

Extensión de la Argumentación Rebatible para considerar Etiquetas

M. J. Gómez L., C. I. Chesñevar, G. R. Simari y A. J. García

Laboratorio de Investigación y Desarrollo de Inteligencia Artificial,
Departamento de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur,
Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET),
Email: {mjpg,cic,grs,ajg}@cs.uns.edu.ar

RESUMEN

Esta línea de investigación explora la incorporación del manejo de meta-información a la Programación en Lógica Rebatible (DeLP) a través de etiquetas. El propósito principal de esta incorporación es brindar la posibilidad de definir un criterio de comparación apropiado para el dominio de aplicación específico. Mediante este artículo se introduce ℓ -DeLP, una extensión a DeLP que incorpora el manejo de meta-información a través de etiquetas y permite basar en ella el criterio de comparación por mejor. Además se presenta un ejemplo de aplicación mediante el cual se ilustran sus ventajas sobre DeLP.

1. INTRODUCCIÓN Y MOTIVACIONES

La Programación en Lógica Rebatible (DeLP, [3]), un exponente de la Argumentación Rebatible, constituye un formalismo efectivo para representar y razonar acerca de conocimiento tentativo y potencialmente contradictorio. Resulta muy adecuado para aplicaciones donde se agrega o quita información en forma dinámica, y por lo tanto conclusiones obtenidas previamente pueden quedar invalidadas. Estas características hacen de DeLP una herramienta atractiva para representar el conocimiento y definir el razonamiento de agentes inteligentes deliberativos.

No obstante, el éxito de la aplicación de DeLP a un dominio específico depende en gran medida de la definición de un criterio de comparación entre argumentos que capture una noción de mejor argumento apropiada para dicho dominio. Esta noción de mejor argumento, específica para el dominio, podría basarse en información más allá de la modelada a través de las propias reglas del programa, por ejemplo información acerca de las reglas o meta-información (confiabilidad de la

fuente, probabilidad, grado de certeza, prioridad, etc).

Este trabajo introduce ℓ -DeLP, una extensión a DeLP que incorpora el manejo de meta-información a través de etiquetas y permite basar en ella el criterio de comparación por mejor.

A continuación se describe la estructura del artículo. En la sección 2 se introduce un ejemplo de aplicación que se utilizará para ilustrar los conceptos e ideas tratados en el resto del artículo. En la sección 3 se muestra como modelar el ejemplo de aplicación empleando DeLP. En la sección 4 se introduce ℓ -DeLP y en la sección 5 se emplea para incorporar el tratamiento de incertidumbre posibilística al ejemplo. Finalmente, en la Sección 6, se presentan las conclusiones del artículo y se menciona el trabajo futuro.

2. EJEMPLO DE APLICACIÓN

Consideremos un agente inteligente que asiste a una persona en la compra de un auto. Concretamente debe razonar acerca de diferentes características del vehículo para determinar si éste se encuentra o no en buen estado.

Asumamos que el agente cuenta con la siguiente información general:

- Un auto se encuentra en buen estado si su motor y su carrocería están en buen estado.
- Si el motor suena bien, existen razones para pensar que se encuentra en buenas condiciones.
- La aprobación de un mecánico constituye una razón fuerte para pensar que el motor se encuentra en buen estado.
- Si fue usado como taxi, existen razones para pensar que el motor no se encuentra en buen estado.

Financiado parcialmente por Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET) PIP 5050, y por la Agencia Nacional de Promoción Científica y Tecnológica (PICT 2002 Nro. 13096)

- Si es de color blanco y funciona a gas es bastante posible que haya sido usado como taxi.

Además, el agente dispone para un auto en particular de la siguiente información:

- es de color blanco
- funciona a gas
- el motor suena bien
- su carrocería se encuentra en buen estado

El agente podrá incorporar nueva información acerca del vehículo en forma dinámica, variando sus conclusiones consecuentemente.

