

# Aprendizaje de reglas para la generación de argumentos a partir de observar cómo otros agentes argumentan

Ariel Monteserin and Analía Amandi

ISISTAN, Fac. Cs. Ex., UNCPBA, Campus Universitario, Pje. Arroyo Seco, Tandil, Argentina.  
CONICET, Consejo Nacional de Investigaciones Científicas y Técnicas, Argentina. Email:  
{amontese, amandi}@exa.unicen.edu.ar.

**Resumen** Aprender cómo argumentar es una habilidad clave para un agente negociador. En este artículo, presentaremos un enfoque que le permite a un agente aprender cómo construir sus argumentos a partir de observar cómo otros agentes argumentan en un contexto de negociación. En particular, nuestro enfoque permite inferir las reglas de generación de argumentos que son utilizadas por los agentes observados. Para llevar a cabo este propósito, el agente registra los argumentos expresados por otros agentes y los hechos que componen el contexto de la negociación en el momento en el que el argumento fue expresado. Luego, utiliza un algoritmo de minería difusa de reglas de asociación generalizadas para extraer las reglas deseadas. Este tipo de algoritmo nos permite (a) obtener reglas generales, que pueden ser aplicadas en distintos contextos, y (b) lidiar con la incertidumbre de no conocer a ciencia cierta qué hechos del contexto son considerados por los agentes para la generación de sus argumentos. Los resultados experimentales demostraron que es posible inferir reglas de generación de argumentos a partir de un número reducido de argumentos observados.

## 1. Introducción

La negociación es una herramienta esencial para que agentes con objetivos conflictivos alcancen acuerdos. Distintos enfoques sobre negociación entre agentes han sido desarrollados. Uno de ellos es el enfoque de negociación basada en argumentación (ver [11,18,17,15,2]). En estos enfoques, los agentes están habilitados a intercambiar argumentos, además de la información expresada en las propuestas. Así, en el contexto de la negociación, un argumento es visto como una pieza de información que da soporte a una propuesta y permite al agente (a) justificar su posición de negociación, o (b) influenciar la posición de otro agente [8].

Cuando un conflicto surge durante la negociación, un agente debe observar el contexto de la negociación y determinar qué argumentos puede expresar para alcanzar un acuerdo. Una forma de generar estos argumentos, ampliamente difundida, es mediante la utilización de reglas explícitas ([11,17]). Estas reglas establecen un conjunto de condiciones que el contexto de la negociación debe cumplir para que se pueda generar un determinado argumento. Por ejemplo, si queremos generar una recompensa, debemos conocer que desea recibir el destinatario del argumento a cambio de nuestro pedido. En términos formales, para generar una recompensa, un agente  $a_i$ , que necesita persuadir a un oponente  $a_j$ , debe observar en el contexto de la negociación la existencia de un objetivo  $g_{j1}$  perteneciente a  $a_j$ , y de una acción  $t_A$  que facilita el cumplimiento de dicho objetivo  $g_{j1}$ . De esta forma, si el agente encuentra estos hechos en el contexto de la negociación, podrá generar una recompensa diciendo: “Si ( $a_j$ ) aceptas mi propuesta, prometo ejecutar la acción  $t_A$ ”.

En la mayoría de los *frameworks* de negociación basada en argumentación, las reglas de generación de argumentos que el agente emplea durante la negociación son definidas en tiempo de diseño, y no se definen técnicas que permitan a un agente aprender cómo construir sus argumentos a partir de su propia experiencia.

En este artículo, proponemos un enfoque que permite a un agente aprender cómo construir argumentos a partir de observar cómo otros agentes argumentan en el contexto de una negociación. Específicamente, nuestro enfoque se centra en el aprendizaje de reglas para la generación de argumentos. Para inferir estas reglas, utilizamos un algoritmo para la extracción difusa de reglas de asociación generalizadas ([7]). Con este algoritmo obtendremos reglas que asocien un conjunto de condiciones (antecedente) con un argumento (consecuente), tal es el formato de las reglas de generación de argumentos que deseamos hallar. Por un lado, este tipo de algoritmo nos permitirá obtener reglas generales, que podrán ser aplicadas en distintos dominios de negociación. Esto es posible gracias a que aprovecha la existencia de una taxonomía de condiciones y argumentos para generar reglas de asociación en diferentes niveles de dicha taxonomía, con diferentes grados de generalidad ([19]).

