Universidad Nacional del Sur

Tesis de Doctor en Ciencias de la Computación

# Razonamiento y Evolución Ontológica con Tolerancia a Inconsistencias.

*Un Enfoque Argumentativo a Revisión de Description Logics.*

Martín O. Moguillansky

Bahía Blanca Argentina

2010

# Prefacio

Esta tésis es presentada como parte de los requisitos para optar por el grado de Doctor en Ciencias de Computación de la Universidad Nacional del Sur, y no ha sido presentada anteriormente para optar por ningún otro grado, en otra ni en esta misma institución académica. Aquí se presentan los resultados de investigación obtenidos durante el período comprendido entre el 12 de septiembre de 2006 y el 30 de noviembre de 2010, bajo la supervisión y dirección del Dr. Marcelo A. Falappa.

Martín O. Moguillansky
mom@cs.uns.edu.ar
Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)
Lab. de Investigación y Desarrollo en Inteligencia Artificial (LIDIA)
Departamento de Ciencias e Ingeniería de Computación (DCIC)
Universidad Nacional del Sur (UNS)
Bahía Blanca, 30 de noviembre de 2010.

UNIVERSIDAD NACIONAL DEL SUR
Secretaría General de Posgrado y Educación Continua

La presente tesis ha sido aprobada el .../.../..., mereciendo la calificación de ......(.........)

# Agradecimientos

Por sus contribuciones a este trabajo de tesis, agradezco muy especialmente a mi director Dr. Marcelo A. Falappa; a la Dra. Renata Wassermann, quien ha colaborado en gran parte de las propuestas de esta tesis; a mi colega y amigo, Dr. Nicolás D. Rotstein, con quien tuve la suerte de discutir y formar las propuestas de esta tesis en sus instancias iniciales; y al Dr. Alejandro J. García y Dr. Guillermo R. Simari, por su colaboración desinteresada en mi formación de post-grado.

A mis compañeros de la sala de becarios de computación Mariano Tucat, Carlos Lorenzetti, Luciano Tamargo, Sebastián Gottifredi, Fernando Saguí, Mauro Gomez, Andrea Cohen, Rocío Cecchini, Julieta Marcos, y Diego García; por brindarme su apoyo y amistad durante mi período de estudiante doctoral.

A mi familia, y especialmente a mis padres, Oscar y Olga, y a mis hermanos, Marcos y Gisela; a mi hermosísima novia y compañera, Sabina Bilder; al arte y la música en mi vida, y a mis compañeros de SCROM, y demás amigos personales. Por su aliento incondicional para sobreponerme a cada situación que he enfrentado en la vida, y sobre todo por el apoyo recibido durante los últimos cinco años en que desarrollé las investigaciones de esta tesis.

A mis raíces. A la vida y por la vida.

*Martín Moguillansky*

# Resumen

El razonamiento y cambio de bases de conocimiento o *knowledge bases* (KBs) por sobre las inconsistencias, es de extrema importancia en áreas como la medicina y el derecho. Esto es, razonar y provocar la evolución del conocimiento sin necesidad de restaurar la consistencia, sino proveyendo una forma de tolerancia a ella. La argumentación puede brindar la posibilidad de dar con ambos problemas. Primeramente, mediante la construcción de un marco argumentativo o *argumentation framework* (`AF`) a partir de la KB inconsistente, podemos decidir si aceptar o rechazar una cierta conclusión o *claim* a través de la interacción entre argumentos y contra-argumentos. Segundo, mediante el manejo de la dinámica de argumentos del `AF`, podemos dar con la dinámica del conocimiento de la KB inconsistente subyacente.

Por un lado, propondremos una nueva familia de marcos argumentativos abstractos a los cuales referimos como generalizados, e identificamos como *generalized abstract argumentation frameworks* (`GenAF`), debido a su habilidad de adaptación a diferentes lenguajes de representación. El objetivo es proveer un marco argumentativo, no completamente abstracto, para razonar por sobre las inconsistencias de KBs representadas a través de cualquier lenguaje que se sepa conforme algún fragmento de primer-orden. Las semánticas estándar de Dung son adaptadas para construir la maquinaria de razonamiento del `GenAF`.

En lo que constituye un primer enfoque a *revisión de creencias* en esta tesis, se propondrá un operador de debugging para la KB subyacente –definido sobre las semánticas argumentativas para `GenAF`s– como un tipo de *consolidación*: operación propuesta por Hansson para restaurar consistencia a KBs. Por ello, nos basaremos sobre sus postulados usuales para caracterizar axiomáticamente la operación de debugging propuesta, mostrando el correspondiente teorema de representación. Luego, proponemos la reificación del lenguaje abstracto para argumentos del `GenAF` al lenguaje básico de descripción $\mathcal{ALC}$. Esto muestra la flexibilidad del formalismo presentado y una forma de aplicar argumen-

tación a ontologías para razonar sobre las inconsistencias. Finalmente, la operación de debugging provee una herramienta para reparar inconsistencias y conceptos insatisfacibles, restaurando consistencia-coherencia a las ontologías $\mathcal{ALC}$.

La dinámica de argumentos ha recientemente atraído atención y aunque algunos enfoques han sido propuestos, una axiomatización completa dentro de la teoría de revisión de creencias constituyó un resultado pendiente hasta el momento. Una revision surge cuando deseamos que las semánticas argumentativas acepten un nuevo argumento. La teoría del cambio argumentativo o *Argument Theory Change* (ATC) define operadores de revision que modifican un `AF` mediante el análisis de árboles de dialéctica –argumentos como nodos y ataques como arcos– como la semántica argumentativa adoptada.

Presentaremos un simple enfoque a ATC basado en KBs proposicionales. Esto nos permite manejar el cambio de KBs inconsistentes basándonos en la teoría clásica de revisión de creencias, aunque al contrario de lo que ella indica, se evitará la restauración de consistencia de la KB a trabajar. Subsecuentemente, un conjunto de postulados de racionalidad será adaptado a argumentación y finalmente, el modelo de cambio propuesto será relacionado a los postulados a través del correspondiente teorema de representación. Seguidamente, los resultados serán extendidos a description logics, para manejar el razonamiento y evolución ontológicos con tolerancia a inconsistencias.

# Palabras clave

Revisión de Creencias, Argumentación, Dinámica de Argumentos, Dinámica del Conocimiento, Lógica Clásica, Lógica de Primer-orden, Lógicas de Descripción, Tolerancia a Inconsistencias, Razonamiento No-monótono, Debugging y Reparación de Ontologías, Evolución Ontológica, Web Semántica.

# Abstract

Reasoning and change over inconsistent knowledge bases (KBs) is of utmost relevance in areas like medicine and law. Argumentation may bring the possibility to cope with both problems. Firstly, by constructing an argumentation framework (`AF`) from the inconsistent KB, we can decide whether to accept or reject a certain claim through the interplay among arguments and counterarguments. Secondly, by handling dynamics of arguments of the `AF`, we might deal with the dynamics of knowledge of the underlying inconsistent KB.

On the one hand, we propose a new family of abstract argumentation frameworks which we refer as generalized (identified through the acronym `GenAF`), due to its ability of adapting to different representation languages. The objective is to provide a not so, but still, abstract argumentation framework for reasoning over inconsistent knowledge bases (KBs) represented through any language known to conform to some first-order fragment. The well known Dung's standard semantics are adapted to construct the `GenAF`'s reasoning machinery.

Constituting the first approach to belief revision in this thesis work, we propose a debugging operator for the underlying KB defined upon the argumentation semantics for `GenAF`s, as a kind of belief revision's consolidation. A consolidation is a well known Hansson's operation for restoring consistency to KBs, therefore, we rely upon its usual postulates to axiomatically characterize the debugging operation proposed here, showing the corresponding representation theorem. Afterwards, we propose a reification of the `GenAF`'s abstract language for arguments to the basic $\mathcal{ALC}$ description logic. This shows both the flexibility of the formalism presented, and a way to apply argumentation for reasoning over inconsistent ontologies, subject of utmost relevance in areas like medicine and law. Finally, the debugging operation provides a tool for debugging inconsistent ontologies and repairing unsatisfiable concepts, for restoring both consistency and coherency to $\mathcal{ALC}$ ontologies.

Dynamics of arguments has recently attracted attention and although some approaches have been proposed, a full axiomatization within the theory of belief revision was still missing. A revision arises when we want the argumentation semantics to accept an argument. Argument Theory Change (ATC) encloses the revision operators that modify the `AF` by analyzing dialectical trees –arguments as nodes and attacks as edges– as the adopted argumentation semantics.

We present a simple approach to ATC based on propositional KBs. This allows to manage change of inconsistent KBs by relying upon classic belief revision, although contrary to it, consistency restoration of the KB is avoided. Subsequently, a set of rationality postulates adapted to argumentation is given, and finally, the proposed model of change is related to the postulates through the corresponding representation theorem. Afterwards, the results are extended to description logics, to handle ontology reasoning and evolution with tolerancy to inconsistency.

## Keywords

Belief Revision, Argumentation, Dynamics of Arguments, Dynamics of Knowledge, Classic Logic, First-order Logic, Description Logics, Inconsistency Tolerance, Non Monotonic Reasoning, Ontology Debugging and Repair, Ontology Evolution, Semantic Web.

# Índice general

# Índice de figuras

# Índice de cuadros

# Capítulo 1

# Introducción y Motivación

El trato de las inconsistencias es de extrema importancia en áreas como la medicina y el derecho. Por ejemplo, en un proceso judicial, las dos partes de la disputa presentan información contradictoria ante un tribunal, a favor o en contra del punto en disputa (en procesos criminales esto se conoce como la presunción de inocencia). Luego el tribunal resuelve la disputa a partir de la evaluación de la evidencia presentada. Esto muestra la necesidad de considerar algún tipo de semántica paraconsistente para lograr razonar apropiadamente sobre bases de conocimiento o *knowledge bases* (KBs) conteniendo información contradictoria.

Para algunos entornos, será también necesario proveer servicios para el manejo de la dinámica del conocimiento con capacidades para tolerar inconsistencias en la KB. En particular, un entorno interesante es el de la promulgación de leyes. Esto usualmente involucra un extenso proceso en el cual artículos y principios de leyes anteriores, e incluso evidencia considerada a partir del estado actual de las cosas, pueden entrar en conflicto con artículos que conforman la nueva ley. Imagine una base conteniendo conocimiento sobre el sistema legal de una nación, incluyendo la Constitución Nacional, la ley internacional, y otros principios políticos fundamentales –como los códigos civil y penal, y otras leyes locales menores. Se espera que tal KB evolucione de forma tal que incorpore la información conformando la nueva ley, asegurando su constitucionalidad. Para este fin, es necesaria la identificación de un conjunto de artículos y/o principios a ser derogados, o reformados (enmendados), como parte del proceso de promulgación.

Como ejemplo, nos referiremos en forma somera a la nueva Ley de Medios de Argentina, reformada durante el 2009. La ley de medios anterior, promulgada por el último régimen *de facto*, otorgó al gobierno facultades para regular los diferentes medios de comunicación, permitiendo un control absoluto de la información. Cuando la democracia fue restituída, la regulación de los medios fue extendida a grupos inversores privados. A medida que los años pasaron, estos grupos se apoderaron de la mayoría de los diferentes tipos de medios, conformando en algunos casos monopolios. Esto dió un poder excesivo a grupos con intereses parciales, permitiéndoles manipular la opinión de la sociedad sobre el gobierno de turno, e incluso condicionar a los políticos y sus decisiones, atentando de esta forma contra la soberanía nacional. Por este motivo, el artículo 161 de la nueva ley de medios se ha convertido en uno de los puntos más controversiales, dado que fuerza a empresas "monopólicas" a deshacerse de sus activos en un período máximo de un año. Incluso, algunas empresas advirtieron que serían forzadas a vender sus activos a precios extremadamente bajos. Esto violaría el artículo 17 de la Constitución Nacional, el cual vela por los derechos de propiedad privada. Más aún, algunos miembros de la Corte Suprema de Justicia de la Nación Argentina advierten que el artículo 161 recuerda al control sobre los medios ejercido por regímenes totalitarios, lo cual violaría el artículo 1 de la Constitución Nacional. De hecho, tal situación evolucionaría a un estado de desconfianza sobre el principio de Seguridad Jurídica. Estos son apenas algunos de los puntos controversiales por los cuales la nueva ley de medios está aún siendo resistida por algunos miembros del Congreso Argentino.

El mencionado ejemplo muestra sólo algunas de las motivaciones para la investigación y desarrollo de teorías cuyo foco es el razonamiento sobre bases inconsistentes y el manejo de su evolución con tolerancia a inconsistencias. Esto enmarca las contribuciones de esta tesis en tres diferentes áreas dentro de la Inteligencia Artificial como son la teoría de revisión de creencias o *belief revision*, las teorías de argumentación, y el razonamiento y evolución sobre lenguajes de representación de ontologías. Es por ello, que las investigaciones de esta tesis se basan en tres pilares fundamentales: por el lado de revisión de creencias, el aspecto novedoso es la propuesta de metodologías de cambio sin restauración de consistencia. Sin embargo, para ello se proveen caracterizaciones axiomáticas en base a los postulados usuales de la teoría del cambio, y otros elementos propios de los modelos de cambio de mayor impacto en el área, como son el modelo AGM (AGM85) (por sus creadores Alchourrón, Gärdenfors, y Makinson) y los modelos basados en Kernel Sets de Hansson (Han94). Por el lado de la argumentación, las contribuciones apuntan principalmente a la dinámica de argumentos en marcos argumentativos generalizados (capaces de aplicarse a diferentes lenguajes de representación). Para ello, nos basamos en el trabajo seminal de Dung (Dun95) sobre marcos argumentativos abstractos y sus semánticas, y otras semánticas argumentativas ampliamente referidas en la literatura especializada como lo son los *árboles de dialéctica*. Por último, las ontologías y en particular las Description Logics (DLs), son estudiadas en detalle como forma de aplicación de las teorías propuestas en esta tesis. Nuestro interés en la aplicación a ontologías yace en la capacidad por parte de las DLs de estructurar diferentes tipos de información compleja, como es el caso de las leyes –mencionado anteriormente. Por otro lado, el estudio de teorías sobre DLs es parte de la tarea global en pos de la construcción de la web semántica. Es por ello, que el objetivo de las contribuciones teóricas de esta tesis es lograr cierto impacto en el área de la World Wide Web (WWW), mediante la propuesta de maquinarias DLs no-estándares para razonar sobre ontologías inconsistentes y manejar su evolución con tolerancia a inconsistencias. Sin embargo, para ello, basamos nuestras investigaciones (por ejempo, la construcción de argumentos a partir de ontologías) en razonadores DLs estándares y sus metodologías clásicas como son el uso de *algoritmos tableaux*. Finalmente, la fundamentación de las nuevas teorías sobre pilares teóricos ampliamente aceptados, logra que las futuras implementaciones de las contribuciones teóricas de esta tesis sean beneficiadas por los diferentes avances correspondientes a las tres áreas de impacto.

## 1.1. Breve Introducción a las Teorías Fundamentales

La teoría de revisión de creencias o *belief revision* (AGM85; Han99), estudia la dinámica del conocimiento, dando con el problema de como cambiar la información sobre la conceptualización del mundo modelado, para reflejar su evolución. Las *revisiones*, como las operaciones de cambio más importantes, se concentran en la incorporación de nuevo conocimiento de forma tal que la base resultante concluya consistentemente. Como un ejemplo simple de revisión, consideremos el siguiente problema: por muchos años se creyó que el punto de ebullición del agua era de 100℃. Sin embargo, investigaciones posteriores mostraron que esto sólo se mantiene bajo condiciones estándares de presión atmosférica. Más aún, fue demostrado que el punto de ebullición del agua decrece a razón de 1℃ cada 285$m$ de elevación. Esta nueva evidencia forzó la revisión de los conocimientos en el área para mantenerlos actualizados y consistentes. Observe sin embargo, que para el caso de promulgación de leyes anteriormente mencionado, es obligatorio mantener intacto (tanto como sea posible) el conocimiento, incluyendo las fuentes de inconsistencia, de la KB para lograr su evolución apropiada. Esto demanda la investigación de nuevos enfoques de revisión de creencias que operen sobre semánticas paraconsistentes, a fin de evitar la restauración de consistencia.

La teoría de argumentación (BCD07a) permite razonar sobre KBs potencialmente inconsistentes. En general, esto es realizado mediante la interpretación de la definición usual de inferencia en lógica clásica, como la noción de *garantía* o *warrant*, en argumentación: el proceso de garantía evalúa piezas conflictivas de información para decidir cuales prevalecen a pesar de la existencia de creencias en contraposición. La noción de garantía es también identificada como el criterio de aceptación de argumentos, correspondiente a las semánticas argumentativas adoptadas. Por lo tanto, a partir de un marco argumentativo o *argumentation framework* ($\mathtt{AF}$), una creencia $\alpha$ es garantizada si existe un argumento soportando $\alpha$ el cual es aceptado por la semántica argumentativa. Entre los trabajos de mayor influencia en semánticas argumentativas, podemos citar aquellos sobre grafos de argumentos tales como (Dun95; BG07). Sin embargo, otra semántica disponible persigue la idea de argumentación dialéctica (PV00; CML00): *árboles de dialectica* son árboles de argumentos construidos a partir del $\mathtt{AF}$ con argumentos como nodos y ataques como arcos. Un par de nodos conectados a través de un arco determina una fuente de inconsistencia en la KB subyacente. El uso de árboles de dialéctica, nos permite concentrar en una consulta específica para construir "en demanda" aquellos argumentos que están de alguna forma relacionados a la consulta. Tal tipo de semánticas permiten la construcción de enfoques prácticos evitando el análisis completo del grafo de argumentos.

La teoría de cambio argumentativa o Argument Theory Change (ATC) (RMF$^+$08b) aplica nociones de la teoría clásica del cambio, y particularmente del reconocido modelo de cambio AGM (AGM85) (nombrado por las iniciales de sus creadores Alchourrón, Gärdenfors, y Makinson) y las Kernel Contractions (Han94) de Hansson, al área de argumentación abstracta (Dun95) basandose en árboles de dialéctica como la semántica argumentativa adoptada. (En argumentación abstracta, no hay especificación alguna para la lógica de argumentos ni para su estructura interna.) Una revisión de argumentos *à la* ATC revisa un $\mathtt{AF}$ por un argumento buscando su garantía. A tal fin, el $\mathtt{AF}$–y el conjunto de argumentos en él– es modificado a fin de que la semántica argumentativa acepte el nuevo argumento. Esta es la condición de *éxito* o *success* adoptada por los diferentes enfoques de ATC. En revisión de creencias clásico, un conjunto básico de postulados es usualmente especificado para caracterizar un comportamiento racional de las operaciones de cambio utilizadas. Entre ellos, éxito y *cambio mínimo* o *minimal change* han concentrado mucho del esfuerzo dedicado a su investigación. Éxito especifica el principal objetivo de la operación de cambio, usualmente, la aceptación (inferencia) de la nueva creencia a incorporar. Por otro lado, cambio mínimo asegura que tan poca información como sea posible será modificada en pos del éxito. Consecuentemente, para los enfoques a ATC, diferentes criterios de cambio mínimo surgen dependiendo del punto de vista: la cantidad de cambio puede ser analizada respecto de (1) el conjunto de argumentos, (2) el árbol de dialéctica enraizado en el argumento a revisar, (3) el conjunto de argumentos aceptados o *warranted*, o (4) cualquier combinación entre ellos.

Entre los usos de mayor relevancia de ATC, podemos mencionar razonamiento hipotético, dinámica en negociación, persuasión, diálogos, estrategias, planeamiento, entre otros. Por ejemplo, consideremos para el caso de planeamiento, el desarrollo de un organizador o *scheduler* de tareas. Asumamos además, que las asignaciones de empleados son manejadas por un agente que interpreta la KB. Seguidamente, la autoridad central incorpora nuevas tareas a la KB. Un agente usa esta información para decidir a qué empleados se les deberían asignar las nuevas tareas. Una medida de relevancia de una tarea para un empleado específico podría surgir a partir de sus asignaciones actuales. Una nueva tarea con un alto nivel de importancia podría ser enviado a un empleado específico por cuestiones de confianza, provocando la reasignación de sus tareas anteriores a otros empleados. ATC puede ser útil en la definición del procedimiento de organización mediante el reconocimiento de las asignaciones conflictivas en términos de factibilidad.

## 1.2. Breve Resumen de las Teorías Propuestas

Como mencionamos anteriormente, en argumentación abstracta no se considera una KB subyacente, y los argumentos son tratados como entidades atómicas del marco o *framework*. Por ello, no es necesario especificar lógica alguna para los argumentos ni para representar ningún tipo de conocimiento. Este es el caso del `AF` de Dung (Dun95), que en esa forma sirvió mayormente para concentrar las investigaciones sobre semánticas argumentativas. Sin embargo, la aplicabilidad de los `AF`s al estilo de Dung sobre diferentes KBs es usualmente una tarea compleja. Varios inconvenientes aparecen al momento de concretizar la lógica abstracta de argumentos para manejar lenguajes de representación específicos.

Por esa razón, la primer contribución de esta tesis presentada en el Capítulo 3, será la propuesta de un marco argumentativo abstracto generalizado o *generalized abstract argumentation framework* (`GenAF`) por el cual formalizaremos una maquinaria de razonamiento para bases inconsistentes sin tener en cuenta ninguna lógica específica. Esto implica que el lenguaje utilizado para representar conocimiento y argumentos se mantendrá abstracto en un nivel teórico, restringiendo su máxima expresividad posible a la lógica de primer orden o *first-order logic* (FOL).

Un lenguaje abstracto $\mathbb{A}\mathtt{rgs}$, denominado *lenguaje argumental*, es propuesto para proveer de cierta estructura a la noción de argumento manteniendo su abstracción. El objetivo de mantener abstracta la configuración completa de la lógica de argumentos en un `GenAF` tiene la intención de ser eventualmente reificado a diferentes fragmentos FOL sin traer mayores inconvenientes. Consecuentemente, la generalización deja una forma de argumentación no completamente abstracta, la cual desarrolla los elementos teóricos necesarios para especificar los componenetes internos de los argumentos, su interrelación, y las estructuras dentro de las cuales los argumentos pueden ser agregados para soportar una consulta, o incluso para derrotar otras estructuras.

Dada su naturaleza generalizada, el `GenAF` podrá beneficiarse de los contínuos avances en el área de argumentación abstracta. Un caso específico es el de las semánticas argumentativas: el `GenAF` aquí propuesto se basa sobre las semánticas estándares de Dung para especificar su maquinaria de razonamiento. Sin embargo, otras nuevas semánticas como las investigadas en (BG07) pueden ser facilmente adaptadas a los `GenAF`s. Recordemos que la semántica argumentativa asegura la obtención de un conjunto de argumentos libre de conflicto, llamado *extensión*, a partir del cual el conocimiento aceptado (*i.e.*, formulas garantizadas) puede ser identificado. A través de sus semánticas argumentativas, un `GenAF` permite no sólo razonar sobre las inconsistencias para un amplio rango de formalismos, sino también dar con el problema de restauración de consistencia de KBs. Dicha tarea es obtenida en forma directa de una extensión obtenida.

En un `GenAF`, cada argumento representa una formula de la KB subyacente. Esto permite compartir los mismos elementos primitivos correspondientes al marco argumentativo (argumentos) y a la KB (fórmulas). Como ventaja resultante, es destacable la facilidad con la que un `GenAF` puede adaptarse para dar con el problema de la dinámica del conocimiento como ha sido realizado en (RMF$^+$08b): la eliminación de un argumento del marco implicaría la remoción de su regla correspondiente de la KB. Por otro lado, la especificación de los componentes internos de argumentos nos permite establecer el formato general de una fórmula del cual se identifica la porción mínima que estará disponible para ser eliminada de la KB. En este sentido, proponemos el uso de técnicas de compilación de conocimiento o *knowledge compilation* como (DM02), entre otros, para transformar la KB de acuerdo a alguna forma normal con el objectivo de especificar el conocimiento mediante una forma granularmente fina a fin de mejorar la eficiencia de las consultas. Las ventajas de tal decisión incluyen la representación de argumentos pequeños, proveyendo una metodología granular de reparación de fórmulas, y reduciendo la complejidad computacional de las consultas lo cual compensará el costo adicional de preprocesamiento inicial enfrentado por la tarea de normalización de la KB.

Nos concentraremos en el debugging de KBs como una forma de restauración de consistencia. A tal fin, proponemos una operación de debugging para identificar los posibles subconjuntos consistentes de la subyacente KB normalizada. Una operación de debugging puede ser vista como un estilo de consolidación de Hansson de la teoría clásica de revisión de creencias. Es por ello que la operación de debugging presentada aquí es caracterizada axiomáticamente a través de los postulados clásicos para consolidación.

Como resultado final de los aportes en esta línea de investigación, propondremos reificar las lógicas abstractas del `GenAF` a la description logic (DL) básica $\mathcal{ALC}$ con un primer objetivo de razonar sobre ontologías inconsistentes. A tal fin, propondremos metodologías de transformación de ontologías $\mathcal{ALC}$ en alguna forma normal para interpretar axiomas $\mathcal{ALC}$ como argumentos del `GenAF` de manera directa. Nuevamente, trabajos de compilación de conocimiento para ontologías $\mathcal{ALC}$ como los de Bienvenu sobre $\mathcal{ALC}$ *prime implicate normal form* (Bie08), son propuestos. Un segundo objetivo que perseguimos es dar con el problema de debugging y reparación de ontologías. Un proceso de debugging restaura la consistencia a una ontología inconsistente, mientras que el proceso de reparación convierte conceptos insatisfacibles a satisfacibles. La principal motivación para aplicar el `GenAF` a ontologías se basa en la necesidad de razonar sobre las inconsistencias en áreas como la medicina y el derecho. La operación de debugging permite obtener una sub-ontología consistente durante el proceso de razonamiento sobre la ontología inconsistente principal a través del `GenAF`.

La segunda contribución principal de esta tesis presentada en el Capítulo 4, consiste en una metodología capaz de provocar la evolución de KBs proposicionales (potencialmente inconsistentes) sin requerir restauración de consistencia. A tal fin, nos basaremos sobre un `AF` con un lenguaje concreto para argumentos: lógica clásica proposicional. Tal `AF` surgirá de la simplificación del `GenAF` para permitir concentrarnos directamente sobre el trabajo de la dinámica del conocimiento. Sin embargo, la metodología propuesta en este capítulo podría ser aplicada en forma directa sobre los marcos generalizados, `GenAF`s, presentados en el Capítulo 3.

Luego propondremos un novedoso enfoque sobre ATC bajo el nombre de *modelo dialéctico-global*, el cual revisa un `AF` construido a partir de una KB por un argumento $\mathcal{R}$ conteniendo un conjunto mínimo de reglas infiriendo su conclusión o *claim* $\alpha$. El resultado de esta revisión es una nueva KB (potencialmente inconsistente) que contendrá la información completa incluida en $\mathcal{R}$. La nueva KB revisada determinará un `AF` cuya semántica argumentativa garantizará $\alpha$ mediante la aceptación de $\mathcal{R}$ a partir del análisis del árbol de dialéctica enraizado en él. Esto será obtenido mediante la identificación de un conjunto de argumentos que deberán ser eliminados del `AF` para permitir que $\mathcal{R}$ logre ser considerado no-derrotado o *undefeated*. Por lo tanto, la búsqueda de aceptación de $\alpha$ puede involucrar no sólo la incorporación de $\mathcal{R}$ al `AF` original, sino también la remoción de otros argumentos. Sin embargo, la remoción de argumentos de un `AF` no puede ser realizada "a la ligera", sino como consecuencia de la remoción de creencias de la KB subyacente de donde el `AF` es construido.

La cantidad de creencias a remover de la KB es controlada por un nuevo criterio de mínimo cambio: tanta información original de la KB como sea posible debe ser mantenida en pos del `AF` revisado. Este será el único criterio de mínimo cambio que adoptaremos para garantizar un comportamiento racional por parte del operador propuesto, y es además la razón principal por la cual el modelo dialéctico-global presentado en esta tesis, constituye el más simple y práctico enfoque a ATC definido hasta el momento.

Considerando el ejemplo dado anteriormente sobre promulgación de leyes, el modelo dialéctico-global podría revisar el sistema legal para incluir un argumento $\mathcal{R}$ en representación de la nueva ley a promulgar. En este caso, los argumentos a ser removidos del `AF` original contendrían diferentes artículos de otras leyes ya promulgadas. De esta forma, se aseguraría la constitucionalidad de la nueva ley mediante la propuesta de derogación, o reforma, de otras leyes lo cual sería logrado removiendo artículos específicos que son parte de argumentos a ser removidos del `AF`. Naturalmente, se espera que la remociones de la base sean de menor importancia que la asociada a la nueva ley. Por ejemplo, las leyes a reformar no deberían corresponder a la Constitución Nacional a menos que la promulgación de la nueva ley se presuma de relevancia suficiente para provocar una reforma constitucional. En el caso en que no surjan leyes menores a ser reformadas, será claro que la nueva ley $\mathcal{R}$ no podrá ser promulgada tal como está planteada y por ello será necesaria su evaluación y posterior modificación para evitar conflictos irrecuperables respecto del sistema legal.

Finalmente, mediante la reificación de la lógica de argumentos del `AF` utilizado a diferentes description logics, logramos aplicar al área de ontologías, el modelo dialéctico-global propuesto –correspondiente a la teoría de cambio argumentativo (ATC)– con el propósito de dar con el problema de evolución de ontogías con tolerancia a inconsistencias, punto central de esta tesis.

## 1.3. Contribuciones

Las contribuciones generales de esta tesis están compuestas por dos teorías diferentes: el `GenAF` introducido en el Capítulo 3 y el *modelo dialéctico-global* de cambio argumentativo, definido en el Capítulo 4. En base al punto de vista teórico, las contribuciones a su vez pueden ser divididas en tres ópticas diferentes, respecto de revisión de creencias, de la argumentación, y de las ontologías.

**(A) respecto de revisión de creencias,**

- **(1)** el aspecto novedoso es la propuesta de modelos de revisión sobre bases inconsistentes sin restauración de consistencia.
  - Los modelos propuestos son axiomáticamente caracterizados de acuerdo a la literatura clásica del área.

**(B) respecto de la argumentación,** se provee

- **(1)** un marco argumentativo generalizado o *generalized abstract argumentation framework* (`GenAF`) para razonar sobre bases inconsistentes definidas como fragmentos de primer orden, y
- **(2)** el *modelo dialéctico-global* de cambio argumentativo para incorporar un nuevo argumento a un framework de forma que sea garantizado por la semántica argumentativa.

**(C) respecto de las ontologías,**

**(1)** el `GenAF` provee una alternativa para razonar sobre ontologías inconsistentes expresadas a través de diferentes description logics (DLs).

- Adicionalmente, se propone un operador de debugging y reparación de inconsistencias ontológicas.

**(2)** el *modelo dialéctico-global* ataca el problema de la evolución ontológica sin restaurar su consistencia, mediante la incorporación de nuevo conocimiento a una ontología potencialmente inconsistente.

- Se propone un algoritmo basado en razonadores DLs estándares para la construcción de argumentos.
- Se analiza la complejidad computacional del algoritmo propuesto.

## 1.4. Publicaciones

Los resultados obtenidos en el Capítulo 3 fueron formados a partir de las publicaciones (MRF08a; MRFS09; MRF10a), donde se presentaron las primeras versiones preliminares del `GenAF`, inspiradas por los marcos argumentativos dinámicos o *dynamic argumentation framework* (`DAF`) presentados en (RMGS08; RMF$^+$08b; RMGS10). Por su parte, las investigaciones y propuestas del `DAF`–contribución central en la tesis (Rot10) de mi colega, el Dr. Nicolás D. Rotstein– también han surgido como resultado indirecto de las contribuciones de esta tesis. El `DAF` no es incluido en ésta disertación, y actualmente una versión completa está bajo revisión para ser enviada a una revista internacional indexada.

Por parte de las contribuciones presentadas en el Capítulo 4, los resultados principales fueron enviados a JIGPAL (the Oxford's Logic Journal of the IGPL) y su reificación al lenguaje de descripción básico $\mathcal{ALC}$, publicado en (MW09; MWF10). El modelo dialéctico-global es uno de los enfoques a ATC. Como resultado de las instancias iniciales de esta tesis, ATC fue definido y presentado mediante diferentes enfoques en (RMF$^+$08b; MRF$^+$08b; RMF$^+$08a; RMS09; MRF$^+$09; MRF$^+$10b). Actualmente un artículo completo en el que se propone ATC para manejar la dinámica del conocimiento en programas rebatibles, se encuentra bajo revisión en TPLP (International Journal of Theory and Practice on Logic Programming).

Otras contribuciones menores surgidas a partir de las investigaciones relacionadas con esta tesis son (MF07; MFS08), y siete artículos cortos presentados en eventos de alcance nacional como son CACIC (Congreso Argentino de Ciencias de la Computación) y WICC (Workshop de Investigadores en Ciencias de la Computación).

# Capítulo 2

# Resumen de Teorías y Nociones Fundamentales

A continuación daremos una muy breve reseña de las teorías y nociones generales que son el fundamento teórico para la correcta comprensión de las propuestas definidas en este trabajo de tesis.

## 2.1. Argumentación Abstracta

Como se mencionó anteriormente, la propuesta de mayor impacto introduciendo un marco argumentativo abstracto fue dada por Dung (Dun95), en un artículo cuyos resultados aún son objeto de análisis en la literatura. Varios autores han tomado esta propuesta para extenderla en diversos sentidos (BCD07b; WBC07), logrando formalismos con una capacidad expresiva de un espectro realmente amplio. La especificación de un sistema argumentativo abstracto implica definir dos conjuntos: el de argumentos y el de ataques entre ellos. Esto permite representar un marco argumentativo como un grafo dirigido, donde los nodos son los argumentos y los arcos, los ataques apuntando al argumento siendo derrotado. Este grafo suele resultar una herramienta útil para visualizar el sistema y analizar propiedades. La finalidad de representar conocimiento en un marco argumentativo es tener la capacidad de obtener conclusiones del sistema a pesar de la potencial inconsistencia del conocimiento. Una semántica aplicada al marco argumentativo determinará el criterio por el cual ciertos argumentos serán garantizados (*i.e.*, aceptados a pesar del conocimiento en contradicción) y otros, rechazados. El conjunto de argumentos garantizados contendrá aquellos argumentos en los cuales se puede creer luego de haber considerado todo el conocimiento relacionado.

A continuación introduciremos las nociones básicas del marco abstracto de Dung.

### 2.1.1. El Marco de Dung

Como mencionamos anteriormente, en un `AF` los argumentos son tratados como piezas indivisibles de información con una estructura interna mantenida en forma abstracta.

**Definición 2.1.1 (Marco Argumentativo Abstracto)** (Dun95) *Un **marco argumentativo abstracto** o abstract argumentation framework (*`AF`*) es una tupla* $\langle \mathbf{A}, \mathbf{R_A} \rangle$*, donde* $\mathbf{A}$ *es un conjunto finito de argumentos, y* $\mathbf{R_A} \subseteq (\mathbf{A} \times \mathbf{A})$*, la relación de ataque sobre* $\mathbf{A}$*.*

Un `AF` es básicamente un grafo dirigido donde los nodos son argumentos de $\mathbf{A}$, y los arcos, los ataques determinados por la relación $\mathbf{R_A}$. La dirección de un arco determina el origen y destino del ataque.

Los `AF` son utilizados para el análisis de la argumentación en su grado de mayor abstracción y simplicidad. Es por ello que los `AF` han sido mayormente utilizados para enfocar las investigaciones en las *semánticas argumentativas*. Muchas extensiones han sido definidas, como el caso de los *marcos value-based* (BC02), *marcos bipolares*(ACLSL08), y *marcos etiquetados* (CS06), entre otros. La incorporación de la noción de *subargumento* en (MGS07), inicia una nueva forma de argumentación en la que cierta estructura interna es dada a los argumentos, sin perder abstracción. Esta idea motivó nuevas extensiones del marco de Dung como por ejemplo el *marco dinámico* o *Dynamic Argumentation Framework* (RMGS10), e incluso, el *marco generalizado* (`GenAF`) presentado luego en el Capítulo 3.

### 2.1.2. Semánticas Argumentativas

En un marco argumentativo, las semánticas permiten obtener conclusiones acerca del conocimiento representado a través de los argumentos y su interrelación mediante el ataque. Tras la aplicación de una semántica argumentativa podemos verificar si un determinado argumento, o un conjunto de ellos, debe ser *aceptado* o *garantizado*.

### Semánticas Estándares de Dung

Las semánticas estánderes de Dung (Dun95) se aplican a un `AF` para obtener un conjunto $\mathbf{S} \subseteq 2^{\mathbf{A}}$ de subconjuntos de $\mathbf{A}$ denominados *extensiones*. Cada extensión es un conjunto de argumentos cuyo requerimiento mínimo es ser *libre de conflicto*.

**Definición 2.1.2 (Libertad de Conflicto y Defensa)** (Dun95; CA07) *Dado un* `AF` $\langle \mathbf{A}, \mathbf{R_A} \rangle$ *y un subconjunto* $\mathbf{S} \subseteq \mathbf{A}$ *de argumentos.*

- $\mathbf{S}$ *es* ***libre de conflicto*** *si y sólo si no existe par* $\mathcal{A}, \mathcal{B} \in \mathbf{S}$ *tal que* $\mathcal{A}\mathbf{R_A}\mathcal{B}$.
- $\mathbf{S}$ ***defiende*** *un argumento* $\mathcal{A} \in \mathbf{A}$ *(*$\mathcal{A}$ *es* ***aceptable*** *con respecto a* $\mathbf{S}$*) si y sólo si para cualquier argumento* $\mathcal{B} \in \mathbf{A}$*, si* $\mathcal{B}\mathbf{R_A}\mathcal{A}$ *entonces existe un argumento* $\mathcal{C} \in \mathbf{S}$ *tal que* $\mathcal{C}\mathbf{R_A}\mathcal{B}$*. Un conjunto de argumentos es aceptable con respecto a* $\mathbf{S}$ *cuando cada uno de sus elementos es aceptable con respecto a* $\mathbf{S}$*. Finalmente,* $\mathbf{S}$ *es llamado* ***propiamente aceptable*** *cuando* $\mathbf{S}$ *es aceptable con respecto a* $\mathbf{S}$.

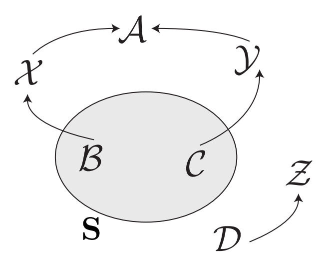



Figura 2.1: Argumento $\mathcal{A}$ aceptable respecto de $\mathbf{S}$.

**Definición 2.1.3 (Conjuntos Admisibles)** (Dun95; CA07) *Dado un* `AF` $\langle \mathbf{A}, \mathbf{R_A} \rangle$*, un subconjunto* $\mathbf{S} \subseteq \mathbf{A}$ *de argumentos es llamado* ***admisible*** *si y sólo si* $\mathbf{S}$ *es propiamente acceptable (i.e., defiende cada argumento contenido en* $\mathbf{S}$*) y es libre de conflicto.*

La expresión "semántica estándares de Dung" es usualmente utilizada para referir a las semánticas *completa*, *grounded*, *preferida*, y *estable*. Además, el término *extensión* puede ser utilizado para referirse a extensiones completa, grounded, preferida, o estable. La importancia de la noción de conjuntos admisibles queda reflejada por el hecho de que cada extensión bajo alguna semántica estándar de Dung es admisible. A continuación se detallan las semánticas estándares de Dung de acuerdo a su introducción original en (Dun95) y su refinamiento en (CA07).

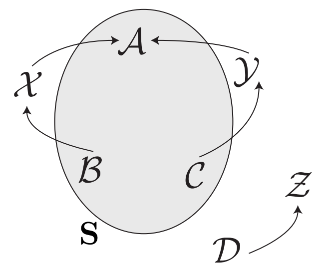



Figura 2.2: Conjunto $\mathbf{S}$ admisible.

**Definición 2.1.4 (Semánticas Estándares de Dung)** (Dun95; CA07) *Dado un* `AF` $\langle \mathbf{A}, \mathbf{R_A} \rangle$*, un conjunto* $\mathbf{S} \subseteq \mathbf{A}$ *de argumentos libre de conflicto, y* $\mathcal{F} : 2^{\mathbf{A}} \longrightarrow 2^{\mathbf{A}}$ *la* ***función característica*** *tal que:*

$$\mathcal{F}(\mathbf{S}) = \{\mathcal{A} \mid \mathcal{A} \in \mathbf{A} \textit{ es un argumento defendido por } \mathbf{S}\}.$$

- $\mathbf{S}$ *es una* ***extensión completa*** *sssi es admisible y se verifica* $\mathbf{S} = \mathcal{F}(\mathbf{S})$*.*
- $\mathbf{S}$ *es una* ***extensión grounded*** *sssi es la mínima extensión completa respecto de inclusión de conjuntos de* $\mathbf{A}$*.*
- $\mathbf{S}$ *es una* ***extensión preferida*** *sssi es la máxima extensión completa respecto de inclusión de conjuntos de* $\mathbf{A}$*.*
- $\mathbf{S}$ *es una* ***extensión estable*** *sssi es una extensión preferida y para cualquier argumento* $\mathcal{A} \in (\mathbf{A} \setminus \mathbf{S})$ *existe un argumento* $\mathcal{B} \in \mathbf{S}$ *tal que* $\mathcal{B}\mathbf{R_A}\mathcal{A}$*.*

Los siguientes ejemplos ilustran las semánticas estándares de Dung.

**Ejemplo 1** (Rot10) *En la Figura 2.3 pueden verse dos* `AF`*s. En el* `AF` *de la figura* (a) *el conjunto de extensiones completas es:* $\{\emptyset, \{\mathcal{A}\}, \{\mathcal{B}\}\}$*, mientras que en el de la figura* (b) *es:* $\{\emptyset, \{\mathcal{A}, \mathcal{C}\}, \{\mathcal{B}\}\}$*.*


(a) (b)

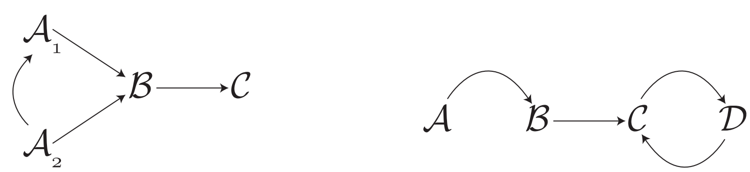
Figura 2.3: Dos `AF`s y sus extensiones completas.

**Ejemplo 2** (Rot10) *En la Figura 2.4 se muestra como la existencia de ciclos puede afectar a las extensiones preferidas. En el* `AF` *de la figura* (a) *el conjunto de extensiones preferidas es:* $\{\{\mathcal{A}_1, \mathcal{C}\}, \{\mathcal{A}_2, \mathcal{C}\}\}$, *mientras que en el de la figura* (b) *es:* $\{\{\mathcal{A}, \mathcal{C}\}, \{\mathcal{A}, \mathcal{D}\}, \{\mathcal{B}, \mathcal{D}\}\}$.


(a) (b)

Figura 2.4: Dos `AF`s y sus extensiones preferidas.

**Ejemplo 3** (Rot10) *En los* `AF`*s de los ejemplos 1 y 2, la extensión grounded es el conjunto vacío, dado que la función característica no encuentra argumentos aceptables con respecto al conjunto vacío, i.e., no derrotados por ningún argumento. Considerando el* `AF` *de la Figura 2.5 en el caso de la figura* (a) *el conjunto de extensiones grounded es:* $\{\{\mathcal{A}_2, \mathcal{C}\}\}$, *mientras que en el de la figura* (b) *es:* $\{\{\mathcal{A}\}\}$.




(a) (b)

Figura 2.5: Dos `AF`s y sus extensiones grounded.

**Lema 2.1.5** (Dun95) *Toda extensión estable es preferida, pero no vice-versa.*

Es claro que cada extensión estable es preferida (ver Definición 2.1.4). Sin embargo, como muestra el lema anterior extraído del trabajo seminal de Dung (Dun95), no siempre es posible que las extensiones preferidas y estables coincidan como sucede en el Ejemplo 2. Para ilustrar el caso, considere el siguiente ejemplo.

**Ejemplo 4** *Dado un* `AF` $\langle \mathbf{A}, \mathbf{R_A} \rangle$ *con* $\mathbf{A} = \{\mathcal{A}\}$ *y* $\mathbf{R_A} = \{(\mathcal{A}, \mathcal{A})\}$. *El conjunto vacío es una extensión preferida del* `AF`, *aunque no estable.*

### Semánticas sobre Árboles de Dialéctica

Una alternativa a las semánticas basadas en grafos son las semánticas basadas en *árboles de dialéctica*. Este tipo de semánticas se dicen "guiadas por consulta" debido a que su construcción parte de una determinada consulta sobre un argumento particular del marco. Por ejemplo, dado el argumento $\mathcal{A}$, para decidir su aceptación se construye un árbol de dialéctica cuya raíz es el mismo argumento consulta, *i.e.*, $\mathcal{A}$. Luego, se analiza el árbol por medio de un criterio de marcado, y finalmente se decide la aceptabilidad del argumento raíz.

Existen diferentes construcciones para árboles de dialéctica como por ejemplo (BH08). En este trabajo de tesis nos basaremos en las utilizadas para la implementación de DeLP (GS04) (sistema de programación en lógica rebatible con razonamiento basado en argumentación). A continuación detallaremos su construcción a partir de un `AF` de Dung, basándonos en las construcciones dadas en (GS04; CS07; RMS09).

La noción de *línea de argumentación* (CS07) identifica un intercambio reiterado de argumentos y contra-argumentos, o derrotadores.

**Definición 2.1.6 (Línea de Argumentación)** *Dado un* `AF` $\langle \mathbf{A}, \mathbf{R_A} \rangle$ *y los argumentos* $\mathcal{B}_1 \ldots, \mathcal{B}_n \in \mathbf{A}$, *una* ***línea de argumentación*** $\lambda$ *es una secuencia no-vacía* $[\mathcal{B}_1 \ldots, \mathcal{B}_n]$ *donde* $\mathcal{B}_i \mathbf{R_A} \mathcal{B}_{i-1}$ $(1 < i \leq n)$*. El argumento* $\mathcal{B}_1$ *es identificado como la* ***raíz*** *de* $\lambda$*, y* $\mathcal{B}_n$*, como su* ***hoja****. La línea* $\lambda$ *se dirá* ***enraizada*** *en* $\mathcal{B}_1$*.*

Una línea de argumentación puede ser vista como "dos partes de una discusión": una a favor de la raíz y la otra argumentando contra ella. En consecuencia, identificamos el *conjunto de argumentos pro (respecto de, con)* de una línea como aquellos argumentos que directa o indirectamente defienden (resp., atacan) la raíz.

**Definición 2.1.7 (Conjunto de Argumentos Con (Pro))** *Dada una línea* $\lambda$*, el conjunto* ***pro*** *(respecto de,* ***con****), notado como* $\lambda^+$ *(resp.,* $\lambda^-$*), contendrá todos los argumentos ubicados en posiciones impares (resp., pares) en* $\lambda$*.*

Se hará abuso de notación para identificar al argumento $\mathcal{B}$ en $\lambda$, escribiendo $\mathcal{B} \in \lambda$. Las llamadas *condiciones de aceptabilidad* o *restricciones dialécticas* son utilizadas para la construcción de líneas libres de *falacias*, llamadas *líneas de argumentación aceptables*.

Ellas pueden varias dependiendo de la especificación del marco argumentativo y la lógica subyacente (si existiera). Por ejemplo, (1) un argumento no debería aparecer dos veces en la misma línea, y (2) el conjunto de argumentos pro (resp., con) en una línea es libre de conflictos. Para mayor detalle sobre las condiciones de aceptabilidad, el lector puede referirse a (GS04). Adicionalmente, una línea aceptable es *exhaustiva* si no puede ser extendida por ningún otro derrotador de su hoja sin comprometer su condición de aceptable. Dado un $\mathtt{AF}$ $\phi = \langle \mathbf{A}, \mathbf{R_A} \rangle$, definimos el dominio de todas las líneas aceptables de $\phi$ como $\mathbf{L}_\phi$, mientras que el dominio de todas las líneas aceptables y exhaustivas de $\phi$ será identificado mediante el conjunto $\mathbb{L}_\phi$.

**Observación 2.1.8** *Dado un* $\mathtt{AF}$ $\phi = \langle \mathbf{A}, \mathbf{R_A} \rangle$, $\mathbb{L}_\phi \subseteq \mathbf{L}_\phi$.

Una secuencia inicial en una línea aceptable podrá ser identificada mediante el uso de la noción de *segmento superior*.

**Definición 2.1.9 (Segmento Superior)** *Dado un* $\mathtt{AF}$ $\phi = \langle \mathbf{A}, \mathbf{R_A} \rangle$, *y una línea aceptable* $\lambda \in \mathbf{L}_\phi$ *tal que* $\lambda = [\mathcal{B}_1, \ldots, \mathcal{B}_n]$, *el* ***segmento superior*** *de* $\lambda$ *con respecto a* $\mathcal{B}_i$ $(1 \leq i \leq n)$ *es definido como* $\lambda^\uparrow[\mathcal{B}_i] = [\mathcal{B}_1, \ldots, \mathcal{B}_i]$, *mientras que el* ***segmento propio superior*** *de* $\lambda$ *con respecto a* $\mathcal{B}_i$ $(i \neq 1)$ *es definido como* $\lambda^\uparrow(\mathcal{B}_i) = [\mathcal{B}_1, \ldots, \mathcal{B}_{i-1}]$.

Nos referiremos simplemente como "segmento superior" para referirnos a cualquier tipo de segmento superior, sea propio o no-propio. Sólo su notación los harán distinguibles: el uso de los paréntesis para un segmento propio superior y de los corchetes para el caso de un segmento (no-propio) superior. Observe que el uso de un segmento superior en una línea constituye una (posiblemente no-exhaustiva) línea por sí misma, *i.e.*, para cualquier $\lambda$ y cualquier $\mathcal{B} \in \lambda$, $\lambda^\uparrow[\mathcal{B}]$ es una línea de argumentación.

**Observación 2.1.10** *Dado un* $\mathtt{AF}$ $\phi = \langle \mathbf{A}, \mathbf{R_A} \rangle$ *y una línea* $\lambda \in \mathbf{L}_\phi$, *para cualquier* $\mathcal{B} \in \lambda$ *se verifica* $\lambda^\uparrow[\mathcal{B}] \in \mathbf{L}_\phi$.

Como fuera mencionado anteriormente, un árbol de dialéctica permite determinar el estado de aceptabilidad del argumento raíz, *i.e.*, si puede ser aceptado o rechazado como una pieza de información racionalmente justificada. Tal árbol es construido a partir de un conjunto de líneas de argumentación enraizadas en un argumento en común. Esta noción es también concordante con la intuiciones consideradas en (PV00).

**Definición 2.1.11 (Árbol de Dialéctica)** *Dado un* $\texttt{AF}$ $\phi = \langle \mathbf{A}, \mathbf{R_A} \rangle$ *y un conjunto* $X \subseteq \mathbf{L}_\phi$ *de líneas aceptables enraizadas en* $\mathcal{R} \in \mathbf{A}$*, el* ***árbol de dialéctica*** $\mathcal{T}(\mathcal{R})$ *determinado por* $X$ *es construido de forma que un argumento* $\mathcal{C}$ *en* $\mathcal{T}(\mathcal{R})$ *es:*

1. *un* ***nodo*** *sssi* $\mathcal{C} \in \lambda$*, para cualquier* $\lambda \in X$*; ó*
2. *un* ***hijo*** *de un nodo* $\mathcal{B}$ *en* $\mathcal{T}(\mathcal{R})$ *sssi*
   a) $\mathcal{C} \in \lambda$, $\mathcal{B} \in \lambda'$*, para cualquier* $\{\lambda, \lambda'\} \subseteq X$*, y*
   b) $\lambda'^{\uparrow}[\mathcal{B}] = \lambda^{\uparrow}(\mathcal{C})$.

*Diremos que* $\mathcal{T}(\mathcal{R})$ *está* ***enraizado*** *en* $\mathcal{R}$*. Las* ***hojas*** *en* $\mathcal{T}(\mathcal{R})$ *son las hojas de cada línea contenida en* $X$*. El dominio de todos los árboles de dialéctica a construir a partir de* $\phi$ *será identificado mediante el conjunto* $\mathbf{T}_\phi$.

**Ejemplo 5** *Dado el conjunto de líneas* $X$ *enraizadas en* $\mathcal{R}$ *determinando el árbol* $\mathcal{T}(\mathcal{R})$ *(ver Figura 2.6) tal que* $X = \{\lambda_1, \lambda_2, \lambda_3\}$*, donde* $\lambda_1 = [\mathcal{R}, \mathcal{B}_1, \mathcal{B}_6]$*,* $\lambda_2 = [\mathcal{R}, \mathcal{B}_1, \mathcal{B}_2]$*, y* $\lambda_3 = [\mathcal{R}, \mathcal{B}_3, \mathcal{B}_4, \mathcal{B}_5]$*, son tres líneas aceptables y exhaustivas. Observe que el argumento* $\mathcal{B}_2$ *es un hijo de* $\mathcal{B}_1$ *en* $\mathcal{T}(\mathcal{R})$ *dado que* $\lambda_1^{\uparrow}[\mathcal{B}_1] = [\mathcal{R}, \mathcal{B}_1] = \lambda_2^{\uparrow}(\mathcal{B}_2)$ *(ver Definición 2.1.11).*

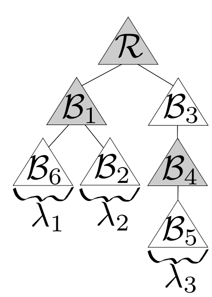



Figura 2.6: Árbol de Dialéctica – Ejemplo 5

La condición de aceptabilidad para árboles de dialéctica requiere que sean utilizadas todas las líneas exhaustivas enraizadas en el mismo argumento $\mathcal{R}$ para su construcción. Por tal motivo definimos el llamado *conjunto bundle* para $\mathcal{R}$.

**Definición 2.1.12 (Conjunto Bundle)** *Dado un* $\texttt{AF}$ $\phi = \langle \mathbf{A}, \mathbf{R_A} \rangle$*, el* ***conjunto bundle para*** $\mathcal{R}$ *es el conjunto* $\mathcal{S}(\mathcal{R}) \subseteq \mathbb{L}_\phi$ *conteniendo todas las líneas aceptables y exhaustivas enraizadas en* $\mathcal{R}$ *que pueden ser construidas a partir de* $\mathbf{A}$ *y* $\mathbf{R_A}$.

**Definición 2.1.13 (Árbol de Dialéctica Aceptable)** *Dado un* $\mathtt{AF}$ $\phi = \langle \mathbf{A}, \mathbf{R_A} \rangle$*, un árbol de dialéctica* $\mathcal{T}(\mathcal{R}) \in \mathbf{T}_\phi$ *es* ***acceptable*** *si es construido a partir del conjunto bundle* $\mathcal{S}(\mathcal{R}) \subseteq \mathbb{L}_\phi$*. El dominio de todos los árboles de dialéctica aceptables de* $\phi$ *será identificado mediante el conjunto* $\mathbb{T}_\phi$*.*

**Observación 2.1.14** *Dado un* $\mathtt{AF}$ $\phi = \langle \mathbf{A}, \mathbf{R_A} \rangle$ *y un argumento* $\mathcal{R} \in \mathbf{A}$*, el conjunto bundle* $\mathcal{S}(\mathcal{R}) \subseteq \mathbb{L}_\phi$ *es único y, por lo tanto, también lo es el árbol aceptable* $\mathcal{T}(\mathcal{R}) \in \mathbb{T}_\phi$ *determinado por* $\mathcal{S}(\mathcal{R})$*.*

Haremos abuso de notación nuevamente para identificar una línea $\lambda \in \mathbf{L}_\phi$ correspondiente a un árbol de dialéctica $\mathcal{T}(\mathcal{R}) \in \mathbf{T}_\phi$, escribiendo $\lambda \in \mathcal{T}(\mathcal{R})$. Por ejemplo, en el Ejemplo 5, $\lambda_1 \in \mathcal{T}(\mathcal{R})$ es verificado.

Dado un $\mathtt{AF}$ $\phi = \langle \mathbf{A}, \mathbf{R_A} \rangle$ y el argumento $\mathcal{R} \in \mathbf{A}$ cuya aceptabilidad es consultada, $\mathcal{R}$ es aceptado (garantizado) por la semántica argumentativa mediante el análisis del árbol de dialéctica aceptable $\mathcal{T}(\mathcal{R}) \in \mathbb{T}_\phi$. Esta evaluación es obtenida pesando la información en el árbol a través de una *función de marcado* $\mathfrak{mark} : \mathbf{A} \times \mathbf{L}_\Sigma \times \mathbf{T}_\Sigma \longrightarrow \mathbb{M}$, la cual define un criterio de aceptación asignando a cada argumento en $\mathcal{T}(\mathcal{R})$ una determinada marca del dominio $\mathbb{M} = \{D, U\}$, donde las marcas $D$ y $U$ identifican a un argumento *derrotado* o *no-derrotado*, respectivamente.

La marca de un nodo interno en el árbol es identificada mediante las marcas de sus hijos, *i.e.*, sus derrotadores, siguiendo algún criterio de marcado específico. Como se mencionó anteriormente, el criterio utilizado en esta tesis se corresponde con aquel utilizado para la implementación de DeLP. Este criterio es referido como escéptico, y definido de forma tal que[1]:

1. todas las hojas son marcadas $U$ y
2. todo nodo interno $\mathcal{B}$ es marcado como $U$ *sssi* todo hijo de $\mathcal{B}$ es marcado como $D$, de lo contrario, $\mathcal{B}$ es marcado como $D$.

A continuación introducimos la definición formal del criterio de marcado.

[1]Para mayor detalle de éste y otros tipos de marcados, el lector puede referirse a (RMS09).

**Definición 2.1.15 (Criterio de Marcado)** *Dado un* AF $\phi = \langle \mathbf{A}, \mathbf{R_A} \rangle$ *y un árbol* $\mathcal{T}(\mathcal{R}) \in \mathbb{T}_\Sigma$*, para cualquier línea* $\lambda \in \mathcal{T}(\mathcal{R})$ *y cualquier* $\mathcal{B} \in \lambda$*:*

$$\mathfrak{mark}(\mathcal{B}, \lambda, \mathcal{T}(\mathcal{R})) = \begin{cases} U & \textit{si } \mathcal{B} \textit{ es la hoja de } \lambda, \textit{ ó} \\ U & \textit{si } \mathfrak{mark}(\mathcal{C}, \lambda', \mathcal{T}(\mathcal{R})) = D, \textit{ para toda } \lambda' \in \mathcal{T}(\mathcal{R}) \\ & \textit{y todo } \mathcal{C} \in \lambda' \textit{ tal que } \lambda^{\uparrow}[\mathcal{B}] = \lambda'^{\uparrow}(\mathcal{C}), \textit{ ó} \\ D & \textit{en caso contrario.} \end{cases}$$

La definición anterior muestra que $\mathcal{B}$ es *no-derrotado* (marcado como $U$) si el derrotador $\mathcal{C}$ de $\mathcal{B}$ en cada línea del árbol es *derrotado* (marca $D$), ó $\mathcal{B}$ es la hoja en $\lambda$. De lo contrario, cuando el derrotador $\mathcal{C}$ de $\mathcal{B}$ en alguna línea del árbol es no-derrotado, el argumento $\mathcal{B}$ concluye derrotado en $\lambda$.

La *función garantizante* $\mathfrak{warrant} : \mathbf{T}_\phi \longrightarrow \{true, false\}$ determina la aceptación de la raíz del árbol verificando $\mathfrak{warrant}(\mathcal{T}(\mathcal{R})) = true$ *sssi* $\mathfrak{mark}(\mathcal{R}, \lambda, \mathcal{T}(\mathcal{R})) = U$. Por lo tanto, $\mathcal{R} \in \mathbf{A}$ es *garantizado* por $\mathcal{T}(\mathcal{R}) \in \mathbb{T}_\phi$ *sssi* $\mathfrak{warrant}(\mathcal{T}(\mathcal{R})) = true$. En tal caso, $\mathcal{T}(\mathcal{R})$ es referido como *árbol garantizante*. Estas nociones son gráficamente ilustradas con argumentos pintados en gris y blanco en representación de marcas $D$ y $U$, respectivamente. En el Ejemplo 5, $\mathcal{R}$ es derrotado y, por lo tanto, $\mathcal{T}(\mathcal{R})$ es no-garantizante.

### 2.1.3. Subargumentos

Normalmente, los nuevos marcos argumentativos que resultan de extender el clásico AF tienen como objetivo ampliar las capacidades de representación del conocimiento. Una de las adiciones más significativas para el curso de esta tesis es la introducción de la noción de *subargumento*. Su intuición puede describirse como "una porción de un argumento que es un argumento por sí misma". Los subargumentos son utilizados para descomponer argumentos en porciones más pequeñas. Dado que ellas a su vez, son argumentos válidos, heredan las mismas capacidades de la noción de argumento en un AF clásico, como por ejemplo, los ataques.

**Definición 2.1.16 (AF Extendido)** (MGS07) *Un* ***marco argumentativo abstracto*** *extendido con subargumentos es una tupla* $\langle \mathbf{A}, \mathbf{R_A}, \trianglelefteq \rangle$*, donde* $\mathbf{A}$ *es un conjunto finito de argumentos,* $\mathbf{R_A} \subseteq (\mathbf{A} \times \mathbf{A})$*, la relación de ataque sobre* $\mathbf{A}$*, y* $\trianglelefteq \subseteq (\mathbf{A} \times \mathbf{A})$*, la relación de subargumentos sobre* $\mathbf{A}$*.*

**Ejemplo 6** *Dado un* `AF` *extendido de acuerdo a la Definición 2.1.16 tal que* $\langle \mathbf{A}, \mathbf{R_A}, \trianglelefteq \rangle$, *donde* $\mathbf{A} = \{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4, \mathcal{A}_5, \mathcal{A}_6, \mathcal{A}_7\}$, $\mathbf{R_A} = \{(\mathcal{A}_1, \mathcal{A}_2), (\mathcal{A}_1, \mathcal{A}_5), (\mathcal{A}_6, \mathcal{A}_3), (\mathcal{A}_7, \mathcal{A}_2), (\mathcal{A}_7, \mathcal{A}_5)\}$, *y* $\trianglelefteq = \{(\mathcal{A}_3, \mathcal{A}_5), (\mathcal{A}_4, \mathcal{A}_5), (\mathcal{A}_6, \mathcal{A}_7)\}$. *La Figura 2.7 ilustra el grafo resultante.*

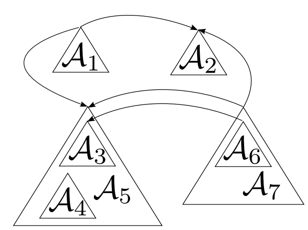



Figura 2.7: Grafo resultande del `AF` extendido con subargumentos.

El uso de subargumentos permite trabajar con un nivel de granularidad más fino. En el transcurso de esta disertación veremos que esta noción es esencial para el manejo apropiado de marcos argumentativos en representación de bases de conocimientos subyacentes. Por otro lado, la construcción de árboles de dialéctica ante el uso de subargumentos requiere una especificación definida de la construcción de las líneas de argumentación aceptables. Esto será apropiadamente definido en el Capítulo 4.

## 2.2. Revisión de Creencias

El área de revisión de creencias o *belief revision* estudia el proceso de cambio de creencias en un estado epistémico para aceptar una nueva pieza de información. Un *estado epistémico* –al cual se aplican las operaciones de cambio– generalmente representa su conocimiento en dos formas posibles: conjuntos de creencias o bases de creencias. Una *base de creencias*, llamada también *base de conocimiento* o *knowledge base* (KB), es un estado epistémico representado por un conjunto de sentencias no necesariamente cerrado bajo consecuencia lógica. Por otro lado, un *conjunto de creencias* o *belief set* es un conjunto de sentencias cerrado bajo consecuencia lógica. En general, un conjunto de creencias es infinito, siendo ésta la principal razón por la cual se hace inviable la representación real de este tipo de conjuntos en la práctica. En su lugar, es posible caracterizar propiedades que cada una de las operaciones de cambio deberían satisfacer en cualquier representación finita de un estado epistémico.

### 2.2.1. El Modelo AGM

El modelo AGM (AGM85) (la sigla aparece por las iniciales de sus creadores Alchourrón, Gärdenfors, and Makinson) es hasta el momento el modelo estándar dentro del área de revisión de creencias. Para una exposición detallada, el lector puede referirse a (Gä88; Han99; GR95). El modelo original AGM es una teoría acerca de como agentes de racionalidad idealizada deberían revisar sus creencias al momento de recibir nueva información. Los agentes son idealizados en lo que se asumen con memoria infinita y habilidad de inferencia. Ellos pueden inmediatamente cerrar un conjunto bajo consecuencia lógica y mantener un conjunto de creencias –usualmente infinito. Las creencias son representadas por fórmulas de un lenguaje proposicional y la lógica subyacente a cada agente para razonar, es clásica. Al representar los estados epistémicos mediante el uso de conjuntos de creencias, esto implica, el manejo de conjuntos de fórmulas $K$ tal que $Cn(K) = K$, donde $Cn$ es el operador de clausura por consecuencia lógica.

El modelo AGM clásico puede ser visto como una teoría que modela sólo los cambios ocurriendo en conjuntos de creencias debido a la nueva información acerca del mundo estático, *i.e.*, el mundo no cambia, sólo la información disponible para el agente acerca del mundo es la que se modifica. No se modelan todos los cambios del estado epistémico de un agente, sino sólo aquellos que afectan al conjunto de creencias.

Las operaciones de cambio tal como son introducidas en el modelo AGM clásico son conocidas como expansiones, contracciones, y revisiones. Una *expansión* incorpora una nueva creencia sin garantizar la consistencia del estado epistémico resultante. Una *contracción* elimina una creencia $\alpha$ y otras que hacen posible su inferencia, del estado epistémico. Finalmente, una *revisión* incorpora una nueva creencia $\alpha$ al estado epistémico garantizando un resultado consistente siempre que $\alpha$ sea también consistente. Esto implica que la revisión incluye una nueva creencia y posiblemente elimine otras a fin de evitar inconsistencias. Las revisiones son usualmente identificadas mediante contracciones y expansiones: assumiendo un operador de revisión "$*$", una nueva creencia $\alpha$, y un estado epistémico $K$; el estado epistémico resultante $K * \alpha$ es asegurado consistente (a menos que $\alpha$ sea inconsistente) a través de una contracción "$-$" por el complemento de la nueva creencia (*i.e.*, $\sim\alpha$) y una expansión "$+$" por $\alpha$. Esta composición de sub-operacions define un operador de revisión mediante la *identidad de Levi:* (Lev77; Gä81).

**(Identidad de Levi)** $K * \alpha = (K - \sim\alpha) + \alpha$

En (Han93) la versión reversa de esta identidad fue estudiada, es decir: $K * \alpha = (K + \alpha) - {\sim}\alpha$. Esta identidad será útil luego en la introducción del modelo de cambio argumentativo definido en el Capítulo 4.

Análogamente, una operación de contracción puede ser definida en términos de la operación de revisión, de acuerdo con la *identidad de Harper*:

**(Identidad de Harper)** $K - \alpha = (K * {\sim}\alpha) \cap K$

### Postulados de Racionalidad

De las tres operaciones AGM, sólo las expansiones son caracterizadas en una única forma. Cuando un conjunto de creencias $K$ es expandido por una sentencia $\alpha$, el resultante conjunto $K + \alpha$ es obtenido simplemente mediante la incorporación de la nueva creencia al viejo conjunto y tomando las consecuencias lógicas del conjunto resultante:

$$K + \alpha = Cn(K \cup \{\alpha\})$$

El nombre de expansión fue justificado por el hecho de que $K \subseteq K + \alpha$.

Las operaciones de contracción y revisión no fueron definidas en forma directa, sino restringidas o caracterizadas por un conjunto básico de postulados de racionalidad. Para la contracción de un conjunto de creencias $K$ por una sentencia $\alpha$, *i.e.*, $K-\alpha$, seis postulados básicos fueron dados en (AGM85)[2].

**(K-1)** $K - \alpha$ es un conjunto de creencias (*clausura*)
**(K-2)** $K - \alpha \subseteq K$ (*inclusión*)
**(K-3)** Si $\alpha \notin K$, entonces $K - \alpha = K$ (*vacuidad*)
**(K-4)** Si $\nvdash \alpha$, entonces $\alpha \notin K - \alpha$ (*success* o *éxito*)
**(K-5)** $K \subseteq (K - \alpha) + \alpha$ (*recovery* o *recuperación*)
**(K-6)** Si $\vdash \alpha \leftrightarrow \beta$, entonces $K - \alpha = K - \beta$ (*equivalencia*)

El postulado **(K-1)** asegura que el resultado de la contracción de un conjunto de creencias por una sentencia debe ser un nuevo conjunto de creencias. Por su parte, **(K-2)**

[2] El operador $\vdash$ es utilizado como la relación de consecuencia asociada a $Cn$.

asegura que en una contracción no aparecerán nuevas fórmulas en el conjunto resultante. **(K-3)** asegura que si la sentencia a contraer no pertenece al conjunto entonces no habrá nada por contraer. **(K-4)** dice que a menos que la sentencia a ser contraída sea una tautología, no debería ser un elemento del conjunto resultante. El postulado de recovery **(K-5)** (es el más controversial de acuerdo con Makinson (Mak87)), asegura que una contracción sea recuperable, es decir, que el conjunto original puede ser recuperado mediante la expansión de la sentencia que fuera contraída. Finalmente, el postulado **(K-6)** asegura que las contracciones por sentencias lógicamente equivalentes resultan en iguales conjuntos.

Para la operación de revisión, los siguientes seis postulados básicos fueron introducidos en (Gä88), dando los mismos nombres a postulados que persiguen principios relacionados.

