

Implementación de Operadores de Consolidación de Ontologías en Datalog +/-

Ignacio Novello Pelayo

Cristian D. Pacifico

Cristhian A. D. Deagustini

Marcelo A. Falappa

Guillermo R. Simari

Area de Agentes y Sistemas Inteligentes
Facultad de Ciencias de la Administración
Universidad Nacional de Entre Ríos
Tavella 1424, (3200) Concordia, Entre Ríos

Laboratorio de Investigación y Desarrollo en Inteligencia Artificial
Instituto de Ciencias e Ingeniería de la Computación,
Universidad Nacional del Sur/CONICET
Alem 1253 - Bahía Blanca - Buenos Aires - Argentina
(0291) 459-5135

(inovello,cripac)@fcad.uner.edu.ar, (cadd,mfalappa,grs)@cs.uns.edu.ar

Resumen

En los últimos tiempos, la colaboración y el intercambio de información se han vuelto aspectos cruciales de muchos sistemas. En estos entornos es de vital importancia definir métodos automáticos para resolver conflictos entre el conocimiento compartido por distintos sistemas. Este conocimiento es frecuentemente expresado a través de ontologías que pueden ser compartidas por los sistemas que utilizan el mismo. Dentro de las diferentes posibilidades para expresar conocimiento en los últimos tiempos un lenguaje que ha recibido cada vez más atención es Datalog+/-; debido a ser un lenguaje que ofrece un alto nivel de expresividad por construir reglas en fragmentos de Lógica de Primer Orden, permitiendo su comprensión natural en la forma de esquemas de lógica clásica. Por otro lado, Datalog+/- como máquina de inferencia, tiene la propiedad de ser

cia de operadores de contracción de kernel y de cluster para manejo de inconsistencias e incoherencias en ontologías Datalog+/- a través de la implementación de los mismos. Para esto, se analizará el diseño teórico de los operadores de contracción de kernel y cluster y su aplicación práctica para eliminar conflictos. A partir de un proyecto de software que implementa un intérprete Datalog+/-, se codificará las funcionalidades requeridas e implementarán los citados operadores.

Palabras Clave: Integración de Bases de Conocimiento, Revisión de Creencias, Representación de Conocimiento, Razonamiento.

1. Contexto

Esta línea de investigación se lleva a cabo en el marco de la investigación:

- **“Representación de conocimiento y razonamiento argumentativo: Herramientas inteligentes para la web y las bases de conocimiento”**. Director: Guillermo R. Simari. 01/01/15 – 31/12/2018. Unidad coordinadora: Universidad Nacional del Sur.
- **“Combinación de Revisión de Creencias y Argumentación para mejorar las capacidades de Razonamiento y modelado de la Dinámica de Conocimiento en Sistemas Multi-agente”**. Director: Marcelo A. Falappa. PIP 112-20110101000. Unidad coordinadora:

tornos reales.

Sin embargo, el uso del conocimiento, especialmente compartido, suele traer aparejados conflictos en el mismo que dificulta su explotación por procesos automatizados. Es decir, aceptar nuevas observaciones y datos suele traer aparejados violaciones a la integridad y consistencia del cuerpo de conocimiento. En este sentido, el principal desafío es eliminar los conflictos las inconsistencias e incoherencias que puedan surgir en el conocimiento expresado.

En la presente investigación se busca comprobar empíricamente la correctitud, computabilidad y eficien-

Consejo Nacional de Investigaciones Científicas y Técnicas.

La línea de investigación se enmarca dentro del ámbito de colaboración entre el Laboratorio de Investigación y Desarrollo en Inteligencia Artificial (LIDIA) del Instituto de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur/CONICET y el Área de Agentes y Sistemas Inteligentes de la Facultad de Ciencias de la Administración, Universidad Nacional de Entre Ríos.

2. Introducción

El trabajo colaborativo y el intercambio de información son aspectos cruciales de todo sistema de información, esto hace necesario contar con métodos automáticos y adecuados para el manejo de conflictos, esto es, inconsistencias e incoherencias que aparecen naturalmente en la medida que el conocimiento evoluciona. En entornos de este tipo, las bases de conocimiento son representadas a través de ontologías.

