

Un enfoque declarativo basado en juegos del razonamiento rebatible

Laura A. Cecchi

Guillermo R. Simari

Depto. de Informática y Estadística - Fa.E.A.
UNIVERSIDAD NACIONAL DEL COMAHUE
e-mail:lcecchi@uncoma.edu.ar

Depto. de Ciencias de la Computación
UNIVERSIDAD NACIONAL DEL SUR
e-mail:grs@cs.uns.edu.ar

PALABRAS CLAVES: Argumentos - Razonamiento Rebatible - Programación en Lógica

Resumen

La Programación en Lógica Rebatible Básica [4, 8, 9] (P.L.R.B.) es una extensión de la Programación en Lógica en la que se permite representar conocimiento contradictorio, a través de la noción de negación fuerte, y conocimiento tentativo, incorporando a la sintaxis una nueva clase de reglas, las reglas rebatibles.

La semántica operacional de la P.L.R.B. está basada en un formalismo no monotónico: los Sistemas Argumentativos. Las consecuencias de un programa lógico rebatible son determinadas a través de un análisis dialéctico de argumentos y contraargumentos. Si bien este mecanismo es suficiente para verificar si una consulta está garantizada o no en un programa lógico rebatible, en los últimos años, la semántica operacional ha sido estudiada desde un punto de vista declarativo[6, 10], con el objeto de determinar el significado preciso de un programa lógico sin recurrir al control.

El propósito de este trabajo es presentar los resultados parciales del estudio del sistema procedural de razonamiento no monótono de la P.L.R.B., en el que el criterio de decisión entre argumentos contradictorios es la especificidad. Se presenta un enfoque declarativo de la semántica operacional de la P.L.R., considerando las tres clases de respuestas posibles para un literal de la signature del programa lógico que se está analizando. Dicha caracterización está basada en el formalismo de juegos propuesto por Abramsky en [1].

1 Introducción

La semántica operacional de los programas lógicos está basada en el método de resolución de Robinson, aplicado a cláusulas de Horn. Esto supone una limitación tanto sintáctica como semántica que, a pesar de las ventajas teóricas y prácticas que trae consigo, restringe en algunos casos la aplicabilidad de la Programación en Lógica (de ahora en más P.L.) en resolución de problemas. La superación de esta limitación puede atacarse de varias maneras. Un modo es extender la ejecución clásica a nuevos procedimientos junto con un cambio en la sintaxis, resultando en un incremento del poder expresivo de la P.L..

La Programación en Lógica Rebatible Básica [4, 8, 9] (P.L.R.B.) es una extensión de la P.L. en la que se permite representar conocimiento contradictorio, a través de la noción de negación fuerte, y conocimiento tentativo, incorporando a la sintaxis una nueva clase de reglas, las reglas rebatibles.

La semántica operacional de la P.L.R.B. está basada en un formalismo no monotónico: los Sistemas Argumentativos. Las consecuencias de un programa lógico rebatible son determinadas a través de un análisis dialéctico de argumentos y contraargumentos. Si bien este mecanismo es suficiente para verificar si una consulta está garantizada o no en un programa lógico rebatible, en los últimos años, la semántica operacional ha sido estudiada desde un punto de vista declarativo [6, 10], con el objeto de determinar el significado preciso de un programa lógico sin recurrir al control.

El propósito de este trabajo es presentar los resultados parciales del estudio del sistema procedural de razonamiento no monótono de la P.L.R.B., en el que el criterio de decisión entre argumentos contradictorios es la especificidad. En la sección 2, se describen brevemente la sintaxis de la P.L.R.B. y su semántica operacional. En la siguiente sección, se describe la semántica declarativa desarrollada, basada en una propuesta matemática presentada por Abramsky en [1], formalizando la idea de familia de juego. Finalmente, en la sección 4, se presentan las conclusiones y los trabajos futuros.

2 Programación en Lógica Rebatible

La P.L.R. permite representar conocimiento tentativo y certero, y con el objeto de distinguirlos consta de dos clases diferentes de reglas: reglas estrictas y reglas rebatibles. El lenguaje de la P.L.R. está formado por todos los posibles literales fijos¹ [7] que se puedan obtener a partir de la signatura del programa y lo notamos Lit . En otras palabras, L pertenece a Lit si L es un átomo fijo A o un átomo fijo negado $\sim A$, siendo \sim la representación de la negación fuerte. Dado un literal L , notamos con \overline{L} al *complemento* de L con respecto a la negación fuerte y lo definimos como sigue: si L es el átomo A entonces $\overline{L} = \sim A$; si L es el átomo negado $\sim A$ entonces $\overline{L} = A$.

