Un sistema multilenguaje para argumentación rebatible

Carlos I. Chesñevar[1] Guillermo R. Simari

Instituto de Ciencias e Ingeniería de Computación (ICIC)
Grupo de Investigación en Inteligencia Artificial (GIIA)
Departamento de Ciencias de la Computación
Universidad Nacional del Sur
Av.Alem 1253 – (8000) Bahía Blanca – REPÚBLICA ARGENTINA
FAX: (54) (91) 563401 – TEL.: (54) (91) 20776 (ext.208) – Email: {grs,ccchesne}@criba.edu.ar

PALABRAS CLAVE: inteligencia artificial, contextos, razonamiento rebatible, sistemas argumentativos

Abstract

Los sistemas multilenguajes (sistemas ML) constituyen una noción alternativa de sistema formal que permite el uso de lenguajes diferentes, cada uno de ellos asociado con su propia teoría. Por otra parte, es posible definir reglas de inferencia cuyas premisas y consecuencia no pertenezcan al mismo lenguaje, permitiéndose así propagar resultados *a través* de teorías. La argumentación rebatible es una variante del razonamiento rebatible para modelar el razonamiento de sentido común. Recientemente la teoría de la argumentación rebatible se ha enriquecido con distintas consideraciones dialécticas, que han redundado en ontologías alternativas para modelar el proceso de razonamiento mediante argumentos. Este trabajo presenta un sistema multilenguaje para argumentación rebatible, modelándose los roles de distintos elementos intervinientes en un sistema argumentativo utilizando distintos lenguajes.

[1]Becario de Perfeccionamiento del Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), República Argentina.

1 Introducción

Los *sistemas multilenguajes* [Giu91, Giu93] constituyen un poderoso formalismo para la representación de conocimiento en términos contextuales [McC87, Guh91]. En un sistema ML se estructura el conocimiento como un conjunto de distintas teorías, cada una de las cuales brinda una *descripción parcial* del mundo. Este acercamiento se suplementa con el uso de múltiples lenguajes, cada uno de los cuales tiene asociada su propia teoría. Un sistema ML permite formalizar así la interacción de deducciones *entre* distintos lenguajes, utilizando un tipo de regla de inferencia denominada *regla puente*. En estas reglas, las premisas y conclusiones pueden pertenecer a distintos lenguajes. Las reglas puente permiten derivar un teorema en una teoría a partir de un teorema en otra teoría, permitiendo un nexo de comunicación entre ellas.[2]

Los sistemas argumentativos [SL92, Vre93, SCG94] son una formalización del razonamiento rebatible de amplia aceptación en la comunidad de Inteligencia Artificial. El formalismo *MTDR* [SL92, SCG94, Che95] ha demostrado ser particularmente apropiado para modelar el proceso de argumentación rebatible. En *MTDR*, un *argumento A* para una hipótesis *h* constituye una prueba tentativa para que un agente inteligente pueda *explicar h*. La inferencia en un sistema argumentativo como *MTDR* es el resultado de un complejo proceso, caracterizado por definiciones dadas a un nivel meta-teórico. Recientemente [Che94, Che95] se han encontrado que un factor importante lo constituyen ciertas pautas y consideraciones dialécticas que permiten 'guiar' el proceso de inferencia, mejorando el comportamiento del formalismo para modelar situaciones reales. En particular, la noción de *contextos* ha demostrado ser extremadamente útil para modelar el proceso de argumentación de una manera modular, facilitando su definición y estudio.

En este trabajo se presenta la definición de un sistema multilenguaje para argumentación rebatible, al cual nos referiremos como MLAR, (sistema MultiLenguaje para Argumentación Rebatible), que intenta brindar un formalismo genérico dentro del cual pueden caracterizarse de manera flexible los distintos elementos intervinientes en el proceso de argumentación rebatible. El sistema formal presentado ha sido definido a partir de los conceptos presentados en [Giu91, Giu93], y modela el razonamiento argumentativo en términos del sistema *MTDR* [SL92, SCG94]. El trabajo está estructurado de la siguiente manera: primeramente, en la sección 2 se presentan los principales conceptos subyacentes a la argumentación rebatible. Luego, en la sección 3 se presentan generalidades acerca de los sistemas multilenguajes. Posteriormente, en la sección 4 se presenta el sistema formal MLAR. Se muestra el comportamiento del sistema a partir de un ejemplo. Finalmente, en la sección 5 se presentan las principales conclusiones obtenidas.

2 Argumentación rebatible

El formalismo *MTDR* [SL92, SCG94] constituye uno de los acercamientos más difundidos para modelar el razonamiento de un agente inteligente utilizando argumentación rebatible. En esta sección se presentan elementos generales que caracterizan a ese formalismo, los que pueden complementarse con las definiciones presentadas en el apéndice A.[3]

[2]Un análisis más detallado de las propiedades y características de los sistemas ML puede consultarse en [Giu91].

[3]Mayores detalles acerca del formalismo *MTDR* pueden consultarse en [SL92, SCG94].

En *MTDR*, un *argumento* $\langle A, h\rangle$ (ver def. A.2) constituye una prueba tentativa que un agente inteligente está dispuesto a aceptar como explicación para una hipótesis h. La *aceptabilidad* final de ese argumento $\langle A, h\rangle$ está caracterizada en términos de un proceso dialéctico, de caracter recursivo. Para decidir si $\langle A, h\rangle$ es aceptable, se lo confronta con todos aquellos *contraargumentos* (def. A.3) $\langle A_1, h_1\rangle$, $\langle A_2, h_2\rangle$, ..., $\langle A_k, h_k\rangle$. Cada uno de éstos constituye una prueba rebatible para *rechazar* $\langle A, h\rangle$. Si algún $\langle A_i, h_i\rangle$ es mejor o igual que $\langle A, h\rangle$ (de acuerdo a un criterio de preferencia entre argumentos), $\langle A_i, h_i\rangle$ será un candidato para *derrotar* a $\langle A, h\rangle$ (def. A.4). Pero puesto que $\langle A_i, h_i\rangle$ es también un argumento, el análsis anterior también se le aplica. Un argumento se considerará finalmente aceptable cuando *no tenga argumentos que lo derrotan*, o bien cuando *todos los argumentos que lo derrotan son a su vez derrotados por argumentos aceptables*. La aceptabilidad del argumento original $\langle A, h\rangle$ resultará entonces de un proceso recursivo, en el cual se tienen en consideración argumentos, derrotadores, derrotadores de derrotadores, y así sucesivamente. La estructura arbórea resultante de este proceso se denomina *árbol dialéctico* (def. A.5). Si $\langle A, h\rangle$ se considerará finalmente aceptable, se dice que $\langle A, h\rangle$ es una *justificación* (def. A.7).

El proceso anterior puede caracterizarse en términos de un *debate* de carácter introspectivo, donde intervienen un *proponente* (que presenta el argumento $\langle A, h\rangle$ inicial), un *oponente* (que intenta derrotar al oponente a través de contraargumentos), y un *árbitro* (que regula el desarrollo del debate). Se ha comprobado [SCG94, Che95] que esta caracterización es equivalente a la anterior, siendo más sencilla su conceptualización por hacer uso de la *dialéctica* [Res77]. El debate consiste de jugadas alternadas de proponente y oponente, continuadas hasta que no existan más jugadas posibles. Según la definición de *argumento aceptable*, el argumento inicial será una justificación si no existen más jugadas posibles, y todos los derrotadores del argumento inicial no son aceptables, o bien el argumento inicial no tiene derrotadores asociados.