Por otro lado, un agente que observa a otros agentes argumentar sólo puede tener certeza de los argumentos generados, pero no puede estar seguro de las condiciones chequeadas para su generación. No obstante, el agente puede acceder a información del contexto de la negociación que podría haber formado parte de dichas condiciones. Por ejemplo, la información contextual alrededor de la recompensa ejemplificada previamente podría ser:  $a_j$  tiene los objetivos  $g_{j1}$ ,  $g_{j2}$  y  $g_{j3}$ ;  $a_k$  tiene el objetivo  $g_{k1}$ ;  $a_i$  conoce que la ejecución de  $t_A$  permite cumplir  $g_{j1}$  y  $g_{k1}$  y la ejecución de  $t_B$  facilita el cumplimiento de  $g_{j2}$ . En este contexto, no todos estos hechos forman parte necesariamente de las condiciones para la generación de la recompensa. Por esta razón, el agente debe diferenciar entre aquellos hechos que son parte de las condiciones, de aquellos que no lo son. Así, teniendo en cuenta la información que puede ser extraída de los argumentos (por ejemplo,  $a_j$  es el destinatario y  $t_A$  es la “recompensa”), podemos determinar cuán probable es que un hecho esté relacionado al argumento observado (el hecho de que  $a_i$  conoce que la ejecución de  $t_A$  permite cumplir  $g_{j1}$  está más relacionado al argumento que el hecho de que  $a_i$  sabe que ejecutando  $t_A$  facilita el cumplimiento de  $g_{k1}$ , dado que  $t_A$  es la acción prometida y  $g_{j1}$  es un objetivo de  $a_j$ , destinatario del argumento).

Por este contexto es que proponemos la utilización de un enfoque difuso para la extracción de reglas de asociación generalizada. La minería de reglas de asociación difusas es el descubrimiento de reglas de asociación utilizando el concepto de conjunto difuso (*fuzzy set*, [12]). En el contexto de nuestro trabajo, vemos al conjunto de condiciones observadas como un conjunto difuso, en donde cada condición tiene un grado de membresía de acuerdo a la relación presumida entre la condición y el argumento. Consecuentemente, la incertidumbre asociada a las condiciones es tenida en cuenta durante el proceso de minería de datos.

El resto del artículo es organizado de la siguiente forma. La Sección 2 presenta conceptos básicos sobre negociación basada en argumentación. En la Sección 3 presentamos el enfoque de aprendizaje de reglas de generación de argumento a partir de la observación de otros agentes. A continuación, en la Sección 4 se presentan los resultados extraídos de la experimentación realizada en un sistema multiagente en un dominio de agenda de reuniones. Por último, la Sección 5 presenta las conclusiones del presente trabajo y los trabajos futuros que se desprenden de éste.

## 2. Negociación basada en argumentación

Existen dos grandes tendencias en la literatura existente sobre negociación basada en argumentación ([16]): (a) enfoques para adaptar lógicas dialécticas para argumentación rebatible

embebido conceptos de negociación dentro de ellas ([14,3]); (b) enfoques que extienden frameworks de negociación para que permitan a los agentes intercambiar argumentos retóricos, como recompensas y amenazas ([11,17]). En esta última categoría se encuadra el enfoque presentado en este artículo.

La variedad de argumentos que los agentes pueden intercambiar durante la negociación depende tanto de los distintos tipos de argumentos como de la forma en que estos tipos pueden ser construidos. Dichos tipos son los comúnmente estudiados en el campo de la persuasión en negociación entre seres humanos ([10,13]). En base a estos estudios, en la literatura actual se definen seis tipos de argumentos que un agente puede utilizar durante la negociación ([11,17,4]). Estos argumentos son: amenazas (advierten sobre una consecuencia negativa), recompensas (prometen un premio) y apelaciones (justifican una propuesta); y de este último se pueden desprender cuatro subtipos de apelaciones: de recompensa pasada (recuerdan una recompensa realizada con anterioridad), contraejemplo (muestran un contraejemplo que respalda una posición de negociación), de práctica predominante (señalan situaciones similares en las que las mismas propuestas han sido aceptadas), y de interés propio (indican un beneficio propio si se acepta una propuesta). En términos generales, un argumento retórico está conformado por cuatro elementos: el emisor, el receptor, una conclusión que normalmente representa la propuesta que se está queriendo soportar, y un conjunto de premisas que sirven de justificación ([18]).

