



UNIVERSIDAD DE MÁLAGA

Escuela de Ingenierías Industriales  
Departamento de Matemática Aplicada  
Programa de Doctorado en Ingeniería Mecánica y Eficiencia Energética

TESIS DOCTORAL

**Extracción de conocimiento usando  
atributos negativos en el  
Análisis de Conceptos Formales  
Aplicaciones en la Ingeniería**

D. José Manuel Rodríguez Jiménez


Director Dr. D. Pablo José Cordero Ortega

Málaga, 2017



UNIVERSIDAD  
DE MÁLAGA

AUTOR: José Manuel Rodríguez Jiménez

 <http://orcid.org/0000-0003-3776-9887>

EDITA: Publicaciones y Divulgación Científica. Universidad de Málaga



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-SinObraDerivada 4.0 Internacional:

<http://creativecommons.org/licenses/by-nc-nd/4.0/legalcode>

Cualquier parte de esta obra se puede reproducir sin autorización pero con el reconocimiento y atribución de los autores.

No se puede hacer uso comercial de la obra y no se puede alterar, transformar o hacer obras derivadas.

Esta Tesis Doctoral está depositada en el Repositorio Institucional de la Universidad de Málaga (RIUMA): [riuma.uma.es](http://riuma.uma.es)





UNIVERSIDAD  
DE MÁLAGA

Dr. Pablo José Cordero Ortega, Profesor Titular de Universidad del Departamento de Matemática Aplicada de la Universidad de Málaga

HACE CONSTAR:

Que D. José Manuel Rodríguez Jiménez, Licenciado en Ciencias (Matemáticas), ha realizado en el Departamento de Matemática Aplicada de la Universidad de Málaga, bajo mi dirección y tutela, el trabajo de investigación correspondiente a su Tesis Doctoral titulada:

**Extracción de conocimiento usando atributos negativos  
en el Análisis de Conceptos Formales. Aplicaciones en la  
Ingeniería**

Revisado el presente trabajo, estimo que puede ser presentado al Tribunal que ha de juzgarlo. Asimismo, informo que las publicaciones científicas que avalan esta tesis no han sido utilizadas como aval para tesis anteriores.

Y para que conste a efectos de lo establecido en el Real Decreto 99/2011, autorizo la presentación de este trabajo en la Universidad de Málaga.

Málaga, a 5 de mayo de 2017

Dr. Pablo José Cordero Ortega



UNIVERSIDAD  
DE MÁLAGA

*Para todos aquellos que han contribuido en el desarrollo de esta Tesis.*

*Hay una fuerza motriz más poderosa que el vapor, la electricidad  
y la energía atómica: la voluntad.*

Albert Einstein



UNIVERSIDAD  
DE MÁLAGA

# Agradecimientos

Al término de esta etapa de mi vida, quiero expresar un profundo agradecimiento a quienes con su ayuda, apoyo y comprensión me alentaron a lograr esta increíble realidad, en especial a todos los que han hecho posible en mayor o menor medida que este trabajo de investigación haya dado sus frutos.

Todo este proyecto de investigación no se podría haber llevado a cabo sin la inestimable ayuda y conocimientos del Dr. Pablo Cordero, tutor de esta Tesis, y de los Drs. Manuel Enciso y Ángel Mora, con los cuales he publicado los artículos aquí presentados.

No puedo dejar de agradecer a mi mujer, la Dra. Inmaculada de las Peñas, su comprensión, apoyo y ayuda en todo este largo proceso.

El recuerdo a la labor de mis padres, que propiciaron que tuviera la educación necesaria en mi juventud y que me enseñaron los valores que me han hecho ser persistente en esta tarea.

Este trabajo de investigación ha sido financiado por los proyectos TIN-2011-28084 y TIN-2014-59471-P del Ministerio de Ciencia en Innovación de España, cofinanciado por el Fondo Regional de Desarrollo Europeo (ERDF).

La base de datos sobre cáncer de mama usada en la Aplicación 7.1.1 ha sido cedida por el University Medical Centre, Institute of Oncology, Ljubljana, Eslovenia. Se agradece a los Drs. M. Zwitter y M. Soklic por



proporcionar dichos datos.

Quiero agradecer especialmente a la Dra. M. D. de las Peñas, oncóloga especialista en radioterapia en el Hospital Universitario Rey Juan Carlos, por su importante contribución, asesoramiento y comentarios al estudio de la Aplicación 7.1.1.

Quedan muchos nombres que harían interminable esta lista, pero todos los que saben que han estado ahí, igualmente saben que no me olvido de ellos ni de su apoyo.

# Índice general

<b>Publicaciones</b>	XIII
<b>1. Introducción</b>	<b>1</b>
<b>2. Preliminares</b>	<b>7</b>
2.1. Conjuntos parcialmente ordenados y retículos . . . . .	7
2.1.1. Retículos y retículos completos . . . . .	10
2.1.2. Subretículos y homomorfismos de retículos . . . . .	14
2.1.3. Retículos booleanos . . . . .	15
2.2. Operadores de clausura y conexiones de Galois . . . . .	16
2.3. Análisis de Conceptos Formales . . . . .	19
2.3.1. Retículo de conceptos formales . . . . .	21
2.3.2. Implicaciones de atributos . . . . .	25
2.3.3. Retículos de conceptos <i>versus</i> implicaciones . . . . .	33
2.3.4. Cálculo del retículo de conceptos y de las implicaciones	36
2.4. Reglas de asociación . . . . .	44
<b>3. Atributos negativos. Planteamiento del problema</b>	<b>47</b>
3.1. Antecedentes en Minería de Datos . . . . .	50



3.1.1.	Reglas de asociación . . . . .	50
3.1.2.	Generalizaciones (negación) . . . . .	51
3.2.	Antecedentes en el Análisis de Conceptos Formales . . . . .	56
3.2.1.	Negación de atributos y/o negación de conceptos . . . . .	57
3.2.2.	Negación de atributos en las implicaciones . . . . .	62
3.3.	Planteamiento de objetivos de la tesis . . . . .	66
<b>4.</b>	<b>Retículos de conceptos mixtos</b>	<b>69</b>
4.1.	Considerando atributos negativos en la conexión de Galois . . . . .	70
4.2.	Propiedades de los retículos de conceptos mixtos . . . . .	77
4.3.	Caracterizando los retículos de conceptos mixtos . . . . .	87
4.4.	La negación: una comparación con enfoques relacionados . . . . .	94
<b>5.</b>	<b>Sistemas implicacionales mixtos</b>	<b>99</b>
5.1.	<b>MixAtL</b> : una lógica para implicaciones entre atributos mixtos	103
5.1.1.	La semántica de <b>MixAtL</b> . . . . .	104
5.1.2.	Un sistema axiomático para <b>MixAtL</b> . . . . .	106
5.1.3.	Corrección y completitud de <b>MixAtL</b> . . . . .	112
5.2.	<b>SimMixAtL</b> : Lógica de simplificación para atributos mixtos	116
<b>6.</b>	<b>Extracción de conocimiento</b>	<b>123</b>
6.1.	Calculando sistemas implicacionales mixtos completos . . . . .	123
6.2.	Calculando el retículo de conceptos mixtos . . . . .	128
6.2.1.	Propuesta basada en el algoritmo Next-Closure . . . . .	132
6.2.2.	Propuesta basada en el algoritmo de Norris . . . . .	134
6.2.3.	Propuesta basada en el algoritmo FCbO. . . . .	135
6.2.4.	Propuesta basada en el algoritmo InClose . . . . .	138

6.2.5. Comparación de los algoritmos para generar retículos de conceptos mixtos. . . . .	140
6.3. La herramienta FuzAtAnalyzer . . . . .	143
<b>7. Aplicabilidad de los resultados obtenidos</b>	<b>149</b>
7.1. Aplicaciones . . . . .	151
7.1.1. Análisis de datos sobre cáncer de mama. . . . .	151
7.1.2. Optimización automatizada en el diseño de motores	157
7.1.3. Análisis de redes criminales . . . . .	160
7.1.4. Sistemas de recomendación . . . . .	166
<b>Conclusiones y trabajo futuro</b>	<b>173</b>
<b>Índice de definiciones</b>	<b>179</b>
<b>Bibliografía</b>	<b>183</b>



UNIVERSIDAD  
DE MÁLAGA

# Publicaciones

1. J. M. Rodríguez-Jiménez, P. Cordero, M. Enciso, y S. Rudolph. Concept lattices with negative information: A characterization theorem. *Information Sciences*, 369, pp. 51–62, 2016.

Factor de impacto en J.C.R. 2015: 3,360. Posición 6<sup>a</sup> de 144 (Q1) en la categoría ‘Computer Sciences, Information Systems’.

2. J. M. Rodríguez-Jiménez, P. Cordero, M. Enciso, y A. Mora. Data mining algorithms to compute mixed concepts with negative attributes: an application to breast cancer data analysis. *Mathematical Methods in the Applied Sciences*, 39 (16), pp. 4829–4845, 2016.

Factor de impacto en J.C.R. 2015: 1,002. Posición 90<sup>o</sup> de 254 (Q2) en la categoría ‘Mathematics, Applied’.

3. J. M. Rodríguez-Jiménez, P. Cordero, M. Enciso, y A. Mora. Analyzing criminal networks using Formal Concept Analysis with negative attributes. *International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE*, 2016.

4. J. M. Rodríguez-Jiménez, P. Cordero, M. Enciso, y A. Mora. Optimization in engine design via formal concept analysis using negative attributes. *The Fourth International Conference on Control, Modelling, Computing and Applications*. Zurich (Suiza), 2016.

5. J. M. Rodríguez-Jiménez, E. Rodríguez-Lorenzo, P. Cordero, M. Enciso, y A. Mora. A Normal Form for Fuzzy Functional Dependencies , *IEEE Symposium Series on Computational Intelligence, SSCI, Cape Town, South Africa, December 7–10, 2015*, pp. 984–989, 2015.
6. E. Rodríguez-Lorenzo, J. M. Rodríguez-Jiménez, P. Cordero, M. Enciso, y A. Mora. Fuzzy functional dependencies and direct bases. *Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA)*, 2015.
7. J. M. Rodríguez-Jiménez, P. Cordero, M. Enciso, y A. Mora. Extending FuzAtAnalyzer to approach the management of classical negation. *International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE*, 2015.
8. J. M. Rodríguez-Jiménez, P. Cordero, M. Enciso, y A. Mora. A Generalized Framework to Consider Positive and Negative Attributes in Formal Concept Analysis. *Proceedings of the Eleventh International Conference on Concept Lattices and Their Applications. Košice, Slovakia, October 7-10*, pp. 267–278, 2014.
9. J. M. Rodríguez-Jiménez, P. Cordero, M. Enciso, y A. Mora. Negative attributes and implications in formal concept analysis. *Procedia Computer Science*, 31: pp. 758–765. Presentado en 2nd International Conference on Information Technology and Quantitative Management, ITQM, Moscow, June 3–5, 2014.

# Capítulo 1

## Introducción

Es común confundir los términos *datos*, *información* y *conocimiento*. Sin embargo, son nociones bien diferenciadas, y estudiadas, que frecuentemente se representan en la bien conocida *Pirámide del Conocimiento*. Llamamos datos a conjuntos de valores (numéricos o de cualquier otro tipo) sin ninguna estructura. Un ejemplo sería el conjunto de medidas que va proporcionando un sensor de temperatura del aceite de un motor. Los datos por sí solos no tienen relevancia, adquiriendo el valor de información cuando se agrupan con cierta estructura. Por ejemplo, si recabamos datos de distintos sensores del motor y los organizamos en función del instante de tiempo en que se han obtenido, disponemos de información sobre el funcionamiento del motor. Hay una expresión popular que resume claramente el problema intrínseco a la información: “el exceso de información es peor que la falta de ésta”. Esto se debe a que, cuando tenemos que tratar con mucha información, se hace indispensable procesarla para sacar conclusiones, resumirla, estudiar reglas o patrones que se repiten, etc. En resumen, obtener de la información aquellos aspectos más relevantes que nos permitan *razonar* sobre ella. Esta última parte se encuentra en la cima de la pirámide y a ella se refiere el término conocimiento. Una gran cantidad de información sí es muy útil si somos



capaces de extraer conocimiento de ella y razonar sobre este conocimiento. A esto se refiere otra popular frase: “la información es poder”.

El término conocimiento engloba tres pilares básicos: extracción del conocimiento, representación de éste y razonamiento. Los tres son fundamentales. Necesitamos herramientas eficientes para extraer conocimiento a partir de la información, alguna forma de representar este conocimiento para que no siga teniendo la misma tacha que el *exceso de información* y mecanismos (automáticos) que permitan razonar sobre este conocimiento, es decir, extraer nuevo conocimiento implícito del conocimiento explícito que tenemos representado. Aquí tiene un papel relevante la *lógica*.

La pirámide del conocimiento da una idea estática de este fenómeno (de los datos obtenemos información y, es éste, conocimiento) que no es real. La inteligencia computacional pretende reproducir el comportamiento de la mente humana y, en ella, el proceso es cíclico. Constantemente recibimos datos e información que transformamos en conocimiento, y este conocimiento lo volvemos a contrastar con nuevos datos e información. Es decir, que estamos constantemente revisando nuestro conocimiento con nueva información: *aprendiendo*.

El Análisis de Conceptos Formales (ACF) es una disciplina que abarca todos estos aspectos. Fue introducida en los años ochenta del siglo pasado por Rudolf Wille [103] y, desde entonces, ha tenido un constante crecimiento. Esta disciplina, firmemente establecida, proporciona herramientas y técnicas para el descubrimiento de conocimiento, la representación de este conocimiento y métodos que permiten el razonamiento automático sobre él.

La forma más exclusiva de representar conocimiento en esta disciplina la constituyen los *conceptos formales*. Se trata de pares de conjuntos de objetos y conjuntos de propiedades que se identifican mutuamente: objetos con propiedades comunes y propiedades compartidas por los objetos. Wille formaliza de esta manera nociones introducidas en los primeros años del

siglo XX por el científico, filósofo y lógico estadounidense Charles S. Peirce en su sistema de *categorías*, alternativo al de Aristóteles y al de Kant. Estos conceptos formales se ordenan en una jerarquía (*subconceptos–superconceptos*) dando lugar a un retículo, conocido como *retículo de conceptos*.

Una representación alternativa al retículo de conceptos la constituyen las *implicaciones de atributos*, que indican conjuntos de atributos cuya presencia implica la de otros atributos. El primer conjunto de atributos se le llama *premisa* y, al segundo, *conclusión*. La ventaja de esta representación es que se parece más a las expresiones con las que habitualmente se trabaja en lógica y, por tanto, permite darle un tratamiento al estilo de la lógica y, por ende, el diseño de mecanismos (automáticos) de razonamiento.

Por otro lado, se dispone de una amplia colección de algoritmos y técnicas para extraer conocimiento desde la información, el *contexto formal*, tanto en forma de retículo de conceptos como de conjuntos de implicaciones de atributos. Del mismo modo, disponemos de algoritmos para pasar del retículo de conceptos a conjuntos de implicaciones, y viceversa.

Las herramientas clásicas de tratamiento de información, y en particular de ACF, se basan en la información positiva, es decir, la presencia de atributos en los objetos. Sin embargo, la no presencia o ausencia de un atributo en un objeto también proporciona un conocimiento significativo que, con el enfoque clásico, solo puede ser considerado parcialmente. Estas herramientas sí permiten considerar, como parte de nuestro conocimiento, negaciones de implicaciones de atributos. Es decir, permiten afirmar que una determinada implicación no es cierta. Sin embargo, esto puede corresponder a dos situaciones bien diferentes en relación con la ausencia de propiedades, o negación de atributos, en los objetos del contexto:

Caso A: Algunos objetos tienen los atributos de la premisa y de la conclusión, mientras que otros, teniendo los atributos de la premisa, no tienen los de la conclusión.

Caso B: Todo objeto que tenga los atributos de la premisa, carece de los atributos de la conclusión.

La única manera de capturar separadamente estas dos situaciones en nuestro conocimiento es enriquecer la expresividad del lenguaje usado de forma que podamos hablar de *atributos negativos* o atributos ausentes. Llamaremos *implicaciones mixtas* a implicaciones en las que, tanto en la premisa como en la conclusión, pueden aparecer atributos positivos y/o negativos. Hay que diferenciar claramente entre negación de implicaciones e implicaciones mixtas. El problema de la negación de implicaciones ya está resuelto en el tratamiento clásico mientras que las implicaciones mixtas, es decir, las distinciones entre los casos A y B, es lo que centra nuestro trabajo. El objetivo es aportar conocimiento a través del enriquecimiento de las implicaciones.

No es la primera vez que se aborda este problema. De hecho, podemos encontrar numerosos trabajos (en el Capítulo 3 presentaremos un análisis de estos trabajos) en la bibliografía de la última década que abordan este tema. Sin embargo, consideramos que son soluciones parciales y es necesario dar un tratamiento más completo y uniforme a este tema, echamos en falta un estudio global de los diferentes aspectos relacionados con este problema que sirva de marco común en el que desarrollar aplicaciones eficientes. Básicamente este es el **objetivo global** de esta tesis: desarrollar un marco general que permita, de una forma uniforme y eficiente, extraer, tratar (razonar) y representar conocimiento a partir de información positiva y negativa.

Los **objetivos concretos** de la tesis son:

- Puesto que el desarrollo de la teoría de ACF se basa en la noción de conexión de Galois, estudiaremos cómo la incorporación de atributos negativos repercute en la conexión de Galois correspondiente. En particular, introducimos una conexión de Galois generalizada que nos

llevará a un nuevo tipo de retículo que recoge toda la información, tanto positiva como negativa.

Estudiaremos sus propiedades y veremos que, a diferencia de lo que ocurre en el caso clásico, no todos los retículos se corresponden con un contexto. El objetivo principal de este apartado será dar un teorema de caracterización de tales retículos.

- El estudio de las implicaciones mixtas y la definición de sistemas axiomáticos que exploten las ventajas inherentes a la semántica de la negación, abriendo las puertas al diseño de mecanismos de razonamiento automático.
- El diseño de herramientas que permitan extraer conocimiento a partir de información positiva y negativa, haciendo uso del marco formal desarrollado.
- El estudio de los problemas en los que se pueden aplicar los resultados y las herramientas desarrolladas.

La tesis está organizada como sigue:

Tras un capítulo de preliminares, que tiene como objetivo que la tesis sea lo más autocontenida posible, en el Capítulo 3 presentamos una revisión exhaustiva de la bibliografía existente en relación con este problema. La finalidad de este capítulo es enmarcar claramente los objetivos de la tesis dentro de la bibliografía existente y el estado actual del área de investigación.

En el Capítulo 4 extenderemos los fundamentos de ACF para incluir la información negativa desde los primeros pasos. Esto nos llevará a una modificación del retículo de conceptos que llamaremos retículo de conceptos mixtos, o retículo mixto para abreviar. A lo largo de toda la tesis utilizamos el adjetivo *mixto* para distinguir las nuevas nociones, que consideran también la información negativa, de las clásicas. Tras destacar que, a diferencia de lo

que ocurre con la construcción clásica, no todo retículo finito es isomorfo a un retículo mixto, nos centraremos en caracterizar los retículos que sí cumplen esta propiedad.

El Capítulo 5 estará dedicado a las implicaciones de atributos mixtos. Los trabajos encontrados en la bibliografía se centran en estudiar qué nuevas implicaciones considerar y añadir a las clásicas para, con los mecanismos de inferencia clásicos, fundamentalmente el conocido como *axiomas de Armstrong*, poder derivar todo el conocimiento. En este capítulo propondremos una solución radicalmente diferente que consiste en añadir nuevos axiomas y reglas de inferencia a los de Armstrong para conseguir un sistema axiomático correcto y completo.

Una vez fijados los fundamentos teóricos de ambos pilares, el retículo de conceptos y los sistemas implicacionales mixtos, en el Capítulo 6 propondremos algoritmos, basados en los resultados de los capítulos anteriores, para la extracción del conocimiento a partir de la información proporcionada en términos de contexto formal.

Concluimos la tesis con un capítulo, el 7, en el que esbozamos distintos problemas donde consideramos que estos resultados se pueden aplicar. De hecho ya hemos comenzado a trabajar en estas líneas.

# Capítulo 2

## Preliminares

En este capítulo, con la finalidad de hacer el trabajo autocontenido en la medida de lo posible, introducimos los conceptos y resultados sobre conjuntos parcialmente ordenados (y, en particular, de retículos), conexiones de Galois y operadores de clausura, o cierre, que serán utilizados a lo largo de todo el trabajo. Aunque la bibliografía sobre estos tópicos es muy amplia, destacamos especialmente [45], [14] y [29].

Así mismo, incluimos las ideas, definiciones y propiedades fundamentales del Análisis de Conceptos Formales, área en la que se enmarca esta tesis. El libro de referencia sobre estos temas es [43].

### 2.1. Conjuntos parcialmente ordenados y retículos

Una *relación binaria* entre dos conjuntos  $A$  y  $B$  es un subconjunto

$$R \subseteq A \times B = \{\langle a, b \rangle \mid a \in A, b \in B\}.$$

Cuando  $A = B$  se denomina *relación binaria interna en  $A$*  y, en estos casos, habitualmente usaremos notación infija, es decir, escribiremos  $aRb$  en lugar de  $\langle a, b \rangle \in R$ .

**Definición 2.1.1** Un par  $\langle A, \leq \rangle$  se denomina **conjunto parcialmente ordenado** si  $A$  es un conjunto no vacío y  $\leq$  una relación binaria interna en  $A$  que satisface las siguientes propiedades:

- *Reflexiva:*  $a \leq a$  para todo  $a \in A$ .
- *Antisimétrica:*  $a \leq b$  y  $b \leq a$  implica  $a = b$  para todo  $a, b \in A$ .
- *Transitiva:*  $a \leq b$  y  $b \leq c$  implica  $a \leq c$  para todo  $a, b, c \in A$ .

Decimos que  $\leq$  es un orden parcial en  $A$ .

Usaremos con frecuencia la siguiente notación de intervalos en un conjunto parcialmente ordenado  $\langle A, \leq \rangle$ : para cada  $a \in A$ ,

$$[a) = \{x \in A \mid a \leq x\} \quad \text{y} \quad (a] = \{x \in A \mid x \leq a\}$$

Usaremos también el símbolo  $<$  para denotar el correspondiente orden estricto, es decir,  $a < b$  si y solo si  $a \leq b$  y  $a \neq b$ .

La forma estándar de representar gráficamente, cuando sea posible, los conjuntos parcialmente ordenados es mediante el conocido como *diagrama de Hasse*. Nosotros lo usaremos para conjuntos finitos, en los que siempre es posible utilizar este tipo de diagramas. Dado un conjunto parcialmente ordenado finito  $\langle A, \leq \rangle$ , su diagrama de Hasse es el grafo dirigido  $\langle A, \prec \rangle$  en el que los elementos de  $A$  son los vértices y las aristas vienen dadas por la *relación de cubrimiento*:  $a \prec b$  si y solo si se satisfacen las dos condiciones siguientes:

1.  $a < b$  y
2.  $a \leq c < b$  implica  $a = c$  para todo  $c \in A$ .

Para simplificar la lectura, los vértices, ordenados de menor a mayor, se representan de abajo hacia arriba, es decir, los más pequeños abajo y los mayores arriba.

La interpretación del diagrama de Hasse se basa en el hecho de que la relación de orden  $\leq$  es el *cierre reflexivo y transitivo* de la relación de cubrimiento  $\prec$ : dados dos elementos  $a, b \in A$ , se tiene que  $a \leq b$  si y solo si  $a = b$  o existe una secuencia  $x_1, \dots, x_n \in A$  tal que  $x_1 = a$ ,  $x_n = b$  y  $x_i \prec x_{i+1}$  para todo  $1 \leq i < n$ , es decir, una cadena ascendente desde  $a$  hasta  $b$  en el grafo.

**Ejemplo 2.1.2** El diagrama de Hasse del conjunto parcialmente ordenado  $\langle A, \leq \rangle$ , donde  $A = \{1, 2, 3, 4, 6, 12\}$  y  $\leq$  es la relación de divisibilidad ( $a \leq b$  si y solo si existe  $c \in \mathbb{Z}$  tal que  $ac = b$ ), es el dibujado en la Figura 2.1.

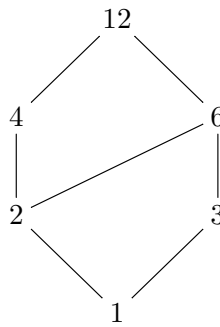


Figura 2.1: Divisores positivos de 12

**Definición 2.1.3** Sea  $\langle A, \leq \rangle$  un conjunto parcialmente ordenado,  $B \subseteq A$  y  $a \in A$ . Decimos que:

- $a$  es *cota superior* de  $B$  si  $B \subseteq (a]$ .
- $a$  es *cota inferior* de  $B$  si  $B \subseteq [a)$ .

Si existe al menos una cota superior y una inferior de  $B$  decimos que  $B$  es acotado.

**Ejemplo 2.1.4** En el conjunto parcialmente ordenado de la Figura 2.1, el conjunto  $B = \{2, 3, 6\}$  tiene como cotas superiores a 6 y a 12 y, como cota inferior, a 1.



**Definición 2.1.5** Sea  $\langle A, \leq \rangle$  un conjunto parcialmente ordenado,  $B \subseteq A$  y  $b \in A$ . Decimos que:

- $b$  es **máximo** de  $B$  si  $b \in B \subseteq (b]$ , es decir, es una cota superior que pertenece al subconjunto.
- $b$  es **mínimo** de  $B$  si  $b \in B \subseteq [b)$ , es decir, es una cota inferior que pertenece al subconjunto.

Por la antisimetría de  $\leq$  se tiene que el máximo y el mínimo de un conjunto, si existen, son únicos. Los denotaremos por  $\text{máx } B$  y  $\text{mín } B$  respectivamente. Además, si en un conjunto parcialmente ordenado  $\langle A, \leq \rangle$ , el propio  $A$  es acotado, entonces existen el máximo y el mínimo de  $A$  que serán denotados por  $\top$  y  $\perp$  respectivamente.

**Definición 2.1.6** Sea  $\langle A, \leq \rangle$  un conjunto parcialmente ordenado y  $B \subseteq A$ . Decimos que un elemento de  $A$  es:

- el **supremo** de  $B$  si es el mínimo de las cotas superiores de  $B$  (denotado como  $\text{sup } B$  o como  $\bigvee B$ ) y
- el **ínfimo** de  $B$  si es el máximo de las cotas inferiores de  $B$  (denotado como  $\text{ínf } B$  o como  $\bigwedge B$ ).

Obsérvese que  $\bigvee \emptyset$  existe si y solo si existe el mínimo de  $A$  y, en tal caso,  $\bigvee \emptyset = \text{mín } A = \perp$ . Del mismo modo,  $\bigwedge \emptyset$  existe si y solo si existe el máximo de  $A$  y, en tal caso,  $\bigwedge \emptyset = \text{máx } A = \top$ .

### 2.1.1. Retículos y retículos completos

Ya tenemos todas las nociones previas que nos permiten introducir la noción de retículo.

**Definición 2.1.7** *Un retículo ordenado es un conjunto parcialmente ordenado  $\langle L, \leq \rangle$  en el que existen  $\bigvee B$  y  $\bigwedge B$  para todo  $B \subseteq L$  finito no vacío.*

Las condiciones de la definición anterior son equivalentes a pedir la existencia del supremo y del ínfimo de los pares de elementos, es decir, que para todo  $a, b \in L$ , existan  $\bigvee\{a, b\}$  y  $\bigwedge\{a, b\}$ .

Como es bien conocido, existe una definición algebraica equivalente para la estructura de retículo.

**Definición 2.1.8** *Un semirretículo algebraico es un par  $\langle A, \star \rangle$  donde  $A$  es un conjunto no vacío y  $\star$  es una operación binaria interna en  $A$  con las siguientes propiedades:*

1. *Idempotencia:*  $a \star a = a$  para todo  $a \in A$ .
2. *Conmutatividad:*  $a \star b = b \star a$  para todo  $a, b \in A$ .
3. *Asociatividad:*  $a \star (b \star c) = (a \star b) \star c$  para todo  $a, b, c \in L$ .

*Un retículo algebraico es una terna  $\langle A, \vee, \wedge \rangle$  donde  $\langle A, \vee \rangle$  y  $\langle A, \wedge \rangle$  son semirretículos algebraicos y además se cumple la siguiente propiedad:*

4. *Absorción:*  $a \vee (a \wedge b) = a$  y  $a \wedge (a \vee b) = a$  para todo  $a, b \in L$ .

La equivalencia mencionada entre retículo ordenado y algebraico hace referencia a los siguientes hechos: Todo retículo ordenado  $\mathbb{L} = \langle L, \leq \rangle$  define un retículo algebraico  $\mathbb{L}^a = \langle L, \vee, \wedge \rangle$  donde  $a \vee b = \bigvee\{a, b\}$  y  $a \wedge b = \bigwedge\{a, b\}$ . Recíprocamente, todo retículo algebraico  $\mathbb{L} = \langle L, \vee, \wedge \rangle$  define un retículo ordenado  $\mathbb{L}^o = \langle L, \leq \rangle$  donde  $a \leq b$  si y solo si  $a \vee b = b$  ó, equivalentemente,  $a \wedge b = a$ . Además, se cumple que  $(\mathbb{L}^o)^a = \mathbb{L}$  y  $(\mathbb{L}^a)^o = \mathbb{L}$ .

Como consecuencia de esta equivalencia, en adelante, hablaremos de retículos, sin diferenciar entre ordenados y algebraicos.

**Definición 2.1.9** *Un conjunto parcialmente ordenado es un retículo completo si todo subconjunto tiene supremo.*

Es fácil probar que un conjunto parcialmente ordenado  $\langle L, \leq \rangle$  es un retículo completo si y solo si satisface alguna de las siguientes propiedades (todas equivalentes entre sí):

1. Existe  $\bigwedge H$  para todo  $H \subseteq L$ .
2. Existe  $\top = \text{máx } L$  y, para todo  $\emptyset \neq H \subseteq L$ , existe  $\bigwedge H$ .
3. Existe  $\bigvee H$  para todo  $H \subseteq L$ .
4. Existe  $\perp = \text{mín } L$  y, para todo  $\emptyset \neq H \subseteq L$ , existe  $\bigvee H$ .

Como consecuencia, todo retículo completo es un retículo acotado.

En un retículo completo  $\mathbb{L} = \langle L, \leq \rangle$ , para cada  $x \in L$ , se definen:

$$x_* = \bigvee \{y \in L \mid y < x\} \quad \text{y} \quad x^* = \bigwedge \{y \in L \mid x < y\} \quad (2.1)$$

Obviamente, se tiene que  $x_* \leq x \leq x^*$  para todo  $x \in L$ .

**Definición 2.1.10** *Sea  $\mathbb{L} = \langle L, \leq \rangle$  un retículo completo. Decimos que  $x \in L$  es  $\vee$ -irreducible si  $x \neq x_*$  y que  $x$  es  $\wedge$ -irreducible si  $x \neq x^*$ .*

El conjunto de los elementos  $\vee$ -irreducibles en  $\mathbb{L}$  se va a denotar por  $J(\mathbb{L})$  (por su nombre en inglés: join-irreducible elements) y el de los  $\wedge$ -irreducibles por  $M(\mathbb{L})$  (meet-irreducible elements).

Como consecuencia de la definición, si  $j \in J(\mathbb{L})$ , se tiene que:

- i.  $\ell_1 \vee \ell_2 = j$  implica que  $\ell_1 = j$  o  $\ell_2 = j$  para todo  $\ell_1, \ell_2 \in L$ .

En el caso de los retículos finitos, esta propiedad i. caracteriza a los elementos que son  $\vee$ -irreducibles. Además, se tiene que  $j_*$  es el único vecino inferior de  $j$ , el único que cumple que  $j_* \prec j$ .

De igual forma, si  $m \in M(\mathbb{L})$ , se cumple que:

ii.  $\ell_1 \wedge \ell_2 = m$  implica que  $\ell_1 = m$  o  $\ell_2 = m$  para todo  $\ell_1, \ell_2 \in L$ .

y, en el caso finito,  $m^*$  es el único vecino superior de  $m$ , con  $m \prec m^*$ .

**Definición 2.1.11** Sea  $\mathbb{L} = \langle L, \leq \rangle$  un retículo acotado. Un elemento  $a \in L$  se dice que es un **átomo** si  $\ell \leq a$  implica  $\ell = \perp$  o  $\ell = a$  para todo  $\ell \in L$ .

El conjunto de los átomos de  $\mathbb{L}$  se denotará por  $\text{At}(\mathbb{L})$ . Se tiene pues que  $a \in \text{At}(\mathbb{L})$  si y solo si  $(a) = \{\perp, a\}$ .

**Ejemplo 2.1.12** El conjunto parcialmente ordenado de la Figura 2.1 es un retículo en el que  $J(\mathbb{L}) = \{2, 3, 4\}$  (las potencias de números primos),  $M(\mathbb{L}) = \{3, 4, 6\}$  y  $\text{At}(\mathbb{L}) = \{2, 3\}$  (los números primos).

**Definición 2.1.13** Sea  $\mathbb{L} = \langle L, \leq \rangle$  un retículo completo. Decimos que un subconjunto  $H \subseteq L$  es  **$\vee$ -denso** (respectivamente  **$\wedge$ -denso**) si para todo  $\ell \in L$  existe un subconjunto  $T \subseteq H$  tal que  $\ell = \vee T$  (respectivamente  $\ell = \wedge T$ ).

Obviamente, si  $H$  es un conjunto  $\vee$ -denso, para todo  $\ell \in L$ , se tiene que  $\ell = \vee \{x \in H \mid x \leq \ell\}$  y si  $H$  es  $\wedge$ -denso entonces  $\ell = \wedge \{x \in H \mid \ell \leq x\}$  para todo  $\ell \in L$ .

**Proposición 2.1.14** Sea  $\mathbb{L} = \langle L, \leq \rangle$  un retículo finito y  $H \subseteq L$ . Se tiene que:

1.  $H$  es  $\vee$ -denso si y solo si  $J(\mathbb{L}) \subseteq H$ .
2.  $H$  es  $\wedge$ -denso si y solo si  $M(\mathbb{L}) \subseteq H$ .

Obsérvese que esto no es cierto, en general, para retículos infinitos. Por ejemplo, en el retículo  $\langle \mathbb{R}, \leq \rangle$  tenemos que  $J(\mathbb{R}) = M(\mathbb{R}) = \emptyset$  y, si el resultado anterior se pudiese aplicar, todos los subconjuntos serían  $\wedge$ -densos y  $\vee$ -densos. Sin embargo, tiene subconjuntos que son  $\wedge$ -densos y  $\vee$ -densos, por ejemplo  $\mathbb{Q}$ , y subconjuntos que no lo son, por ejemplo  $\mathbb{Z}$ .

### 2.1.2. Subretículos y homomorfismos de retículos

Recordamos, a continuación, solo un conjunto de definiciones que vamos a necesitar en los siguientes capítulos.

**Definición 2.1.15** Sea  $\mathbb{L}$  un retículo y  $\emptyset \neq M \subseteq L$ . Decimos que  $M$  es un  $\vee$ -**subsemirretículo** de  $\mathbb{L}$  si es cerrado bajo supremos ( $a, b \in M$  implica  $a \vee b \in M$ ) y que es un  $\wedge$ -**subsemirretículo** de  $\mathbb{L}$  si es cerrado bajo ínfimos ( $a, b \in M$  implica  $a \wedge b \in M$ ). El conjunto  $M$  es **subretículo** de  $\mathbb{L}$  si es  $\vee$ -subsemirretículo y  $\wedge$ -subsemirretículo de  $\mathbb{L}$ .

En los retículos completos, las definiciones de subsemirretículo,  $\vee$ -subsemirretículo y  $\wedge$ -subsemirretículo se obtienen cambiando las condiciones de “ser cerrado”, en lugar de a pares de elementos, a subconjuntos arbitrarios, es decir, si  $\mathbb{L}$  un retículo completo, un subconjunto  $H \subseteq L$  es  $\vee$ -subsemirretículo de  $\mathbb{L}$  si  $\bigvee T \in H$ , para todo  $T \subseteq H$ , y es  $\wedge$ -subsemirretículo de  $\mathbb{L}$  si  $\bigwedge T \in H$ , para todo  $T \subseteq H$ .

Nos centramos ahora en las funciones entre conjuntos parcialmente ordenados o entre retículos.

**Definición 2.1.16** Sean  $\langle A, \leq \rangle$  y  $\langle B, \leq \rangle$  dos conjuntos parcialmente ordenados. Una función  $f: A \rightarrow B$  se dice que es

- **isótona** si  $a \leq b$  implica  $f(a) \leq f(b)$  para todo  $a, b \in A$ .
- **antítóna** si  $a \leq b$  implica  $f(b) \leq f(a)$  para todo  $a, b \in A$ .

**Definición 2.1.17** Sea  $\langle A, \leq \rangle$  un conjunto parcialmente ordenado. Una función  $f: A \rightarrow A$  se dice que es

- **inflacionaria** si  $a \leq f(a)$  para todo  $a \in A$ .
- **deflacionaria** si  $f(a) \leq a$  para todo  $a \in A$ .

La definición de homomorfismo entre retículos sigue el esquema habitual.

**Definición 2.1.18** Sean  $\mathbb{L}_1 = \langle L_1, \leq \rangle$  y  $\mathbb{L}_2 = \langle L_2, \leq \rangle$  dos retículos (completos). Una función  $f: L_1 \rightarrow L_2$  es un **homomorfismo de retículos (completos)** si respeta supremos e ínfimos de subconjuntos finitos (subconjuntos arbitrarios). Si, además,  $f$  es biyectiva, decimos que es un **isomorfismo entre  $\mathbb{L}_1$  y  $\mathbb{L}_2$** .

Decimos que dos retículos (completos) son **isomorfos**, y lo denotamos por  $\mathbb{L}_1 \cong \mathbb{L}_2$ , si existe un isomorfismo entre ellos.

Por tanto, en el caso de los retículos, para todo  $a, b \in L_1$ , se debe cumplir que:

$$f(a \vee b) = f(a) \vee f(b) \quad \text{y} \quad f(a \wedge b) = f(a) \wedge f(b) \quad (2.2)$$

mientras que, en el caso de los retículos completos, la función  $f$  debe satisfacer que, para todo  $H \subseteq L_1$ ,

$$f(\bigvee H) = \bigvee f(H) \quad \text{y} \quad f(\bigwedge H) = \bigwedge f(H) \quad (2.3)$$

Como consecuencia de la definición, todo homomorfismo entre retículos es isótono. Es más,  $f: \mathbb{L}_1 \rightarrow \mathbb{L}_2$  es un isomorfismo de retículos completos si y solo si  $f$  es biyectiva y, para todo  $a, b \in L_1$ ,

$$a \leq b \quad \text{si y solo si} \quad f(a) \leq f(b). \quad (2.4)$$

### 2.1.3. Retículos booleanos

**Definición 2.1.19** Sea  $\mathbb{L} = \langle L, \vee, \wedge, \top, \perp \rangle$  un retículo acotado y  $a \in L$ . Decimos que un elemento  $b \in L$  es **complemento** de  $a$  si  $a \vee b = \top$  y  $a \wedge b = \perp$ .

Decimos que el retículo es **complementado** si todo elemento tiene complemento.

Por ejemplo, los retículos de la Figura 2.2 son complementados.

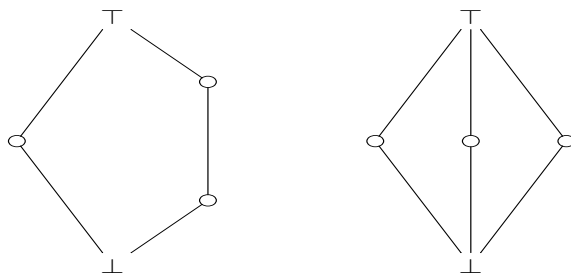


Figura 2.2: Retículos conocidos como *Pentágono* y *Diamante* respectivamente.

**Definición 2.1.20** Un retículo  $\mathbb{L} = \langle L, \vee, \wedge \rangle$  se dice **distributivo** si, para todo  $a, b, c \in L$ , se cumple que

$$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c) \quad \text{y} \quad a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$$

Es bien conocido que un retículo es distributivo si y solo si no tiene ningún subretículo isomorfo al *Pentágono* o al *Diamante*. Véase la Figura 2.2.

**Definición 2.1.21** Llamamos **retículo booleano** a todo retículo acotado, complementado y distributivo.

En los retículos booleanos, el complemento de cada elemento, además de existir, es único. Podemos, por tanto, dar una versión algebraica de la definición anterior:  $\langle L, \vee, \wedge, {}^c, \top, \perp \rangle$  donde  $\langle L, \vee, \wedge, \top, \perp \rangle$  es un retículo acotado y  $(-)^c: L \rightarrow L$  es una operación monaria tal que  $a \wedge a^c = \perp$  y  $a \vee a^c = \top$ . Esta estructura algebraica se conoce como **álgebra de Boole**.

## 2.2. Operadores de clausura y conexiones de Galois

En esta sección se introducen las nociones de operadores de clausura, de sistemas de clausura y de conexión de Galois. Además se recuerdan algunos resultados sobre las relaciones entre operadores de clausura y sistemas de

clausura, así como la relación entre las conexiones de Galois y los operadores de clausura (sistemas de clausura).

**Definición 2.2.1** Sea  $\langle A, \leq \rangle$  un conjunto parcialmente ordenado. Un **operador de clausura** (o de **cierre**) sobre  $A$  es una aplicación  $c: A \rightarrow A$  que es isótona, inflacionaria e idempotente ( $c \circ c = c$ ).

**Definición 2.2.2 ([37])** Sea  $\langle A, \leq \rangle$  un conjunto parcialmente ordenado. Un subconjunto  $S \subseteq A$  se dice que es un **sistema de clausura** (también conocido como **familia de Moore**) si para todo  $a \in A$  el conjunto  $\{s \in S \mid a \leq s\}$  tiene mínimo. Dicho de otro modo, si existe  $\min([a] \cap S)$  para todo  $a \in A$ .

La siguiente proposición muestra la relación biunívoca que existe entre operadores de clausura y sistemas de clausura de un conjunto parcialmente ordenado.

**Proposición 2.2.3** Sea  $\langle A, \leq \rangle$  un conjunto parcialmente ordenado.

1. Si  $c: A \rightarrow A$  es un operador de clausura, entonces el conjunto imagen de  $c$ , que coincide con  $S_c = \{a \in A \mid c(a) = a\}$ , es un sistema de clausura.
2. Si  $S$  es un sistema de clausura, entonces la aplicación  $c_S: A \rightarrow A$ , definida como  $c_S(a) = \min([a] \cap S)$  para todo  $a \in A$ , es un operador de clausura.

Además,  $c = c_{S_c}$  y  $S = S_{c_S}$ .

En la proposición anterior,  $c_S$  se denomina *operador de clausura asociado a  $S$* , y  $S_c$  se llama *sistema de clausura asociado a  $c$* .

En el caso de los retículos, la siguiente proposición da una caracterización alternativa para los sistemas de clausura.

**Proposición 2.2.4** Sea  $\langle L, \leq \rangle$  un retículo completo. Un subconjunto  $S \subseteq L$  es un sistema de clausura si y solo si  $S$  es un  $\wedge$ -subsemirretículo que contiene a  $\top$ .



Obsérvese que, en este caso,  $c_S(a) = \bigwedge \{s \in S \mid a \leq s\}$  y que  $\langle S, \leq \rangle$  es un retículo completo también, aunque no necesariamente un subretículo de  $\langle L, \leq \rangle$ . Por otro lado, todo retículo completo obviamente es un sistema de clausura.

**Definición 2.2.5** Sean  $\langle A, \leq \rangle$  y  $\langle B, \leq \rangle$  dos conjuntos parcialmente ordenados y  $f: A \rightarrow B$  y  $g: B \rightarrow A$ . Decimos que el par  $(f, g)$  es una **conexión de Galois** si para todo  $a \in A$  y todo  $b \in B$  se tiene que

$$b \leq f(a) \quad \text{si y solo si} \quad a \leq g(b) \quad (2.5)$$

**Teorema 2.2.6** Sean  $\langle A, \leq \rangle$  y  $\langle B, \leq \rangle$  dos conjuntos parcialmente ordenados y  $f: A \rightarrow B$  y  $g: B \rightarrow A$ . Entonces son equivalentes las siguientes afirmaciones:

1.  $(f, g)$  es una conexión de Galois.
2.  $f$  y  $g$  son antítonas y las composiciones  $f \circ g$  y  $g \circ f$  son inflacionarias.
3.  $f$  es antítona y  $g(b) = \text{máx}\{a \in A \mid b \leq f(a)\}$  para todo  $b \in B$ .
4.  $g$  es antítona y  $f(a) = \text{máx}\{b \in B \mid a \leq g(b)\}$  para todo  $a \in A$ .

En consecuencia, en toda conexión de Galois, una función del par determina unívocamente a la otra. Por otro lado, de la condición 2. es fácil deducir que  $g \circ f \circ g = g$  y que  $f \circ g \circ f = f$  y, como consecuencia se tiene el siguiente teorema.

**Teorema 2.2.7** Si  $(f, g)$  es una conexión de Galois entre dos conjuntos parcialmente ordenados  $\mathbb{A}$  y  $\mathbb{B}$ , entonces  $g \circ f$  es un operador de clausura en  $\mathbb{A}$  y  $f \circ g$  es un operador de clausura en  $\mathbb{B}$ .

**Teorema 2.2.8** Si  $(f, g)$  es una conexión de Galois entre dos retículos (completos)  $\mathbb{A} = \langle A, \leq \rangle$  y  $\mathbb{B} = \langle B, \leq \rangle$ , entonces los sistemas de clausura

$$S_{gf} = \{a \in A \mid g(f(a)) = a\} \quad \text{y} \quad S_{fg} = \{b \in B \mid f(g(b)) = b\}$$

son retículos (completos) dualmente isomorfos, es decir  $\langle S_{gf}, \leq \rangle \cong \langle S_{fg}, \geq \rangle$ .

### 2.3. Análisis de Conceptos Formales

El Análisis de Conceptos Formales (ACF) fue introducido por Rudolf Wille [103] en la década de los 80. Se trata de un entorno conceptual para estructurar, analizar, minimizar, visualizar y revelar conocimiento a partir de la información proporcionada en términos de una relación binaria entre los elementos de dos conjuntos: objetos y atributos. Las técnicas de ACF han sido utilizadas en las tres últimas décadas con gran éxito en diversas áreas de investigación como minería de datos, análisis de redes sociales, marketing, diagnosis médica, etc.

El punto de partida lo constituye la relación existente entre un conjunto de objetos y sus propiedades.

**Definición 2.3.1** *Un contexto formal  $\mathbb{K}$  es una terna  $\langle G, M, I \rangle$  donde  $G$  y  $M$  son conjuntos no vacíos e  $I$  es una relación binaria entre  $G$  y  $M$ .*

$I$	$m_1$	$m_2$	$m_3$	$m_4$
$o_1$	×	×	×	
$o_2$	×		×	×
$o_3$		×	×	×
$o_4$	×			×

Tabla 2.1: Ejemplo de contexto formal

Los elementos de  $G$  se denominan *objetos*, y los de  $M$  *atributos*. Si  $\langle g, m \rangle \in I$ , decimos que el objeto  $g$  *tiene el atributo*  $m$  o que  $g$  *satisface*  $m$ . Por ejemplo, la Tabla 2.1 representa un contexto formal  $\mathbb{K} = \langle G, M, I \rangle$  donde el conjunto de objetos es  $G = \{o_1, o_2, o_3, o_4\}$ , el conjunto de atributos es  $M = \{m_1, m_2, m_3, m_4\}$  y la relación  $I$  viene dada por la tabla. En ella, por

ejemplo, se puede observar que  $\langle o_1, m_1 \rangle \in I$  (el objeto  $o_1$  tiene el atributo  $m_1$ ) pero  $\langle o_1, m_4 \rangle \notin I$  (el objeto  $o_1$  no satisface  $m_4$ ).

Del contexto formal se extrae conocimiento que es representado, o bien mediante una estructura conocida como *retículo de conceptos*, o bien mediante una *conjunto de implicaciones entre los atributos*. En realidad, ambas formas de representar conocimiento son equivalentes, en el sentido de que representan el mismo conocimiento y se puede reconstruir uno a partir del otro sin necesidad de volver a consultar el contexto formal. Dedicaremos las siguientes subsecciones a cada una de estas representaciones.

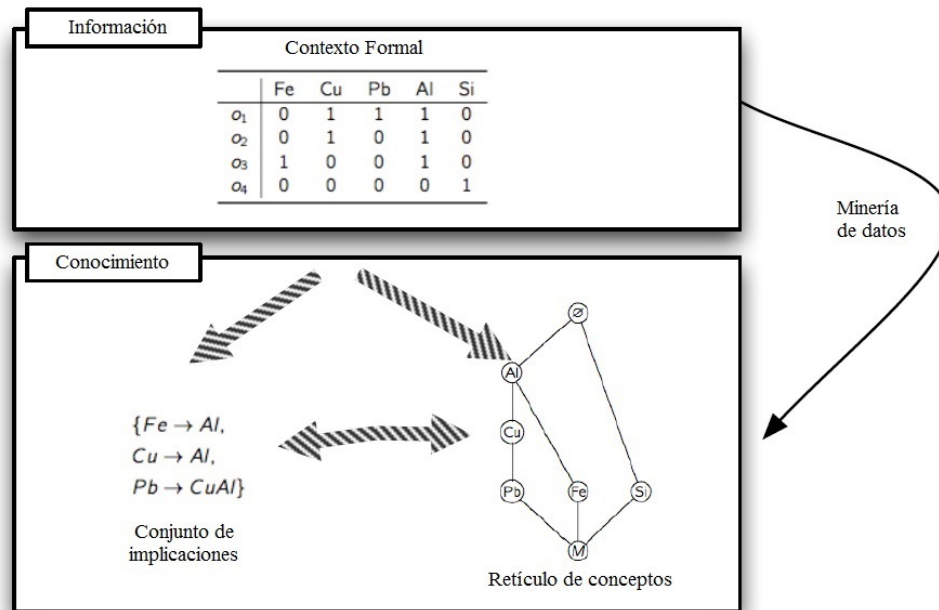


Figura 2.3: Diferentes caras del conocimiento en ACF

### 2.3.1. Retículo de conceptos formales

El siguiente elemento a analizar dentro del ACF son los propios Conceptos Formales. Estos son subconjuntos de objetos y atributos que cumplen una serie de propiedades dentro del contexto. Para definirlos tenemos que empezar definiendo los operadores de derivación.

**Definición 2.3.2** Dado un contexto formal  $\langle G, M, I \rangle$ , llamamos *operadores de derivación* a las funciones  $\uparrow: 2^G \rightarrow 2^M$  y  $\downarrow: 2^M \rightarrow 2^G$  que se definen por:

$$A^\uparrow = \{m \in M \mid \langle g, m \rangle \in I \text{ para todo } g \in A\} \quad (2.6)$$

$$B^\downarrow = \{g \in G \mid \langle g, m \rangle \in I \text{ para todo } m \in B\} \quad (2.7)$$

para cada  $A \subseteq G$  y  $B \subseteq M$ .

El conjunto  $A^\uparrow$  es el de los atributos compartidos por todos los objetos de  $A$  y el conjunto  $B^\downarrow$  es el de los objetos que comparten todos los atributos de  $B$ .