EJEMPLO **2.1** Sea $\mathcal{K}= \{ e_1, e_2, c \rightarrow c_2 \}$, y sea $\Delta= \{ c_1 \wedge c_2 \mathrel{-\!\!\prec} h,\ e_2 \mathrel{-\!\!\prec} c,\ e_1 \mathrel{-\!\!\prec} c_1,\ e_2 \wedge c_4 \mathrel{-\!\!\prec} \neg c_1,\ e_1 \mathrel{-\!\!\prec} c_4,\ e_1 \wedge e_2 \mathrel{-\!\!\prec} \neg c_4, \}$. Entonces $A_1= \{ e_1 \mathrel{-\!\!\prec} c_1, e_2 \mathrel{-\!\!\prec} c, c_1 \wedge c_2 \mathrel{-\!\!\prec} h \}$ es un argumento para h. El argumento $\langle A_1, h\rangle$ tiene como derrotador asociado el argumento $\langle A_2, \neg c_1\rangle$, con $A_2 = \{ e_1 \mathrel{-\!\!\prec} c_4,\ c_4 \wedge e_2 \mathrel{-\!\!\prec} \neg c_1 \}$. Luego, $\langle A_1, h\rangle$ no es aceptable (provisoriamente). Pero este segundo argumento no es aceptable, ya que tiene a su vez otro derrotador asociado, $\langle A_3, \neg c_1\rangle$, con $A_3 = \{ e_1 \wedge e_2 \mathrel{-\!\!\prec} \neg c_4\}$. No existen más argumentos por considerar. Luego $\langle A_3, \neg c_4\rangle$ es una justificación; $\langle A_2, \neg c_1\rangle$ no es aceptable (y por ende no es una justificación). El argumento $\langle A_1, h\rangle$ es aceptable (pues no tiene derrotadores aceptables asociados), y por ende es una justificación.

3 Sistemas formales multilenguajes (SFM)

Los *sistemas formales multilenguajes* [Giu91, Giu93] constituyen una alternativa para *estructurar* el conocimiento como un conjunto de distintas teorías, cada una de las cuales constituye una *descripción parcial* del mundo. Desde la perspectiva de definición e implementación de sistemas basados en conocimiento (SBCs), este acercamiento posee ventajas análogas a las de la programación modular, facilitando el desarrollo y modificación de la base de conocimiento y de la máquina de inferencia del sistema. En un sistema multilenguaje (o sistema ML), cada lenguaje está asociado con su propia teoría. Además, es factible formalizar la interacción entre distintas teorías a través de deducciones dentro de los distintos lenguajes. Esto se formaliza a través de las denominadas *reglas 'puente'*,[4] en las cuales las premisas y la conclusión pertenecen a diferentes lenguajes. Las reglas puente permiten derivar un teorema en una teoría a partir de un teorema de otra teoría.

DEFINICIÓN **3.1** *Sistema Formal Multilenguaje SFM.* Un *sistema formal multilenguaje* (o SFM) es una 3-upla $\langle \mathcal{L}, \mathcal{A}, \mathcal{R} \rangle$, cuyas componentes están definidas como sigue:
Lenguajes: El conjunto $\mathcal{L} = \{L_i\}_{i \in I}$ es una familia de *lenguajes*, siendo I un conjunto de índices. Si f es una fbf en L_i, lo notaremos $L_i{:}f$.[5] Se denotará como S_i un conjunto de fbfs de L_i.
Axiomas: Para cada $L_i \in \mathcal{L}$ se distingue un un subconjunto de fbfs $A_i \subseteq L_i$, denominado los *axiomas* de L_i. El conjunto $\mathcal{A} = \{A_i : A_i \subseteq L_i, i \in I\}$ es el conjunto de *axiomas* del sistema.
Reglas de Inferencia: $\mathcal{R} = \{R_1, R_2, \ldots, R_m\}$ es el conjunto de *reglas de inferencia*, donde cada regla tiene la forma

$$\frac{S_{i_1}, S_{i_2}, \ldots, S_{i_m}}{L_j{:}f}$$

donde $i_k \in I, k = 1 \ldots m$. Así, las *premisas* de una regla de inferencia R_i están dadas por un conjunto de fbfs (eventualmente pertencientes a distintos lenguajes). La *conclusión* es una fbf f perteneciente a un lenguaje $L_j \in \mathcal{L}$. □

DEFINICIÓN **3.2** *Derivabilidad en un SFM.* Sea $F = \langle \mathcal{L}, \mathcal{A}, \mathcal{R} \rangle$ un sistema SFM. Sea S_i un conjunto de fbfs de $L_i \in \mathcal{L}$. Sea $\Gamma = \bigcup\{S_j\}$, con $j \in I$, donde I es el conjunto de indices asociado a F. Diremos que una fbf $L_i{:}f$ es *derivable en F a partir de* Γ, denotado $\Gamma \vdash_F L_i{:}f$, si existe una secuencia $p_1, p_2, \ldots, p_k$, tal que para cada p_i se cumple que (a) p_i es una instancia de un axioma de F; o (b) $p_i \in \Gamma$; o bien (c) p_i es consecuencia de p_j's previos por aplicación de alguna reglas de inferencia en $\mathcal{R}$. Si $\Gamma = \emptyset$ y $\Gamma \vdash_F L_i{:}f$, entonces diremos que $L_i{:}f$ es *probable* en F. □

4 El sistema formal MLAR

Seguidamente se definirá un sistema multilenguaje para argumentación rebatible (MLAR), que permitirá realizar inferencias siguiendo los lineamientos del formalismo *MTDR* [SL92, SCG94]. Nuestro acercamiento consistirá en caracterizar la base de conocimiento de un agente inteligente como *axiomas* del sistema formal multilenguaje. Los axiomas corresponderán a un subconjunto distinguido de dos lenguajes ($L_{\mathcal{K}}$ y L_Δ) utilizados para representación de conocimiento. Las *reglas de inferencia* permitirán simular la realización de un debate introspectivo entre un *proponente* y un *oponente*, regulados por un *árbitro*, siguiendo los lineamientos presentados en la sección 2. La caracterización de la inferencia se realiza utilizando distintos lenguajes ($L_{\mid\!\sim}$, L_{Arbit}, L_{Debate}, L_{Pro}, L_{Opp}). Los subíndices

[4] En inglés *"bridge rules"*.

[5] Nuestra formalización difiere en algunos aspectos de la presentada originalmente en [Giu91]. Entre otras cosas, [Giu91] utiliza la notación $\langle A, i \rangle$ para denotar que A es una fbf de un lenguaje L_i. La notación que se adopta en este trabajo ha sido diferente a fin de evitar ambigüedad con la notación que se utiliza usualmente para denotar argumentos.

utilizados en cada lenguaje reflejan el aspecto del proceso de inferencia que es capturado por ese lenguaje (ej.: el lenguaje L_{Arbit} captura las creencias del árbitro en el debate).

Cabe acotar que para ciertos predicados (ej: `inconsistent`, `nonminimal`, `status`, `debate`) se hace referencia a su *negación*, la cual debe interpretarse en términos de CWA (*closed-world assumption*).[6] Para esos predicados puede demostrarse la existencia de procedimientos efectivos que permiten computarlos, lo que garantiza la aplicabilidad de CWA. En la sección 5 se analizará este punto en mayor detalle.