En los enfoques de negociación basados en argumentación, los agentes pueden intercambiar argumentos con el objeto de justificar sus propuestas, persuadir a sus oponentes, y alcanzar, de este modo, un acuerdo esperado. En contraste con los agentes que carecen de esta habilidad argumentativa, un agente argumentador, además de evaluar y generar propuestas, deberá ser capaz de (a) evaluar argumentos recibidos y actualizar su estado mental como resultado de tal evaluación, (b) generar argumentos candidatos para ser presentados a un oponente, y (c) seleccionar cuál de ellos debe ser presentado en primer lugar [5].

La generación de argumento está relacionada con la construcción de argumentos candidatos que podrían ser utilizados en una determinada situación para persuadir a un oponente. Una de las técnicas desarrolladas para este fin, es la definición de reglas para la generación de argumentos ([11,17]). Dichas reglas especifican las condiciones necesarias que se deberán satisfacer para que un argumento pueda ser generado. De esta forma, cuando tales condiciones son satisfechas por el contexto de negociación el argumento asociado a la regla se convierte en un argumento candidato. Normalmente estas reglas son definidas explícitamente, no obstante, es posible aprenderlas a partir de observar cómo otros agentes argumentan.

Para formalizar los términos que se utilizarán en el resto del artículo, a continuación describiremos el lenguaje de negociación utilizado durante los experimentos y las reglas estándares de generación de argumentos definidas en [11] que se utilizarán como ejemplo.

## 2.1. Lenguaje de negociación

El lenguaje de negociación  $L$  utilizado por los agentes para el intercambio de propuestas y argumentos durante la negociación está compuesto por los siguientes predicados:

- $goal(G)$ :  $G$  es un objetivo.
- $hasgoal(A, goal(G))$ : el agente  $A$  tiene el objetivo  $G$ .
- $believe(A, B)$ : el usuario  $A$  cree  $B$ , en otras palabras  $A$  tiene  $B$  entre sus creencias.
- $prefer(A, goal(G1), goal(G2))$ : el agente  $A$  prefiere cumplir el objetivo  $G_1$  en lugar de cumplir el objetivo  $G_2$ .
- $accept(P)$ : representa la aceptación de la propuesta  $P$ .
- $reject(P)$ : representa el rechazo de la propuesta  $P$ .
- $imply(Q, R)$ :  $Q$  implica  $R$ , representa la inferencia clásica.

- $pastpromise(A_i, A_j, P)$ : el usuario  $A_i$  ha prometido al usuario  $A_j$  cumplir la propuesta  $P$ .
- $do(A, Action)$ : representa la ejecución de la acción  $Action$  por parte de  $A$ .  $Action$  puede ser instanciada con  $accept(P)$  o  $reject(P)$ .
- $wasgoal(A, goal(G))$ :  $A$  persiguió el objetivo  $G$  en el pasado.
- $did(A, Action)$ :  $A$  ejecutó la acción  $Action$  anteriormente.
- $appeal(A_i, A_j, do(A_j, Action), [Just])$ : representa una apelación expresada por el agente  $A_i$  al agente  $A_j$ . El objetivo del argumento es dar soporte a una propuesta  $do(A_j, Action)$ , utilizando la lista de justificaciones  $Just$ .
- $reward(A_i, A_j, do(A_j, Action_h), [do(A_i, Action_k)])$ : representa una recompensa expresada por el agente  $A_i$  al agente  $A_j$ .  $A_i$  promete la ejecución de  $Action_k$  a cambio de que  $A_j$  ejecute  $Action_h$ .
- $threat(A_i, A_j, do(A_j, Action_h), [do(A_i, Action_k)])$ : representa una amenaza o advertencia expresada por el agente  $A_i$  al agente  $A_j$ .  $A_i$  advierte que ejecutará  $Action_k$  si  $A_j$  no ejecuta  $Action_h$ .

