

# Un formalismo para persuasión con DeLP

Guillermo Aguirre, Marcelo Errecalde<sup>1</sup>, Guillermo Simari<sup>2</sup>

<sup>1</sup> Lab. de Investigación y Desarrollo en Inteligencia Computacional (LIDIC),  
{gaguirre,merreca}@uns1.edu.ar

Departamento de Informática. Universidad Nacional de San Luis  
Ejército de los Andes 950 - Local 106. (D5700HHW) - San Luis - Argentina

<sup>2</sup> Laboratorio de Investigación y Desarrollo en Inteligencia Artificial (LIDIA),  
grs@cs.uns.edu.ar

Depto. de Ciencias e Ingeniería de la Computación. Univ. Nacional del Sur,  
Av. Alem 1253, (B8000CPB) Bahía Blanca, Argentina

**Resumen** La comunicación entre agentes requiere estar regulada y debe cumplir ciertos formalismos para facilitar la tarea de cada uno de los agentes involucrados. Bajo esos formalismos, la persuasión permite que dos agentes se pongan de acuerdo sobre alguna situación particular. El acuerdo se logra mediante los protocolos de persuasión, los cuales entre otras cosas requieren de dos lenguajes para formar las frases, por un lado, un lenguaje de comunicación general y por otro, un lenguaje específico del contexto del diálogo. Es necesario que este último lenguaje resuelva situaciones de conflicto en favor de quien haga sus planteos con argumentos más sólidos, para lo cual es conveniente el uso de algún lenguaje de programación lógica. En este trabajo se propone el uso de DeLP como dicho lenguaje, ya que posee las características requeridas, y mediante un programa de ejemplo se muestra que esta propuesta es viable.

Palabras claves juegos de diálogo, argumentación, DeLP.

## 1. Introducción

La comunicación entre los seres humanos consiste en el intercambio estructurado de frases y cuando se respetan las reglas propias del entorno, se consigue transmitir ideas de una manera racional. Cada frase se emite por una determinada razón, la que puede o no ser alcanzada por distintos motivos. Esta característica imprecisa de los diálogos ha dado origen a numerosos estudios: *Cómo hacer las cosas con palabras* [1] y *Teoría de acción comunicativa* [14]. A pesar de los diversos enfoques, existe coincidencia en considerar a las frases cortas como *actos del habla* porque pueden cambiar la realidad del mismo modo que las acciones físicas. Lo que resulta evidente es la necesidad de establecer contextos o sistemas que regulen los distintos diálogos posibles, especialmente cuando los participantes son agentes computacionales. Este trabajo se concentra en una clase particular de comunicación que ocurre habitualmente, una parte sostiene algo mediante argumentos que la otra parte pone a prueba.

El término “diálogo de persuasión” es utilizado por Walton & Krabber [16] como parte de su influyente clasificación de los diálogos en seis tipos de acuerdo a su finalidad. Mientras que la *persuasión* procura resolver una diferencia de opinión, la *negociación* trata de resolver un conflicto de intereses mediante un trato, la *búsqueda de información* alienta la transferencia de información, la *deliberación* procura alcanzar una decisión sobre un curso de acción, una *consulta* pretende “incrementar el conocimiento y el acuerdo” y la *disputa* es el sustituto verbal de la pelea. De acuerdo a esta clasificación es posible que los diálogos de una clase se conviertan en otro tipo. En particular, un diálogo de cualquier tipo puede transformarse en uno de persuasión cuando surge un conflicto de opinión. Por ejemplo en la búsqueda de información, puede surgir un conflicto de opinión sobre la credibilidad de una fuente de información, en una deliberación los participantes pueden disentir sobre los probables resultados de los planes o acciones y en una negociación los actores involucrados pueden no coincidir sobre las razones por las que una propuesta resulta de interés para alguno de ellos.

