas	$\theta_2(ASIG)$ = Matemáticas
$\theta_1(Nom)$ =María	$\theta_2(Nom)$ =María
$\theta_1(Asig_2)$ =Física	$\theta_2(Asig_2)$ = Filosofía

Como en el caso de la semántica de conjuntos, el conjunto de hechos derivados como resultado de la aplicación de la consulta Q a la base de datos $\mathcal{D}$ estará formado por los valores que se le hayan asignado a las variables de la cabeza en cada aplicación de asignación. En la consulta del ejemplo la única variable de la cabeza es Nom y el valor que tanto θ_1 como θ_2 le asignan a dicha variable es *María*. Pero ahora es necesario calcular la multiplicidad con la que se obtiene el hecho derivado mediante cada aplicación de asignación.

Con la aplicación de asignación θ_1 se obtiene un hecho derivado t que tendrá una multiplicidad m_1. Dicha multiplicidad se calcula multiplicando las multiplicidades de los hechos base que alcanzaron los átomos del cuerpo de la consulta mediante la aplicación de asignación θ_1. Obsérvese que:

$$\theta_1(profesorado(Asig, Nom)) = (profesorado(\text{ Matemáticas, María}))$$

$$\theta_1(preferidas(Asig_1, Asig_2)) = (preferidas(\text{ Matemáticas, Física}))$$

Las multiplicidades de los hechos base $profesorado($ Matemáticas, María) y $preferidas($ Matemáticas, Física) en la base de datos son 2 y 5 respectivamente.

Luego $m_{1,1} = 2$ y $m_{1,2} = 5$ por lo que la multiplicidad del hecho derivado que se obtiene por la aplicación de asignación θ_1 es:

$$m_1 = m_{1,1} * m_{1,2} = 2 * 5 = 10$$

Del mismo la multiplicidad obtenida por el hecho derivado como resultado de la aplicación de la consulta mediante la aplicación de asignación θ_2 es:

$$\theta_2(profesorado(Asig, Nom)) = (profesorado\ (\text{ Matemáticas, María}))$$

$$\theta_2(preferidas(Asig_1, Asig_2)) = (preferidas(\text{ Matemáticas, Filosofía})$$

Las multiplicidades de los hechos base $profesorado($ Matemáticas, María) y $preferidas($ Matemáticas, Filosofía) en la base de datos son 2 y 1 respectivamente.

Luego $m_{2,1} = 2$ y $m_{2,2} = 1$ por lo que la multiplicidad del hecho que se obtiene por la aplicación de asignación θ_2 es:

$$m_2 = m_{2,1} * m_{2,2} = 2 * 1 = 2$$

Puesto que las dos aplicaciones de asignación θ_1 y θ_2 obtienen el mismo hecho derivado, se debe sumar la multiplicidad que dicho hecho obtiene con cada aplicación para calcular su multiplicidad final:

$$|t|_{Q(D)} = \sum_{j=1}^{j=k} m_j = m_1 + m_2 = 10 + 2 = 12$$

Por lo tanto el resultado final de la consulta Q cuando se aplica sobre la base de datos extensional $\mathcal{D}$, estará formado por la siguiente bolsa de hechos derivados:

$$Q(\mathcal{D}) \quad = \{q(\text{ María; } [12])\}$$

Evidentemente, puesto que la notación SQL y la notación Datalog son equivalentes para expresar consultas conjuntivas, el resultado es el mismo, no sólo en lo que respecta al conjunto de tuplas obtenidas, sino también en lo que respecta a sus multiplicidades. $\square$

5

B-INCLUSIÓN DE CONSULTAS: RESULTADOS DE CHAUDHURI Y VARDI.

5.1 Introducción.

Las consultas conjuntivas bajo la semántica de bolsas, se abordaron por primera vez en 1982 en un artículo de Dayal, Goodman y Katz [12] en donde esbozabán un algebra relacional extendida que permitía controlar la eliminación de tuplas duplicadas. Posteriormente, en 1986 Klausner estudió el problema de la equivalencia y la inclusión de consultas conjuntivas bajo la semántica de bolsas [19]. Recientemente Ioannidis y Ramakrishnan [18] trabajaron sobre este mismo problema. Pero fueron Chaudhuri y Vardi [8] quienes más avanzaron en el estudio de la inclusión de consultas conjuntivas bajo el marco semántico de bolsas, que en esta tesis denominaremos *b-inclusión*.

Recientemente se ha estudiado cómo se ven afectados los lenguajes de consulta a bases de datos relacionales en su capacidad expresiva y en su complejidad cuando se trabaja sobre el marco semántico de bolsas [15]. Por otro lado, en [21] se presenta un estudio del significado de los duplicados en el modelo de datos relacional.

Chaudhuri y Vardi en [8] exponen dos condiciones necesarias y una condición suficiente para la b-inclusión de consultas, pero no encuentran ningún procedimiento que permita comprobar si, dadas dos consultas conjuntivas cualesquiera, una de ellas está b-incluida en la otra. De hecho no saben siquiera si el problema es decidible. A continuación se exponen los resultados de su trabajo pues, aunque se superan por el procedimiento que se presenta en esta tesis, constituyen un marco de referencia obligado.

5.2 Condiciones necesarias para la b-inclusión de consultas conjuntivas.

Puesto que la relación de b-inclusión es estrictamente más fuerte que la relación de inclusión en el marco semántico de conjuntos, Chaudhuri y Vardi pensaron que sería razonable endurecer la condición de inclusión que se exige en el marco conjuntista para tratar de encontrar las condiciones que garantizasen la b-inclusión.

Recuérdese que, en el marco conjuntista, la condición necesaria y suficiente para que, dadas dos consultas conjuntivas Q y Q' se cumpla que $Q \leq_s Q'$, es la de que exista una **aplicación de inclusión** τ de Q' a Q. Dicha aplicación de inclusión se caracteriza por cumplir las siguientes propiedades:

- A cada una de las variables de Q' se les asignará alguna variable de Q.
- La aplicación sobre los nombres de predicados será la identidad.
- A la cabeza de Q' se le asigna la cabeza de Q.
- A cada átomo de Q' se le asigna un átomo, no necesariamente distinto, de Q.
- Puede ocurrir que algún átomo de Q no sea asignado a ningún átomo de Q'.

Chaudhuri y Vardi, endureciendo la condición de aplicación de inclusión, encontraron dos condiciones necesarias para la b-inclusión que se exponen y demuestran a continuación en sendas proposiciones.

En [8] Chaudhuri y Vardi tan sólo enuncian dichas condiciones necesarias pero, dado que es importante entenderlas bien para comprender su alcance, se presenta aquí no sólo la enunciación de dichas condiciones sino también la demostración que hacemos de las mismas[1]

Proposición 5.2.1 *Sean Q y Q' dos consultas conjuntivas tales que $Q \leq_b Q'$. Se debe cumplir que para todo nombre de predicado p, el número de átomos de predicado p en Q debe ser menor o igual al número de átomos de predicado p en Q'.*

Demostración:
Asúmase que $Q \leq_b Q'$ pero que el número de átomos de predicado p_i en Q es mayor que el número de átomos de predicado p_i en Q'. Siempre se podrá construir una base de datos $\mathcal{D}$ con tantos hechos base como nombres de predicado diferentes existan en el cuerpo de Q. Cada uno de los hechos base de $\mathcal{D}$ estará definido sobre un predicado diferente de igual nombre a uno de los predicados del cuerpo de Q.

De este modo sobre cada nombre de predicado del cuerpo de Q estará definido un único hecho base de $\mathcal{D}$. Además, las multiplicidades de todos los hechos base de $\mathcal{D}$ valdrán 1 excepto la multiplicidad del hecho base definido con el predicado p_i cuya multiplicidad será 2.

Por tanto, cuando se aplique Q sobre esta base de datos, a todos los átomos de Q con igual nombre de predicado se les asignará el mismo hecho base de $\mathcal{D}$. Dicho hecho base será aquel único hecho base que se definió con ese nombre de predicado. Es claro que existe una aplicación de asignación de Q a $\mathcal{D}$

La multiplicidad $|t|_{Q(\mathcal{D})}$ del hecho derivado por la consulta Q al aplicarse sobre la base de datos $\mathcal{D}$, así construida será:

$$|t|_{Q(\mathcal{D})} = m_1^{f(p_1)_Q} * m_2^{f(p_2)_Q} .... * m_n^{f(p_n)_Q}$$

[1]Se hace notar que las demostraciones no son ofrecidas por Chaudhuri y Vardi en su único trabajo presentado sobre el tema [8], sino que han sido deducidas en el proceso de desarrollo de esta tesis. Los autores citados hablan en [8] de un artículo completo que publicarían próximamente en donde aparecerían tales demostraciones, pero dicho artículo no ha sido publicado por el momento.

Donde $f(p_j)_Q$ indica la cardinalidad del predicado p_j en el cuerpo de Q, es decir, el número de átomos en el cuerpo de Q con nombre de predicado p_j. Como las multiplicidades de todos los hechos base de $\mathcal{D}$ valen 1 excepto la multiplicidad del hecho definido sobre el predicado de nombre p_i, se tendrá que la multipicidad del hecho derivado será:

$$|t|_Q = 1^{f(p_1)_Q} * 1^{f(p_2)_Q} ... * 2^{f(p_i)_Q} ... * 1^{f(p_n)_Q} = 2^{f(p_i)_Q}$$

Dado que $Q \leq_b Q'$ es evidente que debe existir una aplicación de asignación de q' a $\mathcal{D}$. Pero en el caso de Q' la multiplicidad del hecho derivado será:

$$|t|_{Q'(\mathcal{D})} = m_1^{f(p_1)_{Q'}} * m_2^{f(p_2)_{Q'}} .... * m_n^{f(p_n)_{Q'}}$$

$$=$$

$$|t|'_Q = 1^{f(p_1)_{Q'}} * 1^{f(p_2)_{Q'}} ... * 2^{f(p_i)_{Q'}} ... * 1^{f(p_n)_{Q'}} = 2^{f(p_i)_{Q'}}$$

Como Q' tiene menos átomos p_i que Q por hipótesis, es evidente que:

$$2^{f(p_i)_Q} \geq 2^{f(p_i)_{Q'}}$$

Por lo tanto $|t|_{Q(\mathcal{D})} \geq |t|_{Q'(\mathcal{D})}$ de donde se deduce, contradiciendo la hipótesis de partida, que la consulta Q no está incluida en la consulta Q'. Es decir:

$$|t|_{Q(\mathcal{D})} \geq |t|_{Q'(\mathcal{D})} \Longrightarrow Q \not\leq_b Q'$$

□

La idea subyacente a esta demostración es que el hecho de que Q tenga más átomos de predicado p_i que Q' va a facilitar que pueda aplicarse sobre más hechos base de predicado extensional p_i de una base de datos y pueda obtener, por tanto, mayor multiplicidad.

Esta proposición hace resaltar la diferencia básica entre inclusión de consultas bajo el marco semántico de conjuntos y bajo el marco semántico de bolsas. Cuando $Q \leq_s Q'$ el cuerpo de Q' debe ser menos "restrictivo" que el de Q, por tanto, cuantos menos átomos tenga Q' menos condiciones se le imponen a los hechos base sobre los que Q' se pueda aplicar y, por tanto, menos restrictiva resulta Q', por lo que más fácil será que $Q \leq_s Q'$. Pero, cuando se trabaja con bolsas, la multiplicidad que obtiene una consulta es el resultado de multiplicar las multiplicidades de todos los hechos base que se le asignan a sus átomos mediante una aplicación de asignación concreta y, por tanto, tener un menor número de átomos redunda en obtener una menor multiplicidad.

La segunda proposición, que enuncia la segunda condición necesaria encontrada por Chaudhuri y Vardi, añade otro matiz.

Proposición 5.2.2 *Sean Q y Q' dos consultas conjuntivas tales que $Q \leq_b Q'$ entonces se debe cumplir que todo átomo de Q debe ser asignado a algún átomo de Q' mediante alguna aplicación de inclusión τ de Q' a Q. Es decir, para todo átomo $p_i(\vec{X_i})$ del cuerpo de Q debe existir una aplicación de inclusión τ de Q' a Q tal que $p_i(\vec{X_i}) \in \tau(Q')^2$.*

Demostración:
Supóngase que $Q \leq_b Q'$ pero que existe un átomo de predicado p_i en el cuerpo de Q que no es asignado a ningún átomo de Q', mediante ninguna aplicación de inclusión τ de Q' a Q.

A partir de Q es posible construir una base de datos $\mathcal{D}$ con tantos hechos base como átomos tenga el cuerpo de Q. Por cada átomo de Q habrá un hecho base en $\mathcal{D}$ con igual nombre de predicado y cuyas constantes tendrán el mismo nombre que las variables del átomo correspondiente.

Así, $\mathcal{D}$ tendrá tantos hechos base como átomos tenga Q y a cada átomo de Q se le podrá asignar un hecho de $\mathcal{D}$ mediante la aplicación de asignación

[2]Obsérvese que aquí se está viendo a la consulta Q' como un conjunto de átomos por lo que $\tau(Q)$ es el conjunto de imágenes de dichos átomos producidas por la aplicación τ. Lo que se dice en el enunciado del Lema es que el átomo $p_i(\vec{X_i})$ de Q debe ser un elemento de dicho conjunto.

identidad θ. Las multiplicidades de todos los hechos de $\mathcal{D}$ valdrán 1, excepto la multiplicidad del hecho que corresponda al átomo p_i de Q que no es alcanzado por ninguna aplicación de inclusión τ de Q' a Q. La multiplicidad de ese hecho valdrá n, un entero arbitrariamente grande.

El que no exista alguna aplicación de inclusión τ de Q' a Q que permita alcanzar un átomo p_i de Q, indica que cuando se aplica Q' sobre $\mathcal{D}$, el hecho p_i de $\mathcal{D}$ no podrá ser alcanzado por ningún átomo de Q' mediante ninguna aplicación de asignación. Por tanto Q' obtendrá como resultado, mediante cualquier aplicación de asignación sobre $\mathcal{D}$, un hecho derivado con multiplicidad 1, mientras que Q podrá obtener al menos un hecho derivado con multipliciad n. Es evidente, por tanto, que Q no está incluida en Q' bajo el marco semántico de bolsas, es decir $Q \not\leq_b Q'$. □

Lo que esta proposición les sugirió a Chaudhuri y Vardi [8] es que se necesita "recubrimiento" de los átomos de Q por los átomos de Q' mediante alguna aplicación de inclusión y, así, estos autores enunciaron una condición suficiente para la inclusión de consultas conjuntivas en el marco semántico de bolsas. Dicha condición se expone en el apartado siguiente.

5.3 Condición suficiente para la b-inclusión de consultas conjuntivas.

Antes de enunciar y probar la condición suficiente de Chaudhuri y Vardi es necesario definir la **aplicación cobertora**.

Definición 5.3.1 Una **aplicación cobertora** τ de una consulta conjuntiva Q' **sobre** una consulta conjuntiva Q es una aplicación de inclusión en la que *todos* los átomos de Q son alcanzados o **cubiertos** por algún átomo de Q'. Es decir, es una aplicación de inclusión sobreyectiva.

Por tanto, el cuerpo de Q es una subbolsa de la imagen del cuerpo de Q' obtenida mediante la aplicación cobertora τ. Es decir $Q \subseteq_b \tau(Q')$[3].

[3]Obsérvese que aquí se está viendo a las consultas como bolsas de átomos por lo que $\tau(Q')$ es la bolsa de imágenes de los átomos de Q'. Lo que se dice en la definición es que el cuerpo de la consulta Q es una bolsa de átomos que es subbolsa de $\tau(Q')$.

Ejemplo 5.3.1 :
Considérense las siguientes consultas conjuntivas:

$$Q = q(X,Y) : -p_1(X,Z), p_2(W,Y), p_2(W,V)$$

$$Q' = q(X,Y) : -p_1(X,Z), p_2(Z,Y)$$

$$Q'' = q(X,Y) : -p_1(X,Z), p_2(Z,Y), p_3(Y,W)$$

$$Q''' = q(X,Y) : -p_1(X,Z), p_2(Z,Y), p_2(Y,Y)$$

Obsérvese que existe una *aplicación cobertora* de Q *sobre* Q' puesto que los dos átomos de Q' son alcanzados o cubiertos por la aplicación de inclusión τ que se define a continuación:

$$\begin{array}{c} \tau(X) = X \\ \tau(Y) = Y \\ \tau(Z) = Z \\ \tau(W) = Z \\ \tau(V) = Y \end{array}$$

Por otro lado existe una aplicación de inclusión (pero no cobertora) de Q a Q''. Dicha aplicación τ se define a continuación:

$$\begin{array}{c} \tau(X) = X \\ \tau(Y) = Y \\ \tau(Z) = Z \\ \tau(W) = Z \\ \tau(V) = Y \end{array}$$

No existe una aplicación cobertora de Q a Q'' porque el átomo p_3 de Q'' no puede ser alcanzado por ninguna aplicación de inclusión de Q a Q''. Es decir, las consultas Q y Q'' ni siquiera cumplen la condición necesaria enunciada en la Proposición 5.2.2.

Por último, obsérvese que existen dos aplicaciones de inclusión (pero ninguna cobertora) de Q a Q'''. Dichas aplicaciones τ_1 y τ_2 se definen a continuación:

τ_1	τ_2
$\tau_1(X) = X$	$\tau_2(X) = Y$
$\tau_1(Y) = Y$	$\tau_2(Y) = Y$
$\tau_1(Z) = Z$	$\tau_2(Z) = Z$
$\tau_1(W) = Z$	$\tau_2(W) = Y$
$\tau_1(V) = Y$	$\tau_2(V) = Y$

Obsérvese que ahora sí se cumplen las dos condiciones necesarias para que Q esté b-incluida en Q''' es decir para que se cumpla $Q \leq_b Q'''$:

Se cumple la condición enunciada en la Proposición 5.2.1 ya que el número de veces que aparece un nombre de predicado en el cuerpo de Q es igual o mayor al número de veces que aparece ese mismo nombre de predicado en el cuerpo de Q'''

Por otro lado, se cumple la condición enunciada en la Proposición 5.2.2 debido a que todos los átomos de Q''' son alcanzados por alguna aplicación de inclusión, aunque como ninguna de ellas cubre todos los átomos de Q''' al mismo tiempo, ninguna de ellas es una aplicación cobertora. $\Box$

Ahora ya es posible enunciar la condición suficiente de Chaudhuri y Vardi expuesta en [8] mediante la siguiente proposición:

Proposición 5.3.1 *Sean Q y Q' dos consultas conjuntivas. Si existe una aplicación de inclusión de Q'* **SOBRE** *Q que sea* **COBERTORA**, *entonces $Q \leq_b Q'$.*

Demostración:
Si existe una *aplicación cobertora* τ de Q' *sobre* Q, entonces todos los hechos de cualquier base de datos $\mathcal{D}$ alcanzados por una aplicación de asignación θ de Q a $\mathcal{D}$, también podrán ser alcanzados por Q' aplicando a Q' sucesivamente la aplicación cobertora τ seguida de la aplicación de asignación θ. Es decir los hechos base de cualquier base de datos $\mathcal{D}$ que sean "alcanzados" por una aplicación de asignación θ aplicada a la consulta conjuntiva Q, serán una

subbolsa de los hechos base que alcance Q' aplicando a Q' sucesivamente la aplicación cobertora τ y después la aplicación de asignación θ. Es evidente que:

$$\forall \mathcal{D},\ \forall \theta \qquad \theta(Q) \subseteq_b \theta(\tau(Q'))$$

Por tanto todos los hechos derivados que obtenga como resultado la consulta Q al aplicarse sobre cualquier base de datos $\mathcal{D}$ serán también obtenidas por la consulta Q' con igual o mayor multiplicidad. $\square$

Después de haber demostrado la Proposición 5.3.1 vamos a utilizarla para extraer conclusiones sobre la inclusión y la b-inclusión de las consultas conjuntivas del ejemplo anterior.