Adicionalmente, pueden existir otras proposiciones, fuertemente relacionadas al dominio de la negociación, que extienden el lenguaje  $L$ , especialmente aquellas relacionadas a objetivos, propuestas y acciones que los agentes pueden realizar. Por ejemplo, en el dominio de una agenda de reuniones las proposiciones extras son:  $discusstopic(T)$ ,  $T$  es un tema que puede ser discutido en la reunión;  $inplace(P)$ , la reunión puede ser concertada en el lugar  $P$ ;  $date(D)$ , la reunión puede ser en el día  $D$ ; y  $time(S)$ , la reunión puede ser en el horario  $S$ . De esta forma,  $hasgoal(a_1, goal(discusstopic(topic_1)))$  representa el objetivo de  $a_1$  de discutir el tema  $topic_1$  en una reunión.

## 2.2. Reglas para la generación de argumentos

Como hemos comentado previamente, es posible obtener distintos tipos de apelaciones variando la forma en que éstas son justificadas. Estos tipos son: apelaciones de promesa pasada, contraejemplo, de práctica prevaleciente y de interés propio. Las reglas para su generación son:

- Apelación de promesa pasada:
  - Condiciones:  $pastpromise(Y, X, do(Y, Action))$
  - Argumento:  $appeal(X, Y, do(Y, Action), [pastpromise(Y, X, do(Y, Action))])$
- Apelación de interés propio:
  - Condiciones:  $hasgoal(Y, goal(Goal)), believe(X, imply(do(Y, Action), Goal))$
  - Argumento:  $appeal(X, Y, do(Y, Action), [imply(do(Y, Action), Goal), hasgoal(Y, goal(Goal))])$
- Apelación de práctica predominante:
  - Condiciones:  $hasgoal(Y, goal(Goal)), believe(Y, imply(do(Y, Action), not(Goal))), wasgoal(Z, goal(Goal)), did(Z, Action)$
  - Argumento:  $appeal(X, Y, do(Y, Action), [wasgoal(Z, goal(Goal)), did(Z, Action)])$
- Contraejemplo:
  - Condiciones:  $hasgoal(Y, goal(Goal)), believe(Y, imply(do(Y, Action), not(Goal))), believe(Y, imply(do(Y, ActionB), not(Goal))), wasgoal(Y, goal(Goal)), did(Y, ActionB)$
  - Argumento:  $appeal(X, Y, do(Y, Action), [did(Y, ActionB), imply(do(Y, ActionB), not(Goal))])$

Adicionalmente, definimos dos reglas ejemplo para generar recompensas y amenazas.

- Recompensa (observando también los propios objetivos del agente emisor):

- Condiciones:  $hasgoal(Y, goal(Goal)), believe(Y, imply(do(X, ActionR), Goal)), hasgoal(X, goal(Goal2)), believe(X, imply(do(Y, ActionP), Goal2))$
- Argumento:  $reward(X, Y, do(Y, ActionP), [do(X, ActionR)])$
- Amenazas:
  - Condiciones:  $hasgoal(Y, goal(GoalA)), hasgoal(Y, goal(GoalB)), prefer(Y, goal(GoalA), goal(GoalB)), believe(X, imply(do(X, ActionT), not(GoalA))), believe(X, imply(do(Y, ActionP), not(GoalB)))$
  - Argumentos:  $threat(X, Y, do(Y, ActionP), [do(X, ActionT)])$

Es importante remarcar que sólo hemos mostrado acciones para los principales argumentos a modo de ejemplo, no obstante, otras reglas pueden ser definidas variando las condiciones.

### 3. Aprendizaje de reglas de generación de argumentos

Como hemos introducido anteriormente, para el aprendizaje de reglas de generación de argumentos utilizaremos un algoritmo de extracción difusa de reglas de asociación generalizadas. Este algoritmo nos permite obtener reglas con diferentes grados de generalidad que podrán ser aplicadas por el agente en distintas situaciones de negociación, y adicionalmente, nos permite considerar la falta de certeza sobre los hechos que compondrán dichas reglas. En esta sección, presentaremos, en primer lugar, una breve descripción del funcionamiento de este tipo de algoritmos, y luego detallaremos cómo inferir reglas de generación de argumentos desde un conjunto de argumentos observados.

