



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
E INTELIGENCIA ARTIFICIAL

Razonamiento mereotopológico automatizado para la depuración de ontologías

Memoria presentada por
Antonia M. Chávez González
para optar al grado de
Doctor en Matemáticas
por la Universidad de Sevilla

Antonia M. Chávez González

V. B. Director

D. Joaquín Borrego Díaz

Sevilla, 29 de Abril de 2005

A Alberto y Javier

Agradecimientos

En un principio, afronté la tarea de elaborar esta tesis valorando el esfuerzo que requeriría para mí, tanto en el aspecto intelectual como a nivel personal y familiar. Poco a poco comprendí que el esfuerzo no era sólo por mi parte: cada día más personas han compartido, alentado, sufrido si cabe, el desarrollo de este proyecto.

En primer lugar quiero agradecer a Joaquín Borrego su amistad y su inestimable ayuda, algo que algún día espero merecer y sin lo cual no podría haber llevado a cabo este trabajo.

A los miembros del Departamento de Ciencias de la Computación e Inteligencia Artificial, por su apoyo y consideración. A aquellos con los que he compartido docencia, por la carga que he podido suponer. A José Antonio y María José, siempre dispuestos a escucharme. Y en especial a Agustín, excelente compañero. Sin vosotros todo habría sido aún más difícil.

Por último, pero no menos, he de dar las gracias a mi familia. A mis padres, a quienes tanto debo, por su ejemplo imposible de imitar. A mis hijos, por esperarme cada tarde. Ellos dan sentido a mi esfuerzo de cada día. A Tomás, por enseñarme a mirar el futuro sin miedo. Gracias.

El trabajo desarrollado en esta memoria ha sido parcialmente financiado por el proyecto TIN2004-03884 *Sistemas Verificados para el Razonamiento en la Web Semántica* del Ministerio de Educación y Ciencia (proyecto cofinanciado por fondos FEDER).

Índice general

Introducción	11
Web Semántica y el papel de las Ontologías	11
OIL, DAML+OIL, OWL	12
Agentes con capacidad deductiva	15
Motivación de este trabajo	17
Contribución	18
Trabajos relacionados	19
Contenido de la memoria	22
Esquema general de la tesis	38
1. Preliminares	41
1.1. Retículos: Definición y resultados	41
1.2. Lógicas de la descripción	45
1.2.1. Sintaxis y Semántica de los lenguajes de conceptos	46
1.2.2. Bases de conocimiento	48
1.2.3. Tareas de razonamiento	50
1.2.4. Normalización	51
1.2.5. Relación de OWL con DL	51
1.3. Formalización en LPO de las Bases de Datos Lógicas	55
1.4. Argumentos	57
1.4.1. Definiciones y resultados fundamentales	58
1.4.2. La estructura argumentativa \mathbb{A}	64
1.5. OTTER y MACE4	67
1.6. Espacios Topológicos	69

1.7.	Satisfacción de restricciones	71
1.8.	Cálculo de Conexión de Regiones (RCC)	72
1.8.1.	Modelos de RCC	74
1.8.2.	PSRs en RCC	76
1.9.	Topología de los Cells	80
2.	Razonamiento automático en RCC	83
2.1.	RCC-8 y RCC-5	85
2.2.	El Retículo de las relaciones en RCC	88
2.3.	Movimientos topológicos y proyecciones reticulares	94
2.4.	Conexión de la proyección reticular	96
2.4.1.	Entornos conceptuales	96
2.4.2.	Razonamiento basado en entornos conceptuales	98
3.	Extensiones. Interpretación de RCC	103
3.1.	Introduciendo una relación de indefinición disjunta con RCC8 .	103
3.2.	Interpretación de la relación de indefinición	114
3.3.	Tabla de composición de una extensión interpretable	123
3.4.	Interpretación de las relaciones en la teoría “huevo-yema” . . .	125
3.4.1.	Interpretación con pulsación en el paradigma huevo/yema	126
3.4.2.	Interpretación de la contracción	127
3.5.	Interpretación con conjuntos del tipo <i>Rough</i>	131
4.	Extensiones con refinamiento	135
4.1.	Introduciendo una nueva relación de conexión	135
4.2.	Interpretación de las nuevas relaciones	143
4.3.	Tabla de composición de un subcálculo	148
5.	Razonamiento metaontológico con RCC	151
5.1.	Interpretación fuerte	151
5.2.	Interpretación débil	155
5.3.	Relación entre las interpretaciones fuerte y débil	159
5.4.	Interpretaciones vagas de los conceptos con respecto a una ontología	159

6. Anomalías en la conceptualización asociada a una ontología	161
6.1. Anomalías en ontologías provisionales	163
6.1.1. Revisión ontológica	164
6.1.2. Actualización	164
6.1.3. Incompletitud y anomalías	165
6.2. Clasificación de Anomalías en BC consistentes	165
6.3. Anomalías de tipo léxico-lógico	169
6.3.1. Sinonimia lógica	169
6.3.2. Polisemia lógica	170
6.3.3. Monosemia lógica	170
6.4. Anomalías por inconsistencia de tipo léxico-lógico	170
6.4.1. Antonimia lógica	170
6.4.2. Anomalías detectables por composición de relaciones me- reotopológicas o de proyecciones reticulares	171
6.5. Anomalías detectadas por el razonador	172
7. Resolución algorítmica de las anomalías	177
7.1. Resolución local: refinamiento por pares de conceptos	177
7.2. Ciclo de limpieza para ontologías	183
7.2.1. Ciclo de limpieza	183
7.2.2. Paso 1: Traducción de la TBox	185
7.2.3. Paso 2: Arreglos espaciales	187
7.2.4. Paso 3: De las relaciones espaciales a la nueva BC	188
7.3. Interpretación geométrica	188
7.3.1. Paso 4: Necesidad de interpretar lo obtenido	190
8. Limpieza de anomalías dirigida por un SRA	193
8.1. Ciclo de limpieza asistido por un demostrador automático de teoremas	194
8.2. Bases de conocimiento en RCC	195
8.3. Interpretación espacial de los argumentos	197
8.3.1. Condiciones sintácticas para la consistencia	197
8.3.2. Grado de credibilidad e interpretación espacial	200

8.3.3. Nivel de compacidad	201
8.4. Un caso de estudio	201
8.4.1. Fichas de experimentos	219
8.4.2. Conclusiones	227
9. Conclusiones y trabajo futuro	229
9.1. Conclusiones	229
9.2. Trabajo futuro	231
Bibliografía	234

Introducción

Web Semántica y el papel de las Ontologías

La idea de **Web Semántica** [8] surge a raíz de la propuesta de una nueva World Wide Web (WWW) que realiza Tim Berners-Lee, creador de la WWW y propulsor de esta iniciativa que propaga el Consorcio Internacional de Estandarización para la Web (W3C). Consiste, a grandes rasgos, en una extensión de la actual Web en la que la información aparece dispuesta de manera inteligente y ordenada, de forma que puede ser procesada por máquinas. Así, la Web Semántica (WS) puede entenderse como una base de datos cuyo contenido se encuentra global y convenientemente enlazado, en la que el *significado* de la información almacenada adquiere mayor relevancia. Ello nos permite no sólo *conocer* el contenido, sino *deducir* nuevo conocimiento e *incorporar* nueva información obtenida a partir de la ya existente. Este escenario representa una mejora sustancial y prometedora de la actual situación.

La información que en la WWW permanece oculta (porque no es explícita o no aparece reflejada durante la navegación por la WWW) resulta inaccesible, inmanejable a gran escala, debido al carácter de universalidad de las conexiones que proporcionan los enlaces de hipertexto. Sólo el usuario puede, en última instancia, seleccionar lo que es *de su interés*. La información, la representación de la misma, está orientada principalmente al consumo humano.

La Web Semántica, construida a partir de la actual Web, proporcionaría la capacidad de cooperación entre máquinas y seres humanos, de manera que procesarán y comprenderán (en el sentido de manipular de manera efectiva y eficiente) la información que, hoy por hoy, únicamente se *muestra* en la Web.

Que la WS sea factible depende en gran medida de que la teoría de representación del conocimiento resuelva varios problemas. La información debe estar representada de manera estructurada y debemos disponer de conjuntos de reglas que permitan y dirijan el razonamiento automatizado (para inferir nueva información, responder preguntas, etc). Se debe disponer, por tanto, de un lenguaje para expresar los datos y las reglas de razonamiento sobre los mismos, que permita exportar a la Web sistemas de representación. Con ese objeti-

vo se han desarrollado **XML** (eXtensible Markup Language) y **RDF** (Resource Description Framework). El primero permite dar estructura a un documento pero no expresa lo que significa tal estructura. El meta-lenguaje XML permite crear nuestro propio lenguaje de etiquetado. El significado lo expresamos con RDF mediante 3-tuplas que pueden estar escritas en XML. Cada 3-tupla representará el *sujeto*, *verbo* y *objeto* de una *frase elemental*, expresando que un determinado elemento posee una propiedad con cierto valor. Por ejemplo, *Juan tiene como DNI el número XXX*. La utilización de RDFS (RDF Schema) asegurará que el objeto es un número de DNI, no la matrícula de un vehículo. Sujeto, verbo y objeto se identifican mediante **URIs** (Universal Resource Identifier), como si fuesen enlaces en una página Web. Todo ello permitirá que la Web pase de ser un espacio hiperdocumental a ser un espacio *hiperconceptual*.

El hecho de utilizar un URI para codificar un concepto en un documento, hace que dicho concepto no sea sólo una palabra en el documento, sino que su definición sea única y resulte accesible a cualquier usuario en la Web. Pero esto no impide que dos bases de datos utilicen diferentes URIs para un mismo concepto. Aquí es donde entran en juego las **ontologías**. En Inteligencia Artificial (IA) se adopta el término *ontología* para referirse a un documento o fichero que contiene las definiciones formales de los conceptos y relaciones entre éstos en un determinado dominio de conocimiento.

Una ontología consta básicamente de un conjunto de clases (una taxonomía relacional de conceptos) y un conjunto de axiomas que permite deducir nuevo conocimiento. Aunque no exista una definición universalmente aceptada -una aproximación podría ser *especificación formal de los términos del dominio y las relaciones entre éstos*-, sí existe en cambio un consenso casi general en algunas de las razones por las que es necesario crear una ontología [55]:

- Compartir el conocimiento común de la estructura de la información entre agentes (incluidos humanos).
- Poder reusar el conocimiento del dominio.
- Poder explicitar lo que se asume acerca del dominio.
- Separar el conocimiento del dominio del conocimiento operacional.
- Analizar el dominio del conocimiento.

OIL, DAML+OIL, OWL

La Web posee una naturaleza descentralizada y abierta (dado que los recursos están disponibles mediante mecanismos de acceso como HTTP), lo que la hace desmesuradamente extensa y dinámica. Por ello, la implementación

de un lenguaje estándar de definición de ontologías y el mantenimiento de la consistencia de dichas ontologías, constituyen todo un reto.

El primer lenguaje de representación de ontologías bajo estándares W3C fue **OIL** (Ontology Inference Layer) [38]. Este lenguaje está definido como una extensión de RDFS y admite sintaxis XML. Está estructurado en capas, partiendo de RDFS (excepto *reificación* - posibilidad de tratar un aserto como parte de otro-) y añadiendo cada capa más funcionalidad (y complejidad) a la capa anterior. Entre sus limitaciones, que aparecen detalladas y justificadas en [38], destacamos la escasa expresividad en la declaración de reglas y la dificultad de identificar un conjunto común de axiomas que se puedan estandarizar.

Para extender el nivel de expresividad de OIL, DARPA (US Defense Advanced Research Projects Agency) desarrolló **DAML**¹ (DARPA's Agent Markup Language) y, en cooperación con el grupo de trabajo de OIL, surgió **DAML+OIL**, un nuevo paso en la construcción de un lenguaje estándar para la definición de ontologías. DAML+OIL hereda muchas características de OIL, pero comienza a potenciar un nuevo aspecto lógico, la utilización de las **Lógicas de la Descripción** (o Descriptivas, DL). Este lenguaje presenta como contrapartida cierta dificultad para el diseñador de ontologías, a nivel de uso y aprendizaje (según [37]), ya que es reducido el número de ontologías construidas con él. No es un problema de complejidad técnica, dado el gran número de herramientas desarrolladas en soporte para este lenguaje, sino de complejidad conceptual.

Para superar este problema, en julio de 2002 se publica un primer borrador de **OWL**² (Web Ontology Language), derivado de DAML+OIL pero cimentado en RDFS. OWL, al igual que OIL, está estructurado en capas que facilitan la adaptación a las necesidades de nivel de expresividad de cada comunidad de usuarios (autores, programadores) y de cada aplicación (agentes, motores de búsqueda, etc.), véase [35].

Para compaginar esas necesidades de expresividad y eficiencia, el grupo de trabajo de ontologías web de W3C define OWL como tres sublenguajes diferentes, cada uno de los cuales solventa un cierto equilibrio entre expresividad y eficiencia. De mayor a menor potencia expresiva, el mayor, *OWL Full*, contiene todas las primitivas de OWL y es totalmente compatible con RDF y RDFS, sintáctica y semánticamente. Este sublenguaje es tan potente como indecidible. Para conseguir mayor eficiencia computacional, *OWL DL* restringe el uso de los constructores de OWL y RDF, perdiendo compatibilidad con este último. Finalmente, una versión más "ligera" la proporciona *OWL Lite*. Éste restringe el uso de DL a un conjunto de constructores, siendo el menos expresivo pero de fácil uso e implementación.

¹<http://www.w3.org/daml>

²<http://www.w3.org/2004/OWL>

Las Lógicas de descripción [4] son una familia de lenguajes de representación del conocimiento que tienen como predecesores la Redes Semánticas y Sistemas de Marcos, pero, a diferencia de ellos, posee una semántica formal, basada en lógica de predicados. Su nombre se debe a que las nociones más importantes, en el dominio de conocimiento que representan, se definen mediante *descripciones de conceptos*, es decir, mediante expresiones construidas a partir de conceptos atómicos (predicados 1-arios) y roles atómicos (predicados binarios) usando los constructores de una lógica descriptiva particular. Cada lógica de descripción se identifica por un conjunto de constructores que la distingue. Así, \mathcal{FL}^- se define como la lógica descriptiva que permite el uso de cuantificación universal, conjunción y un tipo simple de cuantificación existencial (no cualificada). La lógica \mathcal{AL} extiende a la anterior añadiendo el constructor correspondiente a la negación de conceptos atómicos. A continuación, \mathcal{ALC} permite además el uso de la negación general. Análogamente, aumentando cada lógica con nuevos constructores, obtenemos la familia de lógicas descriptivas, cada una de diferente complejidad computacional.

Las Lógicas de la Descripción son un ideal punto de partida para definir lenguajes de definición de ontologías. Estas lógicas proporcionan las tareas de razonamiento necesarias para soportar la construcción, integración y evolución de ontologías de calidad. Es necesario, para su uso en la práctica, que las herramientas para adquisición de conocimiento, mantenimiento, integración e interoperatividad de las ontologías estén basadas en DL. Ya se ha avanzado mucho en este sentido y todos los esfuerzos llevan a la construcción y mantenimiento de bases de conocimiento (BC) en DL. De hecho, OWL DL tiene su contrapartida en la lógica de descripción $\mathcal{SHOIN}D_n^-$ [40].

La WS y las ontologías en sí, mejorarán el funcionamiento de la Web considerablemente, proporcionando búsquedas seguras -los sitios de búsqueda serán las páginas referidas a un concepto concreto, no todas aquellas que posean claves ambiguas o genéricas- y permitiendo el desarrollo de aplicaciones avanzadas que relacionen la información contenida en una página con estructuras de conocimiento asociadas y reglas de inferencia. La potencia real de la WS saldrá a la luz cuando se creen programas que recojan el contenido de la Web procedente de diferentes recursos (entendiendo recurso como todo objeto identificable mediante un URI), procesen la información recogida y la pongan a disposición de otros programas. La efectividad de todo ese software aumentará cuantos más contenidos orientados a máquina y servicios automatizados estén disponibles. También es necesario que el procesamiento de la información esté soportado por una justificación lógica (adecuación, completitud, etc.). La base de todo ello la constituyen los agentes inteligentes.

Agentes con capacidad deductiva

De entre las distintas definiciones de *agente inteligente* contempladas en IA para el trabajo en la web, podemos resumir la idea de agente inteligente como una entidad de software capacitada para filtrar, procesar e inferir información contenida en la Web y, a su vez, interactuar con otros sin la supervisión del usuario. Existe consenso en cuanto a los requerimientos esenciales para considerar un sistema como agente inteligente:

1. *Reactividad*: es capaz de responder *efectivamente* a las percepciones que recibe de su entorno.
2. *Pro-actividad*: es capaz de mostrar una conducta dirigida a alcanzar sus objetivos.
3. *Sociabilidad*: es capaz de comunicarse con otros agentes (para cooperar, pedir información, etc.).

Hay que distinguir, por ejemplo, entre el papel de agente inteligente y el de buscador inteligente. Un buscador puede mejorar su eficiencia aprovechando el enriquecimiento semántico de los recursos, pero funcionará rastreando e indexando la información susceptible de ser recuperada. A diferencia de éste, un agente inteligente goza de la autonomía que le confiere su reactividad, pudiendo consultar un buscador, encontrar la información a través de él y, por último, llevar a cabo una acción sobre el recurso en cuestión (todo ello en virtud de su proactividad y sociabilidad). Sin embargo, un agente debe responder en un tiempo aceptable a las necesidades del usuario. Y existen tareas que necesitarán, en la WS, planificar la búsqueda e inferencia de conocimiento.

Reactividad frente a proactividad

Se plantea por tanto el dilema entre la condición de procesar la información en tiempo real y la necesidad de efectuar tareas de razonamiento con cierto nivel de complejidad. Desde el punto de vista de esta memoria, existen tres soluciones interesantes a dicho dilema:

- *Buscar el equilibrio entre la representación simbólica y el procesamiento efectivo.* Existe una clasificación bastante completa de la complejidad de muchos lenguajes (que extienden en algunos casos a la lógica de Horn). Podemos aumentar la riqueza de la Representación del Conocimiento, permitiendo el análisis de representaciones más específicas. Esta idea fundamenta el uso de lógicas descriptivas [18].
- Algunos Demostradores Automáticos de Teoremas (DAT) pueden especializarse (como extensiones útiles de la Programación Lógica, por ejem-

plo), convirtiéndolos en candidatos naturales a ocupar el puesto de *componente pensante* del agente. Esta opción tiene dos inconvenientes: la autonomía del propio demostrador y la ausencia de métodos para su uso como dicha componente (por ejemplo, en *ciclos de razonamiento*). De hecho, una de las motivaciones de la presente memoria es la fundamentación de tales métodos.

- Puede ser aconsejable, en muchos casos, sustituir las restricciones de tiempo real por otras relacionadas con la efectividad. Esta opción es bastante delicada, pues enfrenta, de nuevo, las dos nociones de *efectividad*: la *eficiencia teórica* versus la *eficiencia en la práctica*.

Sin embargo, todos estas opciones consideran necesaria y útil la incorporación de un demostrador automático a la arquitectura de un agente, por lo que la segunda opción -metodologías asistidas por DAT para la realización de tareas cognitivamente complejas- es especialmente interesante.

Aptitudes deductivas en agentes

La capacidad de razonamiento complejo en un agente inteligente dependería de la potencia de su componente racional. De hecho, es posible que la gestión de objetivos necesite llevar a cabo muchos servicios de razonamiento, y gran parte de las respuestas ofrecidas por el agente son consecuencia de un proceso deductivo. Por tanto, una opción natural consiste en integrar un DAT como componente racional. Pero es necesario estimar el *interés* de los datos que emanan de la componente deductiva para incorporarse a la BC (*capa árbitro*), o para ponerse a disposición de otros agentes del entorno (*capa supervisor*), en el caso de potentes (o autónomos) DAT. La razón que hace necesaria esa valoración del interés de la información es que un DAT puede producir un *desbordamiento* de información indiscriminada, que carezca de interés para el usuario. La solución, apuntada en [17], consiste en incorporar las capas de tipo *árbitro* y *supervisor*. En general, y según ese autor, la proporción de datos útiles es 1000:1 [17], por tanto, la componente árbitro debe ser *estricta* a la hora de aceptar conocimiento nuevo. Con este fin se deberá proporcionar buenos criterios de selección.

En la figura 1 representamos cuál podría ser la arquitectura de un agente que usa un DAT para la depuración de errores lógicos en una BC. Consideramos la BC escrita en texto estructurado y la ontología que traduce el dominio de conocimiento. El agente de limpieza, asistido por un sistema de razonamiento automático (SRA), en este caso un demostrador de teoremas, procesa y actualiza la información contenida tanto en el dominio como en la BC. Además, mediante la capa árbitro medirá la relevancia de la información generada, bajo ciertos criterios que permiten decidir la aceptación o rechazo de esa información. El reto, sin embargo, persiste: ¿cómo se usa el demostrador

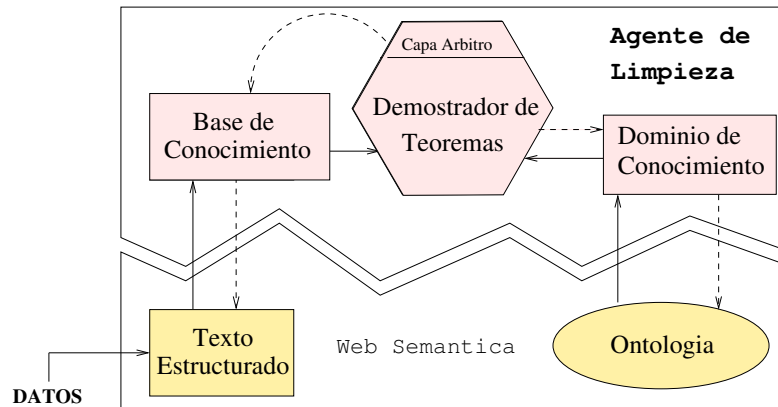


Figura 1: Agentes para la limpieza de BC

en los ciclos de ejecución del agente? Una vez diseñado un método, ¿Es *lógicamente válido*? ¿Se obtienen los resultados requeridos?

Adicionalmente, la propia administración de BC sobre dominios de conocimiento complejos (como en la WS), puede ser defectuosa. Al abordar el problema de la limpieza de BC (eternamente *incompletas* en la WS), será usual detectar anomalías, especialmente cuando están asociadas a una ontología compleja³. Las anomalías deberán ser analizadas, especialmente aquellas que surgen debido a deficiencias en el Dominio de Conocimiento, es decir, debido a una ontología deficiente.

Por otra parte, al margen del problema de la limpieza de BC, existen herramientas para la gestión y mantenimiento de ontologías, muchas de ellas utilizan herramientas visuales, pero tales sistemas no están basados, en general, en una semántica formal que nos asegure que las transformaciones permitidas al usuario no dañan propiedades fundamentales de la ontología. Por tanto, el razonamiento espacial requerido no dispone de una semántica formal asociada al propósito de la gestión. Es el usuario el encargado de asegurar la corrección de las transformaciones que realiza. Este problema de falta de fundamentación está presente no sólo en la WS, es común en la depuración de bases datos [29].

Motivación de este trabajo

La motivación fundamental de esta memoria es la necesidad de formalizar de manera lógica la limpieza de ontologías, presentando un soporte lógico general para la limpieza y tratamiento de éstas, que permita verificar los pro-

³En [3] presentamos un enfoque metodológico de solución al problema, e incluso se propone un *ciclo de limpieza*, que está fundamentado en algunos de los resultados del capítulo 8.

pios procesos de transformación de las mismas. Este paso de formalización es importante, para poder integrarlos, en el futuro, en agentes inteligentes que actúen en la WS para la depuración de datos. Concretamente, nos centramos en dotar de semántica a transformaciones ontológicas basadas en representaciones espaciales.

En el campo de la Ingeniería del Conocimiento, la verificación de BC ocupa una posición significativa. La cuestión de la verificación es un problema complejo y, en general, irresoluble. La creación de entornos con una estructura *cognitiva* compleja (como la Web Semántica) debe llevar a una revisión de algunos de los métodos de resolución clásicos. Un objetivo ya mencionado en el campo de la Inteligencia Artificial será equilibrar las actitudes reactiva y proactiva en el diseño de agentes inteligentes. Ambos problemas (verificación de bases de conocimiento y el equilibrio mencionado) se combinan en la cuestión de la verificación de metadatos de la Web mediante agentes inteligentes. Los agentes deliberativos (agentes con una conducta basada en lógica, producida por un proceso deductivo), deberán estar libres de restricciones de tiempo real en cuestiones como la verificación de ontologías formales, recuperación de información de forma estructurada (en forma de *teoría lógica*), minería de datos en la WS, etc. En un entorno como la WS, cambia la noción de percepción (o mensaje): el agente recibirá un *estímulo más complejo* [12]. Esta cuestión es más importante cuando el número de sistemas heterogéneos que interactúan aumenta (así como el número de ontologías diferentes).

Contribución

Para investigar el problema de la verificación de bases de conocimiento asociadas a ontologías, en esta memoria proponemos un enfoque práctico, diseñando metodologías para la depuración de bases de conocimiento asociados a ontologías, con la asistencia de un DAT. Creemos que esta metodología debe ser el primer paso hacia el diseño de *agentes inteligentes de limpieza* para la WS. Gran parte de los métodos ofrecidos se sustentan en la utilización de un formalismo mereotopológico para representar espacialmente el conocimiento conceptual capturado en las ontologías.

Adicionalmente, en esta memoria se estudia el problema de la verificación y validación del conocimiento espacial cualitativo, y la verificación de BC escritas en un lenguaje concreto y adecuado a ese tipo de conocimiento. Esta aproximación, aunque particularizada a este caso, es aplicable a otros ámbitos.

Por último, la presente memoria formaliza metodologías para la gestión fiable de *ontologías formales*, es decir, a teorías lógicas para la descripción explícita de un dominio. Desde este punto de vista, y para mostrar el nivel de ge-

neralidad de los problemas y las soluciones propuestas, las BC y las ontologías se suponen descritas en DL, o incluso en Lógica de Primer Orden (LPO). Es evidente, por otro lado, que son aplicables a otros lenguajes. Como ya hemos comentado, el análisis y diseño de este tipo de métodos es un paso previo (e imprescindible) a la inserción de ciertas aptitudes deductivas en agentes diseñados para la *depuración inteligente de ontologías*.

Trabajos relacionados

Análisis ontológico

En el campo de las ontologías, es de destacar el trabajo de N. Guarino y C. Welty [36], quienes muestran cómo un análisis riguroso de las *meta-propiedades*⁴ de los nodos taxonómicos puede ayudar a utilizar la relación de subsunción de forma correcta y ordenada, debido a las restricciones que sobre esta relación se establecen. Ese análisis proporciona orden a las taxonomías facilitando su comprensión, comparación e integración y constituye el primer paso hacia una metodología general para el análisis conceptual de ontologías.

Por otra parte, en [62] los autores explotan el hecho de que el lenguaje OWL *DL* se corresponde con $SHOIN\mathcal{D}_n^-$. Ésta, como lógica de descripción, es un fragmento de LPO. Proponen la utilización de métodos estándar de razonamiento automático para LPO como mecanismos para razonar con OWL *DL*, en lugar de usar métodos basados en DL. La idea no es la de sustituir a estos últimos, sino la de combinarlos o suplementarlos para desarrollar y abordar problemas que quedan fuera del nivel expresivo de los razonadores en DL. Los autores analizan experimentalmente esta idea. Concretamente, utilizan un razonador en DL, Fact++, para clasificar tres ontologías, y extraen una serie de tareas de subsunción que son traducidas a LPO y resueltas con Vampire⁵, comparando los dos sistemas. La principal conclusión de su trabajo es que Vampire resuelve de forma efectiva muchos de los problemas derivados de la clasificación de las ontologías, y se sugieren mejoras en el preprocesamiento. No obstante, ciertos experimentos siguen dejando espacio para razonadores en DL. Por otro lado, aunque en la lógica de descripción $SHOIN\mathcal{D}_n^-$ la consistencia es decidible, no existe un procedimiento de decisión efectivo conocido, y por tanto, no hay razonadores implementados. Los experimentos confirman que Vampire es una buena alternativa en este caso.

En [24] se muestra la aplicación de un DAT (el sistema SNARK) para ve-

⁴Basadas en los conceptos filosóficos de rigidez, identidad, unidad y dependencia

⁵Demostrador en LPO de propósito general DAT basado en resolución.

rificar la semántica declarativa de los lenguajes en la WS. Primero se realiza una traducción a lógica de primer orden, para luego poder aplicar el demostrador y hallar las inconsistencias. Esta idea, aplicada a la especificación del lenguaje, es útil para el problema de la verificación de BC.

Soportes visuales

Las ontologías son la base de muchas técnicas de visualización de contenidos Web para navegación y búsqueda así como para el análisis y localización de recursos.

Respecto al uso de la clasificación ontológica para la representación visual de la información, hay dos trabajos que merecen ser mencionados. Un primer ejemplo de visualización de recursos Web es *Hyperbolic Tree* [47], herramienta basada en geometría hiperbólica desarrollada en 1995. Esta visualización se centra en estructuras sintácticas en árbol, y contienen semántica, pero de manera implícita y ad hoc. El método utilizado consiste en trasladar el árbol al plano hiperbólico y, a continuación, aplicar éste al disco unidad. Mediante translaciones, permite desplazar el foco dentro del disco, de manera que el árbol representado puede “verse” como si se dispusiera de un zoom. La ventaja de esta herramienta es que necesita modestos requerimientos computacionales.

En 2002, los autores de [25], proporcionan un método de visualización acompañado de ejemplos de resolución de tareas como análisis y comparación de datos o *debilitación de consultas*⁶. Restringen su estudio a ontologías simples, llamadas *light-weight*, jerarquías de conceptos donde las subclases pueden ser incompletas (la unión de las subclases forman un subconjunto propio de su superclase) o bien solaparse (existiendo individuos comunes). En cuanto al software utilizado, se trata de *Spectacle*, uno de los productos del proveedor Administrator⁷. Más concretamente, es *Cluster Map*, uno de sus componentes clave, el que se usa para la visualización de la ontología. Esta herramienta permite crear presentaciones de información a la medida del usuario. Así, por ejemplo, gracias a la semántica de la WS, la navegación posibilitará la realización de búsquedas en muy pocos pasos (cliks).

El tipo de visualización que debe utilizarse dependerá de la naturaleza de los datos a representar y de las tareas a las que tendrá que proporcionar soporte. En cuanto a la naturaleza, cabe desear que las ontologías, en la WS, sean del tipo *light-weight*, con una taxonomía sin “cruces”. En un segundo trabajo [26], los mismos autores ofrecen un interesante y

⁶La tarea denominada *query-relaxation* consiste en encontrar respuestas aproximadas a consultas para las que no hay respuesta exacta (explícita)

⁷<http://www.administrator.nl>

amplio análisis comparativo de distintas técnicas de visualización e intentan subsanar el hecho de que la mayoría de las herramientas de visualización desarrolladas polaricen su objetivo en la representación de ontologías complejas, olvidando (ignorando según ellos) mostrar las instancias respectivas⁸. Con ese propósito presentan una técnica para ontologías light-weight con gran número de instancias, soportando una variedad de tareas como análisis de datos, consulta y navegación. En general, estos trabajos se centran en la representación visual y adolecen de mecanismos de inferencia o una semántica formal asociada.

Representación de relaciones espaciales

Recientemente, en [45], se aborda la representación de las relaciones espaciales en ontologías para la WS. Los autores enfatizan la importancia de las relaciones espaciales en la construcción de ontologías web y sugieren un método de representación de las mismas, usando las capacidades de OWL basadas en DL. Escogen la lógica \mathcal{ALC} extendiéndola, mediante el uso de un constructor de roles basado en predicados⁹, a \mathcal{ALCRP} , en el dominio de los subconjuntos cerrados regulares no vacíos del espacio topológico \mathbb{R}^2 . Añaden nuevos axiomas para definir dos relaciones de RCC8, la conexión externa y la desconexión. Las definiciones en DL se traducen a OWL y comprueban, con una ontología de ejemplo, que la representación de la misma es fácil y flexible.

Análisis y limpieza de datos

Existen trabajos relacionados con la limpieza de datos anómalos (dirty-data) como paso previo al uso de cualquier aplicación de análisis o gestión sobre grandes bases de datos. Destacamos el realizado por Kim et al.[42], donde se plantea la necesidad de desarrollar una taxonomía de datos anómalos, así como una métrica, que permitan conocer la calidad del conocimiento almacenado en una base de datos antes de derivar conocimiento a partir de ella o tomar decisiones basadas en ese conocimiento. Desarrollan una taxonomía, detallada y exhaustiva, de datos anómalos partiendo de un conjunto de hipótesis como son: suponer que el usuario busca datos escritos en el mismo lenguaje en que éstos están almacenados, restringir el tipo de dato a cadena o numérico (no imágenes, por ejemplo), considerar sólo datos, no meta-datos. A continuación proponen un conjunto de técnicas para prevenir, comprobar o reparar cada uno de los tipos de anomalía descritos por la taxonomía. Estas técnicas, al no tratar metadatos, no necesitan de una semántica asociada, pero tampoco son directamente aplicables a ontologías.

⁸O viceversa, muestran gran número de instancias dejando de lado la información ontológica

⁹Disjoint = $\exists(\text{has_area})(\text{has_area}).\text{dc}$, Touching = $\exists(\text{has_area})(\text{has_area}).\text{ec}$

Diagnosic del diseño de bases de datos deductivas

Las técnicas para representar y explotar dominios de conocimiento (como thesaurus) son utilizadas por herramientas de diseño de bases de datos inteligentes. En [54] se estudia a qué nivel esas técnicas pueden mejorar la capacidad de diagnosis de las herramientas de diseño. En primer lugar, los autores introducen una clasificación de los tipos de inconsistencia de diseño que pueden detectarse y repararse: inconsistencias semánticas (transitividad o falta de información a cerca de relaciones entre conceptos), conceptos inconsistentes (debido a que sus propiedades no están explícitas en el diseño), redundancia en propiedades y relaciones (al ser heredadas), elementos redundantes (por la presencia de conceptos sinónimos). Los resultados de este estudio, que se realiza usando la herramienta IOA de diseño de bases de datos inteligentes, indican que las técnicas de thesaurus, diccionario y reconciliación del conocimiento aumentan la capacidad de detección y resolución de inconsistencias del diseño de bases de conocimiento¹⁰, pero apuntan que dichas capacidades dependen en gran medida de la seguridad y completitud de la estructura del dominio de conocimiento representado. Es decir, en nuestro caso, dependen en gran medida de las propiedades lógicas de la ontología asociada a los datos.

Contenido de la memoria

La presente memoria se estructura en tres grandes bloques:

- I Tratamiento automatizado de la teoría RCC y métodos asistidos por SRA para obtener extensiones de dicha teoría. Generalizaciones del retículo de relaciones asistido por SRA. Este bloque abarca los capítulos 2, 3 y 4.
- II Aplicación de RCC como cálculo *meta-ontológico* para su posterior uso en el análisis de anomalías en ontologías. Este tema se aborda en los capítulos 5 y 6.
- III Métodos asistidos por SRA para la depuración de bases de conocimiento asociadas a ontologías complejas. Comprende los capítulos 7 y 8.

El primer bloque proporciona un método para la extensión de teorías mediante nuevos conceptos/relaciones, generando las extensiones correspondientes, comprobando la unicidad del conjunto de las extensiones obtenidas e interpretando bajo un marco teórico apropiado las relaciones introducidas. Queremos hacer notar que este estudio tiene interés específico para el resto de la

¹⁰Como por ejemplo técnicas para eliminar las inconsistencias producidas por el uso de diferentes tiempos de un mismo verbo, o como la representación explícita de sinónimos.

memoria, pero, además, posee interés propio en el campo del Razonamiento Espacial Cualitativo (un estudio pormenorizado y asistido por SRA de RCC), y para la extensión de ontologías formales, pues proporciona un método práctico para la inserción ontológica. También pone de manifiesto la necesidad de una reinterpretación de los compromisos adoptados en la construcción de la ontología original para así acomodarse a la nueva ontología extendida¹¹.

Con esta metodología se prepara el desarrollo del siguiente bloque, en el que se aborda el refinamiento de ontologías mediante la introducción de nuevos objetos en la ontología (conceptos o elementos), utilizando como soporte formal la mereotopología introducida en el bloque anterior. La tercera parte representa otra aproximación a la depuración de ontologías con un enfoque de carácter práctico, y con la asistencia de SRA.

El último capítulo de la memoria está dedicado a las conclusiones que hemos obtenido en la elaboración de este trabajo, así como una breve descripción del trabajo futuro.

El diagrama de la figura presentada al final de esta introducción, describe el esquema general de la memoria. A continuación presentamos una descripción más detallada de los capítulos que la componen.

PARTE I:

TRATAMIENTO AUTOMATIZADO DEL RAZONAMIENTO ESPACIAL

Capítulo 1: Preliminares

En este capítulo se introducen las herramientas utilizadas en los capítulos siguientes. El tipo básico de conceptualización que utilizaremos, para el razonamiento mereotopológico, está basado en retículos. Esta elección no supone una gran restricción: teorías para la extracción de ontologías como el análisis formal de conceptos [31] proporcionan tales retículos y son utilizadas para la gestión de ontologías (por ejemplo en AKT¹²). Una vez recordadas las propiedades básicas de la teoría de retículos, introducimos el concepto de *extensión conservativa* con respecto a un conjunto de fórmulas. Esta nueva relación será clave en el estudio de las extensiones de la teoría en que se centra esta memoria, el cálculo RCC¹³.

¹¹En el caso de las extensiones de RCC, proporcionamos tales reinterpretaciones, poniendo de manifiesto su necesidad. Véanse las secciones 3.4, 3.5 y 4.2

¹²Advanced Knowledge Technologies. Proyecto del Engineering and Physical Sciences Research Council (EPSRC) que tiene como objetivo desarrollar y extender tecnologías para la gestión del conocimiento: <http://kmi.open.ac.uk/projects/akt/about.html>

¹³De hecho, es una noción que en la práctica puede ser usada como noción de *extensión robusta* de una conceptualización [11].

A continuación se describen brevemente las Lógicas de Descripción (o Lógicas Descriptivas, DL)¹⁴. Consideradas como subconjunto de la Lógica de Primer Orden, poseen una semántica formal que permite una adecuada representación del conocimiento. Su complejidad computacional ha sido ampliamente estudiada [18]. De hecho, confieren una semántica y un soporte de razonamiento a los lenguajes de representación de ontologías. Traducir, cuando sea posible, una ontología a DL nos permite utilizar los SRA existentes para ese formalismo. Se describe la estructura de las bases de conocimiento en DL y las tareas que facilitan al usuario los servicios de razonamiento fundamentales (satisfacibilidad, subsunción, consistencia, etc.). Recordamos algunos aspectos del proceso de normalización (que simplifica las bases de conocimiento facilitando la detección de contradicciones sintácticamente obvias).

Con el objetivo de justificar la aplicación de demostradores automáticos para la detección de anomalías, necesitamos utilizar la formalización lógica clásica de las bases de datos (debida a R. Reiter [58]). Es decir, las hipótesis bajo las que se puede entender la evaluación de las consultas a una base de conocimiento como un procedimiento para conseguir consecuencia lógica: *nombres únicos, clausura de dominio y mundo cerrado*.

En el estudio de la inconsistencia de bases de conocimiento, las lógicas anteriores (apropiadas para la representación y el razonamiento) no tienen suficiente alcance, no abarcan el tratamiento de información inconsistente. En ocasiones es posible dividir el conocimiento a tratar en subconjuntos consistentes sobre los que sí podamos razonar con fiabilidad. Aún así, deberá estudiarse la fiabilidad de la información deducida a partir de ese subconjunto, sobre todo si la base de conocimiento es inconsistente. La teoría de argumentos es una herramienta poderosa a la hora de clasificar las deducciones (argumentos, pares de la forma *hipótesis/conclusión*), mediante una *estructura argumentativa*¹⁵ [23].

En la sección 1.5 resumimos las características de OTTER y MACE4, un demostrador automático y un buscador de modelos asociado, respectivamente. Son los SRA utilizados en esta memoria. Han sido de gran ayuda como asistentes en la prueba de teoremas así como en el cálculo y obtención de modelos,

¹⁴Como hemos comentado, el uso de las lógicas descriptivas podría ser generalizado a la lógica de primer orden en muchos de los estudios y resultados de esta memoria. Su elección viene motivada no sólo por ser el soporte lógico de los lenguajes para representación de ontologías en la Web. Este formalismo proporciona una forma elegante de expresar las propiedades entre conceptos en una conceptualización.

¹⁵En esta memoria nos limitaremos a resolver el problema de la consistencia de los argumentos y cómo determinar la relación de corte conservativo en el uso de DAT para limpiar BC. El uso de la jerarquía obtenida para BC de tamaño medio a partir de experimentos es una línea interesante de investigación para el futuro.

C :	Conexión
DR :	Discreto
O :	Solapamiento
P :	Parte de
PP :	Parte Propia
DC :	Desconexión
EC :	Conexión Externa
PO :	Solapamiento Parcial
EQ :	Igualdad
TPP :	Parte Propia Tangencial
NTPP :	Parte Propia No Tangencial

Figura 2: Relaciones base RCC

concretamente en el caso de las extensiones de la teoría bajo estudio¹⁶.

El capítulo concluye con un breve resumen de las definiciones elementales sobre topología y espacios topológicos así como con la introducción de una teoría para el razonamiento espacial cualitativo. Tal teoría, la denominada *Cálculo de Conexión de Regiones* (RCC) [15] será el eje de esta memoria. Esta teoría, de carácter mereológico, trata sobre entidades espaciales (concretamente, conjuntos regulares) desde el punto de vista topológico. Los problemas de satisfacción de restricciones asociados a su cálculo relacional han sido profundamente estudiados [61] [59] [60]. Las relaciones binarias fundamentales de RCC, denotado por \mathcal{R}_{RCC} , aparecen en la figura 2 (hemos omitido las relaciones inversas). Se describen los modelos topológicos pretendidos de la teoría, se introduce la tabla de composición de sus relaciones y particularizamos el concepto de problema de satisfacción de restricciones a la teoría RCC.

Capítulo 2: Tratamiento automatizado del Razonamiento en RCC

Este capítulo está dedicado a cumplir con dos objetivos: Analizar, utilizando SRA, la potencia deductiva del RCC y a continuación, analizar desde el punto de vista del razonamiento automático, los cálculos inducidos por la *proyección reticular* y los *movimientos topológicos* (cualitativos) en RCC. Estas dos cuestiones son de especial relevancia para razonar con anomalías en ontologías (en el capítulo 6).

Con respecto al primero de los objetivos, certificamos mediante OTTER que RCC prueba que RCC8, subconjunto de RCC formado por las relaciones PO,

¹⁶Una nota importante: como MACE4 no está formalmente verificado, hemos certificado los resultados producidos por este sistema con OTTER (o a mano). Tales pruebas sí son fácilmente verificables.

TPP, NTPP, EQ, TPPi y NTPPi, EC y DC y, de manera análoga, RCC5 formado por DR, PP, PO, PPi y EQ, constituyen conjuntos exhaustivos de relaciones disjuntas dos a dos, es decir, que RCC8 y RCC5 son JEPDs. Estas afirmaciones constituyen dos teoremas cuyas pruebas son similares y, en el caso del primero, la prueba se detalla e incluye. La importancia de la demostración automática de tales resultados estriba en que certificamos que éstos se siguen estrictamente de RCC, es decir, no se hace uso de propiedades espaciales intuitivas que, aún siendo válidas, no sean demostrables por la propia teoría. Este fenómeno ha invalidado algunos resultados obtenidos en el análisis de teorías espaciales [53]. Siguiendo esta idea, demostramos que la estructura de retículo que posee el conjunto \mathcal{R}_{RCC} de relaciones de RCC es demostrable por la propia teoría RCC. Dicha estructura es no modular y no distributiva.

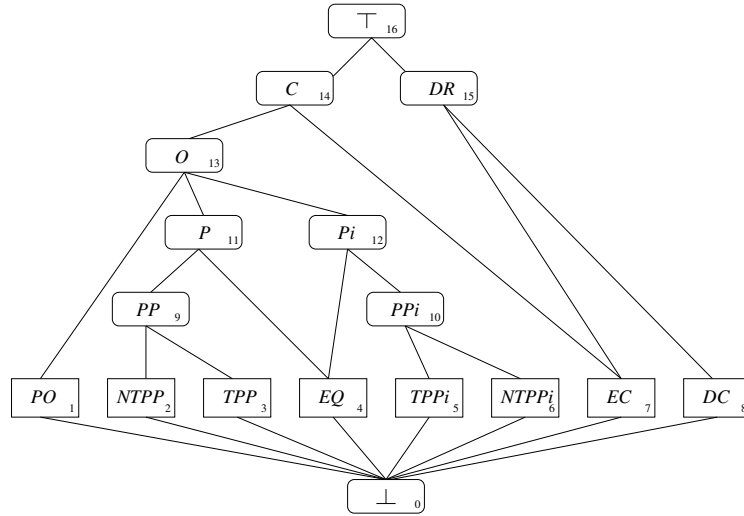
Para probar la estructura de JEPD del conjunto RCC8, necesitamos realizar 28 pequeñas pruebas de resultados del tipo

$$\forall xy(DC(x, y) \rightarrow \neg EQ(x, y))$$

En este capítulo mostramos, como ejemplo, algunas de éstas.

Es importante reseñar que la teoría que facilitamos al demostrador está formada por aquellos axiomas de RCC que definen las relaciones y que podemos ver en la sección 1.8.1. Y en el conjunto soporte sólo la negación del resultado a probar (en el ejemplo anterior, la existencia de dos regiones desconectadas que a su vez coinciden). Como con esta entrada el demostrador proporciona una refutación, hemos demostrado que sólo es necesario utilizar los *axiomas definidores* de RCC para la prueba, y no aquellos que hablan de otras operaciones (véase la sección 2.1). Análogamente ocurre con la demostración de que las ocho relaciones forman un conjunto exhaustivo. Este hecho es clave para el posterior desarrollo de este primer bloque, y se repite en la última tarea que nos planteamos con respecto al primer objetivo, a saber, demostrar que la estructura de retículo de \mathcal{R}_{RCC} es demostrable a partir de los axiomas definidores de RCC (véase su diagrama de Hasse en la figura 3).

Sin embargo, en la prueba de ese resultado procedemos de manera distinta. La utilización sólo de axiomas definidores en las pruebas anteriores induce a pensar que, posiblemente, el diagrama de Hasse puede ser caracterizado *ecuacionalmente*, bajo ciertos compromisos de *cierre del mundo* (es decir, retículos con 17 relaciones). La demostración del resultado que presentamos certifica, de hecho, tal hipótesis. Aplicamos el generador de modelos MACE4 a un fichero de entrada que contiene los axiomas de estructura de retículo y las conexiones entre las relaciones de RCC que aparecen en el diagrama, utilizando `and` y `or` para referirnos a la conjunción y la disyunción respectivamente. El generador proporciona, mediante las tablas de Cayley de las relaciones `and` y `or`, un único retículo que satisface la representación del diagrama de Hasse descrito para RCC. Seguidamente comprobamos -mediante OTTER o a mano- que son

Figura 3: El retículo \mathcal{R}_{RCC}

teoremas de RCC que cada una de las fórmulas que establecen las posiciones relativas de las relaciones de \mathcal{R}_{RCC} en el diagrama. Se trata, por tanto, de probar a partir de los axiomas de RCC y utilizando OTTER, una serie de resultados de carácter algebraico, como los siguientes:

- $RCC \vdash \forall x \forall y (0(x, y) \rightarrow PO(x, y) \vee P(x, y) \vee Pi(x, y))$ (lo que implica que $O = PO \vee P \vee Pi$ en el retículo).
- $RCC \vdash \neg \exists x \exists y (PO(x, y) \wedge P(x, y))$ (lo que implica que $PO \wedge P = \perp$ en el retículo).

De hecho, somos capaces de aislar un conjunto de ecuaciones \mathcal{E}_{RCC} que caracteriza al retículo \mathcal{R}_{RCC} bajo clausura de dominio (y nombres únicos). Nótese que, desde el punto de vista de ese conjunto, estamos describiendo las relaciones en \mathcal{E}_{RCC} de manera *exógena*, es decir, caracterizamos una relación en función del resto pero no hablamos de los pares que la componen. Este carácter permite trabajar de manera más sencilla con las relaciones de RCC, frente al carácter endógeno del conjunto de axiomas de RCC (el cual define internamente las relaciones, desde el punto de vista de los elementos que las identifican). Es decir, con \mathcal{E}_{RCC} hablamos de la conceptualización de la ontología de relaciones espaciales de RCC.

Es importante recordar que el retículo formado por \mathcal{R}_{RCC} no es modular ni distributivo, lo cual impide utilizar las propiedades de este tipo de retículos. Por consiguiente, la teoría de retículos extendida con \mathcal{E}_{RCC} no posee tales restricciones. Esto induce a pensar que existe una cierta facilidad para extender el retículo con nuevas relaciones (lo que haremos en los capítulos 4 y 5).

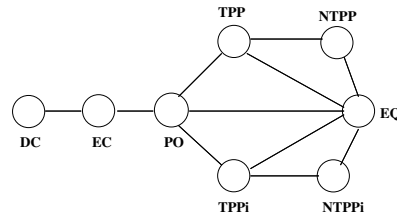


Figura 4: Iconización de RCC8 para determinar los entornos conceptuales

La parte dedicada a cálculos inducidos por otras nociones asociadas a RCC comienza con la obtención de la tabla de composición de la proyección reticular de RCC. Estos cálculos serán de utilidad en el segundo bloque de esta memoria. La proyección reticular asocia a cada relación del retículo \mathcal{R}_{RCC} el subconjunto de RCC8 formado por las relaciones que están contenidas en la primera. Un primer resultado afirma que la definición de proyección reticular de una relación es equivalente a dicha relación, resultado demostrable en RCC (hecho que a su vez es consecuencia de que \mathcal{R}_{RCC} tenga la estructura que determina su diagrama de Hasse).

Mediante la noción de contigüidad, analizamos algunas propiedades de las proyecciones reticulares, basándonos en el método ideado por Freksa sobre entornos conceptuales de las relaciones espaciales y temporales (para el razonamiento temporal basado en intervalos según Allen) [27][28]. Utilizaremos la iconización de la estructura de entorno conceptual que puede verse en la figura 4. En esta figura, dos posiciones son contiguas si se puede pasar de una a otra mediante una transformación topológica elemental¹⁷. Partiendo de esta idea de *contigüidad* entre relaciones del retículo \mathcal{R}_{RCC} , definimos lo que entendemos como *entorno conceptual*, y demostramos que la proyección reticular de toda relación de \mathcal{R}_{RCC} es un tal entorno. Así, podemos aplicar un método de razonamiento sobre proyecciones reticulares para estudiar el retículo que nos ocupa, \mathcal{R}_{RCC} , y como consecuencia estudiar tablas de composición de *todas* las relaciones de RCC. En dicha tabla se identifican ciertos entornos conceptuales, algunos de ellos correspondientes a proyecciones reticulares de relaciones de RCC. Por último, se incluyen las consideraciones que hemos extraído del análisis de la tabla general, entre las que destacamos que el único subcálculo es el que proporciona RCC8, entendiendo como subcálculo un subconjunto de relaciones cerrado bajo composición.

El estudio es exhaustivo, pues es necesario explorar (en el bloque II) cuáles de estas herramientas serán útiles.

¹⁷Es decir, durante la transformación no alcanzamos ninguna otra posición distinta a éstas según RCC8.

Capítulo 3: Extensiones robustas de RCC por indefinición

Este capítulo está dedicado al estudio de las extensiones automatizadas de RCC mediante una relación de indefinición, utilizando como asistentes OTTER y MACE4.

La idea consiste en extender el cálculo RCC8 mediante una nueva relación que representará la indefinición (desconocimiento de la relación concreta existente entre dos regiones, hasta cierto grado). Se necesita, por tanto, considerar a la nueva relación disjunta con RCC8, y por tanto la unión con RCC8 proporcionará un nuevo JEPD. La demostración de la existencia de tales extensiones proporciona un método para obtener extensiones genéricas de \mathcal{R}_{RCC} y, de hecho, de cualquier conceptualización parecida: en primer lugar, procedemos a la generación con MACE4 de los modelos correspondientes a las extensiones y, a continuación, obtenemos con OTTER una prueba de la exhaustividad del conjunto de modelos. Es importante reseñar que las extensiones son conservativas con respecto al conjunto \mathcal{E}_{RCC} definido en el capítulo anterior (el conjunto de fórmulas que caracterizan a \mathcal{R}_{RCC}). En el caso de las extensiones que necesitamos, MACE4 genera ocho distintos. Por otro lado, el demostrador confirma que los ocho modelos descritos por las tablas de Cayley que MACE4 genera son únicos¹⁸.

Es evidente que la introducción de tales extensiones¹⁹ no tiene utilidad si no está acompañada de una interpretación, en el mismo dominio ontológico que RCC, de los elementos de dicha extensión. Por tanto, necesitamos introducir teorías mereotopológicas asociadas a dichas extensiones para poder aceptarlas, un *soporte topológico* (los modelos espaciales que representa). En el caso particular que nos ocupa, proporcionamos cuatro tipos de interpretación, dos de las cuales son esenciales para la clasificación de posibles anomalías en bases de conocimiento (vía la oportuna interpretación mereotopológica de las mismas). Son las obtenidas por *pulsación*, *contracción*. Las dos restantes consisten en adaptar la clásica *yema/huevo* [16] para regiones con indeterminación y en extender la asociada a conjuntos *rough* [56] en trabajos que aplican ese formalismo al razonamiento espacial [63] [10].

Las dos primeras interpretaciones que describimos están basadas en los *espacios topológicos con pulsación* y los *espacios topológicos con contracción*. Para decidir si el proceso de interpretación es correcto (es decir, que el retículo asociado a las relaciones en el espacio topológico elegido corresponde al obtenido

¹⁸Esta prueba consiste en suponer que tenemos una relación que no coincide con siete de las ocho halladas y comprobar que se trata de la octava. El demostrador deberá verificar que esa nueva relación satisface tablas de Cayley del octavo modelo. Se muestran las tablas en cuestión y, como ejemplo, una de las refutaciones.

¹⁹Que, realmente, son extensiones de la ontología RCC por inserción de un nuevo elemento no definible en la propia RCC.

con MACE4), debemos reinterpretar varias relaciones de RCC en este contexto. Denotemos por I_k con $k \in \{1, 2, \dots, 8\}$ las ocho posibles relaciones de indefinición halladas.

Una adecuada interpretación de las relaciones de indefinición en dichos espacios es obtenida, pero cabe reseñar un primer resultado de *no interpretabilidad*: una de las ocho extensiones no es interpretable en espacios con pulsación (teorema 3.2.6), pero sí lo es en espacios con contracción.

El *Teorema de Inserción Ontológica de las relaciones de indefinición* (teorema 3.2.11), certifica que la estructura de retículo de los espacios topológicos junto con la oportuna interpretación de las nuevas relaciones coincide con la obtenida automáticamente. La prueba de este teorema consta de cuatro pasos. Denotamos por L_k el retículo $RCC + \{I_k\}$ con $k \in \{1, 2, \dots, 8\}$ y por R^Ω a la interpretación de una relación R en la estructura topológica Ω , y Ω_k como la estructura en el lenguaje de $RCC + \{I_k\}$ con $k \in \{1, 2, \dots, 8\}$. L_{RCC} representará el retículo de las relaciones de \mathcal{R}_{RCC} .

- * En primer lugar es necesario probar que se conserva el orden original de RCC, es decir, que para cualesquiera $R_1, R_2 \in \mathcal{R}_{RCC}$ se verifica que

$$R_1^\Omega \leq R_2^\Omega \text{ en } L_{RCC} \Rightarrow R_1^{\Omega_k} \leq R_2^{\Omega_k} \text{ en } L_k, \forall k \in \{1, 2, \dots, 7, 8\}$$

Esta condición equivale a que L_{RCC} se puede incorporar al retículo L_k , para cada $k \in \{1, 2, \dots, 8\}$.

- * En segundo lugar hay que demostrar que, para cada $k \in \{1, 2, \dots, 8\}$, la relación I_k está en la posición que indica el diagrama de Hasse correspondiente al retículo L_k . Para ello hay que comprobar que se verifican las siguientes condiciones:

$$\begin{array}{ll} I_1^{\Omega_1} < PP^{\Omega_1} & I_2^{\Omega_2} < P^{\Omega_2} \\ I_3^{\Omega_3} < PP_i^{\Omega_3} & I_4^{\Omega_4} < P_i^{\Omega_4} \\ I_5^{\Omega_5} < O^{\Omega_5} & I_6^{\Omega_6} < C^{\Omega_6} \\ I_7^{\Omega_7} < DR^{\Omega_7} & \end{array}$$

- * En una tercera etapa demostramos que para cada $k \in \{1, 2, \dots, 8\}$, $RCC \cup \{I_k^{\Omega_k}\}$ forma un conjunto de relaciones disjuntas dos a dos en el retículo L_k .
- * Por último, comprobamos que, para cada $k \in \{1, 2, \dots, 8\}$, el diagrama de Hasse de L_k representa la operación unión en el retículo de relaciones de Ω_k .

Nótese que la existencia de una demostración certificada de la estructura del retículo asociado a una teoría no implica, en general, que la tabla de composición asociada sea demostrable. Es también evidente que sin una definición

de la nueva relación en función de las antiguas no es posible un cálculo certificado, en general. Sin embargo, en este caso, debido al estudio ya realizado de RCC, es posible proporcionar la tabla de composición de los nuevos JEPDs. Ilustramos este hecho con el cálculo de éstos.

La interpretación topológica mediante pulsación/contracción no es la única posible. Presentamos una interpretación alternativa basada en el trabajo clásico de Cohn y Gotts [16] sobre la representación *huevo-yema* (egg-yolk) de regiones con frontera indeterminada (llamadas también regiones vagas). Una vez realizada la oportuna adaptación de la teoría RCC²⁰ -esta teoría no fue concebida inicialmente para representar y razonar con este tipo de regiones- existen 46 posibles posiciones relativas entre dos regiones vagas (entre las 4 subregiones que las representan), y debemos considerar como conjunto base de relaciones a RCC5²¹.

Interpretando las relaciones de indefinición con pulsación mediante el formalismo "huevo-yema" se establece que, exceptuando I_6 , la relación de indefinición junto con RCC5 constituye un retículo. La interpretación concreta de una relación consistirá, en definitiva, en un subconjunto de las 46 posiciones relativas posibles²². Este mismo proceso se realiza para la contracción. De manera análoga al caso anterior, obtenemos los retículos que resultan de la extensión de RCC5 por el conjunto de indefiniciones, así como los tres JEPDs resultantes.

Una última alternativa la proporciona la interpretación con conjuntos tipo Rough [56]. Esta teoría proporciona un método para aproximar subconjuntos del universo si éste ha sido previamente dotado de una partición o relación de equivalencia. El enfoque de este tipo debido a L. Vigneron y A. Wasilewska [63] no es directamente aplicable a nuestro caso, pues se basa en espacios topológicos en los que los abiertos son a la vez cerrados y viceversa, es decir, espacios topológicos totalmente disconexos. Para solventar este problema se definen los espacios topológicos *DR-Rough* así como los recubrimientos *DR-Rough*, proporcionando los ejemplos correspondientes. Se interpretan las relaciones de indefinición mediante aproximaciones tipo rough, cerrando con este resultado el presente capítulo.

²⁰La adaptación consiste en representar una región vaga A mediante dos subregiones, una en el interior de la otra, de manera que cada subregión tiene frontera definida, fronteras que limitan la indefinición en A. Así A está representada por el área que abarca la subregión interior, la exterior, o, debido a la indefinición, una zona intermedia.

²¹El hecho de tener como referencia RCC5 en lugar de RCC8, es debido a que, al no conocer el perfil exacto de cada región, carece de sentido considerar la tangencialidad.

²²Además se determinan tres JEPDs de interés.

Capítulo 4: Aplicación del método: extensiones con interpretación por refinamiento

En este capítulo, utilizando la metodología dada en el capítulo anterior, obtenemos y estudiamos ciertas extensiones con refinamiento de la teoría RCC. En este caso pretendemos añadir al conjunto \mathcal{R}_{RCC} una nueva definición de conexión que extiende la conexión externa EC. De esta forma, ilustramos la generalidad del método de extensión automatizada descrito en el capítulo anterior. En un primer apartado demostramos el teorema de existencia de once relaciones verificando tal propiedad, denotadas por μC_k con $k \in \{1, \dots, 11\}$ que dan lugar a once extensiones \mathcal{E}_{RCC} -conservativas de L_{RCC} , denotadas por $RCC + \mu C_k$. Además, se prueba la unicidad del conjunto de extensiones halladas. De manera análoga al capítulo anterior, el generador de modelos proporciona los once modelos diferentes, las once extensiones en cuestión y, haciendo uso del demostrador, comprobamos que son sólo once.

El siguiente paso consiste en interpretar las nuevas relaciones, excepto la primera de ellas μC_1 , dado que es de máxima indeterminación, utilizando un método de representación ideado por Egenhoffer [22]. En primer se necesita un refinamiento del concepto primitivo *región espacial*, para poder obtener una buena interpretación²³. Así, consideramos regiones construidas a partir de *cells* de la misma dimensión. Se basa en utilizar las matrices que recogen información sobre las intersecciones entre los interiores, fronteras y exteriores de las dos regiones cuya relación caracteriza. El valor 0 indica intersección vacía, el 1, el caso contrario. Según la posición, la relación entre A y B viene descrita por la tabla siguiente:

\cap	B°	∂B	$e(B)$
A°	$A^\circ \cap B^\circ$	$A^\circ \cap \partial B$	$A^\circ \cap e(B)$
∂A	$\partial A \cap B^\circ$	$\partial A \cap \partial B$	$\partial A \cap e(B)$
$e(A)$	$e(A) \cap B^\circ$	$e(A) \cap \partial B$	$e(A) \cap e(B)$

Se facilitan las caracterizaciones de las relaciones de $\mathcal{R}_{RCC} \cup \{\mu C_1, \dots, \mu C_{11}\}$.

Una vez elegida la interpretación ontológica, se obtiene la estructura reticular del conjunto de nuevas relaciones junto con la conexión C y la conexión externa EC, facilitándose el correspondiente diagrama de Hasse. Dado que no se obtiene una interpretación de μC_1 , es necesario justificar su posición en el retículo, descartando el resto de posiciones posibles, justificación obtenida mediante MACE4, que proporciona un modelo, cuyas tablas de Cayley se presentan. Finalmente obtenemos un JEPD de relaciones de conexión. El capítulo finaliza presentando la tabla de composición del JEPD formado por las relaciones $\{DC, \mu C_{11}, TPP, NTPP, TPPi, NTPPi, EQ\}$.

²³Este refinamiento no fue necesario en el capítulo anterior.

PARTE II:
 APLICACIÓN DE LA MEREOTOPOLOGÍA AL ANÁLISIS DE
 ONTOLOGÍAS

Capítulo 5: Interpretación metaontológica de RCC

La teoría RCC ha sido ampliamente estudiada en la primera parte de esta memoria desde el punto de vista de formalización para la representación espacial cualitativa. En este capítulo se aborda la cuestión de interpretar dicha teoría como *ontología general sobre las relaciones entre conceptos que conforman el dominio de conocimiento de una ontología*. La idea que subyace en las interpretaciones que vamos a estudiar es la de que los conceptos son susceptibles de ser representados por regiones espaciales. Por tanto, es posible utilizar RCC, y las extensiones estudiadas en los capítulos anteriores, como soporte semántico para razonar sobre la representación espacial de dichas ontologías. De igual forma, el resultado de aplicar RCC para razonar sobre la representación espacial puede traducirse de nuevo a conocimiento ontológico.

En primer lugar definimos la *interpretación fuerte* de RCC como *metaontología*²⁴. Los conceptos en esta metaontología juegan el papel que antes representaban las regiones espaciales. Dos conceptos de una BC estarán conectados respecto tal base si existen elementos comunes a los dos conceptos en algún modelo, aunque no necesariamente se concluye tal hecho de la base, ni tiene que existir un subconcepto común. Es decir, si Σ es una BC en DL,

$$C_{\Sigma}(C_1, C_2) \iff \Sigma \not\models C_1 \sqcap C_2 = \perp$$

Para analizar tal conexión hacemos uso de tipos²⁵. El primer resultado destacado es el teorema que establece que la interpretación fuerte no discrimina RCC8, es decir, no contempla tangencialidad ya que conexión equivale a solapamiento. Por tanto, dos conceptos no pueden estar externamente conectados; tampoco, en caso de que uno sea parte propia del otro, podemos distinguir si lo es de manera tangencial o no. La prueba se obtiene estudiando la traducción a tipos de cada una de las relaciones de RCC, y se basa, fundamentalmente, en que el espacio topológico formado por los tipos de una teoría consistente es un espacio totalmente desconexo, pues los tipos (que son una subbase de la topología) son abiertos y cerrados.

Una desventaja de esta interpretación es que, cuanto más expresivas son las lógicas descriptivas a utilizar y dado que la conexión entre conceptos que establecemos está basada en la implicación lógica, la complejidad algorítmica puede ser muy alta. Por tanto, dicha interpretación no es usable en la práctica

²⁴Es decir, como una ontología sobre las relaciones entre conceptos de las ontologías.

²⁵En su acepción lógica.

en algunas lógicas. Introducimos otra interpretación que se basa en la directa modelización espacial, que denominamos *interpretación débil*, que en la práctica es más útil. Mediante la interpretación débil asociamos a cada BC un problema de satisfacción de restricciones (PSR) de forma que los modelos topológicos solución de tal problema representen modelos de la base en cuestión. En primer lugar, se construye, por casos, el grafo asociado al PSR. Definimos el concepto de modelo espacial de una BC como la interpretación en un espacio topológico (con ciertas propiedades) de los conceptos e individuos de la BC. Las interpretaciones (de conceptos e individuos) serán, respectivamente, abiertos regulares y puntos en el espacio topológico²⁶. Se demuestra que la interpretación débil es equiconsistente con la BC original.

Por último se determina la relación existente entre las interpretaciones fuerte y débil. Esta relación puede expresarse de la siguiente forma: dos conceptos están conectados según la interpretación fuerte si y sólo si existe un modelo espacial (proporcionado bajo la interpretación débil) en el que los abiertos regulares correspondientes a esos conceptos tienen intersección no vacía. De esta forma se traduce la relación de conexión entre conceptos a conexión entre regiones espaciales.

Para finalizar, elaboramos la base lógica sobre la que se sustenta la *interpretación con pulsación de ontologías*. Se basa en la definición de una aplicación σ que a cada concepto de una BC denotada por Σ en DL le asocia una fórmula de DL. A partir de esa aplicación, definimos la Σ -conexión bajo σ en base a la Σ -conexión de la interpretación fuerte. Esta definición puede extenderse al resto de las relaciones de RCC a partir de los axiomas definidores.

Capítulo 6: Estudio de anomalías conceptuales en BC escritas en DL

Previamente a la manipulación de una BC, es necesario analizar la calidad de su contenido: la existencia de datos erróneos, la incoherencia que pueda derivarse de los mismos, etc. En definitiva, de ese análisis depende la confianza en los resultados obtenidos al utilizar cualquier aplicación sobre la BC. En [42], se presenta una clasificación muy completa. Inicialmente, se definen los datos ilícitos como aquellos de los que el usuario (o la aplicación en cuestión) obtiene resultados erróneos, o bien no es posible derivar un resultado, debido a problemas inherentes a los datos en sí. Estos problemas incluyen los generados por entrada y salida de datos erróneos (falsos) así como por fallos en la transmisión y procesamiento de los mismos. Esta clasificación debe ser revisada para las BC en la WS debido a diversos factores, entre los que destacamos el papel de las ontologías.

²⁶Aunque esta definición es demasiado general, sabemos que es posible obtener una representación espacial formada por regiones poligonales en \mathbb{R}^2 [59].

En este capítulo describimos el problema de las anomalías en BC escritas en DL, especialmente en ontologías, las cuales, debido al carácter dinámico de la WS, pueden estar en revisión permanente. Hacemos un breve repaso de la problemática de la revisión ontológica, la actualización de ontologías y la incompletitud, tanto de carácter expresivo como lógico. Adoptamos la idea de *noción* como concepto que no aparece explícitamente en la componente intensional (terminológica) de una BC o en una ontología provisional. Utilizando la base lógica que proporcionamos al finalizar el capítulo anterior, a través de la interpretación fuerte y la idea de pulsación, establecemos el teorema que describe la tabla de posiciones relativas de dos conceptos y sus nociones (como pulsaciones de los mismos). Encontramos posiciones potencialmente conflictivas que se señalan y que nos indican la presencia de posibles anomalías²⁷.

Obtenemos una clasificación de las anomalías de carácter *estático*, es decir, aquellas que no aparecen por experimentación o durante la ejecución de servicios de razonamiento. Distinguiremos las anomalías que denominamos de *tipo léxico-lógico*: entenderemos que dos conceptos tienen el mismo *significante* cuando la BC sea incapaz de distinguirlos (el mismo nombre pero posiblemente diferente significado). De manera análoga, el hecho de que dos conceptos tengan la misma pulsación indicará que tienen el mismo significado (y, posiblemente, diferente nombre). Esto da lugar al estudio de la sinonimia y la polisemia de tipo léxico-lógico como posibles anomalías en la ontología. El caso de la monosemia lógica no se considera anomalía, muy al contrario, representa el caso ideal, el concepto que coincide con su pulsación está totalmente definido.

El caso de la antonimia léxico-lógica sí es, en cambio, un caso de inconsistencia de la ontología (un caso de anomalía por inconsistencia de tipo léxico-lógico). Dos alternativas al uso de nociones en dicho estudio son la utilización de la composición de relaciones mereotopológicas y la composición de proyecciones reticulares para detectar anomalías. Resulta más fácil hacer inferencias espaciales -que luego se traducirán a la BC- que inferencias en la teoría, de ahí que en la reparación, más adelante, de las anomalías detectadas utilicemos las interpretaciones espaciales basadas en RCC8 ya estudiadas.

Por último, presentamos los principios generales para la utilización de un DAT de propósito general (en nuestro caso, OTTER) para la detección de anomalías. El método de resolución que proponemos está basado en el análisis de los argumentos generados por el ARS. En primer lugar particularizamos la

²⁷Hay que hacer notar que tales anomalías deben ser aceptadas como tales por el usuario antes de ser corregidas. Debido a que la confección de una ontología es una tarea potencialmente inconclusa, algunas de las deficiencias son corregidas en el proceso natural de desarrollo y evolución de la ontología, y por tanto, aún siendo una anomalía, son aceptadas. Y otras simplemente no lo son, pues representan propiedades específicas del dominio representado. En cambio, toda anomalía que provoque inconsistencia debe ser reparada.

noción de argumento para OTTER, que denotamos 0-argumento, y definimos la longitud de un 0-argumento.

Se definen las nociones de *grafo de un modelo* de una teoría y *el mundo de un conjunto de fórmulas* así como su entorno cognitivo, para llegar a la definición de *grado de credibilidad de un argumento*, parámetro que servirá para estimar la robustez del mismo.

PARTE III: RESOLUCIÓN DE ANOMALÍAS SOPORTADAS POR LAS INTERPRETACIONES MEREOTOPOLÓGICAS

Capítulo 7: Métodos para la depuración de ontologías

Este capítulo está orientado a la consecución de dos objetivos. Por una parte, el de traducir el análisis de la BC al estudio de un refinamiento gráfico de la misma. Esto se realiza usando como herramienta el cálculo RCC para el razonamiento espacial. La idea es el *análisis local* de las ontologías provisionales analizando el argumento generado. Debido a su tamaño relativamente pequeño (en comparación con la BC completa), es más fácil de representar espacialmente. El segundo objetivo consiste en diseñar un método práctico de reparación de argumentos mediante razonamiento espacial.

Con respecto al primer objetivo, proporcionamos un método para reparar posibles anomalías haciendo uso del estudio realizado en el capítulo anterior de las posiciones relativas de pares de conceptos bajo la interpretación vaga definida en 5.4, en la que un concepto se representa por las dos subregiones concepto/noción. Detectamos las anomalías en las posiciones relativas y definimos el procedimiento, por casos, para repararlas (cuando el usuario acepte que tal configuración constituye una anomalía real). Los arreglos consisten, por ejemplo, en insertar constantes de manera que la relación entre los dos conceptos que motivan la anomalía a reparar, coincida con la relación entre las nociones respectivas. El método se describe con detalle y se ilustra con ejemplos. Los resultados se presentan en una tabla que contiene las posiciones conflictivas, la posición final tras el arreglo (en cada caso particular), el tipo de arreglo aplicado y, por último, el subconjunto de figuras al que corresponde cada arreglo, dentro de las 46 posibilidades descritas en el cuadro 6.1. Este primer método resuelve de forma sistemática anomalías que se detectan entre pares de conceptos.

En segundo lugar, diseñamos un ciclo de limpieza para BC en DL, que consiste básicamente en cuatro pasos:

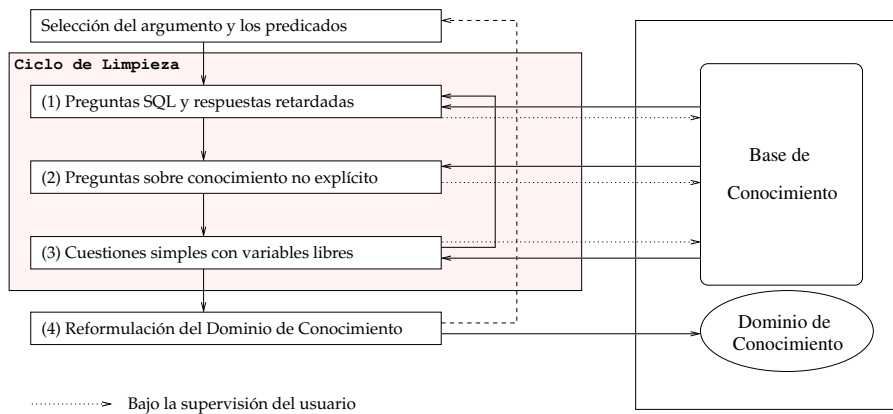
1. En primer lugar, se produce un PSR en RCC8 (o RCC5)²⁸, mediante una traducción cognitivamente adecuada de la Tbox a fórmulas de RCC. El problema se resuelve, obteniendo un escenario consistente, representado en \mathbb{R}^2 , y se añaden los hechos de la Abox.
2. El usuario ejecuta movimientos en la representación espacial (que tienen su correspondencia en proyecciones reticulares o movimientos topológicos). Los reticulares representan refinamientos de las relaciones entre pares de conceptos y los topológicos conllevan la sustitución de una relación por otra, incompatible con la primera pero *cognitivamente cercana* a ella. Es aconsejable que el usuario posteriormente considere que ya dispone de una representación en RCC8 o RCC5. Esto restringe el conjunto de nociones en la ontología provisional.
3. Se aplica una traducción de las fórmulas de RCC a DL. En la BC nueva, han cambiado algunas relaciones y se pueden haber inducido algunos conceptos nuevos.
4. Por último, el usuario interpreta o descarta los nuevos conceptos o elementos que la traducción anterior haya podido inducir.

