

Consolidación de Ontologías Datalog[±]

Cristhian Ariel David Deagustini

Directores: Guillermo R. Simari y Marcelo A. Falappa

Fecha de exposición: 25 de noviembre de 2015

Laboratorio de Investigación y Desarrollo en Inteligencia Artificial
Departamento de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur,
Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)
Av. Alem 1253, (B8000CPB) Bahía Blanca, Argentina
Tel: (0291) 459-5135 / Fax: (0291) 459-5136
e-mail: cadd@cs.uns.edu.ar

Resumen

En la presente tesis nos enfocamos en el manejo de dos problemas diferentes pero relacionados que suelen aparecer en el conocimiento, especialmente en entornos colaborativos: inconsistencias e incoherencias. Inconsistencia es un problema clásico y ampliamente reconocido en la representación de conocimiento, el cual trae importantes consecuencias para los mecanismos clásicos de inferencia. Incoherencia, por otra parte, ha recibido cada vez más atención desde el surgimiento de lenguajes ontológicos; la misma se relaciona con conflictos en el conjunto de reglas ontológicas que hacen a tales reglas imposibles de satisfacer al mismo tiempo. En este trabajo formalizamos la noción de incoherencia en ontologías Datalog[±], que se encontraba ausente en la literatura, en términos de la satisfacibilidad del conjunto de restricciones en las mismas, y mostramos como bajo ciertas condiciones incoherencia puede llevar a ontologías Datalog[±] inconsistentes.

La contribución principal de este trabajo es el desarrollo de dos operadores noveles para la restauración tanto de la consistencia como la coherencia en ontologías Datalog[±]. Los enfoques propuestos se basan en kernel contraction. En el primero de ellos la restauración se realiza mediante la aplicación de funciones de incisión que seleccionan fórmulas para remoción de los conjuntos incoherentes/inconsistentes mínimos encontrados en las ontologías. Tal operador trata los conflictos mínimos de manera local, sin tener en cuenta la relación (si es que existe) entre los diferentes conflictos mínimos. El otro enfoque, que puede ser visto como un enfoque global, tiene en cuenta tal relación mediante el agrupamiento de conflictos mínimos relacionados en nuevas estructuras llamadas clusters, mediante el uso de una relación de solapamiento.

En esta disertación presentamos construcciones tanto para el enfoque local como el global, junto con las propiedades que se espera que los mismos satisfagan, expresadas a través de postulados. Finalmente, establecemos la relación entre las construcciones y las propiedades mediante el uso de teoremas de representación. Si bien la propuesta está enfocada en la consolidación de ontologías Datalog[±], estos operadores pueden ser aplicados a otros tipos de lenguajes ontológicos, tales como las Lógicas Descriptivas, haciéndolos aptos para su uso en ambientes colaborativos como la Web Semántica.

1. Introducción

En tiempos donde la colaboración e interacción entre sistemas se ha vuelto moneda corriente es cada vez más frecuente el encontrarse con problemas derivados de tal interacción, apareciendo a su vez desafiantes oportunidades de investigación asociadas a los mismos. En particular, el manejo de información conflictiva es un importante problema que debe ser atacado [7, 8, 12, 2], especialmente cuando se integra conocimiento proveniente de diferentes fuentes [4, 1], o cuando tal conocimiento será explotado por procesos de razonamiento automáticos. El más conocido dentro de los conflictos en información es el de la inconsistencia. Inconsistencia es un concepto clásico dentro de la Representación de Conocimiento y el Razonamiento, y se refiere a teorías tales que no es posible encontrar para ellas un modelo. La importancia del manejo de la inconsistencia surge del problema que la misma acarrea en las relaciones de consecuencia clásicas: las mismas son *explosivas* [12, 7], en el sentido de que cualquier fórmula es consecuencia de una contradicción lógica, pudiendo por lo tanto derivar el lenguaje completo a partir de una base de conocimiento inconsistente, haciendo a las conclusiones obtenidas completamente irrelevantes.

En adición a la inconsistencia, en entornos ontológicos podemos encontrar otro tipo de conflictos, relacionados a un fenómeno conectado en cierta forma con inconsistencia pero a su vez con sus características propias: incoherencia [6, 5, 3, 13, 17, 16], un concepto que ha surgido con el advenimiento de entornos ontológicos como Datalog[±]. Tal fenómeno versa en una característica del conocimiento expresado a través de las reglas que hace que el mismo no pueda ser aplicado sin generar problemas de consistencia (es decir, un conjunto de reglas las cuales no pueden ser aplicadas sin violar *inevitablemente* alguna de las restricciones impuestas al conocimiento, haciéndolas por lo tanto insatisfacibles). Intuitivamente podemos ver a la incoherencia como una inconsistencia latente o potencial; esto es, el conocimiento incoherente no representa una violación de consistencia en sí mismo, pero cuando un conocimiento ontológico (un conjunto de reglas) incoherente es considerado junto con hechos relevantes (hechos que activen las reglas en cuestión) entonces la violación es inevitable.