**Ejemplo 2.3.3** Para el contexto de la Tabla 2.1 tenemos, por ejemplo, que:

$$\begin{aligned} \{o_1\}^\uparrow &= \{m_1, m_2, m_3\}, & \{o_1, o_2\}^\uparrow &= \{m_1, m_3\}, & G^\uparrow &= \emptyset & \text{ y } & \emptyset^\uparrow &= M. \\ \{m_1\}^\downarrow &= \{o_1, o_2, o_4\}, & \{m_1, m_2\}^\downarrow &= \{o_1\}, & M^\downarrow &= \emptyset, & \text{ y } & \emptyset^\downarrow &= G. \end{aligned}$$

**Definición 2.3.4** Sea  $\mathbb{K} = \langle G, M, I \rangle$  un contexto formal,  $A \subseteq G$  y  $B \subseteq M$ . Decimos que el par  $\langle A, B \rangle$  es un *concepto formal* de  $\mathbb{K}$  si  $A^\uparrow = B$  y  $B^\downarrow = A$ .

En otras palabras,  $\langle A, B \rangle$  es un concepto formal si  $A$  contiene justo todos los objetos que comparten los atributos de  $B$  y, respectivamente,  $B$  contiene justo todos los atributos que comparten los objetos de  $A$ . Podríamos decir que  $A$  es la descripción por “*extensión*” (en inglés “*extent*”) del concepto y  $B$  es la descripción por “*comprensión*” (en inglés “*intent*”) del concepto. Una forma alternativa de definir los conceptos es como pares de subconjuntos maximales tales que  $A \times B \subseteq I$ .

**Ejemplo 2.3.5** En el contexto de la Tabla 2.1, el par  $\langle \{o_2, o_3\}, \{m_3, m_4\} \rangle$  es un concepto formal.

$I$	$m_1$	$m_2$	$m_3$	$m_4$
$o_1$	×	×	×	
$o_2$	×		×	×
$o_3$		×	×	×
$o_4$	×			×

Tabla 2.2: Concepto Formal

**Teorema 2.3.6** Sea  $\mathbb{K} = \langle G, M, I \rangle$  un contexto formal. El par de operadores de derivación  $(\uparrow, \downarrow)$  forma una conexión de Galois entre  $\langle 2^G, \subseteq \rangle$  y  $\langle 2^M, \subseteq \rangle$ .

Como consecuencia directa de este teorema tenemos un conjunto de propiedades de los operadores de derivación (véase la sección anterior) de las que nos interesa especialmente el siguiente corolario.

**Corolario 2.3.7** Sea  $\mathbb{K} = \langle G, M, I \rangle$  un contexto formal. Las composiciones de sus operadores de derivación,  $\uparrow\downarrow: 2^G \rightarrow 2^G$  y  $\downarrow\uparrow: 2^M \rightarrow 2^M$ , son operadores de clausura cuyos sistemas de clausura son retículos completos dualmente isomorfos, es decir,  $\langle S_{\uparrow\downarrow}, \subseteq \rangle \cong \langle S_{\downarrow\uparrow}, \supseteq \rangle$ .

Dado que  $A^\uparrow = A^{\uparrow\downarrow\uparrow}$  para todo  $A \subseteq G$  y  $B^\downarrow = B^{\downarrow\uparrow\downarrow}$  para todo  $B \subseteq M$ , los cerrados de  $S_{\uparrow\downarrow}$  son precisamente las primeras componentes (extents) de los conceptos y los de  $S_{\downarrow\uparrow}$  las segundas componentes (intents) de los conceptos.

**Definición 2.3.8** Sea  $\mathbb{K} = \langle G, M, I \rangle$  un contexto formal. Llamamos **retículo de conceptos** de  $\mathbb{K}$ , y lo denotamos por  $\mathfrak{B}(\mathbb{K})$ , al retículo formado por los conceptos

con la relación de orden

$$\langle A_1, B_1 \rangle \leq \langle A_2, B_2 \rangle \text{ si y solo si } A_1 \subseteq A_2 \quad (2.8)$$

o, equivalentemente,  $B_1 \supseteq B_2$

El orden establecido entre los conceptos formales indica cuál de ellos es más específico. Así,  $\langle A_1, B_1 \rangle \leq \langle A_2, B_2 \rangle$  indica que  $\langle A_1, B_1 \rangle$  es más específico que  $\langle A_2, B_2 \rangle$  denominándose **subconcepto** o, igualmente, que  $\langle A_2, B_2 \rangle$  es más general denominándose **superconcepto**.

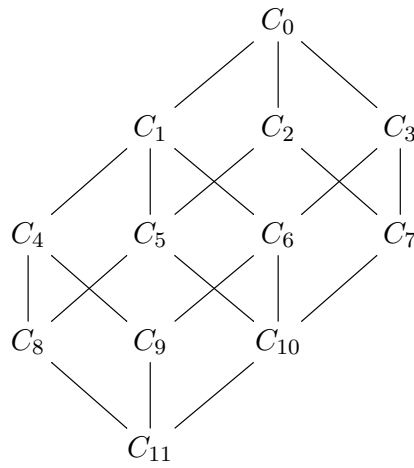


Figura 2.4: Retículo de conceptos del contexto descrito en la Tabla 2.1

**Ejemplo 2.3.9** Para el contexto formal descrito en la Tabla 2.1, su retículo de conceptos viene representado por la Figura 2.4 donde los conceptos son los siguientes:

- $C_0 = \langle G, \emptyset \rangle$
- $C_1 = \langle \{o_1, o_2, o_3\}, \{m_3\} \rangle$
- $C_2 = \langle \{o_1, o_2, o_4\}, \{m_1\} \rangle$
- $C_3 = \langle \{o_2, o_3, o_4\}, \{m_4\} \rangle$
- $C_4 = \langle \{o_1, o_3\}, \{m_2, m_3\} \rangle$
- $C_5 = \langle \{o_1, o_2\}, \{m_1, m_3\} \rangle$
- $C_6 = \langle \{o_2, o_3\}, \{m_3, m_4\} \rangle$
- $C_7 = \langle \{o_2, o_4\}, \{m_1, m_4\} \rangle$

- $C_8 = \langle \{o_1\}, \{m_1, m_2, m_3\} \rangle$
- $C_9 = \langle \{o_3\}, \{m_2, m_3, m_4\} \rangle$
- $C_{10} = \langle \{o_2\}, \{m_1, m_3, m_4\} \rangle$
- $C_{11} = \langle \emptyset, M \rangle$

**Teorema 2.3.10 (Teorema fundamental de los retículos de conceptos)** Para todo contexto  $\mathbb{K} = \langle G, M, I \rangle$  se tiene que  $\mathfrak{B}(\mathbb{K})$  es un retículo completo en el que el supremo y el ínfimo de una familia de conceptos  $\mathcal{F} = \{\langle A_i, B_i \rangle \mid i \in \Lambda\}$  vienen dados por

$$\bigvee \mathcal{F} = \left\langle \left( \bigcup_{i \in \Lambda} A_i \right)^{\uparrow\downarrow}, \bigcap_{i \in \Lambda} B_i \right\rangle \quad \text{y} \quad \bigwedge \mathcal{F} = \left\langle \bigcap_{i \in \Lambda} A_i, \left( \bigcup_{i \in \Lambda} B_i \right)^{\downarrow\uparrow} \right\rangle$$

Un retículo completo  $\mathbb{L} = \langle L, \leq \rangle$  es isomorfo a  $\mathfrak{B}(\mathbb{K})$  si y solo si existen funciones  $\tilde{\gamma}: G \rightarrow L$  y  $\tilde{\mu}: M \rightarrow L$  tales que  $\tilde{\gamma}(G)$  es  $\vee$ -denso en  $\mathbb{L}$ ,  $\tilde{\mu}(M)$  es  $\wedge$ -denso en  $\mathbb{L}$  y, para todo  $g \in G$  y  $m \in M$ , se cumple que:

$$\langle g, m \rangle \in I \quad \text{si y solo si} \quad \tilde{\gamma}(g) \leq \tilde{\mu}(m).$$

En particular,  $\mathbb{L} \cong \mathfrak{B}(L, L, \leq)$ .

En resumen, todo retículo de conceptos es un retículo completo y todo retículo completo  $\langle L, \leq \rangle$  es isomorfo a muchos retículos de conceptos, en particular, el del contexto  $\langle L, L, \leq \rangle$ . Nos preguntamos ahora si hay uno más pequeño. Recuérdese que, en el caso de los retículos finitos,  $\tilde{\gamma}(G)$  es  $\vee$ -denso si y solo si  $\mathcal{J}(\mathbb{L}) \subseteq \tilde{\gamma}(G)$ , y  $\tilde{\mu}(M)$  es  $\wedge$ -denso si y solo si  $\mathcal{M}(\mathbb{L}) \subseteq \tilde{\mu}(G)$  (véase la Proposición 2.1.14).

Es fácil caer en la cuenta de que si duplicamos filas y/o columnas en un contexto, el nuevo retículo de conceptos será isomorfo al inicial. Esto justifica la siguiente definición.

**Definición 2.3.11** Un contexto  $\mathbb{K} = \langle G, M, I \rangle$  se dice que está *clarificado* si cumple que:

1.  $\{g_1\}^\uparrow = \{g_2\}^\uparrow$  implica  $g_1 = g_2$  para todo  $g_1, g_2 \in G$ .

2.  $\{m_1\}^\downarrow = \{m_2\}^\downarrow$  implica  $m_1 = m_2$  para todo  $m_1, m_2 \in M$ .

Por un razonamiento similar llegamos a la conclusión de que, en un contexto finito, si una fila o columna es intersección de otras, puede ser eliminada del contexto sin afectar a la estructura del retículo de conceptos. Esto sucede, en particular, si tenemos una fila/columna completa o vacía.

**Definición 2.3.12** Un contexto finito  $\mathbb{K} = \langle G, M, I \rangle$  se dice que es *reducido* si, para todo  $A \subseteq G$ ,  $g \in G$ ,  $B \subseteq M$  y  $m \in M$ , se cumple que:

1.  $\{g\}^\uparrow \neq \emptyset$  y, si  $\{g\}^\uparrow = A^\uparrow$ , entonces  $g \in A$ .
2.  $\{m\}^\downarrow \neq \emptyset$  y, si  $\{m\}^\downarrow = B^\downarrow$ , entonces  $m \in B$ .

El siguiente teorema responde a la pregunta planteada a continuación del Teorema Fundamental.

**Teorema 2.3.13** Para todo retículo finito  $\mathbb{L} = \langle L, \leq \rangle$  se tiene que el contexto  $\mathbb{K}(\mathbb{L}) = \langle J(\mathbb{L}), M(\mathbb{L}), \leq \rangle$  cumple que  $\mathbb{L} \cong \mathfrak{B}(\mathbb{K}(\mathbb{L}))$ . Además, éste es el único contexto reducido, salvo isomorfismos, que cumple tal condición.

Este teorema, sin embargo, no se puede extender a todo retículo completo ya que, por ejemplo, para  $\langle \mathbb{R}, \leq \rangle$  se tiene que  $J(\mathbb{R}) = M(\mathbb{R}) = \emptyset$ .

**Ejemplo 2.3.14** Consideremos el retículo de los divisores positivos de 12 descrito en la Figura 2.1. El único contexto reducido cuyo retículo de conceptos es isomorfo a él es  $\langle \{2, 3, 4\}, \{3, 4, 6\}, I \rangle$  donde  $I$  viene descrito por la Tabla 2.3.

### 2.3.2. Implicaciones de atributos

Existe una forma de obtener y representar conocimiento sobre un contexto alternativa al retículo de conceptos: las implicaciones de atributos.



$I$	3	4	6
2		×	×
3	×		×
4		×	

Tabla 2.3: Contexto reducido para el retículo de los divisores de 12.

Veremos en esta sección que son equivalentes, en el sentido de que contienen el mismo conocimiento y se puede pasar de una a la otra sin necesidad de consultar el contexto formal. La ventaja del retículo de conceptos es que permite navegar por él, pasando de subconceptos a superconceptos y viceversa. La ventaja de las implicaciones de atributos es que permiten un tratamiento de tipo lógico y, más aún, automatizar los razonamientos. Por otro lado, es muy frecuente que el conocimiento en términos de implicaciones se pueda caracterizar con mucha menos información que el conocimiento en forma de retículo de conceptos.

Presentamos los resultados relacionados con las implicaciones de atributos considerando los componentes habituales de una lógica: lenguaje, semántica, sistema axiomático o inferencia sintáctica, y demostrador automático.

**Definición 2.3.15** *Dado un conjunto de atributos  $M$ , se define el lenguaje como*

$$\mathcal{L}_M = \{X \rightarrow Y \mid X, Y \subseteq M\}$$

*Las formulas en  $\mathcal{L}_M$  se denominan **implicaciones entre atributos**, o simplemente **implicaciones**. En una implicación  $X \rightarrow Y$ , llamamos **premisa** a  $X$  y **conclusión** a  $Y$ . Llamaremos **sistema implicacional** a cualquier subconjunto  $\Sigma \subseteq \mathcal{L}_M$ .*

Es costumbre, y nosotros lo haremos, omitir las llaves y las comas en las premisas y las conclusiones de las implicaciones. Del mismo modo, la

unión de conjuntos dentro de una implicación se denotará por yuxtaposición de los dos conjuntos. Por ejemplo, escribiremos  $m_1m_2 \rightarrow m_3$  en lugar de  $\{m_1, m_2\} \rightarrow \{m_3\}$  y  $AB \rightarrow CD$  en lugar de  $A \cup B \rightarrow C \cup D$ . El objetivo es reducir el tamaño de las fórmulas y simplificar su lectura.

### Semántica:

Una vez definido el lenguaje se va a introducir la semántica para darle un significado a las fórmulas del lenguaje.

**Definición 2.3.16** Sea  $\mathbb{K} = \langle G, M, I \rangle$  un contexto formal y  $A \rightarrow B \in \mathcal{L}_M$ . Decimos que el contexto  $\mathbb{K}$  es un **modelo** para  $A \rightarrow B$ , denotado por  $\mathbb{K} \models A \rightarrow B$ , si  $A^\downarrow \subseteq B^\downarrow$ , o equivalentemente,  $B \subseteq A^{\uparrow}$ .

Como es usual, esta notación se extiende a sistemas implicacionales: dado  $\Sigma \subseteq \mathcal{L}_M$ , la expresión  $\mathbb{K} \models \Sigma$  denota que  $\mathbb{K} \models A \rightarrow B$  para todo  $A \rightarrow B \in \Sigma$ .

I	$m_1$	$m_2$	$m_3$
$g_1$		×	×
$g_2$		×	
$g_3$	×	×	×

Tabla 2.4: Contexto Formal  $\mathbb{K} = \langle G, M, I \rangle$ .

**Ejemplo 2.3.17** Para el contexto formal  $\mathbb{K} = \langle G, M, I \rangle$  donde  $G = \{g_1, g_2, g_3\}$ ,  $M = \{m_1, m_2, m_3\}$  e  $I$  la relación binaria descrita en la Tabla 2.4, se tiene que  $\mathbb{K} \models \{m_1 \rightarrow m_3, \emptyset \rightarrow m_2\}$  mientras que  $\mathbb{K} \not\models m_2 \rightarrow m_3$ .

**Definición 2.3.18** Sea  $M$  un conjunto de atributos. Decimos que una implicación  $A \rightarrow B \in \mathcal{L}_M$  se **deriva semánticamente** de un sistema implicacional  $\Sigma \subseteq \mathcal{L}_M$ ,

denotado por  $\Sigma \models A \rightarrow B$ , si  $\mathbb{K} \models \Sigma$  implica  $\mathbb{K} \models A \rightarrow B$  para todo contexto formal  $\mathbb{K}$ .

Por otro lado, decimos que dos sistemas implicacionales  $\Sigma_1, \Sigma_2 \subseteq \mathcal{L}_M$  son **semánticamente equivalentes**, y lo denotaremos por  $\Sigma_1 \equiv \Sigma_2$ , si, para todo contexto formal  $\mathbb{K}$ , se cumple que  $\mathbb{K} \models \Sigma_1$  si y solo si  $\mathbb{K} \models \Sigma_2$ .

En resumen,  $\Sigma \models A \rightarrow B$  si todo modelo de  $\Sigma$  es modelo de  $A \rightarrow B$  y  $\Sigma_1 \equiv \Sigma_2$  si sus modelos coinciden, es decir, ambos conjuntos representan el mismo conocimiento.

**Ejemplo 2.3.19** Consideremos el conjunto de atributos  $M = \{m_1, m_2, m_3\}$ . Se tiene que  $\{m_1 \rightarrow m_2, m_1 \rightarrow m_3\} \equiv \{m_1 \rightarrow m_2 m_3\}$  porque, para todo contexto formal  $\mathbb{K} = \langle G, M, I \rangle$ , se cumple que  $\{m_2, m_3\}^\downarrow = \{m_2\}^\downarrow \cap \{m_3\}^\downarrow$  y, por tanto,

$$\{m_1\}^\downarrow \subseteq \{m_2\}^\downarrow \text{ y } \{m_1\}^\downarrow \subseteq \{m_3\}^\downarrow \quad \text{si y solo si} \quad \{m_1\}^\downarrow \subseteq \{m_2, m_3\}^\downarrow$$

### El sistema axiomático de Armstrong:

El sistema axiomático conocido como “**axiomas de Armstrong**” considera un esquema de un axioma y dos reglas de inferencia: sean  $A, B, C \subseteq M$ ,

$$\begin{array}{lll} [\text{Inc}] & \text{Inclusión:} & \vdash_{\mathcal{A}} AB \rightarrow A \\ [\text{Augm}] & \text{Aumentación:} & A \rightarrow B \quad \vdash_{\mathcal{A}} AC \rightarrow BC \\ [\text{Trans}] & \text{Transitividad:} & A \rightarrow B, B \rightarrow C \quad \vdash_{\mathcal{A}} A \rightarrow C \end{array}$$

La noción de derivación sintáctica se introduce del modo habitual:

**Definición 2.3.20** Una implicación  $\varphi \in \mathcal{L}_M$  se dice que se **deriva sintácticamente**, o se **infiere**, de un conjunto de implicaciones  $\Sigma \subseteq \mathcal{L}_M$ , denotado por  $\Sigma \vdash_{\mathcal{A}} \varphi$ , si existe una secuencia de implicaciones  $\varphi_1, \dots, \varphi_n \in \mathcal{L}_M$  tal que  $\varphi_n = \varphi$  y, para todo  $1 \leq i \leq n$ , se cumple alguna de las siguientes condiciones:

- $\varphi_i$  pertenece a  $\Sigma$ ,
- $\varphi_i$  es un axioma, o
- $\varphi_i$  se obtiene aplicando las reglas de inferencia descritas anteriormente a las implicaciones  $\{\varphi_j \mid 1 \leq j < i\}$ .

En este caso, se dice que la secuencia  $\{\varphi_i \mid 1 \leq i \leq n\}$  es una prueba o demostración para  $\Sigma \vdash_{\mathcal{A}} \varphi$ .

El origen de este sistema axiomático lo encontramos en [9] donde se utiliza para estudiar las propiedades de las *dependencias funcionales* en el *modelo relacional de Codd* [20]. En trabajos posteriores al mencionado [10,38,52,82] se han propuesto otros sistemas axiomáticos equivalentes a él, pero hemos optado por presentar el original. Por otro lado, hemos llamado al esquema de axioma “inclusión” aunque comúnmente es conocido como “reflexividad”. El motivo es que reservamos este nombre para otro esquema de axioma que encaja mejor con la idea matemática de reflexividad.

**Ejemplo 2.3.21** Sea  $M = \{m_1, m_2, m_3, m_4, m_5\}$ . La siguiente secuencia demuestra que  $\{m_1m_2 \rightarrow m_3m_4, m_2m_3 \rightarrow m_5\} \vdash_{\mathcal{A}} m_1m_2 \rightarrow m_5$ :

$$\begin{array}{ll}
 \varphi_1 = & m_1m_2 \rightarrow m_3m_4 \quad \text{Hipótesis.} \\
 \varphi_2 = & m_2m_3 \rightarrow m_5 \quad \text{Hipótesis.} \\
 \varphi_3 = & m_1m_2 \rightarrow m_2m_3m_4 \quad \text{Aplicando [Augm] a } \varphi_1 \text{ con } m_2. \\
 \varphi_4 = & m_2m_3m_4 \rightarrow m_4m_5 \quad \text{Aplicando [Augm] a } \varphi_2 \text{ con } m_4. \\
 \varphi_5 = & m_1m_2 \rightarrow m_4m_5 \quad \text{Aplicando [Trans] a } \varphi_3 \text{ y } \varphi_4. \\
 \varphi_6 = & m_4m_5 \rightarrow m_5 \quad \text{Por [Inc].} \\
 \varphi_7 = & m_1m_2 \rightarrow m_5 \quad \text{Aplicando [Trans] a } \varphi_5 \text{ y } \varphi_6.
 \end{array}$$

El siguiente teorema asegura que la derivación sintáctica y la semántica coinciden, es decir, todo lo que se puede derivar con el sistema axiomático

de Armstrong se puede derivar semánticamente (es correcto) y viceversa (es completo).

**Teorema 2.3.22 (Corrección y completitud)** *Sea  $M$  un conjunto finito de atributos,  $\Sigma \subseteq \mathcal{L}_M$  y  $A \rightarrow B \in \mathcal{L}_M$ . Entonces  $\Sigma \models A \rightarrow B$  si y solo si  $\Sigma \vdash_{\mathcal{A}} A \rightarrow B$ .*

### La Lógica de Simplificación:

Aunque el sistema axiomático de Armstrong se referencia en multitud de trabajos, en la práctica se usa para el estudio teórico de las implicaciones, y no para el desarrollo de aplicaciones/algoritmos. El problema intrínseco en él es que las demostraciones no son fácilmente automatizables. Por este motivo, en [79] se presentó la Lógica de Simplificación que es más adecuada para el razonamiento automático [78].

La Lógica de Simplificación considera la reflexividad como esquema de axioma y las siguientes reglas de inferencia denominadas fragmentación, composición y simplificación respectivamente.

[Ref]	Reflexividad:	$\vdash_S A \rightarrow A$
[Frag]	Fragmentación:	$A \rightarrow BC \vdash_S A \rightarrow B$
[Comp]	Composición:	$A \rightarrow B, C \rightarrow D \vdash_S AC \rightarrow BD$
[Simp]	Simplificación:	$A \rightarrow B, C \rightarrow D \vdash_S A(C-B) \rightarrow D$

En realidad, podemos quedarnos solamente con [Ref] y [Simp], porque [Frag] y [Comp] son reglas derivadas de las anteriores.

**Teorema 2.3.23 ([27])** *Sea  $M$  un conjunto finito de atributos. Para todo  $\Sigma \in \mathcal{L}_M$  y todo  $A \rightarrow B \in \mathcal{L}_M$ ,  $\Sigma \vdash_S A \rightarrow B$  si y solo si  $\Sigma \vdash_{\mathcal{A}} A \rightarrow B$ .*

**Corolario 2.3.24** *La Lógica de Simplificación es correcta y completa.*

Dada la equivalencia del teorema anterior, en adelante, omitiremos el subíndice y escribiremos simplemente  $\vdash$ .

La principal ventaja de la Lógica de Simplificación es que las reglas de inferencia se pueden considerar reglas de equivalencia que permiten eliminar redundancias, y éstas, a su vez, son suficientes para calcular todas las derivaciones (ver [78] para más detalles y demostraciones).

**Teorema 2.3.25 ([78])** *Las siguientes equivalencias se cumplen:*

1. *Eq. de Fragmentación [EqFr]:*  $\{A \rightarrow B\} \equiv \{A \rightarrow B-A\}$ .
2. *Eq. de Composición [EqCo]:*  $\{A \rightarrow B, A \rightarrow C\} \equiv \{A \rightarrow BC\}$ .
3. *Eq. de Simplificación [EqSi]:* Si  $A \cap B = \emptyset$  y  $A \subseteq C$  entonces

$$\{A \rightarrow B, C \rightarrow D\} \equiv \{A \rightarrow B, C-B \rightarrow D-B\}$$

Se observa que estas equivalencias, leídas de izquierda a derecha, eliminan la información redundante, lo que justifica el nombre de la lógica.

#### Razonamiento automático:

Para cada sistema implicacional, el sistema axiomático define un operador de clausura en  $2^M$  que vamos a llamar “cierre sintáctico”.

**Definición 2.3.26** *Sea  $M$  un conjunto de atributos y  $\Sigma \subseteq \mathcal{L}_M$ . Decimos que un conjunto  $X \subseteq M$  es **cerrado respecto**  $\Sigma$  si, para todo  $A \rightarrow B \in \Sigma$ , se tiene que  $A \subseteq X$  implica  $B \subseteq X$ .*

El conjunto de los cerrados respecto de  $\Sigma$  forma un sistema de clausura (ver Definición 2.2.2) y, por el Teorema 2.2.3, define un operador de clausura en  $\langle 2^M, \subseteq \rangle$ .

**Definición 2.3.27** Sea  $M$  un conjunto de atributos y  $\Sigma \subseteq \mathcal{L}_M$ . Para cada  $X \subseteq M$  se define el *cierre sintáctico de  $X$  respecto  $\Sigma$*  como:

$$X_{\Sigma}^{+} = \bigcap \{C \subseteq M \mid X \subseteq C \text{ y } C \text{ es cerrado respecto } \Sigma\}$$

El siguiente teorema es la base, no solo para calcular cierres, sino también para introducir un método de razonamiento automático.

**Teorema 2.3.28** Sea  $M$  un conjunto finito de atributos,  $\Sigma \subseteq \mathcal{L}_M$  y  $A, B \subseteq M$ . Las siguientes condiciones son equivalentes:

$$\Sigma \vdash A \rightarrow B \text{ si y solo si } B \subseteq A_{\Sigma}^{+} \text{ si y solo si } \{\emptyset \rightarrow A\} \cup \Sigma \vdash \emptyset \rightarrow B.$$

Como consecuencia directa del teorema anterior tenemos que:

$$X_{\Sigma}^{+} = \text{máx}\{Y \subseteq M \mid \Sigma \vdash X \rightarrow Y\}$$

Por otro lado, el método lineal, presentado en [78], para calcular el cierre de un conjunto  $X$  con respecto a  $\Sigma$ , o equivalentemente, para determinar si  $X \rightarrow Y$  se puede inferir de  $\Sigma$ , consiste en:

1. Añadimos  $\emptyset \rightarrow X$  a  $\Sigma$ . Esta implicación sirve de semilla y guía para el procedimiento.
2. Mientras sea posible, compara la guía con el resto de implicaciones y aplica la equivalencia que corresponda (Teorema 2.3.25) de entre las siguientes:
  - **Eq. I:** Si  $B \subseteq A$  entonces  $\{\emptyset \rightarrow A, B \rightarrow C\} \equiv \{\emptyset \rightarrow AC\}$ .
  - **Eq. II:** Si  $C \subseteq A$  entonces  $\{\emptyset \rightarrow A, B \rightarrow C\} \equiv \{\emptyset \rightarrow A\}$ .
  - **Eq. III:** En otro caso,  $\{\emptyset \rightarrow A, B \rightarrow C\} \equiv \{\emptyset \rightarrow A, B-A \rightarrow C-A\}$ .
3. Cuando se alcance un punto fijo, si la guía es  $\emptyset \rightarrow A$ , entonces se tiene que  $X_{\Sigma}^{+} = A$ .





bajo coste. Estos subconjuntos reciben el nombre de *sistemas implicacionales completos*.

**Definición 2.3.30** Sea  $\mathbb{K} = \langle G, M, I \rangle$  un contexto formal. Decimos que un sistema implicacional  $\Sigma \subseteq \mathcal{L}_M$  es **completo para**  $\mathbb{K}$  si, para todo  $A \rightarrow B \in \mathcal{L}_M$ ,

$$\mathbb{K} \models A \rightarrow B \quad \text{si y solo si} \quad \Sigma \vdash A \rightarrow B.$$

Como consecuencia directa de la definición anterior se tiene que, para todo contexto formal  $\mathbb{K}$  y todo sistema implicacional completo  $\Sigma$ , ambos operadores de cierre coinciden, es decir,

$$X^{\downarrow\uparrow} = X_{\Sigma}^{+}, \quad \text{para todo } X \subseteq M.$$

El retículo de conceptos y el sistema implicacional, por tanto, representan el mismo conocimiento sobre el contexto.

Existen diferentes algoritmos para construir el retículo de conceptos a partir de un sistema implicacional completo. Por ejemplo, en [23,24], se proporcionan métodos, basados en el algoritmo del cierre descrito en la sección anterior, para calcular todos los cerrados y sus generadores minimales. Estos métodos usan la ‘guía’ para obtener los cerrados, y el resto de implicaciones para guiar la búsqueda y reducir el tamaño del problema.

Para el problema inverso, obsérvese que existen diferentes sistemas implicacionales completos para el mismo contexto y, por tanto, para el mismo retículo de conceptos. En [13], se expone un estudio exhaustivo sobre diferentes propiedades relacionadas con el tamaño de los sistemas implicacionales completos. Un sistema implicacional se considera **base** de un contexto si, además de ser completo, satisface algún criterio de minimalidad (con respecto al cardinal, con respecto al número de atributos que intervienen en él, etc.), es decir, donde la redundancia se ha reducido a su mínima expresión. Un problema ampliamente estudiado es cómo obtener una base del contexto

con la mínima cardinalidad, es decir, que el sistema implicacional tenga el menor número de implicaciones posible. Para un contexto finito, esta base siempre existe y es conocida como base de Duquenne-Guigues, base raíz (*stem basis*) o base canónica [46], y se construye a partir de la noción de *pseudo-cerrado* (también conocido como *pseudo-intent*) [40].

**Definición 2.3.31** Sea  $c$  un operador de clausura en  $\langle 2^M, \subseteq \rangle$ . Un conjunto  $A \subseteq M$  es *quasi-cerrado* si, para todo  $B \subseteq A$ , se tiene que  $c(B) \subseteq A$  o  $c(B) = c(A)$ .

Por otro lado, un conjunto quasi-cerrado  $A$  es *pseudo-cerrado* si  $c(A) \neq A$  y, para todo conjunto quasi-cerrado  $B \subsetneq A$ , se tiene que  $c(B) \subsetneq A$ .

Un conjunto es quasi-cerrado si es cerrado o es tal que, al añadirlo al sistema de clausura, se obtiene un nuevo sistema de clausura. Por otro lado, un conjunto pseudo-cerrado es un minimal de entre aquellos conjuntos quasi-cerrados que tienen el mismo cierre. Obsérvese que puede haber más de un minimal para cada cerrado. Un estudio más profundo sobre estas nociones se puede encontrar en [64].

La siguiente definición parte de un operador de cierre arbitrario. Este operador puede ser  $(-)^{\downarrow\uparrow}$  o  $(-)^{\uparrow}_{\Sigma}$  dependiendo de si partimos del contexto, del retículo de conceptos o de un sistema implicacional completo.

**Definición 2.3.32** Sea  $c$  un operador de clausura en  $\langle 2^M, \subseteq \rangle$ . La *base Duquenne-Guigues* para  $c$  se define como  $\Sigma_{DG} = \{A \rightarrow c(A) - A \mid A \text{ es pseudo-cerrado para } c\}$ .

La existencia y unicidad de la base Duquenne-Guigues está asegurada y, además, es la base de menor cardinalidad, es decir, que  $\Sigma \equiv \Sigma_{DG}$  implica  $|\Sigma_{DG}| \leq |\Sigma|$ . Esta última afirmación se debe a la siguiente proposición.

**Proposición 2.3.33 ([40])** Sea  $M$  un conjunto de atributos y  $\Sigma \subseteq \mathcal{L}_M$ . Si  $C$  es un conjunto pseudo-cerrado respecto  $(-)^{\uparrow}_{\Sigma}$ , entonces existe una implicación  $A \rightarrow B \in \Sigma$  tal que  $A \subseteq C$  y  $A^{\uparrow}_{\Sigma} = C^{\uparrow}_{\Sigma}$ .

En la bibliografía podemos encontrar también diversos métodos para calcular la base Duquenne-Guigues a partir de, o bien un contexto, o bien un retículo de conceptos, o bien un sistema implicacional arbitrario. El más conocido es una variante del algoritmo `NextClosure` que describiremos en la siguiente sección.

**Ejemplo 2.3.34** En [40], se considera un ejemplo que muestra los datos de países, concretamente se muestran los datos de 130 países y 6 atributos: *Group of 77*, *Non-aligned*, *LDC (Least Developed Countries)*, *MASC (Most Seriously Affected Countries)*, *OPEC (Organization of Petrol Exporting Countries)* y *ACP (African, Caribbean and Pacific Countries)*.

La base es la siguiente:

*OPEC* → *Group of 77*, *Non-aligned*

*MASC* → *Group of 77*

*Non-aligned* → *Group of 77*

*Group of 77*, *Non-aligned*, *MASC*, *OPEC* → *LLDC*, *ACP*

*Group of 77*, *Non-aligned*, *LLDC*, *OPEC* → *MASC*, *ACP*

□

#### 2.3.4. Cálculo del retículo de conceptos y de las implicaciones

Como es bien conocido, la complejidad del problema de generar todos los conceptos es exponencial en el tamaño del contexto de entrada y el problema de determinar el número de conceptos formales es  $\#P$ -completo [63]. En ese artículo, S.O. Kuznetsov and S.A. Obiedkov proponen una comparación significativa entre los principales algoritmos para generar los conceptos formales con respecto a la densidad/dispersión del contexto formal.

Uno de los temas principales de estudio es el orden en el que se generan los conceptos formales para evitar duplicidades y, de esta forma, realizar una

búsqueda más eficiente. La diferencia principal entre los distintos algoritmos de la bibliografía estriba en el orden en que se realiza la búsqueda, para evitar repeticiones, y en diferentes estrategias de poda para esta búsqueda que evitan cálculos innecesarios [12]. Algunos algoritmos sustituyen el coste en tiempo por coste en espacio de almacenamiento al utilizar complejas estructuras de datos.

Los trabajos iniciales en relación a este tema (ver los surveys [6, 12, 63]) los encontramos en: el algoritmo de Malgrange [72] que recorre el retículo por niveles y calcula el siguiente candidato basándose en los conceptos de niveles anteriores; el algoritmo de Chein [19] con una estrategia similar y el algoritmo de Norris [80] que introduce un enfoque incremental.

Cuando apareció el algoritmo `NextClosure` [40, 43] de Ganter, hubo un salto cualitativo al introducir el orden léxico que inserta el orden de la inclusión en orden total, eliminando repeticiones, y evita calcular el cierre de todos los posibles subconjuntos de atributos. Es digno de destacar que este algoritmo no requiere espacio exponencial porque no utiliza ninguna estructura de datos adicional.

Casi coetáneo a este algoritmo encontramos una familia de métodos, entre los que destaca el algoritmo de Bordat [15], que funcionan más rápido que el `NextClosure` a cambio de utilizar espacio exponencial para almacenar información de todos los conceptos. A éstos les sigue el algoritmo de Nourine [81] basado en el de Norris y con una estrategia similar a Bordat pero con una estructura de datos más eficiente para almacenar los conceptos calculados. `Titanic` [97] y `Upper-Neighbour` [70] usan un exhaustivo método de búsqueda de abajo hacia arriba para calcular los candidatos. `Titanic` además calcula las *reglas de asociación* (una generalización de las implicaciones de la que hablaremos en la siguiente sección) y `Upper-Neighbour` las relaciones de vecindad. `Add-Intent` [101], con un proceso incremental para construir el retículo, también proporciona las

relaciones de vecindad.

Otro salto cualitativo en la mejora de estos algoritmos se produjo con el *test de canonicidad* propuesto por S. Kuztnesov et al. en [61,63] en el algoritmo *Close-by-One* (CbO). Como se indica en el recopilatorio más reciente [6], es comúnmente aceptado que las nuevas variantes de CbO, como *FCbO* [58] e *In-Close* [5] son significativamente más rápidas que los algoritmos precedentes. Uno de los aspectos más relevantes es la modificación del test de canonicidad, por ejemplo, *FCbO* propone un test de fallo en la canonicidad heredada antes de calcular el cierre, esto es, un concepto es cerrado antes de que sus descendientes sean calculados. Mientras, *In-Close* añade una nueva forma de comprobar la canonicidad antes de que se cierre un concepto.

*Concept Explorer* (ConExp) [107] y *Lattice Miner* [66], entre otros, son herramientas de libre disposición que permiten dibujar los retículos de conceptos.

### El Algoritmo **NetxClosure**:

Presentamos a continuación el algoritmo *NetxClosure*, aunque no es el más eficiente, por su relevancia y su influencia en otros algoritmos, además de que el mismo algoritmo sirve para calcular los conceptos y la base Duquenne-Guigues.

Dado un contexto finito  $\mathbb{K} = \langle G, M, I \rangle$ , para explorarlo en búsqueda de los conceptos y/o la base Duquenne-Guigues, el algoritmo recorre  $2^M$  siguiendo un orden total conocido como “orden léxico”. Partiendo de una ordenación total de  $M$ , por ejemplo,  $M = \{m_1 < m_2 < \dots < m_7\}$ , identificamos cada subconjunto con su *mapa de bits*. Así, por ejemplo, el subconjunto  $A = \{m_1, m_3, m_4, m_6\}$  se puede describir como 1011010. De este modo podemos considerar que un subconjunto precede a otro si su mapa de bits, como número binario, es anterior.

Por simplicidad en la notación, asumiremos en adelante que  $M = \{1, \dots, m\}$ .

**Definición 2.3.35** Sea  $M = \{1, \dots, m\}$ . Definimos el **orden léxico** en  $2^M$  como sigue: para cada  $A, B \subseteq M$ , se tiene que  $A$  es **más pequeño léxicamente** que  $B$ , denotado por  $A < B$ , si  $A \neq B$  y el menor elemento que los distingue pertenece a  $B$ , es decir, si y solo si existe  $i \in B \setminus A$  tal que  $\{x \in A \mid x < i\} = \{x \in B \mid x < i\}$ .

**Ejemplo 2.3.36** En  $M = \{x \in \mathbb{N} \mid 1 \leq x \leq 9\}$ , consideremos los conjuntos  $A = \{1, 2, 5, 6\}$  y  $B = \{1, 2, 4, 6\}$ . Tenemos que  $A < B$  porque el menor elemento en que difieren es 4 y pertenece a  $B$ .

$M$	1	2	3	4	5	6	7	8	9
$A$	1	1	0	0	1	1	0	0	0
$B$	1	1	0	1	0	1	0	0	0

La principal característica del orden léxico, además de ser total, es que *subsume* al orden de la inclusión de conjuntos.

**Proposición 2.3.37** Sea  $M$  un conjunto finito y  $A, B \subseteq M$ . Si  $A \not\subseteq B$  entonces  $A < B$ .

Este orden léxico se puede describir también del siguiente modo:  $A < B$  si y solo si existe  $i \in M$  tal que  $A <_i B$  donde:

$$A <_i B \quad \text{si y solo si} \quad i \in B \setminus A \text{ y } \{x \in A \mid x < i\} = \{x \in B \mid x < i\}$$

Por otro lado, dado un operador de cierre  $c$  en  $\langle 2^M, \subseteq \rangle$ , y un subconjunto  $A \subseteq M$ , definimos  $A \oplus i$  como el cerrado:

$$A \oplus i := c(\{x \in A \mid x < i\} \cup \{i\})$$

$I$	1	2	3	4
$o_1$	×	×	×	
$o_2$	×		×	×
$o_3$		×	×	×
$o_4$	×			×

Tabla 2.5: Ejemplo de contexto formal

**Ejemplo 2.3.38** Consideremos el contexto descrito en la Tabla 2.5 y su operador de cierre  $(-)^{\downarrow\uparrow}: 2^M \rightarrow 2^M$ . Para el conjunto  $A = \{1, 3, 4\} \subseteq M$ , se tiene que  $A \oplus 2 = \{1, 2\}^{\downarrow\uparrow} = \{1, 2, 3\}$ .

El siguiente teorema es la base del algoritmo `NextClosure`.

**Teorema 2.3.39** Sea  $M$  un conjunto finito,  $c$  un operador de clausura en  $\langle 2^M, \subseteq \rangle$  y  $A \subseteq M$ . El menor conjunto cerrado mayor que  $A$  (llamado **siguiente cerrado**) con respecto al orden léxico es

$$A \oplus i \text{ donde } i = \max\{j \in M \mid A <_j A \oplus j\}.$$

Existen distintas versiones del algoritmo en función de cuáles son los datos de entrada (un contexto formal, un operador de cierre o un sistema implicacional) y cuál es la salida (el retículo de conceptos, el conjunto de los cerrados o la base Duquenne-Guigues). Comenzamos presentando la versión que cronológicamente fue la primera. Esta versión toma como entrada un operador de clausura y devuelve el conjunto de los cerrados. En esencia, el algoritmo comienza calculando el cierre de  $\emptyset$  y, a partir de ahí, calcula el resto de cerrados en orden léxico hasta llegar al mayor, que es el propio  $M$ .

Una característica interesante de este algoritmo es que, si bien tiene coste exponencial en el peor caso, en cada paso calcula el siguiente cerrado en tiempo polinómico.

**Función** NextClosure( $c, A$ )

**Entrada:** Un operador de clausura  $c$  en  $\langle 2^M, \subseteq \rangle$  donde  $M$  es un conjunto finito y un subconjunto  $A \subseteq M$ .

**Salida:** El siguiente cerrado a  $A$  en el orden léxico.

```

1 inicio
2   para cada  $i \in M$  en orden decreciente hacer
3     si  $i \in A$  entonces  $A := A \setminus \{i\}$ ;
4     en otro caso
5        $B := c(A \cup \{i\})$ ;
6       si  $A <_i B$  entonces devolver  $B$ ;
7   devolver  $M$ 

```

**Algoritmo 1:** Cerrados

**Entrada:** Un operador de clausura  $c$  en  $\langle 2^M, \subseteq \rangle$  donde  $M$  es un conjunto finito.

**Salida:** El conjunto  $S_c$  de todos los cerrados en orden léxico.

```

1 inicio
2    $A := c(\emptyset)$ ;  $S_c := \{A\}$ ;
3   repetir
4      $A := \text{NextClosure}(c, A)$ ;
5     Añade  $A$  a  $S_c$ 
6   hasta que  $A = M$ ;
7   devolver  $S_c$ 

```

**Ejemplo 2.3.40** Consideremos el contexto descrito en la Tabla 2.6 y su operador de cierre  $(-)^{\downarrow\uparrow} : 2^M \rightarrow 2^M$ . Para calcular los cerrados, el Algoritmo 1, en lugar de calcular el cierre de los 16 subconjuntos de  $M$ , realiza 7 cierres y devuelve

$$S_c = \{\{4\}, \{3, 4\}, \{1, 4\}, \{1, 2, 4\}, \{1, 2, 3, 4\}\}$$

Los cálculos realizados por el algoritmo se resumen en la Tabla 2.7.

La modificación de este algoritmo para que devuelva, además de los cerrados, la base Duquenne-Guigues se basa en que la unión del conjunto de



$I$	1	2	3	4
$o_1$	×	×		×
$o_2$	×			×
$o_3$			×	×

Tabla 2.6: Ejemplo de contexto formal

Candidato	Cerrado	¿Es el siguiente?	Salida
$\emptyset$	{4}	-	{4}
{3}	{3, 4}	Sí	{3, 4}
{2}	{1, 2, 4}	No	
{1}	{1, 4}	Sí	{1, 4}
{1, 3}	{1, 2, 3, 4}	No	
{1, 2}	{1, 2, 4}	Sí	{1, 2, 4}
{1, 2, 3}	{1, 2, 3, 4}	Sí	{1, 2, 3, 4}

Tabla 2.7: Traza del Algoritmo 1 aplicado al contexto descrito en la Tabla 2.6.

los cerrados y los pseudo-cerrados forma también un sistema de clausura. Combina, para ello, el uso del operador de cierre con el cierre sintáctico  $(-)_\Sigma^+$  obtenido a partir de las implicaciones ya calculadas.

---

**Algoritmo 2:** Cerrados e implicaciones
 

---

**Entrada:** Un operador de clausura  $c$  en  $\langle 2^M, \subseteq \rangle$  donde  $M$  es un conjunto finito.

**Salida:** El conjunto  $S_c$  de todos los cerrados en orden léxico y  $\Sigma_{DG}$  que es la base Duquenne-Guigues.

```

1 inicio
2    $S_c := \emptyset; \Sigma_{DG} := \emptyset; A := c(\emptyset);$ 
3   Añade  $A$  a  $S_c$ ;
4   si  $A \neq \emptyset$  entonces añade  $\emptyset \rightarrow A$  a  $\Sigma_{DG}$ ;
5   repetir
6      $A := \text{NextClosure}((-)_{\Sigma_{DG}}^+, A);$ 
7      $B := c(A);$ 
8     si  $A = B$  entonces añade  $A$  a  $S_c$ ;
9     en otro caso añade  $A \rightarrow B-A$  a  $\Sigma_{DG}$ ;
10  hasta que  $A = M$ ;
11  devolver  $S_c$  y  $\Sigma_{DG}$ 

```

---

**Ejemplo 2.3.41** Consideremos el contexto del Ejemplo 2.3.40 (descrito en la Tabla 2.6). El Algoritmo 2, además de calcular  $S_c$ , devuelve también la base Duquenne-Guigues, que es  $\Sigma_{DG} = \{\emptyset \rightarrow 4, 24 \rightarrow 1, 134 \rightarrow 2\}$ . Los cálculos realizados por el algoritmo se resumen en la Tabla 2.8.

Es un simple ejercicio adaptar el algoritmo anterior para que devuelva, no solo los cerrados, sino los conceptos formales de un contexto. Solo tendríamos que modificar las líneas 5 y 6 de la Función NextClosure para que calcule  $B := (A \cup \{i\})^\downarrow$  y  $C := B^\uparrow$ , y devuelva  $\langle B, C \rangle$ .

$C$	$A = C_{\Sigma_{DG}}^+$	¿Es el siguiente?	$c(A)$	$S_c$	$\Sigma_{DG}$
$\emptyset$	$\emptyset$	–	$\{4\}$	$\{4\}$	$\emptyset \rightarrow 4$
$\{3\}$	$\{3, 4\}$	Sí	$\{3, 4\}$	$\{3, 4\}$	
$\{2\}$	$\{2, 4\}$	Sí	$\{1, 2, 4\}$		$24 \rightarrow 1$
$\{2, 3, 4\}$	$\{1, 2, 3, 4\}$	No			
$\{1\}$	$\{1, 4\}$	Sí	$\{1, 4\}$	$\{1, 4\}$	
$\{1, 3\}$	$\{1, 4\}$	Sí	$\{1, 2, 3, 4\}$		$134 \rightarrow 2$
$\{1, 2\}$	$\{1, 2, 4\}$	Sí	$\{1, 2, 4\}$	$\{1, 2, 4\}$	
$\{1, 2, 3\}$	$\{1, 2, 3, 4\}$	Sí	$\{1, 2, 3, 4\}$	$\{1, 2, 3, 4\}$	

Tabla 2.8: Traza del Algoritmo 2 aplicado al contexto descrito en la Tabla 2.6.

## 2.4. Reglas de asociación

Una generalización de la noción de implicación es la de regla de asociación. Aunque no abordaremos este tema, presentamos aquí una breve introducción porque podemos encontrar algunos trabajos previos (los describiremos en el siguiente capítulo) de otros autores, relacionados con la temática de esta tesis, que hacen referencia a reglas de asociación. Tienen su origen en un área de investigación muy próxima al ACF: la *Minería de Datos*.

El ejemplo típico, que los investigadores en este área utilizan, es el conocido como '*la cesta de la compra*'. Un supermercado tiene una serie de productos a la venta que llaman *ítems*. Supongamos que el conjunto de ítems es  $\{i_1, i_2, i_3, i_4\}$ . Cada comprador se lleva a casa un subconjunto de ítems. Supongamos que hay 7 compradores y, por tanto 7 cestas de compra:  $D = \{t_1, \dots, t_7\}$  donde  $t_1 = \{i_1, i_2\}$ ,  $t_2 = \{i_1, i_3, i_4\}$ ,  $t_3 = \{i_1, i_2, i_3\}$ ,  $t_4 = \{i_1\}$ ,  $t_5 = \{i_2\}$ ,  $t_6 = \{i_2, i_3\}$  y  $t_7 = \{i_2, i_4\}$ . De esta información se desea extraer conocimiento del tipo "los compradores que se llevan tales productos compran también tal otro".

Toda esta información puede ser descrita por un contexto. Por ejemplo,

$D$	$i_1$	$i_2$	$i_3$	$i_4$
$t_1$	×	×		
$t_2$	×		×	×
$t_3$	×	×	×	
$t_4$	×			
$t_5$		×		
$t_6$		×	×	
$t_7$		×		×

Tabla 2.9: Conjunto de *cestas de la compra*.

en este caso, el de la Tabla 2.9. Y, por tanto, el conocimiento al que nos referimos es el que se recoge en términos de implicaciones de atributos: ‘todo comprador que se lleva los productos  $i_1$  e  $i_4$  se lleva también el producto  $i_3$ ’. Sin embargo, el vendedor puede estar interesado en implicaciones que no se cumplen en todos los casos (todos los objetos), aunque sí en la mayoría. Por ello, se añaden a las implicaciones de atributos medidas estadísticas para descubrir/describir regularidades entre atributos (ítems en las cestas). La introducción formal de la noción de regla de asociación se debe a Agrawal et al. [3]. La presentamos a continuación manteniendo la terminología de ACF utilizada en las secciones anteriores.

**Definición 2.4.1** Dado un conjunto finito  $M$ , una **regla de asociación** es una expresión del tipo  $X \rightarrow Y$  donde  $X, Y \subseteq M$  y cumplen que  $X \cap Y = \emptyset$ .

Dado un contexto  $\mathbb{K} = \langle G, M, I \rangle$ , definimos el **soporte** y la **confianza** de la regla  $X \rightarrow Y$  en  $\mathbb{K}$ , respectivamente, como

$$\text{sop}(X \rightarrow Y) = \frac{|(X \cup Y)^\downarrow|}{|G|} \quad \text{y} \quad \text{conf}(X \rightarrow Y) = \frac{|(X \cup Y)^\downarrow|}{|X^\downarrow|}.$$

Obviamente, ambas medidas, soporte y confianza, pertenecen al intervalo

unidad y una implicación de atributos es una regla de asociación con confianza 1. Por ejemplo, la implicación  $i_1 i_4 \rightarrow i_3$  tiene soporte  $\frac{1}{7}$  y confianza 1. Como ejemplo de regla de asociación que no es implicación de atributos de  $\mathbb{K}$  tenemos a  $i_3 \rightarrow i_2$  que tiene soporte  $\frac{2}{7}$  y confianza  $\frac{2}{3}$ .

No tiene mucho sentido calcular el soporte y la confianza de todas las reglas de asociación ya que una regla de asociación con soporte y/o confianza muy pequeña puede no tener ningún interés. Para establecer qué reglas son *interesantes* se suelen utilizar diversas medidas de significancia e interés. Las restricciones más conocidas consisten en establecer umbrales mínimos para soporte y confianza. [28] La mayor parte de los trabajos en esta área se centran en definir medidas de interés/significancia y en diseñar algoritmos que eficientemente encuentren las reglas de asociación interesantes sin calcular el soporte y la confianza de las no interesantes.

## Capítulo 3

# Atributos negativos.

## Planteamiento del problema

Como comentamos en la introducción, el objetivo general de esta tesis es establecer un marco global donde desarrollar los fundamentos que permitan diseñar herramientas para extraer, representar y manipular (razonar) conocimiento extraído, no solo de la información positiva, sino también de la negativa. En este capítulo presentamos los antecedentes, tanto en minería de datos como en ACF, con la finalidad de enmarcar claramente los objetivos de la tesis dentro de la bibliografía existente y el estado actual del área de investigación.

Las herramientas clásicas de ACF permiten considerar, como parte de nuestro conocimiento, negaciones de implicaciones de atributos. Es decir, permiten afirmar que una determinada implicación no es cierta, o bien desde el punto de vista semántico (no se satisface en el contexto), o bien desde el punto de vista sintáctico (no se puede derivar o inferir de nuestra base de implicaciones). Sin embargo, como el siguiente ejemplo muestra, esto puede corresponder a dos situaciones bien diferentes en relación con la ausencia

de propiedades, o negación de atributos, en los objetos del contexto.

