

# Formalización y Generalización del Manejo de Preferencias en Servicios de Razonamiento Rebatible

**Juan Carlos Lionel Teze**

Directores de tesis: **Guillermo R. Simari y Alejandro J. García**

Fecha de exposición: 30/03/2017

Instituto de Ciencias e Ingeniería de la Computación (ICIC),  
Dep. de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur  
Laboratorio de Investigación y Desarrollo en Inteligencia Artificial Concordia,  
Facultad de Ciencias de la Administración, Universidad Nacional de Entre Ríos  
Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)  
Av. Alem 1253, (B8000CPB) Bahía Blanca, Argentina  
Tel: (0291) 459-5135 / Fax: (0291) 459-5136  
email:jct@cs.uns.edu.ar

## Resumen

En los últimos años, la argumentación rebatible ha realizado un importante aporte a la Inteligencia Artificial, hecho que se refleja en el creciente número de aplicaciones del mundo real que la incluyen como formalización del razonamiento del sentido común. En este sentido, los sistemas argumentativos proponen una formalización de este tipo de razonamiento justamente utilizando como mecanismo de inferencia la argumentación rebatible. Intuitivamente, la argumentación rebatible provee formas de confrontar declaraciones contradictorias para determinar si alguna afirmación puede ser aceptada o rechazada. Para obtener una respuesta, el proceso de razonamiento argumentativo lleva a cabo una serie de etapas. Una etapa muy importante es la comparación de argumentos en conflicto para decidir qué argumento prevalece; ésto requiere introducir un criterio de preferencia entre argumentos que haga frente a esta situación.

Esta tesis aborda el estudio, diseño y formalización de herramientas computacionales concretas para seleccionar y cambiar el criterio de preferencia entre argumentos que es utilizado por el sistema de Programación Lógica Rebatible (DeLP) requerido para decidir derrotas al analizar ataques entre argumentos. Para lograr esto, se proponen varios servicios de razonamiento basados en DeLP que disponen de distintos criterios y permiten llevar a cabo esta tarea de diferentes maneras. Como parte de la contribución, se propone un servicio que utiliza expresiones condicionales para programar cómo seleccionar el criterio que mejor se ajusta a las preferencias del usuario o a una situación en particular. Por otra parte, en la tesis se aborda también la definición de un servicio con mecanismos que permiten no solo seleccionar sino también combinar criterios. Estos mecanismos permiten que sea posible comparar argumentos considerando de manera simultánea más de un criterio.

# 1. Introducción y Motivación

Los formalismos propuestos en esta tesis incorporan herramientas concretas para tratar el manejo de múltiples criterios de preferencia entre argumentos, lo cual no ha sido considerado hasta el momento por otros trabajos. De esta manera, los resultados obtenidos brindan una contribución importante a los desarrollos en la comunidad de argumentación, particularmente en el campo de los sistemas basados en Programación Lógica Rebatible, significando además un aporte dentro del área de Inteligencia Artificial en las Ciencias de la Computación. En lo que resta de esta sección se presenta brevemente el contexto que motivo el desarrollo de las contribuciones de esta tesis.

## 1.1. Argumentación Rebatible

Desde hace tiempo, la argumentación ha evolucionado como una propuesta atractiva para modelar el razonamiento basado en sentido común [1, 2, 3], que usualmente sucede en el contexto de información contradictoria, incompleta e incierta. Existen diferentes acercamientos para modelar este proceso, tales como los sistemas argumentativos abstractos [4, 5] que se abstraen de la estructura interna de los argumentos, o los sistemas argumentativos estructurados [6, 7], que sí tienen en cuenta la estructura interna de los mismos. Aquí nos limitamos a estos últimos.

Los sistemas argumentativos estructurados son formalismos de argumentación basados en una lógica subyacente específica, la cual se emplea para representar el conocimiento acerca del dominio sobre el que se razona. Además, esta lógica cuenta con un conjunto de reglas de inferencia que permiten construir argumentos relacionados con una conclusión. Estos sistemas son de gran interés para la comunidad de Inteligencia Artificial dado que las reglas de inferencia permiten representar conocimiento de sentido común, posibilitando la construcción computacional de argumentos. Los sistemas argumentativos estructurados poseen características que los hacen especialmente aptos para su implementación; por otra parte, los sistemas argumentativos son particularmente atractivos para la toma de decisiones y la negociación [8, 9], áreas de gran interés en diversas aplicaciones.