DEFINICIÓN **4.1** *(Sistema MLAR)* Un sistema formal multilenguaje para argumentación rebatible estará dado por la 3-upla $\langle\ \{L_{\mathcal{K}}, L_{\Delta}, L_{\mid\!\sim}, L_{Debate}, L_{Arbit}, L_{Pro}, L_{Opp}\}, \{\ \mathcal{K}, \Delta\}, \mathcal{R}\rangle$ donde $\mathcal{K} \subseteq L_{\mathcal{K}}$, y $\Delta \subseteq L_{\Delta}$, tal que todo $e \in \Delta$ carece de variables instanciadas. El conjunto $\mathcal{R}$ se da por enumeración en el apartado 4.2. □

4.1 Lenguajes $L_{\mathcal{K}}$, L_{Δ}, $L_{\mid\!\sim}$, L_{Debate}, L_{Arbit}, L_{Pro} y L_{Opp}

A continuación se presenta la definición de los distintos lenguajes utilizados en MLAR. Los lenguajes $L_{\mathcal{K}}$ y L_{Δ} son lenguajes para representar conocimiento no-rebatible y rebatible, respectivamente.[7]

a) Lenguaje $L_{\mathcal{K}}$

<term-bas> ::= *<constant>*
<atom> ::= *<pred>(<term>, <term>, ... <term>)*
<lit> ::= *<atom>*| ¬*<atom>*
<conj-lits> ::= *<lit>*| *<lit>*∧*<conj-lits>*
<hecho> ::= *<lit>*
<regla> ::= *<conj-lits>*→*<lit>*
<fbf> ::= *<hecho>*| *<regla>*| ⊥

<term> ::= *<term-bas>*| *<var>*
<atom-bas> ::= *<pred>(<term-bas>, <term-bas>,... <term-bas>)*
<lit-bas> ::= *<atom-bas>*| ¬*<atom-bas>*
<conj-lits-bas> ::= *<lit-bas>*| *<lit-bas>*∧*<conj-lits-bas>*

b) Lenguaje L_{Δ}

<drule> ::= *<conj-lits>* $\succ\!\!-$ *<literal>* (ver def. de *<lit>* y *<conj-lits>* en $L_{\mathcal{K}}$)

El lenguaje $L_{\mid\!\sim}$ será utilizado para representar inferencias a partir de $L_{\mathcal{K}}$ y L_{Δ}. La fbf `dproof`(A,h) de $L_{\mid\!\sim}$ captura la noción de que A es el conjunto de reglas rebatibles instanciadas (denotado como *drules-bas*) usadas en una prueba rebatible para un literal instanciado h (def. A.1). El predicado `subproof`$(Proof1, Proof2)$ indica que el conjunto de reglas rebatibles instanciadas usado en $Proof1$ es un subconjunto del conjunto de reglas rebatibles instanciadas usado en $Proof2$; el predicado `inconsistent`$(Proof)$ refleja que $Proof$ es una prueba rebatible inconsistente. El predicado `nonminimal`$(Proof)$ indica que $Proof$ es una prueba rebatible no minimal.

El lenguaje L_{Arbit} representará el conocimiento del árbitro acerca del debate, permitiendo reflejar los argumentos obtenibles a partir de $L_{\mid\!\sim}$, y las relaciones entre ellos. Así, el predicado `arg`(Arg) denota que Arg es un argumento; `subarg`$(Arg1, Arg2)$ denota que $Arg1$ es un subargumento de $Arg2$ (ver def. A.2); `counterargues`$(Arg1, Arg2)$ denota

[6] La CWA establece que la derivabilidad de $\neg p$ está dada por la imposibilidad de derivar p (*i.e.*, si $S \not\vdash p$ entonces $S \vdash \neg p$).

[7] Los nombres *term-bas*, *atom-bas* y *lit-bas* denotan términos, átomos y literales básicos, respectivamente. Por razones de espacio, no se definen explícitamente variables, constantes y letras predicativas, las cuales son análogas a las usadas en un lenguaje de primer orden. Asimismo, se asumirán como válidas la asociatividad y conmutatividad para toda conjunción de la forma $a_1 \wedge a_2 \ldots \wedge a_n$, permitiéndose la introducción de paréntesis cuando se considere necesario. Estas características podrían eventualmente especificarse a través de reglas de inferencia *dentro* del sistema formal.

que el argumento $Arg1$ contraargumenta a $Arg2$ (ver def. A.3); `disagrees`($Arg1, Arg2$) denota que el argumento $Arg1$ está en desacuerdo a $Arg2$;[8] `defeats`($Arg1, Arg2$) denota que el argumento $Arg1$ derrota a $Arg2$ (ver def. A.4); `preferred`($Arg1, Arg2$) denota que el argumento $Arg1$ es preferido por sobre el argumento $Arg2$ (ver def. A.4); `justified`(Arg) denota que el argumento Arg es una justificación (ver def. A.7).

c) Lenguaje $L_{\succ\!\!\sim}$

<fbf>	::=	*<dproof>* | *<minimal>* | *<inconsistent>* |*<subproof>*
<dproof>	::=	`dproof(` *<drules-bas>*,*<lit-bas>* `)`
<drules-bas>	::=	{ *<drule-bas>* } | { *<drule-bas>* } ∪ *<drules-bas>*
<drule-bas>	::=	*<conj-lits-bas>* $\succ\!\!-$ *<lit-bas>* (ver def. en $L_{\mathcal{K}}$)
<subproof>	::=	`subproof(` *<dproof>*,*<dproof>* `)`
<inconsistent>	::=	`inconsistent(`*<dproof>*`)` | ¬`inconsistent(`*<dproof>*`)`
<minimal>	::=	`nonminimal(`*<dproof>*`)` | ¬`nonminimal(`*<dproof>*`)`

NOTA: puede observarse que los conjuntos $L_{\mathcal{K}}$, L_{Δ} y $L_{\succ\!\!\sim}$ no son disjuntos. Esto no presenta inconvenientes dentro del formalismo propuesto, en virtud de que la *semántica* de una fbf f depende del lenguaje $\mathcal{L}$ al cual pertenezca f.

d) Lenguaje L_{Arbit}

<fbf>	::=	*<argum>* | *<counter>* | *<disag>* | *<defeats>* | *<pref>* | *<justif>*
<argum>	::=	`arg(` *<drules-bas>*, *<lit-bas>* `)` (ver def. en $L_{\succ\!\!\sim}$ y $L_{\mathcal{K}}$).
<counter>	::=	`counterargues(` *<argum>*, *<argum>* `)`
<disag>	::=	`disagrees(` *<argum>*, *<argum>* `)`
<defeats>	::=	`defeats(` *<argum>*, *<argum>* `)`
<pref>	::=	`preferred(` *<argum>*, *<argum>* `)`
<justif>	::=	`justified(` *<argum>* `)`