3. EL EJEMPLO USANDO DeLP

DeLP constituye un formalismo que permite representar y razonar acerca de conocimiento tentativo y potencialmente contradictorio de forma muy natural.

Se utilizará DeLP como formalismo de representación y razonamiento para el agente, empleando especificidad generalizada [3, Sección 3.2.1] como criterio de comparación de argumentos. A continuación se presenta un programa DeLP \mathcal{P} que modela la información descrita en la sección anterior:

$$\mathcal{P} = \left\{ \begin{array}{l} \text{abe} \leftarrow \text{bno_motor}, \text{bno_carr} \\ \text{bno_motor} \prec \text{suen_bien_motor} \\ \text{bno_motor} \prec \text{mecanico_da_ok} \\ \sim \text{auto_buen_estado} \prec \text{fue_taxi} \\ \text{fue_taxi} \prec \text{color_blanco}, \text{tiene_gas} \\ \text{color_blanco} \\ \text{tiene_gas} \\ \text{suen_bien_motor} \\ \text{bno_carr} \end{array} \right\}$$

El comportamiento del agente se define como sigue. Sea q un literal, el agente concluirá q si q se encuentra garantizado a partir de \mathcal{P} . Concluirá $\sim q$ si $\sim q$ se encuentra garantizado a partir de \mathcal{P} . Recordar que nunca pueden estar garantizados dos literales complementarios, como se muestra en [3, Sección 2.9]. Finalmente, presentará *indecisión* respecto a q , si tanto q como $\sim q$ se encuentran no garantizados a partir de \mathcal{P} .

Analicemos como razonará y que concluirá el agente acerca del estado del auto (*abe*) a partir del programa \mathcal{P} . En la figura 1 se muestran los árboles de dialéctica asociados al análisis de la garantía de *abe* y $\sim \text{abe}$. Los argumentos involucrados son $\langle A_1, \text{abe} \rangle$ y $\langle A_2, \sim \text{abe} \rangle$, donde:

$$A_1 = \{ \text{bno_motor} \prec \text{suen_bien_motor} \}$$

y

$$A_2 = \left\{ \begin{array}{l} \sim \text{auto_buen_estado} \prec \text{fue_taxi}, \\ \text{fue_taxi} \prec \text{color_blanco}, \text{tiene_gas} \end{array} \right\}$$

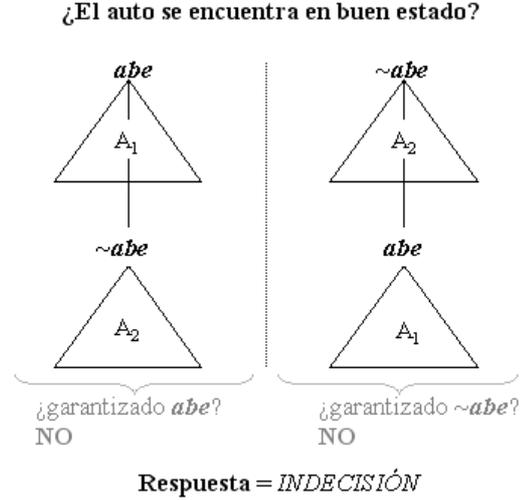


Figura 1: Razonamiento acerca del estado del auto a partir de \mathcal{P} .

Tanto *abe* como $\sim \text{abe}$ se encuentran no garantizados y por lo tanto el agente presentará *indecisión*. La razón de esta situación de indecisión es que ninguno de los dos argumentos en conflicto, A_1 y A_2 , es mejor que el otro de acuerdo al criterio de comparación.