En la Inteligencia Artificial (IA) muchos trabajos se han enfocado en el manejo de los problemas de inconsistencia (y también, aunque en menor medida, incoherencia) en conocimiento de forma tal que se evite de alguna manera los problemas acarreados por la misma. En la presente tesis presentamos dos métodos noveles para la resolución de todos los conflictos de coherencia y consistencia en ontologías Datalog[±]. Tales métodos se centran en modificar de alguna forma el conocimiento inconsistente de forma tal que la consistencia en el mismo sea recuperada, para luego aplicar relaciones de consecuencia clásicas sobre el conocimiento consistente obtenido. De esta forma, si bien la base de conocimiento original es inconsistente y por lo tanto capaz de derivar todo el lenguaje, al utilizar una modificación consistente de la misma se pueden aplicar relaciones de inferencia clásicas de manera segura; esto es conocido en la literatura como *consolidación de conocimiento* [11, 10].

Los operadores propuestos en la tesis toman como entrada una ontología Datalog[±] (la cual es potencialmente inconsistente e incoherente) y retornan otra ontología Datalog[±] donde cada conflicto presente en la original fue subsanado. Siguiendo la tradición en la literatura, un requisito adicional para los procesos de consolidación presentados es que los mismos alteren las ontologías tan poco como sea posible. El enfoque utilizado se basa en *Kernel Contraction* [9, 10], el cual se centra en la resolución de conflictos mínimos conocidos como kernels. De esta forma, el proceso de resolución de conflictos es definido como la identificación de tales conflictos mínimos y la posterior remoción de alguna fórmula de los mismos para de tal forma resolver cada uno de ellos. La ontología resultante del proceso de consolidación es aquella formada por todas las fórmulas de la ontología original que no hayan sido removidas por los operadores.

2. Contribuciones y resultados de la tesis

Las principales contribuciones incluidas en la tesis versan en diferentes direcciones.

2.1. Formalización del concepto de incoherencia para Datalog[±].

Incluso cuando es un concepto muy importante para la representación de conocimiento en entornos ontológicos, la noción de incoherencia en ontologías Datalog[±] no había sido hasta el momento formalizada. Por esto, en el presente trabajo proveemos una definición formal de la misma, adaptando para tal fin una noción similar utilizada para las Lógicas Descriptivas. La formalización esta basada en la identificación de conjuntos de reglas utilizadas para la inferencia de nuevo conocimiento tales que las reglas en el conjunto no pueden ser aplicadas en su totalidad sin caer en una violación de las restricciones que se han impuesto al conocimiento. Además, ahondamos en la relación de la incoherencia con la inconsistencia, enfocándonos especialmente en como tal relación afecta la calidad de los procesos de consolidación de ontologías Datalog[±], particularmente la pérdida de información, al punto de ser necesario definir operadores que tratan ambos problemas pero de forma separada.

2.2. Establecimiento de las propiedades esperadas para procesos de consolidación de ontologías Datalog[±].

Brindamos un conjunto de propiedades que se espera sean satisfechas por operadores de consolidación de ontologías Datalog[±], adaptando las mismas a los casos de operadores trabajando a nivel local o global según el caso. Tales propiedades son presentadas a través de postulados, considerando algunas intuiciones en enfoques tradicionales para la Revisión de Creencias los cuales son adaptados al entorno ontológico de Datalog[±] (pero que pueden ser adaptados para que se adecúen a otros lenguajes ontológicos). Tales postulados proveen una caracterización formal de los operadores de consolidación sin adentrarse en *cómo* el proceso de consolidación es efectivamente realizado, proveyendo de esta forma un marco de comparación formal de operadores de consolidación de ontologías Datalog[±], tanto los definidos en la presente tesis como cualquier otro que trabaje en tales entornos.

2.3. Definición de construcciones para operadores de consolidación locales y globales.

En la tesis introducimos construcciones nóveles completas tanto para operadores de consolidación que tratan kernels por separado como para aquellos que explotan la relación de los mismos en un enfoque global. Tales construcciones toman una ontología Datalog[±] (posiblemente) inconsistente e incoherente y devuelve como resultado un subconjunto consistente y coherente de la misma. Una característica destacable de los operadores es que los mismos involucran enfoques desdoblados, donde primero el operador considera conflictos de incoherencia y resuelve los problemas de consistencia en un paso subsecuente, lo cual ayuda a mejorar el resultado final obtenido.