En particular y citando el análisis en [Dea15, DMFS16], el manejo de información conflictiva en bases de conocimientos es un importante problema que debe ser atacado [GCS10, HvHtT05, BQL07], especialmente cuando se integra conocimiento proveniente de diferentes fuentes [BHP09, AK05], o cuando tal conocimiento será explotado por procesos de razonamiento automáticos. El más conocido dentro de los conflictos es 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; es decir, al menos una interpretación que haga verdaderas a todas las fórmulas de la teoría. En términos sintácticos, un conjunto de fórmulas es consistente si y sólo si para toda fórmula A , no es posible deducir tanto A como $\neg A$. Adicionalmente, en entornos ontológicos se encuentra otro tipo de conflicto, relacionados a un fenómeno conectado en cierta forma con inconsistencia pero a su vez con sus características propias: la incoherencia [FHP⁺06, BB97, KPSH05, SHCvH07, QH07]. Tal fenómeno surge cuando una porción de conocimiento expresado a través de un conjunto de reglas no puede ser aplicado sin generar problemas de consistencia; es decir, el conjunto de reglas no puede ser aplicado 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. Consecuentemente, si este conjunto de reglas incoherente es considerado junto con hechos relevantes (hechos que activen las reglas en cuestión) entonces la violación es inevitable, generando inconsistencia. Esto marca la diferencia más grande con el concepto de inconsistencia: la incoherencia no necesariamente implica una contradicción; al punto de que incoherencia no

puede provocar una violación explícita sin ser considerada junto con instanciaciones de las reglas. Es decir, mientras que la inconsistencia es un conflicto puesto de manifiesto (explícito, una violación que ya está ocurriendo) la incoherencia se refiere a conflictos latentes (implícitos, pero que no necesariamente estén ocurriendo en el momento); se podría pensar por lo tanto en la incoherencia como un preludio de la inconsistencia. En estos entornos, para tratar con los conflictos planteados y recuperar la coherencia y la consistencia, existen dos enfoques:

- El primero de ellos implica modificar la información contenida en la ontología inconsistente de forma tal que la consistencia de la misma sea recuperada, para luego aplicar relaciones de consecuencia clásicas sobre el conocimiento consistente obtenido. De esta forma, se modifica la base de conocimiento original (e inconsistente), de manera que la base de conocimiento modificada, pueda aplicar relaciones de inferencia clásicas de manera segura. Este es uno de los objetivos perseguidos por la teoría de Revisión de Creencias [Gäro3, AGM85, Dal88, QLBo6, Han94].
- El segundo enfoque, propone modificar la forma en cómo el conocimiento es inferido mediante la definición de relaciones de inferencia alternativas, estableciendo de esta forma métodos de razonamiento tolerantes a inconsistencias, que son aquellos que pueden obtener conclusiones consistentes a partir de conocimiento inconsistente. Entre los ejemplos más importantes de tales métodos podemos citar a la Argumentación y el Razonamiento Rebatible [SL92, RSvBo9, GSo4, Pra10].

Siguiendo la línea del primer enfoque, trabajos previos [Dea15, DMFS16] han presentado de manera teórica, una alternativa para conseguir la consolidación de ontologías Datalog+/- . Este es un framework para realizar consultas de ontologías tratables y otras múltiples aplicaciones, basado en el lenguaje Datalog [CGL12]. Extiende el lenguaje incorporando características tales como cuantificadores existenciales para la cabeza de reglas y al mismo tiempo, restricciones en reglas de sintaxis para alcanzar decidibilidad y tratabilidad. La familia de lenguajes de ontologías Datalog permite un estilo modular de representación de conocimiento mediante el uso de reglas de forma similar a la usada en Programación Lógica, y su decidibilidad le permite manejar los volúmenes masivos de datos que podemos encontrar en aplicaciones hoy en día, haciéndola útil en diferentes campos como la consulta de ontologías, extracción de datos en web o intercambio de datos [LMS12]. En particular, 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, e.g., alumno(pedro) (b) Tuple-generating Dependencies - TGDs: reglas que nos permiten obtener nuevos átomos mediante la activación

de las mismas como ser: $\text{alumno}(X) \rightarrow \text{persona}(X)$, (c) Equality-generating Dependencies: reglas que restringen la generación de átomos, por ejemplo: $\text{doctor}(D, P) \wedge \text{doctor}(D', P) \rightarrow D = D'$; y (d) Negative Constraints –NCs: reglas que expresan relaciones que no pueden existir entre átomos, e. g.: $\text{alto}(X) \wedge \text{bajo}(X) \rightarrow \perp$. Los métodos a implementar se enfocarán, por lo tanto, en controlar la relación entre los átomos y las TGDs presentes en la ontología, de forma que se respeten las EGDs y NCs en la KB.