Ejemplo 5.3.2 :
Usando la consultas conjuntivas del ejemplo anterior:

$$Q = q(X,Y) : -p_1(X,Z), p_2(W,Y), p_2(W,V)$$

$$Q' = q(X,Y) : -p_1(X,Z), p_2(Z,Y)$$
$$Q'' = q(X,Y) : -p_1(X,Z), p_2(Z,Y), p_3(Y,W)$$
$$Q''' = q(X,Y) : -p_1(X,Z), p_2(Z,Y), p_2(Y,Y)$$

Debido a que existen aplicaciones de inclusión de Q a Q', Q'' y Q''', se puede establecer usando la condición necesaria y suficiente para la inclusión de consultas en el marco semántico de conjuntos que:

$$Q' \leq_s Q \qquad Q'' \leq_s Q \qquad Q''' \leq_s Q$$

Por otro lado usando la condición suficiente para la b-inclusión de consultas conjuntivas, enunciada en la Proposición 5.3.1, ya que existe una aplicación cobertora de Q sobre Q', se puede decir que:

$$Q' \leq_b Q$$

Además, ya que las consultas Q y Q'' incumplen la condición necesaria para la b-inclusión de Q'' en Q, enunciada en la Proposición 5.2.2 se puede afirmar que:

$$Q'' \not\leq_b Q$$

Finalmente, obsérvese que usando las dos condiciones necesarias y la condición suficiente de Chaudhuri y Vardi no es posible saber si se cumple o no que $Q''' \leq_b Q$ ya que dichas consultas cumplen las dos condiciones necesarias pero no cumplen la condición suficiente. Más adelante en esta tesis se demostrará que el procedimiento que aquí se propone permite comprobar que efectivamente $Q''' \leq_b Q$. $\square$

5.4 Otros resultados sobre b-inclusión.

Además de las dos condiciones necesarias y de la condición suficiente enunciadas en la sección anterior, Chaudhuri y Vardi demuestran en [8] algunos otros resultados que aquí simplemente citaremos esbozando su demostración.

Teorema 5.4.1 *La b-inclusión de dos consultas conjuntivas Q y Q' implica la inclusión de las mismas según la semántica de conjuntos. Es decir*

$$Q \leq_b Q' \Longrightarrow Q \leq_s Q'$$

Demostración:
Es evidente, ya que la relación de subbolsa $\subseteq_b$ implica la de subconjunto $\subseteq_s$.
$\square$

Cuando en Q no hay átomos de igual predicado, entonces la condición suficiente enunciada en la Proposición 5.3.1 se convierte en condición *necesaria y suficiente*

Teorema 5.4.2 *Sean Q y Q' dos consultas conjuntivas. Si Q no tiene dos átomos con igual nombre de predicado, es decir si cada uno de sus átomos corresponde a un predicado diferente, entonces $Q \leq_b Q'$* **si y sólo si** *existe una aplicación cobertora de Q' sobre Q.*

Demostración:
No es necesario demostrar para este caso particular que es una condición suficiente puesto que ya se hizo para el caso general. Simplemente se introduce la demostración de que es también condición necesaria.

Puesto que no hay dos átomos con igual nombre, la no existencia de aplicación cobertora sólo puede deberse a tres razones:

No es posible realizar una aplicación de inclusión de Q' a Q. Esto impediría que existiera una aplicación cobertora, pero en este caso no habría inclusión en el marco de conjuntos y, por tanto, tampoco habría b-inclusión.

Existen más átomos en Q que en Q'. Evidentemente así sería imposible que existiese una aplicación cobertora, pero se estaría incumpliendo la condición necesaria expuesta en la Proposición 5.2.1, por lo que no habría b-inclusión.

Existe un átomo en Q con un nombre de predicado que está en Q'. Este es el caso de la consulta Q'' del ejemplo anterior. Evidentemente, si hay un átomo en Q con un nombre de predicado que no existe en Q', entonces se están incumpliendo las condiciones expresadas en las Proposiciones 5.2.1 y 5.2.2, por lo que no habrá b-inclusión.

Queda claro, por tanto, que no es posible que se dé la b-inclusión, $Q \leq_b Q'$, de esas consultas conjuntivas si no existe una aplicación cobertora de Q' sobre Q cuando en Q no existen átomos de igual predicado.□

Hay que resaltar que la proposición 5.3.1 y el teorema 5.4.2 fueron descubiertos independientemente por Ioannidis y Ramakrishnan en [17].

Otros resultados de Chaudhuri y Vardi en [8] pueden consultarse en su trabajo.

6

BASES DE DATOS Y SISTEMAS DE ETIQUETADO.

6.1 Introducción.

Es necesario definir las bases de datos respecto a dos sistemas de etiquetado debido a que en este trabajo se tratan dos tipos de bases de datos (siempre consideradas como bolsas). El primer tipo de bases de datos que se consideran son aquellas cuyas tuplas tienen multiplicidades expresadas por números enteros que representan el número de ocurrencias de cada tupla. El segundo tipo de bases de datos con las que trabajaremos son aquellas en las que las multiplicidades de sus tuplas se expresan mediante polinomios. Como se verá más adelante serán las bases de datos de este segundo tipo las que se usarán para comprobar la b-inclusión de consultas conjuntivas.

6.2 Definiciones y Notación.

Asumimos las definiciones de [28] y sólo damos algunas definiciones no estándares.

Una *consulta conjuntiva* es una regla Datalog segura no recursiva [28]. Los nombres de predicados y constantes se denotan por cadenas de caracteres comenzando por letra minúscula; las variables en los átomos de las consultas conjuntivas se representan por cadenas de caracteres que comienzan por letras mayúsculas.

Un *hecho* es una cláusula Horn que consta exactamente de un único literal positivo [28]. Dado un predicado p, *un hecho sobre (o definido en)* p es un hecho cuyo nombre de predicado es p.

Un *esquema de base de datos* es un conjunto finito de nombres de predicados. Se asume que todos los predicados usados aquí están implícitamente en $\mathcal{U}$, un esquema de base de datos fijo, y que hay un conjunto fijo de constantes denominadas *Universo Herbrand.* La *base de Herbrand* $B_{\mathcal{U}}$ para $\mathcal{U}$ es el conjunto de todos los hechos *ground o enraizados*[1] que pueden formarse usando los nombres de predicados en $\mathcal{U}$ y las constantes del Universo Herbrand [24].

En el resto de esta subsección, se adopta la definición de Ioannidis y Ramakrishnana [18], para las bases de datos.

6.3 Sistemas de Etiquetado.

Un *sistema de etiquetado* $\mathcal{L}$ es una quintupla $\mathcal{L} = < L, ., +, 0, \leq \ >$ tal que [18]:

- L es un dominio de etiquetas con orden parcial $\leq$;
- . es el *producto*, una operación binaria en L que es asociativa y conmutativa;
- $+$ es la *adición*, una operación binaria en L que es asociativa y conmutativa;
- 0 es la identidad respecto a $+$ y el nulo de la multiplicación; también es el menor elemento de L respecto a $\leq$;
- $\forall\ a, b \in L - \{0\}\ (a \leq a.b)$;
- $\forall\ a, b, c, d \in L\ (\ a \leq b$ y $c \leq d \Rightarrow a + c \leq b + d)$;

[1] Hechos enraizados son aquellos que sólo tienen constantes como argumentos

- $\forall a \in L\ (\exists b \in L\ (\ a \leq b$ y $a \neq b))$

En esta tesis se van a emplear dos sistemas de etiquetado que denominaremos LS1 y LS2 respectivamente:

LS1: Utilizará la aritmética en $\mathcal{Z}^+$, los enteros no negativos, donde $+$, $.$, 0, y $\leq$ son bien conocidos y L es $\mathcal{Z}^+$,el conjunto de los Naturales más el cero. $L = \mathcal{Z}^+ = \mathcal{N} \cup \{0\}$.

LS2: Utilizará la aritmética de polinomios. Los coeficientes de dichos polinomios así como sus exponentes son enteros no negativos. En este sistema $+$ y $.$ son las operaciones habituales de adición y producto de polinomios.

El cero en este sistema es el polinomio **0**, un polinomio que se evalúa a cero para cualquier valor de sus variables.

La relación $\leq$ se define como sigue: Sean $P_1(\vec{X})$ y $P_2(\vec{X})$ dos polinomios. Entonces $P_1(\vec{X}) \leq P_2(\vec{X})$ si se cumple que $P_1(\vec{X}) \leq P_2(\vec{X})$ en $\mathcal{Z}^+$. Es decir, el valor númerico del polinomio $P_1(\vec{X})$ es menor o igual, en $\mathcal{Z}^+$, que el del polinomio $P_2(\vec{X})$ para cualquier valor que tomen sus variables. Dicho de otro modo, para cualquier valor que tengan las variables se cumple que:

$$\mathbf{0} \leq P_1(\vec{X}) \leq P_2(\vec{X})$$

La relación $<$ entre polinomios se define de modo similar.

6.4 Definición de Bases de Datos respecto a un sistema de etiquetado $\mathcal{L}$.

Sea $\mathcal{L} = < L, ., +, 0, \leq\ >$ un sistema etiquetado. Una base de datos $\mathcal{D}$ respecto a $\mathcal{L}$ es una función total desde la base Herbrand $B_{\mathcal{U}}$ a $\mathcal{L}$.

Sea $\mathcal{D}$ una base de datos respecto a $\mathcal{L}$, entonces se puede representar esta base de datos $\mathcal{D}$ como el conjunto de objetos de la forma :

$$p(b_1, \ldots, b_t; [m])$$

Donde $p(b_1, \ldots, b_t)$ es un hecho en $B_{\mathcal{U}}$ que es asignado por $\mathcal{D}$ a m, siendo m un elemento de L distinto de 0. Es decir:

$$m \in L \quad y \quad m \neq 0$$

En otras palabras, aquellos hechos enraizados de la base Herbrand a los que se les asigne el valor cero, no se incluirán en la representación de la base de datos.

Los elementos de una base de datos que, como vimos, tienen la forma $p(b_1, \ldots, b_t; [m])$ los denominaremos de aquí en adelante *tuplas*.

Sea $t = p(a_1, a_2, \ldots, a_n; [m])$ una tupla de $\mathcal{D}$; se dice que m es la *multiplicidad* de t en $\mathcal{D}$ y se denota como $|t|_{\mathcal{D}}$.

Un hecho $p(b_1, \ldots, b_n)$ *está en* $\mathcal{D}$, denotado como $p(b_1, \ldots, b_t) \in \mathcal{D}$, si hay una tupla $p(a_1, \ldots, a_t; [m])$, en la base de datos $\mathcal{D}$ tal que $a_i = b_i$, para $i = 1, ..., n$.

La *multiplicidad del hecho* $q = p(b_1, \ldots, b_t)$ *en* $\mathcal{D}$ *es* m, denotado $|q|_{\mathcal{D}} = m$, si la tupla $p(b_1, \ldots, b_t; [m])$, se encuentra en $\mathcal{D}$. La multiplicidad de un hecho q que no está en $\mathcal{D}$, vale cero, es decir: $|q|_{\mathcal{D}} = 0$.

En los ejemplos, las bases de datos se representan del modo usual en bases de datos relacionales mediante tablas que muestran las tuplas y sus multiplicidades encolumnadas formando relaciones. Cada relación se identifica en la tabla por el nombre de predicado que la representa colocado en la cabecera de la columna. Alternativamente, las bases de datos también se pueden representar como bolsas de tuplas. El siguiente ejemplo ilustra todas las anteriores definiciones.

Ejemplo 6.4.1 :
Sea $\mathcal{D}$ una base de datos respecto a **LS1**. Supongamos que $\mathcal{D}$ asigna 2 a $q(a, b, e)$, 8 a $r(a, c)$, 3 a $r(f, b)$ y 0 al resto de los hechos enraizados en la base Herbrand $B_{\mathcal{U}}$. La base de datos resultante, representada como una bolsa de tuplas es:

$$\mathcal{D} \quad = \quad \{\ q(a, b, e; [2]), \quad r(a, c; [8]), \quad r(f, b; [3])\}$$

La tupla $r(f, b; [3])$ de la base de datos $\mathcal{D}$ indica que hay tres ocurrencias del hecho $r(f, b)$ de la base Herbrand $\mathcal{B}_{\mathcal{U}}$ para $\mathcal{U}$, en la base de datos $\mathcal{D}$.

Alternativamente, la base de datos $\mathcal{D}$ se podría representar como la siguiente tabla:

q	r
$(a, b, e; [2])$	$(f, b; [3])$
	$(a, c; [8])$

Evidentemente, la tupla $q(a, b, e; [2])$, de la base de datos $\mathcal{D}$ indica que hay dos ocurrencias del hecho $q(a, b, e)$ de la base Herbrand $\mathcal{B}_{\mathcal{U}}$ para $\mathcal{U}$, en la base de datos $\mathcal{D}$.$\Box$

Una *relación (definida en el predicado q)* es una base de datos donde todos sus elementos se definen en el mismo predicado (q).

6.5 Bases de Datos y Bases de Datos LS2.

A partir de ahora, reservamos el término *base de datos* para bases de datos relativas al sistema etiquetado **LS1**; y las bases de datos referentes al sistema etiquetado **LS2** se denominarán *bases de datos LS2.* Por lo tanto, cuando se trate con bases de datos LS2 las operaciones binarias $.$ y $+$ y las relaciones $\leq$ (y $<$) serán las definidas para polinomios.

Las multiplicidades de los elementos de las bases de datos **LS2** se denominarán multiplicidades *simbólicas.*

Ejemplo 6.5.1 :
Una base de datos LS2 denominada $\mathcal{D}$ podría ser la formada por el siguiente conjunto de tuplas:

$$\mathcal{D} \quad = \quad \{r(a,b;[m_1]), \quad p(b,a;[m_2+m_3]), \quad r(b,b;[m_4m_5+m_6^3])\}$$

Evidentemente, por como se definió el sistema de etiquetado LS2, las m_i deben ser variables en el rango de $\mathcal{Z}^+$. $\square$

6.5.1 Unión de bases de datos respecto a LS1 y LS2.

La *unión* de dos bases de datos $\mathcal{D}$ y $\mathcal{D}'$, sean éstas LS1 ó LS2, se denota $\mathcal{D} \sqcup \mathcal{D}'$, y se define como:

$$\mathcal{D} \sqcup \mathcal{D}' = \{q(t_1,\ldots,t_{n_q};[m]) \mid s = q(t_1,\ldots,t_{n_q}), s \in \mathcal{D} \vee s \in \mathcal{D}' \text{ y } m = |s|_{\mathcal{D}} + |s|_{\mathcal{D}'}\}$$

Ejemplo 6.5.2 :
Las siguientes bases de datos representan las tres asignaturas preferidas por un grupo de estudiantes, en orden de preferencia:

$$\begin{aligned} \mathcal{R}_1 = \{ \quad & \text{r(Matemáticas,Física,Biología;[5]),} \\ & \text{r(Matemáticas,Filosofía,Química;[1]),} \\ & \text{r(Geografía,Lengua,Filosofía;[8])}\} \end{aligned}$$

$$\begin{aligned} \mathcal{R}_2 = \{ \quad & \text{r(Matemáticas,Física,Biología;[3]),} \\ & \text{r(Geografía,Lengua,Filosofía;[1])}\} \end{aligned}$$

$$\begin{aligned} \mathcal{R}_3 = \{ \quad & \text{r(Matemáticas,Física,Biología;[6]),} \\ & \text{r(Geografía,Lengua,Filosofía;[6])}\} \end{aligned}$$

La unión de las tres bases de datos será:

$$\begin{aligned} \mathcal{R}_1 \sqcup \mathcal{R}_2 \sqcup \mathcal{R}_3 =\{ \quad & \text{r(Matemáticas,Física,Biología;[14]),} \\ & \text{r(Matemáticas,Filosofía,Química;[1]),} \\ & \text{r(Geografía,Lengua,Filosofía;[15])}\} \end{aligned}$$

$\square$

6.5.2 Bases de datos isomórficas.

Dos bases de datos son *isomórficas* si son idénticas al hacer un renombramiento adecuado de sus constantes.

Si una o las dos bases de datos son bases de datos LS2, además de hacer un renombramiento adecuado de las constantes es necesario realizar una asignación de valores a las variables en las multiplicidades simbólicas de modo que se consiga que las dos bases de datos queden idénticas.

Ejemplo 6.5.3 :
Sea $\mathcal{D} = \{\ r(a, b; [m_1]),\ p(b, a; [m_2]),\ r(b, b; [m_3])\ \}$ una base de datos LS2. Una base de datos LS1 isomórfica a $\mathcal{D}$ es la siguiente: $\mathcal{D}' = \{\ r(1, 2; [5]),\ p(2, 1; [2]),\ r(2, 2; [3])\ \}$. $\square$

6.5.3 Relación de Subbolsa entre Bases de Datos.

A continuación se presenta la definición de la relación de subbolsas para bases de datos **LS1** y **LS2**.

Una base de datos $\mathcal{D}$ de tipo **LS1** o **LS2** es una *subbolsa* de otra base de datos $\mathcal{D}'$ también de tipo **LS1** o **LS2**, y se denota $\mathcal{D} \subseteq_b \mathcal{D}'$, si para cada hecho t en $\mathcal{D}$, ocurre que t está también en $\mathcal{D}'$ con multiplicidad igual o mayor. Es decir $|t|_{\mathcal{D}} \leq |t|_{\mathcal{D}'} \quad \forall t \in \mathcal{D}$. Por tanto :

$$\mathcal{D} \subseteq_b \mathcal{D}' \Longleftrightarrow |t|_{\mathcal{D}} \leq |t|_{\mathcal{D}'} \quad \forall t \in \mathcal{D}$$

Cuando $\mathcal{D}$ es una *subbolsa* de $\mathcal{D}'$, es decir, cuando se cumple que $\mathcal{D} \subseteq_b \mathcal{D}'$, se dice que $\mathcal{D}$ está *b-incluida* en $\mathcal{D}'$. Los siguiente dos ejemplos aclaran esta definición.

Ejemplo 6.5.4 :
Sean las siguiente tres bases de datos $\mathcal{R}_1, \mathcal{R}_2$ y $\mathcal{R}_3$, cada una de ellas con una única relación r.

$$
\begin{aligned}
\mathcal{R}_1 = \{ \quad & \text{r(Matemáticas,Física,Biología;[5])}, \\
& \text{r(Matemáticas,Filosofía,Química;[1])}, \\
& \text{r(Geografía,Lengua,Filosofía;[8])}\}
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{R}_2 =\{ \quad & \text{r(Matemáticas,Física,Biología;[3])}, \\
& \text{r(Geografía,Lengua,Filosofía;[1])}\}
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{R}_3 =\{ \quad & \text{r(Matemáticas,Física,Biología;[6])}, \\
& \text{r(Geografía,Lengua,Filosofía;[6])}\}
\end{aligned}
$$

Entre estas bases de datos se dan las siguientes relaciones de b-inclusión:

$$\mathcal{R}_2 \subseteq_b \mathcal{R}_1$$

$$\mathcal{R}_2 \subseteq_b \mathcal{R}_3$$

Pero

$$\mathcal{R}_3 \not\subseteq_b \mathcal{R}_1$$

Esto se debe a que la multiplicidad de la primera tupla de $\mathcal{R}_3$ es mayor que la multiplicidad correspondiente de esa tupla en $\mathcal{R}_1$. □

Veamos a continuación un ejemplo sobre bases de datos LS2

Ejemplo 6.5.5 :
Considérense las siguientes bases de datos LS2 que denominamos $\mathcal{R}_4$ y $\mathcal{R}_5$, cada una de ellas con una única relación r:

$$\mathcal{R}_4 = \quad \{\text{r(Enfermería,Biología,Estadística;}[m_1.m_2.m_3])\}$$

$$\mathcal{R}_5 = \quad \{\text{r(Enfermería,Biología, Estadística;}[m_1.m_2^2 + m_1.m_3^2])\}$$

En este caso es evidente que:

$$\mathcal{R}_4 \subseteq_b \mathcal{R}_5$$

Esto se debe a que $m_1.m_2^2 + m_1.m_3^2 \geq m_1.m_2.m_3$ sobre $\mathcal{Z}^+$.