Definición 2.1. [4, 9] Una *Regla Estricta Básica* es un par ordenado $L \leftarrow L_1, \dots, L_n$ donde $L \in Lit$ y $\{L_1, \dots, L_n\}$ es un subconjunto de Lit , con $n \geq 0$. Si el cuerpo es vacío, i.e., $n = 0$, entonces la regla estricta se transforma en $L \leftarrow true$ o $L \leftarrow$ y es llamada *hecho*. Una *Regla Rebatible Básica* es un par ordenado $L \prec L_1, \dots, L_m$ donde $L \in Lit$ y $\{L_1, \dots, L_m\}$ es un subconjunto de Lit , con $m > 0$. ■

Las reglas rebatibles son reglas que pueden ser derrotadas en presencia de evidencia contraria. Estas reglas pueden interpretarse como *razones para creer en el antecedente* *Cuerpo* *proveen razones para creer en el consecuente* *Cabeza* y sus consecuencias siempre estarán sujetas a debate.

Definición 2.2. [4, 9] Un *programa lógico rebatible básico* (de ahora en más *plrb*) es un conjunto de reglas estrictas básicas y rebatibles básicas. Si \mathcal{P} es un *plrb*, distinguiremos el subconjunto Π de reglas estrictas básicas en \mathcal{P} , y el subconjunto Δ de reglas rebatibles básicas en \mathcal{P} . Cuando se requiera denotaremos \mathcal{P} como (Π, Δ) . ■

Siempre que no sea ambiguo, nos referiremos a los *plrb* simplemente como programas lógicos rebatibles (*plr*).

El lenguaje de un *plr* no permite el uso de variables, y por lo tanto, un *plr* no permitirá en la definición de sus reglas términos no fijos. Es posible pensar a un programa lógico que contenga variables como un *esquema* del programa [4]. Así una regla R que contenga variables es un esquema de regla y puede ser vista

¹ *Ground* en inglés. En la mayoría de la literatura en idioma español, se traduce este vocablo como *básico*. Sin embargo, creemos que la traducción introducida por Fillottrani en [7] es más adecuada.

como una notación abreviada de todas sus posibles instanciaciones fijas. Este concepto puede ser extendido para los *plr*.

La semántica operacional de la P.L.R. se basa en los *sistemas de argumentación*, que constituyen una formalización del razonamiento no monótono. Con el objeto de explicar una creencia, un agente realiza un análisis dialéctico teniendo en cuenta las pruebas tentativas que pueda construir para tal creencia. Las explicaciones que soportan a la creencia se denominan *argumentos* y son el fundamento del razonamiento rebatible.

Definición 2.3. [9] Sea $\mathcal{P} = \langle \Pi, \Delta \rangle$ un programa lógico rebatible. Una *estructura de argumento* para un literal fijo h en el contexto Π , que denotaremos $\langle \mathcal{A}, h \rangle_\Pi$ o simplemente $\langle \mathcal{A}, h \rangle$, si \mathcal{A} es un subconjunto de Δ que cumple las siguientes condiciones:

1. h se deriva rebatiblemente a partir de $\Pi \cup \mathcal{A}$,
2. $\Pi \cup \mathcal{A}$ no es contradictorio, y
3. \mathcal{A} es minimal con respecto a la inclusión de conjuntos, i.e., no existe $\mathcal{A}' \subset \mathcal{A}$ tal que \mathcal{A}' satisface las dos primeras condiciones.

Una estructura de argumento $\langle \mathcal{B}, h' \rangle$ es un *subargumento* de $\langle \mathcal{A}, h \rangle$ si $\mathcal{B} \subseteq \mathcal{A}$.