Los lenguajes L_{Pro} y L_{Opp} permiten capturar los argumentos presentados por proponente y oponente a lo largo del debate, respectivamente. Dado que por razones obvias $L_{Pro} = L_{Opp}$, se los define genéricamente a través de un lenguaje L_{Party}. Se denotará a cualquiera de las partes del debate como $Party$, y se denotará con $\overline{Party}$ al adversario de $Party$. El lenguaje L_{Debate} permitirá describir las distintas instancias *(stages)* a lo largo de las cuales se realiza el debate. Las instancias están identificadas con números naturales consecutivos. El predicado `debate(`*<stage>*, *<argum>*,*<party>*,*<state>*`)` se utiliza para denotar que en la instancia *<stage>*, el argumento *<argum>* presentado por *<party>* se halla derrotado (*dead*) o no derrotado (*alive*). El predicado `status(`*<stage>*,*<situation>*`)` se utiliza para identificar distintas situaciones que se describen a continuación:

a) `curr_defeat`(P, $Arg1$, $Arg2$): el argumento $Arg1$ es el derrotador *en curso* en la instancia *<stage>*, y ha sido presentado por la parte P, derrotando al argumento $Arg2$;

b) `defeater`(P, $Arg1$, $Arg2$): el argumento $Arg1$ es un derrotador activo en la instancia *<stage>*, y ha sido presentado por la parte P, derrotando al argumento $Arg2$;

c) `linked_defeater`(P, $Arg1$, $Arg2$): el argumento $Arg1$ es un derrotador enlazado al derrotador $Arg2$, y ambos han sido presentados por la parte P.[9]

e) Lenguaje L_{Party}

<fbf> ::= *<argum>* (ver def. en L_{Arbit})

f) Lenguaje L_{Debate}

[8] $\langle A_1, h_1\rangle$ está en desacuerdo con $\langle A_2, h_2\rangle$ sssi $\mathcal{K} \cup \{h_1, h_2\} \vdash \bot$

[9] Si $\langle A_1, h_1\rangle$ derrota a $\langle A_2, h_2\rangle$, y $\langle A_2, h_2\rangle$ derrota a $\langle A_3, h_3\rangle$, entonces $\langle A_1, h_1\rangle$ y $\langle A_3, h_3\rangle$ se dicen *derrotadores enlazados.*

<fbf>	::=	debate(*<stage>*, *<argum>*, *<party>*, *<state>*) |
		¬debate(*<stage>*, *<argum>*, *<party>*, *<state>*)
		status(*<stage>*,*<situation>*) |¬status(*<stage>*,*<situation>*)
<state>	::=	*dead*| *alive*
<party>	::=	Pro | Opp
<stage>	::=	0 | succ(*<stage>*)
<argum>	::=	(ver def. en L_{Arbit})
<situation>	::=	curr_defeat(*<party>*,*<argum>*,*<argum>*) |
		defeater(*<party>*,*<argum>*,*<argum>*) |
		linked_defeater(*<party>*,*<argum>*,*<argum>*)

4.2 Reglas de inferencia en MLAR

Las reglas de inferencia en MLAR caracterizan los distintos elementos presentes en un debate entre un proponente y un oponente. Las reglas se presentan en un formato semejante a las usadas para *deducción natural.*[10] Se asume asimismo que se cuenta con las reglas de introducción y eliminación usuales en la deducción natural para una lógica de primer orden, las que pueden aplicarse sobre $\mathcal{K}$.[11]

Generación de argumentos: Las reglas R1 y R2 estipulan cómo obtener pruebas rebatibles en $L_{\mid\!\sim}$ a partir de $L_{\mathcal{K}}$ y L_{Δ}, Las reglas R3 y R5 definen las condiciones necesarias (ver def. A.2) para que una prueba rebatible en $L_{\mid\!\sim}$ constituya un argumento en L_{Arbit}.

Relaciones entre argumentos: A partir de la obtención de argumentos en L_{Arbit}, pueden establecerse distintas relaciones entre los mismos (desacuerdo, contraargumentación y derrota), caracterizadas por las reglas R8, R9 y R10 respectivamente (ver definiciones A.3 y A.4 en el apéndice).

Realización del debate: Las partes del debate (proponente y oponente) pueden introducir cualquier argumento que haya sido aceptado por el árbitro, tal como lo estipula la regla R11. Las reglas restantes caracterizan el desarrollo del debate, el cual está reflejado en aserciones en L_{Debate} que describen cómo se alternan argumentos y derrotadores, de acuerdo al formalismo *MTDR* [SL92]. La regla R12 establece que el comienzo del debate estará siempre a cargo del proponente. Las reglas R13 y R14 definen el ataque de una parte sobre la otra. Las reglas R15, R16, R17 y R18 establecen distintas relaciones entre derrotadores. A través de las reglas R17 y R18 se define el concepto de *derrotadores enlazados* (*linked defeaters*). Los derrotadores enlazados pueden ser *reinstaurados* en la instancia del debate en curso a través de la regla R19.[12]

[10]Como es usual, se hará uso de *cajas* (*boxes*), denotadas con corchetes, para controlar el ámbito de las suposiciones en las premisas. Una *caja* contiene un fragmento de una prueba que puede hacer uso de las suposiciones con las que se "abrió"esa caja. La "descarga" de suposiciones puede hacerse a través de reglas de inferencia que contienen cajas en sus premisas. Mayores detalles pueden verse en [AGE92], págs.45–50.

[11]El lector no familiarizado con deducción natural puede encontrar una enumeración de esas reglas en [AGE92], págs.46–47.

[12]Una descripción más detallada de la formalización de un debate en términos de reglas de inferencia puede encontrarse en [SC95].

1. Introducción de $\succ\!\!-$ (1): [13]

$$\frac{\begin{array}{c} L_{\mathcal{K}}{:}e_1(t_1) \wedge e_2(t_2) \wedge \ldots \wedge e_k(t_k) \\ L_{\Delta}{:}e_1(\overline{X_1}) \wedge e_2(\overline{X_2}) \wedge \ldots \wedge e_k(\overline{X_k}) \succ\!\!- h(\overline{X}) \end{array}}{L_{\mid\!\sim}{:}\mathtt{dproof}(\{e_1(\overline{X_1}\theta) \wedge e_2(\overline{X_2}\theta) \wedge \quad \wedge e_k(\overline{X_k}\theta) \succ\!\!- h(\overline{X}\theta)\}, h(\overline{X}\theta))}$$

2. Introducción de $\succ\!\!-$ (2):

$$\frac{\begin{array}{c} L_{\mid\!\sim}{:}\mathtt{dproof}(A_1, h_1) \\ L_{\mid\!\sim}{:}\mathtt{dproof}(A_2, h_2) \\ \vdots \\ L_{\mid\!\sim}{:}\mathtt{dproof}(A_k, h_k) \\ \left[\begin{array}{c} L_{\mathcal{K}}{:}h_1 \wedge h_2 \ldots \wedge h_k \\ \\ L_{\mid\!\sim}{:}\mathtt{dproof}(\{e_1(\overline{X_1}\theta), e_2(\overline{X_2}\theta), \ldots, e_k(\overline{X_k}\theta) \succ\!\!- h(\overline{X}\theta)\}, h(\overline{X}\theta)) \end{array}\right] \end{array}}{L_{\mid\!\sim}{:}\mathtt{dproof}(A_1 \cup A_2 \cup \ldots A_k \cup \{e_1(\overline{X_1}\theta), e_2(\overline{X_2}\theta), \ldots, e_k(\overline{X_k}\theta) \succ\!\!- h(\overline{X}\theta)\}, h(\overline{X}\theta))}$$