Obsérvese que para saber que $\mathcal{R}_4 \subseteq_b \mathcal{R}_5$ es necesario comparar los polinomios que expresan las multiplicidades simbólicas de sus tuplas. Precisamente la dificultad que puede llegar a entrañar dicha comparación impide contar con un procedimiento capaz de decidir la b-inclusión de consultas en todos los casos. □

6.6 Semántica de las Consultas Conjuntivas sobre bases de datos LS1 y LS2.

La semántica de las consultas conjuntivas se define en términos de aplicaciones de asignación [8]. Una *aplicación de asignación* de una consulta conjuntiva Q sobre una base de datos $\mathcal{D}$, de tipo LS1 o LS2, es una asignación de las constantes en los hechos de $\mathcal{D}$ a las variables de Q de modo que a cada átomo en el cuerpo de Q se le asigna un hecho que está en $\mathcal{D}$.

Ejemplo 6.6.1 :
Sea Q la consulta conjuntiva: $q(\vec{X}) :- p_1(\vec{X}_1), \ldots, p_n(\vec{X}_n)$.
Sea θ_j una aplicación de asignación de Q hacia la base de datos $\mathcal{D}$ de tipo LS2.

Siendo $\vec{X}_i = (V_1, \ldots, V_{k_i})$, el hecho $p_i(\theta_j(V_1), \ldots, \theta_j(V_{k_i}))$ de la base de datos $\mathcal{D}$ alcanzado el átomo $p_i(\vec{X}_i)$ del cuerpo de Q mediante la aplicación de asignación θ_j, se denota $\theta_j(p_i(\vec{X}_i))$.

Sea $m_{j,i} = \mid \theta_j(p_i(\vec{X}_i)) \mid_{\mathcal{D}}$, $i = 1, \ldots, n$. Entonces el *resultado* que obtiene Q *debido a la aplicación de asignación* θ_j cuando se aplica sobre la base de datos $\mathcal{D}$ es:

$$Q_{\theta_j}(\mathcal{D}) = \{q(\theta_j(\vec{X}); [m_j])\}, \quad m_j = m_{j,1}.m_{j,2}. \quad \cdots \quad .m_{j,n} \quad \square$$

Ejemplo 6.6.2 :
Sea la consulta conjuntiva $Q = q(X,Y) :- r(X,U), r(V,U), r(Y,U)$.
Sea la base de datos $\mathcal{D} = \{\ r(a,b;[2]),\ r(c,b;[3])\}$.
Sea θ la aplicación de asignación de Q a $\mathcal{D}$ siguiente:

$$\theta(X) = a, \quad \theta(U) = b, \quad \theta(V) = a, \quad \theta(Y) = c$$

Entonces:

$$Q_\theta(\mathcal{D}) = \{q(a,c;[2 \times 2 \times 3])\} = \{q(a,c;[12])\}$$

Es fácil de entender que θ asigna $r(a,b)$, que tiene multiplicidad 2, a $r(X,U)$ y a $r(V,U)$, y por otro lado asigna $r(c,b)$ cuya multiplicidad es 3 al átomo $r(Y,U)$. Por tanto la multiplicidad que obtiene la tupla resultado es 12. $\square$

Cuando $\mathcal{D}$ es una base de datos LS2, el producto de los $m_{j,i}$ da lugar a un polinomio. Esto se ilustra en el siguiente ejemplo.

Ejemplo 6.6.3 :
Sea $Q = q(X,Y) : -r(X,U), r(V,U), r(Y,U)$.
Sea $\mathcal{D}'$ una base de datos LS2 con una única relación r que tiene la siguiente extensión:

r
$(a,b;[m_1])$
$(c,b;[m_2])$
$(d,b;[m_3])$

y sea θ la aplicación de asignación de Q a $\mathcal{D}'$ siguiente:

$$\theta(X) = a, \quad \theta(U) = b, \quad \theta(V) = c, \quad \theta(Y) = d$$

Entonces, es evidente que para calcular la multiplicidad de la tupla $q(a,d)$, derivada por la consulta Q cuando se aplica sobre la base de datos $\mathcal{D}'$ mediante la aplicación de asignación θ, denotada $Q_\theta(\mathcal{D}')$, es necesario multiplicar las multiplicidades de las tuplas de $\mathcal{D}'$ alcanzadas por los átomos de Q mediante esa aplicación de asignación. Por tanto:

$$Q_\theta(\mathcal{D}') = \{q(a,d;[m_1.m_2.m_3])\}$$

□

Obsérvese que podría ocurrir que la misma tupla t fuese obtenida por diferentes aplicaciones de asignación $\theta_1, ..., \theta_k \quad i \leq k \leq l$, siendo l el número de diferentes aplicaciones de asignación posibles entre la consulta Q y la extensión de la base de datos $\mathcal{D}$ en un momento determinado. Como ya se expuso en el apartado 4.3.2, cuando la misma tupla t puede ser obtenida por Q mediante diferentes aplicaciones de asignación sobre $\mathcal{D}$ la multiplicidad final de dicha tupla, denotada $|t|_{Q(\mathcal{D})}$, se calcula sumando cada una de las multiplicidades $m_j \quad 1 \leq j \leq k$ que la tupla t obtiene con cada una de las diferentes aplicaciones $\theta_j, \quad 1 \leq j \leq k$ que la producen. Es decir:

$$|t|_{Q(\mathcal{D})} = \sum_{j=1}^{j=k} m_j$$

En el apartado 4.3.2 se presenta un ejemplo de cálculo de la multiplicidad de una tupla t que puede ser obtenida, por una consulta conjuntiva aplicada a una base de datos LS1, mediante diferentes aplicaciones de asignación. Aquí se presenta un ejemplo en el que la base de datos es de tipo LS2.

Ejemplo 6.6.4 :
Sea la consulta conjuntiva $Q = q(X,Y) : -r(X,Y), p(X,U), p(Y,V)$ Sea $\mathcal{D}$ la base de datos LS2 siguiente:

r	p
$(a,a;[m_{r1}])$	$(a,b;[m_{p1}])$
	$(a,a;[m_{p2}])$

Las aplicaciones de asignación posibles de Q a la base de datos son cuatro y se representan en la siguiente tabla:

θ_1	θ_2	θ_3	θ_4
$\theta_1(X) = a$	$\theta_2(X) = a$	$\theta_3(X) = a$	$\theta_4(X) = a$
$\theta_1(Y) = a$	$\theta_2(Y) = a$	$\theta_3(Y) = a$	$\theta_4(Y) = a$
$\theta_1(U) = b$	$\theta_2(U) = b$	$\theta_3(U) = a$	$\theta_4(U) = a$
$\theta_1(V) = a$	$\theta_2(V) = b$	$\theta_3(V) = a$	$\theta_4(V) = b$

Con cada una de esas aplicaciones de asignación se obtiene el mimo hecho derivado $q(a,a)$ en la cabecera de la consulta pues en las cuatro aplicaciones de asignación el valor que se le asigna a las variables X e Y es a. Ahora bien, en cada aplicación de asignación el hecho derivado $q(a,a)$ es obtenido con una diferente multiplicidad m_j tal y como se expresa el la siguiente tabla:

θ_j	m_j
θ_1	$m_1 = m_{r1}.m_{p1}.m_{p2}$
θ_2	$m_2 = m_{r1}.m_{p1}^2$
θ_3	$m_3 = m_{r1}.m_{p2}^2$
θ_4	$m_4 = m_{r1}.m_{p2}.m_{p1}$

Por tanto la multiplicidad final con la que se obtiene el hecho derivado $q(a,a)$ cuando se aplica la consulta Q sobre la base de datos $\mathcal{D}$ será:

$$|q(a,a)|_{Q(\mathcal{D})} = \sum_{j=1}^{j=4} m_j$$

Dicha multiplicidad vendrá expresada por el siguiente polinomio:

$$|q(a,a)|_{Q(\mathcal{D})} = m_{r1}.m_{p1}m_{p2} + m_{r1}.m_{p1}^2 + m_{r1}.m_{p2}^2 + m_{r1}.m_{p2}.m_{p1}$$

Es decir:

$$|q(a,a)|_{Q(\mathcal{D})} = 2.m_{r1}m_{p1}m_{p2} + m_{r1}.m_{p1}^2 + m_{r1}.m_{p2}^2 \quad \square$$

6.6.1 Aplicación de una consulta conjuntiva a una base de datos LS1 o LS2.

Supongamos ahora que hay l aplicaciones de asignación, $\theta_1, \ldots, \theta_l$, de Q a la base de datos LS2 denominada $\mathcal{D}$.

El *resultado de la consulta Q sobre $\mathcal{D}$*, denotado $Q(\mathcal{D})$, viene dado por

$$Q(\mathcal{D}) = Q_{\theta_1}(\mathcal{D}) \sqcup \cdots \sqcup Q_{\theta_l}(\mathcal{D})$$

Ejemplo 6.6.5 :
Sea $Q = q(X,Y) : -r(X,U), r(V,U), r(Y,U)$.
Sea $\mathcal{D}$ la base de datos LS1 con una única relación r cuya extensión es:

r
$(a, b; [2])$
$(c, b; [3])$

Puede verificarse que hay ocho aplicaciones de asignación de Q a $\mathcal{D}$; más adelante se muestran tres de ellas. Así:

$$Q(\mathcal{D}) = Q_{\theta_1}(\mathcal{D}) \sqcup Q_{\theta_2}(\mathcal{D}) \sqcup \cdots \sqcup Q_{\theta_8}(\mathcal{D})$$

$$=$$

$$\{q(a,c;[12])\} \sqcup \{q(a,c;[18])\} \sqcup \{q(a,a;[8])\} \sqcup \{q(a,a;[12])\} \sqcup$$

$$\{q(c,a;[12])\} \sqcup \{q(c,a;[18])\} \sqcup \{q(c,c;[18])\} \sqcup \{q(c,c;[27])\}$$

$$=$$

$$\{q(a,c;[30]), q(c,a;[30]), q(a,a;[20]), q(c,c;[45])\}$$

Las tuplas $q(a, c; [12])$, $q(a, c; [18])$, y $q(c, c; [27])$ corresponden a las siguientes aplicaciones de asignación de la consulta Q a la base de datos $\mathcal{D}$:

θ_1	θ_2	θ_8
$\theta_1(X) = a$	$\theta_2(X) = a$	$\theta_8(X) = c$
$\theta_1(U) = b$	$\theta_2(U) = b$	$\theta_8(U) = b$
$\theta_1(V) = a$	$\theta_2(V) = c$	$\theta_8(V) = c$
$\theta_1(Y) = c$	$\theta_2(Y) = c$	$\theta_8(Y) = c$

□

El siguiente ejemplo muestra la aplicación de una consulta conjuntiva a una base de datos LS2.

Ejemplo 6.6.6 :
Sea $Q = q(X, Y) : -r(X, U), r(V, U), r(Y, U)$
Sea $\mathcal{D}$ la base de datos LS2 que se muestra a continuación:

r
$(a, b; [m_1])$
$(c, b; [m_2])$

El resultado de aplicar la consulta Q sobre la base de datos LS2 denotada $\mathcal{D}$ es el siguiente conjunto de tuplas cada una de las cuales tiene asociada su multiplicidad:

$$
\begin{aligned}
Q(\mathcal{D}) = \{ \quad & q(a, c; [m_1^2.m_2 + m_1.m_2^2]), \\
& q(a, a; [m_1^3 + m_1^2.m_2]), \\
& q(c, a; [m_1^2.m_2 + m_1.m_2^2]), \\
& q(c, c; [m_1.m_2^2 + m_2^3])\}
\end{aligned}
$$

□

7

BASES DE DATOS CANÓNICAS.

7.1 Introducción.

En esta tesis se presenta un procedimiento para realizar la comparación de dos consultas conjuntivas decidiendo si efectivamente una está b-incluida en la otra. Como ya se expuso, una consulta conjuntiva Q está *b-incluida* en otra Q', denotado $Q \leq_b Q'$, si y sólo si para todas las bases de datos $\mathcal{D}$ se cumple que $Q(\mathcal{D}) \subseteq_b Q'(\mathcal{D})$. Es decir:

$$Q \leq_b Q' \iff \forall \mathcal{D} \;\; Q(\mathcal{D}) \subseteq_b Q'(\mathcal{D})$$

El procedimiento de comparación que se presenta en esta tesis es válido para cualquier par de consultas conjuntivas. Dicho procedimiento se basa en el hecho, que se demuestra más adelante, de que para saber si dos consultas conjuntivas Q y Q' están efectivamente b-incluidas la una en la otra, es decir $Q \leq_b Q'$, no es necesario comparar los resultados que obtienen sobre las infinitas bases de datos $\mathcal{D}$ posibles tal y como exige la definición de b-inclusión. Por el contrario, se demuestra que se puede probar la b-inclusión de dos consultas conjuntivas Q y Q' usando un pequeño conjunto *finito* de bases de datos canónicas construidas a partir del cuerpo de la consulta Q. Este conjunto se ha denominado el *Conjunto de Bases de Datos Canónicas* de Q, y se ha denotado $\mathcal{CBDC}(Q)$.

Por tanto, mediante el teorema principal que se presenta más adelante, se demuestra que:

$$Q \leq_b Q' \iff \forall d \in \mathcal{CBDC}(Q) \quad Q(d) \subseteq_b Q'(d)$$

Aunque este procedimiento es válido para cualquier par de consultas conjuntivas, lo cierto es que no contamos actualmente con un procedimiento que permita determinar siempre si $Q(d) \subseteq_b Q'(d)$ debido a la dificultad que entraña la comparación de los polinomios que expresarán las multiplicidades simbólicas. Debe destacarse sin embargo que *siempre* es posible realizar dicha comparación cuando las dos consultas cumplen la condición suficiente[1] de Chauhuri y Vardi expuesta en [8] de que exista una aplicación cobertora.

Es posible, por tanto, afirmar que el procedimiento que se expone en esta tesis para estudiar la b-inclusión de consultas conjuntivas, aunque no permite decidir la b-inclusión de consultas en todos los casos, permite sin embargo, no sólo decidirla en *todos* aquellos casos en los que lo permite la condición suficiente[1] de Chauhuri y Vardi, [8] sino también, en muchos otros casos de parejas de consultas que incumplen dicha condición y de las que, por tanto, no era posible decidir su b-inclusión ateniéndose tan sólo a los resultados de Chauhuri y Vardi en [8].

En el resto de este capítulo se presenta:

- Un ejemplo en el que se muestra el conjunto de bases de datos canónicas $\mathcal{CBDC}(Q)$ construidas a partir de una consulta conjuntiva Q.

- El método de cálculo del número de bases de datos canónicas que habrá en $\mathcal{CBDC}(Q)$, es decir, un método de cálculo de la cardinalidad del conjunto de bases de datos canónicas de una consulta Q determinada.

- La definición formal de bases de datos canónicas.

- Un algoritmo que construye el conjunto de bases de datos canónicas $\mathcal{CBDC}(Q)$ para una consulta conjuntiva Q.

[1]Expuesta en esta tesis como Proposición 5.3.1

7.2 Ejemplo de construcción del Conjunto de Bases de Datos Canónicas de una consulta conjuntiva Q dada. $\mathcal{CBDC}(Q)$.

El objetivo de la construcción de un conjunto de bases de datos canónicas, para una consulta Q específica, es cubrir todas las aplicaciones de asignación posibles entre esa consulta y cualquier base de datos. El siguiente ejemplo lo ilustra. Después del ejemplo, se da una definición formal de $\mathcal{CBDC}(Q)$.

Ejemplo 7.2.1 :
Considérese la consulta:

$$Q = q(X, Y, Z) : -r(X, U), r(U, Z), p(U, Y)$$

Se deben usar los átomos del cuerpo de Q para construir el conjunto de bases de datos canónicas para Q, $\mathcal{CBDC}(Q)$.

Para construir $\mathcal{CBDC}(Q)$ es necesario considerar las diferentes formas posibles de asignar, mediante una aplicación de asignación θ, valores de datos, de cualquier base de datos $\mathcal{D}$, a las variables de Q.

Se emplean las letras A, B, C, D (que representan constantes no interpretadas) para identificar los valores que se pueden asignar a las variables de Q.

A continuación, se listan las distintas aplicaciones posibles cada una de las cuales da lugar a una base de datos canónica d_i. Lógicamente cada d_i representa un elemento de $\mathcal{CBDC}(Q)$ y m_{r1}, m_{r2} y m_{p1} representan las multiplicidades simbólicas de sus hechos correspondientes.