Supongamos que el agente averigua que el auto efectivamente fue usado como taxi. El programa $\mathcal{P}_1 = \mathcal{P} \cup \{ \text{fue_taxi} \}$ refleja el nuevo conocimiento del agente. A partir de \mathcal{P}_1 puede construirse un nuevo argumento $\langle A_3, \sim \text{abe} \rangle$ para $\sim \text{abe}$, donde

$$A_3 = \{ \sim \text{auto_buen_estado} \prec \text{fue_taxi} \}$$

que se basa en menos información rebatible que $\langle A_2, \sim \text{abe} \rangle$ y por lo tanto constituye un argumento más fuerte. No obstante, como se muestra en la figura 2, la conclusión obtenida por el agente acerca del estado del auto vuelve a ser *indecisión*. La causa de la indecisión es la misma que antes: ninguno de los argumentos en conflicto es mejor que el otro de acuerdo al criterio de comparación. Nótese que no es posible *romper* la indecisión agregando $\langle A_3, \sim \text{abe} \rangle$ como hijo de $\langle A_1, \text{abe} \rangle$ en el árbol de dialéctica para $\langle A_2, \sim \text{abe} \rangle$, ya que esto implicaría una situación de doble bloqueo [3, Sección 4.1].

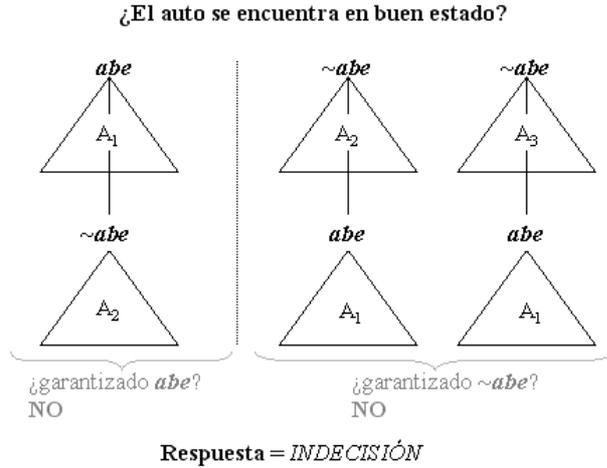


Figura 2: Razonamiento acerca del estado del auto a partir de \mathcal{P}_1 .

Por último, el motor del auto es revisado por un mecánico, quien diagnostica que se encuentra en buen estado. El agente incorpora esta información dando lugar a un nuevo programa $\mathcal{P}_2 = \mathcal{P}_1 \cup \{mecanico_da_ok\}$, lo que le permite construir el argumento $\langle A_4, abe \rangle$, donde:

$$A_4 = \{ bno_motor \prec mecanico_da_ok \}$$

Nuevamente, como se muestra en la figura 3, la conclusión del agente acerca del estado del auto es *indecisión*.

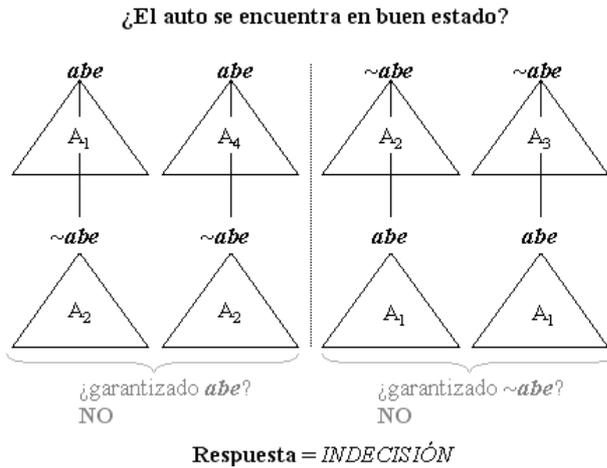


Figura 3: Razonamiento acerca del estado del auto a partir de \mathcal{P}_2 .

4. PROGRAMACIÓN EN LÓGICA REBATIBLE CON ETIQUETAS (ℓ -DeLP)

El éxito de la aplicación de DeLP a un dominio específico depende en gran medida de la definición de un criterio de comparación entre argumentos que capture una noción de mejor argumento apropiada para dicho dominio. Esta noción de mejor argumento, específica para el dominio, podría basarse en información más allá de la modelada a través de las propias reglas del programa. Por ejemplo, podría basarse en información *acerca* de las reglas (meta-información) como probabilidad, grado de certeza, confiabilidad de la fuente, etc.