Para ello se tomará como marco teórico los operadores de consolidación de ontologías Datalog+/- presentados en [Dea15, DMFS16] que, basándose en técnicas de Revisión de Creencias, atacan los conflictos citados anteriormente (inconsistencia e incoherencia). El primer enfoque (basado en las ideas de Hansson [Han94, Han99]) consiste atacar los conflictos, identificando conflictos mínimos conocidos como kernels y removiendo de ellos fórmulas para lograr resolverlos. En el segundo enfoque se utiliza una mirada más global, considerando la relación existente entre los diferentes kernels mediante la agrupación de los mismos en una superestructura denominada cluster, la cual es obtenida explotando una relación de solapamiento entre conflictos.

3. Líneas de Investigación y Desarrollo

Esta línea de investigación se enfoca en la implementación de procesos de resolución de conflictos en ontologías Datalog+/- a través del uso de formalismos enfocados en la resolución de incoherencias e inconsistencias provenientes de las áreas de Revisión de Creencias y Argumentación. Para ello distintos ejes deben ser investigados, que van desde la definición de incoherencias e inconsistencia en el entorno de ontologías Datalog+/- hasta las posibles aplicaciones que un método automático de integración de estas ontologías podría tener.

3.1. Definición de métodos de identificación de Incoherencias e Inconsistencias en Datalog+/-

Datalog+/- se ha vuelto un lenguaje muy popular en los últimos años, y numerosos estudios se han realizado acerca de sus propiedades de decidibilidad y la complejidad asociada a la respuesta de consultas en estas ontologías. Sin embargo, no ha habido mucho estudio acerca de los aspectos de Representación de Conocimiento en Datalog+/- . Muy poco trabajo se ha hecho acerca de inconsistencias en ontologías Datalog+/- . Peor aún es la situación respecto del concepto de incoherencia (*i. e.*, la imposibilidad de satisfacer cierto conjunto de reglas sin violar una restricción impuesta al conocimiento expresado en la ontología).

Uno de los ejes de esta línea de investigación es la definición formal del concepto de incoherencia en Datalog+/-, tomando como partida esfuerzos similares que han sido realizados para otros formalismos de representación de conocimiento, principalmente Description Logics. Adicionalmente, se procederá a identificar las propiedades que llevan a que un conjunto de TGDs sea incoherente, y las que hacen que una ontología Datalog+/- se vuelva inconsistente. De esta manera se podrán identificar tales casos, lo que será el primer paso para la posterior resolución de tales problemas.

Esto es sobremanera importante a fines de los objetivos de la presente línea de investigación, ya que es necesario tener identificadas tanto las características de cada tipo de conflicto como la relación entre las mismas para poder realizar una detección y tratamiento eficientes de conflictos. Para ver este más claramente considere el caso de la incoherencia: la detección de la misma involucra una aplicación iterativa de las TGDs, lo que debido a la suposición de mundo abierto es potencialmente infinita. Por lo tanto se deben definir métodos algorítmicamente realizables de detección de conjuntos insatisfacibles de TGDs minimales.

3.2. Resolución de Incoherencias e Inconsistencias en Datalog+/-

Una vez que se tiene definidos e identificados los conjuntos incoherentes de TGDs y aquellos conjuntos de átomos que provocan inconsistencias en la unión de varias ontologías Datalog+/-, se debe proceder a la resolución de estos conflictos. En esta línea de investigación esto se hará mediante la aplicación de técnicas derivadas de la Revisión de Creencias denominadas Kernel Contraction.

Este tipo de técnicas resuelve conflictos de incoherencia/inconsistencia tomando los conjuntos conflictivos mínimos y eligiendo de alguna forma que elemento remover de los mismos para solucionar el problema. En el caso de integración de ontologías Datalog+/- esto es la remoción de ciertos átomos y ciertas TGDs de la unión de todas las ontologías que se está integrando. Adicionalmente, se puede pensar en la definición de métodos de debilitamiento de reglas, en lugar de la remoción de las mismas. Esto no es una tarea trivial, ya que desde un punto de vista implementacional esto implica diversas decisiones de diseño que pueden tener un alto impacto en la eficiencia de los operadores implementados. Por ejemplo, es importante definir de manera adecuada como elegir el mejor candidato entre los átomos o TGDs que pueden eliminarse, lo que a su vez lleva a definir formas (automáticas) de obtener órdenes entre los candidatos. Como tales órdenes serán utilizados permanentemente por los operadores (cada vez que un conflicto deba ser resuelto) los algoritmos a desarrollar deberán utilizar estructuras de datos lo suficientemente ágiles en su dinamicidad para

permitir un reordenamiento eficiente de sus elementos. En particular, es necesario examinar distintas opciones de estructuras de datos y estructuras de control para encontrar aquellas más adecuadas para los fines planteados, tanto en lo referido a la detección de conflictos como a su posterior resolución.

