Universidad del Bío-Bío, Chile

Facultad de Ciencias Empresariales

Departamento de Sistemas de Información

Algoritmos Polinomiales para Computar Respuestas Aproximadas a Consultas desde Data Warehouses Inconsistentes

Tesis presentada por Juan José Ramírez Lama
para obtener el grado de Magíster en Ciencias de la Computación
Dirigida por:
Dra. Mónica Caniupán, Universidad Del Bío-Bío, Chile
Dra. Loreto Bravo, Universidad de Concepción, Chile

2013

Agradecimientos

Este trabajo no habría sido posible sin el apoyo y el estímulo de mi directora de tesis, Dra. Mónica Caniupán, bajo cuya supervisión escogí este tema y comencé la tesis, un tema que en un principio no comprendía bien y que ahora me ha fascinado con lo que he logrado hacer. Además, agradecer a la Dra. Loreto Bravo quien me ayudó bastante a realizar las definiciones y centrar mi tema durante la etapa en que la profesora Mónica se encontraba en su post-natal.

También me gustaría agradecerle a Raúl Arredondo y Luis Cabrera, por su compañerismo y apoyo en esas interminables horas de estudio y charlas durante el magíster, como durante la tesis.

No puedo terminar sin agradecer a mi familia, en cuyo estímulo constante y amor he confiado a lo largo de mis años de estudio. Estoy agradecido también con mi novia Daniela Villanueva por estar siempre apoyándome y entender que todo este proceso es para algo mejor.

Es a todos ellos a quienes dedico este trabajo.

Abstract

A Data Warehouse (DW) is organized using the multidimensional model where the information is structured according to dimensions. A dimension is an abstract concept that groups data that share a common semantic meaning. The dimensions are modeled using a hierarchical scheme of categories. A dimension is called strict if every element of each category has exactly one ancestor in each parent category, and covering if each element of a category has an ancestor in each parent category. When a dimension is strict and covering we can use pre-computed views to answer queries in an efficient way. If, on the contrary, a dimension is inconsistent with respect to the integrity constraints, that impose these conditions, we can get wrong answers to queries by using pre-computed views. In these cases, it is important to find ways to fix the dimensions (correct them) or find a way to get consistent answers to queries posed on inconsistent dimensions.

A *minimal repair* is a new dimension that satisfies the *strictness* and *covering* integrity constraints, and that is obtained from the original dimension through a minimum number of changes. Repairs are used as a tool to compute consistent answers to aggregation queries. Compute all the minimal repairs for a dimension is NP-hard. Therefore, it becomes interesting to find efficient methods to compute query answers from inconsistent dimensions. In this spirit there exists the *canonical dimension*, that is a new dimension that isolates elements involved in inconsistencies. This new dimension can be used to compute approximate answers to queries. However, to obtain it, it is necessary to compute all the minimal repairs of a dimension.

In this thesis we define a new dimension called the *extended dimension*, which has new elements in the categories. This dimension is used to generate the *compatible repair*, which is a new dimension that is consistent with respect to the integrity constraints and can be used to approximate answers to aggregation queries. We implement polynomial time algorithms to compute the compatible repair. The algorithms do not calculate the minimal repairs of dimensions. Also, we can answer aggregation queries using the aggregation operators `SUM`, `COUNT`, `MAX`, and `MIN`.

Keywords — Data Warehouse, inconsistency, strictness constrains, covering constraints, consistent query answers, aggregation queries, canonical dimension.

Resumen

Un Data Warehouse (DW) se organiza usando el modelo multidimensional donde la información se estructura de acuerdo a dimensiones. Una dimensión es un concepto abstracto que agrupa datos que comparten un significado semántico común. Las dimensiones se modelan mediante un esquema jerárquico de categorías. Una dimensión es *estricta* si todo elemento de cada categoría tiene exactamente un ancestro en cada categoría padre y *homogénea* si cada elemento de las categorías tienen un ancestro en cada categoría padre. Cuando una dimensión satisface las restricciones de integridad (RI), que imponen estas condiciones, se dice que es *consistente* con respecto a las RI, en este caso, es posible utilizar vistas pre-computadas para responder a consultas de manera eficiente. Si al contrario una dimensión es *inconsistente* con respecto a sus RI, al utilizar vistas pre-computadas para computar consultas se pueden obtener respuestas incorrectas. En estos casos es importante buscar formas de reparar (corregir) las dimensiones o encontrar la forma de obtener respuestas consistentes aunque estas sean inconsistentes.

Una *reparación minimal* es una nueva dimensión que satisface las RI estrictas y homogéneas, y es obtenida desde la dimensión original a través de un número mínimo de cambios. Las reparaciones se utilizan como instrumento para computar respuestas consistentes a consultas de agregación. Computar todas las reparaciones minimales para una dimensión inconsistente es NP-complejo. Por lo tanto es interesante generar métodos que permitan computar respuestas a consultas evaluadas en dimensiones inconsistentes de manera eficiente. En esta dirección existe la definición de *reparación canónica*, que es una nueva dimensión que aísla las inconsistencias de la dimensión original y que se obtiene mediante el análisis de todas las reparaciones minimales de una dimensión inconsistente. Esta nueva dimensión puede ser utilizada para computar respuestas aproximadas a consultas.

En esta tesis se define una nueva dimensión *dimensión extendida*, la cual es una nueva dimensión que es consistente con respecto a las RI estrictas y homogéneas, y que permite aproximar respuestas a consultas de agregación. Todo esto se realiza mediante la implementación de algoritmos polinomiales y además, para obtenerla no se realiza el cálculo de todas las reparaciones minimales. Esta dimensión, permite responder de forma aproximada a consultas que utilicen los principales operadores de agregación `SUM`, `COUNT`, `MAX` y `MIN`.

Palabras Clave — Data Warehouse, inconsistencia, restricciones estrictas, restricciones homogéneas, respuestas consistentes, consultas de agregación, dimensión canónica.

Índice general

1. Introducción **1**

2. Preliminares **7**
2.1. Dimensiones y consistencia 7
2.2. Reparación y Limpieza de Dimensiones Inconsistentes 9
2.2.1. Reparaciones y Respuestas Consistentes 9
2.3. Reparación Canónica 12

3. Proyecto de Tesis **15**
3.1. Hipótesis 15
3.2. Objetivos 15
3.2.1. General 15
3.2.2. Específicos 15
3.3. Alcance de la investigación 16
3.4. Estado del Arte 16
3.5. Metodología 18

4. Dimensión Extendida **19**
4.1. Definiciones para la Dimensión Extendida 19
4.2. Dimensión Extendida y Respuestas Aproximadas 21
4.2.1. Respuestas Aproximadas 24

5. Reparación compatible **30**
5.1. Definiciones y Conceptos 30
5.2. Definición y Algoritmo de la Reparación Compatible 34
5.2.1. La Reparación Compatible no es Única 38
5.2.2. Complejidad del Algoritmo 42

6. Experimentos **46**
6.1. Obtención de la Reparación Compatible 46
6.2. Respuestas Aproximadas desde la Reparación Compatible 49
6.3. Ejecución del Algoritmo 1 en un Caso Real 57

7. Conclusión **60**

Referencias **63**

A. Anexos del Capítulo 6 **63**
A.1. Tabla para las dimensiones $\mathcal{D}_e$ y $\mathcal{X}_e$ 63
A.2. Tabla para las dimensiones $\mathcal{D}_{\texttt{FT}}$ y $\mathcal{X}_{\texttt{FT}}$ 66
A.3. Tabla para las dimensiones $\mathcal{D}_{e_1}$ y $\mathcal{X}_{e_1}$ 67

Índice de figuras

1.1. Esquemas de Jerarquía y Dimensiones del Data Warehouse Equipo de Fútbol . . . 2
1.2. Dimensión no estricta y Rollup entre Equipo y Confederación . . . 3
1.3. Algunas de las posibles reparaciones de la Dimensión $\mathcal{D}$ de la Figura 1.2(a). . . . 5

2.1. Reparaciones de la Dimensión $\mathcal{D}$ de la Figura 1.2(a). . . . 10
2.2. Dimensión Canónica de la dimensión $\mathcal{D}$ Equipos de Fútbol . . . 13
2.3. Dimensión Inconsistente con sus reparaciones y la Dimensión Canónica . . . 14

4.1. Esquema y Dimensión Equipos de Fútbol . . . 20
4.2. Ejemplo Dimensión Extendida . . . 23
4.3. Esquema de Jerarquía y Dimensión Extendida $\mathcal{X}_{\mathrm{FT}}$ . . . 26

5.1. Esquemas de Jerarquía y sus niveles de convergencia . . . 31
5.2. Esquema y Dimensión Equipos de Fútbol . . . 32
5.3. Proceso de Reparación de la Dimensión $\mathcal{D}_{\mathrm{FT}}$ de la Figura 5.2 y Reparación Compatible $\mathcal{X}_{\mathrm{FT}}$ obtenida mediante el Algoritmo 1 . . . 35
5.4. Aplicación del Algoritmo para una dimensión inconsistente iniciando la reparación desde distintos elementos en la categoría inferior . . . 38
5.5. Dimensión inconsistente respecto de la restricción Zona $\Rightarrow$ Confederación y su reparación compatible . . . 40
5.6. Dimensión inconsistente respecto de la restricción Zona $\Rightarrow$ Confederación y su reparación compatible (generación de inconsistencia estricta) . . . 40
5.7. Dimensión inconsistente respecto de la restricción Equipo $\rightarrow$ Zona y su reparación compatible . . . 41
5.8. Dimensión inconsistente respecto de la restricción Equipo $\rightarrow$ Confederación y su reparación compatible (múltiple evidencia de consistencia) . . . 42
5.9. Dimensión inconsistente respecto de la restricción Equipo $\rightarrow$ Confederación y su reparación compatible (generación de elemento conjunto) . . . 42

6.1. Esquema y Dimensión $\mathcal{D}_e$ Equipos de Fútbol . . . 47
6.2. Esquema y Reparación compatible $\mathcal{X}_e$ Equipos de Fútbol . . . 48
6.3. Esquema de Jerarquía y Dimensión $\mathcal{D}_{\mathtt{FT}}$ inconsistente . . . 53
6.4. Esquema de Jerarquía y Reparación Compatible $\mathcal{X}_{\mathtt{FT}}$ . . . 54

6.5. Esquema, Dimensión inconsistente $\mathcal{D}_{e_1}$ y Reparación Compatible $\mathcal{X}_{e_1}$ 55
6.6. Esquema jerárquico de la dimensión Teléfonos y cantidad de elementos por categoría 58
6.7. Gráficos obtenidos para computar la reparación compatible en la dimensión Teléfonos 59

Índice de tablas

1.1. Tabla de Hechos Ingresos de la Figura 1.1 2
1.2. Cambios de arco para la Dimensión $\mathcal{D}$ de la Figura 1.2(a) 5
1.3. Respuestas a las consultas de agregación del Ejemplo 1.3 en las reparaciones minimales 5

2.1. Reparación de la Dimensión $\mathcal{D}$ y cambios con respecto de ella. 11

4.1. Tabla de Hechos Ingresos 27
4.2. Respuestas a la consulta sobre la dimensión $\mathcal{X}_{\text{FT}}$ y $\mathcal{X}_{\text{TIME}}$ y la tabla de hechos Ingresos 28
4.3. Respuestas a las consultas de agregación sobre $\mathcal{X}_{\text{FT}}$ y $\mathcal{X}_{\text{TIME}}$ 28

5.1. Restricciones de integridad de la dimensión Equipos de Fútbol $\mathcal{D}_{\text{FT}}$ 33
5.2. Evidencia para ASC para resolver inconsistencia de homogeneidad 45
5.3. Evidencia para AR para resolver inconsistencia estricta 45
5.4. Primera Actualización: Lista de Elementos inconsistentes del Ejemplo 5.5 45
5.5. Segunda Actualización: Lista de Elementos inconsistentes del Ejemplo 5.5 45
5.6. Tercera Actualización: Lista de Elementos inconsistentes del Ejemplo 5.5 45

6.1. Restricciones de integridad de la dimensión Equipos de Fútbol $\mathcal{D}_e$ 47
6.2. Respuestas a la consulta SQL 6.1 agrupada por Confederación y Año para la Figura 6.2 50
6.3. Respuestas aproximadas a la consulta de agregación SQL 6.1 sobre la dimensión extendida de la reparación compatible $\mathcal{X}_e$ y la dimensión Tiempo 51
6.4. Comparación de Respuestas a SQL 6.1 para el operador `sum` y `count` 51
6.5. Comparación de Respuestas a SQL 6.1 para el operador `min` y `max` 52
6.6. Respuestas a SQL 6.1 para la dimensión $\mathcal{D}_{\texttt{FT}}$ 53
6.7. Respuestas a SQL 6.1 para las reparaciones minimales de la dimensión $\mathcal{D}_{\texttt{FT}}$ . . . 53
6.8. Respuestas a SQL 6.1 para la dimensión $\mathcal{X}_{\texttt{FT}}$ 54
6.9. Comparación de respuestas a la consulta SQL 6.1 para los operadores `sum` y `count` 55
6.10. Comparación de respuestas a la consulta SQL 6.1 para los operadores `min` y `max` 56
6.11. Comparación de respuestas a la consulta SQL 6.1 para los operadores `sum` y `count` 57
6.12. Comparación de respuestas a la consulta SQL 6.1 para los operadores `min` y `max` 57

6.13. Resultados de aplicar la reparación compatible a la dimensión Teléfonos en distintos porcentajes de inconsistencia 59

A.1. Tabla de Hechos Ingresos de la dimensión $\mathcal{D}_e$ Equipos de Fútbol 63
A.2. Respuestas a la consulta SQL 6.1 agrupada por Confederación y Año para la Figura 6.1(b) 64
A.3. Respuestas a la consulta SQL 6.1 agrupada por Confederación y Año para la Figura 6.1(b) obtenidas de $\mathcal{A}$ 65
A.4. Rollup entre Equipo y Confederación de la Figura 6.1(b) para los elementos consistentes 65
A.5. Rollup entre Equipo y Confederación de la Figura 6.1(b) 65
A.6. Diferencia entre la $\mathcal{D}_e$ y $\mathcal{X}_e$ en base a la restricción Equipo→Confederación 66
A.7. Rollup entre Equipo y Confederación de la Figura 6.2(b) 66
A.8. Tabla de Hechos Ingresos de la Figura 6.3 66
A.9. Respuesta consistente para $\mathcal{D}_{\mathtt{FT}}$ 67
A.10. Respuestas Aproximadas para $\mathcal{X}_{\mathtt{FT}}$ 67
A.11. Tabla de Hechos Ingresos de la Figura 6.5 67
A.12. Respuestas en la dimensión inconsistente $\mathcal{D}_{e_1}$ 68
A.13. Respuesta consistente para $\mathcal{D}_{e_1}$ 68
A.14. Respuesta Aproximada para $\mathcal{X}_{e_1}$ 68

Capítulo 1

Introducción

Los DataWarehouses (DWs) son *almacenes de datos* que se organizan en dimensiones y hechos, reúnen información de múltiples fuentes y se representan mediante Modelos Multidimensionales. Las dimensiones se modelan como jerarquías de elementos, estos, entregan el orden jerárquico donde cada elemento pertenece a una categoría (**?**). Los hechos corresponden a eventos que se asocian generalmente a valores numéricos conocidos como medidas, y hacen referencia a elementos de dimensión.

Las dimensiones son consideradas la parte estática de los DWs, mientras que los hechos se consideran la parte dinámica, en el sentido de que las operaciones de actualización afectan principalmente a las tablas de hechos (**?**). La estructura multidimensional de un DW permite a los usuarios formular consultas en diferentes niveles de granularidad.

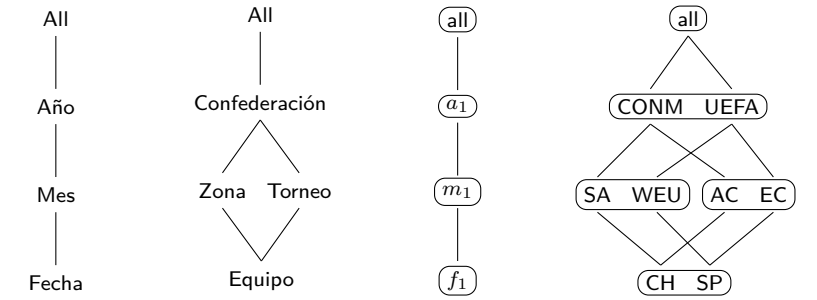
Ejemplo 1.1 *La F.I.F.A.*[1] *posee un DW para administrar la información de las selecciones de fútbol. Para ello ha generado las dimensiones* Equipo de Fútbol *y la dimensión* Tiempo*. La dimensión* Equipo de Fútbol *se organiza de acuerdo a Figura 1.1(b), donde la categoría* Equipo *está conectado (rollup) con* Zona *(Zona Geográfica) y ésta a su vez con* Confederación*. También, la categoría* Equipo *está conectado a la categoría* Torneo *(diferentes competiciones importantes como la Copa América codificada por AC) y ésta a su vez con* Confederación*, esta categoría contiene asociaciones como la Unión Europea de Fútbol Asociada (UEFA). La categoría más alta es* All *a la cual llega* Confederación*. Por otro lado, la dimensión* Tiempo *esta organizada mediante las categorías* Fecha*,* Mes*,* Año *y* All *de acuerdo a la Figura 1.1(a). La Figura 1.1(c) muestra los elementos de la dimensión* Tiempo *junto a las relaciones entre los elementos de las diferentes categorías y de la misma forma la Figura 1.1(d) muestra las relaciones entre los elementos de las categorías de la dimensión* Equipo de Fútbol*. Estas relaciones entre elementos de diferentes categorías se denominan relaciones rollup. Por ejemplo, la relación rollup entre la categoría* Equipo*, y* Zona *posee los pares de elementos (CH, SA) y (SP, WEU).*

La Figura 1.1(d) muestra los elementos de cada categoría en la dimensión Equipo de Fútbol*, y los rollups entre cada uno de ellos. Por ejemplo, los elementos CH (Selección Chilena de Fútbol) y SP (Selección Española de Fútbol) son elementos de la categoría* Equipo *y SA (América del Sur)*

[1] Fédération Internationale de Football Association.

y WEU (Europa del Oeste) son elementos de la categoría Zona. Las relaciones rollup entre estas categorías son los pares (CH, SA), (SP, WEU), (WEU,UEFA) y (SA, CONM).

La tabla de hechos Ingresos (ver Tabla 1.1) almacena los ingresos por concepto de venta, por cada equipo en una fecha en particular. La estructura multidimensional permite a los usuarios computar consultas en diferentes niveles de granularidad. Por ejemplo, para el equipo CH los ingresos por torneo son de $40.000 en el año 2011. También, es sencillo computar resúmenes como: el total de ingresos agrupado por zona y mes, o los ingresos totales por confederación en un año especifico. □



(a) Esquema de Jerarquía Tiempo

(b) Esquema de Jerarquía Equipo de Fútbol

(c) Dimensión Tiempo

(d) Dimensión Equipo de Fútbol

Figura 1.1: Esquemas de Jerarquía y Dimensiones del Data Warehouse Equipo de Fútbol

Ingresos		
Equipo	Fecha	Ingreso
CH	01-01-2011	$40.000
SP	01-10-2011	$30.000
CH	01-01-2012	$60.000
SP	01-10-2012	$30.000

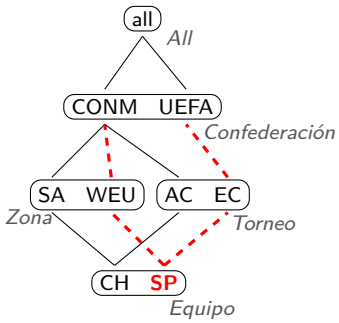
Tabla 1.1: Tabla de Hechos Ingresos de la Figura 1.1

En los DWs se suelen imponer ciertas **restricciones de integridad** (RI) que facilitan la navegación y cómputo de consultas de agregación (**?**), y así comprobar si son coherentes, es decir, si cada hecho se agrega una vez y no más de una vez (**???**). Una dimensión es consistente si satisface todas sus restricciones de integridad, sino lo hace entonces es inconsistente (**?**).

Las restricciones *estrictas* se usan para imponer que las relaciones rollup entre elementos de categorías sean funciones, es decir, todo elemento de una categoría A está conectado a un único elemento en una categoría superior B con A y B conectadas en el esquema jerárquico (de

manera directa o indirecta). Las restricciones *homogéneas* explicitan que las relaciones rollups sean obligatorias para los elementos de dos categorías A y B que están conectadas en el esquema jerárquico (de manera directa o indirecta). Una dimensión que no es homogénea también se conoce como heterogénea.

Ejemplo 1.2 *La dimensión Equipo de Fútbol (Figura 1.1(d)) es consistente con respecto a la restricción de integridad estricta* φ_1 *:Equipo* $\rightarrow$ *Confederación (notación que se introduce en el capítulo 2), que establece que la relación rollup entre Equipo y Confederación debe ser estricta. Sin embargo, la dimensión de la Figura 1.2(a) es inconsistente con respecto a* φ_1*, ya que la serie SP se relaciona transitivamente con dos elementos distintos en la categoría Confederación, tal como se puede apreciar en la tabla de rollup entre los elementos de las categorías Equipo y Confederación.* □



(a) Dimensión Equipo de Fútbol "$\mathcal{D}$" No Estricta

$\mathcal{R}_{\mathcal{D}}^{Confederación}{}_{Equipo}$

Equipo	Confederación
CH	CONM
SP	CONM
SP	UEFA

(b) Rollup entre Equipo y Confederación

Figura 1.2: Dimensión no estricta y Rollup entre Equipo y Confederación

Para el cómputo correcto de consultas de agregación usando o no resultados precomputados, es necesario que las dimensiones cumplan o satisfagan todas sus RI estrictas y homogéneas. Si no es así, es posible que las respuestas a las consultas sean incorrectas o *inconsistentes*, como ocurre con la dimensión inconsistente de la Figura 1.2(a). En el siguiente ejemplo se ilustra el cómputo de consultas en DWs. Las consultas más comunes en los DWs son consultas de agregación con agrupamiento, las que devuelven un valor global por grupo.

Ejemplo 1.3 *[Respuesta inconsistente] Sean las relaciones rollups provenientes de la dimensión Equipo de Fútbol de la Figura 1.2(a) (vistas como tablas):*

$\mathcal{R}_{\mathcal{D}}^{Zona}{}_{Equipo}$

Equipo	*Zona*
CH	*SA*
SP	*WEU*

$\mathcal{R}_{\mathcal{D}}^{Confederación}{}_{Equipo}$

Zona	*Confederación*
SA	*CONM*
WEU	*CONM*

$\mathcal{R}_{\mathcal{D}}{}^{Torneo}_{Equipo}$	
Equipo	Torneo
CH	*AC*
SP	*EC*

$\mathcal{R}_{\mathcal{D}}{}^{Confederación}_{Torneo}$	
Torneo	Confederación
AC	*CONM*
EC	*UEFA*

y la siguiente consulta de agregación SQL que obtiene la suma de los ingresos por torneo, agrupadas por Confederación*, la cual es computada usando las relaciones rollups entre las categorías* Equipo*,* Zona*,* Confederación *y la tabla de hechos* Ingresos *de la Tabla 1.1:*

SQL 1.1: Suma de los ingresos por torneo

```
SELECT R2.Confederacion, SUM(A.Ingreso)
FROM R(Equipo,Zona) R1, R(Zona, Confederacion) R2, Ingresos A
WHERE R1.Equipo = A.Equipo AND R1.Zona = R2.Zona
GROUP BY R2.Confederacion
```

La respuesta a esta consulta evaluada en la dimensión inconsistente de la Figura 1.2(a) es (CONM,160000).

Ahora, si se utilizan las relaciones rollups entre las categorías Equipo*,* Torneo *y* Confederación*, la consulta es de la siguiente forma:*

SQL 1.2: Suma de los ingresos por torneo

```
SELECT R2.Confederacion, SUM(A.Ingreso)
FROM R(Equipo,Torneo) R1, R(Torneo,Confederacion) R2, Ingresos A
WHERE R1.Equipo = A.Equipo AND R1.Torneo = R2.Torneo
GROUP BY R2.Confederacion
```

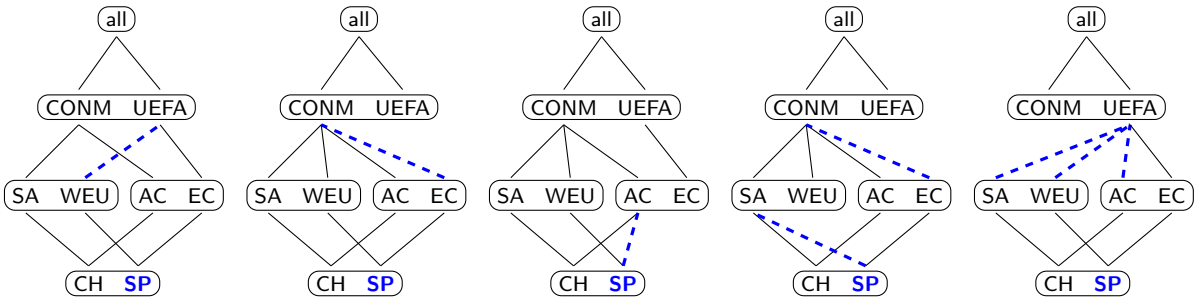
La respuesta a esta consulta es (CONM,100000) y (UEFA,60000).

Si el DW fuera consistente, la respuesta a ambas consultas debería ser la misma. Sin embargo, debido a la inconsistencia en la dimensión se obtienen distintas respuestas a la consulta. □

Para solucionar lo expresado en el Ejemplo 1.3 es necesario corregir o reparar la inconsistencia en la dimensión afectada. Para ello es posible modificar las relaciones rollups. Esto debido a que es usual asumir que las restricciones prevalecen sobre los datos y por lo tanto la dimensión debe ser actualizada para satisfacer las restricciones (**?**).

Para el caso de la dimensión presentada en la Figura 1.2(a) existen al menos 5 reparaciones posibles, las que se aprecian en la Figura 1.3. Estas reparaciones se generan mediante inserción y eliminación de arcos entre elementos de las categorías de la dimensión inconsistente (ver Tabla 1.2). En (**?**) se ha demostrado que el número de reparaciones para una dimensión inconsistente respecto a restricciones estrictas y homogéneas puede ser exponencial.

De acuerdo a la Tabla 1.2 hay reparaciones que involucran más cambios que otras. Una reparación es *minimal* si para obtenerla se realiza un número mínimo de cambios, en el ejemplo,



(a) Reparación $\mathcal{D}_1$ (b) Reparación $\mathcal{D}_2$ (c) Reparación $\mathcal{D}_3$ (d) Reparación $\mathcal{D}_4$ (e) Reparación $\mathcal{D}_5$

Figura 1.3: Algunas de las posibles reparaciones de la Dimensión $\mathcal{D}$ de la Figura 1.2(a).

Operaciones de la Reparación		
Dimensión	Arcos Eliminados en $\mathcal{D}$	Arcos Insertados en $\mathcal{D}'$
$\mathcal{D}_1$	(WEU,CONM)	(WEU,UEFA)
$\mathcal{D}_2$	(EC,UEFA)	(EC,CONM)
$\mathcal{D}_3$	(SP,EC)	(SP,AC)
$\mathcal{D}_4$	(EC,UEFA), (SP,WEU)	(EC,CONM), (SP,SA)
$\mathcal{D}_5$	(SA,CONM), (WEU,CONM), (AC,CONM)	(SA,UEFA), (WEU,UEFA), (AC,UEFA)

Tabla 1.2: Cambios de arco para la Dimensión $\mathcal{D}$ de la Figura 1.2(a)

son reparaciones minimales para $\mathcal{D}$, las reparaciones $\mathcal{D}_1$, $\mathcal{D}_2$ y $\mathcal{D}_3$ con dos cambios cada una. Las reparaciones $\mathcal{D}_4$ y $\mathcal{D}_5$ no son minimales ya que para obtenerlas realizamos más cambios de los necesarios.

Si computamos las consultas de agregación SQL 1.1 y SQL 1.2 presentadas en el Ejemplo 1.3 obtenemos las respuestas que se muestran en la Tabla 1.3 para cada una de las reparaciones minimales de la Figura 1.3. Como se puede apreciar, las respuestas para las consultas (SQL 1.1 y 1.2) son iguales en cada reparación.

Respuestas a las consultas de agregación		
Dimensión	SQL 1.1	SQL 1.2
$\mathcal{D}_1$	(CONM,100000), (UEFA,60000)	(CONM,100000), (UEFA,60000)
$\mathcal{D}_2$	(CONM,160000)	(CONM,160000)
$\mathcal{D}_3$	(CONM,160000)	(CONM,160000)

Tabla 1.3: Respuestas a las consultas de agregación del Ejemplo 1.3 en las reparaciones minimales

El concepto de respuesta consistente con respecto a RI estrictas y homogéneas para consultas de agregación con agrupamiento de datos fue definido inicialmente en (**?**). Una respuesta consistente para este tipo de consultas es un rango para cada valor de grupo que contiene los va-

lores obtenidos en cada una de las reparaciones minimales. Para la consulta anterior la respuesta consistente es (CONM, [100000,160000]). No hay respuesta consistente para UEFA ya que para este grupo no hay respuesta en todas las reparaciones minimales. Éste y otros conceptos serán formalizados en el Capitulo 2.

Es posible utilizar métodos basados en lógica para computar respuestas consistentes a consultas de agregación, desde dimensiones inconsistentes con respecto a RI estrictas y homogéneas (**?**). Sin embargo, no es posible pensar en la aplicación de estos programas sobre una dimensión real que pueda tener terabytes de datos. En (**?**) se presenta un método que computa respuestas aproximadas a consultas de agregación, basadas en una única dimensión que denominan la *dimensión canónica*. Esta dimensión se construye en base a las reparaciones minimales de la dimensión original inconsistente.

En esta tesis se define una nueva dimensión denominada *dimensión extendida*, la cual es una nueva dimensión que permite almacenar conjuntos de elementos en sus categorías, esta dimensión es la base para obtener la *reparación compatible* la cual, a diferencia de la dimensión canónica, no se basa en las reparaciones minimales de la dimensión inconsistente. Además, la dimensión extendida permite computar respuestas aproximadas a consultas de agregación considerando los operadores de agregación `SUM`, `COUNT`, `MAX` y `MIN`.

Esta tesis se organiza de la siguiente manera: En el Capítulo 2 se definen conceptos referente a DWs, reparaciones de una dimensión, restricciones de integridad, respuesta consistente, entre otros. En el Capítulo 3 se presenta la propuesta de esta tesis, su hipótesis, objetivos, alcance y estado del arte. En el Capítulo 4 se definen los conceptos de dimensión extendida y respuestas a consultas en este tipo de dimensiones aplicando distintos operadores de agregación. En el Capítulo 5 se define la reparación compatible y se presentan algoritmos para obtener la misma y su complejidad computacional. En el Capítulo 6 presentan ejemplos y experimentos de la aplicación del algoritmo de la reparación compatible. Finalmente en el Capítulo 7 se presentan las conclusiones y resumen de lo obtenido en esta tesis.