**Ejemplo 3.0.1** La base Duquenne-Guigues obtenida de la tabla 3.1 es  $\{e \rightarrow bc, d \rightarrow c, bc \rightarrow e, a \rightarrow b\}$ .

$I$	$a$	$b$	$c$	$d$	$e$
$o_1$		×	×		×
$o_2$	×	×			
$o_3$		×	×		×
$o_4$			×	×	

Tabla 3.1: Un contexto formal.

Las implicaciones  $b \rightarrow c$  y  $b \rightarrow d$  no se cumplen en la Tabla 3.1 y además no se pueden derivar de la base usando el sistema de inferencia. Sin embargo, ambas implicaciones se corresponden con situaciones diferentes.

**Caso A:** Se tiene que algunos objetos tienen los atributos  $b$  y  $c$  (objetos  $o_1$  y  $o_3$ ) mientras que otros objetos ( $o_2$ ) tienen el atributo  $b$  pero no tienen el  $c$ . El conjunto de objetos que tienen el atributo  $b$  puede ser dividido en función de la presencia o ausencia del atributo  $c$ .

**Caso B:** Cualquier objeto que tiene el atributo  $b$  no tiene el atributo  $d$ , es decir, la presencia del atributo  $b$  implica la ausencia del atributo  $d$ . Más adelante usaremos la expresión  $b \rightarrow \bar{d}$  para denotar esta situación y diremos que se trata de una implicación de atributos mixta o con atributos negativos.

En resumen, cuando una implicación no se cumple en el contexto y no puede ser derivada usando los axiomas de Armstrong, necesariamente viene caracterizada en uno de los dos casos mencionados en el ejemplo anterior. La

única manera de capturar separadamente estas dos situaciones en nuestro conocimiento es enriquecer la expresividad del lenguaje usado.

Hay que diferenciar claramente entre negación de implicaciones e implicaciones mixtas. El problema de la negación de implicaciones ya está resuelto en el tratamiento clásico mientras que las implicaciones mixtas, es decir, las distinciones entre los casos A y B es lo que centra nuestro trabajo. El objetivo es aportar conocimiento a través del enriquecimiento de las implicaciones.

Necesitamos un entorno más general para tratar con este tipo de información. Como primer paso, se introduce una notación extendida que permita considerar la negación de atributos. A partir de ahora, el conjunto de atributos se denota por  $M$ , y sus elementos por la letra  $m$ , posiblemente con subíndices. Esto es, el carácter en minúscula  $m$  se reserva para los atributos positivos. Se usará  $\bar{m}$  para denotar la negación del atributo  $m$  y  $\bar{M}$  para denotar el conjunto  $\{\bar{m} \mid m \in M\}$  cuyos elementos se denominarán atributos negativos.

Los elementos arbitrarios en  $M \cup \bar{M}$  se denotarán con las primeras letras del alfabeto:  $a, b, c$ , etc. y  $\bar{a}$  denotará al opuesto de  $a$ : dado que el símbolo  $a$  podría representar un atributo positivo o negativo, si  $a = m \in M$  entonces  $\bar{a} = \bar{m}$  y si  $a = \bar{m} \in \bar{M}$  entonces  $\bar{a} = m$ .

Las letras mayúsculas  $A, B, C \dots$  denotan los subconjuntos de  $M \cup \bar{M}$ . Si  $A \subseteq M \cup \bar{M}$ , entonces  $\bar{A}$  denota el conjunto de los opuestos de los atributos  $\{\bar{a} \mid a \in A\}$ .

Para comenzar por la idea básica, habría que analizar la expresión lógica "Para todo  $x$ ,  $P(x)$  implica  $Q(x)$ ". Como es bien conocido, esta expresión es equivalente a "para todo  $x$ , o no es cierto  $P(x)$  o es cierto  $Q(x)$ ". Será falsa si existe un  $x$  con " $P(x)$  y no  $Q(x)$ ". Ahora bien, podría suceder que " $P(x)$  y no  $Q(x)$ " se diera en todos los  $x$  o sólo en algunos.

Los casos en los que la expresión "Para todo  $x$ ,  $P(x)$  implica  $Q(x)$ " es



falsa se pueden diferenciar en:

- Caso A: Existe algún  $x$  en el que  $P(x)$  es cierto y  $Q(x)$  también, y algún otro  $x'$  en el que  $P(x')$  es cierto y  $Q(x')$  no.
- Caso B: En todos los  $x$  en los que  $P(x)$  es cierto, se cumple que  $Q(x)$  no lo es.

Este último caso, el caso B, es el que centra nuestro estudio. Buscamos discriminar el caso B del caso A.

## 3.1. Antecedentes en Minería de Datos

### 3.1.1. Reglas de asociación

Tradicionalmente, las implicaciones y nociones similares se han construido usando la información positiva. Comenzando en el área de la minería de datos con la teoría de las reglas de asociación y continuando en el campo del ACF, algunos autores han propuesto la minería de implicaciones (reglas de asociación en minería de datos) con atributos positivos y negativos. En Mannila et al. [74], se introdujo un campo extendido para enriquecer las reglas de asociación, considerando la negación, conjunción y disyunción. Dichas reglas con atributos negados fueron consideradas también en Agrawal y Srikant [3] para representar información mixta: “Si compramos caviar, entonces no compramos atún”.

Estos trabajos preliminares inspiraron el actual progreso de las técnicas de minería de datos para extraer reglas de asociación con atributos positivos y negativos.

Los trabajos incluyen tres niveles:

1. Definición de regla de asociación (ver definición 2.4.1). Todas las definiciones tienen asociadas las medidas de soporte y confianza, si bien algunos autores añaden otras medidas relacionadas.
2. Definición de qué reglas de asociación son relevantes (criterios estadísticos, criterios que dependen del marco específico de trabajo, ...). Por lo general se establecen unos soportes y confianzas mínimos para delimitar la relevancia, aunque existen diversas variantes.
3. Algoritmos de búsqueda que incluyen estrategias y podas. Estos procesos se suelen hacer en dos fases: calcular los conjuntos de atributos frecuentes y generar las reglas de asociación.

Para realizar estas tareas nos encontramos con el problema de que la introducción de los atributos negativos en las reglas de asociación invalida algunas herramientas de poda importantes que se usan para restringir el espacio de búsqueda, que son las que garantizan la eficiencia en los algoritmos clásicos (que solo usan atributos positivos).

### 3.1.2. Generalizaciones (negación)

En la bibliografía, una vez conocida la definición de regla de asociación positiva válida (Definición 2.4.1), se introduce la regla de asociación negativa válida.

Sobre la definición, distinguimos dos casos en función de cómo se define la negación de un conjunto de atributos [28] (para evitar confusiones usaremos diferente notación para cada uno de ellos):

1. Interpretación disyuntiva: La negación de un conjunto de atributos se interpreta como la disyunción de la negación de los atributos, en el sentido clásico, es decir  $\neg A = \bigvee_{a \in A} \neg a$ . Sea  $M$  el conjunto de atributos y  $A, B$  subconjuntos con  $A \cap B = \emptyset$ . Para cada regla de asociación

positiva  $A \rightarrow B$ , se pueden considerar tres negativas,  $A \rightarrow \neg B$ ,  $\neg A \rightarrow B$  y  $\neg A \rightarrow \neg B$  [2, 105, 108]. En [108], Yuan et al. consideran que  $\neg A \rightarrow \neg B$  es equivalente a  $B \rightarrow A$  por su interpretación disyuntiva, lo cual es cierto cuando la confianza es 1, pero no es cierto en general, como se puede comprobar en el ejemplo 3.1.1.

Para un número de atributos  $n$ , el número posible de reglas de asociación positivas es  $3^n - 2^{n+1} + 1$ , mientras que el número de reglas de asociación negativas es el triple.

2. Interpretación conjuntiva: La negación de un conjunto de atributos se interpreta como la conjunción de la negación de los atributos, es decir  $\bar{A} = \bigwedge_{a \in A} \bar{a}$ . Esto equivale a ampliar el conjunto de atributos  $M$  añadiendo cada atributo negado, es decir,  $M \cup \bar{M}$  [7, 18, 106]. La única restricción a tener en cuenta a la hora de calcular las reglas de asociación positivas es que en el subconjunto de atributos no aparezcan un atributo y su negado. Para un número de atributos  $n$ , sin tener en cuenta un nivel de confianza mínimo, el número posible de reglas de asociación es  $5^n - 2(3^n) + 1$ , que es superior al de la definición anterior pero permite una implementación más sencilla y, dado un nivel de confianza mínimo, una poda basada en la denominada "propiedad del cierre del soporte descendente". Por dicho motivo, el uso de esta definición se aconseja en bases de datos pequeñas bajo el enfoque soporte-confianza clásico.

**Ejemplo 3.1.1** Consideremos el conjunto de atributos o ítems  $\{i_1, i_2, i_3, i_4\}$  y el conjunto de tuplas  $D = \{t_1, \dots, t_7\}$ , donde  $t_1 = (i_1, i_2)$ ,  $t_2 = (i_1, i_3, i_4)$ ,  $t_3 = (i_1, i_2, i_3)$ ,  $t_4 = (i_1)$ ,  $t_5 = (i_2)$ ,  $t_6 = (i_2, i_3)$  y  $t_7 = (i_2, i_4)$  (Véase la Tabla 3.2)

En consideración a la interpretación disyuntiva observamos que para la regla de asociación  $i_1 \rightarrow \neg\{i_2, i_3\}$ , su soporte es  $\text{sop}(i_1 \rightarrow \neg\{i_2, i_3\}) = 3/7$  y su confianza

$I$	$i_1$	$i_2$	$i_3$	$i_4$
$t_1$	×	×		
$t_2$	×		×	×
$t_3$	×	×	×	
$t_4$	×			
$t_5$		×		
$t_6$		×	×	
$t_7$		×		×

Tabla 3.2: Conjunto de tuplas.

$$\text{conf}(i_1 \rightarrow \neg\{i_2, i_3\}) = 3/4.$$

Observamos que las reglas  $\neg A \rightarrow \neg B$  y  $B \rightarrow A$  pueden tener soporte y confianza diferentes, por ejemplo, para la regla de asociación  $\neg i_1 \rightarrow \neg\{i_2, i_3\}$  el soporte es  $2/7$  y la confianza es  $2/3$ , mientras que para la regla  $i_2, i_3 \rightarrow i_1$  el soporte es  $1/7$  y la confianza es  $1/2$ .

Si tenemos en cuenta ahora, para la regla  $i_1 \rightarrow \overline{i_2}, \overline{i_3}$ , la interpretación conjuntiva, el soporte es  $1/7$  y la confianza  $1/4$ . Al usar interpretación conjuntiva, para las reglas con confianza 1 se puede usar la regla de inferencia de fragmentación, sin embargo, esto no es cierto en general para reglas de asociación con confianzas menores. Por ejemplo, para la regla de asociación  $i_1 \rightarrow \overline{i_2}, \overline{i_3}$ , se tiene que su fragmentación en  $i_1 \rightarrow \overline{i_2}$  y en  $i_1 \rightarrow \overline{i_3}$  ambas tienen soporte  $2/7$  y confianza  $1/2$ .

Con respecto a lo que se considera relevante, algunos autores limitan la búsqueda a algunos patrones concretos, por ejemplo el caso de las reglas de asociación excluyentes [4], denotadas  $S_1 \cup \overline{S_2} \rightarrow S_3$ , donde  $S_1, S_2$  y  $S_3 \subseteq M$ , es decir, si en una tupla se tienen los atributos del subconjunto  $S_1$  y no se tienen los del subconjunto  $S_2$ , entonces se tienen los atributos del subconjunto  $S_3$ .

La mayor parte de estos artículos se centran en los algoritmos en sí, dando diferentes estrategias de poda. Estas estrategias dependen de que se consideren interpretaciones disyuntivas o conjuntivas.

### Grupo 1

Aggarwal y Yu [2] evaluaron la correlación entre los subconjuntos de atributos definiendo la “fuerza colectiva”, una medida aparte del soporte y la confianza, para usarla en el algoritmo que identifica los denominados “conjuntos fuertemente colectivos”.

En el trabajo de Yuan et al. [108] se presenta un algoritmo para extraer reglas de asociación negativas. Una de las ventajas planteadas es el evitar recorridos innecesarios sobre la base de datos, calculando el soporte y la confianza de las reglas de asociación negativas en base a soportes y confianzas positivos. Son los primeros en tener en cuenta que, ante atributos con dos valores, la negación de uno implica la aparición del otro, lo cual es tenido en cuenta en la poda de su algoritmo. Este algoritmo se divide en dos partes, calculando las reglas positivas con el algoritmo Apriori [3] y prediciendo, según sus propias palabras, posibles reglas negativas interesantes basándose en la taxonomía y la similitud que definen entre los atributos positivos.

En Wu et al. [105] presentan un algoritmo basado en Apriori. Se centra en las medidas de dependencias de las reglas, en particular usa el argumento de Piatetsky-Shapiro [84]. Este algoritmo presenta problemas debido a que no cumple la propiedad de cierre descendente, no pudiendo generarse todas las reglas de asociación negativas.

En [33] se discute sobre la medida de calidad denominada  $M_{GK}$ , la cual se definió en [105], describiéndose su potencial para extraer reglas de asociación con negaciones relevantes y que ha sido ampliamente descrito en [77].

## Grupo 2

Los primeros casos prácticos realizados iban encaminados hacia la investigación de la “cesta de la compra” de usuarios relacionándose las reglas de asociación con correlaciones (Brin et al. [18]) y con reglas de dependencia (Silverstein et al. [96]) usando ambos el modelo del test de la Chi cuadrado para detectar dependencias entre atributos dentro de un subconjunto de ellos.

Antonie y Zaiane [7,8] proponen un algoritmo, basado en Apriori, para extraer simultáneamente reglas de asociación positivas y una subclase de negativas, usando la correlación (coeficiente de Pearson) entre conjuntos de atributos para evitar cálculos repetidos y obtener solo las más representativas. Compara su algoritmo con los trabajos de Wu et al. [105] y Teng et al. [100]. Su enfoque tiene problemas similares a los planteados en [105].

Existen otros trabajos sobre algoritmos para obtener reglas de asociación mixtas como el propuesto por Cornelis et al. [28]. Dicho algoritmo está basado en el algoritmo Apriori y permite encontrar todas las reglas de asociación válidas, positivas y negativas, con respecto al soporte y la confianza. Con este algoritmo se pretende corregir las deficiencias detectadas en los algoritmos de [2] y [18].

Han y Beheshti [50,51] formalizaron la idea de [7] y definieron una regla negativa válida  $A \rightarrow \bar{B}$  (las definiciones para  $\bar{A} \rightarrow B$  y  $\bar{A} \rightarrow \bar{B}$  son análogas).

Con diferentes medidas, existen otros trabajos que presentan nuevos algoritmos de poda, la mayoría basados en el algoritmo Apriori, como los estudiados en [31,34,59,89,102].

Al considerar la negación de atributos, algunos problemas crecen: la densidad de los datos se hace mayor, la complejidad puede incrementarse significativamente y algunas técnicas que usan estrategias basadas en el

reconocimiento de atributos no podrían aplicarse. Los atributos negativos no cumplen la propiedad de cierre ascendente de los conjuntos de atributos frecuentes, por lo cual los métodos de poda se ven afectados por el alto soporte que tienen los atributos negativos.

Existen otros trabajos hacia los que se extienden las reglas de asociación negativas, caso de las reglas de asociación espaciales estudiadas por Sharma et al. [95] y que hacen referencia a atributos específicos que indican lugares.

### **3.2. Antecedentes en el Análisis de Conceptos Formales**

En la sección anterior hemos analizado el “estado de la cuestión” en la consideración de atributos negativos y negaciones en relación con la noción de regla de asociación. En ACF, si bien la noción de regla de asociación y, en particular, de implicación, tiene un rol importante, disponemos de un marco formal mucho más rico, lo que le dota de un potencial mucho mayor. Precisamente por ello, su extensión para considerar atributos negativos requiere de un mayor desarrollo teórico. Nos centramos ahora en los antecedentes que podemos encontrar en esta área.

De por sí, el ACF está diseñado o ideado para tratar información positiva, al igual que los entornos de la minería de datos y el descubrimiento del conocimiento. Las reglas de asociación y las implicaciones se construyen usando solamente información positiva, sin embargo, debido a los requerimientos derivados de situaciones en el mundo real, se han realizado algunos intentos para representar esta información negativa.

### 3.2.1. Negación de atributos y/o negación de conceptos

La incorporación de información negativa se puede hacer a dos niveles: considerando la negación de atributos o de los conceptos.

En esta tesis vamos a considerar la negación de los atributos. Al mismo nivel, Ganter y Wille [43, Página 60] sugieren cómo *dicotomizar* un contexto.

**Definición 3.2.1** Un contexto  $\mathbb{K} = \langle G, M, I \rangle$  es **dicotómico** si, para todo  $m_1 \in M$  existe  $m_2 \in M$  tal que  $\langle g, m_1 \rangle \in I$  si y solo si  $\langle g, m_2 \rangle \notin I$  para todo objeto  $g \in G$ .

**Ejemplo 3.2.2** La tabla de la Figura 3.1 muestra cómo para todo objeto  $g \in G$ ,  $\langle g, a \rangle \in I$  si y solo si  $\langle g, c \rangle \notin I$ . Igualmente, para todo objeto  $g \in G$ ,  $\langle g, b \rangle \in I$  si y solo si  $\langle g, d \rangle \notin I$ .

$I$	$a$	$b$	$c$	$d$
$o_1$		×	×	
$o_2$	×	×		
$o_3$		×	×	
$o_4$			×	×

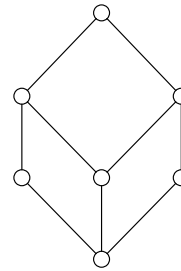


Figura 3.1: Contexto dicotómico y su retículo de conceptos asociado.

Sin embargo, el contexto de la Figura 3.2 no es dicotómico ya que, para el atributo  $b$ , no existe ningún  $m \in M$  que cumpla que, para todo  $g \in G$ ,  $\langle g, m \rangle \in I$  si y solo si  $\langle g, b \rangle \notin I$ .

**Definición 3.2.3** A partir de un contexto formal  $\mathbb{K} = \langle G, M, I \rangle$ , se puede definir un **contexto dual** como  $\overline{\mathbb{K}} = \langle G, \overline{M}, \overline{I} \rangle$  donde  $\overline{I}$  es la relación complementaria de  $I$  (es decir,  $\langle o, m \rangle \in \overline{I}$  si y solo si  $\langle o, m \rangle \notin I$ ).



$I$	$a$	$b$	$c$	$d$
$o_1$		×	×	
$o_2$	×			
$o_3$		×	×	
$o_4$			×	×

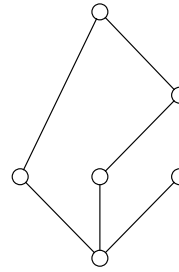


Figura 3.2: Contexto no dicotómico y su retículo de conceptos asociado.

Existen un significativo número de artículos en la bibliografía que siguen esta línea. Por ejemplo, en [77], Missaoui, Nourine y Renaud consideran la aposición del contexto y su dual, es decir  $\mathbb{K}|\overline{\mathbb{K}} = \langle G, M \cup \overline{M}, I \cup \overline{I} \rangle$  donde  $\langle g, \overline{m} \rangle \in \overline{I}$  si y solo si  $\langle g, m \rangle \notin I$ . Es directo ver que  $\mathbb{K}|\overline{\mathbb{K}}$  es dicotómico y que, si un contexto  $\mathbb{K}$  es dicotómico, entonces  $\mathfrak{B}(\mathbb{K}|\overline{\mathbb{K}})$  y  $\mathfrak{B}(\mathbb{K})$  son isomorfos.

**Ejemplo 3.2.4** En la Figura 3.3 podemos observar que el retículo de la aposición de un contexto dicotómico con su dual coincide con el retículo del contexto dicotómico original (Figura 3.1).

$I$	$a$	$b$	$c$	$d$	$\overline{a}$	$\overline{b}$	$\overline{c}$	$\overline{d}$
$o_1$		×	×		×			×
$o_2$	×	×					×	×
$o_3$		×	×		×			×
$o_4$			×	×	×	×		

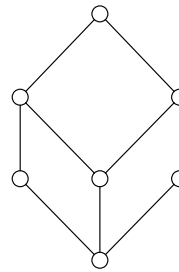


Figura 3.3: Aposición de un contexto dicotómico y su retículo de conceptos asociado.

Sin embargo, el retículo de conceptos en la Figura 3.4 cuyo contexto es la aposición de un contexto no dicotómico con su dual, no coincide con el retículo del

contexto no dicotómico original (Figura 3.2).

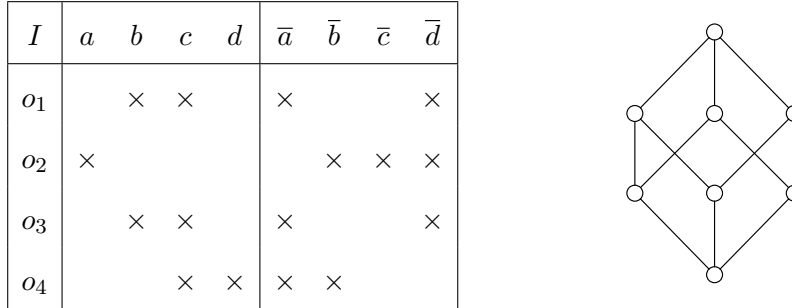


Figura 3.4: Aposición de un contexto no dicotómico y su retículo de conceptos asociado.

Una alternativa aparece en [62] donde, aplicando las técnicas de ACF al aprendizaje de maquinas (*Machine Learning*), Kuznetsov usa escalas dicotómicas  $k$ -dimensionales, como se definen en [43, Página 57]. El contexto se dicotomiza usando la escala  $\mathbb{D}_M = \sum_{m \in M} \mathbb{D}_m$ , que es el semiproducto<sup>1</sup> de las escalas dicotómicas  $\mathbb{D}_m = \langle \{0, 1\}, \{m, \bar{m}\}, \{\langle 1, m \rangle, \langle 0, \bar{m} \rangle\} \rangle$ . La escala  $\mathbb{D}_M$  es dicotómica y tiene  $2^{|M|}$  objetos y  $2^{|M|}$  atributos.

Entre los enfoques que consideran la negación a nivel del concepto, Wille [104] introduce dos diferentes versiones de la negación (llamadas *negación débil* y *oposición débil*) y caracteriza las estructuras que se obtienen. En un contexto formal  $\mathbb{K} = \langle G, M, I \rangle$ , la negación débil de un concepto  $\langle A, B \rangle$  se define como  $\langle A, B \rangle^\Delta = \langle (G \setminus A)^{\uparrow\downarrow}, (G \setminus A)^\uparrow \rangle$  y la oposición débil como  $\langle A, B \rangle^\nabla = \langle (M \setminus B)^\downarrow, (M \setminus B)^{\downarrow\uparrow} \rangle$ . Un retículo de conceptos equipado con estas dos operaciones se denomina un *álgebra de conceptos*. Wille introdujo además la noción de *retículo débilmente dicomplementado* para capturar la teoría ecuacional de álgebras de conceptos. Ganter y Kwuida

<sup>1</sup>El semiproducto de 2 contextos  $\mathbb{K}_1 = \langle G_1, M_1, I_1 \rangle$  y  $\mathbb{K}_2 = \langle G_2, M_2, I_2 \rangle$  se define en [43, Página 46] como  $\mathbb{K}_1 \boxtimes \mathbb{K}_2 = \langle G_1 \times G_2, (\{1\} \times M_1) \cup (\{2\} \times M_2), \nabla \rangle$  donde  $(g_1, g_2) \nabla (j, m)$  si y solo si  $\langle g_i, m \rangle \in I_i$  para  $i \in \{1, 2\}$ .

probaron que las álgebras de conceptos finitas distributivas son exactamente retículos débilmente dicomplementados finitos y distributivos [42].

Hay varias leyes clásicas que cabría esperar que una negación cumpliera. Estas leyes son, conocidas como las tres leyes primarias del pensamiento, *principium exclusi tertii*, *principium contradictionis* y *duplex negation affirmat*.

1. El *principium exclusi tertii* o principio del tercero excluido es un principio de lógica clásica según el cual la disyunción de una proposición y de su negación es siempre verdadera (una tercera cosa no se da), por ejemplo “La casa es roja o no lo es”. A veces se confunde con el principio de bivalencia, según el cual toda proposición o bien es verdadera o bien es falsa.
2. El *principium contradictionis* o principio de exclusión o contradicción, establece que la existencia de dos declaraciones contradictorias hace que no puedan ser verdaderas ambas a la vez. Las proposiciones “A es B” y “A no es B” se excluyen mutuamente con esta propiedad.
3. El *duplex negatio affirmat* o principio de afirmación de la doble negación, admite variantes según el ámbito donde se aplique. En lógica se puede dividir en introducción de la doble negación ( $A \vdash \neg\neg A$ ) y eliminación de la doble negación ( $\neg\neg A \vdash A$ ).

**Ejemplo 3.2.5** *El retículo con cuatro elementos y la negación descrita en la Figura 3.5 cumplen el principio del tercero excluido (es decir,  $x \vee \neg x = \top$  para todo  $x$ ), el de contradicción (es decir  $x \wedge \neg x = \perp$  para todo  $x$ ) que permite considerar falso todo aquello que implique una contradicción, y el de doble negación (es decir  $\neg\neg x = x$ , para todo  $x$ ).*

*Aunque el ejemplo de la Figura 3.5 cumple los tres principios, no tienen por qué cumplirse siempre de forma simultánea, como podemos ver en las Figuras 3.6, 3.7 y 3.8.*

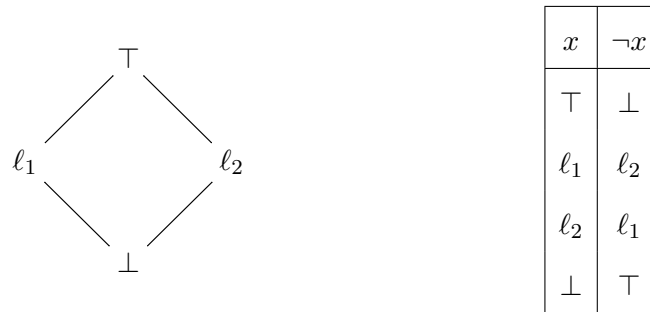


Figura 3.5: Retículo con negación que cumple los tres principios.

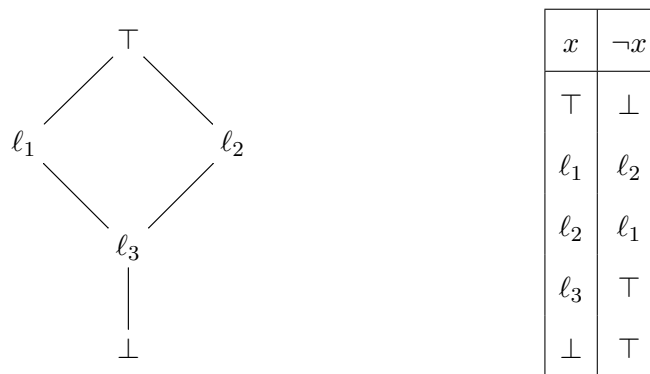


Figura 3.6: Retículo que cumple el principio del tercero excluido pero no los otros dos.

En [65], Kwuida, Tepavčević y Šešelja, prueban que no se cumplen necesariamente estos principios en la negación débil o en la oposición débil, y estudian los subconjuntos de álgebras de conceptos donde se pueden asegurar estas leyes.

Por otro lado, en [41], Ganter y Kwuida definen una negación a nivel de conceptos usando *retículos pseudocomplementados*, también conocidos como *p-álgebras*. En un retículo acotado, el pseudocomplemento de un elemento  $l_1$ , si existe, es el mayor elemento  $l_1^*$  tal que  $l_1 \wedge l_2 = \perp$  si y solo si  $l_2 \leq l_1^*$ . Es decir, si existe  $l_1^*$ , entonces  $l_1^* = \text{máx}\{x \mid l_1 \wedge x = \perp\}$ . Así, una p-álgebra es un retículo acotado donde cualquier elemento tiene un pseudocomplemento.

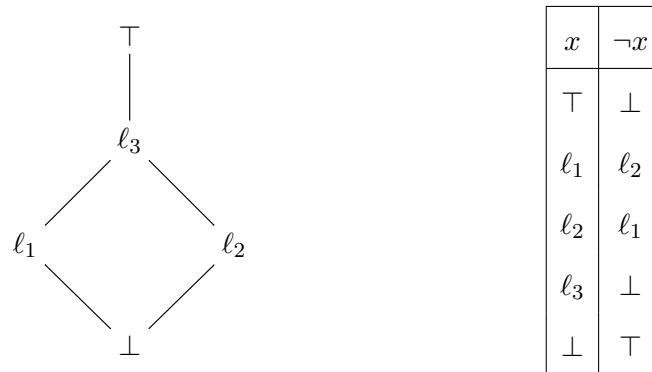


Figura 3.7: Retículo que cumple el principio de exclusión pero no los otros dos.

En un retículo de conceptos, el pseudocomplemento de un concepto  $\langle A, B \rangle$  es el concepto más general (cuando existe) que contradice  $\langle A, B \rangle$ . Así, un pseudocomplemento puede ser interpretado como la negación de un concepto.

### 3.2.2. Negación de atributos en las implicaciones

En adelante nos centraremos especialmente en el uso de reglas de asociación con confianza 1 que, en este marco, se llaman implicaciones entre atributos.

En el campo del ACF, algunos autores han propuesto la minería de implicaciones con atributos positivos y negativos desde la aposición del contexto con su negación  $(\mathbb{K}|\overline{\mathbb{K}})$  [17, 44].

Boulicaut et al. [17], aunque no mencionaron específicamente que introdujeran la consideración de los atributos negativos en el ACF, usaron bases de datos cuyos atributos eran bivaluados, motivo por el cual sus extensiones de métodos previos eran compatibles con el ACF. El coste del algoritmo Apriori en los métodos que consideraban atributos negativos era bastante elevado al usar la aposición de la base de datos con valores opuestos a la

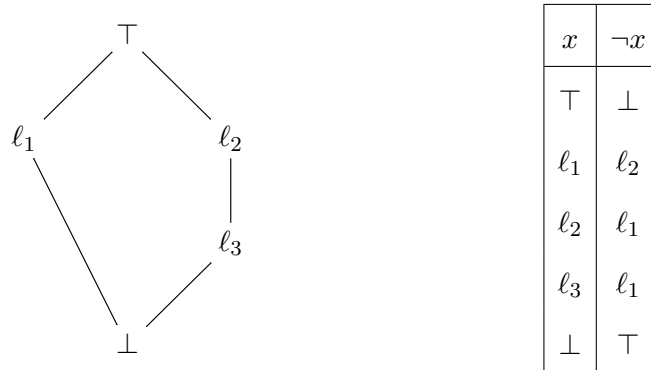


Figura 3.8: Retículo que cumple los principios de tercero excluido y de contradicción pero no la afirmación de la doble negación.

base de datos inicial.

En primera instancia se propone una aproximación para obtener algunas reglas de asociación con negaciones calculando solo el soporte y la confianza de los atributos positivos. Este método presenta algunos problemas de imprecisión cuando no se tienen datos al definir soportes y confianzas que descarten algunos subconjuntos positivos de atributos, lo cual ya fue tratado en [73].

Gasmi et al. [44] también proponen un algoritmo haciendo una adaptación del algoritmo PRINCE [49].

R. Missaoui et al. [75,76] proponen la minería de la información desde un contexto formal  $\mathbb{K}$  para obtener reglas de asociación mixtas exactas o implicaciones mixtas, es decir, con atributos positivos y negativos representando la presencia y la ausencia de atributos. Dado que es el trabajo más cercano a nuestra investigación, se van a detallar más específicamente sus resultados.

Se basa en la idea de [44], donde la primera aproximación se realiza mediante la aposición del contexto opuesto  $\overline{\mathbb{K}}$ . Trabajando con  $(\mathbb{K}|\overline{\mathbb{K}})$  se llega a un problema de crecimiento exponencial y, tal como muestra R. Missaoui et

al. en [77], las aplicaciones reales suelen tener datos dispersos en el contexto  $\mathbb{K}$  mientras que son más densos en  $\overline{\mathbb{K}}$  (o viceversa), y de esta manera “se genera una enorme cantidad de reglas no interesantes”.

Estos autores pretenden calcular las implicaciones que satisface  $\mathbb{K}|\overline{\mathbb{K}}$  a partir de las implicaciones de  $\mathbb{K}$  y  $\overline{\mathbb{K}}$ . Para ello prueban una serie de propiedades (que ellos llaman reglas de inferencia). En concreto, en [75] demuestran:

1. Para todo  $m \in M$ , se tiene que  $\mathbb{K}|\overline{\mathbb{K}} \models m\overline{m} \rightarrow M\overline{M} \setminus \{m\overline{m}\}$  con soporte 0.
2. Si  $\mathbb{K} \models Am \rightarrow n$  con soporte mayor que 0, entonces  $\mathbb{K}|\overline{\mathbb{K}} \models A\overline{n} \rightarrow \overline{m}$ .
3. Si  $\mathbb{K} \models Am \rightarrow M \setminus \{Am\}$  con soporte 0, entonces  $\mathbb{K}|\overline{\mathbb{K}} \models A \rightarrow \overline{m}$  con soporte mayor que 0.
4. Si en  $\mathbb{K}$ , un objeto  $g \in G$  cumple  $\langle g, m \rangle \in I$ , para todo  $m \in M$ , entonces  $\mathbb{K}|\overline{\mathbb{K}} \not\models A \rightarrow \overline{n}$  para cualquier  $A \subseteq M$  y  $n \in M$ .
5. Dado  $A \subseteq M \cup \overline{M}$ ,  $\mathbb{K}|\overline{\mathbb{K}} \models A \rightarrow \overline{m}$  si y solo si  $\mathbb{K}|\overline{\mathbb{K}} \models Am \rightarrow M\overline{M}$  con soporte 0.

En [75] se demuestra que obtener el conjunto de implicaciones mixtas a partir de las reglas puras positivas y negativas no es siempre posible. El siguiente ejemplo justifica esta afirmación.

**Ejemplo 3.2.6** *El conjunto de implicaciones entre atributos positivos y el conjunto de implicaciones entre atributos negativos de los contextos  $\mathbb{K}_1$  y  $\mathbb{K}_2$  presentados en la Figura 3.9 coinciden. La base Duquenne-Guigues de ambos contextos,  $\mathbb{K}_1$  y  $\mathbb{K}_2$ , es  $\{c \rightarrow abe, d \rightarrow abe\}$ . Las correspondientes bases de  $\overline{\mathbb{K}}_1$  y  $\overline{\mathbb{K}}_2$  también coinciden:  $\{\overline{a} \rightarrow \overline{cd}, \overline{b} \rightarrow \overline{cd}, \overline{e} \rightarrow \overline{cd}, \overline{bcde} \rightarrow \overline{a}, \overline{acde} \rightarrow \overline{b}, \overline{abcd} \rightarrow \overline{e}\}$ . En consecuencia:*

- $\mathbb{K}_1 \models A \rightarrow B$  si y solo si  $\mathbb{K}_2 \models A \rightarrow B$ .

- $\mathbb{K}_1 \models A \rightarrow B$  si y solo si  $\mathbb{K}_2 \models A \rightarrow B$ .

Sin embargo, los conjuntos de implicaciones que se satisfacen en la dicotomización de ambos contextos no coinciden, porque, por ejemplo,  $\mathbb{K}_1 | \mathbb{K}_1 \models abc\bar{d} \rightarrow \bar{e}$  mientras que  $\mathbb{K}_2 | \mathbb{K}_2 \not\models abc\bar{d} \rightarrow \bar{e}$ .

$\mathbb{K}_1$	$a$	$b$	$c$	$d$	$e$
$o_1$	×	×			
$o_2$	×				×
$o_3$	×	×	×		×
$o_4$	×	×		×	×
$o_5$		×			×

$\mathbb{K}_2$	$a$	$b$	$c$	$d$	$e$
$o_1$	×	×			
$o_2$	×				×
$o_3$	×	×	×		×
$o_4$	×	×		×	×
$o_5$		×			×
$o_6$	×	×			×

Figura 3.9: Contextos formales  $\mathbb{K}_1$  y  $\mathbb{K}_2$ .

Aunque por lo general, como se ha visto en el ejemplo anterior, extraer las implicaciones mixtas no es posible a partir de un conjunto de implicaciones positivas, estos autores desarrollaron nuevos métodos para extraer implicaciones entre atributos mixtos de un contexto reducido a partir de sus implicaciones positivas basándose en la noción de clave [76].

**Definición 3.2.7** Sea  $A \subseteq M \cup \bar{M}$ . Se dice que  $A$  es una **clave** de  $\mathbb{K} | \bar{\mathbb{K}}$  si y solo si tiene soporte 0.

**Ejemplo 3.2.8** El subconjunto de claves para la Tabla 3.3 formado solo por 2 atributos es  $a\bar{a}$ ,  $ac$ ,  $b\bar{b}$ ,  $c\bar{b}$ ,  $c\bar{c}$ ,  $c\bar{d}$ ,  $d\bar{d}$ ,  $\bar{a}\bar{b}$ ,  $\bar{a}\bar{d}$ ,  $\bar{b}\bar{d}$



$\mathbb{K} \overline{\mathbb{K}}$	$a$	$b$	$c$	$d$	$\bar{a}$	$\bar{b}$	$\bar{c}$	$\bar{d}$
$o_1$	×			×		×	×	
$o_2$		×		×	×		×	
$o_3$	×	×					×	×
$o_4$		×	×	×	×			

Tabla 3.3: Contexto formal  $\mathbb{K}|\overline{\mathbb{K}}$ 

Los autores añaden la siguiente propiedad a su sistema de inferencia [77].

**Propiedad 3.2.9** *Dado  $\mathbb{K}$ ,  $A \subseteq M \cup \overline{M}$  y  $x \in M \cup \overline{M}$ , entonces  $\mathbb{K}|\overline{\mathbb{K}} \models A \rightarrow x$  si y solo si  $\mathbb{K}|\overline{\mathbb{K}} \models A\bar{x} \rightarrow M\overline{M} \setminus A\bar{x}$  con soporte 0. Es decir  $A\bar{x}$  es clave en  $\mathbb{K}|\overline{\mathbb{K}}$  y todas las implicaciones pueden calcularse con el conjunto de claves.*

Es en [77] donde se describe un procedimiento para generar un sistema implicacional mínimo a partir del cual se pueden derivar todas las implicaciones mixtas del contexto usando los axiomas de Armstrong. En esta aproximación, el contexto  $\mathbb{K}$  no es necesariamente reducido y se adaptan algoritmos de hipergrafos [32] debido a su relación con el cálculo de claves. A continuación, proponen usar el algoritmo de Maier [71] para quedarse con un conjunto de implicaciones sin redundancias.

En relación con las aplicaciones, en [90], las implicaciones mixtas han sido utilizadas para el problema de detección de errores en bases de datos.

### 3.3. Planteamiento de objetivos de la tesis

Tras comprobar cómo el uso de los atributos negativos aún presenta un amplio campo por explorar, se ha buscado qué áreas pueden deparar aún re-

sultados que potencien el uso de los mismos para obtener más conocimiento a partir de la información inicial.

El objetivo general de la tesis es desarrollar un marco integral (desde los fundamentos hasta las aplicaciones) que permita tratar la información negativa de una manera uniforme y los objetivos concretos de la tesis son:

- Puesto que el desarrollo de la teoría del ACF se basa en la noción de conexión de Galois, estudiaremos cómo la incorporación de atributos negativos repercute en la conexión de Galois correspondiente.

En particular, introducimos una conexión de Galois generalizada que nos llevará a un nuevo tipo de retículo que recoge toda la información, tanto positiva como negativa.

Estudiaremos las propiedades de estos retículos y veremos que, a diferencia de lo que ocurre en el caso clásico, no todos los retículos se corresponden con un contexto.

El objetivo principal de este apartado será dar un teorema de caracterización de tales retículos.

- El estudio de las implicaciones mixtas (con atributos positivos y/o negativos) y de un sistema axiomático que explote las ventajas inherentes a la semántica de la negación.

El trabajo presentado por Missaoui et al. en [77] sigue estando basado en los axiomas de Armstrong. Buscaremos un sistema axiomático correcto y completo que aproveche las propiedades inherentes a la negación. Esto nos permitirá obtener bases de implicaciones con menor tamaño que las propuestas en [77].

- El diseño de herramientas de minería que permitan extraer conocimiento a partir de información positiva y negativa, haciendo uso del marco formal desarrollado.

Como contraposición a las soluciones propuestas en la bibliografía, que calculan los sistemas implicacionales mixtos usando las herramientas clásicas sobre  $\mathbb{K}|\overline{\mathbb{K}}$ , propondremos algoritmos específicos para obtener dichos sistemas directamente desde  $\mathbb{K}$ .

Igualmente se estudiarán los algoritmos clásicos para calcular conceptos formales y adaptarlos para encontrar los conceptos formales mixtos, y por tanto el retículo que forman. Dichos algoritmos serán estudiados para comprobar cómo evolucionan en tiempo de cómputo al introducir el cambio en el sistema axiomático.

- El estudio de algunas aplicaciones de este marco formal.

Principalmente se buscan aplicaciones en la ingeniería, si bien se realizarán estudios también en otras áreas que indiquen cómo la aplicación de los métodos de extracción de conocimiento que usan atributos mixtos producen mejores resultados que la aplicación de los métodos clásicos.

## Capítulo 4

# Retículos de conceptos mixtos

En el capítulo de preliminares, se introdujo la definición de retículo de conceptos (Definición 2.3.8) y la importancia de dicha noción dentro del ACF para obtener y representar conocimiento a partir de un contexto formal. En nuestro afán de establecer un marco general que considere la información positiva y negativa como alternativa a las soluciones parciales que encontramos en la bibliografía, en este capítulo generalizamos la conexión de Galois formada por los constructores de “extensión” y “comprensión”. El retículo obtenido a partir de dicha conexión de Galois será llamado retículo de conceptos mixtos. En este caso, a diferencia de lo que ocurre en el caso clásico, no todo retículo es isomorfo a un retículo de conceptos mixtos. La mayor parte de este capítulo está dedicada a caracterizar los que sí lo son.

En este punto conviene recordar la notación común en la teoría de retículos para facilitar su lectura (véase la Sección 2.1). Dado un retículo finito  $\mathbb{L} = \langle L, \vee, \wedge, \top, \perp \rangle$ , se denota su conjunto de átomos por  $\text{At}(\mathbb{L})$ , sus elementos  $\vee$ -irreducibles por  $J(\mathbb{L})$  (join-irreducibles) y sus elementos  $\wedge$ -irreducibles por  $M(\mathbb{L})$  (meet-irreducibles).

Dado un elemento  $\ell \in L$ , los conjuntos *superior* e *inferior* de  $\ell$  son  $[\ell) = \{x \in \mathbb{L} \mid \ell \leq x\}$  y  $(\ell] = \{x \in \mathbb{L} \mid x \leq \ell\}$  respectivamente.

También,  $\ell_A$  denota al conjunto de átomos inferiores o iguales a  $\ell$ ,  $(\ell] \cap \text{At}(\mathbb{L})$  y  $\ell^M$  denota el conjunto de elementos  $\wedge$ -irreducibles mayores o iguales a  $\ell$ , es decir,  $(\ell) \cap \text{M}(\mathbb{L})$ .

## 4.1. Considerando atributos negativos en la conexión de Galois

En esta sección introducimos los pilares básicos del nuevo marco algebraico generalizado que nos permitirá tratar de una manera homogénea tanto información positiva como negativa. Se necesita un entorno más general para tratar con este tipo de información. Extenderemos las definiciones de operadores de derivación, de los conceptos formales y, en el siguiente capítulo, de las implicaciones entre atributos.

**Definición 4.1.1** Sea  $\mathbb{K} = \langle G, M, I \rangle$  un contexto formal. Se definen los **operadores de derivación mixtos**  $\uparrow: 2^G \rightarrow 2^{M \cup \overline{M}}$  y  $\downarrow: 2^{M \cup \overline{M}} \rightarrow 2^G$  de la siguiente forma: para cada  $X \subseteq G$  y cada  $Y \subseteq M \cup \overline{M}$ ,

$$\begin{aligned} X^\uparrow = & \{m \in M \mid \text{para todo } g \in X : \langle g, m \rangle \in I\} \\ & \cup \{\overline{m} \in \overline{M} \mid \text{para todo } g \in X : \langle g, m \rangle \notin I\} \end{aligned} \quad (4.1)$$

$$\begin{aligned} Y^\downarrow = & \{g \in G \mid \text{para todo } m \in Y : \langle g, m \rangle \in I\} \\ & \cap \{g \in G \mid \text{para todo } \overline{m} \in Y : \langle g, m \rangle \notin I\} \end{aligned} \quad (4.2)$$

Antes de estudiar las propiedades de estos operadores, definimos los siguientes conjuntos para todo  $A \subseteq M \cup \overline{M}$ :

- $\text{Pos}(A) = A \cap M$
- $\text{Neg}(A) = \overline{A} \cap M$
- $\text{Tot}(A) = \text{Pos}(A) \cup \text{Neg}(A)$

Hay que notar que  $A = \text{Pos}(A) \cup \overline{\text{Neg}(A)}$  y  $\text{Pos}(A), \text{Neg}(A), \text{Tot}(A) \subseteq M$ .

Frecuentemente haremos referencia a subconjuntos de  $M \cup \overline{M}$  que no contienen un atributo y su negado. Diremos que estos conjuntos son consistentes. Si además de ello, todos los atributos aparecen en el conjunto (bien como atributos positivos o bien como negativos) diremos que es consistente completo. Introducimos formalmente estas definiciones a continuación.

**Definición 4.1.2** Sea  $M$  un conjunto de atributos. Diremos que un conjunto  $A \subseteq M \cup \overline{M}$  es **consistente** si  $\text{Pos}(A) \cap \text{Neg}(A) = \emptyset$ .

Dado un conjunto de atributos  $M$ , la familia de conjuntos consistentes se va a denotar por  $\text{Ctts}(M)$ , es decir,

$$\text{Ctts}(M) = \{A \subseteq M \cup \overline{M} \mid \text{Pos}(A) \cap \text{Neg}(A) = \emptyset\}$$

Si  $A \in \text{Ctts}(M)$ , entonces  $|A| \leq |M|$  y, en el caso particular en el que  $|A| = |M|$ , se tiene que  $\text{Tot}(A) = M$ . Esta situación conduce a la noción de conjunto consistente completo.

**Definición 4.1.3** Sea  $M$  un conjunto de atributos. Decimos que un conjunto  $A \subseteq M \cup \overline{M}$  es **consistente completo** si  $A \in \text{Ctts}(M)$  y  $\text{Tot}(A) = M$ .

Dado un conjunto de atributos  $M$ , la familia de conjuntos consistentes completos se va a denotar por  $\text{FC}(M)$ , es decir,

$$\text{FC}(M) = \{A \in \text{Ctts}(M) \mid \text{Tot}(A) = M\}$$

Obsérvese que  $\text{Tot}(M \cup \overline{M}) = \text{Pos}(M \cup \overline{M}) = \text{Neg}(M \cup \overline{M}) = M$  y, por tanto,  $M \cup \overline{M}$  no es consistente.

**Ejemplo 4.1.4** Consideremos el conjunto  $M = \{m_1, m_2, m_3\}$ . El subconjunto  $A = \{m_1, m_2, \overline{m_3}\} \subseteq M \cup \overline{M}$  es consistente completo porque  $\text{Pos}(A) \cap \text{Neg}(A) = \{m_1, m_2\} \cap \{m_3\} = \emptyset$  y  $\text{Tot}(A) = \{m_1, m_2, m_3\}$ .

Por otro lado, el conjunto  $\{m_1, \overline{m_2}\}$  es consistente pero no completo y el conjunto  $\{m_1, \overline{m_1}, m_2\}$  ni es consistente ni es completo.

Estas nociones no son nuevas y en [62] la noción de conjunto consistente completo se denomina conjunto completo no contradictorio. Las siglas Ctts y FC vienen de la traducción del término al inglés: “Consistent set” y “Full Consistent set”.

El siguiente teorema asegura que el nuevo par de operadores de derivación forma también una conexión de Galois:

**Teorema 4.1.5** *Sea  $\mathbb{K} = \langle G, M, I \rangle$  un contexto formal. El par de operadores de derivación  $(\uparrow, \downarrow)$  es una conexión de Galois entre  $(2^G, \subseteq)$  y  $(2^{M \cup \overline{M}}, \subseteq)$ .*

DEMOSTRACIÓN: Se necesita probar que, para todos los subconjuntos  $X \subseteq G$  e  $Y \subseteq M \cup \overline{M}$ ,

$$X \subseteq Y^\downarrow \quad \text{si y solo si} \quad Y \subseteq X^\uparrow$$

Primero, supongamos que  $X \subseteq Y^\downarrow$ . Para todo  $a \in Y$ , se distinguen 2 casos:

1. Si  $a \in \text{Pos}(Y)$ , existe  $m \in M$  con  $a = m$  y, para todo  $g \in X$ , ya que  $X \subseteq Y^\downarrow$ ,  $\langle g, m \rangle \in I$  y, de esta forma  $a = m \in X^\uparrow$ .
2. Si  $a \in \text{Neg}(Y)$ , existe  $m \in M$  con  $a = \overline{m}$  y, para todo  $g \in X$ , ya que  $X \subseteq Y^\downarrow$ ,  $\langle g, m \rangle \notin I$  y, de esta forma  $a = \overline{m} \in X^\uparrow$ .

Recíprocamente, supongamos que  $Y \subseteq X^\uparrow$  y  $g \in X$ . Para asegurar que  $g \in Y^\downarrow$ , se necesita probar que  $\langle g, a \rangle \in I$  para todo  $a \in \text{Pos}(Y)$  y  $\langle g, \overline{a} \rangle \notin I$  para todo  $a \in \text{Neg}(Y)$ , lo cual es directo por  $Y \subseteq X^\uparrow$ .  $\square$

Como consecuencia del teorema previo,  $\uparrow \circ \downarrow$  y  $\downarrow \circ \uparrow$  son operadores de clausura. Es más, como en el caso clásico, ambos operadores de clausura conforman dos retículos duales isomorfos.

**Definición 4.1.6** *Dado un contexto  $\mathbb{K} = \langle G, M, I \rangle$ , un conjunto de objetos  $X \subseteq G$  y un conjunto de atributos  $Y \subseteq M$ , decimos que el par  $\langle X, Y \rangle$  es un **concepto formal mixto** si  $X^\uparrow = Y$  e  $Y^\downarrow = X$ .*

Se define también la relación de orden entre los conceptos

$$\langle X_1, Y_1 \rangle \leq \langle X_2, Y_2 \rangle \text{ si y solo si } X_1 \subseteq X_2 \quad (4.3)$$

(o equivalentemente, si y solo si  $Y_1 \supseteq Y_2$ )

Denotaremos por  $\mathfrak{B}^\sharp(G, M, I)$  al retículo de los conceptos mixtos definido por la relación descrita anteriormente. Es fácil razonar que, en este retículo, de forma análoga a lo que sucede en el caso clásico, se tiene que:

$$\langle X_1, Y_1 \rangle \wedge \langle X_2, Y_2 \rangle = \langle X_1 \cap X_2, (Y_1 \cup Y_2)^{\downarrow\uparrow} \rangle \quad (4.4)$$

$$\langle X_1, Y_1 \rangle \vee \langle X_2, Y_2 \rangle = \langle (X_1 \cup X_2)^{\uparrow\downarrow}, Y_1 \cap Y_2 \rangle \quad (4.5)$$

En adelante, usaremos indistintamente los términos **retículo de conceptos mixtos** y **retículo mixto** para referirnos a  $\mathfrak{B}^\sharp(G, M, I)$ .