3. Incoherencia en ontologías Datalog[±]

Como explicamos, en el trabajo de la tesis nos hemos enfocado en la resolución de conflictos en ontologías Datalog[±]. La familia Datalog[±] de variantes de Datalog es un lenguaje que ha surgido en los últimos tiempos, y se ha vuelto popular debido a combinar una adecuada expresividad con una escalabilidad capaz de manejar los grandes volúmenes de datos presentes en las aplicaciones hoy en día. Por ejemplo, tales lenguajes extienden el Datalog básico con la posibilidad

de utilizar cuantificaciones existenciales en las cabezas de las reglas, entre otras características importantes, manteniendo a su vez una tratabilidad adecuada para entornos de alto volumen de información mediante restricciones a la sintaxis de reglas. Datalog[±] permite un estilo modular de representación de conocimiento basado en reglas, lo que lo hace útil en diversos entornos prácticos, como ser resolución de consultas en ontologías (ontology querying), extracción de datos de la Web (Web data extraction) o intercambio de datos (data exchange) [14].

La representación de conocimiento en ontologías Datalog[±] se lleva a cabo mediante el uso de (a) una Base de Datos: un conjunto de átomos que representan hechos acerca del mundo, v.g., alumno(pedro) (b) Tuple-generating Dependencies - TGDs: reglas que nos permiten obtener nuevos átomos mediante la activación de las mismas como ser: alumno(X) → persona(X), (c) Equality-generating Dependencies: reglas que restringen la generación de átomos, por ejemplo: doctor(D, P) ∧ doctor(D', P) → D = D'; y (d) Negative Constraints - NCs: reglas que expresan relaciones que no pueden existir entre átomos, v.g.,: alto(X) ∧ bajo(X) → ⊥.

Un fenómeno ampliamente estudiado en IA (incluso en Datalog[±]) es el de inconsistencia. Intuitivamente, la inconsistencia en Datalog[±] surge cuando una o más NCs o EGDs en una ontología son violadas, ya sea a través de la presencia de átomos en conflicto en el componente D o la generación de estos átomos a través de la aplicación de TGDs. Es decir, toda ontología inconsistente es tal que no tiene ningún modelo, lo que significa que no hay forma de satisfacer todas las restricciones de la misma al mismo tiempo (recordemos que un modelo es tal que toda fórmula en Σ es satisfecha).

Muchos trabajos enfocados en el manejo de inconsistencia utilizan restricciones para enfocar la resolución de inconsistencias. En general, se asume que tales restricciones no tienen problemas y por lo tanto son satisfacibles. Sin embargo, tales suposiciones no siempre son aplicables. En el trabajo desarrollado para la tesis elegimos considerar que tanto los datos como las restricciones pueden cambiar a medida que una ontología crece, y por lo tanto no sólo consideramos problemas relacionados a los datos sino también a las restricciones impuestas sobre los mismos. En este escenario más general un nuevo fenómeno aparece: incoherencia. Nuestra concepción de (in)coherencia está basada en la insatisfacibilidad de un conjunto de TGDs. Intuitivamente, consideramos un conjunto de TGDs como satisfacible cuando existe un conjunto de átomos que desencadena la aplicación de las TGDs y no produce la violación de ninguna restricción en $\Sigma_{NC} \cup \Sigma_E$, esto es, las TGDs pueden ser satisfechas junto con las NCs y las EGDs en KB .

Ejemplo 1 (Conjuntos insatisfacible de TGDs) Como un ejemplo de insatisfacibilidad (y por lo tanto incoherencia) considere el siguiente conjunto de restricciones:

$$\Sigma_{NC}^1 : \{\tau_1 : garganta_lastimada(X) \wedge puede_cantar(X) \rightarrow \perp\}$$

$$\Sigma_E^1 : \{\nu_1 : representa(X, Y) \wedge representa(X, Z) \rightarrow Y = Z\}$$

$$\Sigma_T^1 : \{\sigma_1 : cantante_rock(X) \rightarrow canta_fuerte(X), \\ \sigma_2 : canta_fuerte(X) \rightarrow garganta_lastimada(X), \\ \sigma_3 : cantante_rock(X) \rightarrow puede_cantar(X)\}$$

El conjunto Σ_T^1 es un conjunto insatisfacible de dependencias, ya que la aplicación de las TGDs $\{\sigma_1, \sigma_2, \sigma_3\}$ en cualquier conjunto relevante de átomos (cualquier conjunto que dispare todas las TGDs) causará la violación de τ_1 , v.g., si consideramos el átomo relevante cantante_rock(axl) tenemos que Axl puede cantar y tiene la garganta lastimada. Nótese que cualquier conjunto relevante de átomos (esto es, cualquier cantante que consideremos) causará la violación de τ_1 .

Bajo el uso de este nuevo concepto de satisfacibilidad de un conjunto de TGDs desarrollado para la tesis tenemos que una ontología es coherente en el caso de que el conjunto de TGDs en la misma es satisfacible con respecto a las EGDs y NCs en ella, e incoherente en el caso contrario. Como veremos más adelante, la cercana relación entre inconsistencia e incoherencia hace que

los operadores de consolidación de ontologías pueden beneficiarse de atacar no sólo problemas de inconsistencia sino también eliminando las incoherencias en el conocimiento expresado en las ontologías Datalog[±].