Capítulo 2

Preliminares

En este capítulo se presentan las definiciones necesarias para comprender mejor el ámbito de los DWs (Sección 2.1). Posteriormente se define el concepto de reparación (Sección 2.2), y Por último se presenta el concepto de dimensión canónica (Sección 2.3) el cual es una dimensión que sirve como auxiliar para obtener respuestas aproximadas a consultas de agregación.

2.1. Dimensiones y consistencia

Definición 2.1 (Esquema de Jerarquía (?)) Un *esquema de jerarquía* $\mathcal{H}$ se define como un par $(\mathcal{C}_\mathcal{H}, \nearrow_\mathcal{H})$, donde $(\mathcal{C}_\mathcal{H}, \nearrow_\mathcal{H})$ es un grafo acíclico dirigido. Los vértices en $\mathcal{C}_\mathcal{H}$ son categorías y los arcos $\nearrow_\mathcal{H}$ son las relaciones hijo/padre entre las categorías. La clausura transitiva y reflexiva de $\nearrow_\mathcal{H}$ se denota por $\nearrow^*_\mathcal{H}$. El conjunto de las categorías $\mathcal{C}_\mathcal{H}$ tiene una categoría *superior* $\mathsf{All}_\mathcal{H}$, que se accede desde cualquier otra categoría en $\mathcal{C}_\mathcal{H}$ y no existe ninguna categoría que sea superior a ella. □

Ejemplo 2.1 *Para el esquema de la Figura 1.1(b) se tiene lo siguiente:*
$\mathcal{H} = (\mathcal{C}_\mathcal{H}, \nearrow_\mathcal{H})$ *donde*
$\mathcal{C}_\mathcal{H}$ = {*Equipo, Zona, Torneo, Confederación, All*},
$\nearrow_\mathcal{H}$ = {*(Equipo, Zona), (Equipo, Torneo), (Zona, Confederación), (Torneo, Confederación), (Confederación, All)*},
$\nearrow^*_\mathcal{H} = \nearrow_\mathcal{H} \bigcup$ {*(Torneo, Torneo), (Equipo, Equipo), (Confederación, Confederación), (Equipo, Confederación), (Equipo, All),. . .,*} □

Definición 2.2 (Dimensión (?)) Una *dimensión* $\mathcal{D}$ es una tupla $(\mathcal{H}_\mathcal{D}, \mathcal{E}_\mathcal{D}, \mathsf{Cat}_\mathcal{D}, <_\mathcal{D})$, donde $\mathcal{H}_\mathcal{D} = (\mathcal{C}_{\mathcal{H}_\mathcal{D}}, \nearrow_{\mathcal{H}_\mathcal{D}})$ es un esquema de jerarquía, $\mathcal{E}_\mathcal{D}$ son los elementos; $\mathsf{Cat}_\mathcal{D} : \mathcal{E}_\mathcal{D} \to \mathcal{C}_{\mathcal{H}_\mathcal{D}}$ es una función de pertenencia de un elemento a su categoría, y $<_\mathcal{D} \subseteq \mathcal{E}_\mathcal{D} \times \mathcal{E}_\mathcal{D}$ son las relaciones hijo/padre entre los elementos de diferentes categorías. $<^*_\mathcal{D}$ denota la clausula reflexiva y transitiva de $<_\mathcal{D}$. Las siguientes condiciones se deben satisfacer:

(a) all es el único elemento de la categoría All.
(b) Por cada par de elementos $a, b \in \mathcal{E}_\mathcal{D}$ si $a <_\mathcal{D} b$, entonces $\mathsf{Cat}_\mathcal{D}(a) \nearrow_\mathcal{H} \mathsf{Cat}_\mathcal{D}(b)$. □

La condición (b) es importante ya que asegura que la relación hijo/padre se refiera a elementos de categorías que están conectadas en el esquema jerárquico.

Ejemplo 2.2 *En la Figura 1.1(d) la dimensión* $\mathcal{D} = (\mathcal{H}_\mathcal{D}, \mathcal{E}_\mathcal{D}, \mathsf{Cat}_\mathcal{D}, <_\mathcal{D})$ *es:*
$\mathcal{E}_\mathcal{D}$ = {*all, CONM, UEFA, SA, WEU, AC, EC, CH, SP*}
$\mathsf{All}_\mathcal{D}$ = *all*
$\mathsf{Cat}_\mathcal{D}$ = {*all* $\mapsto$ *All, CONM* $\mapsto$ *Confederación, UEFA* $\mapsto$ *Confederación, SA* $\mapsto$ *Zona, WEU* $\mapsto$ *Zona, AC* $\mapsto$ *Torneo, EC* $\mapsto$ *Torneo, CH* $\mapsto$ *Equipo, SP* $\mapsto$ *Equipo*}
$<_\mathcal{D}$ = {*(CH,SA), (SP,WEU), (CH, AC), (SP, EC), (SA, CONM), (WEU, UEFA), (AC,CONM), (EC, UEFA), (CONM, all), (UEFA, all)*}
$<^*_\mathcal{D}$ = $<_\mathcal{D} \bigcup$ {*(CH,CH), (SP,SP), (CH, CONM), (SP,UEFA),...*} □

El conjunto de relaciones rollup de la dimensión $\mathcal{D}$ está definida de la siguiente forma.

Definición 2.3 (Relaciones rollup (?)) Para cada par de categorías $c_i, c_j \in \mathcal{C}_{\mathcal{H}_\mathcal{D}}$ tal que $c_i \nearrow^*_{\mathcal{H}_\mathcal{D}} c_j$, se denota la relación rollup $\mathcal{R}_\mathcal{D}(\mathsf{c}_i, \mathsf{c}_j)$ que tiene el siguiente conjunto de pares $\{(a,b) | \mathsf{Cat}_\mathcal{D}(a) = \mathsf{c}_i, \mathsf{Cat}_\mathcal{D}(b) = \mathsf{c}_j \text{ y } a <^*_\mathcal{D} b\}$. □

Ejemplo 2.3 *Sea* $\mathcal{D}$ *la dimensión de la Figura 1.2(a). La relación rollup* $\mathcal{R}_\mathcal{D}$*(Equipo, Zona),* $\mathcal{R}_\mathcal{D}$*(Equipo, Torneo),* $\mathcal{R}_\mathcal{D}$*(Zona, Confederación) y* $\mathcal{R}_\mathcal{D}$*(Torneo, Confederación) son tal cual a las presentadas en el Ejemplo 1.3.* □

Definición 2.4 (Relaciones Rollup Estrictas y Homogéneas (?)) Sea $\mathcal{R}_\mathcal{D}(c_i, c_j)$ una relación rollup, entonces:

(a) $\mathcal{R}_\mathcal{D}(c_i, c_j)$ es estricta si para todos los elementos $x, y, z \in \mathcal{E}_\mathcal{D}$, si $(x, y) \in \mathcal{R}_\mathcal{D}(c_i, c_j)$ y $(x, z) \in \mathcal{R}_\mathcal{D}(c_i, c_j)$ entonces $y = z$

(b) $\mathcal{R}_\mathcal{D}(c_i, c_j)$ es homogénea si para todos los elementos $e \in \mathcal{E}_\mathcal{D}$ de tal forma que $\mathsf{Cat}_\mathcal{D}(e) = c_i$, existe un elemento $e' \in \mathcal{E}_\mathcal{D}$ y $\mathsf{Cat}_\mathcal{D}(e') = c_j$, tal que $(e, e') \in \mathcal{R}_\mathcal{D}(c_i, c_j)$. □

Una rollup es *estricta* si es una función (en caso que no se cumpla es una *relación no estricta*). Luego una relación rollup es *Homogénea* si cada elemento de una categoría inferior está relacionado con algún elemento en una categoría superior (si no se cumple esto es una *relación no Homogénea*).

En general, una dimensión es estricta si todas las relaciones rollups son estrictas. De otra forma, la dimensión no es estricta. Una dimensión es homogénea si todas las relaciones rollups son homogéneas. Una dimensión que no es homogénea también se conoce como heterogénea.

Definición 2.5 (Restricciones Estrictas y Homogéneas (?)) Sea $\mathcal{H} = (\mathcal{C}_\mathcal{H}, \nearrow_\mathcal{H})$ un esquema jerárquico y $\mathcal{D} = (\mathcal{H}_\mathcal{D}, \mathcal{E}_\mathcal{D}, \mathsf{Cat}_\mathcal{D}, <_\mathcal{D})$ una dimensión, tal que: $\mathcal{H}_\mathcal{D} = \mathcal{H}$.

(a) Una restricción estricta sobre $\mathcal{H}$ es una expresión de la forma $c_i \to c_j$ donde c_i, $c_j \in \mathcal{C}_\mathcal{H}$ y $c_i \nearrow^*_\mathcal{H} c_j$. La dimensión $\mathcal{D}$ satisface la restricción $c_i \to c_j$ sí y solo sí la relación rollup $\mathcal{R}_\mathcal{D}(c_i, c_j)$ es estricta.

(b) Una restricción homogénea sobre $\mathcal{H}$ es una expresión de la forma $c_i \Rightarrow c_j$ donde $c_i, c_j \in \mathcal{C}_\mathcal{H}$ y $c_i \nearrow^*_\mathcal{H} c_j$. La dimensión $\mathcal{D}$ satisface la restricción $c_i \Rightarrow c_j$ sí y solo sí la relación rollup $\mathcal{R}_\mathcal{D}(c_i, c_j)$ es homogénea. □

Se denota por $\Sigma_e(\mathcal{H})$ y $\Sigma_h(\mathcal{H})$, al conjunto de todas las posibles restricciones estrictas y homogéneas (respectivamente) para un esquema jerárquico (**?**).

Es posible verificar si una dimensión $\mathcal{D}$ satisface un conjunto de restricciones estrictas y homogéneas en tiempo polinomial (**?**). $\mathcal{D} \models \Sigma$ denota que la dimensión $\mathcal{D}$ es consistente con respecto a las RI en Σ, $\mathcal{D} \not\models \Sigma$ denota que la dimensión $\mathcal{D}$ es inconsistente con respecto a Σ.

Ejemplo 2.4 *Considere la dimensión $\mathcal{D}$ en la Figura 1.2(a). La dimensión satisface el siguiente conjunto Σ de restricciones:*

Restricción Homogenea (Σ_h) *{Zona ⇒ Confederación, Torneo ⇒ Confederación, Equipo ⇒ Zona y Equipo⇒ Torneo}, esta dimensión es homogenea, puesto que cumple todas estas restricciones.*

Restricción Estricta (Σ_e) *{Zona → Confederación, Torneo → Confederación, Equipo → Zona y Equipo → Torneo}. Sin embargo, $\mathcal{D}$ no satisface la restricción Equipo → Confederación, ya que el elemento SP de la categoría Equipo está conectado a los elementos CONM y UEFA de la categoría Confederación.* □

2.2. Reparación y Limpieza de Dimensiones Inconsistentes

Existen distintas técnicas para reparar dimensiones inconsistentes respecto a sus restricciones de integridad. Una de ellas es la propuesta en (**?**) la cual permite eliminar inconsistencias respecto de restricciones estrictas, mediante inserción de elementos y arcos [1]. Otra técnica es la propuesta en (**?**) la cual implica el proceso de inserción y eliminación de arcos entre los elementos de las categorías conflictivas, para restaurar la consistencia respecto de restricciones estrictas y homogéneas.

A medida que la cantidad de elementos involucrados en la inconsistencia crece, la cantidad de posibles reparaciones aumenta de forma muchas veces exponencial, lo que lleva a la necesidad de un método que permita seleccionar y luego unificar todas las reparaciones que involucren un mínimo número de cambios y que permita obtener respuestas aproximadas. Para solucionar esto, en (**?**) se plantea el concepto de reparación canónica el cual se estudia en la Sección 2.3.

2.2.1. Reparaciones y Respuestas Consistentes

La semántica de reparación que se utiliza para corregir inconsistencias es la *agregación* y *eliminación* de arcos entre los elementos de las categorías conflictivas (**?**).

Para definir el concepto de reparación es necesario definir el concepto de *distancia* entre dos dimensiones (**?**).

[1] Esta técnica se revisa en el estado del arte expuesto en el Capítulo 3.

Definición 2.6 (Distancia (?)) Sean $\mathcal{D} = (\mathcal{H}_\mathcal{D}, \mathcal{E}_\mathcal{D}, \mathsf{Cat}_\mathcal{D}, <_\mathcal{D})$ y $\mathcal{D}' = (\mathcal{H}'_\mathcal{D}, \mathcal{E}'_\mathcal{D}, \mathsf{Cat}'_\mathcal{D}, <'_\mathcal{D})$ dos dimensiones sobre un mismo esquema de jerarquía, la distancia entre ellas se define cómo $dist(\mathcal{D}, \mathcal{D}') = |(<'_\mathcal{D} \setminus <_\mathcal{D}) \bigcup (<_\mathcal{D} \setminus <'_\mathcal{D})|$. Es decir, la distancia es el tamaño de la diferencia simétrica entre las relaciones hijo/padre de las dimensiones. □

Este concepto se establece para poder determinar cómo difieren dos dimensiones en base a las eliminación y/o agregación de arcos.

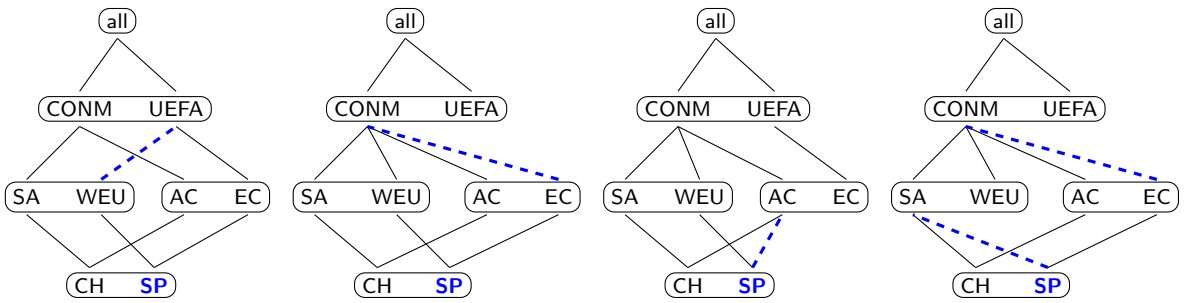
Definición 2.7 (Reparación y Reparación Minimal (?)) Sea $\mathcal{D} = (\mathcal{H}_\mathcal{D}, \mathcal{E}_\mathcal{D}, \mathsf{Cat}_\mathcal{D}, <_\mathcal{D})$ una dimensión y Σ un conjunto de restricciones de integridad en $\mathcal{H}_\mathcal{D}$.

(a) Una *reparación* de $\mathcal{D}$ con respecto de Σ, es una nueva dimensión que satisface Σ.

(b) Una reparación *minimal* de $\mathcal{D}$ con respecto a Σ es una reparación $\mathcal{D}'$, tal que $dist(\mathcal{D}, \mathcal{D}')$ es mínima entre todas las reparaciones de $\mathcal{D}$ respecto a Σ.

Denotaremos por $Rep(\mathcal{D})$ al conjunto de reparaciones minimales. □

Dicho de otra forma una reparación es una nueva dimensión $\mathcal{D}'$ con los mismos elementos y categorías denotadas en la dimensión $\mathcal{D}$, pero que satisface por completo el conjunto de restricciones de integridad en Σ.



(a) Reparación Minimal $\mathcal{D}_1$ (b) Reparación Minimal $\mathcal{D}_2$ (c) Reparación Minimal $\mathcal{D}_3$(d) Reparación No Minimal $\mathcal{D}_4$

Figura 2.1: Reparaciones de la Dimensión $\mathcal{D}$ de la Figura 1.2(a).

Ejemplo 2.5 (Reparación Minimal y Distancia) *Considere la dimensión $\mathcal{D}$ presentada en la Figura 1.2(a) y Σ = {Zona → Confederación, Torneo → Confederación, Equipo → Confederación}. En la Figura 2.1 se presentan reparaciones para la dimensión $\mathcal{D}$ que es inconsistente respecto de Σ, puesto que viola la restriccion Equipo→Confederación. Las reparaciones son obtenidas mediante inserciones y eliminaciones de arcos presentados en la Tabla 2.1. La distancia entre la dimensión inconsistente de la Figura 1.2(a) y cada una de las reparaciones de la Figura 2.1 están representadas en la ultima columna de la Tabla 2.1. Además, se puede decir que $\mathcal{D}_1$, $\mathcal{D}_2$ y $\mathcal{D}_3$ son reparaciones minimales, pero $\mathcal{D}_4$ no lo es ya que posee una distancia mayor con respecto a la dimensión original $\mathcal{D}$ (mayor número de cambios). Por lo tanto la dimensión inconsistente de la Figura 1.2(a) posee solo tres reparaciones minimales.* □

Operación de la Dimensión			
Dimensión	Arcos Eliminados	Arcos Insertados	Distancia a $\mathcal{D}$ (Δ)
$\mathcal{D}_1$	(WEU,CONM)	(WEU,UEFA)	2
$\mathcal{D}_2$	(EC,UEFA)	(EC,CONM)	2
$\mathcal{D}_3$	(SP,EC)	(SP,AC)	2
$\mathcal{D}_4$	(EC,UEFA), (SP,WEU)	(EC,CONM), (SP,SA)	4

Tabla 2.1: Reparación de la Dimensión $\mathcal{D}$ y cambios con respecto de ella.

Una vez determinadas las reparaciones minimales que posee una dimensión inconsistente, es posible realizar en cada una de ellas la consulta de agregación *ingresos por arriendo*, con lo que se obtienen los siguientes resultados: en la reparación minimal D_1 se obtiene la respuesta (CONM, 100000), (UEFA,60000), para la reparación minimal D_2 la respuesta es (CONM,160000) y para D_3 la respuesta es (CONM,160000).

El problema que surge en este punto es que cada una de las reparaciones minimales responden algo distinto (no se descarta que en algunos casos todas pudiesen responder de igual forma, pero eso es muy poco común), para casos como este, en (**?**) definió el concepto de *respuesta consistente* a una consulta de agregación escalar, de la cual (**?**) se basa y extiende para consultas de agregación con agrupamiento de datos. Una respuesta consistente a una consulta de agregación con agrupamiento de datos, es un rango para cada grupo que contiene los valores obtenidos en todas las reparaciones minimales.

Definición 2.8 (Respuesta consistente (?)) Dada una dimensión $\mathcal{D}$ y una consulta de agregación $\mathcal{Q}$, una tupla de la forma $\langle t_1, ..., t_n, [a, b]\rangle$ es una respuesta consistente a la consulta $\mathcal{Q}$ si:

(a) $[a, b]$ es un intervalo numérico;
(b) por cada reparación minimal $\mathcal{D}'$ de $\mathcal{D}$ la tupla $\langle t_1, ..., t_m, f(t_1, ..., t_m)\rangle$ es una respuesta a $\mathcal{Q}$ en $\mathcal{D}'$ y $f(t_1, ..., t_n) \in [a, b]$;
(c) no hay un intervalo más pequeño $[a', b']$ para los que la condición (b) se cumpla. □

Ejemplo 2.6 *Las respuestas a la consulta SQL 1.1 obtenidas en las reparaciones minimales de la Figura 2.1 son: (CONM, 100000), (UEFA,60000) en D_1, (CONM,160000) en D_2 y (CONM,160000) en D_3. Por lo tanto, la respuesta consistente para la consulta es (CONM, [100000,160000]). Esto es así porque solo hay respuesta en cada reparación para el valor CONM, donde en la reparación minimal D_1 la respuesta es* 100000*, que constituye el valor más pequeño del rango consistente, y la respuesta es* 160000 *en D_2 y D_3, que constituye el rango más alto. No hay respuesta consistente para el valor UEFA ya que no existe respuesta para este valor en todas las reparaciones minimales.* □

Muchas veces puede ser costoso calcular las respuestas consistentes a consultas mediante el cómputo de todas las reparaciones minimales, lo que sería un enfoque inspirado directamente en la definición de respuesta consistente. En la siguiente sección se presenta la propuesta de una dimensión canónica para intentar evitar el inconveniente de tener que consultar a cada una de

las reparaciones minimales y después tener que calcular la respuesta consistente. Para resolver este problema se crea una dimensión que resume todas las reparaciones minimales encontradas.

2.3. Reparación Canónica

En el trabajo de (**?**) se define el concepto de *dimensión canónica*, que es una dimensión única que se construye en base a las reparaciones minimales de una dimensión $\mathcal{D}$ que viola restricciones estrictas (se asume que las restricciones homogéneas están satisfechas). Se propone usar esta dimensión para el cómputo de respuestas a consultas de agregación.

La *dimensión canónica* se obtiene mediante el aislamiento de los elementos implicados en las inconsistencias. Por ejemplo, si hay un elemento a que hace un rollup a b_1 en una reparación, y a b_2 en otra; en la dimensión canónica, podemos añadir un elemento $\{b_1, b_2\}$ y hacer que a rollups al elemento $\{b_1, b_2\}$. La dimensión canónica está siempre polinomialmente limitada por el tamaño de la dimensión no estricta (**?**).

Intuitivamente, la dimensión canónica separa los elementos involucrados en las inconsistencias de las que no lo son. El dominio de la dimensión canónica difiere del dominio de la dimensión original, pero todavía se ajusta al mismo esquema. Además las categorías inferiores siempre tienen los mismo elementos que la dimensión original, y por lo tanto, las tablas de hechos pueden ser utilizadas (**?**).

Ejemplo 2.7 *En la Figura 1.2(a) se presenta una dimensión inconsistente para el esquema jerárquico de la Figura 1.1(b). Esta dimensión viola la restricción estricta* Equipo → Confederación *puesto que transitivamente SP se relaciona con dos elementos distintos en* Confederación*. Las reparaciones minimales se muestran en la Figura 2.1(a)-(c).*

La Figura 2.2 muestra la dimensión canónica de la dimensión Equipos de Fútbol*, esta dimensión es generada en base a las tres reparaciones minimales posibles para $\mathcal{D}$. En la dimensión canónica SP hace rollup al elemento {AC, EC}, ya que en la reparación minimal $\mathcal{D}_3$, SP hace rollup al elemento AC y en las reparaciones minimales $\mathcal{D}_1$ y $\mathcal{D}_2$, SP hace rollup a EC, así mismo, WEU y EC hacen rollup al elemento {CONM, UEFA} puesto que en la reparación $\mathcal{D}_1$, WEU hace rollup a UEFA y en la reparación $\mathcal{D}_2$, EC hace rollup a CONM. Por lo tanto, la dimensión canónica contiene los nuevos arcos (SP, {AC, EC}), ({AC, EC}, {CONM,UEFA}), (WEU, {CONM, UEFA}) y (EC, {CONM, UEFA}).* □

De acuerdo a lo presentado en (**?**) las respuestas consistentes a una consulta $\mathcal{Q}$ en $\mathcal{D}$ se pueden aproximar por medio de una forma de composición de respuestas a $\mathcal{Q}$ obtenidas de la dimensión canónica para $\mathcal{D}$.

Una vez generada la dimensión canónica podemos comparar la calidad de su respuesta con las reparaciones minimales anteriores (ver Figura 2.1(a)-(c)).

Ejemplo 2.8 *Las respuesta a la consulta de agregación SQL 1.1 evaluada en la dimensión canónica es: (CONM,100000), ({CONM, UEFA},60000). Por lo tanto, de acuerdo a (**?**) la respuesta aproximada es: (CONM,[100000,160000]), ya que se asegura que para CONM existen $100.000 de ingresos por arriendo, y los otros $60.000 corresponden al elemento conjunto del cual es parte CONM. Esta respuesta coincide con la respuesta consistente a la consulta.* □

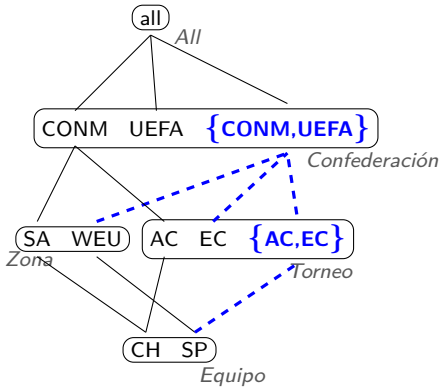


Figura 2.2: Dimensión Canónica de la dimensión $\mathcal{D}$ Equipos de Fútbol

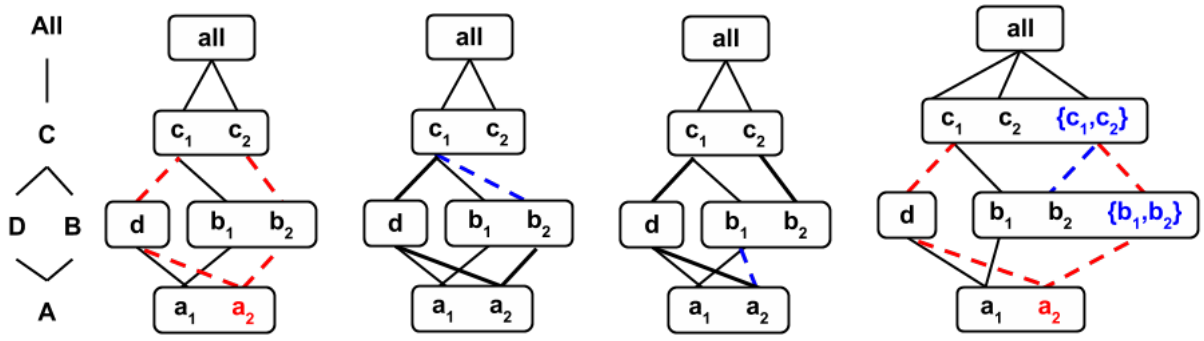
La dimensión canónica del ejemplo anterior es estricta, por lo tanto, las respuestas obtenidas de ella serán las mismas independientemente del uso de las vistas pre-computadas. Sin embargo, podría ocurrir que la dimensión canónica sea inconsistente con respecto a las restricciones de integridad estrictas, como se muestra en el Ejemplo 2.9. Sin embargo, en (**?**) se demuestra que a pesar de esto la dimensión canónica también puede ser utiliza para el cómputo de respuestas aproximadas a las respuestas consistentes esperadas para consultas con operadores `SUM` y `COUNT`.

Ejemplo 2.9 *(**?**) Considere la dimensión inconsistente de la Figura 2.3, la tabla de hechos Hechos(A,N) = {(a_1,1), (a_2,2), (a_2,2), (a_1,2)} y la consulta de agregación Q':* `SELECT` *R.C, SUM(Hechos.N)* `FROM` *Hechos,* $\mathcal{R}_A^C$ *R* `WHERE` *Hechos.A = R.A* `GROUP BY` *R.C. La respuesta consistentes a la consulta anterior, obtenida a partir de las reparaciones* $\mathcal{D}_{1'}$ *y* $\mathcal{D}_{2'}$ *es* $(c_1, 7)$.

En las reparaciones, el elemento a_2 *hace rollup a diferentes elementos en la categoría B pero al mismo elemento en la categoría C. Como resultado, la dimensión canónica (ver Figura 2.3(d)) no es estricta. Las respuestas obtenidas en la dimensión canónica sin el uso de vistas pre-computadas son* $(c_1, 7)$, $(\{c_1, c_2\}, 4)$ *con lo que la respuesta aproximada es* $(c_1, [7, 11])$. *Por otra parte, mediante el uso de vistas pre-computadas de la categoría D se obtiene* $(c_1, 7)$, *que está incluida en la respuesta aproximada* $(c_1, [7, 11])$. *Mediante el uso de vistas pre-computadas desde B se tiene* $(c_1, 3)$, $(\{c_1, c_2\}, 4)$ *que resulta en la respuesta compuesta* $(c_1, [3, 7])$ *que también está incluida en el rango* $(c_1, [7, 11])$ *considerando el rango superior 7 y el inferior de la respuesta aproximada.* □

La definición de Dimensión Canónica presentada en (**?**) depende del cómputo de las reparaciones minimales de una dimensión, lo que provoca que pueda ser demasiado costoso obtener todas estas reparaciones y luego procesarlas para generar la nueva dimensión canónica. Por otra parte, no se formaliza la dimensión canónica, la cual contiene elementos que corresponden a conjuntos de elementos. Además, los autores se restringen a los operadores de agregación `SUM` y `COUNT`. En esta tesis y como se plantea más adelante, se redefinen los conceptos para adaptarlos a este nuevo tipo de dimensiones que permiten la inserción de elementos conjunto en las categorías, se propone un método para obtener respuestas aproximadas sin la necesidad de computar todas las

reparaciones minimales y además se define el tipo de respuestas que genera para los operadores de agregación `SUM`, `COUNT`, `MIN` y `MAX`.



(a) Esquema y $\mathcal{D}'$ (b) $\mathcal{D}'_1$ (c) $\mathcal{D}'_2$ (d) Dimensión Canónica

Figura 2.3: Dimensión Inconsistente con sus reparaciones y la Dimensión Canónica

A continuación se presentan la hipótesis y objetivos de esta tesis, como también se expone el estado del arte de lo que concierne a DWs, reparación de dimensiones inconsistentes, respuestas consistentes, entre otros.

Capítulo 3

Proyecto de Tesis

En este capítulo se presenta todo el tema a tratar, como se abordará el problema, su estado actual, alcance, entre otros. Todo para dejar claro el enfoque y lo que se espera lograr en esta Tesis.

3.1. Hipótesis

Es posible definir una dimensión que permita contener elementos conjuntos en sus categorías, que aísle la información inconsistente de la dimensión, no dependa del cómputo de las reparaciones minimales, y permita responder de forma aproximada a consultas de agregación evaluadas sobre DW inconsistente con respecto a sus RI estrictas y homogéneas.

3.2. Objetivos

3.2.1. General

El objetivo de esta Tesis es definir e implementar una dimensión que permita almacenar elementos conjuntos en sus categorías, que en base a una dimensión inconsistente, sea capaz de generar una dimensión consistente respecto a sus RI para computar respuestas a consultas de agregación.

3.2.2. Específicos

- Formalizar el tipo de dimensión que permita agregar conjuntos de elementos en sus categorías.
  - Esta dimensión debe aislar los elementos inconsistentes de la dimensión original y ser obtenida sin computar las reparaciones minimales.
  - Debe ser consistente con respecto de las RI estrictas y homogéneas.
- Definir la *reparación compatible*.
- Generar algoritmos que permitan computar la reparación compatible.

- Analizar el orden de los algoritmos.
- Probar que los algoritmos son correctos.
- Analizar la calidad de las respuestas a consultas obtenidas desde la reparación compatible.
- Realizar experimentos que muestre el rendimiento del algoritmo.