La estructura de comportamiento requerida en los diálogos se asemeja a las reglas de un juego; un tipo de juegos practicado entre oradores, quienes deben presentar diversas sentencias de acuerdo a reglas estrictas que han ido evolucionando en forma de distintos procedimientos. Por ejemplo, en [4,6,15] se plantean algunos juegos que modelan el razonamiento rebatible como una disputa entre proponente y oponente de una proposición. El proponente comienza con un argumento para la proposición inicial, después de lo cual, cada uno debe atacar algún argumento previo del otro participante con un contraargumento de fuerza suficiente. Los ataques pueden estar dirigidos directamente a la conclusión, lo que se conoce como *refutar* (en inglés rebutting), o bien estar dirigidos a las premisas que soportan la conclusión, *socavar* (en inglés undercutting) [5]. La proposición inicial resulta probada si el proponente tiene una estrategia ganadora, es decir, si él puede hacer que el oponente se quede sin movidas para hacer cualquier ataque. Claramente, este entorno dialéctico se corresponde bien con las aplicaciones de sistemas de diálogo mencionadas anteriormente.

Este trabajo sigue el sistema formal para persuasión presentando en [8] y [9] y plantea crear una instancia del protocolo para los juegos de diálogo propuesto por Prakken pero usando DeLP [2,3]. El fundamento de esta idea son las similitudes existentes entre el sistema argumentativo usado en la propuesta original (el que denominaremos PraTor [10]) y DeLP. En particular el hecho que ambos sistemas usan semántica fijas [11], una condición requerida en [7, pag. 70].

**Ejemplo 1** *Para mostrar el uso de persuasión en el mundo real se adaptó un ejemplo presentado en [13]. El problema es el siguiente: un par de amigos viajan en auto y uno de ellos sugiere, gracias la información del contexto que obtiene dinámicamente del GPS del vehículo, parar en un hotel cercano. ¿Mediante qué elementos se podrán sostener argumentos a favor de la sugerencia?*

El trabajo se organiza así, la sección 2 plantea el uso de persuasión para lograr acuerdos entre agentes racionales y se presenta un ejemplo. Las características generales del sistema de argumentación DeLP se presentan en la Sección 3,

ahí mismo está el Ejemplo 1 como un programa DeLP. Luego en la Sección 4 se explican los principales elementos del protocolo para persuasión y se muestra cómo se puede instanciar el formalismo empleando el lenguaje DeLP. En la Sección 5 se presentan algunas conclusiones y futuras líneas de investigación.

## 2. Persuasión

En los diálogos de persuasión, la situación inicial es un conflicto de opinión lo que impide alcanzar una efectiva comunicación; para superar esta circunstancia se procura hallar una solución basada en la comunicación, una que sea dialógica. En esta búsqueda se encuentran los sistemas de diálogo que se caracterizan por establecer una serie de reglas sobre el uso de los actos del habla, permitiendo que los participantes intercambien afirmaciones o reclamos, planteen desafíos sobre las afirmaciones de otro participante<sup>3</sup>, acepten las afirmaciones del otro, puedan retirar las afirmaciones propias hechas anteriormente o presenten argumentos a favor de ellas. Estos sistemas de diálogo han sido abordados de una manera formal desde mediados del siglo pasado. La idea que la argumentación está relacionada con el diálogo, por lo tanto con las obligaciones y los *compromisos* en el diálogo, es tan vieja como la lógica misma.

**Ejemplo 2** *En la Figura 1 se presenta un posible diálogo entre los agentes involucrados en la situación planteada en el Ejemplo 1.*

A:Sugiero parar en este hotel. (*Hace un reclamo*)  
 B:¿Por qué parar en este hotel?(*Consulta sobre los fundamentos del reclamo*)  
 A:Ya debemos parar y este hotel está cerca.(*Brinda fundamentos del reclamo*)  
 B:¿Por qué debemos parar?(*Consulta sobre los fundamentos de un sub-argumento*)  
 A:Hemos manejado durante muchas horas.(*Brinda fundamentos del reclamo*)  
 B:Pero hemos hecho muchas paradas.(*Establece un contraargumento*)  
 A:Debemos parar debido a que es de noche.(*Establece otro contraargumento*)  
 B:Si es cierto, debemos parar.(*Acepta el sub-argumento anterior*)  
     Pero no en este hotel, esta zona es peligrosa.(*Establece un nuevo contraargumento*)  
 A:¿Por qué esta zona es peligrosa?(*Consulta los fundamentos del contraargumento*)  
 B:Debido a que hay muchos robos.(*Brinda fundamentos del contraargumento*)  
 A:De acuerdo, no fue buena la sugerencia de parar aquí.(*Retira el reclamo inicial*)  
     Mejor no paremos en este hotel.(*Acepta el nuevo contraargumento*)