1. Si a cada variable en Q se le asigna un valor diferente, entonces se genera una base de datos canónica, por ejemplo d_1, con el siguiente patrón:

<table>
<tr><th>$\mathbf{d_i}$</th><th colspan="2">Base de Datos</th><th>APLICACIÓN</th></tr>
<tr><td rowspan="2">d_1</td><td>r</td><td>p</td><td rowspan="2">$\theta_1(X) = A$; $\theta_1(Y) = B$;
$\theta_1(Z) = C$; $\theta_1(U) = D$</td></tr>
<tr><td>$(A, D; [m_{r1}])$
$(D, C; [m_{r2}])$</td><td>$(D, B; [m_{p1}])$</td></tr>
</table>

2. A tres variables se les asigna el mismo valor y a la otra un valor diferente:

$\mathbf{d_i}$	**Base de Datos**		**APLICACIÓN**
	r	p	
d_2	$(A, B; [m_{r1}])$ $(B, A; [m_{r2}])$	$(B, A; [m_{p1}])$	$\theta_2(X) = \theta_2(Y) = \theta_2(Z) = A; \quad \theta_2(U) = B$
d_3	$(A, A; [m_{r1}])$ $(A, B; [m_{r2}])$	$(A, A; [m_{p1}])$	$\theta_3(X) = \theta_3(Y) = \theta_3(U) = A; \quad \theta_3(Z) = B$
d_4	$(A, A; [m_{r1}])$	$(A, B; [m_{p1}])$	$\theta_4(X) = \theta_4(Z) = \theta_4(U) = A; \quad \theta_4(Y) = B$
d_5	$(B, A; [m_{r1}])$ $(A, A; [m_{r2}])$	$(A, A; [m_{p1}])$	$\theta_5(Y) = \theta_5(Z) = \theta_5(U) = A; \quad \theta_5(X) = B$

3. A dos variables se les asigna un mismo valor mientras que a las otras dos se les asigna otro valor diferente:

$\mathbf{d_i}$	**Base de Datos**		**APLICACIÓN**
	r	p	
d_6	$(A, B; [m_{r1}])$ $(B, B; [m_{r2}])$	$(B, A; [m_{p1}])$	$\theta_6(X) = \theta_6(Y) = A; \quad \theta_6(Z) = \theta_6(U) = B$
d_7	$(A, B; [m_{r1}])$ $(B, A; [m_{r2}])$	$(B, B; [m_{p1}])$	$\theta_7(X) = \theta_7(Z) = A; \quad \theta_7(Y) = \theta_7(U) = B$
d_8	$(A, A; [m_{r1}])$ $(A, B; [m_{r2}])$	$(A, B; [m_{p1}])$	$\theta_8(X) = \theta_8(U) = A; \quad \theta_8(Y) = \theta_8(Z) = B$

4. A dos variables se les asigna el mismo valor y a cada una de las otras dos se les asignan dos valores diferentes:

$\mathbf{d_i}$	**Base de Datos**		**APLICACIÓN**
	r	p	
d_9	$(A, C; [m_{r1}])$ $(C, B; [m_{r2}])$	$(C, A; [m_{p1}]))$	$\theta_9(X) = \theta_9(Y) = A;$ $\theta_9(Z) = B; \quad \theta_9(U) = C$
d_{10}	$(A, C; [m_{r1}])$ $(C, A; [m_{r2}])$	$(C, B; [m_{p1}])$	$\theta_{10}(X) = \theta_{10}(Z) = A;$ $\theta_{10}(Y) = B; \quad \theta_{10}(U) = C$
d_{11}	$(A, A; [m_{r1}])$ $(A, C; [m_{r2}])$	$(A, B; [m_{p1}])$	$\theta_{11}(X) = \theta_{11}(U) = A;$ $\theta_{11}(Y) = B; \quad \theta_{11}(Z) = C$
d_{12}	$(B, C; [m_{r1}])$ $(C, A; [m_{r2}])$	$(C, A; [m_{p1}])$	$\theta_{12}(Y) = \theta_{12}(Z) = A;$ $\theta_{12}(X) = B; \quad \theta_{12}(U) = C$
d_{13}	$(B, A; [m_{r1}])$ $(A, C; [m_{r2}])$	$(A, A; [m_{p1}])$	$\theta_{13}(Y) = \theta_{13}(U) = A;$ $\theta_{13}(X) = B; \quad \theta_{13}(Z) = C$
d_{14}	$(B, A; [m_{r1}])$ $(A, A; [m_{r2}])$	$(A, C; [m_{p1}])$	$\theta_{14}(Z) = \theta_{14}(U) = A;$ $\theta_{14}(X) = B; \quad \theta_{14}(Y) = C$

5. A las cuatro variables se les asigna el mismo valor, digamos A:

$\mathbf{d_i}$	**Base de Datos**		**APLICACIÓN**
	r	p	
d_{15}	$(A, A; [m_{r1}])$	$(A, A; [m_{p1}])$	$\theta_{15}(X) = \theta_{15}(U) =$ $= \theta_{15}(Y) = \theta_{15}(Z) = A$

Por tanto, hay 15 bases de datos canónicas para Q. Nótese que no se necesitan todas las bases de datos canónicas posibles. Por ejemplo, la aplicación de X e Y a B y U y Z a D produciría una base de datos canónica con las mismas igualdades que d_6 (sería isomórfica a d_6). Con cuatro variables hay 256 bases de datos canónicas posibles diferentes, pero sólo hay 15 que sean distintas (noisomórficas). □

7.3 Cálculo del número de Bases de Datos Canónicas construidas a partir de una consulta Q dada.

Con cuatro variables en el cuerpo de Q hay 256 diferentes posibles bases de datos canónicas que corresponden a 256 variaciones con repetición de 4 elementos tomados de 4 en 4, pero sólo hay 15 que no sean isomórficas entre sí. Es interesante, aunque no necesario, contar con un procedimiento que permita calcular el número de bases de datos canónicas no isomórficas que se pueden construir a partir de una consulta dada, es decir, el número de bases de datos $d_i \in \mathcal{CBDC}(Q)$. A continuación se expone un método de cálculo a tal fin.

Dada una consulta Q con n variables diferentes, es posible calcular el número total de bases de datos canónicas en $\mathcal{CBDC}(Q)$ aplicando la siguiente fórmula:

$$\text{Número de } \mathcal{CBDC} = \sum_{i=1}^{i=k} PE_n^{\vec{X}_i}$$

donde:

n.- Indica el número de variables en la consulta conjuntiva Q.

$\vec{X}_i$.- Es una lista de números que deben sumar n. Hay una lista $\vec{X}_i$ para cada diferente descomposición del número n en una lista de sumandos.

k.- Indica el número de descomposiciones posibles que el número n puede tener. El sumatorio tendrá por tanto k sumandos, uno por cada diferente posible descomposición de n en una lista de números ($\vec{X}_i$) que sumen n. Es decir $1 \leq i \leq k$.

Ejemplo 7.3.1 :
Con $n = 4$ hay 5 posibles listas $\vec{X}_i$, por tanto $k = 5$ y cada una de esas cinco listas ($\vec{X}_i$) de sumandos se representan a continuación:

$$\begin{array}{rcl}\vec{X}_1 & = & 4\\ \vec{X}_2 & = & 3{,}1\\ \vec{X}_3 & = & 2{,}2\\ \vec{X}_4 & = & 2{,}1{,}1\\ \vec{X}_5 & = & 1{,}1{,}1{,}1\end{array}$$

Obsérvese que el orden de los números de la lista no es relevante. Así, una descomposición de $n = 4$ en una lista $\vec{X}_i = 1, 3$, es exactamente la misma descomposición que $\vec{X}_2$.

Con $n = 5$ hay siete posibles listas $\vec{X}_i$, que se muestran a continuación. Ahora $k = 7$.

$$\begin{array}{rcl}\vec{X}_1 & = & 5\\ \vec{X}_2 & = & 4{,}1\\ \vec{X}_3 & = & 3{,}2\\ \vec{X}_4 & = & 3{,}1{,}1\\ \vec{X}_5 & = & 2{,}2{,}1\\ \vec{X}_6 & = & 2{,}1{,}1{,}1\\ \vec{X}_7 & = & 1{,}1{,}1{,}1{,}1\end{array}$$

$PE_n^{\vec{X}_i}$.- Es una clase especial de permutación. Básicamente es la permutación con repetición de n elementos tomados siguiendo lo indicado por la lista $\vec{X}_i$ dividida por un producto de factores. Es decir:

$$PE_n^{\vec{X}_i} = \frac{PR_n^{\vec{X}_i}}{P_{Z_1}.P_{Z_2}...P_{Z_l}}$$

Donde l indica el número de factores en el divisor que es igual al de clases de equivalencia que se establecen en la lista de números $\vec{X}_i$ mediante la relación de igualdad. Es decir, hay tantos factores en el divisor, como *diferentes* grupos de números con igual valor hay en $\vec{X}_i$.

Ejemplo 7.3.2 :
Sea la lista de números $\vec{X}_i = 4, 3, 3, 2, 2, 2, 1, 1$

Entonces la relación de igualdad establece 4 clases de equivalencia, por tanto $l = 4$:

$$(4) \qquad (3,3), \qquad (2,2,2), \qquad (1,1)$$

Cada clase de equivalencia está formada por todos aquellos números de la lista con igual valor. Por tanto cada clase de equivalencia tiene una cardinalidad que denotamos Z_i siendo i el identificador de la clase de equivalencia. Es decir

Clase	Nombre de la clase i	cardinalidad de la clase Z_i
(4)	$i = 1$	$Z_1 = 1$
$(3,3)$	$i = 2$	$Z_2 = 2$
$(2,2,2)$	$i = 3$	$Z_3 = 3$
$(1,1)$	$i = 4$	$Z_4 = 2$

□

Cada factor divisor es la permutación de Z_i denotada P_{Z_i}, siendo Z_i la cardinalidad de la clase i.

Ejemplo 7.3.3 :
Si $n = 13$ y $\vec{X}_i = 3,2,2,2,2,1,1$, la permutación especial $PE_{13}^{\vec{X}_i}$ es la permutación con repetición de 13 elementos tomados siguiendo las indicaciones de $\vec{X}_i$ que se denota $PR_{13}^{3,2,2,2,2,1,1}$, dividida por el producto de $l = 3$ factores, que son:

$$P_{Z_1} = P_1 = 1!, \qquad P_{Z_2} = P_4 = 4!, \qquad P_{Z_3} = P_2 = 2!$$

Por tanto $PE_n^{\vec{X}_i}$ indica las permutaciones con repetición pero considerando la igualdad entre las variables y su valor, en este ejemplo, es:

$$PE_{13}^{3,2,2,2,2,1,1} = \frac{PR_{13}^{3,2,2,2,2,1,1}}{P_{Z_1}\ .\ P_{Z_2}\ .\ P_{Z_3}} = \frac{\frac{13!}{3!.2!.2!.2!.2!.1!.1!.}}{1!\ .\ 4!\ .\ 2!}$$

□

Interesa considerar las igualdades entre las variables ya que no todas las permutaciones interesan para calcular las bases de datos canónicas no isomórficas. Por ello las permutaciones con repetición de las n variables según lo indicado en $(\vec{X}_i)$ se debe dividir por el producto de las permutaciones de los valores que son iguales. Así, para cada grupo de números iguales en $(\vec{X}_i)$ es necesario dividir por las permutaciones de la cardinalidad del grupo.

Ejemplo 7.3.4 :
El valor de $PE_{13}^{9,2,2}$ será el número de permutaciones con repetición de 13 elementos tomados de $9, 2, 2$ dividido por el producto de las permutaciones de 1(porque el 9 está una vez en la lista) y por las permutaciones de 2 (porque el dos está dos veces). Es decir, $l = 2$, puesto que hay dos clases de equivalencia en la lista $\vec{X}_i$ (las clases (9) y $(2, 2)$). Por tanto $Z_1 = 1$ y $Z_2 = 2$ Así:

$$PE_{13}^{9,2,2} = \frac{PR_{13}^{9,2,2}}{P_1 \ . \ P_2} = \frac{\frac{13!}{9! \ 2! \ 2!}}{1! \ . \ 2!}$$

□

Se presenta a continuación un ejemplo completo de cálculo del número de bases de datos canónicas de una consulta conjuntiva Q. Es decir, se presenta un ejemplo de cálculo de la cardinalidad de $\mathcal{CBDC}(Q)$

Ejemplo 7.3.5 :
Dada la consulta de 4 variables siguiente:

$$Q = \quad q(X, Y, Z) : -r(X, U), r(V, Z), p(U, Y)$$

El número de bases de datos canónicas que genera, es decir, la cardinalidad de $\mathcal{CBDC}(Q)$ será:

$$\begin{array}{llll}
PE_4^4 & = \frac{PR_4^4}{P1} & = \frac{4!/4!}{1!} & = 1 \\
PE_4^{3,1} & = \frac{PR_4^{3,1}}{P1.P1} & = \frac{4!/3!1!}{1!1!} & = 4 \\
PE_4^{2,2} & = \frac{PR_4^{2,2}}{P2} & = \frac{4!/2!2!}{2!} & = 3 \\
PE_4^{2,1,1} & = \frac{PR_4^{2,1,1}}{P1.P2} & = \frac{4!/1!1!2!}{1!2!} & = 6 \\
PE_4^{1,1,1,1} & = \frac{PR_4^{1,1,1,1}}{P4} & = \frac{4!/1!1!1!1!}{4!} & = 1
\end{array}$$

Como se ve el número total de bases de datos canónicas en $CDBC(Q)$ será de $1 + 4 + 3 + 6 + 1 = 15$ como ya se mostró en la Sección 7.2. □

7.4 Definición formal de Bases de Datos Canónicas.

Antes de exponer el algoritmo que permite la construcción automática del conjunto de bases de datos canónicas de una consulta Q dada, es necesario realizar algunas definiciones:

7.4.1 Definición de $db(Q)$.

Sea Q una consulta conjuntiva. Entonces, $db(Q)$ es un conjunto de átomos $\{p_i(\vec{X_i})\}$ que forman el cuerpo de Q. Es decir:

$$db(Q) = \{p(V1, ..., V_{k_i}) \ \mid \ p(V1, ..., V_{k_i}) \text{ es un átomo del cuerpo de Q}\}$$

Es decir, $db(Q)$ es el mismo conjunto de átomos del cuerpo de Q denotado $db(Q)$ para diferenciarlo de la consulta Q propiamente dicha.

7.4.2 Q-aplicaciones θ_i.

Se usará $db(Q)$ para construir el $\mathcal{CBDC}(Q)$. La idea consiste en definir un conjunto de aplicaciones θ_i, que denominamos Q-aplicaciones, que se aplicarán a las variables de $db(Q)$, cada Q-aplicación θ_i que se defina, generará una base de datos canónica $d_i \in \mathcal{CBDC}(Q)$.

Sea $V_Q =< V_1, \ldots, V_q >$ una ordenación de las q variables en átomos de $db(Q)$.

Sea $A_Q = \{A_1, \ldots, A_q\}$ un conjunto de q identificadores distintos, es decir: $A_i \neq A_j \quad \text{si} \quad i \neq j, \quad 1 \leq i, j \leq q$. Dichos identificadores A_i representan constantes no interpretadas.

Las Q-aplicaciones le asignan a cada variable en V_Q una constante de A_Q. Así, una *Q-aplicación* θ es una q-tupla $\quad \theta = (A_{i_1}, ..., A_{i_q})$ donde $\quad 1 \leq i_1, i_2, ..., i_q \leq q \quad$. Esta tupla denota la aplicación $\theta(V_1) = A_{i_1}, ..., \theta(V_q) = A_{i_q}$.

Las Q-aplicaciones son "atajos" que se usarán para producir las bases de datos canónicas.

7.4.3 Aplicación de una Q-aplicación sobre un átomo de $db(Q)$.

Sea θ una Q-aplicación. Sea $p(U_1, ..., U_r)$ un átomo de $db(Q)$. Se define la aplicación de θ a $p(U_1, ..., U_r)$ como:

$$\theta(p(U_1, ..., U_r)) = p(\theta(U_1), ..., \theta(U_r))$$

7.4.4 Aplicación de una Q-aplicación sobre $db(Q)$.

La aplicación de θ a $db(Q)$ se define como la base de datos canónica:

$$d \;=\; \theta(db(Q)) = \bigsqcup_{p(U_1,\ldots,U_r)\in db(Q)} \{t \mid \quad t = p(\theta(U_1),...,\theta(U_r);[m])\}$$

Donde m es una variable que no aparece en ningún otro lugar.

Ejemplo 7.4.1 :
Considérese la consulta :

$$Q = q(X,Y,Z) : -r(X,U), r(U,Z), p(U,Y)$$

Sea $V_Q =< X, Y, Z, U >$ una ordenación de las variables en Q
Sea $A_Q = \{A_1, A_2, A_3, A_4\}$.

En este caso $db(Q) = \{r(X,U), r(U,Z), p(U,Y)\}$.

Sea la Q-aplicación θ :

$$\theta = (A_4, A_3, A_4, A_4)$$

Es decir, θ denota las asignaciones:

$$\theta(X) = \theta(Z) = \theta(U) = A_4; \quad y \quad \theta(Y) = A_3$$

Entonces:

$$\theta(db(Q)) = \{r(A_4, A_{4;}[m_1])\} \sqcup \{r(A_4, A_{4;}[m_2])\} \sqcup \{p(A_4, A_3; [m_3])\} =$$

$$\theta(db(Q)) = \{r(A_4, A_{4;}[m_1 + m_2]), p(A_4, A_3; [m_3])\}$$

□

7.4.5 Q-aplicaciones isomórficas.

Dos Q-aplicaciones $\theta_1 = (A_{i_1}, ..., A_{i_q})$ y $\theta_2 = (A_{j_1}, ..., A_{j_q})$ son *isomórficas* si hay dos aplicaciones γ_1 y γ_2 (de A_Q a A_Q) tales que:

$$(\gamma_1(A_{i_1}), ..., \gamma_1(A_{i_q})) = \theta_2 \quad y \quad (\gamma_2(A_{j_1}), ..., \gamma_2(A_{j_q})) = \theta_1$$

Cuando dos aplicaciones son isomórficas, las dos bases de datos canónicas que generan son también isomórficas, y sólo una de ellas pertenecerá al conjunto de bases de datos canónicas $\mathcal{CBDC}(Q)$

Ejemplo 7.4.2 :
La Q-aplicación $\beta = (A_3, A_3, A_1, A_3)$ es isomórfica a la Q-aplicación $\alpha = (A_4, A_4, A_3, A_4)$.

Del mismo modo la Q-aplicación $\theta = (A_1, A_1, A_1, A_1)$ es isomórfica a la Q-aplicación $\tau(A_2, A_2, A_2, A_2)$. □

7.5 Algoritmo para construir $\mathcal{CBDC}(Q)$.

Usando las definiciones anteriores, se muestra un algoritmo que construye las bases de datos canónicas de Q.

Algoritmo para construir $\mathcal{CBDC}(Q)$
Entrada: $Q = q(\vec{X}) : -p_1(\vec{X}_1), ..., p_n(\vec{X}_n)$.
Salida: $\mathcal{CBDC}(Q)$
Procedimiento:
Paso 1: Sea $db(Q)$ la base de datos que representa el cuerpo de Q;
    Sea $V_Q =< V_1, ..., V_q >$ lista ordenada de todas las variables de $db(Q)$;
    Sea $A_Q = \{A_1, ..., A_q\}$ lista de q constantes distintas (no interpretadas)
    $j = 1$;
    $Aplicaciones = \emptyset$;
    //
Paso 2: // Q-aplicaciones (usando la sintaxis de C).
    for($i_1 = 1; i_1 \leq q; i_1$++)
      for($i_2 = 1; i_2 \leq q; i_2$++)
        ⋮
        for($i_q = 1; i_q \leq q; i_q$++){
          $\theta_j = (A_{i_1}, A_{i_2}, ..., A_{i_q})$;
          if $(\nexists\ una\ Q - aplicación \in Aplicaciones\ \ isomórfica\ a\ \theta_j)${

$Aplicaciones = Aplicaciones \cup \{\theta_j\}$;
j++;
}
...
}
//
Paso 3: // Ahora usando las Q-aplicaciones se generan
// las $j-1$ bases de datos de $\mathcal{CBDC}(Q)$
//
for($l = 1; l < j; l$++)
$d_l = \theta_l(db(Q))$;
Paso 4: devuelve $\{d_1, ..., d_{j-1}\}$;

Ejemplo 7.5.1 :
Sea la consulta conjuntiva:

$$Q \quad = \quad q(X,Y) : -r(X,Y), p(X,U), p(Y,V)$$

Sea $V_Q = < X, Y, U, V >$.
Sea $A_Q = \{A, B, C, D\}$.

Como ya se ha mostrado, con cuatro variables hay 256 Q-aplicaciones diferentes, pero sólo hay 15 θ-aplicaciones no isomórficas. Las 15 bases de datos canónicas correspondientes se detallan en la siguiente tabla:

$d_i \in \mathcal{CBDC}(Q)$	
r	p
$(A, B; [m_{r1}])$	$(A, C; [m_{p1}])$
	$(B, D; [m_{p2}])$
$(A, A; [m_{r1}])$	$(A, A; [m_{p3}])$
$(A, A; [m_{r1}])$	$(A, B; [m_{p1}])$
	$(A, A; [m_{p2}])$
$(A, A; [m_{r1}])$	$(A, A; [m_{p1}])$
	$(A, B; [m_{p2}])$
$(B, A; [m_{r1}])$	$(B, A; [m_{p1}])$
	$(A, A; [m_{p2}])$
$(A, B; [m_{r1}])$	$(A, A; [m_{p1}])$
	$(B, A; [m_{p2}])$
$(A, A; [m_{r1}])$	$(A, B; [m_{p3}])$
$(A, B; [m_{r1}])$	$(A, A; [m_{p1}])$
	$(B, B; [m_{p2}])$
$(A, B; [m_{r1}])$	$(A, B; [m_{p1}])$
	$(B, A; [m_{p2}])$
$(A, A; [m_{r1}])$	$(A, B; [m_{p1}])$
	$(A, C; [m_{p2}])$
$(A, B; [m_{r1}])$	$(A, A; [m_{p1}])$
	$(B, C; [m_{p2}])$
$(A, B; [m_{r1}])$	$(A, C; [m_{p1}])$
	$(B, A; [m_{p2}])$
$(B, A; [m_{r1}])$	$(B, A; [m_{p1})]$
	$(A, C; [m_{p2}])$
$(B, A; [m_{r1}])$	$(B, C; [m_{p1}])$
	$(A, A; [m_{p2}])$
$(B, C; [m_{r1}])$	$(B, A; [m_{p1}])$
	$(C, A; [m_{p2}])$

La primera base de datos canónica en esa tabla corresponde a la Q-aplicación (A, B, C, D); la segunda base de datos canónica corresponde a la Q-aplicación (A, A, A, A).$\square$

8

COMPROBACIÓN DE LA B-INCLUSIÓN.