**(K*1)** $K * \alpha$ es un conjunto de creencias (*clausura*)
**(K*2)** $\alpha \in K * \alpha$ (*success* o *éxito*)
**(K*3)** $K * \alpha \subseteq K + \alpha$ (*inclusión*)
**(K*4)** Si $\sim\alpha \notin K$, entonces $K + \alpha \subseteq K * \alpha$ (*preservación*)
**(K*5)** $K * \alpha \vdash \perp$ si y sólo si $\vdash \sim\alpha$ (*consistencia*)
**(K*6)** Si $\vdash \alpha \leftrightarrow \beta$, entonces $K * \alpha = K * \beta$ (*equivalencia*)

El postulado **(K*1)** asegura que el resultado de la revisión de un conjunto de creencias debe ser un nuevo conjunto de creencias. **(K*2)** asegura que la sentencia por la que un conjunto es revisado debe pertenecer al conjunto revisado resultante. **(K*3)** asegura que ninguna información más allá de la sentencia incorporada y sus consecuencias lógicas serán agregadas. **(K*4)** en conjunción con **(K*3)** asegura que nada será retirado del conjunto y, por lo tanto, la revisión equivaldría a una expansión. El postulado **(K*5)** asegura que a menos que la nueva sentencia sea inconsistente, el conjunto revisado resultante debe ser consistente. Finalmente, el postulado **(K*6)** asegura que las revisiones por sentencias lógicamente equivalentes resultan en iguales conjuntos.

Siguiendo las identidades de Levi y Harper, Gärgenfors mostró la relación entre revisiones y contracciones respecto de los postulados básicos.

**Teorema 2.2.1** (Gä88) *Si "$-$" es una operación de contracción garantizando* **(K-1)**...**(K-4)** *y* **(K-6)**, *entonces su revisión asociada satisface* **(K*1)**...**(K*6)**.

Nótese que el postulado de recovery no es necesario. Las operaciones de contracción que satisfacen **(K-1)**...**(K-4)** y **(K-6)** pero no **(K-5)**, fueron llamadas por Makinson como *operaciones de retractación* o *withdrawal operations* (Mak87).

**Teorema 2.2.2** (Gä88) *Si "$*$" es una operación de revisión garantizando* **(K*1)**...**(K*6)***, entonces su contracción asociada satisface* **(K-1)**...**(K-6)***.*

### 2.2.2. Modelos de Hansson sobre Conjuntos Kernel

El modelo AGM especifica un conjunto de elementos teóricos para manejar el cambio de conjuntos de creencias. Sin embargo, dado que en este trabajo de tesis el objetivo es proveer métodos de cambio para KBs y en particular para ontologías –los cuales pueden ser vistos como diferentes tipos de bases de creencias– nos basaremos en los modelos de Hansson sobre *conjuntos Kernel* (Han94). Dichos modelos fueron propuestos para manejar enfoques prácticos dentro de la teoría de revisión de creencias. Un *conjunto kernel* es un conjunto minimal de creencias infiriendo $\alpha$ (llamado, $\alpha$-*kernel*) del estado epistémico.

**Definición 2.2.3 (Conjunto de Kernels)** (Han94) *Sea $K$ una base de creencias y $\alpha$ una sentencia. El conjunto $K^{\perp\!\!\!\perp}\alpha$, llamado* ***conjunto de kernels*** *es el conjunto de conjuntos $K'$ tal que,*

*(1)* $K' \subseteq K$,

*(2)* $K' \vdash \alpha$, *y*

*(3) si* $K'' \subset K'$ *entonces* $K'' \nvdash \alpha$.

*El conjunto $K^{\perp\!\!\!\perp}\alpha$ es también llamado* ***conjunto de $\alpha$-kernels*** *y cada uno de sus elementos $K'$, es identificado como un* ***$\alpha$-kernel****.*

#### Contracciones Kernel

La *contracción kernel* (Han94) –aplicable tanto a conjuntos como bases de creencias– es una generalización de las *contracciones seguras* o *safe contractions* (AM85). La idea tras las contracciones kernel es remover de la base al menos un elemento de cada $\alpha$-kernel de $K$ para finalmente evitar la inferencia de $\alpha$ (excepto para el caso en que $\alpha$ es una tautología). Los elementos a ser eliminados son seleccionados por una *función de incisión*.

**Definición 2.2.4 (Función de Incisión)** (Han94) *Sea $K$ una base y "$\sigma$" una **función de incisión** tal que, para cualquier sentencia $\alpha$, vale,*

*(1) $\sigma(K^{\perp\!\!\perp}\alpha) \subseteq \bigcup(K^{\perp\!\!\perp}\alpha)$, y*

*(2) Si $K' \in K^{\perp\!\!\perp}\alpha$ y $K' \neq \emptyset$ entonces $K' \cap \sigma(K^{\perp\!\!\perp}\alpha) \neq \emptyset$.*

La base contraída contendrá todas las sentencias de la base original $K$, que hayan permanecido fuera del alcance de $\sigma$. En consecuencia, la operación de contracción kernel eliminará de $K$ todas aquellas sentencias que fueran mapeadas por la función de incisión.

**Definición 2.2.5 (Contracción Kernel)** (Han94) *Sea $K$ una base, $\alpha$ una sentencia, y $K^{\perp\!\!\perp}\alpha$ el conjunto de $\alpha$-kernels de $K$. Sea "$\sigma$" una función de incisión para $K$. El operador "$-_\sigma$", llamado **contracción kernel** determinado por "$\sigma$", es definido como, $K -_\sigma \alpha = K \backslash \sigma(K^{\perp\!\!\perp}\alpha)$.*

*Finalmente, un operador "$-$" es una contracción kernel para $K$ si y sólo si existe una función de incisión "$\sigma$" tal que $K - \alpha = K -_\sigma \alpha$ para toda sentencia $\alpha$.*

Para la caracterización axiomática propuesta en (Han94) para las contracciones kernel para bases, Hansson propuso dos nuevos postulados:

- Si $\beta \in K \setminus (K - \alpha)$, entonces existe algún $K' \subseteq K$ tal que $K' \nvdash \alpha$ y $K' \cup \{\beta\} \vdash \alpha$ (*core-retainment* o *retención de núcleo*)
- Si para todo $K' \subseteq K$, $K' \vdash \alpha$ si y sólo si $K' \vdash \beta$, entonces $K - \alpha = K - \beta$ (*uniformidad*)

**Teorema 2.2.6** (Han94) *Dada una base $K$ y una sentencia $\alpha$, el operador "$-$" es una contracción kernel determinada por alguna función de incision "$\sigma$" si y sólo si $K - \alpha$ satisface los postulados de éxito, inclusión, retención de núcleo y uniformidad.*

### Revisiones Kernel

La operación clásica de revisión AGM (desarrollada en principio para conjuntos de creencias) puede ser reducida a contracción por $\sim\alpha$ seguida de una expansión por $\alpha$. En (Han93) esto fue referido como *revisión interna*. Una alternativa para bases de creencias fue propuesta como *revisión externa* en la que primeramente se expande la base por $\alpha$ y luego se contrae por $\sim\alpha$. Este tipo de revisión no puede ser aplicada a conjuntos de creencias dado que si un conjunto que contiene $\sim\alpha$ fuera expandido por $\alpha$, entonces su conjunto resultante sería igual al lenguaje completo, y consecuentemente toda distinción se perdería.

Obsérvese que mediante la identidad de Levi, asumiendo "$-$" como un operador de *contracción kernel* determinado por una función de incisión "$\sigma$", es posible definir una operación de *revisión kernel interna*.

**Definición 2.2.7 (Revisión Kernel Interna)** *Sea "$-$" una contracción kernel para una base $K$. La* ***revisión kernel interna*** *es definida como* $K \mp_\sigma \alpha = (K - \sim\alpha) + \alpha$.

Mediante la aplicación de la versión reversa de la identidad de Levi (Han93) es posible definir una operación de *revisión kernel externa*.

**Definición 2.2.8 (Revisión Kernel Externa)** *Sea "$-$" una contracción kernel para una base $K$. La* ***revisión kernel externa*** *es definida como* $K \pm_\sigma \alpha = (K + \alpha) - \sim\alpha$.

Nótese que en el caso de la revisión kernel externa, un potencial estado intermedio inconsistente puede ser alcanzado. Finalmente, un operador de revisión kernel "$*$" puede ser definido tanto en forma interna, mediante "$\mp_\sigma$", así como externa, mediante "$\pm_\sigma$".

Las operaciones de revisión externa e interna modelan dos tipos diferentes de cambio. No siempre es claro cual operación es más adecuada para formalizar un ejemplo particular. Levi defiende su expansión determinando un estado intermedio de inconsistencia seguido de la contracción, *i.e.*, revisiones externas como las operaciones naturales (Lev91). Hansson, por su parte, opina que algunas veces una operación es preferida sobre la otra (Han99). Según su apreciación, la diferencia principal subyace en la alternativa de aceptar o no en forma inmediata la nueva pieza de información. En los casos en que el agente se asegura su aceptación inmediata, la revisión externa parece ser más apropiada. En cambio, si el agente debe razonar acerca del cambio, entonces la revisión interna debería ser utilizada.

### Consolidaciones Kernel

En (Han97) fue introducida una operación de cambio adicional sobre bases: las *consolidaciones*, identificadas por el operador "!", cuyo objetivo es restaurar la consistencia de la base. Las consolidaciones pueden ser modeladas mediante una contracción por falso. La idea tras las *consolidaciones kernel* es "romper" todo kernel inconsistente, *i.e.*, $\bot$-kernel, obteniendo así una base consistente.

**Definición 2.2.9 (Consolidación Kernel)** (Han97) *Sea "$\sigma$" una función de incisión, una operación "$!_\sigma$" es una **consolidación kernel** para $K$ determinada por $\sigma$ si y sólo si $K!_\sigma = K \setminus \sigma(K^{\perp\!\!\perp}\bot)$.*

Para su caracterización, Hansson introduce los siguientes postulados, asumiendo "!" es una operación de consolidación:

- $K! \subseteq K$ (*inclusión*)
- $K!$ es consistente (*consistencia*)
- Si $\alpha \in K$ y $\alpha \notin K!$ entonces existe algún $K' \subseteq K$ tal que $K'$ es consistente pero $K' \cup \{\alpha\}$ es inconsistente (*retención de núcleo*)

**Teorema 2.2.10** (Han97) *Una operación "!" es una consolidación kernel para una base $K$ si y sólo si $K!$ satisface inclusión, consistencia, y retención de núcleo.*

## 2.2.3. Revisión de Creencias No-priorizada

Una de las convenciones adoptadas por el modelo AGM es que la nueva información recibe siempre la más alta prioridad. La revisión de creencias no-priorizada, contradice este principio, permitiendo que la nueva información sea rechazada dependiendo del caso. La operación de revisión AGM es aplicada sólo luego de que el agente decide aceptar la nueva pieza de información. Asumiendo al agente como un razonador perfecto, si éste decide rechazar la nueva información entonces nada queda por modificar en el conjunto de creencias. De acuerdo con (Lev91), en (Gä88) nunca se analiza por qué razón un agente debe incorporar una nueva pieza de información a su conjunto de creencias. Siguiendo esta línea de pensamiento, la idea más simple para un operador de revisión no-priorizada es la de analizar la aceptabilidad de la nueva información para aplicar en el caso positivo, una revisión AGM. Esta es la idea de la operación de *revisión monitoreada* o *screened revision* propuesta por Makinson en (Mak97).

### Semi-revisiones Kernel

En (Han97), además de las consolidaciones kernel, también fue introducida otra operación de cambio adicional sobre bases: las *semi-revisiones*, identificadas por el operador "?", cuyo objetivo es proponer una operación de revisión no-priorizada. Esto significa que la nueva información por la cual se revisa la base puede ser tanto aceptada como rechazada. Análogamente al modelo AGM, en el que las revisiones pueden ser definidas en términos de contracciones y expansiones, las semi-revisiones pueden se definidas en términos de consolidaciones y expansiones mediante la siguiente identidad (Han97):

$$K?\alpha = (K+\alpha)!$$

**Definición 2.2.11 (Semi-revisión Kernel)** (Han97) *Sea "$\sigma$" una función de incisión, $K$ una base, y $\alpha$ una sentencia, una operación "$?_\sigma$" es una* ***semi-revisión kernel*** *determinada por $\sigma$ si y sólo si $K?_\sigma\alpha = (K\cup\{\alpha\})\setminus\sigma((K\cup\{\alpha\})^{\perp\!\!\perp}\bot)$.*

Para su caracterización, Hansson introduce los siguientes postulados, asumiendo "?" es una operación de semi-revisión:

- $K?\alpha \subseteq K\cup\{\alpha\}$ (*inclusión*)
- $K?\alpha$ es consistente (*consistencia*)
- Si $\beta\in K$ y $\beta\notin K?\alpha$ entonces existe algún $K'\subseteq K\cup\{\alpha\}$ tal que $K'$ es consistente pero $K'\cup\{\beta\}$ es inconsistente (*retención de núcleo*)
- $(K+\alpha)?\alpha = K?\alpha$ (*pre-expansión*)
- Si $\alpha,\beta\in K$, entonces $K?\alpha = K?\beta$ (*cambio interno*)

**Teorema 2.2.12** (Han97) *Una operación "?" es una semi-revisión kernel para una base $K$ por una sentencia $\alpha$ si y sólo si $K?\alpha$ satisface inclusión, consistencia, retención de núcleo, pre-expansión, y cambio interno.*

## 2.3. Cambio Ontológico

Uno de los conceptos más importantes a ser referidos en esta disertación es el de *ontología*. El término se ha utilizado para referir a un amplio conjunto de representaciones formales, incluyendo taxonomías, vocabularios terminológicos jerárquicos, o teorías lógicas detalladas describiendo un dominio (NK04). Por esta razón, una definición precisa del término es dificultosa. Una definición que es usual en la literatura fue dada en (Gru93) donde una ontología fue definida como "*una especificación explícita de una conceptualización de un dominio compartido*".

El término cambio ontológico (*ontology change*) es utilizado en un amplio sentido para cubrir todos los aspectos de la modificación de ontologías, así como aquellos problemas indirectamente relacionados a la operación de cambio tal como el mantenimiento de versiones de una ontología o la traducción de información ontológica a una terminología común. En resumen, el término cambio ontológico es utilizado para referir al problema de decidir que modificaciones desarrollar sobre una ontología en respuesta a ciertas necesidades de cambio así como a la implementación de estas modificaciones y al manejo de sus efectos sobre datos dependientes, servicios, aplicaciones, agentes, entre otros elementos.

La decisión sobre las modificaciones puede ser desarrollada en forma automática, semiatomática, o manual; la implementación de las modificaciones pueden (pero no necesariamente) mantener una copia de la ontología original. Los cambios sobre una ontología pueden tomar diferentes formas, incluyendo el descubrimiento de nueva información (que podría ser datos de instancia o *instance data*, otra ontología, una nueva observación, etc.), un cambio en el foco o el punto de vista de la conceptualización, información recibida de alguna fuente externa, un cambio en el dominio (un cambio dinámico en el mundo modelado), necesidades de comunicación entre fuentes heterogéneas de información, la fusión de información de diferentes ontologías, entre otros.

Estas nociones son cubiertas por diferentes áreas de investigación relacionadas, estudiadas en forma puntual e individual, en la literatura. En esta tesis, identificaremos sólo tres de tales áreas: *evolución ontológica*, *debugging ontológico* y *reparación ontológica.* Más allá de sus diferencias, estas áreas están estrechamente relacionadas. Tal es así que varios artículos, trabajos y sistemas abordan más de uno de estos problemas. En esta tesis trataremos las tres áreas para proveer metodologías de restauración de consistencia en ontologías (debugging y reparación ontológica), y de manejo de la dinámica del conocimiento ontológico (evolución ontológica).

### 2.3.1. Evolución Ontológica

En esta sección intentaremos contestar en forma intuitiva algunas de las preguntas típicas que surgen como, "¿qué significa evolución ontológica?", "¿bajo qué circunstancias evolucionan las ontologías?", "¿deberíamos alguna vez permitir la pérdida de conocimiento?", y en tal caso, "¿deberíamos remover conocimiento asercional (dato de instancia) o terminológico (esquemático), en primera instancia?".

Como fue mencionado en (Gru93) una *ontología* es *"una especificación explícita de una conceptualización de un dominio compartido"*, donde *conceptualización* hace referencia a *"una visión abstracta y simplificada del mundo que deseamos representar por algún motivo"*. Siguiendo esta definición, tres diferentes contextos aparecen en los cuales diferentes cambios pueden ser aplicados (NK04):

1. *Cambios en el Dominio Compartido:* similares a los cambios en los esquemas de bases de datos afectados por algún cambio en el mundo real (también referido como evolución del dominio). Por ejemplo, una organización industrial puede tener su estructura interna de departamentos manejada por un marco ontológico. Es posible que surja una situación en la cual un grupo de departamentos sea unido dentro de un único departamento. Este tipo de cambios afectarán inevitablemente la disposición del marco ontológico.
2. *Cambios en la Conceptualización:* manejadas como evolución en la forma en que entendemos al mundo y la perspectiva que tenemos de él. Por ejemplo, una ontología describiendo las aerolíneas de un cierto país pueden temporariamente cambiar cuando aparecen conflictos internacionales con alguno de sus vecinos, en el sentido en que una aerolínea puede dejar de ser considerada como una alternativa apropiada.
3. *Cambios en la Especificación Explícita:* hacen referencia a traducciones entre diferentes lenguajes de representación de conocimientos. Cuando los lenguajes difieren en las semánticas y la expresividad, podría no ser trivial preservar las semánticas ontológicas. No trataremos este tipo de cambios en esta tesis. Para detalles en este tipo de cambios, el lector puede referirse a (CGP00).

En evolución de esquemas de base de datos, la preservación de integridad de datos es un punto primordial. Como fuera expuesto en (NK04), las ontologías pueden ser vistas como esquemas (descripciones terminológicas) para KBs publicadas junto con dato de instancia (descripciones asercionales). Las terminologías son utilizadas para conducir búsquedas y proveer una navegación controlada sobre extensas colecciones de documentos a través de consultas. Por ello, en evolución ontológica, no sólamente consideramos implicaciones sobre los datos de instancia, sino que también observamos cómo los cambios pueden afectar las consultas sobre descripciones terminológicas y sus contenidos.

En contraste con las bases de datos, las ontologías incorporan semánticas para la representación del conocimiento via descripciones terminológicas. Estas semánticas formales permiten interpretar definiciones como axiomas lógicos, exponiendo dos puntos importantes respecto del razonamiento: *consistencia ontológica* y *satisfabilidad de conceptos*. Los cambios en las ontologías pueden resultar en restricciones inconsistentes sobre las descripciones terminológicas, dejando conceptos sin instancias posibles. La intuición sobre insatisfabilidad de conceptos puede ser interpretada como: (1) especificación errónea de un concepto, o (2) conceptualización errónea del dominio. Por lo tanto, este tipo de situación debería ser de alguna forma controlada durante el proceso de cambio.

Los cambios en las descripciones terminológicas podrían introducir contradicciones respecto del conocimiento asercional (dato de instancia). En tales casos, surgen algunas alternativas: se puede deber a alguno de los dos tipos de errores especificados anteriormente, o simplemente puede suceder que sea un cambio correcto. En este último caso, el cambio puede corresponder, o bien a la conceptualización, o al dominio, obligando a revisar el dato de instancia, lo que posiblemente determinará alguna pérdida asercional inevitable. Todas estas situaciones indeseables deben ser manejadas en una forma “user-driven”, es decir, un sistema que provee servicios de evolución para ontologías debe permanecer semi-automático, siendo manejado por un ingeniero ontológico en caso de ser necesario.

Esta discusión esta directamente relacionada con las cuestiones pragmáticas de las operaciones de cambio y el lenguaje más allá de otras propiedades como la relevancia epistémica del conocimiento. En tal caso, el ingeniero ontológico aparece como la alternativa más plausible para tomar decisiones finales. Por ejemplo, consideremos una decripción terminológica de un estudiante de posgrado como un subconjunto del conjunto de personas graduadas intersectado con el conjunto de estudiantes. Si sabemos que Carlos es un estudiante de posgrado (información asercional), entonces deberíamos inferir que no sólo es estudiante sino que también está graduado. Repentinamente, alguien informa que por alguna razón él debería dejar de ser considerado como estudiante. Esto puede afectar la ontología mediante el borrado de la información que asegura que Carlos es estudiante, ¿pero debería la consulta "¿Es Carlos estudiante?" ser contestada en forma negativa o indecidida debido a la falta de información?. Más aún, él debería dejar de ser considerado estudiante de posgrado, pero nada debería cambiar respecto de su situación como persona graduada, por lo tanto, sin importar el resultado de la primera decisión, la última debería verificarse de todas formas.

Por otro lado, al contrario de las bases de datos, las ontologías a menudo reutilizan y extienden otras ontologías. Por ello, los cambios en una ontología pueden afectar otras ontologías que la reutilizan. Incluso por naturaleza, las ontologías son distribuídas, compartidas y descentralizadas. Esta ausencia de centralización y control sincronizado descubre la imposibilidad de conocer a cada usuario de una ontología para lograr informarle acerca de cada cambio o actualización que se desarrolla.

Sobre todo lo discutido anteriormente, nos referiremos a la definición formal de evolución de ontologías expuesta en (NK04), entendida como "la habilidad de manejar cambios ontológicos y sus efectos mediante la creación y mantenimiento de diferentes variantes de la ontología". En esta tesis, no trataremos diferentes variantes de la ontología, sino que determinaremos ontologías alternativas que incorporen el cambio deseado, como forma de dar con el problema de la evolución ontológica. Para ello, nuestro enfoque estará puesto únicamente en la definición de operaciones de cambio para ontologías poniendo especial énfasis en la tolerancia a inconsistencias. Por este motivo, necesitaremos además proveer servicios de razonamiento no-estándares (basados en lenguajes de representación ontológicos) para el manejo de ontologías inconsistentes. Dos dimensiones de compatibilidad (tomado de (NK04)) pueden ser tenidas en cuenta a la hora de provocar el cambio ontológico:

1. *Preservación de Dato de Instancia:* Preservar tanto conocimiento asercional como sea posible.
2. *Preservación de Consecuencias:* Preservar tanto conocimiento implícito como sea posible.

Una vez más, debemos destacar la importancia de admitir que algunos cambios (dinámica ontológica) deban finalmente ser realizados mediante previa interacción con el usuario (ingeniero ontológico). Las propuestas teóricas en esta tesis se construirán sobre tales premisas: los servicios de dinámica definidos proveerán, para las futuras implementaciones, alternativas que requerirán la interacción con el usuario, quien finalmente decidirá aceptar o no los cambios ontológicos propuestos.

### 2.3.2. Principios para el Cambio Ontológico

Como fuera descripto en (Lev84) cualquier operación sobre bases de conocimiento debería ser independiente de como la información es representada, esto significa que deberían ser definidas sobre el *nivel del conocimiento* (New80) sin importar el *nivel simbólico*. Por ejemplo, el modelo de la teoría del cambio puede ser dependiente del nivel simbólico dado que sus operaciones están definidas en términos del conjunto de fórmulas de la base de conocimiento. Esto es así, a menos que la base considerada esté cerrada bajo consecuencia lógica, es decir que se consideren conjuntos de creencias –como en el modelo AGM (AGM85)– con los inconvenientes que este tipo de tareas puede acarrear (ver Sección 2.2). Una alternativa es la de formalizar el cambio mediante enfoques con *semánticas modelo-teóricas*, *i.e.*, considerar el conjunto de modelos de la base (Lev84).

Esta discusión está relacionada con la expuesta en (BFL85) sobre el denominado *enfoque funcional* el cual sostiene que el sistema debería resolver las consultas en función de la información contenida en la base de conocimiento sin importar su implementación (el lenguaje utilizado). Sin embargo, si el sistema desarrolla inferencias en tiempo de aserción, *i.e.*, si computa las inferencias luego de que un cambio es aplicado a la base, esta independencia no podrá ser mantenida (Neb90).

Esto muestra que, si bien las operaciones de cambio a nivel de conocimiento o sobre el enfoque funcional son deseables, la identificación precisa del límite que dividide el nivel semántico del sintáctico, puede resultar un tanto ambigua.

Esto conduce a nuevas preguntas acerca de qué debe considerarse para caracterizar las operaciones de cambio ontológico, para resolver problemas propios de la evolución ontológica. En este sentido, un interesante conjunto de principios fue dado en (Dal88) que, en conjunto con los postulados básicos para el cambio vistos en la Sección 2.2, será de gran importancia para las formalizaciones teóricas de esta tesis. Los siguientes principios resultan de los anteriormente mencionados, adaptados al área de evolución ontológica.

1. *Representación Adecuada:* La ontología evolucionada debería respetar la misma representación (lenguaje y expresividad) que su versión anterior.
2. *Irrelevancia de la Sintaxis:* La ontología evolucionada no debería depender de la sintaxis (lenguaje ontológico) del viejo ni del nuevo conocimiento.
3. *Mantenimiento de Consistencia:* Si ambos conocimientos, el viejo y el nuevo, son consistentes en forma independiente entonces así debería serlo también la ontología evolucionada.
4. *Primacía de la Nueva Información:* A menos que sea insatisfacible, la nueva información debería ser aceptada en la ontología evolucionada.
5. *Cambio Mínimo:* Tanta información original como sea posible debería ser retenida en la ontología evolucionada.
6. *Imparcialidad:* Si existieran varios candidatos para obtener la ontología evolucionada verificando los principios anteriores entonces sólo uno de ellos debería ser elegido en forma no arbitraria.

En el caso del último principio, se intenta evitar el no-determinismo. La interacción con el usuario sería vital para su verificación. Es importante remarcar que estos principios pueden ser útiles para dar lugar a la formalización de postulados. Ellos constituyen guías a tener en cuenta a la hora de formalizar las teorías. Sin embargo, no necesariamente debemos restringirnos ni limitarnos a su verificación completa.

## 2.4. Description Logics (DLs)

Un sistema de representación de conocimiento basado en Description Logics (DLs) provee una formalización para especificar el contenido de una base de conocimiento (KB), una forma de razonar sobre él, y un proceso para inferir conocimiento implícito. Una KB basada en DLs, o en particular, una ontología es dividida en dos componentes: una *TBox* y una *ABox*. Una TBox maneja la *terminología* de la aplicación del mundo mientras que una ABox contiene las *aserciones* acerca de individuos determinados en términos de los conceptos previos.

Una terminología está compuesta por conceptos atómicos los cuales denotan conjuntos de individuos y roles atomicos pata manejar relaciones entre individuos. Además, conceptos y roles complejos son construidos a partir de los atómicos utilizando constructores. Los *servicios de razonamiento* están dedicadas a determinar si una descripción es satisfacible (no-contradictoria), o si una descripción es más general que la otra, esto es, si la primera subsume a la segunda.

Para una ABox, el problema es verificar la consistencia de cada conjunto de aserciones (testear si existe un modelo para el conjunto) y encontrar si un individuo particular es una instancia de una descripción conceptual en la TBox dependiendo de las aserciones en la ABox. El entorno interactuará con la ontología mediante consultas y finalmente mediante el agregado de conceptos, roles, y/o aserciones.

En esta sección daremos una muy breve reseña sobre DLs. Para mayor detalle, el lector es invitado a remitirse a (BCM$^+$03).

### 2.4.1. Ontologías

Una *ontología* $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$ detalla el conocimiento en términos de información intesional y extensional descripta en una TBox $\mathcal{T}$ y una ABox $\mathcal{A}$, respectivamente. Los símbolos $A$ y $C$ identifican conceptos atómicos y generales, mientras que $P$ y $E$ identifican roles atómicos y generales, respectivamente. Los símbolos $a, b, \ldots$ identifican constantes en representación de individuos determinados del dominio, y $x, y, \ldots$, variables libres. Los conjuntos $\mathrm{N_V}$ y $\mathrm{N_C}$ contendrán variables y nombres de constantes, respectively. Una TBox usualmente contiene axiomas como $C_1 \sqsubseteq C_2$ llamados *inclusiones generales de concepto* o *general concept inclusions* (GCIs); y una ABox, *membership assertions* o simplemente *aserciones* como $C(a)$ y $E(a, b)$. En algunos casos una TBox puede además hacer referencia a un segundo tipo de axiomas de la forma $C_1 \equiv C_2$, llamado *definición terminológica* o simplemente *definición*. Sin embargo, las definiciones terminológicas son usualmente reemplazadas por un par de GCIs del tipo $C_1 \sqsubseteq C_2$ y $C_2 \sqsubseteq C_1$. Adicionalmente, diferentes restricciones pueden aparecer dependiendo de cada lenguaje de descripción específico.

### 2.4.2. Semántica

Las semánticas DL son dadas en términos del conjunto estándar teórico de semánticas tarskianas, a través de *interpretaciones* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$. Una interpretación $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consiste de un dominio no-vacío $\Delta^{\mathcal{I}}$, y una función de interpretación $\cdot^{\mathcal{I}}$ que mapea cada concepto a un subconjunto de $\Delta^{\mathcal{I}}$, cada rol a un subconjunto de $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, y cada individuo a un elemento de $\Delta^{\mathcal{I}}$.

Asumiendo a $\mathcal{L}$ como la DL subyacente para representar la ontología $\mathcal{O}$, ocasionalmente se hará uso de la función $\mathcal{M}$ la cual mapeará todas las interpretaciones-modelo del argumento entrada de tipo $2^{\mathcal{L}}$. Esto es, $\mathcal{M}(\Phi)$ será el conjunto de todos los modelos de $\Phi \subseteq \mathcal{L}$; y en particular, $\mathcal{M}(\mathcal{O})$ contendrá todos los modelos de la ontología $\mathcal{O}$. Por lo tanto, para todo $\mathcal{I} \in \mathcal{M}(\mathcal{O})$ se verificará $\mathcal{I} \models \mathcal{O}$.

Una ontología contiene conocimiento implícito que hace explícito mediante inferencias: una ontología $\mathcal{O}$ *implica lógicamente* ó *infiere* una sentencia (axioma o aserción) $\phi$, notado $\mathcal{O} \models \phi$, si para cada modelo $\mathcal{I} \in \mathcal{M}(\mathcal{O})$, se verifica $\mathcal{I} \models \phi$. Esto implica que todo modelo de la ontología $\mathcal{O}$ es también modelo de la sentencia $\phi$. Formalmente,

$$\mathcal{O} \models \phi \textit{ sssi } \mathcal{M}(\mathcal{O}) \subseteq \mathcal{M}(\{\phi\}).$$

Una interpretación $\mathcal{I}$ es un modelo para la TBox $\mathcal{T}$, *i.e.*, $\mathcal{I} \models \mathcal{T}$, si $\mathcal{I}$ satisface todos los axiomas en $\mathcal{T}$, es decir $\mathcal{I} \models \phi$, para todo axioma $\phi \in \mathcal{T}$. Una interpretación $\mathcal{I}$ satisface $C_1 \sqsubseteq C_2$ y $C_1 \equiv C_2$ cuando $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$ y $C_1^{\mathcal{I}} = C_2^{\mathcal{I}}$, respectivamente. Por lo tanto, la TBox $\mathcal{T}$ se dice satisfacible si admite al menos un modelo. Además, en la ABox $\mathcal{A}$, $\mathcal{I}$ satisface $C(a)$ si $a \in C^{\mathcal{I}}$, y $E(a, b)$ si $(a, b) \in E^{\mathcal{I}}$. Una interpretación $\mathcal{I}$ es modelo de la ABox $\mathcal{A}$, *i.e.*, $\mathcal{I} \models \mathcal{A}$, si $\mathcal{I}$ satisface a toda aserción de $\mathcal{A}$, es decir $\mathcal{I} \models \phi$, para toda aserción $\phi \in \mathcal{A}$. Consequentemente, la ABox $\mathcal{A}$ se dice satisfacible si admite al menos un modelo.

Respecto de la ontología completa, una interpretación $\mathcal{I}$ es modelo de $\mathcal{O}$, *i.e.*, $\mathcal{I} \models \mathcal{O}$, si toda sentencia (axioma o aserción) en $\mathcal{O}$ es satisfecha por $\mathcal{I}$, o equivalentemente, si $\mathcal{I} \models \mathcal{T}$ y $\mathcal{I} \models \mathcal{A}$, se verifican en forma simultánea. Finalmente, la ontología $\mathcal{O}$ es *satisfacible* (o *consistente*) si admite al menos un modelo, lo cual implica que existe algún $\mathcal{I} \in \mathcal{M}(\mathcal{O})$.

Sólo por simplicidad, haremos abuso de notación, escribiendo $\mathcal{O} = \mathcal{T} \cup \mathcal{A}$ para identificar una ontología $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$. Por ejemplo, identificaremos una ontología $\mathcal{O} = \langle \{C_1 \sqsubseteq C_2\}, \{A(a)\} \rangle$, notándola como un simple conjunto $\mathcal{O} = \{C_1 \sqsubseteq C_2, A(a)\}$.

### 2.4.3. Servicios de Razonamiento

Para verificar implicaciones lógicas de fórmulas DL, nos basamos en los llamados *servicios de razonamiento* o *Reasoning Services* (RS)[3] como los siguientes:

[3] Un quinto tipo de RS es introducido en la Sección 2.4.5.

- **Subsumption (subsunción):** $\mathcal{O} \models C_1 \sqsubseteq C_2$
- **Instance Checking (chequeo de instancia):** $\mathcal{O} \models C(a)$ or $\mathcal{O} \models E(a, b)$
- **KB Satisfiability (satisfabilidad):** $\mathcal{O}$ admite al menos un modelo $\mathcal{I} \neq \emptyset$
- **Query Answering (respuesta de consulta):** $\mathcal{O} \models q(\bar{x})$

Query answering es utilizada para consultar DLs. Una *consulta conjuntiva* o *conjunctive query (cq)* $q(\bar{x})$, con una tupla $\bar{x} \in (\mathrm{N_V})^n$ de aridad $n \geq 0$, es un conjunto no-vacío de átomos $C(z)$, $E(z_1, z_2)$, $z_1{=}z_2$, ó $z_1{\neq}z_2$, donde $C$ y $E$ son respectivamente un concepto general y un rol general de $\mathcal{O}$, y sólo los nombres $\{z, z_1, z_2\} \cap \mathrm{N_V}$ son considerados en $\bar{x}$. La función $\mathfrak{var} : (\mathrm{N_V})^n \longrightarrow 2^{\mathrm{N_V}}$ identifica las variables en $\bar{x}$. Cuando $\bar{x}$ es la tupla vacía, ninguna variable libre es considerada y la consulta es identificada como *booleana*.

Intuitivamente, $q(\bar{x})$ representa la conjunción de sus elementos. Sea $\mathcal{I}$ una interpretación, y $\mathfrak{m} : \mathfrak{var}(\bar{x}) \cup \mathrm{N_C} \longrightarrow \Delta^{\mathcal{I}}$ una función total. Si $z \in \mathrm{N_C}$ entonces $\mathfrak{m}(z) = z^{\mathcal{I}}$ de lo contrario $\mathfrak{m}(z) = a \in \Delta^{\mathcal{I}}$. Escribimos $\mathcal{I} \models^{\mathfrak{m}} C(z)$ si $\mathfrak{m}(z) \in C^{\mathcal{I}}$, $\mathcal{I} \models^{\mathfrak{m}} E(z_1, z_2)$ si $(\mathfrak{m}(z_1), \mathfrak{m}(z_2)) \in E^{\mathcal{I}}$, $\mathcal{I} \models^{\mathfrak{m}} (z_1{=}z_2)$ si $\mathfrak{m}(z_1){=}\mathfrak{m}(z_2)$, y $\mathcal{I} \models^{\mathfrak{m}} (z_1{\neq}z_2)$ si $\mathfrak{m}(z_1){\neq}\mathfrak{m}(z_2)$. Si $\mathcal{I} \models^{\mathfrak{m}} \phi$ para todo $\phi \in q(\bar{x})$, escribimos $\mathcal{I} \models^{\mathfrak{m}} q(\bar{x})$ y llamamos $\mathfrak{m}$ un *match* para $\mathcal{I}$ y $q(\bar{x})$. Decimos que $\mathcal{I}$ satisface $q(\bar{x})$ y escribimos $\mathcal{I} \models q(\bar{x})$ se existe un match $\mathfrak{m}$ para $\mathcal{I}$ y $q(\bar{x})$. Si $\mathcal{I} \models q(\bar{x})$ para todo modelo $\mathcal{I}$ de una ontología $\mathcal{O}$, escribimos $\mathcal{O} \models q(\bar{x})$ y decimos que $\mathcal{O}$ infiere $q(\bar{x})$.

Observe que los casos de instance checking pueden ser resueltos mediante cqs primitivos. Por ejemplo, $\mathcal{O} \models C(a)$ puede ser resuelto mediante $\mathcal{O} \models q(\bar{x})$, donde $q(\bar{x}) = \{C(a)\}$ y $\bar{x} = \langle\rangle$.

### 2.4.4. Lenguajes de Descripción

Los lenguajes de descripción están definidos por sus constructores. En esta tesis consideraremos un pequeño subconjunto del gran conjunto de constructores DLs existentes.

El lenguaje de descripción básico $\mathcal{AL}$, por *attribute language* o lenguaje de atributos, es formado de acuerdo a la sintaxis:

$$C, D \longrightarrow A|\bot|\top|\neg A|C \sqcap D|\forall R.C|\exists R.\top$$

donde $A$ es un concepto atómico, $R$ es un rol (o relación) atómico; y la función de interpretación $\cdot^{\mathcal{I}}$ es extendida de acuerdo al detalle de constructores del Cuadro 2.1.

| **Constructor** | **Escritura** | **Interpretación** |
|---|---|---|
| *Concepto universal* | $\top$ | $\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$ |
| *Concepto bottom* | $\bot$ | $\bot^{\mathcal{I}} = \emptyset$ |
| *Negación atómica* | $\neg A$ | $(\neg A)^{\mathcal{I}} = \Delta^{\mathcal{I}} \backslash A^{\mathcal{I}}$ |
| *Intersección* | $C \sqcap D$ | $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$ |
| *Cuantificación Universal* | $\forall R.C$ | $(\forall R.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \vert \forall b.(a,b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\}$ |
| *Cuantificación Existencial Limitada* | $\exists R.\top$ | $(\exists R.\top)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \vert \exists b.(a,b) \in R^{\mathcal{I}}\}$ |

Cuadro 2.1: Constructores para el lenguaje básico $\mathcal{AL}$.

Otros lenguajes más expresivos surgen mediante la adición de nuevos constructores al lenguaje básico $\mathcal{AL}$ tales como los listados en el Cuadro 2.2. Extendiendo $\mathcal{AL}$ por cualquiera de ellos da lugar a un nuevo lenguaje que será nombrado de acuerdo a una cadena respetando la forma $\mathcal{AL}[\mathcal{U}][\mathcal{E}][\mathcal{C}][\mathcal{N}][\mathcal{Q}]$, donde cada letra en el nombre representa la presencia del constructor correspondiente.

| **Constructor** | **Escritura** | **Interpretación** |
|---|---|---|
| *Unión* ($\mathcal{U}$) | $C \sqcup D$ | $(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$ |
| *Complemento* ($\mathcal{C}$) | $\neg C$ | $(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \backslash C^{\mathcal{I}}$ |
| *Cuantificación Existencial Completa* ($\mathcal{E}$) | $\exists R.C$ | $(\exists R.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \vert \exists b.(a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}$ |
| *Restricciones Numéricas* ($\mathcal{N}$) | $\geqslant nR$<br>$\leqslant nR$<br>$= nR$ | $(\geqslant nR)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \vert\ \Vert\{b \in \Delta^{\mathcal{I}} \vert (a,b) \in R^{\mathcal{I}}\}\Vert \geq n\}$<br>$(\leqslant nR)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \vert\ \Vert\{b \in \Delta^{\mathcal{I}} \vert (a,b) \in R^{\mathcal{I}}\}\Vert \leq n\}$<br>$(= nR)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \vert\ \Vert\{b \in \Delta^{\mathcal{I}} \vert (a,b) \in R^{\mathcal{I}}\}\Vert = n\}$ |
| *Restricciones Numéricas Incalificadas* ($\mathcal{Q}$) | $\geqslant nR.C$<br>$\leqslant nR.C$<br>$= nR.C$ | $(\geqslant nR.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \vert\ \Vert\{b \in \Delta^{\mathcal{I}} \vert (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}\Vert \geq n\}$<br>$(\leqslant nR.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \vert\ \Vert\{b \in \Delta^{\mathcal{I}} \vert (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}\Vert \leq n\}$<br>$(= nR.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \vert\ \Vert\{b \in \Delta^{\mathcal{I}} \vert (a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}\Vert = n\}$ |

Cuadro 2.2: Constructores de extensión del lenguaje básico $\mathcal{AL}$.

**Ejemplo 7** *Consideremos una ontología $\mathcal{O}_1$, basada en el lenguaje $\mathcal{ALN}$. Decimos que un estudiante de ciencias de computación es considerado avanzado (mediante el concepto*

*$\mathsf{EstudianteAvanzado}$) si ha completado 15 cursos de un total de 25 (mediante el uso del rol $\mathsf{cursoCompleto}$), esto es:*

$$\mathsf{EstudianteAvanzado} \sqsubseteq \geqslant 15\ \mathsf{cursoCompleto} \sqcap \leqslant 25\ \mathsf{cursoCompleto}$$

*Por otro lado, consideremos un segunda ontología $\mathcal{O}_2$, basada en el lenguaje $\mathcal{ALQ}$ (las restricciones numéricas hacen referencia a roles limitados a la correspondencia con la clase determinada por un cierto concepto). Luego podemos decir que un estudiante debería completar al menos 6 cursos del área de lógica y matemática (mediante el concepto $\mathsf{LogMat}$) y 9 cursos del área de programación (mediante $\mathsf{Prog}$) para ser considerado avanzado:*

$$\begin{aligned} \mathsf{EstudianteAvanzado} \sqsubseteq\ & \geqslant 6\ \mathsf{cursoCompleto.LogMat} \sqcap \\ & \geqslant 9\ \mathsf{cursoCompleto.Prog} \sqcap \\ & \leqslant 25\ \mathsf{cursoCompleto} \end{aligned}$$

*Si consideramos los siguientes individuos para ambas ontologías $\mathcal{O}_1$ y $\mathcal{O}_2$:*

| | |
|---|---|
| rpa | *(para identificar Resolución de Problemas y Algoritmos)* |
| eag | *(Elementos de Álgebra y de Geometría)* |
| ami | *(Análisis Matemático I)* |
| poo | *(Introducción a la Programación Orientada a Objetos)* |
| lfa | *(Lenguajes Formales y Autómatas)* |
| carlos | *(Carlos A. Bermudez)* |

*y las siguientes aserciones:*

| | |
|---|---|
| Prog(rpa) | cursoCompleto(carlos,rpa) |
| Prog(poo) | cursoCompleto(carlos,eag) |
| LogMat(eag) | cursoCompleto(carlos,ami) |
| LogMat(ami) | cursoCompleto(carlos,poo) |
| LogMat(lfa) | cursoCompleto(carlos,lfa) |

*es evidente que Carlos no corresponderá a la clase correspondiente al concepto de estudiante avanzado, para ninguna de las dos ontologías dadas. Por lo tanto, $\mathsf{carlos} \notin \mathsf{EstudianteAvanzado}^{\mathcal{I}}$, para cualquier interpretación $\mathcal{I}$, donde $\mathcal{I}$ es modelo de la ontología $\mathcal{O}_1$ o de la ontología $\mathcal{O}_2$.*

Para constructores de roles, escribimos las letras/símbolos como superíndices, y para restricciones en la interpretación de roles como subíndices. Algunos constructores de roles

| **Constructor de Roles** | **Escritura** | **Interpretación** |
|---|---|---|
| *Unión* ($\sqcup$) | $R_1 \sqcup R_2$ | $(R_1 \sqcup R_2)^\mathcal{I} = R_1^\mathcal{I} \cup R_2^\mathcal{I}$ |
| *Intersección* ($\sqcap$) | $R_1 \sqcap R_2$ | $(R_1 \sqcap R_2)^\mathcal{I} = R_1^\mathcal{I} \cap R_2^\mathcal{I}$ |
| *Complemento* ($\neg$) | $\neg R$ | $(\neg R)^\mathcal{I} = \Delta^\mathcal{I} \times \Delta^\mathcal{I} \setminus R^\mathcal{I}$ |
| *Inversión* ($-1$) | $R^-$ | $(R^-)^\mathcal{I} = \{(b,a) \in \Delta^\mathcal{I} \times \Delta^\mathcal{I} \vert (a,b) \in R^\mathcal{I}$ |
| *Composición* ($\circ$) | $R_1 \circ R_2$ | $(R_1 \circ R_2)^\mathcal{I} = R_1^\mathcal{I} \circ R_2^\mathcal{I}$ |

Cuadro 2.3: Constructores de roles.

son inroducidos en el Cuadro 2.3. Por ejemplo, un rol $\mathsf{padreDe}$ es obtenido mediante la aplicación del constructor de inversión de roles a un dado rol $\mathsf{hijoDe}$, es decir para cualquier par $(a, b) \in \mathsf{hijoDe}^\mathcal{I}$, tendríamos que $(b, a) \in (\mathsf{hijoDe}^-)^\mathcal{I}$, lo cual equivaldría a $(b, a) \in \mathsf{padreDe}^\mathcal{I}$.

También pueden aparecer restricciones sobre la interpretación de roles como por ejemplo *transitividad de roles* ($R^+$), y *reflexividad-tranitividad de roles* ($R^*$).

Por ejemplo, la DL $\mathcal{ALCQ}^{-1}_{R^+}$ extiende $\mathcal{AL}$ con los constructores conceptuales de negación ($\mathcal{C}$) y restricciones numéricas calificadas ($\mathcal{Q}$), los constructores de roles de inversión ($-1$), y la restricción sobre la iterpretación de roles tal que algunos roles serán considerados transitivos ($R^+$). Un conjunto de axiomas de la forma $R \sqsubseteq S$, donde tanto $R$ como $S$ son atómicos, es llamado *jerarquía de roles*. Por ejemplo, para el caso anterior en que el rol $\mathsf{padreDe}$ es obtenido mediante la aplicación del constructor de inversión de roles al rol $\mathsf{hijoDe}$, podríamos hacer explícita su definición mediante el axioma $\mathsf{padreDe} \sqsubseteq \mathsf{hijoDe}^-$. Una base de conocimiento conteniendo jerarquía de roles es indicada mediante la inclusión de la letra $\mathcal{H}$ al nombre de la DL en cuestión.

Los nombres de individuos, también llamados nominales, no siempre están únicamente contenidos en la ABox. Algunas veces es útil tener información sobre ellos como parte del mismo lenguaje de descripción. El constructor de conceptos más básico para individuos es el *set* (o $\mathsf{one\text{-}of}$), notado como:

$$\{a_1, ..., a_n\}$$

donde $a_1, ..., a_n$ son nombres de individuos, y su interpretación es

$$\{a_1, ..., a_n\}^\mathcal{I} = \{a_1^\mathcal{I}, ..., a_n^\mathcal{I}\}$$

Con conjuntos en el lenguaje de descripción podemos definir, por ejemplo los días que componen la semana como $\{DOM, LUN, MAR, MIE, JUE, VIE, SAB\}$.

Otro constructor de conceptos es *fills*, $R : a$, cuya semántica está definida como,

$$(R : a)^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} \mid (d, a^{\mathcal{I}}) \in R^{\mathcal{I}}\}$$

esto es $R : a$ representa al conjunto de aquellos objetos que tienen a $a$ como *filler* (segundo elemento del par) del rol $R$. Note que $R : a$ y $\exists R.\{a\}$ son equivalentes para un lenguaje con sets y cuantificación existencial completa. Observe además, que *fills* permite expresar aserciones de roles a través de aserciones de conceptos,

Una interpretación satisface $R(a, b)$ *sssi* satisface $(\exists R.\{b\})(a)$

Los lenguajes con constructores nominales son indentificados mediante el agregado de la letra $\mathcal{O}$ a la cadena en representación de su nombre. Finalmente, con el objetivo de evitar cadenas de nombres demasiado extensas para identificar DLs de alta expresividad, la abreviación $\mathcal{S}$ fue introducida en reemplazo de $\mathcal{ALC}_{R^+}$, *i.e.*, la DL que extiende $\mathcal{ALC}$ con transitividad de roles. Entre los miembros más conocidos de la familia $\mathcal{S}$ se encuentran $\mathcal{SHIF}$, que extiende $\mathcal{ALC}_{R^+}$ con restricciones numéricas e inversión de roles. Esta DL en particular, satisface las características de ontologías modeladas por el uso de la recomendación OWL Lite, mientras que $\mathcal{SHOIN}$ es la DL elegida de base de la recomendación OWL DL. Para un detalle exhaustivo de estas y otras DLs, el lector puede referirse a (BCM+03).

### 2.4.5. Expresividad vs. Eficiencia: Nuevos Lenguajes

#### La DL Estándar $\mathcal{ALC}$

El lenguaje de descripción estándar por adopción es el de la DL $\mathcal{ALC}$, construida a partir de la siguiente gramática,

$$C \longrightarrow A|\bot|\top|\neg C|C \sqcap C|C \sqcup C|\forall P.C|\exists P.C$$

El razonamiento en la DL estándar $\mathcal{ALC}$ corresponde a la clase de complejidad EXPTIME-complete[4], sin embargo, mediante la restricción de su uso a *unfoldable* $\mathcal{ALC}$, el

[4]EXPTIME contiene a los problemas que requieren de un tiempo exponencial para su resolución.

problema de satisfabilidad se vuelca a la clase PSPACE-complete[5]. (Una TBox es llamada unfoldable si la parte izquierda (left-hand side) de sus axiomas (conceptos definidos) son atómicos, y si su parte derecha (right-hand side) (definiciones) no contienen referencia directo ni indirecta al concepto definido.) El panorama se torna peor para DLs de expresividad mayor.

#### Las Variantes $\mathcal{EL}$

Diferentes DLs fueron propuestas con el objetivo de reducir la complejidad de razonamiento en detrimento de su expresividad. Este es el caso de $\mathcal{EL}$ (Baa03) en la cual el chequeo de satisfabilidad fue demostrado polinomial. La DL $\mathcal{EL}$ se construye a partir de la gramática siguiente,

$$C \longrightarrow A|\bot|\top|C \sqcap C|\exists P.C$$

En esta tesis, en los casos en que se haga referencia a las DLs $\mathcal{ALC}$ (y algunas de sus extensiones) y para $\mathcal{EL}$, sólamente consideraremos los servicios de razonamiento (RSs) de *subsunción* $\mathcal{O} \models C_1 \sqsubseteq C_2$; y *query answering* $\mathcal{O} \models q(\bar{x})$. Adicionalmente, para aquellos lenguajes cuyos nombres contienen la letra $\mathcal{H}$, la subsunción también corresponderá a $\mathcal{O} \models E_1 \sqsubseteq E_2$. Este es el caso de $\mathcal{ALCH}$, $\mathcal{ELH}$, y $\mathcal{ALCNH}^{-1,\neg}$ (que utilizaremos luego en el Ejemplo 32), entre otros.

#### La Familia DL-Lite

Una familia particular de DLs cuyo objetivo inicial fue reducir la complejidad en detrimento de la expresividad es la familia DL-Lite (CGL$^+$07). Este tipo de lógicas permiten razonar sobre extensas cantidades de datos asercionales, es decir, sobre grandes ABoxes. La denominada *complejidad de datos* o *data complexity* de query answering (con respecto al tamaño de la ABox) se encuentra en la clase LOGSPACE[6] para la mayoría de los miembros de la familia DL-Lite, y polinomial con respecto a la ontología completa (TBox y ABox simultáneamente). Una de las virtudes de las ontologías DL-Lite es su capacidad

[5]PSPACE identifica los problemas que requieren un espacio de tamaño polinomial para su resolución.
[6]LOGSPACE identifica los problemas que requieren un espacio de tamaño logarítmico para su resolución.

para admitir la re-escritura de consultas como consultas SQL. De esta forma, los motores de consulta de bases de datos estándares pueden ser utilizados.

La gramática $\textit{DL-Lite}_{\texttt{core}}$ está dada como,

$$B \longrightarrow A|\exists R \qquad C \longrightarrow B|\neg B \qquad R \longrightarrow P|P^{-} \qquad E \longrightarrow R|\neg R$$

En $\textit{DL-Lite}_{\texttt{core}}$, la TBox está formada por axiomas de tipo $B \sqsubseteq C$, y la ABox por aserciones del tipo $A(a)$ y $P(a,b)$. Agregando a $\textit{DL-Lite}_{\texttt{core}}$ axiomas del tipo $R \sqsubseteq E$, o *restrictiones funcionales* como $(\text{funct}R)$, determina los lenguajes $\textit{DL-Lite}_{\mathcal{R}}$ y $\textit{DL-Lite}_{\mathcal{F}}$, respectivamente. Una interpretación $\mathcal{I}$ es un modelo de una restricción funcional $(\text{funct}R)$ si la relación binaria $R^{\mathcal{I}}$ es una función, *i.e.*, $(x,y_1) \in R^{\mathcal{I}}$ y $(x,y_2) \in R^{\mathcal{I}}$ implica $y_1 = y_2$.

Para ontologías considerando restricciones funcionales, se incorpora un nuevo tipo de servicio de razonamiento, el RS de *chequeo de funcionalidad*:

- **Chequeo de Funcionalidad:** $\mathcal{O} \models (\text{funct}R)$ y $\mathcal{O} \models \neg(\text{funct}R)$

La combinación de ambas extensiones, $\textit{DL-Lite}_{\mathcal{R}}$ y $\textit{DL-Lite}_{\mathcal{F}}$, determina el lenguaje $\textit{DL-Lite}_{(\mathcal{RF})}$, cuyo chequeo de satisfabilidad se encuentra en la clase EXPTIME-hard.

### 2.4.6. Semánticas de Mundo Cerrado vs. Abierto.

La consideración de la familia DL-Lite como una alternativa de acercamiento a bases de datos abre la discusión acerca de las semánticas de mundo cerrado y abierto.

Una analogía entre bases de datos y bases de conocimientos basadas en DLs (ontologías) es posible. El esquema de base de datos puede ser comparado con la TBox, mientras que las instancias de datos pueden ser pensadas como una ABox. Sin embargo, las semánticas son las que hacen la diferencia. Mientras que una instancia de base de datos representa exactamente una interpretación, una ABox representa muchas interpretaciones diferentes, es decir, todos sus modelos. Como consecuencia, la ausencia de información en una instancia de base de datos es interpretada como información negativa, mientras que la ausencia de información en una ABox sólo indica ausencia de conocimiento.

Mientras la información en una base de datos es siempre interpretada como completa, la información en una ABox es en general vista como incompleta. La semántica de las

ABoxes es, por lo tanto, caracterizada como de *mundo abierto* (*open-world*), mientras que la semántica tradicional de bases de datos es caracterizada como de *mundo cerrado* (*closed world*).

### 2.4.7. Algoritmos Tableaux para el Razonamiento

Los problemas de inferencia son usualmente reducidos a problemas de consistencia para ABoxes, para los casos en que la DL subyacente permiten el uso de constructores de conjunción y negación. Sin embargo, para aquellos lenguajes de descripción que no permiten negación, la subsunción de conceptos puede ser computada por los denominados *algoritmos de subsunción estructural*, *i.e.*, algoritmos que comparan la estructura sintáctica de descripciones conceptuales (posiblemente normalizadas). El primer sistema de razonamiento de DLs basado en algoritmos de subsunción estructural fue KL-ONE (1985), y otros sistemas posteriores de este tipo fueron KRYPTON (1983), LOOM (1987), BACK (1988), K-REP (1991) y CLASSIC (1991).

Si bien los algoritmos de subsunción estructural son usualmente muy eficientes (polinomiales (BHS07)), son solamente completos para lenguajes simples y de poca expresividad. En particular, las DLs con negación completa y disyunción no pueden ser manejadas. Para tales lenguajes, se propusieron los denominados *algoritmos basados en tableau*, resultando de extrema utilidad. Estos nuevos algoritmos, entendidos como un tipo de tableaux calculi especializado, surgen para DLs de los enfoques en que se considera chequeo de satisfabilidad como la principal inferencia. Entre los varios sistemas DLs basados en chequeo de satisfabilidad podemos nombrar a KRIS (BH91) [7], CRACK (BFT95), FaCT (Hor98), DLP (PS99), RACER (HM01).

#### Ejemplos de Chequeo de Subsunción por Tableau

Como mencionamos anteriormente, en lugar de testear subsunción de descripciones conceptuales, estos algoritmos están basados en chequeo de satisfabilidad. Para ello, se utiliza negación para reducir subsunción a (in)satisfabilidad de descripciones conceptuales:

$$C \sqsubseteq D \textit{ sssi } C \sqcap \neg D \textit{ es insatisfacible.}$$

[7]De los primeros razonadores basados en tableau-calculi. Mostró una performance aceptable sobre problemas de inferencia típicos, aunque para el peor caso la complejidad ya no era polinomial (BHS07).

Ilustraremos las ideas mediante dos simples ejemplos tomados de (BN02). Sean $A$ y $B$ dos conceptos, y $R$ un rol. Como primer ejemplo, asumamos que deseamos saber si $(\exists R.A) \sqcap (\exists R.B)$ es subsumido por $\exists R.(A \sqcap B)$. Esto significa que debemos chequear si la descripción del concepto $C = (\exists R.A) \sqcap (\exists R.B) \sqcap \neg(\exists R.(A \sqcap B))$ es insatisfacible.

Desplazando los símbolos de negación dentro de la descripción tanto como sea posible, determinaremos el concepto $C_0 = (\exists R.A) \sqcap (\exists R.B) \sqcap \forall R.(\neg A \sqcup \neg B)$, correspondiendo a la forma normal de negación (*negation normal form*[8]).

Luego, construiremos una interpretación finita $\mathcal{I}$ tal que $C_0^{\mathcal{I}} \neq \emptyset$. Esto significa que debe existir un individuo en $\Delta^{\mathcal{I}}$ que es un elemento de $C_0^{\mathcal{I}}$. El algoritmo simplemente genera tal individuo, digamos $b$, e impone la restricción $b \in C_0^{\mathcal{I}}$ sobre él, esto significa que $b$ debe satisfacer las tres conjunciones interpretadas que componen $C_0$.

De $b \in (\exists R.A)^{\mathcal{I}}$ podemos deducir que debe existir un individuo $c$ tal que $(b, c) \in R^{\mathcal{I}}$ y $c \in A^{\mathcal{I}}$. Análogamente, $b \in (\exists R.B)^{\mathcal{I}}$ implica la existencia de un individuo $d$ con $(b, d) \in R^{\mathcal{I}}$ y $d \in B^{\mathcal{I}}$. En esta situación, no se debería asumir que $c = d$. Por ello:

- *Para la restricción existencia, el algoritmo introduce un nuevo individuo como rol filler, y tal individuo debe satisfacer las restricciones expresadas por la restricción.*

Dado que $b$ debe también satisfacer la restricción de valor $\forall R.(\neg A \sqcup \neg B)$, y $c$, $d$ fueron introducidas como R-fillers de $b$, obtenemos las restricciones adicionales $c \in (\neg A \sqcup \neg B)^{\mathcal{I}}$ and $d \in (\neg A \sqcup \neg B)^{\mathcal{I}}$. Por lo tanto:

- *El algoritmo usa restricciones de valores en interacción con roles ya definidos para imponer nuevas restricciones sobre individuos.*

Luego, $c$ debería ser tal que $c \in (\neg A)^{\mathcal{I}}$ ó $c \in (\neg B)^{\mathcal{I}}$. La primera posibilidad conduce a una contradicción; por lo tanto, seleccionaremos la segunda opción. Análogamente, debemos elegir $d \in (\neg A)^{\mathcal{I}}$ para satisfacer la restricción $d \in (\neg A \sqcup \neg B)^{\mathcal{I}}$ sin crear una contradicción en $d \in B^{\mathcal{I}}$. Por lo tanto:

- *Para restricciones disyuntivas, el algoritmo trata ambas posibilidades en intentos sucesivos. Realizará un backtracking si alcanza una contradicción obvia, i.e., si el mismo individuo debe satisfacer restricciones que son obviamente conflictivas.*

[8] Negación ocurre sólo en frente de los nombres de concepto.

En el ejemplo se han verificado todas las restricciones sin encontrar una contradicción obvia. Esto demuestra que $C_0$ es satisfacible, y por ello $(\exists R.A) \sqcap (\exists R.B)$ no es subsumido por $\exists R.(A \sqcap B)$. La interpretación generada por el algoritmo es $\Delta^{\mathcal{I}} = \{b, c, d\}$; $R^{\mathcal{I}} = \{(b, c), (b, d)\}$; $A^{\mathcal{I}} = \{c\}$ y $B^{\mathcal{I}} = \{d\}$.

Para el segundo ejemplo queremos chequear si $(\exists R.A) \sqcap (\exists R.B) \sqcap \leqslant 1R$ es subsumido por $\exists R.(A \sqcap B)$. El algoritmo tableaux para satisfabilidad primeramente procede como en el caso anterior, con la única diferencia de que existe la restricción adicional $b \in (\leqslant 1R)^{\mathcal{I}}$. Para satisfacer esta restricción, los dos R-fillers $c, d$ de $b$ deben ser identificados con el otro. Por lo tanto:

- *Si una restricción numérica de tipo "a lo sumo" es violada entonces el algoritmo debe identificar diferentes fillers de rol.*

El individuo $c = d$ debe pertenecer a ambos $A^{\mathcal{I}}$ y $B^{\mathcal{I}}$, lo cual en conjunción con $c = d \in (\neg A \sqcup \neg B)^{\mathcal{I}}$ determinará una contradicción. Por ello, la búsqueda de un contraejemplo para la la relación de subsunción falla, y el algoritmo concluye $(\exists R.A) \sqcap (\exists R.B) \sqcap \leqslant 1R \sqsubseteq \exists R.(A \sqcap B)$.

### 2.4.8. Inconsistencias Ontológicas

Las diferentes clases de inconsistencias en una ontología son definidas a través de la noción usual de *inconsistencia* en lógica clásica, junto con la noción de *incoherencia* presentada en (FHP+06). En ese artículo, se analiza las clases de inconsistencias en una ontología a partir de la inconsistencia más primitiva, *i.e.*, la insatisfabilidad de un simple concepto.

**Definición 2.4.1 (Concepto Insatisfacible)** (FHP+06) *Un concepto $C$ es **insatisfacible** sssi para cada interpretación $\mathcal{I} \in \mathcal{M}(\mathcal{O})$, se verifica $C^{\mathcal{I}} = \emptyset$.*

Esto puede ser utilizado para reconocer la clase de ontologías conteniendo conceptos insatisfacibles. Tal tipo de ontologías será denominada *incoherente*.

**Definición 2.4.2 (Ontología Incoherente)** (FHP+06) *Una ontología $\mathcal{O}$ es **incoherente** sssi existe un determinado concepto $C$ insatisfacible en $\mathcal{O}$.*

La incoherencia puede ser considerada como un tipo de inconsistencia en una TBox, *i.e.*, la parte terminológica de una ontología. Una ontología incoherente tiene una TBox incoherente. Sin embargo, la incoherencia no completa la noción clásica de inconsistencia debido a que puede existir un modelo para una ontología incoherente. Por ello, también necesitaremos la noción clásica de inconsistencia para las ontologías.

**Definición 2.4.3 (Ontología Inconsistente)** (FHP$^+$06) *Una ontología $\mathcal{O}$ es **inconsistente** sssi no admite modelo alguno, i.e., no existe $\mathcal{I} \in \mathcal{M}(\mathcal{O})$.*

Observe nuevamente que la satisfabilidad para una ontología incoherente puede aún mantenerse. Por lo tanto, aunque incoherencia es considerada un tipo de inconsistencia en la TBox, no reemplaza la noción usual de inconsistencia, dado que una ontología incoherente puede admitir modelos no vacíos.

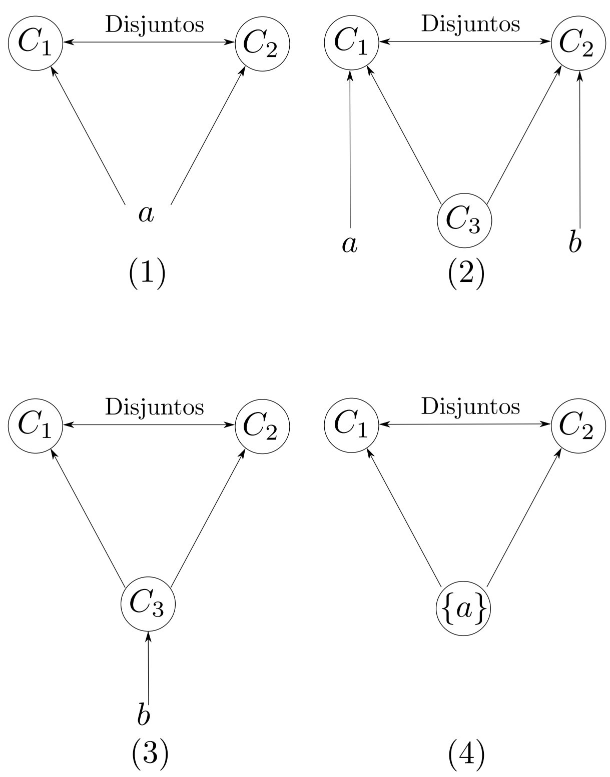



Figura 2.8: Ejemplos de variantes de inconsistencias e incoherencias.

Discutiremos brevemente las relaciones entre las dos clases de inconsistencias en ontologías de acuerdo a su introducción original en (FHP$^+$06). Primeramente, una ontología que es inconsistente no necesariamente debe ser incoherente, y vice versa. Existen diferentes combinaciones de inconsistencias e incoherencias. En la Figura 2.8 (tomada de (FHP$^+$06)) se muestran algunos ejemplos de tales variantes. La Figura 2.8(1) es un ejemplo de una ontología inconsistente coherente, en la cual dos conceptos disjuntos $C_1$ y $C_2$ comparten una instancia $a$. La Figura 2.8(2) es un ejemplo de ontología consistente incoherente, en la cual los dos conceptos disjuntos $C_1$ y $C_2$ comparten un sub-concepto $C_3$. La Figura 2.8(3) es un ejemplo de una ontología inconsistente e incoherente, en la cual los dos conceptos disjuntos $C_1$ y $C_2$ comparten un sub-concepto $C_3$, el cual contiene una instancia $b$. La Figura 2.8(4) es un ejemplo de TBox inconsistente coherente, en la que tanto $C_1$ como $C_2$ comparten un sub-concepto el cual es un nominal $\{a\}$.

Por otro lado, coherencia y consistencia están de alguna forma relacionadas: podemos introducir un nuevo individuo $i_C$ para cada concepto $C$ en una ontología $\mathcal{O}$. En este sentido, una ontología aumentada $\mathcal{O}^+ = \mathcal{O} \cup \{C(i_C)|$ para todo concepto $C$ en $\mathcal{O}\}$ puede ser construida incorporando tales aserciones sobre los nuevos individuos en la ontología.

**Proposición 2.4.4** (FHP$^+$06)

1. *Dada una ontología $\mathcal{O}$, si su ontología aumentada $\mathcal{O}^+$ es consistente, entonces $\mathcal{O}$ es coherente.*
2. *Dada una ontología consistente $\mathcal{O}$, si $\mathcal{O}$ es coherente, entonces su ontología aumentada $\mathcal{O}^+$ es consistente.*

# Capítulo 3

# Argumentación Abstracta Generalizada: Una Maquinaria de Primer-orden para Debugging de Ontologías

Como mencionamos anteriormente, en este capítulo definimos el llamado *marco argumentativo generalizado* o *generalized argumentation framework* (`GenAF`) como una nueva familia de marcos argumentativos abstractos con capacidades de adaptación a diferentes lenguajes de representación cuya expresividad esté contenida en las de la lógica de primer-orden. Utilizaremos tales marcos para razonar sobre KBs inconsistentes. Las semánticas estándares de Dung introducidas en la Sección 2.1.2, son adaptadas para construir la maquinaria de razonamiento del `GenAF`. Sobre ellas, se propondrá un operador de debugging para la KB subyacente. Luego, el comportamiento de la operación de debugging será caracterizado axiomáticamente en base a los postulados de la operación de consolidación (ver Sección 2.2.2), mostrando el correspondiente teorema de representación. Finalmente, nos concentraremos en la aplicación del `GenAF` como modelo de razonamiento y debugging de ontologías inconsistentes. Para ello, propondremos la reificación del lenguaje abstracto para argumentos del `GenAF` al lenguaje básico de descripción $\mathcal{ALC}$.

## 3.1. Fundamentos para un AF Generalizado

El poder máximo de expresividad de un GenAF es impuesto mediante la restricción de sus componentes internos a una lógica $\mathcal{L}^\kappa$ como límite de expresividad superior, donde $\kappa \in \mathbb{N}_0$[1]. Las fórmulas en $\mathcal{L}^\kappa$ serán aquellas correspondientes a FOL que pueden ser construidas con la ayuda de letras predicativas de aridad $\leq \kappa$, incluyendo el uso de la igualdad (*equality*) y símbolos constantes, pero sin símbolos funcionales. En particular, $\mathcal{L}^2$ fue mostrado decidible en (Mor75). Un ejemplo de una lógica $\mathcal{L}^2$ es el caso de la DL estándar $\mathcal{ALC}$ (ver Sección 2.4.5). Para mayor detalles sobre la compatibilidad $\mathcal{L}^2$-$\mathcal{ALC}$ el lector es referido a (Bor96; Baa99).

Para $\mathcal{L}^\kappa$ utilizamos $p, p_1, p_2, \ldots$ y $q, q_1, q_2, \ldots$ para denotar letras predicativas monádicas, $r, r_1, r_2, \ldots$ para letras predicativas diádicas, $x, y$ para variables libres, y $a, b, c, d$ para constantes (nombres de individuos). Además, la lógica $\mathcal{L}_{\mathtt{A}} \subset \mathcal{L}^\kappa$ identifica el fragmento de $\mathcal{L}^\kappa$ describiendo algún tipo de *fórmula asercional*, *i.e.*, cualquier sublenguaje $\mathcal{L}^\kappa$ que considere sólo fórmulas *ground* (sin variables libres).

La lógica $\mathcal{L}^\kappa$ es interpretada en términos del conjunto estándar teórico de semánticas tarskianas, a través de interpretaciones $\mathcal{I} = \langle \Delta^\mathcal{I}, p^\mathcal{I}, p_1^\mathcal{I}, \ldots, q^\mathcal{I}, q_1^\mathcal{I}, \ldots, r^\mathcal{I}, r_1^\mathcal{I}, \ldots \rangle$, donde $\Delta^\mathcal{I}$ es el dominio de interpretación, y $p^\mathcal{I}, p_1^\mathcal{I}, \ldots, q^\mathcal{I}, q_1^\mathcal{I}, \ldots, r^\mathcal{I}, r_1^\mathcal{I}, \ldots$ interpreta $p, p_1, \ldots, q, q_1, \ldots, r, r_1, \ldots$, respectivamente. Para una interpretación $\mathcal{I}$, algún $a \in \Delta^\mathcal{I}$, y una fórmula $\varphi(x)$, escribimos $\mathcal{I} \models \varphi(a)$ si $\mathcal{I}, v \models \varphi(x)$, para la asignación $v$ mapeando $x$ a $a$. De aquí en más, y sólo por simplicidad, omitiremos cuantificadores universales, escribiendo $\varphi(x)$ para referirnos a $(\forall x)(\varphi(x))$.