El ciclo cuyo diseño presentamos está basado en las herramientas estudiadas a lo largo de esta memoria. Como resultado final, se demuestra que las operaciones en que está basado preservan la consistencia de la BC.

Capítulo 8: Depuración de BC asociadas a ontologías complejas asistida por un DAT

En este capítulo, se aborda la limpieza de anomalías desde una nueva perspectiva. Realizamos el estudio mediante argumentos haciendo uso de los fundamentos lógicos que se introducen en la sección correspondiente de este capítulo. Se propone el diseño de un agente, esquematizado en la figura 1, que realiza la detección de inconsistencias mediante la comparación de la información inferida o deducida por el agente con el contenido de la ontología. Así mismo, se propone un ciclo de limpieza asistido por un SRA, concretamente, por un demostrador automático de teoremas. En este caso, suponemos que trabajamos con ontologías consistentes, aunque no es una condición necesaria, de manera que las anomalías provienen mayormente de la componente extensional (la Abox). Presentamos un diagrama que resume el esquema del proceso.

²⁸Es necesario aplicar la proyección reticular para obtener el PSR, pues, para evitar inicialmente refinamientos innecesarios, se utilizarán \hat{P} y \hat{O} (además de ϵ) para obtener el PSR. Refinamientos posteriores del PSR son tareas del usuario, una vez resuelto dicho PSR.



La utilidad del ciclo descrito se ilustra con el caso concreto de una base de conocimiento en RCC. Para ello introducimos la traducción de los axiomas de RCC a DL, comprobando que se utilizan constructores complejos, lo cual confirma el hecho de que RCC es una teoría muy expresiva para el razonamiento clásico en DL.

En la siguiente sección se realiza la interpretación espacial de los argumentos. Comenzamos definiendo el nivel de compacidad, que será un parámetro de gran utilidad. Establecemos las condiciones sintácticas para la consistencia de una BC respecto a un modelo topológico y relacionamos el grado de credibilidad de un argumento con la consistencia del mismo, mediante el teorema y corolario correspondientes.

Finalmente, presentamos un caso de estudio, describiendo una BC como ejemplo. Se presentan los resultados obtenidos en algunos experimentos realizados sobre la BC. Las fichas de tales experimentos, contienen, entre otros datos, el análisis de los resultados obtenidos por el demostrador a las preguntas del ciclo de limpieza, su número y tipo según una clasificación introducida previamente.

Capítulo 9: Conclusiones y líneas de trabajo futuro

Este capítulo presenta las conclusiones sobre el trabajo expuesto y señala las diferentes líneas de investigación que pueden guiar los siguientes avances.

Esquema general de la tesis

En la figura de la página siguiente se describe el diagrama de dependencia entre las distintas partes de esta memoria, así como el tipo de dependencia.

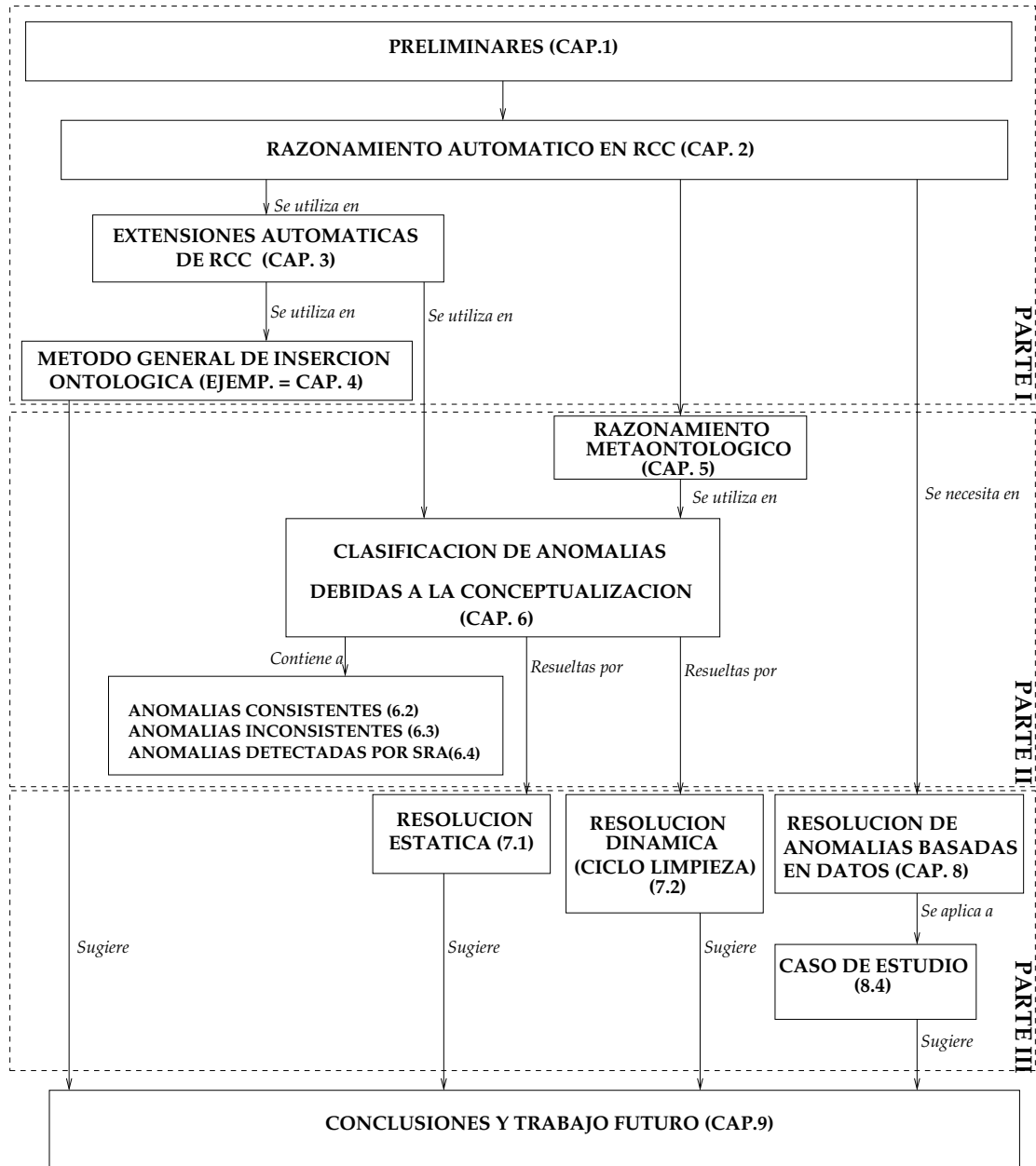


Figura 5: Esquema general de la tesis

Capítulo 1

Preliminares

En este capítulo se introducen las diversas herramientas y teorías que se utilizarán a lo largo de toda la memoria. Concretamente, trataremos la teoría RCC, cálculo para razonamiento espacial cualitativo; resumimos la teoría de retículos e incorporamos a la misma la definición de una nueva relación de extensionalidad de los mismos. Dicha teoría nos servirá más adelante para estudiar posibles extensiones de RCC; haremos también un repaso de la lógica descriptiva, como instrumento para la representación del conocimiento dotado de una componente semántica que lo distingue de otros mecanismos de representación. También describimos los sistemas de razonamiento automático que utilizaremos como asistentes a lo largo de esta memoria, OTTER y MACE4. El primero es un demostrador automático de teoremas que se complementa con el segundo, un buscador de modelos asociado. Se dedica una sección a la topología, ideas generales y definiciones que será útil tener en cuenta en adelante, ya que la topología es la herramienta que permite desarrollar un cálculo para razonamiento espacial cualitativo de manera general, no vinculado a un espacio concreto. Por último, recordamos algunos conceptos relacionados con los problemas de satisfacción de restricciones.

1.1. Retículos: Definición y resultados

Hay dos maneras de introducir la definición de retículo, la primera, de manera algebraica, y la segunda, basada en la noción de orden.

Definición. 1.1.1 *Un retículo es una terna (L, \vee, \wedge) , donde L es un conjunto y \vee, \wedge son dos operaciones binarias definidas sobre L que cumplen las siguientes propiedades, para cualesquiera $a, b, c \in L$:*

L1: Conmutativas: $a \vee b = b \vee a$, $a \wedge b = b \wedge a$.

L2: Asociativas: $a, b, c \in L, a \vee (b \vee c) = (a \vee b) \vee c, a \wedge (b \wedge c) = (a \wedge b) \wedge c.$

L3: Leyes de Idempotencia: $\forall a \in L a \vee a = a, \forall a \in L a \wedge a = a.$

L4: Leyes de absorción: $\forall a, b \in L a \vee (a \wedge b) = a, \forall a, b \in L a \wedge (a \vee b) = a.$

A continuación introducimos la noción de orden parcial en un conjunto.

Definición. 1.1.2 Una relación binaria \leq definida en un conjunto A es un **orden parcial** en A si se tienen las siguientes condiciones:

1. $a \leq a$
2. $a \leq b$ y $b \leq a$ implican $a = b$
3. $a \leq b$ y $b \leq c$ implican $a \leq c$

Si, además, para $a, b \in A$ se tiene:

4. $a \leq b$ o bien $b \leq a$
entonces \leq es un **orden total** en A . Un conjunto no vacío con un orden parcial/total es un **conjunto parcial/totalmente ordenado**.

La existencia de orden en un conjunto induce a pensar en la posible acotación del mismo.

Definición. 1.1.3 Sea A un subconjunto de un conjunto parcialmente ordenado P . Un elemento p en P es una **cota superior** para A si todo a de A verifica $a \leq p$. Además, p será el **supremo** de A si es la menor de las cotas superiores en A . Análogamente, se define la noción de **cota inferior** para A y la mayor cota inferior o **ínfimo** de A . Denotaremos por $\sup(A)$ e $\inf(A)$ al supremo y el ínfimo de A respectivamente. Por último, para a, b de P , diremos que b **está sobre** a , y se denota por $a \prec b$ si $a \leq b$ y además todo c que verifique $a \leq c \leq b$ debe ser $a = c$, o bien $b = c$.

Veamos cómo es posible dotar de estructura de retículo a un conjunto parcialmente ordenado.

Definición. 1.1.4 Un conjunto parcialmente ordenado L tal que para cualesquiera $a, b \in L$ existen $\sup\{a, b\}$ y $\inf\{a, b\}$, es un **retículo**.

Teorema. 1.1.5 Las dos definiciones de retículo que hemos presentado son equivalentes.

Demostración: Si L es un retículo según la primera definición, bastará definir \leq como

$$a \leq b \iff a = a \wedge b$$

para que cumpla la segunda definición. En caso de que L sea un retículo según la segunda definición, entonces definimos las operaciones \wedge y \vee como

$$a \vee b = \sup\{a, b\} \text{ y } a \wedge b = \inf\{a, b\}$$

y así se cumplirá la primera de las definiciones, véase [14]. ■

Los conjuntos parcialmente ordenados tienen la característica especial de que podemos representarlos gráficamente. A cada conjunto finito parcialmente ordenado P se le asocia un diagrama llamado **diagrama de Hasse**. Cada elemento de P se representa por un círculo \circ . Si b está sobre a , dibujamos el círculo que representa a b encima del que representa a a y unimos ambos círculos con una línea. A partir de este diagrama, capturamos la relación de orden \leq , de manera que se tiene $a < b$ si y sólo si existe una secuencia finita c_1, \dots, c_n de elementos de P tales que $a = c_1 < c_2 < \dots < c_{n-1} < c_n = b$.

Definición. 1.1.6 Dos retículos L_1 y L_2 son **retículos isomorfos** si existe una biyección α , de L_1 en L_2 tal que para dos elementos a, b cualesquiera de L_1 se tienen las dos condiciones siguientes:

- $\alpha(a \vee b) = \alpha(a) \vee \alpha(b)$
- $\alpha(a \wedge b) = \alpha(a) \wedge \alpha(b)$

Se dirá que α es un isomorfismo.

Podemos reformular la definición de isomorfismo en función de las respectivas relaciones de orden.

Definición. 1.1.7 Dados P_1 y P_2 dos conjuntos parcialmente ordenados y una aplicación α de P_1 en P_2 , se dice que α **conserva el orden** si $\alpha(a) \leq \alpha(b)$ en P_2 para cualesquiera a, b de P_1 tales que $a \leq b$.

En general, por simplificar, trataremos de identificar directamente los retículos isomorfos.

Teorema. 1.1.8 (Véase [14]) Dos retículos L_1, L_2 son isomorfos si y sólo si existe una biyección α de L_1 en L_2 tal que α y α^{-1} conservan el orden.

Los retículos más estudiados son los distributivos y los modulares.

Definición. 1.1.9 Un *retículo distributivo* es un retículo que verifica una de las siguientes leyes distributivas:

$$D1: \forall a, b, c \in L \quad a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$$

$$D2: \forall a, b, c \in L \quad a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$$

Teorema. 1.1.10 Véase [14]. En un retículo L , las condiciones D1 y D2 son equivalentes.

Definición. 1.1.11 Si L es un retículo y L' un subconjunto no vacío de L tal que para dos elementos cualesquiera $a, b \in L'$, se tiene $(a \wedge b) \in L'$ y también $(a \vee b) \in L'$, siendo \wedge, \vee son las operaciones de L restringidas a L' , entonces decimos que L' es un *subretículo* de L .

Definición. 1.1.12 Un retículo L_1 se puede *incorporar* a un retículo L_2 si existe un subretículo de L_2 isomorfo a L_1 ; en tal caso diremos también que L_2 contiene una copia de L_1 como subretículo.

Resulta interesante, en esta memoria, considerar un tipo especial de relación de extensión entre retículos, la de *extensión conservativa* con respecto a un conjunto de ecuaciones.

Definición. 1.1.13 Sean L, L' dos retículos. Sea \mathcal{E} un conjunto de ecuaciones con constantes denotando elementos de L , tal que $L \models \mathcal{E}$. Diremos que L' es una *extensión \mathcal{E} -conservativa* de L , $L \subset_{\mathcal{E}} L'$, si existe una función $i: L \rightarrow L'$ inyectiva tal que para toda fórmula $\varphi(a_1, \dots, a_n) \in \mathcal{E}$, $L' \models \varphi(i(a_1), \dots, i(a_n))$ y $(i(L), \leq) \subseteq (L', \leq')$.

Nota. 1.1.14 La propiedad de ser subestructura (con respecto al orden) es un caso particular de ser extensión conservativa. Una expresión $a \leq b$, puede expresarse mediante la fórmula $a \wedge b = a$.

Nota. 1.1.15 Sean L_1, L_2 dos retículos:

- Si L_1 se puede incorporar a L_2 , entonces L_2 es una extensión conservativa con respecto a E_{L_1} , el conjunto de ecuaciones cerradas válidas en L_1 .
- La implicación contraria no es cierta en general: Consideremos el conjunto $\mathcal{E} = \{a \wedge b = a, a \wedge c = \perp\}$ y el contraejemplo dado en la figura 1.1. El segundo retículo es una extensión \mathcal{E} -conservativa del primero, pero éste no puede incorporarse al segundo.

Definición. 1.1.16 Un *retículo modular* L es aquel que verifica la siguiente condición, llamada Ley Modular:

$$\forall a, b, c \in L \quad [(a \wedge b) \vee (b \wedge c) = b \wedge ((a \wedge b) \vee c)]$$

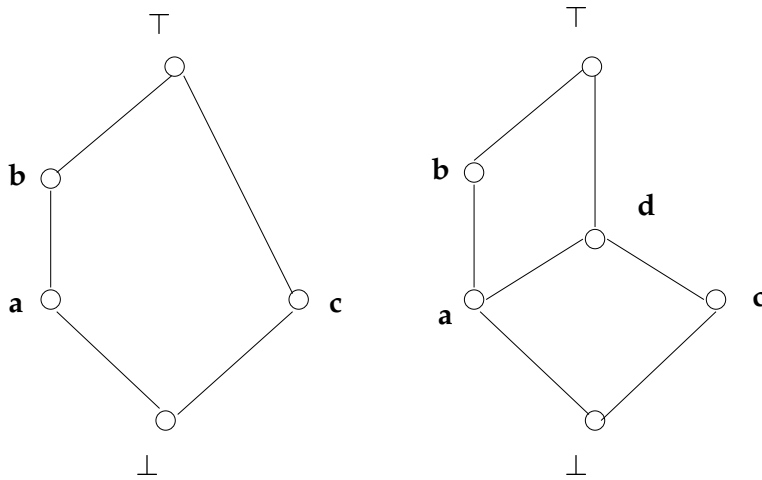


Figura 1.1: L_1 no puede incorporarse a L_2 , una extensión \mathcal{E} -conservativa

Teorema. 1.1.17 (Véase [14]) *Todo retículo distributivo es modular.*

Los dos teoremas siguientes presentan una caracterización de los retículos modulares y distributivos en función de la existencia de ciertos retículos de cinco elementos, llamados M_5 y N_5 , representados en la figura 1.2. Ninguno de ellos es distributivo. Además M_5 es modular pero N_5 no lo es.

Teorema. 1.1.18 (Dedekind). *L es un retículo no-modular si y solo si N_5 puede incorporarse a L .*

Teorema. 1.1.19 (Birkhoff). *L es un retículo no-distributivo si y solo si es posible incorporar N_5 ó M_5 a L .*

Los retículos distributivos y modulares poseen interesantes propiedades y son estructuras básicas para el estudio de retículos. Sin embargo, como veremos en la proposición 2.2.4, los retículos con los que vamos a trabajar son no modulares y no distributivos.

1.2. Lógicas de la descripción

Las lógicas de la descripción (DL) [4] son una familia de lenguajes formales con una semántica claramente especificada, junto con unos mecanismos de inferencia para clasificar el conocimiento. El origen de la investigación en este campo fue el propósito de identificar los fragmentos de la lógica de primer orden y las extensiones capaces de capturar las características necesarias para representar conceptos estructurados, siendo a la vez posible diseñar algoritmos de razonamiento eficientes.

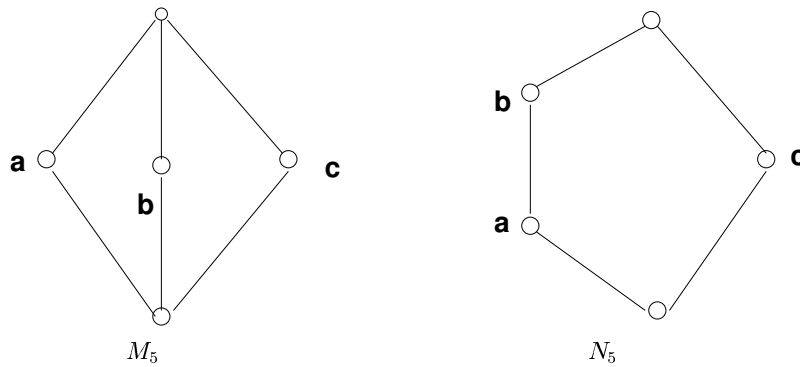


Figura 1.2: Los retículos M_5 y N_5 .

Al ser técnicamente DL un subconjunto de la lógica de primer orden, posee una semántica inherente, a diferencia de otros formalismos anteriormente usados para la representación del conocimiento. Existen cálculos manejables para diversos sistemas DL, con complejidad computacional conocida (cf. [18]).

Un sistema de representación de conocimiento basado en DL está constituido por un lenguaje de conceptos, que puede verse como un conjunto de operadores para denotar las clases y las relaciones entre clases [18]. Los conceptos se usan para representar clases como conjuntos de individuos y los roles son relaciones binarias usadas para especificar sus propiedades o atributos. A su vez, los conceptos se denotan con expresiones formadas mediante unos operadores especiales.

1.2.1. Sintaxis y Semántica de los lenguajes de conceptos

Como hemos mencionado, un lenguaje de conceptos está compuesto por los símbolos de *nombres de conceptos*, *nombres de roles* y *nombres de individuos* (o simplemente *individuos*). Además se incluye un número de operadores que permiten la formación de *expresiones de conceptos y roles*. El conjunto de operadores básico aparece listado en la figura 1.3. Como norma general, los nombres de conceptos se denotarán por las letras A, B, \dots , los nombres de roles por P, Q, \dots y utilizaremos las letras a, d (tal vez con subíndices) para denotar individuos.

Un lenguaje \mathcal{L} se clasificará por su conjunto de operadores. A los conceptos (resp. roles) obtenidos usando los operadores de \mathcal{L} los llamaremos \mathcal{L} -conceptos (resp. \mathcal{L} -roles). Como los operadores caracterizan al lenguaje de conceptos en sí, nombramos dichos lenguajes utilizando los identificadores de los operadores. Tomamos como base el lenguaje \mathcal{FL}^- , que incluye cuantificación universal, conjunción y cuantificación existencial general:

$$\mathcal{FL}^- = \mathcal{L} + \sqcap + \forall R.C + \exists R.T$$

Nombre del operador	Id.	Sintaxis	Semántica
nombre de concepto		A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
top		\top	$\Delta^{\mathcal{I}}$
bottom		\perp	\emptyset
conjunción de conceptos		$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
disyunción de conceptos	(U)	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
negación de concepto	(C)	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
cuantificación universal		$\forall R.C$	$\{d_1 \mid \forall d_2 : (d_1, d_2) \in R^{\mathcal{I}} \rightarrow d_2 \in C^{\mathcal{I}}\}$
cuantificación existencial	(E)	$\exists R.C$	$\{d_1 \mid \exists d_2 : (d_1, d_2) \in R^{\mathcal{I}} \wedge d_2 \in C^{\mathcal{I}}\}$
restricciones de número	(N)	$(\geq n R)$	$\{d_1 \mid \#\{d_2 \mid (d_1, d_2) \in R^{\mathcal{I}}\} \geq n\}$
		$(\leq n R)$	$\{d_1 \mid \#\{d_2 \mid (d_1, d_2) \in R^{\mathcal{I}}\} \leq n\}$
colección de individuos	(O)	$\{a_1, \dots, a_n\}$	$\{a_1^{\mathcal{I}}, \dots, a_n^{\mathcal{I}}\}$
nombre de rol		P	$P^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
conjunción de roles	(R)	$Q \sqcap R$	$Q^{\mathcal{I}} \cap R^{\mathcal{I}}$
negación de rol		$\neg R$	$\Delta^{\mathcal{I}} \setminus R^{\mathcal{I}}$
composición de roles		$P \circ R$	$\{(d_1, d_2) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid \exists d : (d_1, d) \in P^{\mathcal{I}} \wedge (d, d_2) \in R^{\mathcal{I}}\}$
inverso de rol		P^{-}	$\{(d_1, d_2) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (d_2, d_1) \in P^{\mathcal{I}}\}$

Figura 1.3: Sintaxis y semántica para operadores

Añadiendo a \mathcal{FL}^- los constructores \top , \perp y la negación de conceptos atómicos $\neg A$, obtenemos el lenguaje \mathcal{AL} . Para nombrar las extensiones de ambos lenguajes \mathcal{FL}^- y \mathcal{AL} , basta añadir a su nombre la lista de operadores deseados. De esta forma, $\mathcal{FL}^-[\mathcal{E}][\mathcal{U}][\mathcal{C}]$ será el lenguaje de conceptos que resulte de extender \mathcal{FL}^- con los operadores de cuantificación existencial, disyunción de conceptos y negación de conceptos. En la figura 1.3 podemos ver los identificadores de los operadores.

Definición. 1.2.1 Una *interpretación* \mathcal{I} es un par $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, donde $\Delta^{\mathcal{I}}$ es un conjunto no vacío (dominio) y $\cdot^{\mathcal{I}}$ una función (de interpretación) que aplica cada concepto en un subconjunto de $\Delta^{\mathcal{I}}$, cada rol en un subconjunto de $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ y cada individuo en un elemento de $\Delta^{\mathcal{I}}$.

Las expresiones de conceptos y roles se interpretan según la semántica dada en la Fig. 1.3, en la columna derecha.

Como ya mencionamos, los conceptos se corresponden con predicados en LPO y, a partir de éstos, se pueden interpretar los diversos constructores como fórmulas LPO. En la Fig. 1.4 se describen las fórmulas correspondientes a los operadores para conceptos y roles de la Fig. 1.3.

Definición. 1.2.2 Sea \mathcal{I} una interpretación:

1. Una interpretación \mathcal{I} es un **modelo** para un concepto C si $C^{\mathcal{I}}$ es no vacío.

Expresión	Fórmula LPO
\top	Verdadero
\perp	Falso
$C \sqcap D$	$C(x) \wedge D(x)$
$C \sqcup D$	$C(x) \vee D(x)$
$\neg C$	$\neg C(x)$
$\forall R.C$	$\forall y(R(x, y) \rightarrow C(y))$
$\exists R.C$	$\exists y(R(x, y) \wedge C(y))$
$(\geq n R)$	$\exists y_1, \dots, y_n \left(\bigwedge_{1 \leq i \leq n} \left(R(x, y_i) \wedge \bigwedge_{1 \leq i < j \leq n} y_i \neq y_j \right) \right)$
$(\leq n R)$	$\forall y_1, \dots, y_{n+1} \left(\left(\bigwedge_{1 \leq i \leq n+1} R(x, y_i) \right) \rightarrow \bigvee_{1 \leq i < j \leq n+1} y_i = y_j \right)$
$Q \sqcap R$	$Q(x, y) \wedge R(x, y)$
$\neg R$	$\neg R(x, y)$
$P \circ R$	$\exists z(P(x, z) \wedge R(z, y))$
P^-	$P^-(x, y)$

Figura 1.4: Traducción a Fórmulas de Primer Orden

2. Un concepto es **satisfactible** si tiene un modelo. Se dice que es **insatisfactible** en caso contrario.
3. Un concepto D **subsume** a otro C si $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ para toda interpretación \mathcal{I} . Y ambos conceptos serán **equivalentes**, $C \equiv D$, si $C^{\mathcal{I}} = D^{\mathcal{I}}$ para toda interpretación \mathcal{I} .

1.2.2. Bases de conocimiento

Definición. 1.2.3 Dado un lenguaje de conceptos \mathcal{L} , una \mathcal{L} -fórmula es de una de las formas siguientes:

$$\begin{aligned} C(a) & \text{ Pertenencia a un concepto} \\ R(a, b) & \text{ Pertenencia a un rol} \end{aligned}$$

donde C es un \mathcal{L} -concepto, R un \mathcal{L} -rol y a, b individuos.

Ya hemos visto cómo podemos formar descripciones de conceptos para describir clases de objetos. DL contiene un formalismo terminológico y asercional. Los *axiomas terminológicos*, en su forma más simple, se utilizan para introducir nombres de descripciones complejas, para establecer cómo se relacionan conceptos y roles entre sí. En el caso más general, los *axiomas terminológicos* tienen las formas siguientes [18]:

1. $A \sqsubseteq C$, Especificación de concepto primitivo
2. $A \equiv C$, Definición de concepto
3. $C \sqsubseteq D$, Inclusión de concepto
4. $C \equiv D$, Equivalencia de conceptos

donde A es un nombre de concepto y C, D son \mathcal{L} -conceptos. Un concepto que aparezca en el lado izquierdo de una definición de concepto se llamará *concepto definido*. Si aparece en el lado izquierdo de una especificación del tipo (1) será un *concepto primitivo*. Si no aparece a la izquierda en ningún axioma terminológico, se llamará *concepto atómico*.

El formalismo asercional de DL se usa para establecer propiedades de los individuos. La semántica de los axiomas es la esperada: una interpretación \mathcal{I} *satisface* una inclusión $A \sqsubseteq C$ si $A^{\mathcal{I}} \subseteq C^{\mathcal{I}}$ y *satisface* una equivalencia $A \equiv C$ si $A^{\mathcal{I}} = C^{\mathcal{I}}$.

La semántica correspondiente es la que sigue.

Definición. 1.2.4 Dada \mathcal{I} una interpretación, decimos que \mathcal{I} *satisface* $C(a)$ ($\mathcal{I} \models C(a)$) si $a^{\mathcal{I}} \in C^{\mathcal{I}}$, y que \mathcal{I} *satisface* $R(a, b)$ ($\mathcal{I} \models R(a, b)$) si $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$.

Definición. 1.2.5 Una *base de conocimiento en DL*, Σ , es un par $\langle \mathcal{T}, \mathcal{A} \rangle$, donde \mathcal{A} es un conjunto de propiedades de los individuos (que llamaremos *ABox*) y \mathcal{T} un conjunto de axiomas terminológicos, *TBox*.

Definición. 1.2.6 Dada una base de conocimiento $\Sigma = \langle \mathcal{T}, \mathcal{A} \rangle$, el conjunto de *conceptos elementales* en Σ , $\text{conceptos}(\Sigma)$, es el conjunto de conceptos necesarios para construir cada fórmula de Σ , mediante combinaciones booleanas de las mismas. El conjunto de *individuos* de Σ , $\text{indiv}(\Sigma)$, es el conjunto de nombres de individuos que aparecen en \mathcal{A} .

Si \mathcal{T} es un conjunto de axiomas terminológicos, entonces \mathcal{I} *satisface* \mathcal{T} si y solo si \mathcal{I} *satisface* cada axioma de \mathcal{T} . Si \mathcal{I} *satisface* un axioma (resp. un conjunto de axiomas), diremos que es un *modelo* del axioma (resp. del conjunto de axiomas). Dos axiomas o dos conjuntos de axiomas son *equivalentes* si tienen los mismos modelos.

Definición. 1.2.7 Dado un lenguaje \mathcal{L} y una *base de conocimiento* $\Sigma = \langle \mathcal{T}, \mathcal{A} \rangle$, en \mathcal{L} :

1. Una *interpretación* de Σ es un modelo de Σ si lo es de \mathcal{T} y \mathcal{A} .
2. Una fórmula α es *consecuencia lógica* de Σ , $\Sigma \models \alpha$, si α es *válida* en todo modelo de Σ .

1.2.3. Tareas de razonamiento

Las lógicas descriptivas proporcionan al usuario mecanismos de inferencia para deducir conocimiento implícito a partir del conocimiento representado explícitamente.

Definición. 1.2.8 *Algunos de los servicios de razonamiento básicos de los DL-sistemas son los siguientes:*

1. La **Satisfactibilidad** de conceptos, $(\Sigma \models C \equiv \perp)$, consiste en comprobar si C es satisfactible con respecto a Σ , es decir, comprobar si existe un modelo \mathcal{I} de Σ tal que $C^{\mathcal{I}} \neq \emptyset$.
2. La **Subsunción** $(\Sigma \models C \sqsubseteq D)$, permite comprobar si D subsume a C con respecto a Σ , es decir, si $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ en todo modelo \mathcal{I} de Σ .
3. El algoritmo de **Consistencia**, $(\Sigma \models)$, determina si una base de conocimiento es o no contradictoria, si tiene un modelo.
4. **Instanciación**, $(\Sigma \models C(a))$. Se trata de comprobar si todo modelo de Σ satisface $C(a)$.

Una tarea de razonamiento más compleja es la de **clasificación**, en la que juega un gran papel la subsunción. Consiste en, dado un concepto C y una TBox \mathcal{T} , encontrar el lugar correcto de C en la taxonomía de \mathcal{T} , es decir, encontrar los conceptos más específicos que C con respecto a la relación de subsunción.

Otra tarea, en la que interviene la instanciación, es la de **recuperación**. Consiste en, dada una base de conocimiento Σ y un concepto C , hallar el conjunto $\{a : \Sigma \models C(a)\}$, y se puede obtener iterando instanciación sobre cada individuo de Σ .

Lema. 1.2.9 *En lenguajes que disponen del operador \mathcal{C} (para expresar la negación de conceptos), se verifica que:*

- $\Sigma \models C \sqsubseteq D \iff \Sigma \models C \sqcap \neg D \equiv \perp$
- De forma parecida,

$$\Sigma \models C(a) \iff \Sigma \cup \{\neg C(a)\} \text{ es inconsistente.}$$

Por tanto, la subsunción puede reducirse a satisfactibilidad y la instanciación a consistencia.

1.2.4. Normalización

En general, en bases de conocimiento construidas a mano, muchos conceptos complejos se describen utilizando otros nombres de conceptos más simples. Como método para detectar contradicciones de manera rápida (contradicciones directas como por ejemplo la existente entre $C \sqcap D$ y $\neg C \sqcup \neg D$), podemos transformar todos los conceptos a una forma normal sintáctica que permita compararlos. La normalización, proceso de transformación específico de la Forma Normal Disyuntiva, simplifica la base de conocimiento detectando contradicciones sintácticamente obvias y tautologías. Las funciones de normalización y simplificación son las siguientes [39]:

$$\begin{aligned}
 \text{Norm}(A) &= A, \text{ para } A \text{ concepto atómico} \\
 \text{Norm}(\neg C) &= \text{Simp}(\neg \text{Norm}(C)) \\
 \text{Norm}(C_1 \sqcap \dots \sqcap C_n) &= \text{Simp}(\sqcap(\{\text{Norm}(C_1)\} \cup \dots \cup \{\text{Norm}(C_n)\})) \\
 \text{Norm}(C_1 \sqcup \dots \sqcup C_n) &= \text{Norm}(\neg(\neg C_1 \sqcap \dots \sqcap \neg C_n)) \\
 \text{Simp}(A) &= A, \text{ para } A \text{ concepto atómico} \\
 \text{Simp}(\neg C) &= \begin{cases} \perp & \text{si } C = \top \\ \top & \text{si } C = \perp \\ \text{Simp}(D) & \text{si } C = \neg D \\ \neg C & \text{e.o.c.} \end{cases} \\
 \text{Simp}(\sqcap S) &= \begin{cases} \perp & \text{si } \perp \in S \\ \perp & \text{si } \{C, \neg C\} \subseteq S \\ \top & \text{si } S = \emptyset \\ \text{Simp}(S \setminus \{\top\}) & \text{si } \top \in S \\ \text{Simp}(\sqcap P \cup S \setminus \{\sqcap P\}) & \text{si } \sqcap P \in S \\ \sqcap S & \text{e.o.c.} \end{cases}
 \end{aligned}$$

Estas funciones se pueden ampliar, para lógicas más expresivas, añadiendo las normalizaciones oportunas de los constructores elegidos.

1.2.5. Relación de OWL con DL

OWL¹ es la propuesta estándar de W3C como lenguaje para la representación de ontologías en la Web. En principio, un lenguaje para ontologías requiere una sintaxis y una semántica bien definidas. Además es necesario que el nivel de expresividad se mantenga en equilibrio con la eficiencia del mecanismo de razonamiento. OWL [35] está construido a partir de RDF² y RDFS³, usando la sintaxis XML de RDF, los descriptores de RDF y la mayoría de las primitivas

¹<http://www.w3.org/2001/sw/WebOnt/>

²<http://www.w3.org/RDF/>

³<http://www.w3.org/TR/2000/CR-rdf-schema-20000327/>

de RDFS. Aunque OWL ya es suficientemente rico, se están realizando extensiones que amplían las posibilidades en el aspecto lógico, entre ellas el uso de reglas.

La importancia de la sintaxis es clara cuando se pretende que la información la procesen ordenadores. En el caso de OWL, no es muy relevante que su sintaxis, la de XML, resulte manejable al usuario, ya que éste, en última instancia, lo que utilizará será herramientas para el desarrollo de ontologías, no escribirá directamente en OWL.

Por otro lado, una semántica formal debe describir de manera objetiva el significado del conocimiento y a su vez constituye un prerrequisito para la existencia de un soporte para el razonamiento. Así, si la semántica permite razonar a cerca de la pertenencia a una clase o sobre la consistencia del conocimiento, un soporte para el razonamiento permitirá comprobar la consistencia del razonamiento y de la propia ontología de manera rápida y eficiente. Esto facilita considerablemente el diseño de ontologías extensas, en las que pueden intervenir varios autores, y la integración de ontologías compartidas por diferentes recursos. Usualmente, la vía para conseguir una semántica formal y un soporte para razonar, es traducir el lenguaje de la ontología a un formalismo lógico conocido y utilizar razonadores (demostradores) automáticos ya existentes para ese formalismo. Este es el caso de FACT y RACER para DL.

La imposibilidad de compatibilizar los requerimientos de un lenguaje para ontologías que proporcione un soporte para razonar de manera efectiva y una expresividad apropiada, y que a su vez sea tan potente como el resultado de combinar RDFS con una lógica completa, lleva a W3C a definir OWL como tres sublenguajes diferentes. Estos sublenguajes son OWL *completo* (el cual resulta altamente potente pero igualmente indecidible), OWL *DL*, sublenguaje del primero que restringe el uso de los constructores de OWL y RDF, y, por último, OWL *lite*, cuyo conjunto de constructores es un subconjunto del anterior. Así, la elección de uno de esos lenguajes a la hora de desarrollar una ontología depende de la necesidad, bien de constructores que aumenten la expresividad, bien de mejorar la eficiencia en el razonamiento.

La traducción de algunos de los constructores y axiomas de OWL a DL aparece en las figuras 1.5 y 1.6 respectivamente [5].

Las figuras 1.7 y 1.8 muestran una pequeña ontología de la familia escrita en OWL y su traducción a DL, respectivamente.

Constructor	Sintaxis DL	Ejemplo
intersectionOf	$C_1 \sqcap \dots \sqcap C_n$	Human \sqcap Male
unionOf	$C_1 \sqcup \dots \sqcup C_n$	Doctor \sqcup Lawyer
complementOf	$\neg C$	\neg Male
oneOf	$\{x_1 \dots x_n\}$	{john, mary}
allValuesFrom	$\forall P.C$	\forall hasChild.Doctor
someValuesFrom	$\exists R.C$	\exists hasChild.Lawyer
hasValue	$\exists R.\{x\}$	\exists citizenOf.{USA}
minCardinality	$(\geq n R)$	$(\geq 2$ hasChild)
maxCardinality	$(\leq n R)$	$(\leq 1$ hasChild)
inverseOf	R^-	hasChild $^-$

Figura 1.5: Algunos constructores de OWL

Axiomas	Sintaxis DL	Ejemplo
subclassOf	$C_1 \sqsubseteq C_2$	Human \sqsubseteq Animal \sqcap Biped
equivalentClass	$C_1 \equiv C_2$	Man \equiv Human \sqcap Male
subPropertyOf	$P_1 \sqsubseteq P_2$	hasDaughter \sqsubseteq hasChild
equivalentProperty	$P_1 \equiv P_2$	cost \equiv price
disjointWith	$C_1 \sqsubseteq \neg C_2$	Male $\sqsubseteq \neg$ Female
sameAs	$\{x_1\} \equiv \{x_2\}$	{President – Bush} \equiv {G.Bush}
differentFrom	$\{x_1\} \sqsubseteq \neg\{x_2\}$	{john} $\sqsubseteq \neg$ {peter}
TransitiveProperty	$P \in \mathbf{R}_+$	hasAncestor $^+ \in \mathbf{R}_+$
FunctionalProperty	$\top \sqsubseteq (\leq 1 P)$	$\top \sqsubseteq (\leq 1$ hasMother)
InverseFunctionalProperty	$\top \sqsubseteq (\leq 1 P^-)$	$\top \sqsubseteq (\leq 1$ isMotherOf $^-)$
SymmetricProperty	$P \equiv P^-$	isSiblingOf \equiv isSiblingOf $^-$

Figura 1.6: Algunos tipos de axiomas que permite OWL

```
name "Family"
documentation "Example ontology describing family relationships"
definitions
  slot-def hasChid
    inverse isChildOf

  class-def defined Woman
    subclass-of PersonFemale

  class-def defined Man
    subclass-of Person not Woman

  class-def defined Mother
    subclass-of Woman
    slot-constraint hasChid
    has-value Person

  class-def defined Father
    subclass-of Man
    slot-constraint hasChid
    has-value Person

  class-def defined Parent
    subclass-of or Father Mother

  class-def defined Grandmother
    subclass-of Mother
    slot-constraint hasChid
    has-value Parent

  class-def defined MotherWithManyChildren
    subclass-of Mother
    slot-constraint hasChid
    min-cardinality3

  class-def defined MotherWithoutDaughter
    subclass-of Mother
    slot-constraint hasChid
    value-tipe not Woman
```

Figura 1.7: Ontología de la Familia en OWL

hasChild ⁻	\sqsubseteq	isChildOf
isChildOf ⁻	\sqsubseteq	hasChild
Woman	\equiv	Person \sqcap Female
Man	\equiv	Person \sqcap \neg Woman
Mother	\equiv	Woman \sqcap \exists hasChild.Person
Father	\equiv	Man \sqcap \exists hasChild.Person
Parent	\equiv	Father \sqcup Mother
Grandmother	\equiv	Mother \sqcap \exists hasChild.Parent
MotherWithManyChildren	\equiv	Mother \sqcap ≥ 3 hasChild
MotherWithoutDaughter	\equiv	Mother \sqcap \forall hasChild. \neg Woman

Figura 1.8: Traducción de la ontología de 1.7 a DL

1.3. Formalización en LPO de las Bases de Datos Lógicas

Antes de considerar la formalización de las bases de datos en términos lógicos, es necesario aislar las hipótesis que rigen en la evaluación de las consultas y las restricciones de integridad. Por un lado, esas hipótesis expresan una cierta representación implícita de hechos negativos (por ejemplo, “Ana no es la madre de Isabel”) y, por otro, ayudan a hacer más preciso el universo de referencia al que las consultas se puedan referir. Dichas hipótesis son tres:

1. *Hipótesis del mundo cerrado* (CWA): Esta hipótesis establece que la información que no es verdadera (o bien se desconoce si lo es) en la base de datos, se supone falsa. Por ejemplo, en el caso de bases de datos relacionales, se considera verdadero $\neg R(a_1, \dots, a_n)$ si (a_1, \dots, a_n) no se encuentra en la relación R .
2. *Hipótesis de nombres únicos* (UNA): Establece que nombres diferentes de constantes representan individuos distintos.
3. *Hipótesis de clausura de dominio* (DCA): Consiste en suponer que no existen más individuos que aquellos representados por los nombres de las constantes del lenguaje de la base de datos.

Las respuestas a preguntas del tipo “para todo” o preguntas que lleven negación se obtienen usando las hipótesis anteriores.

Ejemplo. 1.3.1 Consideremos la base de datos siguiente:

Alumno(Alberto)
Alumno(Javier)
Director(Pedro)
Profesor(Luis)

La pregunta *¿Quién no es un alumno?* tiene como respuesta $\{Luis, Pedro\}$. La hipótesis de clausura de dominio indica que el conjunto de individuos a considerar se reduce a $\{Alberto, Javier, Pedro, Luis\}$. De acuerdo con el principio de nombres únicos, tenemos lo siguiente: $Luis \neq Alberto, Luis \neq Javier$. En consecuencia, $Luis \notin Alumno$, lo cual, según la hipótesis de mundo cerrado, nos lleva a $\neg Alumno(Luis)$. De manera análoga se obtiene el segundo elemento de la respuesta: $\neg Alumno(Pedro)$.

Hay una manera de evitar la hipótesis de clausura de dominio. Consiste en aceptar como preguntas únicamente expresiones que tienen restringido su propio dominio. Es el caso de la clase de las llamadas fórmulas lógicas definidas o de rango restringido.

Aunque el proceso de evaluación de las consultas a una base de datos funciona implícitamente bajo esas tres hipótesis, éstas se hacen explícitas mediante la formalización de las bases de datos. Desde el punto de vista de la lógica, una base de datos puede verse de dos formas; como una *interpretación* (de una teoría de primer orden) y como una *teoría* de primer orden. Cuando de miran desde el punto de vista de una interpretación, las consultas y las restricciones de integridad son fórmulas que hay que evaluar en la interpretación usando la definición semántica de verdad. Como teoría, consultas y restricciones son teoremas que hay que probar. Los dos puntos de vista mencionados formalizan respectivamente las bases de datos convencionales y las deductivas.

Pasemos a describir la visión de una base de datos BD como base de datos deductiva. Consiste en construir una teoría T que admite a BD como único modelo. Por tanto, dada una fórmula ϕ en L, el lenguaje de T, se tendrá $T \vdash \phi$ si y sólo si ϕ es verdadera en DB. Los axiomas de T serán los siguientes:

1. *Asertos*. Para cada relación R en BD y cada tupla $(a_1, \dots, a_p) \in R$, añadimos el axioma: $R(a_1, \dots, a_p) \in T$.
2. *Axiomas de particularización*.
 - *Axiomas de completación*. Hay uno para cada relación R de BD. Si $(a_1^1, \dots, a_p^1), \dots, (a_1^n, \dots, a_p^n)$ son todas las p -tuplas de R , el axioma correspondiente se escribe: $\forall x_1 \dots \forall x_p (R(x_1, \dots, x_p) \rightarrow (x_1 = a_1^1 \wedge \dots \wedge x_p = a_p^1) \vee \dots \vee (x_1 = a_1^n \wedge \dots \wedge x_p = a_p^n))$
 - *Axiomas de nombres únicos*. Si a_1, \dots, a_q son todos los individuos en BD, los axiomas de nombres únicos son:

$$(a_1 \neq a_2), \dots, (a_1 \neq a_q), (a_2 \neq a_3), \dots, (a_{q-1} \neq a_q)$$

- *Axioma de clausura de dominio*. En este caso, es:

$$\forall x ((x = a_1) \vee (x = a_2) \vee \dots \vee (x = a_q))$$

3. *Axiomas de igualdad* Éstos son necesarios puesto que en los axiomas de particularización aparece el predicado de igualdad. Estos axiomas expresan las propiedades usuales de la igualdad.

- *Reflexividad:* $\forall x(x = x)$
- *Simetría:* $\forall x\forall y((x = y) \rightarrow (y = x))$
- *Transitividad:* $\forall x\forall y\forall z((x = y) \wedge (y = z) \rightarrow (x = z))$
- *Principio de sustitución de términos iguales:*

$$\forall x_1 \dots \forall y_n (P(x_1, \dots, x_n) \wedge (x_1 = y_1) \wedge \dots \wedge (x_n = y_n) \rightarrow P(y_1, \dots, y_n))$$

para cada P relación n -aria.

Tal como hemos definido la teoría T , es claro que admite como único modelo a BD (véase [30]). Además, de acuerdo con esto, BD satisface una restricción ϕ si y sólo si $T \vdash \phi$. Más aún, la respuesta a una consulta formulada como $W(x_1, \dots, x_p)$, donde x_i son variables libres en la fórmula W , consiste en aquellas p -tuplas (a_1, \dots, a_p) tales que $T \vdash W(a_1, \dots, a_p)$.

1.4. Argumentos

La lógica clásica tiene características que la hacen apropiada para la representación del conocimiento y el razonamiento. Pero, desafortunadamente, no tiene ninguna utilidad para razonar con información inconsistente, ya que, en la teoría de demostración clásica, de la inconsistencia se sigue cualquier fórmula. De entre las formas existentes de abordar esta cuestión, destacamos dos. La primera de ellas es el estudio de las *lógicas paraconsistentes*: éstas añaden ciertas premisas que no invalidan la deducción pero toleran las contradicciones.

La segunda consiste en trabajar con subconjuntos consistentes de la base de conocimiento. Trabajando con estos subconjuntos, la lógica clásica se puede utilizar sin restricción; pero este tipo de razonamiento no es monótono, ya que, al añadir ciertas premisas, se puede violar la consistencia del subconjunto en el contexto en que se han realizado las deducciones. No obstante, es preferible trabajar con subconjuntos consistentes porque se mantiene la semántica clásica. Por otro lado, la información que se deduce a partir de subconjuntos de una base de conocimiento inconsistente está, en general, débilmente justificada. Para abordar este problema introducimos las nociones de *argumento* y *aceptabilidad de un argumento*. Estas nociones nos ayudarán a definir el concepto de *estructura argumentativa*.

Un argumento no es más que un subconjunto de la base de datos, junto con

una fórmula inferida de ese subconjunto. Con la noción de aceptabilidad definida sobre el conjunto de todos los argumentos, obtenemos una estratificación de dicho conjunto en subconjuntos con diferentes grados de aceptabilidad. Aquí vamos a describir una estructura argumentativa concreta, que denotaremos \mathbb{A} . Esta estructura tiene una característica que la distingue: está basada en una noción intuitiva de aceptabilidad.

1.4.1. Definiciones y resultados fundamentales

Aplicamos la notación estándar. Así \mathcal{L} representará una lógica clásica usual, y \vdash será el símbolo que represente la inferencia clásica definida sobre \mathcal{L} .

Definición. 1.4.1 Una base de conocimiento Δ es un conjunto de fórmulas en \mathcal{L} .

Definición. 1.4.2 Para una base de conocimiento Δ , $\mathbf{Cn}(\Delta)$ es el conjunto de fórmulas ϕ tales que $\Delta \vdash \phi$.

Definición. 1.4.3 Sea Δ una base de conocimiento. Un **argumento** de Δ es un par $\langle \Pi, \phi \rangle$ tal que $\Pi \subseteq \Delta$ y $\Pi \vdash \phi$. Un argumento es **consistente** si Π es consistente. El **conjunto de los argumentos de Δ** , $\mathbf{An}(\Delta)$, es

$$\mathbf{An}(\Delta) = \{ \langle \Pi, \phi \rangle \mid \Pi \subseteq \Delta \wedge \Pi \vdash \phi \}.$$

Definición. 1.4.4 Sea Δ una base de conocimiento. Sean $\langle \Pi, \phi \rangle$ y $\langle \Theta, \psi \rangle$ dos argumentos de Δ . Si $\vdash \phi \leftrightarrow \neg\psi$, entonces decimos que $\langle \Pi, \phi \rangle$ **rebate a** $\langle \Theta, \psi \rangle$. Si $\gamma \in \Theta$ y $\vdash \phi \leftrightarrow \neg\gamma$, entonces $\langle \Pi, \phi \rangle$ **es un corte de** $\langle \Theta, \psi \rangle$.

La relación “rebate” es simétrica y afecta a (actúa contra) la fórmula inferida, mientras que el corte afecta a (actúa contra) las hipótesis de las que se deduce dicha fórmula.

Definición. 1.4.5 Sea Δ una base de conocimiento y $\Gamma \subseteq \mathbf{An}(\Delta)$. Se define

$$\mathbf{CON}(\Delta) = \{ \Pi \subseteq \Delta \mid \Pi \not\vdash \perp \}$$

$$\mathbf{INC}(\Delta) = \{ \Pi \subseteq \Delta \mid \Pi \vdash \perp \}$$

$$\mathbf{MC}(\Delta) = \{ \Pi \in \mathbf{CON}(\Delta) \mid \forall \Phi \in \mathbf{CON}(\Delta) : \Pi \not\subseteq \Phi \}$$

$$\mathbf{MI}(\Delta) = \{ \Pi \in \mathbf{INC}(\Delta) \mid \forall \Phi \in \mathbf{INC}(\Delta) : \Phi \not\subseteq \Pi \}$$

$$\mathbf{FREE}(\Delta) = \bigcap \mathbf{MC}(\Delta)$$

$$\mathbf{MIN}(\Gamma) = \{ \langle \Pi, \phi \rangle \in \Gamma \mid \forall \langle \Phi, \phi \rangle \in \Gamma : \Phi \not\subseteq \Pi \}$$

Así, $\mathbf{MC}(\Delta)$ es el conjunto de subconjuntos de Δ maximalmente consistentes; $\mathbf{MI}(\Delta)$ es el conjunto de subconjuntos de Δ minimalmente inconsistentes; $\mathbf{FREE}(\Delta)$ es el conjunto de información que tienen en común todos los subconjuntos maximalmente consistentes; y $\mathbf{MIN}(\Gamma)$ es el conjunto de los argumentos minimales de un conjunto de argumentos Γ . Con ánimo de completitud, enunciamos y demostramos las propiedades fundamentales de este tipo de conjuntos.

Lema. 1.4.6 *Sea Δ una base de conocimiento. Entonces:*

$$\mathbf{MI}(\Delta) = \{\Pi \subseteq \Delta \mid (\forall \phi \in \Pi : \Pi - \{\phi\} \in \mathbf{CON}(\Delta)) \wedge \Pi \notin \mathbf{CON}(\Delta)\}$$

Demostración: Queremos probar

$$(1) \mathbf{MI}(\Delta) = \{\Pi \subseteq \Delta \mid (\forall \phi \in \Pi : \Pi - \phi \in \mathbf{CON}(\Delta)) \wedge \Pi \notin \mathbf{CON}(\Delta)\}$$

Por definición,

$$(2) \mathbf{MI}(\Delta) = \{\Pi \in \mathbf{INC}(\Delta) \mid \forall \Phi \in \mathbf{INC}(\Delta) : \Phi \not\subseteq \Pi\}$$

Hacemos la prueba por doble inclusión. Supongamos que $\Pi \in (1)$. Entonces, Π es un subconjunto no consistente de la base de conocimiento Δ tal que todo elemento ϕ verifica que $\Pi - \phi$ sí es consistente. Si $\Pi \notin (2)$, existe un subconjunto inconsistente $\Phi \subset \Pi$. Sea $\psi \in \Pi - \Phi$. Tendremos $\Phi \in \mathbf{INC}(\Delta) \subseteq \Pi - \psi \in \mathbf{CON}(\Delta)$, lo cual es una contradicción.

Veamos la otra inclusión. Supongamos que $\Pi \in (2)$. Si $\Pi \notin (1)$, entonces, o bien existe un $\phi \in \Pi$ tal que $\Pi - \phi$ es inconsistente, o bien Π es consistente. Lo segundo es imposible porque $\Pi \in (2)$. Por otro lado, $\Pi - \phi \subset \Pi$, con lo cual, $\Pi - \phi$ no puede ser inconsistente. ■

Lema. 1.4.7 *Sean Π y Θ bases de conocimiento. Entonces:*

1. $\mathbf{Cn}(\Pi) \cup \mathbf{Cn}(\Theta) \subseteq \mathbf{Cn}(\Pi \cup \Theta)$
2. $\mathbf{Cn}(\Pi \cap \Theta) \subseteq \mathbf{Cn}(\Pi) \cap \mathbf{Cn}(\Theta)$

Demostración:

1. $\phi \in \mathbf{Cn}(\Pi) \cup \mathbf{Cn}(\Theta) \implies \phi \in \mathbf{Cn}(\Pi) \vee \phi \in \mathbf{Cn}(\Theta) \implies \Pi \vdash \phi \vee \Theta \vdash \phi \implies \Pi \cup \Theta \vdash \phi \implies \phi \in \mathbf{Cn}(\Pi \cup \Theta)$
 2. $\phi \in \mathbf{Cn}(\Pi \cap \Theta) \implies \Pi \cap \Theta \vdash \phi \implies \Pi \vdash \phi \wedge \Theta \vdash \phi \implies \Pi \in \mathbf{Cn}(\Pi) \wedge \Pi \in \mathbf{Cn}(\Theta) \implies \Pi \in \mathbf{Cn}(\Pi) \cap \mathbf{Cn}(\Theta)$
-

Lema. 1.4.8 Sea $\phi \in \mathcal{L}$.

$$\forall \Phi \in \mathbf{MC}(\Delta) : \Phi \vdash \phi \implies \forall \Theta \in \mathbf{MC}(\Delta) : \Theta \not\vdash \neg\phi$$

Demostración: Hacemos la prueba por reducción al absurdo. Sea $\phi \in \mathcal{L}$. Supongamos que $\forall \Phi \in \mathbf{MC}(\Delta) : \Phi \vdash \phi$ y $\exists \Theta \in \mathbf{MC}(\Delta)$ tal que $\Theta \vdash \neg\phi$. Como $\Theta \in \mathbf{MC}(\Delta)$, en particular, $\Theta \vdash \perp$, lo cual es imposible al ser consistente. ■

Lema. 1.4.9 Sea $\Pi \in \mathbf{MC}(\Delta)$, $\Phi \subseteq \Delta$.

$$\Phi \not\subseteq \Pi \implies (\exists \phi \in \Phi : \Pi \vdash \neg\phi)$$

Demostración: Sean $\Pi \in \mathbf{MC}(\Delta)$, $\Phi \subseteq \Delta$, y $\Phi \not\subseteq \Pi$. Como $\Phi \not\subseteq \Pi$, entonces $\Phi \neq \emptyset$. Sea $\phi_0 \in \Phi - \Pi$. Supongamos que $\Pi \not\vdash \neg\phi_0$. Entonces $\Pi \cup \{\phi_0\}$ será consistente, lo cual contradice la maximalidad de Π . Por tanto $\Pi \vdash \neg\phi_0$. ■

Lema. 1.4.10 Sean $\Pi \in \mathbf{MI}(\Delta)$, y $\phi \in \Pi$. Entonces $\Pi - \{\phi\} \vdash \neg\phi$.

Demostración: Sean $\Pi \in \mathbf{MI}(\Delta)$ y $\phi \in \Pi$. Entonces $\Pi = (\Pi - \{\phi\}) \cup \{\phi\}$ es inconsistente, y por tanto $(\Pi - \{\phi\}) \cup \{\phi\} \vdash \neg\phi$. Así $\Pi - \{\phi\} \vdash \phi \rightarrow \neg\phi$, y, por tanto, también $\Pi - \{\phi\} \vdash \neg\phi$, porque $\vdash (\phi \rightarrow \neg\phi) \leftrightarrow \neg\phi$. ■

Lema. 1.4.11 Sean $\Pi, \Phi \in \mathbf{CON}(\Delta)$.

$$\Pi \cup \Phi \notin \mathbf{CON}(\Delta) \implies \exists \phi \in \mathcal{L} : ((\Pi \vdash \neg\phi) \wedge (\Phi \vdash \phi))$$

Demostración: Sea $\Theta \subseteq \Phi$ un subconjunto minimal tal que $\Pi \cup \Theta \notin \mathbf{CON}(\Delta)$. Por compacidad de lógica clásica, Θ es un conjunto finito (ya que toda prueba de \perp requiere solamente un número finito de inferencias). Sea ϕ la conjunción de todas las fórmulas de Θ . Entonces $\Pi \cup \{\phi\} \vdash \perp$ y, por tanto, $\Pi \vdash \neg\phi$. También, por construcción, tenemos $\Phi \vdash \phi$. ■

Lema. 1.4.12 Se verifica la siguiente igualdad:

$$\bigcap \mathbf{MC}(\Delta) = \Delta - \bigcup \mathbf{MI}(\Delta)$$

Demostración: Sea $\alpha \in \Delta$.

$$\begin{aligned} \alpha \notin \bigcup \mathbf{MI}(\Delta) &\leftrightarrow \forall \Phi \in \mathbf{MI}(\Delta) : \alpha \notin \Phi \\ &\leftrightarrow \forall \Pi \in \mathbf{CON}(\Delta) : \Pi \not\vdash \neg\alpha \\ &\leftrightarrow \forall \Pi \in \mathbf{MC}(\Delta) : \Pi \not\vdash \neg\alpha \quad [\mathbf{MC}(\Delta) \subseteq \mathbf{CON}(\Delta)] \\ &\leftrightarrow \forall \Pi \in \mathbf{MC}(\Delta) : \alpha \in \Pi \\ &\leftrightarrow \alpha \in \bigcap \mathbf{MC}(\Delta) \end{aligned}$$

■

Podemos considerar un subconjunto maximalmente consistente de una base de conocimiento como una parte coherente de dicha base de conocimiento. Por eso, el conjunto $\mathbf{MC}(\Delta)$ es tan importante en la mayoría de las definiciones presentadas en esta sección. Además, consideramos $\mathbf{FREE}(\Delta)$, que es igual a $\bigcup \mathbf{MC}(\Delta)$, como la información no problemática de Δ . Consideramos el conjunto $\bigcap \mathbf{MI}(\Delta)$ como el conjunto de datos problemáticos de Δ .

El siguiente resultado nos ofrece una forma de obtener, de manera incremental, $\mathbf{MC}(\Delta)$.

Lema. 1.4.13 *Sea \mathbf{max} un operador que escoge los elementos \subseteq -maximales de un conjunto de conjuntos. Entonces*

$$\mathbf{MC}(\Delta \cup \{\alpha\}) = \{\Phi \in \mathbf{MC}(\Delta) : \Phi \vdash \neg\alpha\} \\ \cup \{\Phi \cup \{\alpha\} : \Phi \in \mathbf{max}\{\Psi \in \mathbf{CON}(\Delta) : \Psi \not\vdash \neg\alpha\}\}$$

Demostración: Para simplificar, utilicemos la siguiente notación:

A1: $\Pi \in \mathbf{MC}(\Delta \cup \{\alpha\})$

A2: $\Pi \in \{\Phi \in \mathbf{MC}(\Delta) : \Phi \vdash \neg\alpha\}$

A3: $\Pi \in \{\Phi \cup \{\alpha\} : \Phi \in \mathbf{max}\{\Psi \in \mathbf{CON}(\Delta) : \Psi \not\vdash \neg\alpha\}\}$

En primer lugar probamos, por reducción al absurdo, que A1 implica que se verifica A2 ó bien A3.

- Si $\Pi \vdash \neg\alpha$, entonces es trivial que se tiene A2.
- En otro caso, $\Pi \not\vdash \neg\alpha \implies \alpha \in \Pi$ (al ser Π maximal consistente).

Si $\Pi \subseteq \Delta$, es decir, $\alpha \in \Delta$, entonces Π es maximal en $\mathbf{CON}(\Delta)$, lo cual contradice las hipótesis.

Tenemos entonces: $\alpha \notin \Delta \implies \Pi - \{\alpha\} \subseteq \Delta$ y $\Pi - \{\alpha\}$ es maximal en $\{\Psi \in \mathbf{CON}(\Delta) : \Psi \not\vdash \neg\alpha\}$, ya que en otro caso Π no podría estar en $\mathbf{MC}(\Delta \cup \{\alpha\})$. Pero esto contradice de nuevo las hipótesis, por tanto se tiene A2 ó A3.

Probamos que si se verifica A2 ó A3, entonces debe darse A1, y lo hacemos por reducción al absurdo.

- Si tenemos A2, entonces $\Pi \in \mathbf{MC}(\Delta)$ y Π no es consistente con α . Así que tiene que ser A1, contradiciendo las hipótesis.

- Si tenemos A3, entonces, para algún Φ , se tendrá $\Phi \cup \{\alpha\} = \Pi$ y Φ es maximal en $\{\Psi \in \mathbf{CON}(\Delta) : \Psi \not\vdash \neg\alpha\}$. Como Π no es maximal en $\mathbf{CON}(\Delta \cup \{\alpha\})$, entonces, para algún $\beta \in \Delta - \Pi$ se tendrá $\Phi \cup \{\alpha, \beta\} \not\vdash \neg\alpha$. Pero esto contradice la maximalidad de Φ , porque $\beta \notin \Phi$ y $\Phi \cup \{\beta\} \not\vdash \neg\alpha$. Así podemos afirmar que se tiene A1. ■

Ejemplo. 1.4.14 Sea $\Delta = \{\alpha, \gamma \wedge (\alpha \vee \neg\beta), \neg\gamma \vee (\neg\alpha \vee \neg\beta)\}$. Entonces

- $\mathbf{MC}(\Delta) = \{\{\alpha, \gamma \wedge (\alpha \vee \neg\beta)\}, \{\alpha, \neg\gamma \vee (\neg\alpha \vee \neg\beta)\}\}$.
- $\mathbf{MC}(\Delta \cup \{\beta\}) = \{\{\alpha, \gamma \wedge (\alpha \vee \neg\beta), \beta\}, \{\alpha, \neg\gamma \vee (\neg\alpha \vee \neg\beta)\}, \{\neg\gamma \vee (\neg\alpha \vee \neg\beta), \beta\}\}$.

Este es un ejemplo en que vemos que $\mathbf{MC}(\Delta \cup \{\beta\})$ se puede construir directamente a partir de $\mathbf{MC}(\Delta)$.

Corolario. 1.4.15 Sea $\alpha \in \mathcal{L}$.

1. Sea $\Phi \in \mathbf{MC}(\Delta)$. Entonces $\Phi \cup \{\alpha\} \in \mathbf{MC}(\Delta \cup \{\alpha\}) \iff \Phi \not\vdash \neg\alpha$
2. Sea $\Phi \in \mathbf{MC}(\Delta)$. Entonces $\Phi \vdash \neg\alpha \implies \Phi \in \mathbf{MC}(\Delta \cup \{\alpha\})$
3. Si $\alpha \not\vdash \perp$, entonces
 $(\forall \Phi \in \mathbf{MC}(\Delta) : \Phi \vdash \neg\alpha) \implies \mathbf{MC}(\Delta \cup \{\alpha\}) = \mathbf{MC}(\Delta) \cup \{\{\alpha\}\}$
4. Sea $\Phi \in \mathbf{MC}(\Delta)$. Entonces $\exists \Psi \in \mathbf{MC}(\Delta \cup \{\alpha\}) : \Phi \subseteq \Psi$

Demostración: Detallamos la prueba del primer punto, pues los otros tres son directos por el Lema 1.4.13.

1. Implicación hacia la derecha. Por el Lema 1.4.13, sabemos que

$$\mathbf{MC}(\Delta \cup \{\alpha\})$$

$$= \{\Phi \in \mathbf{MC}(\Delta) : \Phi \vdash \neg\alpha\} \cup \{\Phi \cup \{\alpha\} : \Phi \in \max\{\Psi \in \mathbf{CON}(\Delta) : \Psi \not\vdash \neg\alpha\}\}.$$

Por tanto, si

$$\Phi \in \mathbf{MC}(\Delta) : \Phi \cup \{\alpha\} \in \mathbf{MC}(\Delta \cup \{\alpha\})$$

entonces, será que: o bien Φ es consistente y prueba $\neg\alpha$, o bien $\Phi \in \max\{\Psi \in \mathbf{CON}(\Delta) : \Psi \not\vdash \neg\alpha\}$.

En el primer caso, si Φ es consistente y prueba $\neg\alpha$, entonces $\Phi \cup \{\alpha\}$ no puede ser consistente, con lo cual tiene que ser el segundo caso. De donde deducimos que $\Phi \not\vdash \neg\alpha$.

2. La implicación a la izquierda es trivial. ■

Lema. 1.4.16

$$\mathbf{FREE}(\Delta \cup \{\alpha\}) \subseteq \mathbf{FREE}(\Delta) \cup \{\alpha\}$$

Demostración: Aplicando la definición de **FREE** y el lema 1.4.13, $\mathbf{FREE}(\Delta \cup \{\alpha\})$ es igual a :

(*) $\bigcap(\{\Phi \in \mathbf{MC}(\Delta) \mid \Phi \vdash \neg\alpha\} \cup \{\Phi \cup \{\alpha\} \mid \Phi \in \mathbf{max}\{\Psi \in \mathbf{CON}(\Delta) \mid \Psi \not\vdash \neg\alpha\}\})$
Ahora, consideramos dos casos.

1. Si ningún $\Phi \in \mathbf{MC}(\Delta)$ CUMPLE $\Phi \vdash \neg\alpha$, entonces, expresando el segundo conjunto como $\mathbf{MC}(\Delta) = \mathbf{max}(\mathbf{CON}(\Delta))$, la expresión de (*) será :

$$\begin{aligned} & \bigcap\{\Phi \cup \{\alpha\} \mid \Phi \in \mathbf{max}(\mathbf{CON}(\Delta))\} \\ &= \bigcap\{\Phi \cup \{\alpha\} \mid \Phi \in \mathbf{MC}(\Delta)\} \\ &= \{\alpha\} \cup \bigcap \mathbf{MC}(\Delta) \\ &= \mathbf{FREE}(\Delta) \cup \{\alpha\} \end{aligned}$$

2. Si existe un $\Phi \in \mathbf{MC}(\Delta)$ tal que $\Phi \vdash \neg\alpha$, entonces, (*) será igual a: $\bigcap(A \cup C)$, donde

$$A = \{\Phi \mid \Phi \in \mathbf{MC}(\Delta) \wedge \Phi \vdash \neg\alpha\}$$

$$B = \{\Phi \mid \Phi \in \mathbf{MC}(\Delta) \wedge \Phi \not\vdash \neg\alpha\}$$

$$C = \{\Phi \cup \{\alpha\} \mid \Phi \in \mathbf{max}\{\Psi \in \mathbf{CON}(\Delta) \mid \Psi \not\vdash \alpha\}\}$$

$$D = \{\Phi \mid \Phi \in \mathbf{max}\{\Psi \in \mathbf{CON}(\Delta) \mid \Psi \not\vdash \alpha\}\}$$

Como $\forall \Phi \in A : \Phi \vdash \neg\alpha$, α no puede estar en la intersección de $A \cup C$. Por tanto, $\bigcap(A \cup C) = \bigcap(A \cup D)$. También tenemos $\bigcap(\mathbf{MC}(\Delta)) = \bigcap(A \cup B)$. Además, es directo probar que $B \subseteq D$. Usando la teoría de conjuntos estándar, tenemos $\bigcup(A \cup D) \subseteq \bigcup(A \cup B)$. Así, $\bigcap(\mathbf{MC}(\Delta \cup \{\alpha\})) \subseteq \bigcap(\mathbf{MC}(\Delta))$. ■

Como veremos más adelante, estos resultados son necesarios al derivar inferencias de $\mathbf{FREE}(\Delta)$, ya que la opción de actualización de la forma $\mathbf{FREE}(\Delta \cup \{\alpha\})$ ó bien $\mathbf{FREE}(\Delta) \cup \{\alpha\}$ puede afectar al razonamiento.

1.4.2. La estructura argumentativa \mathbb{A}

Para una base de conocimiento Δ , definimos una estructura argumentativa como un conjunto de subconjuntos de $\mathbf{An}(\Delta)$. La idea es clasificar los subconjuntos de $\mathbf{An}(\Delta)$ según su grado de aceptabilidad. A continuación definimos una estructura argumentativa particular \mathbb{A} y veremos cómo esa definición captura la noción de aceptabilidad. Aunque tal estructura no será plenamente utilizada (salvo los primeros niveles), con ánimo de completitud las analizaremos globalmente

Evidentemente pueden definirse estructuras diferentes. Sin embargo, \mathbb{A} se distingue porque está definida a partir de los conceptos básicos de la lógica clásica. Trataremos con subconjuntos consistentes, subconjuntos consistentes maximales y subconjuntos libres (FREE) tal como se han definido en la sección anterior.

Definición. 1.4.17 (La estructura argumentativa \mathbb{A}) Sea Δ una base de conocimiento. Se definen los siguientes conjuntos:

$$\mathbf{AT}(\Delta) = \{\langle \emptyset, \phi \rangle \mid \emptyset \vdash \phi\}$$

$$\mathbf{AF}(\Delta) = \{\langle \Pi, \phi \rangle \mid \Pi \subseteq \mathbf{FREE}(\Delta) \wedge \Pi \vdash \phi\}$$

$$\mathbf{AB}(\Delta) = \{\langle \Pi, \phi \rangle \mid \Pi \in \mathbf{CON}(\Delta) \wedge \Pi \vdash \phi \wedge (\forall \Phi \in \mathbf{MC}(\Delta), \psi \in \Pi : \Phi \vdash \psi)\}$$

$$\mathbf{ARU}(\Delta) = \{\langle \Pi, \phi \rangle \mid \Pi \in \mathbf{CON}(\Delta) \wedge \Pi \vdash \phi \wedge (\forall \Phi \in \mathbf{MC}(\Delta) : \Phi \not\vdash \neg\phi) \wedge (\forall \Phi \in \mathbf{MC}(\Delta), \psi \in \Pi : \Phi \not\vdash \neg\psi)\}$$

$$\mathbf{AU}(\Delta) = \{\langle \Pi, \phi \rangle \mid \Pi \in \mathbf{CON}(\Delta) \wedge \Pi \vdash \phi \wedge (\forall \Phi \in \mathbf{MC}(\Delta), \psi \in \Pi : \Phi \not\vdash \neg\psi)\}$$

$$\mathbf{A}\forall(\Delta) = \{\langle \Pi, \phi \rangle \mid \Pi \in \mathbf{CON}(\Delta) \wedge \Pi \vdash \phi \wedge (\forall \Phi \in \mathbf{MC}(\Delta) : \Phi \vdash \phi)\}$$

$$\mathbf{AR}(\Delta) = \{\langle \Pi, \phi \rangle \mid \Pi \in \mathbf{CON}(\Delta) \wedge \Pi \vdash \phi \wedge (\forall \Phi \in \mathbf{MC}(\Delta) : \Phi \not\vdash \neg\phi)\}$$

$$\mathbf{A}\exists(\Delta) = \{\langle \Pi, \phi \rangle \mid \Pi \in \mathbf{CON}(\Delta) \wedge \Pi \vdash \phi\}$$

Para nombrarlos hemos seguido el siguiente criterio. Utilizamos **T** para argumentos tautológicos, los que se siguen del conjunto vacío de premisas. **F** es para argumentos libres, los que se siguen de datos libres de inconsistencias. **B** es para los argumentos cuyas premisas se siguen de todos los subconjuntos de datos maximalmente consistentes. **RU** es para los argumentos que no son objeto de corte ni refutación. **U**, para los argumentos que no son objeto de corte. \forall es para argumentos universales (inevitables), que son los que se siguen de todos los subconjuntos maximalmente consistentes de datos. **R** es para los argumentos que no son objeto de refutación. \exists es para argumentos existenciales, los de premisas consistentes.

Las definiciones para $\mathbf{A}\exists$, \mathbf{AF} , \mathbf{AT} son claras. Nos fijamos en el resto. **AR** permite un argumento $\langle \Pi, \phi \rangle$ si y sólo si no hay un subconjunto maximalmente consistente que pruebe $\neg\phi$. **AU** permite un argumento $\langle \Pi, \phi \rangle$ sólo si para todo $\psi \in \Pi$ no hay un subconjunto maximalmente consistente que pruebe $\neg\psi$. **ARU** combina las condiciones de **AR** y **AU**. Hacer notar que **AR** y $\mathbf{A}\forall$ tienen definiciones muy parecidas, con la única diferencia de " $\Phi \not\vdash \neg\phi$ " en **AR** y " $\Phi \vdash \phi$ " en

$\mathbf{A}\forall$. Algo similar ocurre con $\mathbf{A}\mathbf{U}$ y $\mathbf{A}\mathbf{B}$. Por tanto $\mathbf{A}\forall$ y $\mathbf{A}\mathbf{B}$ son los refuerzos de $\mathbf{A}\mathbf{R}$ y $\mathbf{A}\mathbf{U}$ respectivamente, es decir, sustituyen $\not\vdash \neg\phi$ por $\vdash \phi$.

Ejemplo. 1.4.18 Consideremos $\Delta = \{\alpha, \neg\alpha\}$. Entonces, se verifica:

- $\langle \{\alpha, \neg\alpha\}, \alpha \wedge \neg\alpha \rangle \in \mathbf{A}\mathbf{n}(\Delta)$
- $\langle \{\alpha\}, \alpha \rangle \in \mathbf{A}\exists(\Delta)$
- $\langle \{\alpha\}, \alpha \vee \beta \rangle \in \mathbf{A}\mathbf{R}(\Delta)$
- Si $\beta \not\vdash \alpha$, entonces $\langle \{\}, \alpha \vee \neg\alpha \rangle \in \mathbf{A}\forall(\Delta)$. Más aún, $\mathbf{A}\forall(\Delta) = \mathbf{A}\mathbf{F}(\Delta) = \mathbf{A}\mathbf{B}(\Delta) = \mathbf{A}\mathbf{R}\mathbf{U}(\Delta) = \mathbf{A}\mathbf{U}(\Delta) = \mathbf{A}\mathbf{T}(\Delta)$.