8.1 Introducción.

Sean dos consultas conjuntivas cualesquiera Q y Q', donde:

$$Q = q(\vec{X}) :- p_1(\vec{X}_1), \ldots, p_n(\vec{X}_n)$$
$$Q' = q(\vec{X}) :- p_1(\vec{Z}_1), \ldots, p_l(\vec{Z}_l)$$

Es decir, se asume que la cabeza de Q' y la de Q están definidas con el mismo nombre de predicado q.

Lo que se desea probar es que:

$$Q \leq_b Q' \iff \forall d \in \mathcal{CBDC}(Q)\ Q(d) \subseteq_b Q'(d))$$

En el apartado siguiente se demuestra un lema y se introducen algunas definiciones que son necesarias para probar el teorema principal.

8.2 Lema y definiciones previas.

Comenzaremos por probar un lema que resultará más tarde necesario. Antes de enunciar dicho lema es necesario familiarizarse con los elementos que intervendrán en su enunciado. Dichos elementos son:

La consulta conjuntiva $Q = q(\vec{X}) : -p_1(\vec{X}_1), \ldots, p_n(\vec{X}_n)$**.** Obsérvese que cada $p_i(\vec{X}_i)$ es un átomo del cuerpo de Q cuyas variables se asume que son $\vec{X}_i = Y_{i_1}, \cdots Y_{i_r}$.

La aplicación de asignación θ**.** Dicha aplicación de asignación le hace corresponder a cada átomo del cuerpo de Q una tupla de una base de datos $\mathcal{D}$.

La subbolsa de hechos de $\mathcal{D}$ **que denotamos** SD**.** Dicha subbolsa SD está formada por aquellas tuplas de $\mathcal{D}$ que la aplicación θ hace corresponder a los átomos del cuerpo de Q, es decir:

$$SD = \{p_i(\theta(Y_{i_1}), \ldots, \theta(Y_{i_r}); [c_i]) \; | p_i(Y_{i_1}, \ldots, Y_{i_r}) \text{ es un átomo de } \; Q$$

Nótese que c_i es la multiplicidad del hecho $p_i(\theta(Y_{i_1}), \ldots, \theta(Y_{i_r}))$ de $\mathcal{D}$ sobre el que se aplica el átomo $p_i(\vec{X}_i)$ del cuerpo de Q.

Lema 8.2.1 *Sea Q la consulta conjuntiva:*

$$Q = q(\vec{X}) : -p_1(\vec{X}_1), \ldots, p_n(\vec{X}_n)$$

Sea θ una aplicación de asignación de Q a una base de datos $\mathcal{D}$. Sea SD la siguiente subbolsa de $\mathcal{D}$:

$$SD = \{p_i(\theta(Y_{i_1}), \ldots, \theta(Y_{i_r}); [c_i]\}$$

Donde $p_i(\vec{X}_i)$ es un átomo del cuerpo de Q y $\vec{X}_i = Y_{i_1}, ..., Y_{i_r}$. Entonces existe una base de datos canónica $d_i \in \mathcal{CBDC}(Q)$ que es isomórfica a la subbolsa SD. Es decir:

$$\exists d_i \in \mathcal{CBDC}(Q) \;\; tal \; que \quad d_i \quad isomórfica \;\; a \;\; SD$$

Demostración:

Obsérvese que SD es la subbolsa de $\mathcal{D}$ sobre la que se aplica Q mediante la aplicación de asignación θ, es decir, sobre la que se aplica el cuerpo de Q que es isomórfico a $db(Q)$. Recuérdese que cada base de datos canónica

$d \in \mathcal{CBDC}(Q)$ se obtiene también mediante una aplicación de $db(Q)$ a un conjunto de constantes no interpretadas A_Q. Existen diferentes casos a analizar dependiendo de las igualdades generadas por θ entre las variables de Q.

Asúmase que θ asignó a cada variable de Q el mismo valor constante a de $\mathcal{D}$. En el conjunto de base de datos canónicas $\mathcal{CBDC}(Q)$ existe, por cómo se construyó, una base de datos, digamos d_1, que fue obtenida por una Q-aplicación que le asignó a cada variable en $db(Q)$ la misma constante no interpretada, digamos $A \in A_Q$. Es evidente que SD y d_1 son isomórficos ya que si en cada tupla $p_i(A, \ldots, A; [m_i])$ en d_1 remplazamos A por a y m_i por la multiplicidad del hecho $p_i(a, \ldots, a)$ de SD, que denotamos c_i, las dos bases de datos SD y d_1 quedan idénticas. Esto prueba el lema para este caso.

Asúmase ahora que θ le asignó a todas las variables del cuerpo de Q, excepto a una, digamos Y, la misma contante de $\mathcal{D}$, digamos a, mientras que a la variable Y le asignó la constante b. Por construcción de $\mathcal{CBDC}(Q)$ se sabe que debe haber una base de datos canónica, digamos d_2, que se construyó asignándole a todas las variables de $db(Q)$, excepto a la variable Y, la misma constante no interpretada de A_Q. Supongamos que la constante $A \in A_Q$ es la que la Q-aplicación asignó a todas las variables de $db(Q)$ excepto a Y, a la que asignó la constante B. Es evidente que SD y d_2 son isomórficas ya que con un renombramiento adecuado de las constantes de d_2 y con la asignación del adecuado valor a su multiplicidad simbólica se convierten en base de datos idénticas.

Este mismo razonamiento puede hacerse para todas las posibles igualdades entre las constantes de $\mathcal{D}$ que θ asigne a las variables del cuerpo de Q. Esto es evidente a partir de cómo se construyen las bases de datos canónicas $d_i \in \mathcal{CBDC}(Q)$. Recuérdese que las igualdades entre las constantes en los hechos de las bases de datos canónicas cubren todas las igualdades posibles que pueden darse entre las variables de los átomos del cuerpo de Q cuando se aplican sobre cualquier base de datos $\mathcal{D}$. $\square$

8.2.1 Definición de θ_d y t_d.

Para toda base de datos canónica $d \in \mathcal{CBDC}(Q)$ se define θ_d como la Q-aplicación usada para construir dicha base de datos d a partir de $db(Q)$.

Por otro lado se define t_d como el hecho derivado por la consulta Q, aplicada sobre $d \in \mathcal{CBDC}(Q)$, cuando a cada variable de la cabeza de Q se le asigna un valor mediante la Q-aplicación θ_d, es decir, mediante la Q-aplicación que dio lugar a la base de datos canónica d. Por tanto:

$$t_d \;=\; \theta_d(q(\vec{X}))$$

Ejemplo 8.2.1 :
Dada la consulta $Q \;=\; q(X,Y) \;:-\; r(X,U),\; r(V,U),\; r(Y,U)$ Una de sus bases de datos canónicas es:

$$d = \{r(A,B;[m_{r1}]),\quad r(C,B;[m_{r2}]),\quad r(D,B;[m_{r3}])\}$$

La Q-aplicación θ usada para construir d, denotada θ_d es:

$$\theta_d(X) = A,\quad \theta_d(Y) = D,\quad \theta_d(U) = B,\quad \theta_d(V) = C$$

Por tanto, en esta base de datos el hecho derivado por la consulta Q cuando se aplica sobre la base de datos canónica d mediante la aplicación de asignación θ_d es:

$$t_d \;=\; \theta_d(q(X,Y)) \;=\; q(A,D)$$

Aplicando Q sobre d, el hecho t_d puede obtenerse de tres formas diferentes que corresponden a las tres aplicaciones de asignación siguientes de la consulta Q sobre d:

θ_1	θ_2	θ_3
$\theta_1(X) = A$	$\theta_2(X) = A$	$\theta_3(X) = A$
$\theta_1(U) = B$	$\theta_2(U) = B$	$\theta_3(U) = B$
$\theta_1(V) = C$	$\theta_2(V) = A$	$\theta_3(V) = D$
$\theta_1(Y) = D$	$\theta_2(Y) = D$	$\theta_3(Y) = D$

Cada una de las tres aplicaciones de asignación θ_i obtiene el hecho $t_d = q(A,D)$ con una multiplicidad m_j posiblemente diferente. La multiplicidad final del hecho t_d derivado por la consulta Q cuando se aplica sobre la base de datos canónica d, denotada $|t_d|_{Q(d)}$, se obtiene mediante la suma de esa tres multiplicidades m_j.

Por tanto la multiplicidad final del hecho t_d derivado por la consulta Q cuando se aplica sobre la base de datos canónica d será:

$$t_d = q(A,D;[m_{r1}.m_{r2}.m_{r3} + m_{r1}^2.m_{r3} + m_{r1}.m_{r3}^2])$$

Obsérvese que la multiplicidad final del hecho derivado t_d queda representada por el polinomio siguiente:

$$|t_d|_{Q(d)} = m_{r1}.m_{r2}.m_{r3} + m_{r1}^2.m_{r3} + m_{r1}.m_{r3}^2$$

Cada sumando de la multiplicidad procede de una de las aplicaciones de asignación θ_i como puede comprobarse fácilmente.

□

8.3 Demostración del Teorema Principal.

El teorema dice que $Q \leq_b Q'$ exactamente cuando para cada base de datos canónica $d \in \mathcal{CBDC}(Q)$, se cumple que t_d está en $Q'(d)$ y la multiplicidad de t_d en $Q'(d)$ denotada $|t_d|_{Q'(d)}$ no es menor que la multiplicidad de t_d en $Q(d)$ denotada $|t_d|_{Q(d)}$

Nótese que tanto $|t_d|_{Q(d)}$ como $|t_d|_{Q'(d)}$ son polinomios por lo que su comparación no es trivial.

Teorema 8.3.1 *Sean Q y Q' dos consultas conjuntivas. Entonces*

$$Q \leq_b Q' \iff \forall d \in \mathcal{CBDC}(Q) \quad |t_d|_{Q(d)} \leq |t_d|_{Q'(d)}$$

Demostración:

Parte SÓLO SI:

Si $\exists d \in \mathcal{CBDC}(Q)$ tal que $|t_d|_{Q(d)} \not\leq |t_d|_{Q'(d)}$, podemos usar esa base de datos canónica d para construir un contra-ejemplo de $Q \leq_b Q'$.

Parte SI: :

Asumiendo que $|t_d|_{Q(d)} \leq |t_d|_{Q'(d)} \quad \forall d \in \mathcal{CBDC}(Q)$, se quiere demostrar que $Q \leq_b Q'$.

Sea $\mathcal{D}$ una base de datos LS1 arbitraria y sea u un hecho cualquiera en $Q(\mathcal{D})$.

Para demostrar $Q \leq_b Q'$ se debe demostrar $|u|_{Q(\mathcal{D})} \leq |u|_{Q'(\mathcal{D})}$

Sean $\theta_1, ..., \theta_l$ las aplicaciones de asignación de Q a $\mathcal{D}$ que obtienen el hecho u.
Sea SD_i la subbolsa de hechos de $\mathcal{D}$ asignados a los átomos del cuerpo de Q mediante la aplicación de asignación θ_i.

Evidentemente, si con cada aplicación de asignación θ_i ocurre que, la multiplicidad con la que Q obtiene el hecho u, desde la subbola SD_i definida por tal aplicación de asignación, es menor o igual que la multiplicidad con la que la consulta Q' obtine el hecho u desde la misma subbolsa SD_i, entonces se cumplirá que $Q \leq_b Q'$. Es decir:

$$\forall i,\ 1 \leq i \leq l \quad |u|_{Q(SD_i)} \leq |u|_{Q'(SD_i)} \Longrightarrow Q \leq_b Q'$$

Asumamos que, por el contrario, $\exists j,\ 1 \leq j \leq l$, tal que $|u|_{Q(S\mathcal{D}_j)} \not\leq |u|_{Q'(S\mathcal{D}_j)}$. Se sabe además, por el Lema 8.2.1, que $S\mathcal{D}_j$ es isomórfica a alguna base de datos $d_i \in \mathcal{CBDC}(Q)$. Entonces, debido a que $|u|_{Q(S\mathcal{D}_j)} \not\leq |u|_{Q'(S\mathcal{D}_j)}$ y a que existe una base de datos canónica d_i que es isomórfica a $S\mathcal{D}_j$, se concluye que $|t_{d_i}|_{Q(d_i)} \not\leq |t_{d_i}|_{Q'(d_i)}$. Pero esto es una contradicción a la hipótesis de esta parte del teorema.

Por tanto, es imposible que $|u|_{Q(\mathcal{SD}_j)} \not\leq |u|_{Q'(\mathcal{SD}_j)}$ lo que completa esta parte de la demostración. $\square$

Corolario 8.3.1 *Siendo Q y Q' dos consultas conjuntivas siempre se cumplirá que:*

$$Q \leq_b Q' \Longleftrightarrow \forall d \in \mathcal{CBDC}(Q) \quad (Q(d) \subseteq_b Q'(d))$$

Demostración:

Parte SÓLO SI:

Si existe una base de datos canónica $d \in \mathcal{CBDC}(Q)$ tal que $Q(d) \not\subseteq_b Q'(d)$, entonces podemos usar la base de datos d para construir un contra-ejemplo de $Q \leq_b Q'$.

Parte SI: :

Se asume que $\forall d \in \mathcal{CBDC}(Q)$ $(Q(d) \subseteq_b Q'(d))$.

Por construcción, t_d está en $Q(d)$ y por la hipótesis está además en $Q'(d)$ con una multiplicidad mayor o igual a su multiplicidad en $Q(d)$. Por tanto:

$$\forall d \in \mathcal{CBDC} \quad (|t_d|_{Q(d)} \leq |t_d|_{Q'(d)})$$

Usando ahora el teorema anterior se concluye que $Q \leq_b Q'$.$\square$

8.4 Procedimiento para demostrar la b-inclusión de consultas usando $\mathcal{CBDC}(Q)$.

Se puede comprobar si se cumple o no que $Q \leq_b Q'$ usando las bases de datos canónicas de Q construidas mediante el procedimiento visto anteriormente.

El proceso puede no devolver ninguna respuesta porque, en general, no sabemos si es siempre posible comparar los dos polinomios que expresan las multiplicidades de los hechos $t_{Q(d)}$ y $t_{Q'(d)}$, es decir, no siempre es posible saber si se cumple o no que $(|t_{d_l}|_{Q(d_l)} \not\leq |t_{d_l}|_{Q'(d_l)})$. Dicha comparación se realiza en el Paso 2 del procedimiento que se describe a continuación. En ocasiones la comparación es sencilla pero no sabemos si *siempre podrá hacerse*

Procedimiento para demostrar $Q \leq_b Q'$ usando $\mathcal{CBDC}(Q)$

Entrada: $Q = q(\vec{X}) : -p_1(\vec{X}_1), ..., p_n(\vec{X}_n)$
$Q' = q(\vec{Y}) : -g_1(\vec{Y}_1), ..., g_m(\vec{Y}_m)$
Salida: SI, NO, o no hay salida

Paso 1: // Llamar al algoritmo que construye $\mathcal{CBDC}(Q)$
Sean $d_1, ..., d_j$ las Bases de Datos de $\mathcal{CBDC}(Q)$
Sean $\theta_1, ..., \theta_j$ las Q-aplicaciones usadas para construir $d_1, ..., d_j$

Paso 2: for($l = 1; l \leq j; l$++){
Sea $t_{d_l} = \theta_l(q(\vec{X}))$;
//
// Si sobre $\mathcal{Z}^+$ la multiplicidad de t_{d_l} en $Q(d_l)$ no es menor
//o igual que la multiplicidad de t_{d_l} en $Q'(d_l)$ devuelve "NO"
//
si $(|t_{d_l}|_{Q(d_l)} \not\leq |t_{d_l}|_{Q'(d_l)})$ devuelve NO;
}
Paso 3: devuelve SI;

8.5 Ejemplos de comprobación de la b-inclusión.

Se presentan a continuación dos ejemplos de utilización del procedimiento descrito. En el primero de ellos existe efectivamente b-inclusión entre las consultas mientras que en el segundo no.

Ejemplo 8.5.1 :
Consideremos las siguientes consultas:

$Q = q(X,Y) : -r(X,Y), p(X,U), p(Y,V).$
$Q' = q(X,Y) : -r(X,Y), p(Z,N), p(Z,M).$

Las bases de datos canónicas de Q se muestran en la primera columna de la siguiente tabla. En la segunda y tercera columnas, que están etiquetadas $|t_d|_{Q(d)}$ y $|t_d|_{Q'(d)}$ respectivamente, se representan las multiplicidades simbólicas con las que cada una de las consultas obtiene el hecho t_d.

$d \in \mathcal{CBDC}$		$\|t_d\|_{Q(d)}$	$\|t_d\|_{Q'(d)}$
r	p		
$(A,B;[m_{r1}])$	$(A,C;[m_{p1}])$ $(B,D;[m_{p2}])$	$m_{r1}m_{p1}m_{p2}$	$m_{r1}m_{p1}^2 + m_{r1}m_{p2}^2$
$(A,A;[m_{r1}])$	$(A,A;[m_{p3}])$	$m_{r1}m_{p3}^2$	$m_{r1}m_{p3}^2$
$(A,A;[m_{r1}])$	$(A,B;[m_{p1}])$ $(A,A;[m_{p2}])$	$m_{r1}m_{p1}m_{p2} + m_{r1}m_{p2}m_{p1} +$ $m_{r1}m_{p1}^2 + m_{r1}m_{p2}^2$	$m_{r1}m_{p1}m_{p2} + m_{r1}m_{p2}m_{p1} +$ $m_{r1}m_{p1}^2 + m_{r1}m_{p2}^2$
$(A,A;[m_{r1}])$	$(A,A;[m_{p1}])$ $(A,B;[m_{p2}])$	$m_{r1}m_{p1}m_{p2} + m_{r1}m_{p2}m_{p1} +$ $m_{r1}m_{p1}^2 + m_{r1}m_{p2}^2$	$m_{r1}m_{p1}m_{p2} + m_{r1}m_{p2}m_{p1} +$ $m_{r1}m_{p1}^2 + m_{r1}m_{p2}^2$
$(B,A;[m_{r1}])$	$(B,A;[m_{p1}])$ $(A,A;[m_{p2}])$	$m_{r1}m_{p1}m_{p2}$	$m_{r1}m_{p1}^2 + m_{r1}m_{p2}^2$
$(A,B;[m_{r1}])$	$(A,A;[m_{p1}])$ $(B,A;[m_{p2}])$	$m_{r1}m_{p1}m_{p2}$	$m_{r1}m_{p1}^2 + m_{r1}m_{p2}^2$
$(A,A;[m_{r1}])$	$(A,B;[m_{p3}])$	$m_{r1}m_{p3}^2$	$m_{r1}m_{p3}^2$
$(A,B;[m_{r1}])$	$(A,A;[m_{p1}])$ $(B,B;[m_{p2}])$	$m_{r1}m_{p1}m_{p2}$	$m_{r1}m_{p1}^2 + m_{r1}m_{p2}^2$
$(A,B;[m_{r1}])$	$(A,B;[m_{p1}])$ $(B,A;[m_{p2}])$	$m_{r1}m_{p1}m_{p2}$	$m_{r1}m_{p1}^2 + m_{r1}m_{p2}^2$
$(A,A;[m_{r1}])$	$(A,B;[m_{p1}])$ $(A,C;[m_{p2}])$	$m_{r1}m_{p1}m_{p2} + m_{r1}m_{p2}m_{p1} +$ $m_{r1}m_{p1}^2 + m_{r1}m_{p2}^2$	$m_{r1}m_{p1}m_{p2} + m_{r1}m_{p2}m_{p1} +$ $m_{r1}m_{p1}^2 + m_{r1}m_{p2}^2$
$(A,B;[m_{r1}])$	$(A,A;[m_{p1}])$ $(B,C;[m_{p2}])$	$m_{r1}m_{p1}m_{p2}$	$m_{r1}m_{p1}^2 + m_{r1}m_{p2}^2$
$(A,B;[m_{r1}])$	$(A,C;[m_{p1}])$ $(B,A;[m_{p2}])$	$m_{r1}m_{p1}m_{p2}$	$m_{r1}m_{p1}^2 + m_{r1}m_{p2}^2$
$(B,A;[m_{r1}])$	$(B,A;[m_{p1})]$ $(A,C;[m_{p2}])$	$m_{r1}m_{p1}m_{p2}$	$m_{r1}m_{p1}^2 + m_{r1}m_{p2}^2$
$(B,A;[m_{r1}])$	$(B,C;[m_{p1}])$ $(A,A;[m_{p2}])$	$m_{r1}m_{p1}m_{p2}$	$m_{r1}m_{p1}^2 + m_{r1}m_{p2}^2$
$(B,C;[m_{r1}])$	$(B,A;[m_{p1}])$ $(C,A;[m_{p2}])$	$m_{r1}m_{p1}m_{p2}$	$m_{r1}m_{p1}^2 + m_{r1}m_{p2}^2$

Para clarificar la notación, cuando los dos hechos de predicado p de una base de datos son iguales, se representan como un único hecho de multiplicidad m_{p3}, es decir, $m_{p3} = m_{p1} + m_{p2}$.