**Figura 1.** Ejemplo de diálogo entre agentes racionales

Se establecen algunas restricciones sobre las frases admitidas durante la comunicación, de esta manera se facilita tanto la interpretación como la alternancia

<sup>3</sup> Lo que en términos legales se conoce como reconvencción: cuando el demandado inicia acciones contra el demandante.

de los turnos. Cada participante de la comunicación contará con un *almacén de compromisos*, el cual representa el punto de vista públicamente declarado por el agente sobre una proposición; esta perspectiva no necesita coincidir con las creencias internas del agente. Pero sí es necesario que el contenido del almacén de compromisos mantenga consistencia.

### 3. Argumentación en DeLP

Un programa en DeLP[2,3] es  $P = \langle \Theta, \Omega, \Delta \rangle$ , un conjunto de hechos, reglas estrictas, y reglas rebatibles. En ocasiones se usa  $\Pi = \Theta \cup \Omega$ . Los hechos de  $\Theta$ , son literales fijos identificados como  $L_i$ . En los dos tipos de reglas, el literal a la izquierda de la flecha es la *cabeza* y los literales que están a la derecha de la flecha son el *cuerpo*. Las reglas estrictas junto con los hechos representan el conocimiento incuestionable. Las reglas rebatibles sostienen conocimiento tentativo: razones para creer en *cuerpo* proveen razones para creer en *cabeza*. Los elementos del conjunto de reglas estrictas  $\Omega$  se caracterizan como  $L_0 \leftarrow L_1, \dots, L_k$ , por su parte, las reglas rebatibles de  $\Delta$  se muestran como  $L_0 \prec L_1, \dots, L_k$ , puede ocurrir que el cuerpo de estas reglas sea vacío; en ese caso “se supone  $L_0$ ”.

Para explicar cómo funciona DeLP se puede comenzar diciendo que a partir de un programa  $P$  se hace una derivación rebatible de literales. Con  $P \vdash L$  se indica que el programa deriva rebatiblemente el literal  $L$ , el último de una secuencia de literales fijos. Cada elemento de la secuencia es un hecho o la cabeza de una regla (estricta o rebatible) en la cual todos los literales del cuerpo han aparecido antes en la secuencia. Considerando la amplia diversidad de alternativas, es posible derivar literales *contradictorios* (por ejemplo  $L$  y  $\bar{L}$ ).

Luego se continúa con el *procedimiento de prueba*, que es un mecanismo global mediante el cual se determina cuáles son los literales *aceptados* a partir de un programa particular. El mecanismo de aceptación mencionado sirve para seleccionar ciertos *argumentos* adecuados a un literal específico. Para cualquier programa  $P$ , un argumento es una estructura con dos elementos:  $A$  que es un conjunto de reglas rebatibles en  $P$  y  $L$  que es un literal obtenido a partir de una derivación rebatible de  $P$ . También se debe cumplir lo siguiente:

1.  $\Pi \cup A \vdash L$
2.  $\Pi \cup A$  no es contradictorio
3.  $A$  es mínimo.

Para continuar con la explicación, es necesario aclarar lo siguiente:  $\langle B, Q \rangle$  es un *sub-argumento* de  $\langle A, L \rangle$  si  $Q \subseteq A$ . Hecha la aclaración, se introduce un concepto importante. El argumento  $\langle A_1, L_1 \rangle$  *ataca* a otro argumento  $\langle A_2, L_2 \rangle$  en el literal  $L$ , si  $\langle A, L \rangle$  es un sub-argumento de  $\langle A_2, L_2 \rangle$ , tal que  $L_1$  y  $L$  son contradictorios o están en desacuerdo.