Dos estructuras de argumento $\langle \mathcal{B}, h' \rangle$ y $\langle \mathcal{A}, h \rangle$ están en *desacuerdo* cuando $\Pi \cup \{h, h'\}$ sea contradictorio. $\langle \mathcal{B}', q \rangle$ se denomina *contraargumento* de $\langle \mathcal{A}, h \rangle$ si existe un subargumento $\langle \mathcal{A}', h' \rangle$ de $\langle \mathcal{A}, h \rangle$ y $\langle \mathcal{A}', h' \rangle$ y $\langle \mathcal{B}', q \rangle$ están en desacuerdo. ■

Aunque cuando se referencia a un *argumento* para h se hace mención de \mathcal{A} , siempre que no sea ambiguo abusaremos del lenguaje denominando a la estructura de argumento $\langle \mathcal{A}, h \rangle$ simplemente argumento.

De todos los contraargumentos que puedan existir para un argumento, nosotros estaremos interesados solamente en aquellos que lo derroten. Con el objeto de poder decidir entre contraargumentos, se define una relación de preferencia entre argumentos. Esta relación puede ser especificada de diferentes maneras, como por ejemplo, prioridad entre reglas [11] y especificidad[9]. Para este trabajo, asumiremos que la relación de preferencia entre argumentos es un orden total genérico.

El análisis dialéctico, que puede ser visto gráficamente como un árbol, se construye comenzando con un argumento para la consulta, considerando, luego, todos los contraargumentos que lo derroten. Ya que los contraargumentos son argumentos, se deberá considerar los contraargumentos de los contraargumentos y así sucesivamente.

Un literal está garantizado si en el árbol que representa su análisis dialéctico resulta que cada contraargumento que podría derrotar al argumento en la raíz es a su vez derrotado, teniendo en cuenta que las hojas del árbol son argumentos no derrotados. En caso de que un literal h esté garantizado diremos Bh , i.e., el agente cree en h . Las posibles respuestas a una consulta h son[9]:

SI:	si Bh (cree en h)
NO:	si $B\bar{h}$ (cree en \bar{h})
INDECISO:	si $\neg Bh$ y $\neg B\bar{h}$ (no cree en h ni en \bar{h})
DESCONOCIDO:	si h no pertenece a la signatura del programa

La formalización de estas ideas pueden encontrarse en detalle en [9]. En la siguiente sección se analiza la semántica operacional desde un punto de vista declarativo.

3 Semántica declarativa basada en juegos

Los juegos [1] tienen una analogía con el razonamiento basado en argumentos. Un árbol dialéctico puede ser visto como un juego donde de manera alternada, el jugador P propone un argumento para un literal y el jugador O trata de derrotarlo. Dada una estructura de argumento $\langle \mathcal{A}, h \rangle$ para un literal h , un juego que comienza con tal estructura de argumento será denotado $G_{\langle \mathcal{A}, h \rangle}$.

Informalmente, en el contexto de la P.L.R., si \mathcal{P} gana un juego $G_{\langle \mathcal{A}, h \rangle}$ cuya primera movida es una estructura de argumento para un literal h en un programa rebatible \mathcal{P} , entonces h pertenecerá al conjunto de literales garantizados de \mathcal{P} .

En [4] se introdujo la definición declarativa de argumento y se probó su equivalencia con la contraparte operacional. Dicha definición está basada en las consecuencias lógicas rigurosas [4] de un *plr*. Si el programa es consistente, entonces un literal pertenecerá al conjunto de sus consecuencias rigurosas, si es la consecuencia de una regla estricta o rebatible aplicable. Si el programa es inconsistente entonces sus consecuencias coinciden con *Lit*, i.e., todo el lenguaje es consecuencia de un programa inconsistente.

Definición 3.1. [4] Sea $\mathcal{P} = \langle \Pi, \Delta \rangle$ un *plr*. Un argumento \mathcal{A} para un literal fijo q es un subconjunto de las reglas rebatibles de \mathcal{P} , $\mathcal{A} \subseteq \Delta$, tal que:

1. $q \in Cn_R(\Pi \cup \mathcal{A})$
2. $Cn_R(\Pi \cup \mathcal{A})$ es consistente, i.e., $Cn_R(\Pi \cup \mathcal{A}) \neq Lit$
3. \mathcal{A} es minimal con respecto a la inclusión, i.e., no existe $\mathcal{A}' \subseteq \mathcal{A}$ que cumpla (1) y (2). ■