4. Consolidación de ontologías Datalog[±]

Como uno de los aportes principales de la tesis hemos presentado un enfoque que posibilita la consolidación de ontologías Datalog[±]. Tal proceso está basado en las ideas de Hansson [9, 10] de atacar conflictos mínimos conocidos como kernels, removiendo de los mismos fórmulas para resolver los conflictos. Además de atender conflictos de inconsistencia, los operadores definidos también se enfocan en el otro tipo de conflictos que pueden surgir en entornos ontológicos: la incoherencia. De esta forma puede verse a los operadores como unos que trabajan en dos fases: los mismos comienzan por resolver los problemas de incoherencia que aparezcan en la ontología mediante la remoción de TGDs, para luego dar paso a la restauración de la consistencia a través de la eliminación de átomos del componente D . En ambos casos, las selecciones se hacen a través de funciones de incisión que deciden qué remover en base a una relación general \prec , mirando para ello localmente en cada kernel en la ontología. Esto es, los operadores eligen de cada kernel aquella fórmula que es señalada por \prec como la menos deseable dentro del kernel.

Para caracterizar apropiadamente el proceso de consolidación de ontologías hemos introducido un conjunto de postulados, adaptando intuiciones introducidas para otros formalismos de representación de conocimiento. A tales propiedades las hemos aumentado con los postulados de Optimidad Local para Restricciones y Optimidad Local para Datos, los cuales son utilizados para enfocarnos en aquellas consolidaciones que minimizan la pérdida de información con respecto a \prec , donde la misma es considerada localmente, *esto es*, para cada conjunto minimalmente inconsistente/incoherente. El conjunto básico de Postulados presentados versa entonces en diferentes aspectos de la consolidación de ontologías, desde requerir que nada sea agregado a la ontología original hasta que aquello que no involucra conflictos sea retenido. Este conjunto básico de postulados implica a su vez ciertas otras propiedades. En particular, dos propiedades que se derivan del conjunto de postulados básicos y que son muy importantes a efectos del objetivo de los operadores de consolidación son los de **Coherencia** y **Consistencia**. Es importante remarcar que, como es tradicional en la teoría de cambios, los operadores no se encuentran atados a las construcciones particulares que se presentan en la presente tesis, sino que son un aporte general a cualquier operador de consolidación de ontologías Datalog[±] (y que pueden ser incluso generalizados a otros lenguajes ontológicos con relativa sencillez). Es decir, cualquier operador de consolidación de ontologías Datalog[±] puede ser analizado a la luz de los postulados presentados, y no solamente aquellos introducidos en este trabajo. Por lo tanto, el conjunto de postulados introducidos corresponde a un marco formal general del comportamiento de operadores de consolidación.

Luego de presentar los postulados que moldean el comportamiento de los operadores de consolidación de ontologías Datalog[±] nos hemos dedicado a la definición de una clase particular de tales operadores, presentando para ello una construcción completa que puede generarlos. Para ello primeramente procedimos a definir que corresponden con kernels de dependencias (conjuntos insatisfacibles de TGDs mínimos bajo inclusión conjuntista) y kernels de datos (conjuntos mínimos de átomos que hacen a la ontología inconsistente). En particular, para los kernels de dependencias hemos demostrado que los mismos son independientes de la instancia particular del componente D en la ontologías, es decir que los mismos no dependen de los átomos. Esto es particularmente interesante porque nos da la pauta de que el conjunto de kernels de dependencias de una ontología puede ser obtenido sin considerar el componente D de la misma, lo que es de utilidad a la hora de separar la resolución de incoherencias de la resolución de inconsistencias.

Una vez que hemos identificado los kernels sobre los que aplicaremos las operaciones de contracción en pos de resolver incoherencias e inconsistencias, debemos establecer como tales problemas son resueltos. Para ello utilizamos funciones de incisión, que seleccionan qué TGDs

y qué átomos serán removidos de los kernels de dependencias y kernels de datos, respectivamente. Es posible definir funciones de incisión generales, que consideran al mismo tiempo tanto incoherencia como inconsistencia. Sin embargo, tal enfoque acarrea un problema: para hacer esto deberíamos calcular los kernels de dependencias y de datos y resolver los conflictos al mismo tiempo. En la tesis mostramos que en la presencia de incoherencia esto puede no ser la mejor solución, al considerarlo a la luz de la pérdida mínima de información, ya que para el caso de átomos relevantes a conjuntos insatisfacibles de TGDs los mismos deben ser obligatoriamente eliminados por una función de incisión general, incluso cuando al mismo tiempo la función de incisión resuelve el problema de incoherencia que hacía que el átomo en cuestión sea un kernel (resolviendo indirectamente la inconsistencia).