Obsérvese en este ejemplo que para cada d en $\mathcal{CBDC}(Q)$ la multiplicidad (sobre $\mathcal{Z}^+$) de t_d en $Q'(d)$ siempre es mayor o igual a la obtenida en $Q(d)$; es decir:

$$|t_d|_{Q(d)} \leq |t_d|_{Q'(d)} \quad \forall d \in \mathcal{CBDC}(Q)$$

Por el Teorema Principal, esto significa que $Q \leq_b Q'$. Evidentemente por el Corolario 8.3.1 también se cumplirá que:

$$Q(d) \subseteq_b Q'(d) \quad \forall d \in \mathcal{CBDC}(Q)$$

Por otro lado, usando la definición de b-inclusión también se puede afirmar que:

$$Q(\mathcal{D}) \subseteq_b Q'(\mathcal{D}) \quad \forall \mathcal{D}$$

□

A continuación se muestran un ejemplo en el que se puede apreciar que no se da b-inclusión.

Ejemplo 8.5.2 :
Considérense las siguientes consultas conjuntivas:

$Q = q(X,Y) : -r(X,Y), p(X,U), p(Y,V).$
$Q'' = q(X,Y) : -r(X,Y), p(N,Z), p(M,Z).$

La organización de la tabla es igual a la del ejemplo anterior.

$d \in \mathcal{CBDC}$		$\|t_d\|_{Q(d)}$	$\|t_d\|_{Q\,'(d)}$
r	p		
$(A,B;[m_{r1}])$	$(A,C;[m_{p1}])$ $(B,D;[m_{p2}])$	$m_{r1}m_{p1}m_{p2}$	$m_{r1}m_{p1}^2+m_{r1}m_{p2}^2$
$(A,A;[m_{r1}])$	$(A,A;[m_{p3}])$	$m_{r1}m_{p3}^2$	$m_{r1}m_{p3}^2$
$(A,A;[m_{r1}])$	$(A,B;[m_{p1}])$ $(A,A;[m_{p2}])$	$m_{r1}m_{p1}m_{p2}+m_{r1}m_{p2}m_{p1}+$ $m_{r1}m_{p1}^2+m_{r1}m_{p2}^2$	$m_{r1}m_{p1}^2+m_{r1}m_{p2}^2$
$(A,A;[m_{r1}])$	$(A,A;[m_{p1}])$ $(A,B;[m_{p2}])$	$m_{r1}m_{p1}m_{p2}+m_{r1}m_{p2}m_{p1}+$ $m_{r1}m_{p1}^2+m_{r1}m_{p2}^2$	$m_{r1}m_{p1}^2+m_{r1}m_{p2}^2$
$(B,A;[m_{r1}])$	$(B,A;[m_{p1}])$ $(A,A;[m_{p2}])$	$m_{r1}m_{p1}m_{p2}$	$m_{r1}m_{p1}^2+m_{r1}m_{p2}^2$ $m_{r1}m_{p1}m_{p2}+m_{r1}m_{p2}m_{p1}$
$(A,B;[m_{r1}])$	$(A,A;[m_{p1}])$ $(B,A;[m_{p2}])$	$m_{r1}m_{p1}m_{p2}$	$m_{r1}m_{p1}^2+m_{r1}m_{p2}^2$ $m_{r1}m_{p1}m_{p2}+m_{r1}m_{p2}m_{p1}$
$(A,A;[m_{r1}])$	$(A,B;[m_{p3}])$	$m_{r1}m_{p3}^2$	$m_{r1}m_{p3}^2$
$(A,B;[m_{r1}])$	$(A,A;[m_{p1}])$ $(B,B;[m_{p2}])$	$m_{r1}m_{p1}m_{p2}$	$m_{r1}m_{p1}^2+m_{r1}m_{p2}^2$
$(A,B;[m_{r1}])$	$(A,B;[m_{p1}])$ $(B,A;[m_{p2}])$	$m_{r1}m_{p1}m_{p2}$	$m_{r1}m_{p1}^2+m_{r1}m_{p2}^2$
$(A,A;[m_{r1}])$	$(A,B;[m_{p1}])$ $(A,C;[m_{p2}])$	$m_{r1}m_{p1}m_{p2}+m_{r1}m_{p2}m_{p1}+$ $m_{r1}m_{p1}^2+m_{r1}m_{p2}^2$	$m_{r1}m_{p1}^2+m_{r1}m_{p2}^2$
$(A,B;[m_{r1}])$	$(A,A;[m_{p1}])$ $(B,C;[m_{p2}])$	$m_{r1}m_{p1}m_{p2}$	$m_{r1}m_{p1}^2+m_{r1}m_{p2}^2$
$(A,B;[m_{r1}])$	$(A,C;[m_{p1}])$ $(B,A;[m_{p2}])$	$m_{r1}m_{p1}m_{p2}$	$m_{r1}m_{p1}^2+m_{r1}m_{p2}^2$
$(B,A;[m_{r1}])$	$(B,A;[m_{p1})]$ $(A,C;[m_{p2}])$	$m_{r1}m_{p1}m_{p2}$	$m_{r1}m_{p1}^2+m_{r1}m_{p2}^2$
$(B,A;[m_{r1}])$	$(B,C;[m_{p1}])$ $(A,A;[m_{p2}])$	$m_{r1}m_{p1}m_{p2}$	$m_{r1}m_{p1}^2+m_{r1}m_{p2}^2$
$(B,C;[m_{r1}])$	$(B,A;[m_{p1}])$ $(C,A;[m_{p2}])$	$m_{r1}m_{p1}m_{p2}$	$m_{r1}m_{p1}^2+m_{r1}m_{p2}^2$ $m_{r1}m_{p1}m_{p2}+m_{r1}m_{p2}m_{p1}$

No es difícil observar que con la tercera, cuarta y décima base de datos canónica $|t|_{Q(d)} \not\leq |t|_{Q''(d)}$. Por lo tanto, por el Teorema Principal se puede afirmar que $Q \not\leq_b Q''$. $\square$

En estos ejemplos es fácil comparar los polinomios que representan las multiplicidades simbólicas que obtiene cada consulta al aplicarse sobre cada base de datos en $\mathcal{CBDC}(Q)$. Pero, en general, no sabemos si la comparación de polinomios de este tipo es decidible. En la siguiente sección se muestra un procedimiento que puede ayudarnos a comparar una gran clase de pares de polinomios del tipo de aquellos que expresan las multiplicidades simbólicas.

8.6 Comparación con procedimientos anteriores.

Obsérvese que aunque la comparación de los polinomios que expresan las multiplicidades pueda ser difícil (y no sabemos si, incluso, imposible en algunos casos), sin embargo, cuando existe una aplicación cobertora de la consulta Q' sobre la consulta Q, cada monomio del polinomio $|t_{d_i}|_{Q(d_i)}$ estará también presente en el polinomio $|t_{d_i}|_{Q'(d_i)}$, por lo que la comparación resulta trivial.

En otras palabras, cuando existe una aplicación cobertora de la consulta Q' sobre la consulta Q, el polinomio que expresa la multiplicidad de las tuplas t_d obtenidas por Q' cuando se aplica sobre cualquier base de datos canónica d_i, tendrá todos y cada uno de los monomios que aparezcan en el polinomio que expresa la multiplicidad de t_d obtenida por Q y quizá algunos monomios más.

Es evidente, por tanto, que cuando existe una aplicación cobertora, la comparación de los polinomios que expresan las multiplicidades es trivial. Así, el procedimiento descrito en este trabajo permite comprobar la b-inclusión de todas las parejas de consultas que ya podían ser estudiadas aplicando la condición suficiente de [8]. En consecuencia, el procedimiento que se presenta en esta tesis para la comprobación de la b-inclusión constituye el procedimiento más general que se ha desarrollado hasta el momento.

9

COMPARACIÓN DE LOS POLINOMIOS QUE EXPRESAN LAS MULTIPLICIDADES.

Para saber si $Q \leq_b Q'$ es necesario verificar que para cada $d \in \mathcal{CBDC}(Q)$ la multiplicidad de la tupla t_d en $Q'(d)$ no es menor que la multiplicidad de t_d en $Q(d)$. Estas multiplicidades son expresiones polinómicas cuyos coeficientes, exponentes y variables están en el rango de los enteros positivos. Así, para conocer si $Q \leq_b Q'$ necesitamos probar si un polinomio es mayor que otro en el rango de los enteros positivos.

En esta sección se presenta un procedimiento que es capaz de comparar un gran conjunto de pares de polinomios. Evidentemente, no es objetivo de este trabajo profundizar en la teoría matemática subyacente, por tanto el prodecimiento de comparación de pares de polinomios se describe de forma sencilla.

Comienza la presentación con polinomios de una sola variable y luego se generaliza el proceso para expresiones polinómicas de más de una variable.

9.1 Comparación de polinomios de una variable.

$\mathcal{Z}^+[x]$ denota los polinomios de una variable x cuyos coeficientes y variable están en el rango de $\mathcal{Z}^+$, los enteros positivos.

Sean $p(x)$ y $q(x)$ dos polinomios en $\mathcal{Z}^+[x]$. Para averiguar si se cumple para todo valor de $x \in \mathcal{Z}^+$ que $p(x) \geq q(x)$, se puede plantear el problema como la comprobación de si cumple que:

$$h(x) \quad = \quad p(x) - q(x) \geq 0 \quad \forall x \in \mathcal{Z}^+$$

Este nuevo polinomio de una variable $h(x)$ tiene ya coeficientes positivos y negativos. El problema se reduce ahora a comprobar si $h(x) \geq 0 \quad \forall x \in \mathcal{Z}^+$. No es difícil ver que la siguiente proposición aporta una condición suficiente.

Proposición 9.1.1 *Sean $p(x)$ y $q(x)$ dos polinomios en $\mathcal{Z}^+[x]$.*
Sea $h(x) = p(x) - q(x)$.

Si $h(x)$ satisface las tres condiciones siguientes, $h(x) \geq 0 \quad \forall x \geq 1$.

Condición 1. *El coeficiente del monomio de mayor grado es positivo.*

Condición 2. $h(1) \geq 0$.

Condición 3. *Existe al menos una partición de $h(x)$ en grupos de monomios de modo que el valor numérico de cada grupo $\forall x \in \mathcal{Z}^+$ sea positivo. Es decir, en cada grupo, los monomios negativos tendrán siempre un valor menor que los monomios positivos. Cada monomio debe incluirse en un y solo un grupo.*[1]

Existen diferentes patrones de monomios positivos y negativos que garantizan que un grupo de monomios con un patrón determinado sólo toma valores no negativos cuando $x \geq 2$; el valor del grupo cuando $x = 1$ no es importante, porque la Condición 2 garantiza que $h(1) \geq 0$. Tampoco importa qué pasa cuando $x = 0$ porque en los polinomios que expresan las multiplicidades, cada variable representa la multiplicidad simbólica de una tupla de una base de datos y recuérdese que las multiplicidades nunca pueden valer 0.

A continuación describimos cuatro de esos patrones en los Grupos I al IV.

[1]La Condición 3 implica la Condición 1 pero la dejamos para mayor claridad y porque será usada como una prueba rápida para verificar si $h(x) < 0$ para algún $x \geq 1$.

9.1.1 Grupo I.

Esta clase de grupo corresponde al caso trivial. Sólo hay monomios positivos en el grupo, es decir, monomios con coeficientes positivos. Está claro que el valor numérico del grupo siempre va a ser positivo.

9.1.2 Grupo II.

En esta clase de grupos es posible tener cualquier número de monomios positivos y negativos, pero el menor grado de los monomios positivos es mayor que el grado más grande de los monomios negativos. Es decir, el grupo tiene un patrón:

$$C_h.x^h + \cdots + C_i.x^i \; -C_j.x^j - \cdots - C_k.x^k \quad \text{donde} \quad h > i > j > k.$$

Ahora vamos a demostrar que si el valor numérico del grupo es positivo cuando $x = 2$, entonces será positivo para cualquier valor de $x > 2$.

Lema 9.1.1 *Sea $p(x)$ un polinomio en $\mathcal{Z}^+[x]$, y asumimos que $p(x)$ satisface las Condiciones 1 y 2 anteriores. Sea $r(x)$ el siguiente polinomio, que se construye a partir de algunos monomios de $p(x)$:*

$$r(x) = C_h.x^h + \cdots + C_i.x^i \; -C_j.x^j - \cdots - C_k.x^k \quad \text{donde} \quad h > i > j > k.$$

Entonces, $r(2) \geq 0 \Longrightarrow r(x) \geq 0 \;\; \forall x \geq 2, \quad x \in \mathcal{Z}^+.$

Demostración:
Se quiere probar que:

$$C_h.2^h + \cdots + C_i.2^i \; -C_j.2^j - \cdots - C_k.2^k \geq 0$$

$$\Longrightarrow$$

$$C_h.x^h + \cdots + C_i.x^i \; -C_j.x^j - \cdots - C_k.x^k \geq 0 \;\; \forall x \geq 2$$

Pueden considerarse dos casos: cuando hay un único monomio positivo y cuando hay varios monomios positivos. Se demuestran estos dos casos por separado.

Caso de un sólo monomio positivo.
Supongamos que sólo hay un monomio positivo, por ejemplo $C_i x^i$, y más de un monomio negativo. Se debe demostrar en este caso que:

$$C_i.2^i \ - C_j.2^j - \cdots - C_k.2^k \geq 0 \implies C_i.x^i \ - C_j.x^j - \cdots - C_k.x^k \geq 0 \ \ \forall x \geq 2$$

Podemos asumir, sin pérdida de generalidad, que $k = 0$ (dividiendo todos los monomios por x^k), el patrón, por tanto, es:

$$C_i.x^i \ - C_j.x^j - \cdots - C_1.x - C_0$$

Asumimos por hipótesis que:

$$C_i.2^i \geq C_j.2^j + \ldots + C_1.2 + C_0$$

Asúmase ahora un valor de $x > 2, \quad x \in \mathcal{Z}^+$, por ejemplo $x = 2y; \quad y > 1; \quad y \in \mathcal{R}^+$, donde $\mathcal{R}^+$ es el conjunto de los números reales positivos. Entonces para $x > 2$, se tiene que:

$$C_i.x^i \ - C_j.x^j - \cdots - C_1.x - C_0 = C_i.(2y)^i - C_j.(2y)^j - \cdots - C_1.2y - C_0$$

Es necesario demostrar que:

$$C_i.(2y)^i \geq C_j.(2y)^j + \cdots + C_1.2y + C_0 \ \ \forall y \in \mathcal{R}^+$$

Es evidente que esto es cierto debido a que por hipótesis:

$$C_i.2^i \geq C_j.2^j + \cdots + C_1.2 + C_0 \quad e \quad y > 1$$

Es fácil ver que:

$$C_i.2^i \geq C_j.2^j + \cdots + C_1.2 + C_0$$

$$\Longrightarrow$$

$$C_i.2^i.y^i \geq (C_j.2^j + \cdots + C_1.2 + C_0).\ y^i \geq (C_j.(2y)^j + \cdots + C_1.2y + C_0)$$

Esto completa la demostración para este caso.

Caso de varios monomios positivos.
Supongamos ahora que tenemos dos monomios positivos. El patrón en el grupo será:

$$C_h.x^h + C_i.x^i - C_j.x^j - \cdots - C_k.x^k \quad \text{donde} \quad h > i > j > k$$

Queremos demostrar que:

$$C_h.2^h + C_i.2^i \geq C_j.2^j + \cdots + C_k.2^k \Longrightarrow C_h.x^h + C_i.x^i \geq C_j.x^j + \cdots + C_k.x^k \ \ \forall x \geq 2.$$

Pero, evidentemente se cumple siempre que:

$$C_h.x^h + C_i.x^i \quad = \quad (C_h.x^{h-i} + C_i).x^i$$

Por tanto se puede construir un nuevo coeficiente, que denotaremos C_i', cuyo valor será

$$C_i' = C_h.2^{h-i} + C_i$$

Entonces se pueden sustituir los dos monomios positivos por un único monomio con ese coeficiente. A partir de aquí, está claro que:

$$C_h.x^h + C_i.x^i = (C_h.x^{h-i} + C_i).x^i \ \geq \ (C_h.2^{h-i} + C_i).x^i = C_i'.x^i \quad \forall x \geq 2$$

Usando el nuevo coeficiente se puede reducir este caso al anterior con un solo monomio positivo. Se puede utilizar el mismo tipo de reducción con cualquier número de coeficientes positivos. Esto completa la demostración del lema. □

9.1.3 Grupo III.

En este caso, tenemos tres monomios en el grupo con un monomio negativo en el medio de los positivos. Entonces, el patrón es:

$$C_i.x^i - C_{i-1}.x^{i-1} + C_{i-2}.x^{i-2}.$$

Obsérvese que se asume que $C_i.2^i < C_{i-1}.2^{i-1}$, pues en otro caso esos dos monomios pertenecerían al Grupo II.

Se puede asumir, sin pérdida de generalidad, que $i = 2$ (dividiendo los tres monomios por x^{i-2} si es necesario) y que tratamos con grupos cuyo patrón general es:

$$C_2.x^2 - C_1.x + C_0.$$

Demostraremos que la condición que necesitamos probar para este caso es $4.C_2.C_0 - C_1^2 \geq 0$.

Lema 9.1.2 *Considérese el polinomio* $r(x) = C_2.x^2 - C_1.x + C_0$. *Si se cumple que* $4.C_2.C_0 - C_1^2 \geq 0$. *Entonces* $r(x) \geq 0 \;\; \forall x > 1$.