Con la formalización declarativa de argumento, podremos definir las posibles movidas de un juego, que notaremos $M_{G_{\langle \mathcal{A}, h \rangle}}$, como un conjunto de estructuras de argumentos. De este modo, $M_{G_{\langle \mathcal{A}, h \rangle}}$ es un subconjunto de Arg_h , donde $\langle \mathcal{A}, h \rangle$ es el primer movimiento en el juego:

$$\begin{aligned} Arg_h^0 &= \{ \langle \mathcal{A}, h \rangle \mid \mathcal{A} \text{ es un argumento de } h \} \cup \{ \langle \mathcal{A}, \bar{h} \rangle \mid \mathcal{A} \text{ es un argumento de } \bar{h} \} \\ Arg_h^i &= \{ \langle \mathcal{B}, l \rangle \mid l \in Cn_R(\Pi \cup \mathcal{B}') \text{ y } \langle \mathcal{B}', h' \rangle \in Arg_h^{i-1} \} \cup \\ &\quad \{ \langle \mathcal{B}, \bar{l} \rangle \mid l \in Cn_R(\Pi \cup \mathcal{B}') \text{ y } \langle \mathcal{B}', h' \rangle \in Arg_h^{i-1} \} \\ Arg_h &= \bigcup_i Arg_h^i \end{aligned}$$

Arg_h contiene todos los posibles puntos de ataque (argumentos y contraargumentos) para el literal h y su complemento \bar{h} . Dado un literal h , $Arg_h = Arg_{\bar{h}}$. Notaremos $M_{G_{\langle \mathcal{A}, h \rangle}}^*$ al conjunto de todas las secuencias que se puedan obtener a partir de $M_{G_{\langle \mathcal{A}, h \rangle}}$.

Notaremos s_i al i -ésimo elemento de una secuencia s . Sea $s_i = \langle \mathcal{A}, h \rangle$ notaremos la proyección del primer elemento de s_i , $s_i^{\mathcal{A}} = \mathcal{A}$ y $s_i^h = h$ a la proyección del segundo elemento de s_i .

No siempre las discusiones entre dos agentes son correctas. En el proceso dialéctico es posible encontrar fallas que invalidan la discusión. Al haber restringido la relación de preferencia que debe ser usada para decidir entre información contradictoria, no es posible encontrar derrotadores recíprocos y, por lo tanto, se evitan los ciclos en las secuencias. Sin embargo, existen algunas falacias que deberemos tener en cuenta por lo que necesitaremos distinguir entre todas las secuencias aquellas que son legales en nuestro juego.

Definición 3.2. Sea $\langle \Pi, \Delta \rangle$ un programa lógico rebatible donde la relación de preferencia es un orden total. Una secuencia finita $s \in M_{G_{\langle \mathcal{A}, h \rangle}}^*$ es legal si

- (a) evita las inconsistencia entre las jugadas del mismo jugador, i.e., $1 \leq i \leq |s|$

$$\Pi \cup \bigcup_{even(i)} s_i^{\mathcal{A}} \not\models \perp \text{ and } \Pi \cup \bigcup_{odd(i)} s_i^{\mathcal{A}} \not\models \perp$$

- (b) no existe s_k tal que $s_k^{\mathcal{A}} \subset s_i^{\mathcal{A}}$, $i < k \leq |s|$, siendo $|s|$ la longitud de s . ■

Con toda esta artillería estamos en condiciones de definir un juego.

Definición 3.3. Sea $\mathcal{P} = (\Pi, \Delta)$ un programa lógico rebatible, h un literal y $\langle \mathcal{A}, h \rangle$ una estructura de argumento para h . Un *juego* para $\langle \mathcal{A}, h \rangle$ con respecto a \mathcal{P} , que notaremos $G_{\langle \mathcal{A}, h \rangle}$, es una estructura $(M_{G_{\langle \mathcal{A}, h \rangle}}, \lambda_{G_{\langle \mathcal{A}, h \rangle}}, P_{G_{\langle \mathcal{A}, h \rangle}})$, donde

- $M_{G_{\langle \mathcal{A}, h \rangle}} \subseteq \text{Arg}_h$.
- $\lambda_{G_{\langle \mathcal{A}, h \rangle}} : M_{G_{\langle \mathcal{A}, h \rangle}} \rightarrow \{P, O\}$;
- $P_{G_{\langle \mathcal{A}, h \rangle}} \subseteq M_{G_{\langle \mathcal{A}, h \rangle}}^{\text{alt}}$, donde $P_{G_{\langle \mathcal{A}, h \rangle}}$ es no vacío y cerrado con respecto a los prefijos de las secuencias legales [5], y $M_{G_{\langle \mathcal{A}, h \rangle}}^{\text{alt}}$ es el conjunto de todas las secuencias legales finitas $s \in M_{G_{\langle \mathcal{A}, h \rangle}}^*$ tales que
 1. Para todo $i : 1 \leq i \leq |s|$

$$\lambda_{G_{\langle \mathcal{A}, h \rangle}}(s_i) = \begin{cases} O & \text{si } i \text{ es par} \\ P & \text{si } i \text{ es impar} \end{cases}$$

siendo $|s|$ la longitud de s .