Intuitivamente, un *argumento* puede ser visto como *una pieza indivisible de conocimiento infiriendo un claim (conclusión) a partir de una conjunto de premisas*. Dado que el claim y las premisas son entidades distinguibles de cualquier argumento, definiremos dos lenguajes independientes para su especificación. Por lo tanto, proponemos un *lenguaje argumental* $\mathbb{A}\mathtt{rgs}$ caracterizado mediante la interrelación entre sus componentes internos: los sub-lenguajes, $\mathcal{L}_{\mathtt{pr}}$ para premisas, y $\mathcal{L}_{\mathtt{cl}}$ para claims.

**Definición 3.1.1 (Lenguaje Argumental)** *Dada la lógica $\mathcal{L}^\kappa$, un* ***lenguaje argumental*** $\mathbb{A}\mathtt{rgs}$ *es definido como $2^{\mathcal{L}_{\mathtt{pr}}} \times \mathcal{L}_{\mathtt{cl}}$, donde $\mathcal{L}_{\mathtt{cl}} \subseteq \mathcal{L}^\kappa$ y $\mathcal{L}_{\mathtt{pr}} \subseteq \mathcal{L}^\kappa$ son identificados como los respectivos lenguajes para claims y premisas en* $\mathbb{A}\mathtt{rgs}$.

[1]Asumiremos los números naturales contenidos en los conjuntos $\mathbb{N}_0 = \{0, 1, \ldots\}$ and $\mathbb{N}_1 = \{1, 2, \ldots\}$.

Usualmente, los argumentos son interpretados como piezas de conocimiento en justificación de sus claims. Sin embargo, cuando consideramos premisas, ellas tienen necesariamente que ser satisfechas para que los argumentos alcancen sus claims. Esto es referido en la literatura como *relación de soporte* (RMGS10), o como *bipolaridad* en (ACLSL08). Dado que una premisa es soportada por un claim de otro/s argumento/s, la expresividad de ambos lenguajes $\mathcal{L}_{\mathtt{pr}}$ y $\mathcal{L}_{\mathtt{cl}}$ debe ser controlada para permitir que toda premisa representable logre ser soportada por fórmulas del lenguaje para claims. Por lo tanto, para mantener el lenguaje argumental $\mathbb{A}\mathtt{rgs}$ a un nivel abstracto, lo caracterizaremos mediante la interrelación $\mathcal{L}_{\mathtt{pr}}$-$\mathcal{L}_{\mathtt{cl}}$ determinando la *condición de legalidad del lenguaje argumental.*

**Definición 3.1.2 (Lenguaje Argumental Legal)** *Un lenguaje argumental* $2^{\mathcal{L}_{\mathtt{pr}}} \times \mathcal{L}_{\mathtt{cl}}$ *es **legal** sssi para cualquier* $\rho \in \mathcal{L}_{\mathtt{pr}}$ *existe un conjunto* $\Phi \subseteq \mathcal{L}_{\mathtt{cl}}$ *tal que* $\Phi \models \rho$.

Creemos que el uso de un marco argumentativo como herramienta para el razonamiento sobre KBs inconsistentes debería ser tan sólo un recurso teórico. Esto implica interpretar la KB directamente como un marco argumentativo sin necesidad, en la práctica, de transformarla a un `GenAF`. Para tal fin, es importante ajustar los elementos atómicos de cada formalismo (KB y `GenAF`), tal que, cada sentencia de la KB entregue un argumento en el `GenAF`. Intuitivamente, un argumento expone una razón para creer en un claim si se da el caso en que sus premisas son soportadas (o satisfechas). Esta intuición es similar a la noción de *material conditionals* o implicaciones lógicas "$\rightarrow$" en lógica clasica. Por ello, necesitamos especificar el formato apropiado de sentencias en una KB $\mathcal{L}^{\kappa}$ con el objetivo de que compartan la misma configuración que los argumentos del `GenAF`.

**Definición 3.1.3 (Forma Normal pre-Argumental)** *Dada una KB* $\Sigma \subseteq \mathcal{L}^{\kappa}$ *y un lenguaje argumental* $\mathbb{A}\mathtt{rgs}$*,* $\Sigma$ *conforma la **forma normal pre-argumental** (*`pANF`*) sssi toda fórmula* $\varphi \in \Sigma$ *corresponde con la forma* $(\bigwedge \Gamma) \rightarrow \alpha$*, donde* $\alpha \in \mathcal{L}_{\mathtt{cl}}$ *y* $\Gamma \subseteq \mathcal{L}_{\mathtt{pr}}$*. En tal caso, cada* $\varphi \in \Sigma$*, tanto como* $\Sigma$*, se dicen estar en* `pANF`*. Una función* $\mathfrak{af} : \mathcal{L}^{\kappa} \longrightarrow \mathcal{L}^{\kappa}$ *es una **función de traducción*** `pANF` *sssi traduce cualquier KB* $\Sigma \subseteq \mathcal{L}^{\kappa}$ *a una KB* `pANF` $\mathfrak{af}(\Sigma)$ *lógicamente equivalente.*

**Observación 3.1.4** *Fórmulas como* $\alpha \in \mathcal{L}_{\mathtt{A}}$ *o* $\alpha \in \mathcal{L}_{\mathtt{cl}}$*, están en* `pANF` *dado que corresponden a la forma* $(\bigwedge \Gamma) \rightarrow \alpha$*, con* $\Gamma = \emptyset$*.*

La siguiente observación especifica la lógica concreta adoptada para el lenguaje argumental utilizado en los ejemplos de este capítulo (hasta la Sección 3.5).

**Observación 3.1.5 (Lógica concreta referida por los ejemplos)** *Se asumirá un lenguaje $\mathcal{L}_{\mathtt{A}}$ conteniendo sólo átomos ground*[2] *y sus negaciones; y a los lenguajes $\mathcal{L}_{\mathtt{cl}}$ y $\mathcal{L}_{\mathtt{pr}}$ concretos de forma que $\mathcal{L}_{\mathtt{cl}}$ permite disyunciones pero prohibe conjunciones, mientras que $\mathcal{L}_{\mathtt{pr}}$ prohibe ambas formas conjuntivas y disyuntivas. Observe que siguiendo tales convenciones, $2^{\mathcal{L}_{\mathtt{pr}}} \times \mathcal{L}_{\mathtt{cl}}$ determina un lenguaje argumental legal de acuerdo con la Definición 3.1.2. Por simplicidad, los ejemplos estarán (en general) comprendidos dentro de $\mathcal{L}^2$, considerando predicados de aridad $\leq 2$.*

**Ejemplo 8** *Sea $\Sigma = \{(p_1(x) \wedge p_2(x)) \vee (p_3(x) \wedge p_4(x)) \rightarrow q_1(x) \wedge (q_2(x) \vee q_3(x)), (p_1(x) \rightarrow \neg p_1(x)), (p_1(x) \vee \neg r(x, y)), r(a, b)\}$ una KB $\mathcal{L}^\kappa$. Su normalización (de acuerdo a la Observación 3.1.5) dejaría una KB* `pANF`*:*

$$\mathfrak{af}(\Sigma) = \{(p_1(x) \wedge p_2(x) \rightarrow q_1(x)),\ (p_1(x) \wedge p_2(x) \rightarrow q_2(x) \vee q_3(x)),$$
$$(p_3(x) \wedge p_4(x) \rightarrow q_1(x)),\ (p_3(x) \wedge p_4(x) \rightarrow q_2(x) \vee q_3(x)),\ (p_1(x) \rightarrow \neg p_1(x)),$$
$$(p_1(x) \vee \neg r(x, y)),\ r(a, b)\}.$$

Observe que la fórmula $(p_1(x) \vee \neg r(x, y))$ en el Ejemplo 8 podría ser transformada en una fórmula $(\neg p_1(x) \rightarrow \neg r(x, y))$. Sin embargo, esto dependerá de las convenciones adoptadas por el algoritmo utilizado para obtener la normalización. Nos abstraeremos de tales cuestiones, y seguidamente, formalizaremos la noción generalizada de argumento en forma independiente de la KB. La relación entre premisas y claims con respecto a una KB es tratada luego en la Observación 3.1.11.

**Definición 3.1.6 (Argumento)** *Un **argumento** $\mathcal{B} \in \mathbb{A}\mathtt{rgs}$ es un par $\langle \Gamma, \alpha \rangle$, donde $\Gamma \subseteq \mathcal{L}_{\mathtt{pr}}$ es un conjunto finito (posiblemente vacío) de premisas, $\alpha \in \mathcal{L}_{\mathtt{cl}}$ su claim, y se verifica la condición $\Gamma \cup \{\alpha\} \not\models \bot$ **(consistencia)**.*

Usualmente, la *evidencia* es considerada una pieza de conocimiento básica e irrefutable. Esto significa que la evidencia no necesita ser soportada, dado que es "auto-justificada" por definición. Por lo tanto, dos opciones surgen para la especificación de la evidencia: como

[2] Recordemos que los átomos ground son fórmulas atómicas que no consideran variables libres.

una entidad separada en el marco, al estilo del marco argumentativo dinámico (RMGS10); o como *argumentos evidenciales*: argumentos sin premisas que satisfacer. En este capítulo, adoptaremos la última postura.

**Definición 3.1.7 (Evidencia)** *Un argumento $\mathcal{B} \in \mathbb{A}\mathtt{rgs}$ es referido como **argumento evidencial** (o evidencia) sssi $\mathcal{B} = \langle \emptyset, \alpha \rangle$ y $\alpha \in \mathcal{L}_{\mathtt{A}}$ (fórmula asercional).*

Los argumentos evidenciales tienen un conjunto vacío de premisas; sin embargo, los argumentos no-evidenciales, *i.e.*, argumentos sin fórmulas asercionales como claims, pueden también contar con un conjunto vacío de premisas. Tales argumentos serán identificados como *primitivos*. Observe que los argumentos primitivos aparecerán dependiendo de la especificación del lenguaje argumental.

**Definición 3.1.8 (Argumento Primitivo)** *Un argumento $\mathcal{B} \in \mathbb{A}\mathtt{rgs}$ es referido como **argumento primitivo** sssi $\mathcal{B} = \langle \emptyset, \alpha \rangle$ y $\alpha \notin \mathcal{L}_{\mathtt{A}}$ (fórmula no-asercional).*

Dado $\mathcal{B} \in \mathbb{A}\mathtt{rgs}$, su claim y conjunto de premisas son indentificados por las funciones $\mathfrak{cl} : \mathbb{A}\mathtt{rgs} \longrightarrow \mathcal{L}_{\mathtt{cl}}$, y $\mathfrak{pr} : \mathbb{A}\mathtt{rgs} \longrightarrow 2^{\mathcal{L}_{\mathtt{pr}}}$, respectivamente. Por ejemplo, dado $\mathcal{B} = \langle \{\rho_1, \rho_2\}, \alpha \rangle$, sus premisas son $\mathfrak{pr}(\mathcal{B}) = \{\rho_1, \rho_2\}$, y su claim, $\mathfrak{cl}(\mathcal{B}) = \alpha$. Los argumentos serán ontenidos de fórmulas pANF a través de una *función de traducción argumental* "$\mathfrak{arg}$" definida como sigue.

**Definición 3.1.9 (Traducción Argumental)** *Una función $\mathfrak{arg} : \mathcal{L}^{\kappa} \longrightarrow \mathbb{A}\mathtt{rgs}$ es una **traducción argumental** si verifica $\mathfrak{arg}(\varphi) = \langle \Gamma, \alpha \rangle$ sssi $\varphi \in \mathcal{L}^{\kappa}$ es una fórmula* pANF *$(\bigwedge \Gamma) \rightarrow \alpha$. De lo contrario, $\mathfrak{arg}(\varphi) = \langle \emptyset, \bot \rangle$.*

**Ejemplo 9 (Continúa del Ejemplo 8)** *Para las fórmulas en $\mathfrak{af}(\Sigma)$, los siguientes argumentos son obtenidos por efecto de la función "$\mathfrak{arg}$":*

$\mathcal{B}_1 = \langle \{p_1(x), p_2(x)\},\ q_1(x) \rangle$

$\mathcal{B}_2 = \langle \{p_1(x), p_2(x)\}, q_2(x) \vee q_3(x) \rangle$

$\mathcal{B}_3 = \langle \{p_3(x), p_4(x)\},\ q_1(x) \rangle$

$\mathcal{B}_4 = \langle \{p_3(x), p_4(x)\}, q_2(x) \vee q_3(x) \rangle$

$\mathcal{B}_5 = \langle \{p_1(x)\}, \neg p_1(x) \rangle$

$\mathcal{B}_6 = \langle \{\}, p_1(x) \vee \neg r(x, y) \rangle$ *(primitivo)*

$\mathcal{B}_7 = \langle \emptyset, r(a, b) \rangle$ *(evidencia)*

La siguiente observación muestra un caso especial de concretizado para el lenguaje argumental que anulará el uso de premisas. Este caso específico será observado a lo largo de todo el capítulo.

**Observación 3.1.10** *Para cualquier KB* $\Sigma \subseteq \mathcal{L}^{\kappa}$*, si* $\mathcal{L}_{\mathtt{cl}} = \mathcal{L}^{\kappa}$ *entonces* $\Sigma$ *está en* `pANF` *y para toda* $\varphi \in \Sigma$*, si* $\mathfrak{arg}(\varphi) \in \mathbb{A}\mathtt{rgs}$ *entonces* $\mathfrak{arg}(\varphi)$ *será primitivo o bien evidencial.*

Como mencionamos anteriormente, es importante recordar que la noción de argumento adoptada en este capítulo difiere de la usual. Aquí los argumentos son también indivisibles pero dado que aparecen en representación de las fórmulas de la KB subyacente, juegan un rol más pequeño en el `GenAF`: no son auto-conclusivos (excepto para el caso de argumentos evidenciales y primitivos), sino que son agrupados dentro de estructuras[3] que serán interpretadas como argumentos clásicos (en la cual sus contenidos interactúan como una justificación lógica para su claim). Este punto será aclarado luego en la Sección 3.2.

**Observación 3.1.11** *Dada una KB* `pANF` $\Sigma \subseteq \mathcal{L}^{\kappa}$*, una fórmula* $\varphi \in \Sigma$*, y su argumento asociado* $\mathfrak{arg}(\varphi) = \langle \Gamma, \alpha \rangle$*; se verifica* $\Sigma \models (\bigwedge \Gamma) \rightarrow \alpha$*, pero no necesariamente* $\Gamma \models \alpha$*.*

**Definición 3.1.12 (`GenAF`)** *Un* ***marco argumentativo abstracto generalizado*** *(*`GenAF`*) es un par* $\langle \mathbf{A}, \mathbf{R_A} \rangle$*, donde* $\mathbf{A} \subseteq \mathbb{A}\mathtt{rgs}$ *es un conjunto finito de argumentos, y* $\mathbf{R_A} \subseteq \mathbf{A} \times \mathbf{A}$*, una relación de ataque finita. El dominio de* `GenAF`*s es identificado a través del conjunto* $\mathbb{G}$*, y para todo argumento evidencial y primitivo contenido en* $\mathbf{A}$*, sus dominios se identifican a través de los conjuntos* $\mathbf{E} \subseteq \mathbf{A}$ *y* $\mathbf{P} \subseteq \mathbf{A}$*, respectivamente.*

Seguidamente definimos la *función de teorías* "$\mathfrak{genaf}$" para identificar el `GenAF` asociado a una KB $\Sigma \subseteq \mathcal{L}^{\kappa}$. Para tal fin, nos basaremos sobre la función de traducción `pANF` (ver Definición 3.1.3). El conjunto de argumentos conformando el `GenAF` $\mathfrak{genaf}(\Sigma)$ contendrá las estructuras obtenidas a través de $\mathfrak{arg}(\varphi)$, donde $\varphi \in \mathfrak{af}(\Sigma)$ ó $\varphi^{-} \in \mathfrak{af}(\Sigma)$, tal que $\mathfrak{arg}(\varphi)$ es un argumento de acuerdo a la Definición 3.1.6, y $\varphi^{-}$ es la *contrapositiva* de $\varphi$. Observe que los argumentos resultantes de las fórmulas contrapositivas de la KB serán necesarios. Esto es natural dado que las contrapositivas están implicitamente consideradas para el razonamiento clásico en FOL. Sin embargo, en un `GenAF`, esto se hace explícito debido a la naturaleza de la estructura utilizada para la representación de argumentos. A partir de ahora, identificaremos la contrapositiva de una fórmula $\varphi \in \mathcal{L}^{\kappa}$ como $\varphi^{-}$.

**Definición 3.1.13 (Función de Teorías)** *Dada una KB* $\Sigma \subseteq \mathcal{L}^{\kappa}$*, una* ***función de teorías*** $\mathfrak{genaf} : 2^{\mathcal{L}^{\kappa}} \longrightarrow \mathbb{G}$ *identifica un* `GenAF` $\mathfrak{genaf}(\Sigma) = \langle \mathbf{A}, \mathbf{R_A} \rangle$*, donde* $\mathbf{A} = \{\mathfrak{arg}(\varphi) \in \mathbb{A}\mathtt{rgs} | \varphi \in \mathfrak{af}(\Sigma)$ *ó* $\varphi^{-} \in \mathfrak{af}(\Sigma)\}$*.*

[3] Ver *estructuras argumentales*, Definición 3.2.8.

**Ejemplo 10 (Continúa del Ejemplo 9)** *La función de teorías determina el* GenAF $\mathfrak{genaf}(\Sigma) = \langle \mathbf{A}, \mathbf{R}_{\mathbf{A}} \rangle$*, donde* $\mathbf{A} = \{\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4, \mathcal{B}_6, \mathcal{B}_7, \mathcal{B}'_1, \mathcal{B}'_2, \mathcal{B}'_3, \mathcal{B}'_4\}$.

$\mathcal{B}_1 = \langle \{p_1(x), p_2(x)\},\ q_1(x) \rangle$

$\mathcal{B}_2 = \langle \{p_1(x), p_2(x)\}, q_2(x) \vee q_3(x) \rangle$

$\mathcal{B}_3 = \langle \{p_3(x), p_4(x)\},\ q_1(x) \rangle$

$\mathcal{B}_4 = \langle \{p_3(x), p_4(x)\}, q_2(x) \vee q_3(x) \rangle$

$\mathcal{B}_5 = \langle \{p_1(x)\}, \neg p_1(x) \rangle$

$\mathcal{B}_6 = \langle \emptyset, p_1(x) \vee \neg r(x,y) \rangle$

$\mathcal{B}_7 = \langle \emptyset, r(a,b) \rangle$

$\mathcal{B}'_1 = \langle \{\neg q_1(x)\}, \neg p_1(x) \vee \neg p_2(x) \rangle$

$\mathcal{B}'_2 = \langle \{\neg q_2(x), \neg q_3(x)\}, \neg p_1(x) \vee \neg p_2(x) \rangle$

$\mathcal{B}'_3 = \langle \{\neg q_1(x)\},\ \neg p_3(x) \vee \neg p_4(x) \rangle$

$\mathcal{B}'_4 = \langle \{\neg q_2(x), \neg q_3(x)\}, \neg p_3(x) \vee \neg p_4(x) \rangle$

$\mathcal{B}'_6 = \langle \{\neg p_1(x), r(x,y)\}, \bot \rangle$

$\mathcal{B}'_7 = \langle \{\neg r(a,b)\}, \bot \rangle$

*Observe que* $\mathcal{B}_5 = \mathfrak{arg}(p_1(x) \rightarrow \neg p_1(x))$, $\mathcal{B}'_6 = \mathfrak{arg}((p_1(x) \vee \neg r(x,y))^-)$, *y* $\mathcal{B}'_7 = \mathfrak{arg}((r(a,b))^-)$*, no están contenidos en* $\mathbf{A}$ *dado que ninguno de ellos está contenido en* $\mathbb{A}\mathtt{rgs}$*, i.e., no son argumentos (ver consistencia en la Definición 3.1.6).*

## 3.2. La Maquinaria Argumentativa GenAF

El propósito de generalizar un marco argumentativo abstracto se basa en la necesidad de manejar diferentes lenguajes argumentales especificados a través de diferentes fragmentos FOL. Dada la especificación de $\mathbb{A}\mathtt{rgs}$, varias posibilidades pueden surgir, como por ejemplo, el lenguaje adoptado para claims podría aceptar disyunsión de fórmulas. Por ello, es posible inferir una fórmula en $\mathcal{L}_{\mathtt{cl}}$ a partir de los claims de argumentos del GenAF agrupados. Consideremos como ejemplo dos argumentos $\langle \{p_1(x)\}, q_1(x) \vee\ q_2(x) \rangle$ y $\langle \{p_2(x)\}, \neg q_2(x) \rangle$, el claim $q_1(x)$ podría ser inferido. Observe que este tipo de construcciones son similares a los argumentos, pero en contraste, son obtenidas del GenAF en forma implícita. En forma intuitiva, diremos entonces que la noción de *coalición-claiming* identifica *un conjunto de argumentos agrupados en pos de la inferencia de un nuevo claim.*

Para especificar en profundidad la noción de coalición, nos basaremos en las funciones $\mathfrak{clset} : 2^{\mathbb{A}\mathtt{rgs}} \longrightarrow 2^{\mathcal{L}_{\mathtt{cl}}}$ y $\mathfrak{prset} : 2^{\mathbb{A}\mathtt{rgs}} \longrightarrow 2^{\mathcal{L}_{\mathtt{pr}}}$, definidas como $\mathfrak{clset}(\widehat{\mathcal{C}}) = \{\mathfrak{cl}(\mathcal{B}) | \mathcal{B} \in \widehat{\mathcal{C}}\}$, y $\mathfrak{prset}(\widehat{\mathcal{C}}) = \bigcup_{\mathcal{B} \in \widehat{\mathcal{C}}} \mathfrak{pr}(\mathcal{B})$, para identificar respectivamente el conjunto de claims y premisas de un conjunto $\widehat{\mathcal{C}} \subseteq \mathbb{A}\mathtt{rgs}$ de argumentos. En forma intuitiva, una *coalition* puede ser definida como *un conjunto mínimo y consistente de argumentos garantizando la verificación de una meta determinada*. Diremos que $\widehat{\mathcal{C}} \subseteq \mathbb{A}\mathtt{rgs}$ es consistente *sssi* $\mathfrak{prset}(\widehat{\mathcal{C}}) \cup \mathfrak{clset}(\widehat{\mathcal{C}}) \not\models \bot$, mientras que por medio de la noción de minimalidad se asegura que $\widehat{\mathcal{C}}$ verifique una *meta*

$\theta$ *sssi* no existe subconjunto propio de $\widehat{\mathcal{C}}$ verificando $\theta$. Dependiendo de la naturaleza de la meta $\theta$, dos tipos de coaliciones diferentes serán identificadas: *coaliciones-claiming* y *coaliciones-soporte*. Para el primer caso, la meta $\theta$ será una nueva inferencia en $\mathcal{L}_{\texttt{cl}}$ a partir de los argumentos considerados por la coalición $\widehat{\mathcal{C}}$, mientras que para el caso de las coaliciones-soporte, la meta $\theta$ es la obtención de un conjunto de claims –a partir de los argumentos considerados en $\widehat{\mathcal{C}}$– infiriendo (soportando) una premisa en $\mathcal{L}_{\texttt{pr}}$ correspondiente a un argumento del `GenAF`.

**Definición 3.2.1 (Coalición-claiming)** *Dado un* `GenAF` $\langle \mathbf{A}, \mathbf{R_A} \rangle \in \mathbb{G}$*, y una fórmula* $\alpha \in \mathcal{L}_{\texttt{cl}}$*, un conjunto de argumentos* $\widehat{\mathcal{C}} \subseteq \mathbf{A}$ *es una* ***coalición-claiming****, o simplemente un* ***claimer****, de* $\alpha$ *sssi las siguientes condiciones son verificadas:*

**(inferencia)** $\widehat{\mathcal{C}}$ *infiere* $\alpha$*, i.e.,* $\mathfrak{clset}(\widehat{\mathcal{C}}) \models \alpha$*,*

**(consistencia)** $\widehat{\mathcal{C}}$ *es consistente, i.e.,* $\mathfrak{prset}(\widehat{\mathcal{C}}) \cup \mathfrak{clset}(\widehat{\mathcal{C}}) \not\models \bot$*, y*

**(minimalidad)** $\widehat{\mathcal{C}}$ *es mínima, i.e., no existe subconjunto* $X \subset \widehat{\mathcal{C}}$ *verificando* (inferencia)*.*

Una coalición-claiming conteniendo un único argumento $\langle \Gamma, \alpha \rangle$ es referida como *coalición primitiva para* $\alpha$. Como fuera mencionado anteriormente, un argumento requiere que sus premisas sean soportadas como parte funcional del proceso de razonamiento para alcanzar su claim. En este marco, debido a la caracterización del lenguaje argumental $\mathbb{A}\texttt{rgs}$, a veces una fórmula en $\mathcal{L}_{\texttt{pr}}$ puede ser satisfecha únicamente a través de varias fórmulas $\mathcal{L}_{\texttt{cl}}$. Esto significa que un simple argumento no siempre será suficiente para soportar una premisa de otro argumento. Por ello, extenderemos la definición usual de soporte (RMGS10) mediante la noción de *coalición-soporte*.

**Definición 3.2.2 (Coalición-soporte)** *Dado un* `GenAF` $\langle \mathbf{A}, \mathbf{R_A} \rangle \in \mathbb{G}$*, un argumento* $\mathcal{B} \in \mathbf{A}$*, y una premisa* $\rho \in \mathfrak{pr}(\mathcal{B})$*. Un conjunto de argumentos* $\widehat{\mathcal{C}} \subseteq \mathbf{A}$ *es una* ***coalición-soporte****, o simplemente un* ***supporter****, de* $\mathcal{B}$ *en* $\rho$ *sssi las siguientes condiciones son verificadas:*

**(inferencia)** $\widehat{\mathcal{C}}$ *infiere* $\rho$*, i.e.,* $\mathfrak{clset}(\widehat{\mathcal{C}}) \models \rho$*,*

**(consistencia)** $\widehat{\mathcal{C}} \cup \{\mathcal{B}\}$ *es consistente, i.e.,* $\mathfrak{prset}(\widehat{\mathcal{C}} \cup \{\mathcal{B}\}) \cup \mathfrak{clset}(\widehat{\mathcal{C}} \cup \{\mathcal{B}\}) \not\models \bot$*, y*

**(minimalidad)** $\widehat{\mathcal{C}}$ *es mínima, i.e., no existe subconjunto* $X \subset \widehat{\mathcal{C}}$ *verificando* (inferencia)*.*

**Ejemplo 11** *Asuma* $\mathbf{A} = \{\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4\}$*, donde* $\mathcal{B}_1 = \langle\{p_1(x)\}, q_1(x)\rangle$*,* $\mathcal{B}_2 = \langle\{p_1(x)\}, q_2(x)\rangle$*,* $\mathcal{B}_3 = \langle\{p_2(x)\}, p_1(x) \vee q_1(x)\rangle$*, y* $\mathcal{B}_4 = \langle\{p_3(x)\}, \neg q_1(x)\rangle$*. El conjunto* $\widehat{\mathcal{C}} = \{\mathcal{B}_3, \mathcal{B}_4\}$ *es un supporter de* $\mathcal{B}_2$*. Observe que* $\widehat{\mathcal{C}}$ *no puede ser una coalición-soporte de* $\mathcal{B}_1$ *dado que violaría (supporter) consistency (ver Definición 3.2.2).*

Cuando no se encuentran en $\mathbf{A}$ todos los argumentos necesarios para conformar una coalición-soporte, la premisa no-soportada es referida como *premisa libre*.

**Definición 3.2.3 (Premisa Libre)** *Dado un* `GenAF` $\langle \mathbf{A}, \mathbf{R}_{\mathbf{A}} \rangle \in \mathbb{G}$ *y un argumento* $\mathcal{B} \in \mathbf{A}$*, una premisa* $\rho \in \mathfrak{pr}(\mathcal{B})$ *es* ***libre*** *con respecto a* $\mathbf{A}$ *sssi no existe coalición-soporte* $\widehat{\mathcal{C}} \subseteq \mathbf{A}$ *de* $\mathcal{B}$ *en* $\rho$*.*

Observe que en el Ejemplo 11, las premisas $p_2(x) \in \mathfrak{pr}(\mathcal{B}_3)$, $p_3(x) \in \mathfrak{pr}(\mathcal{B}_4)$, y $p_1(x) \in \mathfrak{pr}(\mathcal{B}_1)$ son libres con respecto a $\mathbf{A}$; mientras que, por el contrario, la premisa $p_1(x) \in \mathfrak{pr}(\mathcal{B}_2)$ no lo es.

Cuando un argumento es soportado completamente por evidencia ($\widehat{\mathcal{C}} \subseteq \mathbf{E}$), su claim es finalmente instanciado determinando una fórmula ground. Por ello, un argumento $\mathcal{B}$ puede ser incluido en una coalición-soporte $\widehat{\mathcal{C}}$ de $\mathcal{B}$ mismo debido a la sustitución de variables. Esta situación será aclarada luego en el Teorema 3.2.12 e ilustrada en el Ejemplo 13. La búsqueda de un supporter $\widehat{\mathcal{C}}$ de algún argumento $\mathcal{B}$ en una premisa $\rho \in \mathcal{B}$, describe un proceso recursivo de soporte dado que cada premisa en $\widehat{\mathcal{C}}$ necesita ser soportada. Cuando este proceso finalmente concluye en un supporter conteniendo sólo argumentos evidenciales, distinguiremos a cada premisa (no-libre) $\rho \in \mathfrak{pr}(\mathcal{B})$ como *cerrada*.

**Definición 3.2.4 (Premisa Cerrada)** *Dado un* `GenAF` $\langle \mathbf{A}, \mathbf{R}_{\mathbf{A}} \rangle \in \mathbb{G}$*, y un argumento* $\mathcal{B} \in \mathbf{A}$*, una premisa* $\rho \in \mathfrak{pr}(\mathcal{B})$ *es* ***cerrada*** *con respecto a* $\mathbf{A}$ *sssi existe un supporter* $\widehat{\mathcal{C}} \subseteq \mathbf{A}$ *de* $\mathcal{B}$ *en* $\rho$ *tal que o bien* $\mathfrak{prset}(\widehat{\mathcal{C}}) = \emptyset$*, o toda premisa en* $\mathfrak{prset}(\widehat{\mathcal{C}})$ *es cerrada.*

La idea trás el cerrado de premisas es identificar aquellos argumentos que efectivamente exponen una razón (a partir del `GenAF`) para creer en sus claims. Tales argumentos serán aquellos para los cuales el soporte de cada una de sus premisas finalmente concluye en un conjunto de argumentos evidenciales –y, por lo tanto, no más premisas requerirán ser soportadas. Por ello, toda premisa en un argumento es cerrada *sssi* el claim es *inferible*. Esto es natural dado que los claims inferibles pueden ser efectivamente alcanzados a partir de evidencia. Finalmente, cuando la coalición-claiming de un claim inferible pasa el análisis de aceptabilidad (mediante la semántica argumentativa), el claim finaliza *garantizado*. Las nociones de aceptabilidad y garantía de argumentos en un $\mathtt{GenAF}$ serán detalladas luego, en la Sección 3.3.3.

**Definición 3.2.5 (Fórmula Inferible)** *Dado un* $\mathtt{GenAF}$ $\langle \mathbf{A}, \mathbf{R}_{\mathbf{A}} \rangle \in \mathbb{G}$, *una fórmula* $\alpha \in \mathcal{L}_{\mathtt{cl}}$ *es **inferible** de* $\mathbf{A}$ *sssi existe una coalición-claiming* $\widehat{\mathcal{C}} \subseteq \mathbf{A}$ *para* $\alpha$ *tal que o bien,* $\mathfrak{prset}(\widehat{\mathcal{C}}) = \emptyset$, *o toda premisa en* $\mathfrak{prset}(\widehat{\mathcal{C}})$ *es cerrada.*

El proceso de soporte para cerrar toda premisa de una coalición-claiming $\widehat{\mathcal{C}}$ con el objetivo de verificar si el claim es inferible, conforma un árbol enraizado en la coalición $\widehat{\mathcal{C}}$, al que llamaremos *árbol de soporte*. Por otro lado, cada rama en un árbol de soporte será identificada como *cadena de soporte*.

**Definición 3.2.6 (Cadena de Soporte)** *Dado un* $\mathtt{GenAF}$ $\langle \mathbf{A}, \mathbf{R}_{\mathbf{A}} \rangle \in \mathbb{G}$, *una fórmula* $\alpha \in \mathcal{L}_{\mathtt{cl}}$, *y una secuencia* $\lambda \in (2^{\mathbf{A}})^n$ *tal que* $\lambda = \widehat{\mathcal{C}}_1 \ldots \widehat{\mathcal{C}}_n$, *donde* $n \in \mathbb{N}_1$, $\widehat{\mathcal{C}}_1$ *es una coalición-claiming para* $\alpha$, *y para todo* $0 < i \leq n$ *se verifica* $\widehat{\mathcal{C}}_i \subseteq \mathbf{A}$, *y* $\widehat{\mathcal{C}}_{i+1}$ *es una coalición-soporte de alguna* $\rho \in \mathfrak{prset}(\widehat{\mathcal{C}}_i)$.

*Las notaciones* $|\lambda|$ *y* $\lambda[i]$ *son utilizadas para identificar la **longitud*** $n$ *de* $\lambda$ *y el **nodo*** $\widehat{\mathcal{C}}_i$ *en ella, respectivamente. La última coalición-soporte en* $\lambda$, *llamada **hoja**, es identificada por la función* $\mathfrak{leaf}(\lambda) = \lambda[|\lambda|]$. *La función* $\overleftarrow{\lambda} : (2^{\mathbf{A}})^n \times \mathbb{N}_0 \longrightarrow \mathcal{L}_{\mathtt{cl}} \cup \mathcal{L}_{\mathtt{pr}} \cup \{\bot\}$ *identifica el **link*** $\overleftarrow{\lambda}[0] = \alpha$; *ó* $\overleftarrow{\lambda}[i] = \rho$ $(0 < i < |\lambda|)$, *donde* $\rho \in \mathfrak{prset}(\lambda[i])$ *es soportado por* $\lambda[i+1]$; *ó* $\overleftarrow{\lambda}[i] = \bot$ $(i \geq |\lambda|)$. *El conjunto* $\lambda^* = \bigcup_i \lambda[i]$ *(con* $0 < i \leq |\lambda|$*) identifica al conjunto de argumentos incluido en* $\lambda$.

*Finalmente,* $\lambda$ *es una **cadena de soporte para*** $\alpha$ *con respecto a* $\mathbf{A}$ *sssi se verifica simultáneamente:*

**(minimalidad)** $\widehat{\mathcal{C}} \subseteq \lambda^*$ *es supporter (claimer si* $i = 0$*) de* $\overleftarrow{\lambda}[i]$ *sssi* $\widehat{\mathcal{C}} = \lambda[i+1]$ $(0 \leq i < |\lambda|)$

**(exhaustividad)** *toda* $\rho \in \mathfrak{prset}(\mathfrak{leaf}(\lambda))$ *es libre con respecto a* $\lambda^*$

**(aciclicidad)** $\overleftarrow{\lambda}[i] = \overleftarrow{\lambda}[j]$ *sssi* $i = j$, *con* $\{i, j\} \subseteq \{0, \ldots, |\lambda| - 1\}$

**(consistencia)** $\mathfrak{prset}(\lambda^*) \cup \mathfrak{clset}(\lambda^*) \not\models \bot$

De la Definición 3.2.6 podemos decir que, una cadena de soporte es una secuencia finita de coaliciones-soporte $\widehat{\mathcal{C}}_i$ de un link $\rho \in \mathfrak{prset}(\widehat{\mathcal{C}}_i)$ soportadas por $\widehat{\mathcal{C}}_{i+1}$. La cadena es finita dado que el conjunto $\mathbf{A}$ también es finito, y que ningún link puede ser repetido en la cadena (aciclicidad). La condición de minimalidad (con respecto a inclusión "$\subseteq$" sobre $\lambda^*$) sirve a la consideración de la menor cantidad posible de argumentos diferentes tomados de $\mathbf{A}$ para obtener la misma cadena, mientras que la condición de exhaustividad (con respecto a la longitud $|\lambda|$) asegura que la cadena es tan larga como es posible considerando argumentos de $\lambda^*$ (sin ciclos), esto es, $\lambda$ tiene todos los links posibles que puedan aparecer considerando los argumentos que construyen la cadena. Observe que a partir de minimalidad, ningún par de argumentos para un mismo claim puede ser considerado en forma simultánea por una cadena de soporte. Finalmente, consistencia es requerida dado que la intención de la cadena de soporte es proveer una herramienta para cerrar una premisa de la coalición-claiming. A continuación, los *árboles de soporte* son formalizados basandonos en la definición de cadena de soporte.

**Definición 3.2.7 (Árbol de Soporte)** *Dado un* GenAF $\langle \mathbf{A}, \mathbf{R_A} \rangle \in \mathbb{G}$*, una fórmula* $\alpha \in \mathcal{L}_{\mathtt{cl}}$*, y un árbol* $\mathcal{T}$ *de coaliciones* $\widehat{\mathcal{C}} \subseteq \mathbf{A}$ *tal que cada nodo* $\widehat{\mathcal{C}}$ *es:*

- ***la raíz*** *sssi* $\widehat{\mathcal{C}}$ *es una coalición-claiming para* $\alpha$*; o*
- ***un nodo interno*** *sssi* $\widehat{\mathcal{C}}$ *es una coalición-soporte de* $\rho \in \mathfrak{prset}(\widehat{\mathcal{C}'})$*, dode* $\widehat{\mathcal{C}'} \subseteq \mathbf{A}$ *es, o bien un nodo interno, o el nodo raíz.*

*La relación de pertenencia será sobrecargada escribiendo* $\lambda \in \mathcal{T}$ *y* $\widehat{\mathcal{C}} \in \mathcal{T}$ *para identificar respectivamente la rama* $\lambda$ *y el nodo* $\widehat{\mathcal{C}}$ *de* $\mathcal{T}$*. El conjunto* $\mathcal{T}^* = \bigcup_{\widehat{\mathcal{C}} \in \mathcal{T}} \widehat{\mathcal{C}}$ *identifica el conjunto de argumentos incluido en* $\mathcal{T}$*. Luego,* $\mathcal{T}$ *es un* ***árbol de soporte*** *sssi se verifica simultáneamente:*

**(completitud)** *toda* $\lambda \in \mathcal{T}$ *es una cadena de soporte de* $\alpha$ *con respecto a* $\mathbf{A}$

**(minimalidad)** *para toda* $\lambda \in \mathcal{T}$, $\widehat{\mathcal{C}} \subseteq \mathcal{T}^*$ *es un supporter (claimer si* $i = 0$*) de* $\overleftarrow{\lambda}[i]$ *sssi* $\widehat{\mathcal{C}} = \lambda[i+1]$ *(*$0 \leq i < |\lambda|$*)*

**(exhaustividad)** *para toda* $\rho \in \mathfrak{prset}(\mathcal{T}^*)$*, si no existe* $\lambda \in \mathcal{T}$ *tal que* $\overleftarrow{\lambda}[i] = \rho$ *(*$0 < i < |\lambda|$*) entonces* $\rho$ *es libre con respecto a* $\mathcal{T}^*$

**(consistencia)** $\mathfrak{prset}(\mathcal{T}^*) \cup \mathfrak{clset}(\mathcal{T}^*) \not\models \bot$

*Finalmente, la notación* $\mathfrak{Trees}_{\mathbf{A}}(\alpha)$ *identifica al conjunto de todos los árboles de soporte para* $\alpha$ *construibles de* $\mathbf{A}$.

La condición de completitud es requerida para restringir la construcción del árbol de soporte al uso único de cadenas de soporte como sus ramas. De forma similar a la cadena de soporte, la minimalidad es requerida para evitar la consideración de argumentos extra para la construcción del árbol, mientras que exhaustividad asegura que toda posible coalición-soporte $\widehat{\mathcal{C}} \subseteq \mathcal{T}^*$ de una premisa en $\mathfrak{prset}(\mathcal{T}^*)$ es un nodo interno en el árbol. Finalmente, consistencia asegura que el proceso completo de soporte de las premisas correspondientes a la coalición-claiming concluirá siendo no-contradictorio, aún considerando todas las ramas del árbol. Es importante notar que un árbol de soporte para $\alpha \in \mathcal{L}_{\mathtt{cl}}$ determina el conjunto de argumentos usado en el (posiblemente *inconcluso*) proceso de soporte de una coalición-claiming de $\alpha$. Tal conjunto será denominado *estructura*.

**Definición 3.2.8 (Estructura)** *Dado un* GenAF $\langle \mathbf{A}, \mathbf{R}_{\mathbf{A}} \rangle \in \mathbb{G}$*, y una fórmula* $\alpha \in \mathcal{L}_{\mathtt{cl}}$*, un conjunto* $\mathbb{S} \subseteq \mathbf{A}$ *identifica una* ***estructura*** *para* $\alpha$ *sssi existe un árbol de soporte* $\mathcal{T} \in \mathfrak{Trees}_{\mathbf{A}}(\alpha)$ *para* $\alpha$ *tal que* $\mathbb{S} = \mathcal{T}^*$*. El claim y premisas de* $\mathbb{S}$ *pueden ser respectivamente determinados por las funciones* $\mathfrak{cl} : 2^{\mathbb{A}\mathrm{rgs}} \longrightarrow \mathcal{L}_{\mathtt{cl}}$ *y* $\mathfrak{pr} : 2^{\mathbb{A}\mathrm{rgs}} \longrightarrow 2^{\mathcal{L}_{\mathtt{pr}}}$*, tal que* $\mathfrak{cl}(\mathbb{S}) = \alpha$ *y* $\mathfrak{pr}(\mathbb{S}) = \{\rho \in \mathfrak{prset}(\mathbb{S}) \mid \rho$ *es una premisa libre con respecto a* $\mathbb{S}\}$*. Finalmente, la estructura* $\mathbb{S}$ *es* ***argumental*** *sssi* $\mathfrak{pr}(\mathbb{S}) = \emptyset$*; de lo contrario,* $\mathbb{S}$ *es* ***esquemática****.*

Observe que los procesos de soporte inconclusos son aquellos que determinan estructuras esquemáticas (premisas no-cerradas con respecto a la estructura misma pero libres con respecto a $\mathbf{A}$) dado que tales procesos no dan en forma efectiva con la clausura de las premisas del claimer. La formación de tales estructuras será útil para el reconocimiento de *incoherencias* (definido sobre ontologías en la Sección 2.4.8) de la KB subyacente. Estas nociones serán aclaradas luego en la Sección 3.3.2.

Las funciones "$\mathfrak{pr}$" y "$\mathfrak{cl}$" son sobrecargadas para ser aplicadas tanto a argumentos como a estructuras. Esto no será problemático dado que sus usos diferentes serán explícitos por la misma notación. Cuando sea necesario sobrecargaremos la notación para estructuras escribiendo $\mathbb{S} = \langle \Psi, \Phi, \alpha \rangle$, donde $\Psi \subseteq \mathcal{L}_{\mathtt{pr}}$, $\Phi \subseteq \mathbf{A}$, y $\alpha \in \mathcal{L}_{\mathtt{cl}}$; para identificar la estructura $\mathbb{S} = \Phi$ tal que $\mathfrak{pr}(\mathbb{S}) = \Psi$ y $\mathfrak{cl}(\mathbb{S}) = \alpha$. Además, una estructura $\mathbb{S}$ es referida como *primitiva sssi* $|\mathbb{S}| = 1$, *i.e.*, contiene un único argumento. Por lo tanto, si $\mathbb{S} = \{\mathcal{B}\}$ entonces $\mathfrak{pr}(\mathcal{B}) = \mathfrak{pr}(\mathbb{S})$ y $\mathfrak{cl}(\mathcal{B}) = \mathfrak{cl}(\mathbb{S})$. Sin embargo, no todo argumento puede tener una estructura primitiva asociada. Por ejemplo, ninguna estructura podrá contener un argumento $\langle \{p(x)\}, p(x) \rangle$ dado que violaría la condición de aciclicidad de cadena de soporte (ver Definición 3.2.6). Finalmente, cuando no sea necesaria su distinción, nos referiremos a las estructuras primitivas, esquemáticas, o argumentales, simplemente como estructuras.

**Ejemplo 12** *Dados dos argumentos* $\mathcal{B}_1 = \langle\{p(x)\}, q(x)\rangle$ *y* $\mathcal{B}_2 = \langle\{q(x)\}, p(x)\rangle$*. El conjunto* $\{\mathcal{B}_1, \mathcal{B}_2\}$ *no puede ser una estructura para* $q(x)$ *dado que* $\{\mathcal{B}_1\}\{\mathcal{B}_2\}\{\mathcal{B}_1\}\ldots$ *violaría la condición de aciclicidad de cadena de soporte (Definición 3.2.6). Similarmente,* $\{\mathcal{B}_1, \mathcal{B}_2\}$ *tampoco podrá ser una estructura para* $p(x)$*.*

**Observación 3.2.9 (Continúa de la Observación 3.1.10)** *Dada una KB* $\Sigma \subseteq \mathcal{L}^\kappa$ *y su asociado* `GenAF` $\langle \mathbf{A}, \mathbf{R}_{\mathtt{A}}\rangle = \mathfrak{genaf}(\Sigma)$*, si* $\mathcal{L}_{\mathtt{cl}} = \mathcal{L}^\kappa$ *entonces cualquier estructura* $\mathbb{S} \subseteq \mathbf{A}$ *es argumental y su árbol de soporte contiene un único nodo, i.e., la raíz, la cual es una coalición-claiming conteniendo sólo argumentos evidenciales y/o primitivos.*

La observación anterior muestra una especificación para $\mathbb{A}\mathtt{rgs}$ determinando un marco alternativo "primitivo" en el cual el uso de premisas, coalición-soportes, y estructuras esquemáticas son descartadas. Esto permitirá la formación de estructuras con claims de expresividad máxima (cualquier fórmula en $\mathcal{L}^\kappa$). Sin embargo, como es usual en lógica, esto disminuirá la eficiencia del razonamiento, dado que las ventajas obtenidas del uso de KBs normalizadas serán relegadas. En la Sección 3.4, será claro que este tipo de seteo puede también complicar el proceso de debugging de KBs, haciéndolas más rígidas para la reparación de sus fórmulas.

**Lema 3.2.10** *Dado un* `GenAF` $\langle \mathbf{A}, \mathbf{R}_{\mathtt{A}}\rangle \in \mathbb{G}$*, una estructura* $\mathbb{S} \subseteq \mathbf{A}$ *para* $\alpha \in \mathcal{L}_{\mathtt{cl}}$*, y su árbol de soporte* $\mathcal{T} \in \mathfrak{Trees}_{\mathbb{S}}(\alpha)$*; si* $\mathbb{S}$ *es argumental entonces* $\mathfrak{leaf}(\lambda) \subseteq (\mathbf{E} \cup \mathbf{P})$*, para toda* $\lambda \in \mathcal{T}$*.*

Observe que la recíproca del lema anterior es en general inválida dado que aunque $\mathfrak{leaf}(\lambda) \subseteq \mathbf{E}$ es verificado para toda $\lambda \in \mathcal{T}$, es posible que aparezcan premisas libres $\rho \in \mathfrak{prset}(\widehat{\mathcal{C}})$, para algún $\widehat{\mathcal{C}} = \lambda[i]$ $(0 < i < |\lambda|)$ (ver también exhaustividad de árbol de soporte en la Definición 3.2.7). En consecuencia, $\mathfrak{leaf}(\lambda) \subseteq \mathbf{E}$, para toda $\lambda \in \mathcal{T}$ donde $\mathcal{T} \in \mathfrak{Trees}_{\mathbb{S}}(\alpha)$, no implica $\mathfrak{pr}(\mathbb{S}) = \emptyset$, y por lo tanto, $\mathbb{S}$ no es argumental.

**Lema 3.2.11** *Dado un* `GenAF` $\langle \mathbf{A}, \mathbf{R}_{\mathtt{A}}\rangle \in \mathbb{G}$*, y una fórmula* $\alpha \in \mathcal{L}_{\mathtt{cl}}$*, existe una estructura argumental* $\mathbb{S} \subseteq \mathbf{A}$ *para* $\alpha$ *sssi* $\alpha$ *es inferible.*

Para el siguiente teorema, consideraremos los vectores $\bar{x} = x_1, \ldots, x_n$ y $\bar{a} = a_1, \ldots, a_n$, donde cada $x_i \in \{\bar{x}\}$ es una variable libre y cada $a_i \in \{\bar{a}\}$ es un individuo constante.

Del Lema 3.2.11, si una fórmula $\varphi(\bar{x}) \in \mathcal{L}_{\mathtt{cl}}$ es inferible entonces existe una estructura argumental $\mathbb{S}$ para $\varphi(\bar{x})$. Observe que del Lema 3.2.10, dado que toda estructura argumental contiene un conjunto vacío de premisas, su árbol de soporte $\mathcal{T}$ tiene sólo argumentos evidenciales en sus hojas. Por ello, dado que el claim de un argumento evidencial es expresado como una fórmula del lenguaje $\mathcal{L}_{\mathtt{A}}$–no considera variables libres– el proceso de soporte dentro de $\mathbb{S}$ desarrollado por $\mathcal{T}$ aplica una *sustitución de variables*, por ejemplo mapeando $\bar{x}$ a $\bar{a}$, tal que $\mathfrak{cl}(\mathbb{S}) = \varphi(\bar{a})$.

**Teorema 3.2.12** *Dado un* $\mathtt{GenAF}$ $\langle \mathbf{A}, \mathbf{R}_{\mathtt{A}} \rangle \in \mathbb{G}$*, una fórmula* $\varphi(\bar{x}) \in \mathcal{L}_{\mathtt{cl}}$*, y una estructura* $\mathbb{S} \subseteq \mathbf{A}$ *para* $\varphi(\bar{x})$*; si* $\mathbb{S}$ *es argumental y* $\mathfrak{leaf}(\lambda) \subseteq \mathbf{E}$ *entonces* $\mathfrak{cl}(\mathbb{S}) = \varphi(\bar{a})$ *y* $\varphi(\bar{a}), v \models \varphi(\bar{x})$*, donde* $v$ *mapea cada* $x_i \in \{\bar{x}\}$ *a* $a_i \in \{\bar{a}\}$*.*

Dado que las estructuras son conjuntos de argumentos interrelacionados, es posible identificar subconjuntos dentro de una misma estructura cuyos argumentos conforman un árbol de soporte más pequeño. Identificaremos a tales subconjuntos como *subestructuras* (estructuras dentro de estructuras), siguiendo la misma intuición vista para la noción de subargumento en la Sección 2.1.3.

**Definición 3.2.13 (Subestructura)** *Dadas dos estructuras* $\mathbb{S} \subseteq \mathbb{A}\mathtt{rgs}$ *para* $\alpha \in \mathcal{L}_{\mathtt{cl}}$*, y* $\mathbb{S}' \subseteq \mathbb{A}\mathtt{rgs}$ *para* $\beta' \in \mathcal{L}_{\mathtt{cl}}$*;* $\mathbb{S}'$ *es una* ***subestructura*** *de* $\mathbb{S}$ *(notado como* $\mathbb{S}' \trianglelefteq \mathbb{S}$*) sssi* $\mathbb{S}' \subseteq \mathbb{S}$*. Además,* $\mathbb{S}'$ *es una* ***subestructura propia*** *de* $\mathbb{S}$ *(notado como* $\mathbb{S}' \triangleleft \mathbb{S}$*) sssi* $\mathbb{S}' \subset \mathbb{S}$*.*

**Proposición 3.2.14** *Dado un* $\mathtt{GenAF}$ $\langle \mathbf{A}, \mathbf{R}_{\mathtt{A}} \rangle \in \mathbb{G}$*, una fórmula* $\alpha \in \mathcal{L}_{\mathtt{cl}}$*, y dos estructuras* $\mathbb{S} \subseteq \mathbf{A}$ *para* $\alpha$ *y* $\mathbb{S}' \subseteq \mathbf{A}$ *para* $\alpha$*, si* $\mathbb{S}' \triangleleft \mathbb{S}$ *entonces* $\mathfrak{pr}(\mathbb{S}) \neq \mathfrak{pr}(\mathbb{S}')$*.*

La formación de estructuras complejas es ilustrada en el siguiente ejemplo. Observe que un argumento podría ser usado varias veces dentro de una misma estructura si es soportado por diferentes argumentos. Esto es posible debido a las diferentes instanciaciones de variables.

**Ejemplo 13** *Asuma un* $\mathtt{GenAF}$ $\langle \mathbf{A}, \mathbf{R}_{\mathtt{A}} \rangle$ *tal que* $\{\mathcal{B}_1,\ \mathcal{B}_2,\ \mathcal{B}_3,\ \mathcal{B}_4\} \subseteq \mathbf{A}$ *donde* $\mathcal{B}_1 = \langle \{p(x)\}, (\exists y)(\neg r(x,y) \vee p(y)) \rangle$*,* $\mathcal{B}_2 = \langle \{\}, r(a,b) \rangle$*,* $\mathcal{B}_3 = \langle \{\}, p(a) \rangle$*, y* $\mathcal{B}_4 = \langle \{\}, r(b,c) \rangle$*. El conjunto* $\widehat{\mathcal{C}}_1 = \{\mathcal{B}_1, \mathcal{B}_2\}$ *es un supporter de* $\mathcal{B}_1$ *en* $p(b)$*, y el conjunto* $\widehat{\mathcal{C}}_2 = \{\mathcal{B}_1, \mathcal{B}_4\}$ *es un claimer de* $p(c)$*. Observe que como resultado de la sustitución de variables* $x$ *e* $y$ *(de*

*$\mathcal{B}_1 \in \widehat{\mathcal{C}}_1$) a $a$ y $b$ (de $\mathcal{B}_2 \in \widehat{\mathcal{C}}_1$) respectivamente, tenemos que $\mathfrak{prset}(\widehat{\mathcal{C}}_1) = \{p(a)\}$, lo cual en cambio es soportado por la coalición primitiva $\{\mathcal{B}_3\}$; sustituyendo similarmente $x$ e $y$ (de $\mathcal{B}_1 \in \widehat{\mathcal{C}}_2$) a $b$ y $c$ (de $\mathcal{B}_4 \in \widehat{\mathcal{C}}_2$), el conjunto de premisas $\mathfrak{prset}(\widehat{\mathcal{C}}_2) = \{p(b)\}$ concluye soportado por $\widehat{\mathcal{C}}_1$. Luego, aparecen las estructuras esquemáticas $\mathbb{S}_1 = \{\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3\}$ para $p(b)$, y $\mathbb{S}_2 = \{\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4\}$ para $p(c)$, tal que $\mathbb{S}_1 \lhd \mathbb{S}_2$. Observe que el árbol de soporte en $\mathfrak{Trees}_{\mathbb{S}_2}(p(c))$ tiene $\{\mathcal{B}_1, \mathcal{B}_4\}\{\mathcal{B}_1, \mathcal{B}_2\}\{\mathcal{B}_3\}$ como su única cadena de soporte. En la Figura 3.1 ilustramos las estructuras esquemáticas y sus componentes internos detallados en este ejemplo.*

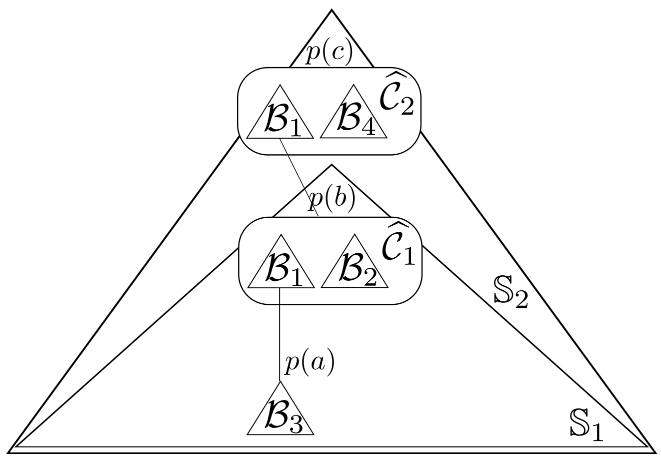



Figura 3.1: Estructuras correspondientes al Ejemplo 13.

## 3.3. El Razonador `GenAF`

En esta sección presentaremos los fundamentos para el reconocimiento de conflictos entre estructuras construidos con argumentos del `GenAF` y de las semánticas para especificar la aceptabilidad de argumentos de un marco generalizado.

### 3.3.1. Reconocimiento de Conflictos

Dos estructuras argumentales están en conflicto cuando sus claims no pueden ser asumidos en forma simultánea. Tales estructuras son referidas como *rebuttals*. Por otro lado, dos estructuras esquemáticas pueden también ser identificadas como conflictivas a través del reconocimiento de una contradicción entre sus claims. Sin embargo, dado que las estructuras esquemáticas consideran premisas libres, deberíamos establecer algún tipo de *dependencia* entre ellas para lograr asegurar que ambos claims serán eventualmente alcanzados. Para una visión intuitiva de este punto imagine un marco en el cual no se encuentra la evidencia necesaria para cerrar toda premisa del par de estructuras consideradas, pero una hipotética adición de la evidencia faltante de una de ellas sería suficiente para determinar dos diferentes estructuras argumentales conflictivas conteniendo cada una de las estructuras esquemáticas en cuestión. Por ello, la dependencia entre dos estructuras $\mathbb{S}_1$ y $\mathbb{S}_2$ será verificada cuando las premisas de $\mathbb{S}_1$ infieren las premisas de $\mathbb{S}_2$. A tavés de la dependencia logramos identificar el conjunto completo de estructuras conflictivas sin necesidad de soportar sus premisas. Esto será útil para dar con el manejo de inconsistencias e incoherencias en ontologías (ver Sección 2.4.8), luego en la Sección 3.5. Adicionalmente, podría ocurrir que el claim de $\mathbb{S}_2$ entre en conflicto con alguna de las premisas de $\mathbb{S}_1$ que no han sido consideradas en la inferencia de las premisas de $\mathbb{S}_2$ (para verificar la condición de dependencia). Esta situación describe un tipo diferente de conflicto, referido como *undercut*: el claim y alguna premisa de dos estructuras diferentes no pueden ser satisfechas en forma simultánea.

**Definición 3.3.1 (Estructuras Conflictivas)** *Dado un* $\texttt{GenAF}$ $\langle \mathbf{A}, \mathbf{R}_{\mathbf{A}} \rangle \in \mathbb{G}$*, dos estructuras* $\mathbb{S}_1 \subseteq \mathbf{A}$ *y* $\mathbb{S}_2 \subseteq \mathbf{A}$ *están en* ***conflicto*** *sssi se verifica:*

**(dependencia)** $\mathfrak{pr}(\mathbb{S}_1) \models \mathfrak{pr}(\mathbb{S}_2)$*, y uno de los siguientes:*

**(rebut)** $\{\mathfrak{cl}(\mathbb{S}_1), \mathfrak{cl}(\mathbb{S}_2)\} \models \perp$ *(entonces* $\mathbb{S}_1$ *es rebuttal de* $\mathbb{S}_2$*, y viceversa), o*

**(undercut)** $\{\mathfrak{cl}(\mathbb{S}_2), \rho\} \models \perp$*, donde* $\rho \in \mathfrak{pr}(\mathbb{S}_1)$ *(en tal caso* $\mathbb{S}_2$ *es un undercut de* $\mathbb{S}_1$*).*

**Observación 3.3.2 (Continúa de la Observación 3.2.9)** *Dada una KB* $\Sigma \subseteq \mathcal{L}^{\kappa}$ *y su asociado* $\texttt{GenAF}$ $\langle \mathbf{A}, \mathbf{R}_{\mathbf{A}} \rangle = \mathfrak{genaf}(\Sigma)$*, si* $\mathcal{L}_{\texttt{cl}} = \mathcal{L}^{\kappa}$ *entonces cualquier par de estructuras conflictivas* $\mathbb{S}_1 \subseteq \mathbf{A}$ *y* $\mathbb{S}_2 \subseteq \mathbf{A}$ *son rebuttals.*

**Proposición 3.3.3** *Dado un* $\texttt{GenAF}$ $\langle \mathbf{A}, \mathbf{R}_{\mathbf{A}} \rangle$ *y dos estructuras* $\mathbb{S}_1 \subseteq \mathbf{A}$ *y* $\mathbb{S}_2 \subseteq \mathbf{A}$*; si* $\mathbb{S}_1$ *y* $\mathbb{S}_2$ *son conflictivas entonces* $\mathbb{S}_1$ *y* $\mathbb{S}_2$ *son simultáneamente esquemáticas o argumentales.*

**Proposición 3.3.4** *Dado un* $\texttt{GenAF}$ $\langle \mathbf{A}, \mathbf{R}_{\mathbf{A}} \rangle$ *y dos estructuras* $\mathbb{S}_1 \subseteq \mathbf{A}$ *y* $\mathbb{S}_2 \subseteq \mathbf{A}$*; si* $\mathbb{S}_2$ *es un undercut de* $\mathbb{S}_1$ *entonces ambas* $\mathbb{S}_1$ *y* $\mathbb{S}_2$ *son estructuras esquemáticas.*

**Ejemplo 14** *Dado el conjunto* $\{\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4, \mathcal{B}_5, \mathcal{B}_6, \mathcal{B}_7\} \subseteq \mathbf{A}$ *donde*

$\mathcal{B}_1 = \langle\{p_1(x)\}, p_2(x)\rangle$

$\mathcal{B}_2 = \langle\{p_2(x)\}, p_3(x)\rangle$

$\mathcal{B}_3 = \langle\{p_1(x)\}, \neg p_3(x)\rangle$

$\mathcal{B}_4 = \langle\{\neg p_3(x)\}, p_1(x)\rangle$

$\mathcal{B}_5 = \langle\{p_1(x), \neg p_2(x)\}, p_3(x)\rangle$

$\mathcal{B}_6 = \langle\{p_4(x)\}, \neg p_3(x) \vee \neg p_1(x)\rangle$

$\mathcal{B}_7 = \langle\{p_5(x)\}, p_1(x)\rangle$

*Algunas de las estructuras que consideraremos son:*

$\mathbb{S}_1 = \langle\{p_1(x)\}, \{\mathcal{B}_1, \mathcal{B}_2\}, p_3(x)\rangle$

$\mathbb{S}_2 = \langle\{p_1(x)\}, \{\mathcal{B}_3\}, \neg p_3(x)\rangle$

$\mathbb{S}_3 = \langle\{p_1(x)\}, \{\mathcal{B}_1\}, p_2(x)\rangle$

$\mathbb{S}_4 = \langle\{p_1(x), \neg p_2(x)\}, \{\mathcal{B}_5\}, p_3(x)\rangle$

$\mathbb{S}_5 = \langle\{p_5(x)\}, \{\mathcal{B}_7, \mathcal{B}_1, \mathcal{B}_2\}, p_3(x)\rangle$

$\mathbb{S}_6 = \langle\{p_4(x), p_5(x)\}, \{\mathcal{B}_6, \mathcal{B}_7\}, \neg p_3(x)\rangle$

$\mathbb{S}_7 = \langle\{p_4(x), p_5(x)\}, \{\mathcal{B}_6, \mathcal{B}_7, \mathcal{B}_4\}, p_1(x)\rangle$

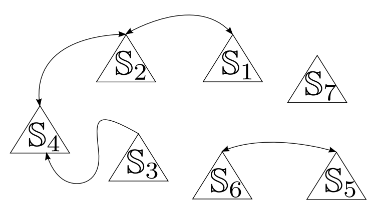



Figura 3.2: Grafo de conflictos entre estructuras – Ejemplo 14.

*Los siguientes conflictos son obtenidos:* $\mathbb{S}_1$ *es rebuttal de* $\mathbb{S}_2$*,* $\mathbb{S}_2$ *de* $\mathbb{S}_1$*,* $\mathbb{S}_2$ *de* $\mathbb{S}_4$*,* $\mathbb{S}_4$ *de* $\mathbb{S}_2$*,* $\mathbb{S}_5$ *de* $\mathbb{S}_6$*,* $\mathbb{S}_6$ *de* $\mathbb{S}_5$*, y* $\mathbb{S}_3$ *es un undercut de* $\mathbb{S}_4$*. Observe además que* $\mathbb{S}_3 \trianglelefteq \mathbb{S}_1 \trianglelefteq \mathbb{S}_5$ *y* $\mathbb{S}_6 \trianglelefteq \mathbb{S}_7$*. Luego, el grafo ilustrado en la Figura 3.2 también puede ser visto como en la Figura 3.3.*

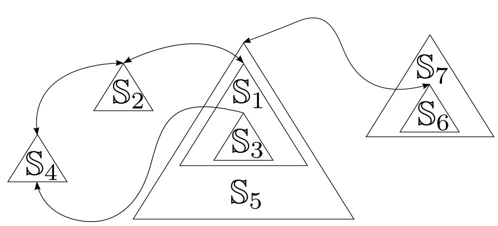



Figura 3.3: Grafo de conflictos entre estructuras c/subestructuras – Ejemplo 14.

Algunas veces, los conflictos pueden ser propagados a super-estructuras, *i.e.*, estructuras conteniendo alguna de las partes del conflicto. Esta noción es ilustrada por el siguiente ejemplo.

**Ejemplo 15 (Continúa del Ejemplo 14)** *Suponga que también tenemos* $\{\mathcal{B}_8, \mathcal{B}_9\} \subseteq \mathbf{A}$ *donde* $\mathcal{B}_8 = \langle\{\}, p_1(a)\rangle$ *y* $\mathcal{B}_9 = \langle\{\}, \neg p_2(a)\rangle$*. Nos concentraremos en las estructuras* $\mathbb{S}_2, \mathbb{S}_3, \mathbb{S}_4$*, y en las nuevas estructuras siguientes:*

$\mathbb{S}_2 = \langle\{p_1(x)\}, \{\mathcal{B}_3\}, \neg p_3(x)\rangle$, *y*
$\mathbb{S}_8 = \langle\{\}, \{\mathcal{B}_8, \mathcal{B}_3\}, \neg p_3(a)\rangle$, *donde* $\mathbb{S}_2 \trianglelefteq \mathbb{S}_8$*;*

$\mathbb{S}_3 = \langle\{p_1(x)\}, \{\mathcal{B}_1\}, p_2(x)\rangle$, *y*
$\mathbb{S}_9 = \langle\{\}, \{\mathcal{B}_8, \mathcal{B}_1\}, p_2(a)\rangle$, *donde* $\mathbb{S}_3 \trianglelefteq \mathbb{S}_9$*; and*

$\mathbb{S}_4 = \langle\{p_1(x), \neg p_2(x)\}, \{\mathcal{B}_5\}, p_3(x)\rangle$,
$\mathbb{S}_{10} = \langle\{\neg p_2(a)\}, \{\mathcal{B}_8, \mathcal{B}_5\}, p_3(a)\rangle$, *y*
$\mathbb{S}_{11} = \langle\{\}, \{\mathcal{B}_9, \mathcal{B}_8, \mathcal{B}_5\}, p_3(a)\rangle$, *donde* $\mathbb{S}_4 \trianglelefteq \mathbb{S}_{10} \trianglelefteq \mathbb{S}_{11}$.

*El grafo ilustrado en la Figura 3.4 es determinado por la identificación de los pares conflictivos.*

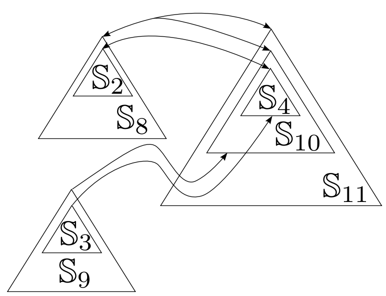



Figura 3.4: Grafo de conflictos – Ejemplo 15.

Dadas dos estructuras conflictivas, si dos subestructuras propias, una de cada una, también determinan un par conflictivo, entonces tales conflictos son referidos como *conflictos anidados*. El reconocimiento de los *conflictos base* a partir de conflictos anidados nos permite identificar fuentes más pequeñas de conflicto del `GenAF`.

**Definición 3.3.5 (Conflicto Base)** *Dado un* `GenAF` $\langle \mathbf{A}, \mathbf{R}_\mathbf{A}\rangle \in \mathbb{G}$, *dos estructuras conflictivas* $\mathbb{S}_1 \subseteq \mathbf{A}$ *y* $\mathbb{S}_2 \subseteq \mathbf{A}$ *determinan un* ***conflicto base*** *sssi para cualquier* $\mathbb{S}'_1 \trianglelefteq \mathbb{S}_1$ *y cualquier* $\mathbb{S}'_2 \trianglelefteq \mathbb{S}_2$, $\mathbb{S}'_1$ *y* $\mathbb{S}'_2$ *son conflictivas sssi* $\mathbb{S}'_1 = \mathbb{S}_1$ *y* $\mathbb{S}'_2 = \mathbb{S}_2$.

Para definitivamente decidir cual estructura prevalece en un par conflictivo de rebuttals, se asumirá la existencia de un *criterio de comparación de argumentos* $\succ \subseteq \mathbf{A} \times \mathbf{A}$ el cual es determinado a partir de un criterio de comparación entre fórmulas de la KB –que podría ser definido, por ejemplo, sobre parámetros de relevancia o *reliability* del conocimiento, *i.e.*, mediante alguna medida de relevancia asignada a fórmulas de la KB. (Por defecto, todo par de fórmulas en una KB se asumen igualmente relevantes, a menos que lo contrario sea especificado.) Luego, sobrecargaremos la definición del criterio de comparación de argumentos, extendiendo su uso a estructuras, *i.e.*, $\succ \subseteq 2^{\mathbf{A}} \times 2^{\mathbf{A}}$. Por ello, dos estructuras $\mathbb{S}_1$ y $\mathbb{S}_2$ se asumen ordenadas mediante alguna *metodología de ordenamiento de estructuras* definida sobre el criterio "$\succ$". Luego, escribimos $\mathbb{S}_1 \succ \mathbb{S}_2$ para decir que $\mathbb{S}_1$ es más relevante que $\mathbb{S}_2$. Diferentes metodologías de ordenamiento de estructuras pueden ser definidas. En esta tesis asumiremos una simple: $\mathbb{S}_1 \succ \mathbb{S}_2$ implica la existencia de algún $\mathcal{B} \in \mathbb{S}_1$ tal que para todo $\mathcal{B}' \in \mathbb{S}_2$ se verifica $\mathcal{B} \succ \mathcal{B}'$.

La relación de ataque $\mathbf{R_A}$ de un `GenAF` $\langle \mathbf{A}, \mathbf{R_A} \rangle \in \mathbb{G}$, es finalmente adjudicada en términos del criterio de comparación de argumentos "$\succ$".