3. Inconsistencia

$$\frac{\begin{array}{c} L_{\mid\!\sim}{:}\mathtt{dproof}(A, h) \\ L_{\mid\!\sim}{:}\mathtt{subproof}(\mathtt{dproof}(B, j), \mathtt{dproof}(A, h)) \\ \left[\begin{array}{c} L_{\mathcal{K}}{:}j \\ \vdots \\ L_{\mathcal{K}}{:}\bot \end{array}\right] \end{array}}{L_{\mid\!\sim}{:}\mathtt{inconsistent}(\mathtt{dproof}(A, h))}$$

4. Definición de inclusión entre pruebas rebatibles:

$$\frac{L_{\mid\!\sim}{:}\mathtt{dproof}(Proof \cup \{g_1, \ldots, g_k \succ\!\!- h\}, h)}{L_{\mid\!\sim}{:}\mathtt{subproof}(\mathtt{dproof}(Proof, g_i), \mathtt{dproof}(Proof \cup \{g_1, \ldots, g_k \succ\!\!- h\}, h)\;)\quad 1 \le i \le k}$$

$$\frac{\begin{array}{c} L_{\mid\!\sim}{:}\mathtt{dproof}(A_1, h_1) \\ L_{\mid\!\sim}{:}\mathtt{subproof}(\mathtt{dproof}(A, h), \mathtt{dproof}(A_1, h_1)\;) \end{array}}{L_{\mid\!\sim}{:}\mathtt{dproof}(A, h)}$$

5. Definición de minimalidad:

$$\frac{\begin{array}{c} L_{\mid\!\sim}{:}\mathtt{dproof}(A, h) \\ L_{\mid\!\sim}{:}\mathtt{subproof}(\mathtt{dproof}(B, h), \mathtt{dproof}(A, h)) \end{array}}{L_{\mid\!\sim}{:}\mathtt{nonminimal}(\mathtt{dproof}(A, h))}$$

6. Definición de argumento:

$$\frac{L_{\mid\!\sim}{:}\neg\mathtt{nonminimal}(\mathtt{dproof}(A, h)) \quad L_{\mid\!\sim}{:}\neg\mathtt{inconsistent}(\mathtt{dproof}(A, h))}{L_{Arbit}{:}\mathtt{arg}(A, h)}$$

7. Definición de subargumento:

$$\frac{\begin{array}{c} L_{Arbit}{:}\mathtt{arg}(A_1, h_1) \\ L_{\mid\!\sim}{:}\mathtt{subproof}(\mathtt{dproof}(A, h), \mathtt{dproof}(A_1, h_1)\;) \end{array}}{L_{Arbit}{:}\mathtt{subarg}(\mathtt{arg}(A, h), \mathtt{arg}(A_1, h_1))}$$

8. Definición de desacuerdo:

[13]Los términos t_i corresponden a términos básicos. $\overline{X}$ denota genéricamente un vector de variables $(X_{i_n}, \ldots, X_{i_m})$, con $1 \le n \le m \le k$. $\overline{X}\theta$ denota un término básico obtenido por la sustitución $\theta = \{X_i/t_i\}, i = 1 \ldots k$.

$$\frac{L_{Arbit}\texttt{:arg(}A_1\texttt{,}h_1\texttt{)} \qquad L_{Arbit}\texttt{:arg(}A_2\texttt{,}h_2\texttt{)} \qquad \left[\begin{array}{cc} L_{\mathcal{K}}\texttt{:}h_1 & L_{\mathcal{K}}\texttt{:}h_2 \\ \\ \multicolumn{2}{c}{L_{\mathcal{K}}\texttt{:}\bot} \end{array}\right]}{L_{Arbit}\texttt{:disagree(arg(}A_1\texttt{,}h_1\texttt{),arg(}A_2\texttt{,}h_2\texttt{))}}$$

9. Definición de contraargumento:

$$\frac{L_{Arbit}\texttt{:arg(}A_1\texttt{,}h_1\texttt{)} \qquad L_{Arbit}\texttt{:arg(}A_2\texttt{,}h_2\texttt{)} \\ L_{Arbit}\texttt{:subarg(arg(}A\texttt{,}h\texttt{),arg(}A_2\texttt{,}h_2\texttt{))} \qquad L_{Arbit}\texttt{:disagree(arg(}A\texttt{,}h\texttt{),arg(}A_2\texttt{,}h_2\texttt{))}}{L_{Arbit}\texttt{:counterargues(arg(}A_1\texttt{,}h_1\texttt{),arg(}A_2\texttt{,}h_2\texttt{))}}$$

10. Definición de derrota:

$$\frac{L_{Arbit}\texttt{:counterargues(arg(}A_1\texttt{,}h_1\texttt{),arg(}A_2\texttt{,}h_2\texttt{))} \\ L_{Arbit}\texttt{:preferred(arg(}A_1\texttt{,}h_1\texttt{),arg(}A_2\texttt{,}h_2\texttt{))}}{L_{Arbit}\texttt{:defeats(arg(}A_1\texttt{,}h_1\texttt{),arg(}A_2\texttt{,}h_2\texttt{))}}$$

11. Incorporación de argumentos

$$\frac{L_{Arbit}\texttt{:arg(}A\texttt{,}h\texttt{)}}{L_{Pro}\texttt{:arg(}A\texttt{,}h\texttt{)}} \qquad \frac{L_{Arbit}\texttt{:arg(}A\texttt{,}h\texttt{)}}{L_{Opp}\texttt{:arg(}A\texttt{,}h\texttt{)}}$$

12. Comienzo del debate

$$\frac{L_{Pro}\texttt{:arg(}A\texttt{,}h\texttt{)}}{L_{Debate}\texttt{:debate(0,arg(}A\texttt{,}h\texttt{),Pro,}alive\texttt{)}}$$

13. Ataque de proponente a oponente (o viceversa)

$$\frac{L_{Pro}\texttt{:arg(}A_1\texttt{,}h_1\texttt{)} \\ L_{Debate}\texttt{:debate(i,arg(}A_2\texttt{,}h_2\texttt{),Opp,}alive\texttt{)} \\ L_{Arbit}\texttt{:defeats(arg(}A_1\texttt{,}h_1\texttt{),arg(}A_2\texttt{,}h_2\texttt{))}}{L_{Debate}\texttt{:status(succ(i), curr\_defeat(Pro,arg(}A_1\texttt{,}h_1\texttt{),arg(}A_2\texttt{,}h_2\texttt{)))}}$$

$$\frac{L_{Opp}\texttt{:arg(}A_1\texttt{,}h_1\texttt{)} \\ L_{Debate}\texttt{:debate(i,arg(}A_2\texttt{,}h_2\texttt{),Pro,}alive\texttt{)} \\ L_{Arbit}\texttt{:defeats(arg(}A_1\texttt{,}h_1\texttt{),arg(}A_2\texttt{,}h_2\texttt{))}}{L_{Debate}\texttt{:status(succ(i), curr\_defeat(Opp,arg(}A_1\texttt{,}h_1\texttt{),arg(}A_2\texttt{,}h_2\texttt{)))}}$$

14. Argumentos vivos y muertos

$$\frac{L_{Debate}\texttt{:status(i, curr\_defeat(}Party\texttt{,arg(}A_1\texttt{,}h_1\texttt{),arg(}A_2\texttt{,}h_2\texttt{)))}}{L_{Debate}\texttt{:debate(i,arg(}A_1\texttt{,}h_1\texttt{),}Party\texttt{,}alive\texttt{)} \\ L_{Debate}\texttt{:debate(i,arg(}A_2\texttt{,}h_2\texttt{),}\overline{Party}\texttt{,}dead\texttt{)}}$$