El hecho que un programa pueda derivar literales contradictorios da origen al ataque entre sus correspondientes argumentos. El ataque entre los argumentos se resuelve mediante comparación, este procedimiento de comparación es el que determina si un argumento derrota a otro.

Se usa  $\langle A_1, L_1 \rangle \succ \langle A, L \rangle$  para indicar la preferencia estricta del argumento que aparece a la izquierda del operador  $\succ$ . Para profundizar sobre la preferencia entre argumentos revisar las referencias [12,3,13].

Es posible ahora formalizar la derrota entre argumentos:  $\langle A_1, L_1 \rangle$  derrota  $\langle A_2, L_2 \rangle$  en  $L$ , si  $\langle A, L \rangle$  es un sub-argumento de  $\langle A_2, L_2 \rangle$  y  $L$  es el punto de *contra-argumentación* del ataque de  $\langle A_1, L_1 \rangle$  a  $\langle A_2, L_2 \rangle$ . Además se debe cumplir una de las dos alternativas:

- $\langle A_1, L_1 \rangle \succ \langle A, L \rangle$ . Es decir,  $\langle A_1, L_1 \rangle$  es un derrotador propio de  $\langle A, L \rangle$ .
- $\langle A_1, L_1 \rangle \asymp \langle A, L \rangle$ <sup>4</sup>. Donde  $\langle A_1, L_1 \rangle$  es llamado un *derrotador por bloqueo*.

Un argumento que no es derrotado brinda razones para creer en el literal que lo integra. Resulta interesante, entonces, considerar lo que ocurre cuando hay una secuencia de derrotas entre los argumentos de un programa. Es decir, cuando un primer argumento derrota al segundo pero un tercer argumento derrota al primero. En ese caso el literal del segundo argumento se mantiene en vigencia como no derrotado. La situación anterior caracteriza lo que se llama una *línea de argumentación*.

Dado  $\langle A_1, L_1 \rangle$ , la secuencia  $\Lambda = [\langle A_1, L_1 \rangle, \dots, \langle A_{i+1}, L_{i+1} \rangle]$  donde cada  $\langle A_j, L_j \rangle$  derrota al anterior, es una línea de argumentación. Los argumentos en las posiciones impares tienen el rol de *soporte* del literal del primer argumento. En ese mismo sentido, los de posiciones pares son llamados argumentos de *interferencia*. Además  $\Lambda$  debe cumplir las siguientes condiciones:

1.  $\Lambda$  es finita,
2. El conjunto de argumentos de soporte en  $\Lambda$  es no contradictorio y el conjunto de argumentos de interferencia en  $\Lambda$  también es no contradictorio,
3. Ningún argumento  $A_j$  en  $\Lambda$  es un sub-argumento de otro  $A_i$  con  $i < j$ .
4. Todo derrotador por bloqueo  $A_i$  en  $\Lambda$  solo puede ser seguido por un derrotador propio  $A_{i+1}$  en  $\Lambda$ .

Como cada argumento puede tener varios derrotadores, la ramificación que ocurre en las líneas de argumentación origina un árbol. Un *árbol de dialéctica*  $T_{\langle A_0, L_0 \rangle}$  se construye partiendo de  $\langle A_0, L_0 \rangle$  como raíz. Todo nodo  $N$  es interior si tiene al menos un derrotador, por lo cual, un nodo es una hoja solamente si no tiene derrotadores.

El *marcado* de un árbol de dialéctica es el resultado de un proceso que consiste en marcar o etiquetar cada nodo como derrotado “D” o no derrotado “U”. Se trabaja de la siguiente forma: los nodos hojas son marcados como “U”. Un nodo interno  $N$  será marcado como “D” si tiene al menos un hijo marcado como “U”,  $N$  será “U” si todos sus hijos están marcados como “D”.