## 1.2. Programación Lógica Rebatible y la Comparación entre Argumentos

Como se ha mencionado, la importancia del uso de sistemas argumentativos como mecanismo de razonamiento en sistemas inteligentes es reconocida en diversos trabajos en la literatura [2, 10, 11]. Esta tesis se enfoca en un sistema argumentativo estructurado, denominado Programación Lógica Rebatible (DeLP por sus siglas en inglés de *Defeasible Logic Programming*) [7]. Este formalismo combina resultados de programación en lógica y argumentación rebatible, y ha sido aplicado exitosamente en diferentes dominios (ver *e.g.*, [12, 13, 14]). DeLP extiende a la Programación en Lógica permitiendo representar conocimiento potencialmente contradictorio, mediante el uso de la negación fuerte y conocimiento tentativo.

El procedimiento de prueba de DeLP se basa en un proceso de análisis dialéctico donde interactúan argumentos a favor y en contra de un literal a fin de determinar si ese literal está garantizado. Este proceso conducirá a la construcción de una estructura arbórea de derrotadores. De este modo, un literal se hallará garantizado si existe un argumento para ese literal que sobreviva a todas las derrotas que recibe en su árbol asociado. Para determinar si un argumento es un derrotador de otro con el que está en conflicto es necesario de un criterio de preferencia que decida preferencia entre estos argumentos.

Si bien al presentarse DeLP en [7] se asoció el criterio de Especificidad Generalizada como criterio por defecto, en DeLP el criterio es un elemento modular. Sin embargo, en la comunidad de argumentación no existe un consenso establecido acerca de qué criterio utilizar para evaluar argumentos. En este sentido, en trabajos como [15, 16, 17] los autores declaran que el criterio de especificidad no corresponde a un criterio general del razonamiento de sentido común, sino que simplemente es uno de muchos métodos de comparación que podrían ser utilizados. Más aún, propuestas como [18, 15] sugieren que la información acerca del dominio suele ser la herramienta principal para evaluar los argumentos del sistema. De allí la importancia de contar con un sistema como [7] que permita tratar la comparación de argumentos de forma modular, permitiendo que el usuario emplee el criterio que mejor se ajusta al dominio de aplicación reemplazando el provisto por defecto como parte del sistema.

Existen varios criterios concretos para comparar argumentos en DeLP (ver *e.g.*, [7, 19]). No obstante, los sistemas que utilizan DeLP como mecanismo de razonamiento y consideran estos criterios en sus formalismos [20, 21, 22, 23, 24], por lo general adoptan un criterio o una combinación fija de criterios (establecida en la configuración del sistema) de acuerdo al dominio que se está representando. Si bien en DeLP la comparación de argumentos es modular, en la literatura existente no se han propuesto mecanismos programables concretos que le permitan al usuario seleccionar y cambiar dinámicamente el criterio de preferencia dependiendo de sus preferencias o necesidades. Si un usuario conoce las razones por las cuales se prioriza cierta información sobre otra, el mecanismo para comparar argumentos deja de ser una caja negra, aumentando la confianza del usuario sobre las respuestas que puede recibir de estos sistemas en particular. Así el sistema se vuelve más confiable y transparente para el usuario ya que éste puede interactuar directamente con el mecanismo de razonamiento de dicho sistema.

Otra característica de los sistemas basados en DeLP, respecto al manejo de los mecanismos de comparación sobre argumentos, es que usualmente tampoco disponen de varios criterios para que el usuario pueda elegir el que más se adecua a un contexto determinado. En este sentido, en la vida real el proceso de razonamiento humano hace frente a una situación particular teniendo en cuenta diferentes criterios. Por ejemplo, supongamos una persona que se quiere hospedar en un hotel y consulta un sitio web que le ofrece información respecto a varios hoteles. Para elegir un hotel, esta persona podría considerar varios criterios que le permitan comparar la información que le brinda el sitio. Un posible criterio podría preferir aquella información que favorece al confort de las habitaciones, mientras que otro criterio podría preferir la información que favorece a la seguridad del lugar en donde se encuentra el hotel. Finalmente, un tercer criterio podría ser uno que combine los dos anteriores, es decir un criterio que priorice la información relacionada al confort y la seguridad. Por lo tanto, disponer de diferentes criterios de preferencias introduce un grado de flexibilidad adicional a los sistemas basados en DeLP. Sin embargo, y como se puntualizó anteriormente, en estos sistemas el criterio es un componente fijo, o si existe multiplicidad de criterios, no hay forma de cambiarlo una vez que uno es seleccionado e integrado al intérprete. De esta manera, una de las motivaciones de esta tesis es mejorar las capacidades del mecanismo de inferencia de un sistema basado en argumentación para que se pueda adaptar de una forma natural a varios criterios, a través de un mecanismo programable concreto.