El siguiente lema, que caracteriza los casos extremos, procede directamente de la definición 4.1.1.

**Lema 4.1.7** *Sea  $\mathbb{K} = \langle G, M, I \rangle$  un contexto formal. Entonces  $\emptyset^{\uparrow} = M \cup \overline{M}$ ,  $\emptyset^{\downarrow} = G$  y  $(M \cup \overline{M})^{\downarrow} = \emptyset$ .*

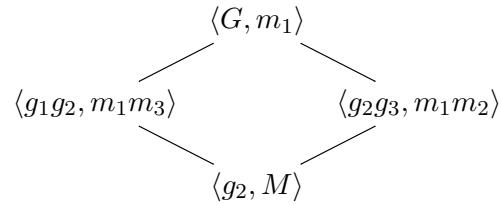
En el entorno clásico, el retículo de conceptos  $\mathfrak{B}(G, M, I)$  está acotado por  $\langle M^{\downarrow}, M \rangle$  y  $\langle G, G^{\uparrow} \rangle$ . Sin embargo, en este entorno generalizado, como consecuencia directa del lema anterior, las cotas inferior y superior de  $\mathfrak{B}^\sharp(G, M, I)$  son  $\langle \emptyset, M \cup \overline{M} \rangle$  y  $\langle G, G^{\uparrow} \rangle$  respectivamente.

**Ejemplo 4.1.8** *En el contexto de la Tabla 4.1, para el retículo de conceptos  $\mathfrak{B}(\mathbb{K})$  (véase Fig. 4.1), las cotas superior e inferior son  $\langle G, m_1 \rangle$  y  $\langle g_2, M \rangle$ , que son diferentes de las cotas de  $\mathfrak{B}^\sharp(\mathbb{K})$  (véase Fig. 4.2), que son  $\langle G, m_1 \rangle$  y  $\langle \emptyset, M \cup \overline{M} \rangle$ .*

**Observación 4.1.9** *Es obvio que, si en un contexto formal duplicamos una fila, el retículo mixto del contexto resultante es isomorfo al inicial. Por ello, a partir de ahora se va a asumir, sin pérdida de generalidad, que  $\{g_1\}^{\uparrow} = \{g_2\}^{\uparrow}$  implica que  $g_1 = g_2$ , es decir, que no hay dos filas iguales dentro de un contexto.*

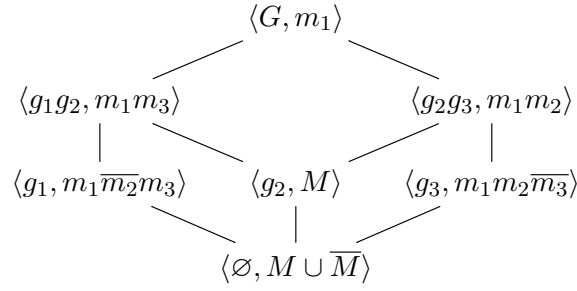


$I$	$m_1$	$m_2$	$m_3$
$g_1$	×		×
$g_2$	×	×	×
$g_3$	×	×	

Tabla 4.1: Contexto para el que  $\mathfrak{B}(\mathbb{K})$  y  $\mathfrak{B}^\sharp(\mathbb{K})$  tienen distintas cotasFigura 4.1: Retículo  $\mathfrak{B}(\mathbb{K})$  del contexto descrito en la Tabla 4.1.

Volviendo al Ejemplo 4.1.8, observamos que, si eliminamos del contexto el atributo  $m_1$ , los nuevos retículos  $\mathfrak{B}(\mathbb{K})$  y  $\mathfrak{B}^\sharp(\mathbb{K})$  serían isomorfos a los dibujados en las Figuras 4.1 y 4.2. Esto se debe a que el atributo  $m_1$  es común a todos los objetos. Si nos fijamos ahora en el objeto  $g_2$ , en el que están presentes todos los atributos, observamos una diferencia en su influencia en  $\mathfrak{B}(\mathbb{K})$  y  $\mathfrak{B}^\sharp(\mathbb{K})$ . La eliminación de  $g_2$  del contexto nos daría un retículo  $\mathfrak{B}(\mathbb{K})$  isomorfo al dibujado en la Figura 4.1 pero, sin embargo, en el caso del retículo mixto sí tendría una influencia significativa: el nuevo  $\mathfrak{B}^\sharp(\mathbb{K})$  no sería isomorfo al de la Figura 4.2. De hecho, en este caso, se tendría que  $\mathfrak{B}(\mathbb{K})$  y  $\mathfrak{B}^\sharp(\mathbb{K})$  serían isomorfos.

**Observación 4.1.10** *Como en el caso clásico, cuando sea conveniente por simplicidad en los razonamientos y sin pérdida de generalidad, asumiremos que  $G^\uparrow = \emptyset$ , es decir, que ni hay un atributo común a todos los objetos ni un atributo ausente en todos ellos.*

Figura 4.2: Retículo  $\mathfrak{B}^\#(\mathbb{K})$  del contexto descrito en la Tabla 4.1.

**Lema 4.1.11** Sea  $\mathbb{K} = \langle G, M, I \rangle$  un contexto formal. Las siguientes propiedades se cumplen:

1.  $\{g\}^\uparrow \in \mathbb{FC}(M)$  y  $\{g\}^{\uparrow\downarrow} = \{g\}$  para todo  $g \in G$ .
2.  $X^\uparrow = \bigcap_{g \in X} \{g\}^\uparrow$  para todo  $X \subseteq G$ .

DEMOSTRACIÓN:

1.  $\{g\}^\uparrow$  es un conjunto consistente completo porque, para todo  $m \in M$ ,  $\langle g, m \rangle \in I$  o  $\langle g, m \rangle \notin I$  y la siguiente unión es disjunta:

$$\{g\}^\uparrow = \{m \in M \mid \langle g, m \rangle \in I\} \cup \{\overline{m} \in \overline{M} \mid \langle g, m \rangle \notin I\}.$$

Por tanto,  $\text{Tot}(\{g\}^\uparrow) = M$  y  $\text{Pos}(\{g\}^\uparrow) \cap \text{Neg}(\{g\}^\uparrow) = \emptyset$ .

Por otro lado, como  $(\uparrow, \downarrow)$  es una conexión de Galois,  $g_1 \in \{g\}^{\uparrow\downarrow}$  implica que  $\{g_1\}^\uparrow \supseteq \{g\}^{\uparrow\downarrow\uparrow} = \{g\}^\uparrow$ . Además, como  $\{g_1\}^\uparrow$  y  $\{g\}^\uparrow$  son conjuntos consistentes completos, se tiene que  $\{g_1\}^\uparrow = \{g\}^\uparrow$  y, la suposición de la no existencia de filas repetidas en el contexto asegura que  $g_1 = g$ .

2. De la misma forma que ocurre en el entorno clásico, como  $(\uparrow, \downarrow)$  es una conexión de Galois entre  $(2^G, \subseteq)$  y  $(2^{M \cup \overline{M}}, \subseteq)$ , para cualquier  $X \subseteq G$ , se tiene que  $X^\uparrow = \left( \bigcup_{g \in X} \{g\} \right)^\uparrow = \bigcap_{g \in X} \{g\}^\uparrow$ .

□

Los lemas elementales señalados anteriormente conducen al siguiente teorema que enfatiza una diferencia significativa con respecto a la construcción clásica y se centra en cómo la inclusión de objetos nuevos influye necesariamente en la estructura del retículo de conceptos mixtos.

Esto es, si un objeto nuevo y diferente (un objeto que difiere en al menos un atributo de cada objeto en el contexto) se añade al contexto formal, entonces el retículo de conceptos mixtos cambia.

**Teorema 4.1.12** *Sea  $\mathbb{K} = \langle G, M, I \rangle$  un contexto formal,  $g_0$  un objeto nuevo, es decir,  $g_0 \notin G$ , e  $Y \subseteq M$  el conjunto de atributos que posee  $g_0$ . Entonces, existe  $g \in G$  tal que  $\text{Pos}(\{g\}^\uparrow) = Y$  si y solo si existe un isomorfismo entre  $\mathfrak{B}^\#(G, M, I)$  y  $\mathfrak{B}^\#(G \cup \{g_0\}, M, I \cup \{\langle g_0, m \rangle \mid m \in Y\})$ .*

DEMOSTRACIÓN: Obviamente, si existe  $g \in G$  tal que  $\text{Pos}(\{g\}^\uparrow) = Y$ , en el contexto  $\langle G \cup \{g_0\}, M, I \cup \{\langle g_0, m \rangle \mid m \in Y\} \rangle$  se tiene que  $\{g_0\}^\uparrow = \{g\}^\uparrow$  y su retículo mixto es isomorfo a  $\mathfrak{B}^\#(\mathbb{K})$ .

Por otro lado, si los retículos de conceptos mixtos son isomorfos, existe  $X \subseteq G$  tal que el conjunto cerrado  $X^\uparrow$  en  $\mathfrak{B}^\#(G, M, I)$  coincide con  $\{g_0\}^\uparrow$ . Así, en el retículo de conceptos mixtos  $\mathfrak{B}^\#(G \cup \{g_0\}, M, I \cup \{\langle g_0, m \rangle \mid m \in X\})$ , por el lema 4.1.11, se tiene que  $\{g_0\}^\uparrow = X^\uparrow = \bigcap_{g \in X} \{g\}^\uparrow$ . Además, como  $\{g_0\}^\uparrow$  es un conjunto consistente completo,  $X \neq \emptyset$  porque por el lema 4.1.7,  $\emptyset^\uparrow = M \cup \overline{M}$ . Así, para todo  $g \in X$  (existe al menos un  $g \in X$ ),  $g_0 \in \{g\}^\uparrow$  y, por el lema 4.1.11,  $\{g\}^\uparrow = \{g_0\}^\uparrow = Y$ . □

**Ejemplo 4.1.13** *Sean los contextos formales  $\mathbb{K}_1 = (\{g_1, g_2\}, \{m_1, m_2, m_3\}, I_1)$  y  $\mathbb{K}_2 = (\{g_1, g_2, g_3\}, \{m_1, m_2, m_3\}, I_2)$  donde  $I_1$  e  $I_2$  son las relaciones binarias definidas en la Tabla 4.2.*

*Nótese que  $\mathbb{K}_2$  se construye desde  $\mathbb{K}_1$  añadiendo el objeto nuevo  $g_3$ .*

$I_1$	$m_1$	$m_2$	$m_3$
$g_1$	×		×
$g_2$	×	×	

$I_2$	$m_1$	$m_2$	$m_3$
$g_1$	×		×
$g_2$	×	×	
$g_3$	×		

Tabla 4.2: Los contextos formales  $\mathbb{K}_1$  y  $\mathbb{K}_2$

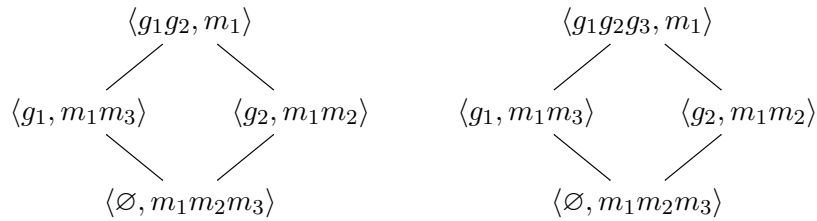


Figura 4.3: Retículos  $\mathfrak{B}(\mathbb{K}_1)$  y  $\mathfrak{B}(\mathbb{K}_2)$  de los contextos dados en la Tabla 4.2.

En el entorno clásico, el retículo de conceptos  $\mathfrak{B}(\mathbb{K}_1)$  y el retículo de conceptos  $\mathfrak{B}(\mathbb{K}_2)$  son isomorfos (Figura 4.3).

Sin embargo, los retículos de conceptos mixtos no pueden ser isomorfos porque el objeto nuevo  $g_3$  no es una repetición de un objeto ya existente (Figura 4.4).

## 4.2. Propiedades de los retículos de conceptos mixtos

Ya hemos adelantado algunas diferencias entre los retículos de conceptos y los retículos de conceptos mixtos. Como se verá posteriormente en esta sección, los retículos de conceptos mixtos se restringen a una subclase específica de retículos. Esto contrasta con lo que sucede en el caso clásico donde todos ellos pueden ser asociados con un contexto formal, es decir, en el entorno clásico, cualquier retículo se corresponde con una colección de contextos formales. Sin embargo, de los 6 retículos representados en la

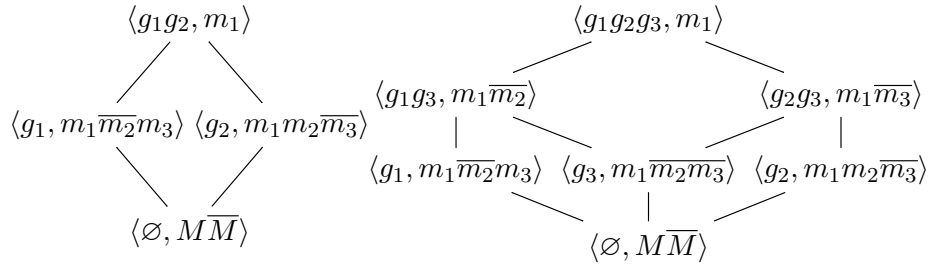
Figura 4.4: Retículos  $\mathfrak{B}^\sharp(\mathbb{K}_1)$  y  $\mathfrak{B}^\sharp(\mathbb{K}_2)$ .

Tabla 4.3, como se demostrará posteriormente, los retículos 3 y 5 no pueden ser asociados a un contexto formal mixto.

Otra diferencia significativa entre los retículos de conceptos y los retículos mixtos se relaciona con las propiedades de los átomos. En el entorno clásico, la inclusión

$$\text{At}(\mathfrak{B}(\mathbb{K})) \subseteq \text{J}(\mathfrak{B}(\mathbb{K})) \subseteq \{\langle \{g\}^{\uparrow\downarrow}, \{g\}^\uparrow \rangle \mid g \in G\} \quad (4.6)$$

se cumple, pero la igualdad no se cumple necesariamente.

**Ejemplo 4.2.1** En la Tabla 4.4, que se corresponde con el retículo 5 de la Tabla 4.3, se tiene que  $\text{At}(\mathfrak{B}(\mathbb{K})) \subseteq \text{J}(\mathfrak{B}(\mathbb{K})) \subseteq \{\langle \{g\}^{\uparrow\downarrow}, \{g\}^\uparrow \rangle \mid g \in G\}$  pero no se cumple la igualdad.  $\text{At}(\mathfrak{B}(\mathbb{K})) \neq \text{J}(\mathfrak{B}(\mathbb{K}))$  porque  $\langle g_1 g_2, m_1 \rangle \in \text{J}(\mathfrak{B}(\mathbb{K}))$  pero no es un átomo.

**Definición 4.2.2** Un retículo  $L$  se dice que es *atomístico* si  $\text{J}(L) = \text{At}(L)$ .

El siguiente teorema asegura que los retículos mixtos son atomísticos y que, en ellos, las inclusiones descritas en (4.6) se transforman en igualdades.

**Teorema 4.2.3** Para cada contexto formal  $\mathbb{K} = \langle G, M, I \rangle$ , su retículo de conceptos mixtos  $\mathfrak{B}^\sharp(\mathbb{K})$  es atomístico y satisface que

$$\text{At}(\mathfrak{B}^\sharp(\mathbb{K})) = \text{J}(\mathfrak{B}^\sharp(\mathbb{K})) = \{\langle \{g\}^{\uparrow\downarrow}, \{g\}^\uparrow \rangle \mid g \in G\}.$$

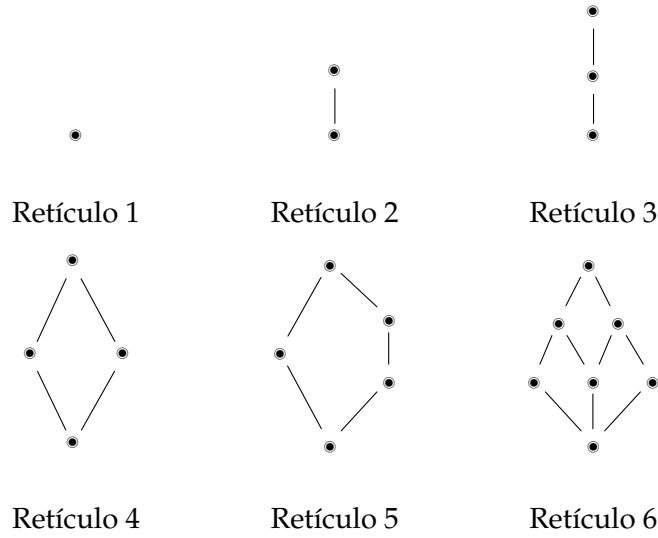


Tabla 4.3: Estructuras de algunos retículos

DEMOSTRACIÓN: Primero, habiendo fijado un objeto  $g_0 \in G$ , vamos a probar que el concepto mixto  $\langle \{g_0\}^{\uparrow\downarrow}, \{g_0\}^{\uparrow} \rangle$  es un átomo en  $\mathfrak{B}^\sharp(\mathbb{K})$ . Si  $\langle X, Y \rangle$  es un concepto mixto tal que  $\langle \emptyset, M \cup \overline{M} \rangle < \langle X, Y \rangle \leq \langle \{g_0\}^{\uparrow\downarrow}, \{g_0\}^{\uparrow} \rangle$ , entonces  $\{g_0\}^{\uparrow} \subseteq Y = X^\uparrow \subsetneq M \cup \overline{M}$ . Por el Lema 4.1.11,  $\{g_0\}^{\uparrow} \subseteq X^\uparrow = \bigcap_{g \in X} \{g\}^{\uparrow}$ . Además, para todo  $g \in X \neq \emptyset$ , por el Lema 4.1.11, ambos  $\{g_0\}^{\uparrow}$  y  $\{g\}^{\uparrow}$  son conjuntos consistentes completos y, como  $\{g_0\}^{\uparrow} \subseteq \{g\}^{\uparrow}$ , se tiene que  $\{g_0\}^{\uparrow} = \{g\}^{\uparrow}$ . Así,  $\{g_0\}^{\uparrow} = X^\uparrow = Y$  y  $\langle X, Y \rangle = \langle \{g_0\}^{\uparrow\downarrow}, \{g_0\}^{\uparrow} \rangle$ .

Recíprocamente, si  $\langle X, Y \rangle$  es un átomo en  $\mathfrak{B}^\sharp(\mathbb{K})$ , entonces  $X \neq \emptyset$  y allí existe  $g_0 \in X$ . Como  $(\uparrow, \downarrow)$  es una conexión de Galois,  $\{g_0\}^{\uparrow} \supseteq X^\uparrow = Y$  y, así,  $\langle \{g_0\}^{\uparrow\downarrow}, \{g_0\}^{\uparrow} \rangle \leq \langle X, Y \rangle$ . Finalmente, como  $\langle X, Y \rangle$  es un átomo, se tiene que  $\langle X, Y \rangle = \langle \{g_0\}^{\uparrow\downarrow}, \{g_0\}^{\uparrow} \rangle$ . □

Obsérvese que, con la suposición dada en la Observación 4.1.9, se tiene que  $\text{At}(\mathfrak{B}^\sharp(\mathbb{K})) = \mathcal{J}(\mathfrak{B}^\sharp(\mathbb{K})) = \{ \langle \{g\}, \{g\}^{\uparrow} \rangle \mid g \in G \}$  y, así, el número de átomos en un retículo de conceptos mixtos coincide con  $|G|$ . Además, para

$I$	$m_1$	$m_2$	$m_3$	$m_4$
$g_1$	×	×		
$g_2$	×		×	
$g_3$				×

Tabla 4.4: Contexto formal  $\mathbb{K}$ 

un concepto mixto  $\langle X, Y \rangle \in \mathfrak{B}^\sharp(\mathbb{K})$ , se tiene que:

1.  $\langle X, Y \rangle \in \text{At}(\mathfrak{B}^\sharp(\mathbb{K}))$  si y solo si  $Y$  es un conjunto consistente completo.
2.  $\langle X, Y \rangle \neq \langle \emptyset, M \cup \overline{M} \rangle$  si y solo si  $Y$  es un conjunto consistente.

Del Teorema 4.2.3 se puede concluir que, a diferencia del entorno clásico, no todos los retículos de conceptos pueden ser asociados con un contexto formal. Así, los retículos 3 y 5 de la Tabla 4.3 no pueden ser asociados con un contexto formal mixto. Ambos tienen un elemento que no es un átomo pero, al mismo tiempo es un elemento  $\vee$ -irreducible en el retículo. Más específicamente, no existe un retículo de conceptos mixtos con 3 elementos.

También se puede deducir del Teorema 4.2.3 que ser un retículo atomístico es una condición necesaria para ser isomorfo a un retículo de conceptos mixtos. Sin embargo, el siguiente ejemplo muestra que no es una condición suficiente.

**Ejemplo 4.2.4** *El retículo  $\mathbb{L}$  descrito en la Figura 4.5 es atomístico, pero no es isomorfo a ningún retículo mixto. Lo probamos por reducción al absurdo: supongamos que existe un contexto  $\mathbb{K} = \langle G, M, I \rangle$  y un isomorfismo  $f : \mathbb{L} \rightarrow \mathfrak{B}^\sharp(\mathbb{K})$ . Por la Observación 4.1.9 y el Teorema 4.2.3, podemos asumir que  $G = \{g_1, g_2, g_3\}$  con  $f(\ell_1) = \langle \{g_1\}, \{g_1\}^\uparrow \rangle$ ,  $f(\ell_2) = \langle \{g_2\}, \{g_2\}^\uparrow \rangle$  y  $f(\ell_3) = \langle \{g_3\}, \{g_3\}^\uparrow \rangle$ . Como  $\ell_1 \neq \ell_2$  y  $\{g_1\}^\uparrow$  y  $\{g_2\}^\uparrow$  son conjuntos consistentes completos (ver Lema 4.1.11),*

uno tiene que existe  $a \in M \cup \overline{M}$  tal que  $a \in \{g_1\}^\uparrow$  y  $\bar{a} \in \{g_2\}^\uparrow$ . Además, como  $\{g_3\}^\uparrow$  es también un conjunto consistente completo, se tiene que  $a \in \{g_3\}^\uparrow$  ó  $\bar{a} \in \{g_3\}^\uparrow$  y la disyunción es exclusiva. Así  $\{g_1\}^\uparrow \cap \{g_3\}^\uparrow \neq \{g_2\}^\uparrow \cap \{g_3\}^\uparrow$ . Esto es contradictorio con  $l_1 \vee l_3 = l_2 \vee l_3 = \top$ .

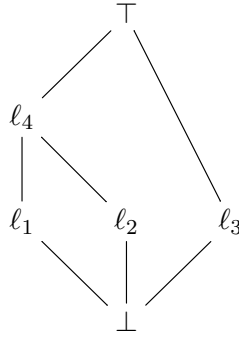


Figura 4.5: Retículo atomístico que no es isomorfo a ningún retículo de conceptos mixtos.

En la búsqueda de una caracterización de aquellos retículos que sí son isomorfos a los retículos de conceptos mixtos, es decir, de condiciones necesarias y suficientes, presentamos a continuación una nueva propiedad y probamos, en el Teorema 4.2.6, que tales retículos la satisfacen.

**Definición 4.2.5 (Ley de  $\vee$ -simplificación atomística)** Decimos que un retículo acotado  $\mathbb{L}$  satisface la ley de  $\vee$ -simplificación atomística si se tiene que, para todo  $a_1, a_2, a_3 \in \text{At}(\mathbb{L})$ ,  $a_1 \vee a_2 = a_1 \vee a_3$  implica  $a_2 = a_3$ .

La ley anterior se asemeja a la bien conocida ley de simplificación de la Teoría de Grupos, en este caso limitada a los elementos atómicos. Nótese que no tendría sentido pedir esta propiedad para todos los elementos del conjunto ya que, si un retículo satisface la ley de  $\vee$ -simplificación, es decir, para todo  $l_1, l_2, l_3 \in L$ ,  $l_1 \vee l_2 = l_1 \vee l_3$  implica  $l_2 = l_3$ , entonces el retículo tendría, como mucho, un elemento. El razonamiento es simple ya que, para todo  $l \in \mathbb{L}$ ,  $\top \vee \top = l \vee \top$  y por tanto  $l = \top$ .



**Teorema 4.2.6** *Para todo contexto  $\mathbb{K}$ , el retículo  $\mathfrak{B}^\#(\mathbb{K})$  satisface la ley de  $\vee$ -simplificación atomística.*

DEMOSTRACIÓN: Consideremos tres átomos  $a_1 = \langle \{g_1\}^{\uparrow\downarrow}, \{g_1\}^{\uparrow} \rangle$ ,  $a_2 = \langle \{g_2\}^{\uparrow\downarrow}, \{g_2\}^{\uparrow} \rangle$ ,  $a_3 = \langle \{g_3\}^{\uparrow\downarrow}, \{g_3\}^{\uparrow} \rangle \in \text{At}(\mathfrak{B}^\#(\mathbb{K}))$ . Por el ítem 1 en el Lema 4.1.11,  $\{g_1\}^{\uparrow}$ ,  $\{g_2\}^{\uparrow}$  y  $\{g_3\}^{\uparrow}$  son conjuntos consistentes completos. Vamos a probar que si se tiene  $\langle \{g_2\}^{\uparrow\downarrow}, \{g_2\}^{\uparrow} \rangle \neq \langle \{g_3\}^{\uparrow\downarrow}, \{g_3\}^{\uparrow} \rangle$ , entonces  $\langle \{g_1\}^{\uparrow\downarrow}, \{g_1\}^{\uparrow} \rangle \vee \langle \{g_2\}^{\uparrow\downarrow}, \{g_2\}^{\uparrow} \rangle \neq \langle \{g_1\}^{\uparrow\downarrow}, \{g_1\}^{\uparrow} \rangle \vee \langle \{g_3\}^{\uparrow\downarrow}, \{g_3\}^{\uparrow} \rangle$ .

En efecto, si  $\{g_2\}^{\uparrow} \neq \{g_3\}^{\uparrow}$  entonces existe  $m \in \{g_2\}^{\uparrow}$  tal que  $\bar{m} \in \{g_3\}^{\uparrow}$ .

- Si  $m \in \{g_1\}^{\uparrow}$  entonces  $m \in \{g_1\}^{\uparrow} \cap \{g_2\}^{\uparrow}$ ,  $m \notin \{g_1\}^{\uparrow} \cap \{g_3\}^{\uparrow}$ , y  $a_1 \vee a_2 \neq a_1 \vee a_3$
- Si  $\bar{m} \in \{g_1\}^{\uparrow}$  entonces  $\bar{m} \notin \{g_1\}^{\uparrow} \cap \{g_2\}^{\uparrow}$ ,  $\bar{m} \in \{g_1\}^{\uparrow} \cap \{g_3\}^{\uparrow}$ , y  $a_1 \vee a_2 \neq a_1 \vee a_3$

□

El siguiente ejemplo muestra cómo las propiedades introducidas hasta ahora son condiciones necesarias pero no suficientes para caracterizar los retículos de conceptos mixtos.

**Ejemplo 4.2.7** *El retículo descrito en la Figura 4.6 es atomístico y cumple la ley de  $\vee$ -simplificación atomística presentada anteriormente, pero no es isomorfo a ningún retículo de conceptos mixto. Esta última afirmación se puede probar por reducción al absurdo:*

Supongamos que existe un isomorfismo  $f : \mathbb{L} \rightarrow \mathfrak{B}^\#(\mathbb{K})$  para un determinado contexto  $\mathbb{K} = \langle G, M, I \rangle$ . Por el Lema 4.1.7, tenemos que  $f(\perp) = \langle \emptyset, M \cup \bar{M} \rangle$  y podemos asumir que  $f(\top) = \langle G, \emptyset \rangle$  (véase la Observación 4.1.10). Sea, por otro lado,  $f(\ell_i) = \langle X_i, Y_i \rangle$  para cada  $1 \leq i \leq 10$ .

Como  $\top \neq \ell_3 = \ell_7 \vee \ell_9$ , por (4.5), se debe cumplir que  $\emptyset \neq Y_3 = Y_7 \cap Y_9$ . Consideremos un elemento arbitrario  $a \in Y_3 \subseteq Y_9$ . Como  $\ell_3 \vee \ell_8 = \ell_3 \vee \ell_{10} = \top$ ,

tenemos que  $Y_3 \cap Y_8 = Y_3 \cap Y_{10} = \emptyset$  y, en consecuencia,  $a \notin Y_8$  y  $a \notin Y_{10}$ . Por otro lado, por el Lema 4.1.11 y el Teorema 4.2.3, tenemos que  $Y_8, Y_{10} \in \mathbb{FC}(M)$  y, por tanto,  $\bar{a} \in Y_8 \cap Y_{10}$ . Como  $l_8 \vee l_{10} = l_1 \geq l_9$ , por (4.3) y (4.5), tenemos también que  $\bar{a} \in Y_9$ , lo que contradice que  $Y_9 \in \mathbb{FC}(M)$ .

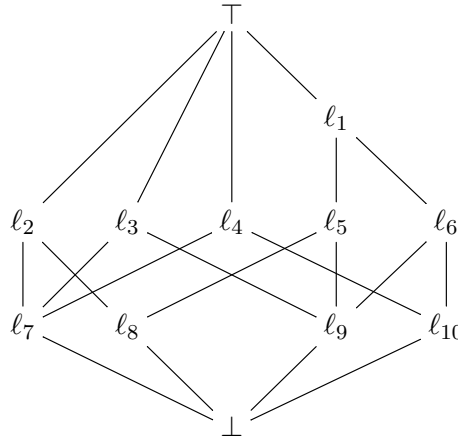


Figura 4.6: Ejemplo de retículo sin contexto mixto asociado.

El ejemplo anterior muestra que se ha de continuar la búsqueda de las condiciones suficientes para caracterizar los retículos de conceptos mixtos.

**Definición 4.2.8** Sea  $\mathbb{L}$  un retículo finito. Para cada  $l \in \mathbb{L}$ , se define su **elemento opuesto** como sigue:

$$l^{op} = \bigvee \{x \in \mathbb{L} \mid l \wedge x = \perp\}$$

**Ejemplo 4.2.9** En la Figura 4.7 tenemos que el opuesto de  $l_1$  es  $l_2$  (y viceversa), que es a la vez  $\vee$ -irreducible y  $\wedge$ -irreducible.

**Ejemplo 4.2.10** En el álgebra de Boole de 8 elementos (Figura 4.8) se observa que el opuesto de un elemento  $\wedge$ -irreducible no es necesariamente  $\wedge$ -irreducible.  $l_1$  es  $\wedge$ -irreducible y su opuesto,  $l_6$ , no lo es. Igualmente se observa que  $l_6$  es  $\vee$ -irreducible y  $l_1$ , que es su opuesto, no lo es.

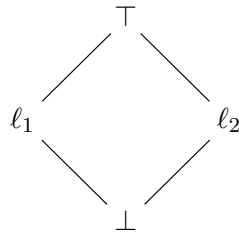


Figura 4.7: Retículo con opuestos  $\wedge$ -irreducibles y  $\vee$ -irreducibles.

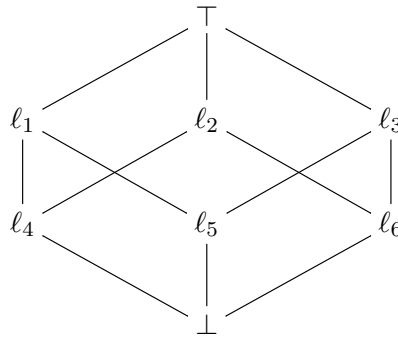


Figura 4.8: Álgebra de Boole de 8 elementos

**Ejemplo 4.2.11** En la Figura 4.9 se tiene que  $\ell_2$  es el opuesto a la vez de  $\ell_1$  y  $\ell_3$  y que  $\ell_1$  no es el opuesto de ningún elemento.

Ganter y Kwuida en [41] consideran pseudocomplementos para el estudio de la negación. El pseudocomplemento de un elemento  $\ell$ , si existe, es  $\ell^* = \text{máx}\{x \mid \ell \wedge x = \perp\}$ . Es obvio que el opuesto de un elemento es pseudocomplemento si cumple que  $\ell \wedge \ell^{op} = \perp$ . Sin embargo, esto no siempre se va a cumplir como podemos ver en el siguiente ejemplo.

**Ejemplo 4.2.12** En el Diamante (Figura 4.10) el opuesto de  $\ell_1$  es  $\top$  y, sin embargo, no tiene pseudocomplemento porque  $\ell_1 \wedge \top \neq \perp$ .

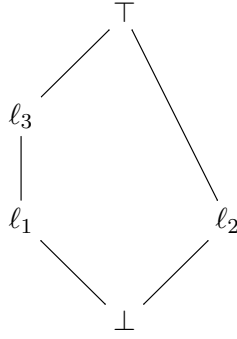


Figura 4.9: Pentágono.

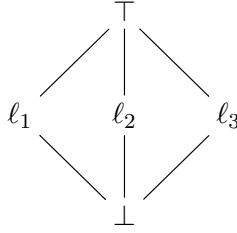


Figura 4.10: Diamante.

**Lema 4.2.13** Sea  $\mathbb{L}$  un retículo finito arbitrario. Para todo  $\ell \in L$ ,

$$\ell^{op} \geq \bigvee \{x \in \text{At}(\mathbb{L}) \mid x \notin \ell_A\}$$

Además, si  $\mathbb{L}$  es atomístico,  $\ell^{op} = \bigvee \{x \in \text{At}(\mathbb{L}) \mid x \notin \ell_A\}$ .

DEMOSTRACIÓN: La desigualdad  $\ell^{op} \geq \bigvee \{x \in \text{At}(\mathbb{L}) \mid x \notin \ell_A\}$  es inmediata porque  $x \notin \ell_A$  implica  $\ell \wedge x = \perp$ , para todo  $x \in \text{At}(\mathbb{L})$ . Es decir,

$$\{x \in \text{At}(\mathbb{L}) \mid x \notin \ell_A\} \subseteq \{x \in \mathbb{L} \mid x \wedge \ell = \perp\}.$$

Se asume ahora que  $L$  es atomístico y se debe probar la desigualdad contraria. Para asegurar que  $\ell^{op} \leq \bigvee \{x \in \text{At}(L) \mid x \notin \ell_A\}$ , por la Definición 4.2.8, es suficiente probar que, para todo  $x \in L$ ,  $\ell \wedge x = \perp$  implica  $x \leq \bigvee \{x \in \text{At}(L) \mid x \notin \ell_A\}$ . En efecto, si  $\ell \wedge x = \perp$  entonces

$x_A \cap \ell_A = \emptyset$ . Así,  $x_A \subseteq \{x \in \text{At}(L) \mid x \notin \ell_A\}$  y, como  $L$  es atomístico,  $x = \bigvee x_A \leq \bigvee \{x \in \text{At}(L) \mid x \notin \ell_A\}$ .  $\square$

**Ejemplo 4.2.14** En la Figura 4.9, que corresponde a un retículo no atomístico, se tiene que  $\ell_2^{op} = \ell_3 > \bigvee \{x \in \text{At}(L) \mid x \notin \ell_2\} = \ell_1$  por lo que no se cumple la igualdad. Esta igualdad sí se da en el Álgebra de Boole de 8 elementos (Figura 4.8) que sí es un retículo atomístico.

**Definición 4.2.15** Un retículo  $L$  se dice que es  $\wedge$ -complementado si para todo  $\ell \in \mathbb{M}(L)$  se tiene que  $\ell^{op}$  es un complemento de  $\ell$ , es decir,  $\ell^{op} \wedge \ell = \perp$  y  $\ell^{op} \vee \ell = \top$ .

**Ejemplo 4.2.16** En el retículo de la Figura 4.6 se puede comprobar que dicho retículo no es  $\wedge$ -complementado debido a que  $\ell_1^{op} = \perp$  pero no es complemento.

El siguiente lema, que es directo, refuerza la conexión entre elementos opuestos y complementos.

**Lema 4.2.17** Sea  $L$  un retículo  $\wedge$ -complementado y  $\ell \in \mathbb{M}(L)$ . Entonces, cualquier complemento de  $\ell$  pertenece a  $(\ell^{op}]$ .

**Ejemplo 4.2.18** En la Figura 4.9, los complementos de  $\ell_2 \in \mathbb{M}(L)$  son  $\ell_1$  y  $\ell_3$ , que pertenecen a  $(\ell_2^{op}] = (\ell_3]$ .

**Teorema 4.2.19** Cualquier retículo de conceptos mixtos es  $\wedge$ -complementado.

DEMOSTRACIÓN: Sea  $\mathbb{K} = \langle G, M, I \rangle$  un contexto formal. Por el Teorema 4.2.3, el retículo de conceptos mixtos  $\mathfrak{B}^\sharp(\mathbb{K})$  es atomístico. Sea  $\langle A_1, B_1 \rangle$  un elemento  $\wedge$ -irreducible de  $\mathfrak{B}^\sharp(\mathbb{K})$  y  $\langle A_2, B_2 \rangle = \langle A_1, B_1 \rangle^{op}$ . Por el Lema 4.2.13, se tiene que  $\langle A_2, B_2 \rangle = \bigvee \{\langle g^{\uparrow\downarrow}, g^{\uparrow} \rangle \mid g \in G, B_1 \not\subseteq g^{\uparrow}\}$  y, por tanto  $\langle A_1, B_1 \rangle \vee \langle A_2, B_2 \rangle = \langle G, G^{\uparrow} \rangle$ .

Para concluir esta demostración se probará que  $\langle A_1, B_1 \rangle \wedge \langle A_2, B_2 \rangle = \langle \emptyset, M \cup \overline{M} \rangle$ . Para ello demostraremos que existe  $a \in B_1$  tal que  $\bar{a} \in B_2$ , lo que implica que  $a, \bar{a} \in B_1 \cup B_2$  y, por tanto,  $(B_1 \cup B_2)^\downarrow = \emptyset$ .

Como  $\langle A_1, B_1 \rangle$  es  $\wedge$ -irreducible, si

$$\langle A_3, B_3 \rangle = \langle A_1, B_1 \rangle^* = \bigwedge \left\{ \langle X, Y \rangle \in \mathfrak{B}^\sharp(\mathbb{K}) \mid \langle A_1, B_1 \rangle < \langle X, Y \rangle \right\} \quad (4.7)$$

(véase Definición 2.1.10) entonces  $\langle A_1, B_1 \rangle < \langle A_3, B_3 \rangle$ , lo que implica que existe  $a \in B_1 \setminus B_3$ .

Para todo  $g \in G$ , probamos que  $a \in g^\uparrow$  implica  $\langle g^{\uparrow\downarrow}, g^\uparrow \rangle \leq \langle A_1, B_1 \rangle$ . Por reducción al absurdo, si suponemos que existe  $g \in G$  tal que  $a \in g^\uparrow$  y  $\langle g^{\uparrow\downarrow}, g^\uparrow \rangle \not\leq \langle A_1, B_1 \rangle$ , se tiene que  $\langle g^{\uparrow\downarrow}, g^\uparrow \rangle \vee \langle A_1, B_1 \rangle > \langle A_1, B_1 \rangle$  y, por (4.7),  $\langle g^{\uparrow\downarrow}, g^\uparrow \rangle \vee \langle A_1, B_1 \rangle \geq \langle A_3, B_3 \rangle$ , lo que implica  $a \in B_3$  que es contradictorio.

Como  $\langle A_2, B_2 \rangle = \bigvee \{ \langle g^{\uparrow\downarrow}, g^\uparrow \rangle \mid g \in G, B_1 \not\subseteq g^\uparrow \}$  se tiene que  $\bar{a} \in B_2$  porque  $a \notin g^\uparrow$  para todo  $g$  tal que  $B_1 \not\subseteq g^\uparrow$  y  $g^\uparrow$  es consistente completo.  $\square$

### 4.3. Caracterizando los retículos de conceptos mixtos

En la sección anterior se probó que, para cada contexto  $\mathbb{K}$ , el retículo de conceptos mixtos  $\mathfrak{B}^\sharp(\mathbb{K})$  satisface:

1. Es atomístico.
2. La ley de  $\vee$ -simplificación atomística se cumple.
3. Es  $\wedge$ -complementado.

Así, estas propiedades dan condiciones necesarias para que un retículo arbitrario sea isomorfo a un retículo de conceptos mixtos.

Queremos probar que estas propiedades son además condiciones suficientes, proporcionando un teorema de caracterización. Como un paso

previo, estudiaremos si algunas de las propiedades pueden ser derivadas de otras. De hecho, existe una fuerte conexión entre las condiciones 2 y 3:

**Teorema 4.3.1** *Cualquier retículo atomístico  $\wedge$ -complementado satisface la ley de  $\vee$ -simplificación atomística.*

DEMOSTRACIÓN: Sean  $a_1, a_2, a_3 \in \text{At}(L)$  donde  $a_1 \vee a_2 = a_1 \vee a_3$ . Trivialmente, si  $a_1 = a_2$  o  $a_1 = a_3$  entonces  $a_1 = a_2 = a_3$ . Suponiendo que  $a_1 \neq a_2$  y  $a_1 \neq a_3$ , es directo que  $(a_1 \vee a_2)^M = a_1^M \cap a_2^M = (a_1 \vee a_3)^M = a_1^M \cap a_3^M$ .

Para probar que  $a_2^M = a_3^M$  (es decir,  $a_2 = a_3$ ) es suficiente probar que para todo  $m \in a_2^M \setminus a_1^M$ , se tiene que  $m \in a_3^M \setminus a_1^M$  y viceversa.

Sea  $m \in a_2^M \setminus a_1^M$ , y supongamos que  $m \notin a_3^M$ . Como  $a_3 \not\leq m$  por el Lema 4.2.13,  $a_3 \leq m^{op}$  y, de forma similar,  $a_1 \leq m^{op}$ . Esto es, se deduce que  $a_1 \vee a_3 \leq m^{op}$ . Se tiene que  $a_2 \leq a_1 \vee a_2 = a_1 \vee a_3 \leq m^{op}$  y, por tanto,  $a_2 \leq m \wedge m^{op} = \perp$  lo que es una contradicción ya que  $a_2 \in \text{At}(L)$ .

La otra inclusión se prueba análogamente. □

La implicación recíproca del teorema anterior no se cumple siempre como muestra el siguiente ejemplo.

**Ejemplo 4.3.2** *El retículo de la Figura 4.6 es atomístico y cumple la ley de  $\vee$ -simplificación atomística, sin embargo no es  $\wedge$ -complementado debido a que  $\ell_3 \in M(L)$ , pero  $\ell_3^{op} = \ell_8 \vee \ell_{10} = \ell_1$  y  $\ell_3 \wedge \ell_3^{op} = \ell_3 \wedge \ell_1 = \ell_9 \neq \perp$ .*

Además, las condiciones 1 y 3 son independientes. El ejemplo previo (Figura 4.6) es un retículo atomístico que no es  $\wedge$ -complementado. Por otro lado, el pentágono (Figura 4.9) es  $\wedge$ -complementado pero no atomístico.

Ya estamos en condiciones de probar el siguiente teorema que proporciona una caracterización de los retículos que se corresponden con retículos mixtos.

**Teorema 4.3.3** *Un retículo finito  $\mathbb{L}$  es  $\wedge$ -complementado y atomístico si y solo si existe un contexto  $\mathbb{K}$  tal que  $\mathbb{L}$  es isomorfo a  $\mathfrak{B}^\sharp(\mathbb{K})$ .*

DEMOSTRACIÓN: Dado un retículo atomístico  $\wedge$ -complementado  $L$ , se considera el contexto  $\mathbb{K}(L) = \langle \text{At}(L), \mathbb{M}(L), \leq \rangle$  y la siguiente función:

$$h: L \rightarrow \mathfrak{B}^\sharp(\mathbb{K}(L)) \text{ con } h(\ell) = \langle \ell_A, \ell^M \cup \ell_{op}^M \rangle$$

donde  $\ell_{op}^M = \{\bar{m} \mid m \in \mathbb{M}(L), \ell \leq m^{op}\}$ .

Primero, se va a probar que la función  $h$  está bien definida, es decir,  $h(\ell)$  es un concepto en el retículo  $\mathfrak{B}^\sharp(\mathbb{K}(L))$ :

- Para probar  $\ell_A^\uparrow = \ell^M \cup \ell_{op}^M$ , es suficiente comprobar las siguientes igualdades:
  - $\ell^M = \{m \in \mathbb{M}(L) \mid a \leq m, \text{ para todo } a \in \ell_A\}$ . Debido a la definición de  $\ell^M$  y  $\ell_A$ , y la transitividad de  $\leq$ , la inclusión  $\ell^M \subseteq \{m \in \mathbb{M}(L) \mid a \leq m, \text{ para todo } a \in \ell_A\}$  es directa. Recíprocamente, si  $m$  es un elemento  $\wedge$ -irreducible con  $a \leq m$  para todo  $a \in \ell_A$ , como  $L$  es atomístico,  $\ell = \bigvee_{a \in \ell_A} a \leq m$  y  $m \in \ell^M$ .
  - $\ell_{op}^M = \{\bar{m} \mid m \in \mathbb{M}(L), a \not\leq m, \text{ para todo } a \in \ell_A\}$ . Para  $m \in \mathbb{M}(L)$ , se va a probar que  $\ell \leq m^{op}$  si y solo si  $a \not\leq m$ , para todo  $a \in \ell_A$ . Si  $\ell \leq m^{op}$  y  $a \in \ell_A$ , entonces  $a$  es un átomo de  $L$  con  $a \leq \ell \leq m^{op}$ . Como  $L$  es  $\wedge$ -complementado,  $m \wedge m^{op} = \perp$  y, además  $a \not\leq m$ . Por otro lado, si  $m \in \mathbb{M}(L)$  con  $a \not\leq m$  para todo  $a \in \ell_A$  entonces  $\ell_A \subseteq \text{At}(L) \setminus m_A$  y, como  $L$  es atomístico, por el Lema 4.2.13,  $\ell = \bigvee_{a \in \ell_A} a \leq \bigvee_{a \in \text{At}(L) \setminus m_A} a = m^{op}$ .
- Por otro lado se prueba que  $\ell_A = \left(\ell^M \cup \ell_{op}^M\right)^\downarrow$ . Esto es, cualquier átomo  $a$  satisface  $a \leq \ell$  si y solo si  $a \leq m$  para todo  $m \in \ell^M$  y  $a \not\leq m$  para todo  $m \in \mathbb{M}(L)$  con  $\ell \leq m^{op}$ .



Si  $a \leq \ell$ , por transitividad,  $a \leq m$  para todo  $m \in \ell^M$  y, para todo  $m \in M(L)$  con  $\ell \leq m^{op}$ , como  $L$  es  $\wedge$ -complementado,  $a \not\leq m$ . Por otro lado, si  $a \leq m$  para todo  $m \in \ell^M$  entonces  $a \leq \bigwedge \ell^M = \ell$ .

Una vez probado que  $h$  está bien definida, se va a probar que  $h$  es un isomorfismo entre retículos:

- Es directo que  $h(\ell_1) \wedge h(\ell_2) = h(\ell_1 \wedge \ell_2)$ , para todo  $\ell_1, \ell_2 \in L$ , porque  $\ell_{1A} \cap \ell_{2A} = (\ell_1 \wedge \ell_2)_A$ .
- $h(\ell_1) \vee h(\ell_2) = h(\ell_1 \vee \ell_2)$ , para todo  $\ell_1, \ell_2 \in L$ , porque:
 
$$(\ell_1^M \cup \ell_{1op}^M) \cap (\ell_2^M \cup \ell_{2op}^M) = (\ell_1^M \cap \ell_2^M) \cup (\ell_1^M \cap \ell_{2op}^M) \cup (\ell_{1op}^M \cap \ell_2^M) \cup (\ell_{1op}^M \cap \ell_{2op}^M) =$$

$$(\ell_1^M \cap \ell_2^M) \cup (\ell_{1op}^M \cap \ell_{2op}^M) = (\ell_1 \vee \ell_2)^M \cup (\ell_1 \vee \ell_2)_{op}^M$$

Finalmente, se prueba que  $h$  es una biyección. Como el retículo es atomístico, si  $h(\ell_1) = h(\ell_2)$  entonces  $\ell_1 = \bigvee \ell_{1A} = \bigvee \ell_{2A} = \ell_2$ . Así,  $h$  es inyectiva. Para probar que  $h$  es sobreyectiva, se considera un concepto  $\langle X, Y \rangle \in \mathfrak{B}^\#(\mathbb{K}(L))$  y el elemento  $\ell = \bigvee X \in L$ . Como  $X \subseteq \ell_A$ , se tiene que  $\langle X, Y \rangle \leq h(\ell)$ . Por el contrario,  $h(\ell) \leq \langle X, Y \rangle$  porque  $Y \subseteq \ell^M \cup \ell_{op}^M$ : Considerando  $m \in M(L)$ , se distinguen dos casos:

- Si  $m \in Y = X^\uparrow$  entonces  $x \leq m$  para todo  $x \in X$  y, así,  $\ell \leq m$ , es decir,  $m \in \ell^M$ .
- Si  $\bar{m} \in Y$ , entonces  $x \not\leq m$  para todo  $x \in X$ . Así  $X \subseteq \text{At}(L) \setminus m_A$  y, por el Lema 4.2.13,  $\ell = \bigvee X \leq \bigvee \{x \in \text{At}(L) \mid x \notin m_A\} = m^{op}$ . Esto es,  $\bar{m} \in \ell_{op}^M$ .

□

Como ni el diamante ni el pentágono cumplen la ley de  $\vee$ -simplificación atomística y los retículos mixtos son atomísticos, es natural preguntarse cómo se relacionan estas propiedades con la distributividad. En esta línea, es fácil probar la siguiente proposición.

**Proposición 4.3.4** *Cualquier retículo booleano finito es un retículo de conceptos mixtos.*

Sin embargo, el resultado contrario no se cumple: existen retículos de conceptos mixtos que incluyen un pentágono o diamante como un subretículo. Véase, por ejemplo, el retículo descrito en la Figura 4.11.

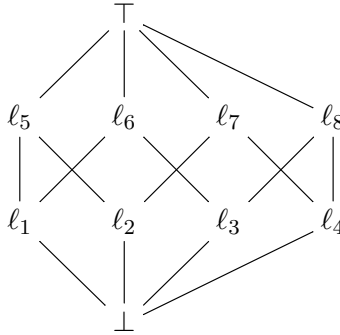


Figura 4.11: Retículo de conceptos mixtos que no es distributivo ni p-álgebra.

La fuerte conexión entre retículos de conceptos y contextos formales en el caso clásico tiene su contrapartida en el caso mixto. En particular, se muestra la forma en la cual, a partir de un retículo atomístico  $\wedge$ -complementado, puede construirse un contexto formal tal que su retículo de conceptos mixtos es isomorfo al retículo inicial.

**Ejemplo 4.3.5** *Consideremos el retículo  $\mathbb{L}$  descrito en la Figura 4.12. Este retículo es atomístico (es decir,  $J(\mathbb{L}) = \text{At}(\mathbb{L}) = \{l_1, l_2, l_3, l_4\}$ ) y  $\wedge$ -complementado porque sus elementos  $\wedge$ -irreducibles son  $l_1, l_8, l_9, l_{10}$  y  $l_{11}$ , y sus elementos opuestos, los cuales son también complementos, son  $l_{11}, l_5, l_4, l_3$  y  $l_1$ , respectivamente.*

*En la Tabla 4.5 se muestra el contexto  $\mathbb{K}(\mathbb{L}) = \langle \text{At}(\mathbb{L}), \mathbf{M}(\mathbb{L}), \leq \rangle$  y se comprueba fácilmente que  $\mathfrak{B}^\sharp(\mathbb{K}(\mathbb{L}))$  es isomorfo a  $\mathbb{L}$ .*

**Ejemplo 4.3.6** *En la Figura 4.13, el retículo de la izquierda es  $\wedge$ -complementado y atomístico y el de la derecha es el retículo mixto  $\mathfrak{B}^\sharp(\mathbb{K}(\mathbb{L}))$ , isomorfo al de la*

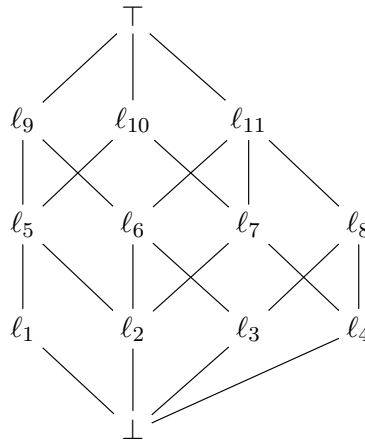


Figura 4.12: Ejemplo de retículo atomístico  $\wedge$ -complementado.

izquierda, donde el contexto  $\mathbb{K}(\mathbb{L}) = \langle \text{At}(\mathbb{L}), M(\mathbb{L}), \leq \rangle$  es el representado en la Tabla 4.6.