#### 3.1. Algoritmo de reglas de asociación generalizadas difusas

La minería de reglas de asociación difusas es el descubrimiento de reglas de asociación utilizando el concepto de conjunto difuso (*fuzzy set*, [12]). La teoría de conjuntos difusos ([20]) ha sido utilizada más y más frecuentemente en sistemas inteligentes gracias a su simplicidad y similitud con el razonamiento humano ([9]). Los conjuntos difusos son conjuntos cuyos elementos tienen cierto grado de membresía. En el contexto de nuestro trabajo, vemos al conjunto de condiciones observadas como un conjunto difuso, en donde cada condición tiene un grado de membresía de acuerdo a la relación presumida entre la condición y el argumento. A continuación, describiremos brevemente los conceptos básicos de la minería de reglas de asociación tradicional, generalizadas y difusas.

**Minería de reglas de asociación tradicional.** Como se introdujo en [1], dado un conjunto de transacciones  $T$ , donde cada transacción es un conjunto de ítems pertenecientes a  $I$ , una regla de asociación es una expresión  $X \rightarrow Y$ , donde  $X$  e  $Y$  son también conjuntos de ítems. Esto significa que las transacciones en la base de datos que contienen los ítems de  $X$  también contendrán los ítems de  $Y$ . Esto es asegurado computando el soporte y la confianza de la regla. El soporte de una regla  $X \rightarrow Y$  es la tasa de transacciones ( $T$ ) que contienen  $X \cup Y$  sobre el total de transacciones en la base de datos ( $|D|$ ). Mientras que la confianza es la tasa de registros que contienen  $X \cup Y$  sobre el número de registros que contienen  $X$ .

**Minería de reglas de asociación generalizadas.** Dado que necesitamos que las reglas de generación de argumentos posean un formato variable, no es suficiente la utilización de un algoritmo tradicional de reglas de asociación, debido a que los argumentos y hechos sobre los cuales deberá trabajar se encuentran expresados en términos constantes.

Para resolver este problema, empleamos un algoritmo de reglas de asociación generalizadas. Estos algoritmos aprovechan la existencia de una taxonomía jerárquica de datos para generar reglas de asociación a diferentes niveles de la taxonomía ([19]). Una regla de asociación generalizada  $X \rightarrow Y$  es definida idénticamente a una regla de asociación regular, excepto que ningún ítem de  $Y$  puede ser ancestro<sup>1</sup> de un ítem de  $X$ . Siguiendo esta idea, nuestro objetivo es construir una taxonomía de condiciones y argumentos en cuyas hojas estén los hechos del contexto de la negociación (condiciones) y argumentos observados por el agente, y en los niveles superiores se encuentren las mismas proposiciones pero expresadas en términos variables, con diferentes grados de generalidad. Entonces, un algoritmo de extracción de reglas de asociación generalizadas será capaz de generar, especialmente, reglas en los niveles superiores de la taxonomía. De este modo, las reglas serán variables.

Un enfoque simple para realizar la minería de reglas de asociación generalizadas es tomar cada transacción y expandir cada ítem incluyendo todos los ítems sobre él en la jerarquía ([19]). Esto es, agregar todos los ancestros de cada ítem de una transacción a la misma transacción. Como es de esperar, cuando las reglas son generadas para los ítems en los niveles superiores de la taxonomía el soporte y la confianza se incrementan. Esto es un aspecto deseable dado que los algoritmos de reglas de asociación buscan reglas con valores de soporte y confianza superiores a los mínimos establecidos.