3.3. Análisis computacional del comportamiento de los operadores de consolidación

Como se mencionó previamente, una línea importante de investigación involucra un estudio del comportamiento de los operadores desarrollados. Tal análisis involucra diferentes aspectos que deben ser considerados. Por un lado, es importante el realizar un estudio de la complejidad computacional de las diferentes tareas que involucra la resolución de incoherencias e inconsistencias, para establecer la tratabilidad o no de la tarea. Por otro lado, es interesante el saber como afecta el crecimiento de las ontologías (y de esta manera posiblemente la cantidad de conflictos a resolver) en el desempeño de los algoritmos desarrollados. Para esto se procederá a realizar diferentes análisis prácticos de la eficiencia de los algoritmos, aplicándolo a diferentes ontologías en distintos entornos de aplicación.

4. Resultados y Objetivos

El objetivo general de este trabajo es comprobar empíricamente la correctitud, computabilidad y eficiencia de los operadores de consolidación de ontologías en Datalog+/- . Para esto se analizará a fondo las diferentes características tanto de las ontologías como de los conflictos que en las mismas pueden aparecer, para de esta forma detectar aquello que pueda afectar, tanto positiva como negativamente, la eficiencia en la resolución de conflictos.

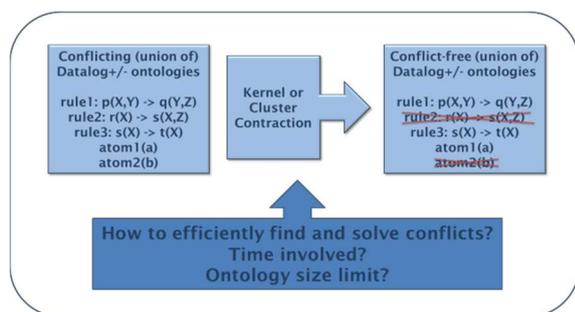


Figura 1: Aspectos a considerar.

Para conseguir este objetivo general hay diferentes objetivos específicos a perseguir

4.1. Estudiar los componentes intervinientes en la ejecución de un programa Datalog+/-

Se analizarán los diferentes componentes intervinientes en la resolución de consultas en Datalog+/- , y cómo los mismos colaboran a tal fin. De ser necesario se complementará la documentación existente en la literatura con diagramas explicativos tanto de funcionamiento de componentes aislados como de interacciones ocurrientes.

4.2. Diseñar e implementar diferentes soluciones para las operaciones requeridas

Se diseñarán de forma abstracta diferentes algoritmos que implementen las operaciones de consolidación en ontologías Datalog+/- . Esta tarea no es trivial ya que los diferentes tipos de conflictos posibles en una ontología Datalog+/- (esto es, incoherencias e inconsistencias) poseen características que los distinguen y pueden dificultar la tarea de encontrar los conflictos minimales (*kernels* en la terminología de Hansson [Han99]). Esto es particularmente cierto para el fenómeno de incoherencia, ya que la aplicación de reglas en Datalog+/- puede provocar una secuencia infinita, por lo que mecanismos adecuados deben ser ideados al momento de buscar los conjuntos insatisfacibles de reglas [DMFS16] para evitar estos inconvenientes derivados del uso de reglas existencialmente cuantificadas en su cabeza. Para ellos se estudiarán el uso de diferentes estructuras de control y de datos según el caso lo requiera.

4.3. Realizar análisis de complejidad y eficiencia de las soluciones propuestas

Las soluciones propuestas en el apartado anterior serán sometidas a análisis de complejidad y eficiencia. Se analizarán los recursos computacionales de tiempo y espacio que cada solución exija, para determinar su aplicabilidad en dominios de aplicación de la vida real.

4.4. Realizar pruebas unitarias, de integración y funcionales

Se utilizarán frameworks de pruebas con la finalidad de asegurar el correcto funcionamiento de los componentes del software de manera aislada y en grupo. Se pondrán a prueba los operadores desarrollados utilizando diferentes programas Datalog+/- .

5. Formación de Recursos Humanos

En la presente línea de investigación se enmarca el desarrollo de una tesis para la Maestría en Sistemas de la Información de la Universidad Nacional de Entre Ríos.