A partir de lo mencionado anteriormente, las contribuciones de esta tesis conciernen a la formalización de varios servicios de razonamiento basados en DeLP que abordan los temas planteados en esta sección. En esta dirección, sería posible lograr una satisfactoria integración de razonadores basados en DeLP en diferentes dominios de aplicación, tales como agentes autónomos, sistemas de soporte a las decisiones, búsqueda inteligente en la web, sistemas de recomendación, y otros dominios de similar importancia.

## 2. Aportes a la disciplina

Como aportes principales de esta tesis se proponen tres servicios de razonamiento basados en DeLP: uno de ellos es el *Servicio de Razonamiento Rebatible basado en Preferencias*, y sus dos extensiones el *Servicio de Razonamiento Rebatible basado en Preferencias Condicionales* y el *Servicio de Razonamiento Rebatible basado en Preferencias Combinadas*. Los resultados de las contribuciones de esta tesis han sido publicados, con referato nacional e internacional, en:

- “*Improving argumentation-based recommender systems through context-adaptable selection criteria*” publicado en la revista *Expert Systems with Applications* vol. 42.
- “*An Approach to Argumentative Reasoning Servers with Multiple Preference Criteria*” publicado en la revista *Inteligencia artificial: Revista Iberoamericana de Inteligencia Artificial* vol. 53.
- “*Servicios de Razonamiento con Múltiples Criterios de Preferencia*” publicado en XVI *Workshop de Investigadores en Ciencias de la Computación (WICC 2014)*.
- “*An Approach to Argumentative Reasoning Servers with Conditions based Preference Criteria*” publicado en XIX *Congreso Argentino de Ciencias de la Computación (CACIC 2013)*.
- “*Modelo de Servicio de Razonamiento con Preferencias*” publicado en el XV *Workshop de Investigadores en Ciencias de la Computación (WICC 2013)*.

## 3. Contribuciones de la tesis

El objetivo de esta línea de investigación es mejorar las capacidades de razonamiento efectivo de sistemas basados en programación lógica rebatible introduciendo diferentes herramientas programables concretas que permitan tratar el manejo de multiplicidad de criterios. En términos generales, en este trabajo se presentan tres diferentes servicios de razonamiento. A continuación los presentaremos brevemente.

### 3.1. Servicio de Razonamiento Rebatible basado en Preferencias

En esta sección se presenta la noción de *Servicio de Razonamiento Rebatible basados en Preferencias (SRPref)*, el cual representa el primer aporte de esta tesis. Estos servicios proveen un razonamiento lógico rebatible que cuenta con la capacidad de modificar las preferencias sobre la información procesada cambiando, a partir de cada consulta recibida, el criterio de preferencia utilizado. En particular, el mecanismo que se utiliza para consultar un *Servicio de Razonamiento Rebatible basado en Preferencias (SRPref)* es el de las *Consulta basada en Preferencias*. Entonces, un servicio de razonamiento rebatible basado en preferencias está conformado por:

- el *intérprete* DeLP que se encarga de procesar las consultas hechas al servicio; y
- un *módulo de comparación de argumentos* que, a través de la *función de cómputo de preferencias*, obtiene las preferencias entre argumentos que el intérprete necesita resolver utilizando para ello una de las *implementaciones de criterios de preferencia* almacenadas en el servicio.