3.3. Alcance de la investigación

El algoritmo para obtener la reparación compatible podría considerar una clase especial de esquema.

En el estudio se considerarán consultas de agregación con operadores `SUM`, `COUNT`, `MIN` y `MAX`.

3.4. Estado del Arte

El concepto de reparación de bases de datos relacionales con respecto a conjuntos de restricciones de integridad, así como también el estudio de respuestas consistentes a consultas ha sido ampliamente estudiado en (**????????**). Se han propuesto métodos para computar respuestas consistentes a consultas sin tener que computar las reparaciones minimales (**????**). Sin embargo, estos métodos solo se pueden aplicar para ciertas clases de consultas (por ejemplo, consultas sin proyección) y ciertas clases de restricciones de integridad (ejemplo, dependencias funcionales).

Durante las ultimas décadas, y el avance en el manejo de las bases de datos, los data warehouse han tenido relevancia, por su gran capacidad de generar resúmenes de información que se encuentra almacenada en el tiempo. Las primeras investigaciones de consistencia en DWs consideraban que las dimensiones eran la parte estática de los DWs y solo los hechos constituían la parte dinámica, en el sentido que eran afectados por las actualizaciones. Sin embargo, en (**??**) se ha mostrado que las dimensiones deben ser adaptadas a los cambios del entorno. Cuando se realiza una actualización en un DW es posible que se generen inconsistencias con respecto a sus restricciones de integridad, lo cual puede ser provocado por errores en los datos de origen, disparidad entre la distintas fuentes, entre otros. Por otro lado, en (**?**) se presentan modelos de datos multidimensional que soportan ambigüedad, imprecisión e incertidumbre en sus datos.

(**?**) muestra que es importante utilizar las restricciones de integridad para guiar las operaciones de actualización, y de esta forma no realizar cambios que puedan producir inconsistencias.

Uno de los primeros en proponer una forma de transformar una dimensión no estricta en una estricta fue (**?**), el cual propone la inserción de nuevos elementos artificiales en algunas categorías. Este método es útil cuando los datos son correctos, pero no se acomodan a la restricción estricta, *por ejemplo, un río que está en el límite de dos países, podría estar asociado a dos elementos en la misma categoría país, en este caso la solución no es reparar en el sentido propuesto en (**?**), sino transformar los datos para cumplir con la restricción de integridad. En el ejemplo, la solución sería agregar un elemento compuesto con el nombre de los dos países a los que el río está asociado en la categoría país.* Esta idea se toma en consideración en la solución propuesta en esta tesis. Otro método el el propuesto en (**?**) donde proponen solucionar la inconsistencia modificando el esquema jerárquico, este tipo de solución es una técnica que el resto de autores intenta evitar, manteniendo en lo posible el esquema de jerarquía sin cambios.

A pesar de que existan varias formas de representar un DW usando modelos relacionales, se ha mostrado que no es posible usar técnicas de reparación de bases de datos relacionales para calcular las reparaciones de un DW (**?**). El principal motivo de esto es la semántica de reparación que se utiliza. En los DWs es más útil una semántica de reparación basada en cardinalidad (**??**), donde hay que reducir al mínimo el número de cambios realizados en una dimensión, mientras que el modelo relacional, además de basarse en cardinalidad, se puede reparar reduciendo al mínimo el conjunto de inserciones y eliminaciones de tuplas.

En (**?**) se presenta una semántica de reparación basada en cardinalidad de cambios, donde se reparan las inconsistencias respecto de RI estrictas y homogéneas mediante la inserción y eliminación de arcos. Se muestra que el número de reparaciones para una dimensión puede ser exponencial, el problema de computar las reparaciones minimales de una dimensión en general, es `NP`[1]`-completo`. Por lo tanto, se requieren algoritmos heurísticos para poder encontrar las reparaciones minimales de dimensiones inconsistentes. En este artículo además se presenta un método basado en programas en lógica con semántica de modelos estables (**?**) para representar y obtener las reparaciones minimales de una dimensión. Estos programas, también podrían ser utilizados para computar respuestas consistentes a consultas de agregación (**?**). Sin embargo, a pesar de que el enfoque lógico es efectivo encontrando las reparaciones minimales, su desempeño puede ser ineficiente.

En (**?**) se propone la generación de una dimensión canónica, que busca obtener respuestas consistentes en DWs inconsistentes con respecto a las restricciones de integridad estrictas. Para este proceso, es necesario computar todas las reparaciones minimales de la dimensión inconsistente para luego crear una nueva dimensión. Esta dimensión canónica, puede ser utilizada para obtener respuestas aproximadas a consultas. Esta tesis propone un método alternativo para obtener una reparación de una dimensión inconsistente que obtenga respuestas aproximadas a consultas de agregación sin tener que computar las reparaciones minimales.

En la mayoría de las investigaciones y aplicaciones industriales OLAP, las dimensiones se consideran para satisfacer todas las restricciones de integridad estrictas y homogéneas definidas en el esquema jerárquico (**?**). Sin embargo las dimensiones podrían no satisfacer estas dimensiones (**??????**). Cuando esto ocurre es necesario especificar las restricciones de integridad para identificar los rollup que son estrictos u homogéneos, y de esta manera poder mantener la capacidad de responder correctamente a consultas de agregación (**?**). En los principales sistemas de gestión de bases de datos (SGBD) existentes en el mercado, las dimensiones se modelan en base de datos relacionales, principalmente optando por el esquema estrella o copo de nieve, el principal problema que posee este tipo de técnicas es que es muy difícil mantener y evaluar la consistencia de la dimensión, puesto que requiere de muchas restricciones en las tablas. Además, es muy probable que solucionar una inconsistencia requiera realizar un cambio que elimine muchos más arcos de la dimensión que hacerlo directamente en ella, puesto que por ejemplo en el esquema estrella, requiere eliminar una tupla de la dimensión.

[1] non-deterministic polynomial time

3.5. Metodología

Las siguientes son las actividades a realizar en esta investigación, en función de los objetivos específicos planteados anteriormente.

- Analizar la literatura relevante al tema del tratamiento de inconsistencias en DWs y cómputo de respuestas desde DWs inconsistentes con respecto a RI estrictas y homogéneas.
- Definir una nueva dimensión que permita cumplir los objetivos.
- Implementar algoritmos para obtener una dimensión estricta y homogénea desde una dimensión inconsistente, sin el cómputo de las reparaciones minimales.
- Finalmente, realizar experimentos que demuestren la efectividad de los algoritmos planteados.

Capítulo 4

Dimensión Extendida

Cuando una dimensión es inconsistente con respecto a sus restricciones de integridad, de alguna forma hay elementos que presentan ambigüedad con respecto a sus rollups. Por ejemplo, una violación de una restricción estricta $\mathsf{c}_i \rightarrow \mathsf{c}_j$ es producida cuando hay un elemento en la categoría c_i que alcanza más de un elemento en la categoría c_j, por lo tanto hay un error o imprecisión en los datos. La incertidumbre se ha analizado en el contexto de los modelos multidimensionales en (**??**).

También, las definiciones presentadas en el Capítulo 2 no son lo suficiente amplias para poder generar la dimensión extendida que se plantea en este capítulo, por este motivo es necesario modificar algunas de las definiciones ya propuestas para poder generar una nueva dimensión que llamaremos *extendida*. En esta dimensión se repararán las inconsistencias de datos mediante la inserción de nuevos elementos en las categorías, estos elementos son del tipo conjunto, por lo tanto, es necesario formalizar el tipo de dimensiones que soportarán elementos conjuntos en sus categorías, a diferencia de la definición original de dimensión, la cual no está definida para soportar conjuntos de elementos. Con esta dimensión extendida será posible centrarnos en el tema de obtener una reparación compatible. Esta nueva reparación compatible es una dimensión extendida que permitirá obtener respuestas aproximadas a consultas de agregación.

En la Sección 4.1 se adaptan algunas de las definiciones presentadas en la Sección 2.1 para poder utilizarlas en la definición de *dimensión extendida* que formalizamos en la Sección 4.2, la cual considera que las categorías pueden contener conjuntos de elementos. Por último, en la Sección 5 se define el concepto de *reparación compatible* que se obtiene a partir de la dimensión original inconsistente sin necesidad de computar todas las reparaciones minimales.

4.1. Definiciones para la Dimensión Extendida

Las definiciones presentadas en esta sección son adaptaciones de las definiciones presentadas en las secciones 2.1 y 2.2. Sin embargo, en ellas el esquema de jerarquía, se mantiene sin cambios, por lo que no es necesario redefinirla o modificarla, por lo cual cada vez que se mencione un esquema de jerarquía $\mathcal{H}$, éste se refiere a la definición 2.1.

Definición 4.1 (Dimensión) Una *dimensión* $\mathcal{D}$ es una tupla $(\mathcal{H}_\mathcal{D}, \mathcal{E}_\mathcal{D}, \mathsf{Elem}_\mathcal{D}, <_\mathcal{D})$, donde $\mathcal{H}_\mathcal{D} = (\mathcal{C}_{\mathcal{H}_\mathcal{D}}, \nearrow_{\mathcal{H}_\mathcal{D}})$ es un esquema de jerarquía, $\mathcal{E}_\mathcal{D}$ contiene los elementos; $\mathsf{Elem}_\mathcal{D} : \mathcal{C}_{\mathcal{H}_\mathcal{D}} \to \mathcal{P}(\mathcal{E}_\mathcal{D})$[I] que devuelve el conjunto de elementos en cada categoría, y la relación $<_\mathcal{D} \subseteq \mathcal{E}_\mathcal{D} \times \mathcal{E}_\mathcal{D}$ son las relaciones hijo/padre entre los elementos de diferentes categorías. $<^*_\mathcal{D}$ denota la clausura reflexiva y transitiva de $<_\mathcal{D}$. Las siguientes condiciones se deben satisfacer:

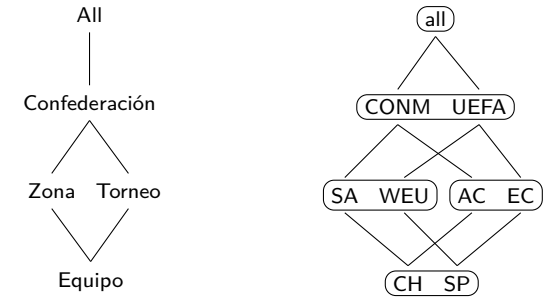
(a) all es el único elemento de la categoría All.

(b) Para todo $\mathsf{c}_i, \mathsf{c}_j \in \mathcal{C}_{\mathcal{H}_\mathcal{D}}$, si $\mathsf{c}_i \neq \mathsf{c}_j$ entonces $\mathsf{Elem}_\mathcal{D}(\mathsf{c}_i) \cap \mathsf{Elem}_\mathcal{D}(\mathsf{c}_j) = \emptyset$.

(c) Por cada par de elementos $a, b \in \mathcal{E}_\mathcal{D}$ tal que $a <_\mathcal{D} b$, entonces existen $\mathsf{c}_i, \mathsf{c}_j \in \mathcal{C}_{\mathcal{H}_\mathcal{D}}$ tal que $\mathsf{c}_i \nearrow_\mathcal{H} \mathsf{c}_j$, $a \in \mathsf{Elem}_\mathcal{D}(\mathsf{c}_i)$ y $b \in \mathsf{Elem}_\mathcal{D}(\mathsf{c}_j)$. □

Note que esta nueva definición difiere de la presentada en la definición 2.2 cuya forma es $\mathcal{D} = (\mathcal{H}_\mathcal{D}, \mathcal{E}_\mathcal{D}, \mathsf{Cat}_\mathcal{D}, <_\mathcal{D})$, en la cual $\mathsf{Cat}_\mathcal{D}$ es una función que entrega la categoría de los elementos. Ahora, en la nueva definición se utiliza la función $\mathsf{Elem}_\mathcal{D}$ en reemplazo de $\mathsf{Cat}_\mathcal{D}$ y ésta es una función que para una categoría dada, devuelve el conjunto de elementos que pertenecen a ella.

De ahora en adelante cuando se mencione una dimensión $\mathcal{D}$, esta será una dimensión como la presentada en la Definición 4.1



(a) Esquema de Jerarquía

(b) Dimensión $\mathcal{D}$ Equipos de Fútbol

Figura 4.1: Esquema y Dimensión Equipos de Fútbol

Ejemplo 4.1 *Para la Figura 4.1(b), las tuplas de la dimensión* $\mathcal{D} = (\mathcal{H}_\mathcal{D}, \mathcal{E}_\mathcal{D}, \mathsf{Elem}_\mathcal{D}, <_\mathcal{D})$ *están compuestas por:*

$\mathcal{H}$ *es el esquema de jerarquía (ver Definición 2.1),*

$\mathcal{E}_\mathcal{D} = \{all, CONM, UEFA, SA, WEU, AC, EC, CH, SP\}$,

$\mathsf{All}_\mathcal{D} = all$,

$\mathsf{Elem}_\mathcal{D}(\mathsf{All}) = \{all\}$,

$\mathsf{Elem}_\mathcal{D}(\mathsf{Confederación}) = \{CONM, UEFA\}$,

$\mathsf{Elem}_\mathcal{D}(\mathsf{Zona}) = \{SA, WEU\}$,

$\mathsf{Elem}_\mathcal{D}(\mathsf{Torneo}) = \{AC, EC\}$,

$\mathsf{Elem}_\mathcal{D}(\mathsf{Equipo}) = \{CH, SP\}$,

[I] $\mathcal{P}$ es el conjunto potencia, conjunto formado por todos los subconjuntos posibles de una dimensión

$<_{\mathcal{D}}$ = *{(CH, SA), (SP, WEU), (CH, AC), (SP, EC), (SA, CONM), (WEU, UEFA), (AC, CONM), (EC, UEFA), (CONM, all), (UEFA, all)}*,
$<^*_{\mathcal{D}} = <_{\mathcal{D}} \bigcup$ *{(CH, CONM), (SP, UEFA), (CH,CH), (SP, SP), (CONM, CONM), (UEFA, UEFA), ...}* □

El conjunto de relaciones rollup de una dimensión $\mathcal{D}$ (Definición 4.1) se define de la siguiente manera:

Definición 4.2 (Relaciones rollup) Dada una dimensión $\mathcal{D} = (\mathcal{H}_{\mathcal{D}}, \mathcal{E}_{\mathcal{D}}, \mathsf{Elem}_{\mathcal{D}}, <_{\mathcal{D}})$, la *relación rollup* $\mathcal{R}_{\mathcal{D}}(\mathsf{c}_i, \mathsf{c}_j)$ para cada par de categorías $\mathsf{c}_i, \mathsf{c}_j \in \mathcal{C}_{\mathcal{H}_{\mathcal{D}}}$ tal que $\mathsf{c}_i \nearrow^*_{\mathcal{H}_{\mathcal{D}}} \mathsf{c}_j$, consiste en $\mathcal{R}_{\mathcal{D}}(\mathsf{c}_i, \mathsf{c}_j) = \{(a, b) | a \in \mathsf{Elem}_{\mathcal{D}}(\mathsf{c}_i), b \in \mathsf{Elem}_{\mathcal{D}}(\mathsf{c}_j)$ y $a <^*_{\mathcal{D}} b\}$. □

Sea $\mathcal{D}$ la dimensión de la Figura 4.1. La relación rollup $\mathcal{R}_{\mathcal{D}}(\mathsf{Equipo}, \mathsf{Zona})$, contiene los pares {(CH, SA), (SP, WEU)}.

Como se puede apreciar, las relaciones rollup no son diferentes de las relaciones rollups definidas para las dimensiones de la forma 2.2. Lo mismo sucede con la definición de restricciones estrictas y homogéneas.

Definición 4.3 (Restricciones Estrictas y Homogéneas) Dado un esquema jerárquico $\mathcal{H} = (\mathcal{C}_{\mathcal{H}}, \nearrow_{\mathcal{H}})$ y categorías $\mathsf{c}_i, \mathsf{c}_j \in \mathcal{C}_{\mathcal{H}}$ tal que $\mathsf{c}_i \nearrow^*_{\mathcal{H}} \mathsf{c}_j$ una restricción estricta entre esas categorías se denota por $\mathsf{c}_i \to \mathsf{c}_j$ y una restricción homogénea por $\mathsf{c}_i \Rightarrow \mathsf{c}_j$.

Una dimensión $\mathcal{D} = (\mathcal{H}, \mathcal{E}_{\mathcal{D}}, \mathsf{Elem}_{\mathcal{D}}, <_{\mathcal{D}})$ satisface una *restricción estricta* $\mathsf{c}_i \to \mathsf{c}_j$ sí y solo sí para todo elemento $e_i \in \mathsf{Elem}_{\mathcal{D}}(\mathsf{c}_i)$ no existe elementos $e_j, e_k \in \mathsf{Elem}_{\mathcal{D}}(\mathsf{c}_j)$ tales que $e_j \neq e_k$, $e_i <^* e_j$ y $e_i <^* e_k$. La dimensión $\mathcal{D}$ satisface la *restricción homogénea* $\mathsf{c}_i \Rightarrow \mathsf{c}_j$ sí y solo sí para todo elemento $e_i \in \mathsf{Elem}_{\mathcal{D}}(\mathsf{c}_i)$ existe un elemento $e_j \in \mathsf{Elem}_{\mathcal{D}}(\mathsf{c}_j)$ tal que $e_i <^* e_j$. □

Las características de una restricción estricta y homogénea permanecen sin cambios excepto en la dimensión que se utiliza, por lo que las características presentadas en la definición 2.5 se mantienen sin cambios.

De la misma forma, también es necesario adaptar el concepto de **distancia** y de **reparación** para una reparación, la diferencia con la presentada en la Sección 2.2.1 basta con cambiar el tipo de dimensión que se utiliza, siendo ahora una dimensión del tipo $\mathcal{D} = (\mathcal{H}_{\mathcal{D}}, \mathcal{E}_{\mathcal{D}}, \mathsf{Elem}_{\mathcal{D}}, <_{\mathcal{D}})$, el resto de las características se mantiene sin alteraciones.

4.2. Dimensión Extendida y Respuestas Aproximadas

El siguiente concepto de dimensión permite la generación de dimensiones donde los elementos de las categorías pueden ser conjuntos de elementos.

Definición 4.4 (Dimensión Extendida) Una *dimensión extendida* $\mathcal{X}$ es una tupla $(\mathcal{H}, \mathcal{E}_{\mathcal{X}}, \mathsf{CElem}_{\mathcal{X}}, \ll_{\mathcal{X}})$, donde $\mathcal{H} = (\mathcal{C}_{\mathcal{H}}, \nearrow_{\mathcal{H}})$ es un esquema de jerarquía; $\mathcal{E}_{\mathcal{X}}$ es un conjunto de constantes, llamadas elementos; $\mathsf{CElem}_{\mathcal{X}} : \mathcal{C}_{\mathcal{H}} \to \mathcal{P}(\mathcal{P}(\mathcal{E}_{\mathcal{X}}))$ es una función que dada una categoría devuelve una familia de subconjuntos de $\mathcal{E}_{\mathcal{X}}$ (es decir, subconjuntos de los subconjuntos de $\mathcal{E}_{\mathcal{X}}$);

y la relación $\ll_{\mathcal{X}} \subseteq \mathcal{P}(\mathcal{E}_{\mathcal{X}}) \times \mathcal{P}(\mathcal{E}_{\mathcal{X}})$ representa las relaciones hijo/padre entre los elementos de diferentes categorías. Se denota por $\ll^*_{\mathcal{X}}$ la cláusula reflexiva y transitiva de $\ll_{\mathcal{X}}$. Además se deben cumplir las siguientes restricciones:

(a) all es el único elemento en la categoría All
(b) Para cada categoría $\mathsf{c}_i, \mathsf{c}_j \in \mathcal{C}_{\mathcal{H}}$, si $\mathsf{c}_i \neq \mathsf{c}_j$ se cumple que $\mathsf{CElem}_{\mathcal{X}}(\mathsf{c}_i) \cap \mathsf{CElem}_{\mathcal{X}}(\mathsf{c}_j) = \emptyset$.
(c) Por cada par de elementos $a \in \mathsf{CElem}_{\mathcal{X}}(\mathsf{c}_i)$ y $b \in \mathsf{CElem}_{\mathcal{X}}(\mathsf{c}_j)$ si $a \ll_{\mathcal{X}} b$ entonces $\mathsf{c}_i \nearrow_{\mathcal{H}} \mathsf{c}_j$.
(d) Para cada categoría $\mathsf{c}_i \in \mathcal{C}_{\mathcal{H}}$ se cumple que (i) $\emptyset \notin \mathsf{CElem}_{\mathcal{X}}(\mathsf{c}_i)$; y (ii) si $e \in \mathsf{CElem}_{\mathcal{X}}(\mathsf{c}_i)$ entonces para cada elemento $e' \in e$, se cumple que $\{e'\} \in \mathsf{CElem}_{\mathcal{X}}(\mathsf{c}_i)$.
(e) Para la categoría inferior $\mathsf{c}_b \in \mathcal{C}_{\mathcal{H}}$ se debe cumplir que por cada $e \in \mathsf{CElem}_{\mathcal{X}}(\mathsf{c}_b)$, e es un elemento unitario. □

Definición 4.5 Para una dimensión extendida $\mathcal{X} = (\mathcal{H}_{\mathcal{X}}, \mathcal{E}_{\mathcal{X}}, \mathsf{CElem}_{\mathcal{X}}, \ll_{\mathcal{X}})$, con $\mathcal{H} = (\mathcal{C}_{\mathcal{H}}, \nearrow_{\mathcal{H}})$, su dimensión clásica asociada ($\mathcal{D} = (\mathcal{H}_{\mathcal{D}}, \mathcal{E}_{\mathcal{D}}, \mathsf{Elem}_{\mathcal{D}}, <_{\mathcal{D}})$) se denota como $\mathcal{T}(\mathcal{X})$, donde:

- $\mathcal{H}_{\mathcal{D}} = \mathcal{H}_{\mathcal{X}}$,
- $\mathcal{E}_{\mathcal{D}} = \bigcup_{c_i \in \mathcal{C}_{\mathcal{H}_{\mathcal{X}}}} \mathsf{CElem}_{\mathcal{X}}(c_i)$,
- $\mathsf{Elem}_{\mathcal{D}} = \mathsf{CElem}_{\mathcal{X}}$ con dominio en $\mathcal{E}_{\mathcal{D}}$, y
- $<_{\mathcal{D}} = \ll_{\mathcal{X}}$ son los elementos en $\mathcal{E}_{\mathcal{D}}$. □

$\mathcal{T}(\mathcal{X})$ no es más que la dimensión tradicional que corresponde a la dimensión extendida X. Sin embargo, tenga en cuenta que $\mathcal{T}(\mathcal{X})$ pierde la información sobre la relación entre los elementos. Por ejemplo, se podría interpretar que no hay relación entre los elementos o_1 y $\{o_1, o_2\}$ a pesar de que el elemento o_1 este contenido en el conjunto, esto es por la interpretación de la dimensión extendida como una dimensión clásica.

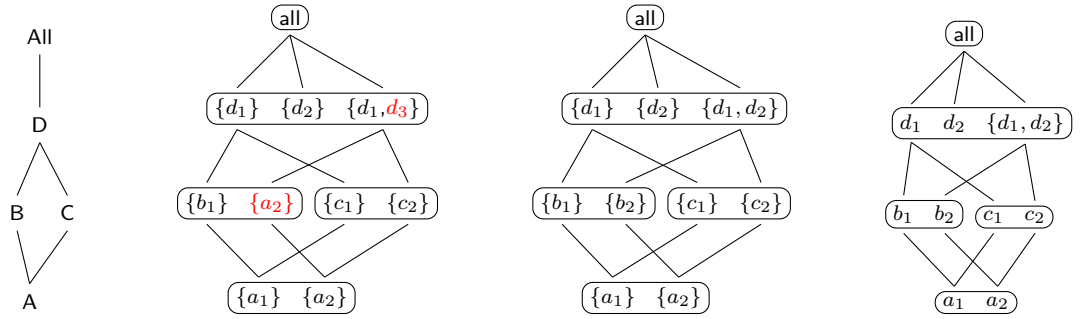
Una dimensión extendida utiliza el esquema jerárquico $\mathcal{H}$ de una dimensión acorde a la Definición 2.1, por lo cual en el resto del documento solo nos referiremos como $\mathcal{H}$ para mencionar un esquema de jerarquía. La Definición 4.4 denota lo siguiente: el elemento all es el único elemento en la categoría All y además, si existen dos elementos que se unen mediante rollup entre categorías, estas categorías están relacionadas en el esquema jerárquico. Cabe destacar que estas dos características, no varían de las presentadas en la Definición 4.1.

La condición (b) hace referencia a que no pueden existir los mismos elementos en categorías distintas, como se muestra en la Figura 4.2(b) en la categoría B esta el elemento a_2 el cual ya existe en la categoría A, por lo tanto, esta dimensión, no cumple con ser una dimensión extendida. La condición (c) se refiere a que cada elementos que pertenezca a la relación hijo/padre ($\ll_{\mathcal{X}}$) solo pueden estar conectados a elementos de categorías que estén conectadas en el esquema jerárquico. La condición (d) hace referencia a que no debe existir el *conjunto vacío* en alguna de las categorías de la dimensión y los conjuntos generados, solo pueden contener elementos que se encuentren en dicha categoría, ya sean dos o más, pero nunca un elemento que no pertenezca a la categoría. En la Figura 4.2(b) se muestra una dimensión que no cumple esta propiedad, puesto que en la categoría D el elemento conjunto es $\{d_1, d_3\}$ y como elementos unitarios solo existen d_1 y d_2. Por último, la condición (e) fuerza que los elementos en la categoría inferior solo sean elementos unitarios y nunca conjuntos (la categoría inferior en la Figura 4.2 es A).

Ejemplo 4.2 *La dimensión extendida $\mathcal{X}$ de la Figura 4.2(c) se formaliza de la siguiente manera:*

$\mathcal{H} = (\mathcal{C}_\mathcal{X}, \nearrow_\mathcal{H})$ *donde:*
$\mathcal{C}_\mathcal{X} = \{A, B, C, D, All\}$,
$\nearrow_\mathcal{H} = \{(A, B), (A, C), (B, D), (C, D), (D, All)\}$,
$\mathcal{E}_\mathcal{X} = \{a_1, a_2, b_1, b_2, c_1, c_2, d_1, d_2, all\}$,
$\mathsf{CElem}_\mathcal{X}(All) = \{\{all\}\}$,
$\mathsf{CElem}_\mathcal{X}(A) = \{\{a_1\}, \{a_2\}\}$,
$\mathsf{CElem}_\mathcal{X}(B) = \{\{b_1\}, \{b_2\}\}$,
$\mathsf{CElem}_\mathcal{X}(C) = \{\{c_1\}, \{c_2\}\}$,
$\mathsf{CElem}_\mathcal{X}(D) = \{\{d_1\}, \{d_2\}, \{d_1, d_2\}\}$,
$\ll_\mathcal{X} = \{(\{a_1\}, \{b_1\}), (\{a_1\}, \{c_1\}), (\{a_2\}, \{b_2\}), (\{a_2\}, \{c_2\}), (\{b_1\}, \{d_1\}), (\{b_2\}, \{d_1, d_2\}),$
$(\{c_1\}, \{d_1\}), (\{c_2\}, \{d_1, d_2\})(\{d_1\}, \{all\}), (\{d_2\}, \{all\})(\{d_1, d_2\}, \{all\})\}$,
$\ll^*_\mathcal{X} = \ll_\mathcal{X} \bigcup \{(a_1, a_1), (a_2, a_2), (a_1, d_1), \ldots,\}$

Notar que la Figura 4.2(c) corresponde a una dimensión extendida ya que cumple todas las propiedades presentadas en la Definición 4.4. □



(a) Esquema de Jerarquía

(b) Dimensión No Extendida

(c) Dimensión Extendida

(d) Dimensión Extendida como dimensión clásica

Figura 4.2: Ejemplo Dimensión Extendida

La *dimensión extendida* $\mathcal{X} = (\mathcal{H}, \mathcal{E}_\mathcal{X}, \mathsf{CElem}_\mathcal{X}, \ll_\mathcal{X})$ de la Figura 4.2(c) también puede ser vista como una dimensión clásica $\mathcal{D} = (\mathcal{H}, \mathcal{E}_\mathcal{D}, \mathsf{Elem}_\mathcal{D}, <_\mathcal{D})$ donde $\mathcal{E}_\mathcal{D} = \mathcal{P}(\mathcal{E}_\mathcal{X}) \setminus \{\}$, $\mathsf{Elem}_\mathcal{D} = \mathsf{CElem}_\mathcal{X}$ y $<_\mathcal{D} = \ll_\mathcal{X}$. Esto es, cuando una dimensión extendida $\mathcal{X}$ que contiene solo conjunto unitarios en sus categorías, se puede ver como una dimensión clásica $\mathcal{D}$, considerando en vez de los conjuntos unitarios sus elementos, esto se puede apreciar en la Figura 4.2(c), la cual, es una dimensión extendida que posee en cada categoría conjuntos unitarios de elementos, que por simplicidad, se representa finalmente como en la Figura 4.2(d), donde solo se toman los elementos, pudiendo esta dimensión extendida, verse tal como una dimensión clásica.

Definición 4.6 Para una dimensión extendida $\mathcal{X} = (\mathcal{H}_\mathcal{X}, \mathcal{E}_\mathcal{X}, \mathsf{CElem}_\mathcal{X}, \ll_\mathcal{X})$, con $\mathcal{H} = (\mathcal{C}_\mathcal{H}, \nearrow_\mathcal{H})$, se relaciona a una dimensión clásica $\mathcal{D} = (\mathcal{H}_\mathcal{D}, \mathcal{E}_\mathcal{D}, \mathsf{Elem}_\mathcal{D}, <_\mathcal{D})$, en:

- $\mathcal{H}_\mathcal{D} = \mathcal{H}_\mathcal{X}$,

- $\mathcal{E}_{\mathcal{D}} = \bigcup_{c_i \in \mathcal{C}_{\mathcal{H}_{\mathcal{X}}}} \mathsf{CElem}_{\mathcal{X}}(c_i)$,
- $\mathsf{Elem}_{\mathcal{D}} = \mathsf{CElem}_{\mathcal{X}}$ con dominio en $\mathcal{E}_{\mathcal{D}}$, y
- $<_{\mathcal{D}} = \ll_{\mathcal{X}}$ son los elementos en $\mathcal{E}_{\mathcal{D}}$. □

Uno de los principales cambios con respecto a la definición original de dimensión es $\mathsf{CElem}_{\mathcal{X}}$, el cual permite ahora establecer que las dimensiones utilizadas poseen en cada categoría conjuntos de elementos y no elementos simples. En este sentido, todas las categorías poseen conjuntos de elementos. Para simplificar la presentación cuando un elemento de una categoría sea un conjunto unitario, éste se representará sin los símbolos "{ }" tal como se aprecia en la Figura 4.2(c) en la categoría D donde los elementos d_1 y d_2 se presentan como elementos simples (unitarios) al igual que en una dimensión clásica $\mathcal{D}$, pero en una dimensión extendida, ahora son conjunto unitarios que serán representados de igual forma (Ver Figura 4.2); por otro lado, el elemento $\{d_1, d_2\}$ es un conjunto que contiene dos elementos de la categoría (por coincidencia son todos los elementos de la categoría), por lo que este sí lleva los símbolos de conjunto, para representarlos.