Es decir:

$$4.C_2.C_0 - C_1^2 \geq 0 \Longrightarrow C_2.x^2 - C_1.x + C_0 \geq 0 \;\; \forall\, x \in \mathcal{Z}^+$$

Demostración:
Asumimos que $4.C_2.C_0 - C_1^2 \geq 0$.
Como es sabido las raíces de $r(x)$ vienen dadas por la ecuación

$$x = \frac{C_1 \pm \sqrt{C_1^2 - 4.C_2.C_0}}{2.C_2}$$

Si $4.C_2.C_0 - C_1^2 \geq 0$ entonces, o hay una sola raíz, cuando $4.C_2.C_0 - C_1^2 = 0$, o ninguna, cuando $4.C_2.C_0 - C_1^2 > 0$; en otro caso, la parábola que representa $r(x)$ nunca cruzaría el eje x.

Como $C_2 \geq 0$, es evidente que $r(x) > 0$ para algunos valores positivos de x; esto añadido al hecho de que $r(x)$ no cruza el eje x implica que $r(x) \geq 0 \;\; \forall x \in \mathcal{Z}^+$. $\square$

Se puede generalizar este caso para cualquier grupo de monomios con el patrón:

$$C_i.x^i - C_j.x^j + C_{j-1}.x^{j-1} \quad donde \;\; i > j.$$

9.1.4 Grupo IV.

En los grupos de esta clase hay más de un monomio negativo entre los dos monomios positivos. El patrón general en este caso es:

$$C_i.x^i - C_j.x^j - C_{j-1}.x^{j-1} - \cdots - C_k.x^k + C_l.x^l, \quad i > j > k > l$$

Se asume que no se cumple que $C_i.2^i \geq C_j.2^j - C_{j-1}.2^{j-1} - \cdots - C_k.2^k$, pues si se cumpliera, estos monomios encajarían en el patrón del Grupo II y el último monomio encajaría en el Grupo I

Se puede asumir que el patrón general de los grupos en este caso es (dividiendo cada monomio por x^l si es necesario):

$$g(x) = C_n.x^n - C_m.x^m - \cdots - C_1.x + C_0, \qquad n > m > 0$$

Lema 9.1.3 *Sea el polinomio:*

$$g(x) = C_n.x^n - C_m.x^m - \cdots - C_1.x + C_0, \quad n > m > 0$$

Entonces $g(x) \geq 0$ *para todo* $x \geq 2$ *si se cumplen las dos condiciones siguientes:*

- $g(2) \geq 1$. *Es decir, si* $x = 2$ *la suma de los monomios en el grupo es positiva.*

- $g'(x) \geq 0$. *Es decir, el polinomio que se obtiene al calcular la primera derivada de* $g(x)$ *es siempre mayor o igual a cero para cualquier valor de* x. *Es decir:*

$$g'(x) = n.C_n.x^{n-1} - m.C_m.x^{m-1} - \cdots - C_1 \geq 0$$

Pero por el Lema 9.1.1, si $g'(x)$ *es mayor que* 0 *cuando* $x = 2$, *entonces será mayor que* 0 *para cualquier valor de* x. *Por tanto la segunda condición es que* $g'(2) \geq 0$. *Es decir:*

$$g'(2) = n.C_n.2^{n-1} - m.C_m.2^{m-1} - \cdots - C_1 \geq 0$$

Demostración:
$g'(2) \geq 0$ indica que la función $g(x)$ es creciente en $x = 2$ y por el lema 9.1.1 será siempre creciente después del valor $x = 2$. Como $g(2) \geq 1$, es claro que $g(x)$ nunca tendrá valores negativos para $x \geq 1$.

□

En estos cuatro casos, los monomios negativos en el grupo se anulan con los monomios positivos del grupo, por lo que el grupo tendrá valor positivo para cualquier valor de $x \geq 2$.

Si es posible dividir el polinomio en grupos de monomios como éstos, y cada grupo satisface las condiciones que demostramos en cada caso, el valor numérico del polinomio siempre será positivo para cualquier valor de $x \in \mathcal{Z}^+$.

9.2 Procedimiento de comparación de polinomios de una variable $p(x)$ y $q(x)$.

Para saber si $p(x) \geq q(x)$ $\forall x > 0$ se debe comprobar si se cumple que:

$$h(x) = p(x) - q(x) \geq 0 \quad \forall x > 0.$$

A partir de la proposición demostrada en el apartado anterior se obtiene un procedimiento que permite averiguar si $h(x) \geq 0$. El proceso responderá `SI`, `NO`, o `DESCONOCIDO` a la pregunta:

$$\textit{Es cierto que } h(x) \geq 0 \;\; \forall x > 0?$$

Después de simplificar $h(x)$ usando los procedimientos algebraicos habituales, y ordenar los monomios de acuerdo al grado de x, podemos comenzar con el procedimiento. Los pasos del proceso son:

1.- Observación del signo del primer monomio: Es necesario que el coeficiente del monomio con mayor grado sea positivo. Si el signo del primer coeficiente es negativo, el proceso se para aquí y la respuesta será `NO`.

2.- Comprobación de que $h(1) \geq 0$: Basta sumar los coeficientes del polinomio y si el resultado es negativo el proceso se para aquí y la respuesta es `NO`.

3.- Búsqueda de una partición adecuada del polinomio: Es necesario dividir el polinomio en grupos de monomios. Cada monomio deberá estar en un y sólo un grupo. Cada grupo debe corresponder con el patrón de uno de los grupos descritos en la sección anterior. Si hay una partición adecuada para el polinomio, y los monomios de cada grupo satisfacen las condiciones para esta clase de grupo, entonces la respuesta al proceso es `SÍ`. En otro caso, la respuesta es `DESCONOCIDO`.

Hay más y mejores procedimientos para averiguar si un polinomio de una variable tiene valores positivos siempre (como la derivación), pero tales procedimientos no se pueden extender a polinomios con más variables. Las condiciones que se describen en la sección anterior, sin embargo, pueden aplicarse de modo recursivo a polinomios con cualquier número de variables como se verá a continuación

9.3 Comparación de polinomios de dos variables.

Queremos comprobar si se cumple que $r(x,y) = p(x,y) - q(x,y) \ \geq 0 \ \forall x, y \in \mathcal{Z}^+$.

Cualquiera de estos polinomios se puede considerar como un polinomio $[h(x)](y)$ cuyos coeficientes son polinomios en $\mathcal{Z}^+[x]$, en lugar de valores enteros. Por lo tanto, el procedimiento anterior para comparar polinomios definidos sobre una variable soporta un proceso para comparar pares de polinomios definidos sobre dos variables.

Ahora se considera que un coeficiente es positivo cuando el polinomio que lo constituye toma siempre valores positivos y se considera que es negativo si existe un conjunto de valores para sus variables que producen que el polinomio que constituye el coeficiente tome un valor negativo. Ahora, para saber si un coeficiente es positivo o negativo es necesario comprobar si el polinomio que lo constituye toma siempre valores positivos para cualesquiera que sean los valores de sus variables (las cuales tomarán valores siempre en $\mathcal{Z}^+$).

Con sólo esa consideración se puede aplicar el mismo procedimiento que se describió para polinomios de una variable.

Después de simplificar el polinomio $r(x,y)$, organizando sus monomios como si fuesen polinomios de una variable y, y obteniendo los nuevos coeficientes $h(x)$ para cada grado de la variable y, el procedimiento es el siguiente:

1.- Observación del signo del primer monomio: Es decir, ver si el polinomio $m(x)$, que es el coeficiente del monomio de mayor grado de y en $[h(x)](y)$, es siempre positivo. Se considera *positivo* un coeficiente que siempre tiene valores positivos, y *no-positivo* en otro caso. Para saber si un coeficiente es positivo, necesitamos ver si el polinomio, que es el coeficiente, tiene siempre valor positivo para cualquier valor de su variable.

Puesto que $m(x)$ es un polinomio de una variable, es posible utilizar el procedimiento descrito para comparar polinomios de una sola variable, aunque de hecho, es también posible aplicar cualquier otro procedimiento que permita determinarlo.

Si $m(x)$ no es positivo, el proceso se para aquí y la respuesta es `NO`.

2.- Comprobación de que $[h(x)](1) \geq 0$**:** Es necesario sustituir y por 1. Se obtiene un polinomio $g(x)$. El nuevo polinomio $g(x)$ es la suma de todos los coeficientes del polinomio original $[h(x)](y)$. Usando el procedimiento para los polinomios con exactamente una variable vemos si $g(x) \geq 0$. Si $\exists x$ tal que $g(x) < 0$, entonces el proceso se para aquí y la respuesta es `NO`.

3.- Cálculo del signo de cada monomio: En el proceso de los polinomios con una variable, los coeficientes son constantes y por tanto este paso es innecesario, pero ahora cada coeficiente es un polinomio y es preciso conocer su signo. Como se mencionó antes, un coeficiente $c(x)$ de un monomio en $[h(x)](y)$ se considera positivo si y sólo si $c(x) \geq 0 \ \forall x > 0$.

4.- Búsqueda de una partición adecuada del polinomio $[h(x)](y)$**:** Es necesario dividir el polinomio en grupos de monomios. Cada monomio puede estar en un solo grupo. Los grupos deben ajustarse al patrón de alguno de los cuatro grupos descritos en la sección anterior. Es necesario ver si cada grupo satisface las condiciones para su grupo.

Si se encuentra una partición adecuada del polinomio, y los monomios en cada grupo satisfacen las condiciones para ese tipo de grupo, entonces la respuesta del proceso será SÍ. En otro caso, la respuesta será DESCONOCIDO.

Obsérvese que este procedimiento puede generalizarse para polinomios de n variables. Los coeficientes de tales polinomios serán a su vez polinomios de $n-1$ variables y así sucesivamente hasta llegar a polinomios de una variable.

Todavía estamos trabajando en buscar un procedimiento más poderoso para comparar los polinomios que representan las multiplicidades simbólicas.

Aunque la comparación de polinomios de varias variables constituye un problema indecidible, en el caso de estos polinomios concretos concurren una serie de “regularidades” (son homogéneos, el número de variables es igual o menor al grado de cada monomio, los monomios negativos tienen grado menor o igual que los positivos ...) que nos hacen pensar que quizá su comparación sea siempre posible.

10

EJEMPLO COMPLETO DE COMPROBACIÓN DE LA B-INCLUSIÓN DE DOS CONSULTAS.

10.1 Introducción.

Se presenta en este capítulo un ejemplo completo del proceso de comprobación de la b-inclusión entre dos consultas conjuntivas dadas. Se usará para ello una pareja de consultas extraídas del artículo de Chaudhuri y Vardi [8]. Dicha pareja de consultas se exponen en el citado trabajo como ejemplo de consultas que cumplen las dos condiciones necesarias pero inclumplen la condición suficiente de Chaudhuri y Vardi [8], expuestas en esta tesis como Proposiciones 5.2.1, 5.2.2 y 5.3.1 respectivamente. Chaudhuri y Vardi recalcan que no cuentan con un procedimiento que permita decidir la b-inclusión de dicha pareja de consultas y que sería de gran interés contar con alguno. Es por ello, que en esta tesis se utilizan precisamente esas consultas, para mostrar que el procedimiento aquí descrito es válido para comprobar la b-inclusión de una clase mucho más amplia de consultas conjuntivas que la clase de aquellas que cumplen la condición suficiente de Chaudhuri y Vardi [8].

10.2 Ejemplo de comprobación de la b-inclusión.

Considérense las consultas:

$$Q'(X,Z) = p(X), q(U,Y), q(V,Y), r(Z).$$

$$Q(X,Z) = p(X), q(U,X), q(V,Z), r(Z).$$

Obsérvese que, aunque cumplen las dos condiciones necesarias expuestas en las Proposiciones 5.2.1 y 5.2.2, no cumplen la condición suficiente de que exista una aplicación cobertora de Q' sobre Q. Por tanto, los resultados de Chaudhuri y Vardi [8] no permiten decidir su b-inclusión.

Se desea saber si se cumple que $Q \leq_b Q'$. Para averiguarlo, siguiendo el procedimiento descrito en esta tesis, es necesario realizar lo siguiente:

Construir el conjunto $\mathcal{CBDC}(Q)$. Para construir el conjunto completo de bases de datos canónicas de Q denotadas d_i es necesario establecer todas las Q-aplicaciones no isomórficas. Dichas aplicaciones y la base de datos d_i a las que cada una de ellas da lugar se representan en formato de tabla en el apartado correspondiente.

Aplicar Q y Q' sobre cada base de datos $d_i \in \mathcal{CBDC}(Q)$. De cada base de datos se debe extraer la tupla t_d y obtener el polinomio que representa su multiplicidad. Es decir, se deben obtener los polinomios $|t_{d_i}|_{Q(d_i)}$ y $|t_{d_i}|_{Q'(d_i)}$.

Comparar los polinomios $|t_{d_i}|_{Q(d_i)}$ y $|t_{d_i}|_{Q'(d_i)}$. Es decir, comparar, para cada base de datos canónica d_i, la multiplicidad simbólica con la que cada consulta obtiene para la tupla t_{d_i}. Esta comparación puede relizarse mediante la aplicación del procedimiento descrito en el Capítulo 9 de esta tesis.

Se describe a continuación la aplicación de estos tres pasos:

10.3 Construcción del conjunto de bases de datos canónicas $\mathcal{CBDC}(Q)$.

Las diferentes Q-aplicaciones, así como la base de datos a las que cada una da lugar, se calculan mediante el procedimiento descrito en el Capítulo 7 de esta tesis. En la siguiente tabla se muestran todas las $d_i \in \mathcal{CBDC}(Q)$ así como las Q-aplicaciones que las produjeron. En la última columna de la tabla se muestra la correspondiente tupla t_d que las consultas Q y Q' deben obtener al aplicarse sobre cada una de las d_i.

d_i	*Q-aplicación* θ_i	*Bases de Datos*			t_{d_i}
	(X, U, V, Z)	p	q	r	
d_1	(A, A, A, A)	$(A; [m_{p1}])$	$(A, A; [m_{s3}])$	$(A; [m_{r1}])$	(A, A)
d_2	(A, A, A, B)	$(A; [m_{p1}])$	$(A, A; [m_{s1}])$ $(A, B; [ms2])$	$(B; [m_{r1}])$	(A, B)
d_3	(A, A, B, A)	$(A; [m_{p1}])$	$(A, A; [m_{s1}])$ $(B, A; [ms2])$	$(A; [m_{r1}])$	(A, A)
d_4	(A, B, A, A)	$(A; [m_{p1}])$	$(B, A; [m_{s1}])$ $(A, A; [ms2])$	$(A; [m_{r1}])$	(A, A)
d_5	(B, A, A, A)	$(B; [m_{p1}])$	$(A, B; [m_{s1}])$ $(A, A; [ms2])$	$(A; [m_{r1}])$	(B, A)
d_6	(A, A, B, B)	$(A; [m_{p1}])$	$(A, A; [m_{s1}])$ $(B, B; [ms2])$	$(B; [m_{r1}])$	(A, A)
d_7	(A, B, B, A)	$(A; [m_{p1}])$	$(B, A; [m_{s3}])$	$(A; [m_{r1}])$	(A, A)
d_8	(A, B, A, B)	$(A; [m_{p1}])$	$(B, A; [m_{s1}])$ $(A, B; [ms2])$	$(B; [m_{r1}])$	(A, B)
d_9	(A, A, B, C)	$(A; [m_{p1}])$	$(A, A; [m_{s1}])$ $(B, C; [ms2])$	$(C; [m_{r1}])$	(A, C)
d_{10}	(A, B, A, C)	$(A; [m_{p1}])$	$(B, A; [m_{s1}])$ $(A, C; [ms2])$	$(C; [m_{r1}])$	(A, C)
d_{11}	(B, A, A, C)	$(B; [m_{p1}])$	$(A, B; [m_{s1}])$ $(A, C; [ms2])$	$(C; [m_{r1}])$	(B, C)
d_{12}	(B, A, C, A)	$(B; [m_{p1}])$	$(A, B; [m_{s1}])$ $(C, A; [ms2])$	$(A; [m_{r1}])$	(B, A)
d_{13}	(B, C, A, A)	$(B; [m_{p1}])$	$(C, B; [m_{s1}])$ $(A, A; [ms2])$	$(A; [m_{r1}])$	(B, A)
d_{14}	(A, B, C, A)	$(A; [m_{p1}])$	$B, A; [m_{s1}])$ $(C, A; [ms2])$	$(A; [m_{r1}])$	(A, A)
d_{15}	(A, B, C, D)	$(A; [m_{p1}])$	$(B, A; [m_{s1}])$ $(C, D; [ms2])$	$(D; [m_{r1}])$	(A, D)

10.4 Aplicación de Q y Q' sobre cada base de datos canónica $d_i \in \mathcal{CBDC}(Q)$.

Al aplicar las consultas conjuntivas Q y Q' sobre cada base de datos canónica d_i, las dos deberán obtener, al menos, la tupla t_{d_i}. De lo que se trata es de comprobar que Q' obtiene dicha tupla con una multiplicidad igual o mayor de Q.

En la siguiente tabla se muestra, para cada base de datos canónica d_i, la tupla t_{d_i} y la multiplicidad con la que cada una de las consultas obtiene dicha tupla.