Esto motiva la siguiente idea: enriquecer el lenguaje DeLP de forma que cada cláusula pueda tener cierta meta-información asociada. La meta-información asociada a una cláusula es simplemente información acerca de o relativa a dicha cláusula. Además, la meta-información asociada a las reglas permitirá sintetizar meta-información asociada a argumentos, en la que luego se basará el criterio de comparación por mejor.

ℓ -DeLP es una forma de materializar la idea recién expresada, utilizando etiquetas para llevar la meta-información. Más concretamente ℓ -DeLP extiende DeLP requiriendo la especificación de:

- una etiqueta junto a cada cláusula del programa.
- una función *ALS* (Argument Label Synthesis) que permite determinar la etiqueta (meta-información) asociada a un argumento en base a las etiquetas de las cláusulas que lo componen.
- una relación o criterio de comparación por mejor entre etiquetas de argumentos *BL* (Better Label).

Describamos informalmente como se obtiene la etiqueta asociada a un argumento mediante la función *ALS*. Para ello consideremos el carácter recursivo de la noción de argumento. Supongamos que contamos con un argumento $\langle A, a \rangle$. Entonces, o bien a es un hecho en el programa y por lo tanto $A = \emptyset$, o a se derivó utilizando una regla r con cabeza a y cuerpo b_1, b_2, \dots, b_n . En este último caso, A resulta de unir B_1, B_2, \dots, B_n , donde $\langle B_i, b_i \rangle$ es subargumento de $\langle A, a \rangle$ para $1 \leq i \leq n$, además de la regla r si es que r es rebatible.

A partir de lo anterior, la etiqueta asociada a $\langle A, a \rangle$ en base a la función *ALS* es

$\ell_{\langle A, a \rangle} = ALS([], \ell_a)$ si a es un hecho y $\ell_{\langle A, a \rangle} = ALS([\ell_{\langle B_1, b_1 \rangle}, \ell_{\langle B_2, b_2 \rangle}, \dots, \ell_{\langle B_n, b_n \rangle}], \ell_r)$ en caso contrario, donde ℓ_a y ℓ_r son las etiquetas asociadas al hecho a y a la regla r por el programa, respectivamente.

Ahora que sabemos como sintetizar la etiqueta para un argumento, describamos el funcionamiento del sistema. La función de síntesis de etiqueta de argumento ALS permite sintetizar la etiqueta (meta-información) asociada a un argumento a partir de las etiquetas de las cláusulas que lo componen. Finalmente, el criterio de comparación por mejor BL define la noción de derrota entre argumentos en que se basa el proceso dialéctico.

A continuación se ilustran los conceptos recién introducidos utilizando el ejemplo de aplicación presentado en la sección 2.

5. EL EJEMPLO USANDO ℓ -DeLP PARA INCORPORAR EL TRATAMIENTO DE INCERTIDUMBRE POSIBILÍSTICA

A modo de ilustración, utilizaremos ℓ -DeLP para incorporar el tratamiento de incertidumbre posibilística al ejemplo de la sección 3. El resultado es equivalente a P-DeLP, una extensión a DeLP introducida en [2].

Cada cláusula del programa tendrá asociado, mediante una etiqueta, un valor entre 0 y 1. Este valor constituye una medida de necesidad o certeza para la cláusula. En particular, la etiqueta asociada a una regla establece el grado de certeza con el que el consecuente se sigue del antecedente. Como es de esperarse, las reglas estrictas y hechos siempre tendrán factor de certeza asociado 1.