Ejemplo. 1.4.19 Consideremos $\Delta = \{\neg\alpha \wedge \beta, \alpha \wedge \beta\}$. Entonces, para $\Pi = \{\alpha \wedge \beta\}$

- $\langle \Pi, \beta \rangle \in \mathbf{A}\exists(\Delta)$, $\langle \Pi, \beta \rangle \in \mathbf{A}\mathbf{R}(\Delta)$ y $\langle \Pi, \beta \rangle \in \mathbf{A}\forall(\Delta)$.
- Sin embargo, no hay un $\Pi \subseteq \Delta$ tal que $\langle \Pi, \beta \rangle \in \mathbf{A}\mathbf{U}(\Delta)$, $\langle \Pi, \beta \rangle \in \mathbf{A}\mathbf{R}\mathbf{U}(\Delta)$, $\langle \Pi, \beta \rangle \in \mathbf{A}\mathbf{B}(\Delta)$ ó $\langle \Pi, \beta \rangle \in \mathbf{A}\mathbf{F}(\Delta)$.

Proposición. 1.4.20 Se verifica la siguiente cadena de igualdades/contenciones:

$$\mathbf{A}\mathbf{T}(\Delta) \subseteq \mathbf{A}\mathbf{F}(\Delta) = \mathbf{A}\mathbf{B}(\Delta) = \mathbf{A}\mathbf{R}\mathbf{U}(\Delta) = \mathbf{A}\mathbf{U}(\Delta) \subseteq \mathbf{A}\forall(\Delta) \subseteq \mathbf{A}\mathbf{R}(\Delta) \subseteq \mathbf{A}\exists(\Delta) \subseteq \mathbf{A}\mathbf{n}(\Delta)$$

Demostración:

- $\mathbf{A}\mathbf{T}(\Delta) \subseteq \mathbf{A}\mathbf{F}(\Delta)$: Dado que $\emptyset \subseteq \mathbf{F}\mathbf{R}\mathbf{E}\mathbf{E}(\Delta)$ para todo Δ , el conjunto de argumentos tautológicos se puede definir como $\{\langle \Pi, \phi \rangle \in \mathbf{A}\mathbf{F}(\Delta) \mid \Pi = \emptyset\}$.
- $\mathbf{A}\mathbf{F}(\Delta) \not\subseteq \mathbf{A}\mathbf{T}(\Delta)$: Para construir un contraejemplo, sea la base de conocimiento $\Delta = \{\alpha\}$, conteniendo una única fórmula. Entonces $\langle \Delta, \alpha \rangle \in \mathbf{A}\mathbf{F}(\Delta)$, pero $\langle \Delta, \alpha \rangle \notin \mathbf{A}\mathbf{T}(\Delta)$.
- $\mathbf{A}\mathbf{B}(\Delta) \subseteq \mathbf{A}\mathbf{F}(\Delta)$:
Supongamos (i) $\langle \Pi, \phi \rangle \notin \mathbf{A}\mathbf{F}(\Delta)$ para probar (ii) $\langle \Pi, \phi \rangle \notin \mathbf{A}\mathbf{B}(\Delta)$.
Si $\Pi \not\vdash \phi$, entonces se tiene (ii). En otro caso $\Pi \vdash \phi$ y $\Pi \not\subseteq \mathbf{F}\mathbf{R}\mathbf{E}\mathbf{E}(\Delta)$.
Escojamos un $\Phi_0 \in \mathbf{M}\mathbf{C}(\Delta)$ tal que $\Pi \not\subseteq \Phi_0$. Por el lema 1.4.9 tenemos $\exists \psi \in \Pi$ tal que $\Phi_0 \vdash \neg\psi$. De aquí se sigue (ii) porque ahora tenemos $\neg\forall\Phi \in \mathbf{M}\mathbf{C}(\Delta)$, $\psi \in \Pi : \phi \vdash \psi$.
- $\mathbf{A}\mathbf{F}(\Delta) \subseteq \mathbf{A}\mathbf{B}(\Delta)$: Supongamos (ii) para demostrar (i). O bien $\Pi \not\vdash \phi$, o bien $\Pi \not\subseteq \mathbf{C}\mathbf{O}\mathbf{N}(\Delta)$; en estos casos se tendrá (i) ó $\Pi \vdash \phi$ y $\Pi \in \mathbf{C}\mathbf{O}\mathbf{N}(\Delta)$ y podemos tomar un Φ_0 y un ψ_0 tales que $\Phi_0 \in \mathbf{M}\mathbf{C}(\Delta)$, $\psi_0 \in \Pi$ y $\Phi_0 \not\vdash$

ψ_0 . Por hipótesis, $\psi_0 \notin \mathbf{Cn}(\Phi_0)$ y por las propiedades de la intersección, tenemos $\psi_0 \notin \bigcap_{\Phi \in \mathbf{MC}(\Delta)} \mathbf{Cn}(\Phi)$. Usando el lema 1.4.7 se obtiene $\phi_0 \notin \mathbf{Cn}(\bigcap_{\Phi \in \mathbf{MC}(\Delta)} (\Phi))$. De aquí se sigue (i), ya que $\Pi \not\subseteq \mathbf{FREE}(\Delta)$.

- $\mathbf{AU}(\Delta) \subseteq \mathbf{AB}(\Delta)$: Supongamos (ii) para demostrar (iii) $\langle \Pi, \phi \rangle \notin \mathbf{AU}(\Delta)$. O bien $\Pi \notin \mathbf{CON}(\Delta)$ o bien $\Pi \not\vdash \phi$, en cuyo caso se sigue (iii), o bien $\Pi \in \mathbf{CON}(\Delta)$ y $\Pi \vdash \phi$ y podemos elegir unos Φ_0 y ψ_0 tales que $\Phi_0 \in \mathbf{MC}(\Delta)$, $\psi_0 \in \Pi$ y $\Phi_0 \not\vdash \psi_0$. Ya que $\Pi \vdash \psi_0$ y $\Phi_0 \not\vdash \psi_0$, tendremos que $\Pi \cup \Phi_0 \notin \mathbf{CON}(\Delta)$, porque en otro caso Φ_0 no sería maximal consistente. Por el Lema 1.4.9, como $\Pi \not\subseteq \Phi_0$, hay algún $\Psi_1 \in \Pi$ tal que $\Phi_0 \vdash \neg \psi_1$. De aquí se sigue (iii).
- $\mathbf{AB}(\Delta) \subseteq \mathbf{AU}(\Delta)$: Esto se sigue del lema 1.4.8.
- $\mathbf{ARU}(\Delta) \subseteq \mathbf{AU}(\Delta)$: $\mathbf{ARU}(\Delta)$ se puede definir como $\mathbf{AR}(\Delta) \cap \mathbf{AU}(\Delta)$, y como $\mathbf{AU}(\Delta) \subseteq \mathbf{AR}(\Delta)$, éste es equivalente a $\mathbf{AU}(\Delta)$.
- $\mathbf{AV}(\Delta) \not\subseteq \mathbf{AU}(\Delta)$: Sea $\Delta = \{\alpha \wedge \beta, \neg \alpha \wedge \beta\}$. Entonces $\langle \Delta, \beta \rangle \in \mathbf{AV}(\Delta)$ pero $\langle \Delta, \beta \rangle \notin \mathbf{AU}(\Delta)$.
- $\mathbf{AU}(\Delta) \subseteq \mathbf{AV}(\Delta)$: Supongamos (iv) $\langle \Pi, \phi \rangle \notin \mathbf{AV}(\Delta)$ para probar (iii). O bien $\Pi \notin \mathbf{MC}(\Delta)$ ó $\Pi \not\vdash \phi$, en ambos casos se tiene (iv), o bien $\Pi \in \mathbf{MC}(\Delta)$ y $\Pi \vdash \phi$ y podemos elegir algún Φ_0 tal que $\Phi_0 \in \mathbf{MC}(\Delta)$ y $\Phi_0 \not\vdash \phi$. Supongamos, para llegar a contradicción, que (v) $\forall \theta \in \mathbf{MC}(\Delta), \psi \in \Pi : \theta \not\vdash \neg \psi$. Usando el lema 1.4.9, como en el caso anterior, podemos encontrar un $\psi_0 \in \Pi$ tal que $\Phi_0 \vdash \neg \psi_0$. Esto contradice (v). De la negación de (v) se deduce que debe cumplirse (iii).
- $\mathbf{AV}(\Delta) \subseteq \mathbf{AR}(\Delta)$: Esto se tiene por el Lema 1.4.8.
- $\mathbf{AR}(\Delta) \not\subseteq \mathbf{AV}(\Delta)$: Sea $\Delta = \{\alpha, \neg \alpha \wedge \beta\}$. Entonces $\langle \Delta, \beta \rangle \in \mathbf{AR}(\Delta)$, pero $\langle \Delta, \beta \rangle \notin \mathbf{AV}(\Delta)$.
- $\mathbf{AR}(\Delta) \subseteq \mathbf{A}\exists(\Delta)$: $\mathbf{AR}(\Delta)$ se puede definir como $\{\langle \Pi, \phi \rangle \in \mathbf{A}\exists(\Delta) \mid \forall \Phi \in \mathbf{MC}(\Delta) : \Phi \not\vdash \neg \phi\}$.
- $\mathbf{A}\exists(\Delta) \not\subseteq \mathbf{AR}(\Delta)$: Sea $\Delta = \{\alpha, \neg \alpha\}$. Entonces $\langle \Delta, \alpha \rangle \in \mathbf{A}\exists(\Delta)$ pero $\langle \Delta, \alpha \rangle \notin \mathbf{AR}(\Delta)$.
- $\mathbf{A}\exists(\Delta) \subseteq \mathbf{An}(\Delta)$: $\mathbf{A}\exists(\Delta)$ se puede definir como $\{\langle \Pi, \phi \rangle \in \mathbf{An}(\Delta) : \Pi \in \mathbf{CON}(\Delta)\}$.
- $\mathbf{An}(\Delta) \not\subseteq \mathbf{A}\exists(\Delta)$: Sea $\Delta = \{\alpha \wedge \alpha\}$. Entonces $\langle \Delta, \beta \rangle \in \mathbf{An}(\Delta)$, pero $\langle \Delta, \beta \rangle \notin \mathbf{A}\exists(\Delta)$.



Resumimos este resultado con el diagrama de la figura siguiente:

$$\begin{array}{c}
 AT(\Delta) \\
 | \cap \\
 AF(\Delta) = AB(\Delta) = ARU(\Delta) = AU(\Delta) \\
 | \cap \\
 A\forall(\Delta) \subseteq AR(\Delta) \subseteq A\exists(\Delta) \subseteq An(\Delta)
 \end{array}$$

La principal característica a destacar es que \mathbb{A} es una estructura lineal y los conjuntos **AF**, **AB**, **ARU** y **AU** son equivalentes. No obstante, señalemos que la definición de \mathbb{A} está basada en la noción de consecuencia y en los conceptos de la lógica clásica. Cambiando la lógica obtendríamos una estructura con una jerarquía totalmente diferente.

1.5. OTTER y MACE4

OTTER es un demostrador de teoremas por resolución para lógica de primer orden con igualdad. Las siglas de OTTER proceden de la expresión *Organized Techniques for Theorem-proving and Effective Research* (Técnicas Organizadas para demostración de Teoremas e Investigación Efectiva). Fue desarrollado a finales de los años ochenta en el Argonne National Laboratory (Argonne, Illinois) por William W. McCune. En esencia es parecido a otros demostradores asociados a Argonne, como AURA (AUtomated Reasoning Assistant) y LMA/ITP (Logic Machine Architecture/Interactive Theorem Prover). Está escrito en lenguaje C. No es interactivo; se distingue principalmente de otros por su portabilidad y la simplicidad del código. Para una introducción al uso de OTTER, véase [19], y el manual de referencia es [50]. Para un estudio en profundidad del razonamiento automático con OTTER, sus fundamentos y uso, se puede consultar [41].

La *operación fundamental* de OTTER es una función del conjunto de espacios de cláusulas en sí mismo. El espacio de cláusulas consiste en cuatro listas de cláusulas (que no tienen que ser no vacías): la lista `usable`, la lista `soporte`, la pasiva `passive` y `demodulators` de demoduladores. Las cláusulas de la lista `usable` son las que utiliza OTTER cuando hace una inferencia y están soportadas por una cláusula del conjunto `soporte`, en el sentido de que al menos uno de los antecesores de cada cláusula `usable` procede de la lista

sos. OTTER tiene la capacidad de demodular las cláusulas inferidas, es decir, reescribirlas usando las reglas de reescritura de la lista de demoduladores junto con las reglas implícitas del procesador. Las cláusulas de la lista pasiva no cambian en el curso del proceso de razonamiento. La operación fundamental de la que hablamos es una operación de *inferencia* que depende de la activación de varias opciones, pero una vez activadas esas opciones, está determinada de manera única. Todas las opciones tienen asignado un valor por defecto.

Aunque OTTER cuenta con un modo autónomo, el cual realiza la búsqueda de pruebas de manera bastante efectiva, el uso del demostrador requiere en la práctica la intervención del usuario, pudiendo éste escoger las reglas y opciones para controlar el proceso. El usuario proporciona a OTTER un fichero de entrada que contiene las fórmulas de primer orden (ó el conjunto de cláusulas correspondiente) y en el que se establecen los cambios en los valores por defecto de las opciones. A continuación, el demostrador genera un fichero de salida que contiene información sobre el proceso de razonamiento aplicado a la entrada. Si OTTER no facilita una prueba, el usuario puede volver a intentarlo variando las condiciones iniciales. Entre las opciones, aparecen diversas reglas de inferencia (resolución binaria, hiperresolución, resolución-UR, etc.), que por defecto están desactivadas. La descripción exacta del bucle principal de OTTER puede encontrarse en [50], pero, en líneas generales, el ciclo es el siguiente:

- Se selecciona una cláusula del soporte *sos*, que llamaremos “cláusula dada” y se mueve a la lista *usable*.
- Usando las reglas disponibles, las de inferencia que estén activadas y las reglas del proceso de evaluación, se infieren todas las cláusulas que resultan de combinar la cláusula dada con cada una de las cláusulas que ya estaban en la lista *usable*.
- Se procesa cada una de las nuevas cláusulas obtenidas, siendo unas descartadas, otras modificadas y luego llevadas al soporte para empezar de nuevo.

El procesamiento de las cláusulas es complejo. En [41] se ofrece un estudio en profundidad del mismo. Con respecto a las reglas, la de hiperresolución es una generalización de la regla *modus ponens* de la lógica clásica. La regla de resolución UR realiza inferencias que hiperresolución no es capaz de hacer y proporciona pruebas indirectas, en contraposición a hiperresolución que produce pruebas directas. En principio, si no obtenemos una prueba usando hiperresolución, podemos intentarlo con resolución UR, ya que las pruebas indirectas son más fáciles de encontrar. OTTER ofrece la posibilidad de utilizar una variante de la hiperresolución que es la hiperresolución negativa. La

regla de resolución binaria es una generalización de la regla de corte de la lógica clásica y se conoce simplemente como “resolución”. De hecho, las tres reglas mencionadas anteriormente pueden obtenerse como resultado de aplicar repetidamente la resolución binaria. En [41] pueden encontrarse diversos ejemplos del uso de dichas reglas. Otro elemento que juega un papel elemental en el procesamiento de las cláusulas es la subsunción. Ésta ofrece un mecanismo para eliminar una cláusula cuando aparece otra más general.

Con respecto a la adecuación y completitud de OTTER, sabemos que la completitud de las reglas de inferencia básicas es importante, pero hay refinamientos y restricciones incompletas que, en ocasiones, son necesarias y útiles para encontrar una prueba. En [50], el autor expone una breve lista de casos. En cuanto a la adecuación, OTTER no ha sido verificado formalmente: en ocasiones, las pruebas proporcionadas pueden no resultar del todo claras y es necesario comprobarlas, pues existe la posibilidad de que no sean correctas.

Por otro lado, MACE4 busca modelos finitos de fórmulas de primer orden. El sistema es compatible con el demostrador que acabamos de describir, de manera que el mismo fichero de entrada puede servir para, en caso de que el demostrador no consiga una prueba, buscar un contraejemplo. El usuario facilita al buscador de modelos un conjunto de fórmulas/cláusulas de primer orden (con igualdad) y el tamaño del modelo a buscar (n). Los elementos del dominio serán entonces $\{0, 1, \dots, n-1\}$. En primer lugar MACE4 crea las tablas ($n \times n$) para cada símbolo de función y predicado y , a continuación, genera todas las instancias de las cláusulas de entrada (en el dominio). Luego, de manera sistemática, va completando (con ceros y unos, según los valores de verdad asignados) las posiciones de las tablas mediante la propagación del efecto de las asignaciones. Cuando se encuentra una contradicción, hace backtraking deshaciendo las asignaciones e intenta una nueva asignación. Si no encuentra contradicciones, genera un modelo, es decir, una interpretación en la que todas las cláusulas de entrada son verdaderas. Para más información, véase [51].

1.6. Espacios Topológicos

En el desarrollo de este trabajo estamos interesados en uno de los aspectos más fundamentales del espacio, su topología. Las relaciones topológicas son aquellas que se mantienen invariantes por transformaciones continuas del espacio. La topología ofrece la posibilidad de clasificar diferentes tipos de espacios (espacios topológicos) según verifiquen o no ciertas propiedades. Cuando se desea desarrollar un cálculo para Razonamiento Espacial Cualitativo (REC) de manera general, que no dependa de un espacio particular, entonces la topología es la herramienta más adecuada.

En esta sección introducimos y definimos los conceptos topológicos que se usarán a lo largo de esta memoria.

Definición. 1.6.1 Sea \mathcal{U} un conjunto no vacío que llamaremos universo. Una **topología** sobre \mathcal{U} es una familia \mathcal{T} de subconjuntos de \mathcal{U} que satisface los siguientes axiomas:

1. $\mathcal{U}, \emptyset \in \mathcal{T}$
2. La unión de conjuntos de \mathcal{T} pertenece a \mathcal{T}
3. La intersección de dos conjuntos de \mathcal{T} pertenece a \mathcal{T}

Un **espacio topológico** es un par $\langle \mathcal{U}, \mathcal{T} \rangle$. Los elementos de \mathcal{T} se llaman **conjuntos abiertos**.

En un espacio topológico $\langle \mathcal{U}, \mathcal{T} \rangle$, un subconjunto X de \mathcal{U} es un conjunto **cerrado** si su complementario $\mathcal{U} \setminus X$ es abierto, es decir, $\mathcal{U} \setminus X \in \mathcal{T}$.

Un concepto íntimamente relacionado con el de conjunto abierto es el de entorno.

Definición. 1.6.2 Sea $\langle \mathcal{U}, \mathcal{T} \rangle$ un espacio topológico y $p \in \mathcal{U}$ un punto de \mathcal{U} .

- $N \subset \mathcal{U}$ es un **entorno** de p si existe un abierto $O \subset \mathcal{U}$ tal que $p \in O \subset N$.
- La familia de todos los entornos de p se llama **sistema de entornos de p** y lo denotamos \mathcal{N}_p .

Un sistema de entornos \mathcal{N}_p tiene la propiedad de que toda intersección finita de elementos de \mathcal{N}_p está en \mathcal{N}_p . Utilizando el concepto de entorno se definen ciertos puntos y áreas de una región.

Definición. 1.6.3 Sea \mathcal{U} un espacio topológico, $X \subset \mathcal{U}$ y $p \in \mathcal{U}$.

- p es un **punto interior** de X si existe un entorno N de p contenido en X . El conjunto de puntos interiores de X se llama **interior de X** , y se denota por $\overset{\circ}{X}$.
- p es un **punto frontera** de X si todo entorno de p contiene al menos un punto de X y un punto que no pertenece a X . El conjunto de los puntos frontera de X se llama **frontera de X** y se denota por ∂X .
- La **clausura** de X , que denotamos \bar{X} , es el menor conjunto cerrado que contiene a X .

Nótese que todo cerrado coincide con su clausura y todo abierto, con su interior.

Definición. 1.6.4 Sea X un subconjunto de un espacio topológico \mathcal{U} .

- X es **abierto regular** si X coincide con el interior de su clausura, es decir, $X = \overset{\circ}{\overline{X}}$.
- X es **cerrado regular** si X coincide con la clausura de su interior, es decir, $X = \overline{\overset{\circ}{X}}$.

Dos conjuntos de un espacio topológico están **separados** si la clausura de uno de ellos es disjunta del otro y vice-versa.

Como hemos comentado, los espacios topológicos se clasifican según ciertas propiedades. Éstas se refieren a la manera en que los puntos o los conjuntos cerrados se pueden separar entre sí por conjuntos abiertos. Los **axiomas de separación** establecen las diferentes posibilidades. Los más relevantes para esta memoria son los siguientes:

- T_1 : Un espacio topológico es un T_1 -**espacio** si dados dos puntos distintos $p, q \in \mathcal{U}$, cada uno de ellos pertenece a un conjunto abierto que no contiene al otro.
- T_2 : Un espacio topológico es un T_2 -**espacio** si dados dos puntos distintos $p, q \in \mathcal{U}$, existen dos abiertos disjuntos $\mathcal{O}_p, \mathcal{O}_q \subseteq \mathcal{U}$ donde $p \in \mathcal{O}_p, q \in \mathcal{O}_q$.
- T_3 : Un espacio topológico es un T_3 -**espacio** si dado un conjunto cerrado X y un punto $p \notin X$, existen dos abiertos disjuntos $\mathcal{O}_X, \mathcal{O}_p \subseteq \mathcal{U}$ tales que $p \in \mathcal{O}_p$ y $X \subseteq \mathcal{O}_X$.

Un **espacio conexo** es un espacio topológico que no puede expresarse como unión disjunta de dos abiertos. Un espacio topológico es **regular** si satisface los axiomas T_2 y T_3 .

1.7. Satisfacción de restricciones

Informalmente, una restricción sobre una sucesión de variables es una relación entre sus dominios. Es una condición que establece qué combinaciones de valores de los respectivos dominios son admisibles. En consecuencia, un **problema de satisfacción de restricciones** (PSR) es un conjunto finito de restricciones, cada una sobre una subsucesión de la sucesión de variables dada. En esta sección definimos formalmente estos conceptos.

Definición. 1.7.1 Sea $Y = y_1, \dots, y_k$ con $k > 0$, una sucesión de variables con D_1, \dots, D_k sus respectivos dominios. Una **restricción** C sobre Y es un subconjunto de $D_1 \times \dots \times D_k$. Para $k = 1$ decimos que la restricción es unaria, para $k = 2$, binaria.

Un **problema de satisfacción de restricciones (PSR)** es una sucesión finita de variables $X = x_1, \dots, x_n$ con dominios respectivos D_1, \dots, D_k , junto con un conjunto finito \mathcal{C} de restricciones, cada una de ellas sobre una subsucesión de X . Denotamos el PSR por $\langle \mathcal{C}; \mathcal{DE} \rangle$, donde $\mathcal{DE} = x_1 \in D_1, \dots, x_n \in D_n$, y llamamos a cada expresión $x \in D$ **expresión de dominio**.

Ahora definimos el concepto de solución de un PSR. Intuitivamente, una solución es una sucesión de valores para las variables que verifica todas las restricciones.

Definición. 1.7.2 Sea $\langle \mathcal{C}; \mathcal{DE} \rangle$ un PSR, con $\mathcal{DE} = x_1 \in D_1, \dots, x_n \in D_n$. Diremos que una n -tupla $(d_1, \dots, d_n) \in D_1 \times \dots \times D_k$ **satisface** una restricción $C \in \mathcal{C}$ sobre las variables x_{i_1}, \dots, x_{i_m} si $(d_{i_1}, \dots, d_{i_m}) \in C$. Decimos que una n -tupla $(d_1, \dots, d_n) \in D_1 \times \dots \times D_k$ es una **solución** a $\langle \mathcal{C}; \mathcal{DE} \rangle$ si satisface cada restricción $C \in \mathcal{C}$. Si un PSR tiene una solución, entonces diremos que es **consistente**, en otro caso, **inconsistente**.

Hay que hacer notar, con respecto a las definiciones anteriores, que no se hace referencia a su sintaxis. Sin embargo, es necesario definir las en un lenguaje específico, escogiendo una representación adecuada. En esa representación, está implícito el hecho de que cada restricción es un subconjunto del producto cartesiano de los dominios de las variables.

En [46] encontramos, entre otros, el siguiente ejemplo para ilustrar los conceptos que acabamos de definir.

Ejemplo. 1.7.3 Consideremos la sucesión de variables x, y, z, u en el conjunto de los números naturales y las tres restricciones siguientes sobre ellas: $x^3 + y^3 + z^3 + u^3 = 100, x < u, x + y = z$. De acuerdo con la notación anterior, expresamos este PSR como $\langle x^3 + y^3 + z^3 + u^3 = 100, x < u, x + y = z; x \in N, y \in N, z \in N, u \in N \rangle$, donde N denota el conjunto de los números naturales. La sucesión $(1, 2, 3, 4)$ es una solución de este PSR, ya que satisface todas las restricciones. La restricción $x < y$ está definida sobre la subsucesión x, y de la sucesión x, y, z, u . Pero la sucesión z, y no es una subsucesión de x, y, z, u , así que la expresión $z = y + 2$ no puede considerarse una restricción sobre z, y . Desde luego, si se reescribe como $y + 2 = z$, es una restricción sobre y, z , y por tanto, una restricción para el PSR.

1.8. Cálculo de Conexión de Regiones (RCC)

El Razonamiento Espacial Cualitativo (QSR ó REC) [60] es una alternativa a la manera cualitativa de representar el conocimiento espacial, es decir, sin usar valores numéricos. En su lugar, la información espacial se representa

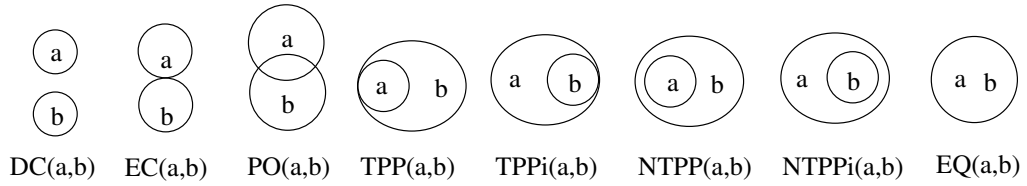


Figura 1.9: Relaciones de RCC8

usando un vocabulario consistente en un número finito de relaciones. La representación cualitativa se ajusta más a la manera en que los seres humanos representan y comunican el conocimiento espacial, así como las relaciones entre entidades espaciales. Hay que añadir que, aunque no se utilicen valores numéricos, el razonamiento cualitativo no es inexacto. Si es necesario, pueden hacerse distinciones, sin más que elegir el nivel de granularidad, el conjunto de relaciones que se usarán para describir una situación espacial. Cuando se desarrolla un formalismo cualitativo, es una práctica común elegir un conjunto JEPD de relaciones base que define cierto nivel de granularidad.

El Cálculo de Conexión de Regiones (RCC) [15] es una teoría axiomatizada de primer orden para la representación topológica de las relaciones entre *entidades espaciales*, que en este caso son conjuntos regulares no vacíos. Como en el razonamiento y la representación espacial cualitativos, nos interesa trabajar sobre un conjunto de relaciones base, elegimos en primer lugar la relación de conexión $\mathcal{C}(x, y)$, que se interpreta como “*las clausuras de x e y se cortan*”. Las ocho relaciones $\{DC, EC, PO, EQ, TPP, NTPP, TPPi, NTPPi\}$ mostradas en la figura 1.9, forman un conjunto de relaciones base que constituyen un conjunto JEPD y constituyen las relaciones base del cálculo relacional RCC8, que ha sido estudiado en profundidad por J. Renz y B. Nebel en [61].

Los axiomas de RCC son los dos axiomas básicos sobre \mathcal{C} , que expresan las propiedades reflexiva y simétrica de la conexión,

$$A_1 := \forall x[\mathcal{C}(x, x)] \quad A_2 := \forall x, y[\mathcal{C}(x, y) \rightarrow \mathcal{C}(y, x)],$$

junto con los axiomas que definen las principales relaciones espaciales (véase la figura 1.10), además de otros axiomas que hacen uso de las relaciones derivadas de \mathcal{C} . Estos últimos aparecen en la siguiente definición⁴ que determina el concepto general de modelo de RCC, (véase [49]).

Definición. 1.8.1 *Un modelo del Cálculo de Conexión de Regiones consta de los siguientes elementos. Un conjunto R , un elemento $u \in R$, un conjunto unitario $\{n\}$ ⁵ disjunto con R , una operación $\text{compl} : R \setminus \{u\} \mapsto R \setminus \{u\}$, las operaciones binarias $\text{sum} : R \times R \mapsto R$, y $\text{prod} : R \times R \mapsto R \cup \{n\}$, y la relación binaria primitiva \mathcal{C} en*

⁴De hecho, los axiomas adicionales que aparecen en esa definición no son utilizados en el resto de la memoria (como justificamos en la introducción).

⁵En elemento u representará el universo y el conjunto unitario $\{n\}$ la región vacía.

R. Estos elementos deben verificar los siguientes axiomas:

$$\text{R1. } (\forall x \in R) \mathcal{C}(x, x)$$

$$\text{R2. } (\forall x, y \in R) [\mathcal{C}(x, y) \longrightarrow \mathcal{C}(y, x)]$$

$$\text{R3. } (\forall x \in R) \mathcal{C}(x, \mathbf{u})$$

$$\text{R4a. } (\forall x \in R)(\forall y \in R \setminus \{\mathbf{u}\}) [\mathcal{C}(x, \text{compl } y) \longleftrightarrow \neg \text{NTPP}(x, y)]$$

$$\text{R4b. } (\forall x \in R)(\forall y \in R \setminus \{\mathbf{u}\}) [\mathcal{O}(x, \text{compl } y) \longleftrightarrow \neg \text{P}(x, y)]$$

$$\text{R5. } (\forall x, y, z \in R) [\mathcal{C}(x, \text{sum}(y, z)) \longleftrightarrow \mathcal{C}(x, y) \vee \mathcal{C}(x, z)]$$

$$\text{R6. } (\forall x, y, z \in R) [\text{prod}(y, z) \in R \mapsto [\mathcal{C}(x, \text{prod}(y, z)) \leftrightarrow (\exists w \in R)[\text{P}(w, y) \wedge \text{P}(w, z) \wedge \mathcal{C}(x, w)]]]]$$

$$\text{R7. } (\forall x, y \in R)[\text{prod}(x, y) \in R \leftrightarrow \mathcal{O}(x, y)]$$

El sistema RCC original contiene un axioma adicional:

$$(\forall x \in R)(\exists y \in R) \text{NTPP}(y, x)$$

Sin embargo, en [20] se demuestra que este último es redundante.

El cálculo RCC5 se define sobre el conjunto $\{\text{DR}, \text{PP}, \text{P0}, \text{PPi}, \text{EQ}\}$. En [44] se analiza experimentalmente cuál de los dos cálculos, RCC8 o RCC5, es más adecuado para el razonamiento espacial cualitativo, desde el punto de vista psicológico, y los autores se decantan por RCC8.

En adelante, denotaremos por \mathcal{R}_{RCC} el conjunto de relaciones definidas en RCC.

Se verifica el siguiente teorema debido a N. Gotts [33]:

Teorema 1.8.2 *Si Ω es un espacio T_3 conexo no trivial, la expansión natural de Ω al lenguaje completo es un modelo de RCC.*

La “expansión natural” de Ω se define a continuación.

1.8.1. Modelos de RCC

Para estudiar la potencia expresiva de esta teoría, debemos analizar qué clase de modelos representan nuestro objeto de estudio. En adelante, todo modelo que consideremos será T_3 conexo no trivial. Concretamente, es necesario considerar tres clases de modelos: la clase de todos los modelos (de acuerdo con la definición clásica de la lógica de primer orden), la clase de los modelos

$A_{DC} :$	$DC(\mathbf{x}, \mathbf{y}) \leftrightarrow \neg C(\mathbf{x}, \mathbf{y})$	(Desconexión)
$A_P :$	$P(\mathbf{x}, \mathbf{y}) \leftrightarrow \forall z[C(\mathbf{z}, \mathbf{x}) \rightarrow C(\mathbf{z}, \mathbf{y})]$	(Parte de)
$A_{PP} :$	$PP(\mathbf{x}, \mathbf{y}) \leftrightarrow P(\mathbf{x}, \mathbf{y}) \wedge \neg P(\mathbf{y}, \mathbf{x})$	(Parte Propia)
$A_{EQ} :$	$EQ(\mathbf{x}, \mathbf{y}) \leftrightarrow P(\mathbf{x}, \mathbf{y}) \wedge P(\mathbf{y}, \mathbf{x})$	(Igualdad)
$A_O :$	$O(\mathbf{x}, \mathbf{y}) \leftrightarrow \exists z[P(\mathbf{z}, \mathbf{x}) \wedge P(\mathbf{z}, \mathbf{y})]$	(Solapamiento)
$A_{DR} :$	$DR(\mathbf{x}, \mathbf{y}) \leftrightarrow \neg O(\mathbf{x}, \mathbf{y})$	(Discreto)
$A_{PO} :$	$PO(\mathbf{x}, \mathbf{y}) \leftrightarrow O(\mathbf{x}, \mathbf{y}) \wedge \neg P(\mathbf{x}, \mathbf{y}) \wedge \neg P(\mathbf{y}, \mathbf{x})$	(Solapamiento Parcial)
$A_{EC} :$	$EC(\mathbf{x}, \mathbf{y}) \leftrightarrow C(\mathbf{x}, \mathbf{y}) \wedge \neg O(\mathbf{x}, \mathbf{y})$	(Conexión Externa)
$A_{TPP} :$	$TPP(\mathbf{x}, \mathbf{y}) \leftrightarrow PP(\mathbf{x}, \mathbf{y}) \wedge \exists z[EC(\mathbf{z}, \mathbf{x}) \wedge EC(\mathbf{z}, \mathbf{y})]$	(PP Tangencial)
$A_{NTPP} :$	$NTPP(\mathbf{x}, \mathbf{y}) \leftrightarrow PP(\mathbf{x}, \mathbf{y}) \wedge \neg \exists z[EC(\mathbf{z}, \mathbf{x}) \wedge EC(\mathbf{z}, \mathbf{y})]$	(PP No Tangencial)

Figura 1.10: Axiomas de RCC

topológicos, y el espacio \mathbb{R}^n , donde las constantes se interpretan como los conjuntos regulares en cuestión. Formalmente, se le puede asociar a cada espacio topológico una estructura en el lenguaje de RCC, de la siguiente forma:

Definición. 1.8.3 Sea Ω un espacio topológico, y \mathbf{X} un conjunto finito de constantes. Una estructura Θ es un **modelo topológico** sobre Ω si es de la forma

$$\langle \mathcal{R}(\Omega)_{/\sim}, C_{\Theta}, \{a_{\Theta} : a \in \mathbf{X}\} \rangle$$

donde $\mathcal{R}(\Omega)$ es la clase de los conjuntos regulares no vacíos, \sim es la relación de equivalencia "las clausuras coinciden".

La relación de equivalencia es necesaria para que el axioma A_P sea un axioma de extensionalidad de P con respecto a C . Al efectuar el cociente, la relación mereológica EQ coincide con la igualdad. C_{Θ} es la interpretación de la conexión, C , y $a_{\Theta} \in \mathcal{R}(\Omega)_{/\sim}$, no vacíos, para cada $a \in \mathbf{X}$.

Teorema. 1.8.4 La estructura Θ asociada a Ω , definida en 1.8.3, es modelo de los axiomas de RCC, y donde EQ^{Θ} coincide con la igualdad⁶.

Cada estructura del tipo de 1.8.3 se expande a una en el lenguaje completo de RCC, mediante la interpretación natural de las otras relaciones. Así, por ejemplo, como $C_{\Theta} = \{(a, b) : \Omega \models C(a, b)\}$, tendremos

$$P_{\Theta} = \{(a, b) : \forall c \in \Omega[\Omega \models C(c, a) \implies \Omega \models C(c, b)]\}.$$

Una consecuencia directa de 1.8.2 es el siguiente resultado:

⁶Aunque esta coincidencia no es, en principio, necesaria (pues se podría evitar si no consideramos la relación \sim), es muy útil para trabajar con las regiones.

Proposición. 1.8.5 *En un espacio topológico Ω que es T_3 , conexo no trivial se verifican las siguientes propiedades:*

$$\begin{array}{ll}
 \text{TPP}^\Omega \subseteq \text{PP}^\Omega & \text{NTPP}^\Omega \subseteq \text{PP}^\Omega \\
 \text{TPPi}^\Omega \subseteq \text{PPi}^\Omega & \text{NTPPi}^\Omega \subseteq \text{PPi}^\Omega \\
 \text{PO}^\Omega \subseteq \text{O}^\Omega & \text{PP}^\Omega \subseteq \text{P}^\Omega \\
 \text{EQ}^\Omega \subseteq \text{P}^\Omega & \text{EQ}^\Omega \subseteq \text{Pi}^\Omega \\
 \text{P}^\Omega \subseteq \text{O}^\Omega & \text{Pi}^\Omega \subseteq \text{O}^\Omega \\
 \text{EC}^\Omega \subseteq \text{C}^\Omega & \text{EC}^\Omega \subseteq \text{DR}^\Omega \\
 \text{DC}^\Omega \subseteq \text{DR}^\Omega &
 \end{array}$$

1.8.2. PSRs en RCC

El conocimiento sobre entidades o relaciones entre entidades espaciales, a menudo, se expresa en forma de restricciones. Como hemos dicho, las restricciones pueden ser 1-arias ó, en general, n -arias, según restrinjan el dominio de 1 ó de n variables. En RCC trabajamos con PSR binarios, es decir, donde sólo se utilizan restricciones binarias. Este tipo concreto de PSR se puede representar mediante una “red de restricciones”, un grafo etiquetado donde cada nodo lleva la etiqueta de una variable y cada eje la de una restricción binaria.

Si el dominio de las variables es finito, el problema se puede resolver por *backtracking* sobre el dominio ordenado de las variables. Esta técnica instancia las variables sucesivamente con valores del dominio ordenado, hasta que o bien todas las variables están instanciadas o bien se encuentra una inconsistencia, en cuyo caso la variable en curso se instancia con el siguiente valor de su dominio. Si todas las posibles instanciaciones de la variable en curso llevan a una inconsistencia, la variable anterior se convierte en variable en curso y el proceso se repite. Esta técnica es exponencial en el número de variables. Si el dominio de las variables es infinito hay que aplicar otros métodos.

Una forma de abordar PSRs con dominios de variables infinitos es usar restricciones sobre un conjunto finito de relaciones binarias. Se pueden formular PSRs binarios como álgebras de relación. Esto permite tratar de manera uniforme los PSRs binarios con dominios finitos e infinitos.

Álgebras de relación

Un álgebra de relación consiste en un conjunto de relaciones binarias cerrado bajo una serie de operaciones y que contiene unas relaciones particulares. Las operaciones son *unión* (\cup), *intersección* (\cap), *composición* (\circ), *simétrico* (\cdot^{-1}),

complementario ($\hat{\cdot}$). La definición de estas operaciones es la siguiente:

$$\begin{aligned}\forall xy : x(R \cup S)y &\leftrightarrow xRy \vee xSy \\ \forall xy : x(R \cap S)y &\leftrightarrow xRy \wedge xSy \\ \forall xy : x(R \circ S)y &\leftrightarrow \exists z(xRz \wedge zSy) \\ \forall xy : xR^{-1}y &\leftrightarrow yRx \\ \forall xy : x\hat{R}y &\leftrightarrow \neg xRy\end{aligned}$$

Como la unión de relaciones es igual a la disyunción de restricciones, escribiremos conjuntos de relaciones para denotar disyunciones de relaciones, por ejemplo, $\{R, S\}$ representará la disyunción de R y S . Son necesarias además tres relaciones básicas, la *relación vacía* \emptyset , que no verifica ningún par de elementos del dominio, la *relación universal* $*$ entre dos elementos cualesquiera del dominio, y la *relación identidad* Id que mantiene cualquier elemento consigo mismo. Se supone que el conjunto de restricciones contiene una restricción para cualquier par de variables, es decir, si no hay información sobre la relación existente entre dos variables, entonces la restricción $x * y$ está implícita en el conjunto de restricciones. Otra hipótesis que mantenemos es que si se tiene la restricción xRy , entonces su inversa $yR^{-1}x$ también está presente. Determinar la consistencia de un CSP con dominios infinitos es en general indecidible. Un método parcial para determinar la inconsistencia de un CSP es el algoritmo de *consistencia por caminos* [60].

Interés particular tienen las álgebras de relación basadas en conjuntos finitos de relaciones que son JEPD (*Jointly Exhaustive and Pairwise Disjoint*), es decir, “exhaustivas y disjuntas dos a dos”. Las relaciones JEPD se llaman relaciones atómicas, básicas o relaciones base. Que las relaciones sean exhaustivas y disjuntas dos a dos, entre pares de elementos del dominio de trabajo, significa que dos objetos cualesquiera del dominio están relacionados mediante una (y solo una) de las relaciones de **Rels**.

Como consecuencia, los conjuntos de relaciones JEPD se utilizan para representar el conocimiento definido con respecto a un nivel de granularidad dado.

Por otra parte, el conocimiento indefinido se especifica mediante disyunciones de posibles relaciones básicas. Dado un conjunto de relaciones base A , el conjunto potencia 2^A , el conjunto de todas las posibles disyunciones de relaciones de A , es un álgebra de relaciones si es cerrado bajo composición. En esas álgebras de relación, la relación universal es la disyunción de todas las relaciones base.

Tablas de composición

La composición de relaciones base tiene que calcularse usando la semántica de las relaciones. La composición de disyunciones de relaciones base se puede obtener calculando la disyunción de la composición de las relaciones. Usualmente, las composiciones de las relaciones base se calculan y almacenan en *tablas de composición*.

Definición. 1.8.6 *Dado un conjunto \mathbf{Rels} de relaciones, una **tabla de composición** es una aplicación $CT : \mathbf{Rels} \times \mathbf{Rels} \rightarrow 2^{\mathbf{Rels}}$, es decir, CT asigna a cada par $\langle R1, R2 \rangle$, donde $R1, R2$ son elementos de \mathbf{Rels} , un subconjunto $\mathbf{S} \subseteq \mathbf{Rels}$, llamado **composición de R y S** que notaremos $CT(R, S)$.*

De manera intuitiva, si dos elementos del dominio a, b están relacionados mediante $R1$ y, a su vez b está relacionado mediante $R2$ con un tercer elemento c , el conjunto $CT(R, S)$ estará constituido por las posibles relaciones entre a y c ,

$$CT(R, S)(a, c) \iff \exists b[R(a, b) \wedge S(b, c)]$$

Usualmente, suponemos que los elementos de \mathbf{Rels} son relaciones que constituyen un JEPD. En estas condiciones, cualquier combinación Booleana de relaciones es equivalente a una disyunción de elementos de \mathbf{Rels} . Evidentemente, el significado de CT depende del contexto en el que se emplee.

Refinamientos

Definición. 1.8.7 (Refinamiento de una relación) Diremos que una relación R es un refinamiento de una relación S si y sólo si $R \subseteq S$.

Por ejemplo, dada una disyunción de relaciones $\{R_1, R_2, R_3\}$, la relación $\{R_1, R_2\}$ es un refinamiento de la primera. Esta definición se extiende al conjunto de restricciones.

Definición. 1.8.8 (Refinamiento de un conjunto de restricciones) Un conjunto de restricciones Θ es un refinamiento de otro Θ' si contienen las mismas variables y para cada par de variables x, y se verifica que si $xR'y \in \Theta'$ y $xRy \in \Theta$, entonces $R' \subseteq R$. Se dirá que Θ' es un **refinamiento consistente** de Θ si Θ' es un refinamiento de Θ y ambos son consistentes.

Hay un tipo de refinamiento que resulta de gran interés y que definimos a continuación.

Definición. 1.8.9 Un **escenario consistente** Θ_s de un conjunto de restricciones Θ es un refinamiento consistente de Θ donde todas las restricciones de Θ_s son elementales, es decir, aparecen sólo relaciones base.

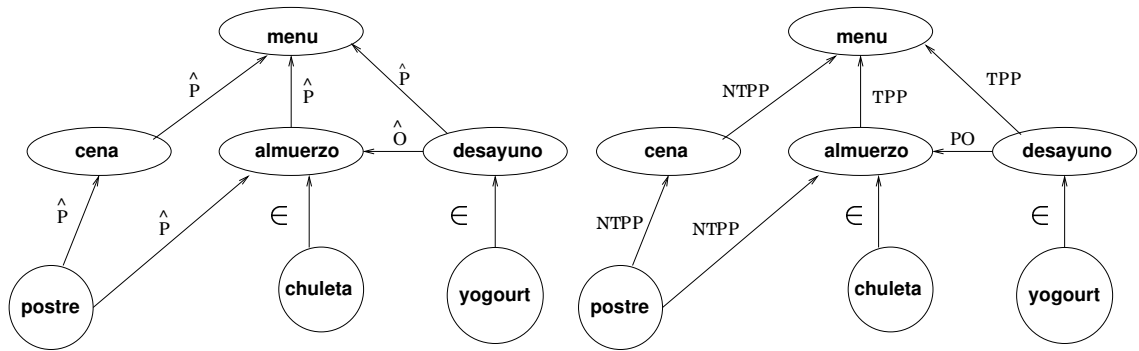


Figura 1.11: Grafo inicial del PSR (izda.) y solución (dcha.)

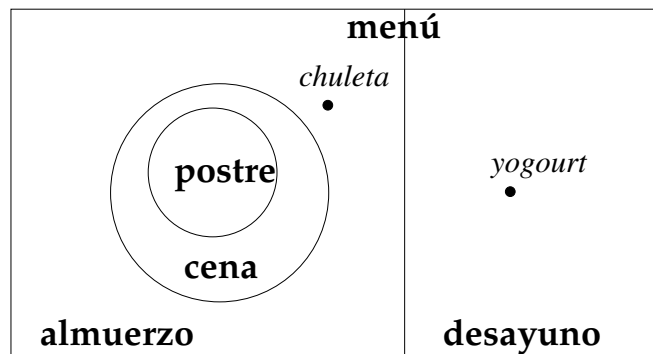


Figura 1.12: Escenario consistente

Consideremos el ejemplo cuyo grafo PSR(Σ) se da en la fig. 1.11, a la izquierda, donde por \hat{P} denotamos la disyunción de P en función de RCC8. Si hacemos una proyección reticular en RCC8, una solución al problema PSR aparece a la derecha en la fig. 1.11.

El escenario consistente asociado a un PSR estará representado espacialmente por regiones regulares del plano (no necesariamente conectadas), ver figura 1.12. La existencia de tal escenario se prueba en los dos siguientes teoremas que extraemos de [59].

Teorema. 1.8.10 *Todo conjunto consistente de fórmulas espaciales se puede realizar en cualquier dimensión $n \geq 1$, donde las regiones son conjuntos de regiones de dimensión n con fronteras poligonales.*

Teorema. 1.8.11 *Todo conjunto consistente de fórmulas espaciales se puede realizar con regiones internamente conectadas en cualquier dimensión $n \geq 3$.*

1.9. Topología de los Cells

A la hora de caracterizar ciertas relaciones espaciales, en el desarrollo de este trabajo, hemos necesitado de formalismos utilizados por la topología algebraica. M. Egenhofer, en [22], los utiliza para caracterizar las relaciones espaciales entre regiones con cierta regularidad. Vamos a introducir brevemente los principales conceptos, necesarios para la caracterización y el estudio de las relaciones espaciales que estudiaremos, dentro del marco de la teoría RCC en el capítulo 4.

La topología algebraica está basada en unos objetos geométricos primarios, los *cells*, que se definen para las diferentes dimensiones espaciales. Un **0-cell** es un nodo (el objeto 0-dimensional minimal); un **1-cell** es el enlace entre dos 0-cells distintos; un **2-cell** es el área descrita por una secuencia cerrada de tres 1-cells disjuntos. Una $(0 \dots n)$ -**cara** de un n-cell A es cualquier $(0 \dots n)$ -cell que esté contenido en A. Técnicamente los cells se definen de manera inductiva como sigue [21]:

Definición. 1.9.1 Los **cells** en $\mathbb{R}^1 = \mathbb{R}$ son los puntos $\{r\}$ y los intervalos (a, b) . Sea $C \subseteq \mathbb{R}^n$ un cell; si $f, g : C \rightarrow \mathbb{R}$ son funciones continuas tales que $f < g$ en C , entonces

$$(f, g) := \{(x, r) \in C \times \mathbb{R} : f(x) < r < g(x)\}$$

es un cell en \mathbb{R}^{n+1} . También, dada una función continua $f : C \rightarrow \mathbb{R}$, los conjuntos

$$\Gamma(f)$$

$$(-\infty, f) := \{(x, r) \in C \times \mathbb{R} : r < f(x)\}$$

$$(f, +\infty) := \{(x, r) \in C \times \mathbb{R} : f(x) < r\}$$

son cells en \mathbb{R}^{n+1} ; finalmente, $C \times \mathbb{R} \subseteq \mathbb{R}^{n+1}$ es una figura.

Este modelo de datos espaciales difiere del modelo simplicial en una propiedad principalmente: los símlices tienen “cierre convexo”, mientras los cells pueden tener interior con forma arbitraria. Por otro lado, un n-cell tiene la misma dimensión que el espacio que lo contiene (si el n-cell existe en él), pero no existe un homeomorfismo del n-cell en un espacio de dimensión $n - 1$.

Las características topológicas que vamos a considerar son la clausura, el interior, la frontera y el exterior de un cell.

Definición. 1.9.2 La clausura de un n-cell A, la denotamos por \bar{A} , es el conjunto de todas las r-caras (c_r) de A, donde $0 \leq r \leq n$, es decir:

$$\bar{A} = \bigcup_{r=0}^n (c_r) \in A$$

Definición. 1.9.3 El conjunto frontera de un n -cell A , que denotamos por ∂A , es el conjunto de todas las r -caras (c_r) de A , donde $0 \leq r \leq (n - 1)$, es decir:

$$\partial A = \bigcup_{r=0}^{n-1} (c_r) \in A$$

Definición. 1.9.4 El interior de un n -cell A , que denotamos por A° , es la diferencia entre la clausura y la frontera de A :

$$A^\circ = \bar{A} - \partial A$$

Definición. 1.9.5 El exterior de un n -cell A , que denotamos por A^- , es el conjunto de todos los cells del universo \mathcal{U} que no son elementos de la clausura de A :

$$A^- = \mathcal{U} - \bar{A}$$

A partir de los cells, como objetos elementales, se pueden formar objetos mas complejos, los llamados *complejos celulares*. Un cell complejo C será la unión de dos o más cells A_1, \dots, A_x . Las operaciones con cells complejos se definen en términos de las operaciones con cells. De estas definiciones se sigue que :

- El interior, la frontera y el exterior de un cell son mutuamente excluyentes
- La unión coincide con el universo

En el contexto de este trabajo nos interesan, al igual que en [22], los cells complejos que son homogéneos, n -dimensionales y no separados en partes disjuntas no vacías. Las características de los puntos, líneas y regiones más usadas son las siguientes:

- Una región es un 2-complejo en \mathbb{R}^2
- Una línea es una sucesión de 1-complejos conectados en \mathbb{R}^2 tales que ni se cruzan entre sí ni forman bucles
- Un punto es un 0-cell en \mathbb{R}^2

Una región puede tener la frontera conexa o no, lo cual dará lugar a regiones sin y con agujeros respectivamente. Una línea será simple o compleja según tenga dos o más extremos desconectados.

Las relaciones binarias entre dos cells que M. Egenhofer considera en su trabajo [22], se basan en las posibles intersecciones entre sus interiores, fronteras y exteriores. Esto da lugar a nueve datos fundamentales que describen las

nueve posibles relaciones entre dos cells. Así, por ejemplo, $\partial A \cap B^-$ indicará la intersección entre la frontera del cell A y la clausura del cell B . Vamos a utilizar una matriz 3×3 para representar la relación entre dos cells A y B , como sigue:

$$R(A, B) = \begin{pmatrix} A^\circ \cap B^\circ & A^\circ \cap \partial B & A^\circ \cap B^- \\ \partial A \cap B^\circ & \partial A \cap \partial B & \partial A \cap B^- \\ A^- \cap B^\circ & A^- \cap \partial B & A^- \cap B^- \end{pmatrix}$$

Para indicar que alguna de las intersecciones es no vacía, pondremos en su lugar un 1 y en caso contrario pondremos un 0. Así tendremos una forma cómoda y abreviada que resume las relaciones con exactitud.

Capítulo 2

Razonamiento automático en RCC

El propósito de este capítulo es el tratamiento automatizado de las principales herramientas utilizadas en las aplicaciones de la teoría RCC. Esto facilitará el estudio de las posibles extensiones de tal teoría introduciendo e interpretando en nuestro caso nuevas relaciones (de indefinición y conexión). Este tratamiento se llevará a cabo con la ayuda del demostrador automático de teoremas OTTER y el programa generador de modelos MACE4. Al trabajar en la formalización de un campo de conocimiento tan intuitivo como el espacial, es aconsejable utilizar herramientas que nos aseguren la objetividad, imprescindible a la hora de obtener pruebas de nuevos resultados: la demostración automática evita el uso de propiedades espaciales intuitivas no demostrables a partir de la teoría base.

Para analizar las relaciones espaciales utilizaremos las *tablas de composición*, técnica que proporciona un mecanismo de inferencia muy eficiente. En [7] los autores proponen, entre otros objetivos, encontrar las condiciones generales bajo las que una tabla de composición para un cálculo relacional proporciona un sistema de inferencia completo y tratable para una teoría.

Las tablas de composición suelen construirse a mano, pero deben verificarse, de manera que cada salida de la tabla constituya un teorema. En [57], los autores hacen uso de OTTER para verificar la tabla de composición de RCC8; ese objetivo también es alcanzado en [53]. En cualquier caso, no solo hay que demostrar que el resultado de cada composición es la disyunción correspondiente: además sería necesario encontrar un modelo que verifique cada una de las relaciones que componen esa disyunción, para así comprobar que la composición es el resultado “más fuerte posible”, hecho que es fácilmente constatable en \mathbb{R}^2 , para el caso de RCC8. De hecho, en la primera parte de este capítulo demostraremos que RCC demuestra que RCC8 y RCC5 forman JEPDs.

Esta técnica es la que hemos usado en este trabajo para el análisis del retículo de las relaciones de RCC y de algunas de sus extensiones. Para trabajar con

OTTER expresaremos los axiomas/definiciones de RCC como en la caja 1.

```

%% Axiomas básicos DE RCC
all x c(x,x).
all x y (c(x,y) <-> c(y,x)).
%% Axiomas/Definiciones de las relaciones de RCC
all x y (dc(x,y) <-> -c(x,y)).
all x y (iq(x,y) <-> (p(x,y) & p(y,x))).
all x y (po(x,y) <-> (o(x,y) & -p(x,y) & -p(y,x))).
all x y (ec(x,y) <-> (c(x,y) & -o(x,y))).
all x y (tpp(x,y) <-> (pp(x,y) &
                        (exists z (ec(z,x) &
                                    ec(z,y)))).
all x y (ntpp(x,y) <-> (pp(x,y) &
                        -(exists z (ec(z,x) &
                                    ec(z,y)))).
all x y (tppi(x,y) <-> tpp(y,x)).
all x y (ntppi(x,y) <-> ntpp(y,x)).
all x y (p(x,y) <-> (all z (c(z,x) -> c(z,y)))).
all x y (pp(x,y) <-> (p(x,y) & -p(y,x))).
all x y (o(x,y) <-> (exists z (p(z,x) & p(z,y)))).
all x y (dr(x,y) <-> -o(x,y)).

```

La forma clausal de RCC producida por OTTER aparece en la siguiente caja , y es la que utilizaremos como teoría base para los experimentos con RCC.

```

0 [] -dc(X1,X2) | -c(X1,X2) .
0 [] dc(X1,X2) | c(X1,X2) .
0 [] -iq(X3,X4) | p(X3,X4) .
0 [] -iq(X3,X4) | p(X4,X3) .
0 [] iq(X3,X4) | -p(X3,X4) | -p(X4,X3) .
0 [] -po(X5,X6) | o(X5,X6) .
0 [] -po(X5,X6) | -p(X5,X6) .
0 [] -po(X5,X6) | -p(X6,X5) .
0 [] po(X5,X6) | -o(X5,X6) | p(X5,X6) | p(X6,X5) .
0 [] -ec(X7,X8) | c(X7,X8) .
0 [] -ec(X7,X8) | -o(X7,X8) .
0 [] ec(X7,X8) | -c(X7,X8) | o(X7,X8) .
0 [] -tpp(X9,X10) | pp(X9,X10) .

```

```

0 [] -tpp(X9, X10) | ec($f1(X9, X10), X9) .
0 [] -tpp(X9, X10) | ec($f1(X9, X10), X10) .
0 [] tpp(X9, X10) | -pp(X9, X10) | -ec(X11, X9) | -ec(X11, X10) .
0 [] -ntpp(X12, X13) | pp(X12, X13) .
0 [] -ntpp(X12, X13) | -ec(X14, X12) | -ec(X14, X13) .
0 [] ntp(X12, X13) | -pp(X12, X13) | ec($f2(X12, X13), X12) .
0 [] ntp(X12, X13) | -pp(X12, X13) | ec($f2(X12, X13), X13) .
0 [] -tppi(X15, X16) | tpp(X16, X15) .
0 [] tppi(X15, X16) | -tpp(X16, X15) .
0 [] -ntppi(X17, X18) | ntp(X18, X17) .
0 [] ntp(X17, X18) | -ntpp(X18, X17) .
0 [] -p(X19, X20) | -c(X21, X19) | c(X21, X20) .
0 [] p(X19, X20) | c($f3(X19, X20), X19) .
0 [] p(X19, X20) | -c($f3(X19, X20), X20) .
0 [] -pp(X22, X23) | p(X22, X23) .
0 [] -pp(X22, X23) | -p(X23, X22) .
0 [] pp(X22, X23) | -p(X22, X23) | p(X23, X22) .
0 [] -o(X24, X25) | p($f4(X24, X25), X24) .
0 [] -o(X24, X25) | p($f4(X24, X25), X25) .
0 [] o(X24, X25) | -p(X26, X24) | -p(X26, X25) .
0 [] -dr(X27, X28) | -o(X27, X28) .
0 [] dr(X27, X28) | o(X27, X28) .

```

2.1. RCC-8 y RCC-5

En este apartado vamos a presentar las demostraciones formales de las propiedades básicas de RCC-8 y RCC-5. La mayoría de los conceptos que utilizaremos se definen o mencionan en la sección 1.8 de este trabajo.

Teorema. 2.1.1 *En RCC se demuestra que el conjunto de relaciones RCC-8 forma un sistema JEPD.*

Demostración: Para demostrar que son disjuntas habrá que demostrar que las siguientes fórmulas son teoremas de RCC:

1. $DC(x, y) \rightarrow [\neg EQ(x, y) \wedge \neg EC(x, y) \wedge \neg PO(x, y) \wedge \neg TPP(x, y) \wedge \neg TPPi(x, y) \wedge \neg NTTP(x, y) \wedge \neg NTTPi(x, y)]$
2. $EQ(x, y) \rightarrow [\neg EC(x, y) \wedge \neg PO(x, y) \wedge \neg TPP(x, y) \wedge \neg TPPi(x, y) \wedge \neg NTTP(x, y) \wedge \neg NTTPi(x, y)]$
3. $EC(x, y) \rightarrow [\neg PO(x, y) \wedge \neg TPP(x, y) \wedge \neg TPPi(x, y) \wedge \neg NTTP(x, y) \wedge \neg NTTPi(x, y)]$

4. $PO(x, y) \rightarrow [\neg TPP(x, y) \wedge \neg TPPi(x, y) \wedge \neg NTPP(x, y) \wedge \neg NTPPi(x, y)]$
5. $TPP(x, y) \rightarrow [\neg TPPi(x, y) \wedge \neg NTPP(x, y) \wedge \neg NTPPi(x, y)]$
6. $TPPi(x, y) \rightarrow [\neg NTPP(x, y) \wedge \neg NTPPi(x, y)]$
7. $NTPP(x, y) \rightarrow \neg NTPPi(x, y)$

Así pues, la demostración de que son disjuntas consiste en realizar 28 pequeñas pruebas, por ejemplo, del tipo

$$\forall xy(DC(x, y) \rightarrow \neg EQ(x, y))$$

Veamos esta última, como muestra. En el fichero de entrada incluimos los axiomas de RCC y las definiciones de las relaciones de RCC [\[1\]](#).

En el conjunto soporte ponemos la negación del resultado a probar, es decir, afirmamos que existen dos conjuntos que satisfacen a la vez los predicados DC y EQ, como en [\[2\]](#).

$dc(a, b) .$ $iq(a, b) .$	2
------------------------------	-------------------

OTTER devuelve la refutación que mostramos en [\[3\]](#). Ésta utiliza la reflexividad de la relación de conexión C .

<pre> 1 [] -dc(A, B) -c(A, B) . 2 [] -iq(A, B) p(A, B) . 5 [] -p(A, B) -c(C, A) c(C, B) . 8 [] dc(a, b) . 9 [] iq(a, b) . 1 c(A, A) . 13 [ur, 8, 1] -c(a, b) . 15 [hyper, 9, 2] p(a, b) . 17 [ur, 13, 5, 10] -p(a, b) . 18 [binary, 17.1, 15.1] \$F.</pre>	3
--	-------------------

Por último, faltaría demostrar que forman un conjunto exhaustivo, es decir, $\forall x y[DC(x, y) \vee EQ(x, y) \vee EC(x, y) \vee PO(x, y) \vee TPP(x, y) \vee TPPi(x, y) \vee NTPP(x, y) \vee NTPPi(x, y)]$

Para probarlo, en el fichero de entrada incluimos las cláusulas de [\[1\]](#) y, en el conjunto soporte, la negación de la cláusula a probar, es decir, que existen dos constantes a, b que no están relacionadas por ninguna de las relaciones de RCC8, como en [\[4\]](#).

```

-dc (a, b) .
-iq (a, b) .
-ec (a, b) .
-po (a, b) .
-tpp (a, b) .
-tppi (a, b) .
-ntpp (a, b) .
-ntppi (a, b) .

```

4

La refutación obtenida por OTTER es la siguiente:

```

4 [] iq(A,B) | -p(A,B) | -p(B,A) .
8 [] po(A,B) | -o(A,B) | p(A,B) | p(B,A) .
11 [] ec(A,B) | -c(A,B) | o(A,B) .
15 [] tpp(A,B) | -pp(A,B) | -ec(C,A) | -ec(C,B) .
18 [] ntp(A,B) | -pp(A,B) | ec($f2(A,B),A) .
19 [] ntp(A,B) | -pp(A,B) | ec($f2(A,B),B) .
21 [] tppi(A,B) | -tpp(B,A) .
23 [] ntp(A,B) | -ntpp(B,A) .
28 [] pp(A,B) | -p(A,B) | p(B,A) .
37 [] -dc(a,b) .
38 [] -iq(a,b) .
39 [] -ec(a,b) .
40 [] -po(a,b) .
41 [] -tpp(a,b) .
42 [] -tppi(a,b) .
43 [] -ntpp(a,b) .
44 [] -ntppi(a,b) .
45 [] dc(A,B) | c(A,B) .
48 [ur,42,21] -tpp(b,a) .
50 [ur,44,23] -ntpp(b,a) .
51 [hyper,45,37] c(a,b) .
52 [hyper,51,11,unit_del,39] o(a,b) .
55 [hyper,52,8,unit_del,40] p(a,b) | p(b,a) .
86 [hyper,55,28,factor_simp] p(b,a) | pp(a,b) .
89 [hyper,55,28,factor_simp] p(a,b) | pp(b,a) .
177 [hyper,86,19,unit_del,43] p(b,a) | ec($f2(a,b),b) .
178 [hyper,86,18,unit_del,43] p(b,a) | ec($f2(a,b),a) .
179 [hyper,89,19,unit_del,50] p(a,b) | ec($f2(b,a),a) .
180 [hyper,89,18,unit_del,50] p(a,b) | ec($f2(b,a),b) .
206 [hyper,179,4,178,unit_del,38] ec($f2(b,a),a) | ec($f2(a,b),a) .
207 [hyper,179,4,177,unit_del,38] ec($f2(b,a),a) | ec($f2(a,b),b) .
214 [hyper,180,4,178,unit_del,38] ec($f2(b,a),b) | ec($f2(a,b),a) .
215 [hyper,180,4,177,unit_del,38] ec($f2(b,a),b) | ec($f2(a,b),b) .
296 [hyper,207,15,86,206,unit_del,41,factor_simp] ec($f2(b,a),a)
    | p(b,a) .
303 [hyper,296,4,179,unit_del,38,factor_simp] ec($f2(b,a),a) .
358 [hyper,214,15,89,303,unit_del,48] ec($f2(a,b),a) | p(a,b) .
374 [hyper,358,4,178,unit_del,38,factor_simp] ec($f2(a,b),a) .
436 [hyper,215,15,89,303,unit_del,48] ec($f2(a,b),b) | p(a,b) .
449 [hyper,436,4,177,unit_del,38,factor_simp] ec($f2(a,b),b) .
450 [hyper,449,15,86,374,unit_del,41] p(b,a) .
451 [hyper,449,15,89,374,unit_del,48] p(a,b) .
465 [ur,450,4,38] -p(a,b) .
466 [binary,465.1,451.1] $F .

```

5

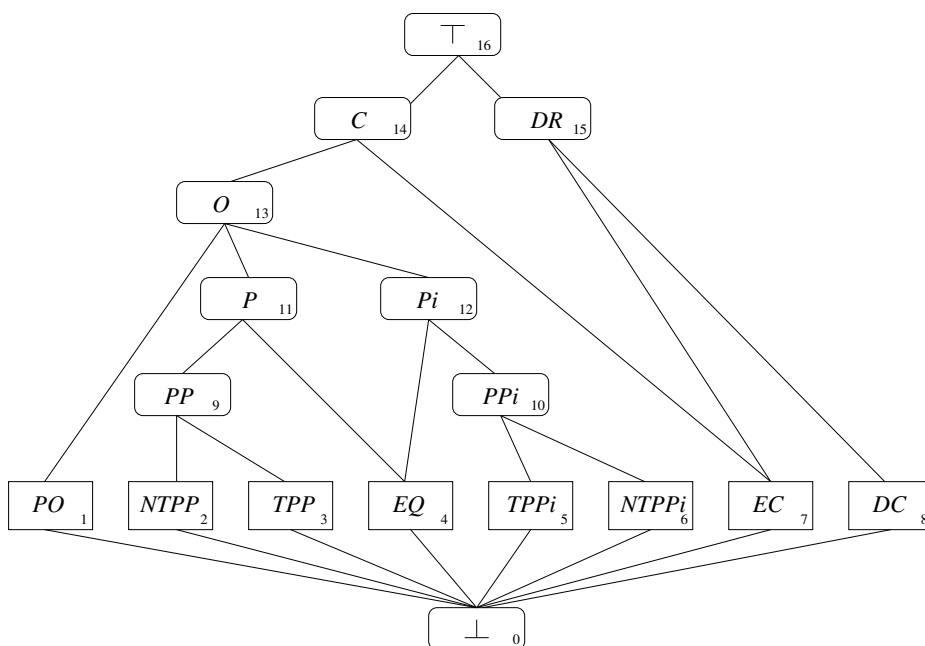


Figura 2.1: Retículo de las relaciones de RCC

Análogamente, se puede demostrar el siguiente resultado para RCC5.

Teorema. 2.1.2 *El conjunto de relaciones RCC-5 forma un sistema JEPD.*

2.2. El Retículo de las relaciones en RCC

La estructura clave asociada a RCC que utilizaremos en este trabajo es la que considera a las relaciones de RCC como una *ontología* de relaciones espaciales. En primer lugar, comprobaremos que tal estructura es demostrable en RCC. Aunque esta tarea sería posible realizarla sólo con OTTER, utilizaremos una técnica distinta para motivar el desarrollo posterior de este capítulo.

Teorema. 2.2.1 *En RCC es demostrable que el conjunto \mathcal{R}_{RCC} tiene la estructura de retículo cuyo diagrama de Hasse se ilustra en la figura 2.1.*

Demostración: La prueba que presentamos consiste en utilizar el generador de modelos automático, en nuestro caso MACE4, asociado con el demostrador OTTER. En el fichero de entrada, introducimos los axiomas que definen la estructura de retículo, [6]. Además, deben aparecer expresadas las conexiones (*and*, *or*) entre las relaciones de RCC (que presentamos en [7] y [8]).

Para facilitar el trabajo, a partir de este momento asignaremos a cada una de esas relaciones, que serán las constantes, un número de 0 a 16, empezando

de abajo hacia arriba y de izquierda a derecha en el gráfico de la figura 2.1, así, el 0 se identificará con \perp , el 16 con \top y las demás relaciones seguirán el siguiente código:

PO	NTPP	TPP	EQ	TPPi	NTPPi	EC	DC	PP	PPi	P	Pi	O	C	DR
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

De esta forma exigimos que se cumpla el Principio de Nombres Únicos cuando trabajemos con MACE4.

<pre>%conmutativas all x all y (or(x,y) = or(y,x)). all x all y (and(x,y) = and(y,x)). %asociativas all x all y all z (or(x, or(y,z)) = or(or(x,y), z)). all x all y all z (and(x, and(y,z)) = and(and(x,y), z)). %idempotencia all x (or(x,x) = x). all x (and(x,x) = x). %absorcion all x all y (x = or(x, (and(x,y))))). all x all y (x = and(x, (or(x,y))))).</pre>	6
---	---

<pre>16 = or(14,15). 13 = or(1,12). 15 = or(7,8). 12 = or(4,10). 14 = or(13,7). 11 = or(9,4). 13 = or(1,11). 10 = or(5,6). 13 = or(11,12). 9 = or(2,3).</pre>	7
--	---

<pre>and(1,11) = 0. and(3,15) = 0. and(1,12) = 0. and(4,10) = 0. and(1,15) = 0. and(4,15) = 0. and(2,12) = 0. and(5,15) = 0. and(2,15) = 0. and(6,15) = 0. and(3,12) = 0.</pre>	8
---	---

Recordemos que la idea que dicho principio expresa es que el nombre de las constantes debe ser identificativo de cada una de ellas; o lo que es lo mismo, que no puede haber dos nombres distintos para un mismo elemento. La forma en que plasmamos este principio consiste en asignar un número a cada relación

de \mathcal{R}_{RCC} , tal como MACE4 permite: de esta forma, los números son tratados como constantes distintas¹.