Por último, la segunda diferencia de una dimensión clásica con una dimensión extendida, es que las relaciones rollup entre elementos de categorías como se hace en las dimensiones clásicas, en las dimensiones extendidas, son entre conjuntos de elementos, por lo que el símbolo que representa esta relación para una dimensión extendida es $\ll_{\mathcal{X}}$ que es análoga a $<_{\mathcal{X}}$.

Definición 4.7 (Restricciones Dimensiones Extendidas) Dado un esquema jerárquico $\mathcal{H} = (\mathcal{C}_{\mathcal{H}}, \nearrow_{\mathcal{H}})$ y categorías c_i, $\mathsf{c}_j \in \mathcal{C}_{\mathcal{H}}$ tal que $\mathsf{c}_i \nearrow^*_{\mathcal{H}} \mathsf{c}_j$ una restricción estricta entre esas categorías se denota por $\mathsf{c}_i \to \mathsf{c}_j$ y una restricción homogénea por $\mathsf{c}_i \Rightarrow \mathsf{c}_j$.

Una dimensión $\mathcal{X} = (\mathcal{H}, \mathcal{E}_{\mathcal{X}}, \mathsf{CElem}_{\mathcal{X}}, \ll_{\mathcal{X}})$ satisface una *restricción estricta* $\mathsf{c}_i \to \mathsf{c}_j$ sí y solo sí para todo elemento $e_i \in \mathsf{CElem}_{\mathcal{X}}(\mathsf{c}_i)$ no existe elementos $e_j, e_k \in \mathsf{CElem}_{\mathcal{X}}(\mathsf{c}_j)$ tales que $e_j \neq e_k$ y, $e_i \ll^* e_j$ y $e_i \ll^* e_k$. La dimensión $\mathcal{X}$ satisface la *restricción homogénea* $\mathsf{c}_i \Rightarrow \mathsf{c}_j$ sí y solo sí para todo elemento $e_i \in \mathsf{CElem}_{\mathcal{X}}(\mathsf{c}_i)$ existe un elemento $e_j \in \mathsf{CElem}_{\mathcal{X}}(\mathsf{c}_j)$ tal que $e_i \ll^* e_j$. □

Las restricciones estrictas y homogéneas para una dimensión extendida funcionan de manera análoga a las definidas previamente para dimensiones clásicas. La diferencia esta en que ahora las restricciones son en base a conjuntos de elementos.

Una dimensión extendida es estricta si todas sus relaciones rollups son estrictas. De otra forma, la dimensión es no estricta. Una dimensión es homogénea si todas sus relaciones rollups son homogéneas.

4.2.1. Respuestas Aproximadas

Dado que para una dimensión extendidas $\mathcal{X}$ tenemos su correspondiente dimensión clásica $\mathcal{T}(\mathcal{X})$ podríamos pensar que al plantear una consulta podríamos usar las técnicas aplicadas actualmente en los sistemas de DWS. Sin embargo, como veremos, las necesidades de respuestas a consultas que se redefinen para las dimensiones extendidas, permiten sacar el máximo provecho de la flexibilidad dada por las dimensiones extendidas.

En los DWs las consultas de agregación más comunes son aquellas que llevan a cabo la agrupación por los valores de un conjunto de atributos, y devuelven un único valor agregado por este

grupo. Una consulta de agregación es de la forma:

```
SELECT A_j, ... A_n, f(A)
FROM T, R_i, ... R_m
WHERE condiciones
GROUP BY A_j, ... A_n
```

Donde A_j,... A_n son los atributos de las relaciones rollup R_i, ... R_m (vistas como tablas), y f es una función, tal como: `MIN(A)`, `MAX(A)`, `COUNT(A)`, `SUM(A)`, aplicado al atributo `A` de la tabla de hechos, con $\mathsf{A} \notin \{\mathsf{A}_j, \ldots \mathsf{A}_n\}$.

Definición 4.8 (Respuestas a consultas) Una respuesta a una consulta de agregación $\mathcal{Q}$ sobre una dimensión $\mathcal{D}_1, \ldots, \mathcal{D}_i, \ldots, \mathcal{D}_n$ y una tabla de hechos F es una tupla de la forma $\langle t_1, \ldots, t_n, c\rangle$, donde cada t_i es un elemento de $\mathcal{D}_i$, y c es el valor retornado por la función de agregación por agrupamiento $\langle t_1, \ldots, t_n\rangle$ en la tabla de hechos F. El conjunto de respuestas de $\mathcal{Q}$ sobre la dimensión $\mathcal{D}_1, \ldots, \mathcal{D}_i, \ldots, \mathcal{D}_n$ y la tabla de hechos F es denotada por $\mathcal{Q}(\{\mathcal{D}_1, \ldots, \mathcal{D}_i, \ldots, \mathcal{D}_n\}, F)$. □

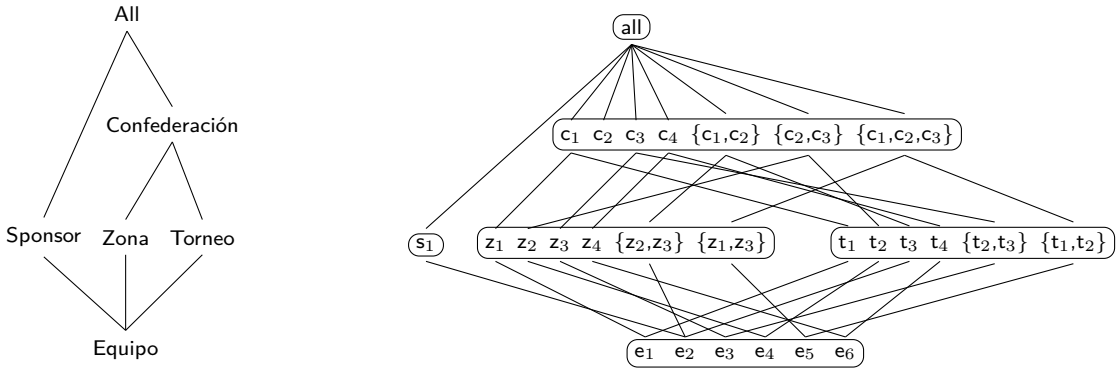
Dado un conjunto de dimensiones extendidas $\mathcal{X}_1, \ldots, \mathcal{X}_i, \ldots, \mathcal{X}_n$, la tabla de hechos F y una consulta $\mathcal{Q}$ podríamos tratar de dar respuestas solo con evaluar la consulta sobre las dimensiones $\mathcal{T}(\mathcal{X}_i)$ con $i \in [1, n]$, esto es $\mathcal{Q}(\{\mathcal{T}(\mathcal{X}_1), \ldots, \mathcal{T}(\mathcal{X}_i), \ldots, \mathcal{T}(\mathcal{X}_n)\}, F)$. En el siguiente ejemplo mostramos que los resultados obtenidos con este método no permiten al usuario ver el efecto de la relación entre los elementos dentro de una categoría.

Ejemplo 4.3 *Considere la Tabla 4.2, esta muestra un conjunto de respuestas que pueden ser obtenida de la tabla de hechos* Ingresos *de la Tabla 4.1 realizando consultas de agregación Q_j obteniendo, respectivamente, la suma (*SUM*) y cantidad (*COUNT*) de ingresos agrupados por* Confederación, y Año *sobre la dimensión $\mathcal{T}(\mathcal{X}_{\text{FT}})$ (ver Figura 6.1), y $\mathcal{T}(\mathcal{X}_{\text{TIME}})$ de la dimensión* Tiempo. *Es fácil ver que los resultados en $Q_j(\{\mathcal{T}(\mathcal{X}_{\text{FT}}), \mathcal{T}(\mathcal{X}_{\text{TIME}})\},$ Ingresos$)$ no toma el hecho de que diferentes elementos pueden pertenecer a la misma agregación. Por ejemplo, la tupla t_6 en la Tabla 4.2 puede contribuir a la respuesta en la tupla t_2 ya que c_1 esta contenido en el elemento $\{c_1, c_2\}$.* □

Dado que $\mathcal{Q}(\{\mathcal{T}(\mathcal{X}_1), \ldots, \mathcal{T}(\mathcal{X}_i), \ldots, \mathcal{T}(\mathcal{X}_n)\}, F)$ no toma la semántica deseada a las respuestas a consultas en una dimensión extendida, se ha formalizado una respuesta para estas dimensiones extendidas, que toman en cuenta la relación entre los elementos dentro de una categoría.

Definición 4.9 (Tuplas asociadas a un elemento conjunto) *Dada una consulta $\mathcal{Q}$, una dimensión extendida $\mathcal{X} = (\mathcal{H}, \mathcal{E}_{\mathcal{X}}, \mathsf{CElem}_{\mathcal{X}}, \ll_{\mathcal{X}})$, una tabla de hechos F, y una tupla $t = \langle e_1, \ldots, e_n\rangle$ con $e_i \in \mathcal{E}_{\mathcal{X}}$ por cada $i \in [1, n]$, el conjunto de tuplas asociadas a t en $\mathcal{Q}(\{\mathcal{T}(\mathcal{X}_1), \ldots, \mathcal{T}(\mathcal{X}_i), \ldots, \mathcal{T}(\mathcal{X}_n)\}, F)$, denotado por $\mathcal{A}(t, \mathcal{Q}, \{\mathcal{X}_1, \ldots, \mathcal{X}_i, \ldots, \mathcal{X}_n\}, F)$, es $\{a | a = \langle s_1, \ldots, s_n, v\rangle, a \in Q(\mathcal{X}_1, \ldots, \mathcal{X}_i, \ldots, \mathcal{X}_n, F), e_i \in s_i$ por cada $i \in [1, n]\}$.* □

Intuitivamente, el conjunto $\mathcal{A}(t, \mathcal{Q}, \{\mathcal{X}_1, \ldots, \mathcal{X}_i, \ldots, \mathcal{X}_n\}, F)$ contiene todas las tuplas que contribuyen a la respuesta de la tupla t.



(a) Esquema Equipo de Futbol (b) Dimensión Extendida Equipo de Futbol $\mathcal{X}_{\mathrm{FT}}$

Figura 4.3: Esquema de Jerarquía y Dimensión Extendida $\mathcal{X}_{\mathrm{FT}}$

Definición 4.10 (Respuestas aproximadas para `SUM` y `COUNT`) Dada una consulta $\mathcal{Q}$ con funciones de agregación `SUM` o `COUNT` las cuales denotaremos como $\mathcal{Q}_{\texttt{SUM}}$ y $\mathcal{Q}_{\texttt{COUNT}}$ respectivamente, un set de dimensiones extendidas $\mathcal{S} = \{\mathcal{X}_1, \ldots, \mathcal{X}_i, \ldots, \mathcal{X}_n\}$ donde $\mathcal{X}_i = (\mathcal{H}_{\mathcal{X}_i}, \mathcal{E}_{\mathcal{X}_i}, \mathsf{CElem}_{\mathcal{X}_i}, \ll_{\mathcal{X}_i})$, y una tabla de hechos F, la *respuesta de $\mathcal{Q}$ sobre $\mathcal{X}$ y F* se denota por $\widetilde{Q}(\mathcal{S}, F)$. Una tupla $\langle e_1, \ldots, e_n, [a, b]\rangle$ con $e_i \in \mathcal{E}_{\mathcal{X}}$ por cada $i \in [1, n]$ y $[a; b]$ un conjunto numérico que pertenece a $\widetilde{Q}(\mathcal{S}, F)$ sí y solo sí: (i) $\mathcal{A}(\langle e_1, \ldots, e_n\rangle, \mathcal{Q}, \mathcal{S}, F) \neq \emptyset$; (ii) una de las siguientes condiciones se cumple: $a = v$ cuando existe algún v tal que $\langle\{e_1\}, \ldots, \{e_n\}, v\rangle \in Q(\{\mathcal{X}_1, \ldots, \mathcal{X}_i, \ldots, \mathcal{X}_n\}, F)$ o $a = 0$ cuando no existe tal v; y $b = \sum_{\langle s_1, \ldots, s_n, v\rangle \in V}, v$ cuando $V = \mathcal{A}(\langle e_1, \ldots, e_n\rangle, \mathcal{Q}, \mathcal{S}, F)$. □

La respuesta aproximada para los operadores de agregación son un conjunto de elementos que esta compuesto el valor que sea verdad para una tupla t y por todos los elementos que sean mayores que el valor seguro para t para el operador de agregación (max) que estén asignados a elementos conjuntos y los valores que sean menores que el valor seguro para t para el operador de agregación (min) que estén asignados a elementos conjuntos.

Si tenemos en cuenta, en vez de los operadores de agregación `SUM` y `COUNT` diferentes operadores, la respuesta aproximada debería ser modificada. Por ejemplo, en el caso de `MIN` y `MAX`, en lugar de un rango, estaríamos interesados en el conjunto de posibles valores (mínimo/máximo) que correspondan con las posibles respuestas.

Ejemplo 4.4 *La Tabla 4.3 muestra las respuestas a consultas de agregación con los operadores de agregación `SUM`, `COUNT`, `MIN` y `MAX` sobre las dimensiones extendidas $\mathcal{X}_{\mathrm{FT}}$ y $\mathcal{X}_{\mathrm{TIME}}$ de la Figura 6.1, que se obtienen teniendo en cuenta las respuestas dadas en la Tabla 4.2.*

Como ilustración, para la tupla $t = \langle c_1, 2010\rangle$ dada por $\mathcal{A}(t, \mathcal{Q}_{\texttt{SUM}}, \{\mathcal{X}_{\mathrm{FT}}, \mathcal{X}_{\mathrm{TIME}}\}, \mathsf{Ingresos})$ contiene las tuplas t_1, t_5, t_9 y t_{10} (ver Tabla 4.2), en este caso, no hay respuesta para t en la Tabla 4.2, por lo tanto, la respuesta final para la consulta con el operador de agregación `SUM` es $\langle c_1, 2010, [0, 1700]\rangle$, donde 1700 es la suma de los valores obtenidos de las tuplas en el conjunto $\mathcal{A}(t, \mathcal{Q}_{\texttt{SUM}}, \{\mathcal{X}_{\mathrm{FT}}, \mathcal{X}_{\mathrm{TIME}}\}, \mathsf{Ingresos})$. Para la tupla $t = \langle c_1, 2011\rangle$ dada por $\mathcal{A}(t, \mathcal{Q}_{\texttt{SUM}}, \{\mathcal{X}_{\mathrm{FT}}, \mathcal{X}_{\mathrm{TIME}}\}, \mathsf{Ingresos})$ contiene las tuplas t_1, t_2, t_6 y t_{10}, ya que hay una respuesta para t en la Tabla

Ingresos		
Equipo	**Fecha**	**Ingreso**
e_1	13-3-{2010,2011}	400
e_1	22-3-2011	400
e_1	18-9-2012	200
e_1	11-10-{2012,2013}	300
e_2	2-3-2010	600
e_2	23-10-2011	100
e_2	17-5-{2012,2013}	200
e_2	10-2-2013	100
e_5	26-1-2010	400
e_5	6-4-{2010,2011}	300
e_5	22-7-{2012,2013}	500
e_5	18-11-2013	600
e_4	18-8-2010	100
e_4	21-1-2011	200
e_4	18-8-{2012,2013}	300
e_4	11-11-2013	500
e_3	2-5-{2010,2012}	300
e_3	7-3-2011	300
e_3	11-5-2012	500
e_3	17-9-2013	200
e_6	3-7-2013	300

Tabla 4.1: Tabla de Hechos Ingresos

4.2 (tupla t_2), la respuesta final para el operador `SUM` *es* $\langle c_1, 2011, [400, 1200]\rangle$ *donde el valor* 400 *es la respuesta para* t *dada por la tupla* t_2 *y el valor* 1200 *es la suma de los valores de las tuplas en* $\mathcal{A}(t, \mathcal{Q}_{\texttt{SUM}}, \{\mathcal{X}_{\text{FT}}, \mathcal{X}_{\text{TIME}}\}, \textit{Ingresos})$*. El mismo proceso se realiza para* $(c_1, 2012)$*,* $(c_1, 2013)$*,* $(c_2, 2010)$*,* $(c_2, 2011)$*,* $(c_2, 2012)$*,* $(c_2, 2013)$*,* $(c_3, 2010)$*,* $(c_3, 2011)$*,* $(c_3, 2012)$*,* $(c_3, 2013)$*. Para la tupla* $t = \langle c_4, 2010\rangle$ *dada por* $\mathcal{A}(t, \mathcal{Q}_{\texttt{SUM}}, \{\mathcal{X}_{\text{FT}}, \mathcal{X}_{\text{TIME}}\}, \textit{Ingresos})$ *es vacía, dado que no hay respuesta para* t *en la Tabla 4.2, entonces, no es una respuesta para este grupo (y lo mismo sucede para* c_4 *con el año* 2011 *y* 2012*). Sin embargo, para* $t = \langle c_4, 2013\rangle$ *dado por* $\mathcal{A}(t, \mathcal{Q}_{\texttt{SUM}}, \{\mathcal{X}_{\text{FT}}, \mathcal{X}_{\text{TIME}}\}, \textit{Ingresos})$ *se obtiene que la única tupla es* t_{21} *en la Tabla 4.2, entonces, la respuesta para* `SUM` *es* $\langle c_4, 2013, [300, 300]\rangle$ *donde el valor* 300 *es una respuesta para* $t = \langle c_4, 2013\rangle$ *en la Tabla 4.2. El mismo análisis se realiza para el operador de agregación* `COUNT`*.* □

Ejemplo 4.5 *Para calcular la respuesta a los operadores* `MIN` *y* `MAX` *se realiza el siguiente procedimiento, la tupla* $t = \langle c_1, 2010\rangle$ *dada por* $\mathcal{A}(t, \mathcal{Q}_{\texttt{MIN}}, \{\mathcal{X}_{\text{FT}}, \mathcal{X}_{\text{TIME}}\}, \textit{Ingresos})$ *contiene las tuplas* t_1, t_5, t_9 *y* t_{10} *(ver Tabla 4.2), en este caso, no hay respuesta para* t *en la Tabla 4.2, por lo tanto, la respuesta final para la consulta con el operador de agregación* `MIN` *es* $\langle c_1, 2010, \{0, 300, 400, 600\}\rangle$*,*

	Confederación	**Año**	SUM	MIN	MAX	COUNT
t_1	c_1	{2010,2011}	400	400	400	1
t_2	c_1	2011	400	400	400	1
t_3	c_1	2012	200	200	200	1
t_4	c_1	{2012,2013}	300	300	300	1
t_5	$\{c_1, c_2\}$	2010	600	600	600	1
t_6	$\{c_1, c_2\}$	2011	100	100	100	1
t_7	$\{c_1, c_2\}$	{2012,2013}	200	200	200	1
t_8	$\{c_1, c_2\}$	2013	100	100	100	1
t_9	$\{c_1, c_2, c_3\}$	2010	400	400	400	1
t_{10}	$\{c_1, c_2, c_3\}$	{2010,2011}	300	300	300	1
t_{11}	$\{c_1, c_2, c_3\}$	{2012,2013}	500	500	500	1
t_{12}	$\{c_1, c_2, c_3\}$	2013	600	600	600	1
t_{13}	$\{c_2, c_3\}$	2010	100	100	100	1
t_{14}	$\{c_2, c_3\}$	2011	200	200	200	1
t_{15}	$\{c_2, c_3\}$	{2012,2013}	300	300	300	1
t_{16}	$\{c_2, c_3\}$	2013	500	500	500	1
t_{17}	c_3	{2010,2012}	300	300	300	1
t_{18}	c_3	2011	300	300	300	1
t_{19}	c_3	2012	500	500	500	1
t_{20}	c_3	2013	200	200	200	1
t_{21}	c_4	2013	300	300	300	1

Tabla 4.2: Respuestas a la consulta sobre la dimensión $\mathcal{X}_{\text{FT}}$ y $\mathcal{X}_{\text{TIME}}$ y la tabla de hechos Ingresos

Confederación	**Año**	SUM	COUNT	MIN	MAX
c_1	2010	$[0, 1700]$	$[0, 4]$	$\{0, 300, 400, 600\}$	
c_1	2011	$[400, 1200]$	$[1, 4]$	$\{400, 100, 300\}$	$\{400\}$
c_1	2012	$[200, 1200]$	$[1, 4]$	$\{200\}$	$\{200, 300, 500\}$
c_1	2013	$[0, 1700]$	$[0, 5]$	$\{0, 100, 200, 300, 500, 600\}$	
c_2	2010	$[0, 1400]$	$[0, 4]$	$\{0, 100, 300, 400, 600\}$	
c_2	2011	$[0, 600]$	$[0, 3]$	$\{0, 100, 200, 300\}$	
c_2	2012	$[0, 1000]$	$[0, 3]$	$\{0, 200, 300, 500\}$	
c_2	2013	$[0, 2200]$	$[0, 6]$	$\{0, 100, 200, 300, 500, 600\}$	
c_3	2010	$[0, 1100]$	$[0, 4]$	$\{0, 100, 300, 400\}$	
c_3	2011	$[300, 800]$	$[1, 3]$	$\{300, 200\}$	$\{300\}$
c_3	2012	$[500, 1600]$	$[1, 4]$	$\{500, 300\}$	$\{500\}$
c_3	2013	$[200, 2100]$	$[1, 5]$	$\{200\}$	$\{200, 300, 500, 600\}$
c_4	2013	$[300, 300]$	$[1, 1]$	$\{300\}$	

Tabla 4.3: Respuestas a las consultas de agregación sobre $\mathcal{X}_{\text{FT}}$ y $\mathcal{X}_{\text{TIME}}$

donde cada elemento corresponde a un valor obtenido de las tuplas en el conjunto $\mathcal{A}(t, \mathcal{Q}_{\texttt{MIN}}, \{\mathcal{X}_{\text{FT}}, \mathcal{X}_{\text{TIME}}\}, \textsf{Ingresos})$*. Para la tupla* $t = \langle c_1, 2011\rangle$ *dada por* $\mathcal{A}(t, \mathcal{Q}_{\texttt{MIN}}, \{\mathcal{X}_{\text{FT}}, \mathcal{X}_{\text{TIME}}\}, \textsf{Ingresos})$ *contiene las tuplas* t_1, t_2, t_6 *y* t_{10}*, ya que hay una respuesta para* t *en la Tabla 4.2 (tupla* t_2*), la respuesta final para el operador* `MIN` *es* $\langle c_1, 2011, \{400, 100, 300\}\rangle$ *donde el valor* 400 *es la respuesta para* t *dada por la tupla* t_2 *y los valores* 100 *y* 300 *son de los valores de las tuplas en* $\mathcal{A}(t, \mathcal{Q}_{\texttt{MIN}}, \{\mathcal{X}_{\text{FT}}, \mathcal{X}_{\text{TIME}}\}, \textsf{Ingresos})$*. El mismo proceso se realiza para* $(c_1, 2012)$*,* $(c_1, 2013)$*,* $(c_2, 2010)$*,* $(c_2, 2011)$*,* $(c_2, 2012)$*,* $(c_2, 2013)$*,* $(c_3, 2010)$*,* $(c_3, 2011)$*,* $(c_3, 2012)$*,* $(c_3, 2013)$*. Para la tupla* $t = \langle c_4, 2010\rangle$ *dada por* $\mathcal{A}(t, \mathcal{Q}_{\texttt{MIN}}, \{\mathcal{X}_{\text{FT}}, \mathcal{X}_{\text{TIME}}\}, \textsf{Ingresos})$ *es vacía, dado que no hay respuesta para* t *en la Tabla 4.2, entonces, no es una respuesta para este grupo (y lo mismo sucede para* c_4 *con el año* 2011 *y* 2012*). Sin embargo, para* $t = \langle c_4, 2013\rangle$ *dado por* $\mathcal{A}(t, \mathcal{Q}_{\texttt{MIN}}, \{\mathcal{X}_{\text{FT}}, \mathcal{X}_{\text{TIME}}\}, \textsf{Ingresos})$ *se obtiene que la única tupla es* t_{21} *en la Tabla 4.2, entonces, la respuesta para MIN es* $\langle c_4, 2013, \{300\}\rangle$ *donde el valor* 300 *es una respuesta para* $t = \langle c_4, 2013\rangle$ *en la Tabla 4.2. El mismo análisis se realiza para el operador de agregación* `MAX`*, el cual coincidentemente obtiene los mismos valores que el operador* `MIN`*.* □

Como se puede apreciar, gracias a esta nueva dimensión podemos obtener respuestas a consultas donde puedan existir elementos conjunto, aplicando los distintos operadores de agregación, no se ha definido respuesta aproximada para el operador `avg`, puesto que no es posible computar el promedio de una vista pre-computada ya que no se conoce el n original de elementos con lo cual obtener el promedio.

A continuación se presenta un algoritmo para reparar una dimensión clásica $\mathcal{D}$ inconsistente respecto a un conjuntos de restricciones de integridad, la cual aplica esta nueva dimensión.

Capítulo 5

Reparación compatible

En este capítulo se define la *reparación compatible*, las cual es una nueva dimensión extendida obtenida de la reparación de una dimensión inconsistente respecto a un conjunto Σ de restricciones de integridad. La idea de la reparación compatible, es obtener una única reparación para computar respuestas aproximadas a consultas de agregación. La definición de la reparación compatible esta inspirado en el método presentado en (**?**) para obtener dimensiones estrictas. De acuerdo a este método, las restauración de la restricción estricta se soluciona insertando elementos fusionados en categorías artificiales. Como ejemplo, si un elemento a hace rollup a los elementos b y c en una categoría D, el elemento $\{b, c\}$ se crea en una nueva categoría y a es asociada con este nuevo elemento compuesto. Además, los enlaces se crean a partir de los elementos fusionados de los padres originales de a en la categoría conflictiva D. El método de la reparación compatible adopta el hecho de la generación de un elemento conjunto, pero solo se creará este elemento en las categorías conflictivas solo cunado no exista otra opción para reparar la inconsistencia estricta. Para formalizar este concepto, se utiliza la *dimensión extendida* definida en el Capítulo 4, la cual es una dimensión que puede contener elementos conjuntos en sus categorías.

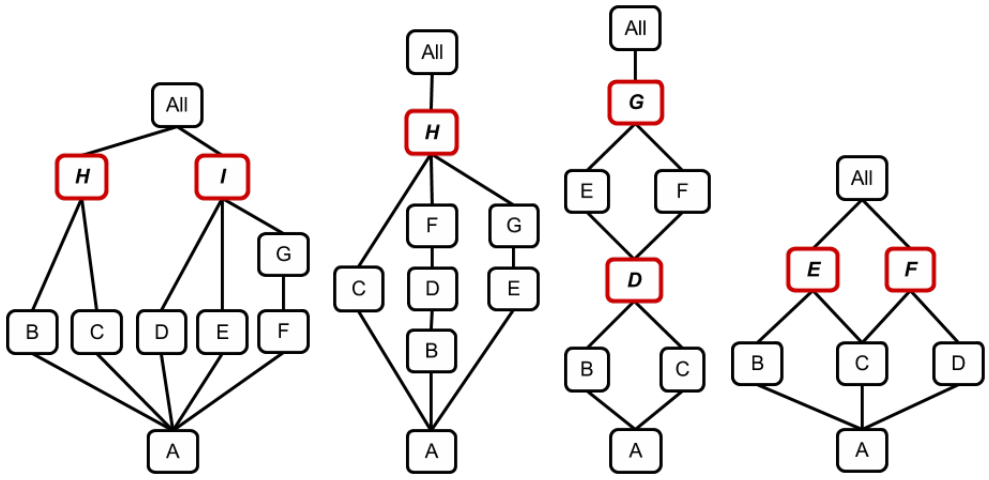
5.1. Definiciones y Conceptos

Antes de definir reparación compatible es necesario definir nivel de convergencia, que intuitivamente, es una categoría superior, a la cual convergen dos o más categorías inferiores (distintas) mediante relaciones rollup.

Definición 5.1 (Nivel de Convergencia) Sea una dimensión $\mathcal{D} = (\mathcal{H}, \mathcal{E}_{\mathcal{D}}, \mathsf{Elem}_{\mathcal{D}}, <_{\mathcal{D}})$, una serie de categorías c_i, c_j y c_k tal que $\mathsf{c}_i \nearrow \mathsf{c}_k$ y $\mathsf{c}_j \nearrow \mathsf{c}_k$, los elementos a, b y c con $\mathsf{Elem}(\mathsf{c}_i) = a$, $\mathsf{Elem}(\mathsf{c}_j) = b$ y $\mathsf{Elem}(\mathsf{c}_k) = c$. La categoría c_k es un nivel de convergencia. □

Una dimensión puede tener varios niveles de convergencia. En la Figura 5.1 se presentan distintos esquemas de jerarquía de dimensiones de DW. Para la Figura 5.1(a) los niveles de convergencia son H e I, en la Figura 5.1(b) el nivel de convergencia es H, para la Figura 5.1(c) los niveles de convergencia son D y G, y por último, en la Figura 5.1(d) los niveles de convergencia son E y F.

Los niveles de convergencia pueden ser *independientes* o no, esto en el sentido de que cambios en las relaciones rollup de categorías inferiores puedan afectar la consistencia de restricciones estrictas que involucren a los niveles de convergencia. Por ejemplo, en la Figura 5.1(a) los niveles de convergencia H e I son independientes, ya que cualquier cambio de arcos realizado entre los elementos de las categorías bajo H no afectarían la consistencia de una restricción estricta entre las categorías A e I, lo mismo ocurre con la Figura 5.1(b) donde la categoría H es una categoría nivel de convergencia independiente. En cambio, en el esquema jerárquico de la Figura 5.1(c) los niveles de convergencia D y G no son independientes, puesto que cambios de arcos entre las categorías bajo D pueden afectar la consistencia en términos de restricciones estrictas entre las categorías D y G. Lo mismo sucede con el esquema jerárquico de la Figura 5.1(d).



(a) (b) (c) (d)

Figura 5.1: Esquemas de Jerarquía y sus niveles de convergencia

Cuando una dimensión con niveles de convergencia es actualizada entonces, el nivel de convergencia pasa a llamarse *nivel de conflicto*, ya que es necesario revisar que la consistencia respecto a restricciones estrictas se mantenga luego de la actualización. La siguiente definición de nivel de conflicto se basa en la definición entregada en (**?**), la cual se aplica a dimensiones clásicas (sin conjuntos de elementos en las categorías) como las de la Definición 2.2.

Tal como se planteo en el Capítulo 4, las dimensiones que trataremos ahora son dimensiones acordes a la Definición 4.1.