Para resolver tal situación es que hemos introducido una separación entre las funciones de incisión que trabajan sobre el conjunto de TGDs de una ontología de aquellas que se encargan de la remoción de átomos de la misma. En base a tal separación definimos a los operadores de consolidación de ontologías $Datalog^\pm$ como aquellos que utilizan una función de incisión en restricciones sobre el componente Σ y una función de incisión en datos sobre D . Se puede considerar que tales operadores trabajan en “fases” separadas: primero se resuelven todos los problemas de incoherencia, y luego la resolución de inconsistencias se hace sobre la ontología intermedia obtenida mediante la remoción de TGDs. Es decir, por un lado, las TGDs son removidas de Σ (solamente TGDs, ya que los kernels de dependencias no contienen EGDs ni NCs). Por el otro lado, la función de incisión en datos usa una ontología intermedia KB^* en lugar de KB . De esta forma logramos que sólo sean removidos átomos de D que estén en conflicto con el conjunto de TGDs resultante de resolver las incoherencias, ya que los kernels de datos son calculados en base a las restricciones obtenidas luego de aplicar sobre Σ la función de incisión en restricciones.

Por último, hemos establecido la relación entre los postulados y la construcción presentada a través de un teorema de representación, el cual establece que la relación biunívoca entre el conjunto de postulados OCP1-OCP7 y la construcción en base a funciones de incisión. La importancia de tal resultado es que determina que cualquier construcción que satisfaga tales postulados se corresponderá necesariamente con los operadores presentados en este capítulo, y por lo tanto los resultados de esos operadores pueden ser obtenidos por nuestra construcción.

5. Refinamiento del proceso de consolidación de ontologías $Datalog^\pm$

Además de los operadores locales mencionados anteriormente, en la tesis hemos introducido una clase alternativa de operadores de consolidación de ontologías que modifican el comportamiento los mencionados. La diferencia entre los enfoques se da en que, mientras que los operadores presentados previamente atacan localmente a cada conflicto mínimo, en el enfoque alternativo se utiliza una mirada más global, considerando la relación (cuando la misma existe) entre los distintos kernels. La consideración de tal relación se hace mediante el agrupamiento de kernels relacionados en una superestructura denominada cluster, la cual es obtenida explotando una relación de overlapping entre conflictos.

Para ello primeramente hemos analizado el comportamiento de los operadores locales, enfocando el análisis en la pérdida innecesaria de información que los mismos pueden sufrir bajo ciertas condiciones, a saber la existencia de kernels relacionados y estrategias de resolución de conflictos donde la elección óptima local en cada kernel hace que debamos de remover de un kernel una fórmula (la indicada por la relación de preferencia) incluso cuando otra fórmula del mismo kernel ya ha sido seleccionada para remoción como estrategia de solución del conflicto expresado por un kernel diferente. Para poder ver lo expresado considere el siguiente ejemplo, que ilustra tanto el comportamiento de un operador local como el problema que puede traer aparejado.

Ejemplo 2 Supongamos que tenemos $\perp\!\!\!\perp_{(D,KB^*)} = \{\{a_1, a_3\}, \{a_2, a_3\}\}$, y \prec donde $a_1 \prec a_3 \prec a_2$, donde $\perp\!\!\!\perp_{(D,KB^*)}$ denota el conjunto de kernels de datos en la ontología. Entonces, tenemos que la función de incisión ϱ es tal que

$$\begin{aligned}\varrho(\{a_2, a_3\}) &= \{a_3\} \\ \varrho(\{a_1, a_3\}) &= \{a_1\}\end{aligned}$$

La remoción de a_1 es innecesaria, ya que el conflicto es resuelto indirectamente mediante la remoción de a_3 del primer kernel. Sin embargo, un operador local lo removería, ya que no considera la interacción entre los kernels. Dados estos dos kernels las incisiones minimales son o bien remover el átomo a_3 o bien remover $\{a_1, a_2\}$.

Esta situación pone en evidencia que el conjunto de postulados introducidos previamente no es suficiente para paliar completamente el problema de pérdida mínima de información, por lo que hemos introducido una propiedad adicional. La propiedad que se espera sea satisfecha por un operador optimal (en el sentido de mínimo cambio) es la de **Mínima Pérdida de Información**. La intuición detrás del postulado es que un operador de consolidación que lo satisface nos dará como resultado una ontología tal que si agregamos cualquiera de las fórmulas eliminadas entonces la KB resultante sería incoherente o inconsistente; es decir, el operador realizó una cantidad de cambios *mínima* en la resolución de conflictos.