Un Árbol marcado  $T_{\langle A_0, L_0 \rangle}^*$  resulta de interés porque el argumento  $\langle A_0, L_0 \rangle$  está *garantizado* si su marca es “U”. Todo argumento que está garantizado es especialmente útil porque ningún otro argumento del programa puede derrotarlo; eso quiere decir que resultará victorioso al ser usado en cualquier disputa. Si  $\langle A_0, L_0 \rangle$  está garantizado, entonces el literal que lo acompaña  $L_0$ , también lo está.

<sup>4</sup> El operador  $\asymp$  se usa cuando los argumentos son indiferentes o incomparables.

**Ejemplo 3** En la Figura 2 se muestra el programa DeLP que corresponde a la realidad presentada en el Ejemplo 1.

$$\Psi_h = \left\{ \begin{array}{l} mParadas \\ mHManejando \\ deNoche \\ hCercano(h1) \\ estrellas(h1,5) \\ zRobos(h1) \\ cVehicular \leftarrow tLento \end{array} \right\} \Delta_h = \left\{ \begin{array}{l} sugerir(X) \prec bHotel(X), hCercano(X) \\ sugerir(X) \prec sParar, hCercano(X) \\ \sim sugerir(X) \prec zPeligrosa(X) \\ zPeligrosa(X) \prec zRobos(X) \\ \sim zPeligrosa(X) \prec zPolicias(X) \\ bHotel(X) \prec estrellas(X,S), S \geq 3 \\ \sim bHotel(X) \prec estrellas(X,S), S < 3 \\ \sim sParar \prec mParadas \\ sParar \prec mHManejando \\ sParar \prec deNoche \end{array} \right\}$$

**Figura 2.** Programa DeLP para sugerir un hotel

Observe que el conjunto  $\Psi_h$  tiene seis hechos y una regla estricta. Estos hechos representan información sobre la situación actual y que puede ir variando durante el viaje. Hasta el momento se sabe que: se realizaron muchas paradas (mParadas), el conductor ha manejado muchas horas (mHManejando), ya es de noche (deNoche), “h1” es un hotel cercano (hCercano), “h1” es un hotel de cinco estrellas (estrellas(h1,5)), y “h1” está en una zona de robos (zRobos(h1)). La regla estricta representa que “hay congestión vehicular cuando el tránsito es muy lento”.

El conjunto  $\Delta_h$  contiene varias reglas rebatibles. Las primeras dos reglas representan razones para sugerir un hotel: “si es cercano y es un buen hotel” o “si es un hotel cercano y hay una razón que sugiere parar”. Las últimas tres reglas representan razones a favor y en contra de sugerir dejar de conducir para descansar, siendo que “si el conductor lleva muchas horas manejando o está manejando de noche éstas son razones para sugerir una parada”, mientras que “si durante el viaje el conductor ha realizado varias paradas, es una razón en contra de sugerir que el conductor pare”. Observe que la sexta y séptima reglas rebatibles son utilizadas para establecer si un hotel es bueno. La tercer regla expresa que “si un hotel está ubicado en una zona peligrosa entonces hay una razón tentativa para no sugerirlo”. Finalmente, la cuarta y quinta reglas pueden ser leídas de la siguiente manera: “un hotel está ubicado en una zona peligrosa si en la zona normalmente se reportan muchos casos de robos”, y por lo contrario “si la zona generalmente cuenta con presencia policial, entonces el hotel no está ubicado en una zona peligrosa”.

## 4. Juego de diálogo

Cada intervención de los participantes de un diálogo se denomina *movida* y consiste en algún acto del habla formado a partir de elementos del lenguaje de comunicación  $L_c$  y del lenguaje del tópico  $L_t$ .  $L_c$  está relacionado con aspectos como la asignación de los turnos para que los interlocutores hagan sus movidas, la vinculación de cada respuesta con la movida a la cual va dirigida o diferenciar claramente cuando una respuesta es un ataque o una admisión; es decir, que este lenguaje está involucrado con el desarrollo general del diálogo. Por su parte el  $L_t$  debe resolver los ataques entre las movidas y establecer de una manera racional cuando una movida derrota a otra. Con estos lenguajes y otras reglas más, el protocolo llega a especificar cómo se determina el ganador actual, cuándo es el turno de cada participante, cuándo se ha llegado a la terminación del diálogo, cuál es el efecto de cada frase sobre los compromisos de los participantes, etc. Para consultar más detalles acerca de los elementos del protocolo ver [8] y [9].