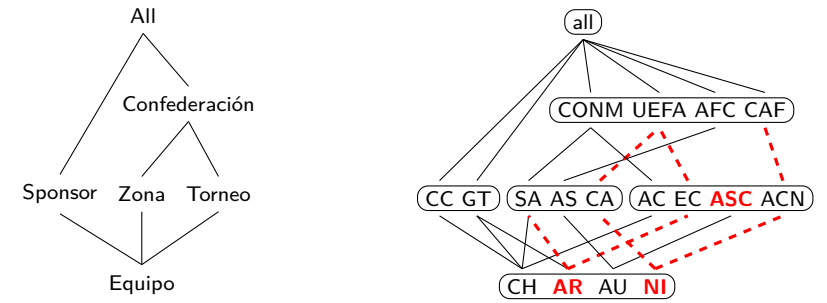
Definición 5.2 (Nivel de Conflicto) Sea una dimensión $\mathcal{D} = (\mathcal{H}, \mathcal{E}_\mathcal{D}, \mathsf{Elem}_\mathcal{D}, <_\mathcal{D})$, un par de categorías c_i y c_j tal que c_i se relaciona con c_j, un par de elementos a y b con $\mathsf{Elem}(\mathsf{c}_i) = a$ y $\mathsf{Elem}(\mathsf{c}_j) = b$. Una categoría c_k, tal que c_j rollup a c_k, es *un nivel de conflicto* respecto a un cambio de arcos entre estas categorías, si hay una categoría c_m tal que c_m rollup a c_i, habiendo una ruta alternativa entre c_m y c_k no incluyendo el arco (c_i,c_j), y a alcanza un elemento en c_m. □

Entonces, cualquier nivel de convergencia es un potencial nivel de conflicto.

El siguiente ejemplo será utilizado para definir el concepto de reparación compatible. Cabe destacar que la definición de restricciones estrictas y homogéneas que utilizaremos es la presentada en la Definición 4.3.

Ejemplo 5.1 *Considere la dimensión $\mathcal{D}_{\text{FT}}$ en la Figura 5.2(b). Las restricciones de integridad para la dimensión son las que se aprecian en la Tabla 5.1. Sin embargo, la dimensión $\mathcal{D}_{\text{FT}}$ no satisface la restricción de integridad estricta* Equipo → Confederación*, ya que AR (Selección de fútbol de Argentina), elemento de la categoría* Equipo *está relacionado mediante rollup a CONM (Confederación sudamericana de fútbol) y UEFA (Unión de federaciones de fútbol europeas), los cuales son elementos de la categoría* Confederación *y NI (Selección de fútbol de Nigeria), elemento de la categoría* Equipo *hace rollup a UEFA y CAF (Confederación africana de fútbol), los cuales son elementos de la categoría* Confederación*. Además, no satisface la restricción de integridad homogénea* Torneo ⇒ Confederación*, ya que ASC (Copa asiática) de la categoría* Torneo *no hace rollup a algún elemento en la categoría* Confederación*. Por lo tanto, esta dimensión no satisface el conjunto Σ de restricciones de integridad y necesita ser reparada. El nivel de convergencia de esta dimensión es* Confederación *el cual, además, es independiente.*

Note que no existe restricción de integridad entre la categoría Equipo *y* Sponsor *por lo que el hecho de que AU (Selección de fútbol de Australia) y NI no hagan rollup a algún elemento en esta categoría no implica que se viole alguna restricción de integridad homogénea, lo mismo sucede con CH Selección de fútbol de Chile que hace rollup a los elementos CC (Coca-Cola) y GT Gillette en la categoría* Sponsor*, donde no se viola ninguna restricción estricta.* □



(a) Esquema de Jerarquía $\mathcal{H}_{\text{FT}}$

(b) Dimensión Equipos de Fútbol $\mathcal{D}_{\text{FT}}$ inconsistente

Figura 5.2: Esquema y Dimensión Equipos de Fútbol

La motivación detrás de la definición de reparación compatible es crear un método que permita obtener respuestas aproximadas desde dimensiones inconsistentes respecto de las RI estrictas y homogéneas, principalmente como una mejora de la dimensión canónica propuesta en (**?**), ya que se desea un método que no requiera computar todas las reparaciones minimales de una dimensión para dejarla consistente respecto al conjunto Σ de RI. Cabe destacar que la reparación

Restricciones Estrictas	**Restricciones Homogénea**
Equipo→Zona	Equipo⇒Zona
Equipo→Torneo	Equipo⇒Torneo
Equipo→Confederación	Zona⇒Confederación
Zona→Confederación	Torneo⇒Confederación
Torneo→Confederación	

Tabla 5.1: Restricciones de integridad de la dimensión Equipos de Fútbol $\mathcal{D}_{\text{FT}}$

compatible adopta conceptos propuestos en (**?**) como la generación de elementos conjuntos. Sin embargo, nosotros, pretendemos que la generación de estos elementos solo se realice como última opción de reclasificación de elementos y solo en las categorías conflictivas. La idea es no tener que agregar más elementos de los necesarios en las categorías, y que los cambios de arcos entre elementos no generen nuevas inconsistencias.

El proceso para obtener una reparación compatible es como sigue. Primero, existe una dimensión de la forma descrita en la Definición 4.1 que es inconsistente con respecto a sus restricciones de integridad. Esta dimensión necesita ser reparada, por lo que se construirá una dimensión extendida, es decir, una nueva dimensión que podría tener en sus categorías elementos de tipo conjunto. En el proceso de obtención de reparación compatible se buscará realizar en lo posible pocos cambios respecto de la dimensión original, y además en cada paso de reparación se buscará no generar nuevas inconsistencias respecto a las restricciones de integridad. Para comprender la definición de reparación compatible es necesario analizar las siguientes observaciones.

Observación 5.1 (Evidencias de consistencia en restricciones estrictas) *Por cada restricción estricta ic: $c_i \to c_j$ y por cada elemento e en una categoría c_i que es inconsistente con respecto a ic, esto es e rollup a dos elementos diferentes en c_j, digamos que a los elementos e_1 y e_2. El algoritmo busca evidencia de consistencia en la dimensión que permite elegir uno de los padres en la categoría c_j. En este caso, la evidencia es la existencia de otro elemento en c_i que es consistente con respecto a la restricción de estricta ic, y hace rollup a uno de los elementos en c_j (e_1 o e_2). La existencia de tal elemento nos indica qué camino de e a los elementos de c_j tienen que ser preservados. Utilizando esta evidencia, se actualiza la dimensión $\mathcal{X}$. En los casos en que no se encuentre evidencia, se agrega en la categoría c_j un nuevo elemento que es el conjunto de elementos a los que e hace rollup en esa categoría, por ejemplo, se inserta en c_j el elemento $\{e_1, e_2\}$.* □

Ejemplo 5.2 *Para la dimensión Equipos de Fútbol de la Figura 5.2(b), la cual es inconsistente respecto a la restricción estricta φ_e : Equipo→Confederación, a través de los elementos AR y NI. Para AR el elemento evidencia es CH, ya que éste no participa en inconsistencias de la restricción φ_e, y ambos comparten al elemento SA en el camino que conduce a CONM en la categoría Confederación. Para NI no hay evidencia de consistencia.* □

El concepto de evidencia de consistencia de restricciones estrictas nos ayudará a decidir qué acciones, en términos de reclasificación de arcos, se deben tomar para restaurar la consistencia.

En el caso del ejemplo anterior, el primer cambio que se llevaría a cabo sería el de cambiar el padre del elemento h_1 al elemento c_2 dada la evidencia del elemento s_2.

En el caso de restricciones estrictas, es necesario verificar que el cambio a realizar en la dimensión para restaurar la consistencia, no provoque nuevas inconsistencias respecto a restricciones estrictas.

Observación 5.2 (Evidencias de consistencia en restricciones homogéneas) *Por cada restricción homogénea $ic : c_i \Rightarrow c_j$ y para cada elemento e en una categoría c_i que es inconsistente con respecto a ic, el algoritmo busca evidencia de consistencia en la dimensión, que pueda ser utilizada para justificar la mejor forma de reparar una inconsistencia del tipo ic, esto con el objetivo de que una reparación para esta restricción no genere violaciones de la restricción estricta. Intuitivamente, la evidencia de consistencia nos dice a qué elemento(s) en c_j debe hacer rollup e. En primer lugar, se busca un elemento e' de una categoría c_a con $c_a \nearrow c_i$ tal que (e', e). Si tal elemento existe, entonces se busca el ancestro de e' en la categoría nivel de convergencia. Luego, se encuentra en la categoría c_j los elementos de tal manera que sus ancestros sean los mismos que el ancestro del elemento e'. La evidencia se utiliza luego para actualizar la dimensión. Si hay más de una evidencia, se elige la primera. De esta manera la reparaciones de restricciones homogénea no generan inconsistencias para las restricciones estrictas. Por otro lado, si no se encuentra evidencia, se reparará dejando solo el arco (e, e''), donde e'' es el primer elemento que se encuentra en la categoría c_j. Este último caso puede crear nuevas inconsistencias del tipo estricta, que serán resueltas posteriormente por el algoritmo.* □

Ejemplo 5.3 *Para la dimensión Equipos de Fútbol de la Figura 5.2(b), la cual es inconsistente respecto a la restricción homogénea φ_h : Torneo⇒Confederación, ya que ASC no alcanza ningún elemento en la categoría Confederación. El elemento AU es evidencia de consistencia para ASC, ya que este elemento alcanza a ASC. AU no participa en la violación de la restricción estricta entre Equipo y Confederación (φ_e), pero podría volverse inconsistente si se le asigna un padre inadecuado a ASC. En este caso, para restaurar la consistencia de homogeneidad se debe conectar ASC al elemento AFC en la categoría Confederación.* □

A continuación se define y ejemplifica el concepto de reparación compatible.

5.2. Definición y Algoritmo de la Reparación Compatible

Definición 5.3 (Reparación Compatible) Sea $\mathcal{D} = (\mathcal{H}_\mathcal{D}, \mathcal{E}_\mathcal{D}, \mathsf{Elem}_\mathcal{D}, <_\mathcal{D})$ una dimensión inconsistente respecto a alguna de sus RI estrictas y homogéneas, su reparación compatible es $\mathcal{X} = (\mathcal{H}_\mathcal{X}, \mathcal{E}_\mathcal{X}, \mathsf{CElem}_\mathcal{X}, \ll_\mathcal{X})$ denotada por $\mathsf{Comp}(\mathcal{D})$, y es obtenida utilizando el Algoritmo 1. □

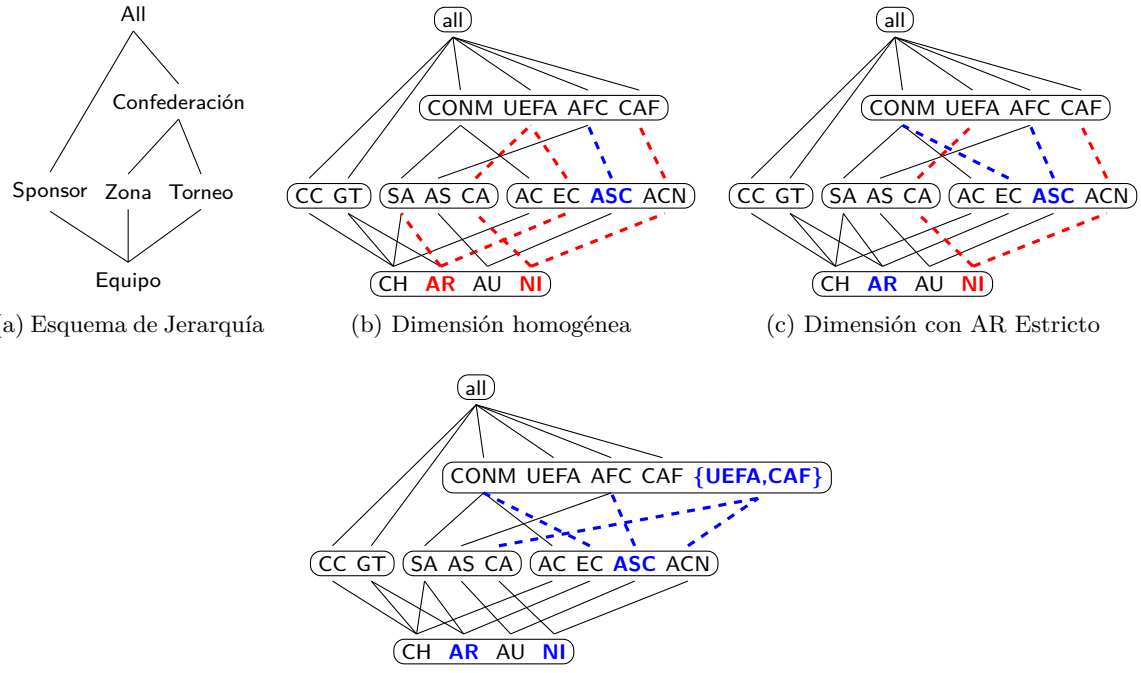
La reparación compatible $\mathcal{X} = (\mathcal{H}_\mathcal{X}, \mathcal{E}_\mathcal{X}, \mathsf{CElem}_\mathcal{X}, \ll_\mathcal{X})$ de $\mathcal{D} = (\mathcal{H}_\mathcal{D}, \mathcal{E}_\mathcal{D}, \mathsf{Elem}_\mathcal{D}, <_\mathcal{D})$ cumple con las siguientes propiedades:

(a) $\mathcal{X}$ es estricta y homogénea.
(b) $\mathcal{H}_\mathcal{D} = \mathcal{H}_\mathcal{X}$.
(c) $\mathcal{E}_\mathcal{D} = \mathcal{E}_\mathcal{X}$.

(d) $\forall x, \forall \mathsf{c} \ / x \in \mathsf{Elem(c)} \leftrightarrow \{x\} \in \mathsf{CElem(c)}$.

A continuación se procede a describir el algoritmo implementado para obtener la reparación compatible, aplicándolo a la dimensión inconsistente $\mathcal{D}_{\mathrm{FT}}$ de la Figura 5.2.

El Algoritmo 1 comienza evaluando la consistencia de la dimensión respecto de las restricciones de integridad homogénea, del conjunto de restricciones de integridad homogéneas que posee la dimensión, se procede a obtener cuales de estas son inconsistentes (línea 1.5), para la Figura 5.2 la única restricción de tipo homogénea que no se cumple es φ_1 : Torneo⇒Confederación, esta inconsistencia se almacena en la lista *CC*. Posteriormente, el algoritmo calcula cuantas restricciones se encuentran en la lista *CC* (línea 1.6) (para el caso solo una), luego, por cada restricción que se viole (línea 1.7) se obtiene la categoría nivel de convergencia de la dimensión donde no se cumple la restricción (línea 1.8), en este caso la categoría nivel de convergencia es Confederación, luego se obtiene la categoría inferior de la restricción que se está analizando (línea 1.9) y la categoría superior de la misma (línea 1.10) las cuales son Torneo y Confederación respectivamente en la dimensión $\mathcal{D}_{\mathrm{FT}}$. Una vez obtenidos estos datos el algoritmo procede a buscar que elementos participan de la inconsistencia φ_1 (línea 1.11) y luego se analiza cuantos elementos son para realizar el ciclo (línea 1.12), en la dimensión $\mathcal{D}_{\mathrm{FT}}$ el solo existe un elemento que participa de esta inconsistencia, el cual es ASC y se almacena en *InconsistentC*.



(a) Esquema de Jerarquía (b) Dimensión homogénea (c) Dimensión con AR Estricto

(d) Reparación Compatible $\mathcal{X}_{\mathrm{FT}}$

Figura 5.3: Proceso de Reparación de la Dimensión $\mathcal{D}_{\mathrm{FT}}$ de la Figura 5.2 y Reparación Compatible $\mathcal{X}_{\mathrm{FT}}$ obtenida mediante el Algoritmo 1

Por cada elemento que participa de la inconsistencia φ_1 (línea 1.13) se procede a evaluar si existe evidencia de consistencia (línea 1.14) para (en lo posible) hacer una reclasificación que no genere inconsistencias estrictas. En la dimensión $\mathcal{D}_{\mathrm{FT}}$ y la restricción φ_1 el único elemento que participa de esta inconsistencia es ASC y el elemento AU es evidencia de consistencia para ASC, ya que este elemento alcanza a ASC en Torneo (ver Tabla 5.2). *AU* no participa en la violación de la restricción estricta entre Equipo y Confederación, en este caso, para restaurar la consistencia de homogeneidad se debe conectar ASC al elemento AFC en la categoría Confederación (línea 1.15). Si no existiese evidencia de consistencias, el algoritmo asigna el primer elemento de la categoría superior (en este caso Confederación) como padre para hacer la actualización (línea 1.17). Como no existen más elementos que participen de la inconsistencia φ_1 y tampoco hay más RI que no sean homogeneas, el proceso finaliza, dejando la dimensión $\mathcal{D}_{\mathrm{FT}}$ tal como se presenta en la Figura 5.3(b).

Una vez resueltas las inconsistencias de homogeneidad, el Algoritmo 1 procede a evaluar que RI estrictas no se cumplen en la dimensión (línea 1.19), almacenándola en *SC* y calcular cuantas son (línea 1.20), para la dimensión $\mathcal{D}_{\mathrm{FT}}$ la única RI que no se cumple es φ_2 : Equipo→Confederación. Posteriormente por cada RI que no se cumple (línea 1.21) se obtiene la categoría nivel de convergencia de la dimensión (línea 1.22), en este caso la categoría nivel de convergencia es Confederación, luego se obtiene la categoría inferior de la restricción que se está analizando (línea 1.23) y la categoría superior de la misma (línea 1.24) las cuales son Equipo y Confederación respectivamente en la dimensión $\mathcal{D}_{\mathrm{FT}}$. Una vez obtenidos estos datos, el algoritmo procede a buscar que elementos participan de la inconsistencia φ_2 (línea 1.25) y luego se analiza cuantos elementos son para realizar el ciclo (línea 1.26), en la dimensión $\mathcal{D}_{\mathrm{FT}}$ existen dos elementos que participan de esta inconsistencia, los cuales son AR y NI y se almacenan en *InconsistentS*.

Por cada elemento que participa de la inconsistencia φ_2 (línea 1.27) se procede a evaluar si existe evidencia de consistencia (línea 1.28). Se comienza por el primer elemento en la lista *InconsistentS*, el cual en este caso es AR, para AR el elemento evidencia es CH, ya que éste no participa en inconsistencias de la restricción φ_2, y ambos comparten al elemento SA en el camino que conduce a CONM en la categoría Confederación, como para AR existe una única evidencia de consistencia (ver Tabla 5.3) (línea 1.29) con lo cual, se procede a hacer un cambio de arcos, haciendo que EU de la categoría Torneo haga rollup a CONM en vez de UEFA (línea 1.29) dejando a AR consistente, tal como se ve en la Figura 5.3(c). Luego el algoritmo continua con el siguiente elemento en *InconsistentS* (línea 1.27), el cual es NI, para el cual procede a evaluar si existe evidencia (línea 1.28). Para NI no hay evidencia de consistencia (línea 1.33) por lo que se procede a obtener los elementos a los cuales este hace rollup, en la dimensión $\mathcal{D}_{\mathrm{FT}}$ para generar el elemento conjunto (línea 1.34), luego se inserta este elemento conjunto en la categoría Confederación (*ClCategory*) (línea 1.35) y luego se procede ha hacer la actualización haciendo que NI haga rollup a este nuevo elemento conjunto (línea 1.36). En el caso de la dimensión $\mathcal{D}_{\mathrm{FT}}$ el elemento conjunto que se genera es {UEFA,CAF} puesto que NI hace rollup a UEFA por medio de la categoría Zona y a CAF por medio de la categoría Torneo, al no haber evidencia se genera este elemento conjunto y se inserta en la categoría correspondiente (Confederación) y luego se hace el cambio de arcos (CA, UEFA) por (CA, {UEFA,CAF}) y (ACN, CAF) por (ACN, {UEFA, CAF}) dejando a NI consistente respecto a la restricción estricta. Si existiese el caso de que un

elemento poseyera más de una evidencia (línea 1.31) hacia elementos distintos, se busca hacer un cambio de arcos en categorías inferiores (más cercanas a cat_c).

Como no existen más elementos que participen de la inconsistencia φ_2 y tampoco hay más RI que no sean estrictas, el proceso finaliza, generando la reparación compatible $\mathcal{X}_{\text{FT}}$ tal como se presenta en la Figura 5.3(d), la cual es una dimensión extendida que es estricta y homogénea.

Utilizando la descripción del Algoritmo 1 y la dimensión de la Figura 5.3(d), se validan las propiedades de la Definición 5.3 por medio del Ejemplo 5.4.

Ejemplo 5.4 *Para la dimensión $\mathcal{X}_{\text{FT}}$ de la Figura 5.3(d) se pueden comparar sus propiedades para validar que la reparación compatible cumple las propiedades presentada en la Definición 5.3.*

(a) $\mathcal{X}$ es estricta y homogénea por lo presentado procedimiento del Algoritmo 1.

(b) Las dimensiones $\mathcal{D}_{\text{FT}}$ y $\mathcal{X}_{\text{FT}}$ poseen el mismo Esquema de Jerarquía $\mathcal{H}$ por lo que se puede apreciar en las Figuras 5.2(a) y 5.3(d) respectivamente, la dimensión $\mathcal{X}_{\text{FT}}$ no genera nuevas categorías, por lo tanto el esquema de jerarquía $\mathcal{H}$ se mantiene intacto.

(c) Para las dimensiones $\mathcal{D}_{\text{FT}}$ y $\mathcal{X}_{\text{FT}}$ se mantienen los elementos $\mathcal{E}_{\mathcal{D}_{\text{FT}}}$ y $\mathcal{E}_{\mathcal{X}_{\text{FT}}}$ respectivamente, puesto que no se crean elementos ficticios, solo se agregan subconjuntos de los mismos.

(d) La similitud que existe entre $\mathsf{Elem}_{\mathcal{D}_{\text{FT}}}$ y $\mathsf{CElem}_{\mathcal{X}_{\text{FT}}}$ es que el primero dado una categoría obtiene los elementos de ésta y para una dimensión $\mathcal{X}_{\text{FT}}$, $\mathsf{CElem}_{\mathcal{X}_{\text{FT}}}$ obtiene los elementos y subconjuntos en esta categoría,

- $\mathsf{Elem}_{\mathcal{D}_{\text{FT}}}(\mathsf{All}) = \{all\}$,
- $\mathsf{Elem}_{\mathcal{D}_{\text{FT}}}(\mathsf{Confederación}) = \{CONM,\ UEFA,\ AFC,\ CAF\}$,
- $\mathsf{Elem}_{\mathcal{D}_{\text{FT}}}(\mathsf{Sponsor}) = \{CC,GT\}$,
- $\mathsf{Elem}_{\mathcal{D}_{\text{FT}}}(\mathsf{Zona}) = \{SA,\ AS,\ CA\}$,
- $\mathsf{Elem}_{\mathcal{D}_{\text{FT}}}(\mathsf{Torneo}) = \{AC,\ EC,\ ASC,\ ACN\}$,
- $\mathsf{Elem}_{\mathcal{D}_{\text{FT}}}(\mathsf{Equipo}) = \{CH,\ AR,\ AU,\ NI\}$,
- $\mathsf{CElem}_{\mathcal{X}_{\text{FT}}}(\mathsf{All}) = \{all\}$,
- $\mathsf{CElem}_{\mathcal{X}_{\text{FT}}}(\mathsf{Confederación}) = \{CONM,\ UEFA,\ AFC,\ CAF,\ \{UEFA,CAF\}\}$,
- $\mathsf{CElem}_{\mathcal{X}_{\text{FT}}}(\mathsf{Sponsor}) = \{CC,GT\}$,
- $\mathsf{CElem}_{\mathcal{X}_{\text{FT}}}(\mathsf{Zona}) = \{SA,\ AS,\ CA\}$,
- $\mathsf{CElem}_{\mathcal{X}_{\text{FT}}}(\mathsf{Torneo}) = \{AC,\ EC,\ ASC,\ ACN\}$,
- $\mathsf{CElem}_{\mathcal{X}_{\text{FT}}}(\mathsf{Equipo}) = \{CH,\ AR,\ AU,\ NI\}$ □

Note que el esquema de jerarquía de una dimensión clásica y el de una dimensión extendida es siempre el mismo por lo tanto solo lo denotaremos como $\mathcal{H}$. De la misma forma que sucede con el esquema de jerarquía, el conjunto de elementos para la dimensión clásica y para la extendida se denotará con $\mathcal{E}$.

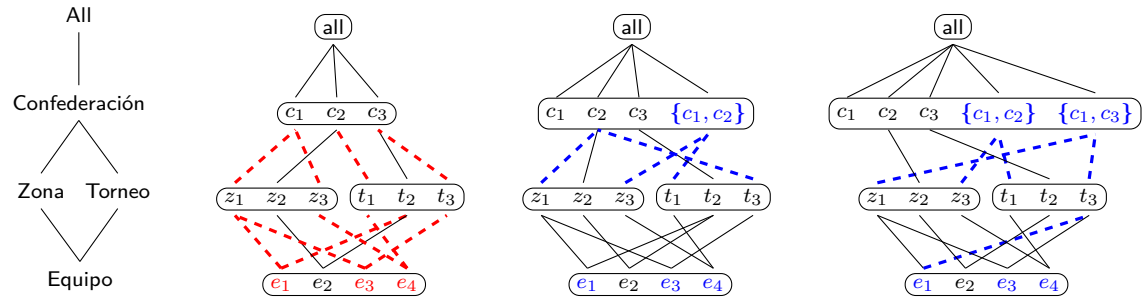
Ambas dimensiones ($\mathcal{D}_{\text{FT}} = (\mathcal{H}, \mathcal{E}, \mathsf{Elem}_{\mathcal{D}_{\text{FT}}}, <_{\mathcal{D}_{\text{FT}}})$ y $\mathcal{X}_{\text{FT}} = (\mathcal{H}, \mathcal{E}, \mathsf{CElem}_{\mathcal{X}_{\text{FT}}}, \ll_{\mathcal{X}_{\text{FT}}})$) en cambio tienen tuplas que si bien son semejantes entregan resultados distintos, una de ellas es el caso de $\mathsf{Elem}_{\mathcal{D}_{\text{FT}}}$ para la dimensión clásica y $\mathsf{CElem}_{\mathcal{X}_{\text{FT}}}$ para la dimensión extendida, la diferencia consiste en que $\mathsf{Elem}_{\mathcal{D}_{\text{FT}}}$ devuelve un conjunto de elementos según cada categoría de la dimensión clásica, en cambio $\mathsf{CElem}_{\mathcal{X}_{\text{FT}}}$ al estar definida para una dimensión extendida, esta devuelve un subconjunto de elementos de cada categoría. Y de forma similar, la clausula transitiva y reflexiva de una dimensión clásica y extendida son similares, pero con la diferencia que la dimensión extendida incluye subconjunto de elementos de las categorías.

Los algoritmos presentados en esta tesis siempre producen una reparación compatible para una dimensión $\mathcal{D}$ que es inconsistente con respecto a un conjunto de RI estrictas y homogéneas.

5.2.1. La Reparación Compatible no es Única

Dependiendo de cómo se implementen las opciones que toma el Algoritmo 1 pueden existir varias reparaciones compatibles para una dimensión dada. Esto se ilustra en el siguiente ejemplo.

Ejemplo 5.5 *Sea el esquema Jerárquico y la dimensión* Equipos de Fútbol *de la Figura 5.4(a)-(b), donde la dimensión* $\mathcal{D}'_{\text{FT}}$ *es inconsistente con respecto a la restricción estricta* φ_2 :Equipo→Confederación *donde el elemento* e_1 *de la categoría* Equipo *rollup a los elementos* c_1 *y* c_2 *de la categoría* Confederación*, el elemento* e_3 *rollup a los elementos* c_1 *y* c_3*, y* e_4 *rollup a los elementos* c_1 *y* c_2*. Con lo anterior se puede decir que los elementos* e_1*,* e_3 *y* e_4 *son inconsistentes respecto de la restricción estricta* φ_2 *y solo el elemento* e_2 *es consistente. No hay inconsistencia respecto a las restricciones homogéneas. El conjunto* Σ *de restricciones de integridad de esta dimensión es el mismo de la Tabla 5.1.*



(a) Esquema Jerárquico $\mathcal{D}'_{\text{FT}}$

(b) Dimensión Inconsistente $\mathcal{D}'_{\text{FT}}$ Equipo→Confederación

(c) Reparación Compatible comenzando del elemento e_1 o e_3

(d) Reparación Compatible comenzando del elemento e_4

Figura 5.4: Aplicación del Algoritmo para una dimensión inconsistente iniciando la reparación desde distintos elementos en la categoría inferior

Para e_1 *existe evidencia por* e_2 *que indica que hacer el rollup hacia* c_2 *es correcto, en cambio, no existe evidencia para elegir el camino hacia* c_1*, por tanto se realiza el cambio de arco de* (z_1, c_1) *a* (z_1, c_2) *(línea 1.29), el cambio anterior hace que* e_3 *ahora rollup a* c_2 *y a* c_3 *en vez de hacer rollup a* c_1 *y* c_3 *(ver Tabla 5.5). Siguiendo con el elemento* e_3 *(línea 1.27), el elemento* e_1 *le evidencia que es correcto alcanzar el elemento* c_2 *y no hay evidencia para el elemento* c_3 *(línea 1.28), por tanto se realiza el cambio de arcos* (t_3, c_3) *a* (t_3, c_2) *(línea 1.29) dejando a* e_3 *consistente (ver Tabla 5.6). Por último, en la lista solo queda el elemento* e_4 *(línea 1.27), este elemento no posee ninguna evidencia de consistencia hacia elementos en la categoría* Confederación *(línea 1.28), por lo tanto se procede a generar el elemento conjunto (línea 1.34)* $\{c_1, c_2\}$ *y se realizan los cambios de arcos* (z_3, c_1) *por* $(z_3, \{c_1, c_2\})$ *y* (t_1, c_2) *por* $(t_1, \{c_1, c_2\})$*, quedando finalmente* e_4 *consistente respecto a la restricción estricta* φ_e*. Por último, al no haber más RI estrictas que no se satisfagan, se obtiene la reparación compatible de la Figura 5.4(c).* □

Ahora, si el algoritmo se implementase con otras heurísticas de elección de elementos para reparar (principalmente el orden de estos) es posible obtener otra dimensión compatible para la dimensión $\mathcal{D}'_{\mathrm{FT}}$ como en el caso del siguiente ejemplo.