Referencias

- [AGM85] Carlos Alchourrón, Peter Gärdenfors, and David Makinson. On the logic of theory change: Partial meet contraction and revision functions. *J. Symb. Log.*, 50(2):510–530, 1985.
- [AK05] L. Amgoud and S. Kaci. An argumentation framework for merging conflicting knowledge bases: The prioritized case. In *ECSQARU 2005*, pages 527–538, 2005.
- [BB97] D. Beneventano and S. Bergamaschi. Incoherence and subsumption for recursive views and queries in object-oriented data models. *Data Knowl. Eng.*, 21(3):217–252, 1997.
- [BHP09] E. Black, A. Hunter, and J. Z. Pan. An argument-based approach to using multiple ontologies. In *SUM*, pages 68–79, 2009.
- [BQL07] D. A. Bell, G. Qi, and W. Liu. Approaches to inconsistency handling in description logic based ontologies. In *OTM Workshops (2)*, pages 1303–1311, 2007.
- [CGL12] A. Cali, G. Gottlob, and T. Lukasiewicz. A general datalog-based framework for tractable query answering over ontologies. *J. Web Sem.*, 14:57–83, 2012.
- [Dal88] Mukesh Dalal. Investigations into a theory of knowledge base revision. In *AAAI*, pages 475–479, 1988.
- [Dea15] Cristhian Ariel David Deagustini. *Consolidación de Ontologías Datalog ±*. PhD thesis, Departamento de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur, 2015.
- [DMFS16] Cristhian Ariel David Deagustini, Maria Vanina Martinez, Marcelo A. Falappa, and Guillermo Ricardo Simari. Datalog+ ontology consolidation. *J. Artif. Intell. Res. (JAIR)*, 56:613–656, 2016.
- [FHP+06] G. Flouris, Z. Huang, J. Z. Pan, D. Plexousakis, and H. Wache. Inconsistencies, negations and changes in ontologies. In *AAAI*, pages 1295–1300. AAAI Press, 2006.
- [Gär03] Peter Gärdenfors. *Belief revision*, volume 29. Cambridge University Press, 2003.
- [GCS10] S. A. Gómez, C. I. Chesñevar, and G. R. Simari. Reasoning with inconsistent ontologies through argumentation. *Appl. Artif. Intell.*, 24(1&2):102–148, 2010.
- [GSo4] A. J. García and G. R. Simari. Defeasible logic programming: An argumentative approach. *TPLP*, 4(1-2):95–138, 2004.
- [Han94] S. O. Hansson. Kernel contraction. *J. Symb. Log.*, 59(3):845–859, 1994.
- [Han99] S. O. Hansson. *A Textbook of Belief Dynamics*. Kluwer Academic Publishers, 1999.
- [HvHtTo5] Zhisheng Huang, Frank van Harmelen, and Annette ten Teije. Reasoning with inconsistent ontologies. In *Proc. of IJCAI 2005*, pages 454–459, 2005.
- [KPSH05] A. Kalyanpur, B. Parsia, E. Sirin, and J. A. Hendler. Debugging unsatisfiable classes in owl ontologies. *J. Web Sem.*, 3(4):268–293, 2005.
- [LMS12] T. Lukasiewicz, M. V. Martinez, and G. I. Simari. Inconsistency handling in datalog+/- ontologies. In *Proc. of ECAI*, pages 558–563, 2012.
- [Pra10] Henry Prakken. An abstract framework for argumentation with structured arguments. *Argument and Computation*, 1:93–124, 2010.
- [QH07] Guilin Qi and Anthony Hunter. Measuring incoherence in description logic-based ontologies. In *ISWC/ASWC*, pages 381–394, 2007.
- [QLBo6] Guilin Qi, Weiru Liu, and David A. Bell. Knowledge base revision in description logics. In *JELIA*, pages 386–398, 2006.
- [RSvBo9] I. Rahwan, G. R. Simari, and J. van Benthem. *Argumentation in artificial intelligence*, volume 47. Springer, 2009.
- [SHCvHo7] Stefan Schlobach, Zhisheng Huang, Ronald Cornet, and Frank van Harmelen. Debugging incoherent terminologies. *J. Autom. Reasoning*, 39(3):317–349, 2007.
- [SL92] G. R. Simari and R. P. Loui. A mathematical treatment of defeasible reasoning and its implementation. *Artif. Intell.*, 53(2-3):125–157, 1992.