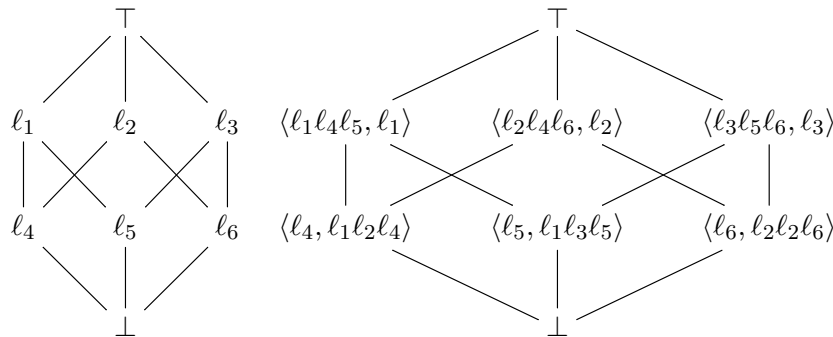


Figura 4.13: Ejemplo de retículo con contexto mixto asociado

Obsérvese que el retículo de conceptos (clásico)  $\mathfrak{B}(\mathbb{K})$  es un subretículo del retículo de conceptos mixtos  $\mathfrak{B}^\sharp(\mathbb{K})$ . Además, la aplicación sobre retículos  $t: \mathfrak{B}^\sharp(\mathbb{K}) \rightarrow \mathfrak{B}(\mathbb{K})$  con  $t(\langle X, Y \rangle) = \langle \text{Pos}(Y)^\downarrow, \text{Pos}(Y) \rangle$  es siempre una aplicación sobreyectiva que conserva el supremo. Sin embargo, esto no es cierto para el ínfimo como muestra el siguiente ejemplo.

$\mathbb{K}(\mathbb{L})$	$l_1$	$l_8$	$l_9$	$l_{10}$	$l_{11}$
$l_1$	×		×	×	
$l_2$			×	×	×
$l_3$		×	×		×
$l_4$		×		×	×

Tabla 4.5: El contexto formal  $\mathbb{K}(\mathbb{L}) = \langle \text{At}(\mathbb{L}), \text{M}(\mathbb{L}), \leq \rangle$

$\mathbb{K}(\mathbb{L})$	$l_1$	$l_2$	$l_3$
$l_4$	×	×	
$l_5$	×		×
$l_6$		×	×

Tabla 4.6: El contexto formal  $\mathbb{K}(\mathbb{L}) = \langle \text{At}(\mathbb{L}), \text{M}(\mathbb{L}), \leq \rangle$  asociado a la Figura 4.13

**Ejemplo 4.3.7** Para el contexto formal  $\mathbb{K}_1 = \langle \{o_1, o_2, o_3, o_4\}, \{m_1, m_2, m_3\}, I \rangle$  donde la relación binaria  $I$  se describe en la Tabla 4.7, el retículo de  $m$ -conceptos  $\mathfrak{B}^\sharp(\mathbb{K}_1)$ , el retículo de conceptos  $\mathfrak{B}(\mathbb{K}_1)$  y la aplicación  $t$  se describen en la Figura 4.14. Se observa en dicha figura que  $\langle \{o_3\}, m_1 \overline{m_2} m_3 \rangle \wedge \langle \{o_4\}, \overline{m_1} \overline{m_2} m_3 \rangle = \langle \emptyset, M \cup \overline{M} \rangle$  en  $\mathfrak{B}^\sharp(\mathbb{K})$ , pero se tiene  $t(\langle \{o_3\}, m_1 \overline{m_2} m_3 \rangle) \wedge t(\langle \{o_4\}, \overline{m_1} \overline{m_2} m_3 \rangle) = \langle \{o_3\}, m_1 m_3 \rangle \wedge \langle \{o_3, o_4\}, m_3 \rangle = \langle \{o_3\}, m_1 m_3 \rangle \neq t(\langle \emptyset, M \cup \overline{M} \rangle) = \langle \emptyset, M \rangle$  en  $\mathfrak{B}(\mathbb{K}_1)$

	$m_1$	$m_2$	$m_3$
$o_1$	×	×	
$o_2$	×		
$o_3$	×		×
$o_4$			×

Tabla 4.7: El contexto formal  $\mathbb{K}_1 = (\{o_1, o_2, o_3, o_4\}, \{m_1, m_2, m_3\}, I)$ .

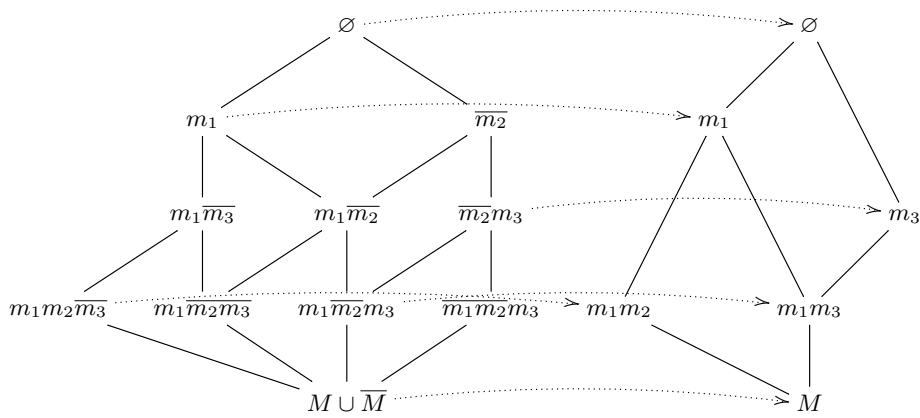


Figura 4.14: Proyección del retículo de conceptos mixtos sobre el retículo de conceptos clásico

### 4.4. La negación: una comparación con enfoques relacionados

Los antecedentes relacionados con la negación de atributos y la negación de conceptos fue vista previamente en la sección 3.2.1, por lo cual no se van a repetir las descripciones y nos limitaremos a comparar los resultados. Como se ha visto, nosotros nos centramos en la negación de los atributos.

Para todo contexto dicotómico  $\mathbb{K}$  se cumple que  $\mathfrak{B}^\#(\mathbb{K}) \cong \mathfrak{B}(\mathbb{K})$  y, por ello, la caracterización proporcionada anteriormente (Teorema 4.3.3) se pue-

de aplicar. En [77], el procedimiento utilizado para trabajar con atributos negativos consiste en transformar cualquier contexto en uno dicotómico mediante la aposición de éste y su negado,  $\mathbb{K}|\overline{\mathbb{K}}$ . Es obvio que, en este caso,  $\mathfrak{B}^\sharp(\mathbb{K}) = \mathfrak{B}(\mathbb{K}|\overline{\mathbb{K}})$  y, por tanto, los resultados obtenidos son aplicables.

El siguiente teorema ya establecido (ver [43, Teorema 7, Página 77] para más detalles) relaciona  $\mathfrak{B}^\sharp(\mathbb{K})$  con  $\mathfrak{B}(\mathbb{K}) \times \mathfrak{B}(\overline{\mathbb{K}})$ .

**Teorema 4.4.1** *Sea  $\mathbb{K}$  un contexto formal. Podemos definir una aplicación de inclusión  $\langle X, Y \rangle \rightarrow (\langle (\text{Pos}(Y))^\Downarrow, \text{Pos}(Y) \rangle, \langle \overline{\text{Neg}(Y)}^\Downarrow, \overline{\text{Neg}(Y)} \rangle)$  que conserva supremos de  $\mathfrak{B}^\sharp(\mathbb{K})$  en el producto directo  $\mathfrak{B}(\mathbb{K}) \times \mathfrak{B}(\overline{\mathbb{K}})$ . Las aplicaciones de los componentes  $\langle X, Y \rangle \rightarrow \langle (\text{Pos}(Y))^\Downarrow, \text{Pos}(Y) \rangle$  y  $\langle X, Y \rangle \rightarrow \langle \overline{\text{Neg}(Y)}^\Downarrow, \overline{\text{Neg}(Y)} \rangle$  son sobreyectivas sobre  $\mathfrak{B}(\mathbb{K})$  y  $\mathfrak{B}(\overline{\mathbb{K}})$ .*

A partir de este resultado se plantea un primer enfoque para abordar los problemas relacionados con los atributos negativos: el retículo de conceptos del contexto de la aposición  $\mathfrak{B}(\mathbb{K}|\overline{\mathbb{K}})$  puede construirse y las técnicas clásicas pueden usarse. Sin embargo, este enfoque lleva a un crecimiento exponencial de los costes y no tiene en cuenta las ventajas de la relación entre la información positiva y negativa. En capítulos posteriores mostraremos que nuestro enfoque conlleva una reducción significativa de costes.

Es directo que, en un retículo de conceptos mixtos  $\mathfrak{B}^\sharp(\mathbb{K})$ , la negación débil (de un concepto, no de atributos) definida por Wille [104] coincide con nuestra noción de elemento opuesto, es decir  $\langle A, B \rangle^\Delta = \langle A, B \rangle^{op}$  (véase el Lema 4.2.13 y la Sección 3.2.1). Por otro lado, la oposición débil está naturalmente definida en nuestro enfoque como  $\langle A, B \rangle^\nabla = \langle ((M \cup \overline{M}) \setminus B)^\Downarrow, ((M \cup \overline{M}) \setminus B)^\Downarrow \uparrow \rangle$ . Así, el papel de oposición débil en nuestro enfoque es trivial ya que se puede describir como sigue:

- Si  $\langle A_1, B_1 \rangle$  es un átomo en  $\mathfrak{B}^\sharp(\mathbb{K})$  y hay otro átomo  $\langle A_2, B_2 \rangle$  tal que  $B_1 = \overline{B_2}$  (recordar que  $B_1$  y  $B_2$  son conjuntos consistentes completos en este caso), entonces  $\langle A_1, B_1 \rangle^\nabla = \langle A_2, B_2 \rangle$ .

- $\langle \emptyset, M \cup \overline{M} \rangle^\nabla = \langle G, G^\uparrow \rangle$  y  $\langle A, B \rangle^\nabla = \langle \emptyset, M \cup \overline{M} \rangle$  en otro caso.

En resumen, la oposición débil en los retículos mixtos es casi trivial, en el sentido de que se va a los extremos, con la única excepción de los átomos cuya oposición sea también un átomo. El siguiente ejemplo ilustra esto.

**Ejemplo 4.4.2** Para el retículo mixto de la Figura 4.15, la oposición débil de cada elemento es:

- $\langle g_1, \overline{abc} \rangle^\nabla = \langle g_2, \overline{abc} \rangle$ ,
- $\langle g_2, \overline{abc} \rangle^\nabla = \langle g_1, \overline{abc} \rangle$ ,
- $\langle \emptyset, M \cup \overline{M} \rangle^\nabla = \langle G, \emptyset \rangle$  y
- $\langle A, B \rangle^\nabla = \langle \emptyset, M \cup \overline{M} \rangle$  en otro caso.

	a	b	c
$g_1$	×		
$g_2$		×	×
$g_3$		×	

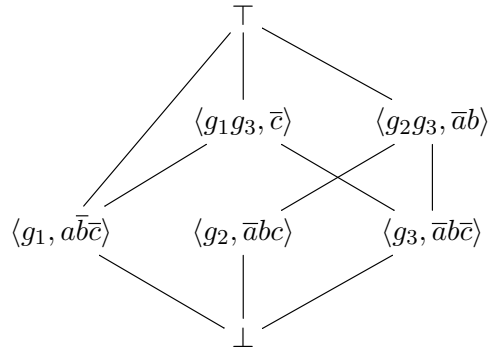


Figura 4.15: Contexto donde la oposición débil  $\nabla$  de uno de los átomos es otro átomo.

En relación a los tres principios clásicos de la negación detallados en la sección 3.2.1, podemos afirmar que, como en el caso general, en los retículos de conceptos mixtos, la negación débil satisface el “principio de tercero excluido”. El principio de contradicción está asegurado por la negación débil

de elementos  $\wedge$ -irreducibles (considerando los retículos de conceptos mixtos como  $\wedge$ -complementados) pero no necesariamente en otros elementos. Véase, por ejemplo, el retículo de conceptos mixtos descrito en la Figura 4.12, donde la negación débil de  $l_7$  es  $l_9$  y  $l_7 \wedge l_9 = l_2 \neq \perp$ . Análogamente, la ley de doble negación puede asegurarse por elementos  $\wedge$ -irreducibles pero no en general, por ejemplo  $l_2^{\Delta\Delta} = \top^\Delta = \perp$  en el mismo ejemplo.

Para el caso de la oposición débil se sigue satisfaciendo el principio de tercero excluido. Sin embargo no cumple el de contradicción ni el de la doble negación ya que en el retículo del ejemplo 4.15:

$$\langle g_3, \overline{abc} \rangle^\nabla = \langle \emptyset, M \cup \overline{M} \rangle, \text{ entonces } \langle g_3, \overline{abc} \rangle \wedge \langle g_3, \overline{abc} \rangle^\nabla = \langle g_3, \overline{abc} \rangle \neq \langle G, G^\uparrow \rangle.$$

$$\langle g_3, \overline{abc} \rangle^{\nabla\nabla} = \langle \emptyset, M \cup \overline{M} \rangle^\nabla = \langle G, G^\uparrow \rangle$$

Relacionada con la definición de Ganter y Kwuida en [41], la siguiente proposición puede ser probada identificando el pseudocomplemento de un elemento con su opuesto.

**Proposición 4.4.3** *Cualquier  $p$ -álgebra atomística finita es un retículo de conceptos mixtos.*

Además, en un retículo de conceptos mixtos, para cualquier elemento  $\wedge$ -irreducible, su elemento opuesto es además su pseudocomplemento. Sin embargo, no todos los retículos de conceptos mixtos son una  $p$ -álgebra. Por ejemplo, el retículo descrito en la Figura 4.11 es un retículo de conceptos mixtos, pero los átomos no tienen pseudocomplemento.





UNIVERSIDAD  
DE MÁLAGA

## Capítulo 5

# Sistemas implicacionales mixtos

La noción de implicación entre conjuntos de atributos puede ser extendida para considerar la información positiva y negativa. Como en ACF clásico, las implicaciones mixtas y los retículos de conceptos mixtos componen las dos caras de la misma moneda, es decir, la información extraída a partir de un contexto formal mixto puede ser representada dualmente mediante un conjunto de implicaciones entre atributos mixtos o un retículo de conceptos mixtos.

La principal ventaja de trabajar con implicaciones, en lugar de con su correspondiente retículo, es que sobre ellas se pueden desarrollar métodos de razonamiento automático. Por otro lado, nos permiten dar diferentes representaciones de un mismo conocimiento, es decir, podemos encontrar diferentes conjuntos de implicaciones (sistemas implicacionales) de los que se pueden derivar (de forma automática y eficiente) todas las implicaciones que se satisfacen en el contexto formal de partida. Se dice que tales sistemas implicacionales son equivalentes. La definición de representaciones canóni-

cas de sistemas implicacionales, con ciertas propiedades (como, por ejemplo, tener el menor número de implicaciones), es un problema bien conocido en el caso clásico.

Como ya comentamos en la Sección 3.2.1, este tema ha sido abordado por otros autores a través de la dicotomización del contexto formal, es decir, con la aposición al contexto de su negado. Allí incidimos en que el principal problema abordado por estos autores es el de la extracción de conocimiento a partir del contexto en términos de implicaciones. Es obvio que, de un contexto  $\mathbb{K}$ , se pueden obtener las implicaciones puramente positivas y, de su negado  $\overline{\mathbb{K}}$ , las puramente negativas. Estos trabajos se centran en obtener implicaciones mixtas a partir de las puramente positivas o negativas. Aún así, en [75], se demuestra que hay implicaciones mixtas que no se pueden obtener de esta manera (véase el Ejemplo 3.2.6) y, por tanto, es necesario dicotomizar el contexto.

Una solución tosca es calcular todas las posibles implicaciones entre atributos desde  $(\mathbb{K}|\overline{\mathbb{K}})$  usando las herramientas clásicas. Sin embargo, esta solución no aprovecha la relación existente entre los atributos y sus negaciones. Además, el número de subconjuntos de atributos para explorar aumenta desde  $2^{|M|}$  a  $4^{|M|}$ .

La solución propuesta allí consiste en calcular las *claves* del contexto  $(\mathbb{K}|\overline{\mathbb{K}}) = \langle G, M \cup \overline{M}, I \cup \overline{I} \rangle$  y, a partir de ellas, un conjunto de implicaciones mixtas que sí representan todo el conocimiento que se puede obtener del contexto en términos de implicaciones. La propiedad en la que se basan asegura que una implicación mixta  $A \rightarrow b$ , con  $A \subseteq M \cup \overline{M}$  y  $b \in M \cup \overline{M}$ , se satisface en  $(\mathbb{K}|\overline{\mathbb{K}})$  si y solo si  $A\overline{b} \rightarrow M\overline{M}$  se cumple en  $(\mathbb{K}|\overline{\mathbb{K}})$ , es decir,  $A \cup \{\overline{b}\}$  es una clave de  $(\mathbb{K}|\overline{\mathbb{K}})$ . Encontrar las claves, equivale a buscar los conjuntos de atributos con soporte cero, lo que es lo mismo, conjuntos de atributos que no están presentes simultáneamente en ningún objeto.

**Ejemplo 5.0.1 ([75])** Consideremos el contexto  $\mathbb{K} = \langle G, M, I \rangle$ , en el cual el

conjunto de objetos es  $G = \{x_1, x_2, x_3, x_4\}$ , el conjunto de atributos es  $M = \{m_1, m_2, m_3, m_4\}$  y cuya relación binaria  $I$  se define en la Tabla 5.1. El contexto formal  $\mathbb{K} = \langle G, \overline{M}, \overline{I} \rangle$  se define en la Tabla 5.2. Finalmente, el contexto formal dicotomizado  $(\mathbb{K}|\mathbb{K}) = \langle G, M \cup \overline{M}, I \cup \overline{I} \rangle$  se define en la Tabla 5.3.

$I$	$m_1$	$m_2$	$m_3$	$m_4$
$x_1$	×			×
$x_2$		×		×
$x_3$	×	×		
$x_4$		×	×	×

Tabla 5.1: Contexto Formal  $\mathbb{K} = \langle G, M, I \rangle$ .

$\overline{I}$	$\overline{m}_1$	$\overline{m}_2$	$\overline{m}_3$	$\overline{m}_4$
$x_1$		×	×	
$x_2$	×		×	
$x_3$			×	×
$x_4$	×			

Tabla 5.2: Contexto Formal  $\overline{\mathbb{K}} = \langle G, \overline{M}, \overline{I} \rangle$ .

$I \cup \overline{I}$	$m_1$	$m_2$	$m_3$	$m_4$	$\overline{m}_1$	$\overline{m}_2$	$\overline{m}_3$	$\overline{m}_4$
$x_1$	×			×		×	×	
$x_2$		×		×	×		×	
$x_3$	×	×					×	×
$x_4$		×	×	×	×			

Tabla 5.3: Contexto Formal  $(\mathbb{K}|\mathbb{K}) = \langle G, M \cup \overline{M}, I \cup \overline{I} \rangle$ .

Obviamente, cualquier implicación entre atributos que es válida en  $\mathbb{K}$ , también es válida en  $(\mathbb{K}|\overline{\mathbb{K}})$ . Por ejemplo, la implicación  $m_2m_3 \rightarrow m_4$  se cumple en el contexto  $\mathbb{K}$  y, además, en  $(\mathbb{K}|\overline{\mathbb{K}})$ . Por el otro lado, cualquier implicación válida en  $\overline{\mathbb{K}}$  también es válida en  $(\mathbb{K}|\overline{\mathbb{K}})$ , como es el caso de  $\overline{m}_2 \rightarrow \overline{m}_3$ . Sin embargo, existen implicaciones que se cumplen en  $(\mathbb{K}|\overline{\mathbb{K}})$  y no se pueden obtener a partir de  $\mathbb{K}$  o a partir de  $\overline{\mathbb{K}}$ . Esto ocurre, por ejemplo, para  $m_2m_4 \rightarrow \overline{m}_1$ . Obsérvese que  $\{m_1, m_2, m_4\}$  es una clave de  $(\mathbb{K}|\overline{\mathbb{K}})$ .

Estos autores, de forma implícita, siguen basando todos sus resultados en los axiomas de Armstrong. Es decir, dado un contexto formal, buscan un conjunto de implicaciones  $\Sigma$  del que se puedan derivar todas las implicaciones mixtas que se cumplen en  $(\mathbb{K}|\overline{\mathbb{K}})$  con los axiomas de Armstrong, es decir, tal que

$$(\mathbb{K}|\overline{\mathbb{K}}) \models A \rightarrow B \quad \text{si y solo si} \quad \Sigma \vdash A \rightarrow B$$

Cuando esta doble implicación se cumple, se dice que  $\Sigma$  es *completo* para  $(\mathbb{K}|\overline{\mathbb{K}})$  respecto de los axiomas de Armstrong.

La solución que aquí se propone va más allá. El sistema axiomático de Armstrong es correcto (es decir,  $\Sigma \vdash A \rightarrow B$  implica  $\Sigma \models A \rightarrow B$ ) y completo ( $\Sigma \models A \rightarrow B$  implica  $\Sigma \vdash A \rightarrow B$ ) en el caso clásico. Sin embargo, cuando consideramos implicaciones mixtas, como veremos a continuación, este sistema deja de ser completo porque no captura las propiedades de la negación. Para compensarlo, los mencionados autores, necesitan aumentar el tamaño del conjunto  $\Sigma$ .

**Ejemplo 5.0.2** *A partir de las claves minimales de la dicotomización del contexto de la Tabla 5.4 se obtiene el siguiente sistema de implicaciones mixtas:  $\{m_1 \rightarrow m_2\overline{m}_3, m_3 \rightarrow \overline{m}_1m_2, \overline{m}_1m_2 \rightarrow m_3, \overline{m}_2 \rightarrow \overline{m}_1\overline{m}_3, \overline{m}_1\overline{m}_3 \rightarrow \overline{m}_2, \overline{m}_3m_2 \rightarrow m_1\}$*

$I$	$m_1$	$m_2$	$m_3$
$o_1$		×	×
$o_2$	×	×	
$o_3$			

Tabla 5.4: Contexto formal

Vemos que hay implicaciones que son redundantes y que no se pueden deducir de los axiomas de Armstrong. Por ejemplo, es fácil razonar que en todo contexto en el que las implicaciones mixtas  $m_1 \rightarrow m_2\bar{m}_3$  y  $m_3 \rightarrow \bar{m}_1m_2$  se cumplen, se debe cumplir también la implicación  $\bar{m}_2 \rightarrow \bar{m}_1\bar{m}_3$ . Sin embargo, los axiomas de Armstrong no permiten inferirla.

La propuesta que abordamos en este capítulo es ampliar el sistema axiomático hasta tener uno completo, que capture la semántica de la negación.

## 5.1. MixAtL : una lógica para implicaciones entre atributos mixtos

Aquí se presenta formalmente una lógica para manejar las implicaciones mixtas considerando tres pilares: el lenguaje, la semántica y un sistema axiomático correcto y completo.

**Definición 5.1.1** Dado un conjunto de atributos  $M$ , el lenguaje de MixAtL es

$$\mathfrak{L}_M = \{X \rightarrow Y \mid X, Y \subseteq M \cup \bar{M}\}$$

Las fórmulas en  $\mathfrak{L}_M$  se denominan implicaciones entre atributos mixtos.

Como es costumbre, y venimos haciendo en capítulos anteriores, mantenemos el convenio de omitir las llaves y las comas en las premisas y las conclusiones de las implicaciones. Del mismo modo, la unión de conjuntos dentro de una implicación se denotará por yuxtaposición de los dos conjuntos. Por ejemplo, escribiremos  $m_1\bar{m}_2 \rightarrow m_3$  en lugar de  $\{m_1, \bar{m}_2\} \rightarrow \{m_3\}$  y  $AB \rightarrow CD$  en lugar de  $A \cup B \rightarrow C \cup D$ . El objetivo es reducir el tamaño de las fórmulas y simplificar su lectura.

Antes de comenzar a describir la semántica y el sistema axiomático, recordamos a continuación una notación introducida en capítulos anteriores y que vamos a usar profusamente a partir de ahora. Dado un conjunto de atributos  $M$ , para todo  $A \subseteq M \cup \bar{M}$ :

- $\text{Pos}(A) = A \cap M$ ,  $\text{Neg}(A) = \bar{A} \cap M$  y  $\text{Tot}(A) = \text{Pos}(A) \cup \text{Neg}(A)$ .
- $\text{Ctts}(M) = \{A \subseteq M \cup \bar{M} \mid \text{Pos}(A) \cap \text{Neg}(A) = \emptyset\}$ .
- $\text{FC}(M) = \{A \in \text{Ctts}(M) \mid \text{Tot}(A) = M\}$ .

### 5.1.1. La semántica de MixAtL

Una vez definido el lenguaje en la sección anterior se va a introducir la semántica de **MixAtL**. Para diferenciarlo del caso clásico, usaremos el símbolo  $\models^\#$  en lugar de  $\models$ .

**Definición 5.1.2** Sea  $\mathbb{K} = \langle G, M, I \rangle$  un contexto formal y  $A \rightarrow B \in \mathcal{L}_M$ . Decimos que el contexto  $\mathbb{K}$  es un **modelo** para  $A \rightarrow B$ , denotado por  $\mathbb{K} \models^\# A \rightarrow B$ , si  $A^\downarrow \subseteq B^\downarrow$ , o equivalentemente,  $B \subseteq A^\downarrow\uparrow$ .

Es inmediato que, dado un contexto  $\mathbb{K} = \langle G, M, I \rangle$  y una implicación  $A \rightarrow B \in \mathcal{L}_M$ , se cumple la siguiente equivalencia

$$\mathbb{K} \models^\# A \rightarrow B \quad \text{si y solo si} \quad \mathbb{K}|\bar{\mathbb{K}} \models A \rightarrow B.$$

**Ejemplo 5.1.3** Considerando el contexto formal  $\mathbb{K} = \langle G, M, I \rangle$  donde  $G = \{g_1, g_2, g_3, g_4\}$ ,  $M = \{m_1, m_2, m_3, m_4, m_5\}$  e  $I$  la relación binaria descrita en la Tabla 5.5, se tiene que  $\mathbb{K} \not\models^\# m_2 \rightarrow m_4$  y  $\mathbb{K} \models^\# m_2 \rightarrow \overline{m_4}$ , mientras que ni  $\mathbb{K} \not\models^\# m_2 \rightarrow m_3$  ni  $\mathbb{K} \not\models^\# m_2 \rightarrow \overline{m_3}$ .

I	$m_1$	$m_2$	$m_3$	$m_4$	$m_5$
$g_1$		×	×		×
$g_2$	×	×			
$g_3$		×	×		×
$g_4$			×	×	

Tabla 5.5: Contexto Formal  $\mathbb{K} = \langle G, M, I \rangle$ .

Como es usual, dado un conjunto de implicaciones entre atributos mixtos  $\Sigma \subseteq \mathcal{L}_M$  y un contexto formal  $\mathbb{K}$ , usaremos la siguiente notación:

- La expresión  $\mathbb{K} \models^\# \Sigma$  denota que  $\mathbb{K} \models^\# A \rightarrow B$  para todo  $A \rightarrow B \in \Sigma$ .
- $\Sigma \models^\# A \rightarrow B$  denota que cualquier modelo para  $\Sigma$  es también modelo para  $A \rightarrow B$ , es decir,  $\mathbb{K} \models^\# \Sigma$  implica  $\mathbb{K} \models^\# A \rightarrow B$  para todo contexto formal  $\mathbb{K}$ . En este caso se dice que  $A \rightarrow B$  se deriva semánticamente desde  $\Sigma$ .

Diremos también que dos conjuntos de implicaciones de atributos mixtos  $\Sigma_1, \Sigma_2 \subseteq \mathcal{L}_M$  son *semánticamente equivalentes*, y lo denotaremos por  $\Sigma_1 \equiv^\# \Sigma_2$ , si, para todo contexto formal  $\mathbb{K}$ , se cumple que  $\mathbb{K} \models^\# \Sigma_1$  si y solo si  $\mathbb{K} \models^\# \Sigma_2$ . Es decir, ambos conjuntos,  $\Sigma_1$  y  $\Sigma_2$ , representan el mismo conocimiento.



### 5.1.2. Un sistema axiomático para MixAtL

En esta sección, se va a proporcionar un sistema axiomático y algunas definiciones y propiedades relacionadas. Posteriormente, en la siguiente sección, se probará que ambas partes, la semántica y la sintáctica coinciden, es decir, que si una implicación entre atributos mixtos se deriva semánticamente desde un conjunto de implicaciones, entonces puede ser inferida usando el sistema axiomático, y viceversa.

El sistema axiomático para MixAtL considera dos axiomas y cuatro reglas de inferencia. Son las siguientes: dados  $a, b \in M \cup \overline{M}$  y  $A, B, C \subseteq M \cup \overline{M}$ ,

[Inc]	Inclusión:	$\vdash^{\#} AB \rightarrow A$
[Cont]	Contradicción:	$\vdash^{\#} a\overline{a} \rightarrow M\overline{M}$
[Augm]	Aumentación:	$A \rightarrow B \quad \vdash^{\#} AC \rightarrow BC$
[Trans]	Transitividad:	$A \rightarrow B, B \rightarrow C \quad \vdash^{\#} A \rightarrow C$
[Rft]	Reflexión:	$Aa \rightarrow b \quad \vdash^{\#} A\overline{b} \rightarrow \overline{a}$
[Tru]	Confianza:	$a \rightarrow \overline{a} \quad \vdash^{\#} \emptyset \rightarrow \overline{a}$

La noción de derivación sintáctica se introduce como es habitual:

**Definición 5.1.4** Una implicación  $\varphi \in \mathcal{L}_M$  se dice que es *sintácticamente derivable*, o *inferida*, desde un conjunto de implicaciones  $\Sigma \subseteq \mathcal{L}_M$  en MixAtL, denotada por  $\Sigma \vdash^{\#} \varphi$ , si existe una secuencia de implicaciones  $\varphi_1, \dots, \varphi_n \in \mathcal{L}_M$  tal que  $\varphi_n = \varphi$  y, para todo  $1 \leq i \leq n$ , se tiene que:

- $\varphi_i$  pertenece a  $\Sigma$ ,
- $\varphi_i$  es un axioma, o
- $\varphi_i$  se obtiene aplicando las reglas de inferencia descritas anteriormente a las implicaciones  $\{\varphi_j \mid 1 \leq j < i\}$ .

En este caso, se dice que la secuencia  $\{\varphi_i \mid 1 \leq i \leq n\}$  es una prueba o demostración para  $\Sigma \vdash^\# \varphi$ .

Obsérvese que el sistema axiomático formado por la terna [Inc], [Augm] y [Trans] es el popularmente conocido como “axiomas de Armstrong”. Reservamos el símbolo  $\vdash$  para la derivación mediante el sistema axiomático de Armstrong y usamos  $\vdash^\#$  para el nuevo sistema aquí presentado.

El sistema de Armstrong, como ya hemos adelantado, no es completo. De hecho, de él no se pueden derivar las nuevas reglas, que, como probaremos, son correctas. En resumen, que  $\Sigma \vdash A \rightarrow B$  implica  $\Sigma \vdash^\# A \rightarrow B$  pero el recíproco no es cierto porque, por ejemplo,  $m_1 \rightarrow \overline{m_2} \vdash^\# m_2 \rightarrow \overline{m_1}$  pero  $m_1 \rightarrow \overline{m_2} \not\vdash m_2 \rightarrow \overline{m_1}$ .

Tanto el esquema de axioma [Cont] como la regla de inferencia [Rft] están inspiradas en propiedades presentadas [76]. Sin embargo, hasta donde sabemos, la regla de inferencia [Tru] es novedosa.

Para simplificar las demostraciones, habitualmente se definen reglas de inferencia derivadas. Las siguientes reglas de inferencia son ampliamente conocidas y derivan de los axiomas de Armstrong y, por ello, se cumplen en **MixAtL**.

**Lema 5.1.5** *Sea  $M$  un conjunto finito de atributos. Las siguientes reglas de inferencia se cumplen en **MixAtL** : para todo  $A, B, C \subseteq M \cup \overline{M}$ ,*

[Frag] *Fragmentación:*  $A \rightarrow BC \vdash^\# A \rightarrow B$

[Un] *Unión:*  $A \rightarrow B, A \rightarrow C \vdash^\# A \rightarrow BC$

[gTr] *Transitividad Generalizada:*  $A \rightarrow BC, B \rightarrow D \vdash^\# A \rightarrow D$

DEMOSTRACIÓN: A pesar de que estas reglas de inferencia son bien conocidas y es un ejercicio fácil su demostración, mostraremos una secuencia que es una demostración para [Frag] como ejemplo:

$$\varphi_1 = (A \rightarrow BC), \varphi_2 = (BC \rightarrow B), \varphi_3 = (A \rightarrow B)$$

donde  $\varphi_1$  es la hipótesis,  $\varphi_2$  es un axioma (ver [Inc]) y  $\varphi_3$  se obtiene desde  $\varphi_1$  y  $\varphi_2$  por [Trans].  $\square$

Como en el caso clásico, el sistema axiomático de **MixAtL** define un cierre sintáctico en  $2^{M \cup \overline{M}}$ . Las siguientes definiciones y resultados pretenden introducirlo.

**Definición 5.1.6** Sea  $M$  un conjunto finito de atributos y  $\Sigma \subseteq \mathfrak{L}_M$  un conjunto de implicaciones entre atributos mixtos. Decimos que un conjunto  $C \subseteq M \cup \overline{M}$  es **m-cerrado** con respecto a  $\Sigma$  si  $A \subseteq C$  implica  $B \subseteq C$  para todo  $A \rightarrow B \in \mathfrak{L}_M$  tal que  $\Sigma \vdash^\# A \rightarrow B$ .

El conjunto de conjuntos m-cerrados con respecto a  $\Sigma$  se denota por  $\text{mcl}(\Sigma)$ .

Ahora probamos que la definición anterior se corresponde con un sistema de clausura o familia de Moore (véase la Definición 2.2.4) y algunas propiedades relacionadas con él.

**Proposición 5.1.7** Sea  $M$  un conjunto finito de atributos,  $\Sigma \subseteq \mathfrak{L}_M$  y  $C_1, C_2 \subseteq M \cup \overline{M}$ . Entonces,

1.  $M \cup \overline{M} \in \text{mcl}(\Sigma)$ .
2. Si  $C_1, C_2 \in \text{mcl}(\Sigma)$  entonces  $C_1 \cap C_2 \in \text{mcl}(\Sigma)$ .
3.  $\text{mcl}(\Sigma) \subseteq \text{Ctts}(M) \cup \{M \cup \overline{M}\}$ .

DEMOSTRACIÓN: Los puntos (1) y (2) son consecuencia directa de la definición. Para el punto (3), probamos que, si  $C \notin \text{Ctts}(M)$  y  $C \neq M \cup \overline{M}$  entonces  $C$  no es m-cerrado. En este caso, dado que  $C$  no es consistente, debe existir  $m \in M$  tal que  $m, \overline{m} \in C$ . Por otro lado, por [Cont],  $\Sigma \vdash^\# m\overline{m} \rightarrow M\overline{M}$  y, por tanto,  $C$  no es m-cerrado ya que  $\{m, \overline{m}\} \subseteq C$  pero  $M \cup \overline{M} \not\subseteq C$ .  $\square$

Dado que, como consecuencia de los puntos (1) y (2) de la Proposición 5.1.7, la familia de conjuntos m-cerrados  $\text{mcl}(\Sigma)$  constituye un sistema

de clausura, ésta define un operador de cierre de la forma usual (véase la Definición 2.2.1). Denominaremos a tal operador *cierre sintáctico mixto*.

**Definición 5.1.8** Sea  $M$  un conjunto finito de atributos,  $\Sigma \subseteq \mathfrak{L}_M$  y  $A \subseteq M \cup \overline{M}$ . El conjunto  $A_\Sigma^\# = \bigcap \{F \in \text{mcl}(\Sigma) \mid A \subseteq F\}$  se dice que es el **cierre sintáctico mixto** de  $A$  con respecto a  $\Sigma$ .

Como hemos dicho,  $(-)_\Sigma^\# : 2^{M \cup \overline{M}} \rightarrow 2^{M \cup \overline{M}}$  es un operador de cierre, es decir:

- $A \subseteq A_\Sigma^\# = (A_\Sigma^\#)_\Sigma^\#$  para todo  $A \subseteq M \cup \overline{M}$  y
- $A_\Sigma^\# \subseteq B_\Sigma^\#$  para todo  $A \subseteq B \subseteq M \cup \overline{M}$ .

**Ejemplo 5.1.9** Sea  $M = \{m_1, m_2, m_3, m_4\}$ . Dado el conjunto de implicaciones

$$\Sigma = \{\overline{m}_4 \rightarrow m_1 m_2 m_3, \overline{m}_3 m_4 \rightarrow m_1 \overline{m}_2, m_2 \rightarrow m_3, \\ \overline{m}_2 m_4 \rightarrow m_1, m_2 m_3 m_4 \rightarrow \overline{m}_1, m_1 m_4 \rightarrow \overline{m}_2\},$$

el conjunto de  $m$ -cerrados respecto de  $\Sigma$  es

$$\text{mcl}(\Sigma) = \{\emptyset, m_1, m_3, m_4, m_1 m_3, m_2 m_3, m_3 m_4, m_1 \overline{m}_2 m_4, \\ m_1 m_2 m_3 \overline{m}_4, m_1 \overline{m}_2 m_3 m_4, \overline{m}_1 m_2 m_3 m_4, m_1 \overline{m}_2 \overline{m}_3 m_4, M \cup \overline{M}\}$$

cuyo retículo está representado en la Fig. 5.1.

**Lema 5.1.10** Sea  $M$  un conjunto finito de atributos,  $\Sigma \subseteq \mathfrak{L}_M$  y  $A \subseteq M \cup \overline{M}$ . Entonces  $\Sigma \vdash^\# A \rightarrow A_\Sigma^\#$ .

DEMOSTRACIÓN: Como  $M$  es finito, existe una familia finita  $\{X_i \subseteq M \cup \overline{M} \mid 1 \leq i \leq n\}$  formada por aquellos conjuntos que satisfacen  $\Sigma \vdash^\# A \rightarrow X_i$ .

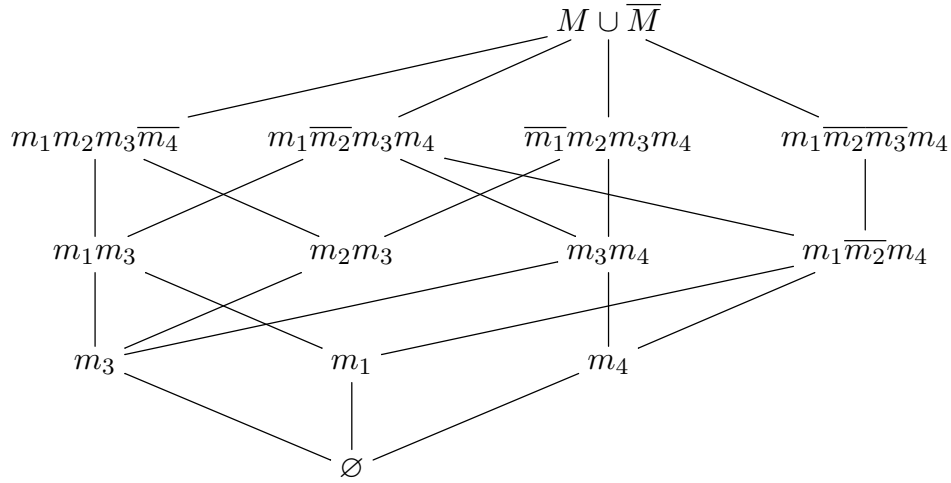


Figura 5.1: Retículo formado por los elementos de  $mcl(\Sigma)$ .

Nótese que, por  $[Inc]$ , esta familia es no vacía. Entonces, para cada  $X_i$  existe una prueba  $\varphi_{i1}, \dots, \varphi_{im_i}$  para  $\Sigma \vdash^\# A \rightarrow X_i$ .

Consideremos ahora  $B := \bigcup_{i=1}^n X_i$  y probaremos que  $\Sigma \vdash^\# A \rightarrow B$ . Sea  $\psi_i$  la implicación de atributos mixtos  $A \rightarrow X_1 \dots X_i$  para cada  $1 < i \leq n$ . La secuencia finita  $\varphi_{11}, \dots, \varphi_{1m_1}, \varphi_{21}, \dots, \varphi_{2m_2}, \dots, \varphi_{n1}, \dots, \varphi_{nm_n}, \psi_2, \dots, \psi_n$  es una prueba para  $\Sigma \vdash^\# A \rightarrow B$  donde  $\psi_2$  se obtiene aplicando  $[Un]$  a  $\varphi_{1m_1}$  y  $\varphi_{2m_2}$ , y  $\psi_j$  se obtiene aplicando  $[Un]$  a  $\varphi_{jm_j}$  y  $\psi_{j-1}$ , para cada  $2 < j \leq n$ .

Por definición, como  $A_\Sigma^\#$  es m-cerrado con respecto a  $\Sigma$  y  $\Sigma \vdash^\# A \rightarrow B$ , se tiene que  $B \subseteq A_\Sigma^\#$ . Como  $A \subseteq B \subseteq A_\Sigma^\#$  (nótese que, por  $[Inc]$ ,  $\Sigma \vdash^\# A \rightarrow A$ ), tenemos que  $A_\Sigma^\# \subseteq B_\Sigma^\# \subseteq (A_\Sigma^\#)_\Sigma^\# = A_\Sigma^\#$  y, por tanto,  $B_\Sigma^\# = A_\Sigma^\#$ . Ahora probemos que  $B = A_\Sigma^\#$  por reducción al absurdo. Supongamos que  $B \subsetneq B_\Sigma^\#$ , es decir, existe  $U, V \subseteq M \cup \overline{M}$  tal que  $\Sigma \vdash^\# U \rightarrow V$  y  $U \subseteq B$ , pero  $V \not\subseteq B$ . Entonces, aplicando  $[gTr]$  sobre  $A \rightarrow B$  y  $U \rightarrow V$ , se tiene que  $\Sigma \vdash^\# A \rightarrow V$  que contradice  $V \not\subseteq B$ . Finalmente, concluimos que  $B = B_\Sigma^\# = A_\Sigma^\#$  y  $\Sigma \vdash^\# A \rightarrow A_\Sigma^\#$ .  $\square$

Este resultado lleva al siguiente teorema que asegura que, dado un

conjunto de implicaciones de atributos mixtos  $\Sigma$ , la derivación sintáctica  $\vdash^\#$  y el operador de cierre sintáctico  $(-)_\Sigma^\#$  contienen la misma información.

**Teorema 5.1.11** *Sea  $M$  un conjunto finito de atributos,  $\Sigma \subseteq \mathcal{L}_M$  y  $A, B \subseteq M \cup \overline{M}$ . Entonces  $\Sigma \vdash^\# A \rightarrow B$  si y solo si  $B \subseteq A_\Sigma^\#$ .*

DEMOSTRACIÓN: Para la implicación directa, supongamos que  $\Sigma \vdash^\# A \rightarrow B$  y, por definición, como  $A_\Sigma^\#$  es m-cerrado con respecto a  $\Sigma$ , tenemos que  $B \subseteq A_\Sigma^\#$ .

Recíprocamente, supongamos ahora que  $B \subseteq A_\Sigma^\#$ . Por [Inc],  $A_\Sigma^\# \rightarrow B$  es un axioma y, por el Lema 5.1.10,  $\Sigma \vdash^\# A \rightarrow A_\Sigma^\#$ . Aplicando la regla [Trans] a estas dos implicaciones tenemos, como consecuencia, que  $\Sigma \vdash^\# A \rightarrow B$ .  $\square$

**Corolario 5.1.12** *Sea  $M$  un conjunto finito de atributos,  $\Sigma \subseteq \mathcal{L}_M$  y  $A \subseteq M \cup \overline{M}$ . Entonces  $A_\Sigma^\# = \text{máx}\{B \subseteq M \cup \overline{M} \mid \Sigma \vdash^\# A \rightarrow B\}$ .*

**Ejemplo 5.1.13** *Sea  $M = \{m_1, m_2, m_3\}$  y  $\Sigma = \{m_1 m_2 \rightarrow m_3, m_1 \overline{m}_3 \rightarrow m_2\}$ . Para el conjunto  $A = \{\overline{m}_3\}$  tenemos que  $A_\Sigma^\# = \{\overline{m}_1, \overline{m}_3\}$  y la siguiente secuencia demuestra que  $\Sigma \vdash^\# \overline{m}_3 \rightarrow \overline{m}_1 \overline{m}_3$ :*

$$\begin{array}{ll}
 \varphi_1 = m_1 m_2 \rightarrow m_3 & \text{Hipótesis.} \\
 \varphi_2 = m_1 \overline{m}_3 \rightarrow m_2 & \text{Hipótesis.} \\
 \varphi_3 = m_1 \overline{m}_3 \rightarrow \overline{m}_2 & \text{Aplicando [Rft] a } \varphi_1. \\
 \varphi_4 = m_1 \overline{m}_3 \rightarrow m_2 \overline{m}_2 & \text{Aplicando [Un] a } \varphi_2 \text{ y } \varphi_3. \\
 \varphi_5 = m_2 \overline{m}_2 \rightarrow M \overline{M} & \text{Por [Cont].} \\
 \varphi_6 = m_1 \overline{m}_3 \rightarrow M \overline{M} & \text{Aplicando [Trans] a } \varphi_4 \text{ y } \varphi_5. \\
 \varphi_7 = m_1 \overline{m}_3 \rightarrow m_3 & \text{Aplicando [Frag] a } \varphi_6. \\
 \varphi_8 = \overline{m}_3 \rightarrow \overline{m}_1 & \text{Aplicando [Rft] a } \varphi_7. \\
 \varphi_9 = \overline{m}_3 \rightarrow \overline{m}_1 \overline{m}_3 & \text{Aplicando [Augm] a } \varphi_8.
 \end{array}$$

### 5.1.3. Corrección y completitud de MixAtL

En esta sección se va a probar que la lógica de implicaciones para implicaciones mixtas es completa y correcta. Primero, el siguiente teorema asegura la corrección del sistema axiomático.

**Teorema 5.1.14** *Sea  $M$  un conjunto finito de atributos,  $A \rightarrow B \in \mathcal{L}_M$  y  $\Sigma \subseteq \mathcal{L}_M$ . Entonces,  $\Sigma \vdash^\# A \rightarrow B$  implica que  $\Sigma \models^\# A \rightarrow B$ .*

DEMOSTRACIÓN: Es suficiente probar que los axiomas se cumplen en cualquier contexto formal y, para cada regla de inferencia, los modelos de la primera parte son también modelos en la segunda parte. Para ello, haremos uso de que  $\downarrow \circ \uparrow$  es un operador de cierre y, por tanto,  $A \subseteq A^{\downarrow\uparrow} = A^{\downarrow\uparrow\downarrow\uparrow}$  para todo  $A \subseteq M \cup \overline{M}$  y  $A^{\downarrow\uparrow} \subseteq B^{\downarrow\uparrow}$  para todo  $A \subseteq B \subseteq M \cup \overline{M}$ .

Considerando un contexto formal arbitrario  $\mathbb{K} = \langle G, M, I \rangle$ ,  $a, b \in M \cup \overline{M}$  y  $A, B, C \subseteq M \cup \overline{M}$ , se tiene que:

- [Inc]  $\mathbb{K} \models^\# AB \rightarrow A$  porque  $A \subseteq A^{\downarrow\uparrow} \subseteq (A \cup B)^{\downarrow\uparrow}$ .
- [Cont]  $\mathbb{K} \models^\# a\overline{a} \rightarrow M\overline{M}$  porque  $\{a, \overline{a}\}^{\downarrow\uparrow} = \emptyset^\uparrow = M \cup \overline{M}$ .
- [Augm] Si  $\mathbb{K} \models^\# A \rightarrow B$ , es decir,  $B \subseteq A^{\downarrow\uparrow}$ , entonces  $B \cup C \subseteq A^{\downarrow\uparrow} \cup C^{\downarrow\uparrow} \subseteq (A \cup C)^{\downarrow\uparrow}$  y, por tanto,  $\mathbb{K} \models^\# AC \rightarrow BC$ .
- [Trans] Si  $\mathbb{K} \models^\# A \rightarrow B$  y  $\mathbb{K} \models^\# B \rightarrow C$ , se tiene que  $B \subseteq A^{\downarrow\uparrow}$  y  $C \subseteq B^{\downarrow\uparrow}$ . Por tanto,  $C \subseteq B^{\downarrow\uparrow} \subseteq A^{\downarrow\uparrow\downarrow\uparrow} = A^{\downarrow\uparrow}$  y  $\mathbb{K} \models^\# A \rightarrow C$ .
- [Rft] Si  $\mathbb{K} \models^\# Aa \rightarrow b$  entonces  $(A \cup \{a\})^\downarrow = A^\downarrow \cap \{a\}^\downarrow \subseteq \{b\}^\downarrow$ . Vamos a probar que  $(A \cup \{\overline{b}\})^\downarrow \subseteq \{\overline{a}\}^\downarrow$ . Considerando cualquier objeto  $o \in (A \cup \{\overline{b}\})^\downarrow = A^\downarrow \cap \{\overline{b}\}^\downarrow$ . Como  $o \in A^\downarrow$  pero  $o \notin \{b\}^\downarrow$ , se tiene que  $o \notin \{a\}^\downarrow$  y  $o \in \{\overline{a}\}^\downarrow$ . Esto es,  $\mathbb{K} \models^\# A\overline{b} \rightarrow \overline{a}$ .
- [Tru] Si  $\mathbb{K} \models^\# a \rightarrow \overline{a}$ , es decir,  $\{a\}^\downarrow \subseteq \{\overline{a}\}^\downarrow$ , entonces, como  $\{a\}^\downarrow \cup \{\overline{a}\}^\downarrow = G$ , se tiene que  $\{\overline{a}\}^\downarrow = G = \emptyset^\downarrow$  y, así,  $\mathbb{K} \models^\# \emptyset \rightarrow \overline{a}$ . □

El siguiente teorema, con su lema previo, caracterizan el cierre sintáctico en términos de conjuntos consistentes completos, y esto será clave para la demostración de la completitud del sistema axiomático.