Con la instrucción:

```
> mace4 -m -1 - n 17 -N 17 >modelo.in
```

el programa generará todos los modelos de tamaño 17. En este caso, sólo genera un modelo. La salida de MACE4 son las tablas de Cayley de las dos operaciones:

or :		9
	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16	

0	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16	
1	1 11 31 31 31 31 31 41 61 31 31 31 31 31 41 61 6	
2	21 3 2 9 11 13 13 14 16 9 13 11 13 13 14 16 6	
3	31 3 9 3 11 13 13 14 16 9 13 11 13 13 14 16 6	
4	41 3 1 1 1 1 4 1 2 1 2 1 4 1 6 1 1 1 2 1 1 1 2 1 3 1 4 1 6 1 6	
5	5 1 3 1 3 1 3 1 2 5 1 0 1 4 1 6 1 3 1 0 1 3 1 2 1 3 1 4 1 6 1 6	
6	6 1 3 1 3 1 3 1 2 1 0 6 1 4 1 6 1 3 1 0 1 3 1 2 1 3 1 4 1 6 1 6	
7	7 1 4 1 4 1 4 1 4 1 4 1 4 7 1 5 1 4 1 4 1 4 1 4 1 4 1 4 1 5 1 6	
8	8 1 6 1 6 1 6 1 6 1 6 1 6 1 5 8 1 6 1 6 1 6 1 6 1 6 1 6 1 5 1 6	
9	9 1 3 9 9 1 1 1 3 1 3 1 4 1 6 9 1 3 1 1 1 3 1 3 1 4 1 6 1 6	
10	10 1 3 1 3 1 3 1 2 1 0 1 0 1 4 1 6 1 3 1 0 1 3 1 2 1 3 1 4 1 6 1 6	
11	11 1 3 1 1 1 1 1 1 1 1 3 1 3 1 4 1 6 1 1 1 3 1 1 1 3 1 3 1 4 1 6 1 6	
12	12 1 3 1 3 1 3 1 2 1 2 1 2 1 4 1 6 1 3 1 2 1 3 1 2 1 3 1 4 1 6 1 6	
13	13 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 4 1 6 1 3 1 3 1 3 1 3 1 3 1 4 1 6 1 6	
14	14 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 6 1 6	
15	15 1 6 1 6 1 6 1 6 1 6 1 6 1 5 1 5 1 6 1 6 1 6 1 6 1 6 1 6 1 5 1 6	
16	16 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6	
and :		
	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16	

0	0 0	
1	0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 1	
2	0 0 2 0 0 0 0 0 0 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2	
3	0 0 0 3 0 0 0 0 0 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3 0 3	
4	0 0 0 0 4 0 0 0 0 0 0 0 4 4 4 4 4 4 4 4 4 4 4 4 4 4 0 4	
5	0 0 0 0 0 5 0 0 0 0 5 0 0 0 5 0 5 5 5 0 5 0 5 0 5 0 5	
6	0 0 0 0 0 0 6 0 0 0 6 0 0 0 6 0 6 6 6 0 6 0 6 0 6 0 6	
7	0 0 0 0 0 0 0 7 0 0 0 0 0 0 0 0 7 7 7 0 7 7 7 0 7 7 7	
8	0 0 0 0 0 0 0 0 8 0 0 0 0 0 0 0 8 8 8 0 8 8 8 0 8 8 8	
9	0 0 2 3 0 0 0 0 0 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9	
10	0 0 0 0 0 5 6 0 0 0 0 10 0 10 10 10 0 10 0 10 10 0 10	
11	0 0 2 3 4 0 0 0 0 0 9 0 11 4 1 1 1 0 11 0 11 0 11 0 11	
12	0 0 0 0 4 5 6 0 0 0 0 10 4 1 2 1 2 0 12 0 12 0 12 0 12	
13	0 1 2 3 4 5 6 0 0 0 9 10 1 1 1 2 1 3 1 3 0 13 0 13	
14	0 1 2 3 4 5 6 7 0 9 10 1 1 1 2 1 3 1 4 7 14 0 14	
15	0 0 0 0 0 0 0 0 7 8 0 0 0 0 0 0 7 15 15 0 15 0 15	
16	0 1 2 3 4 5 6 7 8 9 10 1 1 1 2 1 3 1 4 1 5 1 6 1 6	

Del modelo generado podemos extraer como ejemplo, y tras sustituir la numeración por la relación correspondiente, las dos tablas siguientes, que representan las posiciones relativas de las relaciones C, O, P, Pi y EQ:

¹Otra manera de hacerlo sería incluir una lista de desigualdades del tipo: $c \neq dr$, $c \neq o$, $c \neq p$, ... $dr \neq o$, $dr \neq p$, ..., pero descartamos esta posibilidad porque resultan 120 desigualdades.

or	EQ	P	Pi	O	C	and	EQ	P	Pi	O	C
EQ	EQ	P	Pi	O	C	EQ	EQ	EQ	EQ	EQ	EQ
P	P	P	O	O	C	P	EQ	P	EQ	P	P
Pi	Pi	O	Pi	O	C	Pi	EQ	EQ	Pi	Pi	Pi
O	O	O	O	O	C	O	EQ	P	Pi	O	O
C	C	C	C	C	C	C	EQ	P	Pi	O	C

Para completar la prueba, sólo faltaría comprobar que las fórmulas que aparecen en el fichero de entrada, las que aparecen en $\boxed{7}$ y $\boxed{8}$, son teoremas de RCC, es decir, que sus traducciones al lenguaje de RCC son demostrables en dicha teoría. Por tanto, nos falta demostrar que:

1. $RCC \vdash \forall x y (C(x, y) \vee DR(x, y))$
2. $RCC \vdash \forall x y (DR(x, y) \rightarrow EC(x, y) \vee DC(x, y))$
3. $RCC \vdash \forall x y (C(x, y) \rightarrow O(x, y) \vee EC(x, y))$
4. $RCC \vdash \forall x y (O(x, y) \rightarrow PO(x, y) \vee P(x, y) \vee Pi(x, y))$
5. $RCC \vdash \forall x y (Pi(x, y) \rightarrow EQ(x, y) \vee PPi(x, y))$
6. $RCC \vdash \forall x y (P(x, y) \rightarrow PP(x, y) \vee EQ(x, y))$
7. $RCC \vdash \forall x y (PPi(x, y) \rightarrow TPPi(x, y) \vee NTPPi(x, y))$
8. $RCC \vdash \forall x y (PP(x, y) \rightarrow NTPP(x, y) \vee TPP(x, y))$
9. $RCC \vdash \neg \exists x y (PO(x, y) \wedge P(x, y))$
10. $RCC \vdash \neg \exists x y (PO(x, y) \wedge Pi(x, y))$
11. $RCC \vdash \neg \exists x y (PO(x, y) \wedge DR(x, y))$
12. $RCC \vdash \neg \exists x y (NTPP(x, y) \wedge Pi(x, y))$
13. $RCC \vdash \neg \exists x y (NTPP(x, y) \wedge DR(x, y))$
14. $RCC \vdash \neg \exists x y (TPP(x, y) \wedge Pi(x, y))$
15. $RCC \vdash \neg \exists x y (TPP(x, y) \wedge DR(x, y))$
16. $RCC \vdash \neg \exists x y (EQ(x, y) \wedge PPi(x, y))$
17. $RCC \vdash \neg \exists x y (EQ(x, y) \wedge DR(x, y))$
18. $RCC \vdash \neg \exists x y (TPPi(x, y) \wedge DR(x, y))$

19. $RCC \vdash \neg \exists x y (NTPPi(x, y) \wedge DR(x, y))$

Estas pruebas son fáciles de obtener por OTTER a partir de RCC. Veamos el fichero de entrada para uno de estos resultados y la prueba obtenida por OTTER. Tomemos, por ejemplo, el primero de ellos². La negación del resultado a demostrar constituye el conjunto soporte:

$\neg c(a, b) .$ $\neg dr(a, b) .$	10
---------------------------------------	----

La prueba obtenida por OTTER es la siguiente:

<pre> 1 [] c(X1, X1) . 3 [] c(X2, X3) -c(X3, X2) . 4 [] -p(X4, X5) -c(X6, X4) c(X6, X5) . 7 [] -o(X7, X8) p(\$f2(X7, X8), X7) . 8 [] -o(X7, X8) p(\$f2(X7, X8), X8) . 11 [] dr(X10, X11) o(X10, X11) . 12 [] -c(a, b) . 13 [] -dr(a, b) . 16 [ur, 13, 11] o(a, b) . 20 [ur, 16, 8] p(\$f2(a, b), b) . 21 [ur, 16, 7] p(\$f2(a, b), a) . 29 [ur, 20, 4, 12] -c(a, \$f2(a, b)) . 36 [ur, 21, 4, 1] c(\$f2(a, b), a) . 51 [ur, 29, 3] -c(\$f2(a, b), a) . 52 [binary, 51.1, 36.1] \$F. </pre>
--

De manera análoga se obtienen las demostraciones del resto de los resultados. Con esto queda probado el teorema. ■

Definición. 2.2.2 Llamaremos \mathcal{E}_{RCC} al siguiente conjunto de ecuaciones (and, or) entre las relaciones de RCC:

Observaciones. 2.2.3 En relación con el conjunto \mathcal{E}_{RCC} , deben hacerse las siguientes observaciones:

²El fichero de entrada contendrá en este caso los axiomas de la teoría RCC [1], junto con la negación del resultado a probar [10]. Solicitamos una prueba haciendo uso del modo autónomo de OTTER. Hecho esto, y a la vista de la prueba obtenida, comentamos los axiomas que no son utilizados, desactivamos el modo autónomo, activamos las reglas necesarias, y obtenemos una prueba más corta, que es la que presentamos.

$$\begin{array}{ll}
\top \equiv C \sqcup D & PO \sqsubseteq \neg P \sqcap \neg Pi \sqcap \neg DR \\
DR \equiv EC \sqcup DC & NTPP \sqsubseteq \neg TPP \sqcap \neg Pi \sqcap \neg DR \\
C \equiv O \sqcup EC & TPP \sqsubseteq \neg Pi \sqcap \neg DR \\
O \equiv PO \sqcup P \sqcup Pi & EQ \sqsubseteq \neg P Pi \sqcap \neg DR \\
Pi \equiv EQ \sqcup P Pi & TPPi \sqsubseteq \neg NTPPi \sqcap \neg DR \\
P \equiv EQ \sqcup PP & NTPPi \sqsubseteq \neg DR \\
P Pi \equiv TPPi \sqcup NTPPi & EC \sqsubseteq \neg DC \\
PP \equiv TPP \sqcup NTPP &
\end{array}$$

Figura 2.2: Representación exógena de \mathcal{E}_{RCC} en DL

- La traducción del conjunto \mathcal{E}_{RCC} al lenguaje de retículos está expresada en las cajas [7] y [8].
- El método de prueba de 2.2.1 acorta la prueba de resultados sobre retículos, de hecho seguiremos utilizando la misma idea para simplificar el análisis de extensiones del retículo asociado a RCC.
- El conjunto \mathcal{E}_{RCC} caracteriza (módulo clausura de dominio y nombres únicos) el retículo \mathcal{R}_{RCC} . Este hecho es consecuencia del teorema 2.2.1. Por tanto, para analizar el retículo e escogen las fórmulas de \mathcal{E}_{RCC} y sólo esas pues, debido a las propiedades de absorción y asociativas del retículo, el resto de las propiedades importantes de RCC se deducen de ellas.
- La traducción de \mathcal{E}_{RCC} es en cierto modo más débil que RCC. De hecho, \mathcal{E}_{RCC} tiene carácter exógeno frente al carácter endógeno de la teoría de LPO de RCC, es decir, en \mathcal{E}_{RCC} se define, por ejemplo, solapamiento mediante una caracterización mediante el resto de las relaciones, sin embargo, en RCC se define con exactitud solapamiento, internamente, desde el punto de vista de los elementos que lo identifican.

Una propiedad importante del retículo de RCC es la siguiente, que desactiva la aplicación de importantes propiedades de retículos, pero que, sin embargo, permite estudiar la existencia de diversas extensiones con facilidad.

Proposición. 2.2.4 *El retículo de las relaciones de RCC no es modular ni distributivo.*

Demostración: Basta comprobar que contiene un subretículo del tipo N_5 , (ver los teoremas 1.1.18, y 1.1.19 y la figura 2.1). Dicho subretículo es el formado por el conjunto de relaciones $\{TPP, PP, P, EQ, \perp\}$ (y las restricciones de las operaciones *and*, *or* a ese conjunto). ■

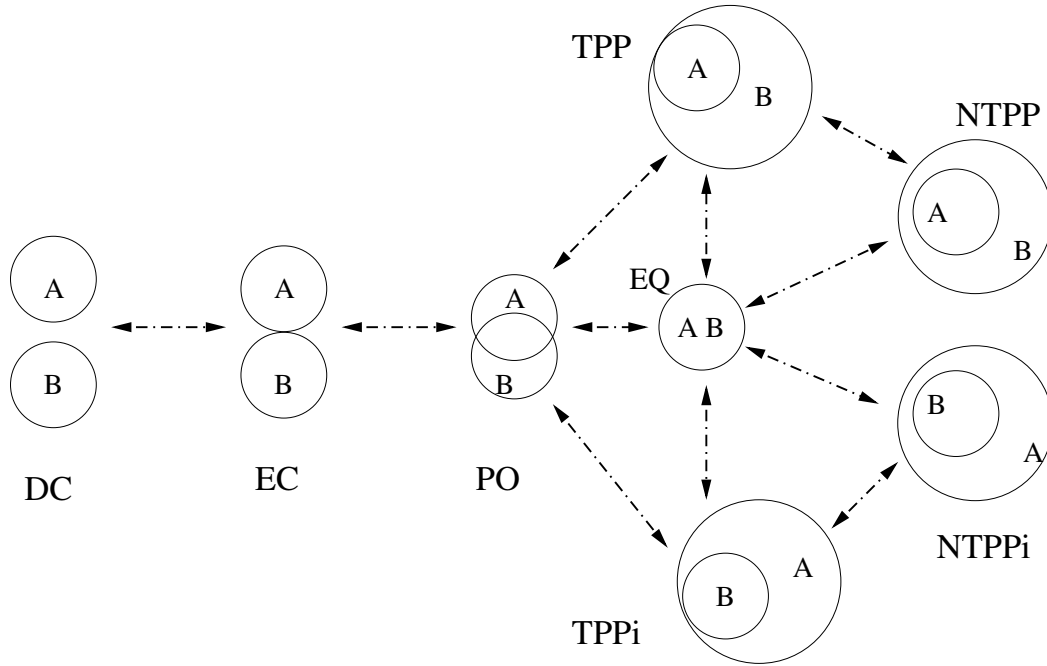


Figura 2.3: Movimientos topológicos en RCC8

2.3. Movimientos topológicos y proyecciones reticulares

Para el desarrollo de esta memoria es importante conocer qué efectos tiene sobre nuestra base de conocimiento un *movimiento* topológico. Los movimientos topológicos son de *mínima distancia topológica*. Son adecuados cognitivamente, y de carácter *continuo*. Estos movimientos están representados en la figura 2.3. En la siguiente sección justificamos este tipo de esquemas.

Otro tipo de movimientos importantes son aquellos que eligen una relación a partir de la proyección de la relación dada. Los *movimientos reticulares* son refinamientos de las relaciones, movimientos hacia abajo en el retículo de las relaciones de RCC. Estos movimientos vienen determinados por la *proyección reticular*.

Definición. 2.3.1 La *proyección reticular* en RCC8 es la función $proy : \mathcal{R}_{RCC} \rightarrow \mathcal{P}(RCC8)$, que a cada $R \in \mathcal{R}_{RCC}$ le asigna el conjunto $\hat{R} := \{R' \in RCC8 : RCC8 \vdash R'(x, y) \rightarrow R(x, y)\}$.

Como ejemplo, la proyección reticular en RCC8 de P , vendrá dada por el conjunto de relaciones de RCC8: $\hat{P} = \{NTPP, TPP, EQ\}$. De manera análoga podemos definir la proyección en RCC5, desde el conjunto restante de relaciones: $\mathcal{R}_{RCC} \setminus \{TPP, NTPP, TPPi, NTPPi\}$.

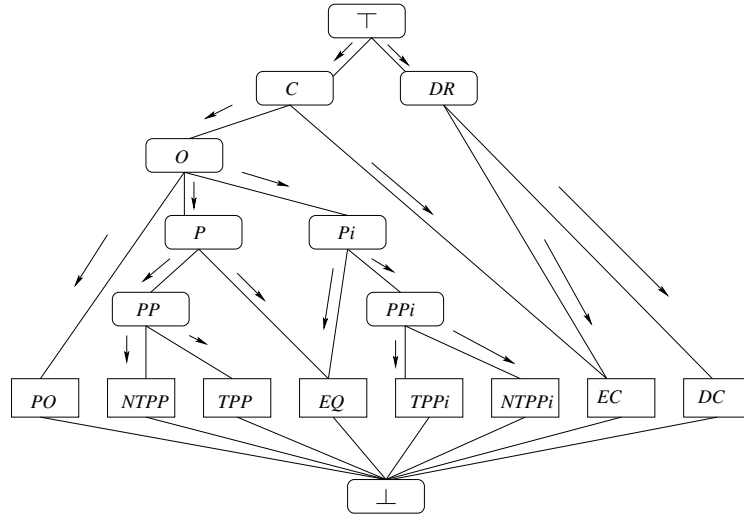


Figura 2.4: Movimientos reticulares en el retículo de RCC

Teorema. 2.3.2 *El diagrama 2.4 representa la proyección reticular, y tal proyección es demostrable en RCC.*

Demostración: Basta demostrar:

1. $RCC \vdash \forall x y (C(x, y) \vee DR(x, y))$
2. $RCC \vdash \forall x y (DR(x, y) \leftrightarrow EC(x, y) \vee DC(x, y))$
3. $RCC \vdash \forall x y (C(x, y) \leftrightarrow O(x, y) \vee EC(x, y))$
4. $RCC \vdash \forall x y (O(x, y) \leftrightarrow PO(x, y) \vee P(x, y) \vee Pi(x, y))$
5. $RCC \vdash \forall x y (P(x, y) \leftrightarrow PP(x, y) \vee EQ(x, y))$
6. $RCC \vdash \forall x y (PP(x, y) \leftrightarrow NTPP(x, y) \vee TPP(x, y))$
7. $RCC \vdash \forall x y (Pi(x, y) \leftrightarrow EQ(x, y) \vee PPi(x, y))$
8. $RCC \vdash \forall x y (PPi(x, y) \leftrightarrow TPPi(x, y) \vee NTPPi(x, y))$

Las pruebas de estos resultados las obtiene OTTER de manera análoga al teorema 2.2.1 (de hecho, son consecuencia del mismo). ■

Por tanto, tal como hemos definido la proyección reticular en RCC8, se obtiene la siguiente definición:

$$\blacksquare \hat{C} = \{PO, NTPP, TPP, EQ, TPPi, NTPPi, EC\}$$

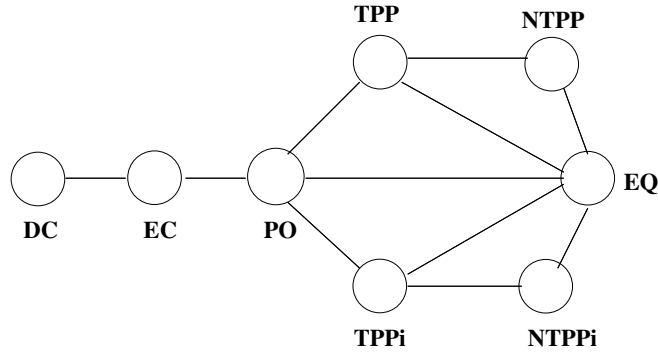


Figura 2.5: Iconización de RCC8 como entorno conceptual

- $\widehat{O} = \{PO, NTTP, TPP, EQ, TPPI, NTPPI\}$
- $\widehat{P} = \{NTTP, TPP, EQ\}$
- $\widehat{P}_i = \{NTPPI, TPPI, EQ\}$
- $\widehat{PP} = \{TPP, NTTP\}$
- $\widehat{P}_i P_i = \{TPPI, NTPPI\}$
- $\widehat{DR} = \{DC, EC\}$
- $\widehat{R} = \{R\}$, para toda relación R de $RCC8$

2.4. Conexión de la proyección reticular

Siguiendo el método de representación ideado por Freksa sobre entornos conceptuales de las relaciones espaciales y temporales (para el razonamiento temporal basado en intervalos según Allen) [27][28], utilizaremos la iconización de la estructura de entorno conceptual que puede verse en la figura 2.5.

Haciendo uso de esta iconización, es posible representar las proyecciones reticulares anteriores como aparece en la figura 2.6.

A la vista de esas representaciones, observamos que la proyección de cualquier relación de \mathcal{R}_{RCC} forma un *entorno conceptual*, en el sentido de que las relaciones de cada proyección son contiguas. Formalizaremos esta idea.

2.4.1. Entornos conceptuales

Definición. 2.4.1 *Dos relaciones de \mathcal{R}_{RCC} se dicen **contiguas** si pueden transformarse directamente una en la otra mediante una deformación continua (en el sentido*

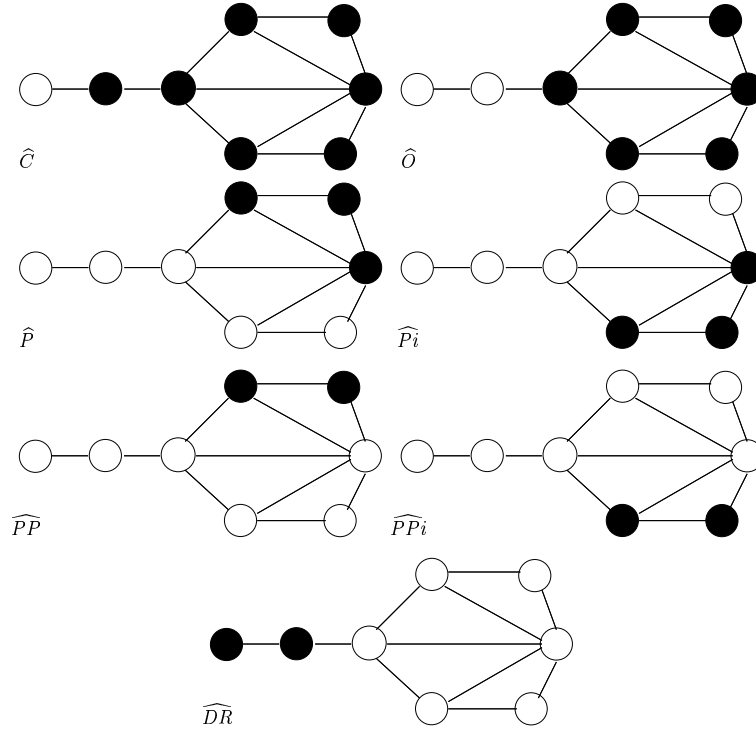


Figura 2.6: Iconización de las propiedades reticulares

topológico). Notaremos este hecho $X \sim_c Y$, para X, Y dos relaciones de \mathcal{R}_{RCC} .

Observar que toda relación es contigua a sí misma.

Definición. 2.4.2 Un conjunto \mathcal{E} de relaciones de RCC8 entre pares de regiones constituye un **entorno conceptual** si determina un subgrafo conexo de la iconización de RCC8 que se muestra en la figura 2.5, es decir,

$$\forall X, Y \in \mathcal{E} \exists Z_1, Z_2, \dots, Z_n \in \mathcal{E} : X \sim_c Z_1, Z_1 \sim_c Z_2, \dots, Z_n \sim_c Y.$$

Obviamente, ser contiguas es una propiedad reflexiva y simétrica. Por otra parte, la definición de entorno conceptual es claramente generalizable a cualquier cálculo para el razonamiento espacial, no es una definición exclusiva de RCC. Observando la figura 2.6, se verifica la siguiente proposición:

Proposición. 2.4.3 Dada $X \in \mathcal{R}_{RCC}$, su proyección reticular, \widehat{X} , en RCC8 es un entorno conceptual.

Cuadro 2.1: Tabla de composición para $RCC8$

$R2(b, c)$ $R1(a, b)$	DC	EC	PO	TPP	$NTPP$	$TPPi$	$NTPPi$	EQ
DC	*	$DC, EC, PO, TPP, NTPP$ (1)	$DC, EC, PO, TPP, NTPP$ (1)	$DC, EC, PO, TPP, NTPP$ (1)	$DC, EC, PO, TPP, NTPP$ (1)	DC	DC	DC
EC	$DC, EC, PO, TPPi, NTPPi$ (2)	$DC, EC, PO, TPP, TPPi, EQ$ (3)	$DC, EC, PO, TPP, NTPP$ (1)	$EC, PO, TPP, NTPP$ (4)	$PO, TPP, NTPP$ (5)	DC, EC (\overline{DR})	DC	EC
PO	$DC, EC, PO, TPPi, NTPPi$ (2)	$DC, EC, PO, TPPi, NTPPi$ (2)	*	$PO, TPP, NTPP$ (5)	$PO, TPP, NTPP$ (5)	$DC, EC, PO, TPPi, NTPPi$ (2)	$DC, EC, PO, TPPi, NTPPi$ (2)	PO
TPP	DC	DC, EC (\overline{DR})	$DC, EC, PO, TPP, NTPP$ (1)	$TPP, NTPP$ (\overline{PP})	$NTPP$	$DC, EC, PO, TPP, TPPi, EQ$ (3)	$DC, EC, PO, TPPi, NTPPi$ (2)	TPP
$NTPP$	DC	DC	$DC, EC, PO, TPP, NTPP$ (1)	$NTPP$	$NTPP$	$DC, EC, PO, TPP, NTPP$ (1)	*	$NTPP$
$TPPi$	$DC, EC, PO, TPPi, NTPPi$ (2)	$EC, PO, TPPi, NTPPi$ (7)	$PO, TPPi, NTPPi$ (8)	$PO, EQ, TPP, TPPi$ (6)	$PO, TPP, NTPP$ (5)	$TPPi, NTPPi$ (\overline{PPi})	$NTPPi$	$TPPi$
$NTPPi$	$DC, EC, PO, TPPi, NTPPi$ (2)	$PO, TPPi, NTPPi$ (8)	$PO, TPPi, NTPPi$ (8)	$PO, TPPi, NTPPi$ (8)	$PO, TPPi, TPP, NTPP, NTPPi, EQ$ (\overline{O})	$NTPPi$	$NTPPi$	$NTPPi$
EQ	DC	EC	PO	TPP	$NTPP$	$TPPi$	$NTPPi$	EQ

2.4.2. Razonamiento basado en entornos conceptuales

Con el objeto de visualizar el uso de las relación de contigüidad y los entornos conceptuales, presentamos la tabla de composición (ver definición 1.8.6) para $RCC8$ según [15] [61], (ver el cuadro 2.1). Recordemos que este tipo de tablas contiene la siguiente información: dadas $R1, R2$ dos relaciones del cálculo $RCC8$ y x, y, z tales que $R1(x, y)$ y $R2(y, z)$, cuáles son las posibles relaciones entre x y z ?

En la tabla 2.1, los números de (1) a (8) se corresponden con los iconos que detallamos en la figura 2.7. El símbolo * representa a $RCC8$.

A continuación realizamos unas observaciones respecto a la tabla expuesta. Algunas coinciden en gran medida con las realizadas por Freksa para el cálculo espacial en [27] y [28], y otras generalizan las citadas observaciones.

- Si el resultado de la composición produce una disyunción de relaciones posibles, dicha disyunción constituye un entorno conceptual (no trivial). Este hecho equivale a razonar con incertidumbre; en otros casos se reduce a una única relación (entorno trivial), y el resto representa $RCC8$, el caso que menos información aporta, por ejemplo, $NTPP \circ NTPPi = \top$.
- De todos los entornos combinatorialmente posibles, sólo hay ocho distintos en la tabla, que precisamente no se corresponden con las proyecciones

de las relaciones sobre RCC8.

- Cuando dos relaciones contiguas se componen con una misma relación, o bien dan lugar a dos entornos idénticos, o bien uno de ellos contiene al otro. Esto quiere decir que, en cierta medida, la composición por una relación fija tiene carácter *continuo*.
- En ningún caso la composición de dos relaciones contiguas se representa por un entorno al que no pertenezcan o del que ambas relaciones no sean contiguas, entendiendo que una relación es contigua a un entorno si éste contiene una relación contigua a la primera. Por tanto, podemos decir que, en cierto modo, la función composición tiene carácter *continuo*.

Las dos últimas observaciones son las que ponen de manifiesto la relevancia del papel de los entornos conceptuales. Más adelante, en el capítulo 7, se hará efectiva la utilidad de éstos en las extensiones de RCC.

De igual manera, tiene sentido estudiar la tabla de composición de las proyecciones reticulares de RCC en RCC8, es decir, la que resulta de componer $\widehat{R1}$ con $\widehat{R2}$, para cada par $R1, R2$ de relaciones de RCC. La tabla resultante nos muestra la siguiente información: denotando por $\widehat{R}(x, y)$ el hecho de que la relación existente entre x e y está en \widehat{R} , nos preguntamos lo siguiente: dados $\widehat{R1}(x, y)$ y $\widehat{R2}(y, z)$, ¿cuál es el conjunto de posibles relaciones entre x y z ? La respuesta nos la proporciona el siguiente teorema.

Teorema. 2.4.4 Sean $R1, R2$ dos relaciones de RCC. Para cualesquiera x, y, z tales que $\widehat{R1}(x, y)$ y $\widehat{R2}(y, z)$, el conjunto de posibles relaciones entre x y z es el contenido en la casilla correspondiente de la tabla 2.2.

Al igual que en el cuadro 2.1, los números de (1) a (12) se corresponden con los iconos que detalla la figura 2.7. El símbolo (*) representa a RCC8.

Demostración: Veamos, como ejemplo, cómo obtenemos $\widehat{O} \circ \widehat{EC}$. Para ello, al ser $\widehat{O} = \{PO, NTPP, TPP, EQ, TPPi, NTPPi\}$, bastará calcular las composición de cada una de éstas con EC (ya que EC coincide con \widehat{EC}). Tenemos, pues:

$$PO \circ EC = \{DC, PO, TPPi, NTPPi, EC\}$$

$$NTPP \circ DC = \{DC\}$$

$$TPP \circ DC = \{EC, DC\}$$

$$EQ \circ DC = \{EC\}$$

$$TPPi \circ DC = \{EC, TPPi, PO\}$$

$$NTPPi \circ DC = \{PO, NTPPi\}$$

La unión de estos conjuntos coincide con el entorno conceptual denotado por (2), es decir, $\{PO, EC, DC, NTPPi, TPPi\}$. ■

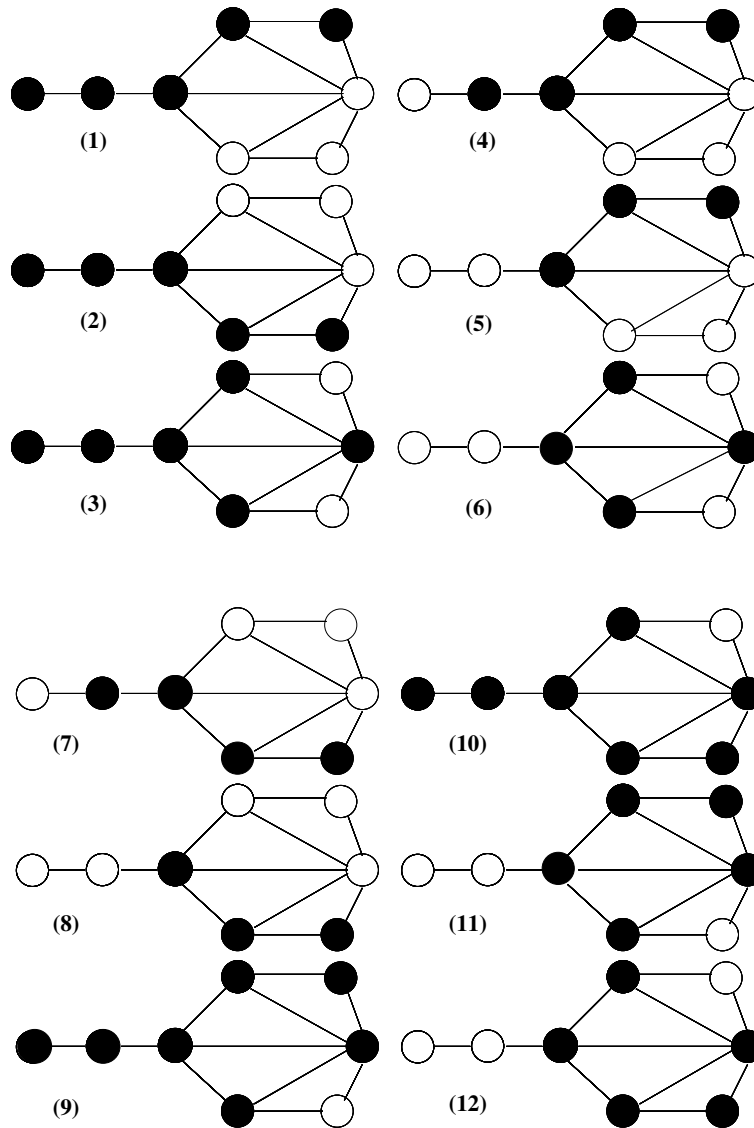


Figura 2.7: Iconos que aparecen en las tablas de composición

		$\widehat{R2}(b, c)$														
		\widehat{C}	\widehat{O}	\widehat{P}	\widehat{P}_i	\widehat{PP}	\widehat{PP}_i	\widehat{DR}	\widehat{DC}	\widehat{EC}	\widehat{PO}	\widehat{TPP}	\widehat{NTPP}	\widehat{EQ}	\widehat{TPP}_i	\widehat{NTPP}_i
$\widehat{R1}(a, b)$	\widehat{C}	*	*	\widehat{C}	*	\widehat{C}	*	10	2	10	*	\widehat{C}	\widehat{O}	\widehat{C}	*	*
	\widehat{O}	*	*	\widehat{O}	*	\widehat{O}	*	2	2	2	*	\widehat{O}	\widehat{O}	\widehat{O}	*	*
	\widehat{P}	*	*	\widehat{P}	*	\widehat{PP}	*	\widehat{DR}	\widehat{DC}	\widehat{DR}	1	\widehat{PP}	\widehat{NTPP}	\widehat{P}	9	*
	\widehat{P}_i	\widehat{C}	\widehat{O}	\widehat{O}	\widehat{P}_i	\widehat{O}	\widehat{PP}_i	2	2	7	8	12	\widehat{O}	\widehat{P}_i	\widehat{PP}_i	\widehat{NTPP}_i
	\widehat{PP}	*	*	\widehat{PP}	*	\widehat{PP}	*	\widehat{DR}	\widehat{DC}	\widehat{DR}	1	\widehat{PP}	\widehat{NTPP}	\widehat{PP}	9	*
	\widehat{PP}_i	\widehat{C}	\widehat{O}	\widehat{O}	\widehat{PP}_i	\widehat{O}	\widehat{PP}_i	2	2	7	8	12	\widehat{O}	\widehat{PP}_i	\widehat{PP}_i	\widehat{NTPP}_i
	\widehat{DR}	9	1	1	\widehat{DR}	1	\widehat{DR}	*	*	9	1	1	1	\widehat{DR}	\widehat{DR}	\widehat{DC}
	\widehat{DC}	1	1	1	\widehat{DC}	1	\widehat{DC}	*	*	1	1	1	1	\widehat{DC}	\widehat{DC}	\widehat{DC}
	\widehat{EC}	9	1	4	\widehat{DR}	4	\widehat{DR}	3	2	3	1	4	5	\widehat{EC}	\widehat{DR}	\widehat{DC}
	\widehat{PO}	*	*	5	2	5	2	2	2	2	*	5	5	\widehat{PO}	2	2
	\widehat{TPP}	*	*	\widehat{PP}	3	\widehat{PP}	3	\widehat{DR}	\widehat{DC}	\widehat{DR}	1	\widehat{PP}	\widehat{NTPP}	\widehat{TPP}	3	2
	\widehat{NTPP}	*	*	\widehat{NTPP}	*	\widehat{NTPP}	*	\widehat{DC}	\widehat{DC}	\widehat{DC}	1	\widehat{NTPP}	\widehat{NTPP}	\widehat{NTPP}	1	*
	\widehat{EQ}	\widehat{C}	\widehat{O}	\widehat{P}	\widehat{P}_i	\widehat{PP}	\widehat{PP}_i	\widehat{DR}	\widehat{DC}	\widehat{EC}	\widehat{PO}	\widehat{TPP}	\widehat{NTPP}	\widehat{EQ}	\widehat{TPP}_i	\widehat{NTPP}_i
	\widehat{TPP}_i	\widehat{C}	\widehat{O}	11	\widehat{PP}_i	11	\widehat{PP}_i	2	2	7	8	6	5	\widehat{TPP}_i	\widehat{PP}_i	\widehat{NTPP}_i
\widehat{NTPP}_i	\widehat{O}	\widehat{O}	\widehat{O}	\widehat{NTPP}_i	\widehat{O}	\widehat{NTPP}_i	2	2	8	8	8	\widehat{O}	\widehat{NTPP}_i	\widehat{NTPP}_i	\widehat{NTPP}_i	

Cuadro 2.2: Tabla de composición para la proyección de RCC sobre $RCC8$

Observando esta última tabla, es interesante hacer las siguientes consideraciones:

- Como toda relación R de RCC8 coincide con su proyección $\hat{R} = \{R\}$, la subtabla que aparece en la esquina inferior derecha es la tabla de composición de RCC8, que vemos ampliada en el cuadro 2.1.
- La mayoría de las composiciones producen la proyección de una relación de RCC. Sólo un tercio aproximadamente de los resultados corresponde a los entornos de la figura 2.7.
- La composición de las proyecciones de dos relaciones, da lugar a un entorno en el que ambas relaciones son contiguas.
- El único subcálculo que encontramos dentro de la composición de las proyecciones de RCC es el propio RCC8. Por subcálculo entendemos un subconjunto de \hat{R} cerrado bajo composición.

Capítulo 3

Extensiones. Interpretación de RCC

A lo largo del capítulo anterior suponemos conocida en todo momento la relación de RCC existente entre dos regiones cualesquiera. Sin embargo, hay que considerar que este hecho no es siempre cierto: en ocasiones, dicha relación no es explícita, lo cual no significa que el desconocimiento sea total, sino que no sabemos qué relación mantienen dos regiones de entre un conjunto determinado de posibilidades. Este capítulo presenta una metodología para el estudio de este tipo de relaciones de indefinición, que es generalizable a otro tipo de extensiones. Finalmente, la extensión del conjunto de relaciones debe estar soportado por una interpretación que extienda una clase de los modelos de la teoría inicial, para manejar dichas extensiones con un soporte semántico.

3.1. Introduciendo una relación de indefinición disjunta con RCC8

Para estudiar la posibilidad de expresar una relación de “indefinición”, introduciremos en RCC una nueva relación $I(x,y)$ que expresaremos “*la relación entre x e y es desconocida (hasta cierto punto)*”. Además pretendemos que dicha relación sea disjunta con RCC8, pues, efectivamente, la indeterminación debe ser un hecho disjunto con la determinación exacta que proporcionan las relaciones de RCC8. De esta forma, puede utilizarse como parte de un cálculo que extienda a RCC8 (para que sea un sistema JEPD). La prueba del siguiente teorema proporciona un método para obtener extensiones genéricas del retículo de relaciones de RCC, método que utilizaremos más adelante en otros casos.

Teorema. 3.1.1 *Existen ocho y sólo ocho extensiones \mathcal{E}_{RCC} -conservativas del retículo \mathcal{R}_{RCC} con una relación de indefinición que, junto con RCC8, forma un JEPD.*

Demostración: Para demostrar la existencia facilitamos a MACE4 la definición de retículo [6] y, de manera análoga a las pruebas anteriores, enumeramos las relaciones de RCC de 0 a 16. Ahora, a la nueva relación se le asignará el número 17. Establecemos las restricciones sobre \wedge y \vee como en el capítulo anterior, [7] y [8], que componen \mathcal{E}_{RCC} . Añadimos también las fórmulas correspondientes a la nueva relación, estableciendo simplemente que I sea disjunta con cada una de las relaciones de RCC8.

```
and(1, 17) = 0.
and(2, 17) = 0.
and(3, 17) = 0.
and(4, 17) = 0.
and(5, 17) = 0.
and(6, 17) = 0.
and(7, 17) = 0.
and(8, 17) = 0.
```

Mediante la instrucción

```
> mace4 -n 18 -N 18 -m -1 <indefinicion.in
```

el programa nos devuelve exactamente ocho modelos, es decir, que existen ocho y sólo ocho retículos que extienden al retículo de las relaciones de RCC y poseen una nueva relación que interpretaremos como indefinición y que denotaremos por I .

Las figuras 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7 y 3.8 muestran los ocho diagramas de Hasse que representan a los modelos generados por MACE4. En ellas puede observarse que la nueva relación I aparece en cada disyunción del retículo. En el caso especial de la relación que aparece en la figura 3.8, puede entenderse que representa el hecho de que “desconocemos totalmente la relación que existe entre x e y ” en RCC8. En el resto de los casos (los otros siete) desconocemos la relación existente entre x e y pero sólo “hasta cierto punto”, como describiremos a continuación.

Para demostrar que no hay más que ocho relaciones posibles de ese tipo, facilitamos a OTTER un fichero de entrada que contenga lo siguiente:

- Definición de retículo, [6]
- Condiciones disyuntivas que lo definen, [7]
- Cada relación es la disyunción de ella y su inmediata inferior, [11]
- Cada relación es la conjunción de ella y su inmediata superior, [12]

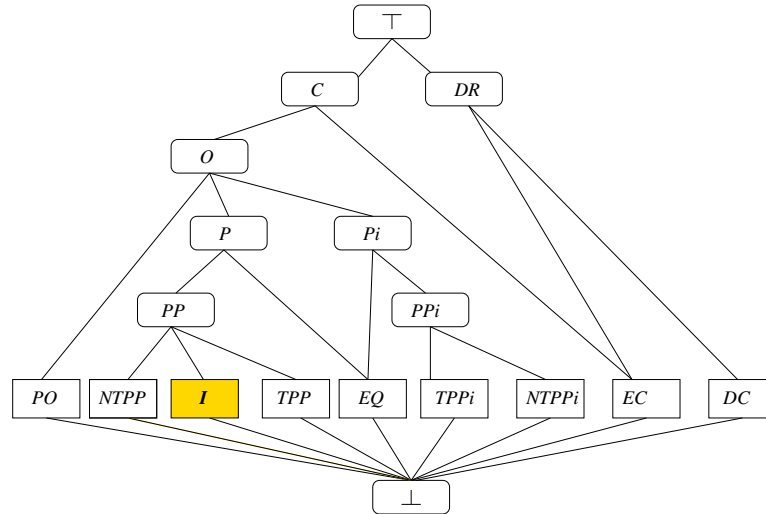


Figura 3.1: Modelo L_1 del Retículo $RCC + I_1$

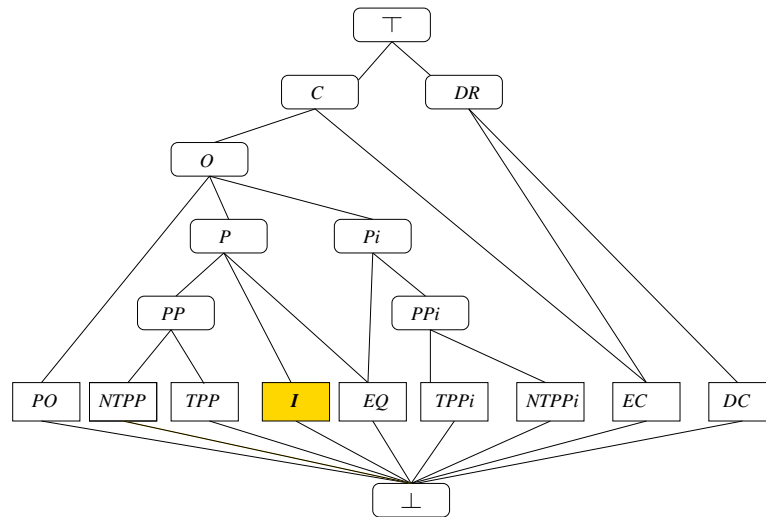


Figura 3.2: Modelo L_2 del Retículo $RCC + I_2$

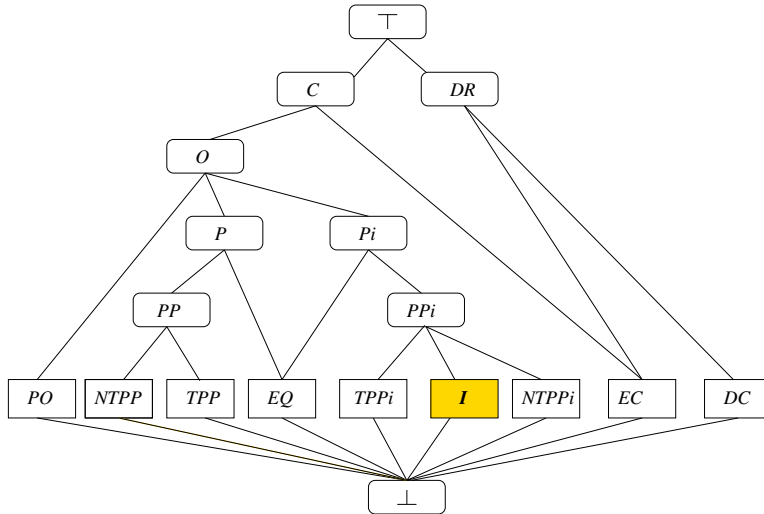


Figura 3.3: Modelo L_3 del Reticulo $RCC + I_3$

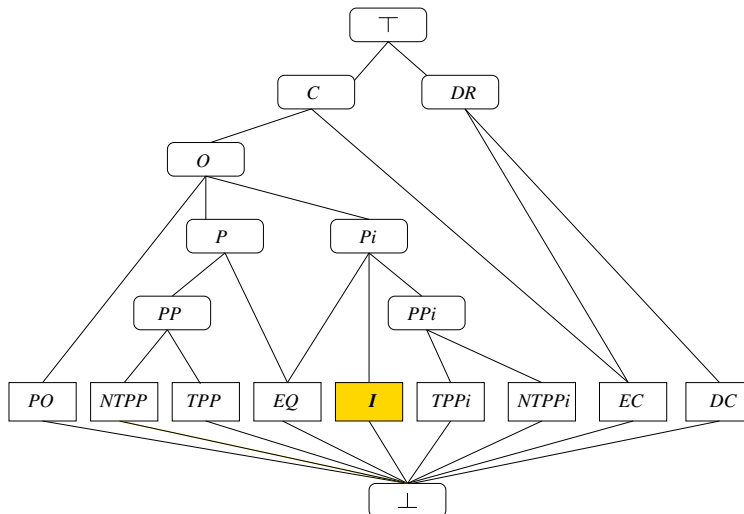


Figura 3.4: Modelo L_4 del Reticulo $RCC + I_4$

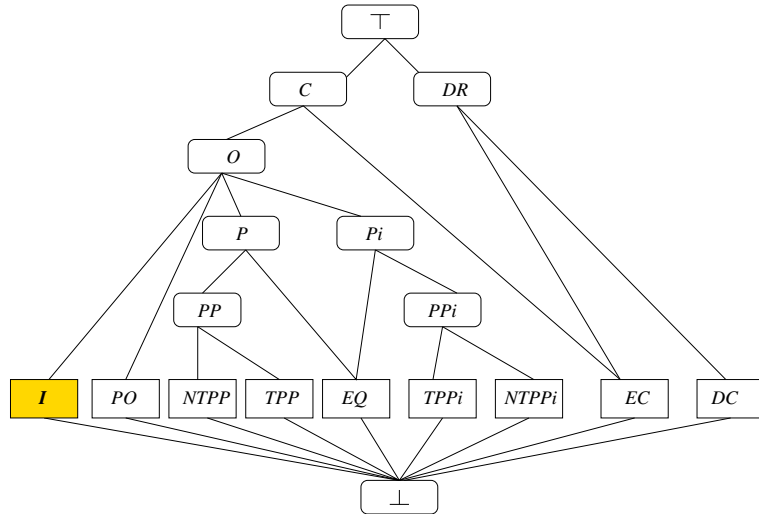


Figura 3.5: Modelo L_5 del Retículo $RCC + I_5$

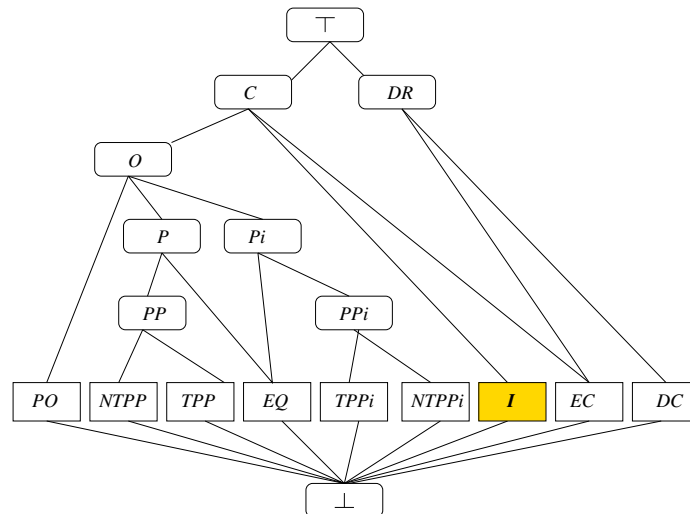


Figura 3.6: Modelo L_6 del Retículo $RCC + I_6$

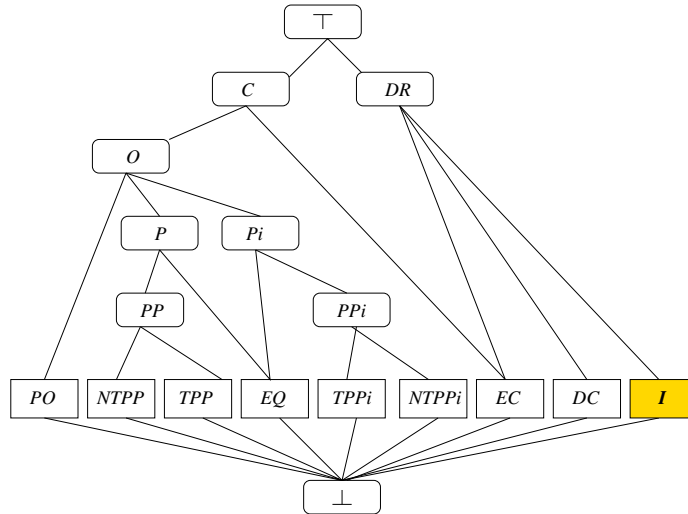


Figura 3.7: Modelo L_7 del Reticulo $RCC + I_7$

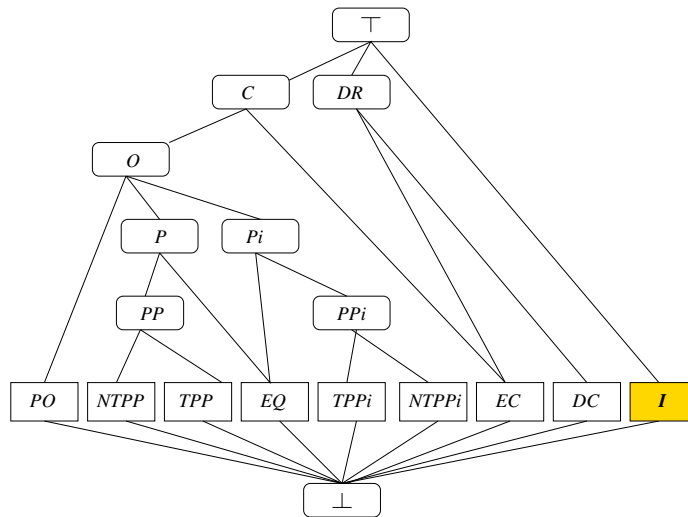


Figura 3.8: Modelo L_8 del Reticulo $RCC + I_8$

- Posiciones del \top y el \perp , [16]
- Las relaciones de RCC8 son disjuntas, [13]
- Resto de relaciones \wedge que faltan, [14]
- La nueva relación que introducimos, in , es disjunta con las de RCC8, [15]
- Axioma de clausura de dominio, [17]
- Axioma de nombres únicos, [18]
- La nueva relación no coincide con las siete anteriores, [19]

$ \begin{array}{ll} c = \text{or}(c, ec) . & p = \text{or}(p, iq) . \\ c = \text{or}(c, o) . & pp = \text{or}(pp, ntp) . \\ dr = \text{or}(dr, ec) . & pp = \text{or}(pp, tpp) . \\ dr = \text{or}(dr, dc) . & pi = \text{or}(pi, iq) . \\ o = \text{or}(o, po) . & pi = \text{or}(pi, ppi) . \\ o = \text{or}(o, p) . & ppi = \text{or}(ppi, tppi) . \\ o = \text{or}(o, pi) . & ppi = \text{or}(ppi, ntpi) . \\ p = \text{or}(p, pp) . & \end{array} $	11
--	----

$ \begin{array}{l} ec = \text{and}(c, ec) . \\ o = \text{and}(c, o) . \\ ec = \text{and}(dr, ec) . \\ dc = \text{and}(dr, dc) . \\ po = \text{and}(o, po) . \\ p = \text{and}(o, p) . \\ pi = \text{and}(o, pi) . \\ pp = \text{and}(p, pp) . \\ iq = \text{and}(p, iq) . \\ ntp = \text{and}(pp, ntp) . \\ tpp = \text{and}(pp, tpp) . \\ iq = \text{and}(pi, iq) . \\ ppi = \text{and}(pi, ppi) . \\ tppi = \text{and}(ppi, tppi) . \\ ntpi = \text{and}(ppi, ntpi) . \end{array} $	12
---	----

13

```

b = and(po, ntp).  b = and(po, tpp).
b = and(po, iq).   b = and(po, ntppi).
b = and(po, tppi). b = and(po, ec).
b = and(po, dc).

b = and(tpp, ntp). b = and(tpp, iq).
b = and(tpp, tppi). b = and(tpp, ntppi).
b = and(tpp, ec).  b = and(tpp, dc).

b = and(iq, ntp).   b = and(tppi, ntp).
b = and(ntpp, ntppi). b = and(ntpp, ec).
b = and(ntpp, dc).

b = and(iq, tppi).   b = and(iq, ntppi).
b = and(iq, ec).     b = and(iq, dc).

b = and(ntppi, tppi). b = and(ec, tppi).
b = and(dc, tppi).

b = and(ec, ntppi).  b = and(dc, ntppi).

b = and(ec, dc).

```

14

```

iq = and(p, pi).   ec = and(c, dr).

```

15

```

b = and(in, po).   b = and(in, ntp).
b = and(in, tpp).  b = and(in, iq).
b = and(in, tppi). b = and(in, ntppi).
b = and(in, ec).   b = and(in, dc).

```

16

```

all x (b = and(b, x)). all x (t = or(t, x)).

```

17

```

all x (x = t | x = c | x = dr | x = o
      | x = p | x = pi | x = po
      | x = pp | x = ppi | x = ntp
      | x = tpp | x = iq | x = tppi
      | x = ntppi | x = ec
      | x = dc | x = b | x = in).

```

18

```

(t != c).      (t != dr).      (t != o).      (t != p).
(t != pi).     (t != pp).     (t != ppi).   (t != po).
(t != ntp).   (t != tpp).   (t != iq).    (t != tppi).
(t != ntpi).  (t != ec).    (t != dc).    (t != b).
(t != in).

(c != dr).     (c != o).     (c != p).     (c != pi).
(c != pp).     (c != ppi).  (c != po).    (c != ntp).
(c != tpp).    (c != iq).   (c != tppi).  (c != ntpi).
(c != ec).     (c != dc).   (c != b).     (c != in).

(dr != o).     (dr != p).   (dr != pi).   (dr != pp).
(dr != ppi).  (dr != po).  (dr != ntp).  (dr != tpp).
(dr != iq).    (dr != tppi). (dr != ntpi). (dr != ec).
(dr != dc).    (dr != b).   (dr != in).

(o != p).      (o != pi).   (o != pp).    (o != ppi).
(o != po).     (o != ntp).  (o != tpp).   (o != iq).
(o != tppi).   (o != ntpi). (o != ec).    (o != dc).
(o != b).      (o != in).

(p != pi).     (p != pp).   (p != ppi).   (p != po).
(p != ntp).   (p != tpp).  (p != iq).    (p != tppi).
(p != ntpi).  (p != ec).   (p != dc).    (p != b).
(p != in).

(pi != pp).    (pi != ppi). (pi != po).   (pi != ntp).
(pi != tpp).  (pi != iq).  (pi != tppi). (pi != ntpi).
(pi != ec).   (pi != dc).  (pi != b).    (pi != in).

(pp != ppi).   (pp != po).   (pp != ntp).
(pp != tpp).   (pp != iq).   (pp != tppi).
(pp != ntpi).  (pp != ec).   (pp != dc).
(pp != b).     (pp != in).   (ppi != po).
(ppi != ntp).  (ppi != tpp). (ppi != iq).
(ppi != tppi). (ppi != ntpi). (ppi != ec).
(ppi != dc).   (ppi != b).   (ppi != in).

```

```

(po != ntp).   (po != tpp).   (po != iq).
(po != tppi). (po != ntpi). (po != ec).
(po != dc).   (po != b).     (po != in).

(ntpp != tpp). (ntpp != iq). (ntpp != tppi).
(ntpp != ntpi). (ntpp != ec). (ntpp != dc).
(ntpp != b).   (ntpp != in).

(tpp != iq).   (tpp != tppi). (tpp != ntpi).
(tpp != ec).   (tpp != dc).   (tpp != b).
(tpp != in).   (iq != tppi).  (iq != ntpi).
(iq != ec).    (iq != dc).    (iq != b).
(iq != in).

(tppi != ec). (tppi != ntpi). (tppi != dc).
(tppi != b).  (tppi != in).

(ntpi != ec). (ntpi != dc).  (ntpi != b).
(ntpi != in).

(ec != dc).   (ec != b).     (ec != in).

(dc != b).    (dc != in).

(in != b).

```

```

and(c, in) != in.
and(dr, in) != in.
and(o, in) != in.
and(p, in) != in.
and(pi, in) != in.
and(pp, in) != in.
and(ppi, in) != in.

```

19

La prueba consiste en comprobar que la nueva relación, *in*, que hemos forzado a ser distinta de las siete primeras halladas por MACE4, coincide con la octava. La idea es demostrar que la relación *in* verifica la última fila y columna de las tablas de Cayley del modelo correspondiente a I_8 . Como las relaciones \wedge y \vee son simétricas, bastará con probar los resultados de la última fila (o columna) de cada tabla. Reproducimos las tablas en cuestión en la caja siguiente:


```

or :
  | 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
  +-----+
0 | 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
1 | 1 1 1 3 1 3 1 3 1 3 1 4 1 6 1 3 1 3 1 3 1 3 1 4 1 6 1 6 1 6
2 | 2 1 3 2 9 1 1 3 1 3 1 4 1 6 9 1 3 1 1 1 3 1 3 1 4 1 6 1 6 1 6
3 | 3 1 3 9 3 1 1 3 1 3 1 4 1 6 9 1 3 1 1 1 3 1 3 1 4 1 6 1 6 1 6
4 | 4 1 3 1 1 1 1 4 1 2 1 2 1 4 1 6 1 1 1 2 1 1 1 2 1 3 1 4 1 6 1 6 1 6
5 | 5 1 3 1 3 1 3 1 2 5 1 0 1 4 1 6 1 3 1 0 1 3 1 2 1 3 1 4 1 6 1 6 1 6
6 | 6 1 3 1 3 1 3 1 2 1 0 6 1 4 1 6 1 3 1 0 1 3 1 2 1 3 1 4 1 6 1 6 1 6
7 | 7 1 4 1 4 1 4 1 4 1 4 1 4 7 1 5 1 4 1 4 1 4 1 4 1 4 1 4 1 5 1 6 1 6
8 | 8 1 6 1 6 1 6 1 6 1 6 1 6 1 5 8 1 6 1 6 1 6 1 6 1 6 1 6 1 5 1 6 1 6
9 | 9 1 3 9 9 1 1 3 1 3 1 4 1 6 9 1 3 1 1 1 3 1 3 1 4 1 6 1 6 1 6 1 6
10 | 10 1 3 1 3 1 3 1 2 1 0 1 0 1 4 1 6 1 3 1 0 1 3 1 2 1 3 1 4 1 6 1 6 1 6
11 | 1 1 1 3 1 1 1 1 1 1 1 3 1 3 1 4 1 6 1 1 1 3 1 1 1 3 1 3 1 4 1 6 1 6 1 6
12 | 1 2 1 3 1 3 1 3 1 2 1 2 1 2 1 4 1 6 1 3 1 2 1 3 1 2 1 3 1 4 1 6 1 6 1 6
13 | 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 4 1 6 1 3 1 3 1 3 1 3 1 3 1 4 1 6 1 6 1 6
14 | 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 6 1 4 1 4 1 4 1 4 1 4 1 4 1 6 1 6 1 6
15 | 1 5 1 6 1 6 1 6 1 6 1 6 1 6 1 5 1 5 1 6 1 6 1 6 1 6 1 6 1 6 1 5 1 6 1 6
16 | 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6
17 | 1 7 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 7

```

```

and :
  | 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
  +-----+
0 | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 | 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0
2 | 0 0 2 0 0 0 0 0 0 0 2 0 2 0 2 2 0 2 0
3 | 0 0 0 3 0 0 0 0 0 0 3 0 3 0 3 3 0 3 0
4 | 0 0 0 0 4 0 0 0 0 0 0 0 4 4 4 4 0 4 0
5 | 0 0 0 0 0 5 0 0 0 0 5 0 5 5 5 0 5 0
6 | 0 0 0 0 0 0 6 0 0 0 6 0 6 6 6 0 6 0
7 | 0 0 0 0 0 0 0 7 0 0 0 0 0 0 7 7 7 0
8 | 0 0 0 0 0 0 0 0 8 0 0 0 0 0 0 8 8 0
9 | 0 0 2 3 0 0 0 0 0 9 0 9 0 9 9 0 9 0
10 | 0 0 0 0 0 5 6 0 0 0 10 0 10 10 10 0 10 0
11 | 0 0 2 3 4 0 0 0 0 9 0 11 4 1 1 1 0 11 0
12 | 0 0 0 0 4 5 6 0 0 0 10 4 1 2 1 2 0 12 0
13 | 0 1 2 3 4 5 6 0 0 9 10 1 1 1 2 1 3 1 3 0 13 0
14 | 0 1 2 3 4 5 6 7 0 9 10 1 1 1 2 1 3 1 4 7 1 4 0
15 | 0 0 0 0 0 0 0 7 8 0 0 0 0 0 0 7 1 5 1 5 0
16 | 0 1 2 3 4 5 6 7 8 9 10 1 1 1 2 1 3 1 4 1 5 1 6 1 7
17 | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 7 1 7

```

Por tanto, para demostrar cada resultado, habrá que añadir al fichero de entrada de OTTER la negación del mismo. Por ejemplo, para demostrar que

$$\text{or}(\text{in}, \perp) = \text{in}$$

(primer dato de la fila *in* en tabla de Cayley para *or*), ponemos en el conjunto soporte del fichero de entrada: $\text{or}(\text{in}, \text{b})! = \text{in}$.

La prueba obtenida por OTTER es la siguiente:

```

1 [] or(A,B)=or(B,A).
13 [] A=or(A,and(A,B)).
14 [copy,13,flip.1] or(A,and(A,B))=A.
19 [] A=A.
233 [] b=and(in,po).
234 [copy,233,flip.1] and(in,po)=b.
486 [] or(b,in)!=in.
852,851 [para_from,234.1.1,14.1.1.2] or(in,b)=in.
902 [para_into,486.1.1,1.1.1,demod,852] in!=in.
903 [binary,902.1,19.1] $F.

```

De manera análoga se obtienen las pruebas del resto de resultados. ■

3.2. Interpretación de la relación de indefinición

Para completar el análisis realizado anteriormente necesitamos interpretar la teoría mereotopológica asociada a cada extensión obtenida. De esta forma, nos aseguramos un soporte topológico adecuado que nos permita razonar con ciertas garantías en dichas extensiones. Las siguientes definiciones y resultados proporcionan una interpretación de las relaciones de indefinición que acabamos de introducir¹.

Definición. 3.2.1 (*Espacio topológico con pulsación*). Sea un espacio topológico $\Omega = (X, \mathcal{T})$. Una **pulsación** es una aplicación $\sigma : \mathcal{R}(\Omega) \mapsto \mathcal{R}(\Omega)$ tal que $\forall X \in \mathcal{R}(\Omega)$ se verifica $\overline{X} \subseteq \overline{\sigma(X)}$, donde $\mathcal{R}(\Omega)$ es el conjunto de los conjuntos regulares de Ω . Al par (Ω, σ) donde Ω es un espacio T_3 conexo y no trivial lo llamaremos **espacio topológico con pulsación**.

¹Si bien se podría haber definido un contexto general donde todas se puedan interpretar, como hacemos en la sección 3.5, es importante para nuestra aplicación sobre ontologías distinguir las interpretaciones que extienden “espacialmente” las regiones. Por este motivo introducimos la siguiente interpretación.

Ejemplo. 3.2.2 *Un ejemplo de espacio topológico con pulsación es el siguiente: Consideremos \mathbb{R}^n con la topología euclídea. Sea $\varepsilon > 0$ y $A \in \mathcal{R}(\mathbb{R}^n)$, es decir, A regular. Definimos $\sigma : \mathcal{R}(\mathbb{R}^n) \rightarrow \mathcal{R}(\mathbb{R}^n)$, como sigue:*

$$\sigma(A) := \bigcup_{x \in A} B(x, \varepsilon)$$

donde $B(x, \varepsilon)$ representa la bola abierta en \mathbb{R}^n de centro x y radio ε .

Proposición. 3.2.3 *(\mathbb{R}^n, σ) con σ la definida en 3.2.1 es un espacio topológico con pulsación.*

Comprobemos que $\sigma(A)$ es un conjunto regular. Sea $x \in \overset{\circ}{\sigma(A)}$. Existe un entorno $\varepsilon_x > 0$ tal que $B(x, \varepsilon) \subseteq \overset{\circ}{\sigma(A)}$. Por otro lado, $\overline{\sigma(A)} = \partial(\sigma(A)) \cup \overset{\circ}{\sigma(A)} = \partial(\sigma(A)) \cup \sigma(A)$.

Por reducción al absurdo, supongamos que $x \notin \sigma(A)$. En ese caso, debe ser $x \in \partial(\sigma(A))$. Sea pues $y \in \sigma(A) \cap B(x, \varepsilon_x)$. Como $B(x, \varepsilon_x)$ es convexa, podemos tomar el segmento $\overline{xy} \subseteq B(x, \varepsilon_x)$.

Si $x \in B(y, \varepsilon)$, habremos concluido, ya que tendríamos $x \in B(x, \varepsilon_x) \cap B(y, \varepsilon) \subseteq \sigma(A)$, y habíamos supuesto $x \notin \sigma(A)$. En caso contrario, $x \notin B(y, \varepsilon)$, la distancia entre x e y es mayor que ε : sea pues $z \in \overline{xy} \cap B(y, \varepsilon) \subseteq \sigma(A)$. De nuevo $x \notin B(z, \varepsilon) \subseteq \sigma(A)$, volviendo a empezar. Aplicando reiteradamente este razonamiento, llegamos por tanto a $x \notin \partial(\sigma(A))$, de donde $x \in \sigma(A)$, concluyendo que $\overline{\overset{\circ}{\sigma(A)}} \subseteq \sigma(A)$.

Definición. 3.2.4 *(Espacio topológico con contracción). Sea un espacio topológico $\Omega = (X, \mathcal{T})$. Una **contracción** es una aplicación $\xi : \mathcal{R}(\mathbb{R}^n) \rightarrow \mathcal{R}(\mathbb{R}^n)$ tal que se verifica $\xi(A) \subseteq A$ para todo A con interior no vacío. Al par (Ω, ξ) , donde Ω es un espacio T_3 conexo y no trivial, lo llamaremos **espacio topológico con contracción**.*

Teorema. 3.2.5 *Siete de las ocho extensiones del teorema 3.1.1 son interpretables mediante un espacio topológico con pulsación.*

Demostración: Sea (Ω, σ) un espacio topológico con pulsación σ . Definimos Ω_k como la estructura en el lenguaje de $RCC + \{I_k\}$ con $k \in \{1, 2, 3, 4, 5, 7, 8\}$, donde para cada $R \in \mathcal{R}_{RCC}$, la interpretación de R en Ω_k es la siguiente²:

²Para simplificar la notación, a lo largo de la demostración, denotaremos

$$R^\sigma := (\sigma \times \sigma)^{-1}[R^\Omega]$$

Análogamente, denotaremos por $\bigvee RCC8^\sigma$ a la disyunción

$$\bigvee_{R \in RCC8} (\sigma \times \sigma)^{-1}[R^\Omega]$$

$$\begin{aligned}
L_1 : R^{\Omega_1} &= R^\Omega \text{ si } R \in \mathcal{R}_{RCC} \setminus \{NTPP, TPP\} \\
TPP^{\Omega_1} &= TPP^\Omega \cap TPP^\sigma \\
NTPP^{\Omega_1} &= NTPP^\Omega \cap NTPP^\sigma \\
I_1^{\Omega_1} &= (TPP^\Omega \cap (\bigvee RCC8^\sigma \setminus \{TPP^\sigma\})) \cup (NTPP^\Omega \cap (\bigvee RCC8^\sigma \setminus \{NTPP^\sigma\})) \\
\\
L_2 : R^{\Omega_2} &= R^\Omega \text{ si } R \in \mathcal{R}_{RCC} \setminus \{PP\} \\
PP^{\Omega_2} &= PP^\Omega \cap PP^\sigma \\
NTPP^{\Omega_2} &= NTPP^\Omega \cap PP^\sigma \\
TPP^{\Omega_2} &= TPP^\Omega \cap PP^\sigma \\
I_2^{\Omega_2} &= PP^\Omega \cap (\bigvee RCC8^\sigma \setminus \{PP^\sigma\}) \\
\\
L_3 : R^{\Omega_3} &= R^\Omega \text{ si } R \in \mathcal{R}_{RCC} \setminus \{TPPi, NTPPi\} \\
TPPi^{\Omega_3} &= TPPi^\Omega \cap TPPi^\sigma \\
NTPPi^{\Omega_3} &= NTPPi^\Omega \cap NTPPi^\sigma \\
I_3^{\Omega_3} &= (TPPi^\Omega \cap (\bigvee RCC8^\sigma \setminus \{TPPi^\sigma\})) \cup (NTPPi^\Omega \cap (\bigvee RCC8^\sigma \setminus \{NTPPi^\sigma\})) \\
\\
L_4 : R^{\Omega_4} &= R^\Omega \text{ si } R \in \mathcal{R}_{RCC} \setminus \{PPi\} \\
PPi^{\Omega_4} &= PPi^\Omega \cap PPi^\sigma \\
TPPi^{\Omega_4} &= TPPi^\Omega \cap PPi^\sigma \\
NTPPi^{\Omega_4} &= NTPPi^\Omega \cap PPi^\sigma \\
I_4^{\Omega_4} &= PPi^\Omega \cap (\bigvee RCC8^\sigma \setminus \{PPi^\sigma\}) \\
\\
L_5 : R^{\Omega_5} &= R^\Omega \text{ si } R \in \mathcal{R}_{RCC} \setminus \{PO\} \\
PO^{\Omega_5} &= PO^\Omega \cap PO^\sigma \\
I_5^{\Omega_5} &= PO^\Omega \cap (\bigvee RCC8^\sigma \setminus \{PO^\sigma\}) \\
\\
L_7 : R^{\Omega_7} &= R^\Omega \text{ si } R \in \mathcal{R}_{RCC} \setminus \{DC\} \\
DC^{\Omega_7} &= DC^\Omega \cap DC^\sigma \\
I_7^{\Omega_7} &= DC^\Omega \cap (\bigvee RCC8^\sigma \setminus \{DC^\sigma\}) \\
\\
L_8 : R^{\Omega_8} &= R^\Omega \wedge C^\sigma \text{ si } R \in \mathcal{R}_{RCC} \setminus \{DR, EC, DC\} \\
DR^{\Omega_8} &= DR^\Omega \cap DR^\sigma \\
EC^{\Omega_8} &= EC^\Omega \cap EC^\sigma \\
DC^{\Omega_8} &= DC^\Omega \cap DR^\sigma \\
I_8^{\Omega_8} &= (C^\Omega \cap DR^\sigma) \cup (DR^\Omega \cap C^\sigma)
\end{aligned}$$

Es fácil comprobar que, con estas interpretaciones, las relaciones $RCC \cup \{I_k\}$ tienen la estructura de retículo L_k en Ω_k . Un resultado importante es el siguiente, que afirma que I_6 no es interpretable en espacios topológicos con pulsación.

Teorema. 3.2.6 *En los espacios topológicos con pulsación no se puede interpretar L_6 .*

Demostración: $\top = C^{\Omega'} \cup DR^{\Omega'}$. Por un lado, $C^{\Omega} \subseteq C^{\sigma}$, por tanto, $C^{\Omega'} = C^{\sigma}$. Además, $DR^{\sigma} \subseteq DR^{\Omega}$, de donde $DR^{\Omega'} = DR^{\Omega}$. En cuanto a O , tenemos $O^{\Omega} \subseteq O^{\sigma}$, luego $O^{\Omega'} = O^{\sigma}$.

Y respecto a EC , sabemos que $EC^{\Omega'} = C^{\Omega'} \cap DR^{\Omega'} = C^{\sigma} \cap DR^{\Omega}$. De todo ello, se concluye que $I_6 = C^{\sigma} \setminus (O^{\sigma} \cup (C^{\sigma} \cap DR^{\Omega}))$.

Sean x, y tales que $C^{\sigma}(x, y)$. Es decir, $C(\sigma(x), \sigma(y))$. Si además, $\neg O^{\sigma}(x, y)$, se tendrá $EC(\sigma(x), \sigma(y))$, es decir, $C^{\sigma} \cap DR^{\Omega}(x, y)$. Por tanto, $I_6 = \perp$. ■

Una consecuencia importante del teorema anterior es el siguiente resultado sobre extensiones genéricas de RCC.

Corolario. 3.2.7 *Sea $I(x, y)$ una relación de indefinición y supongamos que T es una extensión de RCC que tiene un nuevo símbolo de función I . Se verifica:*

$$T \vdash \exists x(I(x, y) \wedge C(x, y)) \rightarrow \forall x(I(x, y) \rightarrow C(x, y))$$

Teorema. 3.2.8 *El retículo L_6 es interpretable en un espacio topológico con contracción.*

Demostración: Sea (Ω, ξ) un espacio topológico con contracción ξ . Definimos Ω_6 como la estructura en el lenguaje de $RCC + \{I_6\}$, donde para cada $R \in \mathcal{R}_{RCC}$, la interpretación de R en Ω_6 es la siguiente:

- $R^{\Omega_6} = R^{\Omega}$ si $R \in \{C, DR, EC, DC\}$
- $R^{\Omega_6} = R^{\Omega} \cap O^{\xi}$ si $R \in \mathcal{R}_{RCC} \setminus \{C, DR, EC, DC\}$
- $I_6^{\Omega_6} = O^{\Omega} \cap DR^{\xi}$

Es fácil comprobar que las relaciones así definidas respetan la estructura de retículo L_6 . ■

Nota. 3.2.9 *Las interpretaciones no son únicas, por ejemplo, con contracción también podemos interpretar el retículo.*

Ejemplo. 3.2.10 *En la figura 3.9 están representadas las diferentes interpretaciones de las relaciones de indefinición, para $a, b \in \mathcal{R}(\Omega)$. Por último, sólo queda comprobar que cada conjunto de relaciones tiene la estructura de retículo que muestra la figura correspondiente.*

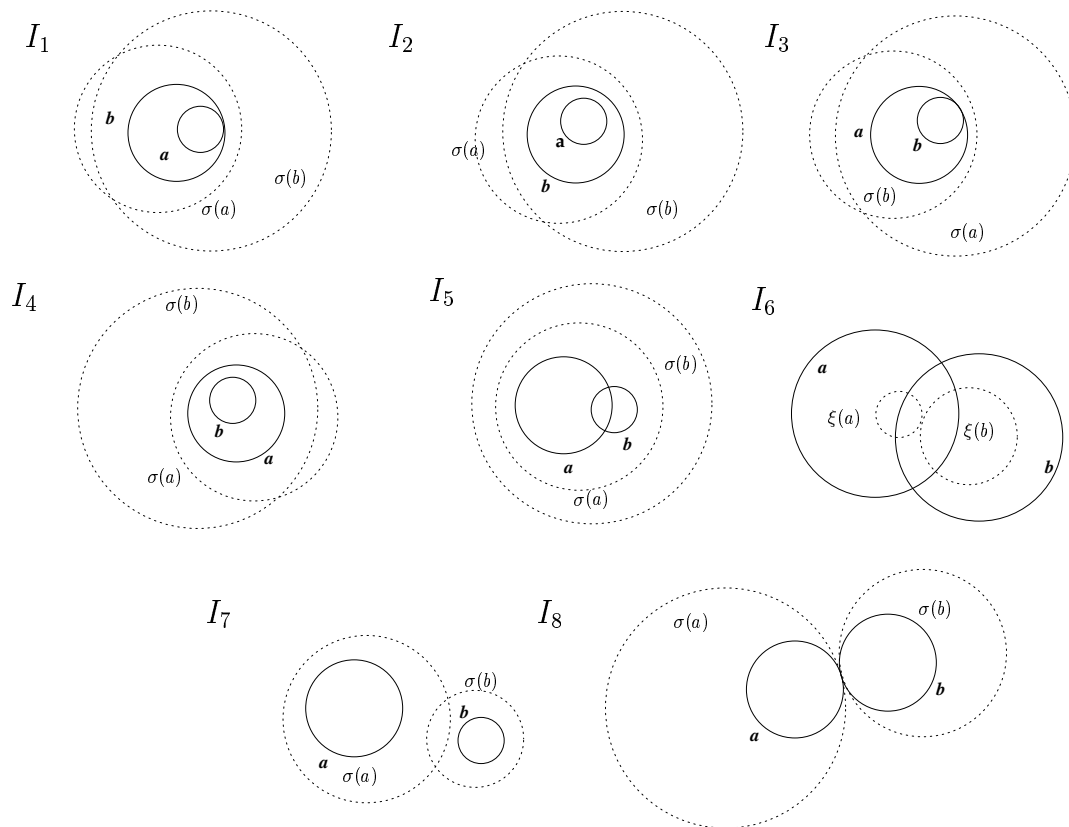


Figura 3.9: Interpretaciones posibles de las relaciones de indefinición

Teorema. 3.2.11 (De inserción ontológica de relaciones con indefinición). El conjunto de las relaciones para Ω_k , con $k \in \{1, 2, \dots, 8\}$, tiene la estructura de retículo dada en la figura correspondiente al modelo L_k .