Un *SRPref* corresponde a una entidad con la capacidad de responder consultas DeLP utilizando el criterio de preferencia entre argumentos que el usuario especifica junto a su consulta y el programa consultado. Un *SRPref* cuenta con la implementación de varios criterios disponibles para el usuario. Por otra parte, como herramienta de interacción entre el usuario y un *SRPref* se introdujo el concepto de *Consulta basada en Preferencias (CPref)*. Una *CPref* se compone además de la consulta propiamente dicha, del programa DeLP que será procesado y una especificación declarativa del criterio de preferencia que el usuario desea utilizar. En la *especificación de criterio* también se puede incluir información adicional que el criterio elegido podría llegar a necesitar para poder ejecutarse. Por lo tanto, todos estos elementos serán los que un *SRPref* tendrá en cuenta al momento de computar la respuesta a una consulta. Es importante destacar que la *especificación de criterio* constituye la herramienta de interacción que utilizarán todos los servicios de razonamientos propuestos en esta tesis para seleccionar y cambiar de criterio.

En la Figura 1, las flechas muestran cómo son utilizados los datos por el *SRPref*. Entonces, a partir de un programa DeLP y una consulta DeLP, el intérprete calcula la respuesta para la consulta, y el *módulo de comparación de argumentos* computa las preferencias entre argumentos en conflicto que recibe del intérprete. La función de cómputo de preferencias se encargará de ejecutar la implementación del criterio indicado en la especificación de criterio de la consulta. Note que la notación gráfica en Figura 1 para los argumentos es mediante triángulos.

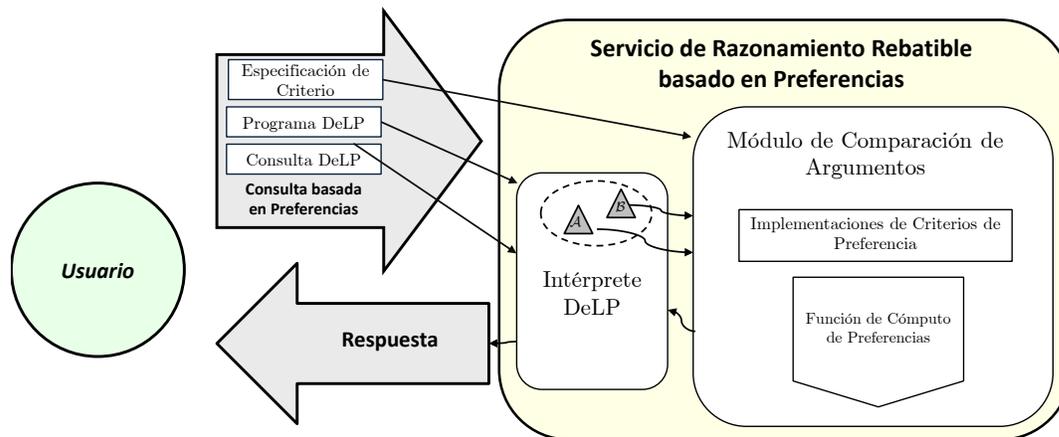


Figura 1: Respuesta a una Consulta basada en Preferencias.

### 3.2. Servicio de Razonamiento Rebatible basado en Preferencias Condicionales

En primer lugar se presentó el concepto de *Consulta basada en Preferencias Condicionales (CPCCond)* que ofrece una alternativa distinta respecto a la forma en la cuál un usuario puede expresar sus preferencias. Estas consultas están constituidas por una expresión formal que permite decidir cuál es el criterio de preferencia que debe ser utilizado en cada situación específica. Para resolver estas consultas se formalizó un nuevo tipo de servicio de razonamiento que extiende las capacidades de los *SRPrefs*, denominado *Servicio de Razonamiento Rebatible basado en Preferencias Condicionales (SRPCCond)*. Estos servicios tienen la capacidad de responder consultas seleccionando un criterio de preferencia que es indicado utilizando una

expresión condicional incluida en la propia consulta.

Entonces, un *SRPCond* está conformado por:

- el *intérprete* DeLP que resuelve las consultas DeLP,
- una *función de evaluación* que, a partir de una *expresión de preferencia condicional*, obtiene la *especificación de criterio* que declara la implementación de criterio que se utilizará en la comparación de argumentos, y
- un *módulo de comparación de argumentos* que computa las preferencias entre argumentos considerando para ello la especificación de criterio que se obtiene desde la función de evaluación.