El principal interés de este marco es enfocarse en los diálogos, mientras se deja suficiente libertad como para realizar réplicas alternativas y para posponer las réplicas. Esta libertad se consigue mediante dos características principales del sistema. Primero,  $\mathcal{L}_c$  tiene una estructura de respuesta explícita, donde cada movida o bien ataca o admite a la movida objetivo. Un  $\mathcal{L}_c$  de ejemplo<sup>5</sup> con este formato se muestra en la Tabla 1. Segundo, el ganador se define para cada uno de los diálogos, ya sea que esté terminado o no; el resultado es determinado por el *estado dialógico* de la raíz del *árbol de diálogo*. Durante el diálogo las movidas pueden ser atacadas o admitidas de distintas formas por el otro participante, es así como se abren distintas posibilidades durante la comunicación; algo similar al árbol de dialéctica pero en este caso cada nodo en lugar de ser un argumento, es una movida como las que aparecen en la Tabla 1. El estado dialógico de una movida se define recursivamente como sigue: para determinar el estado del diálogo, se hace uso de las características del árbol de diálogos generado por la estructura de respuesta del  $\mathcal{L}_c$ , ya que dicho árbol mantiene el estado de las distintas movidas. Una movida está *adentro* si es admitida o si todas las réplicas que la atacan están *afuera*. (Esto implica que una movida sin réplicas está *adentro*). Y una movida está *afuera* si tiene una réplica que esté *adentro*. Entonces un diálogo está siendo ganado (actualmente) por el proponente si su movida inicial está *adentro*, mientras que está (actualmente) siendo ganada por el oponente en otro caso.

El lenguaje de comunicación propuesto es el mostrado en la Tabla 1 y está restringido por el requerimiento de que cada movida debe ser *relevante*. Esto hace que el protocolo sea de *respuesta inmediata*, lo cual implica que cada turno consista de cero o más admisiones seguidas por un ataque. Dentro de estos límites la posposición de respuesta está permitida, a veces incluso indefinidamente.

<sup>5</sup> La movida *se supone*  $\varphi$  no aparece en la propuesta original de Prakken, corresponde a una regla rebatible con cuerpo vacío en DeLP.

Acto	Ataques	Admisiones
reclamo $\varphi$	$\zeta$ por qué $\varphi$ ?	concedo $\varphi$
$\varphi$ dado $S$	$\zeta$ por qué $\psi$ ? ( $\psi \in S$ ) $\varphi'$ dado $S'$ ( $\varphi'$ dado $S'$ derrota $\varphi$ dado $S$ ) se supone $\varphi'$ ( $\Rightarrow \varphi'$ derrota $\varphi$ dado $S$ )	concedo $\psi$ ( $\psi \in S$ ) concedo $\varphi$
$\zeta$ por qué $\varphi$ ?	$\varphi$ dado $S$ se supone $\varphi$	retiro $\varphi$
se supone $\varphi$	$\varphi'$ dado $S$ se supone $\varphi'$	concedo $\varphi$
concedo $\varphi$		
retiro $\varphi$		

**Tabla 1.** Un ejemplo de lenguaje de comunicación

#### 4.1. Administración de los compromisos

A medida que progresa el diálogo se producen cambios en los respectivos almacenes de compromisos de los agentes. El conocimiento del agente no tiene nada que ver con los compromisos que van siendo asumidos durante el diálogo. Dentro de cada almacén, los compromisos deben mantener consistencia. Algunas de las locuciones de la Tabla 1 agregan o quitan proposiciones, a continuación se analizan los casos. Cada *reclamo* implica agregar la proposición al almacén de compromisos del agente que lo emitió. Cuando un agente *concede* un reclamo de otro agente, el agente que admite debe agregar la proposición del reclamo. Al *retirar* la proposición hecha en un reclamo previo, se la debe quitar del almacen de compromisos del emisor. Al usar el acto " $\varphi$  dado  $S$ " se deben incorporar el consecuente y las premisas que aparecen en el argumento involucrado.