Demostración: La prueba de este teorema consta de cuatro pasos:

* En primer lugar es necesario probar que $\forall R_1, R_2 \in RCC$ se tiene

$$R_1^\Omega \leq R_2^\Omega \text{ en } L_{RCC} \Rightarrow R_1^{\Omega_k} \leq R_2^{\Omega_k} \text{ en } L_k, \forall k \in \{1, 2, \dots, 7, 8\}$$

Esto equivale a demostrar que L_{RCC} se puede incorporar al retículo L_k , para cada $k \in \{1, 2, \dots, 8\}$.

* En segundo lugar hay que demostrar que, para cada $k \in \{1, 2, \dots, 8\}$, la relación I_k está en la posición que indica la figura correspondiente al retículo L_k . Para ello hay que comprobar que se verifican las siguientes condiciones:

1. $I_1^{\Omega_1} < PP^{\Omega_1}$
2. $I_2^{\Omega_2} < P^{\Omega_2}$
3. $I_3^{\Omega_3} < PP^i^{\Omega_3}$
4. $I_4^{\Omega_4} < P^i^{\Omega_4}$
5. $I_5^{\Omega_5} < O^{\Omega_5}$
6. $I_6^{\Omega_6} < C^{\Omega_6}$
7. $I_7^{\Omega_7} < DR^{\Omega_7}$

* A continuación, demostraremos que para cada $k \in \{1, 2, \dots, 8\}$, $RCC8 \cup \{I_k^{\Omega_k}\}$ forma un conjunto de relaciones disjuntas dos a dos.

* Por último, comprobamos que, para cada $k \in \{1, 2, \dots, 8\}$, el diagrama de Hasse de L_k representa la operación unión en el retículo de relaciones de Ω_k .

Veamos la prueba paso a paso.

Paso 1. Veamos que $(L_{RCC}, \leq) \subset (L_k, \leq)$ con $k \in \{1, 2, \dots, 8\}$.

$k = 1$: Las pruebas para cada $R \in \mathcal{R}_{RCC} \setminus \{NTPP, TPP\}$ son análogas entre sí. Lo mismo ocurre con las pruebas del resultado para TPP y $NTPP$. Por tanto, lo demostramos para una de ellas, en cada caso.

$$\begin{aligned} PO^{\Omega_1} \leq O^{\Omega_1} : \quad PO^{\Omega_1} &= PO^\Omega \cap \bigvee RCC8^\sigma \\ &\leq O^\Omega \cap \bigvee RCC8^\sigma = O^{\Omega_1} \\ NTPP^{\Omega_1} \leq PP^{\Omega_1} : \quad NTPP^{\Omega_1} &= NTPP^\Omega \cap NTPP^\sigma \\ &\leq PP^\Omega \cap \bigvee RCC8^\sigma = PP^{\Omega_1} \end{aligned}$$

$k = 2$: Las pruebas para cada $R \in \mathcal{R}_{RCC} \setminus \{PP, NTPP, TPP\}$ son análogas entre sí. Por tanto, demostramos el resultado para una de ellas, por ejemplo, veamos que $P^{\Omega_2} \leq O^{\Omega_2}$.

$$P^{\Omega_2} = P^{\Omega} \cap \bigvee RCC8^{\sigma} \leq O^{\Omega} \cap \bigvee RCC8^{\sigma} = O^{\Omega_2}$$

A continuación tenemos que comprobar las posiciones que ocupan PP , $NTPP$ y TPP .

$$PP^{\Omega_2} = PP^{\Omega} \cap PP^{\sigma} \leq P^{\Omega} \cap \bigvee RCC8^{\sigma} = P^{\Omega_2}$$

Los casos de $NTPP$ y TPP son análogos, así que vemos una de ellas.

$$NTPP^{\Omega_2} = NTPP^{\Omega} \cap PP^{\sigma} \leq PP^{\Omega} \cap PP^{\sigma} = PP^{\Omega_2}$$

$k = 3$: Este caso es idéntico a $k = 1$, cambiando Ω_1 , TPP y $NTPP$ por Ω_3 , $TPPi$ y $NTPPi$ respectivamente.

$k = 4$: Este caso es idéntico a $k = 2$, cambiando Ω_2 , PP , TPP y $NTPP$ por Ω_4 , PPi , $TPPi$ y $NTPPi$ respectivamente.

$k = 5$: Las pruebas para cada $R \in \mathcal{R}_{RCC} \setminus \{PO\}$ son análogas entre sí. Por tanto, demostramos el resultado para una de ellas, por ejemplo, veamos que $NTPP^{\Omega_5} \leq PP^{\Omega_5}$.

$$NTPP^{\Omega_5} = NTPP^{\Omega} \cap \bigvee RCC8^{\sigma} \leq PP^{\Omega} \cap \bigvee RCC8^{\sigma} = PP^{\Omega_5}$$

A continuación tenemos que comprobar la posición de PO .

$$PO^{\Omega_5} = PO^{\Omega} \cap PO^{\sigma} \leq O^{\Omega} \cap \bigvee RCC8^{\sigma} = O^{\Omega_5}$$

$k = 6$: En este caso se hace la interpretación de las relaciones con contracción. Las pruebas para cada $R \in \mathcal{R}_{RCC} \setminus \{C, DR, EC, DC\}$ son análogas entre sí. Por tanto, demostramos el resultado para una de ellas, por ejemplo, veamos que $PP^{\Omega_6} \leq P^{\Omega_6}$.

$$PP^{\Omega_6} = NTPP^{\Omega} \cap O^{\xi} \leq P^{\Omega} \cap O^{\xi} = P^{\Omega_6}$$

A continuación tenemos que comprobar las posiciones de C , DR , EC y DC . Veamos dos de ellas.

$$O^{\Omega_6} = O^{\Omega} \cap O^{\xi} \leq O^{\Omega} \leq C^{\Omega} = C^{\Omega_6}$$

$$EC^{\Omega_6} = EC^{\Omega} \leq DR^{\Omega} = DR^{\Omega_6}$$

$k = 7$: Comprobamos la posición de DC .

$$DC^{\Omega_7} = DC^{\Omega} \cap DC^{\sigma} \leq DR^{\Omega} \cap \bigvee RCC8^{\sigma} = DR^{\Omega_7}$$

$k = 8$: Comprobemos, por ejemplo la posición relativa de EC y DR .

$$EC^{\Omega_8} = EC^{\Omega} \cap EC^{\sigma} \leq DR^{\Omega} \cap DR^{\sigma} = DR^{\Omega_8}$$

Paso 2. En efecto, tenemos lo siguiente:

1.
$$\begin{aligned} I_1^{\Omega_1} &= (TPP^{\Omega} \cap (\bigvee RCC8^{\sigma} \setminus \{TPP^{\sigma}\})) \cup (NTPP^{\Omega} \cap (\bigvee RCC8^{\sigma} \setminus \{NTPP^{\sigma}\})) \\ &< (PP^{\Omega} \cap (\bigvee RCC8^{\sigma} \setminus \{TPP^{\sigma}\})) \cup (PP^{\Omega} \cap (\bigvee RCC8^{\sigma} \setminus \{NTPP^{\sigma}\})) \\ &= PP^{\Omega} \cap (\bigvee RCC8^{\sigma} \setminus \{TPP^{\sigma}\} \cup \bigvee RCC8^{\sigma} \setminus \{NTPP^{\sigma}\}) \\ &= PP^{\Omega} \cap \bigvee RCC8^{\sigma} = PP^{\Omega_1} \end{aligned}$$

2.
$$I_2^{\Omega_2} = PP^{\Omega} \cap (\bigvee RCC8^{\sigma} \setminus \{PP^{\sigma}\}) < P^{\Omega} \cap \bigvee RCC8^{\sigma} = P^{\Omega_2}$$

3. $I_3^{\Omega_3} < PPi^{\Omega_3}$: Este caso es idéntico al de $I_1^{\Omega_1}$, sustituyendo Ω_1, PP, TPP y $NTPP$ por $\Omega_3, PPi, TPPi$ y $NTPPi$, respectivamente.
4. $I_4^{\Omega_4} < Pi^{\Omega_4}$: Este caso es idéntico al de $I_2^{\Omega_2}$, sustituyendo Ω_2, P y PP por Ω_4, Pi y PPi , respectivamente.
5. $I_5^{\Omega_5} < O^{\Omega_5}$: Este caso es idéntico al de $I_2^{\Omega_2}$, sustituyendo Ω_2, P y PP por Ω_5, O y PO , respectivamente.
6. $I_6^{\Omega_6} = O^{\Omega} \cap DR^{\xi} < O^{\Omega} < C^{\Omega} = C^{\Omega_6}$
7. $I_7^{\Omega_7} < DR^{\Omega_7}$: Este caso es idéntico al de $I_2^{\Omega_2}$, sustituyendo Ω_2, P y PP por Ω_7, DR y DC , respectivamente.

Paso 3. En este paso basta comprobar que para cada $k \in \{1, 2, \dots, 8\}$, la relación $I_k^{\Omega_k}$ es disjunta con el resto de relaciones del retículo L_k . Esto es consecuencia inmediata de la interpretación de I_k .

$$\begin{aligned}
k = 1: I_1^{\Omega_1} \wedge TPP^{\Omega_1} &= \\
&(TPP^{\Omega} \cap (\bigvee RCC8^{\sigma} \setminus \{TPP^{\sigma}\})) \cup (NTPP^{\Omega} \cap (\bigvee RCC8^{\sigma} \setminus \{NTPP^{\sigma}\})) \cap \\
&TPP^{\Omega_1} = \\
&(TPP^{\Omega} \cap (\bigvee RCC8^{\sigma} \setminus \{TPP^{\sigma}\}) \cap TPP^{\Omega} \cap TPP^{\sigma}) \cup \\
&(NTPP^{\Omega} \cap (\bigvee RCC8^{\sigma} \setminus \{NTPP^{\sigma}\}) \cap TPP^{\Omega} \cap TPP^{\sigma}) = \perp \\
&\text{por ser } (\bigvee RCC8^{\sigma} \setminus TPP^{\sigma}) \cap TPP^{\sigma} = \perp \text{ y } NTPP^{\Omega} \cap TPP^{\Omega} = \perp \\
&\text{De forma análoga se prueba } I_1^{\Omega_1} \wedge NTPP^{\Omega_1} = \perp
\end{aligned}$$

Ahora veamos que I^{Ω_1} es disjunta con PO^{Ω_1}

$$\begin{aligned}
I_1^{\Omega_1} \wedge PO^{\Omega_1} &= I_1^{\Omega_1} \wedge (PO^{\Omega} \cap \bigvee RCC8^{\sigma}) = \\
&(TPP^{\Omega} \cap (\bigvee RCC8^{\sigma} \setminus \{TPP^{\sigma}\})) \cup (NTPP^{\Omega} \cap (\bigvee RCC8^{\sigma} \setminus \{NTPP^{\sigma}\})) \cap \\
&PO^{\Omega} \cap \bigvee RCC8^{\sigma} = \\
&(TPP^{\Omega} \cap PO^{\Omega} \cap (\bigvee RCC8^{\sigma} \setminus \{TPP^{\sigma}\}) \cap \bigvee RCC8^{\sigma}) \cap \\
&(NTPP^{\Omega} \cap PO^{\Omega} \cap (\bigvee RCC8^{\sigma} \setminus \{NTPP^{\sigma}\}) \cap \bigvee RCC8^{\sigma}) = \perp \\
&\text{por ser } (\bigvee RCC8^{\sigma} \setminus NTPP^{\sigma}) \cap NTPP^{\sigma} = \perp \text{ y } TPP^{\Omega} \cap PO^{\Omega} = \perp \\
&\text{De igual forma se prueba para } R \in \mathcal{R}_{RCC} \setminus \{TPP, NTPP\}
\end{aligned}$$

$$\begin{aligned}
k = 2: I_2^{\Omega_2} \wedge PP^{\Omega_2} &= PP^{\Omega} \cap \bigvee RCC8^{\sigma} \setminus \{PP^{\sigma}\} \cap PP^{\Omega} \cap PP^{\sigma} = \perp \\
&\text{por ser } (\bigvee RCC8^{\sigma} \setminus PP^{\sigma}) \cap PP^{\sigma} = \perp \\
&\text{Lo mismo ocurre con } NTPP^{\Omega_2} \text{ y } TPP^{\Omega_2} :
\end{aligned}$$

Veamos el caso de PO^{Ω_2} :

$$\begin{aligned}
I_2^{\Omega_2} \wedge PO^{\Omega_2} &= PP^{\Omega} \cap \bigvee RCC8^{\sigma} \setminus \{PP^{\sigma}\} \cap PO^{\Omega} \cap \bigvee RCC8^{\sigma} = \perp \\
&\text{por ser } PP^{\Omega} \cap PO^{\Omega} = \perp
\end{aligned}$$

De forma análoga se prueba para el resto de las relaciones de \mathcal{R}_{RCC}

- El resto de los casos, excepto el de $k = 6$, se resuelve de forma similar a los dos anteriores.

$$\begin{aligned}
k = 6: I_6^{\Omega_6} \wedge EC^{\Omega_6} &= O^{\Omega} \cap DR^{\xi} \cap EC^{\Omega} = \perp \text{ por ser } O^{\Omega} \cap EC^{\Omega} = \perp. \text{ Por otra} \\
&\text{parte, } I_6^{\Omega_6} \wedge O^{\Omega_6} = O^{\Omega} \cap DR^{\xi} \cap O^{\Omega} \cap O^{\xi} = \perp \text{ por ser } DR^{\xi} \cap O^{\xi} = \perp \\
&\text{ya que } \xi \text{ es una contracción.}
\end{aligned}$$

Paso 4. Por último:

- $k = 1$: Tenemos que demostrar que $TPP^{\Omega_1} \vee NTPP^{\Omega_1} \vee I_1^{\Omega_1} = PP^{\Omega_1}$:

$$TPP^{\Omega_1} \vee NTPP^{\Omega_1} \vee I_1^{\Omega_1} = (TPP^{\Omega} \cap TPP^{\sigma}) \cup (NTPP^{\Omega} \cap NTPP^{\sigma}) \cup (TPP^{\Omega} \cap (\bigvee RCC8^{\sigma} \setminus TPP^{\sigma})) \cup (NTPP^{\Omega} \cap (\bigvee RCC8^{\sigma} \setminus NTPP^{\sigma})) = (TPP^{\Omega} \cap \bigvee RCC8^{\sigma}) \cup (NTPP^{\Omega} \cap \bigvee RCC8^{\sigma}) = (TPP^{\Omega} \cup NTPP^{\Omega}) \cap \bigvee RCC8^{\sigma} = PP^{\Omega} \cap \bigvee RCC8^{\sigma} = PP^{\Omega_1}$$
- $k = 2$: Debemos probar dos resultados: $NTPP^{\Omega_2} \vee TPP^{\Omega_2} = PP^{\Omega_2}$ y $PP^{\Omega_2} \vee EQ^{\Omega_2} \vee I_2^{\Omega_2} = P^{\Omega_2}$:
- $NTPP^{\Omega_2} \vee TPP^{\Omega_2} = (NTPP^{\Omega} \cap PP^{\sigma}) \cup (TPP^{\Omega} \cap PP^{\sigma}) = (NTPP^{\Omega} \cup TPP^{\Omega}) \cap PP^{\sigma} = PP^{\Omega} \cap PP^{\sigma} = PP^{\Omega_2}$
 - $PP^{\Omega_2} \vee EQ^{\Omega_2} \vee I_2^{\Omega_2} = (PP^{\Omega} \cap PP^{\sigma}) \cup (PP^{\Omega} \cap (\bigvee RCC8^{\sigma} \setminus PP^{\sigma})) \cup (EQ^{\Omega} \cap \bigvee RCC8^{\sigma}) = (PP^{\Omega} \cap \bigvee RCC8^{\sigma}) \cup (EQ^{\Omega} \cap \bigvee RCC8^{\sigma}) = P^{\Omega} \cap \bigvee RCC8^{\sigma} = P^{\Omega_2}$
- $k = 3$: Tenemos que demostrar que $TPPi^{\Omega_3} \vee NTPPi^{\Omega_3} \vee I_3^{\Omega_3} = PPi^{\Omega_3}$. La prueba es idéntica a la del caso $k = 1$, sustituyendo PP, TPP y $NTPP$ por $PPi, TPPi$ y $NTPPi$ respectivamente.
- $k = 4$: Lo mismo ocurre en este caso. Basta sustituir en la prueba del caso $k = 2$ las relaciones P, PP, TPP y $NTPP$ por $Pi, PPi, TPPi$ y $NTPPi$ respectivamente.
- $k = 5$: Comprobemos que se verifica $I_5^{\Omega_5} \vee PO^{\Omega_5} \vee P^{\Omega_5} \vee Pi^{\Omega_5} = O^{\Omega_5}$:

$$I_5^{\Omega_5} \vee PO^{\Omega_5} \vee P^{\Omega_5} \vee Pi^{\Omega_5} = (PO^{\Omega} \cap (\bigvee RCC8^{\sigma} \setminus PO^{\sigma})) \cup (PO^{\Omega} \cap PO^{\sigma}) \cup (P^{\Omega} \cap \bigvee RCC8^{\sigma}) \cup (Pi^{\Omega} \cap \bigvee RCC8^{\sigma}) = (PO^{\Omega} \cap \bigvee RCC8^{\sigma}) \cup (P^{\Omega} \cap \bigvee RCC8^{\sigma}) \cup (Pi^{\Omega} \cap \bigvee RCC8^{\sigma}) = O^{\Omega} \cap \bigvee RCC8^{\sigma} = O^{\Omega_5}$$
- $k = 6$: En este caso, vamos a demostrar, como ejemplo, dos resultados. Los demás se obtienen de igual manera.
- $O^{\Omega_6} \vee I_6^{\Omega_6} \vee EC^{\Omega_6} = C^{\Omega_6}$:

$$O^{\Omega_6} \vee I_6^{\Omega_6} \vee EC^{\Omega_6} = (O^{\Omega} \cap O^{\xi}) \cup (O^{\Omega} \cap DR^{\xi}) \cup EC^{\Omega} = (O^{\Omega} \cap (O^{\xi} \cup DR^{\xi})) \cup EC^{\Omega} = O^{\Omega} \cup EC^{\Omega} = C^{\Omega} = C^{\Omega_6}$$
 - $PO^{\Omega_6} \vee P^{\Omega_6} \vee PPi^{\Omega_6} = O^{\Omega_6}$

$$PO^{\Omega_6} \vee P^{\Omega_6} \vee PPi^{\Omega_6} = (PO^{\Omega} \cap O^{\xi}) \cup (P^{\Omega} \cap O^{\xi}) \cup (Pi^{\Omega} \cap O^{\xi}) = (PO^{\Omega} \cup P^{\Omega} \cup Pi^{\Omega}) \cap O^{\xi} = O^{\Omega} \cap O^{\xi} = O^{\Omega_6}$$
- $k = 7$: Se verifica $EC^{\Omega_7} \vee DC^{\Omega_7} \vee I_7^{\Omega_7} = DR^{\Omega_7}$:

$$EC^{\Omega_7} \vee DC^{\Omega_7} \vee I_7^{\Omega_7} = (EC^{\Omega} \cap \bigvee RCC8^{\sigma}) \cup (DC^{\Omega} \cap DC^{\sigma}) \cup (DC^{\Omega} \cap (\bigvee RCC8^{\sigma} \setminus DC^{\sigma})) = (EC^{\Omega} \cap \bigvee RCC8^{\sigma}) \cup (DC^{\Omega} \cap \bigvee RCC8^{\sigma}) = DR^{\Omega} \cap \bigvee RCC8^{\sigma} = DR^{\Omega_7}$$
- $k = 8$: La prueba consiste en probar varios resultados:
- $EC^{\Omega_8} \vee DC^{\Omega_8} = (EC^{\Omega} \cap EC^{\sigma}) \cup (DC^{\Omega} \cap DR^{\sigma}) \subseteq DR^{\Omega} \cap DR^{\sigma} = DR^{\Omega_8}$

La otra contención también se tiene. Sea $(a, b) \in DR^\Omega \cap DR^\sigma$. Por estar en DR^Ω , se cumplirá que $(a, b) \in DC^\Omega$, y por tanto $(a, b) \in (DC^\Omega \cap DR^\sigma)$, o bien, se cumplirá que $(a, b) \in EC^\Omega$, y por tanto $(a, b) \in (EC^\Omega \cap EC^\sigma)$, de donde $(a, b) \in EC^{\Omega_8} \vee DC^{\Omega_8}$.

- $C^{\Omega_8} \vee DR^{\Omega_8} \vee I_8^{\Omega_8} = (C^\Omega \cap C^\sigma) \cup (DR^\Omega \cap DR^\sigma) \cup (C^\Omega \cap DR^\sigma) \cup (DR^\Omega \cap C^\sigma) = ((C^\Omega \cap DR^\Omega) \cap C^\sigma) \cup ((C^\Omega \cap DR^\Omega) \cap DR^\sigma) = \top$
- Hay que comprobar que las relaciones de $\mathcal{R}_{RCC} \setminus \{DR, EC, DC\}$ mantienen la estructura también. Por ejemplo:
 $PP^{\Omega_8} \vee EQ^{\Omega_8} = (PP^\Omega \cap C^\sigma) \cup (EQ^\Omega \cap C^\sigma) = (PP^\Omega \cup EQ^\Omega) \cap C^\sigma = P^\Omega \cap C^\sigma = P^{\Omega_8}$

■

Como consecuencia del teorema que acabamos de demostrar, tenemos el siguiente resultado.

Corolario. 3.2.12 *El conjunto de relaciones $RCC8 + \{I_k\}$ constituye un JEPD con la interpretación Ω_k , para $k = \{1, 2, \dots, 8\}$.*

3.3. Tabla de composición de una extensión interpretable

Para concluir este capítulo, demostraremos que nuestro método generaliza los métodos basados en el estudio de las tablas de composición, comprobando que de la interpretación elegida para la extensión se puede deducir la tabla de composición correspondiente. Como en $RCC8$, es posible calcular las tablas de composición de cada conjunto de relaciones L_k , $k = \{1, \dots, 8\}$. Es esta sección presentamos, a modo de ilustración, la correspondiente a L_1 , pero es evidente que esta misma construcción es posible para el resto de las extensiones.

Teorema. 3.3.1 *La tabla de composición de las relaciones de $RCC8 + \{I_1\}$ es la que aparece descrita en el cuadro 3.1.*

Observaciones. 3.3.2 La parte de la tabla que corresponde a la composición de las relaciones R_1^Ω, R_2^Ω donde $R_1, R_2 \in RCC8$, coincide con la tabla obtenida para $RCC8$ (ver la tabla 2.1), exceptuando los siguientes casos:

- Si de la composición de dos relaciones R_1, R_2 de $RCC8$ se obtienen TPP ó $NTPP$ (ó ambas), ahora aparecen TPP ó $NTPP$ (ó ambas), además de I_1 .

$R2(b, c)$ $R1(a, b)$	DC	EC	PO	TPP	$NTPP$	$TPPi$	$NTPPi$	EQ	I_1
DC	$RCC8[I_1]$	$DC, EC,$ $PO, TPP,$ $NTPP, I_1$	$DC, EC,$ $PO, TPP,$ $NTPP, I_1$	$DC, EC,$ $PO, TPP,$ $NTPP, I_1$	$DC, EC,$ $PO, TPP,$ $NTPP, I_1$	DC	DC	DC	$DC, EC,$ $PO, TPP,$ $NTPP, I_1$
EC	$DC, EC,$ $PO, TPPi,$ $NTPPi$	$DC, EC, PO,$ $TPP, TPPi,$ EQ, I_1	$DC, EC,$ $PO, TPP,$ $NTPP, I_1$	$EC, PO,$ $TPP,$ $NTPP, I_1$	$PO, TPP,$ $NTPP, I_1$	DC, EC	DC	EC	$EC, PO,$ $TPP,$ $NTPP, I_1$
PO	$DC, EC,$ $PO, TPPi,$ $NTPPi$	$DC, EC,$ $PO, TPPi,$ $NTPPi$	$RCC8[I_1]$	$PO, TPP,$ $NTPP, I_1$	$PO, TPP,$ $NTPP, I_1$	$DC, EC,$ $PO, TPPi,$ $NTPPi$	$DC, EC,$ $PO, TPPi,$ $NTPPi$	PO	$PO, TPP,$ $NTPP, I_1$
TPP	DC	DC, EC	$DC, EC,$ $PO, TPP,$ $NTPP, I_1$	$TPP,$ $NTPP, I_1$	$NTPP$	$DC, EC,$ $PO, TPP,$ $TPPi, EQ,$ I_1	$DC, EC,$ $PO, TPPi,$ $NTPPi$	TPP	$TPP,$ $NTPP,$ I_1
$NTPP$	DC	DC	$DC, EC,$ $PO, TPP,$ $NTPP, I_1$	$NTPP$	$NTPP$	$DC, EC,$ $PO, TPP,$ $NTPP, I_1$	$RCC8[I_1]$	$NTPP$	$NTPP, I_1$
$TPPi$	$DC, EC,$ $PO, TPPi,$ $NTPPi$	$EC, PO,$ $TPPi,$ $NTPPi$	$PO,$ $TPPi,$ $NTPPi$	$PO, EQ,$ $TPP,$ $TPPi$	$PO, TPP,$ $NTPP, I_1$	$TPPi,$ $NTPPi$	$NTPPi$	$TPPi$	$PO, EQ,$ $TPP, TPPi,$ $NTPP, I_1$
$NTPPi$	$DC, EC,$ $PO, TPPi,$ $NTPPi$	$PO,$ $TPPi,$ $NTPPi$	$PO,$ $TPPi,$ $NTPPi$	$PO,$ $TPPi,$ $NTPPi$	$PO, TPPi,$ $TPP, NTTP,$ $NTPPi, EQ,$ I_1	$NTPPi$	$NTPPi$	$NTPPi$	$PO, TPPi,$ $TPP, NTTP,$ $NTPPi, EQ,$ I_1
EQ	DC	EC	PO	TPP	$NTPP$	$TPPi$	$NTPPi$	EQ	I_1
I_1	DC	DC, EC	$DC, EC,$ $PO, TPP,$ $NTPP, I_1$	$TPP,$ $NTPP, I_1$	$NTPP, I_1$	$DC, EC,$ $PO, TPP,$ $EQ,$ $NTPP, I_1$	$RCC8[I_1]$	I_1	$TPP,$ $NTPP, I_1$

Cuadro 3.1: Tabla de composición para L_1

- Como consecuencia, si en la tabla de composición de $RCC8$ el resultado de componer dos relaciones es $RCC8$, ahora, el resultado es el conjunto $RCC8 + \{I_1\}$, que hemos denotado por $RCC8[I_1]$.

Finalmente, vamos a detallar, como ejemplo, una de las composiciones.

- $I_1^{\Omega_1}(a, b) \wedge I_1^{\Omega_1}(b, c) =$
 $((TPP^{\Omega}(a, b) \cap (\bigvee RCC8^{\sigma} \setminus \{TPP^{\sigma}\})(a, b))$
 $\cup (NTPP^{\Omega}(a, b) \cap (\bigvee RCC8^{\sigma} \setminus \{NTPP^{\sigma}\})(a, b)))$
 \cap
 $((TPP^{\Omega}(b, c) \cap (\bigvee RCC8^{\sigma} \setminus \{TPP^{\sigma}\})(b, c))$
 $\cup (NTPP^{\Omega}(b, c) \cap (\bigvee RCC8^{\sigma} \setminus \{NTPP^{\sigma}\})(b, c))) =$
 $(NTPP^{\Omega}(a, c) \cap (\bigvee RCC8^{\sigma}(a, c)))$
 \cup
 $(TPP^{\Omega}(a, c) \cap (\bigvee RCC8^{\sigma}(a, c))) =$
 $(NTPP^{\Omega}(a, c) \cap (NTPP^{\sigma}(a, b) \cup (\bigvee RCC8^{\sigma} \setminus \{NTPP^{\sigma}\})(a, c)))$
 \cup
 $(TPP^{\Omega}(a, c) \cap (TPP^{\sigma}(a, b) \cup (\bigvee RCC8^{\sigma} \setminus \{TPP^{\sigma}\})(a, c))) =$
 $(NTPP^{\Omega}(a, c) \cap NTPP^{\sigma}(a, b))$
 $\cup (NTPP^{\Omega}(a, c) \cap (\bigvee RCC8^{\sigma} \setminus \{NTPP^{\sigma}\})(a, c))$
 \cup
 $(TPP^{\Omega}(a, c) \cap TPP^{\sigma}(a, b))$
 $\cup (TPP^{\Omega}(a, c) \cap (\bigvee RCC8^{\sigma} \setminus \{TPP^{\sigma}\})(a, c)) =$
 $NTPP^{\Omega_1}(a, c) \cup (NTPP^{\Omega}(a, c) \cap (\bigvee RCC8^{\sigma} \setminus \{NTPP^{\sigma}\})(a, c))$
 \cup
 $TPP^{\Omega_1}(a, c) \cup (TPP^{\Omega}(a, c) \cap (\bigvee RCC8^{\sigma} \setminus \{TPP^{\sigma}\})(a, c)) =$
 $NTPP^{\Omega_1}(a, c) \vee TPP^{\Omega_1}(a, c) \vee I_1^{\Omega_1}(a, c)$
- Por tanto, la composición de I_1 con I_1 es $NTPP \vee TPP \vee I_1$

3.4. Interpretación de las relaciones en la teoría “huevo-yema”

Como ya hemos comentado, la interpretación topológica presentada no es la única posible. En esta sección vamos a basarnos en el trabajo de Cohn y Gotts

[16] para dar una nueva interpretación de las relaciones I_k , $k = \{1, \dots, 5, 7, 8\}$. Dicho estudio versa sobre la representación “huevo-yema”(egg-yolk) de regiones con frontera indeterminada (regiones vagas) mediante una adaptación de la teoría RCC, ya que esta teoría no fue concebida inicialmente para representar y razonar con este tipo de regiones.

La adaptación que realizan los autores consiste en representar una región vaga A , con frontera indeterminada por tanto, mediante un par de subregiones concéntricas \tilde{a} y a que sí poseen fronteras determinadas. Estas fronteras limitan la indeterminación de la región vaga A a la que ambas regiones representan. Así, podremos expresar el mismo tipo de ideas sobre dos regiones vagas (cada una representada por dos subregiones) que sobre regiones con frontera definida, por ejemplo, que una contiene a otra, que se solapan, que son disjuntas, etc. De esta forma, una región vaga representa el espacio ocupado (el área abarcada) por una de las dos subregiones, las dos a la vez, o bien por una de ellas y no la otra. De las dos subregiones concéntricas, llamamos *contracción* o *yema* a la interior, y es la que denotamos \tilde{a} . La idea es similar a la de representar una región en un espacio topológico con pulsación. Debido a esta similitud, tampoco ahora interpretaremos la relación I_6 .

3.4.1. Interpretación con pulsación en el paradigma huevo/yema

Dadas dos regiones vagas A, B , consideramos ahora, según el formalismo “huevo-yema”, las cinco posibles relaciones de la teoría RCC5, entre la pulsación de una de ellas (\bar{a}) y la otra, b , (o su pulsación \bar{b}), y análogamente, entre a y b (o su pulsación \bar{b}). La razón por la que no es adecuado considerar RCC8 es evidente: si no se conoce el perfil exacto de cada región, no tiene sentido considerar tangencialidad.

Realizando todas las combinaciones, los autores obtienen 46 relaciones posibles entre dos regiones vagas. Esas relaciones son las que aparecen en la figura 3.11, figura que tomaremos como referencia en posteriores capítulos.

Dada $R \in RCC5$, representaremos por \bar{R} el conjunto $(\bar{\cdot} \times \bar{\cdot})^{-1}[R]$;

$$(a, b) \in \bar{R} \iff (\bar{a}, \bar{b}) \in R$$

De esta forma, incluimos RCC5 en la teoría.

Hechas estas consideraciones, podemos interpretar las relaciones de RCC5.

- La interpretación de DR coincide con la idea intuitiva de estar “desconectados”, en el sentido de que si las pulsaciones \bar{a} y \bar{b} , de dos regiones A y B están desconectadas, entonces a y b , lo estarán también. Esta situación es la representada por la relación 1 de la figura 3.11. Así, $DR = \{1\}$.

- $\overline{PP} = \{(A, B) : PP(\bar{a}, \bar{b})\}$, que a su vez es la interpretación de I_1 por su posición en L_1 . Por tanto, $I_1 = \overline{PP} = \{8, 13, 22, 24, 26, 34, 35, 36, 37, 38, 41\}$.
- La posición de I_2 en L_2 sugiere la interpretación:
 $I_2 = \overline{PP} \cup \overline{EQ} = \{8, 13, 22, 24, 26, 34, 35, 36, 37, 38, 41, 42, 43, 44, 45, 46\}$,
 siendo $\overline{EQ} = \{42, 43, 44, 45, 46\}$
- A continuación, y por ser I_3 y I_4 simétricas de I_1 y I_2 respectivamente, tenemos:
 $I_3 = \overline{PPi} = \{7, 12, 21, 23, 25, 28, 29, 30, 31, 33, 40\}$
 $I_4 = \overline{PPi} \cup \overline{EQ} = \{7, 12, 21, 23, 25, 28, 29, 30, 31, 33, 40, 42, 43, 44, 45, 46\}$.
- $\overline{PO} = \{2, 3, 4, 5, 6, 9, 10, 11, 14, 15, 16, 17, 18, 19, 20, 27, 32, 39\}$, por tanto, $I_5 = I_2 \cup I_4 \cup \overline{PO} = \{2, \dots, 46\}$
- $I_7 = \overline{DR} = \{1\}$.
- Finalmente, al ser I_8 la indeterminación total, estará representado por cualquiera de las figuras de 3.11, $I_8 = \{1, 2, \dots, 46\}$.

Como ya vimos, véase 3.2.6, interpretar la relación I_6 con pulsación es imposible, pues no pueden ser discretas las pulsaciones mientras las regiones se solapan. Por otra parte, se tiene:

$$I_8 = \bigcup_{k \in \{1, 2, 3, 4, 5, 7\}} I_k = I_5 \cup I_7$$

De esta forma hemos demostrado el siguiente teorema:

Teorema. 3.4.1 *En la interpretación huevo-yema anteriormente descrita, las relaciones $I_1, \dots, I_5, I_7, I_8$ componen el retículo de la figura 3.10, junto con RCC5.*

Corolario. 3.4.2 *El conjunto de relaciones $\{I_1, \overline{EQ}, I_3, \overline{PO}, I_7\}$ forma un JEPD, al igual que $\{I_2, I_4, \overline{PO}, I_7\}$ y $\{I_5, I_7\}$.*

3.4.2. Interpretación de la contracción

Aunque para el objetivo de esta memoria utilizaremos fundamentalmente la interpretación por pulsación, es posible también interpretar las nuevas relaciones por contracción. Recuérdese que dadas dos regiones vagas A, B , el formalismo “huevo-yema” permite sólo cinco posibles relaciones, las cinco relaciones de la teoría RCC5, entre la contracción de una de ellas (\bar{a}) y las subregiones que representan a la otra (b o su contracción \bar{b}), y análogamente, entre a y la subregiones que representan a B (b o su contracción \bar{b}).

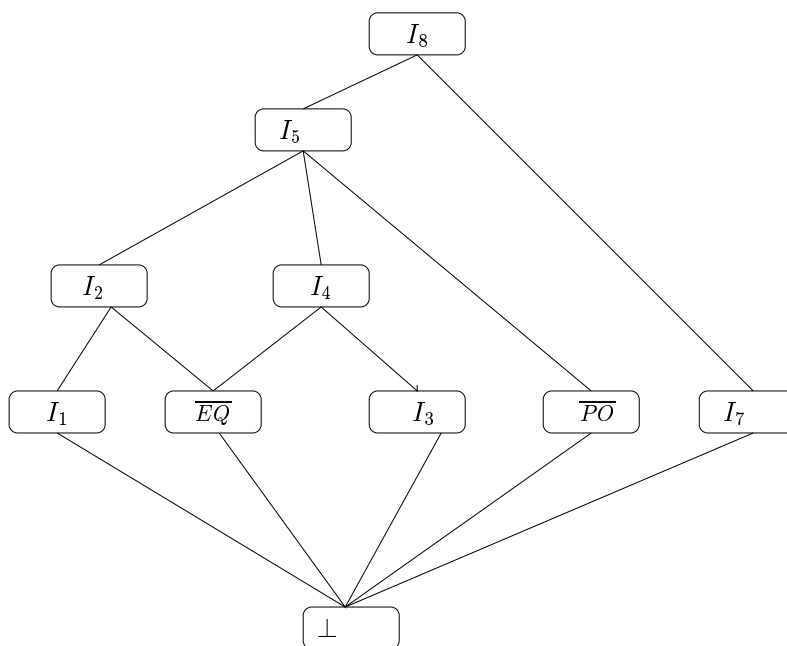


Figura 3.10: Retículo de las relaciones I_j interpretadas con contracción

Dada $R \in RCC5$, representaremos por \tilde{R} el conjunto $(\sim \times \sim)^{-1}[R]$, es decir

$$(a, b) \in \tilde{R} \iff (\tilde{a}, \tilde{b}) \in R$$

De esta forma, se incluye de manera natural RCC5 en la teoría. En la figura 3.11, el huevo a y su yema \tilde{a} estarán representados por la figura de línea intermitente y b y su yema \tilde{b} por la de línea punteada.

Con estas consideraciones, la interpretación de RCC5 y de las nuevas relaciones de indefinición son las siguientes:

- La interpretación de PP coincide con la idea intuitiva de ser “parte propia”. En nuestro caso, si una región A es parte propia de otra B , entonces a y \tilde{a} son parte propia de \tilde{b} y, por consiguiente, de b . Esta situación es la representada por la relación 24 de la figura 3.11. Así, $PP = \{24\}$. De manera similar se pueden interpretar las restantes relaciones de RCC5.
- Por otra parte, $\widetilde{PP} = \{A, B : PP(\tilde{a}, \tilde{b})\}$, que a su vez es la interpretación de I_1 por su posición en L_1 . Por tanto, $I_1 = \widetilde{PP} = \{18, 24, 26, 32, 33, 37, 38, 45\}$.
- La posición de I_2 en L_2 sugiere la interpretación:

$$I_2 = \widetilde{PP} \cup \widetilde{EQ} = \{18, 24, 26, 32, 33, 37, 38, 39, 40, 41, 45, 46\}, \text{ siendo } \widetilde{EQ} = \{39, 40, 41, 46\}$$

- Por simetría, I_3 y I_4 se definen de manera similar a I_1 y I_2 respectivamente:

$$I_3 = \widetilde{PPi} = \{17, 23, 25, 27, 30, 31, 36, 44\}$$

$$I_4 = \widetilde{PPi} \cup \widetilde{EQ} = \{17, 23, 25, 27, 30, 31, 36, 39, 40, 41, 44, 46\}.$$

- $\widetilde{PO} = \{14, 15, 16, 20, 21, 22, 29, 35, 43\}$, por tanto, $I_5 = I_2 \cup I_4 \cup \widetilde{PO} = \{14, 15, 16, 17, 18, 20, 21, 22, 23, 24, 25, 26, 27, 29, 30, 31, 32, 33, 35, 36, 37, 38, 39, 40, 41, 43, 44, 45, 46\}$
- $I_7 = \widetilde{DR} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 19, 28, 34, 42\}$.
- I_6 no es más que un subconjunto de I_7 , ya que contiene las figuras en las que las regiones se solapan pero cuyas contracciones son discretas. Así, $I_6 = \{2, 3, 4, 5, 6, 9, 10, 11, 19\}$
- Finalmente, al ser I_8 la indeterminación total, estará representado por cualquiera de las figuras de 3.11, $I_8 = \{1, 2, \dots, 46\}$.

Como hemos visto, interpretar la relación I_6 no es necesario, aunque sí es posible, pues $I_8 = \bigcup_{k \in \{1,2,3,4,5,7\}} I_k$, que por otra parte coincide con $I_5 \cup I_7$.

Esta observación es una parte del siguiente resultado:

Teorema. 3.4.3 *En la interpretación huevo-yema anteriormente descrita, las relaciones I_1, \dots, I_8 componen el retículo de la figura 3.12, junto con $RCC5$.*

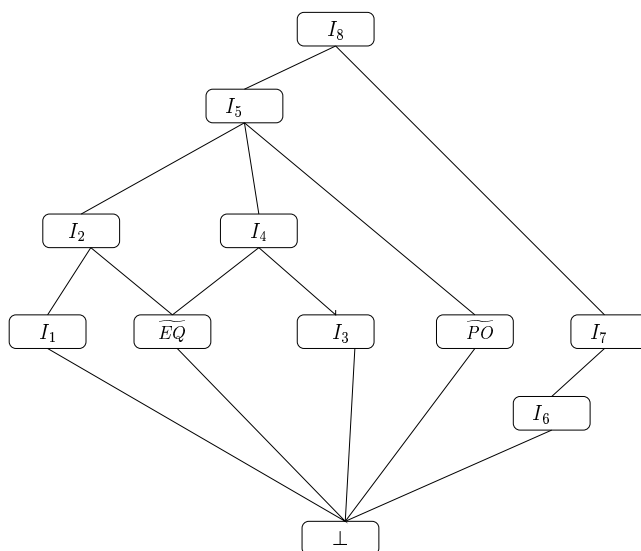


Figura 3.12: Retículo de las relaciones $\bigcup_k I_k \cup RCC5$

Corolario. 3.4.4 *En la interpretación anteriormente descrita, el conjunto de relaciones $\{I_1, \widetilde{EQ}, I_3, \widetilde{PO}, I_7\}$ forma un JEPD, al igual que $\{I_2, I_4, \widetilde{PO}, I_7\}$ y $\{I_5, I_7\}$.*

3.5. Interpretación con conjuntos del tipo *Rough*

Es posible interpretar la noción de aproximación utilizando conjuntos tipo rough [56]. La teoría de conjuntos rough proporciona una forma de aproximar subconjuntos de un conjunto dotado de una partición o relación de equivalencia.

En razonamiento espacial no es necesario aproximar subconjuntos de un conjunto, sino partes de un conjunto que posee estructura geométrica o topológica. Dado un conjunto X con una partición $\{X_i : i \in I\}$, un subconjunto $b \subseteq X$ se aproxima mediante una función $\phi_b : I \mapsto \{\text{fo}, \text{po}, \text{no}\}$, donde los valores $\{\text{fo}, \text{po}, \text{no}\}$ representan respectivamente las ideas de solapamiento total, solapamiento parcial y no solapamiento. Así, $\phi_b(i) = \text{fo}$ si $X_i \subseteq b$, $\phi_b(i) = \text{no}$ si $X_i \cap b = \emptyset$ y $\phi_b(i) = \text{po}$ en otro caso. Es decir, la función ϕ_b indica el grado de solapamiento que existe entre el subconjunto $b \subseteq X$ y los elementos a_i de la partición de X en cuestión.

En relación con esto, \underline{b} y \overline{b} pueden redefinirse en este contexto como:

$$\underline{b} = \bigcup \{X_i : \phi_b(i) = \text{fo}\},$$

$$\overline{b} = \bigcup \{X_i : \phi_b(i) \neq \text{no}\}.$$

Un primer enfoque es el trabajo sobre conjuntos rough [63] que realizaron L. Vigneron y A. Wasilewska. Sin embargo, esta aproximación sólo sirve para espacios topológicos en los que los abiertos son a la vez cerrados y viceversa. Por tanto, dicho estudio se basa en espacios topológicos totalmente disconexos, y no es éste nuestro caso.

Para dar una nueva interpretación de las relaciones de indefinición $\{I_k : k = 1 \dots 8\}$ que obtuvimos en la secciones anteriores, utilizando nociones rough, generalizaremos la noción de partición:

Definición. 3.5.1 *Un espacio topológico Ω se dice **DR-Rough** si existe una familia $\mathcal{F} = \{X_i\}_{i \in \mathbb{N}}$ de subconjuntos de Ω tal que:*

1. $\forall i \in \mathbb{N} : X_i$ es un cerrado regular
2. $\forall i \neq j : \Omega \models DR(X_i, X_j)$
3. $\forall J \subseteq \mathbb{N}$ el conjunto $\bigcup_{j \in J} X_j$ es regular

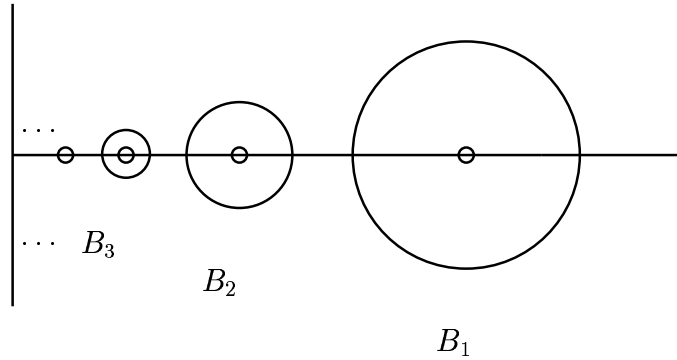


Figura 3.13: La unión de conjuntos regulares no es regular, en general

$$4. \bigcup_{i \in \mathbb{N}} X_i = \Omega$$

A la familia \mathcal{F} se le llama **recubrimiento DR-Rough**.

Notas. 3.5.2 En la definición anterior, es necesaria la condición 3, ya que, en general, la unión de conjuntos regulares no es regular. Veamos el ejemplo representado en la figura 3.13. En \mathbb{R}^2 con la topología usual, consideremos la familia de bolas cerradas $\{B_n : n > 0\}$ de centro $(1/n, 0)$ y radio $1/2(n + 1)$. Se verifica: $\overline{\bigcup_{n \in \mathbb{N}} B_n} \neq \bigcup_{n \in \mathbb{N}} \overline{B_n}$, ya que $(0, 0) \in \overline{\bigcup_{n \in \mathbb{N}} B_n}$ pero $(0, 0) \notin \bigcup_{n \in \mathbb{N}} \overline{B_n}$.

Ejemplo. 3.5.3 Como ejemplo de espacio topológico DR-Rough, consideremos \mathbb{R}^2 y, como recubrimiento, la familia $\mathcal{F} = \{X_{ij}\}_{i,j \in \mathbb{Z}}$ de subconjuntos de \mathbb{R}^2 , definidos por $X_{ij} = (i, i + 1) \times (j, j + 1)$. El ejemplo descrito aparece ilustrado en la figura 3.14.

En espacios DR-rough podemos interpretar las relaciones I_i con aproximaciones del tipo rough. Para ello tomamos la familia $\mathcal{F} = \{X_i\}_{i \in \mathbb{N}}$ de conjuntos cerrados regulares no vacíos, tales que verifican las condiciones de recubrimiento DR-rough.

Teorema. 3.5.4 Sean $\underline{b} = \bigcup\{X_i : X_i \subseteq b\}$ y $\overline{b} = \bigcup\{X_i : X_i \cap b \neq \emptyset\}$. Se verifica entonces, que: $\sigma(b) = \overline{b}$ es una pulsación y $\xi(b) = \underline{b}$ es una contracción.

En el trabajo de T. Bittner y J.G. Stell [10], se estudia RCC5 como esquema de relaciones entre regiones y se muestra cómo puede generalizarse para aproximar regiones. También se describe la manera de extender ese estudio para tratar aproximaciones en las que se tienen en cuenta las fronteras de las regiones, concretamente, RCC8. Los autores, proponen un estilo concreto de

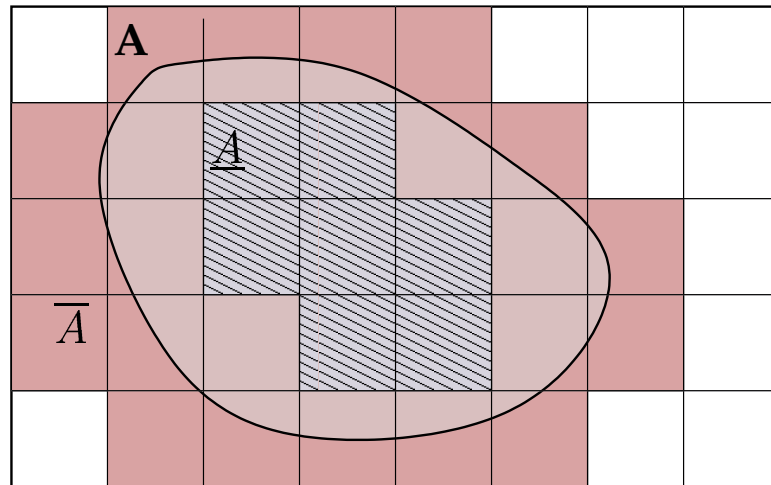


Figura 3.14: Diagrama en el que aparecen $A \subseteq \mathbb{R}^2$, \bar{A} , y \underline{A}

definir las relaciones de RCC. Como ejemplo, en el caso de RCC5, las relación entre dos regiones x, y viene determinada por una 3-upla de valores booleanos:

$$(x \wedge y \neq \perp, x \wedge y = x, x \wedge y = y)$$

Así, por ejemplo, la terna (F, F, F) se corresponde con la relación DR y (T, T, F) se corresponde con la relación PP . Además se establece un orden parcial entre las cinco ternas posibles (las cinco relaciones de RCC5), que es el orden inducido por el grafo conceptual de RCC5.

Capítulo 4

Extensiones con refinamiento

El método descrito en el capítulo anterior es generalizable a otros problemas de inserción de conceptos/relaciones en una ontología. En ese caso, no hemos necesitado refinar la definición de los elementos de los conceptos de la ontología, sin embargo, en ocasiones, es necesario hacerlo. Para ilustrar este hecho, vamos a dedicar este capítulo a estudiar la extensión de dicha teoría mediante la inserción de una nueva relación. La idea de las nuevas relaciones consiste en introducir nuevas definiciones de conexión que extiendan a la conexión externa, EC. La diferencia fundamental entre este ejemplo y el anterior consiste en que ahora la interpretación de las nuevas relaciones se hará mediante el refinamiento de las entidades espaciales, sustituyendo conjuntos regulares por “cells”.

4.1. Introduciendo una nueva relación de conexión

Esta sección está dedicada a la demostración de un teorema, el cual caracteriza todas las relaciones de conexión externa y que se obtienen por extensiones \mathcal{E}_{RCC} -conservativas de L_{RCC} .

Teorema. 4.1.1 *Existen 11 y sólo 11 relaciones de conexión que extienden la relación de conexión externa EC en extensiones \mathcal{E}_{RCC} -conservativas de L_{RCC} .*

Demostración: Hacemos uso de MACE4 para obtener los posibles modelos del nuevo retículo, en el que ahora la relación que denotaremos por el número 17 se especifica sólo por las relaciones:

<pre>and(14,17) = 17. % refina a C and(7,17) = 7. % extiende a EC</pre>	20
---	----

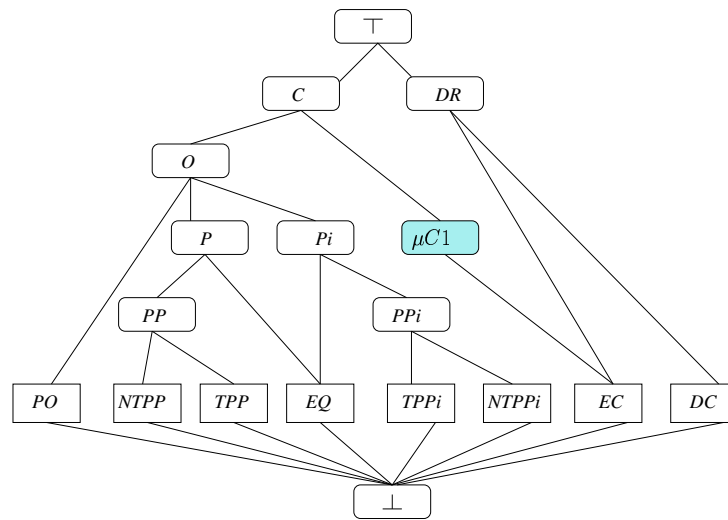


Figura 4.1: Modelo L'_1 del Retículo: $RCC + \mu C_1$

MACE4 nos devuelve 11 modelos, representados en las figuras correspondientes 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9, 4.10 y 4.11. En dichas figuras, utilizamos la notación μC_k , con $k \in \{1, 2, \dots, 11\}$ para referirnos a las nuevas relaciones de conexión.

Falta demostrar que existen únicamente esas 11 nuevas relaciones verificando las condiciones del teorema. Para ello facilitamos a OTTER un fichero de entrada que contenga:

- La definición de retículo, [6]
- Las relaciones \wedge, \vee del retículo, [8] y [7]
- Una nueva relación (a la que damos el nombre 17), verificando que está situada debajo de C y encima de EC, [20]
- Axioma de clausura de dominio, [17]
- Axioma de nombres únicos, [18]
- Las condiciones tipo $\text{and}(6,7) \neq 17$ que indiquen que 17 no es μC_1 , ni $\mu C_2, \dots$, ni μC_{10}

La idea de la prueba es la misma que en la prueba del teorema 3.1.1. Tomamos las tablas de Cayley del modelo correspondiente a μC_{11} y probamos que la nueva relación que hemos introducido verifica la última fila de cada tabla. Por tanto, obtendremos pruebas por reducción al absurdo de cada resultado. Las tablas en cuestión aparecen en [21].

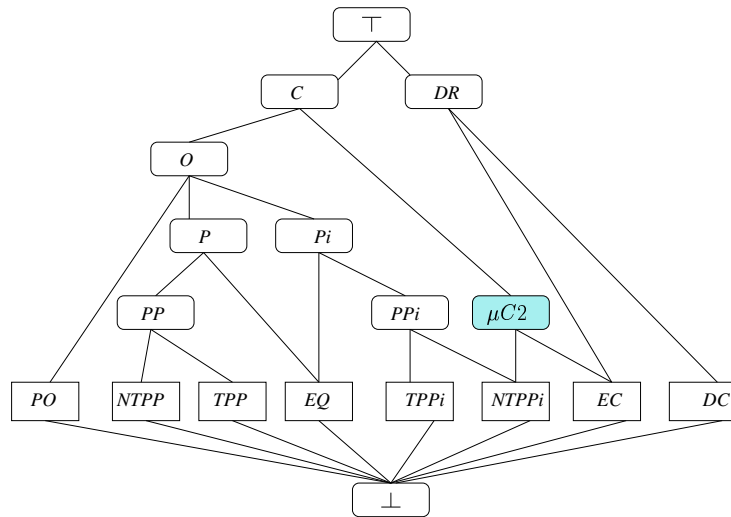


Figura 4.2: Modelo L'_2 del Retículo: $RCC + \mu C_2$

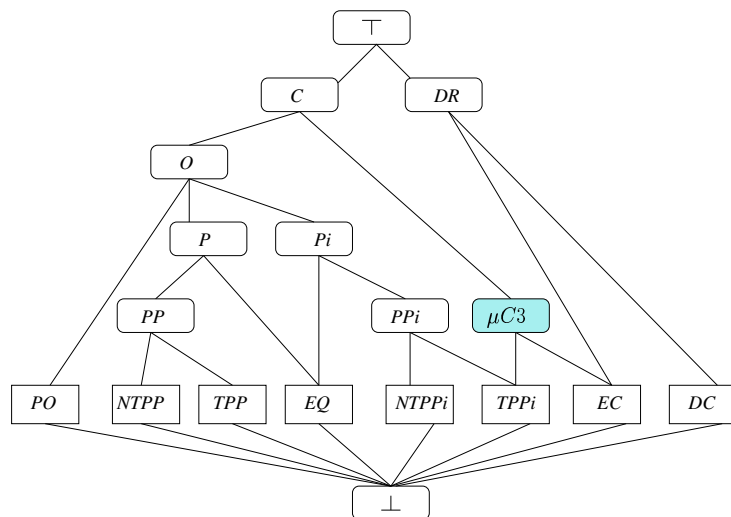


Figura 4.3: Modelo L'_3 del Retículo: $RCC + \mu C_3$

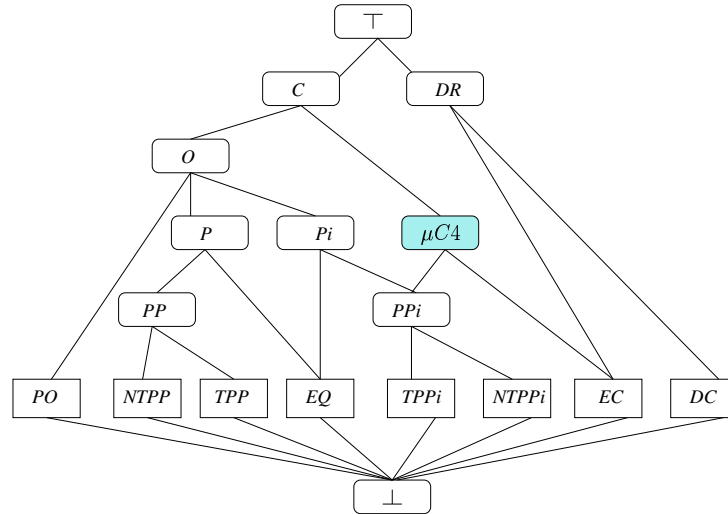


Figura 4.4: Modelo L'_4 del Retículo: $RCC + \mu C_4$

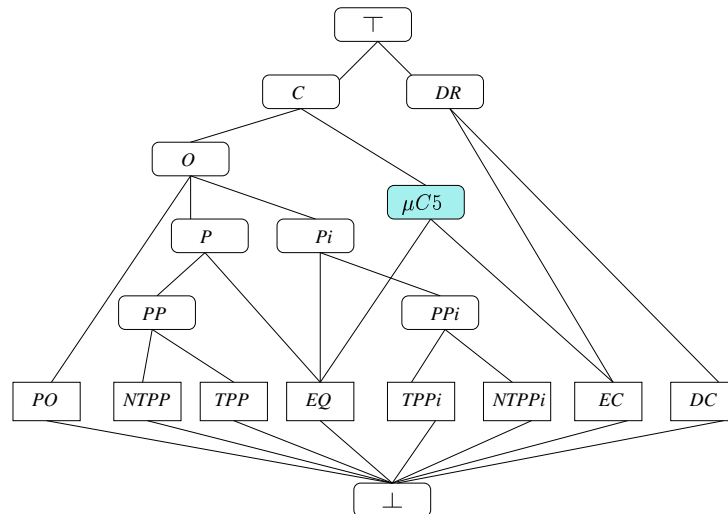


Figura 4.5: Modelo L'_5 del Retículo: $RCC + \mu C_5$

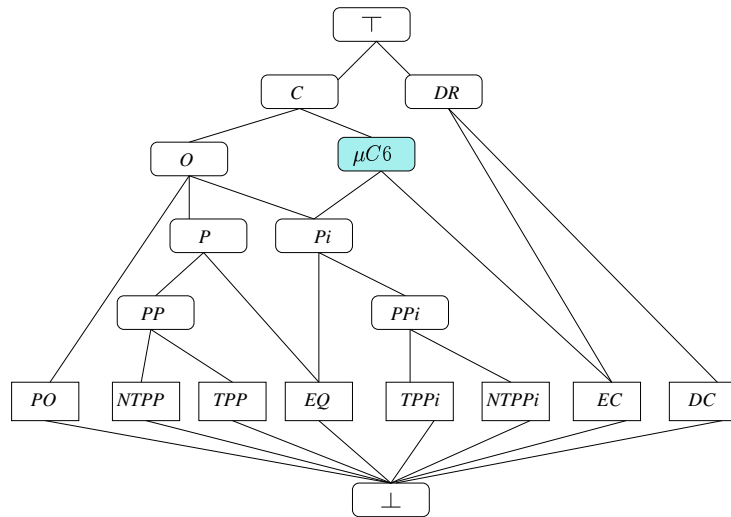


Figura 4.6: Modelo L'_6 del Retículo: $RCC + \mu C_6$

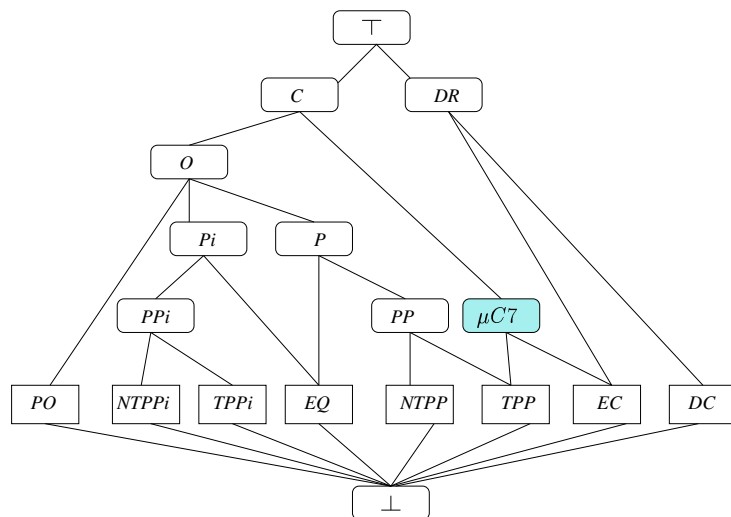


Figura 4.7: Modelo L'_7 del Retículo: $RCC + \mu C_7$

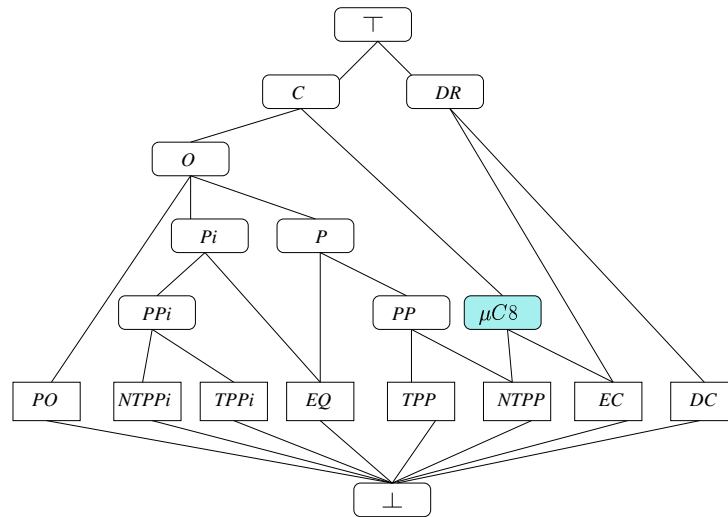


Figura 4.8: Modelo L'_8 del Retículo: $RCC + \mu C_8$

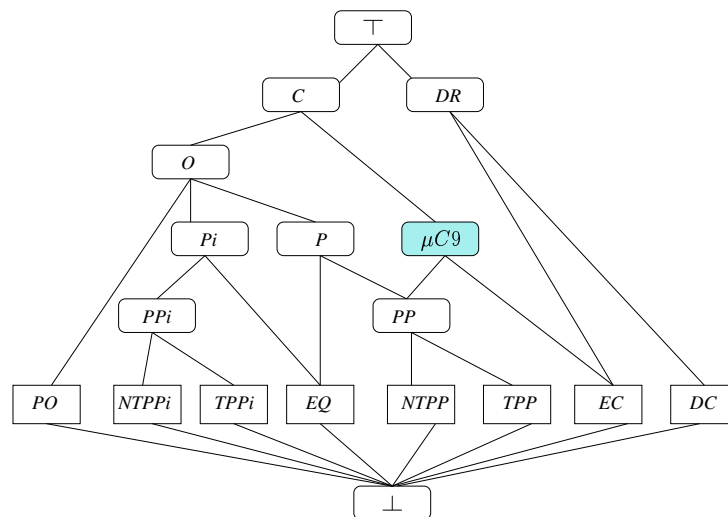


Figura 4.9: Modelo L'_9 del Retículo: $RCC + \mu C_9$

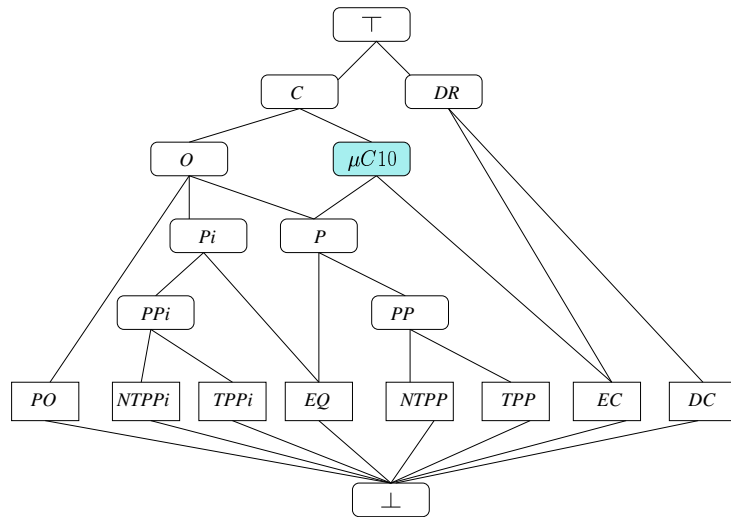


Figura 4.10: Modelo L'_{10} del Retículo: $RCC + \mu C_{10}$

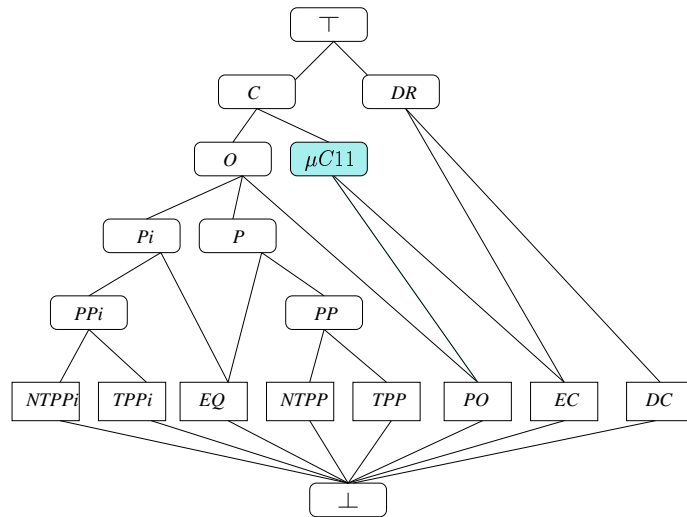


Figura 4.11: Modelo L'_{11} del Retículo: $RCC + \mu C_{11}$

```

or :
  | 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
  +-----+
0 | 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
1 | 1 1 1 3 1 3 1 3 1 3 1 7 1 6 1 3 1 3 1 3 1 3 1 4 1 6 1 6 1 7
2 | 2 1 3 2 9 1 1 1 3 1 3 1 4 1 6 9 1 3 1 1 1 3 1 3 1 4 1 6 1 6 1 4
3 | 3 1 3 9 3 1 1 1 3 1 3 1 4 1 6 9 1 3 1 1 1 3 1 3 1 4 1 6 1 6 1 4
4 | 4 1 3 1 1 1 1 4 1 2 1 2 1 4 1 6 1 1 1 2 1 1 1 2 1 3 1 4 1 6 1 6 1 4
5 | 5 1 3 1 3 1 3 1 2 5 1 0 1 4 1 6 1 3 1 0 1 3 1 2 1 3 1 4 1 6 1 6 1 4
6 | 6 1 3 1 3 1 3 1 2 1 0 6 1 4 1 6 1 3 1 0 1 3 1 2 1 3 1 4 1 6 1 6 1 4
7 | 7 1 7 1 4 1 4 1 4 1 4 1 4 7 1 5 1 4 1 4 1 4 1 4 1 4 1 4 1 5 1 6 1 7
8 | 8 1 6 1 6 1 6 1 6 1 6 1 6 1 5 8 1 6 1 6 1 6 1 6 1 6 1 6 1 5 1 6 1 6
9 | 9 1 3 9 9 1 1 1 3 1 3 1 4 1 6 9 1 3 1 1 1 3 1 3 1 4 1 6 1 6 1 4
10 | 10 1 3 1 3 1 3 1 2 1 0 1 0 1 4 1 6 1 3 1 0 1 3 1 2 1 3 1 4 1 6 1 6 1 4
11 | 1 1 1 3 1 1 1 1 1 1 1 3 1 3 1 4 1 6 1 1 1 3 1 1 1 3 1 3 1 4 1 6 1 6 1 4
12 | 1 2 1 3 1 3 1 3 1 2 1 2 1 2 1 4 1 6 1 3 1 2 1 3 1 2 1 3 1 4 1 6 1 6 1 4
13 | 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 4 1 6 1 3 1 3 1 3 1 3 1 3 1 4 1 6 1 6 1 4
14 | 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 6 1 4 1 4 1 4 1 4 1 4 1 4 1 6 1 6 1 4
15 | 1 5 1 6 1 6 1 6 1 6 1 6 1 6 1 5 1 5 1 6 1 6 1 6 1 6 1 6 1 5 1 6 1 6
16 | 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6 1 6
17 | 1 7 1 7 1 4 1 4 1 4 1 4 1 4 1 4 1 7 1 6 1 4 1 4 1 4 1 4 1 4 1 4 1 6 1 6 1 7

```

```

and :
  | 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
  +-----+
0 | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 | 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1
2 | 0 0 2 0 0 0 0 0 0 2 0 2 0 2 2 0 2 0
3 | 0 0 0 3 0 0 0 0 0 3 0 3 0 3 3 0 3 0
4 | 0 0 0 0 4 0 0 0 0 0 0 4 4 4 4 0 4 0
5 | 0 0 0 0 0 5 0 0 0 0 5 0 5 5 5 0 5 0
6 | 0 0 0 0 0 0 6 0 0 0 6 0 6 6 6 0 6 0
7 | 0 0 0 0 0 0 0 7 0 0 0 0 0 0 7 7 7 7
8 | 0 0 0 0 0 0 0 0 8 0 0 0 0 0 0 8 8 0
9 | 0 0 2 3 0 0 0 0 0 9 0 9 0 9 9 0 9 0
10 | 0 0 0 0 0 5 6 0 0 0 10 0 10 10 10 0 10 0
11 | 0 0 2 3 4 0 0 0 0 9 0 11 4 1 1 1 0 11 0
12 | 0 0 0 0 4 5 6 0 0 0 10 4 1 2 1 2 1 0 12 0
13 | 0 1 2 3 4 5 6 0 0 9 10 1 1 2 1 3 1 3 0 13 1
14 | 0 1 2 3 4 5 6 7 0 9 10 1 1 2 1 3 1 4 7 1 4 1 7
15 | 0 0 0 0 0 0 0 7 8 0 0 0 0 0 0 7 1 5 1 5 7
16 | 0 1 2 3 4 5 6 7 8 9 10 1 1 2 1 3 1 4 1 5 1 6 1 7
17 | 0 1 0 0 0 0 0 7 0 0 0 0 0 0 1 1 7 7 1 7 1 7

```

De igual manera que en la prueba mencionada, llegamos a contradicción. Por tanto, sólo existen los once modelos obtenidos por MACE4.

■

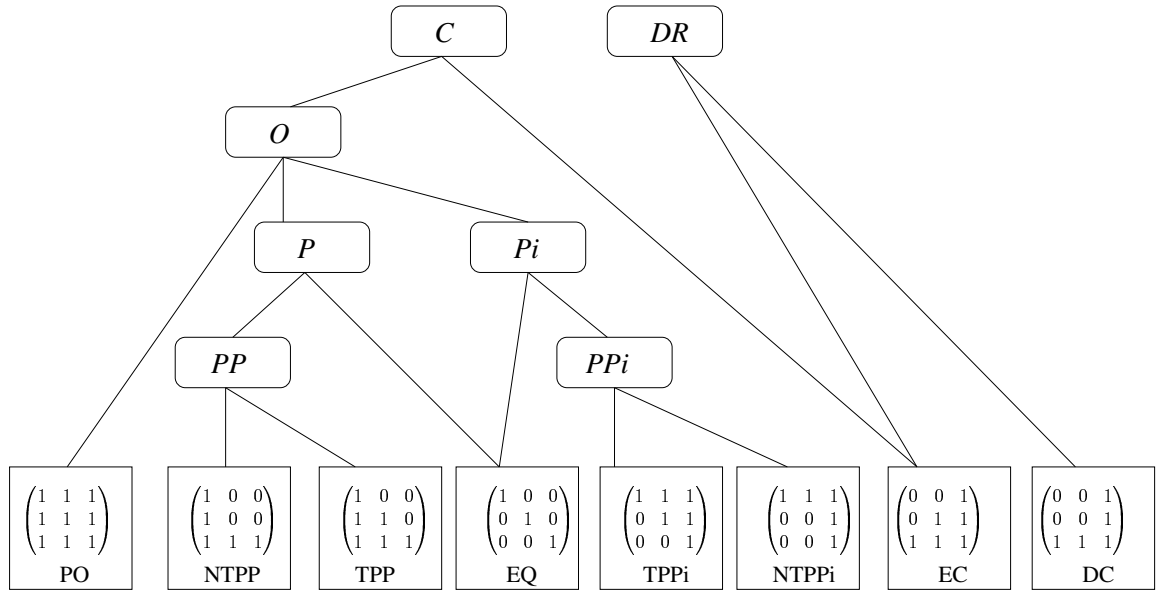


Figura 4.12: Interpretación de las relaciones de RCC8 con las matrices de Egenhofer

4.2. Interpretación de las nuevas relaciones

Como ya comentamos, la obtención de extensiones debe ser complementada con una caracterización de las nuevas relaciones de conexión obtenidas. Excepto para $\mu C1$ (que vuelve a ser una relación de gran indeterminación), interpretaremos las restantes utilizando los *cells* introducidos en este contexto por M. Egenhofer en [22] (véase 1.9). Egenhofer proporciona toda una serie de propiedades acerca del método de representación que son de interés para la caracterización de las nuevas relaciones del cálculo RCC que hemos introducido.

Caracterizamos las relaciones de RCC utilizando una matriz que recoge información sobre las posibles intersecciones entre los interiores, fronteras y exteriores de las dos regiones cuya relación caracteriza. El valor 0 indica intersección vacía, el 1, el caso contrario. Según la posición, la relación entre A y B viene descrita por la tabla siguiente:

\cap	B°	∂B	$e(B)$
A°	$A^\circ \cap B^\circ$	$A^\circ \cap \partial B$	$A^\circ \cap e(B)$
∂A	$\partial A \cap B^\circ$	$\partial A \cap \partial B$	$\partial A \cap e(B)$
$e(A)$	$e(A) \cap B^\circ$	$e(A) \cap \partial B$	$e(A) \cap e(B)$

Las caracterizaciones de las relaciones de RCC8 vienen dadas en la figura 4.12.