A continuación se muestra el programa ℓ -DeLP \mathcal{P}_ℓ correspondiente al programa DeLP \mathcal{P} .

$$\mathcal{P}_\ell = \left\{ \begin{array}{l} 1) \text{ } abe \leftarrow bno_motor, bno_carr : [1] \\ 2) \text{ } bno_motor \prec suena_bien_motor : [0.3] \\ 3) \text{ } bno_motor \prec mecanico_da_ok : [0.9] \\ 4) \text{ } \sim auto_buen_estado \prec fue_taxi : [0.5] \\ 5) \text{ } fue_taxi \prec color_blanco, tiene_gas : [0.2] \\ 6) \text{ } color_blanco : [1] \\ 7) \text{ } tiene_gas : [1] \\ 8) \text{ } suena_bien_motor : [1] \\ 9) \text{ } bno_carr : [1] \end{array} \right\}$$

La cláusula etiquetada 2) establece que si el motor suena bien, entonces se encuentra en buen estado con un grado de certeza de 0.3.

La cláusula etiquetada 3) establece la existencia de una conexión más fuerte entre antecedente

y consecuente: si el mecánico da su aprobación, entonces el motor se encontrará en buen estado con un grado de certeza de 0.9.

Finalmente debemos definir la función ALS (Argument Label Synthesis) y la relación BL (Better Label) de modo que nuestro sistema se comporte como P-DeLP. Esto es, ALS deberá retornar como resultado el menor entre los valores correspondientes a los literales del cuerpo de la regla y el asociado a la regla por el programa recibidos como parámetro. BL captura la relación “posee mayor grado de certeza”.

Más formalmente:

$$ALS([\ell_{b_1}, \ell_{b_2}, \dots, \ell_{b_n}], \ell_c) = MIN(\ell_{b_1}, \ell_{b_2}, \dots, \ell_{b_n}, \ell_c)$$

$$BL(\ell_A, \ell_B) = \ell_A > \ell_B$$

De acuerdo a la semántica de P-DeLP, la etiqueta asociada a un argumento mediante la función ALS corresponderá al factor de certeza para el argumento. La definición de BL como “mejor si mayor grado de certeza” permite resolver la comparación entre argumentos en favor del más certero.

Consideremos nuevamente las tres situaciones analizadas en la sección 3, pero esta vez con ℓ -DeLP. Observar que se contará con los mismos argumentos que en las tres situaciones para DeLP, sólo que esta vez tendrán un factor de certeza asociado. Llamaremos \mathcal{P}_ℓ , \mathcal{P}_{ℓ_1} y \mathcal{P}_{ℓ_2} a las versiones para ℓ -DeLP de los programas DeLP \mathcal{P} , \mathcal{P}_1 y \mathcal{P}_2 , respectivamente.

Para el primer caso se cuenta con $\langle A_1, abe \rangle$ que posee factor de certeza 0.3 y con $\langle A_2, \sim abe \rangle$ que posee factor de certeza 0.2. Estos valores de certeza hacen que $\langle A_2, \sim abe \rangle$ no constituya un derrotador para $\langle A_1, abe \rangle$, y por ende, como se muestra en la figura 4, abe quedará garantizado. El agente concluirá que el auto se encuentra en buen estado con un grado de certeza de 0.3.

Para el segundo caso, correspondiente al programa \mathcal{P}_{ℓ_1} , se cuenta con el argumento $\langle A_3, \sim abe \rangle$ con factor de certeza 0.5. $\langle A_3, \sim abe \rangle$ es un argumento más fuerte que $\langle A_1, abe \rangle$, lo que permite al agente arribar a la conclusión de que el auto no está en buen estado, como se muestra en la figura 5.

Finalmente, para el programa \mathcal{P}_{ℓ_2} , se cuenta con el argumento $\langle A_4, abe \rangle$ con factor de certeza 0.6. Este es un argumento aún más fuerte que $\langle A_3, \sim abe \rangle$, por lo que se vuelve a reinstalar la creencia acerca del buen estado del auto, como se muestra en la figura 6.