Dada la importancia de las *expresiones de preferencias condicionales* como herramienta computacional concreta para guiar la selección de un criterio se presentó una representación de árbol con el fin de proporcionar una forma de analizar varias propiedades de estas expresiones. Estas propiedades son útiles para identificar cuando una expresión puede ser optimizada y de esta manera evitar la computación de literales redundantes, y además caracterizar cuando ciertos caminos en la expresión no son transitables. Estas propiedades son de especial interés en esta tesis por que permiten construir expresiones válidas; es decir, expresiones que mantienen relaciones coherentes entre las guardas que justifican la elección de un criterio en particular.

Un *SRPComb* responde *CPCConds*. Una *CPCCond* se caracteriza por incluir junto a la consulta y el programa a consultar, la especificación de una expresión condicional que servirá para programar cómo seleccionar el criterio que el *SRPCond* consultado utilizará a partir de los criterios especificados. La selección de un criterio en particular depende de la existencia de determinada información alojada en el programa consultado. La Figura 2 muestra de forma esquemática los elementos involucrados en el cálculo de la respuesta para una *CPCCond*. La interpretación de las flechas en el gráfico es la misma que se utilizó en la Figura 1. Por ejemplo, la expresión de preferencia condicional y el programa DeLP son utilizados por la función de evaluación para seleccionar una *especificación de criterio*. Luego, a partir de esta especificación y los argumentos enviados por el intérprete DeLP, el módulo de comparación de argumentos devuelve una preferencia entre dichos argumentos. Finalmente, a partir del programa DeLP y la consulta DeLP, el intérprete obtiene una respuesta para la consulta.

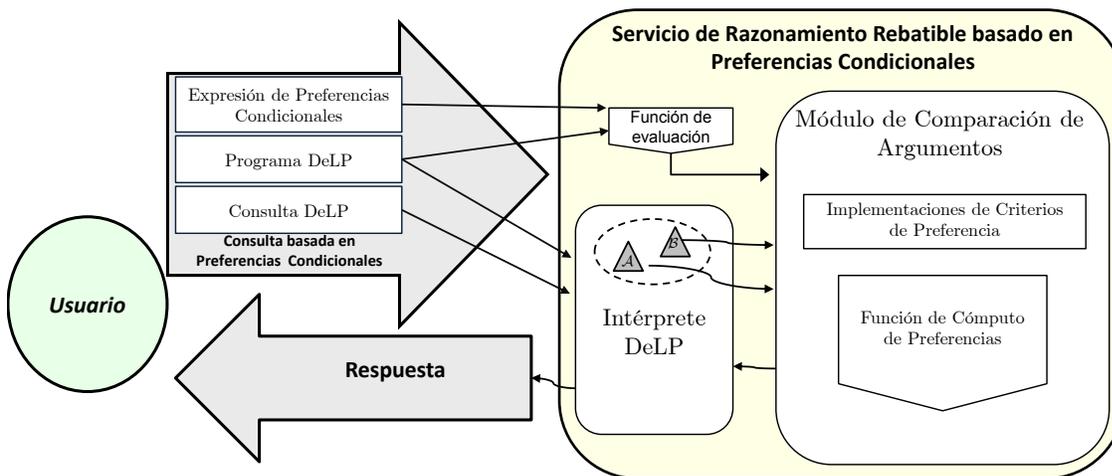


Figura 2: Principales componentes de una *CPCCond* y un *SRPCond*.

### 3.3. Servicio de Razonamiento Rebatible basado en Preferencias Combinadas

Otro tipo de consultas propuesto en este trabajo son las *Consultas basadas en Preferencias Combinadas (CPComb)*, caracterizadas por incluir una expresión que permite combinar la especificaciones de varios criterios. Para resolver las *CPCombs* se definió un tipo de servicio de razonamiento que extiende las capacidades de un *SRPCond*, denominado *Servicio de Razonamiento Rebatible basado en Preferencias Combinadas (SRPComb)*. Una característica distintiva de estos servicios corresponde a la capacidad de poder usar varios criterios a la vez para evaluar la preferencia entre argumentos. Es decir, estos servicios proveen operadores específicos que permiten a las consultas considerar un uso combinado de varios criterios de preferencia.