**Definición 3.3.6 (Relación de Ataque)** *Dado un* `GenAF` $\langle \mathbf{A}, \mathbf{R_A} \rangle \in \mathbb{G}$*, el conjunto* $\mathbf{R_A} \subseteq 2^{\mathbf{A}} \times 2^{\mathbf{A}}$ *es la* ***relación de ataque*** *sssi para todo par de estructuras conflictivas* $\mathbb{S}_1 \subseteq \mathbf{A}$ *y* $\mathbb{S}_2 \subseteq \mathbf{A}$*, se verifica:*

- *si (*$\mathbb{S}_1$ *es rebuttal de* $\mathbb{S}_2$ *y* $\mathbb{S}_1 \succ \mathbb{S}_2$*) o (*$\mathbb{S}_1$ *es undercut de* $\mathbb{S}_2$*) entonces* $(\mathbb{S}_1, \mathbb{S}_2) \in \mathbf{R_A}$.

*La notación infija será utilizada, escribiendo* $\mathbb{S}_1 \mathbf{R_A} \mathbb{S}_2$*, para decir que* $\mathbb{S}_1$ ***ataca/derrota*** *a* $\mathbb{S}_2$*. Identificaremos la* ***relación de ataque base*** $\mathbf{R}^b_{\mathbf{A}} \subseteq \mathbf{R_A}$*, conteniendo las relaciones de ataque dentro de* $\mathbf{R_A}$ *determinadas únicamente por conflictos base.*

**Ejemplo 16 (Continúa de los ejemplos 14 y 15)** *Para los pares de rebuttals determinados por* $\mathbb{S}_1$ *y* $\mathbb{S}_2$*,* $\mathbb{S}_2$ *y* $\mathbb{S}_4$*,* $\mathbb{S}_8$ *y* $\mathbb{S}_{10}$*,* $\mathbb{S}_8$ *y* $\mathbb{S}_{11}$*, y* $\mathbb{S}_5$ *y* $\mathbb{S}_6$*; evitaremos la simetría entre los ataques que modelan en* $\mathbf{R_A}$ *basándonos en el criterio de comparación de argumentos, asumiendo* $\mathcal{B}_3 \succ \mathcal{B}_i$ *y* $\mathcal{B}_6 \succ \mathcal{B}_i$*, donde* $1 \leq i \leq 7$*:*

$$\mathbf{R_A} = \{(\mathbb{S}_2, \mathbb{S}_1),\ (\mathbb{S}_2, \mathbb{S}_4), (\mathbb{S}_3, \mathbb{S}_4), (\mathbb{S}_6, \mathbb{S}_5), (\mathbb{S}_8, \mathbb{S}_{10}), (\mathbb{S}_8, \mathbb{S}_{11}), (\mathbb{S}_9, \mathbb{S}_{10})\}$$

*Por otro lado, la relación de ataque base concluye como:*

$$\mathbf{R}^b_{\mathbf{A}} = \{(\mathbb{S}_2, \mathbb{S}_1), (\mathbb{S}_2, \mathbb{S}_4), (\mathbb{S}_3, \mathbb{S}_4), (\mathbb{S}_6, \mathbb{S}_5)\}$$

### 3.3.2. Coherencia vs. Consistencia en KBs

En esta sección adaptaremos las nociones expuestas en la Sección 2.4.8 a KBs $\mathcal{L}^{\kappa}$ en `pANF`. Recordemos que una KB $\Sigma$ infiere una sentencia $\phi$, escribiendo $\Sigma \models \phi$, si para todo modelo $\mathcal{I}$ de $\Sigma$, $\mathcal{I} \models \phi$. Luego, la KB se dice *satisfacible* o *consistente*, si admite al menos un modelo. La siguiente proposición relaciona la inferencia de sentencias a partir de KBs consistentes en `pANF`, con la construcción de estructuras argumentales.

**Proposición 3.3.7** *Dada una* `pANF` *KB consistente $\Sigma$, para cualquier $\alpha \in \mathcal{L}_{\mathtt{cl}}$, $\Sigma \models \alpha$ sssi existe una estructura argumental $\mathbb{S} \subseteq \mathbf{A}$ tal que $\mathfrak{cl}(\mathbb{S}) = \alpha$, donde $\langle \mathbf{A}, \mathbf{R}_{\mathbf{A}} \rangle = \mathfrak{genaf}(\Sigma)$.*

Dada una `pANF` KB $\Sigma \subseteq \mathcal{L}^{\kappa}$ y una fórmula $\varphi \in \Sigma$ de la forma $\rho_1 \wedge \ldots \wedge \rho_n \rightarrow \alpha$, $\varphi$ es satisfacible si existe un modelo $\mathcal{I} \in \mathcal{M}(\Sigma)$ tal que $\rho_1^{\mathcal{I}} \cap \ldots \cap \rho_n^{\mathcal{I}} \subseteq \alpha^{\mathcal{I}}$ se verifica. Sin embargo, puede suceder que $\rho_1^{\mathcal{I}} \cap \ldots \cap \rho_n^{\mathcal{I}} = \emptyset$. En ese caso la parte izquierda de la fórmula sería insatisfacible (ver Definición 2.4.1). Como fue visto en la Sección 2.4.8, esto no evita que la KB mantenga su condición de satisfabilidad/consistencia. Observe que esta condición determina un tipo de inconsistencia a nivel no-evidencial. Es decir, que si bien no tenemos una inconsistencia clásica, ella aparecería al incorporar nueva información en la KB soportando las premisas $\rho_1, \ldots, \rho_n$ (ver Proposición 2.4.4). Esta es la noción de incoherencia (ver Definición 2.4.2) que aplicaremos a KBs $\mathcal{L}^{\kappa}$.

**Definición 3.3.8 (Incoherencia)** *Dada una* `pANF` *KB $\Sigma \subseteq \mathcal{L}^{\kappa}$, $\Sigma$ es **incoherente** sssi existe una fórmula $\varphi \in \Sigma$ tal que su parte izquierda es insatisfacible, i.e., si $\varphi$ tiene la forma $\rho_1 \wedge \ldots \wedge \rho_n \rightarrow \alpha$ entonces $\rho_1^{\mathcal{I}} \cap \ldots \cap \rho_n^{\mathcal{I}} = \emptyset$, para todo modelo $\mathcal{I}$ de $\Sigma$.*

**Ejemplo 17** *Considere las estructuras $\mathbb{S}_3$ y $\mathbb{S}_4$ del Ejemplo 14, donde $\mathbb{S}_3$ es un undercut de $\mathbb{S}_4$. Aunque tal par conflictivo identifica un conjunto de reglas que no es inconsistente, i.e., $\{p_1(x) \rightarrow p_2(x), p_1(x) \wedge \neg p_2(x) \rightarrow p_3(x)\}$, si buscamos hacer inferible el claim de $\mathbb{S}_4$, necesitaríamos incorporar información a la KB con el objetivo de cerrar las premisas de $\mathbb{S}_4$. Claramente, este proceso concluiría en una nueva inconsistencia. Por ejemplo, para formar una estructura argumental para $p_3(a)$, una posible opción sería la incorporación de $p_1(a)$ y $\neg p_2(a)$ a la KB. Sin embargo, esto concluye en una contradicción con $p_1(x) \rightarrow p_2(x)$. Por lo tanto, el conflicto determinado por el par $\mathbb{S}_3$ y $\mathbb{S}_4$ identifica una fuente de información incoherente en la KB subyacente.*

La definición de la relación $\mathbf{R_A}$ permite reconocer las fuentes de información inconsistente e incoherente de la KB $\Sigma$. Con el objetivo de identificar en forma independiente los casos de inconsistencias, introduciremos la definición del conjunto *relación ataque-consistencia* $\hookrightarrow_{\mathbf{A}} \subseteq \mathbf{R_A}$.

**Definición 3.3.9 (Ataque-Consistencia)** *Dado un* `GenAF` $\langle \mathbf{A}, \mathbf{R_A} \rangle \in \mathbb{G}$ *y una relación de ataque* $\mathbb{S}_1 \mathbf{R_A} \mathbb{S}_2$*, el par* $(\mathbb{S}_1, \mathbb{S}_2)$ *determina una* ***relación ataque-consistencia*** *sssi* $\mathbb{S}_1$ *y* $\mathbb{S}_2$ *son estructuras argumentales. Identificamos el subconjunto* $\hookrightarrow_{\mathbf{A}} \subseteq \mathbf{R_A}$ *como la* ***relación ataque-consistencia****, y escribimos* $\mathbb{S}_1 \hookrightarrow_{\mathbf{A}} \mathbb{S}_2$ *para referirnos al ataque-consistencia. Identificamos además, la* ***relación ataque-consistencia base*** $\hookrightarrow^b_{\mathbf{A}} \subseteq \hookrightarrow_{\mathbf{A}}$*, conteniendo las relaciones ataque-consistencia en* $\hookrightarrow_{\mathbf{A}}$ *determinadas únicamente por conflictos base.*

Seguidamente mostramos que la unión de estructuras determinando un ataque-consistencia base identifica una mínima fuente de inconsistencia en la KB subyacente.

**Proposición 3.3.10** *Dada una KB* $\Sigma \subseteq \mathcal{L}^\kappa$ *y su asociado* `GenAF` $\langle \mathbf{A}, \mathbf{R_A} \rangle = \mathfrak{genaf}(\Sigma)$*,* $\mathbb{S}_1 \hookrightarrow^b_{\mathbf{A}} \mathbb{S}_2$ *sssi existe* $X_1 \subseteq \mathfrak{af}(\Sigma)$ *y* $X_2 \subseteq \mathfrak{af}(\Sigma)$ *tal que* $\varphi \in X_1$ *(*$\varphi \in X_2$*) sssi* $\mathfrak{arg}(\varphi) \in \mathbb{S}_1$ *ó* $\mathfrak{arg}(\varphi^-) \in \mathbb{S}_1$ *(*$\mathfrak{arg}(\varphi) \in \mathbb{S}_2$ *ó* $\mathfrak{arg}(\varphi^-) \in \mathbb{S}_2$*), donde* $\varphi^-$ *es la contrapositiva de* $\varphi$*, y* $X = X_1 \cup X_2$ *es inconsistente pero* $Y$ *es consistente, para cualquier* $Y \subset X$*.*

**Lema 3.3.11** *Dada una* `pANF` *KB* $\Sigma$*, si* $\Sigma$ *es consistente entonces* $\hookrightarrow_{\mathbf{A}} = \emptyset$*, donde* $\langle \mathbf{A}, \mathbf{R_A} \rangle = \mathfrak{genaf}(\Sigma)$*.*

Observe que la contrapositiva del Lema 3.3.11 no es válida dado que una fórmula $\varphi \in \Sigma$ puede verificar $\varphi \models \bot$ y, por lo tanto, $\mathfrak{arg}(\varphi) \notin \mathbb{A}\mathtt{rgs}$. Esto significa que $\hookrightarrow_{\mathbf{A}}$ podría aún ser vacío pero $\Sigma$ no necesariamente consistente.

**Teorema 3.3.12** *Dada una* `pANF` *KB* $\Sigma$*, si* $\Sigma$ *es coherente y consistente entonces* $\mathbf{R_A} = \emptyset$*, donde* $\langle \mathbf{A}, \mathbf{R_A} \rangle = \mathfrak{genaf}(\Sigma)$*.*

### 3.3.3. Análisis de Aceptabilidad

Como fue visto en la sección anterior, una KB $\Sigma$ inconsistente/incoherente conduce a la obtención de argumentos conflictivos dentro de $\mathfrak{genaf}(\Sigma) = \langle \mathbf{A}, \mathbf{R}_{\mathbf{A}} \rangle$ (ver Teorema 3.3.12). Por ello, cada fuente mínima de inconsistencia/incoherencia en una KB es reflejada como un ataque en su `GenAF` asociado. Dado que el objetivo del `GenAF` es razonar sobre KBs inconsistentes, surge la necesidad de formalizar un mecanismo que permita obtener los argumentos que prevalecen sobre el resto. Es decir, los argumentos que pueden ser asumidos simultáneamente (siguiendo algún criterio) en forma consistente. Por ejemplo, las estructuras sin derrotadores deberían prevalecer dado que no existe justificación alguna en contra de ellos. Este mecanismo por el cual nos apoyaremos para razonar sobre las inconsistencias será determinado por las semánticas argumentativas, dado que por medio de ellas se obtienen conjuntos de argumentos libres de conflicto, *i.e.*, *extensiones*. Estas nociones fueron brevemente introducidas en la Sección 2.1.2.

El marco presentado en este capítulo se ubica algunos pasos por delante en la construcción de los clásicos `AF`s al estilo Dung, para obtener una forma de argumentación menos abstracta, especificando cierta estructura interna para los argumentos. En particular, el uso de subargumentación (ver Sección 2.1.3) produce un leve incremento en la complejidad conceptual del marco argumentativo, y requiere la readaptación de las semánticas estándares de Dung a la especificación del `GenAF`.

**Definición 3.3.13 (Libertad de Conflicto y Defensa)** *Sea* $\langle \mathbf{A}, \mathbf{R}_{\mathbf{A}} \rangle \in \mathbb{G}$ *un* `GenAF` *y* $X \subseteq \mathbf{A}$ *un conjunto de argumentos.*

- $X$ *es* ***libre de conflicto*** *sssi no existe par de estructuras* $\mathbb{S}_1 \subseteq X$ *y* $\mathbb{S}_2 \subseteq X$ *tal que* $\mathbb{S}_1 \mathbf{R}_{\mathbf{A}} \mathbb{S}_2$.
- $X$ ***defiende*** *una estructura* $\mathbb{S} \subseteq \mathbf{A}$ *(*$\mathbb{S}$ *es* ***aceptable*** *con respecto a* $X$*) sssi para cualquier estructura* $\mathbb{S}_1 \subseteq \mathbf{A}$ *y cualquier* $\mathbb{S}' \trianglelefteq \mathbb{S}$*, si* $\mathbb{S}_1 \mathbf{R}_{\mathbf{A}} \mathbb{S}'$ *entonces existe una estructura* $\mathbb{S}_2 \subseteq X$ *tal que* $\mathbb{S}_2 \mathbf{R}_{\mathbf{A}} \mathbb{S}'_1$*, donde* $\mathbb{S}'_1 \trianglelefteq \mathbb{S}_1$*. Un conjunto de argumentos es aceptable con respecto a* $X$ *cuando cada uno de sus elementos es aceptable con respecto a* $X$*. Finalmente,* $X$ *es llamado* ***propiamente aceptable*** *cuando* $X$ *es aceptable con respecto a* $X$*.*

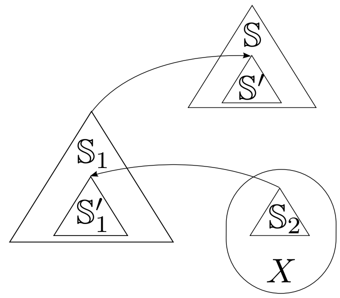



Figura 3.5: El conjunto $X$ defiende la estructura $\mathbb{S}$.

**Definición 3.3.14 (Conjuntos Admisibles)** *Sea* $\langle \mathbf{A}, \mathbf{R}_{\mathbf{A}} \rangle \in \mathbb{G}$ *un* `GenAF`, *un conjunto* $X \subseteq \mathbf{A}$ *de argumentos se dice* ***admisible*** *sssi* $X$ *es propiamente aceptable (i.e., defiende toda estructura construida a partir de* $X$*) y libre de conflicto.*

**Ejemplo 18** *En el grafo ilustrado en la Figura 3.5, tenemos que* $X$ *es admisible. Sin embargo, otros conjuntos también podrían ser admisibles, como* $X \cup \mathbb{S}$.

A continuación mostramos la readaptación a `GenAF`s de las semánticas estándares de Dung, introducidas en la Definición 2.1.4. El uso de extensiones a partir de tales semánticas, será útil para determinar subconjuntos consistentes de la KB subyacente.

**Definición 3.3.15 (Semánticas Estándares de Dung para `GenAF`s)** *Sea* $\langle \mathbf{A}, \mathbf{R}_{\mathbf{A}} \rangle \in \mathbb{G}$ *un* `GenAF`, $X \subseteq \mathbf{A}$ *un conjunto de argumentos libre de conflicto, y* $\mathcal{F} : 2^{\mathbf{A}} \longrightarrow 2^{\mathbf{A}}$ *la* ***función característica*** *tal que:*

$$\mathcal{F}(X) = \{\bigcup \mathbb{S} \mid \mathbb{S} \subseteq \mathbf{A} \text{ es una estructura y } X \text{ defiende } \mathbb{S}\}.$$

*Las siguientes son las* ***Semánticas Estándares de Dung para*** `GenAF`*s:*

- $X$ *es una* ***extensión completa*** *sssi es admisible y verifica* $X = \mathcal{F}(X)$.
- $X$ *es una* ***extensión grounded*** *sssi es la mínima (con respecto a inclusión "$\subseteq$" de* $\mathbf{A}$*) extensión completa.*
- $X$ *es una* ***extensión preferida*** *sssi es la máxima (con respecto a inclusión "$\subseteq$" de* $\mathbf{A}$*) extensión completa.*
- $X$ *es una* ***extensión estable*** *sssi es una extensión preferida y para cualquier estructura* $\mathbb{S} \subseteq (\mathbf{A} \setminus X)$ *existe una estructura* $\mathbb{S}' \subseteq X$ *tal que* $\mathbb{S}' \mathbf{R}^b_{\mathbf{A}} \mathbb{S}$.

Como se ha visto en la Sección 2.1.2, a través de los ejemplos dados para cada semántica Dung, algunos problemas como el de extensiones múltiples pueden surgir de semánticas como la *estable* y la *preferida*. Esto requerirá tomar una decisión entre las extensiones resultantes. Por otro lado, el resultado de las semánticas *grounded* es siempre una única extensión, aunque podría ser el conjunto vacío. Tales inconvenientes deberán ser tenidos en cuenta para la proposición de operaciones sobre el `GenAF` que consideren el uso de las semánticas dadas.

Asumimos una función $\mathfrak{sem} : \mathbb{G} \longrightarrow 2^{\mathbb{A}\mathrm{rgs}}$ la cual implementa alguna de las semánticas estándares de Dung's para `GenAF`s, tal que $\mathfrak{sem}(\langle \mathbf{A}, \mathbf{R_A} \rangle) = X$ es el mapeo de un `GenAF` $\langle \mathbf{A}, \mathbf{R_A} \rangle \in \mathbb{G}$ a un conjunto $X \subseteq \mathbf{A}$ de argumentos, el cual representa alguna de las extensiones determinadas por las semánticas especificadas en la Definición 3.3.15.

**Proposición 3.3.16** *Dado un* `GenAF` $\langle \mathbf{A}, \mathbf{R_A} \rangle \in \mathbb{G}$*, cualquier extensión estándar de Dung determinada por* $\mathfrak{sem}(\langle \mathbf{A}, \mathbf{R_A} \rangle)$ *es libre de conflictos.*

La siguiente proposición muestra que respetando la semántica estable obtenemos una extensión libre de conflictos maximal, del `GenAF` dado.

**Proposición 3.3.17** *Dado un* `GenAF` $\langle \mathbf{A}, \mathbf{R_A} \rangle \in \mathbb{G}$*, si* $\mathfrak{sem}(\langle \mathbf{A}, \mathbf{R_A} \rangle) = X$ *es una extensión estable entonces* $X \cup \{\mathcal{B}\}$ *no es libre de conflictos, para cualquier* $\mathcal{B} \in (\mathbf{A} \setminus X)$*.*

Las consultas hechas al marco argumentativo son resueltas por la construcción de estructuras argumentales que las soporten, para luego examinar su estado de aceptación según las semánticas argumentativas adoptadas. Una consulta satisfecha determinará una fórmula garantizada. Observe que la expresividad máxima de las consultas será restringida a la especificación del lenguaje $\mathcal{L}_{\mathtt{cl}}$.

**Definición 3.3.18 (Garantía)** *Dada una* `pANF` *KB* $\Sigma \subseteq \mathcal{L}^{\kappa}$*, una fórmula* $\alpha \in \mathcal{L}_{\mathtt{cl}}$ *se dice* ***garantizada*** *en* $\Sigma$ *sssi existe una estructura argumental* $\mathbb{S} \subseteq \mathfrak{sem}(\mathfrak{genaf}(\Sigma))$ *tal que* $\mathfrak{cl}(\mathbb{S}) = \alpha$*. Escribiremos* $\Sigma \mid\!\approx \alpha$ *para indicar que* $\alpha$ *es una garantía en* $\Sigma$*.*

**Lema 3.3.19** *Dada una* `pANF` *KB* $\Sigma \subseteq \mathcal{L}^{\kappa}$*, si* $\Sigma \mid\!\approx \alpha$ *entonces* $\alpha$ *es inferible de* $\mathfrak{genaf}(\Sigma)$*.*

Dada la $\texttt{pANF}$ KB $\Sigma \subseteq \mathcal{L}^{\kappa}$, es posible razonar sobre sus inconsistencias mediante el uso del operador " $\approx\!\!\!|$ ". Es importante notar que bajo KBs consistentes, el comportamiento del operador " $\models\!\!\!\approx$ " es análogo al de la inferencia clásica "$\models$" para KBs consistentes del mismo tipo.

**Teorema 3.3.20** *Dada una* $\texttt{pANF}$ *KB* $\Sigma \subseteq \mathcal{L}^{\kappa}$ *coherente-consistente y una fórmula* $\alpha \in \mathcal{L}_{\texttt{cl}}$*,* $\Sigma \models \alpha$ *sssi* $\Sigma \mid\!\approx \alpha$*.*

## 3.4. El $\texttt{GenAF}$ como Herramienta de Debugging

En una $\texttt{pANF}$ KB $\Sigma \subseteq \mathcal{L}^{\kappa}$ inconsistente/incoherente, las fuentes de información contradictoria son reflejadas como ataques en el $\texttt{GenAF}$ $\mathfrak{genaf}(\Sigma)$. Con el objetivo de debuggear (depurar) y reparar $\Sigma$, *i.e.*, restaurar su consistencia y su coherencia, haremos uso de las semánticas argumentativas presentadas en la Sección 3.3.3, considerando una extensión $\mathfrak{sem}(\mathfrak{genaf}(\Sigma))$ para identificar un subconjunto consistente/coherente de $\Sigma$. Primeramente, definiremos una operación de debugging basada en las semánticas argumentativas para $\texttt{GenAF}$s, y luego, analizaremos su racionalidad haciendo referencia a nociones de la teoría clásica de revisión de creencias (ver Sección 2.2).

### 3.4.1. Una Operación de Debugging basada en Semánticas Argumentativas

Intuitivamente, una operación de debugging en este capítulo será desarrollada mediante la interpretación de una $\texttt{pANF}$ $\mathcal{L}^{\kappa}$ KB a través de su $\texttt{GenAF}$ asociado, y luego, una semántica argumentativa determinará una extensión que identificará un subconjunto consistente/coherente de la KB. Consecuentemente, necesitamos definir una función capaz de traducir de regreso la extensión determinada por las semánticas en una nueva $\texttt{pANF}$ $\mathcal{L}^{\kappa}$ KB.

**Definición 3.4.1 (Función Backward)** *Dado un conjunto de argumentos* $X \subseteq \mathbb{A}\texttt{rgs}$*, la **función backward*** $\mathfrak{back} : 2^{\mathbb{A}\texttt{rgs}} \longrightarrow 2^{\mathcal{L}^{\kappa}}$ *es definida como*

$$\mathfrak{back}(X) = \{\varphi | \mathfrak{arg}(\varphi) \in X \ \textit{y si} \ \mathfrak{arg}(\varphi^{-}) \in \mathbb{A}\texttt{rgs} \ \textit{entonces} \ \mathfrak{arg}(\varphi^{-}) \in X\}.$$

Observe que la función backward requiere que un argumento en representación de una fórmula $\varphi$ y otro para la contrapositiva de $\varphi$ (si resulta ser un argumento en $\mathbb{A}\mathtt{rgs}$), estén presentes en el conjunto $X$ para incluir $\varphi$ como parte del resultado de la función. Esto es así, dado que podría suceder que un cierto argumento en representación de una fórmula $\varphi$ sea dejado fuera de la extensión, aunque no necesariamente debe suceder lo mismo con el argumento en representación de $\varphi^-$. Luego, si $\mathfrak{back}(X) \cup \{\varphi\}$ es inconsistente/incoherente entonces $\mathfrak{back}(X) \cup \{\varphi^-\}$ es inconsistente/incoherente. Luego, para el caso en el cual $\mathfrak{arg}(\varphi) \notin X$ y $\mathfrak{arg}(\varphi^-) \in X$; ambas fórmulas, $\varphi$ y $\varphi^-$, deben necesariamente ser dejadas fuera de $\mathfrak{back}(X)$ para asegurar que $\mathfrak{back}(X)$ será consistente y coherente.

**Observación 3.4.2** *Dada una* `pANF` *KB* $\Sigma \subseteq \mathcal{L}^\kappa$*, si* $\varphi \in \mathfrak{back}(\mathfrak{genaf}(\Sigma))$ *entonces o bien,* $\varphi \in \Sigma$ *ó* $\varphi^- \in \Sigma$*.*

**Observación 3.4.3** *Dada una* `pANF` *KB* $\Sigma \subseteq \mathcal{L}^\kappa$ *y su* `GenAF` $\langle \mathbf{A}, \mathbf{R_A} \rangle = \mathfrak{genaf}(\Sigma)$ *asociado, si* $\mathfrak{arg}(\varphi) \in \mathbf{A}$ *entonces* $\varphi \in \mathfrak{back}(\mathbf{A})$*.*

**Proposición 3.4.4** *Dada una* `pANF` *KB* $\Sigma \subseteq \mathcal{L}^\kappa$*, si* $\varphi \in (\Sigma \setminus \mathfrak{back}(\mathfrak{genaf}(\Sigma)))$ *entonces* $\varphi \models \bot$*.*

La siguiente proposición relaciona las nociones de consistencia/coherencia del resultado de la función $\mathfrak{back}$, a las relaciones de ataque correspondientes al `GenAF` asociado a la KB.

**Proposición 3.4.5** *Dado un* `GenAF` $\langle \mathbf{A}, \mathbf{R_A} \rangle \subseteq \mathbb{G}$*,* $\mathbf{R_A} = \emptyset$ *sssi* $\mathfrak{back}(\mathbf{A})$ *es una* `pANF` *KB consistente-coherente.*

La relación expuesta en la Proposición 3.4.5 junto con la Proposición 3.3.16 fundamenta el siguiente resultado.

**Lema 3.4.6** *Dado un* `GenAF` $\langle \mathbf{A}, \mathbf{R_A} \rangle \subseteq \mathbb{G}$*,* $\mathfrak{back}(\mathfrak{sem}(\langle \mathbf{A}, \mathbf{R_A} \rangle))$ *es una* `pANF` *KB consistente-coherente.*

A continuación definimos la *función de debugging* como un mapeo de una KB $\Sigma \subseteq \mathcal{L}^\kappa$ a un subconjunto consistente-coherente de $\Sigma$. Recuerde que la función $\mathfrak{af}$ (ver Definición 3.1.3) traduce una KB $\Sigma \subseteq \mathcal{L}^\kappa$ a una `pANF` KB $\mathfrak{af}(\Sigma)$ lógicamente equivalente.

**Definición 3.4.7 (Debugging)** *Dada una KB* $\Sigma \subseteq \mathcal{L}^{\kappa}$ *y para cualquier semántica estándar de Dung para* GenAF*s adoptada por el operador* $\mathfrak{sem}$*, la **función de debugging*** $\mathfrak{debug} : 2^{\mathcal{L}^{\kappa}} \longrightarrow 2^{\mathcal{L}^{\kappa}}$ *es definida como:*

$$\mathfrak{debug}(\Sigma) = \mathfrak{back}(\mathfrak{sem}(\mathfrak{genaf}(\Sigma))) \cap \mathfrak{af}(\Sigma).$$

El objetivo de la intersección entre el resultado de la función backward y $\mathfrak{af}(\Sigma)$, es la remoción de las fórmulas contrapositivas que no han sido incluidas en $\mathfrak{af}(\Sigma)$. Esto permite la obtención de un subconjunto de la conversión pANF de la KB $\Sigma$ original, tal como se expone a continuación.

**Observación 3.4.8** *Dada una KB* $\Sigma \subseteq \mathcal{L}^{\kappa}$*, se verifica que* $\mathfrak{debug}(\Sigma) \subseteq \mathfrak{af}(\Sigma)$*.*

**Teorema 3.4.9** *Dada una KB* $\Sigma \subseteq \mathcal{L}^{\kappa}$*,* $\mathfrak{debug}(\Sigma)$ *es una* pANF *KB consistente-coherente.*

## 3.4.2. Caracterización Axiomática basada en las Consolidaciones de Hansson

Como fuera originalmente introducido por Hansson en su tesis doctoral y luego en (Han97), una operación de *consolidación* es definida con el objetivo de restaurar la consistencia a una base de creencias. Como hemos detallado en la Sección 2.2.2, una forma de definir la operación es mediante contracción por falso, *i.e.*, mediante la remoción de fórmulas para evitar la inferencia de $\perp$ en la base. En general, cualquier modelo de contraccción podría ser adaptado para definir consolidaciones. Por ejemplo, el uso de *contracciones kernel* (ver Definición 2.2.5) para tal fin define la operación de *consolidación kernel* (ver Definición 2.2.9), donde todo subconjunto mínimo de la base infiriendo bottom (conjunto $\perp$-kernel) es “cortado” por una *función de incisión* (ver Definición 2.2.4).

En este capítulo, y en la tesis en general, nuestro interés particular sobre las operaciones Kernel de Hansson está fundamentado en las similaridades con los métodos utilizados: respecto del GenAF, un ataque base de estructuras, $\mathbb{S}_1 \hookrightarrow^{b}_{\mathbf{A}} \mathbb{S}_2$, identifica una fuente mínima de inconsistencia (al igual que los conjuntos $\perp$-kernel) (ver Proposición 3.3.10), y luego, las semánticas argumentativas dejan fuera de la extensión algunos de los componentes del par conflictivo “rompiendo” el conflicto (al igual que las funciones de incisión). (De acuerdo a la Proposición 3.3.17, es fácil ver que el conflicto es eliminado en forma minimal siempre que se siga la semántica estable.)

Asumiremos al operador de consolidación kernel introducido en la Sección 2.2.2, definido sobre KBs $\mathcal{L}^\kappa$, es decir, $! : \mathcal{L}^\kappa \longrightarrow \mathcal{L}^\kappa$. Luego, consideraremos los postulados dados en la página 30, identificando al postulado de *inclusión* como **C1**, al postulado de *consistencia* como **C2**, y al de *retención de núcleo* como **C3**. Tales postulados fueron utilizados por Hansson para caracterizar axiomáticamente las consolidaciones kernel (ver Teorema 2.2.10), y será nuestro fundamento para finalmente caracterizar axiomáticamente la operación de debugging introducida en la Definición 3.4.7, mediante el Teorema 3.4.10.

A tal fin, asumiremos una `pANF` KB y la función "$\mathfrak{sem}$" implementando la semántica estable aplicada sólo sobre la relación ataque-consistencia de acuerdo a la Definición 3.3.9, *i.e.*, sobre el conjunto $\hookrightarrow_{\mathbf{A}}$.

**Teorema de Representación 3.4.10** *Dada una* `pANF` *KB* $\Sigma \subseteq \mathcal{L}^\kappa$ *y asumiendo que* $\mathfrak{sem}$ *es obtenido a través de la semántica estable sobre* $\hookrightarrow_{\mathbf{A}}$*, la operación* $\mathfrak{debug}(\Sigma)$ *es una consolidación kernel sssi garantiza los postulados* **C1***,* **C2***, y* **C3***.*

**Ejemplo 19** *Asumamos una KB* $\Sigma = \{(p(a) \wedge q(a)), (p(x) \rightarrow \neg q(x))\}$*. Luego,* $\Sigma$ *es reformateada en* `pANF` *como* $\mathfrak{af}(\Sigma) = \{p(a), q(a), (p(x) \rightarrow \neg q(x))\}$ *y luego* $\langle \mathbf{A}, \mathbf{R}_{\mathbf{A}} \rangle = \mathfrak{genaf}(\Sigma)$ *donde* $\mathbf{A} = \{\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4\}$ *tal que:*

$$\mathcal{B}_1 = \langle \{\}, p(a) \rangle \qquad \mathcal{B}_2 = \langle \{\}, q(a) \rangle \qquad \mathcal{B}_3 = \langle \{p(x)\}, \neg q(x) \rangle \qquad \mathcal{B}_4 = \langle \{q(x)\}, \neg p(x) \rangle$$

*Asumiendo* $\beta \succ \alpha$*, para cualquier* $\alpha \in \mathcal{L}_{\mathbf{A}}$ *y cualquier* $\beta \in \mathcal{L}^\kappa$*, tenemos que* $\mathcal{B}_i \succ \mathcal{B}_j$ *para cualquier* $i \in \{3, 4\}$ *y cuaquier* $j \in \{1, 2\}$*. La siguiente lista (no-exahustiva) de estructuras será considerada:*

$$\mathbb{S}_1 = \langle \{\}, \{\mathcal{B}_1\}, p(a) \rangle \quad \trianglelefteq \quad \mathbb{S}_3 = \langle \{\}, \{\mathcal{B}_1, \mathcal{B}_3\}, \neg q(a) \rangle$$
$$\mathbb{S}_2 = \langle \{\}, \{\mathcal{B}_2\}, q(a) \rangle \quad \trianglelefteq \quad \mathbb{S}_4 = \langle \{\}, \{\mathcal{B}_2, \mathcal{B}_4\}, \neg p(a) \rangle$$

*Las relaciones de ataque (ilustradas en la Figura 3.6) son definidas como* $\mathbf{R}_{\mathbf{A}} = \hookrightarrow_{\mathbf{A}} = \{(\mathbb{S}_3, \mathbb{S}_2), (\mathbb{S}_4, \mathbb{S}_1)\}$*. Asumiendo* $X = \{\mathcal{B}_1, \mathcal{B}_3, \mathcal{B}_4\}$*, observe que* $X$ *es propiamente aceptable y libre de conflictos (recuerde que* $\mathcal{B}_4$ *conforma la estructura primitiva*

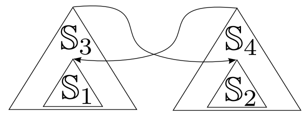



Figura 3.6: Grafo de ataques – Ejemplo 19.

$\langle\{q(x)\},\{\mathcal{B}_4\},\neg p(x)\rangle$*). Luego,* $X$ *es un conjunto admisible, y dado que* $\mathcal{F}(X) = X$*, tenemos que* $X$ *es una extensión completa. Observe que como* $\mathbb{S}_2 \subseteq (\mathbf{A} \setminus X)$ *es rebatido por* $\mathbb{S}_3 \subseteq X$ *(y ninguna otra estructura aparece dentro de* $(\mathbf{A} \setminus X)$*), la extensión completa* $X$ *es también una extensión estable.*

*Asumiendo* $\mathfrak{sem}(\langle\mathbf{A}, \mathbf{R}_{\mathbf{A}}\rangle) = X$*, tenemos* $\mathfrak{debug}(\Sigma) = \mathfrak{back}(\mathfrak{sem}(\mathfrak{genaf}(\Sigma))) \cap \mathfrak{af}(\Sigma) = \mathfrak{back}(X) \cap \mathfrak{af}(\Sigma) = \{p(a), (p(x) \rightarrow \neg q(x))\}$*.*

El resultado expuesto por el Teorema 3.4.10 puede ser extendido para axiomatizar una operación de consolidación que no sólo restaure consistencia a la base sino que también restaure su coherencia. Esto vale mediante la consideración de "$\mathfrak{sem}$" implementando la semántica estable sobre el conjunto completo de relaciones de ataque $\mathbf{R}_{\mathbf{A}}$. A tal fin, es necesario reformular los postulados **C2** y **C3**, cambiando cada requerimiento de consistencia a uno de consistencia-coherencia. El correspondiente teorema de representación puede ser mostrado en forma análoga a la versión original (Teorema 3.4.10) para consistencia.

**Observación 3.4.11 (Continúa de la Observación 3.3.2)** *Dada una KB* $\Sigma \subseteq \mathcal{L}^{\kappa}$ *y su* `GenAF` $\langle\mathbf{A}, \mathbf{R}_{\mathbf{A}}\rangle = \mathfrak{genaf}(\Sigma)$ *asociado, si* $\mathcal{L}_{\texttt{cl}} = \mathcal{L}^{\kappa}$ *entonces la incoherencia no puede ser reparada dado que no existirán estructuras esquemáticas en* $\mathbf{A}$*.*

## 3.5. Debugging de Ontologías a través del `GenAF`

El uso de técnicas de argumentación para razonar sobre ontologías inconsistentes está ganando cierto interés desde los últimos años. Algunos trabajos han sido propuestos, y aunque las investigaciones en el área se encuentran en sus primeras etapas, su usabilidad parece ser promisoria si consideramos adicionalmente los últimos avances en el área de semánticas argumentativas. Por esta razón, el uso de generalizaciones de Dung como la presentada en esta tesis como `GenAF` –cuya maquinaria de razonamiento puede ser manejada por las semánticas clásicas para argumentación abstracta– parece ser una dirección apropiada en esta línea de investigación.

Como fuera mencionado anteriormente, el razonamiento sobre ontologías inconsistentes es de extrema importancia en áreas de aplicación como la medicina y el derecho. En esta sección proponemos una reificación de $\mathcal{L}^{\kappa}$ a la DL $\mathcal{ALC}$ (ver Sección 2.4.5). Esto es posible, dado que las descripciones conceptuales en $\mathcal{ALC}$ han sido mostradas como fórmulas de tipo $\mathcal{L}^2$ en la literatura (Bor96; Baa99). Además, en (LSW01) una extensión de la DL $\mathcal{ALC}$ es presentada y probada equivalente a $\mathcal{L}^2$. Como resultado obtenemos la definición de un $\mathcal{ALC}$-`GenAF` con el objetivo de razonar sobre ontologías $\mathcal{ALC}$ pontencialmente inconsistentes/incoherentes. Sin embargo, en ocasiones es necesaria la identificación de un núcleo consistente de la ontología que contiene nuestro conocimiento. Esto es tratado en el área de debugging de ontologías, que usualmente también incluye la reparación de conceptos insatisfacibles, correspondiente al área de reparación de ontologías.

Primeramente, haremos referencia a los trabajos de Bienvenu sobre $\mathcal{ALC}$ *prime implicates* como formal normal recomendada para conceptos. Una ontología `pANF` $\mathcal{ALC}$ puede ser obtenida mediante la transformación de axiomas $\mathcal{ALC}$ como $C \sqsubseteq D$ en subsunciones entre los prime implicates de $C$ y $D$. De esta forma, podríamos no sólo obtener el $\mathcal{ALC}$-`GenAF`, sino también beneficiarnos de algunas de las conocidas ventajas del uso de prime implicates como forma de *compilación de conocimiento* o *knowledge compilation*. Luego, daremos algunos ejemplos para estudiar el comportamiento de la operación de debugging presentada en la Sección 3.4 sobre ontologías $\mathcal{ALC}$ normalizadas.

### 3.5.1. Especificando el $\mathcal{ALC}$-`GenAF`

Uno de los primeros inconvenientes que enfrentamos en pos de la especificación del $\mathcal{ALC}$-`GenAF`, es la obtención de una ontología `pANF` $\mathcal{ALC}$. Si en primera instancia necesitamos traducir la ontología original a una nuevo formato, entonces cabe la interrogante: ¿por qué valdría la pena? Recientemente, los trabajos de Bienvenu sobre *formas normales* $\mathcal{ALC}$ *prime implicate* (Bie08; Bie09), han mostrado varias propiedades interesantes de las que gozan tales formas normales. Por ejemplo, los conceptos $\mathcal{ALC}$ en la forma normal prime implicate se comportan de forma mucho más eficiente en términos computacionales, que conceptos $\mathcal{ALC}$ arbitrarios: puede ser testeado en tiempo constante si un concepto en forma normal prime implicate es satisfacible o tautológico y en tiempo cuadrático si dos conceptos en forma normal prime implicate son equivalentes o si uno subsume al otro.

La compilación del conocimiento admite un costo inicial de preprocesamiento para computar la forma normal. La transformación de un concepto $\mathcal{ALC}$ a su forma normal prime implicate puede ser alcanzada por un algoritmo propuesto en (Bie08). Dicho algoritmo resulta en una exploción de, a lo sumo, un orden doble-exponencial sobre el tamaño del concepto. Sin embargo, un algoritmo de subsunción estructural fue presentado para decidir entre conceptos en forma normal prime implicate. Tal algoritmo fue mostrado correcto, completo, y logra sus decisiones en tiempo lineal sobre el tamaño de la entrada, *i.e.*, para una consulta $C \sqsubseteq D$, termina en tiempo lineal $|C||D|$, lo cual implica un tiempo cuadrático $|C| + |D|$. Esto implica una muy considerable mejora con respecto al costo de subsunción para axiomas $\mathcal{ALC}$ no-normalizados (correspondiendo a la clase EXPTIME-completo por uso de algoritmos tableaux, o PSPACE-completo en el caso de unfoldable $\mathcal{ALC}$, lo cual también requiere un costo de preprocesamiento inicial para la transformación), en tanto que supone una compensación del costo inicial de las transformaciones con el ahorro computacional que se lograría sobre las consultas posteriores.

Mostraremos que las gramáticas utilizadas para prime implicates (introducidas en (Bie08)) son en general coincidentes para la forma normal pre-argumental, pANF, definida en este capítulo, y por lo tanto, el $\mathcal{ALC}$-GenAF puede ser beneficiado por las facilidades mencionadas, propias del uso de prime implicates.

$$
\begin{aligned}
L &::= \top|\bot|A|\neg A|\forall R.D|\exists R.D \\
Cl &::= L|Cl \sqcup Cl \\
Cb &::= L|Cb \sqcap Cb \\
D &::= \top|\bot|A|\neg A|D \sqcap D|D \sqcup D|\forall R.D|\exists R.D
\end{aligned}
$$

Dada la gramática para conceptos prime implicate $\mathcal{ALC}$, podemos especificar una ontología pANF $\mathcal{ALC}$.

**Definición 3.5.1 (Ontología pANF $\mathcal{ALC}$)** *Dada una ontología $\mathcal{ALC}$ $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$, decimos que $\mathcal{O}$ está en* pANF *sssi $\mathcal{T} \subseteq \mathcal{L}_{\mathcal{T}}$ y $\mathcal{A} \subseteq \mathcal{L}_{\mathtt{A}}$, donde $\mathcal{L}_{\mathcal{T}} ::= Cb \sqsubseteq Cl$ identifica axiomas, y $\mathcal{L}_{\mathtt{A}} ::= Cl(a)|R(a,b)$ identifica aserciones (ground), con $a$ y $b$ como individuos constantes. Identificamos como $\mathcal{ALC}^{\mathbb{A}_\Sigma} \subseteq \mathcal{L}_{\mathcal{T}} \times \mathcal{L}_{\mathtt{A}}$ al dominio de ontologías* pANF $\mathcal{ALC}$.

A continuación especificamos los lenguajes $\mathcal{L}_{\mathtt{cl}}$ para claims y $\mathcal{L}_{\mathtt{pr}}$ para premisas utilizados para la construcción de argumentos del $\mathcal{ALC}$-`GenAF`.

$$\mathcal{L}_{\mathtt{pr}} ::= L$$
$$\mathcal{L}_{\mathtt{cl}} ::= Cl|Cl(a)|R(a,b)$$

**Proposición 3.5.2** *Los lenguajes* $\mathcal{L}_{\mathtt{pr}} ::= L$ *y* $\mathcal{L}_{\mathtt{cl}} ::= Cl|Cl(a)|R(a,b)$ *determinan un lenguaje argumental legal* $\mathbb{A}\mathtt{rgs}$.

Observe que de acuerdo a la Definición 3.1.6, la Definición 3.5.1, los lenguajes para premisas y claims especificados anteriormente, y la gramática para conceptos prime implicate $\mathcal{ALC}$; todo axioma/aserción en una ontología `pANF` $\mathcal{O}$ es un argumento del `GenAF`, por sí mismo, a menos que la condición de consistencia en la Definición 3.1.6 sea violada. Es importante notar que los algoritmos presentados en (Bie08) para la construcción de conceptos prime implicate $\mathcal{ALC}$ determinan un nuevo concepto $\mathcal{ALC}$ el cual conforma la gramática dada anteriormente, sin contener conceptos ni roles innecesarios, ni subconceptos redundantes o irrelevantes. Esta es otra de las interesantes ventajas de la forma normal prime implicate $\mathcal{ALC}$, haciéndola mas fácil para la lectura y comprensión por parte del humano; y aún más interesante para esta tesis, es que los axiomas/aserciones `pANF` $\mathcal{ALC}$, especificadas por la Definición 3.5.1, conformarán trivialmente las condiciones dadas en la Definición 3.1.6 para construir argumentos a partir de fórmulas `pANF`.

Es importante mencionar que el uso detallado de prime implicates (y/u otros tipos de formas normales) exceden el foco de esta tesis, y se propone como alternativa para obtener beneficios de las buenas propiedades de la compilación de conocimiento, enfrentando la necesidad de transformar ontologías $\mathcal{ALC}$ a `pANF`. Más aún, la versatilidad del `GenAF` permite adaptar su lenguaje argumental a diferentes formas normales para $\mathcal{ALC}$, como la forma disyuntiva (tCCMV06), y la forma normal *linkless* (FO07; FGO09). Implementaciones futuras del $\mathcal{ALC}$-`GenAF` considerarán los algoritmos para la transformación de conceptos de acuerdo a la forma normal adoptada, en combinación con las gramáticas apropiadas para obtener ontologías `pANF` $\mathcal{ALC}$. En el resto del capítulo, sólo haremos referencia a ontologías `pANF` satisfaciendo las gramáticas dadas anteriormente –sin considerar los prime implicates de conceptos utilizados en los ejemplos– aunque el lector debe mantener en mente que otras gramáticas pueden adaptarse para la construcción de un diferente, pero aún apropiado, $\mathcal{ALC}$-`GenAF`.

Algunas particularidades surgen para DLs $\mathcal{ALC}$, por ejemplo las equivalencias de conceptos como $C_1 \equiv C_2$, son transformadas en dos axiomas `pANF` $C_1 \sqsubseteq C_2$ y $C_2 \sqsubseteq C_1$. Además, dos axiomas `pANF` como $\bot \sqsubseteq C$ y $C \sqsubseteq \bot$, sólo determinarán dos argumentos de sus axiomas `pANF` contrapositivos $\neg C \sqsubseteq \top$ y $\top \sqsubseteq \neg C$, respectivamente, dado que los argumentos no pueden aceptar $\bot$ en ninguno de sus componentes (ver consistencia en la Definición 3.1.6). Finalmente, cualquier aserción $Cl(a)$ (resp., $R(a,b)$) determina una evidencia $\langle\{\}, Cl(a)\rangle$ (resp., $\langle\{\}, R(a,b)\rangle$).

Sobrecargaremos la función $\mathfrak{af} : 2^{\mathcal{ALC}} \longrightarrow 2^{\mathcal{ALC}^{\mathbb{A}\Sigma}}$, y la identificaremos como *función panf-DL*, para traducir cualquier ontología $\mathcal{ALC}$ $\mathcal{O}$ a una ontología equivalente $\mathcal{ALC}^{\mathbb{A}\Sigma}$ $\mathfrak{af}(\mathcal{O})$. Una propiedad deseable de una ontología $\mathcal{ALC}^{\mathbb{A}\Sigma}$ es que cada sentencia en ella genera un simple argumento (de hecho, un par de argumentos, si se da el caso de que su contrapositiva se encuentra dentro del lenguaje) en su `GenAF` asociado. Esto es válido excepto para inclusiones de concepto insatisfacibles como $A \sqsubseteq \neg A$, las cuales son filtradas por la condición de consistencia en la Definición 3.1.6 –sin determinar argumento alguno en el `GenAF` asociado.

Dada una ontología $\mathcal{ALC}$ $\mathcal{O}$, el `GenAF` $\langle \mathbf{A}, \mathbf{R}_{\mathbf{A}} \rangle$ asociado es obtenido mediante la función de teorías $\mathfrak{genaf}(\mathcal{O})$ (ver Definición 3.1.13). Observe que las terminologías cíclicas como $A \equiv B$ no serán parte de ninguna estructura dado que esto violaría la condición de aciclicidad en la Definición 3.2.6. Un caso análogo en FOL fue ilustrado en el Ejemplo 12. La misma situación ocurre con axiomas como $A \sqsubseteq A$.

**Ejemplo 20** *Dada la ontología* $\mathcal{O} = \{(A_1 \sqcap A_2) \sqcup (\forall R_1.A_3 \sqcap \exists R_2.\forall R_3.\neg A_4) \sqsubseteq (A_1 \sqcup A_2) \sqcap A_5\}$*; la ontología* `pANF` $\mathfrak{af}(\mathcal{O})$ *resultante, determinada de acuerdo a la Definición 3.5.1, contendrá los siguientes cuatro axiomas* `pANF`*:*

- $A_1 \sqcap A_2 \sqsubseteq A_1 \sqcup A_2$,
- $\forall R_1.A_3 \sqcap \exists R_2.\forall R_3.\neg A_4 \sqsubseteq A_1 \sqcup A_2$,
- $A_1 \sqcap A_2 \sqsubseteq A_5$, *y*
- $\forall R_1.A_3 \sqcap \exists R_2.\forall R_3.\neg A_4 \sqsubseteq A_5$.

*Finalmente, de acuerdo a la Definición 3.1.13, los siguientes ocho argumentos son contenidos en el* `GenAF` *asociado a* $\mathcal{O}$*:*

- $\langle\{A_1(x), A_2(x)\}, (A_1 \sqcup A_2)(x)\rangle$,

- $\langle\{(\forall R_1.A_3)(x),\ (\exists R_2.\forall R_3.\neg A_4)(x)\}, (A_1 \sqcup A_2)(x)\rangle$,
- $\langle\{A_1(x), A_2(x)\},\ A_5(x)\rangle$, *y*
- $\langle\{(\forall R_1.A_3)(x),\ (\exists R_2.\forall R_3.\neg A_4)(x)\},\ A_5(x)\rangle$,

*y con respecto a las fórmulas contrapositivas,*

- $\langle\{\neg A_1(x), \neg A_2(x)\}, (\neg A_1 \sqcup \neg A_2)(x)\rangle$,
- $\langle\{\neg A_1(x), \neg A_2(x)\}, (\exists R_1.\neg A_3 \sqcup \forall R_2.\exists R_3.A_4)(x)\rangle$,
- $\langle\{\neg A_5(x)\}, (\neg A_1 \sqcup \neg A_2)(x)\rangle$, *y*
- $\langle\{\neg A_5(x)\}, (\exists R_1.\neg A_3 \sqcup \forall R_2.\exists R_3.A_4)(x)\rangle$.

### 3.5.2. Ejemplo de Razonamiento y Debugging de Ontologías a través del $\mathcal{ALC}$-`GenAF`

El siguiente ejemplo ilustra un caso de debugging para una ontología $\mathcal{ALC}$ con una particularidad: algunos argumentos son reintroducidos dentro de una misma cadena de soporte debido a las diferentes formas de instanciación de sus variables.

**Ejemplo 21** *Dada la ontología* $\mathcal{ALC}$ $\mathcal{O} = \{R(a,b),\ R(b,c),\ R(c,d),\ A(a),\ \neg A(c),\ \neg A(d),\ A \sqsubseteq \forall R.A\}$, *el* $\mathcal{ALC}$-`GenAF` $\langle \mathbf{A}, \mathbf{R}_{\mathbf{A}}\rangle$ *es determinado como* $\mathbf{A} = \{\mathcal{B}_1,\ \mathcal{B}_2,\ \mathcal{B}_3,\ \mathcal{B}_4,\ \mathcal{B}_5,\ \mathcal{B}_6,\ \mathcal{B}_7,\ \mathcal{B}'_7\}$, *donde* $\mathcal{B}_1 = \langle\{\}, R(a,b)\rangle$, $\mathcal{B}_2 = \langle\{\}, R(b,c)\rangle$, $\mathcal{B}_3 = \langle\{\}, R(c,d)\rangle$, $\mathcal{B}_4 = \langle\{\}, A(a)\rangle$, $\mathcal{B}_5 = \langle\{\}, \neg A(c)\rangle$, $\mathcal{B}_6 = \langle\{\}, \neg A(d)\rangle$, $\mathcal{B}_7 = \langle\{A(x)\}, (\forall R.A)(x)\rangle$, *y* $\mathcal{B}'_7 = \langle\{(\exists R.\neg A)(x)\}, \neg A(x)\rangle$.

*El conjunto* $\{\mathcal{B}_7, \mathcal{B}_1\}$ *es una coalición-soporte de* $\mathcal{B}_7$ *en* $A(b)$, $\{\mathcal{B}_7, \mathcal{B}_2\}$ *es una coalición-soporte de* $\mathcal{B}_7$ *en* $A(c)$, *y* $\{\mathcal{B}_7, \mathcal{B}_3\}$ *es una coalición-soporte de* $\mathcal{B}_7$ *en* $A(d)$. *Observe que el conjunto* $\{\mathcal{B}_7, \mathcal{B}_1\}$ *puede conformar dos estructuras esquemáticas diferentes: una para* $A(b)$ *si su cadena de soporte es constituída por una simple coalición* $\{\mathcal{B}_7, \mathcal{B}_1\}$, *y otra para* $(\forall R.A)(b)$ *si su cadena de soporte es* $\{\mathcal{B}_7\}\{\mathcal{B}_7, \mathcal{B}_1\}$. *Por simplicidad, sólo identificaremos las siguientes estructuras argumentales:*

$\mathbb{S}_1 = \langle\{\}, \{\mathcal{B}_4, \mathcal{B}_7, \mathcal{B}_1\}, A(b)\rangle$,

$\mathbb{S}_2 = \langle\{\}, \{\mathcal{B}_4, \mathcal{B}_7, \mathcal{B}_1, \mathcal{B}_2\}, A(c)\rangle$,

$\mathbb{S}_3 = \langle\{\}, \{\mathcal{B}_4, \mathcal{B}_7, \mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3\}, A(d)\rangle$,

$\mathbb{S}_4 = \langle\{\}, \{\mathcal{B}_3, \mathcal{B}_6, \mathcal{B}'_7\}, \neg A(c)\rangle$,

$\mathbb{S}_5 = \langle \{\}, \{\mathcal{B}_3, \mathcal{B}_6, \mathcal{B}'_7, \mathcal{B}_2\}, \neg A(b) \rangle$, *y*
$\mathbb{S}_6 = \langle \{\}, \{\mathcal{B}_3, \mathcal{B}_6, \mathcal{B}'_7, \mathcal{B}_2, \mathcal{B}_1\}, \neg A(a) \rangle$.

*Sus cadenas de soporte correspondientes son:* $\{\mathcal{B}_7, \mathcal{B}_1\}$ $\{\mathcal{B}_4\}$ *para* $\mathbb{S}_1$, $\{\mathcal{B}_7, \mathcal{B}_2\}$ $\{\mathcal{B}_7, \mathcal{B}_1\}$ $\{\mathcal{B}_4\}$ *para* $\mathbb{S}_2$, $\{\mathcal{B}_7, \mathcal{B}_3\}$ $\{\mathcal{B}_7, \mathcal{B}_2\}$ $\{\mathcal{B}_7, \mathcal{B}_1\}$ $\{\mathcal{B}_4\}$ *para* $\mathbb{S}_3$; $\{\mathcal{B}'_7\}$ $\{\mathcal{B}_3, \mathcal{B}_6\}$ *para* $\mathbb{S}_4$, $\{\mathcal{B}'_7\}$ $\{\mathcal{B}'_7, \mathcal{B}_2\}$ $\{\mathcal{B}_3, \mathcal{B}_6\}$ *para* $\mathbb{S}_5$, $\{\mathcal{B}'_7\}$ $\{\mathcal{B}'_7, \mathcal{B}_1\}$ $\{\mathcal{B}'_7, \mathcal{B}_2\}$ $\{\mathcal{B}_3, \mathcal{B}_6\}$ *para* $\mathbb{S}_6$.

*Considere también las estructuras primitivas* $\langle \{\}, \{\mathcal{B}_4\}, A(a) \rangle$, $\langle \{\}, \{\mathcal{B}_5\}, \neg A(c) \rangle$, *y* $\langle \{\}, \{\mathcal{B}_6\}, \neg A(d) \rangle$. *Asumiendo* $A \sqsubseteq \forall R.A$ *como la sentencia de* $\mathcal{O}$ *con mayor relevancia, naturalmente* $\mathcal{B}_7 \succ \mathcal{B}_i$ *y* $\mathcal{B}'_7 \succ \mathcal{B}_i$, *donde* $i \in \{1, \ldots, 6\}$. *Luego, surgen las siguientes relaciones de ataque:* $\mathbb{S}_1 \mathbf{R_A} \mathbb{S}_5$, $\mathbb{S}_5 \mathbf{R_A} \mathbb{S}_1$, $\mathbb{S}_2 \mathbf{R_A} \mathbb{S}_4$, $\mathbb{S}_4 \mathbf{R_A} \mathbb{S}_2$, $\mathbb{S}_2 \mathbf{R_A} \{\mathcal{B}_5\}$, $\mathbb{S}_3 \mathbf{R_A} \{\mathcal{B}_6\}$, *y* $\mathbb{S}_6 \mathbf{R_A} \{\mathcal{B}_4\}$ *(ver Figura 3.7).*

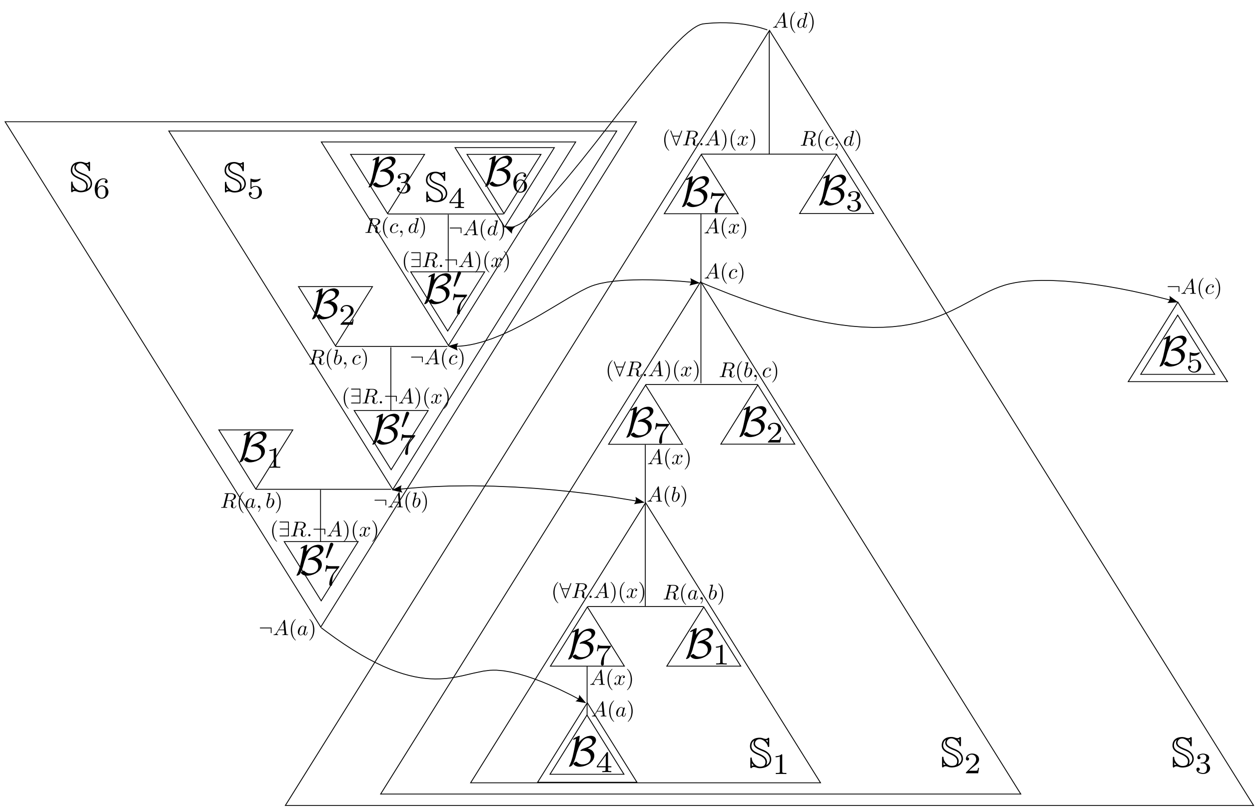



Figura 3.7: Grafo de ataques – Ejemplo 21. Ocurrencias múltiples de un mismo argumento dentro de una estructura hacen referencia a sus diferentes instancias determinadas por distintas formas de sustitución de variables dentro de la cadena de soporte.

*Adoptando la semántica estable, dos extensiones surjen:* $X_1 = \{\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4, \mathcal{B}_7, \mathcal{B}'_7\}$ *y* $X_2 = \{\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_5, \mathcal{B}_6, \mathcal{B}_7, \mathcal{B}'_7\}$. *Luego, mediante cualquiera de las dos extensiones* *obtenidas, $X_1$ o $X_2$, obtendremos la operación* $\mathfrak{debug}(\mathcal{O})$ *que construirá una ontología* $\mathcal{O}^R$ *debuggeada de acuerdo a la extensión considerada:*

- *Considerando $X_1$, se obtiene $\mathcal{O}^R = \{A \sqsubseteq \forall R.A,\ R(a,b),\ R(b,c),\ R(c,d),\ A(a)\}$, y*
- *Considerando $X_2$, $\mathcal{O}^R = \{A \sqsubseteq \forall R.A,\ R(a,b),\ R(b,c),\ R(c,d),\ \neg A(c),\ \neg A(d)\}$.*

## 3.6. Trabajo Relacionado y Futuro

El `GenAF` definido en este capítulo completa la serie (MRF08a; MRFS09; MRF10a) de versiones anteriores, logrando la especificación completa en esta tesis. Sin embargo, varias líneas de investigación quedan pendientes para extender el marco presentado en busca de especificaciones teóricas y algoritmos para otras semánticas argumentativas, construcción de argumentos, técnicas de compilación de conocimiento, otras DLs para el razonamiento sobre ontologías inconistentes, dinámica de conocimiento y argumentos, entre otros. A continuación analizamos parte de la literatura que creemos se relaciona con nuestro trabajo. La organizaremos dividiéndola en diferentes puntos de vista: argumentación, ontologías, y revisión de creencias.

### 3.6.1. Argumentación

La argumentación de primer orden fue estudiada anteriormente en (BH05). Sin embargo, ninguna relación directa fue provista entre argumentos y sentencias de la base subyacente. La principal diferencia con el `GenAF` es que ellos consideran un argumento como un par $\langle \Phi, \alpha \rangle$, donde $\Phi$ es un subconjunto de la KB infiriendo $\alpha$, lo cual es similar a nuestra noción de estructura argumental. Consecuentemente, su marco parece más simple que el `GenAF` dado que ellos construyen estructuras argumentales en forma directa. En contraste, el `GenAF` goza de una versatilidad mayor, siendo posible setear un marco más cercano al propuesto en (BH05), mediante la especificación $\mathcal{L}_{\mathtt{cl}} = \mathcal{L}^{\kappa}$. De esta forma, los argumentos serán siempre primitivos y, por lo tanto, conformarán siempre estructuras argumentales únicamente (ver Observación 3.2.9), como el caso del marco presentado por Besnard y Hunter. Más aún, el `GenAF` maneja un nivel mayor de granularidad, y por ello, es apropiado para el manejo de la dinámica de KBs, como es el caso del operador de debugging presentado aquí. En este sentido, nuestra propuesta es más similar a (Vre97), donde los argumentos son interpretados como implicaciones lógicas "→" en lógica clásica.

Sin embargo, las nociones de deducción y conflicto no son analizadas mediante interpretaciones tarskianas como ocurre en los `GenAF`s.

Desde un punto de vista netamente argumentativo, el `GenAF` posee algunas características similares a los marcos argumentativos bipolares o *Bipolar Argumentation Frameworks* (BAF) (ACLS04; CLS05; ACLSL08). Con bipolaridad se hace referencia a dos tipos de interacción entre argumentos: soporte y conflicto. Recientemente, un marco argumentativo de coaliciones o *Coalition Argumentation Framework* (CAF) (CLS10) fue presentado donde, similar a lo hecho en este capítulo, las coaliciones de argumentos son construidas para proveer soporte y ataque. Semánticas de aceptabilidad son construidas preservando algunas propiedades de los `AF`s de Dung. Los CAFs intentan agrupar tantos argumentos como sea posible en una misma coalición, la cual no podrá ser separada luego por propósitos de defensa. Esta es una de las principales diferencias con el `GenAF`, en el cual los argumentos dentro de coaliciones pueden participar de otras coaliciones con objetivos de ataque y soporte. Otra diferencia con el `GenAF` presentado aquí es que BAFs y CAFs están definidos sobre un nivel de abstracción casi completo, *i.e.*, ninguna suposición acerca de la naturaleza de los argumentos es hecha y, por ello, es que no se provee lenguaje argumental alguno. Esta característica, al igual que el trabajo seminal de Dung sobre `AF`s, permite concentrar el foco mayormente sobre el estudio de las semánticas argumentativas. Sin embargo, el problema de razonamiento sobre KBs inconsistentes no es tratado. Una interesante línea de investigación que esperamos desarrollar es la de establecer relaciones específicas con este este tipo de trabajos y enriquecer así la propuesta del `GenAF` con propiedades y semánticas mostradas para BAFs y CAFs.

Un marco argumentativo colectivo o *collective argumentation framework* (Boc03; NP06) es un marco abstracto construido por un conjunto de argumentos y una relación de ataque entre conjuntos de argumentos. Allí, un conjunto de argumentos puede atacar otros argumentos, pero contrariamente con nuestro marco, esta relación no es reducible a ataques internos. Esto es más similar a las suposiciones hechas por los CAFs, donde la noción de coalición es considerada como un todo y sus miembros no pueden ser utilizados en una relación de ataque separada. Aunque ambas propuestas son similares (ambas definen semánticas sobre subconjuntos de argumentos al igual que lo hecho en el `GenAF`), el marco de Nielsen y Parsons (NP06) permite que conjuntos de argumentos ataquen argumentos simples, y utiliza semánticas similares a las de Dung, mientras que en el marco de Bochman (Boc03), las semánticas argumentativas son basadas en nuevas definiciones específicas para conjuntos de argumentos estables y admisibles. Las diferencias con el `GenAF` son similares a las de los CAFs, basadas en el nivel de abstracción y la atomicidad de las coaliciones. Por otro lado, las similaridades con el enfoque de Bochman se basan en el objetivo de la construcción de los argumentos de un tipo de KB de expresividad restringida, llamada programas lógicos disyuntivos.

Otro enfoque similar es el presentado por García y Simari (GS04), bajo el nombre de programación en lógica rebatible o *Defeasible Logic Programming* (DeLP): una maquinaria de argumentación dedicada al razonamiento sobre un estilo especial de KBs Horn, llamadas *programas lógicos rebatibles*. En forma similar a la construcción de estructuras argumentales en el `GenAF`, un argumento en DeLP es contruído de un programa lógico rebatible como una pieza de información auto-conclusiva respecto de su claim, *i.e.*, no cuentan con premisas para ser soportadas. Las semánticas adoptadas son *semánticas basadas en consulta*, lo que quiere decir que a partir de una consulta, se construye primeramente un argumento que la soporte, y luego un árbol de dialéctica enraizado en el argumento soporte de la consulta (ver Sección 2.1.2). Luego, el árbol es analizado mediante un criterio de aceptabilidad. Estos árboles son un sub-grafo de argumentos en representación de la relación completa de ataques *à la* Dung, y constituyen un estilo diferente de semánticas, mejor orientadas a aplicaciones reales dada su construcción dinámica de derrotadores a partir de una consulta. Trabajo futuro sobre semánticas para `GenAF`s también incluirá la aplicación de árboles de dialéctica como los utlizados en DeLP. Semánticas similares basadas en árboles de argumentos son también propuestas por Besnard y Hunter en varios de sus trabajos como su argumentación de primer orden (BH05).

Las técnicas de compilación de KBs dedicadas a la obtención de un tipo de argumentación más eficiente, fue estudiada con anterioridad por Besnard y Hunter en (BH06). Allí, los autores se basan sobre su argumentación de primer orden (BH05) para aplicar técnicas de compilación para construir argumentos de una dada KB, de forma más eficiente. Como en sus trabajos anteriores, las semánticas son manejadas mediante el uso de árboles de argumentos. Trabajo a futuro también incluye la investigación de sus técnicas para compilación de KBs y algoritmos para construcción de argumentos para eventualmente buscar implementar un `GenAF` aplicado.

### 3.6.2. Ontologías

La argumentación constituye una alternativa interesante para el razonamiento sobre ontologías inconsistentes, área que como se ha dicho, es de extrema importancia en entornos de aplicación como la medicina y el derecho. Aunque algunos investigadores son escépticos de su aplicabilidad debido mayormente a la complejidad computacional, creemos que deben realizaese esfuerzos en investigar esta fusión basándonos en los siguientes argumentos:

- *El razonamiento sobre DLs es mayormente de una alta complejidad computacional:* los algoritmos de razonamiento se comportan bien en práctica, pero para los peores casos la complejidad se mantiene alta. Nuevas DLs inexpresivas aparecen para mejorar la *performance* del razonamiento. La compilación de conocimientos provee otra alternativa para mejorar la eficiencia del razonamiento admitiendo un costo inicial de preprocesamiento. Por ejemplo, unfoldable $\mathcal{ALC}$ constituye un caso de compilación de KBs para ontologías, diferente a los referidos en la Sección 3.5.1. Pero más importante para el área de aplicación que buscamos en esta tesis, el debugging y la reparación de ontologías inconsistentes es, al menos, computacionalmente tan costoso como el razonamiento sobre el mismo tipo de DL bajo consistencia, o incluso peor. Las teorías para el razonamiento sobre ontologías inconsistentes no pueden abstraerse de tales resultados, ni de la necesidad de buscar métodos de optimización para el razonamiento, para dar con niveles altos de complejidad computacional.
- *La argumentación está en constante crecimiento:* algunos algoritmos como (KT99; CDM01; BG02; GS04) fueron desarrollados. Aunque la construcción de argumentos es mayormente costosa, se han propuesto optimizaciones de algoritmos para implementar semánticas argumentativas como árboles de dialéctica, como por ejemplo los llamados *Bonsai de Dialéctica* propuestos en (Rot10). Además, el estudio de nuevas y más eficientes semánticas de argumentación también concentran mucho del esfuerzo en el área. La argumentación está siendo considerada para ser aplicada a la world-wide-web (RZR07), con diferentes objetivos, como el de proveer una alternativa de gobierno con mayor participación de sus ciudadanos. También se están estudiando los llamados *argument bloggers* (WGR09) donde los argumentos son cosntruídos de información de la web, y a partir de foros específicos. Sobre todo lo mencionado, se esperan nuevas aplicaciones argumentativas de relativa eficiencia.