**Minería difusa de reglas de asociación generalizadas.** El concepto de minería difusa de reglas de asociación nace y está asociado a la resolución del problema de la frontera en atributos cuantitativos ([12]). Según el enfoque tradicional, cuando un atributo es cuantitativo (por ejemplo, la edad) se lo divide en intervalos (niño, de 0 a 15; joven, de 16 a 25; adulto, de 26 en adelante). Sin embargo, cuando un valor de un atributo cae cerca de la frontera que divide los intervalos es difícil asegurar que ese valor pertenece exclusivamente a uno de los intervalos y no a otro (¿una persona de 26 es adulto en el mismo grado que una que 50?, ¿y no una de 25?). Para solucionar este problema, se reemplaza los intervalos fijos por los conjuntos difusos. En la teoría de conjuntos difusos, un elemento puede pertenecer a un conjunto con un valor de membresía en  $[0,1]$ . Este valor es asignado por medio de una función de membresía asociada con cada conjunto difuso. Para un atributo  $x$  y su dominio  $D_x$ , el mapeo de la función de membresía es  $m_{f_x}(x) : D_x \rightarrow [0, 1]$ .

Dada una base transaccional  $D = \{t_1, \dots, t_n\}$  y un conjunto de atributos o ítems  $I = \{I_1, \dots, I_m\}$  presentes en las transacciones almacenadas en  $D$ , cada ítem  $I_j$  tendrá asociado una serie de regiones difusas  $F_{I_j} = \{f_{i_j^1}, \dots, f_{i_j^l}\}$ . Entonces, un algoritmo de reglas de asociación difusas será capaz de encontrar reglas de la siguiente forma:  $\langle X, A \rangle \rightarrow \langle Y, B \rangle$ , en donde  $X$  e  $Y$  son conjuntos de atributos, y  $A$  y  $B$  contienen las regiones difusas asociadas con los correspondientes atributos de  $X$  e  $Y$ .

Existen distintos algoritmos para la extracción difusa de reglas de asociación ([12,6]), no obstante utilizaremos en esta propuesta el algoritmo descrito por Hong et al. [7], dado que este algoritmo integra tanto la minería de reglas de asociación generalizadas como difusas.

### 3.2. Extracción difusa de reglas para la generación de argumentos

Mientras un agente interactúa en un sistema multiagente, puede observar los argumentos que otros agentes generan durante una negociación. Adicionalmente, puede observar el contexto de la negociación en la cual un argumento es generado y registrar los hechos que en ese instante lo conforman. Siguiendo esta idea, almacenaremos en una base  $O$  las observaciones realizadas por el agente. Cada observación en  $O$  es una tupla con el siguiente formato:  $(H, a)$ , donde  $H =$

<sup>1</sup> Un ancestro de un ítem es aquel que está sobre él en la taxonomía.

$\{h_1, \dots, h_n\}$  es el conjunto de hechos del contexto donde fue generado el argumento  $a$ . Teniendo en cuenta estas observaciones, queremos encontrar las relaciones entre los hechos observados y los argumentos generados, dado que estas relaciones constituyen las reglas de generación de argumentos que deseamos aprender. Para llevar a cabo esta tarea, los pasos a seguir son:

1. Definición de las transacciones a partir de las observaciones almacenadas.
2. Construcción de una taxonomía de condiciones y argumentos.
3. Ejecución del algoritmo de reglas de asociación generalizadas difusas.
4. Post-procesamiento de las reglas obtenidas.

A continuación, detallaremos estos pasos.

**Definición de las transacciones.** En base a las observaciones almacenadas en  $O$ , debemos definir las transacciones bajo las cuales el algoritmo de reglas de asociación trabajará. Como introdujimos previamente, las observaciones son tuplas con el formato  $(H, a)$ . Dado que el argumento  $a$  fue generado utilizando una regla  $C \rightarrow a$ , donde  $C$  es el conjunto de condiciones que debieron ser cumplidas por los hechos del contexto, por ende, es correcto pensar que este conjunto de condiciones  $C$  es un subconjunto de  $H$ . Si bien el agente no puede tener certeza de los elementos que componen el subconjunto  $C$ , puede definir una función  $m_a(h_i) : H \rightarrow [0, 1]$  que determine el grado de relación  $m_{ai}$  entre el argumento  $a$  y cada  $h_i \in H$ . Por lo tanto, podemos definir a  $C$  como un conjunto difuso en donde  $m_{ai}$  determina el grado de membresía de cada hecho  $h_i$  al conjunto  $C$ . Vale la pena resaltar que cada ítem en  $H$  tendrá asociado una única región difusa, que determinará el grado de inclusión del ítem en las condiciones de una regla de generación de argumentos.