Como se muestra de manera esquemática en la Figura 3 los pasos para responder una *CPComb* coinciden en ciertos aspectos con los descriptos para los servicios de razonamiento presentados en secciones anteriores. Es decir, un *SRPComb* cuenta con un intérprete que responde consultas DeLP, y además se comunica con un módulo encargado de resolver las preferencias que éste necesita. Sin embargo, observe que la arquitectura de un *SRPComb* está diseñada de manera tal que puede recibir expresiones que contienen *operadores de combinación de preferencias* utilizados para combinar criterios. Por lo tanto, una vez que la función de evaluación obtiene desde estas expresiones los criterios que se utilizarán, el *módulo extendido de comparación de argumentos* es capaz de llevar a cabo todas las comparaciones entre argumentos que el intérprete le solicita. Para programar la selección de expresiones que incluyan operadores para combinar criterios se formalizó la noción de *expresión de preferencias combinadas*.

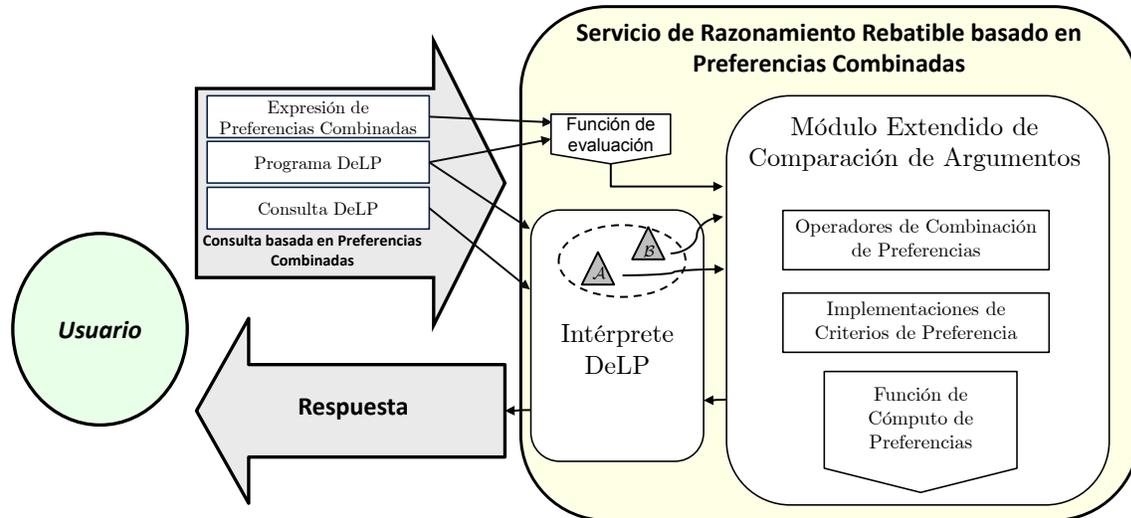


Figura 3: Principales componentes de una *CPComb* y un *SRPComb*.

Se presentaron varios resultados. En primer lugar se estudió un aspecto poco deseable en sistemas argumentativos que es la indecisión por incomparabilidad. Dentro de los resultados mostrados se establecieron algunas propiedades que sirven para establecer cuándo la relación que resulta de aplicar un criterio de preferencia o una operación de combinación de criterios es apropiada a una consulta. Finalmente, se mostraron varios criterios y operadores que mantienen dichas propiedades.

## 4. Conclusión

A pesar de la importancia que tiene la comparación de argumentos en el proceso de argumentación, los formalismos desarrollados hasta el momento no han considerado mecanismos computacionales que permitan cambiar dinámicamente el criterio de preferencia. Esto motivó el estudio de diferentes herramientas concretas que permitan tratar el manejo de múltiples criterios en aquellos formalismos cuyo mecanismo de inferencia sea DeLP, a fin de poder seleccionar y cambiar de criterio de una manera natural y acorde a las preferencias o necesidades del usuario.

En esta tesis se presentó un servicio de razonamiento basado en la programación lógica rebatible, denominado Servicio de Razonamiento Rebatible basado en Preferencias (*SRPref*), que puede cambiar de criterio de preferencia para cada consulta DeLP recibida. Como herramienta de interacción para consultar estos servicios se introdujo la Consulta basada en Preferencias (*CPref*). Esta herramienta permite especificar de forma declarativa el criterio que será utilizado por el *SRPref* que recibe la consulta.