La argumentación fue aplicada al razonamiento sobre ontologías inconsistentes –de forma similar a la que se mostrará en el Capítulo 4– en trabajos como (MWF10; ZZL09b). Sin embargo, los marcos provistos no gozan de una alta versatilidad de aplicación como es el caso de los `GenAF`s propuestos aquí. En el siguiente capítulo, presentaremos la metodología introducida originalmente en (MWF10) para la construcción de derrotadores de un dado argumento $\mathcal{ALC}$. Esta metodología está basada en la ampliamente conocida propuesta de Schlobach (SC03; Sch05) para debugging de ontologías basadas en unfoldable $\mathcal{ALC}$. Mostraremos que tal metodología corresponde a la clase de complejidad PSPACE, al igual que el método de debugging de Schlobach.

Aplicando la noción de relevancia a conceptos DL, en (HvHtT05) se provee un marco (no-argumentativo) para el razonamiento sobre ontologías inconsistentes. Un conjunto de propiedades es propuesta para controlar el buen comportamiento de cualquier razonador DL no-estándar que maneje inconsistencias. Similarmente, el manejo de inconsistencias en ontologías "cambiantes" fue tratado en (HvHH$^+$05), donde algunos enfoques existentes sobre el razonamiento en ontologías inconsistentes fue reformulado en términos de un conjunto de definiciones elementales. Esto permite comparar enfoques bastante diferentes para un mismo problema (*i.e.*, para trabajar con ontologías inconsistentes) con el objetivo de determinar si constituyen soluciones de aplicación factibles bajo diferentes configuraciones: en tiempo de desarrollo, o uso, de la ontología, y requiriendo diferentes piezas de información (la causa de la inconsistencia, o la historia de los cambios de la ontología). Nuestro trabajo a futuro también incluye el estudio de tales propiedades y resultados para caracterizar el `GenAF` de acuerdo a su punto de vista.

Nuestro interés está enfocado mayormente en futuras implementaciones del modelo presentado. Esto podría realizarse, como un módulo a ser incorporado al razonador DL, o como un razonador-DL netamente argumentativo. Para la segunda opción, un razonador-DL basado en argumentación proveería una alternativa a los ya conocidos RACER y FaCT, entre otros (ver Sección 2.4.7). Un enfoque netamente argumentativo para el razonamiento sobre ontologías solucionaría "a la corrida" la decisión de qué mantener o descartar de diferentes fuentes de información sin necesidad de aplicar cambios a las ontologías originales. Es decir, una ontología podría mantener las inconsistencias dejando su resolución al proceso de razonamiento argumentativo, solucionando en forma dinámica el manejo de las inconsistencias. Esta constituye una interesante propuesta para incorporar a la *Web Semántica* la cualidad característica de los razonadores argumentativos: mantener las inconsistencias mientras se razona sobre ellas. Más sobre este punto será referido en el Capítulo 4.

### 3.6.3. Revisión de Creencias

(CdSCLS10) propone una teoría de revisión sobre sistemas abstractos de argumentación al estilo Dung. Considerando como el conjunto de extensiones es modificado bajo un proceso de revisión, proponen una tipología de los diferentes efectos de la revisión: revisión decisiva y revision expansiva. Una fuerte restricción es expuesta: el nuevo argumento incorporado debe poseer a lo sumo una interacción (via ataque) con otro argumento del sistema. Esta restricción simplifica enormemente el problema de revisión de `AF`s, dado que las interacciones múltiples con el sistema original son comunes, e incluso su manejo puede resultar bastante dificultoso, como veremos en el Capítulo 4. En general, ellos estudian como la incorporación de un argumento afectaría el conjunto de extensiones (sin desarrollar ninguna alteración además de la adición del nuevo argumento), mediante el análisis del grafo de argumentos. La revisión de sistemas de decisión basados en argumentación fue aplicada en (AV09) a través de una generalización de la técnica de revisión presentada en (CdSCLS10), la cual evalúa el estado de garantía del nuevo argumento incorporado.

Otro enfoque similar de estudio de dinámica en argumentación fue presentado en (BKvdT09). Allí, la *abstracción* del marco, *i.e.*, la remoción de un conjunto de argumentos o ataques, es considerada, y una serie de principios es propuesta para establecer bajo qué condiciones las semánticas permanecen inalteradas. Este enfoque evita la recomputación de semánticas. Los mismos autores discuten en (BCPTvdT08) la relación existente entre argumentación y revisión de creencias. Ellos consideran argumentación como un tipo de persuasión en la que creer, y esa persuasión es la que debe ser relacionada a revisión de creencias. Más recientemente, (BdCPTvdT08) presentó la interrelación entre argumentación y revisión de creencias en sistemas multi-agentes. Cuando un agente utiliza un argumento para persuadir a otro, debe considerar no sólo la proposición soportada por el argumento, sino el impacto completo del argumento sobre el conocimiento del destinatario.

Todos estos enfoques proponen propiedades generales para analizar las consecuencias de la incorporación de nuevos argumentos a un marco respecto de una extensión. Un profundo análisis sobre este tema sería interesante para el `GenAF`, con el propósito de proponer algoritmos para implementar las semánticas adoptadas y la operación de debugging. Estos resultados podrían ser usados para complementar el criterio de comparación para finalmente decidir cuales argumentos deberían ser excluídos para la construcción de una extensión apropiada.

Los enfoques de la teoría de cambio argumentativa o *Argument Theory Change* (ATC) (RMF$^+$08b; MRF$^+$08b; MRF$^+$10b), proponen la revisión de marcos argumentativos mediante la incorporación de un nuevo argumento y la remoción de algunos otros para asegurar la garantía del nuevo argumento. ATC fue implementado sobre DeLP para la revisión de programas lógicos rebatibles en (MRF$^+$08b). Aunque ese trabajo está basado en árboles de dialéctica para el manejo de las semánticas, y no se realiza restauración de consistencia; el objetivo por el cual el marco es cambiado es similar al que motiva al `GenAF` presentada aquí: alterar una KB subyacente.

Dado que los argumentos de un `GenAF` representan un simple axioma de la ontología subyacente, el enfoque propuesto puede ser útil además en otras subáreas del cambio ontológico como evolución ontológica e integración, lo cual está usualmente basado en revisión de creencias clásico y el modelo AGM (AGM85) (aunque la argumentación es en general no considerada). Parte de este trabajo es investigado en (MLB05; MLBP06; QHHP08; RW09).

## 3.7. Conclusiones

Un novedoso marco argumentativo fue presentado como generalización del clásico `AF` de Dung (Dun95), llamado por su acrónimo `GenAF`. La propuesta del `GenAF` mantiene la abstracción de la lógica utilizada para representar el conocimiento dentro de argumentos mientras que se provee una especificación del lenguaje argumental $\mathbb{A}$`rgs` para dotar de cierta estructura interna a los argumentos. Esto da lugar a una forma generalizada de argumentación abstracta capaz de adaptarse a cualquier lenguaje de representación que respete algún fragmento de primer orden. Por ello, un `GenAF` constituye un marco de argumentación de reificación directa a diferentes lenguajes de representación para el razonamiento sobre un amplio espectro de KBs inconsistentes. Una de las principales características del `GenAF` es que las fórmulas de la KB subyacente son interpretadas como argumentos. Para tal fin, la KB se requiere conformar una *forma normal pre-argumental* para obtener un `GenAF` asociado. Sobre tal necesidad, propusimos el uso de técnicas de compilación de conocimiento, tales como las expuestas en (DM02), las cuales buscan la forma de transformar una KB en pos de la resolución de consultas en tiempo polinomial. En tales teorías, el alto costo de preprocesamiento enfrentado es justificado por la idea de compensación que se adquiere luego mediante ahorros computacionales durante el proceso de razonamiento. Su aplicación al `GenAF` lo beneficia con tales particularidades.

La aplicación del `GenAF` para el manejo de lenguajes de representación específicos para el razonamiento sobre una KB, requiere la concreción del lenguaje argumental $\mathbb{A}\texttt{rgs}$, lo cual involucra la especificación de sub-lenguajes concretos para claims y premisas. Esto puede traer algunos problemas, por ejemplo, el lenguaje para claims puede considerar fórmulas conjuntivas y/o disyuntivas. Para el primer caso, la opción más sencilla es determinar un claim diferente para cada término conjuntivo. Para el caso de fórmulas disyuntivas para claims, el problema es un poco más complicado. Para ello, se introduce la noción de *coalición*, la cual es una estructura capaz de agrupar varios argumentos con la intención de inferir una nueva fórmula mediante la conjunción de los claims disyuntivos de los argumentos incluidos en la coalición. Una coalición tiene también el objetivo de proveer *soporte* a una premisa de un argumento mediante el alcance de su satisfabilidad dentro de la misma coalición. Más aún, dado que una premisa $\rho$ y un claim $\alpha$, pueden considerar variables libres, una coalición con claim ground $\beta$ soportando $\rho$ determina un nuevo tipo de estructura con claim ground mediante la substitución de las variables libres en $\alpha$ con los correspondientes individuos constantes considerados en $\beta$.

Sobre todo lo mencionado, es claro que los argumentos de un `GenAF` son indivisibles al igual que en los marcos argumentativos clásicos. Sin embargo, los argumentos en un `GenAF` juegan un rol menor: son agregados dentro de *estructuras* en pos de la obtención de un claim específico. La idea detrás del agregado de argumentos dentro de una estructura es similar al uso de sub-argumentos (ver Sección 2.1.3). Un `GenAF` considera dos tipos de estructuras diferentes: *estructuras argumentales*, las cuales son auto-concluyentes, al igual que los argumentos de marcos como (BH05) entre otros; y *estructuras esquemáticas*, las cuales no pueden alcanzar sus claims a menos que sus premisas sean soportadas. Observe que la instanciación de variables dentro de una estructura esquemática puede ocurrir como consecuencia del soporte de sus premisas, dando así lugar a estructuras argumentales.

Las derrotas en un `GenAF` son definidas a partir de pares conflictivos de estructuras determinando dos clases diferentes de relaciones de ataque: entre estructuras argumentales y entre estructuras esquemáticas. Ambos tipos de ataque determinan dos tipos de fuentes de información conflictiva de la KB subyacente: inconsistencias e incoherencias. Intuitivamente, una fuente de información incoherente de una KB es una pieza de conocimiento que, aunque no infiere inconsistencia, admite sólo inerpretaciones vacías para cualquier modelo de la KB. Por ello, en una KB consistente, las incoherencias pueden aparecer descubriendo un tipo de "pre-inconsistencia". El manejo de incoherencias es un problema de alta relevancia en el área de reparación de ontologías para conceptos insatisfacibles (ver Sección 2.4.8).

Las semánticas estándares de Dung fueron adaptadas para `GenAF`s, y utilizadas para definir un operador de debugging para la restauración de consistencia/coherencia de KBs. Sin embargo, la abstracción del `GenAF` permite aceptar fácilmente nuevas semánticas argumentativas definidas sobre los marcos argumentativos clásicos de Dung. Luego, el operador de debugging fue axiomáticamente caracterizado mediante el uso de los postulados clásicos para la operación de consolidación de bases de creencias, propuesta por Hansson (Han97) en el área de revisión de creencias. Seguidamente, el correspondiente teorema de representación fue mostrado.

Como fuera mencionado anteriormente, la versatilidad del `GenAF` permite manejar diferentes lógicas para argumentos. En particular, la reificación del lenguaje (abstracto) argumental $\mathbb{A}\mathtt{rgs}$ dentro de las DLs $\mathcal{ALC}$ determina una metodología interesante para el manejo de debugging y la reparación de ontologías. Esto es, a través del ataque entre pares de estructuras esquemáticas y argumentales, logramos dar con la reparación de conceptos insatisfacibles (reparación ontológica), y con la restauración de consistencia (debugging ontológico), respectivamente.

Es importante observar que la maquiaria argumentativa aquí propuesta es semánticamente determinada –por efecto de interpretaciones similar a lo hecho en teorías basadas en el análisis de modelos. Esto provee como ventaja la propuesta futura de implementaciones basadas en otros motores de razonamiento existentes. Por ejemplo, un $\mathcal{ALC}$-`GenAF` podría ser manejado por técnicas tableaux, las cuales son usualmente utilizadas para la implementación de razonadores ontológicos (ver Sección 2.4.7). Por otro lado, una posible implementación de un $\mathcal{ALC}$-`GenAF` podría ser desarrollada en forma directa sobre la ontología $\mathcal{ALC}$ normalizada, emulando la maquinaria argumentativa. Esto se debe a que los axiomas $\mathcal{ALC}$ son interpretados como simples argumentos en el `GenAF`. Con tal consideración, la metodología argumentativa propuesta puede ser utilizada simplemente como una herramienta teórica para el razonamiento –y debugging– sobre ontologías inconsistentes.

# Capítulo 4

# Un Modelo de Cambio para Ontologías Inconsistentes

En este capítulo presentaremos un nuevo modelo de cambio para el manejo de dinámica de argumentos. El mencionado modelo, denominado *dialéctico-global*, es un nuevo enfoque dentro de la *teoría de cambio argumentativa* o *argument theory change* (ATC). El nombre dialéctico-global se debe a su aplicación sobre árboles de dialéctica, y al alcance de la función de incisión definida en forma global al árbol, en contraste con otros modelos ATC cuya aplicabilidad se realiza a nivel de línea de argumentación.

El modelo se aplica sobre un marco argumentativo construido a partir de una KB subyacente. Tal KB se asume dentro de la lógica clásica proposicional con el objetivo de simplificar las características del marco argumentativo para poner el foco de atención en las cuestiones propias de la dinámica de argumentos. Sin embargo, este marco mantiene las características básicas del `GenAF` presentado en el Capítulo 3, por lo cual, sostenemos que su aplicación a `GenAF`s podría ser especificada en forma directa.

Seguidamente, se considerarán algunos de los postulados básicos correspondientes a la teoría clásica de revisión de creencias (ver Sección 2.2) para ser luego adaptados a la argumentación aplicada a árboles de dialéctica (ver Sección 2.1.2). El modelo dialéctico-global será finalmente axiomatizado mediante su relación con el nuevo conjunto de postulados mencionados, y a través del correspondiente teorema de representación.

Los resultados serán extendidos a *description logics*, para manejar el razonamiento y evolución ontológicos con tolerancia a inconsistencias. Para ello, presentaremos el marco argumentativo específico para DLs. Discutiremos la definición de contra-argumentos analizando previamente la dificultad que implica el uso de la negación para axiomas DLs. Finalmente, la construcción de derrotadores para un argumento es propuesta mediante una modificación del ampliamente conocido *algoritmo pinpoint*, originalmente presentado en (SC03).

## 4.1. El Marco Argumentativo Proposicional

Como hemos mencionado, el marco argumentativo sobre el cual definiremos el modelo dialéctico-global, es una versión simplificada del `GenAF`, en el cual, los argumentos serán equivalentes a las estructuras argumentales en el `GenAF`. Esto implica que no trabajaremos la noción de premisa para definir argumentos en este marco simplificado. Por ello, en forma intuitiva diremos que un *argumento* puede ser visto como un *conjunto de piezas interrelacionadas de conocimiento proveyendo soporte a un claim.* Nos basaremos en una estructura $\langle$*body*, *claim*$\rangle$ para representar argumentos, donde el cuerpo o *body* es un conjunto consistente y minimal de creencias infiriendo el claim. El símbolo "$\models$" será utilizado para identificar las implicaciones semánticas utilizadas para obtener inferencias a partir del conocimiento contenido en un lenguaje $\mathcal{L}$. En este capítulo, $\mathcal{L}$ será asumido como la lógica clásica proposicional.

**Definición 4.1.1 (Argumento)** *Un* ***argumento*** $\mathcal{B}$ *es una estructura* $\langle\Delta, \beta\rangle$*, donde* $\Delta \subseteq \mathcal{L}$ *es el* ***cuerpo*** *o* ***body****,* $\beta \in \mathcal{L}$ *el* ***claim****, y tal que se verifica* (1) $\Delta \models \beta$, (2) $\Delta \not\models \bot$*, y* (3) $\nexists X \subset \Delta : X \models \beta$*. Diremos que* $\mathcal{B}$ ***soporta*** *a* $\beta$*.*

Como hemos mencionado, el objetivo es manejar la dinámica del conocimiento de KBs inconsistentes a través de la dinámica de argumentos contenidos en el marco que administra la información de la KB en cuestión. Por lo tanto, a fin de dar soporte a consultas presentadas a la KB, la idea es construir un argumento cuyo claim soporte la consulta dada. En consecuencia, la información contenida en el cuerpo de tal argumento será interpretada como una posible justificación, obtenida de la KB subyacente, para la consulta dada. Por lo tanto, asumiremos que el conocimiento contenido en el cuerpo de los argumentos correponderá a una KB proposicional subyacente $\Sigma \subseteq \mathcal{L}$. Para mayor detalle sobre argumentación basada en lógica proposicional, el lector es referido a (BH01; BH08).

Observe que los claims conforman a $\mathcal{L}$ dado que ellos no estarán necesariamente contenidos en $\Sigma$, sino implicados por $\Sigma$. El *dominio de argumentos de* $\Sigma$ es identificado a través del conjunto $\mathbb{A}_\Sigma$. Dada una consulta $\alpha \in \mathcal{L}$, un argumento $\mathcal{B} \in \mathbb{A}_\Sigma$ es un *soporte de consulta* o *query supporter* si $\mathcal{B}$ soporta a $\alpha$. En ese caso, $\mathcal{B}$ también puede ser identificado como un $\alpha$-*soporte*. Por ejemplo, asumiendo $\{p, p \rightarrow q\} \subseteq \Sigma$, los argumentos *primitivos* $\langle\{p\}, p\rangle$ y $\langle\{p \rightarrow q\}, p \rightarrow q\rangle$, y otros argumentos más comlejos como $\langle\{p, p \rightarrow q\}, q\rangle$ y $\langle\{p \rightarrow q\}, \neg p \vee q\rangle$, aparecen (entre otros) dentro de $\mathbb{A}_\Sigma$. Análogamente a las funciones definidas para el `GenAF`, mediante las funciones $\mathfrak{bd} : \mathbb{A}_\Sigma \longrightarrow 2^{\mathcal{L}}$ y $\mathfrak{cl} : \mathbb{A}_\Sigma \longrightarrow \mathcal{L}$, identificamos el cuerpo $\mathfrak{bd}(\mathcal{B})$ y el claim $\mathfrak{cl}(\mathcal{B})$ de un argumento $\mathcal{B} \in \mathbb{A}_\Sigma$. Las nociones de subargumentación se mantendrán en forma usual, donde $\mathcal{B} \in \mathbb{A}_\Sigma$ es *subargumento* de $\mathcal{C} \in \mathbb{A}_\Sigma$ (y conversamente, $\mathcal{C}$ es *superargumento* de $\mathcal{B}$) si $\mathfrak{bd}(\mathcal{B}) \subseteq \mathfrak{bd}(\mathcal{C})$ se verifica; luego, $\mathcal{B}$ es un *subargumento propio* de $\mathcal{C}$ si $\mathfrak{bd}(\mathcal{B}) \subset \mathfrak{bd}(\mathcal{C})$ se verifica. Observe que $\langle\{p \rightarrow q\}, p \rightarrow q\rangle$ es un subargumento de $\langle\{p, p \rightarrow q\}, q\rangle$.

Similarmente a las fórmulas $\mathcal{L}$ lógicamente equivalentes, los argumentos de $\mathbb{A}_\Sigma$ también podrán ser asociados mediante una relación de equivalencia.

**Definición 4.1.2 (Argumentos Equivalentes)** *Dada una KB $\Sigma \subseteq \mathcal{L}$, dos argumentos $\mathcal{B}_1 \in \mathbb{A}_\Sigma$ y $\mathcal{B}_2 \in \mathbb{A}_\Sigma$ son **equivalentes**, escrito $\mathcal{B}_1 \equiv \mathcal{B}_2$, cuando (1) $\mathfrak{cl}(\mathcal{B}_1)$ es lógicamente equivalente a $\mathfrak{cl}(\mathcal{B}_2)$, y (2) $\mathfrak{bd}(\mathcal{B}_1)$ es lógicamente equivalente a $\mathfrak{bd}(\mathcal{B}_2)$.*

Un *contra-argumento* es un argumento cuyo claim brinda una justificación para descreer de otro argumento. Los contra-argumentos exponen fuentes de inconsistencia de una KB, determinadas por sus cuerpos.

**Definición 4.1.3 (Contra-argumento)** *Dada una KB $\Sigma \subseteq \mathcal{L}$, dos argumentos $\mathcal{B}_1 \in \mathbb{A}_\Sigma$ y $\mathcal{B}_2 \in \mathbb{A}_\Sigma$ están **en conflicto** sssi $\{\mathfrak{cl}(\mathcal{B}_2)\} \cup \mathfrak{bd}(\mathcal{B}_1) \models \bot$. Diremos que $\mathcal{B}_2$ es un **contra-argumento** de $\mathcal{B}_1$, y que $\mathcal{B}_1$ y $\mathcal{B}_2$ determinan un **par conflictivo**.*

La relaciones de preferencia entre argumentos son usualmente utilizadas para determinar *derrotas* entre argumentos. Sin embargo, por simplicidad, evitaremos su uso e identificaremos los *derrotadores* en forma directa. Los *ataques* (o *derrotas*) entre argumentos de $\mathbb{A}_\Sigma$ son finalmente adjudicados a partir de pares conflictivos $\mathcal{B}_1$ y $\mathcal{B}_2$: el argumento $\mathcal{B}_2$ *derrota a* $\mathcal{B}_1$ *sssi* $\mathcal{B}_2$ contra-argumenta a $\mathcal{B}_1$. El argumento $\mathcal{B}_2$ se dice *derrotador* de $\mathcal{B}_1$

(o $\mathcal{B}_1$ es derrotado por $\mathcal{B}_2$), notado como $\mathcal{B}_2 \hookrightarrow \mathcal{B}_1$. En ocaciones, pueden aparecer conjuntos infinitos de derrotadores (cuerpos comunes y claims lógicamente equivalentes) de un mismo argumento. Debido a esto, identificaremos un tipo especial de contra-argumento como representante de todos los derrotadores de un argumento, llamado *undercut canónico* (BH08).

**Definición 4.1.4 (Undercut Canónico)** (BH08) *Dada una KB $\Sigma \subseteq \mathcal{L}$, y dos argumentos $\mathcal{B}_1 \in \mathbb{A}_\Sigma$ y $\mathcal{B}_2 \in \mathbb{A}_\Sigma$, $\mathcal{B}_2$ es un* ***undercut canónico*** *de $\mathcal{B}_1$ sssi $\mathfrak{bd}(\mathcal{B}_1) = \{\beta_1, \ldots, \beta_n\}$ y $\mathfrak{cl}(\mathcal{B}_2) = \neg(\beta_1 \wedge \ldots \wedge \beta_n)$, o equivalentemente $\mathfrak{cl}(\mathcal{B}_2) = \neg(\bigwedge \mathfrak{bd}(\mathcal{B}_1))$.*

**Ejemplo 22** *Asumimos una KB $\Sigma \subseteq \mathcal{L}$ y una consulta p. $\mathcal{R} = \langle \{q, (q \rightarrow p)\}, p \rangle$ es un p-soporte y $\mathcal{B}_1 = \langle \{p_1, (p_1 \rightarrow p_2), (\neg p_2 \vee p_3), (p_3 \rightarrow \neg p)\}, \neg(q \wedge (q \rightarrow p)) \rangle$ un undercut canónico de $\mathcal{R}$ dado que $\mathfrak{cl}(\mathcal{B}_1) = \neg(\bigwedge \mathfrak{bd}(\mathcal{R}))$. La KB $\Sigma$ completa y el resto de los argumentos construidos a partir de ella que serán considerados son:*

$$\Sigma = \left\{ \begin{array}{l} q,\ p_1,\ q_1,\ q_2,\ \neg q_3,\ \neg p_4, \\ (q \rightarrow p), (p_1 \rightarrow p_2), \\ (\neg p_2 \vee p_3), (p_3 \rightarrow \neg p), \\ (q_1 \rightarrow \neg p_3), (q_2 \rightarrow \neg q), \\ (q_2 \rightarrow \neg p_2), (\neg p_2 \rightarrow p_4), \\ (\neg p_2 \vee q_3) \end{array} \right\} \qquad \begin{array}{l} \mathcal{B}_2 = \langle \{q_1, (q_1 \rightarrow \neg p_3)\}, \neg(\bigwedge \mathfrak{bd}(\mathcal{B}_1)) \rangle, \\ \mathcal{B}_3 = \langle \{q_2, (q_2 \rightarrow \neg q)\}, \neg(\bigwedge \mathfrak{bd}(\mathcal{R})) \rangle, \\ \mathcal{B}_4 = \langle \{(q_2 \rightarrow \neg p_2), (\neg p_2 \rightarrow p_4), \neg p_4\}, \\ \qquad \neg(\bigwedge \mathfrak{bd}(\mathcal{B}_3)) \rangle, \\ \mathcal{B}_5 = \langle \{(\neg p_2 \vee q_3), \neg q_3\}, \neg(\bigwedge \mathfrak{bd}(\mathcal{B}_4)) \rangle,\ y \\ \mathcal{B}_6 = \langle \{(\neg p_2 \vee q_3), \neg q_3\}, \neg(\bigwedge \mathfrak{bd}(\mathcal{B}_1)) \rangle. \end{array}$$

*Todo argumento dado (con excepción de $\mathcal{R}$) es un undercut canónico. Observe que la lista no es exhaustiva. Note también que los argumentos con un mismo cuerpo pueden derrotar diferentes argumentos dependiendo de sus claims, tal es el caso de $\mathcal{B}_5$ y $\mathcal{B}_6$. El conjunto de ataques obtenido es: $\mathcal{B}_1 \hookrightarrow \mathcal{R}$, $\mathcal{B}_2 \hookrightarrow \mathcal{B}_1$, $\mathcal{B}_3 \hookrightarrow \mathcal{R}$, $\mathcal{B}_4 \hookrightarrow \mathcal{B}_3$, $\mathcal{B}_5 \hookrightarrow \mathcal{B}_4$, y $\mathcal{B}_6 \hookrightarrow \mathcal{B}_1$.*

La metodología de razonamiento estará dada a partir de la aceptación de algún soporte de consulta obtenido de $\mathbb{A}_\Sigma$. Por otro lado, la teoría de cambio argumentativo, ATC, basa sus modelos de cambio en el análisis y modificación de árboles de dialéctica. Por tales motivos, el marco argumentativo proposicional en este capítulo estará basado en la construcción y análisis de árboles dialécticos, como semántica argumentativa. Los detalles de la semántica a utilizar se corresponderán con las nociones expuestas en la Sección 2.1.2, con algunas salvedades. Por ejemplo, las líneas de argumentación estarán construidas a partir de argumentos obtenidos de $\mathbb{A}_\Sigma$.

Por otro lado, las condiciones de aceptabilidad para construir líneas libres de falacias (líneas aceptables) se basan en dos condiciones: *no-circularidad* y *concordancia*.

(1) **no-circularidad:** la parte derrotada de un argumento no podrá aparecer dos veces en una misma línea. Formalmente, la línea $[\mathcal{B}_1, \ldots, \mathcal{B}_n]$ es *no-circular sssi* para cualquier $1 \leq i \leq n$, el subargumento más chico $\mathcal{C}$ de $\mathcal{B}_i$ tal que $\mathcal{C}$ es derrotado por $\mathcal{B}_{i+1}$, no será reintroducido en el resto de la línea. Esto es $\mathcal{C}$ no es subargumento de ningún argumento $\mathcal{B}_j$, con $i < j \leq n$.

(2) **concordancia:** el conjunto de argumentos pro (resp., con) en una línea no podrá contener pares conflictivos. Formalmente, una línea $\lambda$ es *concordante sssi* los conjuntos $\bigcup_{\mathcal{B}\in\lambda^+} \mathfrak{bd}(\mathcal{B})$ y $\bigcup_{\mathcal{B}\in\lambda^-} \mathfrak{bd}(\mathcal{B})$ de cuerpos de argumentos pro y con, respectivamente, son individualmente consistentes.

Por ejemplo, en el Ejemplo 22, $\mathcal{B}_7 = \langle\{p_1, (p_1 \rightarrow p_2), (\neg p_2 \vee p_3)\}, \neg(q_1 \wedge (q_1 \rightarrow \neg p_3))\rangle$ no puede ser derrotado por $\mathcal{B}_2$ dado que esto determinaría una línea cíclica o circular, violando la condición 1). Esto es así, dado que el cuerpo de $\mathcal{B}_7$ ya fue derrotado por $\mathcal{B}_2$ en la línea $[\mathcal{R}, \mathcal{B}_1, \mathcal{B}_2]$ (observe que $\mathfrak{bd}(\mathcal{B}_7) \subseteq \mathfrak{bd}(\mathcal{B}_1)$). Recordamos al lector que para mayores detalles sobre las condiciones de aceptabilidad, puede referirse a (GS04).

Adicionalmente, las *líneas canónicas* son líneas aceptables construidas mediante el argumento raíz y una secuencia de undercuts canónicos. Definimos al conjunto $\mathbf{L}_\Sigma$ para identificar el dominio de todas la líneas aceptables y canónicas construidas con argumentos de $\mathbb{A}_\Sigma$. Luego, una línea aceptable es *exhaustiva* si no puede ser extendida con un derrotador de su hoja sin comprometer su propia aceptabilidad. Haremos uso del subconjunto $\mathbb{L}_\Sigma \subseteq \mathbf{L}_\Sigma$ para identificar todas las líneas aceptables, canónicas, y exhaustivas.

Finalmente, dada una KB $\Sigma \subseteq \mathcal{L}$, tendremos la noción de árbol de dialéctica $\mathcal{T}(\mathcal{R})$ (ver Definición 2.1.11) construido a partir de un conjunto de líneas $X \subseteq \mathbf{L}_\Sigma$ enraizadas en un argumento común, $\mathcal{R} \in \mathbb{A}_\Sigma$. Tales árboles estarán identificados por el dominio $\mathbf{T}_\Sigma$.

**Ejemplo 23 (Continúa del Ejemplo 22)** *Dado el conjunto $X \subseteq \mathbf{L}_\Sigma$ de líneas de argumentación enraizadas en $\mathcal{R}$ determinando el árbol $\mathcal{T}(\mathcal{R}) \in \mathbf{T}_\Sigma$ (ilustrado en la Figura 4.1) tal que $X = \{\lambda_1, \lambda_2, \lambda_3\}$, donde $\lambda_1 = [\mathcal{R}, \mathcal{B}_1, \mathcal{B}_6]$, $\lambda_2 = [\mathcal{R}, \mathcal{B}_1, \mathcal{B}_2]$, y $\lambda_3 = [\mathcal{R}, \mathcal{B}_3, \mathcal{B}_4, \mathcal{B}_5]$, son tres líneas aceptables y canónicas. (Los argumentos son tomados del Ejemplo 22.)*

Debido al uso de líneas canónicas, un conjunto $X \subseteq \mathbf{L}_\Sigma$ de líneas enraizadas en un argumento en común será finito y, por ello, así también será cualquier árbol construido

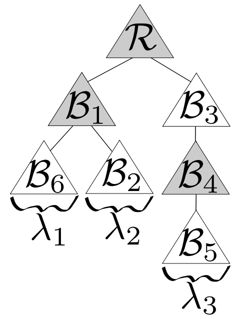



Figura 4.1: Árbol de Dialéctica – Ejemplos 23 y 22.

a partir de $X$. Sin embargo, la condición de aceptabilidad de los árboles de dialéctica también requerirá del uso de líneas exhaustivas. El conjunto bundle (ver Definición 2.1.12 para $\mathcal{R}$ será indentificado como $\mathcal{S}(\mathcal{R}) \subseteq \mathbb{L}_\Sigma$ –el cual contendrá todas las líneas de $\mathbb{L}_\Sigma$ enraizadas en $\mathcal{R}$ que sean aceptables, canónicas, y exhaustivas. En consecuencia, un árbol de dialéctica $\mathcal{T}(\mathcal{R})$ será *aceptable* si y sólo si $\mathcal{T}(\mathcal{R})$ es construido a partir de un conjunto $X \subseteq \mathbf{L}_\Sigma$ (de acuerdo a la Definición 2.1.11) tal que $X = \mathcal{S}(\mathcal{R})$. Identificaremos el dominio de todos los árboles de dialéctica aceptables (ver Definición 2.1.13) relacionados a $\Sigma$ como $\mathbb{T}_\Sigma \subseteq \mathbf{T}_\Sigma$. Nuevamente, dado que $\mathcal{R} \in \mathbb{A}_\Sigma$, el conjunto bundle $\mathcal{S}(\mathcal{R})$ es único y, por lo tanto, así lo será su correspondiente árbol de dialéctica aceptable $\mathcal{T}(\mathcal{R}) \in \mathbb{T}_\Sigma$.

**Observación 4.1.5** *$\mathcal{T}(\mathcal{R}) \in \mathbb{T}_\Sigma$ es el único árbol de dialéctica en $\mathbb{T}_\Sigma$ enraizado en $\mathcal{R}$.*

En el resto del capítulo, las funciones aplicadas sobre líneas y árboles serán generalizadas (cuando así se requiera) mediante su definición sobre $\mathbf{L}_\Sigma$ y $\mathbf{T}_\Sigma$. Esto será hecho indiferentemente de su uso general sobre los dominios $\mathbb{L}_\Sigma \subseteq \mathbf{L}_\Sigma$ y $\mathbb{T}_\Sigma \subseteq \mathbf{T}_\Sigma$.

Dada una KB $\Sigma \subseteq \mathcal{L}$, un soporte de consulta $\mathcal{R} \in \mathbb{A}_\Sigma$ es aceptado (garantizado) si el árbol de dialéctica $\mathcal{T}(\mathcal{R}) \in \mathbb{T}_\Sigma$ es un árbol garantizante (de acuerdo a lo expuesto en la Sección 2.1.2). Nuevamente, esta evaluación se realizará mediante el uso de las funciones de marcado $\mathfrak{mark} : \mathbb{A}_\Sigma \times \mathbf{L}_\Sigma \times \mathbf{T}_\Sigma \longrightarrow \mathbb{M}$, y garantizante $\mathfrak{warrant} : \mathbf{T}_\Sigma \longrightarrow \{true, false\}$, basándose en el criterio de marcado escéptico dado en la Definición 2.1.15. En el caso del Ejemplo 23, $\mathcal{R}$ es derrotado y por ello, $\mathcal{T}(\mathcal{R})$ es un árbol no-garantizante. Recuerde que en las representaciones gráficas adoptadas para los árboles de dialéctica, los argumentos pintados de gris/blanco representan marcas $D/U$, respectivamente.

## 4.2. El Enfoque Dialéctico-global a ATC

Como fuera mencionado anteriormente, ATC define un operador de revisión que revisa un `AF` por un argumento, provocando las modificaciones necesarias para garantizarlo mediante el análisis del árbol de dialéctica enraizado en él. El núcleo de la maquinaria de cambio involucra la *alteración* de algunas líneas en ese árbol de dialéctica cuando se da el caso en que no es garantizante. Por lo tanto, el objetivo de la alteración de líneas es cambiar la morfología del árbol que las contiene a fin de tornarlo garantizante.

En este capítulo, la alteración de líneas aparece mediante la remoción de argumentos pertenecientes al `AF`. Sin embargo, los argumentos no pueden ser simplemente eliminados de $\mathbb{A}_\Sigma$. En cambio, ellos desaparecen como resultado de la remoción de creencias de la KB $\Sigma$ a partir de la cual los argumentos son construidos.

Las líneas de argumentación de un árbol a ser alterado por ATC son identificadas como *líneas de ataque* (RMS09): líneas que son de alguna manera responsables del estatus no-garantizante del árbol al cual pertenecen.

**Definición 4.2.1 (Línea de Ataque)** *Una línea $\lambda \in \mathcal{T}(\mathcal{R})$, con $\mathcal{T}(\mathcal{R}) \in \mathbf{T}_\Sigma$, es una* ***línea de ataque*** *(o* ***línea atacante****) sssi para todo $\mathcal{B} \in \lambda$ se verifica:*

$$\mathfrak{mark}(\mathcal{B}, \lambda, \mathcal{T}(\mathcal{R})) = \begin{cases} D & si\ \mathcal{B} \in \lambda^+,\ ó \\ U & si\ \mathcal{B} \in \lambda^- \end{cases}$$

A modo de ejemplo, considere el árbol de dialéctica ilustrado en el Ejemplo 23. Allí, la línea $\lambda_3$ es de ataque debido a su marcado en el contexto del árbol al cual pertenece.

Cuando una línea alterada se torna de *no-ataque*, la alteración es llamada *efectiva*. Finalmente, alterando efectivamente cada línea de ataque de un árbol permite la obtención de un nuevo árbol –construido a partir de la nueva base revisada– que garantiza su argumento raíz.

**Teorema 4.2.2** *Dado un árbol de dialéctica $\mathcal{T}(\mathcal{R}) \in \mathbf{T}_\Sigma$, no existe línea de ataque $\lambda \in \mathcal{T}(\mathcal{R})$ sssi $\mathcal{T}(\mathcal{R})$ es un árbol garantizante.*

La alteración de una línea $\lambda$ involucra la remoción de algún argumento $\mathcal{B} \in \lambda$, el cual poda el sub-árbol enraizado en $\mathcal{B}$ fuera del árbol dialéctico, dejando un segmento superior $\lambda^\uparrow(\mathcal{B})$. Sólo cortando un *argumento con* de la línea, *i.e.*, $\mathcal{B} \in \lambda^-$, provoca una alteración efectiva.

**Proposición 4.2.3** *Si $\lambda \in \mathcal{T}(\mathcal{R})$ es una línea de ataque entonces para cualquier $\mathcal{B} \in \lambda^-$, el segmento superior $\lambda^{\uparrow}(\mathcal{B})$ se torna línea de no-ataque.*

En la teoría de revisión de creencias clásica (ver Sección 2.2), dado un operador de revisión "$*$", se espera que incorpore a la base $\Sigma$ un dado conocimiento $\alpha$ asegurando que será consistentemente inferido por la base revisada $\Sigma{*}\alpha$. Análogamente, en argumentación, un operador de revisión "$*^{\omega}$" tendría como objetivo la incorporación del nuevo argumento $\mathcal{R}$ (posiblemente no incluido en el conjunto $\mathbb{A}_{\Sigma}$) el cual soporta $\alpha$ (*i.e.*, $\mathfrak{cl}(\mathcal{R}) = \alpha$), asegurando que será aceptado por la semántica argumentativa adoptada a partir de la KB revisada, *i.e.*, $\mathcal{R}$ concluiría garantizado en $\Sigma{*}^{\omega}\mathcal{R}$. Esta propiedad será entendida como la *condición de éxito* del modelo de cambio argumentativo.

Por lo dicho anteriormente, formalizaremos la noción de *argumento externo* como un argumento $\mathcal{B} \in \mathbb{A}_{\Sigma'}$ tal que $\Sigma' = \Sigma \cup \mathfrak{bd}(\mathcal{B})$ y $\mathcal{B} \notin \mathbb{A}_{\Sigma}$. El *dominio de argumentos externos* es identificado a través del conjunto $\mathbb{E}_{\Sigma}$. Por otro lado, haremos referencia al conjunto $\mathbb{A}_{\mathcal{L}}$ para identificar el dominio completo de argumentos $\mathcal{L}$, es decir, argumentos incluidos en $\mathbb{A}_{\Sigma} \cup \mathbb{E}_{\Sigma}$.

En forma intuitiva, la construcción del nuevo modelo de cambio se basará en la definición de una operación de revisión de argumentos la cual incorporará a la KB $\Sigma \subseteq \mathcal{L}$ el cuerpo del argumento $\mathcal{R} \in \mathbb{A}_{\mathcal{L}}$ (incluido o bien en $\mathbb{A}_{\Sigma}$ o en $\mathbb{E}_{\Sigma}$). Luego, el árbol aceptable $\mathcal{T}(\mathcal{R}) \in \mathbb{T}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}))}$ será alterado mediante la remoción de creencias de la base $\Sigma$, con el objetivo de tornarlo garantizante. En consecuencia, una KB revisada $\Sigma{*}^{\omega}\mathcal{R}$ será obtenida, de la cual el árbol $\mathcal{T}'(\mathcal{R}) \in \mathbb{T}_{(\Sigma *^{\omega}\mathcal{R})}$ concluiría garantizante.

Una función de incisión, al estilo de las utilizadas para las operaciones Kernel de Hansson (ver Definición 2.2.4), construida en forma "global" al árbol de dialéctica define una *incisión global* (Definition 4.2.5) la cual determina un conjunto de creencias a ser removido de la KB. Tales remociones permiten descartar argumentos del árbol a fin de alterar efectivamente todas las líneas necesarias a la vez.

Sin embargo, otros argumentos que comparten algunas de las creencias a ser eliminadas también desaparecerán. Por ese motivo, diremos que una línea considerando alguno de esos argumentos será *alterada colateralmente* por efecto de una *incisión colateral*. La remoción de un argumento con de una línea de ataque resulta en un segmento superior no-atacante (ver Proposición 4.2.3). Sin embargo, la remoción de un argumento pro en una línea de argumentación podría producir un efecto no deseado: mientras las líneas de ataque mantendrán su condición de atacantes, las líneas no-atacantes podrían tornarse de ataque. Por lo tanto, el proceso de revisión debería sobreponerse a tales alteraciones involuntarias mediante la alteración de no sólo líneas de ataque, sino también de otras líneas que pueden tornarse de ataque por efecto de incisiones colaterales.

Consecuentemente, dado el árbol de dialéctica $\mathcal{T}(\mathcal{R}) \in \mathbb{T}_\Sigma$, necesitamos identificar un subconjunto de líneas del conjunto bundle $\mathcal{S}(\mathcal{R})$ a ser alteradas de forma efectiva, a fin de obtener un nuevo árbol garantizante. Sin embargo, su identificación precisa se torna no trivial por efecto de las incisiones colaterales. Es decir, este conjunto debería incluir todas las líneas de ataque de $\mathcal{T}(\mathcal{R})$. A su vez, dado que podrían surgir incisiones colaterales, también se deberían considerar el resto de las líneas del árbol colateralmente incididas a fin de evitar que tales alteraciones colaterales las tornen en nuevas líneas de ataque.

Por este motivo, la configuración de la incisión global estará basada en el análisis del denominado *árbol hipotético*, notado $\mathcal{H}(\mathcal{R}, \Psi) \in \mathbf{T}_\Sigma$: árbol de dialéctica que sólo pasa a ser aceptable luego de la remoción, de $\Sigma$, de las creencias incluidas en $\Psi$.

**Definición 4.2.4 (Árbol Hipotético)** *Dado $\mathcal{T}(\mathcal{R}) \in \mathbb{T}_\Sigma$ y un subconjunto $\Psi \subseteq \Sigma$, el* ***árbol hipotético*** *$\mathcal{H}(\mathcal{R}, \Psi) \in \mathbf{T}_\Sigma$ es el árbol (posiblemente no-aceptable) enraizado en $\mathcal{R}$ construido a partir del conjunto $X_1 \cup X_2$ de líneas:*

$$
\begin{aligned}
X_1 &= \{\lambda \in \mathcal{T}(\mathcal{R}) \mid \forall \mathcal{B} \in \lambda : \Psi \cap \mathfrak{bd}(\mathcal{B}) = \emptyset\} \\
X_2 &= \{\lambda^{\uparrow}(\mathcal{B}) \mid \lambda \in \mathcal{T}(\mathcal{R}) \textit{ tal que } (\exists \mathcal{B} \in \lambda : \Psi \cap \mathfrak{bd}(\mathcal{B}) \neq \emptyset) \\
&\qquad\qquad \textit{y } (\forall \mathcal{B}' \in \lambda^{\uparrow}(\mathcal{B}) : \Psi \cap \mathfrak{bd}(\mathcal{B}') = \emptyset)\}
\end{aligned}
$$

**Ejemplo 24 (Continúa del Ejemplo 23)** *Con la intención de alterar $\lambda_3$, $\mathcal{B}_5$ debería ser removido de $\mathcal{T}(\mathcal{R})$ mediante la remoción de $\neg q_3$ de la KB $\Sigma$. Sin embargo, $\mathcal{B}_5$ y $\mathcal{B}_6$ son removidos dado que ambos contienen $\neg q_3$. El conjunto $\{\lambda_1^{\uparrow}(\mathcal{B}_6), \lambda_2, \lambda_3^{\uparrow}(\mathcal{B}_5)\}$ resultante de la Definición 4.2.4, es utilizado para la construcción del árbol hipotético $\mathcal{H}(\mathcal{R}, \{\neg q_3\})$ ilustrado en la Figura 4.2. Observe, sin embargo, que $\lambda_1^{\uparrow}(\mathcal{B}_6) \notin \mathcal{H}(\mathcal{R}, \{\neg q_3\})$ dado que $\lambda_1^{\uparrow}(\mathcal{B}_6) = [\mathcal{R}, \mathcal{B}_1]$, el cual es parte de $\lambda_2 = [\mathcal{R}, \mathcal{B}_1, \mathcal{B}_2]$.*

El uso de árboles de dialéctica permite analizar cómo las incisiones afectarían el árbol original, reconociendo la formación de nuevas líneas de ataque que puedan surgir como resultado de la aparición de incisiones colaterales. Consecuentemente, la incisión global incluirá conocimientos de forma progresiva a fin de alterar en forma efectiva no sólo

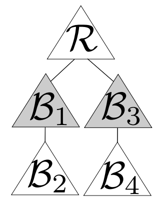


Figura 4.2: Árbol Hipotético $\mathcal{H}(\mathcal{R}, \{\neg q_3\})$ – Ejemplo 24.

líneas de ataque sino también otras líneas que podrían resultar atacantes mediante su observación en el árbol hipotético.

**Definición 4.2.5 (Incisión Global)** *Dado el árbol $\mathcal{T}(\mathcal{R}) \in \mathbb{T}_\Sigma$, una función $\sigma : \mathbb{T}_\Sigma \longrightarrow 2^{\mathcal{L}}$ es una **incisión global** sssi se verifica:*

1. *$\sigma(\mathcal{T}(\mathcal{R})) = \emptyset$ sssi $\mathcal{T}(\mathcal{R})$ no tiene líneas de ataque.*
2. *Para cualquier $\lambda \in \mathcal{T}(\mathcal{R})$, si cualquiera de las siguientes condiciones es verificada,*
   a) *$\lambda$ es de ataque, o*
   b) *existe algún $\mathcal{C} \in \lambda$ tal que*
      - *$\emptyset \neq \sigma(\mathcal{T}(\mathcal{R})) \cap \mathfrak{bd}(\mathcal{C}) = \Psi$, y*
      - *$\lambda^{\uparrow}(\mathcal{C})$ es una línea de ataque en $\mathcal{H}(\mathcal{R}, \Psi)$,*

   *entonces existe algún $\mathcal{B} \in \lambda^{-}$ tal que $\sigma(\mathcal{T}(\mathcal{R})) \cap \mathfrak{bd}(\mathcal{B}) \neq \emptyset$ y para cualquier otro $\mathcal{B}' \in \lambda^{\uparrow}(\mathcal{B})$ se verifica $\sigma(\mathcal{T}(\mathcal{R})) \cap \mathfrak{bd}(\mathcal{B}') = \emptyset$.*
3. *Para cualquier $\beta \in \sigma(\mathcal{T}(\mathcal{R}))$, existe algún $\lambda \in \mathcal{T}(\mathcal{R})$ verificando o bien 2a o 2b, y $\beta \in \mathfrak{bd}(\mathcal{B})$, para algún $\mathcal{B} \in \lambda$.*

Como fuera mencionado previamente, la incisión debe proveer un conjunto de creencias a remover de la KB a fin de tornar un árbol $\mathcal{T}(\mathcal{R})$ no-garantizante en otro garantizante. La condición 1 en la Definición 4.2.5 asegura que la inicisión será vacía sólo en el caso en que $\mathcal{T}(\mathcal{R})$ garantiza su raíz $\mathcal{R}$. Esto surge del Teorema 4.2.2, el cual relaciona la aparición de líneas de ataque a la condición de árbol no-garantizante. En consecuencia, toda línea de ataque debe ser alterada (cond. $2a$), así como también cualquier línea colateralmente incidida que pueda tornarse de ataque (cond. $2b$). Esta última condición es chequeada sobre árboles hipotéticos. De la Proposición 4.2.3, sabemos que cualquier segmento superior $\lambda^{\uparrow}(\mathcal{B})$ es no-atacante si se da el caso en que $\mathcal{B} \in \lambda^{-}$. Por lo tanto, el consecuente de la condición 2 asegura que la incisión contendrá (al menos) una creencia de $\mathfrak{bd}(\mathcal{B})$ a fin de alterar efectivamente $\lambda$ (tornándola a no-atacante). Además, la alteración de $\lambda$ a partir de tal $\mathcal{B}$ se requiere que sea la *alteración superior*, *i.e.*, ningún otro argumento $\mathcal{B}'$ ubicado sobre $\mathcal{B}$ en $\lambda$ debería ser afectado por la incisión dado que esto podría comprometer el estado de ataque de la línea. En tal caso, la incisión sobre el argumento $\mathcal{B} \in \lambda$ se denomina *incisión superior* de $\lambda$. Finalmente, la condición 3 requiere a la incisión tomar creencias sólo de los argumentos en líneas que necesariamente deben ser alteradas (mediante la verificación de $2a$ o $2b$).

**Ejemplo 25** *Dado el árbol de dialéctica $\mathcal{T}(\mathcal{R})$ ilustrado en la Figura 4.3, asumimos $\beta \in (\mathfrak{bd}(\mathcal{C}_3) \cap \mathfrak{bd}(\mathcal{C}_5))$. Supongamos que $\sigma(\mathcal{T}(\mathcal{R}))$ toma creencias de $\mathcal{C}_3$, y en particular $\beta \in \sigma(\mathcal{T}(\mathcal{R}))$. En tal caso, $\sigma(\mathcal{T}(\mathcal{R}))$ no podría considerar creencias de $\mathcal{C}_6$ dado que no sería la incisión superior en $\lambda_2$ ($\mathcal{C}_5$ está ubicado sobre $\mathcal{C}_6$ y $\mathcal{C}_5$ es colateralmente incidido, i.e., $\sigma(\mathcal{T}(\mathcal{R})) \cap \mathfrak{bd}(\mathcal{C}_5) \neq \emptyset$). Por ello, a fin de alterar $\lambda_2$, $\sigma(\mathcal{T}(\mathcal{R}))$ debería también considerar creencias de $\mathcal{C}_4$ (dado que es la única alternativa en $\lambda_2$).*

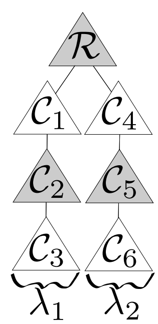



Figura 4.3: Incisión Global – Ejemplo 25.

Mediante la remoción del conjunto de creencias mapeado por la incisión global sobre $\mathcal{T}(\mathcal{R})$, obtenemos una KB determinando un árbol enraizado en $\mathcal{R}$ libre de líneas de ataque, y por lo tanto, a partir del Teorema 4.2.2, $\mathcal{R}$ concluye garantizado. A continuación definimos la revisión de argumentos en términos de la función de incisión global, a fin de revisar una KB $\mathcal{L}$ por un argumento contenido en $\mathbb{A}_{\mathcal{L}}$.

**Definición 4.2.6 (Revisión de Argumento)** *Dada una KB $\Sigma \subseteq \mathcal{L}$ y un argumento $\mathcal{R} \in \mathbb{A}_{\mathcal{L}}$, un operador $*^{\omega}$ es una **revisión de argumento** sssi*

$$\Sigma *^{\omega} \mathcal{R} = (\Sigma \cup \mathfrak{bd}(\mathcal{R})) \setminus \sigma(\mathcal{T}(\mathcal{R})), \text{ donde } \mathcal{T}(\mathcal{R}) \in \mathbb{T}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}))}.$$

**Ejemplo 26** *Como fue visto en el Ejemplo 24, si asumimos $\sigma(\mathcal{T}(\mathcal{R})) = \{\neg q_3\}$, la Definición 4.2.5 es verificada. Por lo tanto, el árbol hipotético ilustrado en el Ejemplo 24 concluye siendo el árbol alterado que resulta de $\mathbb{T}_{(\Sigma *^{\omega} \mathcal{R})}$, garantizando la raíz $\mathcal{R}$.*

*Una alternativa diferente sería asumir otra función de incisión tal que $\sigma(\mathcal{T}(\mathcal{R})) = \{q_2\}$. En este caso, el argumento $\mathcal{B}_3$ sería removido. El árbol alterado resultante (ilustrado en la Figura 4.4) de $\mathbb{T}_{(\Sigma *^{\omega} \mathcal{R})}$, concluye garantizando $\mathcal{R}$.*

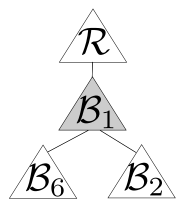



Figura 4.4: Revisión de Argumentos – Ejemplo 26.

**Lema 4.2.7** *Dada una KB $\Sigma \subseteq \mathcal{L}$, si $*^{\omega}$ es una revisión de argumento entonces $\mathcal{T}(\mathcal{R}) \in \mathbb{T}_{(\Sigma *^{\omega} \mathcal{R})}$ no posee líneas de ataque, para ningún $\mathcal{R} \in \mathbb{A}_{\mathcal{L}}$.*

## 4.3. Análisis de Racionalidad

Las revisiones y contracciones son usualmente definidas independientemente con la intención de interralacionarlas luego mediante alguna formalización de dualidad. Sobre la naturaleza de tal independencia, se mantiene una discusión filosófica. Algunos investigadores sostienen que en realidad no existe contracción alguna cuya definición pueda ser justificada sin la existencia de una revisión. De hecho, aseguran que una contracción conforma un estado intermedio en pos de la completa especificación de la operación de revisión. Este es el caso del modelo de revisión ATC.

Como consecuencia, la axiomatización de la revisión presentada aquí es alcanzada mediante el análisis de las diferentes características propias de ambas operaciones de revisión y contracción, tal como son introducidas en la literatura clásica de revisión de creencias (AGM85; Han94; Han99; HW02) (ver Sección 2.2). Los postulados discutidos allí son estudiados como motivación para la propuesta de un conjunto básico de postulados para la operación de revisión de bases de creencias aplicada sobre argumentación y árboles de dialéctica como semántica argumentativa. Tal conjunto básico de postulados, denominado *postulados de revisión argumentativa*, incluirán *éxito*, *consistencia*, *inclusión*, *vacuidad*, *retención de núcleo*, y *uniformidad*.

A continuación introduciremos los postulados de revisión argumentativa uno a uno, analizando sus correspondientes intuiciones con respecto a sus contrapartes en revisión de creencias clásica. Para tal propósito, asumiremos el operador "$*$" como la revisión clásica para bases de la teoría de revisión de creencias, y el operador "$*^{\omega}$" como una revisión de argumentos. Además consideraremos una KB $\mathcal{L}$ $\Sigma$, una creencia $\alpha \in \mathcal{L}$, y un argumento $\mathcal{R} \in \mathbb{A}_{\mathcal{L}}$ tal que $\mathfrak{cl}(\mathcal{R}) = \alpha$. El objetivo es lograr la caracterización axiomática de la operación $\Sigma *^{\omega} \mathcal{R}$ mediante el análisis de la caracterización de la operación clásica $\Sigma * \alpha$.

Mediante el postulado de *éxito* se asegura que la nueva información $\alpha$ sea satisfecha por la KB revisada. Esto es usualmente escrito como $\Sigma * \alpha$ *implies* $\alpha$. Desde la óptica de la argumentación, su interpretación estaría dada por la garantía del nuevo argumento $\mathcal{R}$ el cual soporta a su claim $\alpha$.

**(éxito)** $\mathcal{R}$ es garantizado en $\Sigma *^{\omega} \mathcal{R}$

A través del postulado de *consistencia*, una operación de revisión clásica asegura que la base revisada será consistente siempre que la nueva creencia a incorporar así lo sea. Esto es, $\Sigma * \alpha$ *es consistente si* $\alpha$ *es consistente.* La teoría de argumentación nos da la oportunidad de razonar consistentemente sobre KBs potencialmente inconsistentes. Por lo tanto, no existe necesidad de restauración de la consistencia de la KB revisada. De hecho, ese es nuestro objetivo primordial: el manejo de la dinámica del conocimiento sobre inconsistencias. Por otro lado, cuando hablamos de consistencia en argumentación, hacemos referencia a las semánticas argumentativas: deben asegurar que el conjunto de argumentos garantizados sea libre de conflictos.

**(consistencia)** Para cualquier $\{\mathcal{B}, \mathcal{C}\} \subseteq \mathbb{A}_{(\Sigma *^{\omega} \mathcal{R})}$, si $\mathcal{B}$ y $\mathcal{C}$ son garantizados en $\Sigma *^{\omega} \mathcal{R}$ entonces $\mathcal{B}$ y $\mathcal{C}$ no son conflictivos.

El postulado de consistencia no será necesario para la definición del teorema de representación dado que los modelos de cambio en ATC son dependientes del criterio de marcado el cual conduce a una *semántica argumentativa consistente*[1]. En consecuencia, cualquier

[1]Para mayor detalle sobre este tema, ver la Sección A.3, y en particular, el Teorema A.3.7.

modelo de cambio propuesto sobre tal criterio escéptico asegura la garantía del postulado de consistencia.

El postulado de *inclusión* tiene como objetivo garantizar que ninguna otra pieza de información además de $\alpha$ será incorporada. Esto es, $\Sigma * \alpha \subseteq \Sigma \cup \alpha$. Su adaptación a la argumentación está dada mediante la inclusión del cuerpo de $\mathcal{R}$.

$$\textbf{(inclusión)}\ \Sigma{*^{\omega}}\mathcal{R} \subseteq \Sigma \cup \mathfrak{bd}(\mathcal{R})$$

Inclusión sólo hace referencia a creencias de la KB. Desde el punto de vista del conjunto de argumentos, la relación es bastante diferente: nueva información es adicionada a la KB con la intención de construir $\mathcal{R}$, pero en consecuencia otros argumentos podrían aparecer.

**Observación 4.3.1** $\mathbb{A}_{(\Sigma*^{\omega}\mathcal{R})} \not\subseteq \mathbb{A}_{\Sigma} \cup \{\mathcal{R}\}$

El postulado de *vacuidad* captura las condiciones bajo las cuales la operación de revisión no tiene nada por hacer más que la incorporación de la nueva información $\alpha$. Esto es usualmente escrito como *Si* $\Sigma$ *no implica* $\neg\alpha$ *entonces* $\Sigma \cup \alpha \subseteq \Sigma * \alpha$. Su adaptación a la argumentación puede ser vista como el hecho de que $\mathcal{R}$ sea garantizado en forma directa, sin necesidad de remover información de la base. Por lo tanto, sólo la incorporación de $\mathcal{R}$ es requerida para obtener un árbol garantizante enraizado en $\mathcal{R}$.

$$\textbf{(vacuidad)}\ \text{Si } \mathcal{R} \text{ es garatizado en } \Sigma \cup \mathfrak{bd}(\mathcal{R}) \text{ entonces } \Sigma \cup \mathfrak{bd}(\mathcal{R}) \subseteq \Sigma{*^{\omega}}\mathcal{R}$$

Mediante el postulado de *retención de núcleo*, la cantidad de cambio es controlada evitando remociones que no están relacionadas a la revisión, *i.e.*, toda creencia que es removida sirve para permitir la aceptación de $\alpha$. Este postulado fue presentado en (HW02) como una adaptación para revisiones, tomando en cuenta las intuiciones de su propuesta original sobre contracciones, tal que *Si* $\beta \in \Sigma \setminus (\Sigma * \alpha)$, *entonces existe algún* $\Sigma' \subseteq \Sigma$ *tal que* $\Sigma' \cup \{\alpha\}$ *es consistente pero* $\Sigma' \cup \{\alpha, \beta\}$ *no lo es.* En argumentación, la consistencia no es el punto relevante, sino la condición de garantía de $\mathcal{R}$. Por ello, cualquier creencia $\beta$ que es removida debería ser necesaria para alcanzar una alteración efectiva.

$$\begin{gathered}\textbf{(retención de núcleo)}\ \text{Si } \beta \in \Sigma \setminus (\Sigma{*^{\omega}}\mathcal{R}) \text{ entonces existe algún } \Sigma' \subseteq \Sigma \text{ tal que} \\ \mathcal{R} \text{ es garantizado en } \Sigma' \cup \mathfrak{bd}(\mathcal{R}) \text{ pero no en } \Sigma' \cup \mathfrak{bd}(\mathcal{R}) \cup \{\beta\}\end{gathered}$$

Como es usual en revisión de creencias, es natural asumir que las revisiones aplicadas a $\Sigma$ por $\alpha$ o $\beta$ (dos fórmulas $\mathcal{L}$ lógicamente equivalentes) tienen necesariamente resultados idénticos. Por ejemplo, $\Sigma * (\neg p \vee q)$ debería ser igual a $\Sigma * (p \rightarrow q)$. Esto es capturado por el postulado de *extensionalidad*: *Si* $\alpha$ *sssi* $\beta$ *entonces* $\Sigma * \alpha = \Sigma * \beta$. Sin embargo, extensionalidad permite las siguientes revisiones: (1) $\{r, r \rightarrow p\} * \neg p = \{r \rightarrow p, \neg p\}$ y (2) $\{r, r \rightarrow p\} * \neg(p \vee q) = \{r, \neg(p \vee q)\}$. Aunque $\neg p$ y $\neg(p \vee q)$ no son lógicamente equivalentes, ellos son "equivalentes" si consideramos que sus complementos son igualmente implicados por $\{r, r \rightarrow p\}$ y sus subconjuntos (ver (Han99)). La alternativa de cuáles elementos de la KB retener dependerá de sus relaciones lógicas con la nueva información. Por ello, si dos sentencias son inconsistentes con los mismos subconjuntos de $\Sigma$, ellos deberían eliminar los mismos elementos de $\Sigma$. Esto es conocido como *uniformidad*: *Para toda* $\Sigma' \subseteq \Sigma$*, si* $\Sigma' \cup \{\alpha\}$ *es inconsistente sssi* $\Sigma' \cup \{\beta\}$ *es inconsistente, entonces* $\Sigma \cap (\Sigma * \alpha) = \Sigma \cap (\Sigma * \beta)$. El postulado de uniformidad es usado para bases de creencias como una versión más estricta de extensionalidad, *i.e.*, implica extensionalidad, pero no es implicado por ella.

La cuestión ahora es cómo puede ser adaptado el postulado de uniformidad a la revisión argumentativa. Necesitaremos especificar una relación $\tau \subseteq \mathbb{A}_{\mathcal{L}} \times \mathbb{A}_{\mathcal{L}}$ entre pares de argumentos $\mathcal{R}_1 \in \mathbb{A}_{\mathcal{L}}$ y $\mathcal{R}_2 \in \mathbb{A}_{\mathcal{L}}$ tal que si $\tau(\mathcal{R}_1, \mathcal{R}_2)$ entonces surgirán dos KBs revisadas equivalentes $\Sigma *^{\omega} \mathcal{R}_1$ y $\Sigma *^{\omega} \mathcal{R}_2$. Dado que el modelo de cambio argumentativo que estudiamos está basado en el análisis de árboles de dialéctica, necesitamos asegurar que la adición de cualquiera de esos argumentos determina árboles de dialéctica $\mathcal{T}(\mathcal{R}_1) \in \mathbb{T}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_1))}$ y $\mathcal{T}(\mathcal{R}_2) \in \mathbb{T}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_2))}$, tal que $\mathcal{T}(\mathcal{R}_1)$ es garantizante *sssi* $\mathcal{T}(\mathcal{R}_2)$ es garantizante. Por tal motivo, ambos árboles deberían ser *morfológicamente idénticos*.

**Definición 4.3.2 (Árboles Morfológicamente Idénticos)** *Dos árboles enraizados en* $\mathcal{R}_1$ *y* $\mathcal{R}_2$ *se dicen* ***morfológicamente idénticos*** *sssi o bien ninguno tiene derrotadores, o* $\mathcal{R}_1$ *tiene hijos (derrotadores directos de la raíz)* $\{\mathcal{B}_1, \ldots, \mathcal{B}_n\}$*, y* $\mathcal{R}_2$ *tiene hijos* $\{\mathcal{C}_1, \ldots, \mathcal{C}_m\}$ *tal que (1)* $n = m$*; y (2) para todo* $1 \leq i \leq n$*, los subárboles enraizados en* $\mathcal{B}_i$ *y* $\mathcal{C}_i$ *son árboles morfológicamente idénticos.*

Luego, la alteración de cualquiera de tales árboles, $\mathcal{T}(\mathcal{R}_1)$ y $\mathcal{T}(\mathcal{R}_2)$, debería concluir equivalentemente para que ambas revisiones, $\Sigma *^{\omega} \mathcal{R}_1$ y $\Sigma *^{\omega} \mathcal{R}_2$, se comporten de la misma manera. Esto es, las mismas creencias deberían ser descartadas de $\Sigma$ por cualquiera de las dos revisiones. A fin de provocar alteraciones equivalentes de árboles morfológicamente idénticos, también necesitamos asegurar que para cualquier par de líneas $\lambda_1 \in \mathcal{T}(\mathcal{R}_1)$ y

$\lambda_2 \in \mathcal{T}(\mathcal{R}_2)$, tal que $\lambda_1 = [\mathcal{R}_1, \mathcal{B}_1, \ldots, \mathcal{B}_n]$ y $\lambda_2 = [\mathcal{R}_2, \mathcal{C}_1, \ldots, \mathcal{C}_n]$, si $\tau(\mathcal{R}_1, \mathcal{R}_2)$ entonces $\tau(\mathcal{B}_i, \mathcal{C}_i)$ se verifica para cualquier $1 \leq i \leq n$. Por ejemplo, considerando "$\tau$" como la igualdad "$=$" entre argumentos requerirá los mismos argumentos raíz para ambos árboles, determinando así un único árbol aceptable. Por otro lado, considerando "$\tau$" como la equivalencia "$\equiv$" parece ser demasiado permisivo dado que sus respectivos árboles podrían no concluir siendo morfológicamente idénticos.

**Ejemplo 27** *Dada una KB $\Sigma = \{q \rightarrow \neg r,\ q \rightarrow r\}$, y dos argumentos externos $\mathcal{R}_1 \in \mathbb{E}_\Sigma$ y $\mathcal{R}_2 \in \mathbb{E}_\Sigma$, tal que $\mathcal{R}_1 \equiv \mathcal{R}_2$, donde $\mathcal{R}_1 = \langle\{q, r, (r \wedge q \rightarrow p)\}, p \wedge q \wedge r\rangle$, y $\mathcal{R}_2 = \langle\{r, (r \rightarrow p \wedge q)\}, p \wedge q \wedge r\rangle$. Los argumentos $\mathcal{B}_1 \in \mathbb{A}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_1))}$ y $\mathcal{B}_2 \in \mathbb{A}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_2))}$ son equivalentes ($\mathcal{B}_1 \equiv \mathcal{B}_2$) con cuerpos idénticos $\{q \rightarrow \neg r\}$. Además, $\mathcal{B}_1$ y $\mathcal{B}_2$ son undercuts canónicos de $\mathcal{R}_1$ y $\mathcal{R}_2$, respectivamente. Sin embargo, si bien el argumento $\mathcal{C}_1 \in \mathbb{A}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_1))}$, tal que $\mathcal{C}_1 = \langle\{q, q \rightarrow r\}, \neg(q \rightarrow \neg r)\rangle$, es un undercut canónico de $\mathcal{B}_1$, no existe argumento equivalente que sea undercut canónico de $\mathcal{B}_2$ en $\mathbb{A}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_2))}$.*

Es claro que, considerando argumentos raíz equivalentes no siempre es posible obtener árboles morfológicamente idénticos. Por este motivo, reforzaremos la relación de equivalencia "$\equiv$" entre argumentos, definiendo una *equivalencia estricta* "$\dashv\Vdash$" en representación de la relación "$\tau$".

**Definición 4.3.3 (Argumentos Estrictamente Equivalentes)** *Dos argumentos $\mathcal{R}_1$ y $\mathcal{R}_2$ contenidos en $\mathbb{A}_\mathcal{L}$ se dicen **estrictamente equivalentes**, notado $\mathcal{R}_1 \dashv\Vdash \mathcal{R}_2$ sssi $\mathcal{R}_1 \equiv \mathcal{R}_2$ y para cualquier subconjunto $\Psi_1 \subseteq \mathfrak{bd}(\mathcal{R}_1)$ existe un subconjunto $\Psi_2 \subseteq \mathfrak{bd}(\mathcal{R}_2)$ tal que $\Psi_1$ sssi $\Psi_2$.*

**Ejemplo 28** *Dada una KB $\Sigma = \{q \rightarrow \neg r\}$, y dos argumentos externos $\mathcal{R}_1 \in \mathbb{E}_\Sigma$ y $\mathcal{R}_2 \in \mathbb{E}_\Sigma$, tal que $\mathcal{R}_1 \dashv\Vdash \mathcal{R}_2$, donde $\mathcal{R}_1 = \langle\{q, (q \rightarrow p), (q \rightarrow r)\},\ p \wedge q \wedge r\rangle$, y $\mathcal{R}_2 = \langle\{q, (\neg q \vee p), (\neg q \vee r)\},\ p \wedge q \wedge r\rangle$. El argumento $\mathcal{B}_1 = \langle\{q \rightarrow \neg r\}, \neg(q \wedge (q \rightarrow p) \wedge (q \rightarrow r))\rangle \in \mathbb{A}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_1))}$ es un nuevo undercut canónico de $\mathcal{R}_1$, y $\mathcal{B}_2 = \langle\{q \rightarrow \neg r\}, \neg(q \wedge (\neg q \vee p) \wedge (\neg q \vee r))\rangle \in \mathbb{A}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_2))}$, de $\mathcal{R}_2$. Observe que $\mathcal{B}_1 \dashv\Vdash \mathcal{B}_2$ se verifica y, además, tal condición también se mantiene para los undercuts canónicos de $\mathcal{B}_1$ y $\mathcal{B}_2$ con cuerpos $\{q, q \rightarrow r\}$ y $\{q, \neg q \vee r\}$, respectivamente.*

Dado un par de argumentos externos $\mathcal{R}_1$ y $\mathcal{R}_2$, si son estrictamente equivalentes, sus árboles de dialéctica $\mathcal{T}(\mathcal{R}_1) \in \mathbb{T}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_1))}$ y $\mathcal{T}(\mathcal{R}_2) \in \mathbb{T}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_2))}$ no sólo serán morfológicamente idénticos, sino que también *estrictamente equivalentes*.