Siguiendo esta definición de  $C$ , para cada observación  $\sigma^j = (\{h_1^j, \dots, h_n^j\}, a^j)$  perteneciente a  $O$  definimos una transacción  $t^j = (h_1^j, \dots, h_n^j, a^j)$  con  $m_a^j(h_i^j) : H^j \rightarrow [0, 1]$  como la función de membresía que asocia a cada ítem de la transacción con la región difusa de inclusión en  $C$ .

La función de membresía  $m_a$  puede ser definida de distintas formas. Para este trabajo hemos optado por un enfoque trivial: definirla mediante un conjunto de reglas que determinan la relación entre la conclusión y las premisas que pueden ser extraídas de los argumentos generados y cada hecho del contexto, en base al significado semántico de los mismos. Por ejemplo, dada una recompensa  $reward(a1, a2, do(a2, accept(discusstopic(topic1))), [do(a1, accept(discusstopic(topic2))])]$ , podemos extraer la conclusión  $do(a2, accept(discusstopic(topic1)))$  y la premisa  $do(a1, accept(discusstopic(topic2)))$ . En base a estas proposiciones podemos determinar que el hecho  $believe(a2, do(a2, accept(discusstopic(topic1))), discusstopic(topic1))$  observado en el contexto puede estar relacionado a la conclusión y que el hecho  $believe(a1, do(a1, accept(discusstopic(topic2))), discusstopic(topic2))$  puede estar relacionado a la premisa. Por lo tanto, podemos suponer un valor de 0.8 para representar la posibilidad de que estos hechos formen parte de las condiciones de una regla de generación de argumentos. Consecuentemente, dado el hecho  $believe(a2, do(a2, accept(discusstopic(topic1))), discusstopic(topic1))$  podemos suponer, por transitividad, que un hecho del tipo  $hasgoal(a2, goal(discusstopic(topic2)))$  también puede ser parte de las condiciones de la regla, por lo cual podemos asignarle un valor de 0.6. Asimismo, a partir de  $hasgoal(a2, goal(discusstopic(topic2)))$  podemos identificar un hecho  $prefer(a2, goal(discusstopic(topic2)), goal(discusstopic(topic4)))$  con una probabilidad de 0.5. Por el contrario, en otras situaciones, podemos tener certeza de que un hecho es parte del conjunto de condiciones, por ejemplo, como puede ocurrir con una apelación de promesas pasadas, en cuya premisa podemos encontrar un hecho  $pastpromise$ , el cual debe hallarse inequívocamente en el contexto de la negociación.

**Construcción de la taxonomía.** Para ejecutar el algoritmo de reglas de asociación generalizadas es necesario contar con una taxonomía de los ítems que componen las transacciones. En nuestro trabajo, estos ítems son las condiciones probables y los argumentos observados por el agente.

Para construir la taxonomía, en primer lugar, se comienza poniendo en las hojas las condiciones y argumentos presentes en las transacciones tal como fueron almacenados por el agente (p.e.  $reward(a1, a2, do(a2, accept(discusstopic(topic1))), [do(a1, accept(discusstopic(topic2))])])$  y  $believe(a2, do(a2, accept(discusstopic(topic1))), discusstopic(topic1))$ ), una condición o argumento por cada hoja. Esto es, por cada ítem (condiciones o argumentos) de cada transacción, construimos una rama de la taxonomía comenzando en este ítem (hoja) y finalizando en la raíz de la taxonomía (ver Figura 1).



**Figura 1.** Parte de una taxonomía de condiciones y argumentos.

Para construir esta rama, tomamos un ítem y generamos todos los ancestros que representan la misma condición o argumento pero reemplazando cada término terminal (proposición de  $L$  que no tiene otra proposición como parámetro que puede ser generalizada) por el término respectivo más general. Para determinar esto, mantenemos una estructura de datos (hash table)  $HT$  con las proposiciones y sus correspondientes formas generales. Por ejemplo, para la proposición  $hasgoal$  la forma más general almacenada en  $HT$  es  $hasgoal(Agent, goal(Goal))$ ; para el agente  $a1$ ,  $Agent$ ; para  $goal$ ,  $goal(Goal)$ ; para  $discusstopic$ ,  $discusstopic(Topic)$ , entre otros.