Obsérvese que las multiplicidades con las que se obtiene en cada caso la tupla t_{d_i} se expresan mediante los polinomios, denominados respectivamente, $|t_d|_{Q(d_i)}$ y $|t_d|_{Q'(d_i)}$ que se muestran en las dos últimas columnas de la tabla.

d_i	*Bases de Datos*			$\lvert t_d\rvert_{Q(d_i)}$	$\lvert t_d\rvert_{Q(d_i)}$
	p	q	r		
d_1	$(A;[m_{p1}])$	$(A,A;[m_{s1}])$	$(A;[m_{r1}])$	$m_{p1}m_{s1}^2m_{r1}$	$m_{p1}m_{s1}^2m_{r1}$
d_2	$(A;[m_{p1}])$	$(A,A;[m_{s1}])$ $(A,B;[m_{s2}])$	$(B;[m_{r1}])$	$m_{p1}m_{s1}^2m_{r1}+$ $+m_{p1}m_{s2}^2m_{r1}$	$m_{p1}m_{s1}m_{s2}m_{r1}$
d_3	$(A;[m_{p1}])$	$(A,A;[m_{s1}])$ $(B,A;[m_{s2}])$	$(A;[m_{r1}])$	$m_{p1}m_{s1}^2m_{r1}+$ $+m_{p1}m_{s2}^2m_{r1}+$ $+m_{p1}m_{s1}m_{s2}m_{r1}$	$m_{p1}m_{s1}^2m_{r1}+$ $+m_{p1}m_{s2}^2m_{r1}+$ $+m_{p1}m_{s1}m_{s2}m_{r1}$
d_4	$(A;[m_{p1}])$	$(B,A;[m_{s1}])$ $(A,A;[m_{s2}])$	$(A;[m_{r1}])$	$m_{p1}m_{s1}^2m_{r1}+$ $+m_{p1}m_{s2}^2m_{r1}+$ $+m_{p1}m_{s1}m_{s2}m_{r1}$	$m_{p1}m_{s1}^2m_{r1}+$ $+m_{p1}m_{s2}^2m_{r1}+$ $+m_{p1}m_{s1}m_{s2}m_{r1}$
d_5	$(B;[m_{p1}])$	$(A,B;[m_{s1}])$ $(A,A;[m_{s2}])$	$(A;[m_{r1}])$	$m_{p1}m_{s1}^2m_{r1}$ $+m_{p1}m_{s2}^2m_{r1}$	$m_{p1}m_{s1}m_{s2}m_{r1}$
d_6	$(A;[m_{p1}])$	$(A,A;[m_{s1}])$ $(B,B;[m_{s2}])$	$(B;[m_{r1}])$	$m_{p1}m_{s1}^2m_{r1}$ $+m_{p1}m_{s2}^2m_{r1}$	$m_{p1}m_{s1}m_{s2}m_{r1}$
d_7	$(A;[m_{p1}])$	$(B,A;[m_{s1}])$	$(A;[m_{r1}])$	$m_{p1}m_{s1}^2m_{r1}$	$m_{p1}m_{s1}^2m_{r1}$
d_8	$(A;[m_{p1}])$	$(B,A;[m_{s1}])$ $(A,B;[m_{s2}])$	$(B;[m_{r1}])$	$m_{p1}m_{s1}^2m_{r1}+$ $+m_{p1}m_{s2}^2m_{r1}$	$m_{p1}m_{s1}m_{s2}m_{r1}$
d_9	$(A;[m_{p1}])$	$(A,A;[m_{s1}])$ $(B,C;[m_{s2}])$	$(C;[m_{r1}])$	$m_{p1}m_{s1}^2m_{r1}+$ $+m_{p1}m_{s2}^2m_{r1}$	$m_{p1}m_{s1}m_{s2}m_{r1}$
d_{10}	$(A;[m_{p1}])$	$(B,A;[m_{s1}])$ $(A,C;[m_{s2}])$	$(C;[m_{r1}])$	$m_{p1}m_{s1}^2m_{r1}+$ $+m_{p1}m_{s2}^2m_{r1}$	$m_{p1}m_{s1}m_{s2}m_{r1}$
d_{11}	$(B;[m_{p1}])$	$(A,B;[m_{s1}])$ $(A,C;[m_{s2}])$	$(C;[m_{r1}])$	$m_{p1}m_{s1}^2m_{r1}+$ $+m_{p1}m_{s2}^2m_{r1}$	$m_{p1}m_{s1}m_{s2}m_{r1}$
d_{12}	$(B;[m_{p1}])$	$(A,B;[m_{s1}])$ $(C,A;[m_{s2}])$	$(A;[m_{r1}])$	$m_{p1}m_{s1}^2m_{r1}+$ $+m_{p1}m_{s2}^2m_{r1}$	$m_{p1}m_{s1}m_{s2}m_{r1}$
d_{13}	$(B;[m_{p1}])$	$(C,B;[m_{s1}])$ $(A,A;[m_{s2}])$	$(A;[m_{r1}])$	$m_{p1}m_{s1}^2m_{r1}+$ $+m_{p1}m_{s2}^2m_{r1}$	$m_{p1}m_{s1}m_{s2}m_{r1}$
d_{14}	$(A;[m_{p1}])$	$B,A;[m_{s1}])$ $(C,A;[m_{s2}])$	$(A;[m_{r1}])$	$m_{p1}m_{s1}^2m_{r1}+$ $+m_{p1}m_{s2}^2m_{r1}+$ $+m_{p1}m_{s1}m_{s2}m_{r1}$	$m_{p1}m_{s1}^2m_{r1}+$ $+m_{p1}m_{s2}^2m_{r1}+$ $+m_{p1}m_{s1}m_{s2}m_{r1}$
d_{15}	$(A;[m_{p1}])$	$(B,A;[m_{s1}])$ $(C,D;[m_{s2}])$	$(D;[m_{r1}])$	$m_{p1}m_{s1}^2m_{r1}+$ $+m_{p1}m_{s2}^2m_{r1}$	$m_{p1}m_{s1}m_{s2}m_{r1}$

10.5 Comparación de los polinomios $|t_{d_i}|_{Q(d_i)}$ y $|t_{d_i}|_{Q'(d_i)}$

Usando el procedimiento expuesto en el Capítulo 9, se debe comprobar si para todas las base de datos $d_i \in \mathcal{CBDC}(Q)$ se cumple que:

$$|t_{d_i}|_{Q_{d_i}} \leq |t_{d_i}|_{Q'_{d_i}}$$

Observando la tabla anterior es fácil notar que en las bases de datos d_1, d_3, d_4, d_7 y d_{14} las dos consultas obtienen la tupla t_{d_i} con igual multiplicidad puesto que los polinomios son iguales. Luego para esas bases de datos sí se cumple que:

$$|t_{d_i}|_{Q_{d_i}} \leq |t_{d_i}|_{Q'_{d_i}}$$

En el resto de las bases de datos, la multipliciad con la que Q obtienen la tupla t_{d_i} es $m_{p1}.m_{s1}.m_{s2}.m_{r1}$ y la multipliciad con la que Q' obtienen la tupla t_{d_i} es $m_{p1}.m_{s1}^2.m_{r1} + m_{p1}.m_{s2}^2.m_{r1}$. Por lo tanto sólo es necesario comprobar si se cumple que:

$$m_{p1}.m_{s1}.m_{s2}.m_{r1} \leq m_{p1}.m_{s1}^2.m_{r1}+m_{p1}.m_{s2}^2.m_{r1} \qquad m_{p1},\ m_{s1},\ m_{s2},\ m_{r1} \in \mathcal{Z}^+$$

Para simplificar la representación de los polinomios cambiaremos la denominación de cada variable de los mismos utilizando las letras a, b, c, d para representar a las variables m_{p1}, m_{s1}, m_{s2} y m_{r1} respectivamente. Por lo tanto el polinomio con el que se va a trabajar es:

$$a.b^2.d \ + \ a.c^2.d \ - \ a.b.c.d$$

Simplificando este polinomio, dividiendo por $a.d$, y reordenando sus monomios, se obtiene el nuevo polinomio del que queremos saber si es un polinomio positivo (es decir, si siempre toma valores positivos):

$$[h(b)](c) = c^2 - bc + b^2$$

A continuación se aplica sobre este polinomio el proceso descrito en el Capítulo 9 que consiste en los siguiente pasos:

1.-Observación del signo del primer monomio. El primer monomio es c^2 y evidentemente su signo es positivo ya que la variable $c \in \mathcal{Z}^+$.

2.- Comprobación de que $[h(b)](1) = 1^2 - b + b^2 \geq 0$. Es muy fácil ver, en este ejemplo que el polinomio $[h(b)](1)$ toma siempre valores positivos. Es decir:

$$[h(b)](1) = 1 - b + b^2 \geq 0 \quad \forall b$$

A pesar de que en este ejemplo es evidente el hecho de que el polinomio $[h(b)](1) \geq 0 \quad \forall b$, es de interés aplicar sobre él el método de comparación de polinomios para resaltar como puede aplicarse recursivamente.

El polinomio $[h(b)](1)$ .después de reordenar sus monomios resulta ser el polinomio de una variable $g(b) = b^2 - b + 1$. Aplicando sobre este polinomio de una variable el método descrito se tiene que:

- El signo del monomio de mayor grado de $g(b)$ es positivo.
- Se cumple que $g(1) = 1^2 - 1 + 1 \geq 0$.
- Los monomios de $g(b)$ se pueden partir en dos grupos que denotamos $r(b)$ y $t(b)$ respectivamente:

$$r(b) = b^2 - b \qquad\qquad y \qquad\qquad t(b) = 1$$

El primero de los dos grupos se ajusta al patrón del Grupo II y satisface su condición ya que $r(2) = 2^2 - 2 \geq 0$, mientras que el segundo se ajusta al patrón del Grupo I y también satisface su condición ya que $t(b) = 1 \geq 0$.

Por tanto usando el procedimiento descrito para en el Capitulo 9, se observa que el polinomio $[h(b)](1) = b^2 - b + 1$ es siempre positivo.

3.- Cálculo del signo de cada monomio. Los signos de cada monomio se representan en la siguiente tabla:

Monomio:	c^2	$-bc$	$+b^2$
Coeficiente:	$C_2 = 1$	$C_1 = -b$	$C_0 = b^2$
Signo:	$+$	$-$	$+$

4.- Búsqueda de una partición adecuada. Se puede ver que el patrón de este polinomio se ajusta prefectamente al del Grupo III. Se tendría, por tanto, una única partición en la que estarían todos los monomios del polinomio.

La condición que necesitan satisfacer las particiones del Grupo III es:

$$4.C_2.C_0 \ - \ C_1^2 \ \geq \ 0$$

En el polinomio del ejemplo se tiene que:

$$4 \times 1 \times b^2 - (-b)^2 \quad = \quad 4b^2 - b^2 \quad = 3b^2 \ \geq 0$$

Por tanto, en esta partición se cumple que el monomio negativo siempre es menor que los monomios positivos. Como esta es la única partición que se hizo del polinomio $[h(b)](c)$, se puede concluir que el polinomio tomará siempre un valor positivo sean cuales sean los valores de sus variables b y c en $\mathcal{Z}^+$

Evidentemente el procedimiento responderá `SÍ`, es decir, se comprueba que:

$$[h(b)](c) \ = \ c^2 \ - \ b.c \ + \ b^2 \ \geq \ 0 \qquad \forall\, b, c$$

Usando el Teorema Principal se puede afirmar por tanto que:

$$Q \leq_b Q'$$

ya que:

$$|t_{d_i}|_{Q_{d_i}} \leq |t_{d_i}|_{Q'_{d_i}} \forall d_i \in \mathcal{CBDC}(Q)$$

□

11

CONCLUSIONES.

En esta tesis se estudia el problema de la b-inclusión de consultas conjuntivas. Se comienza por repasar los trabajos de investigación sobre inclusión de consultas conjuntivas en el marco semántico de conjuntos y se analiza y demuestra el por qué los resultados obtenidos por tales trabajos son inaplicables cuando el marco semántico es el de bolsas.

Posteriormente se describe el marco semántico de bolsas y la notación utilizada en dicho marco, y se exponen exhaustivamente los resultados sobre b-inclusión de consultas conjuntivas encontrados por otros investigadores.

Finalmente se aporta un procedimiento para averiguar si, dadas dos consultas conjuntivas Q y Q', se cumple que una está b-incluida en la otra. Dicho procedimiento lo que hace es reducir el problema de la b-inclusión de consultas conjuntivas, al problema de comparar dos polinomios especiales, que representan multiplicidades simbólicas de los hechos derivados por cada una de las consultas cuando se aplican sobre unas bases de datos especiales construidas por el propio procedimiento.

Hay que destacar que en esta tesis, mediante el procedimiento que se propone para decidir la b-inclusión de consultas conjuntivas, se prueba que se puede reducir el número de bases de datos que es necesario utilizar para comprobar si una consulta conjuntiva está b-incluida en otra, desde un número infinito a un número finito.

La mayor aportación de esta tesis es, por tanto, la de proporcionar un procedimiento que permite comprobar la b-inclusión de consultas conjuntivas bajo el marco semántico de bolsas para un número de casos mayor y *englobador* de todos aquellos casos que pudieran ser decididos con procedimientos anteriores a este trabajo.

También se identifica, debido a las dificultades de realizar la comparación de los polinomios que expresan las multiplicidades simbólicas, dónde está la fuente de la posible indecidibilidad de la b-inclusión.

Una aportación, que consideramos que puede ser de interés para otros trabajos, es la de la definición del Conjunto de Bases de Datos Canónicas para una consulta conjuntiva Q dada, $\mathcal{CBDC}(Q)$. Dicho conjunto refleja las diferentes maneras posibles en que una consulta conjuntiva Q puede aplicarse sobre las tuplas de cualquier base de datos $\mathcal{D}$. Es posible que esta idea pueda ser de utilidad en otras investigaciones en las que se trate de averiguar cuáles y cómo son los hechos derivados que una consulta puede extraer al aplicarse sobre una base de datos arbitraria. Esto se debe a que el $\mathcal{CBDC}(Q)$ contiene los diferentes patrones de hechos base que debe tener una base de datos para que sea posible encontrar una aplicación de asignación desde la consulta Q a dicha base de datos.

12

LÍNEAS DE TRABAJO FUTURAS.

El problema de averiguar si la b-inclusión de consultas conjuntivas constituye un problema decidible o no permanece sin resolver.

Consideramos que la idea de tener un conjunto finito de bases de datos canónicas constituye una aportación capital, pues permite centrar el trabajo futuro en dos líneas complementarias:

Formalización de las restricciones sintácticas. Se trata de estudiar y formalizar las restricciones que limitan las aplicaciones de asignación que se pueden establecer. Cuando se aplican las consultas conjuntivas Q y Q' sobre una de las bases de datos canónicas $d \in \mathcal{CBDC}(Q)$, buscando obtener el hecho derivado t_d, las aplicaciones de asignación posibles dependen, por un lado, de las igualdades que existan entre las variables de las tuplas de la base de datos canónica d, y, por otro lado, de las igualdades entre las variables en la consulta Q o en la consulta Q'.

Es fácil intuir que dichas igualdades imprimen ciertas restricciones a las posibles aplicaciones de asignación. No existe un formalismo que permita identificar patrones de igualdades entre variables de la consulta, por un lado, y de la base de datos, por el otro. Dicho formalismo y el estudio de tales igualdades permitiría extraer conclusiones sobre el número de posibles aplicaciones de asignación. Es evidente que de esto depende la multiplicidad con la que se extrae el hecho derivado t_d.

Procedimiento de comparación de polinomios.

Comparar polinomios de más de una variable en $\mathcal{Z}^+$ constituye un problema indecidible pero los polinomios que, de hecho, se encuentran al aplicar el procedimiento que se describe en esta tesis, tienen ciertas regularidades que nos hacen intuir que son siempre comparables.

Obsérvese que dichos polinomios expresan la multiplicidad del hecho derivado t_d obtenido por Q menos la multiplicidad de ese mismo hecho obtenido por Q', cuando ambas consultas conjuntivas se aplican sobre la misma base de datos canónica d.

De hecho, nos ha resultado imposible encontrar una pareja de consultas conjuntivas Q y Q' que, al aplicarse sobre alguna base de datos canónica $d \in \mathcal{CBDC}(Q)$, produjesen el hecho derivado t_d con multiplicidades expresadas por los polinomios $Poli(Q)$ y $Poli(Q')$ respectivamente, sobre los que no pudiéramos decir si se cumplía o no que $Poli(Q') - Poli(Q) \geq 0$ sobre $\mathcal{Z}^+$.

Es decir, nos ha sido imposible encontrar una pareja de consultas conjuntivas sobre las que no pudiésemos decidir su b-inclusión.

En resumen, estas dos líneas de trabajo futuras se orientan, por un lado, a acotar los patrones de polinomios que, de hecho, pueden darse como expresiones de la diferencia entre las multiplicidades simbólicas cuando dos consultas conjuntivas se aplican sobre una de las bases de datos canónicas, y, por el otro, a producir un método de comparación de polinomios que sea válido para comparar todas las posibles parejas de polinomios que correspondan a los patrones de polinomios encontrados.

La profundización en el estudio de estas dos líneas de trabajo que se proponen, permitiría resolver la pregunta fundamental de si la b-inclusión de consultas conjuntivas constituye o no un problema decidible.

Bibliografía

[1] A.V. Aho, Y. Sagiv , J.D. Ullman. "Equivalence of Relational Expressions". *SIAM J. of Computing 8:2*, pp. 218-246, 1979.

[2] A.V. Aho, Y. Sagiv, J.D. Ullman. "Efficient Optimization of a Class of Relational Consultas". *ACM Transactions on Database Systems 4:4*, pp. 435-454, 1979.

[3] J. Albert. "Algebraic properties of bag data types". *Proc. 17th International Conference on Very Large Data Bases*, pp 211-219, 1991.

[4] N. R. Brisaboa, H. J. Hernández. "Testing Bag.Containment of Conjunctive Queries". *Proc. 2nd. International Congress on Computer Science Research*, pp 105-116, Zacatepec, Mexico, September 1995.

[5] N. R. Brisaboa, H. J. Hernández. "Contenimiento de Consultas Conjuntivas bajo la Semántica de Bolsas". *Actas de las Primeras Jornadas de Investigación y Docencia en Bases de Datos*, pp 160-170, A Coruña, Junio 1996.

[6] N. R. Brisaboa, H. J. Hernández. "Testing Bag.Containment of Conjunctive Queries". *Acta Informatica.* Springer-Verlag. Munich. Alemania. (Próxima publicación).

[7] A.K. Chandra, P.M. Merlin. "Optimal implementation of conjunctive Queries in relational databases". *Proc. of 9th ACM Symposium on Theory of Computing*, pp. 77-90, New York, NY, 1977.

[8] S. Chaudhuri, M. Vardi. "Optimization of Real Conjunctive Consultas," *Proc. Twelfth ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems*, pp. 59-70, Washington, DC, Mayo 1993.

[9] C. J. Date *An introduction to Database Systems*, Addison Wesley, Reading. Massachusetts, 1986.

[10] C. J. Date *A Guide to the SQL Standard.* Addison Wesley, Massachusetts, 1987.

[11] M. Davis. *Computability and Unsolvability.* Dover, NY 1982.

[12] U. Dayal, N. Goodman, R.H. Katz. "An Extended Relational Algebra with Control Over Duplicate Elimination." *Proc. First ACM Symposium on Principles of Database Systems*, pp. 117-123, 1982.

[13] S. Grumbach, T. Milo. "Towards tractable algebras for bags." *Proc. 12th ACM Symposium on Principles of Database Systems*, pp. 49-58, Washington, Mayo 1993.

[14] S. Grumbach, T. Milo, Y.Kornatzky. "Calculi for bags and their complexity". *4th. International Workshop on Database Programming Languages*, Septiembre 1993, Springer Verlag, 1994.

[15] S. Grumbach, L.Libkin, T. Milo and L. Wong "Query Languages for Bags: Expressive Power and Complexity". *SIGART NEWS*, pp. 30-37, Junio 1996.

[16] Y. E. Ioannidis and R. Ramakrishnan. "Generalized Containment of Conjunctive Queries", Computer Science Technical Report, Computer Sciences Department, University of Wisconsin, Madison, WI, 1992.

[17] Y. E. Ioannidis and R. Ramakrishnan "Containment of Conjunctive Queries: Beyond Relations as Sets" *ACM Transactions on Database Systems* 20:3, pp. 288-324, 1995.

[18] Y. E. Ioannidis, R. Ramakrishnan. "Containment of Conjunctive Consultas: Beyond Relations As Sets", Computer Science Technical Report, Computer Sciences Department, University of Wisconsin, Madison, WI, 1994.

[19] A. Klausner. "Multirelations in Relational Databases". Ph.D. Dissertation, Harvard University, 1986.

[20] A. Klug. " On Conjunctive Queries containing inequalities". *ACM Transactions on Database Systems* 35:1, pp.146-160, 1988.

[21] K.L. Kwast, S. van Denneheuvel. 'The meaning of duplicates in the Relational Database Model", Technical Report CT-93-04, Institute for Logic, Language and Computation, Department of Filosofia, University of Amsterdam, 1993.

[22] L. Libkin, L.Wong. "Some Properties of Query Languages for Bags". *Proc. 4th. International Workshop on Database Programming Languages*, Septiembre 1993, Springer Verlag, 1994.

[23] L. Libkin, L.Wong. "New Techniques for studying Set Languages, Bag Languages and Aggregate Functions". *Proc. 13th. ACM Symposium on Principles of Database Systems*, pp. 155-166, 1994.

[24] J. W. Lloyd. *Foundations of Logic Programming*, Second, Extended edition, Springer-Verlag, 1987.

[25] I.S. Mumick et al. " The Magic of duplicates and aggregates". *Proc 16th International Conference on Very Large Data Bases*, Brisbane Australia, 1990.

[26] G. Ozsoyoglu, Z.M. Ozsoyoglu, V. Matos. " Extending Relational algebra and relational calculus with set-values attributes and aggregate functions".*ACM Transactions on Database Systems* 12 pp.566-592, 1987.

[27] J. D. Ullman. *Principles of Database Systems*, Computer Science Press, 1982.

[28] J. D. Ullman. *Principles of Database And Knowledge-base Systems*, Vols 1-2, Computer Science Press, 1988-1989.