**Definición 4.3.4 (Árboles y Líneas Estrictamente Equivalentes)** *Dos líneas de argumentación* $\lambda_1 \in \mathbb{L}_\Sigma$ *y* $\lambda_2 \in \mathbb{L}_{\Sigma'}$, *donde* $\lambda_1 = [\mathcal{B}_1 \ldots, \mathcal{B}_n]$ *y* $\lambda_2 = [\mathcal{C}_1 \ldots, \mathcal{C}_n]$, *son* ***estrictamente equivalentes*** *sssi* $\mathcal{B}_i \dashv\Vdash \mathcal{C}_i$ *para* $1 \leq i \leq n$. *Dos árboles de dialéctica* $\mathcal{T}(\mathcal{B}_1) \in \mathbb{T}_\Sigma$ *y* $\mathcal{T}(\mathcal{C}_1) \in \mathbb{T}_{\Sigma'}$ *son* ***estrictamente equivalentes*** *sssi para cualquier* $\lambda \in \mathcal{T}(\mathcal{B}_1)$ *(resp.,* $\lambda \in \mathcal{T}(\mathcal{C}_1)$*) existe una línea estrictamente equivalente* $\lambda' \in \mathcal{T}(\mathcal{C}_1)$ *(resp.,* $\lambda' \in \mathcal{T}(\mathcal{B}_1)$*).*

**Lema 4.3.5** *Dada una KB* $\Sigma \subseteq \mathcal{L}$, *dos argumentos externos* $\mathcal{R}_1 \in \mathbb{E}_\Sigma$ *y* $\mathcal{R}_2 \in \mathbb{E}_\Sigma$, *y sus árboles de dialéctica* $\mathcal{T}(\mathcal{R}_1) \in \mathbb{T}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_1))}$ *y* $\mathcal{T}(\mathcal{R}_2) \in \mathbb{T}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_2))}$; *si* $\mathcal{R}_1 \dashv\Vdash \mathcal{R}_2$ *entonces* $\mathcal{T}(\mathcal{R}_1)$ *y* $\mathcal{T}(\mathcal{R}_2)$ *son árboles estrictamente equivalentes.*

Con el objetivo de revisar una KB $\Sigma$ por un argumento $\mathcal{R}$, posiblemente necesitaremos eliminar alguna/s creencia/s de $\Sigma$. Sin embargo, las creencias incluidas en $\mathcal{R}$ no deberían ser comprometidas. Esto es necesario para asegurar que $\mathcal{R}$ pertenecerá al conjunto resultante de argumentos $\mathbb{A}_{(\Sigma *^\omega \mathcal{R})}$, y también para permitir que $\mathcal{R}$ logre ser garantizado. Esto implica que sólo parte de las creencias originales de la base pueden ser eliminadas. Esto demuestra la cualidad de priorización[2] del operador de revisión argumental, es decir que la nueva información contenida en $\mathcal{R}$ tendrá mayor prioridad que el resto del conocimiento incluido en la KB.

En la consideración para revisar $\Sigma$ por cualquiera de dos (o más) argumentos externos estrictamente equivalentes, los conjuntos de creencias a remover de $\Sigma$ concluirán siendo idénticos dado que, a partir del Lema 4.3.5, ambos (o todos los) árboles son estrictamente equivalentes, y que ninguna nueva información será afectada. El postulado de *uniformidad* es finalmente adaptado a la argumentación de la siguiente manera:

**(uniformidad)** Si $\mathcal{R}_1 \dashv\Vdash \mathcal{R}_2$ entonces $\Sigma \cap \Sigma *^\omega \mathcal{R}_1 = \Sigma \cap \Sigma *^\omega \mathcal{R}_2$

La construcción de una revisión de argumento requiere asegurar que la alteración de diferentes árboles de dialéctica estrictamente equivalentes se desarrollará de la misma forma para ambos árboles. Por tal motivo, introducimos una condición adicional sobre las funciones de incisión para garantizar tal propiedad. Las intuiciones seguidas aquí fueron inspiradas por las incisiones suaves o *smooth* del las Contracciones Kernel Suaves de Hansson (Han94).

[2]Ver en contraposición las revisiones de creencias no priorizadas en la Sección 2.2.3.

**Definición 4.3.6 (Incisión Suave de Argumentos)** *Una incisión $\sigma$ es **suave** sssi para cualquier KB $\mathcal{L}$ $\Sigma$, y cualquier $\{\mathcal{R}_1, \mathcal{R}_2\} \subseteq \mathbb{E}_\Sigma$; si $\mathcal{R}_1 \dashv\Vdash \mathcal{R}_2$ entonces para cualquier $\mathcal{B}_1 \in \mathbb{A}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_1))}$ existe $\mathcal{B}_2 \in \mathbb{A}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_2))}$ tal que $\mathcal{B}_1 \dashv\Vdash \mathcal{B}_2$ y $\sigma(\mathcal{T}(\mathcal{R}_1)) \cap \mathfrak{bd}(\mathcal{B}_1) = \sigma(\mathcal{T}(\mathcal{R}_2)) \cap \mathfrak{bd}(\mathcal{B}_2)$.*

**Proposición 4.3.7** *Dada una KB $\Sigma \subseteq \mathcal{L}$, y dos argumentos $\mathcal{R}_1 \in \mathbb{E}_\Sigma$ y $\mathcal{R}_2 \in \mathbb{E}_\Sigma$ tal que $\mathcal{R}_1 \dashv\Vdash \mathcal{R}_2$, siempre existe una función de incisión de argumentos $\sigma$ la cual es suave.*

Una revisión de argumento será referida como suave si y sólo si es definida en términos de una incisión suave de argumentos. A continuación presentamos el teorema de representación para el modelo dialéctico-global. Es importante observar que, si bien en revisión de creencias clásica, la garantía de los postulados de éxito, inclusión, y retención de núcleo, implican vacuidad, en argumentación esto no es posible debido a la naturaleza no-monótona de las semánticas argumentativas.

**Teorema de Representación 4.3.8** *Dada una $\Sigma \subseteq \mathcal{L}$ y $\mathcal{R} \in \mathbb{A}_\mathcal{L}$, $\Sigma *^\omega \mathcal{R}$ es una revisión suave de argumentos sssi garantiza éxito, inclusión, vacuidad, retención de núcleo, y uniformidad.*

## 4.4. Argumentación DL y Evolución Ontológica

En esta sección, nos basaremos en el marco proposicional simple presentado en la Sección 4.1 y en algunas de las construcciones utilizadas para proveer el `GenAF` en el Capítulo 3, a fin de definir un marco argumentativo para el manejo de DLs en general. Nuestro objetivo es la propuesta de un marco simple para DLs que admita la aplicación del modelo dialéctico-global definido anteriormente. Por lo tanto, la noción de argumentos estará dada de la misma forma en que se propuso el marco proposicional. Esto es, un argumento será auto-conclusivo para inferir su claim, similar a la Definición 4.1.1, y a la noción de estructura argumental dada para el `GenAF` en la Definición 3.2.8.

Sin embargo, para definir el *marco DL-argumentativo* necesitaremos diferenciar el lenguaje de los claims del de las fórmulas contenidas en los argumentos, *i.e.*, de las fórmulas que componen la ontología. Esta necesidad es justificada por los siguientes puntos: necesitamos definir un marco basado en consultas, dado que el modelo dialéctico-global se aplica sobre árboles de dialéctica. Además, como ya hemos mencionado anteriormente, un marco basado en consultas construye sólo los argumentos necesarios para la formación del árbol de dialéctica enraizado en el argumento soporte de la consulta. Por otro lado, dado que la KB subyacente será una ontología basada en DLs, las consultas deberán corresponderse con la forma de los servicios de razonamiento (RSs) dados para la DL en uso específica (ver Sección 2.4.3).

Teniendo en mente tales necesidades, diremos que un *argumento DL* será una pieza de información interrelacionada (correspondiente a una DL $\mathcal{L}$) soportando (infiriendo) un claim. Un lenguaje $\mathcal{L}_{\mathtt{cl}}$ para claims será asumido. Similar a la noción de lenguaje argumental legal (ver Definición 3.1.2), si asumimos que el lenguaje argumental DL está dado como $2^{\mathcal{L}} \times \mathcal{L}_{\mathtt{cl}}$, entonces su condición de legalidad será: para cualquier $\phi \in \mathcal{L}_{\mathtt{cl}}$, existe un conjunto $\Phi \subseteq \mathcal{L}$ tal que $\Phi \models \phi$. La verificación de esta condición dependerá de la DL específica subyacente. Es claro que para cualquier DL, si el lenguaje $\mathcal{L}_{\mathtt{cl}}$ conforma los diferentes RSs propios de una DL $\mathcal{L}$, el lenguaje argumental será legal. Asumiendo su veracidad, seguiremos adelante en la definición del marco DL argumentativo, mediante la formalización de la noción de argumento DL.

**Definición 4.4.1 (Argumento DL)** *Un **argumento DL**, o simplemente **argumento**, $\mathcal{B}$ es una estructura $\langle \Delta, \beta \rangle$, donde $\Delta \subseteq \mathcal{L}$ es el **cuerpo**, $\beta \in \mathcal{L}_{\mathtt{cl}}$ el **claim**, y se verifica* (1) $\Delta \models \beta$, (2) $\Delta$ *es satisfacible (ó $\Delta \not\models \bot$), y* (3) $\nexists X \subset \Delta : X \models \beta$.

En el resto del capítulo, haremos uso del lenguaje $\mathcal{L}$ para referirnos a alguna DL como las introducidas en la Sección 2.4, a menos que $\mathcal{L}$ sea notado mediante la cadena identificatoria del nombre de una DL específica. Para el razonamiento ontológico asumimos una ontología $\Sigma$ expresada según la especificación de una DL $\mathcal{L}$. El *dominio de argumentos* de la ontología $\Sigma \subseteq \mathcal{L}$ es identificado por el conjunto $\mathbb{A}_{\Sigma}$. Observe que la inferencia "$\models$" será obtenida a través de las interpretaciones DL usuales, específicas de la DL $\mathcal{L}$ en uso. Para argumentos $\mathcal{B}$, se asumirá al cuerpo $\mathfrak{bd}(\mathcal{B})$ contenido en $\Sigma$. Además, como hemos mencionado, las consultas a $\Sigma$ serán soportadas a través del claim de los argumentos, por ello asumimos que el claim $\mathfrak{cl}(\mathcal{B}) \subseteq \mathcal{L}_{\mathtt{cl}}$ corresponderá a alguno de los RSs correspondientes a la DL $\mathcal{L}$.

En consecuencia, definimos el lenguaje de claims como $\mathcal{L}_{\mathtt{cl}} \longrightarrow C_1 \sqsubseteq C_2 | q(\langle\rangle)$, y adicionalmente consideraremos $E_1 \sqsubseteq E_2$, cuando $\mathcal{L} = \textit{DL-Lite}_{\mathcal{R}}$ o cuando la cadena $\mathcal{L}$ contenga la letra $\mathcal{H}$; y $(\mathrm{funct} R) | \neg(\mathrm{funct} R)$, cuando $\mathcal{L} = \textit{DL-Lite}_{(\mathcal{RF})}$ o $\textit{DL-Lite}_{\mathcal{F}}$. Observe que a partir de la sintaxis $\mathcal{L}_{\mathtt{cl}}$, $q(\langle\rangle)$ hace referencia a una consulta conjuntiva (cq) con una tupla vacía de $\mathrm{N_V}$. Luego, ningún argumento soportará un claim con variables libres. Esto es, para cualquier ontología $\Sigma \subseteq \mathcal{L}$ y cualquier argumento $\mathcal{B} \in \mathbb{A}_\Sigma$, si $\mathfrak{cl}(\mathcal{B}) = q(\bar{x})$ entonces $\bar{x} = \langle\rangle$. Dada una consulta $\alpha \in \mathcal{L}_{\mathtt{cl}}$, un argumento $\mathcal{B}$ es un *soporte de consulta* si $\mathcal{B}$ soporta a $\alpha$. Definiremos la noción de soporte de consulta para dar con una RS basándonos en las nociones de interpretación y *match* de DLs vistas en la Sección 2.4.3.

**Definición 4.4.2 (Soporte de Consulta)** *Dada una ontología $\Sigma \subseteq \mathcal{L}$ y una consulta $\alpha \in \mathcal{L}_{\mathtt{cl}} \cup \{q(\bar{x})\}$; un argumento $\mathcal{B} \in \mathbb{A}_\Sigma$ es un $\alpha$-**soporte** sssi para todo modelo $\mathcal{I}$ de $\mathfrak{cl}(\mathcal{B})$ existe un match $\mathfrak{m}$ para $\mathcal{I}$ y $\alpha$ tal que $\mathcal{I} \models^{\mathfrak{m}} \alpha$ se verifica, o equivalentemente $\mathfrak{cl}(\mathcal{B}) \models \alpha$.*

Similar a lo visto para el marco proposicional de la Sección 4.1, los argumentos *primitivos* como $\langle\{A(a)\}, A(a)\rangle$ surgen cuando $A(a) \in \Sigma$; asumiendo $\{A(a),\ A \sqsubseteq B\} \subseteq \Sigma$, argumentos más complejos aparecen dentro de $\mathbb{A}_\Sigma$, como $\langle\{A \sqsubseteq B\}, A \sqsubseteq B\rangle$ y $\langle\{A(a), A \sqsubseteq B\}, B(a)\rangle$. Las nociones de subargumentación se mantendrán en forma usual. Observe que $\langle\{A \sqsubseteq B\}, A \sqsubseteq B\rangle$ es un subargumento de $\langle\{A(a), A \sqsubseteq B\}, B(a)\rangle$. Un chequeo de funcionalidad $(\mathrm{funct}R)$ es soportado a través de un argumento $\langle\{(\mathrm{funct}R)\}, (\mathrm{funct}R)\rangle$. Otros argumentos pueden considerar aserciones funcionales, como por ejemplo, $\neg P(a,c)$ es soportado por ambos $\langle\{P(a,b), (\mathrm{funct}P)\}, \{\neg P(a,c)\}\rangle$ y $\langle\{P(b,c), (\mathrm{funct}P^-)\}, \{\neg P(a,c)\}\rangle$. CQs como $q(\bar{x})$ donde, por ejemplo, $q(\bar{x}) = \{A(x), B(y)\}$ y $\bar{x} = \langle x, y\rangle$ pueden ser soportadas por un argumento $\mathcal{B} = \langle\{A(a), A \sqsubseteq B\}, q(\bar{y})\rangle$ donde, $q(\bar{y}) = \{A(a), B(a)\}$, $\bar{y} = \langle\rangle$, y surge un match $\mathfrak{m}(x) = \mathfrak{m}(y) = a$. Observe que cambiando la consulta a $q(\bar{x}) = \{A(x), B(y), x \neq y\}$, el argumento $\mathcal{B}$ ya no será un $q(\bar{x})$-soporte. A partir de ahora haremos referencia a argumentos como $\mathcal{B}$ escribiendo $\langle\{A(a), A \sqsubseteq B\}, \{A(a), B(a)\}\rangle$, evitando la notación explícita de cqs como $q(\langle\rangle) = \{A(a), B(a)\}$ en sus claims.

Para la noción de *contra-argumento*, mantendremos la formalización dada en la Definición 4.1.3. Observe que la condición $\{\mathfrak{cl}(\mathcal{B}_2)\} \cup \mathfrak{bd}(\mathcal{B}_1) \models \bot$ determina la insatisfabilidad respecto del cuerpo de $\mathcal{B}_1$ en conjunción con el claim de $\mathcal{B}_2$. Esto es natural, dado que desde el punto de vista de una ontología $\Sigma$, los contra-argumentos determinan fuentes de incoherencia/inconsistencia determinadas por sus cuerpos.

Usualmente, los argumentos pueden ser *rebuttals* de argumentos con claims opuestos o pueden ser *undercuts* de argumentos cuyo claim se opone en forma explícita o implícita al cuerpo del argumento derrotado. Por ejemplo, dos argumentos $\mathcal{B}$ y $\mathcal{C}$ tal que $\mathcal{C}$ contra-argumenta a $\mathcal{B}$, usualmente decimos que $\mathcal{C}$ es un rebuttal para $\mathcal{B}$ si se verifica $\mathfrak{cl}(\mathcal{B}) =$

$\neg\mathfrak{cl}(\mathcal{C})$. Sin embargo, en DLs esto puede ser conflictivo dado que la negación de axiomas DL pueden determinar axiomas que no corresponden al lenguaje de descripción utilizado. Por ese motivo, es que la formalización de contra-argumentos debe ser dada sobre la noción de satisfabilidad de DLs.

Analicemos con mayor detalle el problema de la negación de axiomas en el contexto de los argumentos DL. Dado un argumento $\mathcal{B} = \langle\{A \sqsubseteq B, B \sqsubseteq C\}, A \sqsubseteq C\rangle$, un posible rebuttal soportaría un axioma como $\neg(A \sqsubseteq C)$. La negación de axiomas como $A \sqsubseteq C$ fue estudiada en (FHP$^+$06) determinando dos tipos de negación: *negación por consistencia* $\neg(A \sqsubseteq C) = \exists(A \sqcap \neg C)$ y *negación por coherencia* $\sim(A \sqsubseteq C) = A \sqsubseteq \neg C$. Para el primer caso, una aserción existencial como $\exists(A \sqcap \neg C)(x)$ serviría, sin embargo, para los lenguajes de descripción sin conjunción de conceptos como *DL-Lite*$_{(\mathcal{RF})}$, caería fuera del alcance del lenguaje. Para tales casos, la aserción existencial será reescrita como una cq $q(\langle x\rangle) = \{A(x), \neg C(x)\}$. Por ejemplo, $\mathcal{C} = \langle\{A(a), A \sqsubseteq \neg C\}, \{A(a), \neg C(a)\}\rangle$ sería un posible rebuttal soportando $q(\langle x\rangle)$ con $\mathfrak{m}(x) = a$. Por otro lado, la negación por coherencia puede ser alcanzada simplemente buscando otro argumento soportando $A \sqsubseteq \neg C$. Adicionalmente, para *DL-Lite*$_{\mathcal{F}}$, extendemos la negación de axiomas a aserciones funcionales, interpretando $\neg(\text{funct}R)$ como un rol $R$ que no conforma la definición de función. Un argumento soportando tal negación debería considerar información extensional (obtenida de la ABox). Por ejemplo, $\langle\{P(a,b), P'(a,c), P' \sqsubseteq P\}, \neg(\text{funct}P)\rangle$.

Sin embargo, el mayor inconveniente aparece cuando pasamos a considerar el RS de *query answering*, dado que en general, la negación de cqs está indefinida. Como solución alternativa, nos concentraremos en encontrar fuentes de insatisfabilidad DL a través de *undercuts*. Observe que varios undercuts pueden aparecer para un mismo argumento. Además, como es mostrado en el siguiente ejemplo, un undercut de un argumento $\mathcal{B}$ puede contener como subargumentos, otros derrotadores de $\mathcal{B}$.

**Ejemplo 29** *Considere los argumentos DL-Lite*$_{\mathsf{core}}$ $\mathcal{B} = \langle\{A(a), A \sqsubseteq B\}, \{B(a)\}\rangle$ *y* $\mathcal{C} = \langle\{A(b), A \sqsubseteq C, C \sqsubseteq \neg B\}, \{A(b), \neg B(b)\}\rangle$*, y el subargumento* $\mathcal{C}'$ *de* $\mathcal{C}$*,* $\mathcal{C}' = \langle\{A \sqsubseteq C, C \sqsubseteq \neg B\}, A \sqsubseteq \neg B\rangle$*. Ambos* $\mathcal{C}$ *y* $\mathcal{C}'$ *son contra-argumentos de* $\mathcal{B}$*.*

Los *undercuts canónicos* (ver Definición 4.1.4) fueron definidos (sobre lógica clásica) como representantes de todos los derrotadores de un argumento. El claim de un undercut canónico $\mathcal{C}$ niega la enumeración conjuntiva de todas las creencias contenidas en el cuerpo del argumento contra-argumentado $\mathcal{B}$, *i.e.*, $\mathfrak{cl}(\mathcal{C}) = \neg(\alpha_1 \wedge \ldots \wedge \alpha_n)$, donde $\mathfrak{bd}(\mathcal{B}) =$

$\{\alpha_1, \ldots, \alpha_n\}$. Sin embargo, tal clase de claims caería fuera de la representación del lenguaje de descripción $\mathcal{L}$. Los undercuts canónicos constituyen fuentes mínimas de inconsistencia (en la KB a partir de la cual son construidos) respecto del cuerpo de un argumento contra-argumentado. Por lo tanto, siguiendo tal intuición, propondremos un nuevo tipo de undercut, al cual denominaremos *undercut mínimo*.

**Definición 4.4.3 (Undercut Mínimo)** *Dados $\mathcal{B}, \mathcal{C} \in \mathbb{A}_\Sigma$, tal que $\mathcal{C}$ es un contra-argumento de $\mathcal{B}$. $\mathcal{C}$ es un* ***undercut mínimo*** *de $\mathcal{B}$ sssi no existe subargumento propio de $\mathcal{C}$ contra-argumentando $\mathcal{B}$.*

Para ilustrar esta noción, observe en el Ejemplo 29, que $\mathcal{C}'$ es un undercut mínimo de $\mathcal{B}$. Mediante el uso de undercuts mínimos, hemos logrado restringir la consideración de contra-argumentos sólo a aquellos cuyo cuerpo es mínimo para la determinación de la insatisfabilidad con respecto al argumento contra-argumentado. Sin embargo, para algunas DLs pueden aparecer varios undercuts mínimos con diferentes claims (ver Ejemplo 30). En tales casos, nos quedaremos con aquellos de inferencia máxima, es decir, aquellos cuyos claims infieran mayor cantidad de conocimiento. De acuerdo a esta intuición, identificaremos a tales tipos de undercuts mínimos como *undercuts maximalmente conservativos* o *maximally conservative undercuts* (*mcu*, por simplicidad).

**Definición 4.4.4 (Undercuts Maximalmente Conservativos)** *Dados $\mathcal{B} \in \mathbb{A}_\Sigma$ y $\mathcal{C} \in \mathbb{A}_\Sigma$ tal que $\mathcal{C}$ es un undercut mínimo de $\mathcal{B}$. $\mathcal{C}$ es un* ***undercut maximalmente conservativo*** *o* ***maximally conservative undercut (mcu)*** *de $\mathcal{B}$ sssi para todo subargumento $\mathcal{C}'$ de $\mathcal{C}$ tal que $\mathcal{C}'$ es un undercut mínimo de $\mathcal{B}$, se verifica $\mathfrak{cl}(\mathcal{C}) \models \mathfrak{cl}(\mathcal{C}')$.*

**Ejemplo 30** *Sean $\mathcal{C} = \langle \Psi, A \sqcup B \sqsubseteq \neg C \rangle$, $\mathcal{C}' = \langle \Psi, A \sqsubseteq \neg C \rangle$, y $\mathcal{C}'' = \langle \Psi, B \sqsubseteq \neg C \rangle$, con $\Psi = \{A \sqcup B \sqsubseteq D, D \sqsubseteq \neg C\}$; tres $\mathcal{ALC}$ undercuts mínimos de $\mathcal{B} = \langle \{(A \sqcap B)(a), A \sqcup B \sqsubseteq C\}, \{(A \sqcup B)(a), C(a)\} \rangle$. Observe que sólo $\mathcal{C}$ es un mcu de $\mathcal{B}$.*

Como hemos visto anteriormente, mediante el uso de árboles de dialéctica, decidimos la aceptabilidad, o estado de garantía, de un soporte de consulta $\mathcal{R} \in \mathbb{A}_\Sigma$, siguiendo el criterio de marcado sobre el árbol enraizado en $\mathcal{R}$. Tales árboles son construidos a partir de un conjunto máximo de líneas de argumentación enraizadas en $\mathcal{R}$.

La reducción del número de argumentos derrotadores nos permite encoger el conjunto de líneas de argumentación utilizadas para construir los árboles de dialéctica. En este sentido, el uso de mcus puede resultar beneficioso para tal fin, permitiendo la consideración de un único derrotador a partir de un conjunto de undercuts mínimos de un argumento en común. Sin embargo, para DLs donde la equivalencia lógica de claims es posible, varios mcus pueden aparecer.

**Ejemplo 31** *Dada* $\Sigma \subseteq \mathcal{ALC}$ *y* $\mathcal{B} = \langle \{A \sqsubseteq B', B' \sqsubseteq \neg B\}, A \sqsubseteq \neg B \rangle$ *contenido en* $\mathbb{A}_\Sigma$*. Asumiendo la construcción de los argumentos* $\langle \{A(a), B(a)\}, \{(A \sqcap B)(a)\} \rangle$ *y* $\langle \{A(a), B(a)\}, \{A(a), B(a)\} \rangle$*, ambos argumentos son mcus de* $\mathcal{B}$ *con idénticos cuerpos y claims lógicamente equivalentes.*

La construcción de árboles de dialéctica aceptables requiere la identificación de los denominados *conjuntos bundle* (ver Definición 2.1.12). A tal fin, especificaremos el dominio $\mathbb{L}_\Sigma$ para argumentación DL, conteniendo sólo líneas aceptables y exhaustivas. Nos aseguraremos de que el conjunto $\mathbb{L}_\Sigma$ sea libre de redundancias como las vistas en el Ejemplo 31, mediante la restricción del uso de mcus con claims equivalentes, condición que denominaremos *no-redundancia*.

**Definición 4.4.5 (Dominio de Líneas de Argumentación)** *Dada una ontología* $\Sigma \subseteq \mathcal{L}$*, el* ***dominio de líneas de argumentación*** $\mathbb{L}_\Sigma$*, es el máximo conjunto de líneas de argumentación* $\lambda = [\mathcal{B}_1, \ldots, \mathcal{B}_n]$*, donde* $\mathcal{B}_i \in \mathbb{A}_\Sigma$ *(para* $1 \leq i \leq n$*), tal que* $\lambda$ *es aceptable, exhaustiva,* ***maximalmente conservativa*** *(*$\mathcal{B}_j$ *es un mcu de* $\mathcal{B}_{j-1}$ *(para* $1 < j \leq n$*)), y* ***no-redundancia*** *es garantizada:*

> **(no-redundancia)** *para cualquier* $\{\lambda, \lambda'\} \subseteq \mathbb{L}_\Sigma$*,* $\mathcal{B} \in \lambda$ *y* $\mathcal{C} \in \lambda'$*,*
> *si* $\mathfrak{bd}(\mathcal{B}) = \mathfrak{bd}(\mathcal{C})$ *y* $\lambda^{\uparrow}(\mathcal{B}) = \lambda'^{\uparrow}(\mathcal{C})$ *entonces* $\lambda = \lambda'$*.*

El conjunto bundle $\mathcal{S}(\mathcal{R})$ contendrá todas las líneas de argumentación incluidas en $\mathbb{L}_\Sigma$ enraizadas en $\mathcal{R}$. A partir del conjunto bundle $\mathcal{S}(\mathcal{R})$, el árbol de dialéctica aceptable $\mathcal{T}(\mathcal{R})$ es construido (ver Definición 2.1.13), por el cual decidiremos la garantía del argumento $\mathcal{R}$. Asumiremos al conjunto $\mathbf{L}_\Sigma$ para identificar el dominio de todas la líneas aceptables y maximalmente conservativas, construidas con argumentos de $\mathbb{A}_\Sigma$. Observe nuevamente, la dependencia determinada entre los dominios de líneas de argumentación: $\mathbb{L}_\Sigma \subseteq \mathbf{L}_\Sigma$.

Dada una ontología $\Sigma \subseteq \mathcal{L}$, un soporte de consulta $\mathcal{R} \in \mathbb{A}_\Sigma$ es aceptado (garantizado) si el árbol de dialéctica $\mathcal{T}(\mathcal{R}) \in \mathbb{T}_\Sigma$ es un árbol garantizante (de acuerdo a lo expuesto en la Sección 2.1.2). Nuevamente, esta evaluación se realizará mediante el uso de las funciones de marcado $\mathfrak{mark}$ y garantizante $\mathfrak{warrant}$, basándose en el criterio de marcado escéptico dado en la Definición 2.1.15.

**Definición 4.4.6 (Inferencia DL Argumentativa)** *Dada la ontología $\Sigma \subseteq \mathcal{L}$ y una consulta $\alpha \in \mathcal{L}_{\texttt{cl}} \cup \{q(\bar{x})\}$; $\Sigma \mid\!\approx \alpha$ sssi existe un $\alpha$-soporte garantizado $\mathcal{R} \in \mathbb{A}_{\Sigma}$. Si no existe $\alpha$-soporter garantizado en $\mathbb{A}_{\Sigma}$ entonces $\alpha$ no es inferida por $\Sigma$, notado como $\Sigma \mid\!\not\approx \alpha$.*

**Teorema 4.4.7** *Dada una ontología $\Sigma \subseteq \mathcal{L}$, si $\Sigma$ es coherente y consistente entonces para cualquier consulta $\alpha \in \mathcal{L}_{\texttt{cl}} \cup \{q(\bar{x})\}$ se verifica $\Sigma \models \alpha$ sssi $\Sigma \mid\!\approx \alpha$.*

En el siguiente ejemplo, estudiamos la revisión de argumentos presentada en la Definición 4.2.6, aplicada sobre una DL $\mathcal{ALCNH}^{-1,\neg}$.

**Ejemplo 32** *Durante el período previo a las elecciones presidenciales los candidatos son presentados en sociedad. Una fórmula presidencial es dada mediante un par de candidatos, el primero para presidente y el segundo para vice-presidente. Sería interesante analizar si una dada fórmula presidencial es confiable. A tal fin, proponemos analizar algunas de las decisiones tomadas por los candidatos incluidos en el par: por ejemplo, como se han declarado/votado respecto de las últimas leyes de mayor relevancia tratadas en el congreso. De esta forma, tenemos una manera de decidir si un par de candidatos es coalicionable, es decir, si es confiable su candidatura conjunta. Para la ontología a analizar, consideraremos los siguientes conceptos y roles:*

- *el rol $P$, representando la fórmula presidencial tal que para cualquier par $(x, y) \in P^{\mathcal{I}}$, el individuo $x$ es el candidato a presidente e $y$ es el candidato a la vice-presidencia.*
- *el rol $C$, representando candidatos coalicionables tal que para cualquier par $(x, y) \in C^{\mathcal{I}}$, los individuos $x$ e $y$ son dos políticos que se asumiremos acuerdan en términos ideológicos sobre los temas de mayor iterés nacional.*
- *los conceptos $L$ y $L'$, representando las dos leyes de mayor relevancia que han sido promulgadas durante el último período presidencial. Los individuos $a \in L^{\mathcal{I}}$ (resp., $a \in L'^{\mathcal{I}}$) identifican a los políticos que han votado a favor de la promulgación de $L$ (resp, $L'$).*

- *el concepto $L_1$, representa uno de los artículos más controversiales de la promulgación de la ley L. Los individuos $a \in L_1^{\mathcal{I}}$ identifican al político que ha votado a favor de la promulgación de $L_1$.*

*Incluiremos en la ontología $\Sigma \subseteq \mathcal{ALCNH}^{-1,\neg}$ la siguiente lista de axiomas terminológicos:*

1. *$P \sqsubseteq C$ toda fórmula presidencial es coalicionable.*
2. *$C \sqsubseteq C^-$ todo par coalicionable de políticos es conmutativo.*
3. *$L_1 \sqsubseteq L$ los políticos que han votado a favor del artículo $L_1$ deberían haber votado a favor de la ley $L$.*
4. *$\forall P.\top \sqsubseteq = 1P$ (equivalente a $(\mathrm{funct}P)$ en DL-Lite$_{(\mathcal{RF})}$) y*
5. *$\forall P^-.\top \sqsubseteq = 1P^-$ (equivalente a $(\mathrm{funct}P^-)$ en DL-Lite$_{(\mathcal{RF})}$), una fórmula presidencial debería ser única y los candidatos deberían anunciar explícitamente el cargo para el cual se postulan. Los candidatos presentados en varias fórmulas presidenciales se asumen menos confiables.*
6. *$\forall C.L \sqsubseteq L$, y*
7. *$\forall C.L' \sqsubseteq L'$, dos políticos que acuerdan sobre las leyes $L$ y $L'$ son coalicionables.*

*La ontología $\Sigma$ también incluirá $P(a,c)$, $P(d,e)$, $\neg C(d,e)$, $L(a)$, $\neg L(b)$, $L_1(b)$, $\neg L'(a)$, y $L'(c)$, donde los individuos $a, b, c, d$, y $e$, representan políticos de la escena nacional.*

*Necesitamos revisar la ontología $\Sigma$ por el argumento externo $\mathcal{R} = \langle\{P(a,b)\}, \{P(a,b)\}\rangle$. El árbol de dialéctica $\mathcal{T}(\mathcal{R})$ enraizado en $\mathcal{R}$ es ilustrado en la Figura 4.5.*

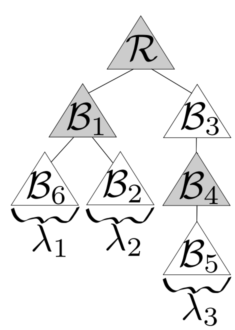



Figura 4.5: Árbol de Dialéctica $\mathcal{T}(\mathcal{R})$– Ejemplo 32.

*Además de $\mathcal{R}$, los siguientes argumentos serán considerados:*

- $\mathcal{B}_1 = \langle\{P \sqsubseteq C, C \sqsubseteq C^-, \forall C.L \sqsubseteq L, L(a), \neg L(b)\}, \{\neg P(a,b)\}\rangle$
- $\mathcal{B}_2 = \langle\{L_1(b), L_1 \sqsubseteq L\}, \{L(b)\}\rangle$
- $\mathcal{B}_3 = \langle\{P(a,c), \forall P.\top \sqsubseteq = 1P\}, \{\neg P(a,b)\}\rangle$
- $\mathcal{B}_4 = \langle\{P \sqsubseteq C, \forall C.L' \sqsubseteq L', \neg L'(a), L'(c)\}, \{\neg P(a,c)\}\rangle$
- $\mathcal{B}_5 = \langle\{P(d,e), \neg C(d,e)\}, \{P(d,e), \neg C(d,e)\}\rangle$

*La línea $\lambda_3$ es la única línea de ataque que corresponde a $\mathcal{T}(\mathcal{R})$, por ello, la función de incisión la alterará a fin de tornarla a no-atacante. Supongamos que $\sigma(\mathcal{T}(\mathcal{R})) = \{\neg C(d,e)\}$, entonces el argumento $\mathcal{B}_5$ será removido. Observe que en tal caso, la línea $\lambda_1$ será colateralmente alterada. Sin embargo, el segmento superior no es una línea de argumentación en el árbol hipotético $\mathcal{H}(\mathcal{R}, \{\neg C(d,e)\})$. Por lo tanto, el árbol de dialéctica resultante (ilustrado en la Figura 4.6.*(a)*) concluirá garantizando el argumento raíz $\mathcal{R}$.*


(a) (b)

Figura 4.6: Dos posibles árboles $\mathcal{T}(\mathcal{R})$ resultantes de $\Sigma *^{\omega} \mathcal{R}$ – Ejemplo 32.

*Por otro lado, supongamos una función de incisión $\sigma(\mathcal{T}(\mathcal{R})) = \{P(a,c)\}$. En ese caso, el argumento $\mathcal{B}_3$ sería removido. Finalmente, el árbol de dialéctica resultante (ilustrado en la Figura 4.6.*(b)*) concluirá garantizando el argumento raíz $\mathcal{R}$.*

### 4.4.1. ¿Cuán Factible es esta Metodología de Razonamiento DL No-estándar?

Para implementar nuesta maquinaria argumentativa DL dos cuestiones necesistan ser atendidas: (1) cómo construir un argumento soportando una consulta, y (2) cómo construir los derrotadores de un dado argumento. Para (1) podemos utilizar diferentes técnicas basadas en el proceso de razonamiento sobre la DL correspondiente. Algunos trabajos relacionados con este punto son (MLBP06) (aplicado sobre $\mathcal{ALC}$) y (BPS07) (sobre $\mathcal{EL}$). En esta sección, nos concentraremos en (2) basándonos en MUPS (minimal unsatisfiability-preserving sub-TBoxes) y MIPS (minimal incoherence-preserving sub-TBoxes) (SC03). Tales estructuras son definidas siguiendo una extensión del algoritmo tableau para la DL estándard $\mathcal{ALC}$ (BCM$^+$03) aplicada sobre unfoldable $\mathcal{ALC}$ (ver Sección 2.4.5). Este algoritmo es referido por los autores como *axiom pinpointing*. A continuación introduciremos las intuiciones para el cálculo de MUPS, y luego las extenderemos para proponer un algoritmo para construir los derrotadores de un argumento dado. Para ello, nos basaremos en ontologías $\mathcal{ALC}$ unfolded.

La insatisfabilidad de un concepto es detectada mediante un *tableau etiquetado saturado*[3]. Un *tableau etiquetado* es un conjunto de ramas etiquetadas. Una rama etiquetada es un conjunto de fórmulas etiquetadas de la forma $(a : C)^X$, donde $a$ es un individo, $C$ es un concepto, y $X$ es la etiqueta conteniendo un conjunto de axiomas determinando la inferencia de la fórmula $(a : C)$. Una fórmula puede ocurrir con diferentes etiquetas sobre la misma rama. Un tableau etiquetado se dice *saturado* si todas sus ramas están cerradas. Una rama está cerrada si contiene una contradicción, *i.e.*, si existe al menos un par de fórmulas con átomos contradictorios acerca del mismo individuo. Por lo tanto, la información sobre qué axiomas son relevantes para la clausura (contradicción) de una rama está contenida en las etiquetas de las fórmulas contradictorias. Por ejemplo, una rama $\lambda$ está cerrada si existe un par $(a : A)^X \in \lambda$ y $(a : \neg A)^Y \in \lambda$, donde $A$ es un concepto atómico. Esto es, los axiomas en $X$ y $Y$ determinan contradicciones, *i.e.*, $X \cup Y$ es insatisfacible.

La clausura de una rama es lograda mediante la aplicación de reglas de expansión las cuales abren (*unfold*) axiomas progresivamente de forma *lazy*. Un ejemplo de regla de expansión sobre una rama $\lambda$ es:

$$\text{Si } (a : A)^X \in \lambda \text{ y } A \sqsubseteq C \in \Sigma \text{ entonces } \lambda \text{ es reemplazado en el tableaux por}$$
$$\lambda \cup \{(a : C)^{X \cup \{A \sqsubseteq C\}}\}$$

Ramas adicionales pueden ser progresivamente incluidas en el tableaux siguiendo una regla disyuntiva la cual opera sobre fórmulas como $(a : C_1 \sqcap C_2)$. (Para mayor detalle acerca del conjunto de reglas de expansión, el lector es referido a (SC03).) Una vez que ya no queden reglas para ser aplicadas, un tableau cerrado es obtenido a partir del conjunto $S$ de ramas cerradas.

[3] La noción de tableau generalizado es introducida en Sección 2.4.7.

La estructura MUPS es formada mediante la construcción de un tableau etiquetado para una rama inicialmente conteniendo sólo $(a : A)^{\emptyset}$, y mediante la aplicación de una *función de minimización* $\varphi$ la cual también comienza en $(a : A)^{\emptyset}$. A medida que las reglas de expansión son aplicadas para cerrar el tableau, otras reglas diferentes son utilizadas para provocar la expansión de la función de minimización. La idea es obtener la menor conjunción de axiomas $\alpha_1 \wedge \ldots \wedge \alpha_n$, llamada *prime implicant*, implicando $\varphi$. Esto es construido a partir de las etiquetas de las fórmulas contradictorias contenidas en cada rama cerrada del tableaux, luego $\alpha_i \in \Sigma$, para cualquier $1 \leq i \leq n$. Como $\varphi$ es una función de minimización todo implicante de $\varphi$ es también una función de minimización. Finalmente, el prime implicant es también una función de minimización. Esto significa que $A$ es insatisfacible cuando $\alpha_1 \wedge \ldots \wedge \alpha_n$ es verdadero, o equivalentemente $A$ es insatisfacible con respecto al conjunto $M = \{\alpha_1, \ldots, \alpha_n\}$, el cual es mínimo dado que proviene de un prime implicant (la menor conjunción de axiomas implicando $\varphi$). La estructura MUPS para $A$ con respecto a $\Sigma$ es definida como:

$$mups(A, \Sigma) = \{M \subseteq \Sigma | A \text{ es insatisfacible en } M \text{ pero } A \text{ es satisfacible en todo } M' \subset M\}.$$

Para calcular los derrotadores de un argumento $\mathcal{B}$ dado, primeramente calculamos $mups(A, \Sigma)$ donde $A \sqsubseteq C \in \mathfrak{bd}(\mathcal{B})$. Para cualquier axioma $A' \sqsubseteq C' \in \mathfrak{bd}(\mathcal{B})$ que no fue considerado por el proceso de clausura del tableau (*i.e.*, no fue incluido en ninguna etiqueta), un nuevo MUPS $mups(A', \Sigma)$ deberá ser obtenido. (Sólo en el peor caso un MUPS para todo axioma dentro de $\mathcal{B}$ deberá ser construido.) Una vez que todos los MUPS necesarios son obtenidos, los unimos dentro del conjunto $mupsArg(\mathcal{B}, \Sigma) = \bigcup_{A \sqsubseteq C \in \mathfrak{bd}(\mathcal{B})} mups(A, \Sigma)$. Finalmente, para cualquier $M \in mupsArg(\mathcal{B}, \Sigma)$, se verifica que $M \setminus \mathfrak{bd}(\mathcal{B})$ es el cuerpo de un undercut mínimo de $\mathcal{B}$, conformando las definiciones 4.4.1, 4.1.3, y 4.4.3 (un mcu conformando la Definición 4.4.4 puede ser obtenido fácilmente reacomodando el claim). A fin de identificar inconsistencias además de incoherencias, *i.e.*, para encontrar aserciones contradictorias de la ABox, el algoritmo original MUPS debería considerar reglas de expansión adicionales, y de esta forma, tornando MUPS de sub-terminologías a sub-ontologías. Esta suposición no debería afectar el análisis siguiente.

El cálculo de la estructura MUPS se basa en la construcción de una función de minimización a partir de un tableau. Construyéndola de forma *depth-first* permite mantener en memoria una única rama a la vez. Por ello, la clase de complejidad del problema MUPS corresponde al problema de chequeo de satisfabilidad en unfoldable $\mathcal{ALC}$, *i.e.*, PSPACE.

Dado que el tamaño de los prime implicants puede ser exponencial con respecto al número de axiomas en la TBox, métodos de aproximación podrían evitar la construcción de tableaux completamente saturados a fin de reducir el tamaño de las funciones de minimización. Adicionalmente, a fin de determinar todos los derrotadores necesarios de un argumento $\mathcal{B}$ dado, la construcción de varios MUPS podría ser necesaria. Sin embargo, el proceso de unfolding encontraría (en general) la mayoría de los axiomas de $\mathcal{B}$.

**Teorema 4.4.8** *El cálculo de todos los derrotadores de un argumento $\mathcal{ALC}$ dado se encuentra en PSPACE.*

## 4.5. Trabajo Relacionado

En esta sección presentamos algunos de los trabajos existentes sobre revisión de creencias y argumentación. Parte de ellos, como (CdSCLS10; BKvdT09; BCPTvdT08; BdCPTvdT08; AV09), no proponen realmente modelos argumentativos de cambio, sino que más bien proveen un análisis sobre impacto que sufre el conjunto de garantías de un `AF` luego de la incorporación de un nuevo argumento. Primeramente describimos tales trabajos y los relacionamos con ATC. Luego, damos un breve resumen de otros artículos en los que se trabaja la relación entre revisión de creencias y otras teorías que de alguna forma están relacionadas con nuestro modelo de cambio argumentativo en pos de la revisión de KBs. Entre tales trabajos, sólo (PG00; PC06) utilizan argumentación, sin embargo, en una forma bastante diferente a lo hecho en los enfoques ATC: (PG00) se enfoca en la dinámica en epistemología, mientras que (PC06) lo hace sobre una KB particular para agentes, basándose en un modelo de dinámica del conocimiento, que es alternativo al clásico modelo AGM. Finalmente, describimos la relación entre el modelo dialéctico-global presentado en este capítulo y otros enfoques ATC existentes, propuestas en las cuales también ha participado el tesista.

### 4.5.1. Argumentación y Revisión de Creencias

La bibliografía que analizaremos respecto de la relación entre argumentación y revisión de creencias, fue introducida previamente en la Sección 3.6.3. Aquí relacionaremos tales trabajos con el modelo ATC dialéctico-global presentado en este capítulo.

En el caso de (CdSCLS10), como hemos mencionado, el modelo es restringido mediante la suposición de que el nuevo argumento incorporado poseerá a lo sumo una interacción (via ataque) con otro argumento del sistema. Esta restricción simplifica enormemente el problema de revisión de `AF`s, dado que las interacciones múltiples con el sistema original son comunes, e incluso su manejo puede resultar bastante dificultoso. Respecto de la revisión de KB con tolerancia a inconsistencias, esta restricción haría imposible la aplicación de tal modelo a la base de donde se construyen argumentos y ataques. En ATC, esto es solucionado mediante la inclusión de subargumentos y a través del manejo de las colateralidades. Adicionalmente, el objetivo de (CdSCLS10) difiere del nuestro en que nosotros aplicamos cambios adicionales (asumiendo que está permitido) al `AF` original (y consecuentemente, a la KB) en busca de la garantía de un argumento a través del análisis de árboles de dialéctica, mientras que ellos estudian como la adición de un dado argumento afectaría al conjunto de extensiones (sin desarrollar alteraciones adicionales más allá de la incorporación del nuevo argumento), mediante el análisis del grafo completo de argumentos. El trabajo sobre árboles de dialéctica parece ser una alternativa más apropiada para el operador de revisión, brindando la posibilidad de manejar un conjunto de argumentos de tamaño tratable, en forma controlada.

Existe una principal diferencia entre los modelos presentados en la bibliografía (CdSCLS10; AV09; BKvdT09; BCPTvdT08; BdCPTvdT08) (ver Sección 3.6.3) y ATC: ellos asumen que el marco argumentatico sólo puede ser cambiado como un reflejo de la evolución del mundo modelado y, por lo tanto, sus teorías sólo analizan el impacto del cambio sobre las semánticas adoptadas. Por otro lado, ATC asume la existencia de cierto mecanismo externo con capacidades para afectar al mundo y la forma en que evoluciona. Por lo tanto, ATC permite decidir qué cambios aplicar a fin de provocar la evolución del mundo modelado en la dirección deseada. Sin embargo, nuestra intención no es conducir a usos deshonestos, "escondiendo" argumentos o evidencia de la maquinaria argumentativa. Al contrario, los modelos ATC proveen una herramienta poderosa para diferentes propósitos de investigación. Por ejemplo, consideremos un proceso legal en el cual la evidencia disponible determina un árbol de dialéctica conformando un escenario inesperado. El desarrollo de razonamiento hipotético (o abductivo) mediante la consideración de argumentos que aún no están disponibles puede dar intuiciones acerca de la evidencia faltante. Esto puede ser utilizado como guía para las investigaciones del caso en pos de su exposición ante el tribunal correspondiente. Un ejemplo detallado sobre este punto basado en un proceso penal real de nuestro país, fue presentado por el tesista en (MRF$^+$09).

### 4.5.2. Revisión de Creencias y Bases de Conocimiento

Con respecto a ideas propias de la teoría clásica de revisión de creencias aplicada a teorías no-monótonas, en (BAGM99), los autores estudian las dinámicas de una variante simple de *lógica rebatible* a través de la definición de operadores de expansión, contracción y revisión. Aquí, una *teoría rebatible* contiene hechos, reglas rebatibles y *derrotadores*. Los primeros dos elementos son similares a aquellos de DeLP, mientras que los derrotadores son reglas que, en lugar de ser utilizadas para obtener conclusiones (claims), por el contrario, previenen que sean alcanzadas. El foco del artículo es proveer una cuenta completa de postulados los cuales están cercanamente relacionados a los del modelo AGM. Las intuiciones detrás de cada operador no necesitan consideración especial alguna, y cada una de ellas es formalmente chequeada de acuerdo a los correspondientes postulados. Como revisión de lógicas rebatibles, este trabajo es similar a (MRF$^{+}$08b), el cual propone la revisión de programas DeLP basándose en un enfoque ATC (ver Sección 4.5.3 para mayor detalle). Sin embargo, contrario a los enfoques ATC –incluyendo el modelo dialéctico-global– el trabajo hecho en (BAGM99) requiere restauración de consistencia.

El trabajo presentado en (BDP95) está primariamente orientado al trato de inconsistencias causado por el uso de múltiples fuentes de información. Las KBs son estratificadas, es decir, cada fórmula contenida en la base es asociada con su nivel de certidumbre correspondiente al estrato al cual pertenece. Ellos sugieren que no es necesaria la restauración de consistencia para lograr inferencias sensibles a partir de una KB inconsistente. De forma similar, la inferencia basada en argumentación puede derivar conclusiones (claims) soportadas por razones para creer en ellas, independientemente de la consistencia de la base de conocimientos.

(PG00) estudia la dinámica de un sistema de revisión de creencias considerando relaciones entre creencias en un “enfoque derivacional”, intentando obtener una teoría de revisión de creencias a partir de una teoría epistemológica más concreta. Según ellos, una de las metas de la revisión de creencias es generar una base en la cual cada pieza de información es justificada (por percepción) o garantizada por argumentos conteniendo creencias previamente incluidas en la base. La dificultad es que el conjunto de creencias justificadas puede exhibir todo tipo de incoherencias lógicas debido a que representa un estado intermedio del razonamiento. Por lo tanto, ellos proponen una teoría de revisión de creencias focalizada en la garantía en lugar de la justificación.

(FKIS02) propone un tipo de revisión no-priorizada basada en el uso de explicaciones. La idea es que un agente, antes de incorporar información que es inconsistente con su conocimiento, requiera una explicación que la soporte. El marco utilizado es orientado al razonamiento rebatible. Una de las ideas más interesantes de este trabajo es la generación de condicionales rebatibles de un proceso de revisión. Este enfoque preserva consistencia respecto del conocimiento estricto y provee un mecanismo para calificar en forma dinámica las creencias como estrictas o rebatibles.

(PC06) unió argumentación y revisión de creencias en un mismo marco conceptual, resaltando la importancia del rol jugado por el *layout* de argumentos de Toulmin (Tou58) en el desarrolo de tal integración. Ellos consideran argumentación como "persuasión para creer" y esta restricción es útil para hacer más explícita la conexión con revisión de creencias. Proponen un modelo de dinámica de creencias alternativo al modelo AGM: *revisión de creencias orientada a datos* o *data-oriented belief revision* (DBR). Dos categorías informacionales (datos y creencias) son presentadas en su modelo para identificar la distinción entre piezas de información que son simplemente agrupadas y almacenadas por el agente (*dato*), y piezas de información que el agente considera (hasta un cierto grado) representaciones confiables de estados del mundo (*beliefs*). Cuando una nueva pieza de evidencia es adquirida a través de la percepción o comunicación, afecta directamente a la estructura de datos del agente y sólo indirectamente a sus creencias. La revisión de creencias es a menudo manejada mediante una actualización de la información o bien sobre un hecho o sobre una fuente: el agente recibe una nueva pieza de información, reacomoda su estructura de datos de forma apropiada, y posiblemente cambia sus creencias.

(DSTW08) atacó el problema de revisión de creencias en programación en lógica (no-monótona) bajo semánticas *answer set*: dados dos programas lógicos $P$ y $Q$, el objetivo es determinar un programa $R$ que corresponde a la revisión de $P$ por $Q$, denotado $P * Q$. Propusieron técnicas formales análogas a las propuestas de *distance-based belief revision* o revisión de creencias basada en distancia (Dal88; Sat88). Se investigan dos operadores específicos: operadores de expansión y revisión de programas lógicos basados en la distancia entre los *strong equivalence models* (SE-models) de programas lógicos. En comparación, nuestro enfoque es muy diferente. Primeramente, aquí usamos KBs proposicionales en lugar de programas lógicos, haciendo nuestro enfoque más expresivo y general. Segundo, dado que nuestro interés se basa en la propuesta de una operación de revisión que garantice un argumento externo $\mathcal{R}$ luego de su incorporación, debemos modificar la KB para que el claim del nuevo argumento quede garantizado, determinando así una revisión priorizada. Tercero, y más importante, nuestro trabajo no busca una base consistente como resultado de la revisión, contrastando así con el objetivo común en (DSTW08; Dal88; Sat88).

### 4.5.3. El Modelo Dialéctico-Global y otros Enfoques ATC

Como fuera mencionado anteriormente, el modelo ATC presentado en (MRF$^+$08b) reifica el primer enfoque ATC abstracto, presentado en (RMF$^+$08b), sobre DeLP con el objetivo de revisar programas DeLP. Similar al caso de (DSTW08), utilizamos KBs proposicionales en lugar de programas lógicos rebatibles, lo cual hace al enfoque presentado en esta tesis, más expresivo y general. En contraste con el modelo dialéctico-global, la alteración de árboles de dialéctica en (MRF$^+$08b) es lograda de manera diferente: las incisiones son aplicadas en composición con una *función de selección*, lo cual determina el argumento preciso de cada línea de argumentación al cual la incisión será aplicada. El uso de funciones de selección en tal enfoque permite especificar diferentes criterios de mínimo cambio: remover tan pocas creencias de la KB como sea posible, alterando tan pocas líneas de argumentación del árbol como sea posible, y preservando la estructura del árbol tanto como sea posible mediante la remoción de argumentos ubicados tan "abajo" como sea posible en cada línea, acercándose a las hojas. Por otro lado, el modelo dialéctico-global sigue un punto de vista alternativo pero más general para lograr la alteración de los árboles de dialéctica: nos basamos sobre una función de incisión que es definida "globalmente" al árbol de dialéctica. Por lo tanto, una función de *incisión global* determina un posible conjunto de creencias a remover a fin de alterar efectivamente toda línea que sea necesario de una única vez.

Adicionalmente, el objetivo del modelo presentado en (MRF$^+$08b) difiere del modelo dialéctico-global en que el último no persigue tan extensa variedad de criterios de mínimo cambio, sino que sólo evita la pérdida de creencias que no están relacionadas con la revisión a través del uso del postulatdo de retención de núcleo. El manejo de diferentes criterios de mínimo cambio hace a los modelos ATC más intrincados tanto para su definición como para su posterior axiomatización. En este sentido, es importante resaltar que la noción de minimalidad es usualmente subjetiva: la mayoría de los modelos en la teoría clásica de revisión de creencias no obtienen real minimalidad, sino aproximaciones a ella mediante la especificación de criterios que interpretan el significado del mínimo cambio.

Con respecto a otros enfoques ATC como (RMF$^+$08b) y más recientemente, (MRF$^+$10b), el foco está puesto en el cambio de `AF`s abstractos sin considerar KBs subyacentes. Además, la alteración de árboles de dialéctica en tales trabajos, es realizada en la misma forma que mencionamos anteriormente para el modelo presentado en (MRF$^+$08b).

### 4.5.4. Argumentación y Description Logics

Parte de la bibliografía que relaciona argumentación a DLs ha sido discutida en la Sección 3.6.2. En particular, un marco argumentativo $\mathcal{ALC}$ (sin soporte de consultas conjuntivas, cqs) similar al presentado aquí, fue propuesto en (ZZL09a). Las diferencias con nuestra propuesta, aparecen desde la forma de resolver las dificultades que surgen respecto de la negación de axiomas DL: ellos especifican semánticas especializadas enriquecidas con los conectores lógico-clásicos $\neg$, $\wedge$, y $\vee$, para definir derrotadores. Desde nuestro punto de vista, esto requeriría extender las metodologías ampliamente conocidas y aceptadas para el razonamiento $\mathcal{ALC}$ (algoritmos tableau) mediante el uso de características propias de la lógica clásica, a fin de construir argumentos en la práctica.

## 4.6. Conclusiones y Trabajo a Futuro

En este capítulo presentamos un novedoso modelo de cambio para manejar (1) la dinámica del conocimiento en KBs proposicionales inconsistentes, y (2) la dinámica de argumentos en `AF`s. Este modelo, llamado *dialéctico-global*, se inspira en la literatura clásica de revisión de creencias para caracterizar axiomáticamente el operador de revisión propuesto. Sin embargo, contrario a los enfoques clásicos en revisión de creencias, nuestro modelo no está basado sobre restauración de consistencia. Por ello, el cambio es provocado proveyendo un resultado consistente, no en un sentido estándar (mediante semánticas tarskianas), sino basándonos sobre semánticas argumentativas. Desde la óptica de (1), el operador de revisión recibe un conjunto mínimo $\mathcal{R}$ de creencias proposicionales infiriendo un claim $\alpha$ y lo incluye en la KB (proposicional e inconsistente) de forma que $\alpha$ sea finalmente aceptado por la semántica (con tolerancia a inconsistencias) adoptada. Por otro lado, para (2), la revisión incluye el argumento $\mathcal{R}$ al `AF` asegurando su garantía y por lo tanto, aceptando $\alpha$.

El capítulo se centra sobre la teoría de cambio argumentativa o Argument Theory Change (ATC), la cual estudia ciertos aspectos de la teoría clásica de revisión de creencias a fin de adaptarlos a la argumentación. Sin embargo, el modelo dialéctico-global provee un nuevo enfoque a ATC haciendo concreto el lenguaje para argumentos dentro de la lógica clásica proposicional.

Según nuestro conocimiento, el modelo dialéctico-global es el único modelo argumentativo de cambio que ha sido completamente axiomatizado (de acuerdo a la teoría de revisión de creencias). Primeramente, un conjunto básico de postulados fue adaptado de revisión de creencias a argumentación, y luego, la racionalidad del modelo propuesto es mostrada a través del correspondiente teorema de representación. Su racionalización completa ha sido inspirada por los trabajos de Hansson sobre contracciones kernels introducidas en (Han94). Allí, los conjuntos kernel –conjuntos mínimos infiriendo la fórmula a contraer– son "cortados" en forma similar a la alteración efectiva de las *líneas de ataque*, tal como es hecho en ATC.

Las líneas de ataque son reconocidas como líneas de argumentación que son de alguna forma responsabilizadas por el estado no-garantizante del árbol de dialéctica, los cuales son adoptados como semántica argumentativa para decidir la aceptación (garantía) del argumento raíz. La alteración de la composición de las líneas de ataque en un árbol de dialéctica determina un nuevo árbol libre de líneas de ataque, lo cual asegura que su argumento raíz será garantizado. Adicionalmente, la *alteración* de las líneas de argumentación es desarrollada mediante la remoción de algún argumento incluido en ella. Sin embargo, dado que el `AF` es construido a partir de una KB proposicional subyacente, la *remoción de argumentos* es finalmente alcanzada mediante el borrado de creencias de la KB. A este fin, el cambio mínimo es asegurado mediante la eliminación de creencias que únicamente están relacionadas a la revisión, *i.e.*, creencias que son parte de argumentos ubicados en líneas que determinan el estado no-garantizante del árbol de dialéctica enraizado en $\mathcal{R}$.

Relaciones de preferencia entre argumentos son usualmente utilizadas para decidir si un contra-argumento finalmente prevalece para determinar un ataque. Por simplicidad nos hemos abstraído de tales relaciones, y dejamos los ataques para ser determinados sólo mediante contradicción lógica. Sin embargo, en un proceso de revisión, las relaciones de preferencia son necesarias para decidir si un argumento puede ser eliminado en beneficio de la búsqueda de garantía del nuevo argumento a incluir. Por ejemplo, en la promulgación de leyes, sería inadmisible la consideración de remociones de artículos correspondientes a leyes de mayor jerarquía que la nueva que se pretende promulgar: la Constitución Nacional nunca debería ser afectada a menos que la nueva ley deba determinar su reforma. Tales consideraciones son parte del trabajo en curso.

El modelo dialéctico-global puede ser fácilmente extendido a lógica de primer orden (FOL). De hecho, sólo la maquinaria argumentativa necesita ser adaptada para manejar KBs de primer orden mientras que el modelo de cambio permanecerá inalterado. Sin embargo, una maquinaria argumentativa de primer orden es ciertamente más compleja que la maquinaria simple adoptada aquí sobre lógica proposicional. De esa forma, además, logramos mantener el foco central del capítulo en la dinámica del conocimiento y de los argumentos. Para mayor detalle sobre argumentación de primer orden el lector es referido a (BH05; BH08; MRFS09).

Luego, en la Sección 4.4, se propuso una maquinaria argumentativa para DLs en general. Esta propuesta constituye una nueva metodología de razonamiento ontológico la cual provee técnicas de argumentación para el razonamiento sobre ontologías inconsistentes/incoherentes, y se comporta como un razonador ontológico clásico cuando considera ontologías consistentes (Teorema 4.4.7).

Mientras las metodologías clásicas para el razonamiento sobre DLs sean reutilizadas para la construcción de maquinarias argumentativas DL, la argumentación proveerá una alternativa útil para el razonamiento sobre ontologías inconsistentes –la complejidad del razonador argumentativo dependerá de la correspondiente al razonador DL clásico subyacente. Sin embargo, cierto incremento en la complejidad debe ser esperado al momento de construir grandes árboles de dialéctica. Esto establece una relación con el "nivel de inconsistencia" (número de axiomas contradictorios) de la ontología consultada.

El análisis de complejidad para decidir si una consulta $\alpha$ es aceptada por la maquinaria argumentativa DL requiere construir y marcar el árbol dialéctico enraizado en un argumento $\alpha$-soporte. En unfolded $\mathcal{ALC}$, este problema se acerca al del cálculo de la estructura MIPS a partir de todas las MUPS. Más aún, dado que el número de derrotadores para un árbol dialéctico completo puede crecer exponencialmente en el número de axiomas de la TBox, creemos que la construcción del árbol en unfolded $\mathcal{ALC}$ estaría al menos dentro de la clase de complejidad de EXPTIME (tal cual es el caso de satisfabilidad en $\mathcal{ALC}$). Un análisis profundo del tema constituye parte del trabajo corriente en esta dirección. Trabajo a futuro incluye además la propuesta del algoritmo completo para $\mathcal{ALC}$ y más importante, para familias DL eficientes como DL-Lite y $\mathcal{EL}$.

# Capítulo 5

# Trabajo Futuro y Conclusiones Generales

Como trabajo futuro, aunque en términos más generales y "abarcativos" que los propuestos en los capítulos 3 y 4, en esta sección evaluamos el impacto y aplicabilidad de las teorías propuestas en esta tesis sobre entornos reales. Para ello, introduciremos de forma somera una de sus proyecciones posibles al servicio de un entorno de aplicación específico, como es el derecho. Tales intuiciones constituyen parte del proyecto post-doctoral del tesista. Finalmente, en la Sección 5.2, esbozaremos algunas ideas a modo de conclusión de esta tesis, en la que consideraremos las contribuciones propuestas como herramientas para el manejo de la evolución ontológica en general con tolerancia a inconsistencias.

## 5.1. Proyección de las Teorías Propuestas en Entornos Legales

Primeramente definiremos el problema en forma intuitiva, explicitando los objetivos buscados. Luego, fundamentaremos algunas hipótesis de trabajo basándonos en las contribuciones de esta tesis.

### 5.1.1. Intuiciones para la Introducción del Problema

Consideremos el desarrollo de sistemas de apoyo al proceso judicial (BCS03) mediante la implementación de un Sistema Judicial Argumentativo o Judicial Argumentation System (JAS). Asumiremos al JAS como un prototipo experimental que será utilizado para la validación de las propuestas teóricas que surjan como parte del objetivo específico: el desarrollo de una teoría de argumentación abstracta que brinde servicios para el manejo de la dinámica, incluyendo capacidades de adaptación a diferentes contextos, a fin de lograr su aplicación en entornos legales reducidos. El enfoque práctico de dicha teoría permitirá la implementación del prototipo JAS, con el cual se espera lograr un análisis semi-automatizado de la dinámica y el razonamiento sobre casos específicos de estudio en entornos legales. Estos puntos serán detallados a continuación.

Como hemos visto, la dinámica del conocimiento, tratada en el área de revisión de creencias, hace referencia al estudio de metodologías de cambio para provocar la evolución del conocimiento de acuerdo a un conjunto de principios básicos de cambio. Si se utiliza la argumentación para razonar sobre bases inconsistentes, el manejo de la dinámica del conocimiento puede ser trasladado al manejo de la dinámica de argumentos. La dinámica de argumentos es provocada mediante el uso de modelos de cambio argumentativos, con diferentes objetivos, como por ejemplo, el de alterar el marco argumentativo de forma tal que admita, y finalmente garantice, un dado nuevo argumento. Para ello, es necesario eliminar aquellos argumentos que interfieren con la garantía del nuevo argumento según se infiera de la semántica argumentativa adoptada.

El contexto en el cual un marco argumentativo es aplicado determina el conjunto de argumentos a ser considerados (o admitidos) por el razonador. Para ello usamos parámetros de evaluación interna de argumentos. Por ejemplo, factores de confianza sobre la evidencia que conforma un argumento (llamado "nivel de prueba"), o la relevancia del propio argumento (fuerza) medido a partir del conocimiento que lo forma. A partir de información inherente al contexto, como un nivel mínimo de prueba aceptable, se puede determinar un subconjunto de argumentos admisibles en el contexto en cuestión, y los ataques relevantes. Esto es referido como sensibilidad al contexto. La definición de una teoría argumentativa dinámica requiere, además de modelos de cambio, capacidades de adaptación a diferentes contextos. Esto hace referencia a la habilidad de obtener las garantías apropiadas al contexto particular en el cual un mismo marco argumentativo es aplicado. Claramente, una garantía en ciertos contextos podría no ser tal en otros.

Las teorías argumentativas dinámicas son necesarias en áreas de las ciencias en donde el manejo de inconsistencias y la evolución del conocimiento conforman puntos críticos. El razonamiento legal, dentro de la Inteligencia Artificial, es una de ellas. Por ejemplo, la promulgación de una ley (como hemos mencionado en la Sección 1) provoca la evolución del sistema legal de forma que su constitucionalidad sea indiscutible más allá de la existencia de conflictos entre artículos de leyes previamente promulgadas (e incluso estándares sociales). Para ello es necesario identificar un conjunto de artículos a derogar o reformar, como parte del proceso de promulgación.

Imaginemos una base de conocimiento representando una porción del sistema legal: conteniendo subconjuntos de la constitución nacional, tratados de ley internacional, u otros principios políticos fundamentales tales como los códigos civil, penal, decretos y resoluciones, entre otros. Asumiendo que tal base será interpretada mediante la argumentación, la promulgación de leyes sería manejada mediante un modelo de cambio argumentativo, el cual incluiría un nuevo argumento (conteniendo la nueva ley) asegurando su garantía (y consecuentemente su constitucionalidad). Observe que aquellos argumentos que el modelo de cambio "elimine" representarían artículos y/o principios de otras leyes, idealmente de menor jerarquía. Para ello será necesario representar el orden jerárquico de las leyes utilizadas (denominada "pirámide jurídica") para determinar la fuerza de un argumento y, consecuentemente, las relaciones de preferencia y ataque entre argumentos, y el conjunto de argumentos a "eliminar". En pos de la garantía (constitucionalidad) del nuevo argumento (de la nueva ley), los modelos argumentativos podrán eliminar argumentos (derogación) o alterar su composición interna (reforma/enmienda). Ambas alternativas buscan la garantía del nuevo argumento eliminando ataques específicos.

Respecto del contexto, la promulgación de leyes es un caso de dinámica propia del derecho constitucional. La dinámica a nivel inter-contextual implicaría servicios para el cambio entre contextos. Desde la óptica legal, esto ocurre cuando el desarrollo de un caso jurídico particular determina la apertura de un nuevo caso de carátula alternativa, requiriendo la intervención de un fuero de diferente clase. Supongamos un caso típico de divorcio en el que las partes se disputan la tenencia de los hijos. Durante el proceso, se presenta evidencia acusando a una de las partes de abuso infantil. Consecuentemente, el caso originalmente propio de un fuero de familia, debe ser puesto en suspenso para abrir un nuevo caso de orden penal. Una de las diferencias entre tales fueros, es que la evidencia presentada en un fuero penal es sometida a un nivel de prueba mayor. Esto significa que un argumento válido para el fuero de familia podría ser inadmisible en el fuero penal. Situaciones similares occurren con los ataques entre argumentos. Otro tipo de cambio de contexto puede ser entendido como un cambio de instancia de un caso legal en particular, haciendo referencia a las distintas instancias de apelación de nuestro sistema legal. Los inconvenientes suelen surgir a partir de la admisibilidad de argumentos al considerar la jurisprudencia (sentencias de casos jurídicos similares).

El JAS proveería una herramienta de estudio y evaluación de sistemas legales, teniendo impacto no solo a nivel académico, dentro de la Inteligencia Artificial y del Derecho, sino también a nivel público, mediante el uso de internet. La propuesta de diferentes modelos de cambio argumentativos facilitará el análisis de las diferentes formas de evolución del sistema legal. La promulgación de la, tan largamente discutida, nueva ley de medios, constituye un claro ejemplo en nuestro país en el que el JAS colaboraría incluso a nivel político para lograr conclusiones más ágilmente. La publicación del JAS en la web proveería al usuario de cierto contexto para participar activamente de la vida política del país, lo cual es estudiado en el área de *e-goverment*.

### 5.1.2. Fundamentos Teóricos

Las teorías propuestas en esta tesis constituyen la base de la formalización teórica para las intuiciones detalladas anteriormente. A continuación describiremos como pueden ser utilizados el `GenAF` (Capítulo 3) y el modelo dialéctico-global (Capítulo 4) en pos de la especificación de la teoría argumentativa dinámica para sistemas legales. Para ello, las hipótesis son organizadas en tres ópticas diferentes: el marco argumentativo, el manejo de la dinámica, y la representación del conocimiento.

#### Marco Argumentativo

Como hemos mencionado, el *Dynamic Abstract Argumentation Framework* (DAF) (RMGS10; Rot10) fue diseñado para facilitar el manejo de la dinámica de argumentos, basado en el `AF` clásico propuesto por Dung (Dun95). Como característica principal, un DAF reconoce un subconjunto activo del universo de argumentos, como el único a ser considerado por la semántica argumentativa. Un argumento puede resultar inactivo como resultado de la remoción de información durante un proceso de revisión.