Se presentó el concepto de Servicio de Razonamiento Rebatible basado en Preferencias Condicionales (*SRPCond*) y se introdujo un tipo especial de consulta para estos servicios, denominada Consulta basada en Preferencias Condicionales (*CPCond*), que le permite a un *SRPCond* seleccionar el criterio que deberá utilizar dependiendo de ciertas condiciones. Para esto la *CPCond* incorpora una expresión de preferencia condicional. Se presentó también una representación de árbol para el estudio de las expresiones condicionales.

Una de las motivaciones importantes en esta tesis ha sido introducir la posibilidad de comparar argumentos considerando varios criterios al mismo tiempo. Siguiendo esta idea se presentó la Consulta basada en Preferencias Combinadas (*CPComb*). Una *CPComb* permite que se pueda especificar el uso de más de un criterio a partir de una expresión especial construida mediante operadores de combinación de preferencias que el servicio consultado dispone.

## 5. Líneas de investigación futura

Como trabajo futuro se implementarán los desarrollos de esta tesis para poder ejercitar los formalismos y desarrollar nuevas alternativas a partir de las aplicaciones. A partir de los trabajos desarrollados en la tesis se abren varias líneas de investigación sobre las cuales se planea seguir trabajando.

- Se buscará tratar con la formalización de otras herramientas de interacción. Ésto permitirá definir nuevas consultas basadas en preferencias que cumplan con nuevos requisitos del usuario con respecto a sus necesidades o preferencias. Por ejemplo, un usuario podría querer realizar diferentes consultas DeLP simultáneamente utilizando el mismo criterio y el mismo programa DeLP. En este sentido tener un tipo de consulta que cumpla con estos requisitos mejoraría los tiempos de respuesta de los servicios de razonamiento propuestos en esta tesis.
- A partir de las tareas de investigación se observó que la forma en la que es tratada la comparación de argumentos por los formalismos no siempre es la misma. Algunos sistemas utilizan un enfoque modular [7], otros argumentan sobre el criterio de preferencia [6], mientras que otros utilizan un criterio fijo [25]. En este sentido, un tópico interesante es estudiar si es posible construir un marco general para sistemas argumentativos estructurados que permita el manejo de multiplicidad de criterios. Un primer paso para lograr esto,

es analizar la posibilidad de definir servicios de razonamiento en donde el mecanismo de razonamiento sea el de otro sistema argumentativo estructurado, y no el de DeLP. Para esto será necesario diseñar un marco lo suficientemente general como para ser instanciado por diferentes sistemas de este tipo en particular.

- Finalmente, un tema interesante para analizar consiste en el diseño e implementación de una arquitectura de agentes basada en el esquema de alguno de los servicios de razonamiento propuestos.

## Referencias

- [1] C. I. Chesñevar, A. G. Maguitman, and R. P. Loui, “Logical models of argument,” *ACM Computing Surveys*, vol. 32, no. 4, pp. 337–383, 2000.
- [2] T. J. M. Bench-Capon and P. E. Dunne, “Argumentation in artificial intelligence,” *Artif. Intell.*, vol. 171, no. 10-15, pp. 619–641, 2007.
- [3] P. Besnard and A. Hunter, *Elements of argumentation*. MIT press Cambridge, 2008, vol. 47.
- [4] P. M. Dung, “On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games,” *Artif. Intell.*, vol. 77, no. 2, pp. 321–358, 1995.
- [5] G. Vreeswijk, “Abstract argumentation systems,” *Artif. Intell.*, vol. 90, no. 1-2, pp. 225–279, 1997.
- [6] H. Prakken and G. Sartor, “Argument-based extended logic programming with defeasible priorities,” *Journal of Applied Non-classical Logics*, vol. 7, pp. 25–75, 1997.
- [7] A. J. García and G. R. Simari, “Defeasible logic programming: An argumentative approach,” *Theory and Practice of Logic Programming (TPLP)*, vol. 4, no. 1-2, pp. 95–138, 2004.
- [8] L. Amgoud, S. Parsons, and N. Maudet, “Arguments, dialogue, and negotiation,” in *ECAI 2000, Proceedings of the 14th European Conference on Artificial Intelligence, Berlin, Germany, August 20-25, 2000*, 2000, pp. 338–342.
- [9] E. Black and A. Hunter, “An inquiry dialogue system,” *Autonomous Agents and Multi-Agent Systems*, vol. 19, no. 2, pp. 173–209, 2009.
- [10] C. I. Chesñevar, A. G. Maguitman, and G. R. Simari, “Recommender system technologies based on argumentation 1,” in *Emerging Artificial Intelligence Applications in Computer Engineering*, 2007, pp. 50–73.
- [11] I. Rahwan and G. R. Simari, *Argumentation in Artificial Intelligence*, 1st ed. Springer Publishing Company, Incorporated, 2009.
- [12] N. D. Rotstein, A. J. García, and G. R. Simari, “Reasoning from desires to intentions: A dialectical framework,” in *AAAI*, 2007, pp. 136–141.