2. Cada secuencia s de $P_{G_{\langle \mathcal{A}, h \rangle}}$ satisface:
 - (a) $(s = \epsilon)$ siendo ϵ la secuencia vacía o existe s' , tal que $s = [\langle \mathcal{A}, h \rangle]s'$
 - (b) Para todo $1 \leq i \leq (|s| - 1)$ existe una estructura de argumento s' para h' tal que $s'^{\mathcal{A}} \subseteq s_i^{\mathcal{A}}$ y $\Pi \cup \{s'^h \cup s_{i+1}^h\}$ es inconsistente.
 - (c) Si $s = s'[\langle \mathcal{A}_1, h_1 \rangle] \in P_{G_{\langle \mathcal{A}, h \rangle}}$ donde s' es una secuencia posiblemente vacía, entonces para cada estructura de contraargumento $\langle \mathcal{A}_2, h_2 \rangle$ de $\langle \mathcal{A}_1, h_1 \rangle$, existe una secuencia $t \in P_{G_{\langle \mathcal{A}, h \rangle}}$ tal que $t = s[\langle \mathcal{A}_2, h_2 \rangle] = s'[\langle \mathcal{A}_1, h_1 \rangle, \langle \mathcal{A}_2, h_2 \rangle]$, siempre y cuando t sea legal. ■

Dependiendo del jugador, cada movida captura un argumento de soporte o de interferencia con respecto al argumento inicial. Por esta razón, estamos solamente interesados en secuencias de movimiento de cierta clase: aquellas que capturan la construcción del árbol dialéctico. Estas secuencias son capturadas a través de las tres condiciones impuestas en la definición 3.3. La condición (1) expresa que los jugadores deberán jugar en forma alternada. El punto (2) captura en el inciso (a) que el juego es vacío o bien comienza con la estructura de argumento $\langle \mathcal{A}, h \rangle$. El siguiente inciso condiciona los movimientos a argumentos que se ataquen, i.e., cada movimiento es un contraargumento para el argumento en la movida precedente. por último, se exige que cada juego contemple todos los contraargumentos de cada argumento jugado.

Definición 3.4. [3] Sea a el primer movimiento del proponente en el juego. Una secuencia s es completa si $s = [a]s_1$, con s_1 posiblemente vacía, entonces no existe ningún movimiento $b \in M_{G_{\langle \mathcal{A}, h \rangle}}$ tal que $[a]s_1[b] \in P_{G_{\langle \mathcal{A}, h \rangle}}$. Una secuencia s' se dice preferida si cada movimiento del oponente tiene una respuesta del proponente. En otras palabras, una secuencia s' es preferida si $|s'|$ es impar. Definimos a una estrategia σ sobre un juego G como un conjunto de secuencias preferidas de P_G . ■

Una secuencia completa es una línea de argumentación [9], i.e. una camino desde el primer movimiento hasta un movimiento que nos permita alcanzar una hoja. Una secuencia preferida s' captura la idea de que s' termina con un movimiento del proponente.

Dado un literal h , el proponente podría jugar como movimiento inicial, cualquiera de los argumentos que soporta a h . De este modo, tendríamos para cada literal un juego por cada argumento que lo soporta.