Sin embargo, el proceso de revisión puede estar aplicado en forma local al contexto. Esto significa que un argumento puede resultar activo o inactivo dependiendo del tipo de consulta. Otra de las virtudes del uso de argumentos activos es que la información nunca se pierde de forma definitiva. Esto es interesante si consideramos el paso inverso, es decir, la activación de argumentos en pos de un proceso de revisión (MRF$^+$09; MRF$^+$10b), explotando así la característica no-monótona de las semánticas argumentativas.

La distinción entre argumentos activos e inactivos brinda la estructura necesaria para manejar la admisibilidad de argumentos en contextos específicos. Se necesitarán especificar condiciones de admisibilidad propias del razonamiento legal, como ser diferentes tipos de parámetros externos como variables de tiempo. Por ejemplo, en un caso legal, existe un período apelatorio previo al inicio del proceso judicial. Un argumento, aunque válido en términos de inferencia, puede ser inactivo en el período post-apelatorio dada su imposibilidad de ser utilizado para el razonamiento. Sin embargo, en determinadas situaciones excepcionales, un juez puede conceder una extensión al período apelatorio, lo cual requeriría activar el argumento en cuestión.

Por otro lado, se necesitará variar la granularidad en la especificación de argumentos dependiendo del tipo de ley que se intente representar. Es decir, los argumentos pueden ser configurados para representar leyes completas, artículos que las conforman, o incluso incisos de artículos. Esto dará la posibilidad de, en el caso de la promulgación de una nueva ley, identificar al nivel más bajo posible los puntos conflictivos con la porción del sistema legal en evaluación. Una granularidad más pequeña permitirá analizar la desactivación de incisos conflictivos al precio de una mayor complejidad de representación. Por el contrario, una representación de mayor simpleza puede ser lograda mediante una granularidad mayor, permitiendo reconocer sólo leyes en contraposición con la nueva ley a promulgar. Por otro lado, el caso legal particular a evaluar requerirá los elementos necesarios para variar el lenguaje de representación específico basado en la expresividad y el tipo de granularidad requeridos. Tales condiciones no hacen más que describir las virtudes propias del `GenAF`.

En definitiva, se precisará una Teoría Argumentativa Judicial o *Judicial Argumentation Theory* (JAT), la cual considerará un `GenAF`, un subconjunto activo del conjunto $\mathbf{A}$, un conjunto universal de contextos jurídicos (cada uno con su nivel mínimo de prueba), un conjunto de cortes judiciales estratificado de acuerdo a su jerarquía jurídica, parámetros de instancia identificando contexto judicial e instancia apelatoria actuales, y una relación de preferencia entre pares conflictivos del `GenAF`. Por su parte, el `GenAF` necesitará organizar la información del conjunto **A**, estratificando las estructuras de acuerdo a la pirámide jurídica (por ejemplo, estructuras correspondientes a la Constitución Nacional estarán por encima de las relativas a una constitución provincial). Cada argumento del `GenAF` contendrá información sobre su nivel de prueba. En una JAT, tanto el conjunto de argumentos activos como la relación de preferencia, variarán dinámicamente de acuerdo a los parámetros de instancia. Los ataques deberán ser adjudicados a partir de la relación de preferencia de la JAT en conjunción con la relación de conflicto del `GenAF`.

**Dinámica**

Con respecto a los modelos de cambio argumentativo, nos basaremos en trabajos recientes como (BKvdT09; CdSCLS10) y mayormente en *Argument Theory Change* (ATC) (RMF$^+$08b; MRF$^+$10b) (propuesta por el tesista en principio para argumentación abstracta, sobre el DAF). Tales propuestas bibliográficas han sido relacionadas a las contribuciones de esta tesis en las secciones 3.6 y 4.5.

Se definirán nuevos modelos de cambio argumentativo *à la* ATC sobre el razonamiento legal para manejar la evolución de una JAT. Para ello, se utilizará el modelo dialéctico-global como punto de partida para ser aplicado al JAT, considerando su conjunto de parámetros específicos adicionales, como han sido mencionados anteriormente. (ATC fue propuesto por el tesista en (MRF$^+$09; MWF10) para manejar la dinámica de casos legales específicos –por ejemplo, en (MWF10) se analizaron algunos puntos conflictivos de la nueva ley de medios.) Además, propondremos principios para regular la dinámica del razonamiento de una JAT manteniendo consistencia en relación a los precedentes (jurisprudencia). La aplicación de lógicas dinámicas (vB07) y dinámica de lógicas deónticas (Han06), para la adjudicación de ataques, podría proveer soluciones a tal fin. La dinámica de ataques de argumentos aparecen cuando los jueces consideran cambios a nivel social, previos a la evolución normativa. Ejemplos específicos ocurren al considerar regulaciones sobre el uso de cannabis para consumo personal o la aceptación del matrimonio homosexual. Desde la óptica teórica de una JAT esto implica dejar de considerar ciertos argumentos como conflictivos, lo cual implica la definición de operaciones de revisión sobre la relación de ataque.

**Representación del Conocimiento**

El uso de ontologías como bases de conocimiento para representar normas y la evidencia sobre el "estado actual de las cosas", es necesario para la implementación del JAS con acceso público a través de internet. Por otro lado, las ontologías y su especificación mediante diferentes tipos de DLs, brindan una forma de representación del conocimiento para lograr estructuración altamente compleja. Esto es vital para representar los fragmentos legales a analizar. Además, será necesario analizar en qué medida relegar cuestiones de expresividad en favor de eficiencia, para elegir una lógica apropiada de representación de normas. En este sentido, serán consideradas familias de DLs como DL-Lite (CGL$^{+}$07) y $\mathcal{EL}$ (Baa03), introducidas en la Sección 2.4.5. Es importante mencionar la extrema relevancia para la especificación del JAT e implementación del JAS, del estudio de las teorías propuestas en esta tesis –basadas en argumentación y dinámica– sobre DLs como forma de representación del conocimiento.

## 5.2. Comentarios y Observaciones Finales

En esta tesis hemos adoptado la argumentación como alternativa teórica para el razonamiento sobre KBs inconsistentes. La especificación de los llamados `GenAF`s, introduce una novedosa familia de marcos argumentativos generalizados. Las componentes internas de un `GenAF` son provistas de cierta estructura, manteniendo abstracta su especificación respecto de un lenguaje de representación concreto. Sin embargo, su caracterización lógica permite que un `GenAF` pueda ser concretizado sobre diferentes lenguajes de representación que correspondan a algún fragmento de primer-orden, como ocurre con las DLs en general.

En términos teóricos, las semánticas del `GenAF` son propuestas sobre las semánticas estándares de Dung. Esto permite un estudio profundo de las particularidades propias de los marcos abstractos en general. Por ejemplo, a partir del estudio de las extensiones estables, logramos obtener un subconjunto maximal consistente de la KB subyacente (ver Sección 3.4). Esta operación es caracterizada a través de los postulados y representación axiomática de las consolidaciones kernel de Hansson, dentro de la teoría de revisión de creencias.

Sin embargo, una de las propuestas de esta tesis señala al análisis de árboles de dialéctica como semántica argumentativa de enfoque meramente práctico. Esto es así debido a la naturaleza de su construcción, es decir, un árbol de dialéctica necesitará el uso de derrotadores que surgen a partir de la construcción de un argumento soporte de una consulta dada. Luego, sólo tales argumentos son considerados (en contraste con las semánticas basadas sobre el grafo completo), en principio para decidir si su argumento raíz es garantizado, lo cual implica la verificación de la consulta en cuestión. Sin embargo, el análisis y posterior alteración de un árbol de dialéctica permite, de acuerdo a la definición del modelo dialéctico-global, proveer una metodología de cambio argumentativo en pos de la garantía del argumento raíz.

Consecuentemente, asumiendo la construcción de un marco argumentativo a partir de la KB subyacente, el problema de la dinámica del conocimiento sobre tal KB es manejado a través de la dinámica de argumentos sobre su marco argumentativo asociado. El cambio y evolución de KBs dedicadas a la representación de conocimiento potencialmente inconsistente, requiere necesariamente de métodos no-clásicos de revisión de creencias en los cuales la restauración de consistencia debe ser evitada. Este es el caso del modelo dialéctico-global, el cual es aplicado directamente sobre árboles de dialéctica, requiriendo así la readaptación de los postulados básicos de revisión de creencias a la teoría de argumentación.

Si bien este nuevo modelo de cambio argumentativo es aplicado sobre un marco proposicional simplificado, su definición sobre `GenAF`s puede ser realizada en forma cuasi-directa, siendo que las incisiones del modelo dialéctico-global incluyen sentencias de la KB, o lo que sería equivalente en un `GenAF`, argumentos. A su vez, el árbol de dialéctica al cual se aplica la incisión, sería construido a partir de estructuras argumentales del `GenAF`. Esto es así, dada su condición de estructuras auto-conclusivas (al igual que los argumentos del marco proposicional) y que sus componentes atómicas son argumentos del `GenAF` o equivalentemente, sentencias de la KB subyacente (emulando nuevamente al marco proposicional).

Por otro lado, la alta versatilidad de la que goza el `GenAF`, le permite no sólo su adaptación a distintos lenguajes de representación, sino también trabajar con bases normalizadas respondiendo a las especificaciones de granularidad deseadas. Esto es, un `GenAF` permite configurar la forma de la mínima porción del conocimiento (indivisible) de la KB. Al aplicar el modelo dialéctico-global al `GenAF`, obtenemos la posibilidad de asegurar que las porciones de conocimiento a eliminar en pos de la revisión de un nuevo conocimiento, corresponderán a un formato específico. Por ejemplo, un formato de alta granularidad permite asegurar que se perderá la menor porción de información de la KB subyacente, no sólo en términos sintácticos, sino también en términos de conclusiones inferidas por la KB. En este caso, la propuesta colaboraría con la verificación de *preservación de consecuencias*, dimensión de compatibilidad introducida en la Sección 2.3.1.

Para la verificación de la dimensión de compatibilidad restante, *i.e.*, *preservación de dato de instancia*, sería suficiente adoptar como política que todo argumento evidencial (contenido en $\mathbf{E}$) del `GenAF` tenga mayor importancia (relevancia) que el resto de los argumentos contenidos en $\mathbf{A}$. De esta forma, el uso del criterio de comparación de los argumentos del `GenAF` puede ser utilizado como método de decisión entre diferentes alternativas para la construcción de la función de incisión del modelo dialéctico-global. Observe, sin embargo, que en forma indirecta en tal caso, podríamos estar descuidando la dimensión de preservación de consecuencias.

Por otro lado, la normalización del conocimiento de la KB a considerar tiene influencia sobre la eficiencia para la construcción de árboles de dialéctica. Esto es así, dado que uno de los objetivos usuales en la definición de formas normales es lograr la disminución en la complejidad computacional del razonamiento. Tal virtud, se asume compensaría el costo inicial de preprocesamiento para la normalización de la KB en cuestión. En este sentido, el uso de la *forma normal prime implicate* para $\mathcal{ALC}$ de Bienvenu, permite la definición de una maquinaria `GenAF` específica para el lenguaje $\mathcal{ALC}$ con ciertas virtudes.

Una maquinaria $\mathcal{ALC}$-`GenAF` donde su forma normal pre-argumental está basada en la forma normal *prime implicate* $\mathcal{ALC}$, utilizará fórmulas *prime implicate* para la construcción de argumentos y, por lo tanto, para sus estructuras argumentales. Esto facilitaría la contrucción del árbol de dialéctica según la metodología propuesta en la Sección 4.4.1. Observe que el algoritmo $\mathcal{ALC}$-tableau que construye la estructura MUPS, construye funciones de minimización que finalmente determinarán *prime implicates*. Dado que los axiomas manejados estarán formados por *prime implicates*, la construcción de las funciones de minimización sería trivial. Luego, a partir de este algoritmo logramos construir los derrotadores de cada nodo del árbol. Esta construcción presupone un tiempo lineal del orden de la suma del tamaño de los axiomas que componen el argumento para el cual buscamos sus derrotadores. Claro que en estos casos, se deberá aceptar un costo inicial de preprocesamiento de orden doble exponencial sobre el tamaño de los conceptos a normalizar. El estudio detallado de estas cuestiones queda propuesto como parte del trabajo futuro.

Finalmente, considerando los principios de cambio propuestos en la Sección 2.3.2, observamos que el modelo dialéctico-global aplicado sobre una maquinaria DL-`GenAF`, se comporta de la siguiente forma. Respecto del principio de:

- *representación adecuada*, la ontología evolucionada respeta la representación original en términos del lenguaje de descripción y, por lo tanto, mantiene su expresividad. Por otro lado, la normalización de la DL subyacente puede alterar la concepción de los axiomas respecto de su conceptualización original. Sin embargo, esto tiene influencia sólo en el aspecto sintáctico.
- *irrelevancia de sintaxis*, al basarnos en métodos semánticos, *i.e.*, mediante interpretaciones tarskianas, nos alejamos de dependencias directas de la sintaxis. Sin embargo, la sintaxis no es completamente irrelevante, dada la necesidad de trabajar la estructura de los axiomas mediante el uso de formas normales.
- *mantenimiento de consistencia*, observamos que no es verificado. Dado que nuestro modelo tiene como objetivo la tolerancia a inconsistencias, la inclusión de un nuevo conocimiento consistente a una ontología consistente, no necesariamente tiene como resultado una ontología evolucionada consistente, sino que la ontología evolucionada garantizará el nuevo conocimiento por sobre la aparición de inconsistencias. Consecuentemente, en un sentido más apropiado para el tipo de evolución que se modela en esta tesis, proponemos el principio de *mantenimiento de inconsistencia*, que se adiciona a los seis principios dados en la Sección 2.3.2:

  7. *Mantenimiento de Inconsistencia:* La incorporación de un nuevo conocimiento consistente a una ontología inconsistente no provocará restauración de consistencia, sino que preservará las fuentes de conflicto tanto como sea posible.

- *mantenimiento de inconsistencia*, la ontología evolucionada verifica el principio, dado que este describe la intuición principal del modelo dialéctico-global.
- *primacía de la nueva información*, dado que el modelo dialéctico-global es un modelo priorizado, *i.e.*, la nueva información recibe máxima prioridad por encima de la información original, si el nuevo conocimiento es consistente, podrá conformar un argumento que concluirá garantizado por la ontología evolucionada.
- *cambio mínimo*, observamos su verificación en el modelo dialéctico-global a través del postulado de retención de núcleo. Como hemos discutido anteriormente, los criterios de mínimo cambio pueden variar respecto de la interpretación que se le

asigne. Diferentes criterios de mínimo cambio para los enfoques ATC en general, han sido propuestos por el tesista en (MRF$^+$08b). Por simplicidad, en esta tesis se abstrae la propuesta de diferentes criterios de mínimo cambio.

- *imparcialidad*, como se ha mencionado en la Sección 2.3, la participación del usuario es primordial para su verificación. Prototipos futuros del modelo dialéctico-global deberán ser implementados teniendo en cuenta este principio. De tal forma, el sistema deberá proveer al ingeniero ontológico, la alternativa de aprobar una determinada construcción de una función de incisión.

En síntesis, en esta tesis las DLs son adoptadas como los lenguajes de representación de mayor interés. De esta forma, se proveen marcos argumentativos especialmente definidos para DLs, lo cual brinda la posibilidad de razonar sobre ontologías inconsistentes.

Las maquinarias argumentativas introducidas son semánticamente determinadas – interpretaciones tarskianas. Esto provee como ventaja la futura propuesta de implementaciones a ser construidas sobre motores estándares de razonamiento DL. Por otro lado, las maquinarias argumentativas propuestas pueden ser desarrolladas en forma directa sobre la ontología. Esto se debe a que los axiomas DL son interpretados como simples argumentos en el `GenAF`. Como hemos mencionado, asumiendo tal consideración, la metodología argumentativa propuesta puede ser utilizada simplemente como una herramienta teórica para el razonamiento sobre ontologías inconsistentes, sin necesidad de construir el marco argumentativo asociado.

Asimismo, el modelo dialéctico-global provee un modelo de cambio ontológico sin restauración de consistencia. Este punto, como hemos visto en la Sección 5.1, es de suma importancia para determinadas áreas de las ciencias como el Derecho, en las que el trato de las inconsistencias es vital tanto para el razonamiento como para la evolución del conocimiento.

# Apéndice A

# Pruebas y Nociones Complementarias

En este apéndice se proveen las pruebas correspondientes a propiedades establecidas a lo largo de este trabajo de tesis. Ocasionalmente el lector podrá encontrar definiciones, proposiciones, lemas, y teoremas adicionales (no introducidos hasta aquí) numerados según el patrón A.XX. Su propósito es el de complementar las pruebas de propiedades provistas en los capítulos anteriores. Sin embargo, tales propiedades y nociones adicionales no son necesarias para una correcta comprensión de las teorías presentadas en la tesis.

## A.1. Respecto del `GenAF` (Cap. 3)

**Lema 3.2.10** *Dado un* `GenAF` $\langle \mathbf{A}, \mathbf{R_A} \rangle \in \mathbb{G}$*, una estructura* $\mathbb{S} \subseteq \mathbf{A}$ *para* $\alpha \in \mathcal{L}_{\mathtt{cl}}$*, y su árbol de soporte* $\mathcal{T} \in \mathfrak{Trees}_{\mathbb{S}}(\alpha)$*; si* $\mathbb{S}$ *es argumental entonces* $\mathfrak{leaf}(\lambda) \subseteq (\mathbf{E} \cup \mathbf{P})$*, para toda* $\lambda \in \mathcal{T}$*.*

<u>*Demostración:*</u> Asumiendo la estructura $\mathbb{S}$ como argumental, de la Definición 3.2.8, sabemos que $\mathfrak{pr}(\mathbb{S}) = \emptyset$, y por lo tanto que no existe premisa $\rho \in \mathfrak{prset}(\mathbb{S})$ libre con respecto a $\mathbb{S}$. Luego, toda premisa $\rho \in \mathfrak{prset}(\mathbb{S})$ es cerrada dentro de $\mathbb{S}$, lo que significa que existe siempre alguna $\lambda \in \mathcal{T}$ tal que $\overleftarrow{\lambda}[i] = \rho$ $(0 < i < |\lambda|)$ (ver Definición 3.2.7). Dado que las cadenas de soporte son acíclicas (ver Definición 3.2.6), para todo $\lambda \in \mathcal{T}$, sabemos que $\mathfrak{prset}(\mathfrak{leaf}(\lambda)) = \emptyset$. Es claro que para todo argumento $\mathcal{B} \in \mathfrak{leaf}(\lambda)$ se verifica $\mathfrak{pr}(\mathcal{B}) = \emptyset$.

Finalmente, de la Definición 3.1.7, todo $\mathcal{B}$ es o bien un argumento evidencial o primitivo, y por lo tanto, $\mathfrak{leaf}(\lambda) \subseteq (\mathbf{E} \cup \mathbf{P})$, para toda $\lambda \in \mathcal{T}$. $\square$

**Lema 3.2.11** *Dado un* `GenAF` $\langle \mathbf{A}, \mathbf{R}_{\mathtt{A}} \rangle \in \mathbb{G}$, *y una fórmula* $\alpha \in \mathcal{L}_{\mathtt{cl}}$, *existe una estructura argumental* $\mathbb{S} \subseteq \mathbf{A}$ *para* $\alpha$ *sssi* $\alpha$ *es inferible.*

*Demostración:* Asumiendo $\mathbb{S}$ como argumental, de la Definición 3.2.8 tenemos que $\mathfrak{pr}(\mathbb{S}) = \emptyset$. Esto significa que no existe premisa $\rho \in \mathfrak{prset}(\mathbb{S})$ libre con respecto a $\mathbb{S}$. Finalmente, de la Definición 3.2.5, $\alpha$ es inferible.

La reversa de esta prueba puede ser ontenida de forma similar. $\square$

**Teorema 3.2.12** *Dado un* `GenAF` $\langle \mathbf{A}, \mathbf{R}_{\mathtt{A}} \rangle \in \mathbb{G}$, *una fórmula* $\varphi(\bar{x}) \in \mathcal{L}_{\mathtt{cl}}$, *y una estructura* $\mathbb{S} \subseteq \mathbf{A}$ *para* $\varphi(\bar{x})$*; si* $\mathbb{S}$ *es argumental y* $\mathfrak{leaf}(\lambda) \subseteq \mathbf{E}$ *entonces* $\mathfrak{cl}(\mathbb{S}) = \varphi(\bar{a})$ *y* $\varphi(\bar{a}), v \models \varphi(\bar{x})$, *donde* $v$ *mapea cada* $x_i \in \{\bar{x}\}$ *a* $a_i \in \{\bar{a}\}$.

*Demostración:* Asumiendo $\mathbb{S}$ como argumental, de la Definición 3.2.8 tenemos que $\mathfrak{pr}(\mathbb{S}) = \emptyset$. Esto significa que el árbol de soporte de $\mathbb{S}$ tiene para todas sus hojas, coaliciones de argumentos con un conjunto vacío de premisas. Luego, los componentes de tales coaliciones podrían ser o bien argumentos evidenciales o primitivos $\mathcal{B}_1, \ldots, \mathcal{B}_m$. Sin embargo, dado que $\mathfrak{leaf}(\lambda) \subseteq \mathbf{E}$, sabemos que ellos son sólo evidenciales, y por lo tanto de la Definición 3.1.7, sabemos que sus claims $\mathfrak{cl}(\mathcal{B}_1), \ldots, \mathfrak{cl}(\mathcal{B}_m)$ corresponden al lenguaje $\mathcal{L}_{\mathtt{A}}$, el cual acepta sólo conocimiento asercional, *i.e.*, fórmulas ground. Luego, la secuencia $\mathfrak{cl}(\mathcal{B}_1), \ldots, \mathfrak{cl}(\mathcal{B}_m)$ no tiene variables libres.

De la Definición 3.2.8 y Definición 3.2.7 el árbol de soporte $\mathcal{T} \in \mathfrak{Trees}_{\mathbf{A}}(\alpha)$ dentro de $\mathbb{S}$ (*i.e.*, $\mathbb{S} = \mathcal{T}^*$) tiene como raíz una coalición-claiming $\widehat{\mathcal{C}}$ para $\varphi(\bar{x})$. Sin embargo, dado que $\mathbb{S}$ es argumental y que toda premisa es cerrada por un argumento evidencial $\mathcal{B}_j$ $(1 \leq j \leq m)$, sabemos que cada premisa de $\widehat{\mathcal{C}}$ es también cerrada y por lo tanto para cada $\rho \in \mathfrak{prset}(\widehat{\mathcal{C}})$ asumiendo $\rho = \phi(\bar{x}')$, se verifica que $\bar{x}'$ es mapeada a $\bar{a}'$, donde $\{\bar{a}'\} \subseteq \bigcup_{j=1}^{m}\{\bar{a}_j\}$, y cada $\bar{a}_j$ es el vector de individuos asociados al claim de $\mathcal{B}_j$ (*i.e.*, $\mathfrak{cl}(\mathcal{B}_j) = \phi'(\bar{a}_j)$). Por ello, cada variable de $\mathfrak{prset}(\widehat{\mathcal{C}})$ es instanciada en el contexto de $\mathcal{T}$ con algún individuo de $\bigcup_{j=1}^{m}\{\bar{a}_j\}$. Luego, una asignación $v$ aparece donde cada $x_i \in \{\bar{x}\}$ es mapeada a $a_i \in \{\bar{a}\}$, y $\{\bar{a}\} \subseteq \bigcup_{j=1}^{m}\{\bar{a}_j\}$. Finalmente, $\varphi(\bar{a}), v \models \varphi(\bar{x})$ se verifica y por lo tanto $\mathfrak{cl}(\mathbb{S}) = \varphi(\bar{a})$. $\square$

**Proposición 3.2.14** *Dado un* `GenAF` $\langle \mathbf{A}, \mathbf{R}_{\mathtt{A}} \rangle \in \mathbb{G}$, *una fórmula* $\alpha \in \mathcal{L}_{\mathtt{cl}}$, *y dos estructuras* $\mathbb{S} \subseteq \mathbf{A}$ *para* $\alpha$ *y* $\mathbb{S}' \subseteq \mathbf{A}$ *para* $\alpha$, *si* $\mathbb{S}' \lhd \mathbb{S}$ *entonces* $\mathfrak{pr}(\mathbb{S}) \neq \mathfrak{pr}(\mathbb{S}')$.

<u>*Demostración:*</u> Asumiendo $\mathbb{S}' \lhd \mathbb{S}$, de la Definición 3.2.8, sabemos que $\mathbb{S}' \subset \mathbb{S}$. Luego, existe al menos un argumento $\mathcal{B} \in \mathbb{S}$ tal que $\mathcal{B} \notin \mathbb{S}'$. Por *reductio ad absurdum*, asumiremos $\mathfrak{pr}(\mathbb{S}) = \mathfrak{pr}(\mathbb{S}')$ (ambas estructuras $\mathbb{S}$ y $\mathbb{S}'$ tienen el mismo conjunto de premisas libres). Consideremos los árboles de soporte $\{\mathcal{T}, \mathcal{T}'\} \subseteq \mathfrak{Trees}_{\mathbf{A}}(\alpha)$, donde $\mathcal{T}$ está asociado a $\mathbb{S}$, y $\mathcal{T}'$ a $\mathbb{S}'$. Dos opciones aparecen: o bien $\mathcal{T} = \mathcal{T}'$ ó $\mathcal{T} \neq \mathcal{T}'$. Para el primero, de la Definición 3.2.8, sabemos que $\mathcal{T}^* = \mathbb{S}$ y $\mathcal{T}'^* = \mathbb{S}'$, y por ello tenemos que $\mathbb{S} = \mathbb{S}'$ lo cual contradice la hipótesis. Para el segundo caso, asumiendo $\mathcal{T} \neq \mathcal{T}'$, consideremos las coaliciones-claiming $\widehat{\mathcal{C}}$ y $\widehat{\mathcal{C}'}$ para $\alpha$, tal que $\widehat{\mathcal{C}}$ es la raíz de $\mathcal{T}$ y $\widehat{\mathcal{C}'}$ es la raíz de $\mathcal{T}'$. Luego tenemos que $\widehat{\mathcal{C}'} \subseteq \mathbb{S}'$ y dado que $\mathbb{S}' \lhd \mathbb{S}$ vale, $\widehat{\mathcal{C}'} \subseteq \mathbb{S}$ se verifica. Luego, necesariamente tendremos que $\widehat{\mathcal{C}} = \widehat{\mathcal{C}'}$ dado que de la condición de minimalidad en árbol de soporte (ver Definición 3.2.7) dos diferentes coaliciones-claiming no podrán existir en el contexto del mismo árbol, formalmente, para toda $\lambda \in \mathcal{T}$, $\widehat{\mathcal{C}} \subseteq \mathcal{T}^*$ es una coalición-claiming en $\overline{\lambda}[0]$ (*i.e.*, $\alpha$) *sssi* $\widehat{\mathcal{C}} = \lambda[1]$. Luego, tenemos que para ambos árboles $\mathcal{T}$ y $\mathcal{T}'$, las premisas a ser cerraradas, correspondientes a sus respectivas coaliciones-claiming, coinciden. Una vez más, a partir de la condición de minimalidad en árbol de soporte tenemos que dos diferentes coaliciones-soporte no podrán existir en el contexto del mismo árbol, formalmente, para toda $\lambda \in \mathcal{T}$, $\widehat{\mathcal{C}_i} \subseteq \mathcal{T}^*$ es una coalición-soporte en $\overline{\lambda}[i]$ *sssi* $\widehat{\mathcal{C}_i} = \lambda[i+1]$ $(0 < i < |\lambda|)$. En consecuencia, las coaliciones-soporte de cada premisa de la raíz de $\mathcal{T}'$ necesariamente coincidirán con las coaliciones-soporte de cada premisa de la raíz de $\mathcal{T}$. Siguiendo esta línea de razonamiento, inevitablemente concluimos en $\mathcal{T} = \mathcal{T}'$, alcanzando el absurdo. Finalmente, $\mathfrak{pr}(\mathbb{S}) \neq \mathfrak{pr}(\mathbb{S}')$ es verificado. □

**Proposición 3.3.3** *Dado un* GenAF $\langle \mathbf{A}, \mathbf{R}_{\mathbf{A}} \rangle$ *y dos estructuras* $\mathbb{S}_1 \subseteq \mathbf{A}$ *y* $\mathbb{S}_2 \subseteq \mathbf{A}$*; si* $\mathbb{S}_1$ *y* $\mathbb{S}_2$ *son conflictivas entonces* $\mathbb{S}_1$ *y* $\mathbb{S}_2$ *son simultáneamente esquemáticas o argumentales.*

<u>*Demostración:*</u> Por *reductio ad absurdum*, asumimos que las estructuras $\mathbb{S}_1$ y $\mathbb{S}_2$ no son simultáneamente argumentales o esquemáticas. Dos opciones surgen, o bien (1) $\mathbb{S}_1$ es esquemática y $\mathbb{S}_2$ es argumental o (2) $\mathbb{S}_2$ es esquemática y $\mathbb{S}_1$ es argumental. Por (1), sabemos $\mathfrak{pr}(\mathbb{S}_1) \neq \emptyset$ pero $\mathfrak{pr}(\mathbb{S}_2) = \emptyset$, luego, $\mathfrak{pr}(\mathbb{S}_1) \models \perp$ (ver dependencia en Definición 3.3.1), lo cual es absurdo dado que viola la condición de consistencia en árbol de soporte (ver Definición 3.2.7), del árbol de soporte subyacente en $\mathbb{S}_1$ (ver Definición 3.2.8).

Para (2), sabemos que $\mathfrak{pr}(\mathbb{S}_2) \neq \emptyset$ pero $\mathfrak{pr}(\mathbb{S}_1) = \emptyset$, luego, $\emptyset \models \mathfrak{pr}(\mathbb{S}_2)$ (ver dependencia en Definición 3.3.1), lo cual implica que $\mathfrak{pr}(\mathbb{S}_2)$ contiene premisas tautológicas. Sin embargo, esto es absurdo dado que en tal caso, una coalición vacía de argumentos soportaría toda premisa en $\mathfrak{pr}(\mathbb{S}_2)$, y por luego, $\mathbb{S}_2$ no debería contener premisas libres.

Finalmente, tenemos que ambas $\mathbb{S}_1$ y $\mathbb{S}_2$ son simultáneamente esquemáticas o argumentales. $\square$

**Proposición 3.3.4** *Dado un* GenAF $\langle \mathbf{A}, \mathbf{R}_\mathbf{A} \rangle$ *y dos estructuras* $\mathbb{S}_1 \subseteq \mathbf{A}$ *y* $\mathbb{S}_2 \subseteq \mathbf{A}$*; si* $\mathbb{S}_2$ *es un undercut de* $\mathbb{S}_1$ *entonces ambas* $\mathbb{S}_1$ *y* $\mathbb{S}_2$ *son estructuras esquemáticas.*

Demostración: Dado $\{\mathfrak{cl}(\mathbb{S}_2), \rho\} \models \perp$, donde $\rho \in \mathfrak{pr}(\mathbb{S}_1)$, se verifica $\mathfrak{pr}(\mathbb{S}_1) \neq \emptyset$ dado que de la Definición 3.2.8, $\mathfrak{cl}(\mathbb{S}_2) \not\models \perp$. Luego, $\mathbb{S}_1$ es esquemática. Finalmente, de la Proposición 3.3.3, sabemos que $\mathbb{S}_2$ es también esquemática. $\square$

**Proposición 3.3.7** *Dada una* pANF *KB consistente* $\Sigma$*, para cualquier* $\alpha \in \mathcal{L}_{\texttt{cl}}$*,* $\Sigma \models \alpha$ *sssi existe una estructura argumental* $\mathbb{S} \subseteq \mathbf{A}$ *tal que* $\mathfrak{cl}(\mathbb{S}) = \alpha$*, donde* $\langle \mathbf{A}, \mathbf{R}_\mathbf{A} \rangle = \mathfrak{genaf}(\Sigma)$*.*

Demostración: Dado $\Sigma \models \alpha$ sabemos que existe un subconjunto $X \subseteq \Sigma$ tal que $X \models \alpha$ pero $X' \not\models \alpha$ para cualquier $X' \subset X$. Observe que $X$ conforma una estructura $\mathbb{S} \subseteq \mathbf{A}$ para $\alpha$, dado que existe un árbol de soporte $\mathcal{T} \in \mathfrak{Trees}_\mathbf{A}(\alpha)$ tal que $\mathcal{T}^* = \mathbb{S}$ y $\mathcal{T}$ describe un procedimiento de prueba en $\mathbb{S}$ para alcanzar $\alpha$. Además, dado $X \models \alpha$ sabemos que $X$ contiene la evidencia necesaria (o argumentos primitivos) para alcanzar $\alpha$ lo cual implica que ninguna premisa será libre en $\mathbb{S}$. Luego, $\mathbb{S}$ es argumental.

Por otro lado, si tenemos una estructura argumental $\mathbb{S} \subseteq \mathbf{A}$ para $\alpha$, sabemos que existe una forma de deducir efectivamente $\alpha$. Esto implica que el árbol de soporte $\mathcal{T} \in \mathfrak{Trees}_\mathbf{A}(\alpha)$ tal que $\mathcal{T}^* = \mathbb{S}$, describe un procedimiento de prueba dentro de $\mathbb{S}$ para alcanzar $\alpha$. Dado $\langle \mathbf{A}, \mathbf{R}_\mathbf{A} \rangle = \mathfrak{genaf}(\Sigma)$, de la Definición 3.1.13 sabemos que existe un conjunto $X \subseteq \Sigma$ tal que $\mathbb{S} = \{\mathfrak{arg}(\varphi) | \varphi \in X \text{ ó } \varphi^- \in X\}$, donde $\varphi^-$ es la contrapositiva de $\varphi$. Claramente, existe una prueba $X \subseteq \Sigma$ para $\alpha$, y dado que $\Sigma$ es consistente, $\Sigma \models \alpha$ se verifica. $\square$

**Proposición 3.3.10** *Dada una KB* $\Sigma \subseteq \mathcal{L}^\kappa$ *y su asociado* GenAF $\langle \mathbf{A}, \mathbf{R}_\mathbf{A} \rangle = \mathfrak{genaf}(\Sigma)$*,* $\mathbb{S}_1 \hookrightarrow^b_\mathbf{A} \mathbb{S}_2$ *sssi existe* $X_1 \subseteq \mathfrak{af}(\Sigma)$ *y* $X_2 \subseteq \mathfrak{af}(\Sigma)$ *tal que* $\varphi \in X_1$ *(*$\varphi \in X_2$*) sssi* $\mathfrak{arg}(\varphi) \in \mathbb{S}_1$ *ó* $\mathfrak{arg}(\varphi^-) \in \mathbb{S}_1$ *(*$\mathfrak{arg}(\varphi) \in \mathbb{S}_2$ *ó* $\mathfrak{arg}(\varphi^-) \in \mathbb{S}_2$*), donde* $\varphi^-$ *es la contrapositiva de* $\varphi$*, y* $X = X_1 \cup X_2$ *es inconsistente pero* $Y$ *es consistente, para cualquier* $Y \subset X$*.*

Demostración: Dados $\mathbb{S}_1 \subseteq \mathbf{A}$, $\mathbb{S}_2 \subseteq \mathbf{A}$, y $\langle \mathbf{A}, \mathbf{R}_\mathbf{A} \rangle = \mathfrak{genaf}(\Sigma)$, es fácil ver la existencia de tales subconjuntos $X_1 \subseteq \mathfrak{af}(\Sigma)$ y $X_2 \subseteq \mathfrak{af}(\Sigma)$, tal como es referido por la hipótesis. Necesitamos mostrar que $X$ es inconsistente pero $Y$ es consistente, para cualquier $Y \subset X$. Dado que ambas $\mathbb{S}_1$ y $\mathbb{S}_2$ son estructuras conflictivas en $\hookrightarrow^b_\mathbf{A}$, de la Definición 3.3.9, sabemos que ambas son argumentales y $\mathbb{S}_1$ es un rebuttal de $\mathbb{S}_2$. Por ello, $\{\mathfrak{cl}(\mathbb{S}_1), \mathfrak{cl}(\mathbb{S}_2)\} \models \perp$ y $X_1 \models \mathfrak{cl}(\mathbb{S}_1)$ y $X_2 \models \mathfrak{cl}(\mathbb{S}_2)$. Luego, es fácil ver que $X \models \{\mathfrak{cl}(\mathbb{S}_1), \mathfrak{cl}(\mathbb{S}_2)\}$ y por lo tanto

$X \models \bot$, pero por la condición de minimalidad de árbol de soporte (ver Definición 3.2.7) del árbol de soporte subyacente en $\mathbb{S}_1$ ó $\mathbb{S}_2$, sabemos que para cualquier $Y \subset X$, o bien $Y \not\models \mathfrak{cl}(\mathbb{S}_1)$ ó $Y \not\models \mathfrak{cl}(\mathbb{S}_2)$. Más aún, dado que ambas estructuras modelan un conflicto base, de la Definición 3.3.5 sabemos que no existe par conflictivo interno en ellos. Esto es, para cualquier $\mathbb{S}'_1 \trianglelefteq \mathbb{S}_1$ y cualquier $\mathbb{S}'_2 \trianglelefteq \mathbb{S}_2$, $\mathbb{S}'_1$ y $\mathbb{S}'_2$ son conflictivas *sssi* $\mathbb{S}'_1 = \mathbb{S}_1$ y $\mathbb{S}'_2 = \mathbb{S}_2$. Finalmente, es claro que $Y \not\models \bot$.

Para la dirección opuesta, la prueba sigue de forma análoga. □

**Lema 3.3.11** *Dada una* `pANF` *KB* $\Sigma$*, si* $\Sigma$ *es consistente entonces* $\hookrightarrow_{\mathbf{A}} = \emptyset$*, donde* $\langle \mathbf{A}, \mathbf{R}_{\mathbf{A}} \rangle = \mathfrak{genaf}(\Sigma)$.

<u>*Demostración:*</u> Por *reductio ad absurdum*, asumimos que $\Sigma$ es consistente pero $\hookrightarrow_{\mathbf{A}} \neq \emptyset$. También asumimos $(\mathbb{S}_1, \mathbb{S}_2) \in \hookrightarrow_{\mathbf{A}}$, donde $\mathbb{S}_1 \subseteq \mathbf{A}$ y $\mathbb{S}_2 \subseteq \mathbf{A}$ son estructuras argumentales; y también la existencia de dos subconjuntos $X_1 \subseteq \Sigma$ y $X_2 \subseteq \Sigma$ tal que $\varphi \in X_1$ ($\varphi \in X_2$) *sssi* $\mathfrak{arg}(\varphi) \in \mathbb{S}_1$ ó $\mathfrak{arg}(\varphi^-) \in \mathbb{S}_1$ ($\mathfrak{arg}(\varphi) \in \mathbb{S}_2$ ó $\mathfrak{arg}(\varphi^-) \in \mathbb{S}_2$), donde $\varphi^-$ es la contrapositiva de $\varphi$. De la Definición 3.2.8, es claro que $X_1 \models \mathfrak{cl}(\mathbb{S}_1)$ y $X_2 \models \mathfrak{cl}(\mathbb{S}_2)$. Luego, $\Sigma \models \{\mathfrak{cl}(\mathbb{S}_1), \mathfrak{cl}(\mathbb{S}_2)\}$. Sin embargo, dado que $\{\mathfrak{cl}(\mathbb{S}_1), \mathfrak{cl}(\mathbb{S}_2)\} \models \bot$ (ver Definición 3.3.1), podemos inferir $\Sigma \models \bot$, *i.e.*, $\Sigma$ es inconsistente lo cual es absurdo. Finalmente, $\hookrightarrow_{\mathbf{A}} = \emptyset$ se verifica. □

**Teorema 3.3.12** *Dada una* `pANF` *KB* $\Sigma$*, si* $\Sigma$ *es coherente y consistente entonces* $\mathbf{R}_{\mathbf{A}} = \emptyset$*, donde* $\langle \mathbf{A}, \mathbf{R}_{\mathbf{A}} \rangle = \mathfrak{genaf}(\Sigma)$.

<u>*Demostración:*</u> Por *reductio ad absurdum*, asumiremos que $\Sigma$ es coherente y consistente, pero $\mathbf{R}_{\mathbf{A}} \neq \emptyset$. Del Lema 3.3.11, sabemos que $\hookrightarrow_{\mathbf{A}} = \emptyset$ luego, de la Definición 3.3.9 y Proposición 3.3.3, sabemos que ambos $\mathbb{S}_1$ y $\mathbb{S}_2$ son esquemáticas, para cualquier $(\mathbb{S}_1, \mathbb{S}_2) \in \mathbf{R}_{\mathbf{A}}$. De la Definición 3.3.1, sabemos que $\mathfrak{pr}(\mathbb{S}_2) \models \mathfrak{pr}(\mathbb{S}_1)$, y obtenemos dos opciones: o bien (1) $\mathbb{S}_1$ es undercut de $\mathbb{S}_2$, ó (2) $\mathbb{S}_1$ es rebuttal de $\mathbb{S}_2$.

Asumiremos sin pérdida de generalidad $(\mathbb{S}_1, \mathbb{S}_2) \in \mathbf{R}^b_{\mathbf{A}}$. Además, asumimos la existencia de dos subconjuntos $X_1 \subseteq \Sigma$ y $X_2 \subseteq \Sigma$ tal que $\varphi \in X_1$ ($\varphi \in X_2$) *sssi* $\mathfrak{arg}(\varphi) \in \mathbb{S}_1$ ó $\mathfrak{arg}(\varphi^-) \in \mathbb{S}_1$ ($\mathfrak{arg}(\varphi) \in \mathbb{S}_2$ ó $\mathfrak{arg}(\varphi^-) \in \mathbb{S}_2$), donde $\varphi^-$ es la contrapositiva de $\varphi$.

Para (1), tenemos que $\{\mathfrak{cl}(\mathbb{S}_1), \rho\} \models \bot$, donde $\rho \in \mathfrak{pr}(\mathbb{S}_2)$ (ver Definición 3.3.1), y luego $\{\mathfrak{cl}(\mathbb{S}_1), \rho\}$ es insatisfacible, lo cual implica que $\mathfrak{cl}(\mathbb{S}_1)^{\mathcal{I}} \cap \rho^{\mathcal{I}} = \emptyset$ para todo modelo $\mathcal{I}$ de $\Sigma$. Observe que para cualquier interpretación $\mathcal{I}$ tal que $\mathcal{I} \models \mathfrak{pr}(\mathbb{S}_2)$, tenemos que $\mathcal{I} \models \{\mathfrak{cl}(\mathbb{S}_1), \rho\}$ (dado que $\mathfrak{pr}(\mathbb{S}_2) \models \mathfrak{pr}(\mathbb{S}_1)$) lo cual concluye verificando $\mathcal{I} \models \bot$. Por lo tanto, para cualquier modelo $\mathcal{I}$ se verifica $\mathfrak{pr}(\mathbb{S}_2)^{\mathcal{I}} = \emptyset$, y por ello también $\rho^{\mathcal{I}} = \emptyset$. Asumiendo una fórmula $\varphi \in X_2$ tal que $\varphi$ tiene la forma $\rho \wedge \rho_1 \wedge \ldots \wedge \rho_n \rightarrow \beta$ (fórmula $\mathtt{pANF}$), es claro que $\rho^{\mathcal{I}} \cap \rho_1^{\mathcal{I}} \cap \ldots \cap \rho_n^{\mathcal{I}} = \emptyset$, para todo modelo $\mathcal{I}$. Luego, $\Sigma$ es incoherente lo cual es absurdo, y finalmente, $\mathbb{S}_1$ no puede ser undercut de $\mathbb{S}_2$.

La prueba sigue en forma similar para el caso de (2): dado que $\{\mathfrak{cl}(\mathbb{S}_1), \mathfrak{cl}(\mathbb{S}_2)\} \models \perp$, necesariamente tenemos que $\mathfrak{pr}(\mathbb{S}_2)^{\mathcal{I}} = \emptyset$, para cualquier modelo $\mathcal{I}$ de $\Sigma$ (dado que $\mathfrak{pr}(\mathbb{S}_2) \models \mathfrak{pr}(\mathbb{S}_1)$). Luego, podemos asumir la existencia de una fórmula $\varphi \in X_2$ tal que $\varphi$ tiene la forma $\rho \wedge \rho_1 \wedge \ldots \wedge \rho_n \rightarrow \beta$ (fórmula $\mathtt{pANF}$), y además $\rho \in \mathfrak{pr}(\mathbb{S}_2)$. Es claro que $\rho^{\mathcal{I}} = \emptyset$ y entonces $\rho^{\mathcal{I}} \cap \rho_1^{\mathcal{I}} \cap \ldots \cap \rho_n^{\mathcal{I}} = \emptyset$, para todo modelo $\mathcal{I}$. Luego, $\Sigma$ es incoherente lo cual es nuevamente absurdo, y por lo tanto, $\mathbb{S}_1$ no puede ser rebuttal de $\mathbb{S}_2$.

Finalmente, es claro que $\mathbf{R_A} = \emptyset$ se verifica. □

**Proposición 3.3.16** *Dado un* $\mathtt{GenAF}$ $\langle \mathbf{A}, \mathbf{R_A} \rangle \in \mathbb{G}$*, cualquier extensión estándar de Dung determinada por* $\mathfrak{sem}(\langle \mathbf{A}, \mathbf{R_A} \rangle)$ *es libre de conflictos.*

*Demostración:* Directo a partir de la Definición 3.3.15, Definición 3.3.14, y Definición 3.3.13. □

**Proposición 3.3.17** *Dado un* $\mathtt{GenAF}$ $\langle \mathbf{A}, \mathbf{R_A} \rangle \in \mathbb{G}$*, si* $\mathfrak{sem}(\langle \mathbf{A}, \mathbf{R_A} \rangle) = X$ *es una extensión estable entonces* $X \cup \{\mathcal{B}\}$ *no es libre de conflictos, para cualquier* $\mathcal{B} \in (\mathbf{A} \setminus X)$*.*

*Demostración:* De la Proposición 3.3.16 sabemos que $X$ eslibre de conflictos. Además, dado que $X$ es una extensión estable, de la Definición 3.3.15 sabemos que para cualquier estructura $\mathbb{S} \subseteq (\mathbf{A} \setminus X)$ existe una estructura $\mathbb{S}' \subseteq X$ tal que $\mathbb{S}' \mathbf{R}^b_{\mathbf{A}} \mathbb{S}$. Esto incluye cualquier estructura primitiva, por lo tanto, podemos asumir la existencia de una argumento $\mathcal{B} \in \mathbf{A} \setminus X$ tal que $\mathbb{S} = \{\mathcal{B}\}$. Luego, sabemos que $\mathbb{S}$ es derrotada por $\mathbb{S}' \subseteq X$, y también que $(\mathbb{S}', \mathbb{S})$ es un ataque base. Por lo tanto, siendo que el ataque se realciona en forma directa con el argumento $\mathcal{B}$, la inclusión de $\mathcal{B}$ en $X$ hace al conjunto $X \cup \{\mathcal{B}\}$ conflictivo. □

**Lema 3.3.19** *Dada una* $\mathtt{pANF}$ *KB* $\Sigma \subseteq \mathcal{L}^{\kappa}$*, si* $\Sigma \approx\!\!\!\!| \ \alpha$ *entonces* $\alpha$ *es inferible de* $\mathfrak{genaf}(\Sigma)$*.*

*Demostración:* De la Definición 3.3.18, sabemos que existe al menos una estructura argumental $\mathbb{S} \subseteq \mathbf{A}$, donde $\langle \mathbf{A}, \mathbf{R_A} \rangle = \mathfrak{genaf}(\Sigma)$, tal que $\mathfrak{cl}(\mathbb{S}) = \alpha$. Finalmente, del Lema 3.2.11, sabemos que $\alpha$ es inferible. □

**Teorema 3.3.20** *Dada una* $\mathtt{pANF}$ *KB* $\Sigma \subseteq \mathcal{L}^{\kappa}$ *coherente-consistente y una fórmula* $\alpha \in \mathcal{L}_{\mathtt{cl}}$*,* $\Sigma \models \alpha$ *sssi* $\Sigma \approx\!\!\!\!| \ \alpha$*.*

<u>*Demostración:*</u> Asumiendo $\langle \mathbf{A}, \mathbf{R_A} \rangle = \mathfrak{genaf}(\Sigma)$, y cualquiera de las semánticas estándares de Dung adaptadas al GenAF, tenemos al menos una extensión completa $\mathfrak{sem}(\langle \mathbf{A}, \mathbf{R_A} \rangle) = X$ (ver Proposición 3.3.16) lo cual determina la obtención del último punto fijo de la función característica, *i.e.*, $\mathcal{F}(X) = X$. Luego, es claro que $X = \mathbf{A}$ dado que $\mathbf{R_A} = \emptyset$ (ver Teorema 3.3.12).

Para cualquier fórmula $\alpha \in \mathcal{L}_{\mathtt{cl}}$, si $\Sigma \models \alpha$ se verifica entonces (de la Proposición 3.3.7) existe una estructura argumental $\mathbb{S} \subseteq \mathbf{A}$ tal que $\mathfrak{cl}(\mathbb{S}) = \alpha$. Finalmente, dado $X = \mathbf{A}$, tenemos que $\mathbb{S} \subseteq X$ y dado que $X$ es una extensión, tenemos que $\mathbb{S}$ no tiene derrotadores en $X$ lo cual implica que $\mathbb{S}$ es garantizado, y luego $\Sigma \mid\!\approx \alpha$.

Por otro lado, si $\Sigma \mid\!\approx \alpha$ entonces del Lema 3.3.19 tenemos que $\alpha$ es inferible y del Lema 3.2.11, que existe una estructura argumental $\mathbb{S} \subseteq \mathbf{A}$ tal que $\mathfrak{cl}(\mathbb{S}) = \alpha$. Dado que $\Sigma$ está en pANF, puede ser asumida la existencia de un conjunto $Y \subseteq \Sigma$ tal que para cualquier fórmula $\varphi \in Y$ existe un argumento $\mathcal{B} \in \mathbb{S}$ tal que $\mathfrak{arg}(\varphi) = \mathcal{B}$. Por lo tanto, $\mathbb{S}$ es construida en forma directa de $Y$. Observe que $Y$ es una prueba para $\alpha$ dado que contiene el árbol de soporte para la construcción de $\mathbb{S}$, el cual describe un procedimiento de prueba para $\alpha$. Finalmente, dado que $\Sigma$ es consistente, $\Sigma \models \alpha$ se verifica. □

**Proposición 3.4.4** *Dada una* pANF *KB* $\Sigma \subseteq \mathcal{L}^{\kappa}$*, si* $\varphi \in (\Sigma \setminus \mathfrak{back}(\mathfrak{genaf}(\Sigma)))$ *entonces* $\varphi \models \bot$*.*

<u>*Demostración:*</u> Esta situación ocurre para una $\varphi$ que viola la condición de consistencia de la Definición 3.1.6, y por ello $\mathfrak{arg}(\varphi) \notin \mathbb{A}\mathtt{rgs}$. Para demostrarlo, asumiremos por *reductio ad absurdum* que existe algún $\varphi \in \Sigma$ pero $\varphi \notin \mathfrak{back}(\mathfrak{genaf}(\Sigma))$ tal que $\varphi \not\models \bot$. Dado que $\Sigma$ está en pANF, sabemos que $\varphi$ también está en pANF. Es claro que $\mathfrak{arg}(\varphi) \in \mathbb{A}\mathtt{rgs}$. Luego, de la Definición 3.1.13 y Definición 3.1.6 tenemos que $\mathfrak{arg}(\varphi) \in \mathbf{A}$, donde $\langle \mathbf{A}, \mathbf{R_A} \rangle = \mathfrak{genaf}(\Sigma)$. Además, dado que $\varphi \notin \mathfrak{back}(\mathfrak{genaf}(\Sigma))$, de la Definición 3.4.1 la única opción que tenemos es $\mathfrak{arg}(\varphi^-) \in \mathbb{A}\mathtt{rgs}$ pero $\mathfrak{arg}(\varphi^-) \notin \mathbf{A}$.Esto implica que $\mathfrak{arg}(\varphi^-)$ viola la condición de consistencia en la Definición 3.1.6. Por lo tanto, $\varphi^- \models \bot$ y entonces es claro que $\varphi \models \bot$, alcanzando el absurdo. □

**Proposición 3.4.5** *Dado un* GenAF $\langle \mathbf{A}, \mathbf{R_A} \rangle \subseteq \mathbb{G}$*,* $\mathbf{R_A} = \emptyset$ *sssi* $\mathfrak{back}(\mathbf{A})$ *es una* pANF *KB consistente-coherente.*

<u>*Demostración:*</u> Si $\mathbf{R_A} = \emptyset$ es verificado, sabemos que no existe par de estructuras conflictivas en $\mathbf{A}$. Por lo tanto no habrá rebuttals ni undercuts en $\langle \mathbf{A}, \mathbf{R_A} \rangle$. Primeramente, dado que $\hookrightarrow_{\mathbf{A}} = \emptyset$ sabemos que no existe par de estructuras argumentales con-

flictivas (*i.e.*, rebuttals). Esto implica que no existe par $\alpha_1 \in \mathcal{L}_{\mathtt{cl}}$ y $\alpha_2 \in \mathcal{L}_{\mathtt{cl}}$ tal que $\mathfrak{back}(\mathbf{A}) \models \{\alpha_1, \alpha_2\}$ y $\{\alpha_1, \alpha_2\} \models \bot$. Luego, $\mathfrak{back}(\mathbf{A})$ admite al menos un modelo, y por lo tanto será consistente.

Segundo, dado que no existe par de estructuras esquemáticas conflictivas, sabemos que no existe par de (1) rebuttals esquemáticos, ni de (2) undercuts esquemáticos. Luego, sabemos que para cualquier par de estructuras $\mathbb{S}_1 \subseteq \mathbf{A}$, $\mathbb{S}_2 \subseteq \mathbf{A}$, si tenemos que $\mathfrak{pr}(\mathbb{S}_2) \models \mathfrak{pr}(\mathbb{S}_1)$, a partir de (1) sabemos que $\{\mathfrak{cl}(\mathbb{S}_1), \mathfrak{cl}(\mathbb{S}_2)\} \not\models \bot$, y de (2), que $\{\mathfrak{cl}(\mathbb{S}_1)\} \cup \mathfrak{pr}(\mathbb{S}_2) \not\models \bot$. Esto implica que para cualquier modelo $\mathcal{I}$ de $\mathfrak{back}(\mathbf{A})$, si $\mathcal{I} \models \mathfrak{pr}(\mathbb{S}_2)$ entonces $\mathfrak{back}(\mathbf{A})^{\mathcal{I}} \neq \emptyset$. Luego, es claro que para cualquier lado izquierdo de una fórmula en $\mathfrak{back}(\mathbf{A})$ existe una interpretación no-vacía la cual es modelo de $\mathfrak{back}(\mathbf{A})$. Finalmente, $\mathfrak{back}(\mathbf{A})$ es coherente.

Por otro lado, si $\mathfrak{back}(\mathbf{A})$ es consistente-coherente, a partir del Teorema 3.3.12, sabemos que $\mathbf{R}_{\mathbf{A}} = \emptyset$ es verificado. □

**Lema 3.4.6** *Dado un* $\mathtt{GenAF}$ $\langle \mathbf{A}, \mathbf{R}_{\mathbf{A}} \rangle \subseteq \mathbb{G}$*,* $\mathfrak{back}(\mathfrak{sem}(\langle \mathbf{A}, \mathbf{R}_{\mathbf{A}} \rangle))$ *es una* $\mathtt{pANF}$ *KB consistente-coherente.*

<u>*Demostración:*</u> A partir de la Proposición 3.3.16, sabemos que $\mathfrak{sem}(\langle \mathbf{A}, \mathbf{R}_{\mathbf{A}} \rangle)$ es libre de conflicto, lo cual siguiendo la Definición 3.3.1, implica $\mathbf{R}_{\mathbf{A}} = \emptyset$. Finalmente, por la Proposición 3.4.5, sabemos que $\mathfrak{back}(\mathfrak{sem}(\langle \mathbf{A}, \mathbf{R}_{\mathbf{A}} \rangle))$ es una $\mathtt{pANF}$ KB consistente-coherente. □

**Teorema 3.4.9** *Dada una KB* $\Sigma \subseteq \mathcal{L}^{\kappa}$*,* $\mathfrak{debug}(\Sigma)$ *es una* $\mathtt{pANF}$ *KB consistente-coherente.*

<u>*Demostración:*</u> Por la Proposición 3.3.16 y Definición 3.3.15, para cualquier semántica $\mathtt{GenAF}$ adoptada por el operador "$\mathfrak{sem}$" sabemos que $\mathfrak{sem}(\mathfrak{genaf}(\Sigma))$ es libre de conflictos. Luego, por el Lema 3.4.6, Definición 3.4.1, y Definición 3.4.7, sabemos que $\mathfrak{back}(\mathfrak{sem}(\mathfrak{genaf}(\Sigma))) \cap \mathfrak{af}(\Sigma)$ es una $\mathtt{pANF}$ KB consistente-coherente. □

**Teorema de Representación 3.4.10** *Dada una* $\mathtt{pANF}$ *KB* $\Sigma \subseteq \mathcal{L}^{\kappa}$ *y asumiendo que* $\mathfrak{sem}$ *es obtenido a través de la semántica estable sobre* $\hookrightarrow_{\mathbf{A}}$*, la operación* $\mathfrak{debug}(\Sigma)$ *es una consolidación kernel sssi garantiza los postulados* **C1***,* **C2***, y* **C3***.*

<u>*Demostración:*</u> **Construcción a postulados)** Dado que $\Sigma$ está en $\mathtt{pANF}$, por la Definición 3.1.3, tenemos que $\mathfrak{af}(\Sigma) = \Sigma$. Luego, por la Definición 3.4.7, $\mathfrak{debug}(\Sigma) = \mathfrak{back}(\mathfrak{sem}(\mathfrak{genaf}(\Sigma))) \cap \mathfrak{af}(\Sigma) = \mathfrak{back}(\mathfrak{sem}(\mathfrak{genaf}(\Sigma))) \cap \Sigma$. Claramente, $\mathfrak{debug}(\Sigma) \subseteq \Sigma$, y por ello el postulado **C1** es verificado trivialmente.

El postulado **C2** es verificado en forma directa del Teorema 3.4.9. En tanto que para el postulado **C3**, dado que $\mathfrak{sem}(\mathfrak{genaf}(\Sigma))$ determina una extensión estable $X \subseteq \mathbf{A}$, donde $\langle \mathbf{A}, \mathbf{R}_{\mathbf{A}} \rangle = \mathfrak{genaf}(\Sigma)$, por la Definición 3.3.15 sabemos que $X$ es una extensión preferida y para cualquier estructura $\mathbb{S} \subseteq (\mathbf{A} \setminus X)$ existe una estructura $\mathbb{S}' \subseteq X$ tal que $\mathbb{S}'\mathbf{R}_{\mathbf{A}}\mathbb{S}''$, para alguna $\mathbb{S}''$ tal que $\mathbb{S} \trianglelefteq \mathbb{S}''$. Asumamos sin pérdida de generalidad, que la estructura $\mathbb{S}$ es una estructura primitiva conteniendo el argumento $\mathcal{B} = \langle \{\rho_1, \ldots, \rho_n\}, \alpha \rangle$. Es claro que $\mathcal{B} \in \mathbf{A}$ pero $\mathcal{B} \notin X$, y por la Observation 3.4.3, asumiendo $\varphi = \rho_1 \wedge \ldots \wedge \rho_n \rightarrow \alpha$, tenemos que $\varphi \in \Sigma$ pero $\varphi \notin \mathfrak{debug}(\Sigma)$ (ver también la Definición 3.4.1). Asumamos que $\mathfrak{back}(X) = \Sigma'$, dado que $X$ es libre de conflictos (ver Proposición 3.3.16) por la Proposición 3.4.5 sabemos que $\Sigma'$ es consistente. Luego, por la Proposición 3.3.17, sabemos que $X \cup \{\mathcal{B}\}$ no es libre de conflictos, y por lo tanto $\Sigma' \cup \{\varphi\}$ no es consistente. Finalmente, el postulado **C3** es verificado.

**Postulados a construcción)** Por el Teorema 2.2.10 sabemos que $\Sigma!$ satisface los postulados **C1**, **C2**, y **C3**. Necesitamos mostrar que $\mathfrak{debug}(\Sigma)$ es una consolidación kernel $\Sigma!$. A tal fin, por la Proposición 3.3.10 sabemos que para cualquier par $(\mathbb{S}_1, \mathbb{S}_2) \in \hookrightarrow^b_{\mathbf{A}}$ (con $\langle \mathbf{A}, \mathbf{R}_{\mathbf{A}} \rangle = \mathfrak{genaf}(\Sigma)$), tenemos un conjunto $(X_1 \cup X_2) \subseteq \Sigma$, tal que $\varphi \in X_1$ ($\varphi \in X_2$) *sssi* $\mathfrak{arg}(\varphi) \in \mathbb{S}_1$ ó $\mathfrak{arg}(\varphi^-) \in \mathbb{S}_1$ ($\mathfrak{arg}(\varphi) \in \mathbb{S}_2$ ó $\mathfrak{arg}(\varphi^-) \in \mathbb{S}_2$); el cual determina una fuente mínima de inconsistencia en $\Sigma$. Por lo tanto, $(X_1 \cup X_2)$ es un conjunto $\perp$-kernel de $\Sigma$. Luego, por la Proposición 3.3.17, sabemos que una extensión obtenida mediante la semántica estable (ver Definición 3.3.15) determina un subconjunto $\mathfrak{sem}(\mathfrak{genaf}(\Sigma)) \subseteq \mathbf{A}$ maximal y libre de conflicto, de argumentos. Es claro que $(\mathbb{S}_1 \cup \mathbb{S}_2) \not\subseteq \mathfrak{sem}(\mathfrak{genaf}(\Sigma))$, y por lo tanto, $(X_1 \cup X_2) \not\subseteq \mathfrak{back}(\mathfrak{sem}(\mathfrak{genaf}(\Sigma)))$, y claramente $(X_1 \cup X_2) \not\subseteq \mathfrak{debug}(\Sigma)$. Finalmente, dado que $\mathfrak{debug}(\Sigma)$ es una consolidación kernel $\Sigma!$, por el Teorema 2.2.10, sabemos que $\mathfrak{debug}(\Sigma)$ satisface los postulados **C1**, **C2**, y **C3**. □

**Proposición 3.5.2** *Los lenguajes $\mathcal{L}_{\mathtt{pr}} ::= L$ y $\mathcal{L}_{\mathtt{cl}} ::= Cl|Cl(a)|R(a,b)$ determinan un lenguaje argumental legal $\mathbb{A}\mathtt{rgs}$.*

<u>*Demostración:*</u> La prueba es directa: de acuerdo a la Definición 3.1.2, para cualquier $\rho \in \mathcal{L}_{\mathtt{pr}}$ existe un conjunto $\Phi \subseteq \mathcal{L}_{\mathtt{cl}}$ tal que $\Phi \models \rho$. Dado que $Cl$ es una disyunción de $L$, siempre es posible verificar $\Phi \models \rho$. Por lo tanto, $\mathbb{A}\mathtt{rgs}$ es un lenguaje argumental legal. □

## A.2. Respecto del Modelo Dialéctico-global (Cap. 4)

**Teorema 4.2.2** *Dado un árbol de dialéctica $\mathcal{T}(\mathcal{R}) \in \mathbf{T}_{\Sigma}$, no existe línea de ataque $\lambda \in \mathcal{T}(\mathcal{R})$ sssi $\mathcal{T}(\mathcal{R})$ es un árbol garantizante.*

*Demostración:* Considerando un árbol de dialéctica, tenemos dos opciones: o bien es garantizante, o no lo es. Si el árbol es garantizante, su argumento raíz será no-derrotado. Dado que la raíz es un argumento pro, por la Definición 4.2.1 sabemos que no existen líneas de ataque. Por otro lado, si el árbol es no-garantizante, es claro que su argumento raíz está marcado como $D$. En tal caso, sabemos que existe al menos uno de sus hijos el cual es un argumento con marcado como $U$. Además, tal argumento con concluye marcado como $U$ dado que o bien (1) es la hoja de una línea $\lambda$, ó (2) necesariamente no tendrá derrotadores no-derrotados. Por 1), es claro que $\lambda$ es una línea de ataque. Por 2), cada uno de sus hijos se sabe será un argumento pro marcado como $D$. Dado que este es el caso del argumento raíz, la prueba necesariamente vuelve a 1). Por lo tanto, si el árbol es no-garantizante sabemos que siempre existe una línea de ataque.

En sentido contrario, asumiendo que no existen líneas de ataque, surgen dos opciones, o bien el argumento raíz es marcado como $D$, ó $U$. Para el segundo caso, es claro que el árbol es garantizante. Para el primer caso en que la raíz es marcada como $D$, obtenemos un absurdo dado que hemos mostrado que en este caso siempre aparecerá una línea de ataque. Por otro lado, asumiendo que tenemos alguna línea de ataque, la única alternativa para el argumento raíz es que sea marcado como $D$, y luego, el árbol será no-garantizante.
□

**Proposición 4.2.3** *Si $\lambda \in \mathcal{T}(\mathcal{R})$ es una línea de ataque entonces para cualquier $\mathcal{B} \in \lambda^{-}$, el segmento superior $\lambda^{\uparrow}(\mathcal{B})$ se torna línea de no-ataque.*

*Demostración:* Cuando cortamos una línea mediante la remoción de un argumento con, el segmento superior que queda determinado tendrá logitud impar. Finalmente, por la Proposición A.2.3, sólo líneas de argumentación de longitud par pueden resultar de ataque.
□

**Lema 4.2.7** *Dada una KB $\Sigma \subseteq \mathcal{L}$, si $*^{\omega}$ es una revisión de argumento entonces $\mathcal{T}(\mathcal{R}) \in \mathbb{T}_{(\Sigma *^{\omega} \mathcal{R})}$ no posee líneas de ataque, para ningún $\mathcal{R} \in \mathbb{A}_{\mathcal{L}}$.*

*Demostración:* Por la condición 1 en la Definición 4.2.5, sabemos que la incisión $\sigma$ es no-vacía si hay líneas de ataque en $\mathcal{T}'(\mathcal{R}) \in \mathbb{T}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}))}$. Además, por la condición 2

(Definición 4.2.5), la función de incisión toma una creencia de un argumento con, $\mathcal{B} \in \lambda^-$, correspondiente a alguna línea $\lambda$ de ataque (o cualquier otra que pueda tornarse atacante) asegurando que no habrá otra creencia tomada por $\sigma$ dentro de un argumento ubicado por encima de $\mathcal{B}$ en $\lambda$. Por la Proposición 4.2.3, $\lambda$ es tornada a no-atacante. Finalmente, por la Definición 4.2.6, $\mathcal{T}(\mathcal{R})$ no tiene líneas de ataque. $\square$

Extenderemos la noción de equivalencia estricta a conjuntos de argumentos.

**Definición A.2.1 (Conjuntos de Argumentos Estrictamente Equivalentes)** *Dadas dos KBs $\Sigma \subseteq \mathcal{L}$ y $\Sigma' \subseteq \mathcal{L}$, los conjuntos de argumentos $\mathbb{A}_\Sigma$ y $\mathbb{A}_{\Sigma'}$ son* ***conjuntos estrictamente equivalentes****, notado $\mathbb{A}_\Sigma \dashv\Vdash \mathbb{A}_{\Sigma'}$ sssi para cualquier $\mathcal{B} \in \mathbb{A}_\Sigma$ (resp., $\mathcal{B} \in \mathbb{A}_{\Sigma'}$) existe $\mathcal{C} \in \mathbb{A}_{\Sigma'}$ (resp., $\mathcal{C} \in \mathbb{A}_\Sigma$) tal que $\mathcal{B} \dashv\Vdash \mathcal{C}$.*

Observe que el símbolo "$\dashv\Vdash$" ha sido sobrecargado para identificar argumentos estrictamente equivalentes, y conjuntos de argumentos estrictamente equivalentes, dado que su uso será explícito en forma trivial: $\mathcal{B} \dashv\Vdash \mathcal{B}'$ y $\mathbb{A}_\Sigma \dashv\Vdash \mathbb{A}_{\Sigma'}$, respectivamente.

**Definición A.2.2 (Clase de Argumentos Estrictamente Equivalentes)** *Dada $\Sigma \subseteq \mathcal{L}$, una* ***clase de argumentos estrictamente equivalentes****, por simplicidad* ***cse****, es cualquier conjunto de argumentos $\Psi \subseteq \mathbb{A}_\Sigma$ tal que (1) para cualquier par $\mathcal{B}_1 \in \Psi$ y $\mathcal{B}_2 \in \Psi$, $\mathcal{B}_1 \dashv\Vdash \mathcal{B}_2$, y (2) para cualquier $\mathcal{B}_1 \in \Psi$ no existe $\mathcal{B}_2 \notin \Psi$ tal que $\mathcal{B}_1 \dashv\Vdash \mathcal{B}_2$.*

Subconjuntos $\Psi$ de argumentos estrictamente equivalentes pueden ser reconocidos a partir de un conjunto de argumentos. Esto es importante para mostrar que si un argumento en $\Psi$ es un nodo de un árbol de dialéctica $\mathcal{T}(\mathcal{R})$, entonces todo argumento dentro de $\Psi$ es un nodo en $\mathcal{T}(\mathcal{R})$, y más aún, el uso de cualquiera de ellos determina líneas estrictamente equivalentes (ver Proposición A.2.8).