Las interpretaciones del resto de relaciones de RCC se expresan como disyunción de las anteriores. Teniendo en cuenta el diagrama de Hasse asociado a L_{RCC} , deducimos cuáles son las interpretaciones de las relaciones de RCC que están más arriba en el retículo \mathcal{R}_{RCC} .

$$PP: \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \vee \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \quad PPi: \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \vee \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

$$P: \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \vee \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \vee \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$Pi: \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \vee \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \vee \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$O: \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \vee \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \vee \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \vee \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \vee \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \vee \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

De igual forma se obtiene la caracterización de la conexión C , que será la disyunción de las matrices que determinan el solapamiento O junto con la que caracteriza a la conexión externa EC :

$$C: \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \vee \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \vee \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ \vee \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \vee \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \vee \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \vee \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Por último:

$$DR: \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \vee \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Las relaciones de conexión μC_k , con $k = 2 \dots 11$, se interpretan como sigue, basta ver las figuras 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7 y 3.8 para observar la disyunción que corresponde a cada una:

$$\mu C_2: \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \vee \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

$$\mu C_3: \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \vee \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

$$\begin{aligned} \mu C_4 &: \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \vee \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \vee \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \\ \mu C_5 &: \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \vee \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \\ \mu C_6 &: \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \vee \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \vee \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \vee \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \\ \mu C_7 &: \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \vee \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \\ \mu C_8 &: \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \vee \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \\ \mu C_9 &: \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \vee \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \vee \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \\ \mu C_{10} &: \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \vee \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \vee \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \vee \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \\ \mu C_{11} &: \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \vee \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \end{aligned}$$

Nótese que, realmente, hemos obtenido las formas normales disyuntivas de las relaciones con respecto a la intersección de interiores, fronteras, etc.

Teorema. 4.2.1 *Sea Ω un espacio topológico. Con la interpretación dada por las matrices de Egenhoffer, las once nuevas relaciones de conexión introducidas anteriormente, junto con C y EC componen el retículo de la figura 4.13, donde C es el \top y EC el \perp .*

Demostración: Para demostrar el teorema basta considerar las interpretaciones que acabamos de asignar a cada una de las relaciones de $RCC \cup \{\mu C_k : k = 2 \dots 11\}$ junto con μC_1 .

Queda, sin embargo, demostrar que la posición de μC_1 es la que aparece en la figura. Como no hemos obtenido una caracterización de μC_1 , necesitamos demostrar que su posición es exactamente la que aparece en la figura 4.13. Esa posición no es la única posible, en principio. Por ejemplo, podría ocurrir que μC_1 apareciera entre EC y una de las relaciones $\mu C_2, \mu C_3, \mu C_7, \mu C_8, \mu C_{11}$ y aún así seguiría siendo disjunta con $RCC \setminus \{EC, C\}$.

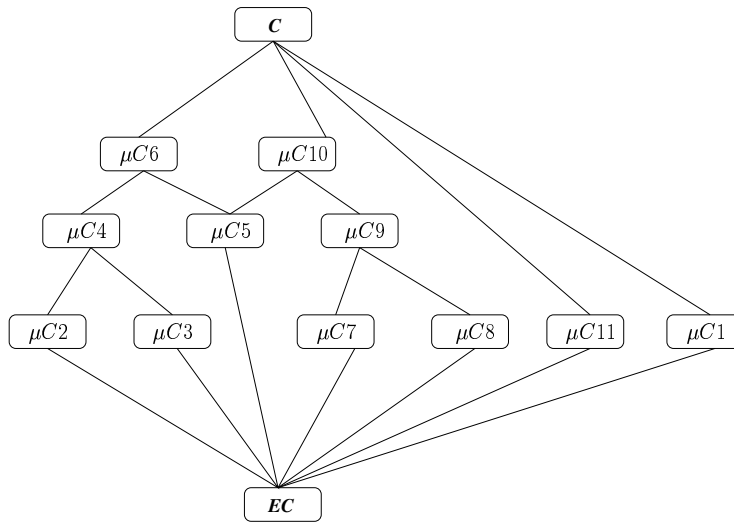


Figura 4.13: Posiciones relativas de las nuevas relaciones de conexión

Por tanto habrá que descartar esas posiciones. Para ello aplicamos MACE4 a un fichero con la definición de retículo [6], el retículo RCC [7] y [8], y las 11 relaciones que hemos obtenido. A μC_{11} únicamente le exigimos que sea una relación de conexión sin solapamiento (que extiende a EC). Añadimos como es habitual el Principio de nombres únicos [18], y la Clausura de Dominio [17].

MACE4 sólo obtiene un modelo de tamaño 28 (las 15 relaciones de RCC, \top , \perp y las 11 nuevas). Obtenemos un solo modelo y en él μC_1 es disjunta con RCC8 salvo que está bajo C y sobre EC y no está debajo de ninguna otra nueva relación de conexión, salvo C. Los ficheros correspondientes son donde-1.in y donde-numeros.in. El primero contiene la entrada con las relaciones según sus nombres en RCC. Luego MACE4 les asigna una numeración y calcula el modelo. Para tener la misma numeración que en los modelos anteriores, en el segundo fichero hemos numerado las relaciones de RCC como en los ejemplos que preceden a éste en el capítulo y hemos calculado el modelo en cuestión.

PO	NTPP	TPP	EQ	TPPi	NTPPi	EC	DC	PP	PPi	P	Pi	O	C	DR
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
\perp	\top	μC_1	μC_2	μC_3	μC_4	μC_5	μC_6	μC_7	μC_8	μC_9	μC_{10}	μC_{11}		
0	16	17	18	19	20	21	22	23	24	25	26	27		

El modelo obtenido viene dado por las tablas de Cayley que mostramos en la tabla 22:

Observaciones. 4.2.2 *En principio, en el retículo que presenta la figura 4.13, las relaciones μC_1 y μC_{11} son aparentemente iguales. Esto es debido al hecho de que las relaciones μC_k con $k = 2 \dots 11$ están conectadas con otras relaciones de RCC (distintas de C y EC), mientras que la relación μC_1 no verifica tal condición.*

Como consecuencia, tenemos el siguiente resultado.

Corolario. 4.2.3 *Con la interpretación de Egenhofer, el conjunto de las relaciones*

$$\{\mu C_1, \mu C_2, \mu C_3, \mu C_5, \mu C_7, \mu C_8, \mu C_{11}\}$$

forman un JEPD del retículo de la figura 4.13 (considerando $C \equiv \top$ y $EC \equiv \perp$).

4.3. Tabla de composición de un subcálculo

El método de interpretación con refinamiento sigue siendo un método más general que la extensión de las tablas de composición. Veamos cómo se puede calcular la tabla de composición a partir de la interpretación descrita para las extensiones en un caso concreto. En la extensión L'_{11} formada por el conjunto de relaciones $\mathcal{R}_{RCC} + \{\mu C_{11}\}$, si tomamos el conjunto de relaciones $RCC8$ y sustituimos las relaciones PO, EC por μC_{11} , obtenemos un subconjunto de dicha extensión con siete relaciones que constituyen un JEPD y que, como tal, tiene una tabla de composición asociada que construiremos a continuación. El interés que tiene dicha extensión, es que aumenta la granularidad del cálculo, es decir, obtenemos un cálculo en el que no se pueden distinguir el solapamiento parcial y la conexión externa de dos regiones. La interpretación de este cálculo es la de las relaciones de conexión entre sólidos, cálculo que ha sido estudiado en [32].

Teorema. 4.3.1 *En la extensión L'_{11} formada por el conjunto de relaciones $\mathcal{R}_{RCC} + \{\mu C_{11}\}$, el subconjunto $\{DC, \mu C_{11}, TPP, NTPP, TPPi, NTPPi, EQ\}$ constituye un JEPD cuya tabla de composición para la interpretación con refinamiento aparece en el cuadro 4.1.*

Demostración: En la tabla de 4.1 resumimos todos los cálculos. Como ejemplo, podemos ver cómo demostramos que la composición de μC_{11} y $NTPPi$ es el conjunto de relaciones $\{DC, \mu C_{11}, TPPi, NTPPi\}$. Con la interpretación antes descrita,

$$\begin{aligned} \mu C_{11}(a, b) \wedge NTPPi(a, b) &\iff (PO(a, b) \vee EC(a, b)) \wedge NTPPi(b, c) \iff \\ (PO(a, b) \wedge NTPPi(b, c)) \vee (EC(a, b) \wedge NTPPi(b, c)) &\iff \\ (DC(a, c) \vee \mu C_{11}(a, c) \vee TPPi(a, c) \vee NTPPi(a, c)) \vee DC(a, c) &\iff \\ DC(a, c) \vee \mu C_{11}(a, c) \vee TPPi(a, c) \vee NTPPi(a, c). & \end{aligned}$$

Cuadro 4.1: Tabla de composición para $RCC7$

$R2(b, c)$ $R1(a, b)$	DC	μC_{11}	TPP	$NTPP$	$TPPi$	$NTPPi$	EQ
DC	$RCC7$	$DC, \mu C_{11},$ $TPP,$ $NTPP$	$DC, \mu C_{11},$ $TPP,$ $NTPP$	$DC, \mu C_{11},$ $TPP,$ $NTPP$	DC	DC	DC
μC_{11}	$DC, \mu C_{11},$ $TPPi,$ $NTPPi$	$RCC7$	$TPP,$ $NTPP,$ μC_{11}	$TPP,$ $NTPP,$ μC_{11}	$DC, \mu C_{11}$ $TPPi,$ $NTPPi$	$DC, \mu C_{11}$ $TPPi,$ $NTPPi$	μC_{11}
TPP	DC	$DC, \mu C_{11},$ $TPP, NTPP$	$TPP,$ $NTPP$	$NTPP$	$DC, \mu C_{11},$ $TPP, TPPi,$ EQ	$DC, \mu C_{11},$ $TPPi,$ $NTPPi$	TPP
$NTPP$	DC	$DC, \mu C_{11},$ TPP $NTPP$	$NTPP$	$NTPP$	$DC, \mu C_{11},$ TPP $NTPP$	$RCC7$	$NTPP$
$TPPi$	$DC, \mu C_{11},$ $TPPi,$ $NTPPi$	$\mu C_{11},$ $TPPi,$ $NTPPi$	$\mu C_{11}, EQ,$ $TPP,$ $TPPi$	$\mu C_{11}, TPP,$ $NTPP$	$TPPi,$ $NTPPi$	$NTPPi$	$TPPi$
$NTPPi$	$DC, \mu C_{11},$ $TPPi,$ $NTPPi$	$\mu C_{11},$ $TPPi,$ $NTPPi$	$\mu C_{11},$ $TPPi,$ $NTPPi$	$\mu C_{11}, TPPi$ $TPP, NTPP$ $NTPPi, EQ$	$NTPPi$	$NTPPi$	$NTPPi$
EQ	DC	μC_{11}	TPP	$NTPP$	$TPPi$	$NTPPi$	EQ

De manera análoga se calcula el resto de las entradas de la tabla. █

Capítulo 5

Razonamiento metaontológico con RCC

En este capítulo interpretaremos RCC como una metaontología: una ontología sobre las relaciones entre los conceptos de una ontología. Con este objetivo utilizaremos tanto RCC como algunas de las extensiones \mathcal{E}_{RCC} -conservativas que hemos diseñado en los capítulos anteriores. El principio que subyace en las siguientes interpretaciones es que, en cualquier ontología robusta de pequeño tamaño, los conceptos deben de ser susceptibles de ser representados por regiones espaciales de manera natural y atendiendo a su relación lógica. Adicionalmente, RCC puede servir para justificar la adecuación de ciertas transformaciones en dichas regiones (es decir, que ciertas propiedades esenciales de la ontología se conservan).

5.1. Interpretación fuerte

En la que denominaremos como *interpretación fuerte*, la relación $\mathcal{C}(x, y)$ se entenderá intuitivamente como: *existen elementos comunes a los dos conceptos x, y , pero no podemos afirmar que comparten un subconcepto común*. A partir de esta definición se pueden interpretar todas las relaciones de RCC, y por tanto, el cálculo mediante tablas de composición se puede traducir en *inferencia ontológica*. Además, como trabajaremos con la representación espacial del concepto entendida como la extensión de éste, a partir de ahora utilizaremos sólo conceptos tales que $\Sigma \models C \not\equiv \perp$.

Definición. 5.1.1 (Interpretación fuerte del lenguaje RCC como metaontología) *Dados C_1, C_2 , dos conceptos de la base de conocimiento Σ , diremos que están Σ -conectados, y lo notaremos $\mathcal{C}_\Sigma(C_1, C_2)$, si*

$$\Sigma \not\models C_1 \sqcap C_2 \equiv \perp$$

Para el análisis de este tipo de conexión, haremos uso de tipos. Recordemos que, dada Σ una base de conocimiento consistente, un n -tipo (o simplemente tipo, si $n = 1$) de Σ es un conjunto (maximal) de fórmulas $\{\varphi_k(x_1, \dots, x_n)\}_k$ que es finitamente satisfactible, es decir, tal que para todo $F \subseteq \mathbb{N}$ finito, la teoría

$$\Sigma + \{\exists x_1 \cdots \exists x_n \bigwedge_{k \in F} \varphi_k(x_1, \dots, x_n)\}$$

es consistente. Diremos que Σ realiza el tipo si dicha propiedad es cierta para $F = \mathbb{N}$, es decir, si dadas a_1, \dots, a_n nuevas constantes, la teoría

$$\Sigma + \{\varphi_k(a_1, \dots, a_n) : k \in \mathbb{N}\}$$

es consistente. Nótese que este hecho quiere decir que existe un modelo de Σ donde todas las fórmulas del tipo son satisfactibles para una n -tupla fija. Nos limitaremos a trabajar con 1-tipos, y para una correcta definición hay que considerar las fórmulas expresadas en LPO, y por tanto los tipos contienen fórmulas no traducibles a DL. Esta consideración no es un problema, pues realmente no se necesitan fórmulas auxiliares que no sean traducibles a DL en la demostración del teorema 5.1.2, siempre y cuando los constructores *negación* y *intersección* sean admisibles por la lógica descriptiva utilizada.

El espacio de los tipos de Σ lo notaremos por

$$S(\Sigma) = \{p : p \text{ es tipo (maximal) de } \Sigma\}$$

Dado $C \in \text{conceptos}(\Sigma)$, $[C] = S_\Sigma(C) = \{p \in S(\Sigma) : C \in p\}$. Dada una constante a y un modelo $I \models \Sigma$, el tipo de a en I es $[a] = tp_I(a) = \{C(x) : I \models C(a)\}$. Dos siguientes resultados básicos muy útiles son que $[C] \cap [D] = [C \wedge D]$ y que $[\neg C] = S(\Sigma) \setminus [C]$.

Teorema. 5.1.2 *La interpretación fuerte no discrimina RCC8. Más concretamente:*

1. $C_\Sigma(C_1, C_2) \iff S_\Sigma(C_1) \cap S_\Sigma(C_2) \neq \emptyset$
2. $DC_\Sigma(C_1, C_2) \iff S_\Sigma(C_1) \cap S_\Sigma(C_2) = \emptyset$
3. $P_\Sigma(C_1, C_2) \iff S_\Sigma(C_1) \subseteq S_\Sigma(C_2)$
4. $PP_\Sigma(C_1, C_2) \iff S_\Sigma(C_1) \subsetneq S_\Sigma(C_2)$
5. $EQ_\Sigma(C_1, C_2) \iff S_\Sigma(C_1) = S_\Sigma(C_2)$
6. $O_\Sigma(C_1, C_2) \iff C_\Sigma(C_1, C_2)$
7. $PO_\Sigma(C_1, C_2) \iff S_\Sigma(C_1) \cap S_\Sigma(C_2) \neq \emptyset \wedge S_\Sigma(C_1) \not\subseteq S_\Sigma(C_2) \wedge S_\Sigma(C_2) \not\subseteq S_\Sigma(C_1)$

8. $Pi_{\Sigma}(C_1, C_2) \iff S_{\Sigma}(C_2) \subseteq S_{\Sigma}(C_1)$
9. $PPi_{\Sigma}(C_1, C_2) \iff S_{\Sigma}(C_2) \subsetneq S_{\Sigma}(C_1)$
10. $DR_{\Sigma}(C_1, C_2) \iff DC_{\Sigma}(C_1, C_2)$
11. Para cualesquiera C_1, C_2 :
 - $\neg EC_{\Sigma}(C_1, C_2)$
 - $\neg TPP_{\Sigma}(C_1, C_2)$
 - $\neg NTPP_{\Sigma}(C_1, C_2)$
 - $\neg TPPi_{\Sigma}(C_1, C_2)$
 - $\neg NTPPi_{\Sigma}(C_1, C_2)$

Demostración: Pasemos a demostrar cada caso:

1. Por definición, $C_{\Sigma}(C_1, C_2)$ si y sólo si $\Sigma \not\models C_1 \sqcap C_2 \equiv \perp$. Sea I un modelo tal que $C_1^I \cap C_2^I \neq \emptyset$. Interpretando a , una nueva constante, como un elemento de esa intersección, demostramos que la teoría $\Sigma \cup \{(C_1 \sqcap C_2)(a)\}$ es consistente. De hecho,

$$[a] \in [C_1 \sqcap C_2]$$

por tanto, $S_{\Sigma}(C_1 \sqcap C_2) \neq \emptyset$. De donde se sigue que $S_{\Sigma}(C_1) \cap S_{\Sigma}(C_2) \neq \emptyset$.

2. $DC_{\Sigma}(C_1, C_2) \iff \neg C_{\Sigma}(C_1, C_2) \iff S_{\Sigma}(C_1) \cap S_{\Sigma}(C_2) = \emptyset$
3. Por la definición en RCC de P ,

$$P_{\Sigma}(C_1, C_2) \iff \forall C_3 (S_{\Sigma}(C_3) \cap S_{\Sigma}(C_1) \neq \emptyset \rightarrow S_{\Sigma}(C_3) \cap S_{\Sigma}(C_2) \neq \emptyset)$$

Veamos que esta última implicación se verifica si y solo si $S_{\Sigma}(C_1) \subseteq S_{\Sigma}(C_2)$:

Probar que es condición suficiente es trivial. Veamos (por reducción al absurdo) que es necesaria. Sea $p \in S_{\Sigma}(C_1)$; es decir, $C_1 \in p$ y supongamos que $p \notin S_{\Sigma}(C_2)$. Entonces $C_2 \notin p$ y, por ser p maximal, $\neg C_2 \in p$. Así, $C_1 \in p$ y $\neg C_2 \in p$, de donde $p \in S_{\Sigma}(C_1) \cap S_{\Sigma}(C_1 \sqcap \neg C_2)$; luego

$$S_{\Sigma}(C_1) \cap S_{\Sigma}(C_1 \sqcap \neg C_2) \neq \emptyset$$

Entonces debe ocurrir que

$$S_{\Sigma}(C_2) \cap S_{\Sigma}(C_1 \sqcap \neg C_2) \neq \emptyset$$

Lo que implicaría que $p' \in C_1 \sqcap \neg C_2$ y $p' \in C_2$, lo cual no es posible, pues $\{C_1 \sqcap \neg C_2, C_2\}$ no puede estar contenido en un tipo (es un conjunto inconsistente).

$$4. \quad \begin{array}{l} PP_{\Sigma}(C_1, C_2) \xleftrightarrow{\text{def.}} P_{\Sigma}(C_1, C_2) \wedge \neg P_{\Sigma}(C_2, C_1) \xleftrightarrow{(3)} \\ S_{\Sigma}(C_1) \subseteq S_{\Sigma}(C_2) \wedge \neg(S_{\Sigma}(C_2) \subseteq S_{\Sigma}(C_1)) \iff S_{\Sigma}(C_1) \subsetneq S_{\Sigma}(C_2) \end{array}$$

5. Se tiene que

$$\begin{array}{l} EQ_{\Sigma}(C_1, C_2) \iff P_{\Sigma}(C_1, C_2) \wedge P_{\Sigma}(C_2, C_1) \iff \\ S_{\Sigma}(C_1) \subseteq S_{\Sigma}(C_2) \wedge (S_{\Sigma}(C_2) \subseteq S_{\Sigma}(C_1)) \iff S_{\Sigma}(C_1) = S_{\Sigma}(C_2) \end{array}$$

Tenemos que demostrar que esta última condición se verifica si y solo si $\Sigma \models C_1 \equiv C_2$.

Es trivial ver que la condición es suficiente. Veamos que es necesaria.

Sea $I \models \Sigma$ y a cualquier realización de C_1 en I , es decir, $I \models C_1(a)$. Entonces

$$tp_I(a) = \{\phi : I \models \phi(a)\} = [a] \in S_{\Sigma}(C_1)$$

Así, $[a] \in S_{\Sigma}(C_2)$. Por tanto, $I \models \forall x(C_1(x) \rightarrow C_2(x))$; es decir, $I \models C_1 \sqsubseteq C_2$.

De igual forma $I \models C_2 \sqsubseteq C_1$. Por tanto, $\Sigma \models C_1 \equiv C_2$.

$$6. \quad \begin{array}{l} O_{\Sigma}(C_1, C_2) \iff \exists C_3(P_{\Sigma}(C_3, C_1) \wedge P_{\Sigma}(C_3, C_2)) \iff \\ \exists C_3(S_{\Sigma}(C_3) \subseteq S_{\Sigma}(C_1) \wedge S_{\Sigma}(C_3) \subseteq S_{\Sigma}(C_2)) \iff \\ \exists C_3(S_{\Sigma}(C_3) \subseteq S_{\Sigma}(C_1) \cap S_{\Sigma}(C_2)) \end{array}$$

Por tanto, $S_{\Sigma}(C_1) \cap S_{\Sigma}(C_2) \neq \emptyset$, es decir, $C_{\Sigma}(C_1, C_2)$. La implicación contraria se debe al hecho de que $S_{\Sigma}(C_1 \sqcap C_2) \subseteq S_{\Sigma}(C_1) \cap S_{\Sigma}(C_2)$ ¹.

$$7. \quad \text{Este caso se deduce de que } PO_{\Sigma}(C_1, C_2) \iff C_{\Sigma}(C_1, C_2) \wedge \neg P_{\Sigma}(C_1, C_2) \wedge \neg P_{\Sigma}(C_2, C_1)$$

El resto de los casos se deben a que, como hemos visto, Σ -conexión equivale a Σ -solapamiento.

A modo de ejemplo, veamos que $\neg EC_{\Sigma}(C_1, C_2)$:

$$\begin{array}{l} \neg EC_{\Sigma}(C_1, C_2) \iff \neg[C_{\Sigma}(C_1, C_2) \wedge \neg O_{\Sigma}(C_1, C_2)] \iff \\ \neg C_{\Sigma}(C_1, C_2) \vee O_{\Sigma}(C_1, C_2) \iff \neg C_{\Sigma}(C_1, C_2) \vee C_{\Sigma}(C_1, C_2), \text{ que es una tautología. } \blacksquare \end{array}$$

Sin embargo, la interpretación anterior puede no ser usable en la práctica cuando tratamos con lógicas descriptivas expresivas: el concepto de conexión entre conceptos se basa en el problema de implicación lógica, que puede ser de alta complejidad algorítmica.

Las *implicaciones lógicas* que se tratan en DL son, fundamentalmente, axiomas de inclusión del tipo $C_1 \sqsubseteq C_2$. En el caso que nos ocupa, la interpretación fuerte de la conexión, tenemos

$$C_{\Sigma}(C_1, C_2) \iff \Sigma \not\models C_1 \sqcap C_2 \equiv \perp$$

¹De hecho, si consideramos la conceptualización como un retículo de conceptos, la interpretación queda como $O_{\Sigma}(C_1, C_2) \iff S_{\Sigma}(C_1 \sqcap C_2) \neq \emptyset$.

o de manera equivalente

$$C_{\Sigma}(C_1, C_2) \iff \Sigma \not\models C_1 \sqsubseteq \neg C_2$$

Por tanto, decidir si dos conceptos no están Σ -conectados será equivalente a comprobar que para cada modelo \mathcal{I} de Σ se tiene $C_1^{\mathcal{I}} \subseteq \neg C_2^{\mathcal{I}}$. En [13], los autores demuestran que la consistencia de conceptos y, por tanto, la satisficibilidad y la implicación lógica es EXPTIME-hard para la lógica \mathcal{ALC} . Es, por tanto, más conveniente utilizar la interpretación ontológica siguiente, a la que llamaremos *interpretación débil*. Es más, tal interpretación volverá a ser debilitada (en el *ciclo de limpieza*, secc. 7.2) si la ontología que estamos representando es considerada como defectuosa, sujeta a cambios, y por tanto no es importante hacer una caracterización espacial tan detallada como la que ofrece la siguiente interpretación.

5.2. Interpretación débil

Ya hemos comentado que la interpretación fuerte puede ser desaconsejable en la práctica debido a su complejidad computacional. Otra limitación de la interpretación fuerte estriba en que reduce nuestra capacidad mereotológica a RCC5 (según 5.1.2). En esta sección introducimos otro tipo de interpretación que llamaremos *débil*. La interpretación débil consiste en asociar a cada base de conocimiento un PSR de tal forma que sus modelos topológicos representen un modelo de la base de conocimiento original.

La interpretación psicológica de la representación espacial obtenida mediante la resolución del PSR se completa identificando la frontera espacial de una región con los elementos susceptibles de cambiar de propiedades conceptuales. Esta idea será utilizada en el capítulo 7².

La construcción del grafo \mathcal{G} asociado al PSR se realiza según el siguiente método.

En primer lugar inicializamos el grafo: $\mathcal{G}_0 = \{C : C \in \text{conceptos}(\Sigma)\} \cup \{\neg C : C \in \text{conceptos}(\Sigma)\}$. Recordemos que los conceptos atómicos son los que no aparecen en el lado izquierdo de ningún axioma de conceptos definidos ni de conceptos primitivos. Como paso preliminar, cada axioma $C \equiv D$ lo sustituyo en Σ por $\{C \sqsubseteq D, \neg C \sqsubseteq \neg D\}$. A continuación añadimos nodos y arcos al grafo atendiendo a los siguientes casos:

²Aunque esta identificación de la frontera con los elementos *inseguros* del concepto induce a pensar en utilizar mejor regiones vagas, consideramos que el número de elementos de este tipo será relativamente pequeño, y no determinable por ningún tipo de región, es decir, no deseamos que el conjunto de tales elementos representen una región espacial, pues éstas se usan exclusivamente para los conceptos.



Figura 5.1: Subgrafo asociado al axioma $C \sqsubseteq D$, con D atómico

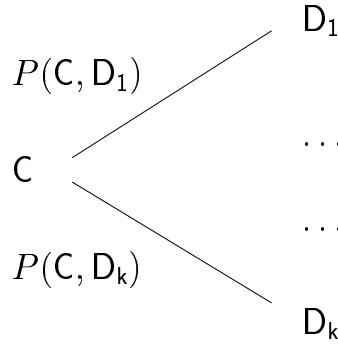
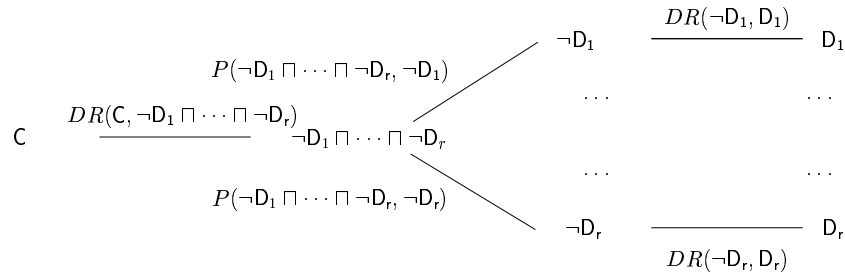


Figura 5.2: Grafo asociado a $C \sqsubseteq D_1 \sqcap \dots \sqcap D_k$

- Para cada axioma $C \sqsubseteq D$, con D atómico, añadimos a \mathcal{G}_0 la arista $P(C, D)$, uniendo los nodos C y D . El grafo aparece en la figura 5.1.
- Para cada axioma $C \sqsubseteq D$, con D no atómico, consideramos las siguientes posibilidades:
 - $C \sqsubseteq D_1 \sqcap D_2$, con D_1, D_2 atómicos. Añadimos dos aristas al grafo: $P(C, D_1)$ que une C y D_1 , y $P(C, D_2)$ uniendo C y D_2 . Este caso se puede generalizar para $C \sqsubseteq D_1 \sqcap \dots \sqcap D_k$, añadiéndose las k aristas correspondientes. Ver figura 5.2.
 - $C \sqsubseteq D_1 \sqcup D_2$, con D_1, D_2 atómicos. Se añaden tres nuevos nodos al grafo, $\neg D_1 \sqcap \neg D_2$, $\neg D_1$ y $\neg D_2$ y las siguientes aristas: $DR(C, \neg D_1 \sqcap \neg D_2)$, $P(\neg D_1 \sqcap \neg D_2, \neg D_i)$ y $DR(\neg D_i, D_i)$, con $i = 1, 2$. De manera análoga podemos generalizar el caso para $C \sqsubseteq D_1 \sqcup \dots \sqcup D_r$, añadiendo los $r + 1$ nodos y las $2r + 1$ aristas resultantes. El grafo puede verse en la figura 5.3.
 - Por último, si $C \sqsubseteq D$, y D no es uno de los anteriores, entonces obtenemos la forma normal conjuntiva de D (ver 1.2.4). Tendremos pues:

$$C \sqsubseteq \bigsqcap_i^r D_{i1} \sqcap \bigsqcap_i^r D_{i2} \sqcap \dots \sqcap \bigsqcap_i^r D_{ik}$$

En este caso, el grafo será el resultante de unir los grafos de cada una de las fórmulas $C \sqsubseteq \bigsqcap_i^r D_{i1}, \dots, C \sqsubseteq \bigsqcap_i^r D_{ik}$ que se obtienen aplicando el caso anterior.

Figura 5.3: Subgrafo asociado a $C \sqsubseteq D_1 \sqcup \dots \sqcup D_r$

Definición. 5.2.1 Un *modelo espacial* de Σ es una interpretación en el lenguaje de Σ , tal que $I : \text{conceptos}(\Sigma) \cup \text{indiv}(\Sigma) \rightarrow \Omega$, donde Ω es un espacio topológico T_3 conexo tal que $I \models \Sigma$ y para cada $C \in \text{conceptos}(\Sigma)$, $I(C)$ es un conjunto abierto regular en Ω y para cada $a \in \text{indiv}(\Sigma)$, $I(a)$ es un punto.

Proposición. 5.2.2 Sean $\Sigma = \langle \mathcal{T}, \mathcal{A} \rangle$ una base de conocimiento finita, I un modelo espacial y a_1, \dots, a_k los nombres de individuos que aparecen en \mathcal{A} . Para cada a_i existe un entorno E_i en Ω tal que:

1. $I(a_i) \in E_i$
2. $\forall j \neq i : I(a_j) \notin E_i$
3. Para todo C concepto de Σ :

$$C(a_i) \iff E_i \subseteq I(C)$$

Demostración: Sean Σ , I , Ω y $a_i \in \text{indiv}(\Sigma)$ como en el enunciado. Para cada concepto C , denotamos por B_C el abierto regular $I(C)$. Tomamos el entorno

$\bigcap_{\{C: C(a_i) \in \mathcal{A}\}} B_C$ (que es abierto por ser intersección finita de abiertos). Al estar en

un espacio topológico T_3 , podemos separar un punto y un cerrado por abiertos, es decir, para cada a_j , con $j \neq i$, existe un abierto \mathcal{O}_j y tal que $a_j \notin \mathcal{O}_j$ y \mathcal{O}_j contiene al cerrado

$\bigcap_{\{C: C(a_i) \in \mathcal{A}\}} B_C$. El conjunto $\bigcap_{j \neq i} \mathcal{O}_j$ es abierto por ser intersección de abiertos. Así, el entorno de a_i :

$$E_i = \bigcap_{\{C: C(a_i) \in \mathcal{A}\}} B_C \cap \bigcap_{j \neq i} \mathcal{O}_j$$

por construcción, verifica las condiciones del enunciado. ■

Corolario. 5.2.3 Sea $PSR'(\Sigma)$ el PSR asociado a Σ que se obtiene como resultado de aplicar la proposición anterior (es decir, tras cambiar cada símbolo a_i por el entorno E_i correspondiente y cada símbolo \in por el de \subseteq). Entonces, se verifica: $PSR(\Sigma)$ es espacialmente consistente $\iff PSR(\Sigma)'$ lo es.

Demostración: La implicación a la derecha es trivial por la proposición. Para el recíproco, basta escoger un elemento de E_i como interpretación de a_i . ■

El siguiente teorema garantiza que la interpretación débil es útil para analizar bases de conocimiento en DL, pues respeta la consistencia.

Teorema. 5.2.4 El PSR asociado a Σ es espacialmente consistente si y sólo si Σ es consistente.

Demostración:

- $PSR(\Sigma)$ consistente $\implies \Sigma$ consistente :

Supongamos que $PSR(\Sigma)$ es espacialmente consistente. Entonces el problema $PSR'(\Sigma)$ asociado construido en 5.2.2 es consistente por el corolario anterior. Si es consistente, por un resultado de J. Renz [59], tal problema posee una solución que es un modelo espacial I en \mathbb{R}^2 (que incluso interpreta las constantes como regiones poligonales). Nótese que en este modelo las interpretaciones de los conceptos son regiones regulares (no necesariamente abiertas).

Una primera transformación de I consiste en elegir para cada constante una interpretación de ésta que pertenezca al interior de la región $I(a)$ original. Denotemos esa interpretación como $J(a)$.

Para construir el modelo de la base de conocimiento inicial, basta transformar las regiones que interpretan cada concepto de tal forma que las que están DR -relacionadas pasen a ser disjuntas. Procedemos de la siguiente forma: Para cada D_i, D_j nodos tales que exista una arista etiquetada por $DR(D_i, D_j)$ en el I , sea $F_{i,j} = D_i \cap D_j$. Nótese que $F_{i,j}$ es un subconjunto de la frontera de tales regiones.

Como $F_{i,j}$ tiene interior vacío, no contiene a la interpretación de ninguna constante. Por tanto, si interpretamos los conceptos como:

$$J(C_i) := I(C_i) \setminus \bigcup_{\{C_j : I \models DR(C_i, C_j)\}} F_{i,j}$$

La interpretación J , como DL-interpretación, valida los axiomas de Σ : veamos uno de los ejemplos.

Si $C \sqsubseteq D_1 \sqcup D_2$ es un axioma de Σ , entonces en la interpretación J , considerada como modelo topológico, se tiene que $DR(C, \neg D_1 \sqcap \neg D_2)$. Como la intersección de dichas regiones está contenida $F_{1,2}$, entonces $J \models C \sqcap (\neg D_1 \sqcap \neg D_2) = \perp$, y por tanto $J \models C \sqsubseteq D_1 \sqcup D_2$.

- Σ consistente $\implies PSR(\Sigma)$ consistente:

Si Σ es consistente, entonces $PSR(\Sigma)$ es trivialmente consistente (el modelo de Σ es una solución, por la interpretación fuerte³. Por el corolario 5.2.3 se tiene que $PSR'(\Sigma)$ es consistente. Aplicando otra vez el resultado de Renz [59], $PSR'(\Sigma)$ es espacialmente consistente. Del corolario 5.2.3 se deduce que $PSR(\Sigma)$ es consistente.

■

5.3. Relación entre las interpretaciones fuerte y débil

Sea Σ una base de conocimiento. La relación entre ambas interpretaciones es directa:

Teorema. 5.3.1

$$\mathcal{C}_{\Sigma}(C_1, C_2) \iff \exists I : I \models \mathcal{C}(C_1, C_2)$$

Demostración: Se tiene que $\mathcal{C}_{\Sigma}(C_1, C_2)$ si y sólo si $\Sigma \not\models C_1 \sqcap C_2 \equiv \perp$. Supongamos que $\mathcal{C}_{\Sigma}(C_1, C_2)$. Haciendo uso del teorema 5.2.4, existe un modelo espacial I donde $C_1^I \sqcap C_2^I \neq \emptyset$. Tomando a una nueva constante, la teoría $\Sigma \cup \{(C_1 \sqcap C_2)(a)\}$ es consistente. Luego existe un modelo de $\Sigma \cup \{C_1(a), C_2(a)\}$. Aplicando la proposición 5.2.2, debe existir un entorno E_a tal que $a \in E_a \subseteq I(C_1) \cap I(C_2)$, de donde se concluye que $\overline{I(C_1)} \cap \overline{I(C_2)} \neq \emptyset$.

El recíproco es trivial

■

5.4. Interpretaciones vagas de los conceptos con respecto a una ontología

Definición. 5.4.1 Sea Σ una base de conocimiento (en DL). Sea σ la aplicación que a cada concepto $C \in \text{conceptos}(\Sigma)$ le asocia una fórmula de DL definida como sigue:

³Aunque ese modelo satisface los arcos del tipo DR a través de DC , por 5.1.2.(10).

$$\sigma(C) = \begin{cases} C, & \text{si } C \text{ es un concepto definido} \\ \sqcap\{D : C \sqsubseteq D \in \Sigma\}, & \text{si } C \text{ es un concepto primitivo} \\ \top, & \text{si } C \text{ es un concepto atómico} \end{cases}$$

Dados $C_1, C_2 \in \text{conceptos}(\Sigma)$, diremos que están Σ -conectados bajo σ , y lo notaremos $\mathcal{C}_\Sigma^\sigma(C_1, C_2)$, si $\mathcal{C}_\Sigma(\sigma(C_1), \sigma(C_2))$.

A $\sigma(C_1)$ lo llamaremos la **noción** asociada a C en Σ .

La noción está definida para todo concepto, y el análisis que realizamos de las relaciones entre nociones es válido para cualesquiera de ellos. Sin embargo, en la práctica no se usa de manera intensiva esta definición para conceptos atómicos (en el análisis de anomalías). La razón es que la total indefinición de la noción asociada a un concepto atómico puede ser intencionada: son los conceptos primitivos de la ontología (en muchos casos, conceptos abstractos) y en muchos casos no se debería forzar al usuario a que refine su definición.

Observaciones. 5.4.2 Nótese que esta nueva conexión está relacionada con la anterior:

- Dados dos conceptos C_1, C_2 , se verifica:

$$\mathcal{C}_\Sigma^\sigma(C_1, C_2) \iff \mathcal{C}_\Sigma(\sigma(C_1), \sigma(C_2)) \iff \Sigma \not\models \sigma(C_1) \sqcap \sigma(C_2) \equiv \perp$$

- La idea de la Σ -conexión bajo σ es obviamente la de pulsación. Ahora, $\sigma(C)$ representa una fórmula en DL asociada a un concepto C .

Proposición. 5.4.3 Sea Σ un base de conocimiento (en DL) y σ como en la definición anterior. Dados C_1, C_2 , dos conceptos de la base de conocimiento Σ , para cualquier $R \in RCC8$, se verifica:

$$R_\Sigma^\sigma(C_1, C_2) \iff R_\Sigma(\sigma(C_1), \sigma(C_2))$$

Demostración: La demostración se reduce a utilizar el teorema 5.1.2. ■

Por tanto, el análisis espacial es similar para conceptos y nociones, una vez realizada la interpretación espacial correspondiente. Este hecho facilitará el análisis de las relaciones entre los conceptos de una ontología que realizamos en el segundo bloque de esta memoria.

Capítulo 6

Anomalías en la conceptualización asociada a una ontología

Antes de utilizar cualquier aplicación (de análisis o manipulación) sobre una base de datos, es necesario corregir o, en su caso, eliminar los posibles datos ilícitos. Tenemos que disponer de una clasificación de los tipos de datos ilícitos para proceder de una forma u otra a su corrección, o bien de una métrica para evaluar la calidad de los mismos. Sin ellas no es posible medir el grado de confianza en los resultados que se deriven del uso de una aplicación concreta.

En [42], Kim et al. presentan una clasificación muy completa e, inicialmente, definen los datos ilícitos como aquellos de los que el usuario (o la aplicación en cuestión) obtiene resultados erróneos, o bien no es posible derivar un resultado, debido a problemas inherentes a los datos en sí. Las fuentes de datos ilícitos incluyen por un lado entradas y salidas erróneas y, por otro, errores en la transmisión y el procesamiento de los mismos.

En líneas generales, encontraremos cuatro tipos de anomalías en el trabajo con BC en la Web Semántica.

(A1): Conflictos debidos a defectos en la implementación de los datos. Una primera clasificación de este tipo de anomalías puede ser la que aparece en [42].

Así, los datos pueden ser nulos (debido a que no se hayan introducido siendo obligatorios, o bien debido a que esté permitido el tipo de dato nulo). También es posible que el dato se haya introducido pero sea erróneo: debido a que no cumpla las restricciones de integridad requeridas, o a que dichas restricciones de integridad no sean satisfactibles. Por último, los datos pueden no ser erróneos pero sí resultar inútiles. Este sería el caso de que aparezcan datos contradictorios o diferentes para una misma entidad en distintas bases que tienen que interoperar. Dentro de esta

última posibilidad entra también el caso de ambigüedad en los datos por el uso de abreviaturas, o la utilización de representaciones no estándar de los datos.

(A2): Las anomalías debidas a la inconsistencia del modelo. En ciertos casos tienen como origen el propio funcionamiento del demostrador. Esto es debido a que la consecuencia lógica no es equivalente a la obtención de respuestas cuando la parte intensional de la BC no es clausal.

(A3): Respuestas disyuntivas (una deficiencia lógica).

(A4): Inconsistencia en la terminología (TBox). Es decir, utilizamos una conceptualización inconsistente.

Las anomalías se deben a diversas razones, entre las que destacamos las siguientes¹:

- El conjunto de datos es inconsistente con el dominio de conocimiento, debido a las inconsistencias formales producidas por datos incorrectos.
- La base de datos no está nunca completa, esto es, el usuario seguirá introduciendo datos. Es posible que la componente intensional de la base de datos, la ontología, no refleje las creencias del usuario y, por tanto, sean necesarias nuevas actualizaciones. Esto ocurrirá frecuentemente en la WS.
- Es posible que algún criterio considerado en el ciclo de limpieza lleve a tomar una decisión que haga inconsistente la BC, en algún paso. Este tipo de errores, que pueden aparecer al gestionar la ontología visualmente, son los que tratamos de evitar con la interpretación mereotopológica del capítulo 5.
- Los sistemas de integración de datos necesitan una reformulación o una interpretación lógica, ya que proporcionan acceso simultáneo a múltiples fuentes de información. Si, por ejemplo, la transformación no respeta criterios lógicos, podemos transformar una BC consistente en una inconsistente.
- Un algoritmo pregunta-respuesta, si resulta ser incompleto, puede no generar testigos de alguna restricción de integridad de carácter existencial. Un caso extremo aparece cuando lo que se usa es un DAT para obtener respuestas [3].

¹Hemos destacado las que atañen al razonamiento automático con BC en DL al utilizar SRA.

- La inconsistencia puede producirse en la especificación del lenguaje de la ontología [24].
- El método elegido para el mantenimiento de la consistencia puede no ser lo suficientemente sólido y fallar con cierto tipo de actualizaciones.
- Los sistemas de minería de datos pueden producir salidas que no satisfacen las restricciones de integridad o los requerimientos de la ontología (Tipo A2).
- A veces, la expresividad se ve limitada y no permite usar nuevas características en la representación usual del conocimiento. Esto implica que algunas definiciones sean incorrectas. Es un problema de pobreza del lenguaje de representación.
- Por último, una ontología pobre (poco detallada o poco cuidada) asociada a datos consistentes puede producir inconsistencias. Esto puede ocurrir en las fases iniciales de la vida de la ontología.

Las bases de conocimiento en DL pueden verse afectadas por las anomalías clásicas de cualquier BC. Al ser una BC en DL, un par $\langle \mathcal{T}, \mathcal{A} \rangle$, donde \mathcal{A} es un conjunto de hechos (la componente *extensional*), y \mathcal{T} un conjunto de relaciones entre los conceptos (la componente *intensional*), las anomalías pueden producirse en cada una de esas componentes de manera individual o bien ser anomalías de carácter mixto. Por tanto, las razones antes expuestas son factibles en nuestro caso.

6.1. Anomalías en ontologías provisionales

Los formalismos de representación del conocimiento proporcionan estructuras para organizar el conocimiento, pero no mecanismos para compartirlo. Una ontología en sí es el resultado de formalizar una conceptualización compartida acerca de un dominio de conocimiento.

Sin embargo, las ontologías están en continuo cambio, es difícil considerarlas “terminadas” o cerradas, pues es necesario acceder al conocimiento que tienen como base, el conocimiento al que representan y, en general, dicho conocimiento es susceptible de evolucionar, haciéndose más preciso y definido. Es ese carácter dinámico el que hace necesario la revisión de forma permanente, (siendo posible, además, que dichas revisiones no siempre las realice una misma persona). A su vez, el contenido de las ontologías puede ser utilizado, simultáneamente, por aplicaciones diversas. Todas estas circunstancias causan incompatibilidades en las aplicaciones y, en muchos casos, hacen inaccesible el conocimiento almacenado en las ontologías.

6.1.1. Revisión ontológica

Con el nombre de (ontology versioning) *revisión ontológica*, [43], se conoce la capacidad de administrar los cambios que sufren las ontologías y los efectos que tienen esos cambios, mediante la creación y mantenimiento de las diferentes variantes que se van generando de las mismas.

Es necesario que las diferentes versiones de una misma ontología contengan el conocimiento de forma que sea compatible y reinterpretable desde cada una de ellas. Pero es aún más importante aclarar en qué contexto es necesario realizar un cambio en una ontología, y cuándo ese cambio da lugar a una nueva versión. No es lo mismo la relación conceptual entre dos conceptos distintos de una ontología, que la relación entre dos versiones de un mismo concepto que ha sufrido un cambio [43].

Otra cuestión es la necesidad de diferenciar entre los cambios que puedan producirse en el dominio y los producidos en la conceptualización: puede haber dos conceptualizaciones distintas de un mismo dominio y también puede cambiar el dominio sin cambiar la conceptualización (aunque esto último nos lleva a una modelización errónea del concepto). En cualquier caso, no puede determinarse de forma automática si un cambio es de un tipo u otro. Es un proceso en el que debe intervenir el usuario (diseñador, ingeniero, ...). Sí es posible utilizar la heurística para sugerir los efectos que tienen los cambios, como la generación de información errónea, incompatibilidad (de diferentes grados) entre las interpretaciones de los datos según las versiones anteriores ó posteriores, etc. (véase [43]).

En el contexto de esta memoria, las versiones nuevas son producidas por la detección de anomalías (no necesariamente de tipo inconsistencia).

6.1.2. Actualización

Por otra parte, habrá que decidir la manera en que se va a aplicar la actualización. A la hora de manipular los cambios en las ontologías, hay que tener en cuenta a qué nivel se realizan: a nivel de definiciones o en todo el fichero en bloque. Hay varias posibilidades para especificar los cambios [43]: en forma de lista de operaciones, mediante la sustitución de una ontología por su nueva versión, ó realizando una traducción de una versión a la siguiente (ontology mapping). Esta última opción es la más completa, pues proporciona la relación entre las distintas versiones de un mismo concepto.

Hay que destacar la importancia de que la evolución de la ontología sea transparente y que hay cambios que dan lugar a una revisión de la ontología en cuestión y cambios que producen una nueva ontología. En el caso de actualización por la presencia de anomalías sugeridas por argumentos, la docu-

mentación de dicha actualización (cuando no es corrección), es inducida por la corrección que hace el usuario, (véase capítulo 7).

6.1.3. Incompletitud y anomalías

La incompletitud de una KB se debe entender de dos formas distintas: la incompletitud lógica (con respecto a cierto tipo de preguntas) y la incompletitud debida a la falta de conceptos o roles (incompletitud de carácter expresivo). En el problema de la verificación de KB en DL aparece, por supuesto, el problema de la inconsistencia lógica.

Ya hemos hablado sobre cierto tipo de anomalías debido a la pobreza expresiva, las causadas por la carencia, desde el punto de vista del usuario, del perfil exacto de un determinado concepto. Cuando esto ocurre, el usuario trabaja con creencias sobre tales conceptos que aún no aparecen de forma explícita en la BC. A los conceptos de este tipo, existentes en las ontologías provisionales, los denominaremos *nociones*. La existencia de nociones en una ontología, provoca, además, que no sea posible distinguir dos conceptos amparados bajo la misma noción, es decir, que la ontología sea *gruesa*.

6.2. Clasificación de Anomalías en BC consistentes

En esta sección realizamos una clasificación de las anomalías utilizando las relaciones de indefinición insertables en la ontología RCC y su interpretación con regiones tipo “huevo-yema”. Recuérdese (figura 6.1), que el huevo a y su yema \tilde{a} estarán representados por la figura de línea intermitente y b y su yema \tilde{b} por la de línea punteada. La yema representará el concepto, y el huevo, la pulsación (noción) correspondiente. El siguiente teorema es consecuencia de aplicar la interpretación vaga a una BC, junto con la interpretación de las indefiniciones descritas en el capítulo 3.

Teorema. 6.2.1 *Al clasificar las posibles posiciones relativas de dos conceptos C_1, C_2 (representados respectivamente por la línea intermitente y la punteada), y sus nociones $\sigma(C_1), \sigma(C_2)$, obtenemos la tabla² 6.1.*

Hacemos uso, por tanto, de la interpretación fuerte del lenguaje de RCC aplicado a conceptos y nociones. Por ejemplo, el conjunto de figuras $\{22, 35\}$ se corresponde con el caso en que los conceptos se solapan parcialmente y las nociones están una contenida en la otra. Esto, con la interpretación fuerte de

²El guión entre los números 2 – 6, y 9 – 11 indica la sucesión de figuras comprendida entre ambos, es decir, 2 – 6 representa el conjunto de figuras $\{2, 3, 4, 5, 6\}$ de 6.1.

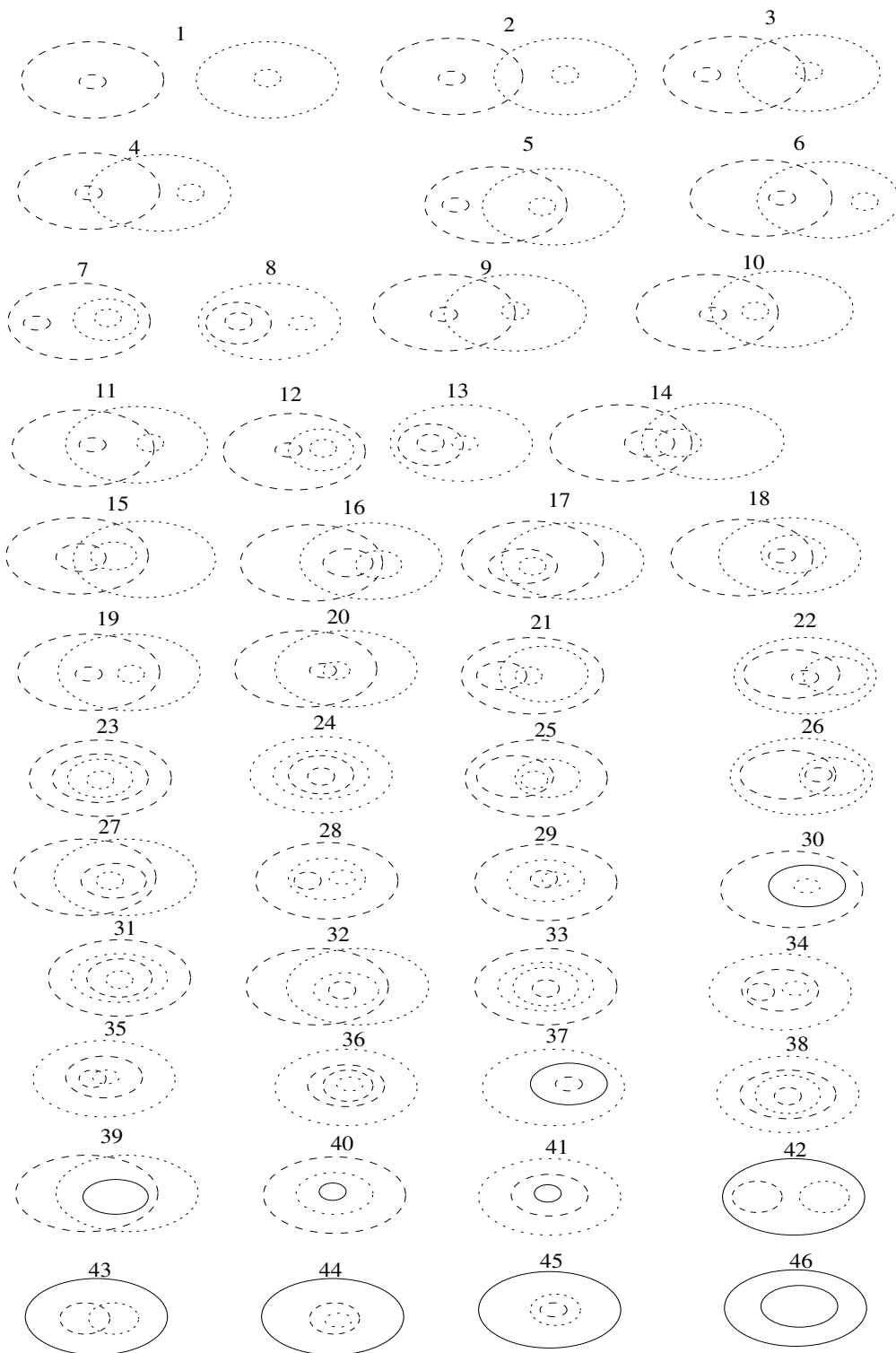


Figura 6.1: Todas las Posibles posiciones relativas de dos regiones tipo vagas

Cuadro 6.1: Tabla de posiciones relativas

		C_1, C_2				
		PPi_Σ	EQ_Σ	PP_Σ	PO_Σ	DR_Σ
$\sigma(C_1), \sigma(C_2)$	PPi_Σ	23, 25, 30, 31	41	33	21, 29	7, 12, 28
	EQ_Σ	45	46	44	43	42
	PP_Σ	36	40	24, 26, 37, 38	22, 35	8, 13, 34
	PO_Σ	17, 27	39	18, 32	14, 15, 16, 20	2-6,9-11,19
	DR_Σ	-	-	-	-	1
		I_3	I_2		\widetilde{PO}	I_7
		I_4		I_1		
		I_5				
INTERPRETACIÓN CON PULSACIÓN						

RCC, se expresa

$$PO_\Sigma(C_1, C_2) \wedge PP_\Sigma(\sigma(C_1), \sigma(C_2))$$

A la vista de la tabla, debemos hacer las siguientes consideraciones:

- En relación con la interpretación mediante pulsación introducidas en 3.4.1, podemos reconocer los distintos tipos de anomalías por indefinición que podrían aparecer en una base de conocimiento Σ :

- La columna encabezada por la relación PP_Σ corresponde con la interpretación de I_1 . Por tanto, la caracterización de I_1 es

$$I_1 \equiv PP_\Sigma \wedge C_\Sigma^\sigma$$

- Las columnas PP_Σ y EQ_Σ corresponden a la interpretación con pulsación de I_2 . Caracterización de I_2 :

$$I_2 \equiv (PP_\Sigma \vee EQ_\Sigma) \wedge C_\Sigma^\sigma$$

- Análogamente, la columna PPi_Σ coincide con la interpretación de I_3 y ésta, junto con la columna EQ_Σ , resulta ser la interpretación de I_4 . Las caracterizaciones de éstas son, pues:

$$I_3 \equiv PPi_\Sigma \wedge C_\Sigma^\sigma$$

$$I_4 \equiv (PPi_\Sigma \vee EQ_\Sigma) \wedge C_\Sigma^\sigma$$

- La interpretación con pulsación de la relación I_5 coincide con el contenido de las columnas PP_Σ , PPi_Σ , EQ_Σ y PO_Σ . Por tanto,

$$I_5 \equiv (PPi_\Sigma \vee EQ_\Sigma \vee PP_\Sigma \vee PO_\Sigma) \wedge C_\Sigma^\sigma$$

- La columna DR_Σ corresponde a la interpretación de I_7 . Luego

$$I_7 \equiv DR_\Sigma \wedge C_\Sigma^\sigma$$

- Por último, la tabla completa (unión de todas las posibles posiciones entre los dos conceptos y sus nociones) se corresponde con la unión de I_5 y I_7 , es decir, la interpretación con pulsación de I_8 , que es la relación de *indeterminación total*. No disponemos de una relación entre los dos conceptos.
- En relación con la figura 3.12, que aparece en 3.4.1, nótese que la intersección de las columnas correspondientes a I_2 y I_4 se corresponde con \widetilde{EQ} .
- Las casillas en blanco representan posiciones imposibles, como el caso en que, estando dos conceptos conectados, las respectivas nociones sean disjuntas (discretas). Esto no es posible ya que siempre se tiene $C \sqsubseteq \sigma(C)$, por la propia definición de noción. Corresponden, por tanto, a anomalías de tipo inconsistencia (ver subsección 6.4).
- Las casillas de la primera diagonal (de izquierda a derecha y de arriba abajo) contienen las posiciones que intuitivamente no presentan un tipo de anomalía. Este es el caso de 1, 46, 24, 25, Los casos en los que no se plantea conflicto son aquellos en que la pulsación conserva la relación entre los conceptos, de ahí que aparezcan en la diagonal.
- Las casillas restantes albergan las anomalías producidas al no conservarse la relación original mediante la pulsación. En las figuras 33 y 36, la relación PP se invierte al realizar la pulsación. En 39 y 43, los conceptos coinciden y sus pulsaciones se solapan (ó viceversa). Además hay casos menos aceptables en general, como el de 7, 12 y 28, en los que ambos conceptos son disjuntos (discretos), mientras sus pulsaciones pueden estar contenidas una en otra (o al revés, como en 8, 13, 34), e incluso coincidir, como en 42.

Teorema. 6.2.2 *La tabla 6.1, contiene todos los casos posibles.*

Demostración: Se probó, (en 3.1.1), que sólo existen ocho modelos de las extensiones de RCC con indefinición. Cada modelo L_i es un retículo y cada relación de indefinición I_i se interpretó con pulsación en el mismo capítulo, 3.4.1. En la tabla, aparecen exactamente esas relaciones. ■

6.3. Anomalías de tipo léxico-lógico

En el estudio de la semántica del lenguaje, el signo lingüístico tiene dos caras: *significante* y *significado*. El significante es la “parte material” del signo, el significado es la imagen mental que sugiere el significante. En el contexto en el que trabajamos en este capítulo, el hecho $\Sigma \models A \equiv B$, que expresa que dos conceptos A y B son equivalentes, se interpreta como “la base de conocimiento no es capaz de distinguir” los conceptos A y B entre sí, es decir, son dos conceptos con el mismo significante. De manera análoga, cuando ocurre que dos conceptos tienen la misma pulsación, es decir, $\Sigma \models \sigma(A) \equiv \sigma(B)$, se entenderá que tienen el mismo significado, o bien que sus definiciones coinciden. Para aplicar la clasificación clásica, identificaremos “mismo significante” con $EQ_{\Sigma}(A, B)$.

Desde esta perspectiva podemos estudiar las relaciones entre significante y significado, como son la monosemia, sinonimia y la polisemia. Estas dos últimas pertenecen al grupo de anomalías de tipo léxico-lógico. Por otro lado, podemos estudiar también las relaciones entre los significados, concretamente, la antonimia.

La reparación de las anomalías que describimos en esta sección se estudia extensamente en el siguiente capítulo.

6.3.1. Sinonimia lógica

La sinonimia entre dos conceptos A y B de la base de conocimiento Σ , ocurre cuando los conceptos no son equivalentes, pero tienen el mismo significado.

$$\begin{aligned}\Sigma \models A \neq B \\ \Sigma \models \sigma(A) \equiv \sigma(B)\end{aligned}$$

Por tanto, los conceptos en cuestión verifican:

$$\neg EQ_{\Sigma} \wedge EQ_{\Sigma}^{\sigma}$$

Como la interpretación fuerte no distingue RCC8, (teorema 5.1.2),

$$\neg EQ_{\Sigma} \rightarrow PO_{\Sigma} \vee DR_{\Sigma} \vee PP_{\Sigma} \vee PPi_{\Sigma}$$

De todo ello, la sinonimia viene descrita por la expresión

$$(PO_{\Sigma} \vee DR_{\Sigma} \vee PP_{\Sigma} \vee PPi_{\Sigma}) \wedge EQ_{\Sigma}^{\sigma}$$

Expresión que da lugar a cuatro casos que se corresponden con las posiciones desde la 42 a la 45 de la figura 6.1.

6.3.2. Polisemia lógica

La polisemia es fenómeno contrario al que acabamos de exponer: conceptos equivalentes (indistinguibles) en la base de conocimiento, pero con definiciones (pulsaciones) no equivalentes:

$$\Sigma \models A \equiv B$$

$$\Sigma \models \sigma(A) \neq \sigma(B)$$

En este caso, los conceptos en cuestión verifican:

$$EQ_{\Sigma} \wedge \neg EQ_{\Sigma}^{\sigma}$$

De nuevo, por el teorema 5.1.2, la polisemia viene descrita por la expresión

$$(PO_{\Sigma}^{\sigma} \vee DR_{\Sigma}^{\sigma} \vee PP_{\Sigma}^{\sigma} \vee PPi_{\Sigma}^{\sigma}) \wedge EQ_{\Sigma}$$

Las posiciones de la figura 6.1 que reflejan esta anomalía son 39, 40 y 41.

Nótese que no es posible el caso $DR_{\Sigma}^{\sigma} \wedge EQ_{\Sigma}$.

6.3.3. Monosemia lógica

La monosemia no es un caso de anomalía, por el contrario, este fenómeno expresa el caso ideal: el concepto coincide con su definición en la base de conocimiento, es decir, $\Sigma \models A \equiv \sigma(A)$.

6.4. Anomalías por inconsistencia de tipo léxico-lógico

A continuación describimos anomalías de tipo léxico-lógico que son inaceptables en cualquier caso, pues son signo de inconsistencia.

6.4.1. Antonimia lógica

Las relaciones entre los significados de los conceptos pueden ser de inclusión y de oposición. En sentido general, se entiende por antonimia el hecho

de que dos conceptos tengan significados contrarios. Esta realidad puede matizarse, dando lugar a tres casos distintos de oposición: complementariedad (fiel/infiel), reciprocidad (entrega/recepción) y antonimia propiamente dicha, que considera entre los significados una gradación (frío/caliente). Esta última es una anomalía por inconsistencia que expresamos de la manera siguiente:

$$\Sigma \models A \sqcap B \neq \perp$$

$$\Sigma \models \sigma(A) \sqcap \sigma(B) \equiv \perp$$

Es obvio que la conjunción de ambas circunstancias llevan a la inconsistencia de la base de conocimiento Σ , pues $\Sigma \models A \sqsubseteq \sigma(A)$.

6.4.2. Anomalías detectables por composición de relaciones me-reotopológicas o de proyecciones reticulares

En apartados anteriores se han estudiado anomalías por pares de conceptos. Otra opción es realizar la aproximación mediante composición de proyecciones reticulares. Esto es especialmente interesante porque más adelante utilizaremos interpretaciones espaciales basadas en RCC8. La idea es que resulta más sencillo hacer inferencias espaciales en el modelo (la representación gráfica) que en la teoría. Para detectar este tipo de anomalías consideramos RCC-bajo interpretación fuerte- como una metaontología, y utilizamos las tablas de composición como tablas de inferencia metaontológica. Veamos el siguiente ejemplo en el que consideremos la base de conocimiento Σ :

$$\mathcal{T} = \begin{cases} \text{Niño} \sqsubseteq \text{Hombre} \\ \text{Abuelo} \sqsubseteq \text{Hombre} \sqcap \text{Padre} \\ \text{Padre} \sqsubseteq \exists \text{TieneHijo.}(\text{Niño} \cup \text{Niña}) \sqcap \text{Niño} \end{cases} \quad \mathcal{A} = \begin{cases} \text{Padre(Pedro)} \\ \text{Hombre(Pedro)} \\ \text{Hombre(Juan)} \\ \neg \text{Niño(Juan)} \\ \neg \text{Niño(Pedro)} \end{cases}$$

Se tiene que $PP_{\Sigma}(\text{Niño}, \text{Hombre}) \wedge PO_{\Sigma}(\text{Hombre}, \text{Padre})$. Si Σ fuese consistente, entonces $\Sigma \models \bigvee_{R \in \widehat{PP} \circ \widehat{PO}} R(\text{Niño}, \text{Padre})$, es decir, $DR_{\Sigma}(\text{Niño}, \text{Padre}) \vee PO_{\Sigma}(\text{Niño}, \text{Padre}) \vee PP_{\Sigma}(\text{Niño}, \text{Padre})$. En este caso, $\Sigma \models PP(\text{Padre}, \text{Niño})$, lo cual es una contradicción. Luego Σ es inconsistente.

6.5. Anomalías detectadas por el razonador

En esta sección vamos a establecer las bases para detectar algunos tipos de inconsistencia mediante el análisis de los argumentos producidos por un demostrador automático (en nuestro caso OTTER).

Recordemos que, dada una base de conocimiento T , un **argumento** en T es un par $\langle \Pi, \phi \rangle$ donde $\Pi \subseteq T$ y $\Pi \vdash \phi$. Este concepto puede particularizarse en el contexto del uso de OTTER.

Definición. 6.5.1

1. Un **0-argumento** (un argumento para OTTER) es un par $\langle \Pi, \phi \rangle$ tal que Π es el conjunto de axiomas de una refutación de $\{\neg\phi\}$ que OTTER proporcione (lo cual escribiremos como $\Pi \vdash^0 \phi$).
2. Si $\langle \Pi, \phi \rangle$ es un 0-argumento, la **longitud de** $\langle \Pi, \phi \rangle$ es la longitud de la refutación de $\Pi \cup \{\neg\phi\}$ alcanzada por OTTER y la denotamos por $len(\langle \Pi, \phi \rangle)$.

De igual forma, la jerarquía de argumentos puede particularizarse al ámbito de uso del un demostrador. Por ejemplo, la clase de argumentos $A\exists(\Sigma)$ (definida de manera general en 1.4.17) se puede adaptar al demostrador OTTER: $A^0\exists(\Sigma) = \{\langle \Pi, \phi \rangle : \Pi \text{ es consistente y } \Pi \vdash^0 \phi\}$. En esta memoria nos limitaremos a analizar este nivel de la jerarquía argumentativa.

Definición. 6.5.2 Sea M un modelo de una teoría T . El **grafo** de M , denotado por \mathcal{M}_G , es la subestructura de M cuyos elementos son la interpretación de las constantes.

En general, cuando Σ no es clausal, es más conveniente utilizar el *entorno cognitivo*.

Definición. 6.5.3 Sea Σ un conjunto de fórmulas y M un modelo de Σ .

- El **mundo** de Σ , denotado por $\mathbb{W}(\Sigma)$, es el conjunto de las interpretaciones en Θ de las constantes en el lenguaje de Σ .
- Consideremos una interpretación de las funciones de Skolem de la forma clausal de Σ . El **entorno cognitivo** de Σ , $\Gamma(\Sigma)$, es la menor subestructura de la expansión de M al lenguaje clausal de Σ , que contiene a $\mathbb{W}(\Sigma)$.

Parece que la consistencia de un argumento depende únicamente de su entorno cognitivo. Esto es así para argumentos con la suficiente *credibilidad*. Particularicemos algunas de las definiciones sobre argumentos que presentamos en preliminares para facilitar el análisis de argumentos.

Definición. 6.5.4 Un *corte* de $\langle \Pi_1, \phi \rangle$ es un argumento $\langle \Pi, \neg(\phi_1 \wedge \dots \wedge \phi_n) \rangle$ donde $\{\phi_1, \dots, \phi_n\} \subseteq \Pi_1$. El corte es un *corte local* si $\Gamma(\Pi) \subseteq \Gamma(\Pi_1)$.

Definición. 6.5.5 $\langle \Pi, \alpha \rangle$ es *más conservativo que* $\langle \Pi', \beta \rangle$ si $\Pi \subseteq \Pi'$ y $\beta \vdash^0 \alpha$.

Definición. 6.5.6 Sea T una teoría, y ϕ una fórmula del lenguaje clausal de T .

- Una clausula tiene *ruido de Skolem* si tiene ocurrencias de símbolos de funciones de Skolem.
- El *grado de credibilidad* de un argumento $\langle \Pi, \phi \rangle$ es:

$$gr(\langle \Pi, \phi \rangle) = \frac{len(\langle \Pi, \phi \rangle) - |\{\eta \in Prueba^0(\Pi, \phi) : \eta \text{ tiene ruido de Skolem}\}|}{len(\langle \Pi, \phi \rangle)}$$

El grado de credibilidad estima la robustez del argumento de acuerdo con el uso de funciones de Skolem, funciones que pueden representar elementos *fantasma*, es decir, elementos sin nombre.

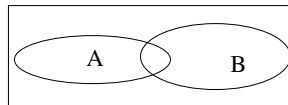
Una evaluación de Σ que ofrezca como respuesta un término con funciones de Skolem puede ser indicio de una anomalía.

Ejemplo. 6.5.7 La representación del conocimiento expresado en la base que vamos a considerar viene dada por el gráfico de la figura 6.2. En él aparecen dos regiones A y B que se solapan. La base de conocimiento está formada por las fórmulas que se indican en dicha figura. Éstas fórmulas describen la siguiente situación:

- Sólo existen esas dos regiones (A y B). (Axioma de clausura de dominio).
- Dichas regiones son distintas. (Axioma de nombres únicos).
- La posición de ambas regiones es de solapamiento y, a su vez, de conexión. Las fórmulas tercera y cuarta son los axiomas de completación correspondientes.
- Definición de solapamiento en función de la relación “ser parte de”.
- Por último, mediante el predicado de respuesta de OTTER, solicitamos los nombres de todas las regiones que sean parte de A .

OTTER clasifica las fórmulas de la base de conocimiento y numera las cláusulas generadas en dicho proceso. Como resultado de la clasificación de la fórmula que define el solapamiento, y debido a la cuantificación existencial, aparece el ruido de Skolem: dos regiones x e y se solapan si existe una región que es parte de ambas, esa región recibe el nombre de $f1(x, y)$.

Más adelante, haremos referencia a esta función de Skolem como $f_0(x, y)$ (cuando estudiemos la interpretación espacial de los argumentos, 8.3.2). Las cláusulas que utiliza en la prueba son cuatro. Estas cuatro cláusulas junto con $\text{Ans}(f_1(A, B))$ constituyen el argumento para OTTER. Se obtiene una prueba de $\text{Ans}(f_1(A, B))$ que contiene las cuatro cláusulas de partida mas las tres que se deducen de ellas. De esas siete (la longitud del argumento), tres tienen ruido de Skolem. Por tanto, el grado de credibilidad del argumento que se muestra en la fig.6.2 es $4/7$.



Base de Datos

```
-----
all x (x=A|x=B) .
A!=B.
all x y (x=A & y=B | x=B & y=A | x=A & y=A | x=B & y=B -> O(x,y)) .
all x y (x=A & y=B | x=B & y=A | x=A & y=A | x=B & y=B -> C(x,y)) .
all x y ((exists z (P(z,x) & P(z,y))) <-> O(x,y)) .
all x (P(x,A) -> $Ans(x)) .
```

Prueba facilitada por OTTER

```
-----
1[] x=x.
5[] x!=A|y!=B|O(x,y) .
14[] P($f1(x,y),x) | -O(x,y) .
16[] -P(x,A) | $Ans(x) .
27[hyper,5,1,1]O(A,B) .
65[hyper,14,27]P($f1(A,B),A) .
66[binary,65.1,16.1]$Ans($f1(A,B)) .
```

Figura 6.2: Una anomalía debida al ruido de Skolem y un 0-argumento

Capítulo 7

Resolución algorítmica de las anomalías

En este capítulo se muestra cómo traducir, de forma cognitivamente correcta, el análisis de una BC a un refinamiento gráfico, usando una herramienta de razonamiento espacial, el cálculo RCC. Dicha traducción consiste por tanto en aplicar el cálculo RCC al análisis local de ontologías provisionales. En segundo lugar, haremos un estudio global de carácter más práctico (mediante el uso de argumentos), pues trabajaremos sobre un argumento anómalo obtenido de la base de conocimiento y que ha sido producido, posiblemente, por un demostrador automático. El tamaño relativamente pequeño de éste permite manejar con cierta facilidad su representación gráfica.

7.1. Resolución local: refinamiento por pares de conceptos

En este apartado proporcionamos un método para obtener refinamientos de una ontología a partir de la interpretación fuerte de RCC. Estudiando las posiciones relativas de los conceptos por pares, así como las de sus respectivas nociones podemos detectar posibles anomalías (como por ejemplo, que dos conceptos coincidan y no lo hagan sus nociones; caso que podría interpretarse como polisemia: un mismo concepto con dos acepciones distintas). A su vez proponemos una forma de repararla mediante una serie de arreglos espaciales¹. Dichos arreglos consisten en hacer que la relación espacial que existe entre los conceptos coincida con la relación existente entre sus respectivas nociones. En algunos casos, bastará con modificar únicamente la relación entre

¹La propuesta de reparación que se aconseja no es la única posible, está basada en criterios de *minimalidad* de movimientos, como indicaremos más adelante.

los conceptos (*resp. nociones*) para que sea la misma que la existente entre las nociones (*resp. conceptos*), véase el ejemplo 7.1.1. Otras veces, es necesario transformar conceptos y nociones para alcanzar una posición intermedia dentro del entorno conceptual, (véase 7.1.2).

Hay que tener en cuenta que, basándonos en la interpretación mereotopológica, hemos clasificado las posiciones relativas entre conceptos y nociones, véase la tabla 6.1. De ellas, hemos seleccionado aquellas que indican posibles anomalías. Nótese que, en cualquier caso, es el usuario quien debe confirmar tal extremo y determinar si es necesario el arreglo espacial. Puede ocurrir que las posiciones relativas entre conceptos y entre nociones coincidan con un caso de posible anomalía en la tabla 6.1 pero el usuario, en ese caso concreto, no vea conflicto con el modelo pretendido. En este caso no es necesario, en principio, considerar arreglo alguno. Los movimientos de reparación de anomalías que presentamos están dirigidos a obtener definiciones de los conceptos involucrados a través del refinamiento de las nociones y, en algunos casos, de los propios conceptos.