**Lema 5.1.15** *Sea  $M$  un conjunto finito de atributos y  $\Sigma \subseteq \mathcal{L}_M$ . Para todo conjunto  $C \in \text{mcl}(\Sigma) \cap \text{Ctts}(M)$  existe  $F \in \text{mcl}(\Sigma) \cap \text{FC}(M)$  tal que  $C \subseteq F$ .*

DEMOSTRACIÓN: Consideremos  $C \in \text{mcl}(\Sigma) \cap \text{Ctts}(M)$ . Como  $M$  es finito, existe un conjunto maximal  $C_1 \in \text{mcl}(\Sigma) \cap \text{Ctts}(M)$  tal que  $C \subseteq C_1$  y probamos que  $C_1 \in \text{FC}(M)$  por reducción al absurdo. Supongamos  $C_1 \notin \text{FC}(M)$ . Entonces existe  $m \in M$  tal que  $m, \bar{m} \notin C_1$  y, por la Proposición 5.1.7,  $(C_1 \cup \{m\})_{\Sigma}^{\#} = M \cup \bar{M}$  porque  $C_1$  es maximal en  $\text{mcl}(\Sigma) \cap \text{Ctts}(M)$ . Por tanto, el Lema 5.1.10 asegura que  $\Sigma \vdash^{\#} C_1 m \rightarrow M\bar{M}$ . Ahora, distinguimos dos casos:

- Si  $C_1 = \emptyset$  entonces  $\Sigma \vdash^{\#} m \rightarrow M\bar{M}$  y, por [Frag],  $\Sigma \vdash^{\#} m \rightarrow \bar{m}$ . Ahora, por [Tru],  $\Sigma \vdash^{\#} \emptyset \rightarrow \bar{m}$  que contradice  $C_1 \in \text{mcl}(\Sigma)$  ya que, como dijimos antes,  $\bar{m} \notin C_1$ .
- En el caso en que  $C_1$  no es vacío, consideremos un  $c \in C_1$  arbitrario. Como  $\Sigma \vdash^{\#} C_1 m \rightarrow M\bar{M}$ , aplicando [Frag], se tiene que  $\Sigma \vdash^{\#} C_1 m \rightarrow \bar{c}$  y, por [Rft],  $\Sigma \vdash^{\#} C_1 \rightarrow \bar{m}$  porque  $C_1 \cup \{c\} = C_1$ . Como consecuencia,  $\bar{m} \in C_1^{\#}_{\Sigma}$  que vuelve a contradecir que  $C_1 \in \text{mcl}(\Sigma)$ .

Concluimos, por tanto, que  $C \subseteq C_1 \in \text{mcl}(\Sigma) \cap \text{FC}(M)$ . □

El siguiente teorema asegura que el operador de cierre sintáctico puede ser descrito en términos de conjuntos m-cerrados que son consistentes completos.

**Teorema 5.1.16** *Sea  $M$  un conjunto finito de atributos,  $\Sigma \subseteq \mathcal{L}_M$  y  $A \subseteq M \cup \bar{M}$ . Entonces  $A_{\Sigma}^{\#} = \bigcap \{F \in \text{mcl}(\Sigma) \cap \text{FC}(M) \mid A \subseteq F\}$ .*



DEMOSTRACIÓN: Es suficiente probar que  $C = \bigcap \{F \in \text{mcl}(\Sigma) \cap \mathbb{FC}(M) \mid C \subseteq F\}$  para cualquier conjunto m-cerrado  $C \in \text{mcl}(\Sigma)$ .

Si  $C = M \cup \overline{M}$ , es trivial porque la intersección vacía es  $M \cup \overline{M}$ .

Si  $C = \emptyset$  debemos probar que  $\bigcap \{F \in \mathbb{FC}(M) \mid F \in \text{mcl}(\Sigma)\} = \emptyset$  y lo hacemos por reducción al absurdo. Supongamos que existe  $b \in F$  para todo  $F \in \text{mcl}(\Sigma) \cap \mathbb{FC}(M)$ . Luego no existe  $F \in \text{mcl}(\Sigma) \cap \mathbb{FC}(M)$  tal que  $\bar{b} \in F$ . Por el Lema 5.1.15 y la Proposición 5.1.7,  $\{\bar{b}\}_\Sigma^\# = M \cup \overline{M}$  y, por el Lema 5.1.10,  $\Sigma \vdash^\# \bar{b} \rightarrow M\overline{M}$ . Por [Frag],  $\Sigma \vdash^\# \bar{b} \rightarrow b$  y, por [Tru],  $\Sigma \vdash^\# \emptyset \rightarrow b$  que contradice  $C \in \text{mcl}(\Sigma)$ .

Nos queda por probar el caso en que  $\emptyset \neq C \neq M \cup \overline{M}$ . Es trivial que  $C \subseteq \bigcap \{F \in \text{mcl}(\Sigma) \cap \mathbb{FC}(M) \mid C \subseteq F\}$  y probamos la otra inclusión, es decir, que  $b \in \bigcap \{F \in \text{mcl}(\Sigma) \cap \mathbb{FC}(M) \mid C \subseteq F\}$  implica  $b \in C$ .

Si  $b \in \bigcap \{F \in \text{mcl}(\Sigma) \cap \mathbb{FC}(M) \mid C \subseteq F\}$ , no existe  $F \in \text{mcl}(\Sigma) \cap \mathbb{FC}(M)$  tal que  $C \cup \{\bar{b}\} \subseteq F$ . Por el Lema 5.1.15 y la Proposición 5.1.7 tenemos que  $(C \cup \{\bar{b}\})_\Sigma^\# = M \cup \overline{M}$  y, por el Lema 5.1.10,  $\Sigma \vdash^\# C\bar{b} \rightarrow M\overline{M}$ . Considerando ahora un  $c \in C$  arbitrario y aplicando [Frag], se tiene que  $\Sigma \vdash^\# C\bar{b} \rightarrow \bar{c}$  y, por [Rft],  $\Sigma \vdash^\# C \rightarrow b$ . Finalmente, como  $C$  es m-cerrado,  $b \in C$ .  $\square$

Añadiendo a este teorema los resultados de la sección anterior, tenemos tres caracterizaciones equivalentes para el cierre sintáctico:

$$\begin{aligned} A_\Sigma^\# &= \bigcap \{C \in \text{mcl}(\Sigma) \mid A \subseteq C\} \\ &= \bigcap \{F \in \text{mcl}(\Sigma) \cap \mathbb{FC}(M) \mid A \subseteq F\} \\ &= \text{máx}\{B \subseteq M \cup \overline{M} \mid \Sigma \vdash^\# A \rightarrow B\} \end{aligned}$$

Ahora, vamos a probar la completitud del sistema axiomático **MixAtL**.

**Teorema 5.1.17** *Sea  $M$  un conjunto finito de atributos,  $A \rightarrow B \in \mathfrak{L}_M$  y  $\Sigma \subseteq \mathfrak{L}_M$ . Entonces,  $\Sigma \models^\# A \rightarrow B$  implica  $\Sigma \vdash^\# A \rightarrow B$ .*

DEMOSTRACIÓN: Probamos que  $\Gamma \not\vdash^\# A \rightarrow B$  implica  $\Gamma \not\models^\# A \rightarrow B$ . Por el Teorema 5.1.11 se tiene que  $\Gamma \not\vdash^\# A \rightarrow B$  implica  $B \not\subseteq A_\Sigma^\#$ . Considerando el

contexto formal  $\mathbb{K} = \langle G, M, I \rangle$  donde  $G = \text{mcl}(\Sigma) \cap \mathbb{F}\mathbb{C}(M)$  y  $I \subseteq G \times M$  donde  $\langle F, m \rangle \in I$  si y solo si  $m \in F$ . Para asegurar que  $\mathbb{K} \models^\# \Sigma$  pero  $\mathbb{K} \not\models^\# A \rightarrow B$ , probamos que  $X^{\downarrow\uparrow} = X_\Sigma^\#$  para todo  $X \subseteq M \cup \overline{M}$  haciendo uso del Teorema 5.1.16:

$$\begin{aligned} X^\downarrow &= \{F \in G \mid m \in F \text{ para todo } m \in X\} \\ &\quad \cap \{F \in G \mid m \notin F \text{ para todo } \overline{m} \in X, \} \\ &= \{F \in G \mid X \subseteq F\} \\ X^{\downarrow\uparrow} &= \{m \in M \mid m \in F \text{ para todo } F \in X^\downarrow\} \\ &\quad \cup \{\overline{m} \in \overline{M} \mid m \notin F \text{ para todo } F \in X^\downarrow\} \\ &= \{a \in M \cup \overline{M} \mid a \in F \text{ para todo } F \in X^\downarrow\} \\ &= \bigcap \{F \in G \mid X \subseteq F\} = X_\Sigma^\# \end{aligned}$$

Por tanto,  $\mathbb{K} \models^\# \Sigma$  y  $\mathbb{K} \not\models^\# A \rightarrow B$ , y, en consecuencia,  $\Sigma \not\models^\# A \rightarrow B$ .  $\square$

Un problema interesante cuando se quiere trabajar con implicaciones de atributos es determinar si un conjunto de implicaciones representa todo el conocimiento que de un contexto formal se puede obtener. De este modo surge la noción de sistema implicacional completo para un contexto dado. En la siguiente definición se extiende al tratamiento de implicaciones mixtas.

**Definición 5.1.18** Sea  $\mathbb{K}$  un contexto formal. Un conjunto de implicaciones entre atributos mixtos  $\Sigma \subseteq \mathcal{L}_M$  se dice que es **m-completo para  $\mathbb{K}$**  si se cumple la siguiente condición: para todo  $A \rightarrow B \in \mathcal{L}_M$ ,

$$\mathbb{K} \models^\# A \rightarrow B \quad \text{si y solo si} \quad \Sigma \vdash^\# A \rightarrow B.$$

Como consecuencia de la corrección y completitud del sistema axiomático de MixAtL se tiene el siguiente resultado que asegura que el cierre semántico y el sintáctico coinciden.

**Teorema 5.1.19** Sea  $\mathbb{K} = \langle G, M, I \rangle$  un contexto formal finito y  $\Sigma \subseteq \mathcal{L}_M$  un conjunto de implicaciones entre atributos mixtos. Si  $\Sigma$  es  $m$ -completo para  $\mathbb{K}$ , entonces  $A^{\downarrow\uparrow} = A_{\Sigma}^{\#}$  para todo  $A \subseteq M \cup \overline{M}$ .

DEMOSTRACIÓN: La demostración es directa ya que, para todo  $A, B \subseteq M \cup \overline{M}$ , se tiene que:

$$\begin{aligned} B \subseteq A^{\downarrow\uparrow} & \quad \text{si y solo si} \quad \mathbb{K} \models^{\#} A \rightarrow B \\ & \quad \text{si y solo si} \quad \Sigma \vdash^{\#} A \rightarrow B \quad \text{si y solo si} \quad B \subseteq A_{\Sigma}^{\#}. \end{aligned}$$

□

## 5.2. SimMixAtL: Lógica de simplificación para atributos mixtos

Introducimos en esta sección un sistema axiomático alternativo al de **MixAtL** inspirado en el paradigma de simplificación y demostraremos que es equivalente al de **MixAtL**.

El sistema axiomático de **SimMixAtL** considera un axioma y cuatro reglas de inferencia. Son las siguientes:

$$\begin{aligned} [\text{Ref}] \quad \text{Reflexividad:} & \quad \vdash_s^{\#} A \rightarrow A \\ [\text{Simp}] \quad \text{Simplificación:} & \quad A \rightarrow B, C \rightarrow D \quad \vdash_s^{\#} A(C-B) \rightarrow D \\ [\text{Ky}] \quad \text{Clave:} & \quad A \rightarrow b \quad \vdash_s^{\#} A\bar{b} \rightarrow M\overline{M} \\ [\text{InKy}] \quad \text{Clave Inversa:} & \quad Ab \rightarrow M\overline{M} \quad \vdash_s^{\#} A \rightarrow \bar{b} \\ [\text{Rd}] \quad \text{Reducción:} & \quad Ab \rightarrow C, A\bar{b} \rightarrow C \quad \vdash_s^{\#} A \rightarrow C \end{aligned}$$

para todo  $a, b \in M \cup \overline{M}$  y todo  $A, B, C \subseteq M \cup \overline{M}$ .

El siguiente teorema establece la equivalencia entre los sistemas axiomáticos de **MixAtL** y **SimMixAtL**.

**Teorema 5.2.1** *Sea  $M$  un conjunto finito de atributos,  $\Sigma \subseteq \mathcal{L}_M$  y  $A \rightarrow B \in \mathcal{L}_M$ . Se tiene que:*

$$\Sigma \vdash^\# A \rightarrow B \quad \text{si y solo si} \quad \Sigma \vdash_s^\# A \rightarrow B$$

DEMOSTRACIÓN: En primer lugar demostramos que los axiomas y las reglas de inferencia de **SimMixAtL** son derivadas de las de **MixAtL** :

[Ref] Es un caso particular de [Inc].

[Simp] La siguiente secuencia lo demuestra:

$$\begin{aligned} \varphi_1 &= A \rightarrow B && \text{Hipótesis.} \\ \varphi_2 &= C \rightarrow D && \text{Hipótesis.} \\ \varphi_3 &= A(C-B) \rightarrow BC && \text{Aplicando [Augm] a } \varphi_1 \text{ y teniendo} \\ &&& \text{en cuenta que } B \cup (C \setminus B) = B \cup C. \\ \varphi_4 &= BC \rightarrow BD && \text{Aplicando [Augm] a } \varphi_2. \\ \varphi_5 &= A(C-B) \rightarrow BD && \text{Aplicando [Trans] a } \varphi_3 \text{ y } \varphi_4. \\ \varphi_6 &= BD \rightarrow D && \text{Por [Inc].} \\ \varphi_7 &= A(C-B) \rightarrow D && \text{Aplicando [Trans] a } \varphi_5 \text{ y } \varphi_6. \end{aligned}$$

[Ky] La siguiente secuencia prueba que  $A \rightarrow b \vdash^\# A\bar{b} \rightarrow M\bar{M}$ :

$$\begin{aligned} \varphi_1 &= A \rightarrow b && \text{Hipótesis.} \\ \varphi_2 &= A\bar{b} \rightarrow b\bar{b} && \text{Aplicando [Augm] a } \varphi_1. \\ \varphi_3 &= b\bar{b} \rightarrow M\bar{M} && \text{Por [Cont].} \\ \varphi_4 &= A\bar{b} \rightarrow M\bar{M} && \text{Aplicando [Trans] a } \varphi_2 \text{ y } \varphi_3. \end{aligned}$$

[InKy] Para demostrar que  $A\bar{b} \rightarrow M\bar{M} \vdash^\# A \rightarrow b$  distinguimos dos casos:

- Si  $A = \emptyset$ , la siguiente secuencia demuestra que  $\bar{b} \rightarrow M\bar{M} \vdash^\# \emptyset \rightarrow b$ .

$$\begin{aligned}
\varphi_1 &= \bar{b} \rightarrow M\bar{M} && \text{Hipótesis.} \\
\varphi_2 &= M\bar{M} \rightarrow b && \text{Por [Ref].} \\
\varphi_3 &= \bar{b} \rightarrow b && \text{Aplicando [Trans] a } \varphi_1 \text{ y } \varphi_2. \\
\varphi_4 &= \emptyset \rightarrow b && \text{Aplicando [Tru] a } \varphi_3.
\end{aligned}$$

- Si  $A \neq \emptyset$ , considerando un  $a \in A$  arbitrario, podemos dar la siguiente demostración para  $A\bar{b} \rightarrow M\bar{M} \vdash^\# A \rightarrow b$ :

$$\begin{aligned}
\varphi_1 &= A\bar{b} \rightarrow M\bar{M} && \text{Hipótesis.} \\
\varphi_2 &= M\bar{M} \rightarrow \bar{a} && \text{Por [Ref].} \\
\varphi_3 &= A\bar{b} \rightarrow \bar{a} && \text{Aplicando [Trans] a } \varphi_1 \text{ y } \varphi_2. \\
\varphi_4 &= A \rightarrow b && \text{Aplicando [Rft] a } \varphi_3 \text{ ya que } a \in A.
\end{aligned}$$

[Rd] La siguiente secuencia prueba que es una regla de inferencia derivada en **MixAtL** :

$$\begin{aligned}
\varphi_1 &= Ab \rightarrow c && \text{Hipótesis.} \\
\varphi_2 &= A\bar{b} \rightarrow c && \text{Hipótesis.} \\
\varphi_3 &= A\bar{c} \rightarrow \bar{b} && \text{Aplicando [Rft] a } \varphi_1. \\
\varphi_4 &= A\bar{c} \rightarrow b && \text{Aplicando [Rft] a } \varphi_2. \\
\varphi_5 &= A\bar{c} \rightarrow b\bar{b} && \text{Aplicando [Un] a } \varphi_3 \text{ y } \varphi_4. \\
\varphi_6 &= b\bar{b} \rightarrow M\bar{M} && \text{Por [Cont].} \\
\varphi_7 &= A\bar{c} \rightarrow M\bar{M} && \text{Aplicando [Trans] a } \varphi_5 \text{ y } \varphi_6. \\
\varphi_8 &= A \rightarrow c && \text{Aplicando [InKy] a } \varphi_7.
\end{aligned}$$

Obsérvese que, en el último paso de esta demostración, hemos utilizado [InKy] que ya habíamos demostrado que era una regla de inferencia derivada en **MixAtL** .

Con todo esto hemos demostrado que  $\Sigma \vdash_s^\# A \rightarrow B$  implica  $\Sigma \vdash^\# A \rightarrow B$ . Demostramos ahora que  $\Sigma \vdash^\# A \rightarrow B$  implica  $\Sigma \vdash_s^\# A \rightarrow B$  probando que todos los axiomas y reglas de inferencia de **MixAtL** son derivados de los de **SimMixAtL**.

[Inc] La siguiente secuencia lo prueba:

$$\begin{aligned}\varphi_1 &= A \rightarrow A && \text{Por [Ref].} \\ \varphi_2 &= B \rightarrow B && \text{Por [Ref].} \\ \varphi_3 &= AB \rightarrow A && \text{Aplicando [Simp] a } \varphi_2 \text{ y } \varphi_1 \text{ y teniendo} \\ &&& \text{en cuenta que } B \cup (A \setminus B) = A \cup B.\end{aligned}$$

[Cont] Lo probamos con la siguiente secuencia:

$$\begin{aligned}\varphi_1 &= a \rightarrow a && \text{Por [Ref].} \\ \varphi_2 &= a\bar{a} \rightarrow M\bar{M} && \text{Aplicando [KY] a } \varphi_1.\end{aligned}$$

[Trans] Es un caso particular de [Simp]:

$$\begin{aligned}\varphi_1 &= A \rightarrow B && \text{Hipótesis.} \\ \varphi_2 &= B \rightarrow C && \text{Hipótesis.} \\ \varphi_3 &= A \rightarrow C && \text{Aplicando [Simp] a } \varphi_1 \text{ y } \varphi_2 \text{ y teniendo} \\ &&& \text{en cuenta que } A \cup (B \setminus B) = A.\end{aligned}$$

[Augm] La demostración es la siguiente:

$$\begin{aligned}\varphi_1 &= A \rightarrow B && \text{Hipótesis.} \\ \varphi_2 &= B \rightarrow B && \text{Por [Ref].} \\ \varphi_3 &= B-C \rightarrow B-C && \text{Por [Ref].} \\ \varphi_4 &= B \rightarrow B-C && \text{Aplicando [Simp] a } \varphi_2 \text{ y } \varphi_3 \text{ y teniendo} \\ &&& \text{en cuenta que } (B \setminus C) \setminus B = \emptyset. \\ \varphi_5 &= A \rightarrow B-C && \text{Aplicando [Simp] a } \varphi_1 \text{ y } \varphi_4 \text{ y teniendo} \\ &&& \text{en cuenta que } A \cup (B \setminus B) = A. \\ \varphi_6 &= BC \rightarrow BC && \text{Por [Ref].} \\ \varphi_7 &= AC \rightarrow BC && \text{Aplicando [Simp] a } \varphi_5 \text{ y } \varphi_6 \text{ y teniendo} \\ &&& \text{en cuenta que } A \cup ((B \cup C) \setminus (B \setminus C)) = \\ &&& A \cup C.\end{aligned}$$

[Rft] Se prueba como sigue:

$$\begin{aligned}
\varphi_1 &= Aa \rightarrow b && \text{Hipotesis.} \\
\varphi_2 &= Aa\bar{b} \rightarrow M\bar{M} && \text{Aplicando [KY] a } \varphi_1. \\
\varphi_3 &= A\bar{b} \rightarrow \bar{a} && \text{Aplicando [InKY] a } \varphi_2.
\end{aligned}$$

[Tru] La siguiente secuencia lo demuestra:

$$\begin{aligned}
\varphi_1 &= a \rightarrow \bar{a} && \text{Hipotesis.} \\
\varphi_2 &= a \rightarrow a && \text{Por [Ref].} \\
\varphi_3 &= a\bar{a} \rightarrow M\bar{M} && \text{Aplicando [KY] a } \varphi_2. \\
\varphi_4 &= a \rightarrow M\bar{M} && \text{Aplicando [Simp] a } \varphi_1 \text{ y } \varphi_3. \\
\varphi_5 &= \emptyset \rightarrow \bar{a} && \text{Aplicando [InKY] a } \varphi_4.
\end{aligned}$$

□

**Corolario 5.2.2** *El sistema axiomático de **SimMixAtL** es correcto y completo.*

**Ejemplo 5.2.3** *Sea  $M = \{m_1, m_2, m_3\}$  y  $\Sigma = \{m_1m_2 \rightarrow m_3, m_1\bar{m}_3 \rightarrow m_2\}$ . En el ejemplo 5.1.13 dimos una demostración de  $\Sigma \vdash^{\#} \bar{m}_3 \rightarrow \bar{m}_1\bar{m}_3$  usando **MixAtL**. A continuación presentamos una demostración basada en el sistema axiomático **SimMixAtL**:*

$$\begin{aligned}
\varphi_1 &= m_1m_2 \rightarrow m_3 && \text{Hipótesis.} \\
\varphi_2 &= m_1\bar{m}_3 \rightarrow m_2 && \text{Hipótesis.} \\
\varphi_3 &= m_1m_2\bar{m}_3 \rightarrow M\bar{M} && \text{Aplicando [KY] a } \varphi_1. \\
\varphi_4 &= m_1\bar{m}_2\bar{m}_3 \rightarrow M\bar{M} && \text{Aplicando [KY] a } \varphi_2. \\
\varphi_5 &= m_1\bar{m}_3 \rightarrow M\bar{M} && \text{Aplicando [Rd] a } \varphi_3 \text{ y } \varphi_4. \\
\varphi_6 &= \bar{m}_3 \rightarrow \bar{m}_1 && \text{Aplicando [InKY] a } \varphi_5. \\
\varphi_7 &= \bar{m}_1\bar{m}_3 \rightarrow \bar{m}_1\bar{m}_3 && \text{Por [Ref].} \\
\varphi_8 &= \bar{m}_3 \rightarrow \bar{m}_1\bar{m}_3 && \text{Aplicando [Simp] a } \varphi_6 \text{ y } \varphi_7.
\end{aligned}$$

Del mismo modo que en la Lógica de Simplificación clásica, las reglas de inferencia de este nuevo sistema axiomático se pueden ver, en realidad, como equivalencias.

Como dijimos en la sección 5.1.1, dos conjuntos  $\Sigma_1, \Sigma_2 \subseteq \mathcal{L}_M$  son semánticamente equivalentes, denotado por  $\Sigma_1 \equiv \Sigma_2$ , si para todo contexto formal  $\mathbb{K}$  se tiene que

$$\mathbb{K} \models^\# \Sigma_1 \quad \text{si y solo si} \quad \mathbb{K} \models^\# \Sigma_2$$

Dado que el sistema axiomático de **MixAtL** es correcto y completo se tiene que  $\Sigma_1 \equiv \Sigma_2$  si y solo si las dos siguientes implicaciones se cumplen:

1.  $\Sigma_1 \vdash^\# A \rightarrow B$  para todo  $A \rightarrow B \in \Sigma_2$ .
2.  $\Sigma_2 \vdash^\# A \rightarrow B$  para todo  $A \rightarrow B \in \Sigma_1$ .

A las equivalencias conocidas de la Lógica de Simplificación (véase Teorema 2.3.25) se añaden en **SimMixAtL** las enunciadas en el siguiente teorema.

**Teorema 5.2.4** *Sea  $M$  un conjunto finito de atributos. Las siguientes equivalencias se cumplen en **SimMixAtL**: para todo  $A, C \subseteq M \cup \overline{M}$  y  $b \in M \cup \overline{M}$ ,*

$$(\text{Rd} \cdot \text{Eq}) \quad \textit{Equivalencia de reducción: } \{Ab \rightarrow C, A\bar{b} \rightarrow C\} \equiv \{A \rightarrow C\}.$$

$$(\text{Ky} \cdot \text{Eq}) \quad \textit{Equivalencia clave: } \{A \rightarrow b\} \equiv \{A\bar{b} \rightarrow M\overline{M}\}.$$

DEMOSTRACIÓN: Por [Rd] tenemos que  $Ab \rightarrow C, A\bar{b} \rightarrow C \vdash_s^\# A \rightarrow C$  y la siguiente secuencia demuestra que  $A \rightarrow C \vdash_s^\# Ab \rightarrow C$ :

$$\begin{aligned} \varphi_1 &= A \rightarrow C && \text{Hipótesis.} \\ \varphi_2 &= Ab \rightarrow Ab && \text{Por [Ref].} \\ \varphi_3 &= Ab \rightarrow C && \text{Aplicando [Simp] a } \varphi_2 \text{ y } \varphi_1. \end{aligned}$$

La demostración para  $A \rightarrow C \vdash_s^\# A\bar{b} \rightarrow C$  es análoga.

Finalmente, la equivalencia clave es consecuencia directa de [Ky] y [InKy].  $\square$





UNIVERSIDAD  
DE MÁLAGA

## Capítulo 6

# Extracción de conocimiento

En los capítulos anteriores se han estudiado a fondo las propiedades de los retículos de conceptos mixtos y de los sistemas implicacionales mixtos. También hemos presentado dos sistemas axiomáticos equivalentes que permiten inferir nuevas implicaciones mixtas a partir de un sistema implicacional. Teniendo en cuenta todos estos resultados, en este capítulo abordamos el problema de la extracción de conocimiento a partir de la información contenida en un contexto formal. En particular, presentamos varios algoritmos, inspirados en los clásicos, que nos permiten extraer conocimiento a partir del contexto formal en términos de, o bien el retículo de conceptos mixtos, o bien un sistema implicacional mixto completo.

### 6.1. Calculando sistemas implicacionales mixtos completos

Uno de los algoritmos de referencia para calcular un sistema implicacional completo a partir de un contexto formal es el algoritmo conocido como `Next-Closure` [40]. Este algoritmo devuelve la base Duquenne-Guigues,

también llamada “*stem base*”, que es el sistema implicacional completo con menor cardinal. La esencia del algoritmo es recorrer la familia de subconjuntos de atributos en orden léxico (orden lineal que subsume a la inclusión de conjuntos) aplicando una estrategia que evita repeticiones y el cálculo de cierres innecesarios.

El algoritmo presentado en esta sección, está inspirado en el algoritmo *Next-Closure*. Básicamente recorre los subconjuntos  $A$  de  $M \cup \overline{M}$  que pueden ser la premisa de una implicación  $A \rightarrow A^{\downarrow\uparrow}$  que no sea trivial (que no es un axioma) o que no se pueda inferir de una previamente calculada. Así, el algoritmo considera solamente subconjuntos consistentes (ver Definiciones 4.1.2 y 4.1.3).

**Lema 6.1.1** Sea  $\mathbb{K} = \langle G, M, I \rangle$  un contexto formal y  $A \subseteq M \cup \overline{M}$ .

1. Si  $A \notin \text{Ctts}(M)$  entonces  $A \rightarrow A^{\downarrow\uparrow}$  es un axioma.
2. Si  $A \in \text{FC}(M)$  entonces  $A \rightarrow A^{\downarrow\uparrow}$  es un axioma o existe  $B \subsetneq A$  tal que  $B \rightarrow B^{\downarrow\uparrow} \vdash^{\#} A \rightarrow A^{\downarrow\uparrow}$ .

**DEMOSTRACIÓN:** Para el ítem 1, debemos probar que  $\emptyset \vdash^{\#} A \rightarrow A^{\downarrow\uparrow}$  para todo  $A \notin \text{Ctts}(M)$ . Como  $A$  no es consistente, debe existir  $m \in M$  tal que  $m, \overline{m} \in A$  y  $A^{\downarrow\uparrow} = M \cup \overline{M}$ . La siguiente secuencia lo demuestra:

$$\begin{aligned} \varphi_1 &= A \rightarrow m\overline{m} && \text{por [Inc]}. \\ \varphi_2 &= m\overline{m} \rightarrow M\overline{M} && \text{por [Cont]}. \\ \varphi_3 &= A \rightarrow M\overline{M} && \text{aplicando [Trans] a } \varphi_1 \text{ y } \varphi_2. \end{aligned}$$

En el caso del ítem 2, si  $A \in \text{FC}(M)$ , debemos distinguir dos casos:

1. Si existe  $g \in G$  tal que  $g^{\uparrow} = A$ , por el Teorema 4.2.3,  $A^{\downarrow\uparrow} = A$  y, por [Inc], tenemos que  $A \rightarrow A^{\downarrow\uparrow}$  es un axioma.
2. Si no existe  $g \in G$  tal que  $g^{\uparrow} = A$ , por el Teorema 5.1.19 y la Proposición 5.1.7, tenemos que  $A^{\downarrow\uparrow} = M \cup \overline{M}$ . Consideremos cualquier  $b \in A$  y

$B = A \setminus \{b\}$ . Por el Teorema 5.2.4, se tiene que  $\{A \rightarrow A^{\downarrow\uparrow}\} \equiv \{B \rightarrow \bar{b}\}$  y, por tanto,  $B \rightarrow B^{\downarrow\uparrow} \vdash A \rightarrow A^{\downarrow\uparrow}$ .

□

Como consecuencia del lema anterior, tenemos que, para encontrar un sistema implicacional completo, podemos limitar la búsqueda a aquellas implicaciones cuya premisa pertenece a  $\mathbb{C}tts(M) \setminus \mathbb{F}C(M)$ . Por tal motivo el algoritmo va a extender el orden léxico para recorrer todo este conjunto.

Para ello, vamos a dar una caracterización de conjuntos en  $\mathbb{C}tts(M)$  en términos de pares de subconjuntos de  $M$ .

Existe una relación uno-a-uno entre conjuntos en  $\mathbb{C}tts(M)$  y pares de conjuntos  $\langle X, Y \rangle$  tal que  $X \subseteq Y \subseteq M$ . Si  $A \in \mathbb{C}tts(M)$  entonces  $\text{Neg}(A) \subseteq \text{Tot}(A) \subseteq M$ . Por otro lado, si  $X \subseteq Y \subseteq M$  entonces  $(Y \setminus X) \cup \bar{X} \in \mathbb{C}tts(M)$ . Es más,

$$A = \left( \text{Tot}(A) \setminus \text{Neg}(A) \right) \cup \overline{\text{Neg}(A)} \quad (6.1)$$

Esta relación uno-a-uno permite extender el orden léxico a  $\mathbb{C}tts(M)$  como sigue: para todo  $A, B \in \mathbb{C}tts(M)$ , el conjunto  $A$  es previo a  $B$ , denotado como  $A \ll B$ , si se da una de las siguientes condiciones:

1.  $\text{Tot}(A) < \text{Tot}(B)$
2.  $\text{Tot}(A) = \text{Tot}(B)$  y  $\text{Neg}(A) < \text{Neg}(B)$

Obsérvese que, si  $A \in \mathbb{C}tts(M) \setminus \mathbb{F}C(M)$  y  $B \in \mathbb{F}C(M)$ , entonces  $A \ll B$ .

Este nuevo orden conserva el orden de la inclusión para subconjuntos (es decir,  $A \subseteq B$  implica  $A \ll B$ ) y permite podar la búsqueda de nuevas implicaciones como mínimo hasta la mitad de los candidatos. En lugar de recorrer  $2^{M \cup \bar{M}}$  recorreremos  $\mathbb{C}tts(M)$  en el orden  $\ll$  hasta que aparezca el primer elemento de  $\mathbb{F}C(M)$ .

Por otro lado, el algoritmo para calcular el sistema implicacional no necesita recorrer exhaustivamente todos los subconjuntos de atributos mixtos,

sino solamente aquellos que son cerrados con respecto al sistema implicacional previamente calculado. La función **Closed**, con coste lineal, devuelve “verdadero” siempre que el conjunto  $A$  es cerrado respecto del conjunto de implicaciones  $\Sigma$ .

---

**Función Closed( $A, \Sigma$ ): booleano**


---

**Entrada:**  $A \in \text{Ctts}(M)$ , y  $\Sigma$  un conjunto de implicaciones mixtas.

**Salida:** “verdadero” si  $A$  es cerrado con respecto a  $\Sigma$ .

```

1 inicio
2   para cada  $B \rightarrow C \in \Sigma$  hacer
3     si  $B \subseteq A$  y  $C \not\subseteq A$  entonces devolver falso;
4     si  $B \setminus A = \{a\}$ ,  $A \cap \bar{C} \neq \emptyset$ , y  $\bar{a} \notin A$  entonces devolver falso;
5   devolver verdadero

```

---



---

**Algoritmo 3: Extracción de implicaciones mixtas**


---

**Entrada:**  $\mathbb{K} = \langle G, M, I \rangle$

**Salida:** Un sistema implicacional  $\Sigma$  que es completo para  $\mathbb{K}$ .

```

1 inicio
2    $\Sigma := \emptyset$ ;
3    $Y := \emptyset$ ;
4   mientras  $Y < M$  hacer
5     para cada  $X \subseteq Y$  hacer
6        $A := (Y \setminus X) \cup \bar{X}$ ;
7       si Closed( $A, \Sigma$ ) entonces
8          $C := A^{\downarrow\uparrow}$ ;
9         si  $A \neq C$  entonces añadir  $A \rightarrow C \setminus A$  a  $\Sigma$ ;
10     $Y := \text{Siguiente}(Y)$  // Esta llamada al método Siguiente( $Y$ )
    devuelve al sucesor de  $Y$  en el orden léxico clásico.
11  devolver  $\Sigma$ 

```

---

Obsérvese que al aplicar los algoritmos clásicos al contexto dicotomizado  $\mathbb{K}|\bar{\mathbb{K}}$  tendríamos  $4^{|M|}$  candidatos a ser premisa de las implicaciones. Además, sería necesario aplicar técnicas de eliminación de redundancias para reducir el tamaño del sistema implicacional conforme a los sistemas de inferencia

descritos en el capítulo anterior. Con el método que acabamos de introducir, el número de candidatos se reduce a  $3^{|M|} - 2^{|M|}$  y, de ellos, solo conllevan consulta al contexto aquellos que son cerrados respecto de las implicaciones calculadas con anterioridad.

**Ejemplo 6.1.2** Consideremos el contexto de los “países en vías de desarrollo” presentado en [40] y ya mencionado en el Ejemplo 2.3.34. Forman el contexto 130 países y 6 atributos: Group of 77, Non-aligned, LDC (Least Developed Countries), MASC (Most Seriously Affected Countries), OPEC (Organization of Petrol Exporting Countries) and ACP (African, Caribbean and Pacific Countries). Las implicaciones (positivas) que conforman la base Duquenne-Guigues son las siguientes:

OPEC  $\rightarrow$  Group of 77, Non-aligned  
 MASC  $\rightarrow$  Group of 77  
 Non-aligned  $\rightarrow$  Group of 77  
 Group of 77, Non-aligned, MASC, OPEC  $\rightarrow$  LLDC, ACP  
 Group of 77, Non-aligned, LLDC, OPEC  $\rightarrow$  MASC, ACP

El Algoritmo 3 calcula el siguiente sistema implicacional mixto:

OPEC  $\rightarrow$  Group of 77, Non-aligned,  $\overline{LLDC}$ ,  $\overline{MASC}$   
 MASC,  $\overline{OPEC}$   $\rightarrow$  Group of 77  
 $\overline{LLDC}$ , ACP  $\rightarrow$  Group of 77  
 $\overline{Non-aligned}$ , LLDC,  $\overline{MASC}$ ,  $\overline{OPEC}$   $\rightarrow$   $\overline{Group\ of\ 77}$   
 Non-aligned  $\rightarrow$  Group of 77  
 Group of 77,  $\overline{LLDC}$ , MASC,  $\overline{OPEC}$ , ACP  $\rightarrow$  Non-aligned

Como conclusión, en la ejecución del método, el contexto formal se comprueba solamente 124 veces, mientras que el número de conjuntos consistentes es 729. El número de implicaciones en el sistema implicacional mixto es 6, justamente solo una implicación más que la base clásica Duquenne-Guigues.

Obsérvese que el número de implicaciones mixtas generadas es ligeramente superior al número de las implicaciones clásicas.

**Ejemplo 6.1.3** *En este segundo ejemplo, se compara el sistema implicacional obtenido mediante el método presentado con el obtenido con el algoritmo propuesto en [75]. Para el contexto formal proporcionado en la Tabla 5.1, el algoritmo de Missaoui et al. proporciona el siguiente sistema implicacional mixto:*

$$\begin{aligned} \Sigma_1 = \{ & \overline{m_4} \rightarrow m_1 m_2 \overline{m_3}, m_3 \rightarrow \overline{m_1} m_2 m_4, \overline{m_2} \rightarrow m_1 \overline{m_3} m_4, \\ & m_2 m_4 \rightarrow \overline{m_1}, \overline{m_1} \rightarrow m_2 m_4, m_1 \rightarrow \overline{m_3}, m_1 m_4 \rightarrow \overline{m_2} \overline{m_3}, \\ & m_1 m_2 \rightarrow \overline{m_3} \overline{m_4}, \overline{m_1} \overline{m_3} \rightarrow m_2 m_4, m_2 \overline{m_3} m_4 \rightarrow \overline{m_1} \} \end{aligned}$$

*El algoritmo propuesto proporciona el siguiente sistema implicacional:*

$$\begin{aligned} \Sigma_2 = \{ & \overline{m_4} \rightarrow m_1 m_2 \overline{m_3}, m_3 m_4 \rightarrow \overline{m_1} m_2, m_2 m_4 \rightarrow \overline{m_1}, \\ & \overline{m_2} \overline{m_3} m_4 \rightarrow m_1, m_1 \rightarrow \overline{m_3}, \overline{m_1} m_4 \rightarrow m_3 \} \end{aligned}$$

*El método propuesto consulta el contexto formal 14 veces mientras que los posibles subconjuntos de atributos asciende a 81. El sistema implicacional mixto generado es menor que el obtenido con el algoritmo anterior: tiene 6 implicaciones en contra de las 10 obtenidas con el algoritmo de Missaoui et al.*

Este método permite generar un sistema implicacional que caracteriza, en forma de implicaciones, las relaciones entre los atributos positivos y negativos. La principal novedad de este enfoque es que se trabaja directamente con el contexto formal original  $\mathbb{K}$  y no con el modificado  $(\mathbb{K}|\overline{\mathbb{K}})$ .

## 6.2. Calculando el retículo de conceptos mixtos

El desarrollo de métodos para calcular eficientemente el retículo de conceptos a partir de un contexto, en el caso clásico, sigue siendo un área muy

activa de investigación. En [63], los autores aseguran que “la elección de un algoritmo para la construcción del retículo de conceptos puede depender de las propiedades de los datos introducidos”, es decir, que aunque hay métodos que, en la mayor parte de los casos, son más eficientes, dependiendo de la naturaleza de los datos a explorar, puede que resulte más interesante utilizar otros.

En este apartado se van a proponer nuevos métodos para extraer los retículos de conceptos mixtos asociados a un contexto formal. Hemos seleccionado los algoritmos más relevantes desarrollados en el caso clásico (Next-Closure, Norris, FCbO e In-Close) para adaptarlos al cálculo del retículo de conceptos mixtos. Para ello, hemos tenido en cuenta las propiedades estudiadas en capítulos anteriores. No nos hemos limitado a los más eficientes del caso clásico para comprobar si siguen siéndolo o no, en esta generalización.

Todos estos métodos trabajan recorriendo los conjuntos de atributos en una búsqueda de arriba hacia abajo o en el sentido inverso. Recuérdese que el Lema 4.1.7 asegura que, para un contexto formal  $\mathbb{K} = \langle G, M, I \rangle$ , el mayor elemento del retículo de conceptos mixto es  $\langle G, G^\uparrow \rangle$  y el menor elemento es  $\langle \emptyset, M \cup \overline{M} \rangle$ . En la Observación 4.1.10 asegurábamos que podíamos suponer, sin pérdida de generalidad, que el mayor elemento es  $\langle G, \emptyset \rangle$  porque, de lo contrario, indicaría la presencia en el contexto de un atributo común a todos los objetos o ausente en todos ellos.

Esta situación es bien conocida en el caso clásico [40], donde se llama “clarificación” o “reducción” al proceso de eliminación de aquellos objetos o atributos que no modifican la estructura del retículo de conceptos, es decir, que cumplen que, si se eliminan, el retículo de conceptos es isomorfo al inicial y además éste puede ser reconstruido. Como recogimos en las Observaciones 4.1.9 y 4.1.10 no todas estas simplificaciones clásicas conservan la estructura del retículo de conceptos mixto, es decir, se puede producir



un nuevo contexto formal con un retículo de conceptos mixto asociado no isomorfo. Así, por ejemplo, si en un contexto formal hay un objeto que tiene todos los atributos y se elimina, debido a la reducción en el número de átomos en el retículo de conceptos mixto, no se puede construir un isomorfismo. Por otro lado, hay nuevas reducciones específicas para el retículo de conceptos mixto.

La siguiente definición extiende la noción de clarificación presentada en [40].

**Definición 6.2.1** *Un contexto  $\mathbb{K} = \langle G, M, I \rangle$  se denomina **clarificado mixto** (acortando, **m-clarificado**) si las siguientes condiciones se cumplen:*

1.  $g_1^\uparrow = g_2^\uparrow$  implica  $g_1 = g_2$  para todo  $g_1, g_2 \in G$ .
2.  $m_1^\downarrow = m_2^\downarrow$  implica  $m_1 = m_2$  para todo  $m_1, m_2 \in M$ .
3.  $m_1^\downarrow \neq \overline{m}_2^\downarrow$  para todo  $m_1, m_2 \in M$ .
4.  $m^\downarrow \neq \emptyset \neq \overline{m}^\downarrow$  para todo  $m \in M$ .

**Ejemplo 6.2.2** *Los contextos presentados en la Tabla 6.1 tienen retículos de conceptos mixtos isomorfos entre sí, e isomorfos al retículo de la Figura 6.1. El contexto  $\mathbb{K}_1$  no es m-clarificado:*

- No se cumple la condición 1 porque  $g_1^\uparrow = g_3^\uparrow$  y  $g_1 \neq g_3$ .
- La condición 2 también falla ya que  $m_3^\downarrow = m_7^\downarrow$ .
- No se cumple 3 ya que  $m_2^\downarrow = \overline{m}_4^\downarrow$  y  $m_5^\downarrow = \overline{m}_6^\downarrow$ .
- Tampoco se cumple la condición 4 porque se tiene que  $m_1^\downarrow = \overline{m}_8^\downarrow = \emptyset$ .

Sin embargo, el contexto  $\mathbb{K}_2$  sí es m-clarificado.

$\mathbb{K}_1$	$m_1$	$m_2$	$m_3$	$m_4$	$m_5$	$m_6$	$m_7$	$m_8$
$g_1$			×	×		×	×	×
$g_2$		×	×		×		×	×
$g_3$			×	×		×	×	×
$g_4$				×	×			×

$\mathbb{K}_2$	$m_2$	$m_3$	$m_5$
$g_1$		×	
$g_2$	×	×	×
$g_4$			×

Tabla 6.1: Contexto formal y su equivalente m-clarificado.

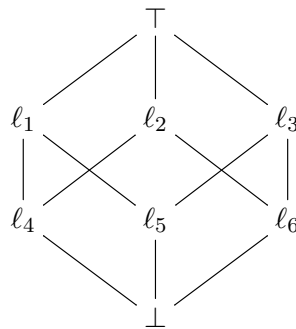


Figura 6.1: Retículo isomorfo a  $\mathfrak{B}^\#(\mathbb{K}_1)$  y  $\mathfrak{B}^\#(\mathbb{K}_2)$  donde  $\mathbb{K}_1$  y  $\mathbb{K}_2$  son los contextos presentados en la Tabla 6.1.

Es fácil probar que, para cada contexto formal, existe un contexto m-clarificado cuyo retículo de conceptos mixto es isomorfo al de partida. El nuevo se obtiene eliminando columnas completas, columnas vacías, columnas duales y filas/columnas repetidas. Nótese que este proceso se puede llevar a cabo con coste cuadrático y que, una vez m-clarificado, se debe cumplir que  $|G| \leq 2^{|M|}$  y que  $|M| < 2^{|G|-1}$ .

De aquí en adelante, se supondrá que todos los contextos formales son m-clarificados.

### 6.2.1. Propuesta basada en el algoritmo Next-Closure

Anteriormente, en la Sección 6.1, se ha propuesto el Algoritmo 3 para calcular un sistema implicacional completo desde un contexto que está inspirado en el algoritmo Next-Closure [40]. En esta sección se va a extender este algoritmo de forma que calcule el conjunto de conceptos mixtos y para ello se va a usar nuevamente el orden léxico extendido.

El núcleo de funcionamiento del algoritmo consiste en recorrer el conjunto de candidatos (que ya hemos reducido) siguiendo el orden léxico y simultáneamente calcular los conjuntos cerrados de atributos y un sistema implicacional. Este conjunto de implicaciones se usa en el algoritmo para evitar tener que comprobar el contexto formal en los casos en los que, del conocimiento previamente descubierto, podamos deducir que un candidato no es cerrado. De esta forma evitamos tener que comprobarlo usando el contexto formal. Obviamente, es interesante solo en el caso en el cual el coste de comprobar el cierre desde las implicaciones sea menor que comprobarlo a partir del contexto.

El nuevo algoritmo (Algoritmo 4) tiene un primer ciclo que recorre la familia de conjuntos consistentes no completos en el orden léxico extendido, buscando conjuntos cerrados y conceptos mixtos. En este punto, se obtiene un sistema implicacional mixto pero no todos los conceptos mixtos son calculados. Es necesario añadir aquellos conceptos mixtos cuyos conjuntos de atributos sean conjuntos consistentes completos (los átomos del retículo de conceptos mixtos) y el concepto mixto mínimo del retículo. El siguiente teorema proporciona una forma directa para calcular esos conceptos mixtos.

**Teorema 6.2.3** *Sea  $\mathbb{K} = \langle G, M, I \rangle$  un contexto formal  $m$ -clarificado y  $A$  un conjunto consistente completo. El par  $\langle A^\downarrow, A \rangle$  es un concepto mixto si y solo si existe  $g \in G$  tal que  $\{g\}^\uparrow = A$  (en este caso,  $A^\downarrow = \{g\}$ ).*

DEMOSTRACIÓN: Si  $\langle A^\downarrow, A \rangle$  es un concepto mixto siendo  $A$  un conjunto

consistente completo,  $A^{\downarrow\uparrow} = A \neq M \cup \overline{M}$  y además  $A^{\downarrow} \neq \emptyset$ . Entonces, para cada  $g \in A^{\downarrow}$ ,  $\{g\}^{\uparrow}$  es un conjunto consistente completo contenido en  $A$ , es decir,  $\{g\}^{\uparrow} = A$ . Finalmente, como  $\mathbb{K}$  es m-clarificado,  $A^{\downarrow} = \{g\}^{\uparrow\downarrow} = \{g\}$ .

A la inversa, para cada  $g \in G$ , como  $\mathbb{K}$  es m-clarificado, se tiene que el par  $\langle \{g\}, \{g\}^{\uparrow} \rangle$  es un concepto mixto tal que  $\{g\}^{\uparrow}$  es un conjunto consistente completo.  $\square$

---

**Algoritmo 4: Next-Closure Mixto**


---

**Entrada:** Un contexto formal m-clarificado  $\mathbb{K} = \langle G, M, I \rangle$ .

**Salida:** El conjunto  $\mathcal{B}$  de conceptos mixtos de  $\mathbb{K}$ .

```

1 inicio
2    $\mathcal{B} := \emptyset; Y := \emptyset; \Sigma := \emptyset;$ 
3   mientras  $Y < M$  hacer
4     para cada  $X \subseteq Y$  hacer
5        $A := (Y \setminus X) \cup \overline{X};$ 
6       si  $\text{Closed}(A, \Sigma)$  entonces
7          $C := A^{\downarrow};$ 
8          $D := C^{\uparrow};$ 
9         si  $A = D$  entonces añadir  $\langle C, D \rangle$  a  $\mathcal{B}$  en otro caso añadir  $A \rightarrow D \setminus A$  a  $\Sigma;$ 
10       $Y := \text{Siguiente}(Y)$  // Esta llamada al método Siguiente(Y)
        devuelve al sucesor de  $Y$  en el orden léxico clásico.
11   para cada  $g \in G$  hacer añadir  $\langle \{g\}, \{g\}^{\uparrow} \rangle$  a  $\mathcal{B};$ 
12   añadir  $\langle \emptyset, M \cup \overline{M} \rangle$  a  $\mathcal{B};$ 
13   devolver  $\mathcal{B}$ 

```

---

Nótese que, gracias al Teorema 6.2.3, los átomos en el retículo de conceptos mixtos son los que se obtienen como  $\langle \{g\}, \{g\}^{\uparrow} \rangle$  para cada  $g \in G$ . Evitamos, por tanto, tener que buscar, de entre todos los conjuntos consistentes completos, los que cumplen que el par  $\langle A^{\downarrow}, A \rangle$  es un concepto mixto (hay  $2^{|M|}$  candidatos para comprobar). Cuando el contexto es m-clarificado, existe una aplicación biyectiva entre los elementos de  $G$  y los átomos en el retículo de conceptos mixtos. La línea 11 en el Algoritmo 4 realiza estos cálculos.

Finalmente, en la línea 12, el algoritmo añade el elemento inferior del

retículo de conceptos mixtos, el cual es  $\langle \emptyset, M \cup \overline{M} \rangle$ .

### 6.2.2. Propuesta basada en el algoritmo de Norris

El algoritmo de Norris es un algoritmo incremental que solo usa el contexto para calcular los atributos que tiene cada objeto. Esto es, nunca calcula explícitamente los cierres usando el contexto. Para los conceptos mixtos es más fácil que en el caso clásico porque el conjunto de los átomos del retículo de conceptos mixtos (cuando el contexto formal es m-clarificado) es precisamente el siguiente conjunto:

$$\text{Atomos} = \{ \langle \{g\}, \{g\}^{\uparrow} \rangle \mid g \in G \} \quad (6.2)$$

Además, estos conceptos mixtos son exactamente los elementos  $\vee$ -irreducibles en el retículo de conceptos mixtos (Teorema 4.2.3).

Obsérvese que, en el caso clásico, incluso cuando el contexto formal es clarificado,  $\{g\}^{\uparrow\downarrow}$  no es necesariamente igual a  $\{g\}$  y, además, el concepto  $\langle \{g\}^{\uparrow\downarrow}, \{g\}^{\uparrow} \rangle$  no es necesariamente un átomo del retículo de conceptos. Así, por ejemplo, para el contexto formal descrito en la Tabla 5.1,  $\langle \{x_2\}^{\uparrow\downarrow}, \{x_2\}^{\uparrow} \rangle = \langle \{x_2, x_4\}, \{m_2, m_4\} \rangle$ , que no es un átomo.

El Algoritmo 5 es una versión del algoritmo de Norris que tiene en cuenta estas propiedades para calcular el retículo de conceptos mixto evitando cálculos extra. La función Novel se usa con este fin.

La principal ventaja de este método es que es incremental y, cuando se añade una nueva entrada al contexto, los conceptos mixtos previamente calculados no requieren ser recalculados de nuevo. Otro hecho interesante es que el ciclo de este algoritmo calcula subretículos que crecen hasta que se obtiene el retículo de conceptos mixto.

**Función Novel( $A, D, \text{Atomos}$ ): booleano**

**Entrada:**  $A$  un conjunto de objetos,  $D$  un conjunto de atributos positivos/negativos,  $\text{Atomos}$  un conjunto de átomos para el retículo de conceptos mixtos.