15. Derrotador en curso es derrotador:

$$\frac{L_{Debate}\texttt{:status(i, curr\_defeat(}Party\texttt{,arg(}A_1\texttt{,}h_1\texttt{),arg(}A_2\texttt{,}h_2\texttt{)))}}{L_{Debate}\texttt{:status(i, defeater(}Party\texttt{,arg(}A_1\texttt{,}h_1\texttt{),arg(}A_2\texttt{,}h_2\texttt{)))}}$$

16. Preservación de derrotador:

$$\frac{L_{Debate}\texttt{:status(i, defeater(}Party\texttt{,arg(}A_1\texttt{,}h_1\texttt{),arg(}A_2\texttt{,}h_2\texttt{)))} \qquad i < j}{L_{Debate}\texttt{:status(j, defeater(}Party\texttt{,arg(}A_1\texttt{,}h_1\texttt{),arg(}A_2\texttt{,}h_2\texttt{)))}}$$

17. Definición de enlace:

$$\frac{L_{Debate}\texttt{:status(i, defeater(}Party\texttt{,arg(}A_1\texttt{,}h_1\texttt{),arg(}A_2\texttt{,}h_2\texttt{)))} \\ L_{Debate}\texttt{:status(i, defeater(}\overline{Party}\texttt{,arg(}A_2\texttt{,}h_2\texttt{),arg(}A_3\texttt{,}h_3\texttt{)))}}{L_{Debate}\texttt{:status(i, linked\_defeater(}Party\texttt{,arg(}A_1\texttt{,}h_1\texttt{),arg(}A_3\texttt{,}h_3\texttt{)))}}$$

18. Transitividad de derrotadores enlazados:

$$\frac{\begin{array}{c} L_{Debate}\texttt{:status(i, linked\_defeater(}Party\texttt{,arg(}A_1\texttt{,}h_1\texttt{),arg(}A_2\texttt{,}h_2\texttt{)))} \\ L_{Debate}\texttt{:status(i, linked\_defeater(}Party\texttt{,arg(}A_2\texttt{,}h_2\texttt{),arg(}A_3\texttt{,}h_3\texttt{)))} \end{array}}{L_{Debate}\texttt{:status(i, linked\_defeater(}Party\texttt{,arg(}A_1\texttt{,}h_1\texttt{),arg(}A_3\texttt{,}h_3\texttt{)))}}$$

19. *Reinstatement* de argumentos (1):

$$\frac{\begin{array}{c} L_{Debate}\texttt{:status(i, curr\_defeat(}Party\texttt{,arg(}A_1\texttt{,}h_1\texttt{),arg(}B\texttt{,}j\texttt{)))} \\ L_{Debate}\texttt{:status(i, linked\_defeater(}Party\texttt{,arg(}A_1\texttt{,}h_1\texttt{),arg(}A_2\texttt{,}h_2\texttt{)))} \end{array}}{L_{Debate}\texttt{:debate(i,arg(}A_2\texttt{,}h_2\texttt{),}Party\texttt{,}alive\texttt{)}}$$

$$\frac{\begin{array}{c} L_{Debate}\texttt{:status(i, curr\_defeat(}Party\texttt{,arg(}A_1\texttt{,}h_1\texttt{),arg(}B\texttt{,}j\texttt{)))} \\ L_{Debate}\texttt{:status(i, linked\_defeater(}\overline{Party}\texttt{,arg(}B\texttt{,}j\texttt{),arg(}C\texttt{,}q\texttt{)))} \end{array}}{L_{Debate}\texttt{:debate(i,arg(}C\texttt{,}q\texttt{),}\overline{Party}\texttt{,}dead\texttt{)}}$$

20. *Reinstatement* de argumentos (2):

$$\frac{\begin{array}{c} L_{Debate}\texttt{:debate(i,arg(}C\texttt{,}q\texttt{),}Party\texttt{,}alive\texttt{)} \quad i < j \\ L_{Debate}\texttt{:status(j,curr\_defeat(}Party\texttt{,arg(}A\texttt{,}h\texttt{),arg(}B\texttt{,}j\texttt{)))} \\ L_{Debate}\texttt{:}\neg\texttt{status(j,linked\_defeater(Party,arg(}A\texttt{,}h\texttt{),arg(}C\texttt{,}q\texttt{)))} \\ L_{Debate}\texttt{:}\neg\texttt{status(j,linked\_defeater(Party,arg(}B\texttt{,}j\texttt{),arg(}C\texttt{,}q\texttt{)))} \end{array}}{L_{Debate}\texttt{:debate(j,arg(}C\texttt{,}q\texttt{),}Party\texttt{,}alive\texttt{)}}$$

21. Definición de justificación:

$$\frac{\begin{array}{c} L_{Debate}\texttt{:debate(i,arg(}A\texttt{,}h\texttt{),}Party\texttt{,}alive\texttt{)} \\ L_{Debate}\texttt{:}\neg\texttt{debate(succ(i),arg(}A\texttt{,}h\texttt{),}Party\texttt{,}dead\texttt{)} \end{array}}{L_{Arbit}\texttt{:justified(arg(}A\texttt{,}h\texttt{))}}$$

4.3 Inferencia en MLAR

A partir de los lenguajes y las reglas de inferencias especificadas anteriormente, un *debate* en términos argumentativos[SCG94] puede conceptualizarse como una *prueba* dentro de MLAR. De esta manera, el concepto de justificación queda caracterizado como sigue.

DEFINICIÓN **4.2** *(justificación)* Sea $h \in L_{\mathcal{K}} \cup L_{\Delta}$, donde $L_{\mathcal{K}}$ y L_{Δ} son los lenguajes de representación de conocimiento de un sistema multilenguaje de argumentación rebatible $S = \langle \{L_{\mathcal{K}}, L_{\Delta}, L_{\succ}, L_{Debate}, L_{Arbit}, L_{Pro}, L_{Opp}\}, \{\mathcal{K}, \Delta\}, \mathcal{R}\rangle$. Entonces h está *justificado* si L_{Arbit}:`justified(arg(`A, h`))` es probable, esto es, si $\vdash_S L_{Arbit}$: `justified(arg(`A,h`))`. □

4.4 Un ejemplo de argumentación rebatible (revisitado)

En la figura 1 de la página siguiente se muestra un ejemplo del proceso de inferencia en MLAR, a partir de la base de conocimiento $(\mathcal{K}, \Delta)$ dada en el ejemplo 2.1. (Obs: los argumentos obtenidos por el proceso de inferencia se abrevian a través de los mismos nombres utilizados en el ejemplo; no se incluyen todos los pasos del proceso de inferencia por razones de espacio).