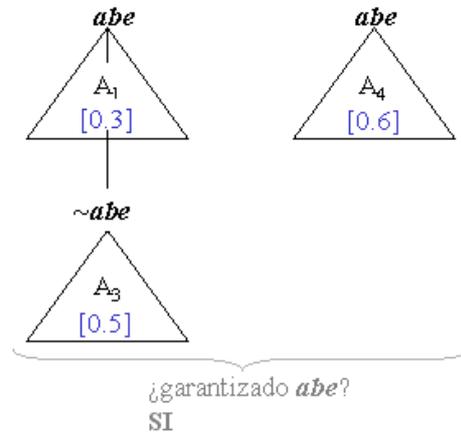
¿El auto se encuentra en buen estado?



Respuesta = SI

Figura 4: Razonamiento acerca del estado del auto a partir de \mathcal{P}_ℓ .

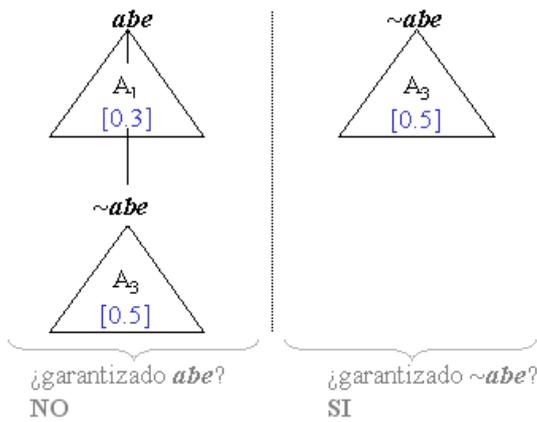
¿El auto se encuentra en buen estado?



Respuesta = SI

Figura 6: Razonamiento acerca del estado del auto a partir de $\mathcal{P}_{\ell 2}$.

¿El auto se encuentra en buen estado?



Respuesta = NO

Figura 5: Razonamiento acerca del estado del auto a partir de $\mathcal{P}_{\ell 1}$.

6. CONCLUSIÓN Y TRABAJO FUTURO

En este paper se presentó ℓ -DeLP, una extensión a la Programación en Lógica Rebatible (DeLP) para introducir el manejo de meta-información mediante etiquetas. Se mostró como esta extensión agrega expresividad a DeLP, permitiendo: 1) asociar cierta meta-información a las cláusulas de un programa, 2) especificar como sintetizar meta-información para los argumentos a partir de la meta-información correspondiente a las cláusulas que lo conforman y 3) definir un criterio de comparación por mejor entre argumentos basado en la meta-información asociada a los mismos. Esto hace posible la definición de un criterio de comparación de argumentos que capture una noción de mejor argumento apropiada para el do-

minio de aplicación específico.

Como trabajo futuro se terminará de definir y formalizar la incorporación de etiquetas a la Programación en Lógica Rebatible. Esto podría realizarse extendiendo la formalización mediante Sistemas Lógicos Etiquetados de DeLP presentada en [1].

Referencias

- [1] Carlos I. Chesñevar and Guillermo R. Simari. A framework for combining defeasible argumentation with labelled deduction. In Javier Legris Claudio Delrieux, editor, *Computer Modeling of Scientific Reasoning*, ISBN 987-89281-89-6, pages 43–56. EDIUNS, 2003.
- [2] C. Chesñevar, G. Simari, T. Alsinet, and L. Godo. A Logic Programming Framework for Possibilistic Argumentation with Vague Knowledge. In *Proc. of the Intl. Conference in Uncertainty in Artificial Intelligence (UAI 2004). Banff, Canada*, pages 76–84, July 2004.
- [3] A. García and G. Simari. Defeasible Logic Programming an Argumentative Approach. *Theory and Prac. of Logic Program.*, 4(1):95–138, 2004.