Ejemplo 5.6 *Si en vez de comenzar a reparar la inconsistencia por el elemento e_1 se comienza por el elemento e_4, el cual, como ya mostramos anteriormente, no posee evidencias de consistencia, por lo tanto, algoritmo procede a generar el elemento conjunto $\{c_1, c_2\}$ en la categoría* Confederación*, y se realiza la respectiva actualización. Luego si se continua con el elemento e_3, el cual, ahora tampoco posee evidencia de consistencia, se genera otro elemento conjunto, esta vez $\{c_1, c_3\}$ y se realiza su respectiva actualización. Por último se continua con e_1 el cual ahora, posee una doble evidencia (línea 1.31), la primera indica que por z_1 de* Zona *debe hacer rollup a $\{c_1, c_3\}$, esta evidencia es dada por el elemento e_3, y la otra evidencia viene dada por el elemento e_2, el cual hace rollup a c_2, por lo cual, es necesario un cambio de arco en un nivel inferior a la categoría cat_h por lo que la actualización genera el cambio de arcos (e_1, t_2) por $(e_1, \{c_1, c_3\})$ lo cual no genera nuevas inconsistencias y deja a e_1 consistente respecto a la restricción estricta φ_e. Finalmente, se genera la reparación compatible de la Figura 5.4(d). Otras reparaciones compatibles es posible generar si se siguen otros métodos para reparar.* □

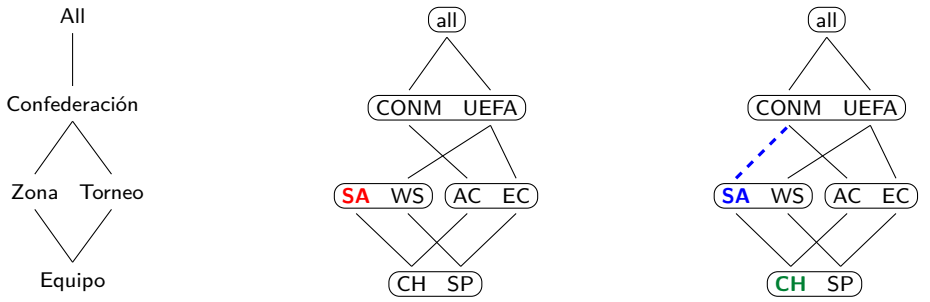
Como se puede apreciar en los ejemplos anteriores, para una dimensión inconsistente respecto a sus RI, se pueden obtener varias reparaciones compatibles dependiendo de la heurística para reparar que se implemente.

A continuación se presentan casos individualizados de los tipos de inconsistencias que se pueden presentar en una dimensión y el resultado obtenido del Algoritmo de la Reparación Compatible, estos casos se presentan individualmente pero no implica que puedan darse todos en una misma dimensión inconsistente.

Ejemplo 5.7 *Dado el esquema jerárquico de la Figura 5.5(a) y la dimensión inconsistente de la Figura 5.5(b) respecto a la restricción homogénea* Zona⇒Confederación*, donde se tiene que el elemento SA de la categoría* Zona *no posee rollup hacia algún elemento en la categoría* Confederación*, en este caso, el algoritmo comprueba si hay elementos que hagan rollup al elemento que participa de la inconsistente, en este caso existe el elemento CH de la categoría* Equipo *que hace rollup a SA en la categoría* Zona*, como CH hace rollup a CONM en la categoría* Confederación*, este elemento le proporciona evidencia a SA, por lo que el algoritmo inserta el arco correspondiente entre SA y CONM, obteniendo la dimensión compatible de la Figura 5.5(c).* □

Hay casos en que el Algoritmo 1 al momento de reparar una inconsistencia respecto a las RI homogéneas, genera una inconsistencia del tipo estricta, a pesar de esto, el algoritmo obtiene finalmente una reparación compatible estricta y homogénea, como en el caso del Ejemplo 5.8.

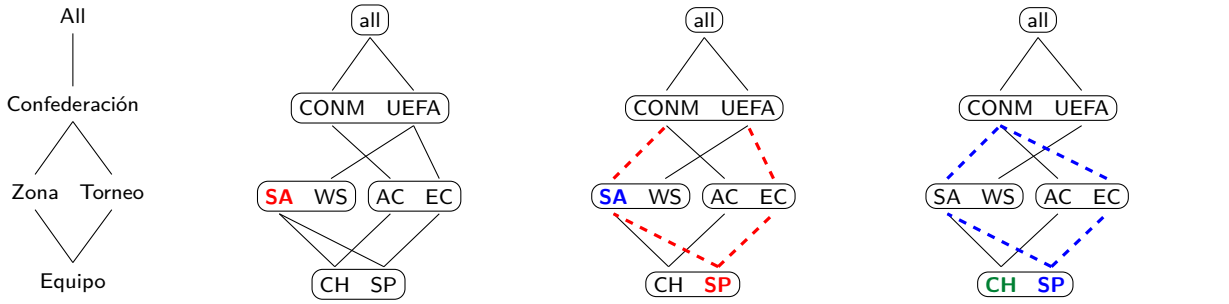
Ejemplo 5.8 *Dado el esquema jerárquico de la Figura 5.6(a) y la dimensión inconsistente de la Figura 5.6(b) respecto a la restricción homogénea* Zona⇒Confederación*, donde se tiene que el elemento SA de la categoría* Zona *no posee rollup hacia algún elemento en la categoría* Confederación*, en este caso, el algoritmo comprueba si hay elementos que hagan rollup al elemento que*



(a) Esquema Jerárquico (b) Dimensión $\mathcal{D}$ No Homogénea (c) Reparación Compatible

Figura 5.5: Dimensión inconsistente respecto de la restricción Zona ⇒ Confederación y su reparación compatible

participa de la inconsistente, en este caso existen los elementos CH y SP de la categoría Equipo que hacen rollup a SA en la categoría Zona, donde, CH hace rollup a CONM y SP hace rollup a UEFA en la categoría Confederación, en este caos a SA se le proporciona una doble evidencia distinta hacia Confederación, por lo que el algoritmo inserta el arco para devolver la homogeneidad a la dimensión, asignándole a SA el primer elemento de la categoría Confederación, en este caso CONM, obteniendo la dimensión de la Figura 5.6(c), el problema de haber hecho esta asignación es que se genera una inconsistencia del tipo estricta, donde se viola la RI Equipo→Confederación, puesto que ahora SP hace rollup a CONM y a UEFA en Confederación, como el algoritmo termino de resolver las inconsistencias del tipo homogéneo, prosigue evaluando si hay violación de alguna de las RI estrictas, donde encuentra que se viola la restricción Equipo→Confederación, por lo que procede a buscar evidencias para el elemento SP que participa de esta inconsistencia. La evidencia se la otorga CH, puesto que poseen un arco en común que es (SA,CONM), por lo que se hace que SP ahora haga rollup hacia CONM, obteniendo finalmente la reparación compatible de la Figura 5.6(d), la cual es estricta y homogénea. □

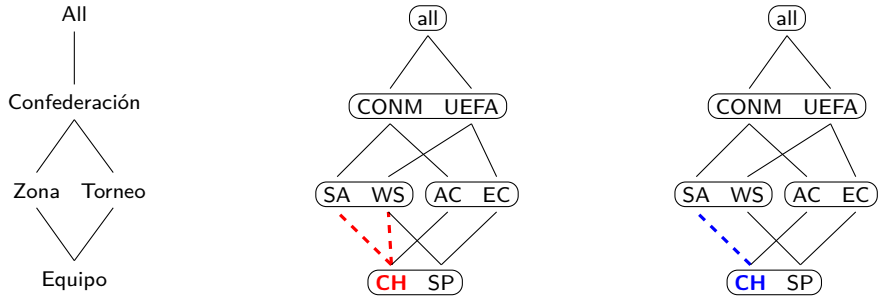


(a) Esquema Jerárquico (b) Dimensión $\mathcal{D}$ No Homogénea (c) Dimensión $\mathcal{X}$ Homogénea pero No Estricta (d) Reparación Compatible

Figura 5.6: Dimensión inconsistente respecto de la restricción Zona ⇒ Confederación y su reparación compatible (generación de inconsistencia estricta)

Los casos anteriores son las principales inconsistencias del tipo homogénea que se pueden dar, a continuación se presentan casos para el tipo de restricciones estrictas posibles, excluyendo el presentado en el Ejemplo 5.8.

Ejemplo 5.9 *Dado el esquema jerárquico de la Figura 5.7(a) y la dimensión inconsistente de la Figura 5.7(b) respecto a la restricción estricta* Equipo→Zona*, donde se tiene que el elemento CH de la categoría* Equipo *hace rollup a los elementos SA y WS en la categoría* Zona*, en este caso, el algoritmo comprueba ha que elemento le hace rollup CH en la categoría nivel de convergencia (*Confederación*) como por SA y AC hace rollup a CONM y por WS hace rollup a UEFA, el algoritmo mantiene el arco (CH, SA) eliminando el arco (CH, WS), por lo que CH ahora hace rollup a CONM, obteniendo finalmente la reparación compatible de la Figura 5.7(c), la cual es estricta y homogénea..* □



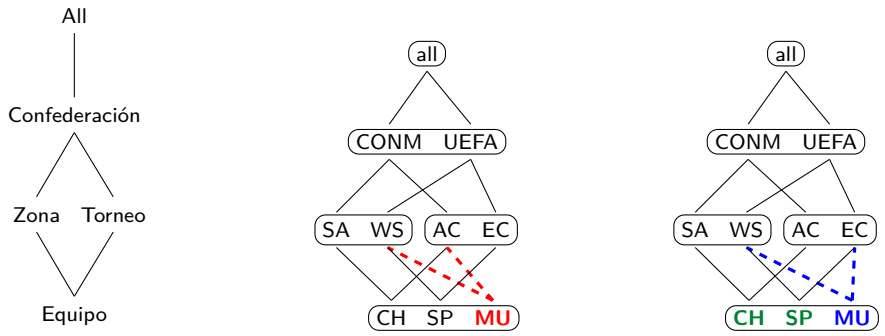
(a) Esquema Jerárquico (b) Dimensión $\mathcal{D}$ No Estricta (c) Reparación Compatible

Figura 5.7: Dimensión inconsistente respecto de la restricción Equipo → Zona y su reparación compatible

Casos como este si suceden en categorías más altas en el esquema jerárquico, como entre las categorías Zona y Confederación es posible que una reparación genere nuevas inconsistencias estrictas en restricciones transitivas como Equipo→Confederación (viendo el esquema de la Figura 5.7(a)) este tipo de inconsistencias son solucionadas al final del algoritmo para evitar que se generen inconsistencias que impidan que el algoritmo termine.

Ejemplo 5.10 *Dado el esquema jerárquico de la Figura 5.8(a) y la dimensión inconsistente de la Figura 5.8(b) respecto a la restricción estricta* Equipo→Confederación*, donde se tiene que el elemento MU de la categoría* Equipo *hace rollup a los elementos CONM y UEFA en la categoría* Confederación*, en este caso, el algoritmo comprueba la evidencia de consistencia, pero se encuentra que CH le indica evidencia hacia CONM y SP le indica evidencia hacia UEFA, por lo tanto el algoritmo procede a reparar en una categoría más baja, en este caso decidiendo cambiar el arco (MU, AC) por (MU, EC), por lo que MU ahora hace rollup a UEFA, obteniendo finalmente la reparación compatible de la Figura 5.8(c), la cual es estricta y homogénea.* □

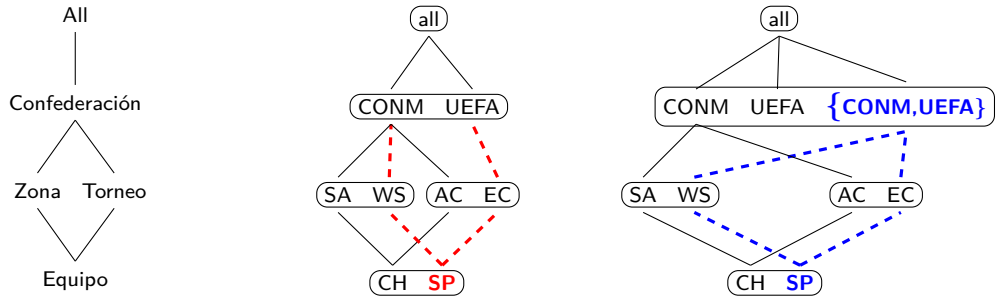
Ejemplo 5.11 *Dado el esquema jerárquico de la Figura 5.9(a) y la dimensión inconsistente de la Figura 5.9(b) respecto a la restricción estricta* Equipo→Confederación*, donde se tiene que el*



(a) Esquema Jerárquico (b) Dimensión $\mathcal{D}$ No Estricta (c) Reparación Compatible

Figura 5.8: Dimensión inconsistente respecto de la restricción Equipo → Confederación y su reparación compatible (múltiple evidencia de consistencia)

elemento SP de la categoría Equipo hace rollup a los elementos CONM y UEFA en la categoría Confederación, en este caso, el algoritmo comprueba la evidencia de consistencia, pero no encuentra ningún elemento consistente que comparta arco con SP, en este caso, el algoritmo inserta el elemento conjunto {CONM, UEFA} en la categoría Confederación y haciendo posteriormente los cambios de arcos correspondiente para que SP ahora haga rollup a este nuevo elemento, este cambio no genera nuevas inconsistencias y finalmente se obtiene la reparación compatible de la Figura 5.9(c), la cual es estricta y homogénea. □



(a) Esquema Jerárquico (b) Dimensión $\mathcal{D}$ No Estricta (c) Reparación Compatible

Figura 5.9: Dimensión inconsistente respecto de la restricción Equipo → Confederación y su reparación compatible (generación de elemento conjunto)

5.2.2. Complejidad del Algoritmo

Sea n es el número máximo de elementos de una categoría, r es el número máximo de las relaciones del rollup entre dos categorías, m_c es el número de restricciones homogéneas y m_s es el número de restricciones estrictas. El costo de verificar si una restricción homogénea $c_i \Rightarrow c_j$ se viola es $O(n^2)$ ya que tenemos que comprobar si hay una relación rollup entre cada elemento

en c_i y un elemento en c_j . El costo de comprobar si una restricción estricta $c_i \rightarrow c_j$ se viola es $O(r^2)$ ya que la relación $r_{c_i}^{c_j}$ es de tamaño $O(r)$ y lo necesitamos para comparar cada tupla en ella con todas sus tuplas. Por lo tanto, el costo de las líneas 1.5 y 1.19 en el Algoritmo 1, es $O((m_c * n^2) + (m_s * r^2))$. A pesar de que en el peor de los casos r es $O(n^2)$, se espera que r sea $O(n)$ ya que en general el número de violaciones de una restricción es pequeña, y por lo tanto la mayoría de los elementos en una categoría se resumen en un único elemento de una categoría superior.

Dada una restricción ic de la forma $c_i \rightarrow c_j$ o $c_i \Rightarrow c_j$, SearchEvidence calcula las rutas entre un elemento e' que pertenece a la categoría c_a con $c_a \nearrow c_i$, tal que (e', e) con $e \in c_i$, y los elementos en el nivel de conflicto de $\mathcal{D}$. El número de rutas y el costo de computarla, en el peor de los casos es $O(n^{|\mathcal{C}|})$, pero en la mayoría de los casos, en que el número de inconsistencias es bajo, es de $O(|\mathcal{C}|)$. Por último, las funciones C_Repair($\mathcal{X}$,Evidence) y C_Repair($\mathcal{X}$, Evidence) poseen un costo de $O(|\mathcal{C}|)$.

Por lo tanto, el Algoritmo 1 se ejecuta en el peor de los casos en $O(m_c(n^2+n^{|\mathcal{C}|})+m_s(r^2+n^{|\mathcal{C}|}))$ y en el caso esperado en $O(n^2(m_c + m_s))$.

En el siguiente capítulo se realizan las pruebas evaluar el tiempo de ejecución en casos más reales y las respuestas aproximadas que se pueden obtener de la reparación compatible, aplicando las definición de respuesta aproximada (ver Sección 4.2), aplicándolo a los operadores de agregación `SUM`, `MIN`, `MAX` y `COUNT`.

Algoritmo 1: Algoritmo para Computar la Reparación Compatible

Input: Dimension $\mathcal{D}$ and Set of constraints Σ
Output: Compatible Repair $\mathcal{X}$

```
1.1  List CC, SC, Evidence
1.2  char *cat_h, *cat_p, *Set, InconsistentC, InconsistentS
1.3  int x, i, j
1.4  X ← D
1.5  GetInconsistenceC(Σc,D,CC)
1.6  int num_h ← total(CC)
1.7  for i ← 1 to num_h do
1.8      ClCategory ← GetConflictingCategory()
1.9      cat_c ← Category(CC[i],1)
1.10     cat_p ← Category(CC[i],2)
1.11     InconsistentC ← GetInconsistentCC(CC[i])
1.12     x ← total(InconsistentC)
1.13     for j ← 1 to x do
1.14         Evidence ← SearchEvidence(InconsistentC)
1.15         if Evidence ≠ null then X ← C_Repair(X,Evidence)
1.16
1.17         else X ← C_Repair(X,GetFirtsElement(cat_p))
1.18
1.19 GetInconsistenceS(Σs,D, SC)
1.20 int num_s ← total(SC)
1.21 for i ← 1 to num_s do
1.22     ClCategory ← GetConflictingCategory()
1.23     cat_c ← Category(SC[i],1)
1.24     cat_p ← Category(SC[i],2)
1.25     InconsistentS=GetInconsistentSC(SC[i])
1.26     x ← total(InconsistentS)
1.27     for j ← 1 to x do
1.28         Evidence ← SearchEvidence(InconsistentS)
1.29         if Evidence ≠ null and Leng(Evidence)=1 then X ←C_Repair(X, Evidence)
1.30
1.31         else if Evidence ≠ null and Leng(Evidence) ≥ 2 then X ←C_Repair(X,
             Evidence)
1.32
1.33         else
1.34             Set ← Elements(InconsistentS,cat_c,cat_p,ClCategory)
1.35             InsertSet(ClCategory,Set)
1.36             X ←C_Repair(X, Set)
```

Caminos de Evidencia	RI inconsistente
AU|Equipo $\mapsto$ AS|Zona $\mapsto$ **AFC**| **Confederación**	Torneo $\Rightarrow$ Confederación

Tabla 5.2: Evidencia para ASC para resolver inconsistencia de homogeneidad

Caminos Elementos Inconsistentes	Elementos de Evidencia para φ_e
AR|Equipo $\rightarrow$ **SA**| **Zona** $\rightarrow$ **CONM**| **Confederación**	CH|Equipo $\rightarrow$ **SA**|**Zona** $\rightarrow$ **CONM**| **Confederación**
AR| Equipo $\rightarrow$ EC| Torneo $\rightarrow$ UEFA| Confederación	CH| Equipo $\rightarrow$ AC| Torneo $\rightarrow$ CONM| Confederación

Tabla 5.3: Evidencia para AR para resolver inconsistencia estricta

Lista Inconsistente	Caminos Consistentes
$e_1 \rightarrow z_1 \rightarrow c_1$	$e_2 \rightarrow z_2 \rightarrow c_2$
$e_1 \rightarrow t_2 \rightarrow c_2$	$e_2 \rightarrow t_2 \rightarrow c_2$
$e_3 \rightarrow z_1 \rightarrow c_1$	
$e_3 \rightarrow t_3 \rightarrow c_3$	
$e_4 \rightarrow z_3 \rightarrow c_1$	
$e_4 \rightarrow t_1 \rightarrow c_2$	

Tabla 5.4: Primera Actualización: Lista de Elementos inconsistentes del Ejemplo 5.5

Lista Inconsistente	Caminos Consistentes
$e_3 \rightarrow z_1 \rightarrow c_2$	$e_1 \rightarrow z_1 \rightarrow c_2$
$e_3 \rightarrow t_3 \rightarrow c_3$	$e_1 \rightarrow t_2 \rightarrow c_2$
$e_4 \rightarrow z_3 \rightarrow c_1$	$e_2 \rightarrow z_2 \rightarrow c_2$
$e_4 \rightarrow t_1 \rightarrow c_2$	$e_2 \rightarrow t_2 \rightarrow c_2$

Tabla 5.5: Segunda Actualización: Lista de Elementos inconsistentes del Ejemplo 5.5

Lista Inconsistente	Caminos Consistentes
$e_4 \rightarrow z_3 \rightarrow c_1$	$e_1 \rightarrow z_1 \rightarrow c_2$
$e_4 \rightarrow t_1 \rightarrow c_2$	$e_1 \rightarrow t_2 \rightarrow c_2$
	$e_2 \rightarrow z_2 \rightarrow c_2$
	$e_2 \rightarrow t_2 \rightarrow c_2$
	$e_3 \rightarrow z_1 \rightarrow c_2$
	$e_3 \rightarrow t_3 \rightarrow c_2$

Tabla 5.6: Tercera Actualización: Lista de Elementos inconsistentes del Ejemplo 5.5

Capítulo 6

Experimentos

En este capítulo se realizan pruebas para presentar otros ejemplos de reparaciones compatibles que son posible obtener, las respuestas aproximadas que se obtienen utilizando los distintos operadores de agregación `SUM`, `MIN`, `MAX` y `COUNT` por medio de la reparación compatible, su comparación con la respuesta consistente y la respuesta desde la dimensión inconsistente y por último el tiempo de cómputo para obtener la reparación compatible en dimensiones reales con millones de elementos. Este análisis se llevará a cabo por medio de ejemplos.

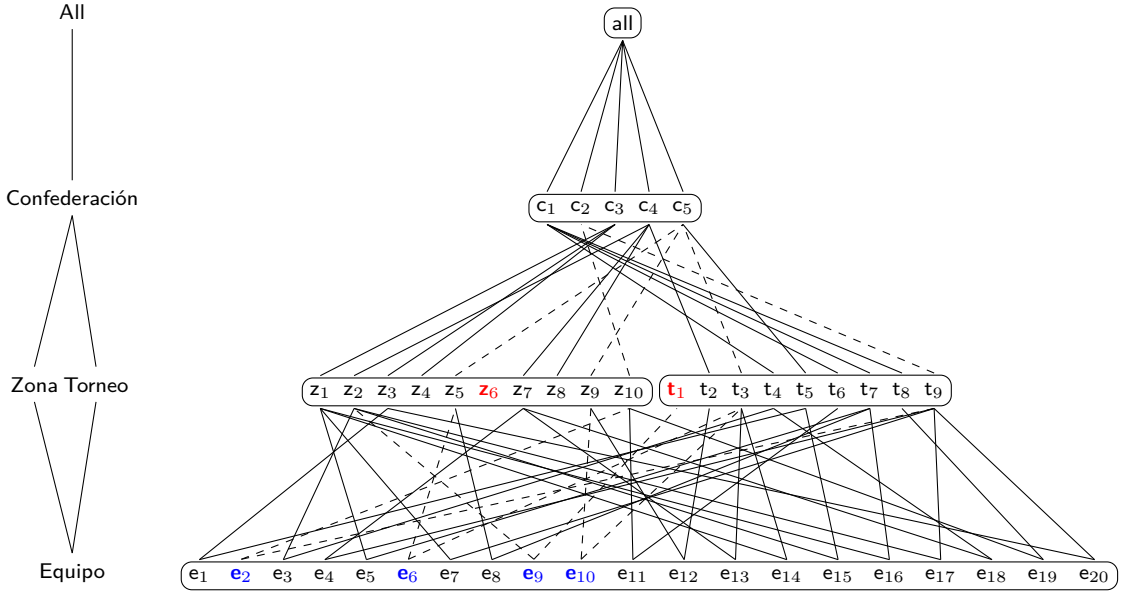
6.1. Obtención de la Reparación Compatible

En la Figura 6.1 se presenta el esquema de jerarquía y la dimensión $\mathcal{D}_e$ Equipos de Fútbol, donde la categoría Equipo está conectado (rollup) con Zona (Zona Geográfica) y ésta a su vez con Confederación. También, la categoría Equipo está conectado a la categoría Torneo y ésta a su vez con Confederación. La categoría más alta es All a la cual llega Confederación. La dimensión $\mathcal{D}_e$ se organiza como se aprecia en la Figura 6.1(b), donde los rollup de los elementos consistentes se presentan con lineas punteadas a diferencia de los casos anteriores, esto es, porque solo cuatro elementos de esta dimensión son consistentes, por lo que hacerlo de forma contraria dificultaría más la visualización de los rollup.[I]

Las restricciones de integridad que debe cumplir la dimensión $\mathcal{D}_e$ vienen dadas por el conjunto Σ de RI que se presentan en la Tabla 6.1, sin embargo, la dimensión $\mathcal{D}_e$ es inconsistente respecto a las restricciones de integridad homogéneas Zona⇒Confederación (elemento z_6) y Torneo⇒Confederación (elemento t_1). También, es inconsistente respecto a la restricción de integridad estricta Equipo→Confederación (elementos $e_1, e_3, e_4, e_5, e_7, e_8, e_{11}, \ldots, e_{20}$). La dimensión $\mathcal{D}_e$ posee además, 4 elementos que no participan de la violación de la restricción estricta Equipo→Confederación, estos elementos se aprecian en la Tabla A.4.

En el caso de la dimensión $\mathcal{D}_e$ cuando se viola la RI estricta Equipo→Confederación se provoca que un equipo pertenezca a más de una confederación, lo que luego se transforma en un conteo

[I] Esta dimensión es un caso muy improbable que suceda en la vida real, lo común es que se obtengan dimensiones inconsistentes con porcentajes inferiores al 50 % de la dimensión, pero para el caso de estas pruebas no hay problema.



(a) Esquema Jerárquico

(b) Dimensión $\mathcal{D}_e$ Inconsistente

Figura 6.1: Esquema y Dimensión $\mathcal{D}_e$ Equipos de Fútbol

Restricciones Estrictas	**Restricciones Homogénea**
Equipo→Zona	Equipo⇒Zona
Equipo→Torneo	Equipo⇒Torneo
Equipo→Confederación	Zona⇒Confederación
Zona→Confederación	Torneo⇒Confederación
Torneo→Confederación	

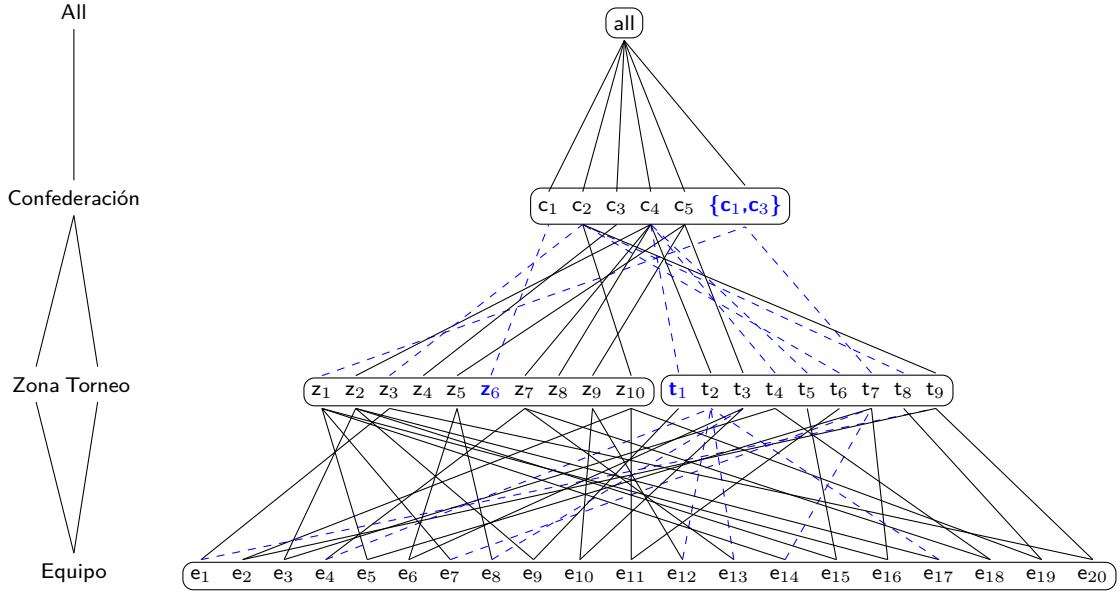
Tabla 6.1: Restricciones de integridad de la dimensión Equipos de Fútbol $\mathcal{D}_e$

erróneo al momento de evaluar consultas pre-computadas (tal como se verá más adelante), en este caso, existe un doble conteo, osea hay equipos que hacen rollup a dos confederaciones distintas, esto se puede apreciar en la Tabla A.5 donde, por ejemplo, el equipo e_3 pertenece a la confederación c_1 y c_4.

Si se computa esta dimensión ($\mathcal{D}_e$) en el Algoritmo 1 se obtiene la reparación compatible de la Figura 6.2(b)[II], la cual es una dimensión extendida que es consistente respecto al conjunto Σ de RI presentados en la Tabla 6.1.

Una vez obtenida la reparación compatible $\mathcal{X}_e$ se puede apreciar la diferencia con la dimensión $\mathcal{D}_e$, en particular se puede ver que en la reparación compatible los elementos consistentes no su-

[II]Los cambios de arcos respecto de la dimensión inconsistente $\mathcal{D}_e$ se representan con lineas seccionadas, en cambio las lineas continuas, son las que no sufrieron cambios al momento de resolver la inconsistencia. Notar que no se afectaron los arcos de los elementos que originalmente eran consistentes respecto a la restricción estricta



(a) Esquema Jerárquico

(b) Reparación Compatible $\mathcal{X}_e$

Figura 6.2: Esquema y Reparación compatible $\mathcal{X}_e$ Equipos de Fútbol

frieron alteraciones respecto de sus confederaciones originales, además solo 4 de los 16 elementos inconsistentes originales no poseían evidencia alguna, por lo que la reparación compatible generó un elemento conjunto al cual le fueron asignados, en este caso, se generó el elemento $\{c_1, c_3\}$, al cual le hacen rollup los elementos e_5, e_7, e_{14} y e_{16}, esto provoca un doble conteo de los datos de la dimensión, pero se disminuirá al momento de obtener las respuestas aproximadas, para el resto de elementos se encontró evidencia, lo que finalmente generó la dimensión $\mathcal{X}_e$, en la Tabla A.6 se pueden apreciar a que confederación finalmente pertenece cada equipo en la dimensión $\mathcal{X}_e$.

Otros ejemplos de reparaciones compatibles han sido presentados en los Capítulos 4 y 5.

Si las vistas pre-computadas se formulan sobre dimensiones inconsistentes obtenemos respuestas incorrectas, entonces podemos utilizar la reparación compatible para obtener respuestas aproximadas a las consultas de agregación. Las respuestas aproximadas se calculan tal como se definieron en la Sección 4.2 para dimensiones extendidas. A continuación se presentan algunas reparaciones compatibles y sus respuestas aproximadas y la comparación con las respuestas obtenidas de la dimensión inconsistente y la respuesta consistentes (ver Definición 2.8). Finalizando con el comportamiento del Algoritmo 1 para una dimensión real.