Ejemplo. 7.1.1 Consideremos el caso en que dos conceptos C_1, C_2 se solapan con respecto a una base de conocimiento Σ , es decir, tenemos $PO_{\Sigma}(C_1, C_2)$ y sus nociones respectivas, $\sigma(C_1), \sigma(C_2)$ coinciden: $EQ_{\Sigma}^g(C_1, C_2)$. Una forma de solucionar este posible conflicto, siempre bajo la supervisión del usuario, consiste en añadir dos constantes que aseguren el solapamiento parcial de las nociones, dos constantes b_1 y b_2 que estén en la noción de un concepto y no en la del otro. De esta forma, $PO_{\Sigma}(C_1, C_2)$ y $PO_{\Sigma'}^g(C_1, C_2)$. Véase la figura 7.1 a la izquierda. Nótese que, en este caso, se necesitaría un refinamiento de la noción. Veamos un ejemplo:

$$\Sigma = \begin{cases} \text{Omnívoro} \sqsubseteq \text{Carnívoro} \sqcap \text{Herbívoro} \\ \text{Carnívoro} \sqsubseteq \text{Animal} \\ \text{Herbívoro} \sqsubseteq \text{Animal} \\ \text{Omnívoro}(\text{Oso}) \end{cases}$$

La situación es: $PO_{\Sigma}(\text{Carnívoro}, \text{Herbívoro}) \wedge EQ_{\Sigma}(\text{Carnívoro}, \text{Herbívoro})$. Como solución, añadimos dos constantes b_1 y b_2 . La nueva base de conocimiento es:

$$\Sigma' = \begin{cases} \text{Carnívoro} \sqsubseteq \text{Animal} \sqcap \neg\{b_2\} \\ \text{Herbívoro} \sqsubseteq \text{Animal} \sqcap \neg\{b_1\} \\ \text{Herbívoro}(b_2) \\ \text{Carnívoro}(b_1) \\ \text{Omnívoro}(\text{Oso}) \end{cases}$$

Se tiene que $PO_{\Sigma'}^g(\text{Carnívoro}, \text{Herbívoro}) \wedge PO_{\Sigma'}^g(\text{Carnívoro}, \text{Herbívoro})$.

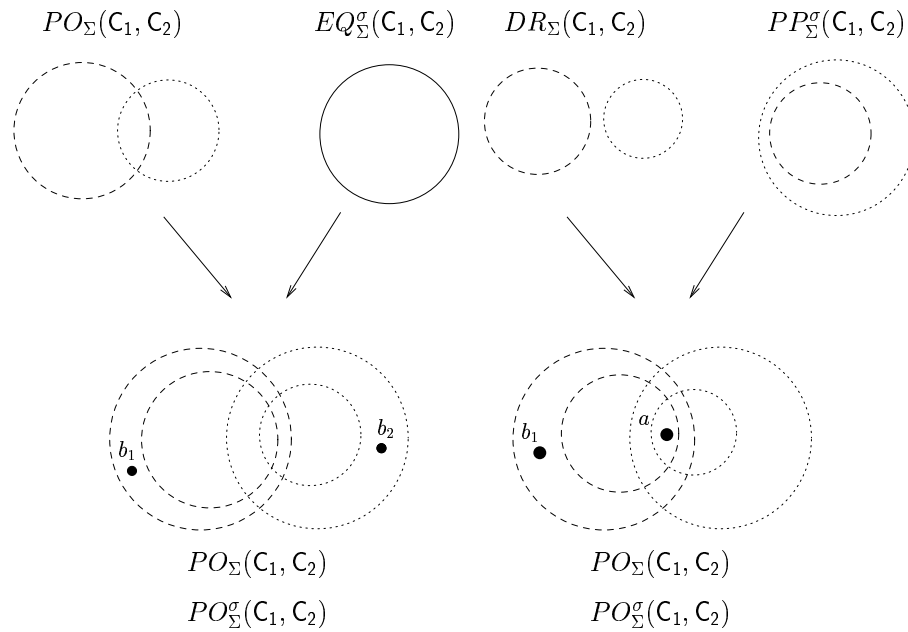


Figura 7.1: Dos ejemplos de arreglos espaciales

Ejemplo. 7.1.2 Sean Σ , una base de conocimiento, y C_1, C_2 , dos conceptos tales que $DR_{\Sigma}(C_1, C_2)$. Supongamos que sus nociones verifican $PP_{\Sigma}^g(C_1, C_2)$. Si el usuario considera el caso como anomalía, se sugiere el siguiente movimiento para solucionarlo: añadir sendas constantes, una común a los dos conceptos C_1, C_2 , y otra en la noción de C_1 pero que no esté en la noción de C_2 . De esta forma, llevamos conceptos y nociones a la misma posición de solapamiento. Véase la figura 7.1 a la derecha.

Consideremos los siguientes tipos de inserciones:

A : Añadir una constante a tal que $C_1(a), C_2(a)$

A_1 : Añadir una constante a_1 tal que $C_1(a_1), \neg C_2(a_1)$

A_2 : Añadir una constante a_2 tal que $\neg C_1(a_2), C_2(a_2)$

B : Añadir una constante b tal que $\sigma(C_1)(b), \sigma(C_2)(b)$

B_1 : Añadir una constante b_1 tal que $\sigma(C_1)(b_1), \neg \sigma(C_2)(b_1)$

B_2 : Añadir una constante b_2 tal que $\neg \sigma(C_1)(b_2), \sigma(C_2)(b_2)$

Con esta notación, los ejemplos de la figura 7.1 representan los arreglos $\langle B_1, B_2 \rangle$ y $\langle A, B_1 \rangle$ respectivamente.

Notación. 7.1.3 La notación que hemos adoptado para indicar el arreglo elegido es la usual en el caso de los conceptos. Recordemos que la noción de un concepto C es $\sigma(C) = \sqcap \{D : C \sqsubseteq D \in \Sigma\}$. Traducimos esta fórmula a su forma normal conjuntiva. Así, $\sigma(C)$ será equivalente a una fórmula $C'_1 \sqcap C'_2 \sqcap \dots \sqcap C'_n$. Pues bien, interpretamos $\sigma(C)(b)$ como la adición del conjunto de asertos $\{C'_1(b), C'_2(b), \dots, C'_n(b)\}$ a la ABox, o bien (de manera equivalente), es posible la transformación de las nociones como se ilustró en el ejemplo 7.1.1

Para presentar un procedimiento que abarque todos los casos recogidos en el cuadro 6.1, utilizamos la tabla 7.3, cuyo contenido pasamos a describir:

- En la primera columna, aparecen las posiciones relativas entre dos conceptos según Σ .
- Para cada una de las posiciones de la columna anterior, en la segunda, tenemos las posibles posiciones consistentes entre sus nociones. Ya hemos eliminado, por tanto, los casos imposibles para bases de conocimiento consistentes. Es, por tanto, una tabla para el tratamiento de bases de conocimiento consistentes.
- En la tercera columna, presentamos la posición relativa de conceptos y nociones tras efectuar el arreglo oportuno junto con su formalización en la interpretación débil.
- En la siguiente, aparece el conjunto de inserciones sugerido para resolver cada caso. Cada posibilidad engloba un conjunto de figuras de la tabla 6.1 cuyos números de referencia aparecen en la quinta columna.

Hay que recordar que cada caso se corresponde, con cada uno de los tipos de indefinición que aparecen en la tabla 6.1. Recordemos que tales indefiniciones son las obtenidas en el estudio de las extensiones de la teoría RCC mediante inserción de una relación de indefinición (capítulo 3).

Debemos precisar que la solución no es única. Por ejemplo, de $PP_\Sigma(C_1, C_2)$ y $PO_\Sigma^g(C_1, C_2)$: una posible solución es $PO_\Sigma(C_1, C_2)$, $PO_\Sigma^g(C_1, C_2)$ y una segunda posibilidad es $EQ_\Sigma(C_1, C_2)$, $EQ_\Sigma^g(C_1, C_2)$. El principio que dirige todo arreglo que presentamos en la tabla de arreglos espaciales es evitar retocar simultáneamente conceptos y nociones². De ahí que elijamos la primera opción. Algo parecido ocurre cuando tenemos $EQ_\Sigma(C_1, C_2)$ y a la vez $PP_\Sigma^g(C_1, C_2)$: podríamos solucionarlo de tres formas distintas:

- $EQ_\Sigma(C_1, C_2)$, $EQ_\Sigma^g(C_1, C_2)$

²La razón para esta restricción es sencilla: una transformación simultánea es prácticamente equivalente a eliminar conceptos y definiciones antiguas e insertar otras nuevas. Parece que este tipo de transformaciones puede afectar a la propia conceptualización del dominio.

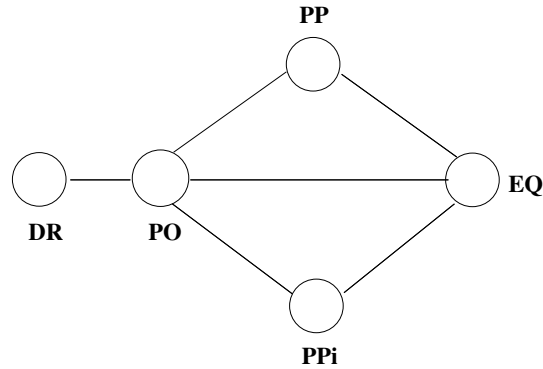


Figura 7.2: Movimientos en entornos conceptuales de R y R^σ

- $PO_\Sigma(C_1, C_2), PO_\Sigma^\sigma(C_1, C_2)$
- $PP_\Sigma(C_1, C_2), PP_\Sigma^\sigma(C_1, C_2)$

Descartamos la segunda opción porque lleva consigo modificar conceptos y nociones. La primera significa hacer equivalentes las nociones, es decir, introducir un número indeterminado de “elementos” de una noción en otra. Escoger la tercera nos asegura añadir una única constante, lo que la convierte en más aceptable que la primera, por criterios de prudencia (pocos cambios en la ontología original).

Resumiendo, estos dos ejemplos ilustran claramente los dos los criterios adoptados para elegir la solución: que los cambios afecten sólo a los conceptos o bien sólo a las nociones, y que el número de constantes de tipo Skolem a añadir sea mínimo.

Otro criterio que hemos elegido es el que las transformaciones sean elementales, es decir, que sean movimientos topológicos cualitativamente directos. Consideremos, teniendo en cuenta lo comentado en 2.4.1, la figura 7.2. Dicha figura representa la iconización de RCC5 como entorno conceptual. A la hora de elegir la solución adecuada para resolver las posiciones conflictivas, se ha decidido siempre mover el concepto/noción en la interpretación de los entornos conceptuales (de RCC5) de las relaciones entre los conceptos (R_1) y las nociones (R_2^σ), de manera que, tras el arreglo, tengamos una misma relación R entre los primeros y R^σ entre las segundas. Esta propiedad se recoge en el siguiente teorema y su validez es fácilmente comprobable con los datos recogidos en la tercera columna de la tabla de arreglos espaciales 7.3. Así, observemos que, en el caso comentado anteriormente, $PP_\Sigma(C_1, C_2)$ y $PO_\Sigma^\sigma(C_1, C_2)$, las tres posibles soluciones se obtienen de analizar qué movimientos, a partir del nodo PP y del nodo PO , llevan a una relación común, y esos son los tres señalados más arriba.

Teorema. 7.1.4 Sean $C_1 \sqsubseteq \sigma(C_1), C_2 \sqsubseteq \sigma(C_2)$ en una base de conocimiento Σ . Tras

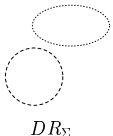

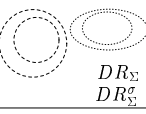
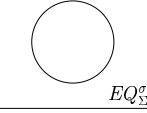
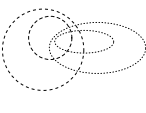
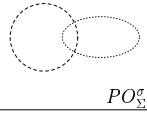
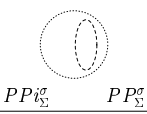

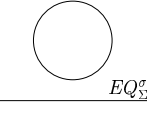
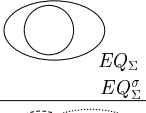
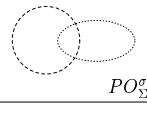
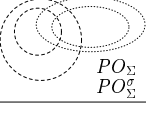
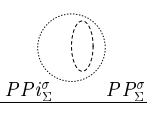
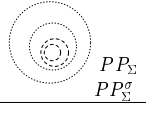

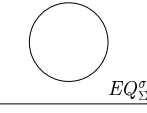
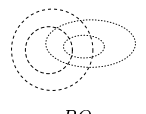
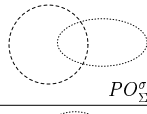
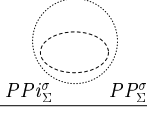

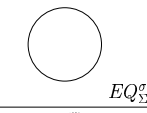
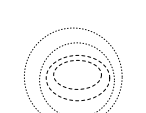
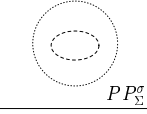
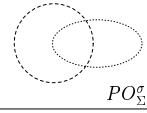

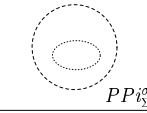
CONCEPTOS	NOCIONES	POSICIÓN FINAL	ARREGLOS	CASOS
 DR_{Σ}	 DR_{Σ}^{σ}	 DR_{Σ} DR_{Σ}^{σ}	--	1
	 EQ_{Σ}^{σ}	 PO_{Σ} PO_{Σ}^{σ}	A, B1, B2	42
	 PO_{Σ}^{σ}		A	2, 3, 4, 5, 6 9, 10, 11, 19
	 PP_{Σ}^{σ} PP_{Σ}^{σ}	PO_{Σ} PO_{Σ}^{σ}	A, B1	7, 12, 28 8, 13, 34
 EQ_{Σ}	 EQ_{Σ}^{σ}	 EQ_{Σ} EQ_{Σ}^{σ}	--	46
	 PO_{Σ}^{σ}	 PO_{Σ} PO_{Σ}^{σ}	A1, A2	39
	 PP_{Σ}^{σ} PP_{Σ}^{σ}	 PP_{Σ} PP_{Σ}^{σ}	A2	40, 41
 PO_{Σ}	 EQ_{Σ}^{σ}	 PO_{Σ} PO_{Σ}^{σ}	B1, B2	43
	 PO_{Σ}^{σ}		--	14, 15, 16, 20
	 PP_{Σ}^{σ} PP_{Σ}^{σ}		A2	21, 29, 22, 35
 PP_{Σ}	 EQ_{Σ}^{σ}	 PP_{Σ} PP_{Σ}^{σ}	B2	44, 45
	 PP_{Σ}^{σ}		--	23, 25, 30, 31 24, 26, 37, 38
	 PO_{Σ}^{σ}		A1	17, 27, 18, 32
	 PP_{Σ}^{σ}	PO_{Σ} PO_{Σ}^{σ}	A1, B2	33, 36

Figura 7.3: Tabla de arreglos espaciales

aplicar los arreglos, existe una relación $R \in RCC$ tal que $R_{\Sigma}(C_1, C_2) \wedge R_{\Sigma}^g(C_1, C_2)$.

ya hemos comentado, nos hemos limitado al caso de ontologías con datos consistentes, ya que en la tabla de arreglos espaciales hemos eliminado los casos de inconsistencia, como el mencionado en 7.1.

7.2. Ciclo de limpieza para ontologías

Cuando la ontología determina una estructura simple sobre sus conceptos, no es necesario recurrir a la interpretación débil, es más sencillo. Tener en cuenta representaciones espaciales que consideran sólo relaciones del tipo O, P . De esta forma, una vez representados espacialmente los conceptos, permitimos al usuario especificar la relación de la proyección reticular de cada una de éstas más conveniente. El siguiente *ciclo de limpieza* se basa en esa idea [1].

7.2.1. Ciclo de limpieza

La BC inicial, denotada por Σ , es una base de conocimiento en DL, es decir, un par Tbox/Abox. Mostraremos brevemente el esquema de los cuatro pasos del proceso de limpieza de Σ (ver Fig. 7.4):

1. En primer lugar, se produce un problema de satisfacción de restricciones (CSP) en el cálculo espacial relacional RCC8 (o RCC5), mediante una traducción adecuada cognitivamente de la Tbox a fórmulas de RCC. El problema se resuelve, obteniendo un escenario consistente, representado en 2D. Se añaden los hechos de la Abox. Para obtener el PSR, utilizamos a lo sumo \widehat{P} y \widehat{PO} . Estas relaciones se establecen utilizando como nodos del PSR los elementos de *conceptos*(Σ).
2. A continuación, el usuario debe hacer los correspondientes arreglos *reticulares* o *topológicos* en la representación gráfica. Los reticulares representan refinamientos de las relaciones entre pares de conceptos y los topológicos conllevan la sustitución de una relación por otra, incompatible con la primera pero *cognitivamente cercana* a ella. Es aconsejable que el usuario posteriormente considere que tiene una representación en RCC8 o RCC5. Esto restringe el conjunto de nociones en la ontología provisional.
3. Se aplica una traducción de las fórmulas de RCC a la BC. En la BC nueva, han cambiado algunas relaciones y se pueden haber inducido algunos conceptos nuevos. Esta traducción se basa, fundamentalmente, en reflejar

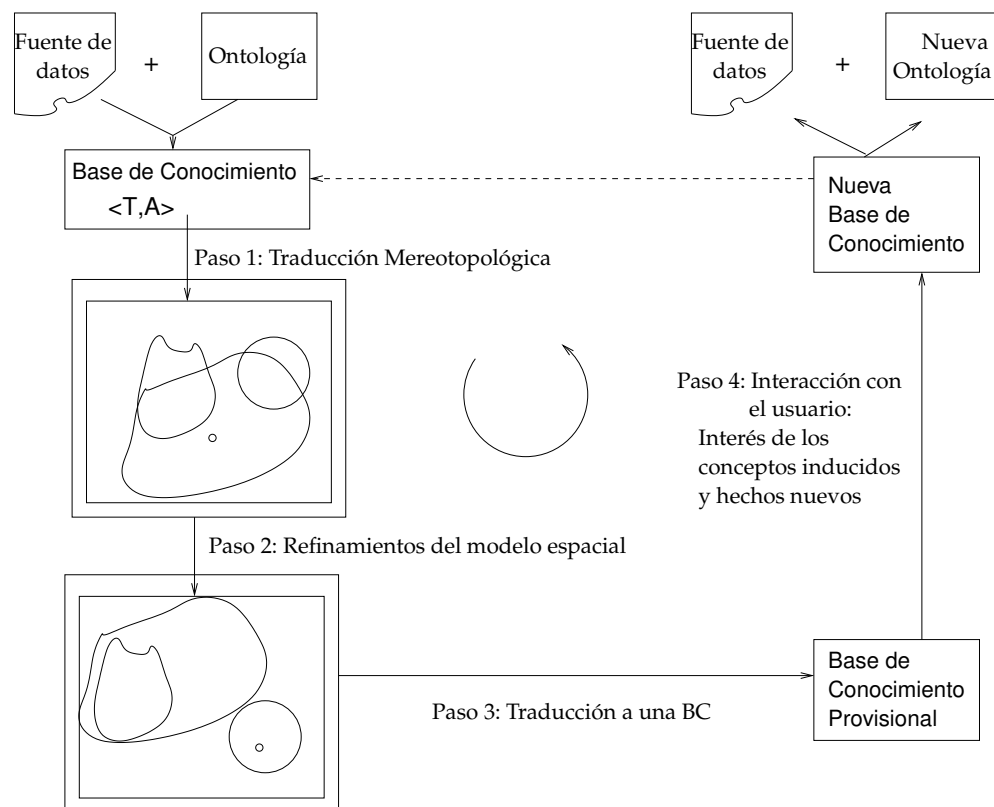


Figura 7.4: El proceso de transformación de ontologías

la disposición espacial final, considerando “frontera” los elementos que reflejen casos excepcionales.

4. Por último, el usuario interpreta o descarta los nuevos conceptos o elementos que la traducción anterior haya podido inducir.

El resultado del proceso es una nueva BC modificada por las creencias del usuario. Realmente, el proceso debe de ser un ciclo, puesto que es posible que la ontología sufra cambios y se proporcionen nuevos datos que motiven una nueva revisión. Realmente, el ciclo de limpieza arregla argumentos. Es decir, el método es aplicado para reparar una anomalía puesta de manifiesto por dicho argumento. Sin embargo, para simplificar, consideramos Σ tal argumento.

Para ilustrar el caso vamos a aplicar la metodología expuesta sobre una pequeña BC sobre la ontología de las relaciones familiares que mostramos en la fig. 7.5. Esta base de conocimiento puede ser, por ejemplo, un argumento.

$$\mathcal{T} = \begin{cases} \text{Woman} \sqsubseteq \text{Person} \sqcap \text{Female} \\ \text{Man} \sqsubseteq \text{Person} \sqcap \neg \text{Woman} \\ \text{Father} \sqsubseteq \text{Man} \sqcap \exists \text{hasChild}.\text{Person} \\ \text{Father} \sqsubseteq \text{Parent} \end{cases} \quad \mathcal{A} = \begin{cases} \text{Father}(\text{John}) \\ \text{Man}(\text{John}) \\ \text{Female}(\text{RuPaul}) \\ \text{Man}(\text{RuPaul}) \\ \text{Woman}(\text{Ann}) \\ \exists \text{hasChild}.\text{Person}(\text{Ann}) \end{cases}$$

Figura 7.5: Una BC provisional, $\Sigma = \langle \mathcal{T}, \mathcal{A} \rangle$, en la ontología de la familia

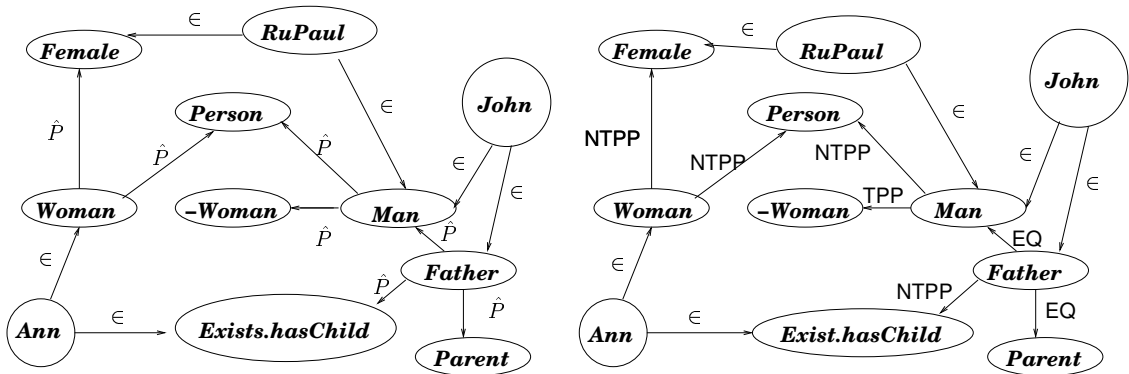


Figura 7.6: Grafo inicial del PSR (izda.) y solución (dcha.)

7.2.2. Paso 1: Traducción de la TBox

Cada concepto de Σ se interpreta como una región (regular) en el plano. Para realizar esta interpretación, aplicamos una traducción de Σ a un problema de CSP en RCC8 (o RCC5), traduciendo las fórmulas de la Tbox a un conjunto de fórmulas de RCC como vimos en 5.2, pero con la salvedad de que, en este caso, y teniendo en cuenta que trabajamos con $\text{conceptos}(\Sigma)$, basta con utilizar \hat{P} y \hat{PO} .

En nuestro ejemplo, el grafo CSP se da en la fig. 7.6, a la izquierda. Si hacemos una proyección reticular en RCC8, la solución del CSP generada por dicha proyección aparece a la derecha en la fig. 7.6.

La solución proporciona un escenario consistente que está representado espacialmente por regiones regulares del plano (no necesariamente conectadas) (ver fig.7.7). La existencia de este escenario es un resultado de J. Renz [59], como ya hemos comentado.

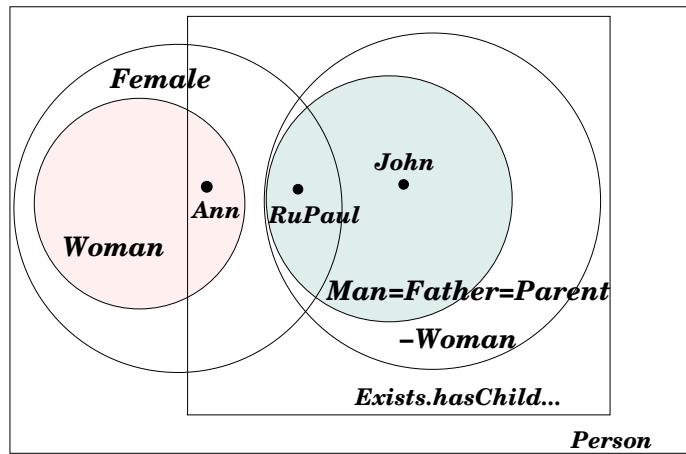


Figura 7.7: Representación espacial de la solución

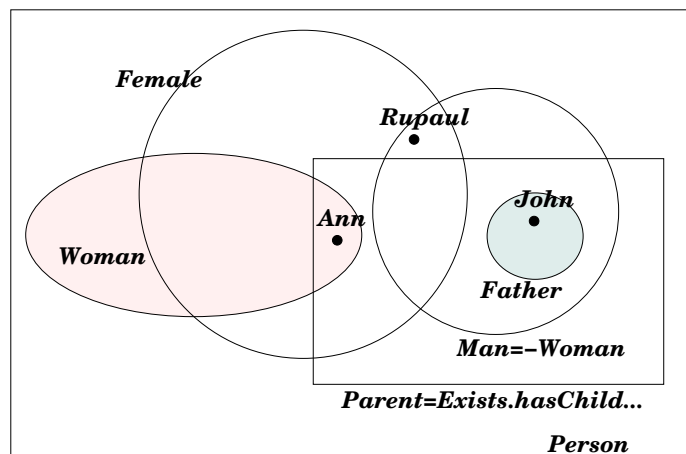


Figura 7.8: Representación espacial tras las reparaciones

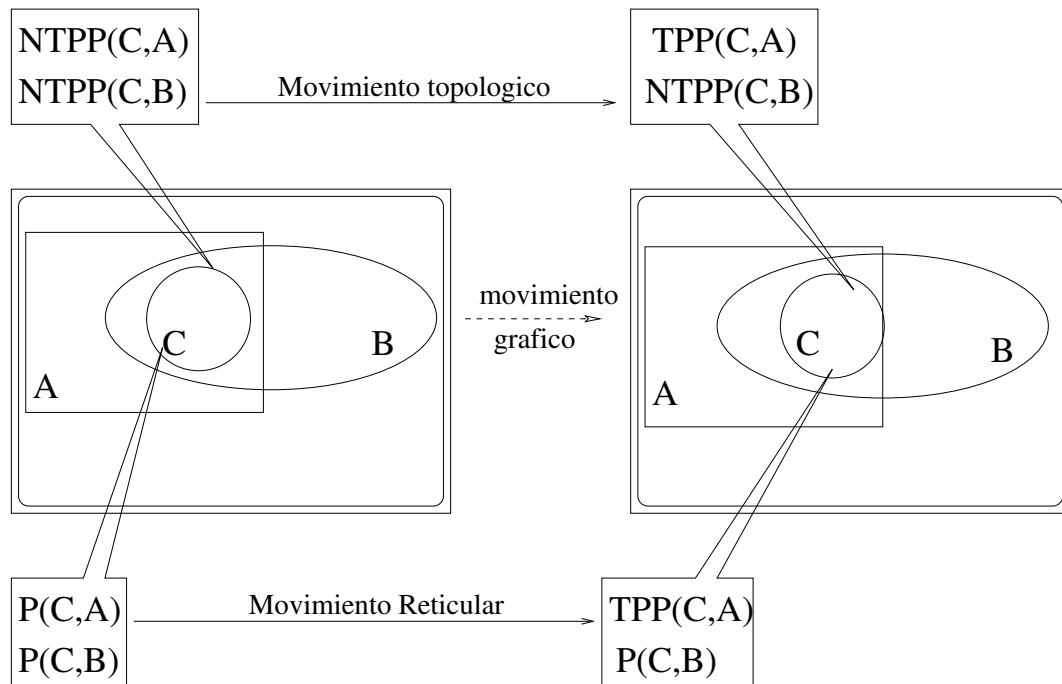


Figura 7.9: Ejemplo de refinamiento gráfico. El topológico está en RCC8

7.2.3. Paso 2: Arreglos espaciales

Son dos los tipos de movimientos sobre RCC que utilizaremos en adelante, los *movimientos reticulares* y los *topológicos*. Mediante dichos movimientos, corregimos las posiciones de las regiones Father, Man y Parent con respecto a \neg Woman, \exists hasChild.Person y a John y RuPaul. Distinguimos entre sí las tres primeras, que en realidad no tienen por qué coincidir, apareciendo en el escenario nuevas regiones, como por ejemplo $Man \setminus Father$, en las que se sitúan las interpretaciones de las constantes de la Abox, concretamente RuPaul.

La introducción de nuevas regiones auxiliares provoca a su vez la introducción de nuevos conceptos. En nuestro ejemplo, la nueva representación aparece en la fig. 7.8.

Con respecto a la interpretación de los hechos de la TBox, se deben entender, en principio, desde el punto de vista de una versión espacial en la Hipótesis del Mundo Cerrado: si el usuario aceptó $d \notin A^I$ (es decir, no ha colocado d dentro de la región espacial A), entonces razona como si fuese verdad que $\neg A(d)$.

7.2.4. Paso 3: De las relaciones espaciales a la nueva BC

Cuando el usuario cree que se ha alcanzado una representación adecuada del *escenario espacial real*, (véase fig. 7.8), el sistema la traduce a una nueva BC. En este sentido hay que hacer notar que la representación puede no ser adecuada con respecto a la BC original: esto ocurre cuando, por ejemplo, se usan arreglos topológicos. Las nuevas regiones regulares que hayan podido generarse no se corresponden con la BC original.

Por otro lado, las relaciones espaciales pueden ser inadecuadas con respecto a la ontología *mental*, las creencias del usuario. Esta anomalía se detecta cuando el usuario rechaza la BC traducida, produciendo en ese caso un refinamiento gráfico. En realidad, la traducción anterior se debe aplicar a la tabla de relaciones que se construirá a partir del grafo CSP que el usuario considere adecuado.

Dicha tabla de relaciones, se obtiene estableciendo las relaciones existentes entre los conceptos escogidos dos a dos.

7.3. Interpretación geométrica

A continuación definimos la traducción $R \in RCC \mapsto R^*$ de cada una de las relaciones de RCC a un conjunto de fórmulas de DL por recursión en el orden de la axiomatización de RCC (fig. 1.10). Cada relación del tipo $a \in A$ se traduce en $A(a)$. En adelante, los “elementos” serán las “interpretaciones espaciales de los símbolos de constante de Σ ”.

1. $C(A, B)^*$: Si en la representación existe algún elemento común a las regiones A y B , entonces $C(A, B)^* = \emptyset$. Si las dos regiones no tienen elementos en común, se introduce una constante de Skolem, que denotamos por $c_{A,B}$, y $C(A, B)^* = \{A(c_{A,B}), B(c_{A,B})\}$.
2. $DC(A, B)^* = \{A \sqsubseteq \neg B\}$.
3. Se debería de tomar $P(A, B)^* = \bigcup_{D \cap A \neq \emptyset} C(D, B)^*$, pero esta definición provoca la aparición en la práctica de muchas constantes de Skolem. Así que escogemos $P(A, B)^* = \{A \sqsubseteq B\}$.
4. $PP(A, B)^*$: Si existe una región D tal que $C(B, D) \wedge \neg C(A, D)$, entonces $PP(A, B)^* = P(A, B)^*$. En otro caso se introduce un nuevo nombre de concepto, el cual denotaremos mediante la expresión $N_{B \setminus A}$, y $PP(A, B)^* = P(A, B)^* \cup \{N_{B \setminus A} \sqsubseteq B, N_{B \setminus A} \sqsubseteq \neg A\}$.
5. $EQ(A, B)^* = \{A \equiv B\}$.

6. $O(A, B)^*$: Si existe D tal que $P(D, A) \wedge P(D, B)$, entonces: $O(A, B)^* = P(D, A)^* \cup P(D, B)^*$. Si no existe, se introduce un nuevo nombre de concepto, $N_{A \cap B}$, entonces: $O(A, B)^* = \{N_{A \cap B} \sqsubseteq A, N_{A \cap B} \sqsubseteq B\}$. La idea espacial del nuevo conjunto es minimal, es decir, no contiene ningún elemento.
7. $DR(A, B)^*$: En la representación no existe D tal que $P(D, A) \wedge P(D, B)$, pero puede existir un elemento en $A \cap B$. Si existiese tal elemento, basta razonar por defecto: se introduce un concepto de tipo *anormal*, $Ab_{A,B}$ y entonces $DR(A, B)^* = \{Ab_{A,B}(d) : d \in A \cap B\}$. En caso de que existiese, $DR(A, B)^* = \emptyset$.
8. $PO(A, B)^*$ contiene a $O(A, B)^*$ y, razonando para las fórmulas $\neg P(A, B)$ y $\neg P(B, A)$ de manera similar a (4), tenemos
 $PO(A, B)^* = O(A, B)^* \cup \{N_{B \setminus A} \sqsubseteq B, N_{B \setminus A} \sqsubseteq \neg A, N_{A \setminus B} \sqsubseteq A, N_{A \setminus B} \sqsubseteq \neg B\}$.
9. $EC(A, B)^*$ contiene a $C(A, B)^*$ y se añaden las fórmulas obtenidas al razonar como en (7) con la fórmula $\neg O(A, B)$. Así, obtendremos, en primer lugar $C(A, B)^*$. Este puede ser \emptyset (caso de que exista en la representación un elemento común a A y B), entonces añadiremos un nuevo concepto tipo Anormal $Ab_{A,B}$ y será: $EC(A, B)^* = \{Ab_{A,B}(d) : d \in A \cap B\}$. En caso de que al calcular $C(A, B)^*$ aparezca un nuevo elemento $c_{A,B}$, tendremos: $EC(A, B)^* = C(A, B)^* \cup \{Ab_{A,B}(c_{A,B})\}$.
10. $TPP(A, B)^*$: Si existe en la representación C tal que $EC(C, A) \wedge EC(C, B)$, entonces
 $TPP(A, B)^* = PP(A, B)^* \cup EC(C, A)^* \cup EC(C, B)^*$.
- Si no, es necesario añadir un nuevo nombre de concepto, $N_{A \supset B}$, y entonces
 $TPP(A, B)^* = PP(A, B)^* \cup EC(N_{A \supset B}, A)^* \cup EC(N_{A \supset B}, B)^*$
- La interpretación espacial deseada de $N_{A \supset B}$ es *minimal*, es decir, sin elementos en su interior.
11. $NTPP(A, B)^*$ contiene a $PP(A, B)^*$ y se razona de manera similar a (1) para asegurar que no existe una región externamente conectada a las dos. Si no existe una región D tal que $EC(A, D)$, entonces, $NTPP(A, B)^* = PP(A, B)^*$. En otro caso, será $NTPP(A, B)^* = PP(A, B)^* \cup O(D, B)^*$.

La notación utilizada en los nuevos nombres de conceptos no refleja ninguna característica deseada del concepto. Por ejemplo, no es válido, en principio, que $N_{A \cap B} \equiv A \sqcap B$.

	Female	Woman	Man	Father	Parent	Person	\neg Woman	\exists hasChild.P
Female	EQ	PO	PO	DC	PO	NTPP	PO	PO
Woman	PO	EQ	DC	DC	PO	NTPP	EC	PO
Man	PO	DC	EQ	NTPPi	PO	NTPP	EQ	PO
Father	DC	DC	NTPP	EQ	NTPP	NTPP	NTPP	NTPP
Parent	PO	PO	PO	NTPPi	EQ	NTPP	PO	EQ
Person	NTPPi	NTPPi	TPPi	NTPPi	NTPPi	EQ	NTPPi	NTPPi
\neg Woman	PO	EC	EQ	NTPPi	PO	NTPP	EQ	PO
\exists hasChild.P	PO	PO	PO	NTPPi	EQ	NTPP	PO	EQ

Figura 7.10: Tabla de relaciones resultante

En nuestro ejemplo de trabajo, la tabla de relaciones que resulta aparece en la fig. 7.10. En ella, P es la abreviatura escogida para Person. La BC obtenida está en la fig. 7.11.

7.3.1. Paso 4: Necesidad de interpretar lo obtenido

Una vez obtenida una nueva base de conocimiento, nos vemos en la necesidad de interpretar lo obtenido, ponerle nombre a los conceptos que han surgido en el proceso, y decidir si los elementos que tenía antes pertenecen a los nuevos conceptos que se han generado. Es decir, resituar los elementos, los conceptos y las relaciones entre ellos. Por ejemplo, el usuario debe decidir si $RuPaul \in N_{Female \setminus Woman}$. Es posible que se descarte algún nuevo concepto, pues la información que aporte sea irrelevante para el usuario, o que la representación gráfica que el usuario ha realizado sea inadecuada para sus creencias. En nuestro caso, el usuario llama a $N_{Female \setminus Woman}$ con el nombre *Crossdresser*, y llama a $N_{Parent \cap Woman}$ con el nombre *Mother*. La BC final aparece a la derecha en la fig. 7.12.

La traducción produce el nombre de concepto $Ab_{Woman, \neg Woman}$ y el hecho $Ab_{Woman, \neg Woman}(C_{Woman, \neg Woman})$ está incluido en la Abox, pero no se añade ninguna relación entre $Ab_{Woman, \neg Woman}$ y *Woman* ó *\neg Woman*. Es preferible que siga siendo una noción (el usuario no ha decidido si es un nuevo concepto o sólo es un caso excepcional que hay que eliminar).

Teorema. 7.3.1 *Las operaciones introducidas aquí preservan la consistencia.*

Demostración: Los modelos espaciales del PSR del paso 2 son modelos de Σ . La configuración espacial del paso 2 es un modelo de la ontología obtenida en el paso 3. El paso 4 sólo renombra conceptos o elimina fórmulas, por tanto preserva la consistencia. ■

$$\mathcal{T}' = \left\{ \begin{array}{l}
\text{Father} \sqsubseteq \text{Person} \sqcap \text{Parent} \sqcap \text{Man} \\
\text{Father} \sqsubseteq \neg\text{Woman} \sqcap \neg\text{Female} \\
\text{Female} \sqsubseteq \text{Person} \\
\text{Man} \equiv \neg\text{Woman} \\
\text{Father} \sqsubseteq \text{Parent} \\
\text{Woman} \sqsubseteq \text{Person} \\
\text{Man} \sqsubseteq \text{Person} \\
\text{Parent} \sqsubseteq \text{Person} \\
\exists\text{hasChild}.\text{Person} \equiv \text{Parent} \\
N_{\text{Woman} \sqcap \text{Female}} \sqsubseteq \text{Woman} \sqcap \text{Female} \\
N_{\text{Woman} \setminus \text{Female}} \sqsubseteq \text{Woman} \sqcap \neg\text{Female} \\
N_{\text{Female} \setminus \text{Woman}} \sqsubseteq \text{Female} \sqcap \neg\text{Woman} \\
N_{\text{Parent} \sqcap \text{Woman}} \sqsubseteq \text{Parent} \sqcap \text{Woman} \\
N_{\text{Parent} \setminus \text{Woman}} \sqsubseteq \text{Parent} \sqcap \neg\text{Woman} \\
N_{\text{Woman} \setminus \text{Parent}} \sqsubseteq \text{Woman} \sqcap \neg\text{Parent} \\
N_{\text{Man} \sqcap \text{Female}} \sqsubseteq \text{Man} \sqcap \text{Female} \\
N_{\text{Man} \setminus \text{Female}} \sqsubseteq \text{Man} \sqcap \neg\text{Female} \\
N_{\text{Female} \setminus \text{Man}} \sqsubseteq \text{Female} \sqcap \neg\text{Man} \\
N_{\text{Parent} \sqcap \text{Female}} \sqsubseteq \text{Parent} \sqcap \text{Female} \\
N_{\text{Parent} \setminus \text{Female}} \sqsubseteq \text{Parent} \sqcap \neg\text{Female} \\
N_{\text{Female} \setminus \text{Parent}} \sqsubseteq \text{Female} \sqcap \neg\text{Parent} \\
N_{\text{Parent} \setminus \text{Man}} \sqsubseteq \text{Parent} \sqcap \neg\text{Man} \\
N_{\text{Man} \setminus \text{Parent}} \sqsubseteq \text{Man} \sqcap \neg\text{Parent}
\end{array} \right.$$

$$\mathcal{A}' = \left\{ \begin{array}{l}
\text{Female}(\text{RuPaul}) \\
\text{Man}(\text{RuPaul}) \\
\text{Father}(\text{John}) \\
\text{Man}(\text{John}) \\
\text{Parent}(\text{John}) \\
\text{Woman}(\text{Ann}) \\
\exists\text{hasChild}.\text{Person}(\text{Ann}) \\
\text{Female}(\text{Ann}) \\
\text{Woman}(C_{\text{Woman}, \neg\text{Woman}}) \\
\neg\text{Woman}(C_{\text{Woman}, \neg\text{Woman}}) \\
N_{\text{Man} \sqsubseteq \text{Female}}(\text{RuPaul}) \\
\text{Ab}_{\text{Woman}, \neg\text{Woman}}(C_{\text{Woman}, \neg\text{Woman}})
\end{array} \right.$$

Figura 7.11: BC de la representación espacial

$$\mathcal{T}'' = \left\{ \begin{array}{l} \text{Father} \sqsubseteq \text{Parent} \sqcap \text{Man} \sqcap \neg \text{Female} \\ \text{Female} \sqsubseteq \text{Person} \\ \text{Woman} \sqsubseteq \text{Person} \\ \text{Man} \sqsubseteq \text{Person} \\ \text{Parent} \sqsubseteq \text{Person} \\ \text{Man} \equiv \neg \text{Woman} \\ \exists \text{hasChild}.\text{Person} \equiv \text{Parent} \\ \text{Crossdresser} \sqsubseteq \text{Female} \\ \text{Crossdresser} \sqsubseteq \neg \text{Woman} \\ \text{Mother} \sqsubseteq \text{Parent} \\ \text{Mother} \sqsubseteq \neg \text{Woman} \\ \text{Ab}_{\text{Woman}, \neg \text{Woman}} \sqsubseteq \text{Woman} \sqcap \neg \text{Woman} \end{array} \right.$$

$$\mathcal{A}'' = \mathcal{A}' \cup \left\{ \begin{array}{l} \text{Mother}(\text{Ann}) \\ \text{Crossdresser}(\text{RuPaul}) \end{array} \right\}$$

Figura 7.12: BC después del último paso

Capítulo 8

Limpieza de anomalías dirigida por un SRA

En este capítulo afrontaremos la reparación de anomalías desde un nuevo punto de vista. En este caso, se analizan los argumentos atendiendo a parámetros que vienen asociados a características específicas del argumento y de la ontología sometida a estudio. Estas características deben provenir fundamentalmente de un estudio específico de la terminología de la ontología. Para no incidir en este estudio, que no es objetivo de la memoria, vamos a elegir como ontología para los datos la propia RCC. Una vez detectado un argumento anómalo, el experto en el dominio, localiza en ese conjunto relativamente pequeño de fórmulas dónde está el error. La reparación consistirá, usualmente, en la corrección de un dato (aunque en algún caso hay que reparar la ontología).

La detección de inconsistencias se realiza mediante la comparación de la información que deduce el agente con la que contiene la ontología y el propio

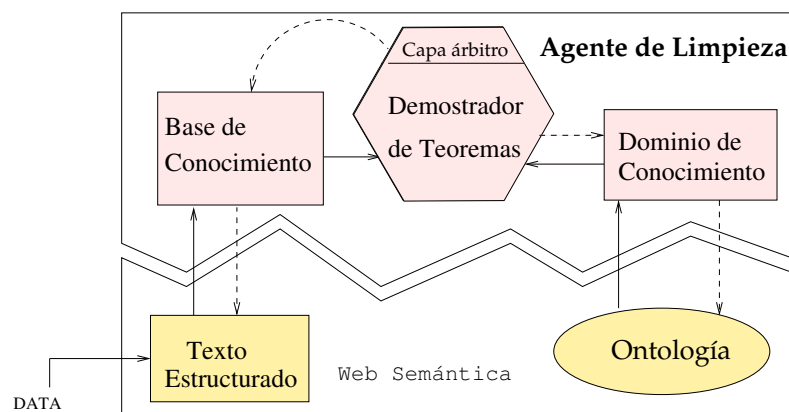


Figura 8.1: Proceso de limpieza

dominio de conocimiento. En la fig. 8.1 mostramos de manera esquemática la propuesta para el diseño de un agente de limpieza.

La base de conocimiento está expresada en texto estructurado y la ontología proporciona la traducción del dominio de conocimiento. El demostrador automático detectará y deberá reparar las anomalías existentes, tanto en la base como en el dominio.

Durante el proceso, una componente asociada al demostrador, la capa árbitro, clasifica la información que va siendo generada. La clasificación se realiza en función de los valores que toman ciertos parámetros que estiman el grado de interés de dicha información, como por ejemplo el número de pasos necesarios en el proceso. El objetivo de obtener esa clasificación es el de incluir o no la información en la BC. Las tareas principales en el proceso son las de recargar y actualizar la información, además de la de valorar la relevancia de la información generada. La capa árbitro debería actuar muy rápidamente. Aunque los criterios de clasificación dependen de la ontología, en esta memoria proponemos uno para detectar, al menos, consistencia.

El problema que surge es el de la necesidad de razonar con inconsistencias en texto estructurado. Este tipo de razonamiento puede ser tratado con el demostrador de teoremas, pero debido a la dificultad que entraña la verificación de grandes bases de conocimiento -asociadas a ontologías complejas-, esta cuestión hará que el agente demore su reacción.

Apuntamos, pues, la principal desventaja para la integración de un demostrador automático de teoremas en el agente, la pérdida del requisito de tiempo real. Sin embargo, dicho requisito no es relevante para este problema. Se puede asumir que el sistema funcione como un servicio de limpieza nocturno, como un asistente personal; el sistema depura los metadatos que el usuario ha incluido en la Web Semántica, usando el *tiempo libre* del ordenador.

8.1. Ciclo de limpieza asistido por un demostrador automático de teoremas

La metodología que seguimos está pensada para BC donde la ontología se supone consistente, aunque no es una condición necesaria. De ese modo las anomalías provienen de la ABox. Dicha metodología está esquematizada en la figura 8.2. En primer lugar seleccionamos un argumento y un predicado e iniciamos el ciclo realizando preguntas sencillas del tipo SQL. Las respuestas a algunas de esas preguntas se obtienen directamente en la BC, pues aparecen en ella de manera explícita. Por otro lado, podemos hacer preguntas complejas, que utilizan incluso predicados que no aparecen en los hechos explícitos de la

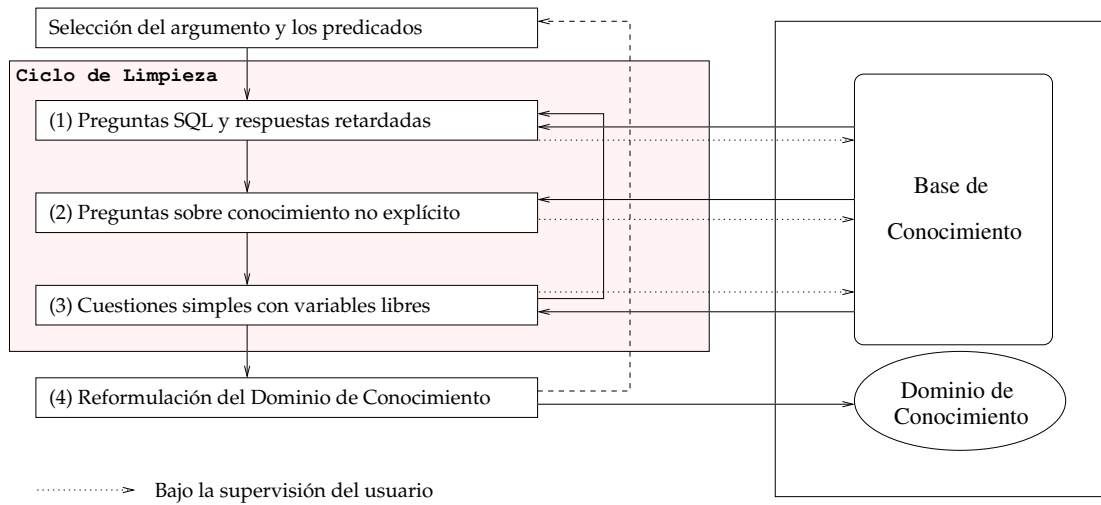


Figura 8.2: Esquema del proceso de limpieza usando un DAT

base de conocimiento. La idea común es la de añadir información a la BC. En determinadas ocasiones, el proceso nos llevará a realizar una reformulación del dominio de conocimiento, con el objeto de repararlo, debiéndose reiniciar el ciclo.

Tenemos que hacer notar que, mientras el lenguaje SQL está orientado a dar respuesta a las consultas efectuadas (respuestas que se encuentran en la base de conocimiento), el demostrador automático OTTER, mediante su predicado respuesta \$Ans(), está orientado a prueba, siendo incompleto e indecidible, es decir, no ofrecerá todas las respuestas y repetirá muchas.

El proceso descrito no termina de aplicarse nunca, ya que, aunque el usuario puede considerar que ha finalizado la generación de argumentos que dan lugar a anomalías, la BC está abierta a recibir nueva información, hecho que obliga a reiniciar el ciclo. Ilustraremos la utilidad de este ciclo analizando un caso donde la parte intensional de la base de conocimiento es sintácticamente compleja: una base de conocimiento con RCC como ontología asociada.

8.2. Bases de conocimiento en RCC

Más adelante, aplicaremos la metodología anteriormente descrita al caso de las bases espaciales cualitativas en el lenguaje RCC. Desde el punto de vista de DL, RCC es una teoría muy expresiva: al traducir los axiomas de RCC a DL, se utilizan constructores de roles muy complejos:

Rol	RCC	DL
<i>DC</i>	$\neg C(x, y)$	$DC \equiv \neg C$
<i>P</i>	$\forall z[C(z, x) \rightarrow C(z, y)]$	$P \equiv \neg(C^- \circ \neg C)$
<i>PP</i>	$P(x, y) \wedge \neg P(y, x)$	$PP \equiv P \sqcap (\neg P)^-$
<i>O</i>	$\exists z[P(z, x) \wedge P(z, y)]$	$O \equiv P^- \circ P$
<i>DR</i>	$\neg O(x, y)$	$DR \equiv \neg O$
<i>EC</i>	$C(x, y) \wedge \neg O(x, y)$	$EC \equiv C \sqcap \neg O$
<i>TPP</i>	$PP(x, y) \wedge \exists z[EC(z, x) \wedge EC(z, y)]$	$TPP \equiv PP \sqcap EC^- \circ EC$
<i>NTPP</i>	$PP(x, y) \wedge \neg \exists z[EC(z, x) \wedge EC(z, y)]$	$NTPP \equiv PP \sqcap \neg(EC^- \circ EC)$
<i>EQ</i>	$P(x, y) \wedge P(y, x)$	$EQ \equiv P \sqcap P^-$
<i>PO</i>	$O(x, y) \wedge \neg P(x, y) \wedge \neg P(y, x)$	$PO \equiv O \sqcap \neg P \sqcap (\neg P)^-$

Estos resultados se obtienen de manera directa, utilizando operadores tales como la composición de roles y el inverso de rol que aparecen en la tabla 1.3.

Axioma en RCC	DL
$\forall x C(x, x)$	$id(\top) \sqsubseteq C$
$\forall xy[C(x, y) \rightarrow C(y, x)]$	$C \sqsubseteq C^-$

Existen propuestas de extensiones del lenguaje OWL donde las relaciones espaciales de RCC aparecen como primitivas (véase [45]). Por ejemplo, la traducción de DL a OWL del segundo axioma de la tabla: $\forall xy[C(x, y) \rightarrow C(y, x)]$, para dos regiones x e y , se reescribe como sigue:

```
< owl:Class rdf:ID = "Region" / >
< owl:SymmetricProperty rdf:ID = "C" >
  < rdfs:domain rdf:resource = "#Region" / >
  < rdfs:range rdf:resource = "#Region" / >
< /owl:SymmetricProperty >
```

En realidad, lo que hemos expresado es la doble implicación en el axioma, ya que si la conexión C es simétrica, entonces también su inversa C^- lo es.

Por último, veamos la traducción a DL de algunos teoremas en RCC.

Teorema en RCC	DL
$\forall xy[x = y \leftrightarrow \forall z[C(z, x) \leftrightarrow C(z, y)]]$	$\neg(C^- \circ \neg C) \equiv EQ$
$\forall xy[x = y \leftrightarrow \forall z[O(z, x) \leftrightarrow O(z, y)]]$	$\neg(O^- \circ \neg O) \equiv EQ$
$\forall xy[PP(x, y) \rightarrow \exists z[P(z, y) \wedge \neg O(z, x)]]$	$PP \sqsubseteq (\neg O)^- \circ P$
$\forall xy[PO(x, y) \rightarrow (\exists z[P(z, y) \wedge \neg O(z, x)] \wedge \exists w[P(w, x) \wedge \neg O(w, y)])]$	$PO \sqsubseteq ((\neg O)^- \circ P) \sqcap (P^- \circ \neg O)$

Formalmente se definió en el capítulo 1 la noción de modelo topológico, véase 1.8.3.

Cada estructura se expande a una en el lenguaje completo de RCC, mediante la interpretación natural de las otras relaciones [33]. Recuérdese que si Ω es un espacio T_3 conectado no trivial, la expansión natural de cualquier estructura topológica en Ω al lenguaje completo es un modelo de RCC.

8.3. Interpretación espacial de los argumentos

El objetivo de esta sección es relacionar la consistencia de argumentos con características espaciales.

A partir de ahora fijaremos un modelo topológico Θ , el modelo espacial con el que vamos a trabajar, y K una base de datos que representa a Θ (esto es, K es un conjunto de fórmulas atómicas básicas tal que $\Theta \models K$). Para simplificar suponemos que el modelo satisface el axioma de nombres únicos. Tres teorías describen el modelo: la formalización de la teoría de base de datos de Reiter $T_{DB}(K)$, la teoría $RCC(K)$, cuyos axiomas son los de K junto con RCC , y $RCC(T_{DB}(K))$. Las tres teorías tienen un lenguaje común, L_K .

8.3.1. Condiciones sintácticas para la consistencia

Definición. 8.3.1 Sea K una base de conocimiento (un conjunto de fórmulas atómicas básicas) para Θ . La base K

- es **C-completa** si para $a, b \in L_K$

$$\Theta \models C(a, b) \implies C(a, b) \in K$$

- es **extensional** para P si para $a, b \in L_K$

$$P(a, b) \notin K \implies \exists c \in L_K [C(c, a) \in K \wedge C(c, b) \notin K]$$

- es **refinada** si para $a, b \in L_K$

$$\Theta \models O(a, b) \implies \exists c \in L_K [\{P(c, a), P(c, b)\} \subseteq K]$$

- es **cerrada bajo solapamiento**¹ si para $a, b \in L_K$

$$\exists c \in L_K [\{P(c, a), P(c, b)\} \subseteq K] \implies O(a, b) \in K$$

¹Es relativamente fácil obtener el cierre de K , que denotaremos $O(K)$. Con una mera inspección de dicha base K , observamos si aparecen $P(c, a)$ ó $P(c, b)$, entonces se añade a K la fórmula $O(a, b)$.

- *reconoce fronteras si para $a, b \in L_K$ tales que $\Theta \models P(a, b)$*

$$\Theta \models \text{TPP}(a, b) \implies \exists c \in L_K [\{C(c, a), C(c, b)\} \subseteq K \wedge \{O(c, a), O(c, b)\} \cap K = \emptyset]$$

En general, una base de datos no es refinada. Hay que notar que cuando una base de datos K es refinada, la función de Skolem f_0 es interpretable en el propio $\mathbb{W}(K)$. Por tanto, podemos añadir a la base de datos un conjunto de axiomas con propiedades básicas de tal función, y f_0 puede *definirse sintácticamente* en K , simplificarse por nivel de compacidad, si es posible. Si no es refinada, la definición parcial también es útil. Esto se podrá comprobar en los experimentos cuyos resultados reflejan las tablas 8.4.1 y 8.4.1.

La definición anterior muestra una interpretación práctica en alguna BC de las relaciones en RCC. De hecho, tenemos el siguiente teorema [2].

Teorema. 8.3.2 *Si K tiene las cinco propiedades anteriores, entonces $\Theta_G \upharpoonright_{\mathbb{W}(K)} \models \text{RCC}(K)$.*

Demostración: En primer lugar hay que observar que, como $\Theta_G \upharpoonright_{\mathbb{W}(K)} \subset \Theta$ (como subestructura), se sigue que toda fórmula universal válida en Θ es una fórmula válida también en la subestructura $\Theta_G \upharpoonright_{\mathbb{W}(K)}$. Además, $\text{RCC}(K)$ está formado por los axiomas de RCC junto con el conjunto de fórmulas (universales) de K . Por tanto, basta probar que $\Theta_G \upharpoonright_{\mathbb{W}(K)} \models \text{RCC}$. Necesitamos probar únicamente que $\Theta_G \upharpoonright_{\mathbb{W}(K)} \models A_P + A_O + A_{\text{TPP}} + A_{\text{NTPP}}$ ya que los otros axiomas de RCC son fórmulas universales.

1. $\Theta_G \upharpoonright_{\mathbb{W}(K)} \models A_P$:

Como puede verse en la fig.1.10, $A_P : P(x, y) \leftrightarrow \forall z[C(z, x) \rightarrow C(z, y)]$. Como $P(x, y) \rightarrow \forall z[C(z, x) \rightarrow C(z, y)]$ es universal, bastará demostrar $\Theta_G \upharpoonright_{\mathbb{W}(K)} \models \forall z[C(z, x) \rightarrow C(z, y)] \rightarrow P(x, y)$.

Supongamos que $\Theta_G \upharpoonright_{\mathbb{W}(K)} \models \neg P(a, b)$ se tiene para ciertos $a, b \in L_K$.

Como $\neg P(a, b)$ es una fórmula abierta, tenemos $\Theta \models \neg P(a, b)$. De $\Theta \models K$ se sigue que $P(a, b) \notin K$.

$$\begin{aligned} & \left\{ \begin{array}{l} K \text{ es extensional} \\ P(a, b) \notin K \end{array} \right. \implies \exists c \in L_K : \left\{ \begin{array}{l} C(a, b) \in K \\ \wedge \\ C(c, b) \notin K \end{array} \right. \\ \implies & \exists c \in L_K : \left\{ \begin{array}{l} \Theta \models C(c, a), \text{ porque } \Theta \models K \\ \wedge \\ \Theta \models \neg C(c, b), \text{ porque } K \text{ is } C\text{-completo} \end{array} \right. \\ \implies & \Theta_G \upharpoonright_{\mathbb{W}(K)} \models \exists c \in L_K [C(c, a) \wedge \neg C(c, b)] \text{ (de } \Theta_G \upharpoonright_{\mathbb{W}(K)} \subset \Theta \text{)}. \end{aligned}$$

$$\text{ASERTO I: } \Theta_G \upharpoonright_{W(K)} \models P(a, b) \iff P(a, b) \in K$$

Prueba:

- Supongamos que $P(a, b) \notin K$. Como en la demostración de $\Theta_G \upharpoonright_{W(K)} \models A_P$, concluimos que $\Theta_G \upharpoonright_{W(K)} \models \exists c \in L_K [C(c, a) \wedge \neg C(c, b)]$ y, como $\Theta_G \upharpoonright_{W(K)} \models A_P$, tenemos $\Theta_G \upharpoonright_{W(K)} \models \neg P(a, b)$.
- $P(a, b) \in K \xrightarrow{\Theta \models K} \Theta \models P(a, b) \xrightarrow{\Theta_G \upharpoonright_{W(K)} \subset \Theta} \Theta_G \upharpoonright_{W(K)} \models P(a, b)$.

2. $\Theta_G \upharpoonright_{W(K)} \models A_0$:

El axioma A_0 establece que $0(x, y) \leftrightarrow \exists z[P(z, x) \wedge P(z, y)]$. Como antes, tenemos que $\exists z[P(z, x) \wedge P(z, y)] \rightarrow 0(x, y)$ es una fórmula universal, por tanto sólo falta probar que $\Theta_G \upharpoonright_{W(K)} \models 0(x, y) \rightarrow \exists z[P(z, x) \wedge P(z, y)]$.

Supongamos que se tiene $\Theta_G \upharpoonright_{W(K)} \models 0(a, b)$ para ciertos $a, b \in L_K$. Como $\Theta_G \upharpoonright_{W(K)} \subset \Theta$, tenemos $\Theta \models 0(a, b)$. Además,

$$K \text{ refinada} \implies \exists c \in L_K : \begin{cases} P(c, a) \in K \\ \wedge \\ P(c, b) \in K \end{cases}$$

$$\xrightarrow{(\text{Aserto I})} \exists c \in L_K : \begin{cases} \Theta_G \upharpoonright_{W(K)} \models P(c, a) \\ \wedge \\ \Theta_G \upharpoonright_{W(K)} \models P(c, b) \end{cases}$$

$$\text{ASERTO II: } 0(a, b) \in K \iff \Theta_G \upharpoonright_{W(K)} \models 0(a, b)$$

Prueba:

- $\Theta_G \upharpoonright_{W(K)} \models 0(a, b) \implies \Theta \models 0(a, b) \xrightarrow{K \text{ es refinada}} 0(a, b) \in K$.
- $0(a, b) \in K \xrightarrow{\Theta \models K} \Theta \models 0(a, b) \xrightarrow{\Theta_G \upharpoonright_{W(K)} \subset \Theta} \Theta_G \upharpoonright_{W(K)} \models 0(a, b)$.

3. $(\Theta_G \upharpoonright_{W(K)} \models A_{\text{TPP}})$

Por el axioma A_{TPP} tenemos $\text{TPP}(x, y) \leftrightarrow \text{PP}(x, y) \wedge \exists z[\text{EC}(z, x) \wedge \text{EC}(z, y)]$. Como $\text{PP}(x, y) \wedge \exists z[\text{EC}(z, x) \wedge \text{EC}(z, y)] \rightarrow \text{TPP}(x, y)$ es una fórmula universal, basta probar que $\text{TPP}(x, y) \rightarrow \text{PP}(x, y) \wedge \exists z[\text{EC}(z, x) \wedge \text{EC}(z, y)]$.

Supongamos que $\Theta_G \upharpoonright_{W(K)} \models \text{TPP}(a, b)$ para ciertos $a, b \in L_K$. Como $\Theta_G \upharpoonright_{W(K)} \subset \Theta$, tenemos $\Theta \models \text{TPP}(a, b)$. De aquí,

$$\begin{cases} \Theta \models \text{PP}(a, b), \text{ porque } \Theta \models A_{\text{TPP}} \\ \Theta \models P(a, b), \text{ porque } \Theta \models A_{\text{PP}} \end{cases} \xrightarrow{(\Theta_G \upharpoonright_{W(K)} \subset \Theta)} \begin{cases} \Theta_G \upharpoonright_{W(K)} \models \text{PP}(a, b) (*) \\ \Theta \models P(a, b) \end{cases}$$

Por tanto:

$$\begin{array}{l}
 \left. \begin{array}{l} \Theta \models P(a, b) \\ K \text{ reconoce fronteras} \\ \Theta \models TPP(a, b) \end{array} \right\} \\
 \\
 \Rightarrow \exists c \in L_K : \left\{ \begin{array}{l} C(c, a) \in K \\ C(c, b) \in K \\ O(c, a) \notin K \\ O(c, b) \notin K \end{array} \right\} \xrightarrow{\Theta \models K} \left\{ \begin{array}{l} \Theta \models C(c, a) \\ \Theta \models C(c, b) \\ \Theta_G \upharpoonright_{W(K)} \not\models O(c, a) \\ \Theta_G \upharpoonright_{W(K)} \not\models O(c, b) \end{array} \right\} \\
 \\
 \xrightarrow{(\Theta_G \upharpoonright_{W(K)} \subset \Theta)} \exists c \in L_K : \left\{ \begin{array}{l} \Theta_G \upharpoonright_{W(K)} \models C(c, a) \wedge \neg O(c, a) \\ \Theta_G \upharpoonright_{W(K)} \models C(c, b) \wedge \neg O(c, b) \end{array} \right\} \\
 \\
 \xrightarrow{(\Theta_G \upharpoonright_{W(K)} \models A_{EC})} \exists c \in L_K : \left\{ \begin{array}{l} \Theta_G \upharpoonright_{W(K)} \models EC(c, a) \\ \Theta_G \upharpoonright_{W(K)} \models EC(c, b) \end{array} \right\} \quad (**)
 \end{array}$$

Y (*), (**) demuestran el resultado.

4. $\Theta_G \upharpoonright_{W(K)} \models A_{NTPP}$, como $\Theta \models NTPP(x, y) \leftrightarrow PP(x, y) \wedge \neg TPP(x, y)$ y es una fórmula universal. ■

8.3.2. Grado de credibilidad e interpretación espacial

Teorema. 8.3.3 Sea $\langle \Pi, \phi \rangle$ un 0-argumento de $RCC(K)$. Si $gr(\langle \Pi, \phi \rangle) = 1$ entonces $\langle \Pi, \phi \rangle \in A^0 \exists (RCC(K))$ y $\Gamma(\Pi) \models \Pi + \phi$.

Demostración: Supongamos que $gr(\langle \Pi, \phi \rangle) = 1$. Por definición, el cardinal del conjunto $\{\eta \in Prueba^0(\Pi, \phi) : \eta \text{ tiene ruido de Skolem}\} = 0$

\Rightarrow no hay fórmulas de Skolem en la prueba de $\Pi \vdash^0 \phi$

\Rightarrow los axiomas de RCC que aparecen en la prueba de $\Pi \vdash^0 \phi$ son fórmulas abiertas, en particular, Π .

Como $\Theta \models RCC + K$ y $RCC + K \vdash \Pi$, tenemos $\Theta \models \Pi$.

Pero $\Pi \vdash \phi$ al ser $\langle \Pi, \phi \rangle$ un argumento.

Por tanto se sigue que $\Theta \models \Pi + \phi$.

$\Pi + \phi$ son fórmulas abiertas y $\Gamma(\Pi) \subset \Theta$, así $\Gamma(\Pi) \models \Pi + \phi$. ■

Corolario. 8.3.4 Si $gr(\langle \Pi, \phi \rangle) = gr(\langle \Pi', \phi' \rangle) = 1$ y el primer argumento es un corte del segundo, entonces

1. $\Gamma(\Pi) \not\subseteq \Gamma(\Pi')$ (por tanto no hay un corte local con grado de credibilidad 1).

2. Si $\langle \Pi, \phi \rangle$ es un corte canónico (maximal conservativo), esto es, $\phi \equiv \neg(\phi_1 \wedge \dots \wedge \phi_n)$ con $\phi_i \in \Pi'$, entonces $\Gamma(\Pi') \subsetneq \Gamma(\Pi)$.

Demostración:

1. Como $\langle \Pi, \phi \rangle$ es un corte de $\langle \Pi', \phi' \rangle$ tenemos $\phi \equiv \neg(\phi_1 \wedge \dots \wedge \phi_n)$ donde $\phi_1 \dots \phi_n \subseteq \Pi'$. Supongamos que el resultado a probar es falso. En ese caso $\Gamma(\Pi) \subseteq \Gamma(\Pi')$. Sabemos que $\Gamma(\Pi) \models \Pi + \phi$, $\Gamma(\Pi') \models \phi'$ y $\phi_1 \wedge \dots \wedge \phi_n$ son fórmulas abiertas. Por el teorema 8.3.3 podemos concluir que $\Gamma(\Pi') \models \phi_1 \wedge \dots \wedge \phi_n$, $\Gamma(\Pi') \models \neg(\phi_1 \wedge \dots \wedge \phi_n)$, y $\Gamma(\Pi')$ es consistente, una contradicción.
2. Ya que $\langle \Pi, \phi \rangle$ es un corte canónico, $\phi \equiv \neg(\phi_1 \wedge \dots \wedge \phi_n)$, donde $\phi_1 \dots \phi_n$ es la enumeración canónica de Π' , y $\Pi \vdash \neg(\phi_1 \wedge \dots \wedge \phi_n)$, se sigue que $\Gamma(\Pi') \subseteq \Gamma(\Pi)$. Por el punto anterior, tenemos $\Gamma(\Pi') \subsetneq \Gamma(\Pi)$.

■

8.3.3. Nivel de compacidad

Un parámetro útil del grafo Θ_G de un modelo topológico Θ es el *nivel de compacidad*.

Definición. 8.3.5 *El nivel de compacidad de Θ es el menor $n > 0$ tal que la intersección de cualquier conjunto de regiones de Θ_G es igual a la intersección de n de ellas.*

En el caso de RCC, las funciones de Skolem proceden de los axiomas existenciales A_P , A_0 , A_{TPP} y A_{NTPP} . Es posible dar una interpretación parcial de tales funciones. Recordemos el ejemplo 6.5.7 que aparece en capítulo 6 de esta memoria. En él mencionamos la función de Skolem para A_0 , que denotamos por $f_0(x, y)$: dicha función representa la intersección de x e y , pues describe la región cuya existencia afirma la definición de solapamiento y es parte de x e y simultáneamente.

Esta idea nos permite eliminar resultados que no son útiles mediante una axiomatización parcial de la intersección (ver fig.8.3). Igualmente, se pueden axiomatizar niveles de compacidad superiores (fig. 8.4).

8.4. Un caso de estudio

Reflejamos aquí una serie de experimentos realizados con el servicio de limpieza, trabajando en una base de datos asociada a RCC. Aunque el sub-

$$\begin{array}{l}
\text{Int}(x, x) = x \\
P(x, y) \rightarrow \text{Int}(x, y) = x \\
O(x, y) \rightarrow \text{Int}(x, y) = \text{Int}(y, x) \\
O(y, z) \wedge O(x, \text{Int}(y, z)) \rightarrow \text{Int}(x, \text{Int}(y, z)) = \text{Int}(\text{Int}(x, y), z) \\
O(y, z) \wedge O(x, \text{Int}(y, z)) \rightarrow \begin{cases} \text{Int}(x, \text{Int}(y, z)) = \text{Int}(y, z) \vee \\ \text{Int}(x, \text{Int}(y, z)) = \text{Int}(x, y) \vee \\ \text{Int}(x, \text{Int}(y, z)) = \text{Int}(x, z) \end{cases}
\end{array}$$

Figura 8.3: Una axiomatización de f_0 (como Int) en nivel de compacidad 2

$$\begin{array}{l}
(1) \text{Int}(x, x) = x \\
(2) P(x, y) \rightarrow \text{Int}(x, y) = x \\
(3) O(x, y) \rightarrow \text{Int}(x, y) = \text{Int}(y, x) \\
(4) O(y, z) \wedge O(x, \text{Int}(y, z)) \rightarrow \text{Int}(x, \text{Int}(y, z)) = \text{Int}(\text{Int}(x, y), z) \\
\begin{cases} \text{Int}^{(1)}(x) := x \\ \text{Int}^{(n+1)}(x_1, \dots, x_{n+1}) := \text{Int}(\text{Int}^{(n)}(x_1, \dots, x_n), x_{n+1}) \end{cases} \\
O^{(n)}(x_1, \dots, x_n) = \bigwedge_{i=1}^{n-1} O(\text{Int}^{(i)}(x_1, \dots, x_i), x_{i+1}) \\
\text{Nivel} = k : \\
(1), (2), (3), (4) + O^{(n)}(x_1, \dots, x_n) \\
\rightarrow \left(\bigvee_{\{b_1, \dots, b_k\} \in \{x_1, \dots, x_n\}^k} \text{Int}^{(n)}(x_1, \dots, x_n) \right) = \text{Int}^{(k)}(b_1, \dots, b_k)
\end{array}$$

Figura 8.4: Axiomatización de f_0 (nivel de compacidad = k)

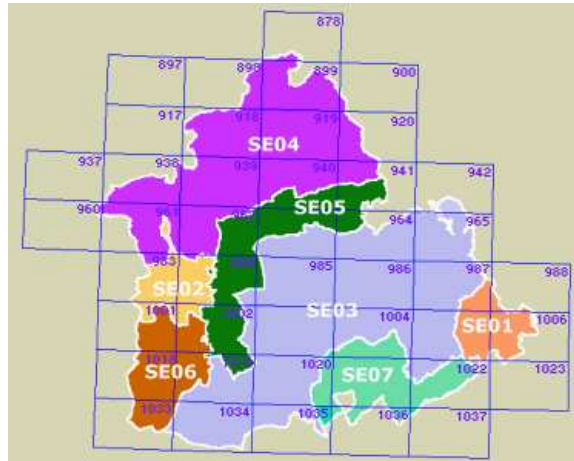


Figura 8.5: Mapa provincial

```

Connect(SE04, SE05)
Connect(SE04, Map - 941)
Overlaps(Map - 920, SE04)
Part - of(SE04, SEVILLA)
⋮

```

Figura 8.6: Algunos hechos de la base de datos espacial de la fig.8.5

lenguaje relacional de RCC se ha estudiado extensamente en el campo de los Sistemas de Información Geográfica [6] para tratar configuraciones espaciales, la teoría resulta computacionalmente inaceptable [34].

El caso de estudio reproduce la depuración de una base de datos espacial, que suponemos escrita en texto estructurado. Concretamente, trabajaremos con una base de datos sobre las relaciones entre tres tipos de regiones: provincias, comarcas, y zonas cartográficas de Andalucía (ver fig. 8.5). El sistema trabaja con una base de datos construida en base a las relaciones de conexión (*Connect*), intersección no vacía o solapamiento (*Overlaps*) y parte-de como (*Part-of*) (ver fig. 8.6). Por tanto existe una gran cantidad de información oculta, de conocimiento relativo a otras relaciones topológicas entre regiones que no aparecen de forma explícita en la base de datos, que el DAT debe deducir (y, eventualmente, añadir a la base de datos).

El universo de trabajo está formado por 258 regiones correspondientes a las ocho provincias andaluzas, las comarcas (que aparecen listadas junto a los mapas de cada provincia) y las zonas cartográficas. Las zonas cartográficas son las celdas que dividen el mapa en cuadrículas y se nombran con la letra **Z** seguida de los números 807 al 1078. Se producen algunos saltos en la numeración debido a que, al estar extraído de un mapa de España, faltan las zonas que no

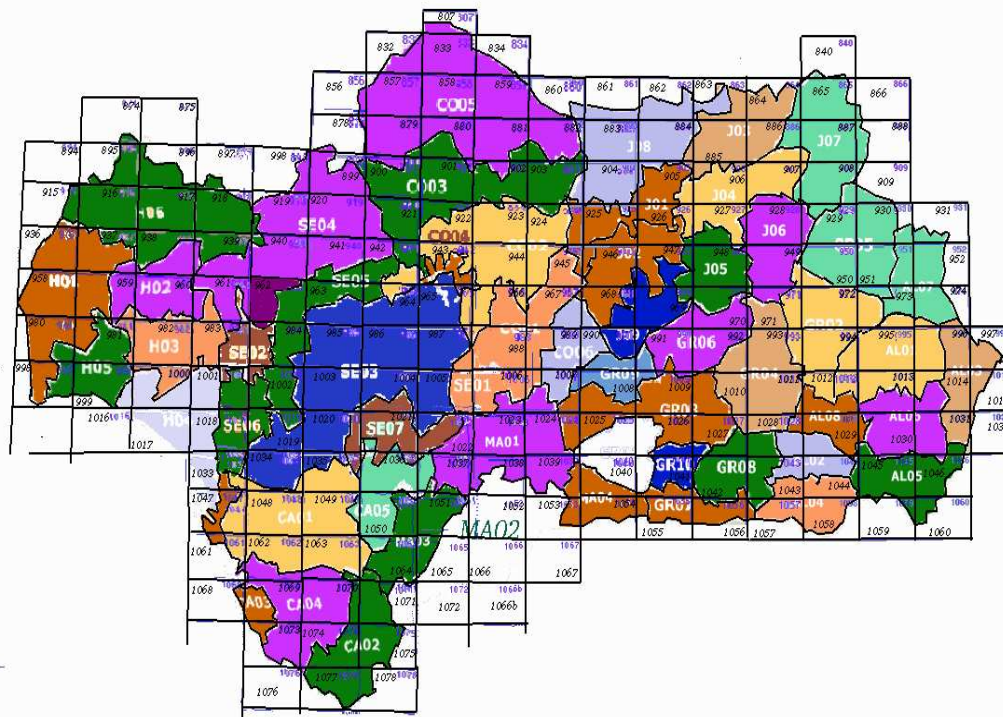


Figura 8.7: Mapa de Andalucía

están conectadas con Andalucía.