**Salida:** “verdadero” cuando  $A$  lleva a introducir un nuevo concepto mixto y “falso” en otro caso.

```

1 inicio
2   para cada  $\langle \{g\}, Y \rangle \in \text{Atomos}$  hacer
3     si  $g \notin A$  y  $D \subseteq Y$  entonces devolver falso;
4   devolver verdadero

```

**Algoritmo 5: Norris Mixto**

**Entrada:**  $\mathbb{K} = \langle G, M, I \rangle$  un contexto formal  $m$ -clarificado

**Salida:** El conjunto de conceptos mixtos  $\mathcal{B}$ .

```

1 inicio
2    $\mathcal{B} := \emptyset$ ;  $\text{Atomos} := \emptyset$ ;
3   para cada  $g \in G$  hacer
4      $Y := \{g\}^\uparrow$ ;  $\mathcal{B}_{nuevo} := \emptyset$ ;
5     para cada  $\langle A, B \rangle \in \mathcal{B}$  hacer
6       si  $B \subseteq Y$  entonces Reemplazar  $\langle A, B \rangle$  por  $\langle A \cup \{g\}, B \rangle$  en  $\mathcal{B}$ ;
7       si no, si  $\text{Novel}(A, B \cap Y, \text{Atomos})$  entonces Añadir  $\langle A \cup \{g\}, B \cap Y \rangle$  a  $\mathcal{B}_{nuevo}$ ;
8      $\mathcal{B} := \mathcal{B} \cup \mathcal{B}_{nuevo}$ ;
9     Añadir  $\langle \{g\}, Y \rangle$  a  $\mathcal{B}$  y a  $\text{Atomos}$ ;
10  Añadir  $\langle \emptyset, M \cup \overline{M} \rangle$  a  $\mathcal{B}$ ;
11  devolver  $\mathcal{B}$ 

```

**6.2.3. Propuesta basada en el algoritmo FCbO.**

En este caso, se va a extender el algoritmo Fast Close-by-One (FCbO) propuesto en [58] que es una mejora de la versión proporcionada en [57] y que, a su vez, es una extensión de algoritmo CbO [60]. Básicamente, el algoritmo CbO realiza un recorrido en profundidad del retículo de conceptos comenzando por el concepto superior y usa un test de canonicidad para evitar el recálculo innecesario de conceptos. En cada paso, a partir de un concepto calcula conceptos ‘hijos’, es decir, conceptos que están justo debajo

de él. Para evitar repeticiones, considera un orden total  $<$  en  $M$  y, dado un concepto  $\langle A, B \rangle$  y un nuevo atributo  $m \notin B$ , se calcula  $D = (B \cup \{m\})^{\downarrow\uparrow}$  y solamente añade el concepto  $\langle D^{\downarrow}, D \rangle$  cuando la siguiente condición se cumple:  $m' \in B$  si y solo si  $m' \in D$  para todo  $m' < m$ .

Para extender este criterio al marco generalizado se va a introducir la siguiente notación:

**Notación 6.2.4** Sea  $\mathbb{K} = \langle G, M, I \rangle$  un contexto formal finito y  $<$  una ordenación total de  $M$ .

- Para cada  $a \in M \cup \overline{M}$ , se define  $\overleftarrow{a}$ ,  $\overrightarrow{a}$  y  $a^+$  como sigue:

$$\overleftarrow{a} = \{b \in M \cup \overline{M} \mid \text{Pos}(b) < \text{Pos}(a)\}.$$

$$\overrightarrow{a} = \{b \in M \cup \overline{M} \mid \text{Pos}(a) < \text{Pos}(b) \text{ o } \text{Pos}(a) = \text{Pos}(b)\}.$$

$$a^+ = \begin{cases} \overline{a} & \text{si } a \in M, \\ \text{mín}\{b \in M \mid \overline{a} < b\} & \text{en otro caso.} \end{cases}$$

- Para un concepto mixto  $\langle A, B \rangle$  y  $a \in M \cup \overline{M}$  tal que  $a \notin B$ , se define

$$B \circ a = \left( (B \cup \{a\})^{\downarrow\uparrow} \setminus B \right) \cap \overleftarrow{a}.$$

Con esta notación, el test de canonicidad puede ser extendido al caso mixto de la siguiente forma: Dado un concepto mixto  $\langle A, B \rangle$  y un atributo  $a \in M \cup \overline{M}$  tal que  $a \notin B$ , el concepto mixto  $\langle (B \cup \{a\})^{\downarrow}, (B \cup \{a\})^{\downarrow\uparrow} \rangle$  se añade a la lista de conceptos mixtos solo en el caso de que  $B \circ a = \emptyset$ .

Obsérvese que este test evita incluir bastantes veces el mismo concepto mixto pero no evita calcular  $(B \cup \{a\})^{\downarrow\uparrow}$  en cada caso. La ventaja del algoritmo FCbO con respecto a CbO es que introduce un pre-test para reducir el número de esos cálculos basado en [58, Lema 1]. El siguiente teorema extiende dicho lema.

**Teorema 6.2.5** Sea  $\mathbb{K} = \langle G, M, I \rangle$  un contexto formal,  $B \subseteq M \cup \overline{M}$  y  $a \in (M \cup \overline{M}) \setminus B$ . Si existe  $B' \subseteq B$  tal que  $\emptyset \neq B' \circ a \not\subseteq B$  entonces  $B \circ a \neq \emptyset$ .

DEMOSTRACIÓN: Considérese  $B' \subseteq B$  tal que  $\emptyset \neq B' \circ a \not\subseteq B$ . Entonces debe existir  $b \in B' \circ a \subseteq (B' \cup \{a\})^{\downarrow\uparrow} \cap \overleftarrow{a}$  con  $b \notin B$ . Por tanto, como  $B' \subseteq B$ , se tiene que

$$b \in \left( (B' \cup \{a\})^{\downarrow\uparrow} \setminus B \right) \cap \overleftarrow{a} \subseteq \left( (B \cup \{a\})^{\downarrow\uparrow} \setminus B \right) \cap \overleftarrow{a} = B \circ a.$$

□

---

**Algoritmo 6: FCbO Mixto**


---

**Entrada:**  $\mathbb{K} = \langle G, M, I \rangle$  un contexto formal m-clarificado con  $M$  totalmente ordenado por  $<$ .

**Salida:** El conjunto de conceptos mixtos  $\mathcal{B}$ .

```

1 inicio
2   Sea  $m$  el primer elemento en  $M$ ;
3   para cada  $x \in M \cup \overline{M}$  hacer  $N_x := \emptyset$ ;
4    $\mathcal{B} := \{ \langle \emptyset, M \cup \overline{M} \rangle \} \cup \text{FastGeneratedMixedFrom}(\langle G, G^\uparrow \rangle, m, \{N_x \mid x \in M \cup \overline{M}\})$ ;
5   devolver  $\mathcal{B}$ 

```

---

Este método toma ventaja de este teorema, donde quiera que sea posible, tal como hace FCbO. Para ello sustituye el recorrido del retículo en profundidad por una combinación de un proceso recursivo (búsqueda en profundidad) con un ciclo controlado por una cola que recorre el retículo de forma horizontal.

El Algoritmo 6 utiliza los conjuntos  $N_a$ , inicializados como conjunto vacío en (línea 3), para almacenar los conjuntos  $B \circ a$  (cuando no son vacíos) y posibilitar así el uso del Teorema 6.2.5.

En la línea 7 de la función 'FastMixedGeneratedFrom', la condición considera la mejora proporcionada por el Teorema 6.2.5 y además evita calcular los conceptos mixtos desde un conjunto de atributos no consistentes porque ello necesariamente conduce a otro concepto mixto  $\langle \emptyset, M \cup \overline{M} \rangle$  que



---

**Función FastMixedGeneratedFrom( $\langle A, B \rangle, a, \{N_x \mid x \in M \cup \overline{M}\}$ )**


---

**Entrada:** Un concepto mixto  $\langle A, B \rangle$ ,  $a \in M \cup \overline{M}$ ,  $\{N_x \mid x \in M \cup \overline{M}\}$  una familia de conjuntos con atributos positivos/negativos

**Salida:** Un conjunto de conceptos mixtos.

```

1 inicio
2    $\mathcal{B} := \{\langle A, B \rangle\};$ 
3    $Cola := \emptyset;$ 
4    $Ultimo := a;$ 
5   para cada  $b \in \vec{a}$  en orden creciente hacer
6      $Nn_b := N_b;$ 
7     si  $b \notin B \cup \overline{B}$  y  $N_b \cap \overleftarrow{b} \subseteq B$  entonces
8       si  $\overline{b} = Ultimo$  entonces  $Abajo_b := G \setminus Abajo_b;$ 
9       en otro caso
10          $Abajo_b := \{b\}^\downarrow;$ 
11          $Ultimo := b;$ 
12          $C := A \cap Abajo_b;$ 
13          $D := C^\uparrow;$ 
14         si  $(D \setminus B) \cap \overleftarrow{b} = \emptyset$  entonces añadir  $(\langle C, D \rangle, b)$  a  $Cola$ ;
15         en otro caso  $Nn_b := D;$ 
16   para cada  $(\langle C, D \rangle, b)$  in  $Cola$  hacer
17     si  $b$  es el último elemento de  $M \cup \overline{M}$  entonces añadir  $\langle C, D \rangle$  a  $\mathcal{B};$ 
18     en otro caso  $\mathcal{B} := \mathcal{B} \cup \text{FastMixedGeneratedFrom}(\langle C, D \rangle, b^+, \{N_{n_x} \mid x \in M \cup \overline{M}\});$ 
19   devolver  $\mathcal{B}$ 

```

---

ha sido incluido al principio del algoritmo. Por otro lado, en la línea 14 se usa el test de canonicidad.

#### 6.2.4. Propuesta basada en el algoritmo InClose

Como último algoritmo de esta sección, se presenta la extensión del algoritmo InClose [5]. Al igual que el algoritmo anterior, InClose se basa en CbO. InClose recorre los atributos usando el orden lexicográfico. Calcula los cierres incrementalmente solo una vez por concepto. Ambos métodos usan funciones recursivas para generar todos los conceptos de forma ordenada.

El algoritmo empieza por el concepto inicial  $\langle G, \emptyset \rangle$  y llama al proceso *MixedInClose* por primera vez. Entonces, esto hace un recorrido del retículo desde el extremo superior al inferior en el cual cada concepto mixto se calcula solo una vez. Finalmente, el elemento inferior  $\langle \emptyset, M \cup \overline{M} \rangle$  ha de ser añadido.

---

**Algoritmo 7: InClose Mixto**


---

**Entrada:**  $\mathbb{K} = \langle G, M, I \rangle$  un contexto formal  $m$ -clarificado con  $M$  totalmente ordenado por  $<$ .

**Salida:** El conjunto de conceptos mixtos  $\mathcal{B}$ .

```

1 inicio
2   Sea  $a$  el primer atributo en  $M$  ;
3    $\mathcal{B} := \text{MixedInClose}(G, \emptyset, a)$ ;
4   devolver  $\mathcal{B} \cup \{\langle \emptyset, M \cup \overline{M} \rangle\}$ 

```

---

InClose, al igual que CbO, usa el siguiente test de canonicidad: Dado el concepto  $\langle A, B \rangle$  el siguiente concepto es canónico (nuevo) si entre sus objetos no hay ninguno que preceda al objeto actual.

---

**Función MixedInClose( $A, B, a$ )**


---

**Entrada:**  $A \subseteq G, B \subseteq M \cup \overline{M}$  y  $a \in M \cup \overline{M}$ .

**Salida:** Esta función añade nuevos conceptos mixtos.

```

1 inicio
2    $\mathcal{B} := \emptyset$ ;
3   para cada  $b \in \vec{a} \setminus \{\bar{a}\}$  en orden creciente hacer
4      $A' := \emptyset$ ;
5     para cada  $o \in A$  hacer
6       si  $o$  tiene el atributo  $b$  entonces  $A' := A' \cup \{o\}$ ;
7     si  $|A'| > 0$  entonces
8       si  $|A'| = |A|$  entonces  $\mathcal{B} := \mathcal{B} \cup \{b\}$ ;
9       si no, si IsCanonical( $B, A', b^-$ ) entonces
10         $B' := B \cup \{b\}$ ;
11         $\mathcal{B} := \mathcal{B} \cup \text{MixedInClose}(A', B', b^+)$ ;
12   devolver  $\mathcal{B} \cup \{A, B\}$ 

```

---

La principal ventaja de InClose es que, en el mismo recorrido, calcula

simultáneamente los objetos del concepto actual y los siguientes candidatos canónicos. En cada paso, añade ordenadamente un atributo  $b$  al concepto actual y calcula el correspondiente conjunto de atributos  $A' = A \cap \{b\}^\downarrow$  que puede satisfacer una de las siguientes condiciones:

- (C1)  $A'$  es un conjunto vacío.
- (C2)  $A'$  es un conjunto no vacío menor que  $A$ .
- (C3)  $A' = A$ .

Para calcular el cierre incrementalmente, la condición (C3) se usa como test cuando se añade un nuevo atributo al concepto actual para podar la recursión hasta que el concepto sea cerrado (línea 8 en la función `MixedInClose`). Si la condición (C1) se satisface, entonces el nuevo conjunto de atributos es vacío y no realiza llamadas recursivas con él (línea 7 en la función `MixedInClose`). Solo cuando se cumple la condición (C2) se usa el test de canonicidad para decidir si es necesario realizar un nueva llamada recursiva (líneas 7 y 9 en la función `MixedInClose`):  $A'$  es canónico si, para todo  $m \in \overleftarrow{b} \setminus B$ , existen  $i, j \in A'$  tales que  $\langle i, m \rangle \in I$  y  $\langle j, m \rangle \notin I$ .

Cuando la búsqueda en anchura finaliza, todos los conceptos que tienen a  $b$  como primer atributo son cerrados.

### 6.2.5. Comparación de los algoritmos para generar retículos de conceptos mixtos.

En esta sección se va a estudiar cómo se comportan los diferentes algoritmos presentados en la sección anterior. Para ello, hemos utilizado un conjunto de datos obtenido del repositorio de la University of California (UCI) [69]. Este repositorio es considerado un referente para comparar algoritmos de tratamiento de la información, especialmente en el área de “*Machine Learning*”. En concreto, hemos utilizado para la comparativa el

**Función IsCanonical( $B, A', b$ )****Entrada:**  $B \subseteq M \cup \overline{M}, A' \subseteq G$  y  $b \in M \cup \overline{M}$ .**Resultado:** "verdadero" si cumple el test; "falso" en otro caso.

```

1 inicio
2   para cada  $m \in \overline{b} \setminus B$  hacer
3     Pos:=falso; Neg:=falso;
4     para cada  $o \in A'$  hacer
5       si  $o$  satisface  $m$  entonces Pos:=verdadero en otro caso Neg:=verdadero;
6       si Pos y Neg entonces interrumpe el bucle;
7     si No(Pos y Neg) entonces devolver falso;
8   devolver verdadero;

```

conjunto de datos "*Congressional Voting Records Data Set*" que almacena los votos de cada miembro del congreso de los Estados Unidos. Contiene datos sobre el voto de cada uno de los 435 votantes (objetos) en 17 votaciones (atributos) binarias (a favor o en contra de la propuesta). El conjunto de datos ha sido m-clarificado para evitar los objetos y atributos repetidos.

Los algoritmos presentados en la sección anterior se han implementado en Java y se han ejecutado sobre el conjunto de datos en un Mac OS X Yosemite 64 bits - Intel Core i7 (2,93 GHz) con 8 GB de RAM.

Con el fin de, no solo comparar los tiempos de ejecución de los diferentes algoritmos, sino también analizar su evolución en función del tamaño de la entrada, se han seleccionado subconjuntos de datos de forma incremental, considerando inicialmente 5 atributos e incrementando el número de atributos hasta los 17 (el conjunto de datos completo). Esto ha afectado, además de al número de atributos, al número de objetos debido a la m-clarificación.

Además del número de objetos y del número de atributos, se ha medido también la densidad de los contextos (proporción del número de votos positivos respecto del total) y el número de conceptos resultantes.

Primero, se han comparado los 4 algoritmos (Next-Closure, Norris, FCbO e InClose) variando los atributos entre 5 y 12. El límite superior lo imponen

los dos primeros algoritmos debido a que sus ejecuciones con más atributos sobrepasaban los recursos del ordenador. Los tiempos de ejecución pueden ser comprobados en la Tabla 6.2 .

	Atributos	5	6	7	8	9	10	11	12
Datos	Objetos clarificados	24	40	59	80	97	124	160	193
	Densidad (%)	48,33	49,58	51,08	52,03	54,52	55,16	55	54,74
	Conceptos	175	438	963	2028	3403	6524	13490	23943
Tiempo (ms)	<i>NextClosure</i> Mixto	5	9	20	38	147	663	3132	11861
	<i>Norris</i> Mixto	7	28	71	129	219	766	2434	6733
	<i>FCbO</i> Mixto	2	4	5	12	11	20	54	121
	<i>InClose</i> Mixto	1	4	2	3	3	6	13	30

Tabla 6.2: Comparación cambiando atributos en el conjunto de datos “Congressional Voting Records” (tiempo en milisegundos)

En un segundo experimento, *FCbO* e *InClose* se han enfrentado con un mayor número de atributos para superar los límites de los otros dos algoritmos. Los resultados del nuevo experimento se muestran en la Tabla 6.3.

Como conclusión general, la versión para atributos mixtos del algoritmo *InClose* consigue el mejor resultado en el experimento mostrado. *FCbO* Mixto también consigue un gran resultado, particularmente cuando el conjunto de datos tiene un número pequeño de atributos.

El algoritmo *NextClosure* mixto, con la función *Closed*, tiene un mejor resultado que el algoritmo de *Norris* mixto, pero ambos tienen peores resultados comparados con *FCbO* e *InClose* mixtos que son considerablemente

	Atributos	13	14	15	16	17
Datos	Objetos clarificados	219	240	259	279	289
	Densidad (%)	54,02	53,8	54,38	53,89	56,01
	Conceptos	42799	70618	110733	189343	241924
Tiempo (ms)	<i>FCbO</i> Mixto	451	490	868	1758	2219
	<i>InClose</i> Mixto	111	117	196	305	495

Tabla 6.3: Comparación cambiando atributos en el conjunto de datos “Congressional Voting Records” (tiempo en milisegundos)

más rápidos.

### 6.3. La herramienta FuzAtAnalyzer

Con el propósito de servir de apoyo a la investigación de nuestro grupo, se diseñó una herramienta en lenguaje JAVA que denominamos FuzAtAnalyzer porque fue inicialmente concebida para trabajar con Tablas de Atributos Difusos [22]. En la versión original, centramos nuestra investigación en obtener y trabajar con sistemas implicacionales con algoritmos que aparecen en los artículos [21–23, 25, 26, 78]. Esta herramienta fue completada con la implementación de algoritmos clásicos de ACF.

Finalmente, se han incorporado a esta herramienta los algoritmos que hemos descrito, lo que nos permite trabajar de forma más sencilla y eficiente que la alternativa consistente en dicotomizar el contexto para usar las herramientas habituales.

Se han implementado también algunos métodos que no se incluyen en esta tesis pero que nos han servido de apoyo para obtener sus resultados.

Presentamos, a continuación, una breve descripción de la herramienta:

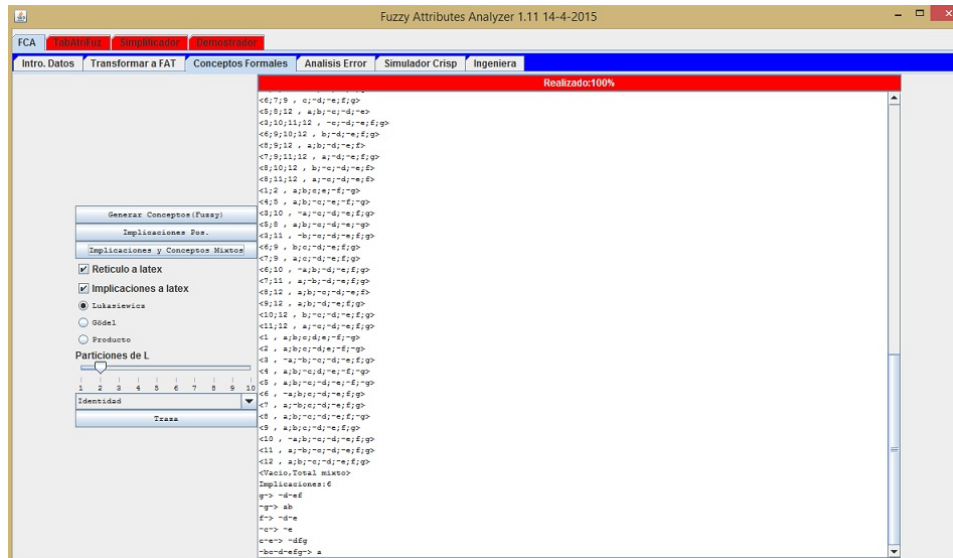


Figura 6.2: Captura de pantalla del FuzAtAnalyzer

- Pestaña 'Introducción de datos': Permite importar datos en el formato específico de la herramienta (botón 'Importar') o desde archivos Microsoft Excel usando Java Excel API [99] (botón 'Importar Excel') y guardar los datos (botón 'Grabar'). Permite también incluir (botón 'Nueva Tupla') o eliminar (botón 'Borrar Tupla') un nuevo objeto, intercambiar ceros y unos a una parte del contexto ('Invertir Valor'), etc.
- Pestaña 'Transformar a FAT (Fuzzy Attributes Table)' muestra los valores en la 'Pantalla de Datos' e incorpora un botón 'Transformar' para realizar ciertas modificaciones de datos difusos usando similitudes, un botón 'Enviar FAT' para trasladar los datos al motor que los gestiona, y permite seleccionar la t-norma deseada con 'Opciones Łukasiewicz-Gödel-Producto'.
- En la pantalla 'Conceptos Formales' tenemos los siguientes botones:

‘Generar Conceptos (Fuzzy)’, para calcular los conceptos difusos usando el método de [11], ‘Implicaciones Positivas’, para obtener la base Duquenne-Guigues clásica, ‘Conceptos Positivos’, para calcular los conceptos (positivos) usando el algoritmo InClose, ‘Implicaciones y Conceptos Mixtos’, que genera un sistema implicacional mixto completo y el conjunto de conceptos mixtos usando la extensión del algoritmo Next-Closure.

Permite seleccionar los métodos para la comparación de rendimiento en la generación de conceptos mixtos entre Next-Closure, UpperNeighbor, FCbO, InClose, Norris, InClose1 y FastInClose.

Además de todo ello, incorpora las siguientes opciones: ‘Retículo a  $\LaTeX$ ’, para que la salida del botón ‘Implicaciones y Conceptos Mixtos’ sea un archivo en formato  $\LaTeX$  donde se dibuja el retículo de conceptos (Ver Figura 6.4) usando el paquete *XYPic* [94] de  $\LaTeX$  (los retículos que aparecen en esta tesis han sido generados con esa opción de esta herramienta), ‘Implicaciones a  $\LaTeX$ ’, para que, al pulsar el botón ‘Implicaciones y Conceptos Mixtos’, se genere un archivo en formato  $\LaTeX$  donde se enumeran las diferentes implicaciones mixtas. Otros botones y opciones que incorpora son: ‘Particiones de  $L$ ’, ‘Identidad-Globalización’, ‘Traza’, ‘Simulador Conceptos Mixtos’ y ‘Simplificación Mixta’.

- Incorpora también una pestaña ‘Análisis Error’ con los siguientes botones y opciones: botón ‘Error’ (usa un método de detección de errores en contextos formales difusos), opciones ‘Łukasiewicz-Gödel-Producto’ (permite seleccionar una de las 3 t-normas para usar en la comprobación de errores), barra ‘Particiones de  $L$ ’ (permite establecer particiones en la generación de implicaciones que permitirán la detección de errores), opción ‘Identidad-Globalización’ (permite generar implicaciones usando la globalización definida por Belohlavek y



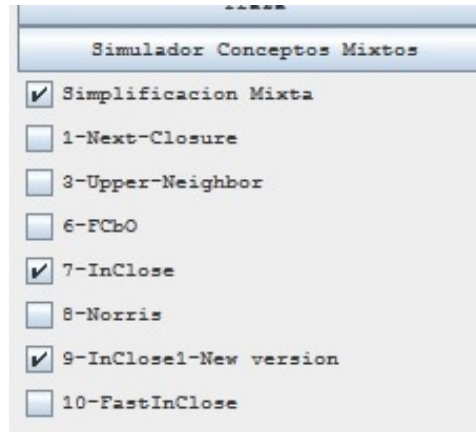


Figura 6.3: Detalle de los diferentes métodos para generar conceptos mixtos en FuzAtAnalyzer

Vichodyl en sus trabajos para usarla en la comprobación de errores), botón 'AR (Association Rules)' (permite extraer reglas de asociación desde el contexto con un soporte definido), botón 'Neg. AR' (permite extraer reglas de asociación mixtas desde el contexto con un soporte definido), 'Grado de soporte' (para indicar el soporte mínimo para el cálculo de las reglas de asociación), botón 'Base' (genera la base Duquenne-Guigues) y botón 'Base Neg.' (genera un sistema implicacional mixto). La 'Pantalla de datos' muestra los diferentes mensajes y resultados que se producen durante el desarrollo de los diferentes métodos - Figura 6.5.

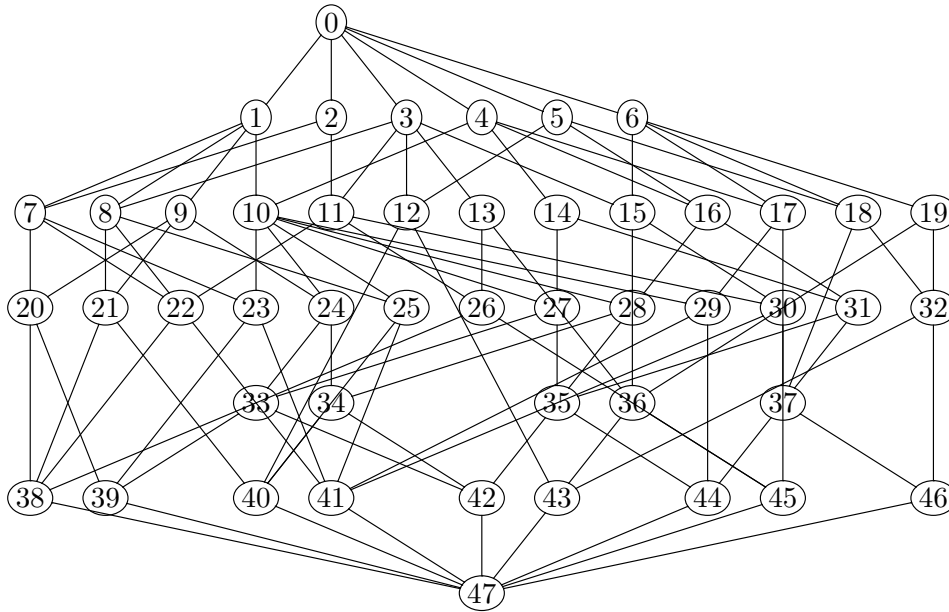


Figura 6.4: Retículo de conceptos mixto.

<p>Encontrados 32 subconjuntos de tamaño 5 con soporte 50.0%</p> <p>Creando subconjuntos de tamaño 6 derivados de 32 subconjuntos de tamaño 5</p> <p>Creados 35 subconjuntos de tamaño 6</p> <p>[_Cu_Fb_Si_Sn_KFe] (0.50)</p> <p>[_Cu_Fb_Si_Sn_NaFe] (0.50)</p> <p>[_Pb_Si_Sn_Na_KFe] (0.54)</p> <p>[_Cu_Fb_Cr_Sn_Na_K] (0.50)</p> <p>[_Cu_Fb_Si_Sn_Na_K] (0.54)</p> <p>[_Cu_Fb_Sn_Na_KFe] (0.50)</p> <p>[_Cu_Fb_Al_Sn_Na_K] (0.50)</p> <p>[_Cu_Fb_Si_Na_KFe] (0.50)</p> <p>[_Cu_Si_Sn_Na_KFe] (0.50)</p> <p>Encontrados 9 subconjuntos de tamaño 6 con soporte 50.0%</p> <p>Creando subconjuntos de tamaño 7 derivados de 9 subconjuntos de tamaño 6</p> <p>Creados 6 subconjuntos de tamaño 7</p> <p>[_Cu_Fb_Si_Sn_Na_KFe] (0.50)</p> <p>Encontrados 1 subconjuntos de tamaño 7 con soporte 50.0%</p> <p>Creando subconjuntos de tamaño 8 derivados de 1 subconjuntos de tamaño 7</p> <p>Creados 0 subconjuntos de tamaño 8</p> <p>Encontrados 205 subconjuntos con soporte 50.0%</p> <p>Subsets 205</p> <p>0.00/Cu ==&gt; 0.00/K: 0.9444444444444444</p> <p>0.00/K ==&gt; 0.00/Cu: 0.7391304347826086</p> <p>0.00/Cu ==&gt; 0.00/Sn: 1.0</p> <p>0.00/Sn ==&gt; 0.00/Cu: 0.9</p> <p>1.00/Fe ==&gt; 0.00/Fb: 1.0</p> <p>0.00/Fb ==&gt; 1.00/Fe: 0.65</p>	<p>Error</p> <p><input checked="" type="radio"/> Lukasiewics</p> <p><input type="radio"/> Gödel</p> <p><input type="radio"/> Producto</p> <p>Particiones de L</p> <p>1 2 3 4 5 6 7 8 9 10</p> <p>Identidad</p> <p>AR Base</p> <p>Neg. AR Base Neg.</p> <p>0.5</p>
---	---

Figura 6.5: Pestaña de los métodos de detección de errores en el FuzAtAnalyzer.



UNIVERSIDAD  
DE MÁLAGA

## Capítulo 7

# Aplicabilidad de los resultados obtenidos

A lo largo de los capítulos anteriores hemos cubierto una deficiencia detectada en la bibliografía relacionada con el manejo de información negativa en ACF.

Aunque, como explicamos en el Capítulo 3, se pueden encontrar trabajos que tratan la información negativa para obtener conocimiento en términos de conceptos formales, éstos se basan en adaptar el contexto, por ejemplo, dicotomizándolo, para poder hacer uso de los algoritmos clásicos. Por tanto, aumentan muy significativamente los costes de ejecución sin aprovechar las propiedades intrínsecas a la relación entre atributos positivos y negativos.

Lo mismo sucede en los trabajos de la bibliografía que abordan el descubrimiento y tratamiento de conocimiento en términos de implicaciones de atributos. Aunque habían estudiado la relación existente entre implicaciones positivas, implicaciones negativas e implicaciones mixtas, siguen haciendo uso de los axiomas de Armstrong que, al no ser un sistema axiomático completo, obliga a considerar implicaciones redundantes en los sistemas

implicacionales para poder caracterizar todas las implicaciones válidas en el contexto.

En el Capítulo 4, modificamos la conexión de Galois formada por el par de operadores de derivación para incluir la información negativa. Tras estudiar las propiedades de esta conexión de Galois, nos centramos en estudiar las propiedades de los retículos de conceptos mixtos. Encontramos una diferencia significativa con respecto a los retículos de conceptos clásicos. Mientras, en ellos, todo retículo finito es isomorfo a un retículo de conceptos, existen retículos finitos que no son isomorfos a ningún retículo de conceptos mixtos. El principal resultado de este capítulo es, precisamente, el Teorema 4.3.3 que caracteriza a los retículos que sí son isomorfos a un retículo mixto.

La extensión de los axiomas de Armstrong, para obtener un sistema axiomático completo, se ha abordado en el Capítulo 5. Hemos propuesto dos sistemas correctos y completos. Uno de ellos incluye los axiomas de Armstrong y, el otro, está basado en la Lógica de Simplificación. Obviamente, ambos sistemas axiomáticos son equivalentes. Dejamos, como trabajo futuro, la adaptación de los métodos de razonamiento automático a estas nuevas lógicas y el estudio de qué propiedades deben tener los sistemas implicacionales mixtos para que se pueda razonar sobre ellos de la forma más eficiente.

En el Capítulo 6 hemos presentado algoritmos, tanto para calcular sistemas implicacionales completos como retículos de conceptos mixtos, basados en los algoritmos más populares del caso clásico. Hemos intentado que los costes aumenten lo mínimo posible aprovechando la relación existente entre los atributos positivos y negativos. De todos los presentados, como cabía esperar, el más eficiente es el algoritmo InClose, al igual que en el caso clásico.

## 7.1. Aplicaciones

Concluimos la tesis con esta sección en la que mostramos algunas ideas de los distintos campos en los que hemos empezado a trabajar de cara a diseñar herramientas para aplicar los resultados de los capítulos anteriores.

### 7.1.1. Análisis de datos sobre cáncer de mama.

En [93] presentamos una aplicación de estos resultados al diagnóstico del cáncer de mama y, de forma general, el diagnóstico de enfermedades. Fiebre alta, náuseas, vómitos, dolor de cabeza, dolor de estómago, dolor sin localizar, etc. son síntomas que algunas enfermedades pueden tener en común o no, y que pueden ser representados como atributos. Estos síntomas permiten al personal médico distinguir y diagnosticar las posibles enfermedades. Las enfermedades comunes son fáciles de diagnosticar pero, por ejemplo, las enfermedades tropicales, son raras y difíciles de diagnosticar porque los síntomas son diferentes. Crear un árbol de decisión completo o un modelo matemático para diagnosticar es, a la vez, una tarea compleja y una herramienta útil para el personal médico.

Proponemos el uso de los retículos de conceptos, en este caso los mixtos, como una valiosa ayuda en dicha tarea. Sabiendo que un paciente tiene fiebre alta y náuseas, se apunta a un subretículo que contiene todas las enfermedades relacionadas con estos síntomas. Un subretículo compartiendo el atributo fiebre alta y ausencia de náuseas (negación del atributo náuseas) apunta a enfermedades diferentes. La presencia o ausencia de cada síntoma reduce el proceso de diagnóstico y la posibilidad de un error en el mismo.

Por otro lado, uno de los objetivos es reducir el número de síntomas a explorar en los análisis preliminares para el diagnóstico, priorizando los esfuerzos en aquellos síntomas que ayuden al personal médico a realizar un diagnóstico más rápido y certero. Hay que tener en cuenta que, para detectar

algunos síntomas, a veces, es necesario realizar pruebas muy costosas o pruebas que son invasivas o dolorosas para el paciente. Realizar más tests de los necesarios puede conllevar un retraso en el diagnóstico que puede tener consecuencias en la eficacia del tratamiento.

La modelización de problemas médicos requiere la selección de un conjunto de parámetros que han de ser considerados en el modelo matemático. La elección de estos parámetros tiene un impacto directo en la validez del modelo y sus resultados finales. En los estudios relacionados con la salud, estos parámetros se corresponden con una serie de indicadores reales seleccionados previamente por los investigadores. Por ejemplo, en muchas enfermedades, la fiebre es considerada un síntoma y, en estos casos, sabemos que la variable fiebre ha de ser introducida en nuestro modelo. Una vez que se ha tenido en cuenta como parámetro, cada valor particular influye en la solución que proporciona el modelo. Dos pasos han de ser considerados en este proceso: la selección de un parámetro (decisión binaria) y su medición (con la precisión necesaria).

Por tanto, una tarea previa al desarrollo de nuevos modelos es identificar aquellos datos que están fuertemente relacionados y que tienen una gran influencia sobre el resultado final. Es más, debido a los recursos extra en tiempo y económicos que producirían una gran cantidad de tests, podría ser útil identificar un número limitado de parámetros para introducirlos en el modelo, conservando su exactitud. Por otro lado, descubrir conexiones entre los síntomas (algunos síntomas pueden ser consecuencia de otros) constituye un gran desafío porque permitiría reducir el número de parámetros a considerar en el modelo matemático.

Como caso real en la ciencia de la medicina, hemos usado un conjunto de datos proporcionado por el University Medical Centre, Institute of Oncology, Ljubljana, Slovenia. Estos datos están accesibles en [69] y constituyen un completo conjunto de datos actual sobre información médica de pacientes

con cáncer de mama formado por 281 referencias y 41 atributos que han de ser reducidos a 277 referencias descartando aquellas instancias que tienen algún dato no registrado. Este conjunto de datos ha sido estudiado en varias publicaciones médicas y de minería de datos.

Atributo	Valores
Recurrencia	sí, no
Edad	20-29,30-39,40-49,50-59,60-69,70-79
Menopausia	lt40- <40 años, ge40- >40 años, premenopausia
Tamaño del tumor	0-4, 5-9, 10-14, 15-19, 20-24, 25-29, 30-34, 35-39, 40-44, 45-49, 50-54, 55-59
Nódulos linfáticos invadidos	0-2, 3-5, 6-8, 9-11, 12-14, 15-17, 18-20, 21-23, 24-26, 27-29, 30-32, 33-35, 36-39
Invasión de la cápsula de los nodos	si,no
Grado de malignidad	1, 2, 3
Mama	izquierda, derecha
Cuadrante de la mama	izq.-sup., izq.-inf., der.-sup., der.-inf., central
Irradiado	si,no

Tabla 7.1: Atributos del ejemplo de cáncer de mama.

En la Tabla 7.1 vienen referenciados los diferentes atributos estudiados en esta aplicación que pasamos a describir para mayor entendimiento de los mismos:



- **Recurrencia:** Reparición del cáncer después de 5 años.
- **Edad:** El factor de mayor riesgo para el cáncer de mama es la edad. Cuanto mayor sea la mujer, mayor riesgo de contraer cáncer (y presumiblemente recurrencia).
- **Menopausia:** El estado de premenopausia es una razón para creer que no habrá recurrencia.
- **Tamaño del tumor:** Cuanto mayor sea el diámetro del tumor, mayor posibilidad de recurrencia.
- **Nódulos linfáticos invadidos:** Debido a que los nódulos linfáticos axilares actúan como uno de los primeros sitios dañados para el cáncer de mama, ellos representan el sitio más común para una metástasis inicial. A mayor número de nódulos linfáticos invadidos, mayor probabilidad de recurrencia. Posiblemente sea el factor más influyente para la recurrencia.
- **Invasión de la cápsula de los nodos:** Si el cáncer hace metástasis en un nodo linfático, incluso fuera de la zona del tumor original, puede ser contenido por la cápsula del nódulo linfático. Sin embargo, el tumor puede reemplazar el nódulo linfático penetrando la cápsula, invadiendo los tejidos circundantes. Si hay invasión capsular, la recurrencia es más frecuente.
- **Grado de malignidad:** El grado histológico del tumor afecta a la recurrencia. Varía desde el tumor cuyas células mantienen muchas de sus características usuales hasta aquellos casos con alto índice de actividad mitótica en las células. Para grados 2 y 3, la recurrencia es más probable.
- **Mama:** Aunque el cáncer puede ocurrir en cualquier mama, no hay diferencia en las incidencias entre ellas. Existe una ligera (e inexplicada)

probabilidad de recurrencia superior en la izquierda.

- Cuadrante de la mama: La mama se divide en cuatro cuadrantes usando el pezón como punto central. A menudo ocurre en los cuadrantes superiores incrementando la posibilidad de recurrencia.
- Irradiado: La radioterapia reduce la recurrencia para este tipo de tumor.

‘Recurrencia’, ‘mama’ e ‘irradiado’ son almacenados en tres atributos atómicos, indicando su valor positivo o negativo. Las variables numéricas, como edad, tamaño del tumor o nódulos invadidos, se escalan por intervalos como es habitual en este tipo de problemas. Usando el algoritmo InClose mixto, descrito en el capítulo anterior, hemos calculado el retículo mixto del contexto y de varios subcontextos y, de su análisis, hemos sacado algunas conclusiones que resumimos a continuación.

En una primera etapa, se ha usado un conjunto de datos parcial eliminando tamaño del tumor y nódulo invadidos para centrar nuestra investigación sobre información fuertemente conectada con atributos etiquetados con valores positivos o negativos. Para medir la magnitud del estudio cabe considerar que este conjunto de datos tiene 193 objetos y 21 atributos, hallándose 229100 conceptos diferentes.

En una segunda etapa, se considera el tamaño del tumor, prescindiendo de los atributos de edad y nódulos invadidos. Usando el mismo método de exploración de subretículos se extrae el conocimiento de este subcontexto.

En una tercera etapa, se estudió el atributo de número de nódulos invadidos eliminando del conjunto de datos las referencias a los atributos de edad y tamaño del tumor. El retículo correspondiente se compone de 166299 conceptos mixtos.

Todo este conocimiento extraído viene detallado ampliamente en [93] por lo que no se va a reproducir aquí dicho contenido. El estudio permite

deducir, entre otras cosas, que algunos atributos no son relevantes para el diagnóstico del cáncer de mama: la localización de la mama (izquierda o derecha) o el cuadrante donde aparece el tumor.

Finalmente, hemos contrastado las conclusiones obtenidas con bibliografía especializada [30, 54]. Dentro de las conclusiones más importantes de nuestro estudio, parece existir una conclusión diferente en el hecho de que una paciente con edad por debajo de 30 años o por encima de 69 y menopausia lt40 podría no tener recurrencia, pero sí es cierto que este grupo tiene un alto riesgo (o probabilidad) para este caso. En España, la edad media para la menopausia es 50-52 años y pacientes sanas con lt50 tienen una alta probabilidad de tener un tumor a lo largo de su vida. En pacientes con edad en el rango 40-49, la menopausia determina el tratamiento, que es más agresivo en pacientes sin menopausia aún por su expectativa de vida. La radioterapia siempre reduce la recurrencia y se usa en la mayoría de pacientes jóvenes. Es un buen resultado que la recurrencia y la irradiación no estén relacionadas, señalando que la recurrencia se ha debido a otros factores y no a la ineficiencia del tratamiento con radioterapia. La presencia de nódulos con la cápsula invadida es frecuente en pacientes jóvenes y sus tumores tienen un alto nivel de malignidad. Los tumores de pequeño tamaño no han pasado mucho tiempo en la mama, por eso la presencia de nódulos con la cápsula invadida y la probabilidad de recurrencia es bastante baja.

En este caso de ejemplo real sobre cáncer de mama, los resultados obtenidos difieren parcialmente de aquellos obtenidos por otros métodos usados en la literatura: en ellos se usan técnicas más complejas y los resultados se expresan en términos de probabilidades. El retículo de conceptos que se propone proporciona relaciones específicas entre atributos que pueden ser tenidas en cuenta por investigadores en Oncología.

### 7.1.2. Optimización automatizada en el diseño de motores

Un motor es una máquina diseñada para convertir una forma determinada de energía en energía mecánica. En el diseño de un motor se evalúan diferentes tipos de propiedades/magnitudes como, por ejemplo, la velocidad, la potencia, la eficiencia o el nivel de sonido. Centrándonos en obtener el mejor resultado en una de estas propiedades podemos hacer que se obtenga un resultado pobre en otras propiedades, por ello los ingenieros han de tener en cuenta todas estas propiedades para alcanzar un equilibrio óptimo.

Por otro lado, el coste de fabricación de prototipos de un motor donde evaluar los resultados es muy alto, motivo por el cual los ingenieros han de usar simuladores para configurar los modelos. En el diseño de motores, estos simuladores son herramientas excelentes para ahorrar costes de producción porque en ellos los ingenieros pueden ajustar diferentes parámetros y observar el resultado global. Los motores más complejos tienen miles de parámetros que tienen que ser ajustados para obtener resultados óptimos en diferentes vías. El tamaño o la forma de la sección de un simple tubo puede incrementar o disminuir la cantidad de combustible que recibe el motor, por eso todas estas medidas son importantes en el desarrollo de cada motor. Los problemas a los que el diseñador se enfrenta son: ¿qué parámetros hay que ajustar? ¿cómo han de ser modificados esos parámetros?

Cada medida puede ser considerada como un atributo del motor y, además, la relación entre ellas puede determinar la eficiencia global. Todas estas medidas se recogen en un conjunto de datos que muestra cómo trabaja el motor, pero es difícil decidir específicamente cuál de ellas podemos cambiar y cómo tenemos que hacerlo porque no sabemos cómo afectan estos posibles cambios al rendimiento del motor.

Taguchi [98] argumenta que los ingenieros de desarrollo han de aplicar métodos de diseño para los parámetros y hacer que funciones básicas se aproximen a las funciones “ideales” bajo condiciones reales. Estas activida-

des de diseño pueden ser dirigidas por los departamentos de investigación y desarrollo antes de que el producto se cree finalmente. El trabajo de Taguchi se basa en métodos estadísticos que actualmente se han sustituido por procesos de redes neuronales o métodos multi-objetivo con matrices de covarianza usando estrategias de adaptación evolutiva [53,67].

En [92] presentamos un método, basado en la construcción del retículo de conceptos mixtos y en la navegación a través de él, como herramienta útil para decidir qué conjuntos de parámetros deben ser revisados de cara a mejorar el rendimiento. Aunque hay diferentes tipos de datos en el diseño de motores, éstos suelen ser valores numéricos. El método que se propone en [92] transforma estos datos en un contexto formal considerando intervalos de valores admisibles para cada magnitud.

Estos datos pueden reducirse, sin pérdida de información, a datos compatibles con el ACF donde una relación binaria determina si un objeto tiene una propiedad o atributo. En esta área se pueden usar diferentes métodos y propiedades que ya han sido desarrolladas y que no pueden usarse con los datos originales.

El método que se propone en [92] intenta centrarse en esas partes específicas del motor que pueden aumentar el rendimiento, cambiando estos valores para obtener el mejor resultado. Los investigadores pueden observar que es similar a un mapa autoorganizado de Kohonen [56], con la diferencia de que usa propiedades de los retículos de conceptos mixtos y podemos cambiar solamente un número reducido de atributos, disminuyendo el coste del procedimiento.

Partimos de un conjunto  $G$  de diferentes configuraciones experimentales del motor que los expertos consideran como datos iniciales. Los datos vienen dispuestos en una tabla  $T$  en la que las filas se corresponden con las diferentes configuraciones (objetos) y las columnas se corresponden con el conjunto  $M$  de los atributos que pueden ser medidos y que pueden tener repercusión

en la eficiencia del motor. Algunos de ellos pueden ser dependientes de otros como, por ejemplo, el número de revoluciones en el que se alcanza el par óptimo. Esta tabla  $T$  no es un contexto sino una tabla de valores numéricos. A la tabla se le añade un atributo nuevo, denominado **función objetivo**, que recoge el parámetro a optimizar. Este parámetro corresponde con una función que recibe como entrada otros parámetros del contexto, pero que habitualmente es desconocida.

Optimizar la función objetivo no siempre consiste en alcanzar el máximo o el mínimo, sino que puede ser acercarse todo lo posible a un valor óptimo fijado por los expertos o incluso estar dentro de un intervalo.

El método descrito en [92], en líneas generales, consiste en dividir la tabla de datos  $T$  en dos partes: una tabla  $T_1$  con las configuraciones en las que la función objetivo alcanza los  $k$  valores ( $1 < k < |G|$ ) más cercanos al valor óptimo y otra tabla  $T_2$  con el resto de las configuraciones.

Construimos un contexto formal  $\mathbb{K}$  a partir de la tabla  $T_2$  de la siguiente forma: para cada atributo, calculamos el valor máximo y el mínimo en la tabla  $T_1$  y marcamos en el contexto  $\mathbb{K}$  aquellos valores, en la tabla  $T_2$ , estén comprendidos entre el máximo y el mínimo mencionado. En resumen, en el contexto tenemos las mismas filas y columnas que en  $T_2$  y un par  $\langle c, m \rangle$  estará en  $I$  si y solo si el valor del atributo  $m$  en la configuración  $c$  (en la tabla  $T_2$ ) es mayor que el mínimo y menor que el máximo de  $m$  en la tabla  $T_1$ . En este contexto, los atributos negados indican que estos valores están fuera de los intervalos fijados (por  $T_1$ ) y apuntan a posibles candidatos para que sus valores sean reemplazados.

Una vez calculado el retículo de conceptos mixto con alguno de los algoritmos descritos en el capítulo anterior, consideramos el mayor concepto mixto y modificamos los valores de los atributos que aparecen negados añadiendo nuevas configuraciones. Si, con esto, no obtenemos mejores valores para la función objetivo, hemos de seguir el retículo de conceptos

mixto en orden descendente y escoger el siguiente concepto mixto para obtener nuevas configuraciones. Este proceso continúa hasta encontrar el valor más próximo al óptimo.

Este método puede ser incluido en simuladores como una herramienta de inteligencia artificial para ajustar parámetros automáticamente. Siendo una herramienta automática, los ingenieros no necesitan examinar ni calibrar el simulador tratando de detectar qué parámetro tienen que cambiar y qué valor se necesita. La configuración para el valor óptimo no se alcanza siempre debido a restricciones en los parámetros y a las configuraciones originales propuestas, pero se puede obtener un valor cercano.

Como otros métodos, detectar los puntos fijos locales es un problema que se puede resolver tomando un conjunto diferente de configuraciones iniciales como punto de partida.

### 7.1.3. Análisis de redes criminales

Las técnicas clásicas de ACF han sido aplicadas al análisis de las actividades criminales en diferentes áreas. La mayor parte de los trabajos han sido publicados por Elzinga y Poelmans, de los cuales se puede encontrar un resumen en [85].

- La radicalización y el terrorismo han sido investigados por el servicio de la Agencia de la Policía Nacional de Holanda [35] que desarrolló un modelo para clasificar yihadistas (potenciales). El objetivo de este modelo es detectar potenciales yihadistas para prevenir que él o ella entren en la fase peligrosa de la radicalización. Se usa Análisis de Conceptos Temporal para visualizar cómo un posible yihadista se radicaliza a través del tiempo.
- La violencia doméstica es uno de los problemas que es difícil de resolver debido a la falta de información sobre víctimas potenciales que no

denuncian su situación. En [87,88] los autores usan mapas emergentes autoorganizados con ACF que analizan los diferentes informes policiales para localizar víctimas potenciales. La minería de datos sobre el texto de estos informes policiales se usa para mostrar que existe un problema real en la forma de poner “etiquetas” a las situaciones, existiendo situaciones confusas, falta de datos, etc.

- El tráfico de personas y prostitución relacionada fueron estudiados en [86] tratando de descubrir en los informes policiales, usando minería de datos sobre el texto, estas situaciones para filtrar personas relevantes para posibles investigaciones más completas y usando ACF Temporal para crear un perfil visual de estas personas, su evolución a lo largo del tiempo y su entorno social. Para estos propósitos, el hecho de encontrar en los informes indicadores específicos diferentes (proxeneta, gran cantidad de dinero, coches caros, etc.) permite a los investigadores obtener un retículo donde los sospechosos y las víctimas pueden ser relacionados.
- Las áreas de gran intensidad de criminalidad, llamadas “puntos calientes”, indican dónde se han confeccionado un mayor número de informes policiales. Con la ayuda de herramientas para la geolocalización, esta investigación [39,55] permite la mejor distribución de recursos, como el reparto de patrullas, agentes y cámaras de seguridad así como la mejor definición de una estrategia para combatir el crimen y su prevención.
- Otra aplicación del ACF es el análisis de conversaciones de chats de pedófilos [36] para prevenir el abuso a menores y el empleo de violencia contra estos.

Las redes criminales son grupos organizados que trabajan en actividades ilegales y tienen una estructura extendida que es difícil de descubrir, por



ejemplo, como se ha mencionado anteriormente, los grupos relacionados con el tráfico de personas. Las relaciones entre sus diferentes miembros son difíciles de descubrir y, cuando aparecen individuos que no son miembros del grupo, las investigaciones crecen y, a veces, la policía pierde una gran cantidad de tiempo siguiendo una pista falsa.