Definición 3.5. Sean \mathcal{P} un *plr*, h un literal bajo la signatura de \mathcal{P} , $\langle \mathcal{A}_1, h \rangle, \dots, \langle \mathcal{A}_n, h \rangle$ todos las estructuras de argumento de h bajo \mathcal{P} y $G_{\langle \mathcal{A}_1, h \rangle}, G_{\langle \mathcal{A}_2, h \rangle}, \dots, G_{\langle \mathcal{A}_n, h \rangle}$ los juegos correspondientes a todos los argumentos de h . Diremos que $\{G_{\langle \mathcal{A}_1, h \rangle}, \dots, G_{\langle \mathcal{A}_n, h \rangle}\}$ es la familia de juegos de h y la notaremos \mathcal{F}_h . ■

Definición 3.6. Sean \mathcal{P} un *plr*, h un literal bajo la signatura de \mathcal{P} , \mathcal{F}_h la familia de juegos del literal h y $\mathcal{F}_{\bar{h}}$ la familia de juegos del literal \bar{h} para el *plr* \mathcal{P} . La semántica de juegos que notaremos \mathcal{GS} es una tupla $\langle V, F \rangle$ tal que :

- Si existe un juego $G_{\langle \mathcal{A}, h \rangle}$ en la familia \mathcal{F}_h , tal que el conjunto de las secuencias completas de $P_{G_{\langle \mathcal{A}, h \rangle}}$ es una estrategia, entonces h pertenece a V .

- Si existe un juego $G_{\langle \mathcal{A}, \overline{h} \rangle}$ en la familia $\mathcal{F}_{\overline{h}}$, tal que el conjunto de las secuencias completas de $P_{G_{\langle \mathcal{A}, \overline{h} \rangle}}$ es una estrategia, entonces h pertenece a F .
- De otro modo, h pertenece a I . ■

La definición anterior no contempla la respuesta DESCONOCIDO, ya que sólo consideramos literales bajo la signatura del *plr*. A partir de \mathcal{GS} no sólo caracterizamos declarativamente a la P.L.R.B., sino que podemos distinguir las diferentes formas de garantizar un literal a través de la familia de juegos.

4 Conclusiones y Trabajos Futuros

Se ha presentado un enfoque declarativo de la semántica operacional de la P.L.R., considerando las tres clases de respuestas posibles para un literal de la signatura del *plr* que se está analizando. Dicha caracterización está basada en el formalismo de juegos propuesto por Abramsky en [1]. Este resultado parcial motivó la etapa actual en la investigación que involucra la eliminación de la restricción sobre el criterio de decisión entre argumentos. En particular, es nuestro interés estudiar el criterio de preferencia entre argumentos basado en la especificidad.

Entre nuestros trabajos futuros se encuentran adaptar dicha semántica a otros sistemas, como el propuesto en [2] y decidir si esta caracterización puede ayudar a dar significado a programas lógicos rebatibles que incluyan la negación por falla.

Referencias

- [1] Samson Abramsky. Semantics of Interaction. In A.Pitts and P. Dibyer, editors, *Semantics and Logic Computation*. Cambridge, 1997.
- [2] G. Antoniou, D. Billington, and M.J. Maher. Normal forms for defeasible logic. In J. Jaffar, editor, *Proceedings International Joint Conference and Symposium on Logic Programming*, pages 160–174. MIT Press, 1998.
- [3] Laura A. Cecchi and Guillermo R. Simari. Game-based approach for modeling dialectical analysis: Preliminary Report. In *Proceedings of V CACiC*, 1999.
- [4] Laura A. Cecchi and Guillermo R. Simari. Sobre la Relación entre la Definición Declarativa y Procedural de Argumento. Ushuaia, 2000. VI CACiC.
- [5] Laura A. Cecchi and Guillermo R. Simari. Una semántica declarativa basada en juegos para la programación en lógica rebatible básica. In *Proceedings of ICIE*, 2000.
- [6] Phan M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning and logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.
- [7] Pablo Rubén Fillottrani. *La negación en la Programación en Lógica*. PhD thesis, Universidad Nacional del Sur, 2001.
- [8] A. García and G. R. Simari. Strong and Default Negation in Defeasible Logic Programming. In *4th Dutch/German Workshop on Nonmonotonic Reasoning Techniques and their applications*, Amsterdam, 25 - 27, Marzo 1999.
- [9] Alejandro J. García. *Programación en Lógica Rebatible: Lenguaje, Semántica Operacional y Paralelismo*. PhD thesis, Universidad Nacional del Sur, 2000.
- [10] A. Kakas and F. Toni. Computing argumentation in logic programming. *Journal of Logic and Computation*, (4):515–562, 1999.
- [11] Henry Prakken and Gerard Vreeswijk. Logics for defeasible argumentation. In D. Gabbay, editor, *Handbook of Philosophical Logic*. Kluwer Academic, second edition, 2000.