Como se ve en Ejemplo 2, la propiedad de mínima pérdida no es satisfecha por los operadores locales, debido al uso de kernels de manera aislada, sin tener en cuenta aquellas ocasiones donde los conflictos minimales están en una relación de “colisión”. Es por esto que en la tesis introducimos una construcción alternativa que se basa en la detección de la relación entre kernels, para de esta forma encontrar la solución que implique la menor pérdida posible. Para lograr esto, en lugar de realizar incisiones sobre kernels el nuevo operador realiza las mismas sobre *clusters*. La estructura de los clusters hace posible identificar conflictos relacionados a través de una relación de *solapamiento* (overlapping) entre kernels.

Sin embargo, el uso de clusters trae aparejado un problema: al remover alguna fórmula de un kernel el conjunto resultante siempre es coherente/consistente ya que los kernels son conjuntos mínimos, pero en el caso de los clusters no podemos asegurar que la remoción de una fórmula nos de como resultado un conjunto con todos los conflictos agrupados resueltos [14]. Para paliar este problema hemos presentado una nueva clase de funciones de incisión que aseguran que una vez que el conjunto de fórmulas seleccionado por la función es removido todos los conflictos mínimos incluidos en el cluster quedan resueltos. En base a tales funciones de incisión hemos definido entonces a los operadores de consolidación de ontologías Datalog^\pm basados en Cluster Contraction los cuales, al igual que los operadores locales, utilizan remoción de átomos y TGDs como estrategia de resolución de incoherencias e inconsistencias.

Una vez presentada la construcción propuesta para los operadores de consolidación de ontologías basados en Cluster Contraction hemos ahondado en la relación existente entre tal construcción y las propiedades presentadas, a través de un teorema de representación. Tal teorema muestra que estos operadores efectivamente satisfacen el postulado de mínima pérdida de información, y por lo tanto representación soluciones óptimas desde tal punto de vista. Es decir, que los operadores basados en Cluster Contraction son tales que remueven lo mínimo que es requerido para solucionar los conflictos, y por lo tanto si fuéramos a dejar siquiera una fórmula de aquellas removidas por el operador entonces la ontología seguiría siendo incoherente/inconsistente.

Un hallazgo interesante presentado en la tesis es que existen casos donde los enfoques coinciden. Un claro caso (trivial) de ésta situación se da al tratar con KB s coherentes y consistentes. Sin embargo, no es el único caso donde tal cosa puede suceder. Otro caso que llevaría a lo mismo es aquél donde los conjuntos de kernels y de clusters coinciden, *esto es*, no hay overlapping entre conflictos mínimos; nuevamente, para tales casos los enfoques coincidirían. Esto, si bien puede parecer trivial y no reportar mucho interés en una primera mirada, se vuelve importante a la hora

de analizar a los operadores desde un punto de vista computacional. Claramente, el manejo de kernels es más sencillo que el manejo de clusters; por ejemplo, como remover una fórmula alcanza para solucionar conflictos mínimos entonces podemos comparar conjuntos unitarios de fórmulas, sin necesidad de mirar en todos los subconjuntos como sucede en Cluster Contraction. Por lo tanto, si tenemos que el conjunto de clusters coincide con el de kernels (lo que sólo involucraría una comparación de conjuntos adicional) podríamos directamente tomar el enfoque local, consiguiendo una mejor performance computacional. Es más, esto puede ser llevado un paso más allá, definiendo operadores mixtos: podríamos utilizar un enfoque local para el caso donde *un* cluster coincide con algún kernel, y utilizar un enfoque global sobre aquellos que involucran dos o más conflictos mínimos.

6. Conclusiones

El trabajo colaborativo y el intercambio de información se han transformado en los últimos tiempos en aspectos cruciales de prácticamente todo sistema; por lo tanto, es de vital importancia disponer de métodos automáticos y adecuados para el manejo de conflictos: en entornos colaborativos, a medida que el conocimiento evoluciona inconsistencias e incoherencias suelen aparecer de manera natural. Es más, en entornos colaborativos tal conocimiento es frecuentemente representado a través de ontologías que suelen ser mantenidas en forma conjunta por muchas entidades, *esto es*, son compartidas por entidades que no sólo las utilizan sino que también las modifican.

Una forma de lidiar con los conflictos que pueden aparecer en tales entornos de aplicación es tratar de modificar la información contenida en la ontología para de esta forma recuperar la coherencia y la consistencia del conocimiento expresado por las mismas. En este trabajo hemos presentado una forma de conseguir la consolidación de ontologías Datalog^{\pm} , *esto es*, la resolución de todos los conflictos presentes en las mismas, tanto de inconsistencia como de incoherencia.

En particular, el fenómeno de incoherencia es uno que suele ser dejado de lado, a pesar de tener gran influencia en la definición de procesos de manejo de inconsistencia. Para paliar esto hemos presentado una caracterización del fenómeno de incoherencia en Datalog^{\pm} , tomando como punto de partida esfuerzos similares en DLs, donde se relaciona a tal fenómeno con la aparición de conceptos insatisfacibles [6, 5, 3, 13, 17, 16].