6.2. Respuestas Aproximadas desde la Reparación Compatible

Considere la dimensión inconsistente $\mathcal{D}_e$, la cual se aprecia en la Figura 6.1(b), y la tabla de hechos Ingresos (Tabla A.1) que almacena los ingresos por concepto de ventas (en miles de dólares estadounidenses) para cada una de las selecciones nacionales que contiene la dimensión.

Si evaluamos la consulta de agregación SQL 6.1 que obtiene los ingresos agrupados por confederación y año en la dimensión $\mathcal{D}_e$ obtenemos respuestas que son incorrectas (ver Tabla A.2), puesto que en esta dimensión existen equipos que pertenecen a dos confederaciones, lo que genera un doble conteo en los ingresos por cada confederación (ver Tabla A.7).

SQL 6.1: Ingresos por Equipo y Año

```
SELECT R1."Confederacion", T."Anio", SUM(I."Ingreso"), MIN(I."Ingreso"), MAX(I."↩
    Ingreso"), COUNT(I."Ingreso")
FROM "R(Equipo,Confederacion)" R1, "Tiempo" T, "Ingresos" I
WHERE R1."Equipo"=I."Equipo" AND T."Fecha"=I."Fecha"
GROUP BY R1."Confederacion", T."Anio"
ORDER BY R1."Confederacion"
```

Si en cambio consideramos la reparación compatible de $\mathcal{D}_e$ obtenida por medio del Algoritmo 1, el cual se aprecia en la Figura 6.2, y la misma tabla de hechos Ingresos (Tabla A.1) evaluados sobre consulta de agregación SQL 6.1 en la reparación compatible $\mathcal{X}_e$ obtenemos las respuestas de la Tabla 6.2.

Como se puede apreciar y tal como se menciono en cuando se definieron las respuestas aproximadas (ver Sección 4.2.1), las respuestas obtenidas en la Tabla 6.2 no permite al usuario ver el efecto de la relación entre los elementos dentro de una categoría.

La Tabla 6.3 muestra las respuestas aproximadas a la consulta de agregación SQL 6.1 sobre la dimensión $\mathcal{X}_e$ y que es obtenida considerando las respuestas que entrega la Tabla 6.2 y aplicando la Definición 4.10 para las respuestas aproximadas de los operadores `SUM/COUNT`, para los operadores de agregación `MIN/MAX` se ha dejado como concepto para definir en trabajo futuro.

Las respuestas presentadas en la Tabla 6.3 indican el rango de valor mínimo y máximo que puede tomar cada tupla (en le caso de `SUM` y `COUNT`) y los posibles valores que puede tomar para los operadores `MIN` y `MAX`.

Si comparamos las respuestas obtenidas de la dimensión inconsistente $\mathcal{D}_e$ (ver Tabla A.3) y las respuestas obtenidas de la reparación compatible $\mathcal{X}_e$ presentadas en la Tabla 6.3 se puede apreciar que esta dimensión reduce considerablemente el doble conteo de datos (ver Tabla A.6) lo que provoca una respuesta mantiene los elementos consistentes sin modificaciones y genera una respuesta en base a la evidencia existente en la dimensión para responder a los elementos inconsistentes.

Para las respuestas presentadas en la Tabla 6.4 podemos decir que es una respuesta mucho mejor que la obtenida de la dimensión inconsistente, puesto que aquí se ha reducido la cantidad de elementos que provocan un doble conteo de datos en la dimensión, además, en la respuesta aproximada, estos elementos que son imprecisos a la hora de entregar respuesta, siempre comienzan como valor mínimo en cero, puesto que como no se esta seguro de su valor se deja a discreción del administrador decidir cuando es correcto o no el valor, por otro lado, el resto de elementos pertenecientes a la dimensión $\mathcal{D}_e$ les fue asignado un único ancestro en la categoría Confederación,

Confederación	Año	SUM	MIN	MAX	COUNT
$\{c_1, c_3\}$	2010	60	9	18	5
$\{c_1, c_3\}$	{2010,2012}	10	10	10	1
$\{c_1, c_3\}$	{2010,2013}	14	14	14	1
$\{c_1, c_3\}$	2011	45	4	18	4
$\{c_1, c_3\}$	2012	19	1	7	4
$\{c_1, c_3\}$	2013	14	4	6	3
c_2	2010	36	6	15	3
c_2	{2010,2011}	4	4	4	1
c_2	{2010,2013}	14	1	13	2
c_2	2011	46	3	16	5
c_2	2012	63	2	19	5
c_2	2013	73	8	16	6
c_4	2010	70	1	20	6
c_4	{2010,2011}	15	15	15	1
c_4	{2010,2012}	26	9	17	2
c_4	{2010,2013}	1	1	1	1
c_4	2011	26	1	16	5
c_4	{2011,2012}	1	1	1	1
c_4	2012	61	3	18	5
c_4	{2012,2013}	8	8	8	1
c_4	2013	52	4	15	6
c_5	2010	21	2	10	3
c_5	{2010,2011}	4	4	4	1
c_5	2011	45	10	19	3
c_5	{2011,2013}	39	19	20	2
c_5	2012	37	1	17	5
c_5	2013	22	3	11	3

Tabla 6.2: Respuestas a la consulta SQL 6.1 agrupada por Confederación y Año para la Figura 6.2

puesto que existía evidencia para hacer un cambio de arcos y dejarlo consistente. Esto implica que se pueda tener una respuesta para los distintos equipos. De la misma forma, el conteo de elementos por cada tupla se redujo considerablemente en la reparación compatible $\mathcal{X}_e$.

Algo similar sucede con las respuestas presentadas en la Tabla 6.5, donde ahora las respuestas aproximadas son un rango de valores que puede tomar la dimensión, donde el primer valor es el que es seguro para respuesta, el resto de elementos existentes, son los que están asignados a elementos conjuntos en la dimensión $\mathcal{X}_e$ (ver Tabla 6.2), de esta forma, se responde a la incertidumbre para los operadores de agregación `min` y `max` en la respuesta aproximada, el primer valor es cero, esto indica que no hay valor seguro para esa tupla, por lo que queda a decisión del administrador nuevamente elegir el valor más adecuado para la respuesta.

Si bien, las respuestas aproximadas son en este caso mucho mejores que las obtenidas desde la dimensión inconsistente $\mathcal{D}_e$, lo ideal es poder comparar la respuesta aproximada con la respuesta consistente, la que recordemos, son los elementos que son verdad en cada una de las reparaciones minimales. Como ya se menciono, el cómputo de las reparaciones minimales es NP-Complejo, sin embargo en (**?**) proponen utilizar DLV[III] para computar estas reparaciones minimales. El

[III] Sistema deductivo de base de datos basado en programación lógica disyuntiva (**?**).

Conf.	Año	SUM	MIN	MAX	COUNT
c_1	2010	[0, 84]	{0,9,10,14}	{0,18,10,14}	[0,3]
c_1	2011	[0, 45]	{0,4}	{0,18}	[0,1]
c_1	2012	[0, 29]	{0,1,10}	{0,7,10}	[0,2]
c_1	2013	[0, 28]	{0,4,14}	{0,6,14}	[0,2]
c_2	2010	[36, 54]	{6,4,1}	{15}	[1,3]
c_2	2011	[46, 50]	{3}	{16}	[1,2]
c_2	2012	[63]	{2}	{19}	[1]
c_2	2013	[73, 87]	{8,1}	{16}	[1,2]
c_3	2010	[0, 84]	{0,9,10,14}	{0,18,10,14}	[0,3]
c_3	2011	[0, 45]	{0,4}	{0,18}	[0,1]
c_3	2012	[0, 29]	{0,1,10}	{0,7,10}	[0,2]
c_3	2013	[0, 28]	{0,4,14}	{0,6,14}	[0,2]
c_4	2010	[70, 112]	{1}	{20}	[1,4]
c_4	2011	[26, 42]	{1}	{16}	[1,3]
c_4	2012	[61, 96]	{3,1}	{18}	[1,4]
c_4	2013	[52, 61]	{4,1}	{15}	[1,3]
c_5	2010	[21, 25]	{2}	{10}	[1,2]
c_5	2011	[45, 88]	{10,4}	{19,20}	[1,3]
c_5	2012	[37]	{1}	{17}	[1]
c_5	2013	[22, 61]	{3}	{11,20}	[1,2]

Tabla 6.3: Respuestas aproximadas a la consulta de agregación SQL 6.1 sobre la dimensión extendida de la reparación compatible $\mathcal{X}_e$ y la dimensión Tiempo

(Confederación, Año)	$\mathcal{Q}_{\texttt{sum}}$		$\mathcal{Q}_{\texttt{count}}$	
	Dimensión Inconsistente	Respuesta Aproximada	Dimensión Inconsistente	Respuesta Aproximada
(c_1, 2010)	122	[0,84]	11	[0,3]
(c_1, 2011)	77	[0,45]	8	[0,1]
(c_1, 2012)	96	[0,29]	9	[0,2]
(c_1, 2013)	66	[0,28]	9	[0,2]
(c_2, 2010)	56	[36,54]	8	[1,3]
(c_2, 2011)	85	[46,50]	8	[1,2]
(c_2, 2012)	109	[63]	8	[1]
(c_2, 2013)	80	[73,87]	9	[1,2]
(c_3, 2010)	83	[0,84]	7	[0,3]
(c_3, 2011)	53	[0,45]	6	[0,1]
(c_3, 2012)	52	[0,29]	6	[0,2]
(c_3, 2013)	68	[0,28]	7	[0,2]
(c_4, 2010)	115	[70,112]	10	[1,4]
(c_4, 2011)	60	[26,42]	7	[1,3]
(c_4, 2012)	105	[61,96]	10	[1,4]
(c_4, 2013)	73	[52,61]	8	[1,3]
(c_5, 2010)	96	[21,25]	9	[1,2]
(c_5, 2011)	110	[45,88]	10	[1,3]
(c_5, 2012)	82	[37]	9	[1]
(c_5, 2013)	131	[22,61]	10	[1,2]

Tabla 6.4: Comparación de Respuestas a SQL 6.1 para el operador `sum` y `count`

problema es que este sistema, no es capaz de computar las reparaciones minimales de dimensiones que posean muchos elementos en sus categorías, es por este motivo que no ha sido posible comparar la respuesta aproximada de la dimensión $\mathcal{X}_e$ con la respuesta consistente, puesto que

(Confederación, Año)	$\mathcal{Q}_{\texttt{min}}$		$\mathcal{Q}_{\texttt{max}}$	
	Dimensión Inconsistente	**Respuesta Aproximada**	**Dimensión Inconsistente**	**Respuesta Aproximada**
(c_1, 2010)	7	{0,9,10,14}	19	{0,18,10,14}
(c_1, 2011)	1	{0,4}	16	{0,18}
(c_1, 2012)	1	{0,1,10}	19	{0,7,10}
(c_1, 2013)	1	{0,4,14}	16	{0,6,14}
(c_2, 2010)	1	{6,4,1}	15	{15}
(c_2, 2011)	4	{3}	18	{16}
(c_2, 2012)	8	{2}	19	{19}
(c_2, 2013)	1	{8,1}	16	{16}
(c_3, 2010)	6	{0,9,10,14}	19	{0,18,10,14}
(c_3, 2011)	3	{0,4}	18	{0,18}
(c_3, 2012)	1	{0,1,10}	19	{0,7,10}
(c_3, 2013)	4	{0,4,14}	15	{0,6,14}
(c_4, 2010)	1	{1}	20	{20}
(c_4, 2011)	1	{1}	19	{16}
(c_4, 2012)	1	{3,1}	18	{18}
(c_4, 2013)	1	{4,1}	20	{15}
(c_5, 2010)	2	{2}	20	{10}
(c_5, 2011)	1	{10,4}	20	{19,20}
(c_5, 2012)	1	{1}	17	{17}
(c_5, 2013)	3	{3}	20	{11,20}

Tabla 6.5: Comparación de Respuestas a SQL 6.1 para el operador `min` y `max`

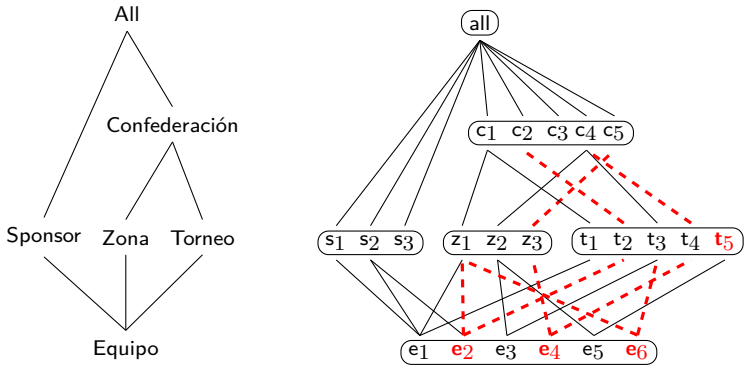
DLV para la dimensión $\mathcal{D}_e$ no entrega respuesta. Para poder comparar la respuesta aproximada con la respuesta consistente, se presentan otras dimensiones con una menor cantidad de elementos en sus categorías, para que sea posible obtener sus reparaciones minimales.

Sea el esquema de jerarquía de la Figura 6.3(a) y la dimensión $\mathcal{D}_{\texttt{FT}}$ Equipos de Fútbol de la Figura 6.3(b). Esta dimensión posee las categorías Equipo que rollup con Zona y ésta a su vez con Confederación. También, la categoría Equipo rollup Torneo y ésta a su vez con Confederación, además la categoría Equipo rollup con Sponsor (Patrocinadores). La categoría más alta es All a la cual llega Confederación y Sponsor. La dimensión $\mathcal{D}_{\texttt{FT}}$ se organiza como se aprecia en la Figura 6.3(b), donde los rollup de los elementos que participan de las inconsistencias se presentan con lineas punteadas.

Las restricciones de integridad que debe cumplir la dimensión $\mathcal{D}_{\texttt{FT}}$ vienen dadas por el conjunto Σ de RI que se presentan en la Tabla 6.1, sin embargo, la dimensión $\mathcal{D}_{\texttt{FT}}$ es inconsistente respecto a la restricción de integridad homogéneas Torneo⇒Confederación por el elemento t_5 y también, viola la restricción de integridad estricta Equipo→Confederación por los elementos e_2, e_4 y e_6. Las respuesta obtenidas desde esta dimensión inconsistente, se presentan en la Tabla 6.6 (ver Tabla de hechos A.10).

Si se computa esta dimensión por medio de DLV para obtener sus reparaciones minimales, se obtienen 17 reparaciones minimales con 7 cambios cada una (4 inserciones y 3 eliminaciones de arcos). Aplicando a las reparaciones minimales obtenidas, la consulta de agregación SQL 6.1 se obtienen las respuestas presentadas en la Tabla 6.7, donde solo los equipos c_1 y c_4 existen en todas las reparaciones minimales.

Si se computa ahora la dimensión $\mathcal{D}_{\texttt{FT}}$ en el Algoritmo 1 se obtiene la reparación compatible



(a) Esquema de Jerarquía (b) Dimensión Inconsistente $\mathcal{D}_{\mathtt{FT}}$

Figura 6.3: Esquema de Jerarquía y Dimensión $\mathcal{D}_{\mathtt{FT}}$ inconsistente

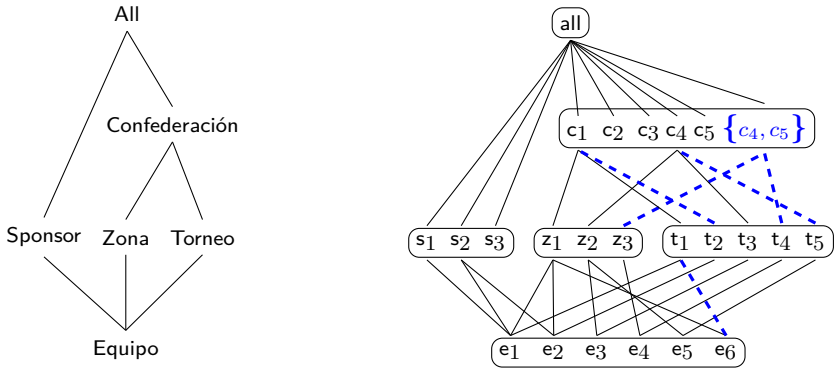
Confederación	**Año**	`sum`	`count`	`min`	`max`
c_1	2011	208	3	42	84
c_1	2012	199	3	25	92
c_2	2011	82	1	82	82
c_2	2012	25	1	25	25
c_4	2011	323	4	65	100
c_4	2012	203	4	27	92
c_5	2011	74	1	74	74
c_5	2012	27	1	27	27

Tabla 6.6: Respuestas a SQL 6.1 para la dimensión $\mathcal{D}_{\mathtt{FT}}$

Confederación	**Año**	`sum`	`count`	`min`	`max`
c_1	2011	208	42	84	3
c_1	2011	124	42	82	2
c_1	2012	199	25	92	3
c_1	2012	107	25	82	2
c_4	2011	323	65	100	4
c_4	2011	165	65	100	2
c_4	2012	203	27	92	4
c_4	2012	84	30	54	2

Tabla 6.7: Respuestas a SQL 6.1 para las reparaciones minimales de la dimensión $\mathcal{D}_{\mathtt{FT}}$

$\mathcal{X}_{\mathtt{FT}}$ de la Figura 6.4(b), la cual es una dimensión extendida que es consistente respecto al conjunto Σ de RI presentados en la tabla 6.1, desde la cual se obtienen las respuestas presentadas en la Tabla 6.8.



(a) Esquema de Jerarquía (b) Reparación Compatible $\mathcal{X}_{\mathtt{FT}}$

Figura 6.4: Esquema de Jerarquía y Reparación Compatible $\mathcal{X}_{\mathtt{FT}}$

Confederción	**Año**	`sum`	`count`	`min`	`max`
c_1	2011	208	42	84	3
c_1	2012	199	25	92	3
c_4	2011	165	65	100	2
c_4	2012	84	30	54	2
$\{c_4, c_5\}$	2011	74	74	74	1
$\{c_4, c_5\}$	2012	27	27	27	1

Tabla 6.8: Respuestas a SQL 6.1 para la dimensión $\mathcal{X}_{\mathtt{FT}}$

Luego si comparamos las 3 respuestas obtenidas, para los operadores de agregación `sum` y `count` (ver Tabla 6.9), se puede apreciar que la respuesta aproximada está contenida en la respuesta consistente (para los casos en que exististe valor seguro para la tupla t), en los casos que no hay valor seguro, en la respuesta consistente sus arcos fueron asignados a otras confederaciones. En este caso, la respuesta aproximada es una buena respuesta, puesto que esta contenida en la respuesta consistente, pero, casos como este no siempre se dan, en particular, mientras menor sea la cantidad de elementos en las categorías superiores, en este caso Confederación, mayor es la probabilidad de que la respuesta consistente este contenida o sea igual a la respuesta aproximada, pero en cambio, mientras más elementos existan en las categorías, mayor es la probabilidad de que no se generen evidencias, lo que provocaría la generación de elementos conjuntos, por lo que en casos como este la respuesta aproximada puede ser totalmente distinta a la respuesta consistente.

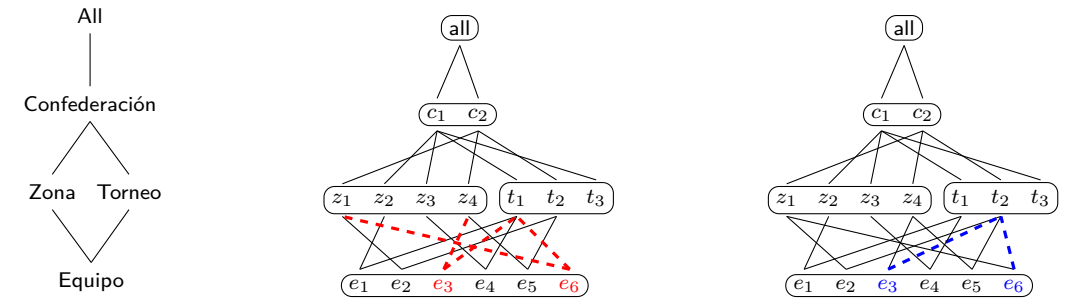
No comparamos la calidad de la respuesta con la respuesta obtenida de la dimensión inconsistente, puesto que ya se explico anteriormente que esta respuesta es mala porque existe un doble

$\mathcal{Q}_{\texttt{sum}}$ (Conf., Año)	Dimensión Inconsistente	Respuesta Consistente	Respuesta Aproximada
(c_1, 2011)	208	[124,208]	[208]
(c_1, 2012)	199	[107,199]	[199]
(c_2, 2011)	82	–	–
(c_2, 2012)	25	–	–
(c_4, 2011)	323	[165,323]	[165,239]
(c_4, 2012)	203	[84,203]	[84,111]
(c_5, 2011)	74	–	[0,74]
(c_5, 2012)	27	–	[0,27]

$\mathcal{Q}_{\texttt{count}}$ (Conf., Año)	Dimensión Inconsistente	Respuesta Consistente	Respuesta Aproximada
(c_1, 2011)	3	[2,3]	[3]
(c_1, 2012)	3	[2,3]	[3]
(c_2, 2011)	1	–	–
(c_2, 2012)	1	–	–
(c_4, 2011)	4	[2,4]	[2,3]
(c_4, 2012)	4	[2,4]	[2,3]
(c_5, 2011)	1	–	[0,1]
(c_5, 2012)	1	–	[0,1]

Tabla 6.9: Comparación de respuestas a la consulta SQL 6.1 para los operadores `sum` y `count`

conteo de los datos. Algo similar sucede con las respuestas obtenidas para los para los operadores de agregación `min` y `max` (ver Tabla 6.10), en este caso la respuesta aproximada es un conjunto de valores, lo que permite responder de forma más precisa que la respuesta consistente, la cual es un rango de valores, ya que en la respuesta aproximada, están las respuestas para los valores que se tienen certeza de que son respuesta y los que posiblemente lo son para una tupla t, de todas formas el que la respuesta aproximada y la respuesta consistente sean similares depende del tamaño de la dimensión.



(a) Esquema de Jerarquía Equipos de Fútbol

(b) Dimensión Inconsistente $\mathcal{D}_{e_1}$

(c) Reparación Compatible $\mathcal{X}_{e_1}$

Figura 6.5: Esquema, Dimensión inconsistente $\mathcal{D}_{e_1}$ y Reparación Compatible $\mathcal{X}_{e_1}$

$\mathcal{Q}_{\texttt{min}}$ (Conf., Año)	Dimensión Inconsistente	Respuesta Consistente	Respuesta Aproximada
(c_1, 2011)	42	[42]	{42}
(c_1, 2012)	25	[25]	{25}
(c_2, 2011)	82	–	–
(c_2, 2012)	25	–	–
(c_4, 2011)	65	[65]	{65}
(c_4, 2012)	27	[27,30]	{30,27}
(c_5, 2011)	74	–	{0,74}
(c_5, 2012)	27	–	{0,27}

$\mathcal{Q}_{\texttt{max}}$ (Conf., Año)	Dimensión Inconsistente	Respuesta Consistente	Respuesta Aproximada
(c_1, 2011)	84	[82,84]	{84}
(c_1, 2012)	92	[82,92]	{92}
(c_2, 2011)	82	–	–
(c_2, 2012)	25	–	–
(c_4, 2011)	100	[100]	{100}
(c_4, 2012)	92	[54,92]	{54}
(c_5, 2011)	74	–	{0,74}
(c_5, 2012)	27	–	{0,27}

Tabla 6.10: Comparación de respuestas a la consulta SQL 6.1 para los operadores `min` y `max`

Para la dimensión inconsistente respecto a la restricción de integridad estricta Equipo → Confederación $\mathcal{D}_{e_1}$ de la Figura 6.5(b) existen 9 Reparaciones minimales que involucran 2 inserciones de arcos y 2 eliminaciones para devolver la consistencia a la dimensión $\mathcal{D}_{e_1}$. Además, computando la misma dimensión en el Algoritmo propuesto para obtener la reparación compatible, se obtiene la dimensión de la Figura 6.5(c)[IV].

En este caso se obtienen las respuestas de las Tablas 6.11 y 6.12, donde se obtiene que la respuesta aproximada para cada uno de los operadores de agregación están completamente contenida en la respuesta consistente, esto sucede principalmente porque la cantidad de elementos en la categoría nivel de convergencia es muy reducida, lo que obliga a que no existan muchas opciones para hacer rollup y devolver la consistencia a la dimensión. En el caso de la reparación compatible, para la dimensión $\mathcal{D}_{e_1}$ los elementos que participan de esta inconsistencia poseen evidencias por lo que solo se realizan cambios de arcos y no se genera elemento conjunto.

A pesar de los resultados obtenidos, podemos decir que la respuesta aproximada obtenida de la reparación es una buena alternativa a la respuesta consistente, ya que nos permite poder responder a la incertidumbre, ya que mantiene la información consistente de la dimensión original, en base a la evidencia dada por estos elementos, la reparación compatible genera los cambios de arcos, y solo en casos donde no exista evidencia se genera el elemento conjunto, pero es muy probable que durante la reparación de inconsistencias, elementos que no posean evidencia, se les asigne una durante el proceso, lo que de todas formas disminuye la cantidad de elementos conjuntos que la reparación compatible genera. Por otro lado, también, la respuesta aproximada es mucho más factible de obtener, puesto que computar las reparaciones minimales

[IV] Detalle de las Tablas de Hechos y tablas de respuestas, ver Anexo A.11

$\mathcal{Q}_{\texttt{sum}}$ (Conf., Año)	Dimensión Inconsistente	Respuesta Consistente	Respuesta Aproximada
(c_1, 2011)	218	[113,218]	[113]
(c_1, 2012)	140	[83,140]	[83]
(c_2, 2011)	235	[130,235]	[235]
(c_2, 2012)	110	[53,110]	[110]
$\mathcal{Q}_{\texttt{count}}$ (Conf., Año)	Dimensión Inconsistente	Respuesta Consistente	Respuesta Aproximada
(c_1, 2011)	4	[2,4]	[2]
(c_1, 2012)	4	[2,4]	[2]
(c_2, 2011)	4	[2,4]	[4]
(c_2, 2012)	4	[2,4]	[4]

Tabla 6.11: Comparación de respuestas a la consulta SQL 6.1 para los operadores `sum` y `count`

$\mathcal{Q}_{\texttt{min}}$ (Conf., Año)	Dimensión Inconsistente	Respuesta Consistente	Respuesta Aproximada
(c_1, 2011)	27	[27,31]	{31}
(c_1, 2012)	6	[6]	{6}
(c_2, 2011)	27	[27,40]	{27}
(c_2, 2012)	9	[9]	{9}
$\mathcal{Q}_{\texttt{max}}$ (Conf., Año)	Dimensión Inconsistente	Respuesta Consistente	Respuesta Aproximada
(c_1, 2011)	82	[82]	{82}
(c_1, 2012)	77	[77]	{77}
(c_2, 2011)	90	[90]	{90}
(c_2, 2012)	44	[44]	{44}

Tabla 6.12: Comparación de respuestas a la consulta SQL 6.1 para los operadores `min` y `max`

de una dimensión inconsistente respecto a sus RI estrictas y homogéneas posee una complejidad computacional de NP-Complejo, con lo cual calcular estas reparaciones en dimensiones pequeñas es complicado y es casi imposible en dimensiones reales.

A continuación se realizarán pruebas para comprobar el comportamiento de la reparación compatible en dimensiones reales, las cuales poseen millones de tuplas.

6.3. Ejecución del Algoritmo 1 en un Caso Real

Una implementación del Algoritmo 1 ha sido realizada en lenguaje de programación `C`, con el cual se han obtenido las dimensiones compatibles presentadas hasta ahora, esta implementación es una primera versión del algoritmo propuesto, para el cual se ha estudiado su comportamiento en un caso real tomado del caso de los Teléfonos de Chile, una dimensión propuesta originalmente en (**?**), para computar la dimensión canónica. Tomando esta misma dimensión, se estudiará el tiempo que demora en generar la reparación compatible en distintos porcentajes de inconsistencia.

La dimensión Teléfonos esta basada en el país de Chile, el cual esta divido políticamente en comunas que en la actualidad ascienden a 346. Cada comuna tiene un cierto número de teléfonos

(de acuerdo a su población), los cuales están asociados con un código de área. Ese código de área se relaciona con una única región, lo mismo sucede para una comuna. El esquema jerárquico que modela dicha situación es el que se presenta en la Figura 6.6. Cabe señalar que se incluye la cantidad de elementos que pertenecen a cada categoría en forma de cardinalidad (# número).

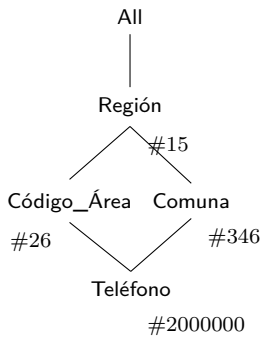


Figura 6.6: Esquema jerárquico de la dimensión Teléfonos y cantidad de elementos por categoría

Los resultados obtenidos para esta dimensión en cada caso son los que se muestran en la Tabla 6.13, además, para una mejor interpretación, se proporciona el gráfico de Tiempo (Figura 6.7) el cual da una idea del tiempo que tarda en realizar el algoritmo la reparación compatible.

Basado en los resultados anteriores se puede mencionar la reparación compatible es obtenida con éxito en todos los casos, con el cual se obtiene una dimensión extendida que es consistente respecto a sus RI. Este es un algoritmo polinomial que genera una reparación compatible de una dimensión inconsistente respecto a un conjunto Σ de RI, a pesar de esto, el tiempo que tarde el algoritmo dependerá bastante de la maquina donde se este ejecutando, y es de asumir que una aplicación de reparación de un DW se realizará en un servidor o un equipo con capacidad suficiente para realizar este proceso. El tiempo que tarde el algoritmo en obtener una reparación para un elemento consistente depende de la cantidad de evidencias distintas que existan, puesto que muchas evidencias implica realizar un cambio de arcos a niveles más cerca de la categoría inferior (categoría Equipo en la dimensión $\mathcal{D}_e$) por lo que ese cambio no afectará a muchos elementos inconsistentes, de hecho si se hace explícitamente en la categoría inferior, solo afectará al elemento inconsistente y a ningún otro, es por esto que cuando hay muchas evidencias para un elemento que participa de la inconsistencia el tiempo de cómputo aumenta.