**Lema 4.3.5** *Dada una KB $\Sigma \subseteq \mathcal{L}$, dos argumentos externos $\mathcal{R}_1 \in \mathbb{E}_\Sigma$ y $\mathcal{R}_2 \in \mathbb{E}_\Sigma$, y sus árboles de dialéctica $\mathcal{T}(\mathcal{R}_1) \in \mathbb{T}_{(\Sigma\cup\mathfrak{bd}(\mathcal{R}_1))}$ y $\mathcal{T}(\mathcal{R}_2) \in \mathbb{T}_{(\Sigma\cup\mathfrak{bd}(\mathcal{R}_2))}$; si $\mathcal{R}_1 \dashv\Vdash \mathcal{R}_2$ entonces $\mathcal{T}(\mathcal{R}_1)$ y $\mathcal{T}(\mathcal{R}_2)$ son árboles estrictamente equivalentes.*

*Demostración:* Asumiendo $\mathcal{R}_1 \dashv\Vdash \mathcal{R}_2$ es verificado, tenemos que $\mathbb{A}_{(\Sigma\cup\mathfrak{bd}(\mathcal{R}_1))} \dashv\Vdash \mathbb{A}_{(\Sigma\cup\mathfrak{bd}(\mathcal{R}_2))}$ se verifica por la Proposición A.2.6. Luego, por la Proposición A.2.7 sabemos que para todo derrotador $\mathcal{B} \in \mathbb{A}_{(\Sigma\cup\mathfrak{bd}(\mathcal{R}_1))}$ de $\mathcal{R}_1$, hay un derrotador $\mathcal{C} \in \mathbb{A}_{(\Sigma\cup\mathfrak{bd}(\mathcal{R}_2))}$ de $\mathcal{R}_2$ tal que $\mathcal{C} \dashv\Vdash \mathcal{B}$. Surgen dos alternativas: si $\mathfrak{bd}(\mathcal{B}) \cap \mathfrak{bd}(\mathcal{R}_1) = \emptyset$ entonces sabemos que existe un cse $\Psi \subseteq \mathbb{A}_\Sigma$

tal que $\{\mathcal{B}, \mathcal{C}\} \subseteq \Psi$, y por la Proposición A.2.8, la línea $[\mathcal{R}_1, \mathcal{B}, \ldots] \in \mathcal{T}(\mathcal{R}_1)$ es estrictamente equivalente a $[\mathcal{R}_1, \mathcal{C}, \ldots] \in \mathcal{T}(\mathcal{R}_1)$. Análogamente, la línea $[\mathcal{R}_2, \mathcal{B}, \ldots] \in \mathcal{T}(\mathcal{R}_2)$ es estrictamente equivalente a $[\mathcal{R}_2, \mathcal{C}, \ldots] \in \mathcal{T}(\mathcal{R}_2)$. En sentido contrario, si es el caso en que $\mathfrak{bd}(\mathcal{B}) \cap \mathfrak{bd}(\mathcal{R}_1) \neq \emptyset$, sabemos que $\mathfrak{bd}(\mathcal{C}) \cap \mathfrak{bd}(\mathcal{R}_2) \neq \emptyset$, y dado que $\mathbb{A}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_1))} \dashv\Vdash \mathbb{A}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_2))}$ y $\mathcal{C} \dashv\Vdash \mathcal{B}$, por la Proposición A.2.7 tenemos que para todo derrotador $\mathcal{D} \in \mathbb{A}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_1))}$ de $\mathcal{B}$ hay un derrotador $\mathcal{D}' \in \mathbb{A}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_2))}$ de $\mathcal{C}$ tal que $\mathcal{D} \dashv\Vdash \mathcal{D}'$. Finalmente, para toda línea en $\mathcal{T}(\mathcal{R}_1)$ hay una línea estrictamente equivalente en $\mathcal{T}(\mathcal{R}_2)$ y por lo tanto ambos $\mathcal{T}(\mathcal{R}_1)$ y $\mathcal{T}(\mathcal{R}_2)$ son árboles estrictamente equivalentes. □

**Proposición 4.3.7** *Dada una KB $\Sigma \subseteq \mathcal{L}$, y dos argumentos $\mathcal{R}_1 \in \mathbb{E}_\Sigma$ y $\mathcal{R}_2 \in \mathbb{E}_\Sigma$ tal que $\mathcal{R}_1 \dashv\Vdash \mathcal{R}_2$, siempre existe una función de incisión de argumentos $\sigma$ la cual es suave.*

*Demostración:* Dado que $\mathcal{R}_1 \dashv\Vdash \mathcal{R}_2$ se verifica, $\mathbb{A}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_1))} \dashv\Vdash \mathbb{A}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_2))}$ surge por la Proposición A.2.6, por lo tanto para cualquier argumento $\mathcal{B}_1 \in \mathbb{A}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_1))}$ sabemos que existe un argumento $\mathcal{B}_2 \in \mathbb{A}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_2))}$, tal que $\mathcal{B}_1 \dashv\Vdash \mathcal{B}_2$. Más aún, por el Lema 4.3.5 también sabemos que $\mathcal{T}(\mathcal{R}_1) \dashv\Vdash \mathcal{T}(\mathcal{R}_2)$, donde $\mathcal{T}(\mathcal{R}_1) \in \mathbb{T}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_1))}$ y $\mathcal{T}(\mathcal{R}_2) \in \mathbb{T}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_2))}$. Surgen dos opciones: o bien $\mathcal{B}_1 \in \mathcal{T}(\mathcal{R}_1)$ o no. Para el caso primero, dado $\mathcal{T}(\mathcal{R}_1) \dashv\Vdash \mathcal{T}(\mathcal{R}_2)$, sabemos que $\mathcal{B}_2 \in \mathcal{T}(\mathcal{R}_2)$, y por la Definición 4.3.4 asumiendo $\mathcal{B}_1 \in \lambda_1$ donde $\lambda_1 \in \mathcal{T}(\mathcal{R}_1)$, sabemos que hay una línea $\lambda_2 \in \mathcal{T}(\mathcal{R}_2)$ que es estrictamente equivalente a $\lambda_1$ y por lo tanto $\mathcal{B}_2 \in \lambda_2$. Asumiendo que "$\sigma$" es una función de incisión de argumentos, por la Proposición A.2.4, sabemos que ambos $\sigma(\mathcal{T}(\mathcal{R}_1)) \cap \mathfrak{bd}(\mathcal{B}_1) \subseteq \mathfrak{bd}(\mathcal{B}_1) \setminus \mathfrak{bd}(\mathcal{R}_1)$ y $\sigma(\mathcal{T}(\mathcal{R}_2)) \cap \mathfrak{bd}(\mathcal{B}_2) \subseteq \mathfrak{bd}(\mathcal{B}_2) \setminus \mathfrak{bd}(\mathcal{R}_2)$ se verifican. Luego, es claro que $\mathfrak{bd}(\mathcal{B}_1) \setminus \mathfrak{bd}(\mathcal{R}_1) = \mathfrak{bd}(\mathcal{B}_2) \setminus \mathfrak{bd}(\mathcal{R}_2)$ vale, donde $\mathcal{B}_2$ es algún argumento incluido en un cse $\Psi \subseteq \mathbb{A}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_2))}$ –por la Proposición A.2.8 sabemos que todo argumento en $\Psi$ es incluido en una línea $\lambda'_2 \in \mathcal{T}(\mathcal{R}_2)$ la cual es estrictamente equivalente a $\lambda_2$. Por lo tanto, $\sigma(\mathcal{T}(\mathcal{R}_1)) \cap \mathfrak{bd}(\mathcal{B}_1) = \sigma(\mathcal{T}(\mathcal{R}_2)) \cap \mathfrak{bd}(\mathcal{B}_2)$.

Por otro lado, si se verifica $\mathcal{B}_1 \notin \mathcal{T}(\mathcal{R}_1)$, dado $\mathcal{T}(\mathcal{R}_1) \dashv\Vdash \mathcal{T}(\mathcal{R}_2)$ también sabemos que $\mathcal{B}_2 \notin \mathcal{T}(\mathcal{R}_2)$. Finalmente, por la Definición 4.2.5, $\sigma(\mathcal{T}(\mathcal{R}_1)) \cap \mathfrak{bd}(\mathcal{B}_1) = \emptyset$ y $\sigma(\mathcal{T}(\mathcal{R}_2)) \cap \mathfrak{bd}(\mathcal{B}_2) = \emptyset$ se verifican. □

**Teorema de Representación 4.3.8** *Dada una KB $\Sigma \subseteq \mathcal{L}$ y $\mathcal{R} \in \mathbb{A}_\mathcal{L}$, $\Sigma *^\omega \mathcal{R}$ es una revisión suave de argumentos sssi garantiza éxito, inclusión, vacuidad, retención de núcleo, y uniformidad.*

*Demostración:* **(Construcción a postulados)** La prueba para **éxito** es trivial por el Lema 4.2.7 y el Teorema 4.2.2. **Inclusión** es trivialmente implicada por la Definición 4.2.6 y Definición 4.2.5. Para **vacuidad**, por *reductio ad absurdum*, si asumimos que $\Sigma \cup \mathfrak{bd}(\mathcal{R}) \subseteq$

$\Sigma *^{\omega} \mathcal{R}$ no se verifica, esto significa que la incisión es no-vacía, implicando que existe una línea de ataque en $\mathcal{T}(\mathcal{R}) \in \mathbb{T}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}))}$. Por lo tanto, por el Teorema 4.2.2 sabemos que $\mathcal{T}(\mathcal{R})$ es no-garantizante, y luego $\mathcal{R}$ no es garantizado en $\Sigma \cup \mathfrak{bd}(\mathcal{R})$ en contradicción con la hipótesis.

Para **retención de núcleo**, si existe algún $\beta \in \Sigma \setminus (\Sigma *^{\omega} \mathcal{R})$ entonces de la equivalencia para $\sigma$ adoptada como hipótesis en este teorema, sabes que $\beta \in \sigma(\mathcal{T}(\mathcal{R}))$. Por la condición 3, en la Definición 4.2.5, sabemos que $\beta \in \mathfrak{bd}(\mathcal{B})$, para algún $\mathcal{B} \in \lambda$, $\lambda \in \mathcal{T}(\mathcal{R})$. Además, sabemos que tal $\lambda$ necesariamente es o bien atacante (cond. $2a$) o tornada a atacante por una incisión colateral (cond. $2b$). Más aún, por el consecuente de la condición 2 en la Definición 4.2.5, sabemos que $\mathcal{B} \in \lambda^{-}$. Por lo tanto, por la Proposición 4.2.3, sabemos que $\lambda$ es efectivamente altereda mediante la eliminación de $\beta$. Por inclusión sabemos que $\Sigma *^{\omega} \mathcal{R} \subseteq \Sigma \cup \mathfrak{bd}(\mathcal{R})$, y por éxito también sabemos que $\mathcal{R}$ es garantizado en $\Sigma *^{\omega} \mathcal{R}$, y junto con la Proposición A.2.5, tenemos que $\mathfrak{bd}(\mathcal{R}) \subseteq \Sigma *^{\omega} \mathcal{R}$. Sea $\Sigma' = (\Sigma *^{\omega} \mathcal{R}) \setminus \mathfrak{bd}(\mathcal{R})$, entonces $\Sigma' \subseteq \Sigma$. Finalmente, es claro que $\mathcal{R}$ es garantizado en $\Sigma' \cup \mathfrak{bd}(\mathcal{R})$ (dado que iguala a $\Sigma *^{\omega} \mathcal{R}$), pero $\mathcal{R}$ no es garantizado en $\Sigma' \cup \mathfrak{bd}(\mathcal{R}) \cup \{\beta\}$, dado que incluyendo $\beta$, $\lambda$ no logra ser efectivamente alterada. Por lo tanto, retención de núcleo es verificado.

Finalmente, por **uniformidad**, si $\mathcal{R}_1 \dashv\Vdash \mathcal{R}_2$ es verificado, por el Lema 4.3.5 ambos árboles $\mathcal{T}(\mathcal{R}_1) \in \mathbb{T}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_1))}$ y $\mathcal{T}(\mathcal{R}_2) \in \mathbb{T}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_2))}$ son estrictamente equivalentes, luego para culquier línea $\lambda \in \mathcal{T}(\mathcal{R}_1)$ hay una línea $\lambda' \in \mathcal{T}(\mathcal{R}_2)$ estrictamente equivalente. Más aún, para cualquier argumento $\mathcal{B}_1 \in \lambda$ sabemos que existe un argumento $\mathcal{B}_2 \in \lambda'$ tal que $\mathcal{B}_1 \dashv\Vdash \mathcal{B}_2$. Por la Definición 4.3.6, sabemos que $\sigma(\mathcal{T}(\mathcal{R}_1)) \cap \mathfrak{bd}(\mathcal{B}_1) = \sigma(\mathcal{T}(\mathcal{R}_2)) \cap \mathfrak{bd}(\mathcal{B}_2)$, y dado que esto es válido para todo argumento en cualquiera de los dos árboles, también sabemos que $\sigma(\mathcal{T}(\mathcal{R}_1)) = \sigma(\mathcal{T}(\mathcal{R}_2))$. Por la Definición 4.2.5, la incisión superior en una línea $\lambda$ aparece sobre un argumento en $\lambda^{-}$ (lo cual excluye al argumento raíz), luego sabemos que $\sigma(\mathcal{T}(\mathcal{R}_1)) \subseteq \Sigma$ y $\sigma(\mathcal{T}(\mathcal{R}_2)) \subseteq \Sigma$. Finalmente, por la Definición 4.2.6, tenemos que $\Sigma *^{\omega} \mathcal{R}_1 = (\Sigma \cup \mathfrak{bd}(\mathcal{R}_1)) \setminus \sigma(\mathcal{T}(\mathcal{R}_1))$, y $\Sigma *^{\omega} \mathcal{R}_2 = (\Sigma \cup \mathfrak{bd}(\mathcal{R}_2)) \setminus \sigma(\mathcal{T}(\mathcal{R}_2))$. Y dado que sabemos que ambas incisiones toman la misma creencia de $\Sigma$, entonces tenemos que $\Sigma \cap \Sigma *^{\omega} \mathcal{R}_1 = \Sigma \cap \Sigma *^{\omega} \mathcal{R}_2$ es verificado.

**(Postulados a construcción)** Supongamos que tenemos una operación $*^{\omega}$ satisfaciendo los cinco postulados para la revisión de argumentos. Necesitamos mostrar que existe una función de incisión suave $\sigma$ tal que $\Sigma \cup \mathfrak{bd}(\mathcal{R}) \setminus \sigma(\mathcal{T}(\mathcal{R})) = \Sigma *^{\omega} \mathcal{R}$.

Primero, definimos $\sigma$ como: $\sigma(\mathcal{T}(\mathcal{R})) = \Sigma \setminus \Sigma *^{\omega} \mathcal{R}$, donde $\mathcal{T}(\mathcal{R}) \in \mathbb{T}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}))}$. Ahora, necesitamos mostrar que $\sigma$ es una función de incisión de acuerdo a la Definición 4.2.5 la cual es suave. Por lo tanto, necesitamos mostrar que:

1. $\sigma$ es una función mapeando árboles a conjuntos de fórmulas,
2. Las condiciones en la Definición 4.2.5 se verifican para $\sigma$, y
3. $\sigma$ es una función de incisión suave de acuerdo a la Definición 4.3.6.

Para 1, asumimos dos árboles de dialéctica $\mathcal{T}(\mathcal{R}_1) \in \mathbb{T}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_1))}$ y $\mathcal{T}(\mathcal{R}_2) \in \mathbb{T}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_2))}$. Si $\mathcal{T}(\mathcal{R}_1) = \mathcal{T}(\mathcal{R}_2)$ entonces necesitamos mostrar que $\sigma(\mathcal{T}(\mathcal{R}_1)) = \sigma(\mathcal{T}(\mathcal{R}_2))$. Dado que ambos árboles son iguales sabemos que tienen exactamente las mismas líneas de argumentación. Dado que todas las líneas comienzan a partir de la raíz, esto implica que $\mathcal{R}_1 = \mathcal{R}_2$ y luego $\mathcal{R}_1 \dashv\Vdash \mathcal{R}_2$. Por **uniformidad** tenemos que $\Sigma \cap \Sigma *^{\omega} \mathcal{R}_1 = \Sigma \cap \Sigma *^{\omega} \mathcal{R}_2$, y entonces $\Sigma \setminus \Sigma *^{\omega} \mathcal{R}_1 = \Sigma \setminus \Sigma *^{\omega} \mathcal{R}_2$. Finalmente, por la definición de $\sigma$ tenemos que $\sigma(\mathcal{T}(\mathcal{R}_1)) = \sigma(\mathcal{T}(\mathcal{R}_2))$.

Para 2 necesitamos mostrar las condiciones 1, 2, y 3, de la Definición 4.2.5. Para la condición 1, si $\sigma(\mathcal{T}(\mathcal{R})) = \emptyset$ entonces necesitamos mostrar que no existe línea de ataque en $\mathcal{T}(\mathcal{R}) \in \mathbb{T}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}))}$. Por la definición de $\sigma$ tenemos que $\Sigma \setminus \Sigma *^{\omega} \mathcal{R} = \emptyset$, entonces se verifica que $\Sigma \subseteq \Sigma *^{\omega} \mathcal{R}$. Por **éxito** y la Proposición A.2.5, sabemos que $\mathfrak{bd}(\mathcal{R}) \subseteq \Sigma *^{\omega} \mathcal{R}$, entonces $\Sigma \cup \mathfrak{bd}(\mathcal{R}) \subseteq \Sigma *^{\omega} \mathcal{R}$. Luego, por **inclusión** tenemos que $\Sigma *^{\omega} \mathcal{R} = \Sigma \cup \mathfrak{bd}(\mathcal{R})$. Esto implica que $\mathbb{T}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}))} = \mathbb{T}_{(\Sigma *^{\omega} \mathcal{R})}$, y entonces $\mathcal{T}(\mathcal{R}) \in \mathbb{T}_{(\Sigma *^{\omega} \mathcal{R})}$ se verifica. Finalmente, por **éxito** sabemos que $\mathcal{R}$ es garantizado en $\Sigma *^{\omega} \mathcal{R}$, y por el Teorema 4.2.2 concluimos que $\mathcal{T}(\mathcal{R})$ no tiene líneas de ataque. Para el sentido contrario, si no existe lína de ataque en $\mathcal{T}(\mathcal{R}) \in \mathbb{T}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}))}$ entonces necesitamos mostrar que $\sigma(\mathcal{T}(\mathcal{R})) = \emptyset$. Luego, ninguna línea en $\mathcal{T}(\mathcal{R})$ tiene su segundo argumento no-derrotado, esto es, todo derrotador de la raíz está derrotado (marcado como $D$). Por lo tanto, $\mathcal{R}$ es garantizado en $\Sigma \cup \mathfrak{bd}(\mathcal{R})$, y por **vacuidad** tenemos que $\Sigma \cup \mathfrak{bd}(\mathcal{R}) \subseteq \Sigma *^{\omega} \mathcal{R}$ es verificado. Finalmente, tenemos que $\Sigma \subseteq \Sigma *^{\omega} \mathcal{R}$, y por la definición de $\sigma$ adoptada en la hipótesis, $\sigma(\mathcal{T}(\mathcal{R})) = \emptyset$ se verifica.

Para la condición 2 en la Definición 4.2.5, para cualquier $\lambda \in \mathcal{T}(\mathcal{R})$ sólo necesitamos considerar dos casos: o bien $\lambda$ es una línea de ataque (condición $2a$), ó $\lambda$ puede tornarse en atacante dada la existencia de incisiones colaterales $\Psi = \sigma(\mathcal{T}(\mathcal{R})) \cap \mathfrak{bd}(\mathcal{C}) \neq \emptyset$ sobre un argumento $\mathcal{C} \in \lambda$, tal que $\lambda^{\uparrow}(\mathcal{C})$ conluye como línea de ataque en el árbol hipotético $\mathcal{H}(\mathcal{R}, \Psi)$ (condición $2b$). En ambos casos, esto significa que $\lambda$ podría amenazar el estado de garantía de la raíz $\mathcal{R}$ en el árbol $\mathcal{T}(\mathcal{R})$. Necesitamos mostrar que existe un argumento $\mathcal{B} \in \lambda$ tal que $\mathcal{B} \in \lambda^{-}$ y $\mathcal{B}$ es el argumento incidido superiormente en $\lambda$. Por *reductio ad absurdum*, es claro que si $\mathcal{B} \notin \lambda^{-}$ es el argumento incidido superiormente, por la

Proposición 4.2.3, $\lambda^{\uparrow}(\mathcal{B})$ se torna a línea de ataque (sólo bajo las condiciones $2a$ ó $2b$), y dado que no existe otra incisión sobre otro argumento ubicado por encima de $\mathcal{B}$ en $\lambda$ (argumento icidido superiormente), entonces el árbol resultante tendrá al menos una línea de ataque. Luego, por **éxito** sabemos que $\mathcal{R}$ es garantizado y por el Teorema 4.2.2, sabemos que esto sólo es posible a partir de un árbol de dialéctica sin líneas de ataque. Luego, alcanzamos el absurdo, y por lo tanto la condición 2 en la Definición 4.2.5 es verificada.

Para la condición 3 en la Definición 4.2.5, sea $\beta \in \sigma(\mathcal{T}(\mathcal{R}))$, entonces por la equivalencia para $\sigma$ adoptada como hipótesis en este teorema, sabemos que $\beta \in \Sigma \setminus (\Sigma *^{\omega} \mathcal{R})$. Por **retención de núcleo** existe alguna $\Sigma' \subseteq \Sigma$ tal que $\mathcal{R}$ es garantizado en $\Sigma' \cup \mathfrak{bd}(\mathcal{R})$ pero $\mathcal{R}$ no es garantizado en $\Sigma' \cup \mathfrak{bd}(\mathcal{R}) \cup \{\beta\}$. Por lo tanto, el árbol enraizado en $\mathcal{R}$ de $\mathbb{T}_{(\Sigma' \cup \mathfrak{bd}(\mathcal{R}) \cup \{\beta\})}$ incluye alguna línea de ataque $\lambda$, y mediante la remoción de $\beta$ sabemos que provocamos una alteración efectiva de tal $\lambda$, dado que $\mathcal{R}$ es garantizado en $\Sigma' \cup \mathfrak{bd}(\mathcal{R})$. Luego, por la Proposición 4.2.3, sabemos existe algún $\mathcal{B} \in \lambda^{-}$, tal que $\beta \in \mathfrak{bd}(\mathcal{B})$.

Para 3, asumimos un par de argumentos externos $\mathcal{R}_1 \in \mathbb{E}_{\Sigma}$, y $\mathcal{R}_2 \in \mathbb{E}_{\Sigma}$, tal que $\mathcal{R}_1 \dashv\Vdash \mathcal{R}_2$. Por la Proposición A.2.6, sabemos que $\mathbb{A}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_1))} \dashv\Vdash \mathbb{A}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_2))}$. Esto implica que para cualquier argumento $\mathcal{B}_1 \in \mathbb{A}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_1))}$, sabemos existe $\mathcal{B}_2 \in \mathbb{A}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_2))}$ tal que $\mathcal{B}_1 \dashv\Vdash \mathcal{B}_2$. Por el Lema 4.3.5 ambos árboles $\mathcal{T}(\mathcal{R}_1) \in \mathbb{T}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_1))}$ y $\mathcal{T}(\mathcal{R}_2) \in \mathbb{T}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_2))}$ se saben son estrictamente equivalentes, por lo tanto la función de incisión será aplicada sobre árboles estrictamente equivalentes. Asumamos cualquier nodo $\mathcal{B}_1$ en $\mathcal{T}(\mathcal{R}_1)$, si $\mathfrak{bd}(\mathcal{B}_1) \cap \mathfrak{bd}(\mathcal{R}_1) = \emptyset$, por la Proposición A.2.8 sabemos que $\mathcal{B}_1$ está también ubicado en la misma posición en el contexto de $\mathcal{T}(\mathcal{R}_2)$. Por otro lado, si $\mathfrak{bd}(\mathcal{B}_1) \cap \mathfrak{bd}(\mathcal{R}_1) \neq \emptyset$ entonces sabemos que existe un nodo $\mathcal{B}_2$ en $\mathcal{T}(\mathcal{R}_2)$ tal que $\mathcal{B}_1 \dashv\Vdash \mathcal{B}_2$ se verifica, pero $\mathcal{B}_1 \neq \mathcal{B}_2$. Pero además, $\mathfrak{bd}(\mathcal{B}_1) \setminus \mathfrak{bd}(\mathcal{R}_1) = \mathfrak{bd}(\mathcal{B}_2) \setminus \mathfrak{bd}(\mathcal{R}_2)$. Por **éxito** y la Proposición A.2.5, tenemos que $\mathfrak{bd}(\mathcal{R}_1) \subseteq \Sigma *^{\omega} \mathcal{R}_1$, y por la definición de $\sigma$ (en la hipótesis) $\Sigma *^{\omega} \mathcal{R}_1 = \Sigma \cup \mathfrak{bd}(\mathcal{R}_1) \setminus \sigma(\mathcal{T}(\mathcal{R}_1))$, sabemos que $\sigma(\mathcal{T}(\mathcal{R}_1)) \cap \mathfrak{bd}(\mathcal{R}_1) = \emptyset = \sigma(\mathcal{T}(\mathcal{R}_2)) \cap \mathfrak{bd}(\mathcal{R}_2)$. Luego, tenemos que $\sigma(\mathcal{T}(\mathcal{R}_1)) \cap \mathfrak{bd}(\mathcal{B}_1) \subseteq \Sigma$ y $\sigma(\mathcal{T}(\mathcal{R}_2)) \cap \mathfrak{bd}(\mathcal{B}_2) \subseteq \Sigma$. Finalmente, $\sigma(\mathcal{T}(\mathcal{R}_1)) \cap \mathfrak{bd}(\mathcal{B}_1) = \sigma(\mathcal{T}(\mathcal{R}_2)) \cap \mathfrak{bd}(\mathcal{B}_2)$ es verificado, para cualquier tal $\mathcal{B}_1$. □

**Proposición A.2.3** *Dado el árbol de dialéctica $\mathcal{T}(\mathcal{R}) \in \mathbb{T}_{\Sigma}$, si $\lambda \in \mathcal{T}(\mathcal{R})$ es atacante entonces $\lambda$ tiene logitud par.*

<u>*Demostración:*</u> Por *reductio ad absurdum*, si asumimos una línea de ataque $\lambda$ de logitud impar, su hoja será un argumento pro. Por el cirterio de marcado adoptado, sabemos que

toda hoja es marcada como $U$ dado que no tiene derrotadores. Por la Definición 4.2.1, sabemos que una línea de ataque tiene sus argumentos pro marcados como $D$. Esto implica que $\lambda$ no es de ataque, contrario a la hipótesis. $\square$

Se evita que la incisión afecte al argumento raíz dado que buscamos su garantía (revisión priorizada).

**Proposición A.2.4** *Dada una KB* $\Sigma \subseteq \mathcal{L}$*, si "*$\sigma$*" es una incisión global entonces* $\sigma(\mathcal{T}(\mathcal{R})) \cap \mathfrak{bd}(\mathcal{R}) = \emptyset$ *se verifica para cualquier* $\mathcal{R} \in \mathbb{A}_\Sigma$*.*

*Demostración:* Por la Definición 4.2.5 sabemos que el argumento de uicación superior en cualquier línea $\lambda \in \mathcal{T}(\mathcal{R})$ que puede ser incidido es un argumento con. Dado que $\mathcal{R}$ es el argumento raíz, sabemos que $\mathcal{R}$ es un argumento pro. Luego, $\sigma(\mathcal{T}(\mathcal{R})) \cap \mathfrak{bd}(\mathcal{R}) = \emptyset$ es verificado. $\square$

**Proposición A.2.5** *Si* $*^\omega$ *satisface éxito entonces* $\mathfrak{bd}(\mathcal{R}) \subseteq \Sigma *^\omega \mathcal{R}$*.*

*Demostración:* Por *reductio ad absurdum*, si $\mathfrak{bd}(\mathcal{R}) \subseteq \Sigma *^\omega \mathcal{R}$ no es verificado entonces $\mathcal{R}$ no podría ser formado en $\Sigma *^\omega \mathcal{R}$ y por lo tanto $\mathcal{R}$ no podría ser garantizado en $\Sigma *^\omega \mathcal{R}$, contrario a éxito. $\square$

**Proposición A.2.6** *Dado* $\Sigma \subseteq \mathcal{L}$*, si* $\mathcal{R}_1 \dashv\Vdash \mathcal{R}_2$ *entonces* $\mathbb{A}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_1))} \dashv\Vdash \mathbb{A}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_2))}$*, para cualquier* $\{\mathcal{R}_1, \mathcal{R}_2\} \subseteq \mathbb{E}_\Sigma$*.*

*Demostración:* Necesitamos mostrar que para cualquier $\mathcal{B} \in \mathbb{A}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_1))}$ existe un $\mathcal{C} \in \mathbb{A}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_2))}$ tal que $\mathcal{B} \dashv\Vdash \mathcal{C}$. Es claro que $\mathbb{A}_\Sigma \subseteq (\mathbb{A}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_1))} \cap \mathbb{A}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_2))})$, por lo tanto el problema es reducido a la prueba de que para todo argumento $\mathcal{B} \in \mathbb{A}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_1))}$ tal que $\mathcal{B} \notin \mathbb{A}_\Sigma$ existe un $\mathcal{C} \in \mathbb{A}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_2))}$ tal que $\mathcal{C} \notin \mathbb{A}_\Sigma$ y $\mathcal{B} \dashv\Vdash \mathcal{C}$. Asumiendo $\mathcal{B} \in \mathbb{A}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_1))}$ y $\mathcal{B} \notin \mathbb{A}_\Sigma$, $\mathcal{B}$ no necesariamente contiene parte de $\mathcal{R}_1$. Luego, si $\mathfrak{bd}(\mathcal{B}) \cap \mathfrak{bd}(\mathcal{R}_1) = X$, dado que $X \subseteq \mathfrak{bd}(\mathcal{R}_1)$ y $\mathcal{R}_1 \dashv\Vdash \mathcal{R}_2$, existe un $Y \subseteq \mathfrak{bd}(\mathcal{R}_2)$ tal que $X$ *sssi* $Y$. Entonces existe un argumento $\mathcal{C} \in \mathbb{A}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_2))}$ con $\mathfrak{bd}(\mathcal{C}) = (\mathfrak{bd}(\mathcal{B}) \setminus X) \cup Y$ tal que $\mathfrak{cl}(\mathcal{B}) = \mathfrak{cl}(\mathcal{C})$ y $\mathcal{B} \dashv\Vdash \mathcal{C}$. Finalmente, $\mathbb{A}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_1))} \dashv\Vdash \mathbb{A}_{(\Sigma \cup \mathfrak{bd}(\mathcal{R}_2))}$ es verificado. $\square$

La proposición anterior muestra que dados dos argumentos externos, $\mathcal{R}_1$ y $\mathcal{R}_2$, estrictamente equivalentes, mediante la incorporación de uno u otro a la KB generamos conjuntos de argumentos estrictamente equivalentes, esto es $\mathbb{A}_{(\Sigma\cup\mathfrak{bd}(\mathcal{R}_1))} \dashv\Vdash \mathbb{A}_{(\Sigma\cup\mathfrak{bd}(\mathcal{R}_2))}$. A continuación mostramos que los conjuntos de derrotadores para cualquier par de argumentos estrictamente equivalentes de $\mathbb{A}_{(\Sigma\cup\mathfrak{bd}(\mathcal{R}_1))}$ y $\mathbb{A}_{(\Sigma\cup\mathfrak{bd}(\mathcal{R}_2))}$, también serán estrictamente equivalentes.

**Proposición A.2.7** *Dados dos conjuntos de argumentos $\mathbb{A}_\Sigma$ y $\mathbb{A}_{\Sigma'}$, y dos argumentos $\mathcal{B} \in \mathbb{A}_\Sigma$ y $\mathcal{C} \in \mathbb{A}_{\Sigma'}$, si $\mathbb{A}_\Sigma \dashv\Vdash \mathbb{A}_{\Sigma'}$ y $\mathcal{B} \dashv\Vdash \mathcal{C}$ entonces para todo derrotador $\mathcal{D} \in \mathbb{A}_\Sigma$ de $\mathcal{B}$ existe un derrotador $\mathcal{D}' \in \mathbb{A}_{\Sigma'}$ de $\mathcal{C}$ tal que $\mathcal{D} \dashv\Vdash \mathcal{D}'$.*

<u>*Demostración:*</u> Dado que $\mathcal{D} \hookrightarrow \mathcal{B}$ y $\mathcal{B} \dashv\Vdash \mathcal{C}$ sabemos que $\mathcal{D} \hookrightarrow \mathcal{C}$. Por la Definición A.2.1 y $\mathbb{A}_\Sigma \dashv\Vdash \mathbb{A}_{\Sigma'}$ sabemos que existe un argumento $\mathcal{D}' \in \mathbb{A}_{\Sigma'}$ tal que $\mathcal{D} \dashv\Vdash \mathcal{D}'$. Luego, $\mathcal{D}' \hookrightarrow \mathcal{C}$ se verifica, o equivalentemente, $\mathcal{D}'$ derrota a $\mathcal{C}$. □

**Proposición A.2.8** *Para cualquier cse $\Psi \subseteq \mathbb{A}_\Sigma$, cualquier argumento $\mathcal{C} \in \Psi$, $\mathcal{B} \in \mathbb{A}_\Sigma$, y $\mathcal{R} \in \mathbb{A}_\Sigma$, y un árbol $\mathcal{T}(\mathcal{R}) \in \mathbb{T}_\Sigma$; si existe una línea $\lambda \in \mathcal{T}(\mathcal{R})$ tal que $\lambda = [\mathcal{R}, \ldots, \mathcal{B}, \mathcal{C}, \ldots]$ entonces para todo $\mathcal{C}' \in \Psi$ existe una línea $\lambda' \in \mathcal{T}(\mathcal{R})$ tal que $\lambda' = [\mathcal{R}, \ldots, \mathcal{B}, \mathcal{C}', \ldots]$ y ambos $\lambda$ y $\lambda'$ son estrictamente equivalentes.*

<u>*Demostración:*</u> Dada $\lambda = [\mathcal{R}, \ldots, \mathcal{B}, \mathcal{C}, \ldots]$ donde $\mathcal{C} \in \Psi$, asumimos por *reductio ad absurdum* que no habrá $\lambda' = [\mathcal{R}, \ldots, \mathcal{B}, \mathcal{C}', \ldots]$ para algún otro $\mathcal{C}' \in \Psi$. Dado que $\lambda^{\uparrow}(\mathcal{C}) = \lambda'^{\uparrow}(\mathcal{C}')$ se verifica, la única opción que tenemos es que $\mathcal{C}'$ no sea un contra-argumento de $\mathcal{B}$. Por la Definición A.2.2 sabemos que $\mathcal{C} \dashv\Vdash \mathcal{C}'$ y por la Definición 4.3.3 es claro que $\mathcal{C}'$ contra-argumenta a $\mathcal{B}$ dado que $\mathcal{C}$ es contra-argumento de $\mathcal{B}$. Si $\lambda$ es aceptable, así también lo será $\lambda'$. Luego, por la Definición 3.2.7, $\lambda'$ existe y es una línea de $\mathcal{T}(\mathcal{R})$. Finalmente, dado que $\mathcal{C} \dashv\Vdash \mathcal{C}'$, por la Proposición A.2.7 sabemos que para cualquier derrotador $\mathcal{D}$ de $\mathcal{C}$ hay un derrotador estrictamente equivalente $\mathcal{D}'$ de $\mathcal{C}'$ (*i.e.*, $\mathcal{D} \dashv\Vdash \mathcal{D}'$), y por lo tanto es claro que ambos $\lambda$ y $\lambda'$ son líneas estrictamente equivalentes. □

**Teorema 4.4.7** *Dada una ontología $\Sigma \subseteq \mathcal{L}$, si $\Sigma$ es coherente y consistente entonces para cualquier consulta $\alpha \in \mathcal{L}_{\mathtt{cl}} \cup \{q(\bar{x})\}$ se verifica $\Sigma \models \alpha$ sssi $\Sigma \approx\!\!\!\!|\;\, \alpha$.*

<u>*Demostración:*</u> La prueba de este teorema es obtenida en forma directa a partir de la especificación del marco argumentativo definido en la Sección 4.4, junto con la demostración del Teorema 3.3.20. □

**Teorema 4.4.8** *El cálculo de todos los derrotadores de un argumento $\mathcal{ALC}$ dado se encuentra en PSPACE.*

<u>*Demostración:*</u> La prueba esquemática de este teorema está dada en la misma Sección 4.4.1. Su prueba formal depende de la especificación completa del algoritmo de construcción de derrotadores, propuesto como trabajo a futuro. □

## A.3. Consistencia de la Semántica basada en Árboles de Dialéctica

En esta sección nos concentraremos en demostrar que las semánticas argumentativas adoptadas, basadas en árboles de dialéctica siempre determinan un conjunto consistente de argumentos garantizados. A tal fin, introduciremos la noción de *línea alternante*, para identificar a las líneas de argumentación cuyo marcado, en el contexto de un árbol de dialéctica al cual pertenecen, es una alternacia perfecta de marcas $U$ y $D$, sin importar la marca de la raíz.

**Definición A.3.1 (Líneas Alternantes)** *Una línea de argumentación $\lambda \in \mathcal{T}(\mathcal{R})$, con $\mathcal{T}(\mathcal{R}) \in \mathbf{T}_\Sigma$, es llamada **alternante** si para todo par $\mathcal{B} \in \lambda$ y $\mathcal{C} \in \lambda$ se verifica $\mathfrak{mark}(\mathcal{B}, \lambda, \mathcal{T}(\mathcal{R})) = \mathfrak{mark}(\mathcal{C}, \lambda, \mathcal{T}(\mathcal{R}))$ sssi o bien $\{\mathcal{B}, \mathcal{C}\} \subseteq \lambda^-$ ó $\{\mathcal{B}, \mathcal{C}\} \subseteq \lambda^+$. Cuando $\mathcal{T}(\mathcal{R})$ es garantizante, $\lambda$ es referida como **línea garantizante-alternante**.*

**Observación A.3.2** *Una línea es de ataque sssi es no-garantizante-alternante.*

**Proposición A.3.3** *Dada $\Sigma \subseteq \mathcal{L}$, y el árbol no-garantizante $\mathcal{T}(\mathcal{R}) \in \mathbb{T}_\Sigma$; siempre existe una línea no-garantizante-alternante $\lambda \in \mathcal{T}(\mathcal{R})$.*

<u>*Demostración:*</u> Asumamos un árbol de dialéctica $\mathcal{T}(\mathcal{R})$ no-garantizante enraizado en un argumento $\mathcal{R} \in \mathbb{A}_\Sigma$, tenemos que $\mathcal{R}$ está marcado como $D$. A partir del criterio de marcado adoptado definido en la página 21, sabemos que (*) cualquier argumento marcado como $D$ tiene al menos un hijo el cual es no-derrotado. Este es el caso de $\mathcal{R}$ que asumimos tiene un hijo $\mathcal{B}$ marcado como $U$. Identificaremos como $\lambda$ a la línea enraizada en $\mathcal{R}$ seguida de $\mathcal{B}$. Si $\mathcal{B}$ es la hoja de $\lambda$ entonces tenemos una línea alternante, mientras que si $\mathcal{B}$ es un nodo interno, la única opción para su hijo $\mathcal{C}$ es estar marcado como $D$, lo cual no deja en la misma sitación del argumento raíz $\mathcal{R}$. Luego, sabemos que existe un argumento

no-derrotado el cual es hijo de $\mathcal{C}$. Finalmente, siguiendo la misma construcción a partir de (*), tenemos que siempre existirá una línea $\lambda \in \mathcal{T}(\mathcal{R})$ no-garantizante-alternante. $\square$

Para la siguiente proposición, haremos uso de la operación de *composición* "$\circ$" definida entre líneas y argumentos como $\circ : \mathbb{A}_\Sigma \times \mathbf{L}_\Sigma \rightarrow \mathbf{L}_\Sigma$ y sobrecargada como $\circ : \mathbf{L}_\Sigma \times \mathbb{A}_\Sigma \rightarrow \mathbf{L}_\Sigma$, tal que para cualquier $\lambda \in \mathbf{L}_\Sigma$, si $\lambda = [\mathcal{B}_1, \ldots, \mathcal{B}_n]$ entonces $\lambda = \mathcal{B}_1 \circ [\mathcal{B}_2, \ldots, \mathcal{B}_n] = [\mathcal{B}_1, \ldots, \mathcal{B}_{n-1}] \circ \mathcal{B}_n$. Observe que cualquier subsecuencia de una línea aceptable es también aceptable, por lo tanto, dado que $\lambda \in \mathbf{L}_\Sigma$, ambas $[\mathcal{B}_2, \ldots, \mathcal{B}_n]$ y $[\mathcal{B}_1, \ldots, \mathcal{B}_{n-1}]$, son líneas aceptables y luego, ambas estarán contenidas en $\mathbf{L}_\Sigma$.

**Proposición A.3.4** *Dada $\Sigma \subseteq \mathcal{L}$ y un árbol de dialéctica garantizante $\mathcal{T}(\mathcal{R}) \in \mathbb{T}_\Sigma$. Para toda $\lambda \in \mathcal{T}(\mathcal{R})$ tal que $\lambda$ es alternante, si $\lambda = \mathcal{R} \circ \lambda'$ y $\mathcal{B}$ es la raíz de $\lambda'$ entonces $\lambda'$ es una línea de $\mathcal{T}(\mathcal{B}) \in \mathbb{T}_\Sigma$ y $\lambda' \in \mathcal{T}(\mathcal{B})$ es línea de ataque.*

*Demostración:* Primeramente, mostraremos que para cualquier argumento no-derrotado $\mathcal{C} \in \lambda$ tal que $\mathcal{C} \neq \mathcal{R}$, $\mathcal{C} \in \lambda'$ no puede ser derrotado en el contexto de $\mathcal{T}(\mathcal{B})$. Observe que, dado que $\lambda$ es una línea garantizante-alternante, $\mathcal{C} \in \lambda^+$ se verifica.

Asumamos que existe algún $\mathcal{D} \in \mathbb{A}_\Sigma$ derrotando a $\mathcal{C}$, tal que para cualquier $\lambda'' \in \mathcal{T}(\mathcal{R})$, si $\mathcal{D} \in \lambda''$ entonces $\lambda''^{\uparrow}[\mathcal{D}] \neq \lambda^{\uparrow}[\mathcal{C}] \circ \mathcal{D}$. Esto significa que $\mathcal{D}$ no es hijo de $\mathcal{C} \in \lambda$ en $\mathcal{T}(\mathcal{R})$ dado que o bien tal $\mathcal{D}$ no existe, ó $\lambda^{\uparrow}[\mathcal{C}] \circ \mathcal{D}$ no es aceptable. Asumiendo la última opción (la primera deja a $\mathcal{C}$ no-derrotado en $\mathcal{T}(\mathcal{B})$), sabemos que $\lambda_{\mathcal{R}} = \lambda^{\uparrow}[\mathcal{C}] \circ \mathcal{D}$ violará alguna de las condiciones de aceptabilidad, (1) o (2) (como fue visto en la página 101). Mostraremos que $\lambda_{\mathcal{B}} = \lambda'^{\uparrow}[\mathcal{C}] \circ \mathcal{D}$ no es aceptable. Observe que $\mathcal{B} \in \lambda_{\mathcal{B}}^+$ y dado que $\mathcal{C} \in \lambda_{\mathcal{B}}^-$, también sabemos que $\mathcal{D} \in \lambda_{\mathcal{B}}^+$. Asumiendo (1) es violada para $\lambda_{\mathcal{R}}$, tenemos que $\mathcal{D}$ incluye el subconjunto de $\mathcal{R}$ derrotado por $\mathcal{B}$, y por lo tanto $\mathcal{B}$ contra-argumenta a $\mathcal{D}$. Esto violaría la condición de aceptabilidad (2) sobre $\lambda_{\mathcal{B}}$ dado que $\{\mathcal{B}, \mathcal{D}\} \subseteq \lambda_{\mathcal{B}}^+$. Por otro lado, asumiendo (2) es violado para $\lambda_{\mathcal{R}}$, tenemos que $\mathcal{D}$ contra-argumenta algún argumento con en $\lambda^{\uparrow}[\mathcal{C}]$ (observe que $\mathcal{D} \in \lambda_{\mathcal{R}}^-$), y dado que $\lambda_{\mathcal{R}}^- = \lambda_{\mathcal{B}}^+$, también tenemos que $\lambda_{\mathcal{B}}$ viola la condición (2). Luego, $\lambda_{\mathcal{B}}$ no es aceptable.

Claramente, todo argumento no-derrotado $\mathcal{C} \in \lambda$ concluye también no-derrotado en $\mathcal{C} \in \lambda'$. Y además, esto incluye a la hoja de $\lambda'$, la cual también es no-derrotada. Luego, es claro que $\lambda' \in \mathcal{T}(\mathcal{B})$, dado que $\lambda'$ es exhaustiva (dado que $\lambda$ se sabe aceptable y exhaustiva, es fácil de ver que la línea $\lambda'$ también es aceptable dado que las condiciones de aceptabilidad son más rígidas sobre $\lambda$ que sobre $\lambda'$). Finalmente, dado que $\lambda$ es garantizante-alternante, $\lambda'$ concluye siendo una línea de ataque en $\mathcal{T}(\mathcal{B})$. $\square$

**Lema A.3.5** *Dada $\Sigma \subseteq \mathcal{L}$, para cualquier árbol de dialéctica $\mathcal{T}(\mathcal{R}) \in \mathbb{T}_\Sigma$ garantizante y cualquier argumento $\mathcal{B} \in \mathbb{A}_\Sigma$, si $\mathcal{B}$ derrota a $\mathcal{R}$ entonces $\mathcal{T}(\mathcal{B}) \in \mathbb{T}_\Sigma$ es no-garantizante.*

*Demostración:* Directo por la Proposición A.3.3, Proposición A.3.4, y por el Teorema 4.2.2. □

**Lema A.3.6** *Dados dos argumentos $\mathcal{B} \in \mathbb{A}_\Sigma$ y $\mathcal{C} \in \mathbb{A}_\Sigma$, si $\mathcal{C}$ es un contra-argumento de $\mathcal{B}$ entonces existe un subargumento $\mathcal{D} \in \mathbb{A}_\Sigma$ de $\mathcal{C}$ tal que $\mathcal{D}$ es un undercut canónico de $\mathcal{B}$.*

*Demostración:* Dado que $\mathcal{C}$ es un contra-argumento de $\mathcal{B}$, por definición tenemos que $\{\mathfrak{cl}(\mathcal{C})\} \cup \mathfrak{bd}(\mathcal{B}) \models \perp$, y por la Definición 4.1.1, dado que $\mathfrak{bd}(\mathcal{C}) \models \mathfrak{cl}(\mathcal{C})$, tenemos que $\mathfrak{bd}(\mathcal{B}) \cup \mathfrak{bd}(\mathcal{C}) \models \perp$. Es claro que $\mathfrak{bd}(\mathcal{C}) \models \bigvee_{\beta \in \mathfrak{bd}(\mathcal{B})} \neg\beta$ es verificado. Por lo tanto, asumiendo $\mathfrak{bd}(\mathcal{B}) = \{\beta_1 \ldots, \beta_n\}$, tenemos que $\mathfrak{bd}(\mathcal{C}) \models \neg(\beta_1 \wedge \ldots \wedge \beta_n)$. Sea $\varphi$ una fórmula $\neg(\beta_1 \wedge \ldots \wedge \beta_n)$. Luego, existe un subconjunto minimal $\Psi \subseteq \mathfrak{bd}(\mathcal{C})$ tal que $\Psi \models \varphi$. Adicionalmente, dada la Definición 4.1.1, y $\Psi \subseteq \mathfrak{bd}(\mathcal{C})$, sabemos que $\Psi \not\models \perp$; y dado que $\Psi$ es minimal para $\varphi$, existe un argumento $\mathcal{D} \in \mathbb{A}_\Sigma$ tal que $\mathfrak{bd}(\mathcal{D}) = \Psi$ y $\mathfrak{cl}(\mathcal{D}) = \varphi$. Finalmente, es fácil de ver que $\mathcal{D}$ –el cual es subargumento de $\mathcal{C}$– es un undercut canónico de $\mathcal{B}$. □

**Teorema A.3.7** *Dada una KB $\Sigma \subseteq \mathcal{L}$, para cualquier par de argumentos $\mathcal{R} \in \mathbb{A}_\Sigma$ y $\mathcal{R}' \in \mathbb{A}_\Sigma$, si ambos $\mathcal{R}$ y $\mathcal{R}'$ son garantizados entonces no son conflictivos.*

*Demostración:* Por *reductio ad absurdum*, supongamos que $\mathcal{R}$ y $\mathcal{R}'$ son garantizados y conflictivos. Asumamos sin pérdida de generalidad que $\mathcal{R}$ derrota a $\mathcal{R}'$, y por el Lema A.3.6, supongamos también que $\mathcal{R}''$ es un undercut canónico de $\mathcal{R}'$ tal que $\mathcal{R}'' \subseteq \mathcal{R}$. Es claro que, si $\mathcal{R}$ es garantizado, así también lo es $\mathcal{R}''$. Luego, $[\mathcal{R}', \mathcal{R}'', \ldots] \in \mathcal{T}(\mathcal{R}')$ se verifica. Dado que $\mathcal{R}'$ es garantizado, por el Lema A.3.5 sabemos que $\mathcal{T}(\mathcal{R}'')$ es no-garantizante. Finalmente, sabemos que $\mathcal{R}$ es no-garantizado, lo cual es absurdo. □

# Bibliografía


[ACLS04] Amgoud, L., Cayrol, C., and Lagasquie-Schiex, M.-C. On the Bipolarity in Argumentation Frameworks. In *NMR* (2004), J. P. Delgrande and T. Schaub, Eds., pp. 1–9.

[ACLSL08] Amgoud, L., Cayrol, C., Lagasquie-Schiex, M.-C., and Livet, P. On Bipolarity in Argumentation Frameworks. *Int. J. Intell. Syst. 23*, 10 (2008), 1062–1093.

[AGM85] Alchourrón, C., Gärdenfors, P., and Makinson, D. On the Logic of Theory Change: Partial Meet Contraction and Revision Functions. *The Journal of Symbolic Logic 50* (1985), 510–530.

[AM85] Alchourrón, C., and Makinson, D. On the logic of theory change: Safe contraction. *Studia Logica*, 44 (1985), 405–422.

[AV09] Amgoud, L., and Vesic, S. On Revising Argumentation-Based Decision Systems. In *ECSQARU* (2009), pp. 71–82.

[Baa99] Baader, F. Logic-Based Knowledge Representation. In *Artif. Intell. Today*. 1999, pp. 13–41.

[Baa03] Baader, F. Terminological Cycles in a Description Logic with Existential Restrictions. In *IJCAI* (2003), pp. 325–330.

[BAGM99] Billington, D., Antoniou, G., Governatori, G., and Maher, M. Revising Nonmonotonic Theories: The Case of Defeasible Logic. *KI-99: Advances in Artificial Intelligence* (1999), 695–695.

[BC02] Bench-Capon, T. J. M. Value-based argumentation frameworks. In *NMR* (2002), S. Benferhat and E. Giunchiglia, Eds., pp. 443–454.

[BCD07a] Bench-Capon, T. J. M., and Dunne, P. E. Argumentation in Artificial Intelligence. *Artif. Intell. 171*, 10-15 (2007), 619–641.

[BCD07b] Bench-Capon, T. J. M., and Dunne, P. E. Argumentation in Artificial Intelligence. *Artif. Intell. 171*, 10-15 (2007), 619–641.

[BCGS10] Baroni, P., Cerutti, F., Giacomin, M., and Simari, G. R., Eds. *Computational Models of Argument: Proceedings of COMMA 2010* (Amsterdam, The Netherlands, The Netherlands, 2010), IOS Press.

[BCM+03] Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Patel-Schneider, P., Eds. *Description Logic Handbook: Theory, Implementation and Application.* Cambridge University Press, Cambridge, 2003.

[BCPTvdT08] Boella, G., Costa Perera, C. d., Tettamanzi, A., and van der Torre, L. Dung Argumentation and AGM Belief Revision (Position Statement). In *ArgMAS (Argumentation in Multi-Agent Systems)* (2008).

[BCS03] Bench-Capon, T. J. M., and Sartor, G. A Model of Legal Reasoning with Cases Incorporating Theories and Values. *Artif. Intell. 150*, 1-2 (2003), 97–143.

[BdCPTvdT08] Boella, G., da Costa Pereira, C., Tettamanzi, A., and van der Torre, L. Making Others Believe What They Want. In *IFIP AI (Artificial Intelligence in Theory and Practice II)* (2008), M. Bramer, Ed., vol. 276 of *IFIP*, Springer, pp. 215–224.

[BDP95] Benferhat, S., Dubois, D., and Prade, H. How to Infer from Inconsistent Beliefs without Revising. In *Proceedings of IJCAI'95* (1995), pp. 1449–1455.

[BFL85] Brachman, R. J., Fikes, R. E., and Levesque, H. J. KRYPTON: A Functional Approach to Knowledge Representation. In *Readings in Knowledge Representation*, R. J. Brachman and H. J. Levesque, Eds. Kaufmann, 1985, pp. 411–429.

[BFT95] Bresciani, P., Franconi, E., and Tessaris, S. Implementing and Testing Expressive Description Logics: A Preliminary Report. In *DL* (Roma, IT, 1995), pp. 131–139.

[BG02] Baroni, P., and Giacomin, M. Argumentation through a Distributed Self-stabilizing Approach. *J. Exp. Theor. Artif. Intell. 14*, 4 (2002), 273–301.

[BG07] Baroni, P., and Giacomin, M. On Principle-Based Evaluation of Extension-Based Argumentation Semantics. *Artificial Intelligence 171*, 10-15 (2007), 675–700.

[BH91] Baader, F., and Hollunder, B. A Terminological Knowledge Representation System with Complete Inference Algorithms. In *Proceedings of the First International Workshop on Processing Declarative Knowledge* (Kaiserslautern (Germany), 1991), vol. 572, Springer–Verlag, pp. 67–85.

[BH01] Besnard, P., and Hunter, A. A Logic-based Theory of Deductive Arguments. *Artif. Intell. 128*, 1-2 (2001), 203–235.

[BH05] Besnard, P., and Hunter, A. Practical First-Order Argumentation. In *AAAI* (2005), pp. 590–595.

[BH06] Besnard, P., and Hunter, A. Knowledgebase Compilation for Efficient Logical Argumentation. In Doherty et al. (DMW06), pp. 123–133.

[BH08] Besnard, P., and Hunter, A. *Elements of Argumentation*. The MIT Press, 2008.

[BHS07] Baader, F., Horrocks, I., and Sattler, U. *Description Logics*. 2007, ch. 3.

[Bie08] Bienvenu, M. Prime Implicate Normal Form for ALC Concepts. In *AAAI* (2008), pp. 412–417.

[Bie09] Bienvenu, M. Prime Implicates and Prime Implicants: From Propositional to Modal Logic. *J. Artif. Intell. Res. (JAIR) 36* (2009), 71–128.

[BKvdT09] Boella, G., Kaci, S., and van der Torre, L. Dynamics in Argumentation with Single Extensions: Abstraction Principles and the Grounded Extension. In *ECSQARU* (2009), pp. 107–118.

[BN02] Baader, F., and Nutt, W. Basic Description Logics. *In the Description Logic Handbook, Cambridge University Press* (2002), 47–100.

[Boc03] Bochman, A. Collective Argumentation and Disjunctive Logic Programming. *J. Log. Comput. 13*, 3 (2003), 405–428.

[Bor96] Borgida, A. On the relative expressiveness of description logics and predicate logics. *Artif. Intell. 82*, 1-2 (1996), 353–367.

[BPS07] Baader, F., Peñaloza, R., and Suntisrivaraporn, B. Pinpointing in the Description Logic EL. In *Description Logics* (2007).

[CA07] Caminada, M., and Amgoud, L. On the Evaluation of Argumentation Formalisms. *Artif. Intell. 171*, 5-6 (2007), 286–310.

[CDM01] Cayrol, C., Doutre, S., and Mengin, J. Dialectical Proof Theories for the Credulous Preferred Semantics of Argumentation Frameworks. In *ECSQARU* (2001), S. Benferhat and P. Besnard, Eds., vol. 2143 of *Lecture Notes in Computer Science*, Springer, pp. 668–679.

[CdSCLS10] Cayrol, C., de Saint-Cyr, F. D., and Lagasquie-Schiex, M.-C. Change in Abstract Argumentation Frameworks: Adding an Argument. *Journal of Artificial Intelligence Research (JAIR) 38* (2010), 49–84.

[CGL+07] Calvanese, D., Giacomo, G. D., Lembo, D., Lenzerini, M., and Rosati, R. Tractable Reasoning and Efficient Query Answering in Description Logics: The DL-Lite family. *JAR 39*, 3 (2007), 385–429.

[CGP00] Corcho, O., and Gómez-Pérez, A. A Roadmap to Ontology Specification Languages. In *EKAW '00: Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling and Management* (London, UK, 2000), Springer-Verlag, pp. 80–96.

[CLS05] Cayrol, C., and Lagasquie-Schiex, M.-C. On the Acceptability of Arguments in Bipolar Argumentation Frameworks. In *ECSQARU*

(2005), L. Godo, Ed., vol. 3571 of *Lecture Notes in Computer Science*, Springer, pp. 378–389.

[CLS10] Cayrol, C., and Lagasquie-Schiex, M.-C. Coalitions of Arguments: A tool for handling bipolar argumentation frameworks. *Int. J. Intell. Syst. 25*, 1 (2010), 83–109.

[CML00] Chesñevar, C., Maguitman, A., and Loui, R. Logical Models of Argument. *ACM Computing Surveys 32*, 4 (2000), 337–383.

[CS06] Chesñevar, C., and Simari, G. Modelling Inference in Argumentation through Labelled Deduction: Formalization and Logical Properties. *Logica Universalis 1*, 1 (2006), (in press).

[CS07] Chesñevar, C., and Simari, G. R. A Lattice-based Approach to Computing Warranted Belief in Skeptical Argumentation Frameworks. In *IJCAI* (2007), pp. 280–285.

[Dal88] Dalal, M. Investigations into a Theory of Knowledge Base Revision. In *AAAI* (1988), pp. 475–479.

[DM02] Darwiche, A., and Marquis, P. A Knowledge Compilation Map. *J. Artif. Intell. Res. (JAIR) 17* (2002), 229–264.

[DMW06] Doherty, P., Mylopoulos, J., and Welty, C. A., Eds. *Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning, Lake District of the United Kingdom, June 2-5, 2006* (2006), AAAI Press.

[DSTW08] Delgrande, J., Schaub, T., Tompits, H., and Woltran, S. Belief Revision of Logic Programs under Answer Set Semantics. In *Proceedings of the Eleventh International Conference on Principles of Knowledge Representation and Reasoning* (2008), pp. 411–421.

[Dun95] Dung, P. M. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning and Logic Programming and $n$-person Games. *Artif. Intell. 77* (1995), 321–357.

[FGO09] Furbach, U., Günther, H., and Obermaier, C. A Knowledge Compilation Technique for ALC Tboxes. In *FLAIRS Conference* (2009), H. C. Lane and H. W. Guesgen, Eds., AAAI Press.

[FHP+06] Flouris, G., Huang, Z., Pan, J., Plexousakis, D., and Wache, H. Inconsistencies, Negations and Changes in Ontologies. In *AAAI* (2006), pp. 1295–1300.

[FKIS02] Falappa, M., Kern-Isberner, G., and Simari, G. R. Explanations, Belief Revision and Defeasible Reasoning. *Artificial Intelligence Journal 141(1-2)* (2002), 1–28.

[FO07] Furbach, U., and Obermaier, C. Knowledge Compilation for Description Logics. In *KESE* (2007), J. Baumeister and D. Seipel, Eds., vol. 282 of *CEUR Workshop Proceedings*, CEUR-WS.org.

[Gä81] Gärdenfors, P. An Epistemic Approach to Conditionals. *American Philosophical Quarterly 18*, 3 (1981), 203–211.

[Gä88] Gärdenfors, P. Knowledge in Flux: Modelling the Dynamics of Epistemic States. *The MIT Press, Bradford Books, Cambridge, Massachusetts* (1988).

[GR95] Gärdenfors, P., and Rott, H. Belief Revision. In *Handbook of Logic in Artificial Intelligence and Logic Programming*, vol. 4. Oxford University Press, 1995.

[Gru93] Gruber, T. R. A Translation Approach to Portable Ontology Specifications. *Knowl. Acquis. 5*, 2 (June 1993), 199–220.

[GS04] García, A., and Simari, G. R. Defeasible Logic Programming: An Argumentative Approach. *TPLP 4*, 1-2 (2004), 95–138.

[Han93] Hansson, S. O. Reversing the Levi Identity. *Journal of Philosophical Logic 22*, 6 (1993), 637–669.

[Han94] Hansson, S. O. Kernel Contraction. *J. of Symbolic Logic 59* (1994), 845–859.

[Han97] Hansson, S. O. Semi-Revision. *Journal of Applied Non-Classical Logic 7* (1997), 151–175.

[Han99] Hansson, S. O. *A Textbook of Belief Dynamics: Theory Change and Database Updating. Springer*. 1999.

[Han06] Hansson, S. O. Ideal Worlds - Wishful Thinking in Deontic Logic. *Studia Logica 82*, 3 (2006), 329–336.

[HM01] Haarslev, V., and Moller, R. RACER System Description. *In Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR 2001) 2083* (2001), 701–705.

[Hor98] Horrocks, I. Using an Expressive Description Logic: FaCT or Fiction? *In Proc. of the 6th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'98)* (1998), 636–647.

[HvHH+05] Haase, P., van Harmelen, F., Huang, Z., Stuckenschmidt, H., and Sure, Y. A Framework for Handling Inconsistency in Changing Ontologies. In *ISWC* (2005), pp. 353–367.

[HvHtT05] Huang, Z., van Harmelen, F., and ten Teije, A. Reasoning with Inconsistent Ontologies. In *IJCAI* (2005), L. P. Kaelbling and A. Saffiotti, Eds., Professional Book Center, pp. 454–459.

[HW02] Hansson, S. O., and Wassermann, R. Local Change. *Studia Logica 70*, 1 (2002), 49–76.

[KT99] Kakas, A. C., and Toni, F. Computing Argumentation in Logic Programming. *J. Log. Comput. 9*, 4 (1999), 515–562.

[Lev77] Levi, I. Subjunctives, Dispositions, and Chances. *Synthèse 34* (1977), 423–455.

[Lev84] Levesque, H. J. Foundations of a Functional Approach to Knowledge Representation. *Artif. Intell. 23*, 2 (1984), 155–212.

[Lev91] Levi, I. *The Fixation of Belief and its Undoing*. Cambridge University Press, 1991.

[LSW01] Lutz, C., Sattler, U., and Wolter, F. Description Logics and the Two-Variable Fragment. In *Description Logics* (2001).

[Mak87] Makinson, D. On the Status of the Postulate of Recovery in the Logic of Theory Change. *Journal of Philosophical Logic 16* (1987), 383–394.

[Mak97] Makinson, D. Screened Revision. *Theoria 63* (1997), 14–23.

[MF07] Moguillansky, M. O., and Falappa, M. A. A Non-monotonic Description Logics Model for Merging Terminologies. *Revista Iberoamericana de Inteligencia Artificial (AEPIA), ISSN 1137-3601* (2007), 35: 77–88.

[MFS08] Moguillansky, M. O., Falappa, M. A., and Simari, G. R. Model-Based Contractions for Description Logics. In *NMR* (2008), pp. 34–42.

[MGS07] Martínez, D. C., García, A. J., and Simari, G. R. Modelling Well-Structured Argumentation Lines. In *IJCAI* (2007), M. M. Veloso, Ed., pp. 465–470.

[MLB05] Meyer, T., Lee, K., and Booth, R. Knowledge Integration for Description Logics. In *AAAI* (2005), pp. 645–650.

[MLBP06] Meyer, T., Lee, K., Booth, R., and Pan, J. Z. Finding Maximally Satisfiable Terminologies for the Description Logic ALC. In *AAAI* (2006).

[Mor75] Mortimer, M. On Languages with Two Variables. *Zeitschr. f. math. Logik und Grundlagen d. Math. 21* (1975), 135–140.

[MRF08a] Moguillansky, M. O., Rotstein, N. D., and Falappa, M. A. A Theoretical Model to Handle Ontology Debugging & Change Through Argumentation. In *IWOD* (2008), pp. 29–42.

[MRF+08b] Moguillansky, M. O., Rotstein, N. D., Falappa, M. A., García, A. J., and Simari, G. R. Argument Theory Change Applied to Defeasible Logic Programming. In *AAAI* (2008), D. Fox and C. P. Gomes, Eds., AAAI Press, pp. 132–137.

[MRF+09] Moguillansky, M. O., Rotstein, N. D., Falappa, M. A., García, A. J., and Simari, G. R. Argument Theory Change Through Defeater Activation. In *CMNA* (2009), pp. 24–33.

[MRF10a] Moguillansky, M. O., Rotstein, N. D., and Falappa, M. A. Generalized Abstract Argumentation: A First-order Machinery towards Ontology Debugging. *Revista Iberoamericana de Inteligencia Artificial (AEPIA), ISSN 1988-3064* (2010), 46: 17–33.

[MRF+10b] Moguillansky, M. O., Rotstein, N. D., Falappa, M. A., García, A. J., and Simari, G. R. Argument Theory Change Through Defeater Activation. In Baroni et al. (BCGS10), pp. 359–366.

[MRFS09] Moguillansky, M. O., Rotstein, N. D., Falappa, M. A., and Simari, G. R. Generalized Abstract Argumentation: Handling Arguments in FOL Fragments. In *ECSQARU* (2009), C. Sossai and G. Chemello, Eds., vol. 5590 of *Lecture Notes in Computer Science*, Springer, pp. 144–155.

[MW09] Moguillansky, M. O., and Wassermann, R. Inconsistent-Tolerant DL-Lite Reasoning: An Argumentative Approach. In *ARCOE* (2009), pp. 7–9.

[MWF10] Moguillansky, M. O., Wassermann, R., and Falappa, M. A. An Argumentation Machinery to Reason over Inconsistent Ontologies. In *IBERAMIA* (2010), Á. F. K. Morales and G. R. Simari, Eds., vol. 6433 of *Lecture Notes in Computer Science*, Springer, pp. 100–109.

[Neb90] Nebel, B. *Reasoning and Revision in Hybrid Representation Systems*, vol. 422 of *Lecture Notes in Computer Science*. Springer, 1990.

[New80] Newell, A. The Knowledge Level (Presidential Address). *AI Magazine 2*, 2 (1980), 1–20, 33.

[NK04] Noy, N. F., and Klein, M. Ontology Evolution: Not the Same as Schema Evolution. *Knowl. Inf. Syst. 6*, 4 (2004), 428–440.

[NP06] Nielsen, S. H., and Parsons, S. A Generalization of Dung's Abstract Framework for Argumentation: Arguing with Sets of Attacking Arguments. In *ArgMAS* (2006), N. Maudet, S. Parsons, and I. Rahwan, Eds., vol. 4766 of *Lecture Notes in Computer Science*, Springer, pp. 54–73.

[PC06] Paglieri, F., and Castelfranchi, C. *The Toulmin Test: Framing argumentation within belief revision theories.* Berlin, Springer, 2006, pp. 359–377.

[PG00] Pollock, J. L., and Gillies, A. S. Belief Revision and Epistemology. *Synthese 122*, 1–2 (2000), 69–92.

[PS99] Patel-Schneider, P. F. Dlp. *In Proc. of the 1999 Description Logic Workshop (DL'99)* (1999), 9–13.

[PV00] Prakken, H., and Vreeswijk, G. Logical Systems for Defeasible Argumentation. In *Handbook of Philosophical Logic, 2nd ed.* 2000.

[QHHP08] Qi, G., Haase, P., Huang, Z., and Pan, J. Z. A Kernel Revision Operator for Terminologies. In *DL* (2008).

[RMF+08a] Rotstein, N. D., Moguillansky, M. O., Falappa, M. A., García, A. J., and Simari, G. R. A Preliminary Reification of Argument Theory Change. *Revista Iberoamericana de Inteligencia Artificial (AEPIA), ISSN 1137-3601* (2008), 40: 51–62.

[RMF+08b] Rotstein, N. D., Moguillansky, M. O., Falappa, M. A., García, A. J., and Simari, G. R. Argument Theory Change: Revision Upon Warrant. In *COMMA* (2008), P. Besnard, S. Doutre, and A. Hunter, Eds., vol. 172 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, pp. 336–347.

[RMGS08] Rotstein, N. D., Moguillansky, M. O., García, A. A., and Simari, G. R. An Abstract Argumentation Framework for Handling Dynamics. In *NMR* (2008), pp. 131–139.

[RMGS10] Rotstein, N. D., Moguillansky, M. O., García, A. J., and Simari, G. R. A Dynamic Argumentation Framework. In Baroni et al. (BCGS10), pp. 427–438.

[RMS09] Rotstein, N. D., Moguillansky, M. O., and Simari, G. R. Dialectical Abstract Argumentation: A Characterization of the Marking Criterion. In *IJCAI* (2009), C. Boutilier, Ed., pp. 898–903.

[Rot10] Rotstein, N. D. Un Marco Argumentativo Abstracto Dinámico. *Tesis Doctoral, Departamento de Ciencias e Ingeniería de Computación, Universidad Nacional del Sur (UNS)* (Abril de 2010).

[RW09] Ribeiro, M. M., and Wassermann, R. Base Revision for Ontology Debugging. *J. of Logic and Computation 19*, 5 (2009), 721–743.

[RZR07] Rahwan, I., Zablith, F., and Reed, C. Laying the Foundations for a World Wide Argument Web. *Artif. Intell. 171*, 10-15 (2007), 897–921.

[Sat88] Satoh, K. Nonmonotonic Reasoning by Minimal Belief Revision. In *FGCS* (1988), pp. 455–462.

[SC03] Schlobach, S., and Cornet, R. Non-Standard Reasoning Services for the Debugging of Description Logic Terminologies. In *IJCAI* (2003), pp. 355–362.

[Sch05] Schlobach, S. Debugging and Semantic Clarification by Pinpointing. In *ESWC* (2005), pp. 226–240.

[tCCMV06] ten Cate, B., Conradie, W., Marx, M., and Venema, Y. Definitorially Complete Description Logics. In Doherty et al. (DMW06), pp. 79–89.

[Tou58] Toulmin, S. *The Uses of Argument*. Cambridge University Press, Cambridge, 1958.

[vB07] van Benthem, J. Dynamic Logic for Belief Revision. *Journal of Applied Non-Classical Logics 17*, 2 (2007), 129–155.

[Vre97] Vreeswijk, G. Abstract Argumentation Systems. *Artif. Intell. 90*, 1-2 (1997), 225–279.

[WBC07] Wyner, A. Z., and Bench-Capon, T. J. M. Towards an Extensible Argumentation System. In *ECSQARU* (2007), K. Mellouli, Ed., vol. 4724 of *Lecture Notes in Computer Science*, Springer, pp. 283–294.

[WGR09] Wells, S., Gourlay, C., and Reed, C. Argument Blogging. In *CMNA* (2009), pp. 57–61.

[ZZL09a] Zhang, X., Zhang, Z., and Lin, Z. An Argumentative Semantics for Paraconsistent Reasoning in Description Logic ALC. In *Description Logics* (2009).

[ZZL09b] Zhang, X., Zhang, Z., and Lin, Z. An Argumentative Semantics for Paraconsistent Reasoning in Description Logic ALC. In *Description Logics* (2009), B. C. Grau, I. Horrocks, B. Motik, and U. Sattler, Eds., vol. 477 of *CEUR Workshop Proceedings*, CEUR-WS.org.