Además de introducir los conceptos de satisfacibilidad y coherencia para Datalog^{\pm} , en el presente trabajo hemos ahondado en la relación entre inconsistencia e incoherencia. En particular, el trabajo muestra que tal relación toma la forma de causa-efecto: cuando tenemos una ontología incoherente la misma es inevitablemente inconsistente cuando el componente D hace que un conjunto insatisfacible sea activado. Esta relación es de vital importancia para la definición de procesos de manejo de inconsistencia: es fácil demostrar que en presencia de una ontología incoherente puede ser una mejor solución aplicar un enfoque de eliminación de incoherencia como paso previo al manejo de inconsistencia, lo que permitirá aumentar la cantidad de respuestas posibles para la ontología previamente incoherente.

De la mano de diferenciar y atacar de forma correspondiente cada tipo de conflicto es que en la presente tesis hemos propuesto la utilización de operadores de consolidación de ontologías Datalog^{\pm} que tengan en cuenta ambas clases de conflictos, *esto es*, que ataquen tanto incoherencia como inconsistencia. Esto último explicado es central dentro de la construcción de operadores de consolidación propuestos, ya que la misma está basada en la separación en dos tipos de funciones de incisión de modo que se pueda atacar primeramente los problemas de coherencia utilizando una función de incisión en restricciones ρ , y luego resolver los problemas de consistencia basándonos en el resultado de remover de la ontología original aquellas TGDs seleccionadas por ρ . Esto es; el resultado de una operación de consolidación $\Upsilon_{\rho, \varrho}(KB)$ es la ontología obtenida removiendo TGDs (seleccionadas por una función de incisión en restricciones ρ del conjunto de kernels de dependencias $\perp\!\!\!\perp_{(\Sigma, KB)}$) y átomos (seleccionados por una función de incisión en datos ϱ de $\perp\!\!\!\perp_{(D, KB^*)}$, que es el conjunto de kernels de datos para la ontología resultante de resolver las

incoherencias como primer paso) de la ontología original *KB*.

La primer construcción propuesta ataca directamente kernels de manera local, lo que acarrea un problema al examinarla a la luz de la mínima pérdida de información; en ocasiones, cuando dos o más kernels están relacionados su tratamiento local no es óptimo. Para paliar tal problema es que hemos definido una estructura adicional denominada cluster, que tiene la función de agrupar kernels relacionados de modo que podamos resolver tales conflictos utilizando un enfoque global del problema. Sin embargo, la aplicación de funciones de incisión definidas para trabajar en kernels directamente a clusters no fue posible, por lo que se presentan nuevas funciones de incisión adaptadas a su uso en clusters, y se define una nueva clase de operadores de consolidación que aseguran que siempre la ontología resultante de la consolidación es la más cercana a la ontología original (en términos de pérdida de información), aunque como contrapartida el esfuerzo computacional puede llegar a ser mayor.

Paralelamente a la presentación de las construcciones antes mencionadas, en la presente tesis hemos presentado una serie de propiedades que operadores de consolidación de ontologías *Datalog*[±] deberían respetar para exhibir un comportamiento que preserve los requisitos de resolver los conflictos que aparecen en las ontologías, tratando a su vez de minimizar las pérdida de información. Como es característico en Revisión de Creencias, estas propiedades son presentadas como postulados, donde los mismos son independientes de la construcción propuesta; esto es, los postulados pueden ser utilizados para medir el desempeño de cualquier operador de consolidación de ontologías *Datalog*[±], y no solamente aquellos propuestos en este trabajo.

Respecto de la relación entre las construcciones presentadas (local/kernel y global/cluster) y las propiedades esperadas que expresan los postulados, hemos establecido claramente la relación entre ambos aportes mediante distintos teoremas de representación. El aporte de tales teoremas es doble: (a) muestran claramente como las construcciones propuestas satisfacen las propiedades esperadas para cada tipo de operador, y (b) muestran que cada operador de consolidación que sea definido de forma tal que satisfaga las propiedades en los teoremas se corresponderá necesariamente con alguna de las construcciones propuestas en este trabajo, es decir, que dada la misma ontología original obtendrá como resultado la misma ontología consolidada que se obtiene utilizando las construcciones presentadas .

7. Líneas de investigación futuras

Obtención de vistas unificadas de varias bases de datos relacionales.: Una aplicación posible de los operadores de consolidación de ontologías *Datalog*[±] es su utilización para la consecución de una vista única de diferentes fuentes de información expresables por *Datalog*[±] (como por ejemplo Bases de Datos Relacionales), mediante un proceso de transformación de las bases de datos relacionales a ontologías *Datalog*[±] primeramente, para luego consolidar (la unión de) las mismas y obtener la vista.