Equipo de pruebas y obtención de datos

El equipo donde se realizaron las pruebas y computaron los distintos ejemplos en esta Tesis, posee las siguientes características:

Hardware

- Procesador Intel® Core™ i5-2430M CPU @ 2.40GHz × 4
- Memoria 7,7 GiB

Inconsistencia (%)	Elementos Inconsistentes	Tiempo (Min)
0.03	500	15.29s
0.61	12120	4m11.687s
1.59	31720	4m10.345s
2.35	46940	18m4.076s
5.79	115420	125m49.978s

Tabla 6.13: Resultados de aplicar la reparación compatible a la dimensión Teléfonos en distintos porcentajes de inconsistencia

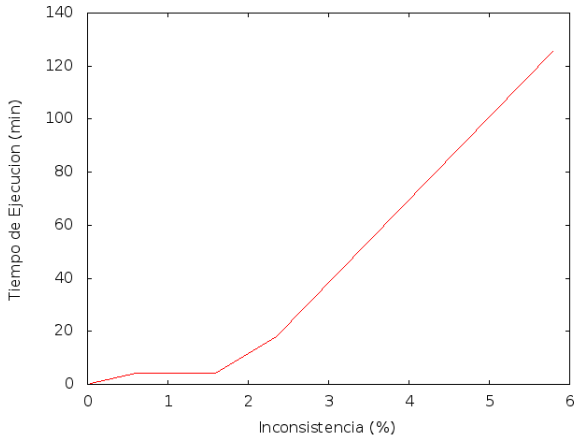


Figura 6.7: Gráficos obtenidos para computar la reparación compatible en la dimensión Teléfonos

- Intercambio 41 GiB

Sistema

- Debian Versión 7.1 (wheezy) de 64-bit
- Núcleo Linux 3.2.0-4-amd64
- GNOME 3.4.2

Software

- **PostgresSQL** versión 9.1.9-1 y **pgadminIII** versión 1.14.2-2 para manejo de Bases de Datos
- **gcc** versión 4.7.2-5 y **Geany** versión 1.22 para programación en C
- **DLV** versión x86-64-linux-elf-unixodbc (2012-12-17).

Para calcular el tiempo de ejecución del programa se a utilizado el comando `time` de Linux, este comando permite obtener el tiempo total de ejecución de un programa. Su sintaxis es muy sencilla: `time` seguido del comando cuya ejecución queremos medir (con los parámetros correspondientes). El resultado de `time` se escribe en la salida de error estándar (en C `stderr`, en C++ `cerr`, y en línea de comandos).

Capítulo 7

Conclusión

Esta tesis se originó con el objetivo principal de definir e implementar una nueva dimensión que aislara las inconsistencias de una dimensión respecto de restricciones estrictas y homogéneas y pudiera se utilizada para responder correctamente o de forma aproximada a las consultas de agregación sin que dependa del cómputo de las reparaciones minimales. En este sentido, se ha definido una nueva dimensión que llamamos *dimensión extendida* (ver Sección 4.2), la cual es una nueva dimensión que permite insertar conjuntos de elementos en sus categorías y sin modificar su esquema de jerarquía. Esta nueva dimensión toma como base la definición de dimensión canónica propuesta en (**?**). En este sentido, encontramos que este es un aporte relevante de esta tesis, ya que en (**?**) no se entregó una definición formal de la dimensión canónica y en particular una definición de dimensión que permita insertar elementos conjuntos en sus categorías. Además la dimensión extendidas es de interés por sí misma, ya que se esta nueva dimensión extendida nos permite expresar la imprecisión de los datos. La incertidumbre y la imprecisión se han analizado en el contexto de los modelos multidimensionales en (**??**). En particular, en (**?**) la imprecisión se encuentra en los valores de las dimensiones, y la incertidumbre en los valores de las medidas en las tablas de hechos.

Para definir correctamente la dimensión extendida y sus propiedades, fue necesario adaptar las definiciones originales dimensión, relaciones rollup, restricciones de integridad y entre otros, con el objeto de incorporar elementos del tipo conjunto en las categorías (ver Sección 4.1).

Para poder expresar la imprecisión de los datos, se han definido las respuestas aproximadas a consultas de agregación que utilicen los principales operadores de agregación SUM, COUNT, MAX y MIN (ver Sección 4.2.1). En este sentido se ha definido la respuesta aproximada para los operadores SUM y COUNT, donde, para una tupla, su valor mínimo viene dado por la suma de las respuestas seguras para esa tupla y su valor máximo viene dado para todas las respuestas donde exista esta tupla. De forma, similar se plantea la respuesta aproximada para los operadores MIN y MAX, donde su valor mínimo o máximo son los valores seguros para la tupla t y el resto de valores, son todos los valores donde exista la tupla que sean menores (MIN) o mayores (MAX) que el valor mínimo o máximo que se esta seguro. para todos estos operadores, de no existir valor seguro, su valor es cero.

Una vez definida esta nueva dimensión extendida, se procedió a definir la *reparación compati-*

ble (ver Capítulo 5), esta definición se llevó a cabo por medio de la implementación de algoritmos polinomiales. Esta reparación compatible, permite obtener una dimensión extendida que es consistente respecto a sus RI estrictas y homogéneas, la que en su definición y cómputo no apela a la definición de las reparaciones minimales de una dimensión inconsistente respecto de sus RI.

La reparación compatible es una dimensión extendida obtenida a partir de la dimensión incompatible que satisface las restricciones estrictas y homogéneas. Esta reparación permite hacer frente a las ambigüedades que surgen de las inconsistencias y puede proporcionar los rangos de las respuestas a las consultas de agregación con funciones como `SUM` y `COUNT` y grupos de valores para los operadores `MIN` y `MAX`. La definición de la dimensión extendida se inspira en el trabajo propuesto en (**?**), donde restricciones estrictas se restauran mediante la inserción de nuevos elementos en categorías artificiales. Aquí creamos elementos compuestos para restaurar la restricción de integridad estricta, pero solo como última opción, tampoco creamos nuevas categorías en las dimensiones y tampoco se altera el esquema jerárquico como se propone en (**?**). La dimensión extendida se diferencia de la reparación canónica presentada en (**?**), en que es una nueva dimensión que si bien no representa todas las reparaciones mínimales de una dimensión inconsistente, pero se puede obtener en tiempo polinómico (ver Sección 6.3).

Se demostró que para una dimensión $\mathcal{D}$ inconsistente con respecto a restricciones estrictas y homogéneas siempre existe una reparación compatible (Capítulo 4).

El problema de reparar bases de datos relacionales con respecto a un conjunto de restricciones de integridad se ha estudiado ampliamente (**?**). En (**?**) se demostró que a pesar de que hay varias maneras de representar un DW utilizando modelos relacionales (por ejemplo, esquema en estrella o copo de nieve esquema (**?**)) no es posible utilizar las técnicas de reparación relacionales para calcular reparaciones de DW.

Las respuestas aproximadas obtenidas de la reparación compatible (ver Capítulo 6), siempre son mejores que la respuesta obtenida de una dimensión inconsistente, puesto que estas dimensiones generan un doble conteo al realizar una consulta pre-computada, cosa que la reparación compatible evita en gran manera. Por otro lado, obtener las reparaciones minimales para comparar la respuesta aproximada con la respuesta consistente (**?**) requiere de un trabajo arduo, puesto que no es posible obtener las reparaciones minimales en dimensiones de DW reales, por lo que las pruebas implementadas en esta tesis se han realizado en dimensiones pequeñas, lo cual no ha permitido encontrar un patrón que determine la calidad de la respuesta aproximada en relación con la consistente, pero se puede decir que el proceso de obtención de las reparaciones minimales no mantiene la información que originalmente era consistente, lo que, provoca que las reparaciones minimales no mantengan la semántica original de los datos consistentes, esto porque el proceso de obtener la reparación minimal se preocupa de obtener una reparación con el mínimo número de cambios sin importar la semántica. En cambio la reparación compatible no modifica estos rollups, lo que permite mantener la semántica, pero provoca que la respuesta aproximada no sea siempre igual o este contenida en la respuesta consistente.

Además, la idea detrás de los algoritmos implementados en esta tesis es aislar la consistencia de los elementos involucrados en inconsistencias en base a la evidencia que entregan los elementos consistentes. El algoritmo nunca altera información que es consistente en la dimensión original, principalmente el algoritmo repara en base a evidencias y solo cuando no existen tales evidencias

se generan elementos conjuntos. Por lo tanto, en este sentido podríamos decir que una gran parte de la dimensión original está contenida en la reparación compatible, lo cual permite mantener la semántica de estos elementos a diferencia de métodos como las reparaciones minimales, las cuales buscan obtener una reparación con el mínimo número de cambios pero sin importarles la semántica, por lo que nuestro método además de reparar, mantiene la semántica de los datos consistentes en la reparación. En el peor de los caso, si la dimensión es totalmente inconsistente, al principio se generarán elementos conjuntos, pero es muy probable que luego de vayan dando evidencias, lo que la respuesta aproximada obtenida de la reparación compatible, seguirá siendo mejor que la respuesta obtenida de la dimensión inconsistente, de todas formas en el mundo real, nunca se dan estos casos.

Por lo anterior, La reparación compatible puede ser considerada en la categoría de limpieza de bases de datos, basado en (**???**). Así mismo, la reparación compatible podría ser de gran valor para el administrador del DW ya que al ser consistente puede ser utilizada para realizar consultas de agregación.

Los algoritmos implementados en esta tesis, permite obtener la reparación compatible en un número finito de pasos, determinado principalmente por el número de elementos inconsistentes de la dimensión original. La complejidad del algoritmo se presentó en el Capítulo 5. Estos algoritmos se desarrollaron en lenguaje `C` (Sección 5.2) y el DW de prueba se almacenó en una base de datos PostgreSQL.

Queda finalmente como trabajo futuro mejorar el proceso de generación de la reparación compatible, definir la respuesta aproximada para consultas de agregación para los operadores de agregación MIN y MAX, implementando alternativas de restauración de consistencia, principalmente mejorando la elección de elementos a los cuales insertar un rollup y por último, optimizar el algoritmo utilizando técnicas algorítmicas que permitan optimizar el consumo de memoria.

Referencias

Foto Afrati y Phokion Kolaitis. Repair checking in inconsistent databases: algorithms and complexity. En *Proceedings of the 12th International Conference on Database Theory*, ICDT '09, págs. 31–41. ACM, 2009.

Mario Alviano, Wolfgang Faber, Nicola Leone, Simona Perri, Gerald Pfeifer, y Giorgio Terracina. *The Disjunctive Datalog System DLV*, tomo 6702 de *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2011.

Marcelo Arenas y Leopoldo Bertossi. Answer Sets for Consistent Query Answering in Inconsistent Databases. *Theory and Practice of Logic Programming*, 3(4):393 – 424, 2003.

Marcelo Arenas, Leopoldo Bertossi, y Jan Chomicki. Consistent query answers in inconsistent databases. En *Proceedings of the eighteenth ACM symposium on Principles of database systems PODS'99*, págs. 68–79. ACM Press, 1999.

Marcelo Arenas, Leopoldo Bertossi, Jan Chomicki, Xin He, Vijay Raghavan, y Jeremy Spinrad. Scalar aggregation in inconsistent databases. *Theor. Comput. Sci.*, 296(3):405–434, 2003. ISSN 0304-3975.

Sina Ariyan y Leopoldo Bertossi. Structural repairs of multidimensional databases. En Pablo Barceló y Val Tannen, eds., *AMW*, tomo 749 de *CEUR Workshop Proceedings*. CEUR-WS.org, 2011.

Leopoldo Bertossi. Consistent query answering in databases. *ACM SIGMOD Record*, 35(2):68, 2006.

Leopoldo Bertossi y Loreto Bravo. Generic and Declarative Approaches to Data Cleaning: Some Recent Developments. En S Sadiq, ed., *Handbook of Data Quality - Research and Practice*. Springer-Verlag Berlin Heidelberg, 2013.

Leopoldo Bertossi, Loreto Bravo, y Mónica Caniupán. Consistent query answering in data warehouses. En Marcelo Arenas y Leopoldo Bertossi, eds., *AMW*, tomo 450 de *CEUR Workshop Proceedings*. CEUR-WS.org, 2009.

Leopoldo Bertossi y Jan Chomicki. Query answering in inconsistent databases. En Jan Chomicki, Ron van der Meyden, y Gunter Saake, eds., *Logics for Emerging Applications of Databases*, págs. 43–83. Springer, 2003.

Loreto Bravo y Leopoldo Bertossi. Consistent query answering under inclusion dependencies. En *Proceedings of the 2004 conference of the Centre for Advanced Studies on Collaborative research*, CASCON '04, págs. 202–216. IBM Press, 2004.

Loreto Bravo, Mónica Caniupán, y Carlos Hurtado. Logic programs for repairing inconsistent dimensions in data warehouses. En Alberto H. F. Laender y Laks V. S. Lakshmanan, eds., *AMW*, tomo 619 de *CEUR Workshop Proceedings*. CEUR-WS.org, 2010.

Doug Burdick, Prasad Deshpande, TS Jayram, Raghu Ramakrishnan, y Shivakumar Vaithyanathan. OLAP over uncertain and imprecise data. *The VLDB Journal*, 16(1):123–144, 2007.

Mónica Caniupán, Loreto Bravo, y Carlos Hurtado. Repairing inconsistent dimensions in data warehouses. *Data & Knowledge Engineering*, 79-80:17–39, 2012.

Surajit Chaudhuri y Umeshwar Dayal. An overview of data warehousing and OLAP technology. *ACM SIGMOD Record*, 26(1):65–74, 1997.

Jan Chomicki. Consistent query answering: The first ten years. En *Proceedings of the 2nd international conference on Scalable Uncertainty Management*, SUM '08, págs. 1–3. Springer-Verlag, 2008.

Tina Dell Armi, Wolfgang Faber, Giuseppe Ielpa, Nicola Leone, y Gerald Pfeifer. Aggregate Functions in DLV. En *Answer Set Programming: Advances in Theory and Implementation, volume 78 of CEUR Workshop proceedings, 274–288. Online: CEUR-WS.org/Vol78*, págs. 274–288. 2003.

Wenfei Fan. Extending Dependencies with Conditions for Data Cleaning. En *The 8th IEEE International Conference on Computer and Information Technology*, págs. 185–190. 2008.

Wenfei Fan, Floris Fan, Xibei Jia, y Anastasios Kementsietsidis. Conditional Functional Dependencies for Capturing Data Inconsistencies. *ACM Trans. Database Syst.*, 33(2):6:1—-6:48, 2008.

Carlos Hurtado y Claudio Gutierrez. *Data Warehouses and OLAP: Concepts, Architectures and Solutions*, cap. Handling Structural Heterogeneity in OLAP. Idea Group, Inc, 2007.

Carlos Hurtado, Claudio Gutierrez, y Alberto Mendelzon. Capturing summarizability with integrity constraints in OLAP. *ACM Transactions on Database Systems*, 30(3):854–886, 2005.

Carlos Hurtado y Alberto Mendelzon. Reasoning about Summarizability in Heterogeneous Multidimensional Schemas. En *Proceedings of the 8th International Conference on Database Theory*, págs. 375–389. Springer-Verlag, 2001.

Carlos Hurtado, Alberto Mendelzon, y Alejandro Vaisman. Maintaining Data Cubes under Dimension Updates. En *Proceedings of the 15th International Conference on Data Engineering*, ICDE '99, págs. 346—-. IEEE Computer Society, 1999a. ISBN 0-7695-0071-4.

Carlos Hurtado, Alberto Mendelzon, y Alejandro Vaisman. Updating OLAP dimensions. En *Proceedings of the 2nd ACM international workshop on Data warehousing and OLAP*, DOLAP '99, págs. 60–66. ACM, 1999b.

Hans-Joachim Lenz y Arie Shoshani. Summarizability in olap and statistical data bases. En Yannis Ioannidis y David Hansen, eds., *SSDBM*, págs. 132–143. IEEE Computer Society, 1997.

Nicola Leone, Gerald Pfeifer, Wolfgang Faber, Thomas Eiter, Georg Gottlob, Simona Perri, y Francesco Scarcello. The DLV system for knowledge representation and reasoning. *ACM Trans. Comput. Logic*, 7(3):499–562, 2006.

Carolin Letz, Eric Henn, y Gottfried Vossen. Consistency in Data Warehouse Dimensions. En *Proceedings of the 2002 International Symposium on Database Engineering & Applications*, IDEAS '02, págs. 224–232. IEEE Computer Society, 2002.

Andrei Lopatenko y Leopoldo Bertossi. Complexity of consistent query answering in databases under cardinality-based and incremental repair semantics. En *Proceedings of the 11th international conference on Database Theory*, ICDT'07, págs. 179–193. Springer-Verlag, 2006. ISBN 3-540-69269-X, 978-3-540-69269-0.

Elzbieta Malinowski y Esteban Zimányi. Hierarchies in a Multidimensional Model: from Conceptual Modeling to Logical Representation. *Data & Knowledge Engineering*, 59(2):348–377, 2006.

Torben Pedersen, Christian Jensen, y Curtis Dyreson. Extending Practical Pre-Aggregation in On-Line Analytical Processing. En *Proceedings of the 25th International Conference on Very Large Data Bases*, VLDB '99, págs. 663–674. Morgan Kaufmann Publishers Inc., 1999a.

Torben Pedersen, Christian Jensen, y Curtis Dyreson. Supporting imprecision in multidimensional databases using granularities. En *Proceedings of the 11th International Conference on Scientific and Statistical Database Management*, SSDBM '99, págs. 90–. IEEE Computer Society, 1999b.

Torben Pedersen, Christian Jensen, y Curtis Dyreson. A Foundation for Capturing and Querying Complex Multidimensional Data. *Inf. Syst.*, 26(5):383–423, 2001.

Maurizio Rafanelli y Arie Shoshani. STORM: A Statistical Object Representation Model. En *In Proc. of SSDBM*, págs. 14–29. 1990.

Alejandro Vaisman. *Updates, View Maintenance and Materialized Views in Multidimnensional Databases*. Tesis Doctoral, Universidad de Buenos Aires, 2001.

Winter Corporation. TopTen Program Survey of Users of Large Databases. 2005. URL `http://www.wintercorp.com/VLDB/2005_TopTen_Survey/TopTenProgram.html`.

Apéndice A

Anexos del Capítulo 6

A.1. Tabla para las dimensiones $\mathcal{D}_e$ y $\mathcal{X}_e$

Ingresos					
Equipo	**Fecha**	**Ingreso**	**Equipo**	**Fecha**	**Ingreso**
e_1	23 - 9 - {2010,2013}	13	e_1	19 - 1 - 2011	3
e_2	24 - 11 - {2010,2011}	4	e_2	19 - 6 - 2011	7
e_3	18 - 1 - {2010,2012}	17	e_3	1 - 1 - 2011	4
e_4	27 - 11 - 2010	1	e_4	25 - 9 - 2011	16
e_5	9 - 8 - 2010	11	e_5	16 - 6 - 2011	4
e_6	17 - 2 - {2010,2011}	4	e_6	11 - 10 - 2011	10
e_7	3 - 4 - 2010	9	e_7	22 - 2 - 2011	18
e_8	2 - 7 - 2010	9	e_8	14 - 5 - 2011	16
e_9	12 - 1 - 2010	7	e_9	11 - 3 - 2011	1
e_{10}	16 - 5 - 2010	2	e_{10}	9 - 1 - {2011,2013}	19
e_{11}	16 - 11 - {2010,2013}	1	e_{11}	26 - 2 - 2011	15
e_{12}	6 - 4 - 2010	10	e_{12}	24 - 5 - 2011	19
e_{13}	8 - 6 - 2010	20	e_{13}	5 - 7 - 2011	1
e_{14}	22 - 8 - 2010	11	e_{14}	2 - 5 - 2011	17
e_{15}	18 - 4 - 2010	13	e_{15}	25 - 4 - {2011,2012}	1
e_{16}	5 - 8 - 2010	18	e_{16}	6 - 7 - 2011	6
e_{17}	12 - 7 - 2010	10	e_{17}	17 - 7 - 2011	4
e_{18}	8 - 10 - 2010	19	e_{18}	21 - 6 - {2010,2011}	15
e_{19}	22 - 8 - 2010	15	e_{19}	8 - 6 - 2011	16
e_{20}	6 - 8 - 2010	6	e_{20}	1 - 9 - 2011	5
Equipo	**Fecha**	**Ingreso**	**Equipo**	**Fecha**	**Ingreso**
e_1	4 - 5 - 2012	2	e_1	7 - 10 - 2013	12
e_2	20 - 5 - 2012	19	e_2	10 - 6 - 2013	10
e_3	1 - 10 - 2012	3	e_3	17 - 6 - 2013	4
e_4	24 - 3 - {2012,2013}	8	e_4	15 - 7 - {2012,2013}	12
e_5	19 - 2 - 2012	4	e_5	11 - 2 - 2013	4
e_6	3 - 8 - 2012	17	e_6	7 - 3 - 2013	8
e_7	12 - 5 - {2010,2012}	10	e_7	19 - 6 - 2013	4
e_8	15 - 10 - 2012	6	e_8	26 - 6 - 2013	3
e_9	23 - 11 - 2012	15	e_9	10 - 7 - 2013	13
e_{10}	5 - 10 - 2012	12	e_{10}	20 - 10 - {2011,2013}	20
e_{11}	9 - 7 - 2012	13	e_{11}	4 - 10 - 2013	16
e_{12}	27 - 9 - 2012	1	e_{12}	19 - 2 - 2013	11
e_{13}	11 - 8 - 2012	14	e_{13}	7 - 10 - 2013	7
e_{14}	27 - 9 - 2012	7	e_{14}	6 - 10 - {2010,2013}	14
e_{15}	23 - 4 - 2012	11	e_{15}	14 - 4 - 2013	15
e_{16}	26 - 9 - 2012	1	e_{16}	10 - 2 - 2013	6
e_{17}	8 - 9 - 2012	18	e_{17}	24 - 10 - 2013	6
e_{18}	5 - 9 - {2010,2012}	9	e_{18}	18 - 2 - {2010,2013}	1
e_{19}	16 - 7 - 2012	19	e_{19}	18 - 7 - 2013	8
e_{20}	12 - 8 - 2012	10	e_{20}	24 - 4 - 2013	15

Tabla A.1: Tabla de Hechos Ingresos de la dimensión $\mathcal{D}_e$ Equipos de Fútbol

Confederación	Año	SUM	COUNT	MIX	MAX
c_1	{2010,2011}	15	1	15	15
c_1	{2010,2012}	26	2	9	17
c_1	{2010,2013}	2	2	1	1
c_1	2010	79	6	7	19
c_1	2011	62	7	1	16
c_1	2012	70	7	1	19
c_1	2013	64	7	4	16
c_2	{2010,2011}	4	1	4	4
c_2	{2010,2012}	10	1	10	10
c_2	{2010,2013}	1	1	1	1
c_2	{2012,2013}	20	2	8	12
c_2	2010	41	5	1	15
c_2	2011	81	7	4	18
c_2	2012	79	5	10	19
c_2	2013	59	6	4	16
c_3	{2010,2012}	19	1	19	19
c_3	{2010,2013}	27	2	13	14
c_3	2010	37	4	6	11
c_3	2011	53	6	3	18
c_3	2012	33	5	1	19
c_3	2013	41	5	4	15
c_4	{2010,2011}	15	1	15	15
c_4	{2010,2012}	26	2	9	17
c_4	{2010,2013}	1	1	1	1
c_4	{2011,2011}	1	1	1	1
c_4	{2012,2013}	20	2	8	12
c_4	2010	73	6	1	20
c_4	2011	44	5	1	19
c_4	2012	58	5	3	18
c_4	2013	52	5	4	20
c_5	{2010,2011}	4	1	4	4
c_5	{2010,2013}	27	2	13	14
c_5	{2011,2012}	1	1	1	1
c_5	{2011,2013}	39	2	19	20
c_5	2010	65	6	2	20
c_5	2011	66	6	1	19
c_5	2012	81	8	2	17
c_5	2013	65	6	3	20

Tabla A.2: Respuestas a la consulta SQL 6.1 agrupada por Confederación y Año para la Figura 6.1(b)

Confederación	**Año**	**SUM**	**COUNT**	**MIX**	**MAX**
c_1	2010	122	11	7	19
c_1	2011	77	8	1	16
c_1	2012	96	9	1	19
c_1	2013	66	9	1	16
c_2	2010	56	8	1	15
c_2	2011	85	8	4	18
c_2	2012	109	8	8	19
c_2	2013	80	9	1	16
c_3	2010	83	7	6	19
c_3	2011	53	6	3	18
c_3	2012	52	6	1	19
c_3	2013	68	7	4	15
c_4	2010	115	10	1	20
c_4	2011	60	7	1	19
c_4	2012	105	10	1	18
c_4	2013	73	8	1	20
c_5	2010	96	9	2	20
c_5	2011	110	10	1	20
c_5	2012	82	9	1	17
c_5	2013	131	10	3	20

Tabla A.3: Respuestas a la consulta SQL 6.1 agrupada por Confederación y Año para la Figura 6.1(b) obtenidas de $\mathcal{A}$

$\mathcal{R}_{\mathcal{D}_{Equipo}}^{Confederación}$	
Confederación	Equipo
c_1	e_9
c_2	e_2
c_5	e_6, e_{10}

Tabla A.4: Rollup entre Equipo y Confederación de la Figura 6.1(b) para los elementos consistentes

$\mathcal{R}_{\mathcal{D}_{Equipo}}^{Confederación}$	
Conferación	Equipo
c_1	$e_3, e_5, e_8, e_9, e_{11}, e_{16}, e_{18}, e_{19}$
c_2	$e_2, e_4, e_7, e_{11}, e_{17}, e_{19}, e_{20}$
c_3	$e_1, e_5, e_7, e_{14}, e_{16}, e_{20}$
c_4	$e_3, e_4, e_{12}, e_{13}, e_{15}, e_{17}, e_{18}$
c_5	$e_1, e_6, e_8, e_{10}, e_{12}, e_{13}, e_{14}, e_{15}$

Tabla A.5: Rollup entre Equipo y Confederación de la Figura 6.1(b)

$\mathcal{R}_{\mathcal{D}_{Equipo}}^{Confederación}$		$\mathcal{R}_{\mathcal{X}_{Equipo}}^{Confederación}$	
Conferación	Equipo	Conferación	Equipo
c_1	$e_3, e_5, e_8, e_9, e_{11}, e_{16}, e_{18}, e_{19}$	c_1	e_5, e_7, e_{14}, e_{16}
c_2	$e_2, e_4, e_7, e_{11}, e_{17}, e_{19}, e_{20}$	c_2	$e_1, e_2, e_{11}, e_{19}, e_{20}$
c_3	$e_1, e_5, e_7, e_{14}, e_{16}, e_{20}$	c_3	e_5, e_7, e_{14}, e_{16}
c_4	$e_3, e_4, e_{12}, e_{13}, e_{15}, e_{17}, e_{18}$	c_4	$e_3, e_4, e_9, e_{12}, e_{13}, e_{15}, e_{17}, e_{18}$
c_5	$e_1, e_6, e_8, e_{10}, e_{12}, e_{13}, e_{14}, e_{15}$	c_5	e_6, e_8, e_{10}

Tabla A.6: Diferencia entre la $\mathcal{D}_e$ y $\mathcal{X}_e$ en base a la restricción Equipo→Confederación

$\mathcal{R}_{\mathcal{X}_{Equipo}}^{Confederación}$	
Equipo	Confederación
c_1	e_5, e_7, e_{14}, e_{16}
c_2	$e_1, e_2, e_{11}, e_{19}, e_{20}$
c_3	e_5, e_7, e_{14}, e_{16}
c_4	$e_3, e_4, e_9, e_{12}, e_{13}, e_{15}, e_{17}, e_{18}$
c_5	e_6, e_8, e_{10}

Tabla A.7: Rollup entre Equipo y Confederación de la Figura 6.2(b)

A.2. Tabla para las dimensiones $\mathcal{D}_{\mathtt{FT}}$ y $\mathcal{X}_{\mathtt{FT}}$

Ingresos		
Equipo	**Fecha**	**Ingreso**
t_1	11 - 8 - 2011	42
t_1	5 - 11 - 2012	82
t_2	16 - 8 - 2012	25
t_2	24 - 8 - 2011	82
t_3	12 - 10 - 2012	54
t_3	14 - 4 - 2011	100
t_4	19 - 10 - 2012	27
t_4	20 - 3 - 2011	74
t_5	23 - 2 - 2012	30
t_5	25 - 10 - 2011	65
t_6	21 - 7 - 2011	84
t_6	27 - 10 - 2012	92

Tabla A.8: Tabla de Hechos Ingresos de la Figura 6.3

(Conf., Año)	SUM	COUNT	MIN	MAX
(c_1,2011)	[124,208]	[2,3]	[42]	[82,84]
(c_1,2012)	[107,199]	[2,3]	[25]	[82,92]
(c_4,2011)	[165,323]	[2,4]	[65]	[100]
(c_4,2012)	[84,203]	[2,4]	[27,30]	[54,92]

Tabla A.9: Respuesta consistente para $\mathcal{D}_{\texttt{FT}}$

(Conf., Año)	SUM	COUNT	MIN	MAX
(c_1, 2011)	208	3	42	84
(c_1, 2012)	199	3	25	92
(c_4, 2011)	[165,239]	[2,3]	[65;74]	[100;74]
(c_4, 2012)	[84,111]	[2,3]	30	[54;27]
(c_5, 2011)	[0,74]	[0,1]	[0;74]	[0;74]
(c_5, 2012)	[0,27]	[0,1]	[0;27]	[0;27]

Tabla A.10: Respuestas Aproximadas para $\mathcal{X}_{\texttt{FT}}$

A.3. Tabla para las dimensiones $\mathcal{D}_{e_1}$ y $\mathcal{X}_{e_1}$

Ingresos		
Equipo	**Fecha**	**Ingreso**
e_1	19 - 4 - 2011	82
e_1	2 - 3 - 2012	77
e_2	13 - 10 - 2012	9
e_2	27 - 4 - 2011	40
e_3	22 - 8 - 2012	29
e_3	7 - 10 - 2011	27
e_4	23 - 11 - 2011	31
e_4	25 - 2 - 2012	6
e_5	4 - 4 - 2012	44
e_5	6 - 6 - 2011	90
e_6	27 - 6 - 2011	78
e_6	9 - 6 - 2012	28

Tabla A.11: Tabla de Hechos Ingresos de la Figura 6.5

(Conf., Año)	SUM	COUNT	MIN	MAX
(c_1, 2011)	218	27	82	4
(c_1, 2012)	140	6	77	4
(c_2, 2011)	235	27	90	4
(c_2, 2012)	110	9	44	4

Tabla A.12: Respuestas en la dimensión inconsistente $\mathcal{D}_{e_1}$

(Conf., Año)	SUM	COUNT	MIN	MAX
(c_1, 2011)	[113,218]	[27,31]	[82]	[2,4]
(c_1, 2012)	[83,140]	[6]	[77]	[2,4]
(c_2, 2011)	[130,235]	[27,40]	[90]	[2,4]
(c_2, 2012)	[53,110]	[9]	[44]	[2,4]

Tabla A.13: Respuesta consistente para $\mathcal{D}_{e_1}$

(Conf., Año)	SUM	COUNT	MIN	MAX
(c_1, 2011)	113	31	82	2
(c_1, 2012)	83	6	77	2
(c_2, 2011)	235	27	90	4
(c_2, 2012)	110	9	44	4

Tabla A.14: Respuesta Aproximada para $\mathcal{X}_{e_1}$