Los informes policiales toman constancia de los diferentes hechos que ocurren durante su servicio. Estos informes contienen información importante y específica que consideramos como atributos. Los trabajos de Elzinga y Poelmans usan minería de datos sobre el texto con el fin de extraer información pero, en nuestra investigación, se han usado solo datos específicos fáciles de extraer de los informes. Estos atributos se corresponden con personas y vehículos. Cada persona se puede asociar con una única referencia a su documento de identificación y, cada vehículo, con su número de matrícula. Las localizaciones son información auxiliar que ha de ser tratada con precaución porque sus valores han de ser agrupados por zonas predeterminadas, es decir, debemos considerar que a veces en los informes aparece el nombre de un establecimiento y otras veces la calle donde se ubica, siendo éste un motivo por el cual en nuestro análisis inicial se va a descartar esta información.

Debido a diferentes razones, los agentes de la policía a menudo desconocen toda la información recogida en informes previos de otros agentes, por lo que la información ha de ser tratada por personal dedicado exclusivamente a monitorizar y analizar estos informes.

Si usamos un conjunto de datos donde aparecen, en una serie temporal, los conjuntos de atributos que suceden o se encuentran juntos, se puede generar un retículo de conceptos que representa la posible red criminal.

Usando los retículos con conceptos de atributos positivos, se puede detectar y comprobar una posible jerarquía entre los miembros, pero usando un retículo donde aparecen los conceptos con atributos mixtos, podemos

detectar posibles sospechosos que se pueden descartar hasta que se posea más información sobre los mismos, o que no están relacionados entre ellos.

Un sistema implicacional mixto completo aporta información que la base Duquenne-Guigues no proporciona. Esta información permite, de alguna forma, detectar *clusters* de grupos de delincuentes. Veamos un ejemplo para comprender esta idea intuitiva.

**Ejemplo 7.1.1** La tabla 7.2 recoge información de 9 informes de patrullas policiales que identifican a diferentes grupos de sospechosos que se han localizado juntos cerca de zonas específicas asociadas con actividades criminales. La Figura 7.1 muestra el retículo de conceptos correspondiente.

Informe	a	b	c	d	e	f	g	h	i	j	k
1	×			×					×		
2		×	×		×					×	
3	×							×	×		
4						×	×				×
5	×			×					×		×
6	×			×				×	×		
7						×	×				
8	×			×				×			
9							×				×

Tabla 7.2: Ejemplo de informes sobre datos criminales

Los conceptos con atributos (sospechosos) *a*, *g* y *k* se localizan en la parte superior del retículo, lo que significa que estos atributos aparecen en común en diversos objetos (informes) pero no siempre han de aparecer juntos. La base Duquenne-

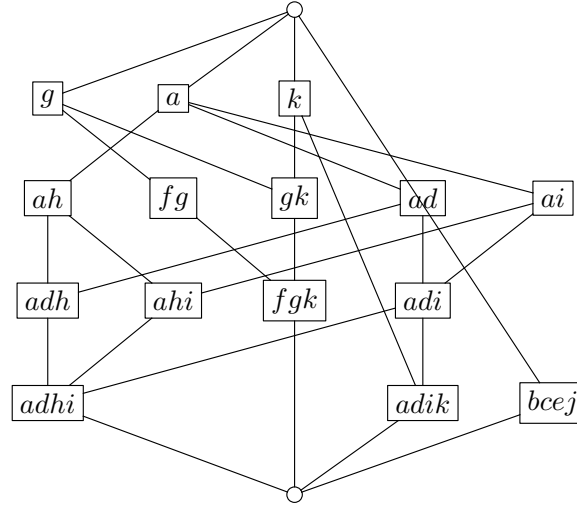


Figura 7.1: Retículo de conceptos asociado con la Tabla 7.2.

Guigues para este contexto es:

$$\Sigma_{DG} = \{j \rightarrow bce, i \rightarrow a, h \rightarrow a, f \rightarrow g, e \rightarrow bcj, d \rightarrow a, c \rightarrow bej, b \rightarrow cej, \\ bcej k \rightarrow adfghi, bcej g \rightarrow adfhik, ak \rightarrow di, ag \rightarrow bcdefhijk, \\ adhik \rightarrow bcejgj, abcej \rightarrow dfghik\}$$

En esta base no encontramos información que relacione los atributos  $a$ ,  $g$  y  $k$  por separado. Aplicando las técnicas descritas en el capítulo anterior obtenemos el siguiente sistema implicacional mixto (que es completo para este contexto):

$$\Sigma = \{k \rightarrow \overline{bceh\bar{j}}, \bar{j} \rightarrow \overline{bce}, j\bar{k} \rightarrow \overline{abcde\bar{f}\bar{g}\bar{h}\bar{i}}, i \rightarrow \overline{abc\bar{e}\bar{f}\bar{g}\bar{j}}, \bar{h}\bar{i} \rightarrow \overline{ad}, \\ \bar{g} \rightarrow \bar{f}, g \rightarrow \overline{abc\bar{d}\bar{e}\bar{h}\bar{i}\bar{j}}, f\bar{k} \rightarrow \bar{g}, \bar{f}\bar{g}\bar{i} \rightarrow \bar{k}, \bar{e} \rightarrow \overline{bc\bar{j}}, d \rightarrow \overline{abc\bar{e}\bar{f}\bar{g}\bar{j}}, \\ \bar{d}\bar{i} \rightarrow \bar{a}\bar{h}, \bar{d}\bar{h} \rightarrow \bar{a}\bar{i}, \bar{d}\bar{f}\bar{g} \rightarrow \bar{k}, \bar{c} \rightarrow \overline{bc\bar{j}}, \bar{b} \rightarrow \overline{ce\bar{j}}, \overline{bceh\bar{j}\bar{k}} \rightarrow \bar{a}\bar{f}\bar{g}, \\ \overline{bc\bar{e}\bar{f}\bar{g}\bar{j}} \rightarrow a, a \rightarrow \overline{bc\bar{e}\bar{f}\bar{g}\bar{j}}, \overline{ad\bar{f}\bar{g}\bar{h}\bar{i}\bar{k}} \rightarrow bcej, \overline{abc\bar{d}\bar{e}\bar{h}\bar{i}\bar{j}} \rightarrow g\}$$

Ponemos especial atención a las implicaciones  $a \rightarrow \overline{bc\bar{e}\bar{f}\bar{g}\bar{j}}$ ,  $g \rightarrow \overline{abc\bar{d}\bar{e}\bar{h}\bar{i}\bar{j}}$  y  $k \rightarrow \overline{bceh\bar{j}}$  que nos muestran que atributos no están relacionados con ellos. El

conjunto de atributos *bcej* no está relacionado con los atributos *a, g* y *k*; *f* y *g* no están relacionados con *a, d, h* e *i*. Se puede observar que hay tres conjuntos no relacionados *adhi*, *fg* y *bcej* y el atributo *k* que conecta los dos primeros conjuntos. Podemos adaptar el contexto para mostrar esta situación (ver Tabla 7.3). Si descartamos el objeto 2, podemos simplificar el retículo y tener una estructura más clara sobre la red criminal y la relación entre atributos (Figura 7.2).

Informe	a	d	h	i	k	f	g	b	c	e	j
1	×	×		×							
3	×		×	×							
5	×	×		×	×						
6	×	×	×	×							
8	×	×	×								
4					×	×	×				
7						×	×				
9					×		×				
2								×	×	×	×

Tabla 7.3: Ejemplo de informes sobre datos criminales con atributos reubicados

Cómo interpretar el retículo es muy importante para tener en cuenta los roles de los componentes de la red criminal. En el retículo no se puede deducir una estructura jerárquica para la red, pero los subconjuntos de atributos de los conceptos localizados en el nivel superior del retículo apuntan a miembros que son interesantes para ser investigados de forma específica.

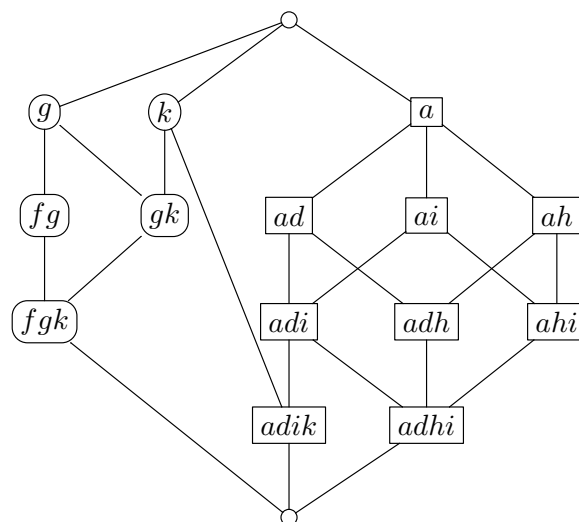


Figura 7.2: Retículo de conceptos con relaciones aclaradas.

#### 7.1.4. Sistemas de recomendación

Los sistemas de recomendación [91], también llamados sistemas de apoyo para la toma de decisiones, constituyen un método eficiente y con un uso creciente en la tecnología de la información. Permite establecer relaciones entre el usuario o demandante y el producto demandado para hacer que éste se adapte de una manera más eficiente a sus preferencias.

Estos sistemas se basan en técnicas de minería de datos para obtener información, analizarla y proporcionar el conocimiento deseado o, al menos, el que el sistema considera más adecuado para que el gestor pueda usarlo para tomar una decisión. Cuanta más información se proporcione al sistema, más datos podrá analizar y generará una respuesta mejor. Esto conlleva que el punto de partida sea una información conocida ya que el sistema no puede proporcionar conocimiento desde la nada y ha de ser iniciado por el gestor.

El software desarrollado en torno a los sistemas de recomendación es

diverso, abarcando un amplio rango de técnicas y aplicaciones para proporcionar la mejor opción para la decisión del usuario. Su evolución está siendo considerable en estos últimos años [1] aunque la adaptación de los diferentes sistemas a los múltiples casos de estudio y necesidades hace que aún quede mucha investigación pendiente para resolver cada uno de los problemas planteados.

Las diferentes aplicaciones basadas en la información que aporta el usuario se extienden por la red para servir de consejeros en la elección de música, viajes, libros, etc. Sin embargo, pueden englobar otras facetas menos públicas y accesibles como comercio electrónico o diagnosis médica entre otras áreas. Todas ellas hacen que las empresas se hayan volcado en dedicar parte de sus esfuerzos en investigar sus sistemas de recomendación internos para optimizar los beneficios o conllevar mejores decisiones para el buen desarrollo de la misma ofreciendo un mejor producto al cliente.

Las redes sociales son un ejemplo de sistemas de recomendación sencillos en los cuales se analizan las diferentes relaciones que tiene un usuario para mostrarle que quizás deberían conectar con un nuevo usuario. Lógicamente, este tipo de recomendaciones aumenta a medida que el usuario va formando su red ya que, ante la falta de información inicial, el sistema es incapaz de “adivinar” en qué o quién puede estar interesado el usuario [47, 109].

El uso de la información negativa abre un punto de vista diferente en los sistemas de recomendación pues hay casos en los que la información negativa no es tomada en cuenta en la toma de decisiones y ésta puede ser relevante en dichos casos.

El caso de las empresas dedicadas a buscar empleo a sus clientes cotejan el perfil de cada uno de los usuarios con los trabajos ofertados y las características que buscan las empresas. Dentro de la gran cantidad de atributos posibles, la consulta se basa en un grupo reducido de ellos, pudiendo optar únicamente a exigir que dentro del perfil se tenga un atributo, pero no que

se carezca de ellos. Por ejemplo, hay empresas informáticas que buscan en su perfil de programador a personas que sean matemáticos pero que no tengan conocimientos de programación previos para enseñarles su estilo propio y adaptarlos a su forma de desarrollar software evitando los posibles “vicios de programación”. Otro ejemplo es el caso de las búsquedas discriminatorias, como que el candidato al puesto de trabajo no esté casado o no fume. O que, por ejemplo, no tenga una titulación superior al perfil del puesto a desempeñar para evitar que el contrato sea de poca duración y no compense formar al trabajador en las labores específicas del puesto de trabajo.

Dentro de este ejemplo de las empresas dedicadas a la búsqueda de empleo, podríamos usar la información que ya tienen para obtener nuevos resultados y recomendaciones en un área de negocio diferente: la formación. Si un usuario es rechazado, o no elegido al no ser la primera opción para el sistema, repetidamente ante tipos parecidos de empleo, podríamos comparar el perfil de los usuarios aceptados con el del rechazado para ver qué atributos tienen en común y cuáles no. Posiblemente, el valor de los atributos del usuario tenga diferencias con el conjunto de atributos del concepto formal cuyos objetos sean los usuarios aceptados, aunque este conjunto de atributos no tiene por qué ser un conjunto consistente completo ya que puede haber ciertas diferencias entre los usuarios aceptados, pero marcará las diferencias entre el usuario rechazado y el grupo y podrá orientar al usuario rechazado para mejorar sus opciones de manera eficiente.

Una herramienta útil es plantear una distancia entre usuarios para saber la similitud entre los diferentes perfiles basada en la distancia de Levenshtein [68] tomando cada uno de los valores de los atributos de forma binaria. Como en el caso definido el número de atributos podríamos prefijarlo, usaríamos la distancia de Hamming [48]. La aplicación de estas distancias sobre objetos que tengan atributos positivos y negativos conllevaría aumentar las

mismas con respecto al caso de solo considerar la coincidencia de atributos positivos, pero mediría una distancia más “real”.

El uso del retículo de conceptos mixtos puede orientar de una manera diferente comparando el retículo del contexto completo con el del contexto formado solo por los aceptados, y estudiando qué atributos del usuario rechazado hay que cambiar para que ambos retículos se asemejen. Para ello habría que eliminar conceptos del primero progresivamente y el orden de eliminación es el que nos interesa.

La relación de orden entre los elementos del retículo permite saber cuáles son los conceptos minimales dentro de la clausura del concepto cuyo objeto es el usuario rechazado. Dentro de estos conceptos minimales habría que estudiar cómo influye en la estructura del retículo la eliminación de alguno de sus atributos negativos, quedándonos con aquel que redujera la estructura del retículo formado por los usuarios aceptados y el rechazado.

**Ejemplo 7.1.2** *Consideremos los datos de 4 usuarios que han sido elegidos por una empresa para el puesto de programador. El usuario rechazado (UR) es comparado con el resto de usuarios aceptados (UA) para analizar el motivo de su no elección y ofrecerle una orientación adecuada. Para ello se divide sus atributos en si sabe Inglés o no, qué lenguajes de programación domina y qué cursos, entre los valorados por la empresa contratante, ha realizado.*

*Para el caso indicado en la Tabla 7.4, vemos que el conjunto de atributos del concepto que contiene a todos los usuarios aceptados es el vacío al no haber atributos comunes ni positivos ni negativos, por lo cual vamos a dibujar el retículo de conceptos (Figura 7.3) para ver las relaciones entre los diferentes grupos de atributos. Con una línea continua se indican aquellas relaciones de orden que se encuentran en el retículo formado solo por los usuarios aceptados y con línea punteada aquellas que surgen al añadir el usuario rechazado.*

*Analizando las distancias de Hamming entre los usuarios aceptados y el rechazado, resultan las siguientes:*



Ref.	Inglés ( $In$ )	Programación			Cursos		
		C ( $C$ )	Java ( $Ja$ )	SQL ( $SQ$ )	A ( $CA$ )	B ( $CB$ )	C ( $CC$ )
UA1	×	×		×		×	×
UA2	×		×	×		×	
UA3		×	×	×	×		×
UA4	×		×			×	
UR		×				×	

Tabla 7.4: Contexto formal de los perfiles de los usuarios aceptados y rechazado.

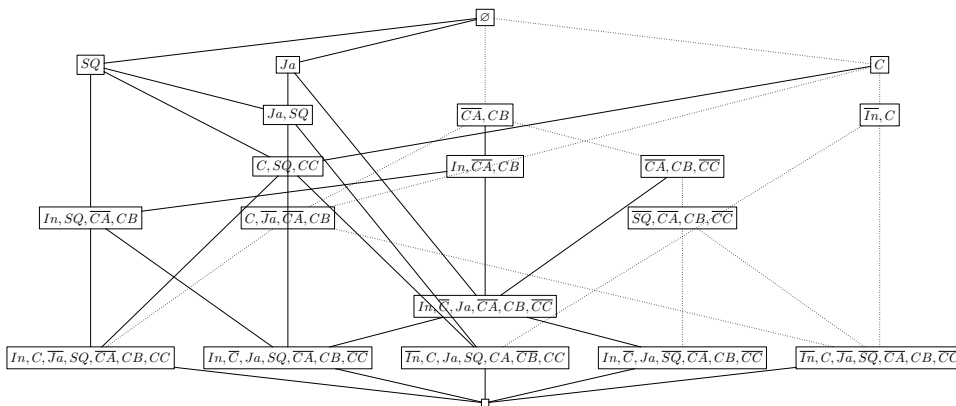


Figura 7.3: Retículo de conceptos mixto de los diferentes atributos de la Tabla 7.4.

- Distancia con UA1 es 3:  $In, SQ, CC$ .
- Distancia con UA2 es 4:  $In, \overline{C}, Ja, SQ$ .
- Distancia con UA3 es 5:  $Ja, SQ, CA, \overline{CB}, CC$ .
- Distancia con UA4 es 3:  $In, \overline{C}, Ja$ .

Reducir las distancias solo es posible aumentando el conocimiento del usuario rechazado. Es obvio que no es posible reducir la distancia aumentando o reduciendo

la formación de los usuarios aceptados con lo cual este punto de vista de la distancia no es controlable.

Los conceptos minimales dentro de la clausura del concepto cuyo objeto es el usuario rechazado son tres y por simplificación serán referenciados por sus atributos. Se quiere comprobar cómo varía la estructura del retículo añadiendo atributos al usuario rechazado para asimilarse al retículo de usuarios aceptados:

- Concepto  $\overline{In}$ ,  $C$ : Añadir el atributo positivo  $In$  al usuario rechazado haría que desapareciera este concepto del retículo además del concepto  $\overline{CA}$ ,  $CB$ . Otros conceptos añadirían ese atributo.
- Concepto  $C$ ,  $\overline{Ja}$ ,  $\overline{CA}$ ,  $CC$ : Añadir el atributo positivo  $Java$  al usuario rechazado solo haría que se modificaran los atributos de varios conceptos manteniéndose el mismo número de ellos. Lo mismo ocurriría al añadir el atributo positivo curso  $A$ .
- Concepto  $\overline{SQ}$ ,  $\overline{CA}$ ,  $CB$ ,  $\overline{CC}$ : Añadir el atributo  $SQL$  haría que aumentara el número de conceptos del retículo al añadirse uno, modificándose los atributos de varios de ellos. Añadir el atributo  $CC$  haría que se redujera el número de conceptos del retículo en uno al desaparecer el concepto  $\overline{CA}$ ,  $CB$ ,  $\overline{CC}$ .

Basándonos en la idea de que el atributo que habría que añadir al objeto  $UR$  es aquel que redujera la diferencia en número de conceptos del retículo formado por los cinco usuarios y el formado solo por los usuarios aceptados, la recomendación sería añadir el atributo  $In$ , que sería el que redujera el número de conceptos en dos.

La aplicación de este sistema de recomendación a la formación no sería el único caso ya que, aplicando el mismo sistema con respecto a otros usuarios que hayan conseguido otro tipo de trabajo, se le puede orientar laboralmente y formativamente hacia otro destino laboral apoyado en una formación orientada hacia dicho trabajo.



UNIVERSIDAD  
DE MÁLAGA

# Conclusiones y trabajo futuro

Presentamos, para finalizar, un resumen de los resultados obtenidos, de la discusión sobre cómo los objetivos marcados se han conseguido y cómo cubren un hueco existente en la bibliografía actual del área y de las líneas de trabajo futuro, a corto y medio plazo, que hemos ido desarrollando a lo largo de la tesis.

En la introducción y en el capítulo 3, tras una presentación exhaustiva de los antecedentes, tanto en minería de datos como en Análisis de Conceptos Formales, se han planteado los objetivos de la tesis y se han enmarcado dentro de la bibliografía existente y el estado actual del área de investigación. Básicamente, el objetivo era desarrollar un marco general que permita, de una forma uniforme y eficiente, extraer, tratar (razonar) y representar conocimiento a partir de información positiva y negativa. Consideramos que el objetivo se ha alcanzado plenamente.

A lo largo de la tesis, el adjetivo *mixto* se ha usado para distinguir las nuevas nociones de las clásicas.

En el capítulo 4, los fundamentos del Análisis de Conceptos Formales se han extendido para incluir la información negativa. Partiendo de la noción de conexión de Galois, se ha estudiado cómo la incorporación de atributos negativos repercute en la conexión de Galois generalizada, lo que nos ha llevado a un nuevo tipo de retículo que recoge toda la información, tanto positiva como negativa, denominado retículo de conceptos mixtos.

Como cabe esperar, al considerar también la información negativa, la cantidad de conocimiento que podemos extraer, y sobre la que podemos razonar, aumenta. La consideración de los atributos negativos en los conceptos formales hace que, generalmente, aumente el número de conceptos y, por tanto, el tamaño del retículo de conceptos. De hecho, el retículo con los conceptos clásicos se encuentra inserto en el retículo de conceptos mixto, es decir, se puede definir un función inyectiva entre el clásico y el mixto.

Las propiedades de este nuevo retículo, a diferencia de lo que ocurre en el caso clásico, nos han conducido a comprobar que no todos los retículos se corresponden con un contexto, es decir, hay retículos que no son isomorfos a ningún retículo de conceptos mixto. Hemos realizado un estudio detallado de las propiedades de los retículos de conceptos mixtos que nos ha permitido proporcionar un teorema de caracterización de tales retículos.

El capítulo 4 concluye, retomando el análisis de otros enfoques existentes en la bibliografía, con una comparación de los resultados obtenidos aquí y las otras propuestas. En particular, se muestra cómo los enfoques que consideran la negación de los conceptos, aunque aportan una riqueza extra al conocimiento extraído del contexto, sigue sin incorporar la información negativa del contexto. Por otro lado, se ha mostrado cómo estas negaciones de conceptos, negación débil y oposición débil, se trivializan en los retículos de conceptos mixtos. Finalmente, en el caso de los trabajos que consideran negación en los atributos, que sí consideran información negativa, quedan todos subsumidos en éste. La diferencia fundamental es que los demás trabajos intentan de alguna forma redefinir el problema para convertirlo en uno clásico, mientras que nosotros proponemos un marco global que considera la información negativa desde los fundamentos.

En el capítulo 5, se han estudiado las implicaciones mixtas entre atributos con un enfoque al estilo de la lógica. Se ha aumentado la expresividad del lenguaje formal utilizado lo que, como es bien conocido, aumenta la

complejidad del sistema y, al mismo tiempo, permite enriquecer la semántica. Ésta se ha definido en base a los resultados introducidos en el capítulo anterior. Sobre dicha semántica se han construido dos sistemas axiomáticos equivalentes y se ha demostrado su corrección y completitud. Estos sistemas axiomáticos explotan las ventajas inherentes a la semántica de la negación y abren las puertas al diseño de mecanismos de razonamiento automático, en especial el segundo presentado que se basa en el paradigma de simplificación. Esta tarea la hemos dejado propuesta como trabajo futuro a desarrollar a medio plazo.

La principal diferencia con los enfoques que encontramos en la bibliografía es que ellos se centran en un problema concreto: dado un contexto formal, encontrar, con el menor coste posible, un conjunto de implicaciones mixtas que permitan derivar todas las demás haciendo uso de los axiomas de Armstrong. Para ello, en algunos casos, proponen considerar por separado la información positiva y la negativa, para después completarlo con alguna técnica específica. En otros casos, la propuesta consiste en calcular las claves del contexto dicotomizado y, a partir de ella, obtener el conjunto de implicaciones mixtas. Es importante destacar que, si bien el sistema axiomático de Armstrong es correcto para la nueva semántica, no es completo.

La principal ventaja de nuestro enfoque es que, al proponer un sistema axiomático correcto y completo, podemos sacar el máximo partido sintácticamente a la semántica. Esto nos permite abordar una variedad mayor de problemas: podemos razonar sobre implicaciones en sí, independientemente de que representen la información contenida en un contexto particular; desarrollar una teoría de modelos; estudiar el problema de la implicación; el cálculo sintáctico de claves; abordar el estudio de la noción de base de implicaciones; etc. Estos trabajos también se proponen para su desarrollo a corto y medio plazo.

Por otro lado, en el problema específico que tratan otros autores (en-

contrar un conjunto de implicaciones completo para un contexto) el seguir considerando los axiomas de Armstrong obliga a compensar la no completitud con un aumento muy considerable en el tamaño de los conjuntos de implicaciones. Además, estos conjuntos contienen una inmensa cantidad de información que, semánticamente, es redundante. Con nuestro enfoque, toda esta información se puede eliminar y, por tanto, podemos trabajar con conjuntos más pequeños de implicaciones. Esto tiene una gran repercusión en los costes de los algoritmos, que ya de por sí han crecido considerablemente.

Aprovechando las propiedades de los nuevos sistemas axiomáticos, en el capítulo 6, hemos reconvertido los algoritmos más significativos de extracción de conocimiento existentes en la bibliografía con el fin de que consideren también la información negativa. En este apartado no hemos podido hacer ninguna comparativa con los enfoques de otros autores porque no hemos encontrado ningún trabajo en esta línea. Lo que sí hemos hecho es comparar, entre sí, los algoritmos nuevos y hemos llegado a la conclusión de que, como en el caso clásico, el que produce menores tiempos de ejecución es la versión mixta del conocido como InClose y que existe una diferencia significativa entre InClose y FCbO y el resto de los algoritmos.

Finalmente, en el capítulo 7, se han estudiado algunos problemas en los que consideramos que los resultados obtenidos en capítulos anteriores tienen mucho que aportar. En ellos hemos incluido herramientas para el diagnóstico del cáncer y, en general, de apoyo al diagnóstico de enfermedades; herramientas para reducción de costes en las fases iniciales del diseño de motores; herramientas de ayuda a la detección de redes criminales a partir de la base de datos de intervenciones policiales; y herramientas para mejorar los sistemas de recomendación. Éstas son tareas en las que hemos empezado a trabajar y que proponemos como trabajo a corto plazo.

Como trabajo futuro a medio plazo, a los casos ya mencionados en el

Capítulo 7, podemos añadir la extensión del trabajo desarrollado por Revenko y Kuznetsov en [90] para la detección de errores. Como nota anecdótica, pero que ayuda a entender que el disponer de más conocimiento puede permitir una representación más concisa de éste, destacamos que, para el contexto de los cuadriláteros usado como ejemplo en [90], la base implicacional mixta obtenida con Mixed-NextClosure tiene un menor número de implicaciones que la base Duquenne-Guigues (recuérdese que la base Duquenne-Guigues es la de menor tamaño de entre las clásicas)

Como trabajo futuro a largo plazo proponemos, como primer paso, la extensión de estos resultados a lógicas trivaluadas que permitan tratar atributos desconocidos. A partir de ahí, abordaremos la extensión a lógicas difusas, donde el problema de la negación resulta mucho más complejo, pero, al mismo tiempo, permite una mayor riqueza semántica. En especial, pretendemos centrarnos en la lógica de Atanasov que consideramos la más cercana a este problema.





UNIVERSIDAD  
DE MÁLAGA

# Índice de definiciones

- Átomo ..... Def. 2.1.11, p. 13
- Cierre sintáctico ..... Def. 2.3.27, p. 32
- Cierre sintáctico mixto ..... Def. 5.1.8, p. 109
- Concepto formal ..... Def. 2.3.4, p. 21
- Concepto formal mixto ..... Def. 4.1.6, p. 72
- Conexión de Galois ..... Def. 2.2.5, p. 18
- Confianza ..... Def. 2.4.1, p. 45
- Conjunto consistente ..... Def. 4.1.2, p. 71
- Conjunto consistente completo ..... Def. 4.1.3, p. 71
- Conjunto de implicaciones completo ..... Def. 2.3.30, p. 34
- Conjunto de implicaciones m-completo ..... Def. 5.1.18, p. 115
- Conjunto m-cerrado ..... Def. 5.1.6, p. 108
- Conjunto parcialmente ordenado ..... Def. 2.1.1, p. 8
- Contexto dual ..... Def. 3.2.3, p. 57

- Contexto formal ..... Def. 2.3.1, p. 19
- Contexto clarificado ..... Def. 2.3.11, p. 24
- Contexto clarificado mixto ..... Def. 6.2.1, p. 130
- Contexto dicotómico ..... Def. 3.2.1, p. 57
- Contexto reducido ..... Def. 2.3.12, p. 25
- Cota inferior ..... Def. 2.1.3, p. 9
- Cota superior ..... Def. 2.1.3, p. 9
- Elemento  $\vee$ -irreducible ..... Def. 2.1.10, p. 12
- Elemento  $\wedge$ -irreducible ..... Def. 2.1.10, p. 12
- Elementos opuestos ..... Def. 4.2.8, p. 83
- Extent ..... Def. 2.3.4, p. 21
- Homomorfismo entre retículos completos ..... Def. 2.1.18, p. 15
- Implicación ..... Def. 2.3.15, p. 26
- Implicación mixta ..... Def. 5.1.2, p. 104
- Ínfimo ..... Def. 2.1.6, p. 10
- Intent ..... Def. 2.3.4, p. 21
- Lenguaje de la Lógica ..... Def. 2.3.15, p. 26
- Lenguaje de MixAtL ..... Def. 5.1.1, p. 103
- Máximo de un conjunto ..... Def. 2.1.5, p. 10
- Mínimo de un conjunto ..... Def. 2.1.5, p. 10

- Modelo para la Lógica ..... Def. 2.3.16, p. 27
- Modelo mixto ..... Def. 5.1.2, p. 104
- Operador de clausura (o cierre) ..... Def. 2.2.1, p. 17
- Operadores de derivación ..... Def. 2.3.2, p. 21
- Operadores de derivación mixtos ..... Def. 4.1.1, p. 70
- Orden Léctico ..... Def. 2.3.35, p. 39
- Propiedad antisimétrica ..... Def. 2.1.1, p. 8
- Propiedad reflexiva ..... Def. 2.1.1, p. 8
- Propiedad transitiva ..... Def. 2.1.1, p. 8
- Pseudo-cerrado ..... Def. 2.3.31, p. 35
- Quasi-cerrado ..... Def. 2.3.31, p. 35
- Regla de asociación ..... Def. 2.4.1, p. 45
- Retículo algebraico ..... Def. 2.1.8, p. 11
- Retículo atomístico ..... Def. 4.2.2, p. 78
- Retículo booleano ..... Def. 2.1.21, p. 16
- Retículo complementado ..... Def. 2.1.19, p. 15
- Retículo completo ..... Def. 2.1.9, p. 12
- Retículo de conceptos ..... Def. 2.3.8, p. 22
- Retículo distributivo ..... Def. 2.1.20, p. 16
- Retículo ordenado ..... Def. 2.1.7, p. 11

- Retículo  $\wedge$ -complementado ..... Def. 4.2.15, p. 86
- Semiretículo ..... Def. 2.1.8, p. 11
- Sistema de clausura ..... Def. 2.2.4, p. 17
- Sistema implicacional ..... Def. 2.3.15, p. 26
- Sistema implicacional completo ..... Def. 5.1.18, p. 115
- Soporte ..... Def. 2.4.1, p. 45
- Subconcepto ..... Def. 2.3.1, p. 23
- Subconjunto denso ..... Def. 2.1.13, p. 13
- Subretículo ..... Def. 2.1.15, p. 14
- Superconcepto ..... Def. 2.3.1, p. 23
- Supremo ..... Def. 2.1.6, p. 10

# Bibliografía

- [1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734–749, June 2005.
- [2] C. C. Aggarwal and P. S. Yu. A new framework for itemset generation. In *Proc. ACM SIGACT.SIGMOD-SIGART, Symposium on Principles of Database Systems*, pages 18–24, 1998.
- [3] R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules in Large Databases. In *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB)*, pages 487–499, Santiago de Chile, Chile, 1994. Morgan Kaufmann Publishers Inc.
- [4] A. Amir, R. Feldman, and R. Kashi. A new and versatile method for association generation. *Information Systems*, 22(6/7):333–347, 1997.
- [5] S. Andrews. In-close, a fast algorithm for computing formal concepts. *International Conference on Conceptual Structures (ICCS), Moscow, CEUR-Workshop Proceedings*, 483:1–14, 2009.
- [6] S. Andrews. A 'best-of-breed' approach for designing a fast algorithm for computing fixpoints of Galois connections. *Information Sciences*, 295:633–649, 2015.
- [7] M.-L. Antonie and O. R. Zaiane. Mining positive and negative association rules: An approach for confined rules. In *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, pages 27–38, 2004.

- [8] M.-L. Antonie and O. Zaïane. An associative classifier based on positive and negative rules. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 64–69, 2004.
- [9] W. Armstrong. Dependency structures of data base relationships. *Proc. IFIP Congress. North Holland, Amsterdam*, pages 580–583, 1974.
- [10] P. Atzeni and V. D. Antonellis. Relational database theory. *The Benjamin/Cummings Publishing Company Inc.*, 1993.
- [11] R. Belohlavek. Algorithms for fuzzy concept lattices. In *Proc. Fourth Int. Conf. on Recent Advances in Soft Computing, RASC 2002, Nottingham, UK, December 12-13*, pages 200–205, 2002.
- [12] A. Berry, J. Bordat, and A. Sigayret. A local approach to concept generation. *Annals of Mathematics and Artificial Intelligence*, 49(1-4):117–136, 2007.
- [13] K. Bertet and B. Monjardet. The multiple facets of the canonical direct unit implicational basis. *Theoretical Computer Science*, 411(22-24):2155–2166, 2010.
- [14] G. Birkhoff. Lattice theory. In *Colloquium Publications*, volume 25. Amer. Math. Soc., 3. edition, 1967.
- [15] J. P. Bordat. Calcul pratique du treillis de Galois d’une correspondance. *Informatiques et Sciences Humaines*, 96:31–47, 1986.
- [16] J. F. Boulicaut and A. Bykowski. Frequent closures as a concise representation for binary data mining. In *Knowledge Discovery and Data Mining, Current Issues and New Applications, 4th Pacific-Asia Conference, PADKK 2000, Kyoto, Japan, April 18-20, 2000, Proceedings*, pages 62–73, 2000.
- [17] J. F. Boulicaut, A. Bykowski, and B. Jeudy. Towards the tractable discovery of association rules with negations. In *FQAS*, pages 425–434, 2000.
- [18] S. Brin, R. Motwani, and C. Silverstein. Beyond market baskets: Generalizing association rules to correlation. In *Proceedings of the 1997 ACM SIGMOD international conference on Management of Data (SIGMOD’97)*, pages 265–276. ACM Press, May 1997.
- [19] M. Chein. Algorithme de recherche des sous-matrices premières d’une matrice. *Bull. Math. Soc. Sc. Math. de Roumanie*, 1(13):21–25, 1969.

- [20] E. F. Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387, 1970.
- [21] P. Cordero, M. Enciso, A. Mora, I. P. de Guzmán, and J. M. Rodríguez-Jiménez. An efficient algorithm for reasoning about fuzzy functional dependencies. In *Advances in Computational Intelligence: 11th International Work-Conference on Artificial Neural Networks, IWANN 2011, Torremolinos-Málaga, Spain, June 8-10, 2011, Proceedings, Part II*, pages 412–420, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.
- [22] P. Cordero, M. Enciso, A. Mora, I. P. de Guzmán, and J. M. Rodríguez-Jiménez. Specification and inference of fuzzy attributes. In *2011 IEEE Symposium on Foundations of Computational Intelligence (FOCI)*, pages 107–114, 2011.
- [23] P. Cordero, M. Enciso, A. Mora, and M. Ojeda-Aciego. Computing minimal generators from implications: a logic-guided approach. In *Proceedings of the 9th International Conference on Concept Lattices and Their Applications, CLA 2012*, pages 187–198, 2012.
- [24] P. Cordero, M. Enciso, A. Mora, M. Ojeda-Aciego, and C. Rossi. Knowledge discovery in social networks by using a logic-based treatment of implications. *Knowledge-Based Systems*, 87:16 – 25, 2015.
- [25] P. Cordero, M. Enciso, A. Mora, and J. M. Rodríguez-Jiménez. Automated inference with fuzzy functional dependencies over graded data. In *Advances in Computational Intelligence - 12th International Work-Conference on Artificial Neural Networks, IWANN 2013*, pages 254–265, 2013.
- [26] P. Cordero, M. Enciso, A. Mora, and J. M. Rodríguez-Jiménez. Computing non-redundant sets of functional dependencies via simplification. In *IEEE Symposium on Foundations of Computational Intelligence, FOCI 2013, Singapore, Singapore, April 16-19, 2013*, pages 9–14, 2013.
- [27] P. Cordero, A. Mora, M. Enciso, G. Aguilera, and I. P. de Guzmán. Reasoning about fuzzy functional dependencies. *Actas del XIV Congreso Español sobre Tecnologías y Lógica Fuzzy*, pages 121–126, 2008.



- [28] C. Cornelis, P. Yan, X. Zhang, and G. Chen. Mining positive and negative association rules from large databases. In *Cybernetics and Intelligent Systems, 2006 IEEE Conference on*, pages 1–6, 2006.
- [29] B. A. Davey and H. A. Priestley. *Introduction to lattices and order*. Cambridge University Press, Cambridge, 1990.
- [30] R. Dávila and M. D. de las Peñas. *Radioterapia conformada en cáncer de mama. Definición de volúmenes y técnicas de tratamiento*. FEOR (Fundacion de la SEOR), 2010.
- [31] M. Delgado, M. D. Ruiz, D. Sánchez, and J. M. Serrano. A formal model for mining fuzzy rules using the RL representation theory. *Information Sciences*, 181(23):5194–5213, 2011.
- [32] J. Demetrovics and V. D. Thi. Describing candidate keys by hypergraphs. *Computers and Artificial Intelligence*, 18(2), 1999.
- [33] J. Diatta, D. R. Feno, and A. Totohasina. Galois lattices and bases for MGK-valid association rules. In *Proceedings of the 4th International Conference on Concept Lattices and Their Applications, CLA 2006*, pages 186–197, 2006.
- [34] X. Dong, Z. Niu, X. Shi, X. Zhang, and D. Zhu. Mining both positive and negative association rules from frequent and infrequent itemsets. In *Advanced Data Mining and Applications*, volume 4632 of *Lecture Notes in Computer Science*, pages 122–133. Springer Berlin Heidelberg, 2007.
- [35] P. Elzinga, J. Poelmans, S. Viaene, G. Dedene, and S. Morsing. Terrorist threat assessment with formal concept analysis. In *IEEE International Conference on Intelligence and Security Informatics, ISI 2010, Vancouver, BC, Canada, May 23-26, 2010, Proceedings*, pages 77–82, 2010.
- [36] P. Elzinga, K. E. Wolff, and J. Poelmans. Analyzing chat conversations of pedophiles with temporal relational semantic systems. In *2012 European Intelligence and Security Informatics Conference, EISIC 2012, Odense, Denmark, August 22-24, 2012*, pages 242–249, 2012.
- [37] M. Erné. Distributive laws for concept lattices. *Algebra Universalis*, 30:538–580, 1993.

- [38] R. Fagin. Functional dependencies in a relational database and propositional logic. *IBM Journal of research and development*, 21:534–544, 1977.
- [39] A. Farias, M. Cintra, A. F. Castro, and D. Lopes. Criminal hot spot detection using formal concept analysis and clustering algorithms. In *Encontro Nacional de Inteligência Computacional (ENIAC)*, volume 1, pages 1–6, 2014.
- [40] B. Ganter. Two basic algorithms in concept analysis. *Technische Hochschule, Darmstadt*, 1984.
- [41] B. Ganter and L. Kwuida. Which concept lattices are pseudocomplemented? *Lecture Notes in Computer Science*, 3403:408–416, 2005.
- [42] B. Ganter and L. Kwuida. Finite distributive concept algebras. *Order*, 23:235–248, 2006.
- [43] B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer-Verlag New York, Inc., 1st edition, 1997.
- [44] G. Gasmi, S. B. Yahia, E. M. Nguifo, and S. Bouker. Extraction of association rules based on literalsets. *Lecture Notes in Computer Science*, 4654:293–302, 2007.
- [45] G. Grätzer. *General Lattice Theory*. Birkhäuser Verlag, 1998.
- [46] J. Guigues and V. Duquenne. Familles minimales d implications informatives resultant d un tableau de donnees binaires. *Mathematiques et Sciences Sociales*, 95:5–18, 1986.
- [47] G. Guo. Resolving data sparsity and cold start in recommender systems. In *Proceedings of the 20th International Conference on User Modeling, Adaptation, and Personalization, UMAP'12*, pages 361–364, Berlin, Heidelberg, 2012. Springer-Verlag.
- [48] R. W. Hamming. Error Detecting and Error Correcting Codes. *Bell System Technical Journal*, 26(2):147–160, 1950.
- [49] T. Hamrouni, S. B. Yahia, and Y. Slimani. Prince: An algorithm for generating rule bases without closure computations. In *Data Warehousing and Knowledge Discovery, 7th International Conference, Copenhagen, Denmark, Proceedings*, pages 346–355, 2005.

- [50] J. Han and M. Beheshti. Discovering both positive and negative fuzzy association rules in large transaction databases. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 10(3):287–294, 2006.
- [51] J. Han and M. Beheshti. Mining fuzzy association rules: interestingness measure and algorithm. In *2006 IEEE International Conference on Granular Computing*, pages 659–662, 2006.
- [52] T. Ibaraki, A. Kogan, and K. Makino. Functional dependencies in horn theories. *Artificial Intelligence*, 108:1–30, 1999.
- [53] C. Igel, N. Hansen, and S. Roth. Covariance matrix adaptation for multi-objective optimization. *Evolutionary Computation*, 15(1):1–28, Mar. 2007.
- [54] E. A. J. M. Baena, M. D. de las Peñas and M. Pajares. Protocolo de diagnóstico y tratamiento del cáncer de mama precoz y localmente avanzado. *Oncoguía Hospital universitario Puerta del Mar*, 2012.
- [55] Q.-A. Kester. Visualization and analysis of geographical crime patterns using formal concept analysis. *CoRR*, abs/1307.8112, 2013.
- [56] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69, 1982.
- [57] P. Krajca, J. Outrata, and V. Vychodil. Parallel recursive algorithm for FCA. In *Proceedings of the 6th International Conference on Concept Lattices and Their Applications, CLA 2008*, volume 433, pages 71–82. CEUR-WS.org, 2008.
- [58] P. Krajca, J. Outrata, and V. Vychodil. Advances in algorithms based on CbO. In *Proceedings of the 7th International Conference on Concept Lattices and Their Applications, CLA 2010*, pages 325–337, 2010.
- [59] M. Kryszkiewicz. Non-derivable item set and non-derivable literal set representations of patterns admitting negation. *Data Warehousing and Knowledge Discovery: 11th International Conference*, pages 138–150, 2009.
- [60] S. O. Kuznetsov. A fast algorithm for computing all intersections of objects in a finite semi-lattice. *Automatic documentation and Mathematical linguistics*, 27(5):11–21, 1993.

- [61] S. O. Kuznetsov. Mathematical aspects of concept analysis. *Journal of Mathematical Sciences*, 80(2):1654–1698, 1996.
- [62] S. O. Kuznetsov. Machine learning and formal concept analysis. In *Concept Lattices, Second International Conference on Formal Concept Analysis, ICFA 2004*, pages 287–312, 2004.
- [63] S. O. Kuznetsov and S. A. Obiedkov. Comparing performance of algorithms for generating concept lattices. *Journal of Experimental and Theoretical Artificial Intelligence*, 14(2-3):189–216, 2002.
- [64] S. O. Kuznetsov and S. A. Obiedkov. Some decision and counting problems of the Duquenne-Guigues basis of implications. *Discrete Applied Mathematics*, 156(11):1994–2003, 2008.
- [65] L. Kwuida, A. Tepavcevic, and B. Seselja. Negation in contextual logic. *Lecture Notes in Computer Science*, 3127:227–241, 2004.
- [66] Laboratoire LARIM. Lattice miner. <https://lattice-miner.sourceforge.io/>, 2007.
- [67] H. Langouet, L. Metivier, D. Sinoquet, and Q. Tran. Optimization for engine calibration. In *International Conference on Engineering Optimization*, 2008.
- [68] V. I. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707, 1966.
- [69] M. Lichman. UCI machine learning repository. <http://archive.ics.uci.edu/ml>, 2013.
- [70] C. Lindig. Fast concept analysis. In *Working with Conceptual Structures – Contributions to ICCS 2000*, pages 152–161. Shaker Verlag, 2000.
- [71] D. Maier. *The theory of relational databases*, volume 11. Computer science press Rockville, MD, 1983.
- [72] Y. Malgrange. Recherche des sous-matrices premières d’une matrice à coefficients binaires. *Applications à certains problèmes de graphe, dans: Deuxième Congrès de l’AFCALTI*, pages 231–242, 1961.

- [73] H. Mannila and H. Toivonen. Multiple uses of frequent sets and condensed representations (extended abstract). In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 189–194, 1996.
- [74] H. Mannila, H. Toivonen, and A. I. Verkamo. Efficient algorithms for discovering association rules. In *AAAIWS'94 Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, pages 181–192, 1994.
- [75] R. Missaoui, L. Nourine, and Y. Renaud. Generating positive and negative exact rules using formal concept analysis: Problems and solutions. *Lecture Notes in Computer Science*, 4933:169–181, 2008.
- [76] R. Missaoui, L. Nourine, and Y. Renaud. An inference system for exhaustive generation of mixed and purely negative implications from purely positive ones. In *Proceedings of the 7th International Conference on Concept Lattices and Their Applications, CLA 2010*, pages 271–282, 2010.
- [77] R. Missaoui, L. Nourine, and Y. Renaud. Computing implications with negation from a formal context. *Fundamenta Informaticae*, 115(4):357–375, 2012.
- [78] A. Mora, M. Enciso, and P. Cordero. Closure via functional dependence simplification. *International Journal of Computer Mathematics*, 89:510–526, 2012.
- [79] A. Mora, M. Enciso, P. Cordero, and I. P. de Guzmán. An efficient pre-processing transformation for functional dependencies sets based on the substitution paradigm. *Lecture Notes in Computer Science*, 3040:136–146, 2004.
- [80] E. M. Norris. An algorithm for computing the maximal rectangles in a binary relation. *Revue Roumaine de Mathématiques Pures et Appliquées*, 23(2):243–250, 1978.
- [81] L. Nourine and O. Raynaud. A fast algorithm for building lattices. *Information Processing Letters*, 71:199–204, 1999.
- [82] J. Paredaens, P. D. Bra, M. Gyssens, and D. V. V. Gucht. *The structure of the relational database model*. Springer-Verlag, 1989.

- [83] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Efficient mining of association rules using closed itemset lattices. *Information Systems*, 24:25–46, 1999.
- [84] G. Piatetsky-Shapiro. Discovery, analysis and presentation of strong rules. In *Knowledge Discovery in Databases*, pages 229–248. AAAI Press, 1991.
- [85] J. Poelmans, P. Elzinga, and G. Dedene. Retrieval of criminal trajectories with an FCA-based approach. *Proceedings of the Formal Concept Analysis Meets Information Retrieval Workshop, Moscow*, 83, 2013.
- [86] J. Poelmans, P. Elzinga, G. Dedene, S. Viaene, and S. O. Kuznetsov. A concept discovery approach for fighting human trafficking and forced prostitution. In *Conceptual Structures for Discovering Knowledge - 19th International Conference on Conceptual Structures, ICCS 2011*, pages 201–214, 2011.
- [87] J. Poelmans, P. Elzinga, S. Viaene, and G. Dedene. A case of using formal concept analysis in combination with emergent self organizing maps for detecting domestic violence. In *Advances in Data Mining. Applications and Theoretical Aspects, 9th Industrial Conference, ICDM 2009*, pages 247–260, 2009.
- [88] J. Poelmans, P. Elzinga, S. Viaene, and G. Dedene. Curbing domestic violence: instantiating C-K theory with formal concept analysis and emergent self-organizing maps. *Int. Syst. in Accounting, Finance and Management*, 17(3-4):167–191, 2010.
- [89] N. S. Rai, S. Jain, and A. Jain. Mining interesting positive and negative association rule based on improved genetic algorithm. *International Journal of Advanced Computer Science and Applications(IJACSA)*, 5(1):160–165, 2014.
- [90] A. Revenko and S. Kuznetsov. Finding errors in new object intents. In *CLA*, pages 151–162, 2012.
- [91] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor. *Recommender Systems Handbook*. Springer-Verlag New York, 2010.
- [92] J. M. Rodríguez-Jiménez, P. Cordero, M. Enciso, and A. Mora. Optimization in engine design via formal concept analysis using negative attributes. In *The Fourth International Conference on Control, Modelling, Computing and Applications*, pages 159–172, Zurich (Suiza), 2016.

- [93] J. M. Rodríguez-Jiménez, P. Cordero, M. Enciso, and A. Mora. Data mining algorithms to compute mixed concepts with negative attributes: an application to breast cancer data analysis. *Mathematical Methods in the Applied Sciences*, 39:4829–4845, 2016.
- [94] K. Rose and R. Moore. XY-pic. <https://tug.org/applications/xy-pic/xy-pic.html>, 2010.
- [95] L. Sharma, O. Vyas, U. Tiwary, and R. Vyas. A novel approach of multilevel positive and negative association rule mining for spatial databases. *Lecture Notes in Computer Science*, 3587:620–629, 2005.
- [96] C. Silverstein, S. Brin, and R. Motwani. Beyond market baskets: Generalizing association rules to dependence rules. *Data Mining and Knowledge Discovery*, 2(1):39–68, 1998.
- [97] G. Stumme, R. Taouil, Y. Bastide, N. Pasquier, and L. Lakhal. Computing iceberg concept lattices with titanic. *Data and Knowledge Engineering*, 42(2):189–222, 2002.
- [98] G. Taguchi. Quality engineering (taguchi methods) for the development of electronic circuit technology. *Reliability, IEEE Transactions on*, 44(2):225–229, 1995.
- [99] TeamDev. Java excel api. <http://jexcelapi.sourceforge.net/>, 2012.
- [100] W.-G. Teng, M.-J. Hsieh, and M.-S. Chen. On the mining of substitution rules for statistically dependent items. In *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM)*, pages 442–449. IEEE Computer Society, 2002.
- [101] D. van der Merwe, S. A. Obiedkov, and D. G. Kourie. Addintent: A new incremental algorithm for constructing concept lattices. In *Concept Lattices, Second International Conference on Formal Concept Analysis, ICFCA 2004*, pages 372–385, 2004.
- [102] H. Wang, X. Zhang, and G. Chen. Mining a complete set of both positive and negative association rules from large databases. *Lecture Notes in Computer Science*, 5012:777–784, 2008.

- [103] R. Wille. Restructuring lattice theory: an approach based on hierarchies of concepts. In *Rival, I. (ed.): Ordered Sets*, pages 445–470. Boston, 1982.
- [104] R. Wille. Boolean concept logic. In *Conceptual Structures: Logical, Linguistic, and Computational Issues, 8th International Conference on Conceptual Structures, ICCS 2000*, pages 317–331, 2000.
- [105] X. Wu, C. Zhang, and S. Zhang. Efficient mining of both positive and negative association rules. *ACM Transactions on Information Systems (TOIS)*, 22(3):381–405, 2004.
- [106] P. Yan, G. Chen, C. Cornelis, M. De Cock, and E. E. Kerre. Mining positive and negative fuzzy association rules. *Lecture Notes in Computer Science*, 3213:270–276, 2004.
- [107] Y. Yevtushenko. Concept explorer. <http://conexp.sourceforge.net/>, 2000.
- [108] X. Yuan, B. P. Buckles, Z. Yuan, and J. Zhang. Mining negative association rules. In *ISCC*, pages 623–628. IEEE Computer Society, 2002.
- [109] M. Zhang, J. Tang, X. Zhang, and X. Xue. Addressing cold start in recommender systems: A semi-supervised co-training algorithm. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR '14*, pages 73–82, New York, NY, USA, 2014. ACM.