A continuación presentamos los mapas de las provincias por orden alfabético. Los mapas contienen las comarcas codificadas según la lista que aparece al pie de cada uno. Además pueden observarse en ellos las zonas cartográficas que los cubren, con su numeración correspondiente. Ver figuras 8.8, 8.9, 8.10, 8.11, 8.12, 8.13, 8.14 y 8.15.

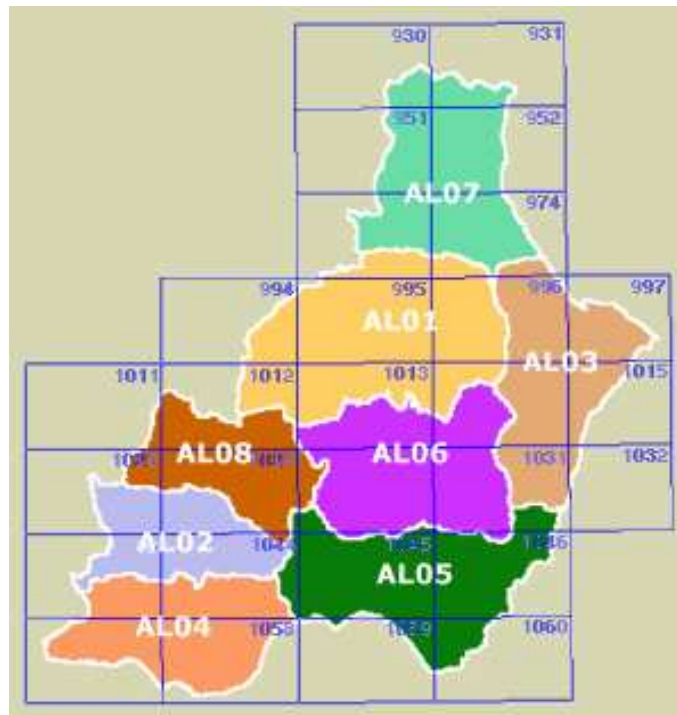


Figura 8.8: La provincia de Almería

COMARCAS:

AL01: ALTO_ALMANZORA.

AL02: ALTO_ANDARAX .

AL03: BAJO_ALMANZORA .

AL04: CAMPO_DALIAS.

AL05: CAMPO_NIJAR.

AL06: CAMPO_TABERNAS.

AL07: LOS_VELEZ.

AL08: RIO_NACIMIENTO.

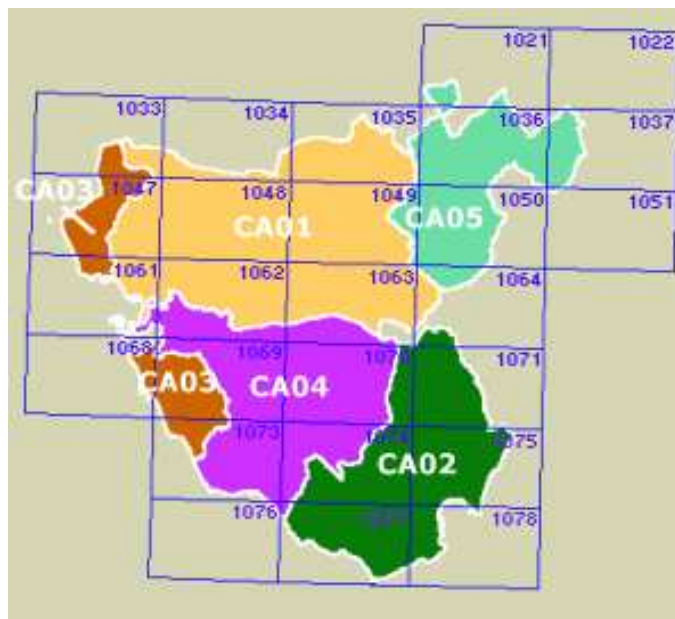


Figura 8.9: La provincia de Cádiz

COMARCAS:

CA01: CAMPINA_DE_CADIZ.

CA02: CAMPO_GIBRALTAR.

CA03: COSTA_NW_CADIZ.

CA04: DE_LA_JANDA.

CA05: SIERRA_DE_CADIZ.

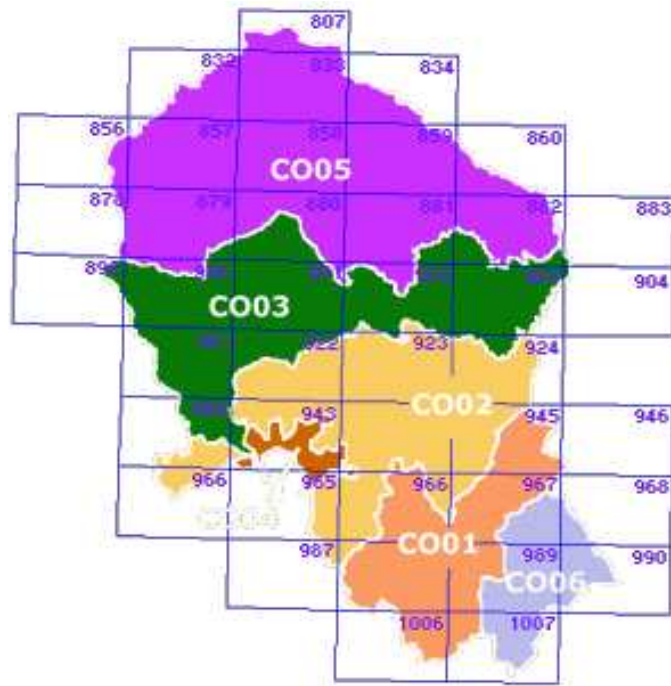


Figura 8.10: La provincia de Córdoba

COMARCAS:

CO01: CAMPINA_ALTA.

CO02: CAMPINA_BAJA.

CO03: LA_SIERRA.

CO04: LAS_COLONIAS.

CO05: PEDROCHES.

CO06: PENIBETICA.

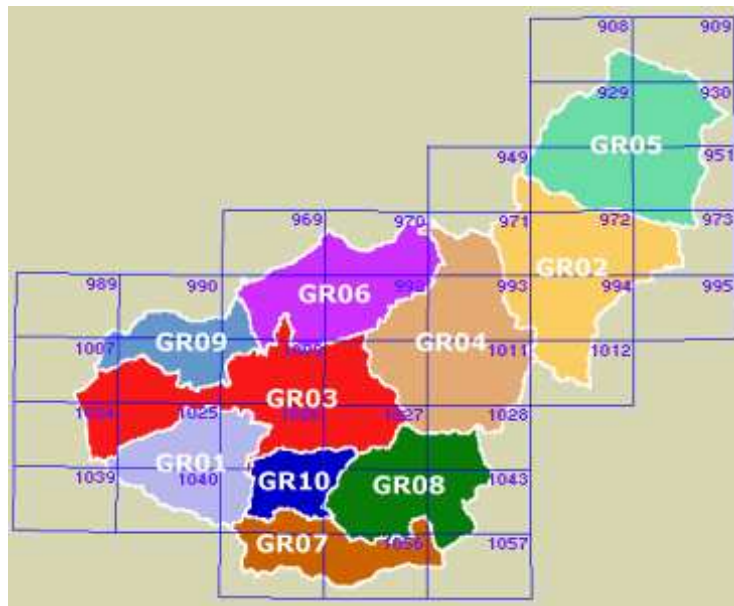


Figura 8.11: La provincia de Granada

COMARCAS:

GR01: ALHAMA.

GR02: BAZA.

GR03: DE_LA_VEGA.

GR04: GUADIX.

GR05: HUESCAR.

GR06: IZNALLOZ.

GR07: LA_COSTA.

GR08: LAS_ALPUJARRAS.

GR09: MONTEFRIO.

GR10: VALLE_DE_LECRIN.

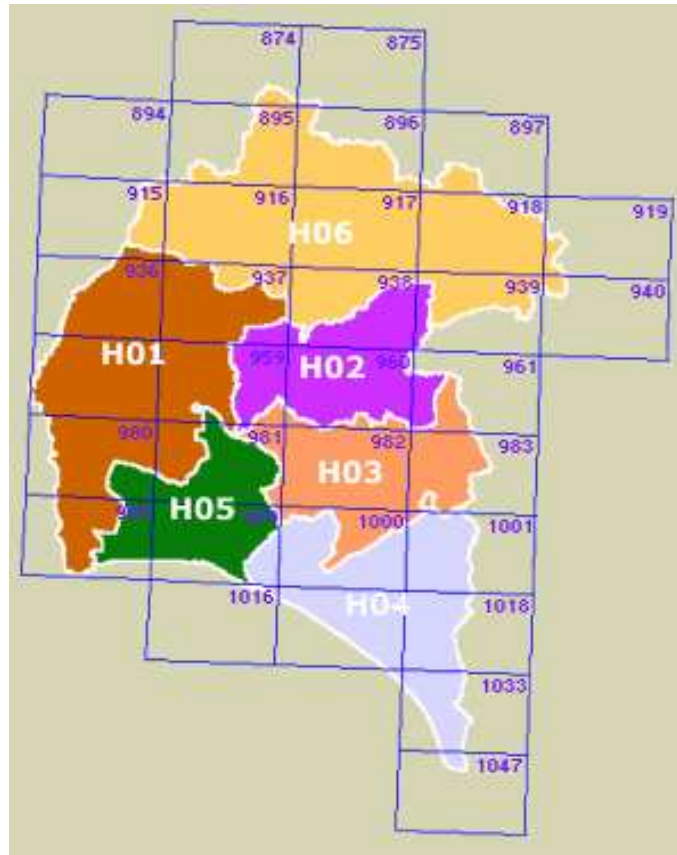


Figura 8.12: La provincia de Huelva

COMARCAS:

H01: ANDEVALO_OCCIDENTAL.

H02: ANDEVALO_ORIENTAL.

H03: CONDADO_CAMPINA.

H04: CONDADO_LITORAL.

H05: COSTA.

H06: SIERRA.

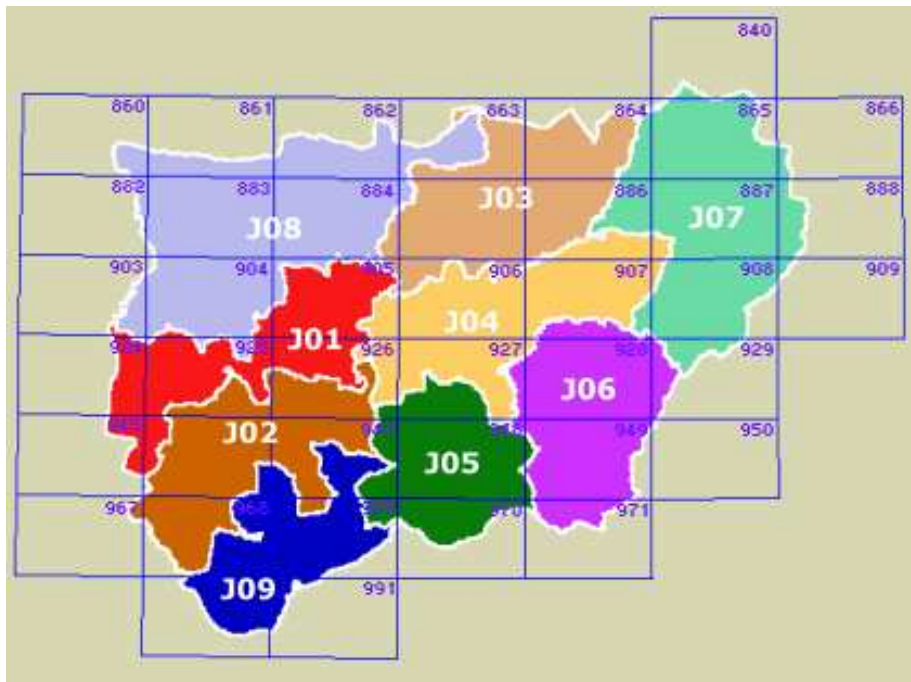


Figura 8.13: La provincia de Jaén

COMARCAS:

J01: CAMPINA_NORTE.

J02: CAMPINA_SUR.

J03: EL_CONDADO.

J04: LA_LOMA.

J05: MAGINA.

J06: SIERRA_DE_CAZORLA.

J07: SIERRA_DE_SEGURA.

J08: SIERRA_MORENA.

J09: SIERRA_SUR_JAEN.

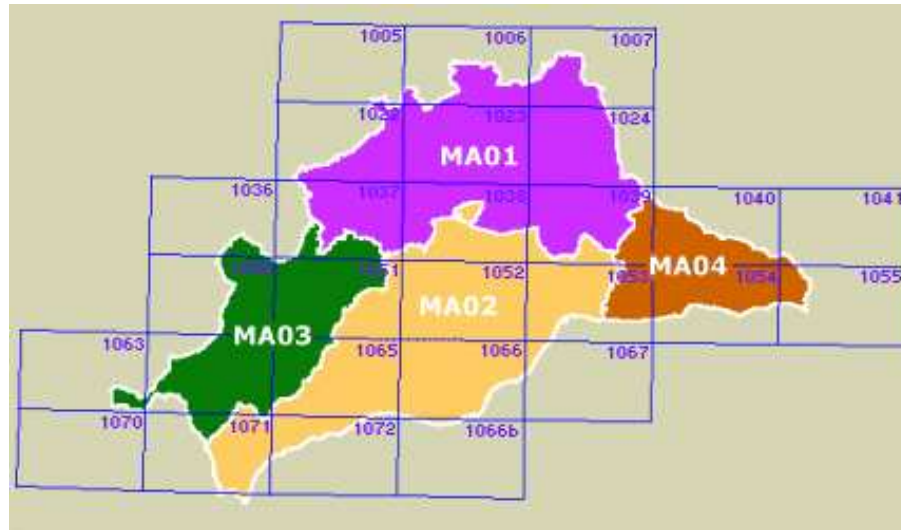


Figura 8.14: La provincia de Málaga

COMARCAS:

MA01: ANTEQUERA.

MA02: GUADALORCE.

MA03: SERRANIA_RONDA.

MA04: VELEZ_MALAGA.

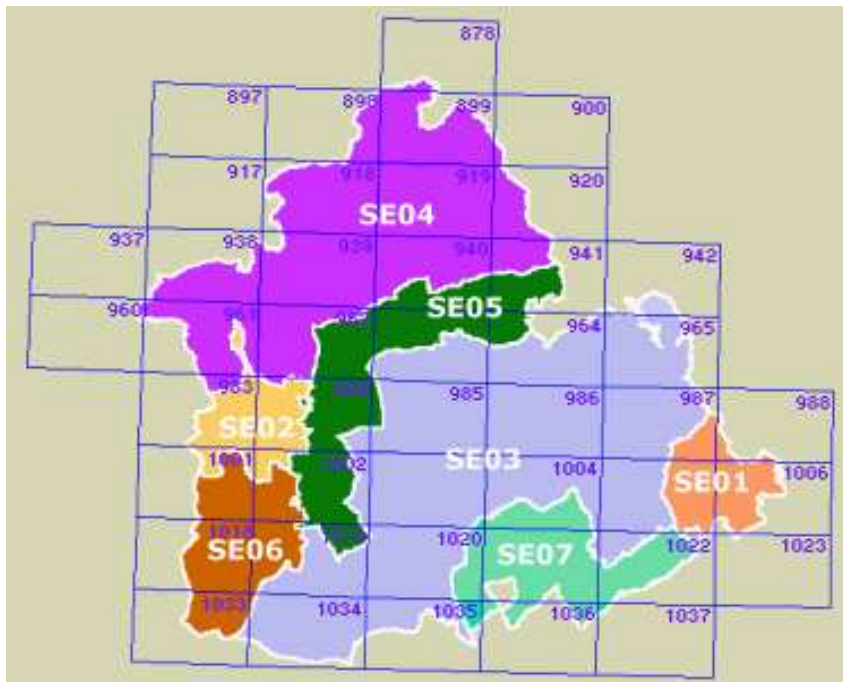


Figura 8.15: La provincia de Sevilla

COMARCAS:

SE01: DE_ESTEPA.

SE02: EL_ALJARAFE.

SE03: LA_CAMPINA.

SE04: SIERRA_NORTE.

SE05: LA_VEGA.

SE06: LAS_MARISMAS.

SE07: SIERRA_SUR_SEVILLA.

La base de conocimiento constará de la siguiente información:

■ La teoría RCC

```

all x (C(x,x)).
all x y (C(x,y) <-> C(y,x)).
all x y (-C(x,y) <-> DC(x,y)).
all x y ((all z (C(z,x) -> C(z,y))) <-> P(x,y)).
all x y (P(x,y) & -P(y,x) <-> PP(x,y)).
all x y (P(x,y) & P(y,x) <-> EQ(x,y)).
all x y ((exists z (P(z,x) & P(z,y))) <-> O(x,y)).
all x y (-O(x,y) <-> DR(x,y)).
all x y (O(x,y) & -P(x,y) & -P(y,x) <-> PO(x,y)).
all x y (C(x,y) & -O(x,y) <-> EC(x,y)).
all x y (PP(x,y) & (exists z (EC(z,x) & EC(z,y)))<->TPP(x,y)).
all x y (PP(x,y) & -(exists z (EC(z,y) & EC(z,y)))<->NTPP(x,y)).

```

■ Teoremas de la tabla de composición de RCC: El contenido de la tabla se introduce para que sirva de álgebra de relaciones en la búsqueda de respuestas a las preguntas realizadas a la BC. De esta forma algunas de las inferencias del sistema son más cortas.

```

all x y z (DC(x,y) & EC(y,z) -> DR(x,z) | PO(x,z) | PP(x,z)).
all x y z (DC(x,y) & PO(y,z) -> DR(x,z) | PO(x,z) | PP(x,z)).
all x y z (DC(x,y) & TPP(y,z) -> DR(x,z) | PO(x,z) | PP(x,z)).
all x y z (DC(x,y) & NTPP(y,z) -> DR(x,z) | PO(x,z) | PP(x,z)).
all x y z (DC(x,y) & TPP(z,y) -> DC(x,z)).
all x y z (DC(x,y) & NTPP(z,y) -> DC(x,z)).
all x y z (DC(x,y) & EQ(y,z) -> DC(x,z)).
all x y z (EC(x,y) & DC(y,z) -> DR(x,z) | PO(x,z) | PP(z,x)).
all x y z (EC(x,y) & EC(y,z) -> DR(x,z) | PO(x,z) | TPP(x,z)
| TPP(z,x) | EQ(x,z)).
all x y z (EC(x,y) & PO(y,z) -> DR(x,z) | PO(x,z) | PP(x,z)).
all x y z (EC(x,y) & TPP(y,z) -> EC(x,z) | PO(x,z) | PP(x,z)).
all x y z (EC(x,y) & NTPP(y,z) -> PO(x,z) | PP(x,z)).
all x y z (EC(x,y) & TPP(z,y) -> DR(x,z)).
all x y z (EC(x,y) & NTPP(z,y) -> DC(x,z)).
all x y z (EC(x,y) & EQ(y,z) -> EC(x,z)).
all x y z (PO(x,y) & DC(y,z) -> DR(x,z) | PO(x,z) | PP(z,x)).
all x y z (PO(x,y) & EC(y,z) -> DR(x,z) | PO(x,z) | PP(z,x)).
all x y z (PO(x,y) & TPP(y,z) -> PO(x,z) | PP(x,z)).
all x y z (PO(x,y) & NTPP(y,z) -> PO(x,z) | PP(x,z)).
all x y z (PO(x,y) & TPP(z,y) -> DR(x,z) | PO(x,z) | PP(z,x)).
all x y z (PO(x,y) & NTPP(z,y) -> DR(x,z) | PO(x,z) | PP(z,x)).
all x y z (PO(x,y) & EQ(y,z) -> PO(x,z)).
all x y z (TPP(x,y) & DC(y,z) -> DC(x,z)).
all x y z (TPP(x,y) & EC(y,z) -> DR(x,z)).
all x y z (TPP(x,y) & PO(y,z) -> DR(x,z) | PO(x,z) | PP(x,z)).
all x y z (TPP(x,y) & TPP(y,z) -> PP(x,z)).

```

```

all x y z (TPP(x,y) & NTPP(y,z) -> NTPP(x,z)).
all x y z (TPP(x,y) & TPP(z,y) -> DR(x,z) | PO(x,z) | TPP(x,z)
          | TPP(z,x) | EQ(x,z)).
all x y z (TPP(x,y) & NTPP(z,y) -> DR(x,z) | PO(x,z) | PP(z,x)).
all x y z (TPP(x,y) & EQ(y,z) -> TPP(x,z)).
all x y z (NTPP(x,y) & DC(y,z) -> DC(x,z)).
all x y z (NTPP(x,y) & EC(y,z) -> DC(x,z)).
all x y z (NTPP(x,y) & PO(y,z) -> DR(x,z) | PO(x,z) | PP(x,z)).
all x y z (NTPP(x,y) & TPP(y,z) -> NTPP(x,z)).
all x y z (NTPP(x,y) & NTPP(y,z) -> NTPP(x,z)).
all x y z (NTPP(x,y) & TPP(z,y) -> DR(x,z) | PO(x,z) | PP(x,z)).
all x y z (NTPP(x,y) & EQ(y,z) -> NTPP(x,z)).
all x y z (TPP(y,x) & DC(y,z) -> DR(x,z) | PO(x,z) | PP(z,x)).
all x y z (TPP(y,x) & EC(y,z) -> EC(x,z) | PO(x,z) | PP(z,x)).
all x y z (TPP(y,x) & PO(y,z) -> PO(x,z) | PP(z,x)).
all x y z (TPP(y,x) & TPP(y,z) -> PO(x,z) | TPP(x,z) | TPP(z,x)
          | EQ(x,z)).
all x y z (TPP(y,x) & NTPP(y,z) -> PO(x,z) | PP(x,z)).
all x y z (TPP(y,x) & TPP(z,y) -> PP(z,x)).
all x y z (TPP(y,x) & NTPP(z,y) -> NTPP(z,x)).
all x y z (TPP(y,x) & EQ(y,z) -> TPP(z,x)).
all x y z (NTPP(y,x) & DC(y,z) -> DR(x,z) | PO(x,z) | PP(z,x)).
all x y z (NTPP(y,x) & EC(y,z) -> PO(x,z) | PP(z,x)).
all x y z (NTPP(y,x) & PO(y,z) -> PO(x,z) | PP(z,x)).
all x y z (NTPP(y,x) & TPP(y,z) -> PO(x,z) | PP(z,x)).
all x y z (NTPP(y,x) & NTPP(y,z) -> O(x,z)).
...

```

- Otros teoremas (que resultan de la propia estructura de retículo de RCC) y que son consecuencia del teorema 2.2.1.

```

all x y (P(x,y) -> C(x,y))
all x y (P(x,y) -> O(x,y))
all x y (O(x,y) -> C(x,y))
all x y (O(x,y) <-> O(y,x))

```

- Axiomas de nombres únicos: Cada nombre corresponde biunivocamente a una región.

```

ALMERIA!=CADIZ.
ALMERIA!=CORDOBA.
ALMERIA!=GRANADA.
ALMERIA!=HUELVA.
ALMERIA!=JAEN.
ALMERIA!=MALAGA.
.....
%
```

```

%
CADIZ!=CORDOBA.
CADIZ!=GRANADA.
CADIZ!=HUELVA.
CADIZ!=JAEN.
CADIZ!=MALAGA.
.....
%
%
Z807!=Z832.
Z807!=Z833.
Z807!=Z834.
Z807!=Z840.
Z807!=Z856.
.....
%
%
Z1076!=Z1077.
Z1076!=Z1078.
%
%
Z1077!=Z1078.

```

- **Axiomas de clausura de dominio:** Este axioma determina el conjunto exacto de constantes que contiene nuestra base de datos. Para que el demostrador pueda trabajar con este axioma, que contendría en 258 literales (el número de regiones a tratar), se agrupan las fórmulas de manera lógica, por provincias, comarcas y zonas del tipo (Z8...), (Z9...) y (Z10...)².

```

all x (Provincia(x) | Comarca(x) | Zona8(x) | Zona9(x) | Zona10(x)).

all x (Provincia(x) <-> (x = ALMERIA
                        | (x = CADIZ) | ... | (x = SEVILLA))).

all x (Comarca(x) <-> (x = ALTO_ALMANZORA
                      | ... | (x = LAS_MARISMAS)
                      | (x = SIERRA_SUR_SEVILLA))).

all x (Zona8(x) <-> (x = Z807) | ... | (x = Z899)).

all x (Zona9(x) <-> (x = Z900) | ... | (x = Z999)).

all x (Zona10(x) <-> (x = Z1000) | ... | (x = Z1078)).

```

- **Axiomas de completitud:** Los agrupamos según la relación de conexión, parte y solapamiento.

²Esta forma de describir la clausura de dominio evita el desbordamiento de OTTER.

```

all x (C(x,ALMERIA) <-> x = ALMERIA | x = GRANADA | x = BAZA
    | x = GUADIX | x = HUESCAR
    | x = LA_COSTA | x = LAS_ALPUJARRAS | x = ALTO_ALMANZORA
    | x = ALTO_ANDARAX | x = BAJO_ALMANZORA
    | x = CAMPO_DALIAS | x = CAMPO_NIJAR | x = CAMPO_TABERNAS
    | x = LOS_VELEZ | x = RIO_NACIMIENTO | x = Z930 | x = Z931
    | x = Z951 | x = Z952 | x = Z973 | x = Z974 | x = Z994
    | x = Z995 | x = Z996 | x = Z997 | x = Z1011 | x = Z1012
    | x = Z1013 | x = Z1014 | x = Z1015 | x = Z1028 | x = Z1029
    | x = Z1030 | x = Z1031 | x = Z1032 | x = Z1043 | x = Z1044
    | x = Z1045 | x = Z1046 | x = Z1057 | x = Z1058 | x = Z1059
    | x = Z1060).

```

.....

```

all x (C(x,SEVILLA) <-> x = SEVILLA | x = CADIZ | x = CORDOBA
    | x = HUELVA | x = MALAGA | x = CAMPINA_DE_CADIZ
    | x = COSTA_NW_CADIZ | x = SIERRA_DE_CADIZ
    | x = CAMPINA_ALTA | x = CAMPINA_BAJA | x = LA_SIERRA
    | x = LAS_COLONIAS | x = PEDROCHES
    | x = ANDEVALO_OCCIDENTAL | x = ANDEVALO_ORIENTAL
    | x = CONDADO_CAMPINA | x = CONDADO_LITORAL | x = SIERRA
    | x = ANTEQUERA | x = DE_ESTEPA | x = EL_ALJARAFE
    | x = LA_CAMPINA | x = SIERRA_NORTE | x = LA_VEGA
    | x = LAS_MARISMAS | x = SIERRA_SUR_SEVILLA | x = Z878
    | x = Z898 | x = Z899 | x = Z900 | x = Z918 | x = Z919
    | x = Z920 | x = Z921 | x = Z938 | x = Z939 | x = Z940
    | x = Z941 | x = Z942 | x = Z943 | x = Z960 | x = Z961
    | x = Z962 | x = Z963 | x = Z964 | x = Z965 | x = Z983
    | x = Z984 | x = Z985 | x = Z986 | x = Z987 | x = Z988
    | x = Z1001 | x = Z1002 | x = Z1003 | x = Z1004
    | x = Z1005 | x = Z1006 | x = Z1018 | x = Z1019
    | x = Z1020 | x = Z1021 | x = Z1022 | x = Z1023
    | x = Z1033 | x = Z1034 | x = Z1035 | x = Z1036
    | x = Z1037 | x = Z1048).

```

.....

```

all x (C(x,CAMPO_NIJAR) <-> x = CAMPO_NIJAR | x = ALMERIA
    | x = ALTO_ANDARAX | x = BAJO_ALMANZORA | x = CAMPO_DALIAS
    | x = CAMPO_TABERNAS | x = RIO_NACIMIENTO | x = Z1029
    | x = Z1030 | x = Z1031 | x = Z1044 | x = Z1045
    | x = Z1046 | x = Z1058 | x = Z1059 | x = Z1060).

```

.....

```

all x (C(x,Z1078) <-> x = Z1078 | x = Z1074 | x = Z1075
    | x = Z1077 | x = CADIZ
    | x = CAMPO_GIBRALTAR).

```

```

all y (P(ALMERIA,y) <-> y = ALMERIA).
all y (P(CADIZ,y) <-> y = CADIZ).
all y (P(CORDOBA,y) <-> y = CORDOBA).

```



```

all y (P(GRANADA,y) <-> y = GRANADA) .

.....

all y (P(Z1047,y) <-> y = Z1047) .
all y (P(Z1048,y) <-> y = Z1048 | y = CADIZ
      | y = CAMPINA_DE_CADIZ) .
all y (P(Z1049,y) <-> y = Z1049 | y = CADIZ) .
all y (P(Z1050,y) <-> y = Z1050) .

.....

all y (P(Z1076,y) <-> y = Z1076) .
all y (P(Z1077,y) <-> y = Z1077) .
all y (P(Z1078,y) <-> y = Z1078) .

all y (O(ALMERIA,y) <-> y = ALMERIA | y = ALTO_ALMANZORA
      | y = ALTO_ANDARAX | y = BAJO_ALMANZORA
      | y = CAMPO_DALIAS | y = CAMPO_NIJAR | y = CAMPO_TABERNAS
      | y = LOS_VELEZ | y = RIO_NACIMIENTO | y = Z930 | y = Z931
      | y = Z951 | y = Z973 | y = Z974 | y = Z994 | y = Z995
      | y = Z996 | y = Z997 | y = Z1011 | y = Z1012 | y = Z1013
      | y = Z1014 | y = Z1015 | y = Z1028 | y = Z1029 | y = Z1030
      | y = Z1031 | y = Z1032 | y = ZZ043 | y = Z1044 | y = Z1045
      | y = Z1046 | y = Z1057 | y = Z1058 | y = Z1059 | y = Z1060) .

.....

all y (O(Z861,y) <-> y = Z861 | y = SIERRA_MORENA | y = JAEN) .
all y (O(Z862,y) <-> y = Z862 | y = SIERRA_MORENA | y = JAEN) .
all y (O(Z863,y) <-> y = Z863 | y = EL_CONDADO
      | y = SIERRA_MORENA | y = JAEN) .
all y (O(Z864,y) <-> y = Z864 | y = EL_CONDADO
      | y = SIERRA_DE_SEGURA | y = JAEN) .
all y (O(Z865,y) <-> y = Z865 | y = SIERRA_DE_SEGURA | y = JAEN) .

.....

all y (O(Z1076,y) <-> y = Z1076 | y = DE_LA_JANDA | y = CADIZ) .
all y (O(Z1077,y) <-> y = Z1077 | y = CAMPO_GIBRALTAR
      | y = DE_LA_JANDA | y = CADIZ) .
all y (O(Z1078,y) <-> y = Z1078 | y = CAMPO_GIBRALTAR
      | y = CADIZ) .

```

- **Axiomas de igualdad:** Los axiomas de igualdad son utilizados por OTTER sin necesidad de suministrarlos, salvo $x=x$.

La base de datos que resulta contiene 34000 hechos (incluida la formalización de primer orden), pero el número se puede reducir usando algunas características del demostrador. Los datos han sido introducidos a mano y, posiblemente, se han cometido errores. Éstos pueden clasificarse:

- A1 Errores de implementación en los datos que llevan a la inconsistencia. Corresponden a respuestas tipo \$F
- A2 En la respuesta obtenida aparece ruido de Skolem, regiones cuyo nombre no aparece de forma explícita en los axiomas de clausura de dominio
- A3 La respuesta aparece en forma disyuntiva, lo que la hace inútil
- A4 Errores que indican fallos en el dominio de conocimiento.

La base de datos es \mathcal{C} -completa pero no refinada. Por tanto, es muy posible que el demostrador detecte anomalías de tipo (A2), es decir, que aparezcan funciones de Skolem. Accidentalmente reconoce fronteras: su nivel de compacidad es 2 y se puede axiomatizar e incorporar a la teoría si usamos la interpretación espacial parcial de la función de Skolem como la *intersección parcial* (ver fig. 8.3).

Seleccionando diferentes tipos de argumentos es posible derivar conocimiento útil a partir de bases de datos inconsistentes [23]. Usando el literal de respuesta \$Ans de OTTER podemos determinar las respuestas asociadas a los argumentos en cada experimento (ver fig. 8.4.1).

No se pretende encontrar inconsistencias únicamente en el dominio de conocimiento (anomalías tipo A4 que corresponderían a fallos en los axiomas de la teoría RCC o en la tabla de composición). Suponemos que el dominio de conocimiento (la teoría RCC y, eventualmente, la tabla de composición para las ocho relaciones), es consistente, y que es muy posible que RCC junto con la base de datos sea inconsistente. Sin embargo, el paso (4) del proceso de limpieza fue necesario en unos de los experimentos, ya que el demostrador encontró un error en la tabla de composición para el cálculo RCC que se muestra en [15].

Se utilizó un ordenador con dos procesadores Pentium III (800 Mhz) y 256 Mb RAM. Lleva sistema operativo Red Hat Linux 7.0. El procesamiento de la base de datos consta de 40242 cláusulas, y se procesó en 6.5 segundos.

No es nuestro objetivo usar el demostrador como lenguaje de programación. La idea es realizar al sistema una serie de preguntas complejas que no se puedan resolver mediante algoritmos de satisfacción de restricciones o simples comandos de SQL. Las cuestiones van dirigidas a obtener conocimiento, acerca de las relaciones espaciales, que no esté explícito en la base de datos, (como `Proper-part` o combinaciones booleanas de relaciones espaciales complejas). Algunas de las cuestiones requieren un excesivo tiempo CPU. De manera sorprendente, el coste en tiempo está justificado; el demostrador *pensó* todo el tiempo sobre la base de datos y encontró muchos errores de implementación -del tipo A1- errores que no son aceptables. El número de resultados inútiles

debido al ruido de Skolem -del tipo A2- se puede reducir de manera significativa mediante una interpretación espacial de algunas funciones de skolem de la forma clausal del dominio de conocimiento. Esto podemos observarlo en los experimentos recogidos en las tablas 8.4.1 y 8.4.1.

Seleccionamos tres predicados que serán las etiquetas a estudiar en cada paso 3: *Part-of*, *Proper-part*, *Externally-connect*. Los resultados se recogen en tablas de manera que cada fila contiene la siguiente información:

- R1** Tiempo transcurrido y número de cláusulas generadas hasta el momento en que se obtiene la primera respuesta correcta a la pregunta
- R2** Muestra el número de resultados obtenidos y el tipo de anomalías encontradas 5 segundos más tarde
- R3** Tiempo, cláusulas generadas y resultados obtenidos en el momento en que se alcanza el primer resultado no válido
- R4** Datos para el primer error encontrado (en la base de datos o el dominio de conocimiento), si es el caso

Las columnas recogen los datos de los resultados obtenidos, según el tipo de error de que se trate.

Notas. 8.4.1 *Es importante reseñar que, aunque no hemos realizado un análisis completo de todos los argumentos obtenidos, los que producen inconsistencias tienen como premisas conjuntos minimalmente inconsistentes. Esto es debido a que (salvo en un experimento) la parte de la clausificación de $RCC(K)$ era consistente, y la contradicción era producida por los asertos.*

Para finalizar, hemos creído más conveniente que proporcionar los datos de todo el ciclo de limpieza sobre dicha base, ilustrar algunos de los pasos con experimentos donde se detallan algunos aspectos interesantes de los resultados obtenidos.

8.4.1. Fichas de experimentos

A continuación presentamos las fichas de experimentos realizados.

EXPERIMENTO 1:

BC: RCC + Tabla de composición + Ax. nombres únicos + Ax. clausura de dominio + Ax. completitud + Ax. igualdad

Pregunta: Regiones que son parte propia de Huelva.

Proper – part(x , HUELVA) \rightarrow \$Ans(x)

Resultado: Una respuesta obtenida a la pregunta es:

Length of proof is 4. Level of proof is 4.

```
----- PROOF -----
5[] -P(x, y) | P(y, x) | PP(x, y) .
8[] -P(x, y) | -P(y, x) | EQ(x, y) .
446[] P(ANDEVALO_ORIENTAL, x) | x! = HUELVA.
8946[] HUELVA! = ANDEVALO_ORIENTAL.
63366[] -PP(x, HUELVA) | $Ans(x) .
63371[] x = x.
69229[hyper, 63371, 446] P(ANDEVALO_OesRIENTAL, HUELVA) .
77764[hyper, 69229, 5, unit_del, 63366]
    P(HUELVA, ANDEVALO_ORIENTAL) | $Ans(ANDEVALO_ORIENTAL) .
83179[hyper, 77764, 8, 69229, flip.2] $Ans(ANDEVALO_ORIENTAL)
    | EQ(HUELVA, ANDEVALO_ORIENTAL) .
89721[para_from, 83179.1.1, 8946.1.1]
    ANDEVALO_ORIENTAL! = ANDEVALO_ORIENTAL
    | $Ans(ANDEVALO_ORIENTAL) .
89722[binary, 89721.1, 63371.1] $Ans(ANDEVALO_ORIENTAL) $ .
----- end of proof -----
```

Comentarios: Este experimento muestra un ejemplo de respuesta válida. La región ANDEVALO – ORIENTAL es parte propia de la región HUELVA. Véase la figura 8.12.

Las cláusulas

5[] – P(x , y) | P(y , x) | PP(x , y).

8[] – P(x , y) | – P(y , x) | EQ(x , y).

proceden de las definiciones de PP y EQ que aparecen en la axiomatización de RCC.

La cláusula

446[] P(ANDEVALO – ORIENTAL, x) | $x!$ = HUELVA.

procede de los axiomas de completitud para la relación P.

La siguiente,

8946[]HUELVA! = ANDEVALO – ORIENTAL.

aparece entre los axiomas de nombres únicos.

La clausificación de la pregunta es

63366[] – PP(x, HUELVA) | \$Ans(x).

y la última utilizada en la prueba es el axioma de igualdad.

Evidentemente, la pregunta no tiene respuesta única, la mostrada aquí es un mero ejemplo. Además es posible que una misma respuesta sea alcanzada mediante pruebas distintas, por lo que algunas aparecerán repetidas. También se da la circunstancia de que el demostrador deduce soluciones a la pregunta en forma de disyunción: no consideramos válidas este tipo de respuesta.

En ocasiones, el demostrador detecta inconsistencias en la base de conocimiento, de manera que genera muchas pruebas en las que alcanza la cláusula vacía: \$F.

La tabla de resultados es la siguiente:

Exp.	T. CPU(seg.)	c. gen.	res.	A1	A2	A3	A4
R1	2395.31	195222	1	113	0	0	0
R2	2400	201797	8	113	0	0	0
R3	2514.46	287088	14	117	0	1	0
R4	54.15	286	0	1	0	0	0

EXPERIMENTO 2:

BC: RCC + Tabla de composición + Ax. nombres únicos + Ax. clausura de dominio + Ax. completitud + Ax. igualdad

Pregunta: Regiones que son parte de Jaén.

$$P(x, \text{Jaen}) \rightarrow \text{\$Ans}(x)$$

Resultado: Una respuesta obtenida a la pregunta es:

Length of proof is 1. Level of proof is 1.

----- PROOF -----

```
715 [] P(Z948, x) | x!=JAEN.
64013 [] -P(x, JAEN) | \$Ans(x) .
64021 [] x=x.
64023 [ur, 64013, 715] \$Ans(Z948) | JAEN!=JAEN.
64024 [binary, 64023.1, 64021.1] \$Ans(Z948) $.
```

----- end of proof -----

Aquí una respuesta donde aparece la función de Skolem, para indicar la intersección entre la zona cartográfica Z991 y Jaén. Esa intersección es parte de Jaén y, por tanto, una respuesta. Véase la figura 8.13.

Length of proof is 2. Level of proof is 1.

----- PROOF -----

```
13 [] P($f2(x, y), y) | -O(x, y) .
6667 [] O(Z991, x) | x!=JAEN.
64013 [] -P(x, JAEN) | \$Ans(x) .
64021 [] x=x.
64026 [ur, 64013, 13] \$Ans($f2(x, JAEN)) | -O(x, JAEN) .
64501 [hyper, 64021, 6667] O(Z991, JAEN) .
64502 [binary, 64501.1, 64026.1] \$Ans($f2(Z991, JAEN)) .
```

----- end of proof -----

Comentarios:

La tabla de resultados es la siguiente:

Exp.	T. CPU(seg.)	c. gen.	res.	A1	A2	A3
R1	53.79	175	1	0	0	0
R2	59	6661	25	4	102	0
R3	58.4	1098	2	0	1	0
R4	54.79	570	1	0	1	0

EXPERIMENTO 2B:

NIVEL DE COMPACIDAD

BC: RCC + Tabla de composición + Ax. nombres únicos + Ax. clausura de dominio + Ax. completitud + Ax. igualdad + Ax. nivel de compacidad

Pregunta: Regiones que son parte de Jaén.

$P(x, \text{Jaen}) \rightarrow \$\text{Ans}(x)$

Resultado: Una respuesta obtenida a la pregunta es la siguiente:

Length of proof is 3. Level of proof is 3.

----- PROOF -----

```

12 [] -O(x,y) | P(Int(x,y),x) .
5275 [] O(JAEN,x) | x!=Z863 .
64021 [] -P(x,JAEN) | $Ans(x) .
64031 [] x=x .
64038 [ur,64021,12] $Ans(Int(JAEN,x)) | -O(JAEN,x) .
70990 [ur,64038,5275] $Ans(Int(JAEN,x)) | x!=Z863 .
71110 [copy,70990,flip.1] $Ans(Int(JAEN,x)) | Z863!=x .
71111 [binary,71110.1,64031.1] $Ans(Int(JAEN,Z863))$.

```

----- end of proof -----

Comentarios: Al haber añadido la axiomatización del nivel de compacidad 2 desaparecen las respuestas anteriormente obtenidas con funciones de Skolem.

Exp.	T. CPU(seg.)	c. gen.	res.	(A1)	(A2)	(A3)
R1	55.20	180	1	0	0	0
R2	60.26	677	2	2	0	0
R3	320	31855	287	5	0	1
R4	55.6	575	1	1	0	0

EXPERIMENTO 3:

BC: RCC + Tabla de composición + Ax. nombres únicos + Ax. clausura de dominio + Ax. completitud + Ax. igualdad

Pregunta: Regiones que solapan a la comarca Sierra-Norte y a Córdoba.

$$0(\text{SIERRA} - \text{NORTE}, x) \wedge 0(\text{Cordoba}, x) \rightarrow \$\text{Ans}(x)$$

Comentarios: Aparecen errores tipo A1 que indican (fallos en los datos de entrada)

Exp.	T. CPU(seg.)	c. gen.	res.	A1	A2	A3	A4
R1	592.55	32473	1	22	0	0	0
R2	597	32517	5	22	0	0	0
R3	54.19	182	0	1	0	0	0

EXPERIMENTO 3B:

BC: RCC + Tabla de composición + Ax. nombres únicos CORREGIDOS + Ax. clausura de dominio CORREGIDOS + Ax. completitud CORREGIDOS + Ax. igualdad + Ax. Nivel de compacidad

Pregunta: Regiones que solapan a la comarca Sierra-Norte y a Córdoba.

$$0(\text{SIERRA} - \text{NORTE}, x) \wedge 0(\text{Cordoba}, x) \rightarrow \$\text{Ans}(x)$$

Comentarios: Desaparecen los errores tipo A1 que indican (fallos en los datos de entrada) y tipo A2 correspondientes a funciones de Skolem, al corregir los datos de entrada e introducir la axiomatización del nivel de compacidad.

Exp.	T. CPU(seg.)	c. gen.	res.	A1	A2	A3	A4
R1	706.41	26379	1	0	0	0	0
R2	711	26452	5	0	0	0	0
R3	-	-	-	-	-	-	-

EXPERIMENTO 4:

BC: RCC + Tabla de composición + Ax. nombres únicos + Ax. clausura de dominio + Ax. completitud + Ax. igualdad

Pregunta: Regiones externamente conectadas con la provincia de Sevilla.
 $EC(x, Sevilla) \rightarrow \$Ans(x)$

Resultado: Una respuesta obtenida a la pregunta es:

Length of proof is 4. Level of proof is 2.

----- PROOF -----

```

19 [] -C(x,y) | O(x,y) | EC(x,y) .
1442 [] C(x, SEVILLA) | x! = Z1048 .
7041 [] -O(Z1048, x) | x = Z1048 | x = CAMPINA_DE_CADIZ | x = CADIZ .
7732 [] CADIZ! = SEVILLA .
7733 [copy, 7732, flip.1] SEVILLA! = CADIZ .
10633 [] SEVILLA! = CAMPINA_DE_CADIZ .
11018 [] SEVILLA! = Z1048 .
11019 [copy, 11018, flip.1] Z1048! = SEVILLA .
64024 [] -EC(x, SEVILLA) | $Ans(x) .
64030 [] x = x .
68842 [hyper, 64030, 1442] C(Z1048, SEVILLA) .
86840 [hyper, 68842, 19, unit_del, 64024] O(Z1048, SEVILLA)
                                     | $Ans(Z1048) .
116984 [hyper, 86840, 7041, unit_del,
        11019, 10633, 7733] $Ans(Z1048)$.

```

----- end of proof -----

Una respuesta en forma de disyunción es la obtenida en la siguiente prueba de OTTER, que aparece en la figura 8.16. De las dos regiones que aparecen en la misma, la primera de ellas no está externamente conectada con Sevilla, pero la segunda sí, véase la figura 8.15. Esta respuesta procede de un error en la implementación de los datos. La cláusula $1442[C(x, SEVILLA)|x! = Z1048]$ es la causante de la respuesta en cuestión, ya que esa zona cartográfica es parte propia de Cádiz, no externamente conectada con Sevilla.

Length of proof is 14. Level of proof is 4.

```

----- PROOF -----
14 [] -O(x,y) | -DR(x,y) .
19 [] -C(x,y) | O(x,y) | EC(x,y) .
33 [] -O(x,y) | O(y,x) .
1407 [] C(x,SEVILLA) | x!=Z938 .
1442 [] C(x,SEVILLA) | x!=Z1048 .
7041 [] -O(Z1048,x) | x=Z1048 | x=CAMPINA_DE_CADIZ | x=CADIZ .
7732 [] CADIZ!=SEVILLA .
7733 [copy,7732,flip.1] SEVILLA!=CADIZ .
7964 [] CADIZ!=Z938 .
7965 [copy,7964,flip.1] Z938!=CADIZ .
10633 [] SEVILLA!=CAMPINA_DE_CADIZ .
15133 [] CAMPINA_DE_CADIZ!=Z938 .
15134 [copy,15133,flip.1] Z938!=CAMPINA_DE_CADIZ .
51076 [] Z938!=Z1048 .
64024 [] -EC(x,SEVILLA) | $Ans(x) .
64028 [] O(x,y) | DR(x,y) .
64030 [] x=x .
68842 [hyper,64030,1442] C(Z1048,SEVILLA) .
68877 [hyper,64030,1407] C(Z938,SEVILLA) .
70212 [hyper,64028,7041] DR(Z1048,x) | x=Z1048
      | x=CAMPINA_DE_CADIZ | x=CADIZ .
70440 [hyper,64028,33] DR(x,y) | O(y,x) .
86840 [hyper,68842,19,unit_del,64024] O(Z1048,SEVILLA)
      | $Ans(Z1048) .
86911 [hyper,68877,19,unit_del,64024] O(Z938,SEVILLA)
      | $Ans(Z938) .
96671 [ur,70212,51076,15134,7965] DR(Z1048,Z938) .
116987 [ur,86840,7041,10633,7733,flip.2] $Ans(Z1048)
      | Z1048=SEVILLA .
116993 [ur,86911,14] $Ans(Z938) | -DR(Z938,SEVILLA) .
125939 [hyper,96671,14,70440] DR(Z938,Z1048) .
179924 [para_into,125939.1.2,116987.1.1] DR(Z938,SEVILLA)
      | $Ans(Z1048) .
179925 [binary,179924.1,116993.1] $Ans(Z1048) | $Ans(Z938) .

----- end of proof -----

```

Figura 8.16: Ejemplo de respuesta disyuntiva

Comentarios: En este caso se detectaron errores en el dominio de conocimiento. De las respuestas tipo A1 obtenidas, 72 eran consecuencia de dicho error, además de las 6 respuestas generadas en forma de disyunción. Se trata de deficiencias en la implementación de la tabla de composición de RCC, fallos comentados en [53].

La tabla de resultados es la siguiente:

Exp.	T. CPU(seg.)	c. gen.	res.	A1	A2	A3	A4
R1	2360	188169	1	113	0	0	0
R2	2366	196575	12	113	0	0	0
R3	53,33	184	0	1	0	0	0
R4	3845	11673078	25	113+72	0	6	1

8.4.2. Conclusiones

Una consecuencia interesante del experimento es que la búsqueda de inconsistencias (A1) produce, en algunos casos, *esquemas de argumentos inconsistentes*: esquemas que, por ligadura de variables, producen muchas inconsistencias desde que se encuentra la primera. Esta conducta se puede interpretar como proceso de aprendizaje porque el sistema genera un argumento general inconsistente para localizar carencias o errores concretos en la base de datos. Este fenómeno nos lleva a un hecho comprobado experimentalmente: una vez encontrado el primero, los siguientes aparecen rápidamente vía una sustitución de variables.

Es interesante notar que entre las características del sistema, podemos usar una opción que nos permita encontrar muchas pruebas de la misma pregunta, es decir, muchos argumentos. Por tanto, podemos localizar varias anomalías en la base de datos con una sola respuesta incorrecta.

El ciclo de limpieza permite detectar anomalías en los datos con el dominio de conocimiento, así como en el propio dominio. El siguiente objetivo es desarrollar un agente basado en este ciclo. Previamente debemos seleccionar banderas (etiquetas, comandos..., etc. de OTTER) para manejar grandes conjuntos de información o datos no útiles. Esta puede ser una aplicación prometedor del campo de la deducción automática para hacer consistentes los datos en entornos como la Web Semántica.

Capítulo 9

Conclusiones y trabajo futuro

En este capítulo incluimos las conclusiones del trabajo realizado en esta memoria, así como una breve descripción de cuáles podrían ser las líneas de investigación futura, justificando el interés que tiene abordar cada una de ellas.

9.1. Conclusiones

A lo largo de esta memoria se muestra la utilidad de los DAT para el razonamiento con ontologías. Hemos constatado que el uso de ellos y el diseño técnicas de demostración de teoremas en LPO para tareas derivadas de los procesos de creación y gestión de ontologías será de gran interés.

Se apuesta por la utilización de un DAT para LPO en el análisis de ontologías, no con el propósito de sustituir a los razonadores en DL, sino como mecanismo complementario que sirva para abordar problemas que, por razones de expresividad, pueden quedar fuera del alcance de los SRA basados en lógicas de la descripción. De hecho, hemos planteado su utilización para el *razonamiento metaontológico*.

La elección de OTTER no ha sido casual. Su modo autónomo ha permitido obtener pruebas sin la intervención del usuario¹. Su uso combinado con el generador automático de modelos asociado MACE4 ha sido fundamental para obtener la información necesaria sobre las extensiones de la teoría RCC, pudiendo establecerse el número y estructura de dichas extensiones. No obstante, podríamos haber utilizado otro sistema de razonamiento en LPO, teniendo en cuenta que cada demostrador se comporta de manera distinta y que

¹Si bien, a posteriori, y tras observar la información utilizada en una prueba, es posible simularla optimizándola respecto a los parámetros estadísticos del demostrador. También ha sido una ventaja el uso de su predicado de respuesta. Sin embargo, no ha sido la optimización de las pruebas uno de los objetivos de este trabajo.

dicho comportamiento está íntimamente relacionado con el tipo de problema planteado. Es decir, los resultados pueden estar afectados del comportamiento del DAT si se utiliza este para tareas distintas de la certificación de pruebas (que es el uso que le hemos dado).

Una opción a considerar es investigar la posible optimización de razonadores en DL existentes, como FaCT++² o, mejor aún, crear herramientas combinadas para facilitar el uso de lenguajes más expresivos como OWL DL, en las que el componente en LPO entre en juego allí donde su complementario en DL esté lejos del campo de los algoritmos existentes. En el caso de SWRL, donde el lenguaje completo ya no tiene una contrapartida en lógicas de descripción (ni en un fragmento decidible de LPO), podría usarse un demostrador en LPO para calcular, al menos, las inferencias más relevantes. Tal como se apunta en [62], aunque inevitablemente existen problemas relacionados con la velocidad de respuesta y, más importantes aún, con la incompletitud, podrían mejorarse algunos razonadores DL existentes que no dominan del todo esas cuestiones.

Existen otras conclusiones más específicas del presente trabajo que concretamos a continuación.

El trabajo desarrollado en la primera parte de esta memoria demuestra que es posible automatizar el Razonamiento espacial cualitativo mediante un SRA (en nuestro caso mediante un DAT, cap. 2). Esta automatización permite obtener extensiones del cálculo RCC (cap. 3), así como analizar la estructura de dichas extensiones, creando una metodología de inserción ontológica asistida por SRAs (cap. 4) que hemos aislado como método general [11]. Nótese que esta aproximación ha permitido trabajar con espacios topológicos a través de estructuras discretas (ecuacionales).

La segunda parte del trabajo demuestra que es posible establecer una semántica formal que justifique (valide) las transformaciones de las ontologías, mediante razonamiento espacial (cap. 5). Por medio de la interpretación débil, a una BC, denotada por Σ , le asociamos un PSR. A continuación obtenemos un modelo espacial, y, por tanto, un PSR equivalente al anterior (en el sentido de que uno es espacialmente consistente si lo es el otro). Por último, mediante el correspondiente teorema, traducimos la consistencia espacial del PSR a la consistencia de la BC asociada. Por tanto, **concluimos** que es posible traducir el conocimiento representado en una BC a una representación espacial, razonar en esa representación con las herramientas de cálculo espacial y traducir los efectos correspondientes en términos de nuevo conocimiento y modificaciones a la BC inicial. Todo ello, con las garantías que aporta la semántica formal subyacente.

Esta interpretación espacial constituye una base para obtener una clasifi-

²Una nueva generación de FaCT pero con arquitectura más moderna escrita en C++ en lugar de LISP. Véase (<http://wonderweb.semanticweb.org/>)

cación general de anomalías de tipo cognitivo intuitiva y, desde esta perspectiva, completa.

El análisis de los argumentos generados por un DAT puede servir de base para establecer un método de resolución de anomalías asistido por un ARS a través de la reparación de argumentos anómalos.

El análisis realizado justifica que las interpretaciones espaciales basadas en RCC pueden ser utilizables para la reparación de las anomalías, ya que es más fácil hacer inferencias en un marco espacial que en un marco teórico.

En el caso de que la ontología sea robusta (y por tanto es posible que las anomalías tengan su origen en los datos), es posible obtener mecanismos lógicamente fiables -asistidos por DATs- de reparación de anomalías (con respecto a la clasificación obtenida en el bloque anterior). Existen métodos de limpieza de datos, pero no suele justificarse el razonamiento que constituye la base de dichos métodos. En esta memoria, justificamos el razonamiento que subyace bajo un método para la depuración de ontologías.

9.2. Trabajo futuro

El trabajo presentado en esta memoria, resuelve los problemas abordados inicialmente combinando materias muy diversas. Por ello plantea nuevas líneas de investigación que pueden enmarcarse en distintas disciplinas. Dado que estructuramos el contenido en tres bloques, describamos las líneas de trabajo futuro atendiendo a dicha estructura:

- Partiendo de los capítulos 3 y 4, en los que trabajamos con extensiones del cálculo de RCC obtenidas con la asistencia de un DAT y se establece un método general de inserción ontológica, es interesante aplicar la metodología al estudio de extensiones \mathcal{E} -conservativas³ de ontologías en general, trabajo que ya hemos iniciado en [11].
- En segundo lugar, es especialmente interesante el desarrollo de una herramienta de razonamiento espacial cualitativo *robusta* que permita razonamiento espacial con regiones y líneas. Esto posibilitaría la extensión de nuestro trabajo al análisis, más amplio, no solo de la conceptualización sino del uso explícito de roles en el razonamiento espacial. Aunque existen algunas, no están profundamente estudiadas y no presentan una semántica clara. Nos hemos dedicado a explorar, clasificar y posteriormente resolver, anomalías en la conceptualización, pero no se han tratado específicamente los roles, su representación espacial y el razonamiento con ésta.

³Conservativas con respecto a un conjunto de fórmulas \mathcal{E} procedentes de la axiomatización.

- El uso que hemos hecho de MACE4 se reduce a la búsqueda de modelos de extensiones de la teoría y al análisis de sus tablas de Cayley para estudiar la estructura de las extensiones. Sería también muy útil la ampliación del uso de buscadores al análisis de argumentos generados por un DAT, para detectar rápidamente su consistencia en el caso general (y no sólo en el caso específico de bases de datos espaciales). Este análisis fundamentaría el uso de MACE4 para otras tareas.
- Un objetivo a medio plazo sería la implementación de una herramienta visual asociada al ciclo de limpieza diseñado en la sección 7.2.
- Un *árbol argumental* describe las diversas formas de alcanzar un argumento (o bien los contra-argumentos respecto de un argumento inicial). El resultado 8.3.4 relaciona los cortes y las configuraciones espaciales, que pueden ser útiles para estimar el tamaño de los *árboles argumentales* [9]. Para que este análisis sea completo es necesario determinar:
 1. Cuáles de las anomalías señaladas pueden clasificarse mediante argumentos, y
 2. Qué parte de la teoría de argumentos puede ser reproducida con OTTER
- Un objetivo relacionado con las anomalías originadas por el *ruido de Skolem*, dentro del conjunto de anomalías basadas en datos, consiste en el diseño de una metodología que posibilite su eliminación al menos cuando no sea testigo de una anomalía grave. Recordemos que este tipo de errores se detecta cuando la respuesta obtenida por el demostrador incluye funciones de Skolem para denotar a elementos cuyo nombre no figura expresamente en la base de datos. Una solución propuesta en nuestro trabajo consiste en el estudio del nivel de compacidad de la interpretación de la función en cuestión y la inclusión de la axiomatización de dicho nivel de compacidad en la base de conocimiento. Pero se necesita una metodología general para tratar con este problema.
- Por medio de los agentes inteligentes es posible filtrar, procesar e inferir información procedente de la Web sin la intervención del usuario. Ya hemos comentado la necesidad de establecer un equilibrio entre la efectividad del razonamiento en tareas complejas y la condición de conseguir un procesamiento en tiempo real por parte de un agente. Apuntamos como medida para alcanzar dicho equilibrio la incorporación de un DAT como *componente pensante* del agente. El desarrollo de nuevas metodologías asistidas por DATs para la realización de tareas de razonamiento cognitivamente complejas es especialmente interesante. Por otro lado, ya apuntamos que una de las desventajas de la incorporación de un

DAT a la arquitectura de un agente inteligente es la propia autonomía del demostrador. Señalamos por tanto el interés de trabajar en esa dirección.

- En esta memoria hemos diseñado y propuesto un ciclo de depuración de ontologías asistido por un DAT (8.1). Mediante un conjunto de preguntas, la aplicación reiterada de dicho ciclo permite detectar errores y corregirlos. No hemos estudiado el número de tests necesarios para poder considerar *limpia* la BC. Aunque el sentido común nos indica que a mayor número de pruebas, mayor calidad del conocimiento almacenado, hay estudios que demuestran que en realidad sólo es necesario un reducido número de comprobaciones para simular el conjunto de posibles respuestas de un sistema basado en conocimiento. Concretamente, sería interesante investigar el equilibrio entre el criterio de completitud y el número de tests necesarios. En el trabajo de Menzies y Cukies ([52]), se presentan resultados a cerca de cuál es el número de tests necesarios para *detectar* defectos. Sin embargo, habría que continuar esa investigación para responder a la pregunta de cuál es el número de test necesarios para *limpiar* una BC, o establecer unos parámetros para establecer la relación entre el número de tests a los que se somete la BC y la medida de la calidad del conocimiento almacenado.

Bibliografía

- [1] ALONSO-JIMÉNEZ, J.A., BORREGO-DÍAZ, J., CHÁVEZ-GONZÁLEZ, A.M.: *Ontology Cleaning by Mereotopological Reasoning* Proceedings of the 15th International Workshop on Database and Expert Systems Applications (DEXA'04), pp. 132-137. IEEE Computer Society, CA (USA) (2004).
- [2] ALONSO-JIMÉNEZ, J.A., BORREGO-DÍAZ, J., CHÁVEZ-GONZÁLEZ, A.M., GUTIÉRREZ-NARANJO, M.A., NAVARRO-MARÍN, J.D.: *Towards a Practical Argumentative Reasoning with Qualitative Spatial Databases* Developments in Applied Artificial Intelligence. Chung, P., Hinde, C, Ali, M. (eds.): IEA/AIE 2003, Lecture Notes in Artificial Intelligence 2718, pp.789-798. Springer-Verlag (2003).
- [3] ALONSO-JIMÉNEZ, J.A., BORREGO-DÍAZ, J., CHÁVEZ-GONZÁLEZ, A.M., GUTIÉRREZ-NARANJO, M.A., NAVARRO-MARÍN, J.D.: *A Methodology for the Computer-Aided Cleaning of Complex Knowledge Databases*. Proceedings of IECON 2002, pp. 1806-1811 IEEE Press (2002).
- [4] BAADER, F., CALVANESE, D., MCGUINNESS, D., NARDI, D., PATEL-SCHNEIDER, P.: *The Description Logic Handbook*. Cambridge University Press (2003).
- [5] BAADER, F., HORROCKS, I., SATTLER, U.: *Description Logics* In S. Staab and R. Studer (eds.) Handbook on Ontologies, pp. 3-28. Springer-Verlag (2004).
- [6] BENNETT, B.: *The application of Qualitative Spatial Reasoning to GIS*. In R.J: Abrahart (ed.) Proceedings of the First International Conference on Geo-Computation, pp. 44-47. Leeds, (1996).
- [7] BENNETT, B.; ISLI, A.; COHN, A.: *When does a Composition Table Provide a Complete and Tractable Proof Procedure for a Relational Constraint Language?* In Proceedings of the IJCAI-97 Workshop on Spatial and Temporal Reasoning, Nagoya, Japan (1997).
- [8] BERNS-LEE, T., HENDLER, J., LASSILA, O.: *The Semantic Web*, Scientific American 284 pp.34-43 (2001).

- [9] BESNARD, P., HUNTER, A.: *A logic-based theory of deductive arguments*. Artificial Intelligence 128(1-2), pp. 203-235 (2001)
- [10] BITTNER, T., STELL, J.G.: *Rough Sets in Approximate Spatial Reasoning*. Proceedings of RSCTC 2000, Lecture Notes in Artificial Intelligence, Springer-Verlag, pp. 445-453 (2000).
- [11] BORREGO-DÍAZ, J., CHÁVEZ-GONZÁLEZ, A.M.: *Extension of Ontologies assisted by Automated Reasoning Systems*. 10 th International Conference on Computer Aided Systems Theory (EUROCAST'05). Lecture Notes in Computer Sciences (Próxima aparición) Springer Verlag (2005).
- [12] BROEKSTRA, J., KLEIN, M., DECKER, S., FENSEL, D., HORROCKS, J.: *Adding formal semantics to the Web: building on the top of RDF Schema*. ECDL (2000).
- [13] CALVANESE, K., DE GIACOMO, G., LENZERINI, M., NARDI, D.: *Reasoning in Expressive Description Logics*. In Alan Robinson y Andrei Voronkov (eds.) Handbook of Automated Reasoning, pp. 1581-1634 Elsevier Science Publishers (2001).
- [14] BURRIS, S., SANKAPPANAVAR, H.P.: *A Course in Universal Algebra*. <http://www.math.sc.edu/~mcnulty/alglatvar/burrishsanka.pdf>
- [15] COHN, A.G., BENNETT, B., GOODAY, J. M., GOTTS, N. M.: *Representing and Reasoning with Qualitative Spatial Relations about Regions*. In Oliverio Stock (ed.) Temporal and Spatial Reasoning, pp. 97-134. Kluwer Academic Publisher (1997).
- [16] COHN, A.G., GOTTS, N.M.: *The 'Egg-Yolk' Representation of Regions with Indeterminate Boundaries* in P. Burrough and A. M. Frank (eds), Proceedings of GISDATA Specialist Meeting on Geographical Objects with Undetermined Boundaries, GISDATA Series, vol. 3, Taylor and Francis, pp. 171-187 (1996).
- [17] DAHN, I., DENZINGER, J.: *Cooperating Theorem Provers*. In W. Bibel and P.H. Schmitt (eds.) Automated Deduction. A basis for applications. Vol. II: Systems and Implementation Techniques, pp. 383-416. Kluwer Academic Publishers (1998).
- [18] DONINI, F.M., LENZERINI, M., NARDI, D., SCHAERF, A.: *Reasoning in description logics*. In Gerhard Brewka (ed.) Foundations of Knowledge Representation, pp. 191-236. CSLI-Publications (1996).

- [19] DOETS, K.; DE NIVELLE, H.: *Otter: A brief Introduction*. (1997).
<http://staff.science.uva.nl/mdr/CALG/Provers/Otter/Papers/ottman.ps.gz>
- [20] DÜNTSCH, I., WANG, H., MCCLOSKEY, S.: *A Relation-algebraic Approach to the Region Connection Calculus* Theoret. Comput. Sci. 255, pp. 63-83 (2001).
- [21] DRIES, L.: *Tame Topology and 0-minimal Structures*. London Mathematical Society. Lecture Note Serie 248. Cambridge University Press (1998).
- [22] EGENHOFER, M.J., HERRING, J.R.: *Categorizing Binary Topological Relations Between Regions, Lines, and Points in Geographic Databases*. Technical Report, Dept. of Surveing Engineering, University of Maine, (1990). (Revised version in NCGIA Technical Report 94-1).
- [23] ELVANG-GORANSSON, M., HUNTER, A.: *Argumentative logics: Reasoning from classically inconsistent information*. Data and Knowledge Engineering 16, pp. 125-145 (1995).
- [24] FIKES, R., MCGUINNESS, D.L., WALDINGER, R.: *A First-Order Logic Semantics for Semantic Web Markup Languages*. Tech. Rep. n. KSL-01-01. Knowledge Systems Laboratory, Stanford University (2002).
- [25] FLUIT, C., SABOU, M., HARMELEN, F.: *Ontology-based Information Visualization V*. Geroimenko, C. Chen (eds.), Visualizing the Semantic Web. Springer-Verlag (2003).
- [26] FLUIT, C., SABOU, M., HARMELEN, F.: *Supporting User Tasks through Visualisation of Light-weight Ontologies*. S. Staab and R. Studer (eds.), Handbook on Ontologies in Information Systems. Springer-Verlag (2003).
- [27] FREKSA, C.: *Conceptual Neighborhood and its Role in Temporal and Spatial Reasoning*. In Procc. of the IMACS Workshop on Decision Support Systems and Qualitative Reasoning, pp. 181-187, North-Holland, Amsterdam 1991.
- [28] FREKSA, C.: *Temporal Reasoning based on semi-intervals*. Artificial Intelligence 54, pp.199-227 (1992).
- [29] GALHARDAS, H., FLORESCU, D., SHASHA, D.: *Declarative Data Cleaning: Language, Model and Algorithms* Proc. de 27th VLDB Conference. Roma, Italy (2001).
- [30] GALLAIRE, H., MINKER, J., NICOLAS, J.M.: *Logic and Databases: A Deductive Approach* Computing Surveys, vol. 16, (2), (1984).

- [31] GANTER, B., WILLE, R.: *Formal Concept Analysis: Mathematical Foundations*. Springer Verlag (1999).
- [32] GEREVINI, A., RENZ, J.: *Combining topological and size information for spatial reasoning*. *Artificial Intelligence* 137, pp. 1-42 (2002)
- [33] GOTTS, N.: *An Axiomatic Approach to Topology for Spatial Information Systems*. Tech. Rep. n. 96.25. School of Computer Studies. University of Leeds (1996).
- [34] GRIGNI, M., PAPADIAS, D., PAPADIMITRIOU, C.: *Topological inference*. In C. Mellish (ed.) *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, (1) pp. 901-906. Morgan Kaufmann (1995).
- [35] GRIGORIS, A., HARMELEN, F.: *Web Ontology Language: OWL* En S. Staab and R. Studer (eds.) *Handbook on Ontologies*, pp. 67-92. Springer-Verlag (2004).
- [36] GUARINO, N., AND WELTY, C.: *Towards a methodology for ontology-based model engineering*. *Proceedings of the ECOOP-2000 Workshop on Model Engineering* (2000).
- [37] HARMELEN, F.: *The Complexity of the Web Ontology Language*. *IEEE Intelligent Systems*, vol. 17, pp. 67-92 (2002).
- [38] HORROCKS, I., FENSEL, D., BROEKSRA, J., DECKER, S., ERDMANN, M., GOBLE, C., HARMELEN, F., KLEIN, M., STAAB, S., STUDER, R., MOTTA, E.: *The Ontology Inference Layer OIL*. <http://www.ontoknowledge.org/oil/TR/oil.long.html>
- [39] HORROCKS, I., MCGUINNES, C.A., WELTY.: *Digital Libraries and Web-Based Information Systems*. In F. Baader et al. (eds.) *The Description Logic Handbook*, pp. 427-449. Cambridge University Press (2003).
- [40] HORROCKS, I., PATEL-SCHNEIDER, P. F., HARMELEN, F.: *From SHIQ and RDF to OWL: The making of a web ontology language*. *Journal of Web Semantics*, 1(1):7-26 (2003).
- [41] KALMAN, J.A.: *Automated Reasoning with Otter*. Rinton Press (2001).
- [42] KIM, W., CHOI, B.J., HONG, E.K., KIM, S.K., LEE, D.: *A Taxonomy of Dirty Data*. *Data Mining and Knowledge Discovery*, 7 pp. 81-99 (2003).
- [43] KLEIN, M., FENSEL, D.: *Ontology Versioning on the Semantic Web*. In I. Cruz (ed.) *Proc. of the the 1st Semantic Web Working Symposium*, pp. 75-91 (2001).

- [44] KNAUFF, M., RAUH, R., RENZ, J.: *A cognitive assessment of topological spatial relationships: Results from an empirical evaluation*. Proceedings of the 3rd Int. Conf. on Spatial Information Theory (COSIT'97). Lecture Notes in Computer Science 1329, pp. 193-206 Springer-Verlag (1997).
- [45] KONG, H., JUNG, K., CHOI, J., KIM, W., KIM, P., PARK, J.: *Representing the Spatial Relations in the Semantic Web Ontologies*. In T.D. Gedeon & L.C.C. Fung (Eds.): AI 2003, Lecture Notes in Artificial Intelligence 2903, pp. 77-87 Springer-Verlag (2003).
- [46] KRZYSZTOF, R. APT: *Principles of Constraint Programming*. Cambridge University Press (2003).
- [47] LAMPING, J., RAO, R., PIROLI, P.: *A Focus+Content Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies* ACM Conference on Human Factors in Software (CHI'95) pp. 401-408 (1995).
- [48] LASSILA, O.: *Web Metadata: A Matter of Semantics*. IEEE Internet Computing. 2(4), pp. 30-37 (1998).
- [49] LI, S., YING, M.: *Region Connection Calculus: Its Models and Composition Table* Artificial Intelligence 145, pp. 121-146 (2003).
- [50] MCCUNE, W.: *OTTER's user manual*. Argonne National Laboratory (1994). <http://www-unix.mcs.anl.gov/AR/otter/>
- [51] MCCUNE, W.: *MACE4 Reference Manual and Guide*. Argonne National Laboratory (2003).
- [52] MENZIES, T., CUKIC, B.: *Adequacy of limited Testing for Knowledge Based Systems* International Journal on Artificial Intelligence Tools, 9(1), pp.153-172 (2000).
- [53] NAVARRO MARÍN, J.D.: *Aplicación de Técnicas de Razonamiento espacial a Sistemas de Información Geográfica Utilizando una Automatización del Cálculo RCC*. Proyecto Fin de Carrera dirigido por Dr. D. Joaquín Borrego Díaz. Sevilla (2001).
- [54] NOAH S.A., WILLIAMS, M.D.: *Intelligent Database Design Diagnosis: Performance Assessment with the Provision of Domain Knowledge*. Artificial Intelligence Review, 21(1), pp.57-84(2004).
- [55] NOY, N.F., MCGUINNESS, D.I. *Ontology Development 101: A Guide to Creating Your First Ontology* KSL Technical Report KSL-01-05.
- [56] PAWLAK, Z.: *Rough Sets*. Int. Journal of Computation and Information Sciences, 11. pp.341-356 (1982).

- [57] RANDELL, D.S.; COHN, A.G.; CUI, Z.: *Computing Transitivity Tables: A Challenge For Automated Theorem Provers*. Proceedings of Conference on Automated Deduction (CADE), Lecture Notes in Computer Science. Springer Verlag (1992).
- [58] REITER, R.: *Towards a Logical Reconstruction of Relational Database Theory*. En M.L. Brodie, J.L. Mylopoulos and J.W. Schmidt (eds.). On Conceptual Modelling, pp. 191-233. Springer, N.Y. (1982).
- [59] RENZ, J.: *A Canonical Model of the Region Connection Calculus*. Proceedings of the 6th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'98) (1998).
- [60] RENZ, J.: *Qualitative Spatial Reasoning with Topological Information*. Lecture Notes in Artificial Intelligence 2293. Springer Verlag (2002)
- [61] RENZ, J., NEBEL, B.: *On the Complexity of Qualitative Spatial Reasoning: A Maximal Tractable Fragment of the Region Connection Calculus*. Artificial Intelligence 108(1-2) pp. 69-123 (1999).
- [62] TSARKOV, D., RIAZANOV, A., BECHHOFFER, S., HORROCKS, I.: *Using Vampire to Reason with OWL*. Research Paper at the 3rd International Semantic Web Conference (ISWC2004), 7-11 Hiroshima, Japan (2004).
- [63] VIGNERON, L., WASILEWSKA, A.: *Rough Diagrams* In T. Y. Lin, editor, 6th International Workshop on Rough Sets, Data Mining and Granular Computing (RSDMGrC'98) at the 4th Joint Conference on Information Sciences, Research Triangle Park (NC, USA), 1998.