1	$L_{\mathcal{K}}$:e_1	(axioma)
2	L_{Δ}:$e_1 \succ\!\!- c_1$	(axioma)
3	$L_{\mid\!\sim}$:dproof($\{e_1 \succ\!\!- c_1\}$,c_1)	De 1,2 y R1
4	$L_{\mathcal{K}}$:e_2	(axioma)
5	L_{Δ}:$e_2 \succ\!\!- c$	(axioma)
6	$L_{\mid\!\sim}$:dproof($\{e_2 \succ\!\!- c\}$,c)	De 4,5 y R1
7	$L_{\mid\!\sim}$:dproof($\{e_2 \succ\!\!- c\}$,c_2)	De 6 por R2
8	L_{Δ}:$c_1 \wedge c_2 \succ\!\!- h$	(axioma)
9	$L_{\mid\!\sim}$: dproof($\{ e_1 \succ\!\!- c_1,\ e_2 \succ\!\!- c,\ c_1 \wedge c_2 \succ\!\!- h \}$, h)	De 3,7, y 8 por R2.
10	$L_{\mid\!\sim}$:¬nonminimal(dproof($\{ e_1 \succ\!\!- c_1,\ e_2 \succ\!\!- c,$ $c_1 \wedge c_2 \succ\!\!- h \}$,$h$))	Por CWA y R5.
11	$L_{\mid\!\sim}$:¬inconsistent(dproof($\{ e_1 \succ\!\!- c_1,\ e_2 \succ\!\!- c,$ $c_1 \wedge c_2 \succ\!\!- h \}$,$h$))	Por CWA y R3.
12	L_{Arbit}:arg($\{e_1 \succ\!\!- c_1, e_2 \succ\!\!- c, c_1 \wedge c_2 \succ\!\!- h\}$,$h$)	(arg(A_1,h)). De 10, 11 y R6.
13	L_{Pro}:arg(A_1,h)	De 12 y R11.
14	L_{Opp}:arg($\{e_1 \succ\!\!- c_4, c_4 \wedge e_2 \succ\!\!- \neg c_1\}$,$\neg c_1$)	(arg(A_2,$\neg c_1$)) *via reglas análogas a 13*
15	L_{Debate}:debate(0,arg(A_1,h) ,Pro,$alive$)	De 13 y R12.
16	L_{Arbit}:defeats(arg(A_2,$\neg c_1$) ,arg(A_1,h))	De 13, 14 y R10.
$\vdots$		
k	L_{Debate}:debate(succ(succ(0)),arg(A_1,h),Pro,$alive$)	
$k+1$	L_{Arbit}:justified(arg(A_1,h))	De k, por CWA y R21

Luego el argumento inicial $\langle A_1, h\rangle$ es una *justificación* para h.

Figura 1: Una secuencia de prueba en MLAR

5 Conclusiones

En este trabajo se ha mostrado como reflejar el comportamiento de un sistema argumentativo en términos de un sistema formal multilenguaje. Las ventajas del acercamiento empleado son múltiples. Por una parte, puede analizarse la definición de *protocolos* de debate desde una perspectiva formal, especificándolos a través de un ordenamiento en aplicación de reglas de inferencia. Por otro lado, el sistema formal es fácilmente modificable para capturar distintas alternativas dentro de un sistema argumentativo (ej: permitir la construcción de *esquemas de argumentos* empleando reglas rebatibles no instanciadas). En particular, dentro de una teoría cuyo lenguaje asociado es L, una alternativa interesante surge de equiparar la *negación por falla* (NAF) para un predicado p con la *negación clásica* para p[14] de la manera siguiente:

$$\frac{L{:}not\ p}{L{:}\neg p}$$

La computación de CWA para p podría reemplazarse por NAF, donde la capacidad de inferir *not* p está vinculada a los *recursos* disponibles en la maquinaria deductiva de

[14]Este concepto sería análogo al empleado en los *programas lógicos extendidos* [GL90]; una aplicación de este concepto a la programación en lógica basada en argumentación rebatible puede verse en [SG95].

L. Estos recursos podrían repartirse entre proponente, oponente y árbitro. Todos estos elementos están siendo estudiados para la definición de un marco formal de caracter genérico que posibilite la definición y estudio de la argumentación rebatible. Actualmente este tema está siendo objeto de investigación.

Referencias

[AGE92] S. Abramsky, D. Gabbay, y T. Maibaum (Eds.). *Handbook of Logic in Computer Science (vol.1)*. Clarendon Press, Oxford, 1992.

[Che94] Carlos I. Chesñevar. Stratifying defeasible rules in argumentative reasoning: a consistency-based approach. En *Primer Workshop Argentino sobre Aspectos Teóricos en Inteligencia Artificial (Buenos Aires)*, Abril 1994.

[Che95] Carlos I. Chesñevar. On the use of contexts for representing knowledge in defeasible argumentation. En *2do. Workshop sobre Aspectos Teóricos en Inteligencia Artificial*, 1er. Congreso Argentino en Cs. de la Computación. B.Blanca, Oct.1995.

[Giu91] Fausto Giunchiglia. Multilanguage systems. En *Proceedings of the AAAI Spring Symposium on Logical Formalizations of Commonsense reasoning*, 1991.

[Giu93] Fausto Giunchiglia. Contextual Reasoning. *EPISTEMOLOGIA*, 16:345–364, 1993.

[GL90] Michael Gelfond y Vladimir Lifschitz. Logic programs with classical negation. En *Proceedings of the 7th International Conference on Logic Programming. Jerusalem.*, 1990.

[Guh91] R. V. Guha. *Contexts: A Formalization and some applications.* Tesis Ph.D., Stanford University, Departament of Computer Science, 1991.

[McC87] John McCarthy. Generality in artificial intelligence. *CACM, 30(12)*, 1030–1035, 1987.

[Res77] Nicholas Rescher. *Dialectics, a Controversy-Oriented Approach to the Theory of Knowledge.* State University of New York Press, Albany, USA, 1977.

[SC95] Guillermo R. Simari y Carlos I. Chesñevar. Arguments and contexts. En *XXI Conferencia Latinoamericana de Informática*, CLEI/PANEL 95, Canela (Brasil)., Julio 1995.

[SCG94] Guillermo R. Simari, Carlos I. Chesñevar, y Alejandro J. García. The role of dialectics in defeasible argumentation. En *Anales de la XIV Conferencia Internacional de la Sociedad Chilena para Ciencias de la Computación*, Universidad de Concepción, Concepción (Chile), Noviembre 1994.

[SG95] Guillermo R. Simari y Alejandro J. García. A knowledge representation language for defeasible argumentation. En *XXI Conferencia Latinoamericana de Informática*, CLEI/PANEL 95, Canela (Brasil)., Julio 1995.

[SL92] Guillermo R. Simari y Ronald P. Loui. A Mathematical Treatment of Defeasible Reasoning and its Implementation. *Artificial Intelligence*, 53:125–157, 1992.

[Vre93] Gerard A.W. Vreeswijk. *Studies in Defeasible Argumentation.* Tesis Ph.D., Vrije University, Amsterdam (Holanda), 1993.

A El formalismo *MTDR*

Representación de conocimiento: El conocimiento de un agente inteligente $\mathcal{A}$ se representa utilizando un lenguaje $\mathcal{L}$ de primer orden, más una relación binaria metalingüística " $\succ\!\!-$ ", definida sobre $\mathcal{L}$ entre un conjunto de literales no básicos (antecedente) y un literal no básico (consecuente). Los miembros de esta relación binaria metalingüística se denominan *reglas rebatibles*. La relación "$\alpha \succ\!\!- \beta$" expresa que "razones para creer en α proveen razones para creer en β." Se restringirá el lenguaje $\mathcal{L}$ a un subconjunto que involucra únicamente cláusulas Horn.