Uso de la consolidación como soporte a procesos de evolución de ontologías: La evolución de ontologías (ontology evolution) trata de modificar las ontologías para reflejar cambios en el dominio que las mismas representan de forma tal que no haya conflictos en la ontología resultante. Para conseguir esto debemos definir procesos de revisión en *Datalog*[±] mediante el enfoque de consolidación presentado como la base del desarrollo de operadores de revisión de ontologías *Datalog*[±]. Para esto, tales operadores comienzan agregando la fórmula en consideración (ya sea un átomo o una TGD) para luego asegurarse de que lo resultante de tal expansión no sea inconsistente/incoherente mediante la remoción de átomos o TGDs. En base a (clases particulares de) tales operadores podemos definir un proceso de evolución de ontologías como aquel que revisa a la ontología actual con el nuevo átomo a agregar, logrando de esta forma incluir esta nueva información sin que aparezcan por ellos conflictos en la ontología resultante.

Manejo de incoherencias a través de debilitación de reglas: Defeasible *Datalog*[±]: Una propuesta alternativa para el manejo de incoherencias que pensamos explorar es la basada en de-

bilitamiento de reglas, en lugar de remoción de las mismas. Para lograr esto pensamos explotar unos desarrollos recientes que han surgido dentro del grupo de investigación que anidó el trabajo presentado en esta disertación, pero que se centran en la modificación de la relación de consecuencia para el manejo de ontologías inconsistentes: Datalog[±] rebatible [15].

Implementación de operadores: Finalmente, en la actualidad estamos trabajando en la implementación de nuestros operadores; planeamos para esto estudiar diferentes técnicas que puedan ser utilizadas para obtener una implementación eficiente de los operadores, posiblemente enfocados en fragmentos específicos de Datalog[±].

Referencias

- [1] L. Amgoud and S. Kaci. An argumentation framework for merging conflicting knowledge bases: The prioritized case. In *ECSQUARU 2005*, pages 527–538, 2005.
- [2] D.A. Bell, G. Qi, and W. Liu. Approaches to inconsistency handling in description-logic based ontologies. In *OTM Workshops*, pages 1303–1311, 2007.
- [3] D. Beneventano and S. Bergamaschi. Incoherence and subsumption for recursive views and queries in object-oriented data models. *DKE*, 21(3):217–252, 1997.
- [4] E. Black, A. Hunter, and J. Z. Pan. An argument-based approach to using multiple ontologies. In *SUM 2009*, pages 68–79, 2009.
- [5] A. Borgida. Description logics in data management. *TKDE*, 7(5):671–682, 1995.
- [6] G. Flouris, Z. Huang, J. Z. Pan, D. Plexousakis, and H. Wache. Inconsistencies, negations and changes in ontologies. In *AAAI 2006*, pages 1295–1300, 2006.
- [7] S. A. Gómez, C. I. Chesñevar, and G. R. Simari. Reasoning with inconsistent ontologies through argumentation. *App Artif. Intell.*, 24(1&2):102–148, 2010.
- [8] P. Haase, F. van Harmelen, Z. Huang, H. Stuckenschmidt, and Y. Sure. A framework for handling inconsistency in changing ontologies. In *ISWC 2005*, pages 353–367, 2005.
- [9] S. O. Hansson. Kernel contraction. *J. of Symb. Log.*, 59(3):845–859, 1994.
- [10] S. O. Hansson. *A Textbook of Belief Dynamics: Theory Change and Database Updating*. Kluwer Academic Publishers, Norwell, MA, USA, 1999.
- [11] Sven Ove Hansson. *Belief Base Dynamics*. PhD thesis, Uppsala University, Department of Philosophy, Uppsala, Sweden, 1991.
- [12] Z. Huang, F. van Harmelen, and A. ten Teije. Reasoning with inconsistent ontologies. In *IJCAI 2005*, pages 454–459, 2005.
- [13] A. Kalyanpur, B. Parsia, E. Sirin, and J. A. Hendler. Debugging unsatisfiable classes in owl ontologies. *Web Semantics: Science, Services and Agents on the World Wide Web*, 3(4):268–293, 2005.
- [14] T. Lukasiewicz, M. V. Martínez, and G. I. Simari. Inconsistency handling in datalog+/- ontologies. In *ECAI 2012*, pages 558–563, 2012.
- [15] M. V. Martínez, C. A. D. Deagustini, M. A. Falappa, and G. R. Simari. Inconsistency-tolerant reasoning in datalog[±] ontologies via an argumentative semantics. In *IBERAMIA 2014*, pages 15–27, 2014.
- [16] G. Qi and A. Hunter. Measuring incoherence in description logic-based ontologies. In *ISWC/ASWC 2007*, pages 381–394, 2007.
- [17] S. Schlobach, Z. Huang, R. Cornet, and F. van Harmelen. Debugging incoherent terminologies. *J. of Autom. Reason.*, 39(3):317–349, 2007.