- [13] S. A. Gómez, C. I. Chesñevar, and G. R. Simari, “Defeasible reasoning in web-based forms through argumentation,” *International Journal of Information Technology and Decision Making*, vol. 7, no. 1, pp. 71–101, 2008. [Online]. Available: <http://dx.doi.org/10.1142/S021962200800282X>
- [14] S. Gottifredi, A. J. García, and G. R. Simari, “Query-based argumentation in agent programming,” in *Advances in Artificial Intelligence - IBERAMIA 2010, 12th Ibero-American Conference on AI, Bahía Blanca, Argentina, November 1-5, 2010. Proceedings*, 2010, pp. 284–295.
- [15] G. Vreeswijk, “The feasibility of defeat in defeasible reasoning,” in *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR’91). Cambridge, MA, USA, April 22-25, 1991.*, 1991, pp. 526–534.
- [16] J. L. Pollock, *Cognitive Carpentry: A Blueprint for How to Build a Person*. MIT Press, 1995.
- [17] H. Prakken and G. Sartor, “A dialectical model of assessing conflicting arguments in legal reasoning,” in *Logical Models of Legal Argumentation*. Springer, 1996, pp. 175–211.
- [18] K. Konolige, “Defeasible argumentation in reasoning about events.” in *ISMIS*, 1988, pp. 380–390.
- [19] E. Ferretti, M. Errecalde, A. J. García, and G. R. Simari, “Decision rules and arguments in defeasible decision making,” in *Computational Models of Argument: Proceedings of COMMA 2008, Toulouse, France, May 28-30, 2008.*, 2008, pp. 171–182.
- [20] C. A. D. Deagustini, S. E. F. Dalibón, S. Gottifredi, M. A. Falappa, and G. R. Simari, “Consistent query answering using relational databases through argumentation,” in *Database and Expert Systems Applications - 23rd International Conference, DEXA 2012, Vienna, Austria, September 3-6, 2012. Proceedings, Part II*, 2012, pp. 1–15.
- [21] L. H. Tamargo, S. Gottifredi, A. J. García, M. A. Falappa, and G. R. Simari, “Deliberative delp agents with multiple informants,” *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, vol. 15, no. 49, pp. 13–30, 2012.
- [22] L. Godo, E. Marchioni, and P. Pardo, “Extending a temporal defeasible argumentation framework with possibilistic weights,” in *Logics in Artificial Intelligence - 13th European Conference, JELIA 2012, Toulouse, France, September 26-28, 2012. Proceedings*, 2012, pp. 242–254.
- [23] C. E. Briguez, M. Capobianco, and A. G. Maguitman, “A theoretical framework for trust-based news recommender systems and its implementation using defeasible argumentation,” *International Journal on Artificial Intelligence Tools*, vol. 22, no. 4, 2013.
- [24] C. E. Briguez, M. C. Budán, C. A. D. Deagustini, A. G. Maguitman, M. Capobianco, and G. R. Simari, “Argument-based mixed recommenders and their application to movie suggestion,” *Expert Systems with Applications*, vol. 41, no. 14, pp. 6467–6482, 2014.
- [25] T. Wakaki, “Preference-based argumentation capturing prioritized logic programming,” in *Argumentation in Multi-Agent Systems - 7th International Workshop, ArgMAS 2010, Toronto, ON, Canada, May 10, 2010 Revised, Selected and Invited Papers*, 2010, pp. 306–325.