**Ejemplo 4** *Continuación del ejemplo 2. Usando un lenguaje de comunicación con respuesta explícita como el propuesto, el diálogo entre los agente A y B, sería similar al mostrado en la Tabla 2.*

$P_1[-]:$ reclamo sugerir(h1)	$O_2[P_1]:$ $\zeta$ por qué sugerir(h1)?
$P_3[O_2]:$ sugerir(h1) dado sParar, hCercano(h1)	$O_4[P_3]:$ $\zeta$ por qué sParar?
$P_5[O_4]:$ sParar dado mHManejando	$O_6[P_5]:$ $\neg$ sParar dado mParadas
$P_7[O_6]:$ sParar dado deNoche	$O_8[P_7]:$ concedo sParar
$P_{10}[O_9]:$ $\zeta$ por qué zPeligrosa(h1)?	$O_9[P_1]:$ $\neg$ sugerir(h1) dado zPeligrosa(h1)
$P_{12}[O_2]:$ retiro sugerir(h1)	$O_{11}[P_{10}]:$ zPeligrosa(h1) dado zRobos(h1)
$P_{13}[O_9]:$ concedo $\neg$ sugerir(h1)	

**Tabla 2.** Protocolo relevante para el diálogo ejemplo



**Ejemplo 5** Al finalizar el diálogo del Ejemplo 4 los compromisos de cada agente son los siguientes:

Compromisos de  $A = \{sParar, hCercano(h1), mHManejando, deNoche, \neg sugerir(h1)\}$

Compromisos de  $B = \{sParar, mParadas, zPeligrosa(h1), zRobos(h1), \neg sugerir(h1)\}$

A lo largo del artículo se han presentado distintas situaciones de ataque y admisión entre movidas; para mostrar cómo se aplican estas ideas concretamente en el caso del diálogo del Ejemplo 4, se muestra a continuación el correspondiente árbol de diálogo.

**Ejemplo 6** Como se observa en la Figura 3, la raíz del árbol de diálogo se muestra en línea puntos indicando que el oponente es el actual ganador del diálogo.

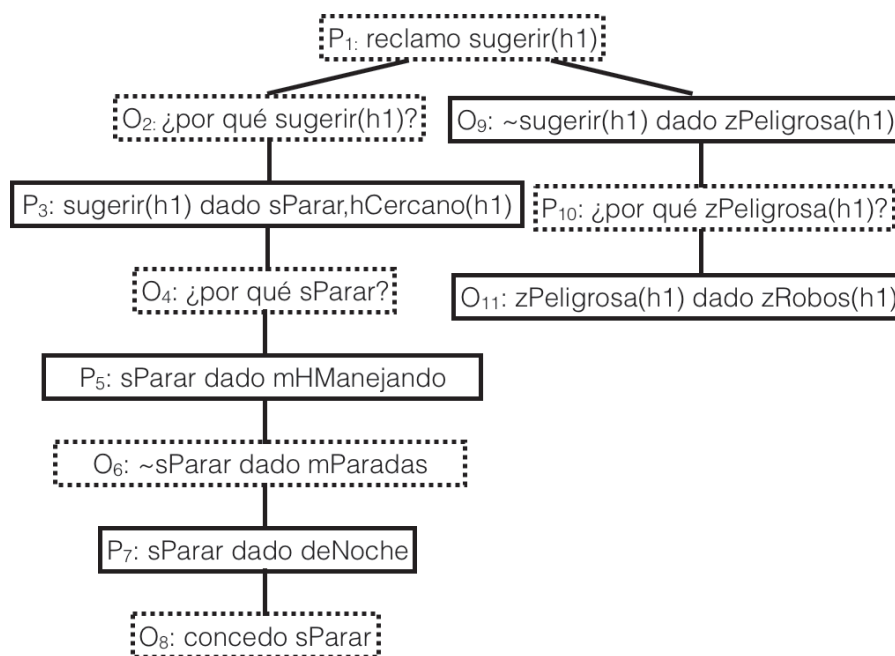


Figura 3. Árbol de diálogo

## 5. Consideraciones Finales

PraTor y DeLP son sistemas de argumentación que tienen suficientes similitudes como para sugerir el uso de cualquiera de los dos como lenguaje del tópico en el sistema formal para diálogos de persuasión propuesto por Prakken. En este