El conjunto $\mathcal{K}$ será un subconjunto finito de $\mathcal{L}$ que representa la parte no-rebatible del conocimiento de $\mathcal{A}$. Δ denota un conjunto finito de reglas rebatibles no básicas que representan información que $\mathcal{A}$ está dispuesto a aceptar como válida. Si $A \subseteq \Delta$, se denotará con $A^{\downarrow}$ al conjunto de todas las instancias básicas de miembros de A. El conjunto $\mathcal{K}$ puede particionarse en dos subconjuntos: $\mathcal{K}_G$ (conocimiento *general*) y $\mathcal{K}_P$ (conocimiento *particular* o *contingente*). Las sentencias en $\mathcal{K}_P$ serán literales básicos (*E.g.*: $vuela(\mathsf{tweety})$. Las sentencias en $\mathcal{K}_G$ serán implicaciones materiales de la forma $a_1, a_2, \ldots, a_k \rightarrow b$, *e.g.* $emu(X) \rightarrow pájaro(X)$. Las reglas rebatibles tienen la forma $a_1, a_2, \ldots, a_k \succ\!\!- b$, *e.g.* $pájaro(X) \succ\!\!- vuela(X)$.

Inferencia: Las definiciones A.1 a A.7 sintetizan la noción de inferencia en *MTDR*.

DEFINICIÓN **A.1** Sea Γ un subconjunto de $\mathcal{K} \cup \Delta^{\downarrow}$. Un literal básico h es una *consecuencia rebatible* de Γ, denotado $\Gamma \mid\!\sim h$, sssi existe una secuencia finita $B_1, \ldots, B_n$ tal que $B_n = h$ y para $1 \leq i < n$, o bien $B_i \in \Gamma$, o bien B_i es una consecuencia directa de los elementos precedentes en la secuencia por aplicación de cualquier regla de inferencia de la teoría de primer orden asociada con el lenguaje $\mathcal{L}$. Las instancias básicas de las reglas rebatibles se consideran implicaciones materiales para la aplicación de reglas de inferencia. Se escribirá $\mathcal{K} \cup A \mid\!\sim h$ para distinguir el conjunto A de reglas rebatibles utilizadas en la derivación a partir del conjunto $\mathcal{K}$. □

DEFINICIÓN **A.2** Dado un conjunto $\mathcal{K}$, un conjunto Δ de reglas rebatibles, y un literal básico $h \in \mathcal{L}$, se dice que un subconjunto A de $\Delta^{\downarrow}$ es una *estructura de argumento* (o simplemente *argumento*) para h (denotado $\langle A, h \rangle$) sssi: 1) $\mathcal{K} \cup A \mid\!\sim h$, 2) $\mathcal{K} \cup A \not\mid\!\sim \bot$ and 3) $\nexists A' \subset A$, $\mathcal{K} \cup A' \mid\!\sim h$. Un *subargumento* de $\langle A, h \rangle$ es un argumento $\langle S, j \rangle$ tal que $S \subseteq A$. □

DEFINICIÓN **A.3** Dados dos argumentos $\langle A_1, h_1 \rangle$ y $\langle A_2, h_2 \rangle$, se dirá que $\langle A_1, h_1 \rangle$ *contraargumenta* $\langle A_2, h_2 \rangle$, denotado $\langle A_1, h_1 \rangle \overset{h}{\otimes\!\!\rightarrow} \langle A_2, h_2 \rangle$, sssi existe un subargumento $\langle A, h \rangle$ de $\langle A_2, h_2 \rangle$ tal que $\mathcal{K} \cup \{h_1, h_2\} \vdash \bot$. El literal h será denominado *literal de counterargumentación.* □

DEFINICIÓN **A.4** Dadas dos estructuras de argumento $\langle A_1, h_1 \rangle$ y $\langle A_2, h_2 \rangle$, diremos que $\langle A_1, h_1 \rangle$ *derrota* $\langle A_2, h_2 \rangle$ en el literal h, denotado $\langle A_1, h_1 \rangle \gg_{\mathbf{def}} \langle A_2, h_2 \rangle$, sssi existe un subargumento $\langle A, h \rangle$ de $\langle A_2, h_2 \rangle$ tal que: $\langle A_1, h_1 \rangle$ contraargumenta $\langle A_2, h_2 \rangle$ en el literal h y
(1). $\langle A_1, h_1 \rangle$ es *estrictamente más específico*[15] que $\langle A, h \rangle$, o bien (2). $\langle A_1, h_1 \rangle$ no está relacionado por especificidad con $\langle A, h \rangle$.
Si $\langle A_1, h_1 \rangle \gg_{\mathbf{def}} \langle A_2, h_2 \rangle$, también diremos que $\langle A_1, h_1 \rangle$ es un *derrotador* para $\langle A_2, h_2 \rangle$. □

DEFINICIÓN **A.5** Un *árbol dialéctico* $\mathcal{T}_{\langle A, h \rangle}$ para un argumento $\langle A, h \rangle$ se define recursivamente como sigue: a) Un único nodo que contiene una estructura de argumento $\langle A, h \rangle$ sin derrotadores es en si mismo un árbol dialéctico para $\langle A, h \rangle$. Este nodo es también la raíz del árbol; b) Sea $\langle A, h \rangle$ un argumento con derrotadores $\langle A_1, h_1 \rangle$, $\langle A_2, h_2 \rangle$, $\ldots$, $\langle A_n, h_n \rangle$. El árbol dialéctico $\mathcal{T}_{\langle A, h \rangle}$, se construye colocando a $\langle A, h \rangle$ como el nodo raíz de $\mathcal{T}_{\langle A, h \rangle}$ y haciendo que este nodo sea el padre de las raíces de los árboles dialécticos para $\langle A_1, h_1 \rangle$, $\langle A_2, h_2 \rangle$, $\ldots$, $\langle A_n, h_n \rangle$. □

DEFINICIÓN **A.6** Sea $\mathcal{T}_{\langle A, h \rangle}$ un árbol dialéctico para una estructura de argumento $\langle A, h \rangle$. Los nodos de $\mathcal{T}_{\langle A, h \rangle}$ pueden etiquetarse recursivamente como *nodos no-derrotados* (nodos-U) y *nodos derrotados* (nodos-D) de la manera siguiente: a) Las hojas de $\mathcal{T}_{\langle A, h \rangle}$ son *nodos-U*; b) Sea $\langle B, q \rangle$ un nodo interno de $\mathcal{T}_{\langle A, h \rangle}$. Entonces $\langle B, q \rangle$ será un *nodo-U* sssi cada hijo de $\langle B, q \rangle$ es un *nodo-D*. $\langle B, q \rangle$ será un *nodo-D* sssi tiene al menos un nodo hijo que sea *nodo-U*. □

DEFINICIÓN **A.7** Sea $\langle A, h \rangle$ una estructura de argumento, y sea $\mathcal{T}_{\langle A, h \rangle}$ su árbol dialéctico asociado. [16] Se dice que A es una *justificación* para h (o bien se dice que $\langle A, h \rangle$ es una *justificación*) sssi el nodo raíz de $\mathcal{T}_{\langle A, h \rangle}$ es un nodo-U. □

---

[15] La especificidad [SCG94] establece un orden parcial entre argumentos, y establece un orden de preferencia entre ellos.

[16] De hecho, los árboles dialécticos deben satisfacer ciertos requerimientos para ser considerados árboles dialécticos *aceptables* (ver [SCG94]). Estos requerimientos involucran la consideración de ciclos y detección de inconsistencia dentro de árboles dialécticos. Estos aspectos, no obstante, merecen un análisis más detallado que excede los alcances de este trabajo.