trabajo no se han discutido en detalle las similitudes entre Prator y DeLP, simplemente se ha mostrado un ejemplo en el que DeLP se utiliza como lenguaje del t3pico particular de dicho ejemplo. Este trabajo busca destacar la flexibilidad del protocolo de persuasi3n que permite ser aplicado en una situaci3n particular empleando un sistema argumentativo diferente al usado en la propuesta original. De esta manera se refuerza la idea de que la propuesta de Prakken es robusta, ya que puede ser empleada tambi3n con DeLP. La diferencia entre DeLP y la programaci3n l3gica extendida con prioridades rebatibles, usada en la propuesta original, es que el primero ha sido implementado, por lo que es posible que los di3logos se ejecuten en el DeLP server.

La sugerencia planteada en este trabajo requiere una discusi3n m3s profunda y t3cnica que se realizar3 en futuros trabajos.

Los autores agradecen las observaciones realizadas por los evaluadores.

## Referencias

1. Austin, J.L.: *How to Do Things with Words*. Harvard University Press, Cambridge, MA, Cambridge, MA (1962)
2. Garc3a, A.J.: *Defeasible Logic Programming: Definition, Operational Semantics and Parallelism*. Ph.D. thesis, Universidad Nacional del Sur, Bah3a (2000)
3. Garc3a, A.J., Simari, G.R.: *Defeasible logic programming: An argumentative approach*. *TPLP* 4(1-2), 95–138 (2004)
4. Loui, R.P.: *Process and policy: Resource-bounded nondemonstrative reasoning*. *Computational intelligence* 14(1), 1–38 (1998)
5. Pollock, J.: *Cognitive Carpentry: A Blueprint for how to Build a Person*. Bradford Bks, MIT Press (1995)
6. Prakken, H.: *Relating protocols for dynamic dispute with logics for defeasible argumentation*. *Synthese* 127(1-2), 187–219 (2001)
7. Prakken, H.: *Coherence and flexibility in dialogue games for argumentation*. *Journal of logic and computation* 15(6) (2005)
8. Prakken, H.: *Formal systems for persuasion dialogue*. *The knowledge engineering review* 21(02), 163–188 (2006)
9. Prakken, H.: *Models of persuasion dialogue*. In: Rahwan, I., Simari, G.R. (eds.) *Argumentation in artificial intelligence*, pp. 281–300. Springer (2009)
10. Prakken, H., Sartor, G.: *Argument-based extended logic programming with defeasible priorities*. *Journal of applied non-classical logics* 7(1-2), 25–75 (1997)
11. Prakken, H., Vreeswijk, G.: *Logics for Defeasible Argumentation*, pp. 219–318. Springer Netherlands, Dordrecht (2002)
12. Simari, G.R., Loui, R.P.: *A mathematical treatment of defeasible reasoning and its implementation*. *Artif. Intell.* 53(2-3), 125–157 (1992)
13. Teze, J.C.L.: *Formalizaci3n y Generalizaci3n del Manejo de Preferencias en Servicios de Razonamiento Rebatible*. Ph.D. thesis, U.N.S., Bah3a Blanca (2017)
14. Verharen, E.: *A Language-action Perspective on the Design of Cooperative Information Agents*. E.M. Verharen (1997)
15. Vreeswijk, G.A.: *The computational value of debate in defeasible reasoning*. *Argumentation* 9(2), 305–342 (1995)
16. Walton, D., Krabbe, E.: *Commitment in Dialog: Basic Concepts of Interpersonal Reasoning*. State University of New York Press, NJ (1995)