Entonces, dada la condición  $hasgoal(a1, goal(discusstopic(topic1)))$ , agregamos una hoja cuyo dato es dicha condición y, teniendo en cuenta que sus términos terminales son  $a1$  y  $discusstopic(topic1)$ , creamos los siguientes ancestros  $anc1$ :  $hasgoal(Agent, goal(discusstopic(topic1)))$ , reemplazando la proposición  $a1$  con  $Agent$ , donde  $Agent$  es la forma más general de  $a1$ ; y  $anc2$ :  $hasgoal(a1, goal(discusstopic(Topic)))$ , reemplazando la proposición  $discusstopic(topic1)$  con  $discusstopic(Topic)$ , donde  $discusstopic(Topic)$  es la forma más general de  $discusstopic(topic1)$ .

Luego, ejecutamos sucesivamente la misma acción, pero con cada ancestro, y creamos un nuevo nodo en la taxonomía que representa el ítem, cuyo padre son los ancestros generados previamente. Siguiendo con el ejemplo, el nuevo ancestro de  $anc1$  es  $hasgoal(Agent, goal(discusstopic(Topic)))$  (el mismo que para  $anc2$ ); reemplazando  $goal(discusstopic(Topic))$  con  $goal(Goal)$  y obtenemos la expresión más general de la condición inicial. Cuando la expresión más general es encontrada, un nuevo nodo es creado en la taxonomía cuyo padre es la raíz. En la Figura 1, podemos observar un ejemplo de esta parte de la taxonomía.

**Ejecución del algoritmo de reglas de asociación generalizadas difusas.** Luego de construir la taxonomía, recorreremos cada transacción  $t_i \in T$  y reemplazamos cada ítem  $i_{ij} \in t_i$  por su

ancestro más cercano que no posea constantes. De esta manera, eliminamos cualquier posible regla constante, ya que nos interesan, como dijimos anteriormente, aquellas reglas que pueden ser completamente instanciadas en un nuevo contexto de negociación. Por ejemplo, el ítem  $hasgoal(a1, goal(discusstopic(topic1)))$  será reemplazado por el ancestro  $hasgoal( Agent, goal(discusstopic(Topic)))$ . Adicionalmente, los términos variables son numerados correlativamente para conservar la referencia entre los distintos ítems de la transacción. Por lo tanto, la versión final del ítem utilizado como ejemplo será:  $hasgoal(Agent0, goal(discusstopic(Topic0)))$ , en consecuencia, cada vez que reemplacemos  $a1$  en un ítem de la misma transacción será cambiado por  $Agent0$ .

El conjunto de transacciones que pueden ser obtenidas luego de este pre-procesamiento forman los datos de entrada del algoritmo de reglas de asociación generalizadas difusas.

**Post-procesamiento de las reglas obtenidas.** El post-procesamiento de las reglas de asociación puede ser dividido en tres partes. Primero, filtramos las reglas cuyo formato no se ajusta al formato  $\{c_i, \dots, c_p\} \rightarrow a_j$ , es decir, una vez que las reglas han sido obtenidas, seleccionamos sólo las reglas cuyo antecedente está compuesto por condiciones y cuyo consecuente está compuesto por un único argumento, el resto son descartadas.

La segunda parte del post-procesamiento consiste en determinar cuan representativa es la regla con respecto a los argumentos generados por el usuario y recolectados en las observaciones  $O$ . Para desarrollar esta tarea, definimos una métrica de suficiencia de una regla de asociación. Esta métrica representa la relación entre las condiciones de las transacciones (observaciones) que soportan la regla y las condiciones de la regla. Es calculada como el porcentaje entre el número total de condiciones de una regla sobre el promedio de las condiciones de las transacciones que la soportan:

$$Suficiencia(r) = \frac{condicionesTotales(r)}{condicionesPromedio(transaccionesSoportando(r))}$$

Por ejemplo, si tenemos las transacciones  $t_1 = (c_1, c_3, c_5, a_1)$ ,  $t_2 = (c_1, c_2, c_4, a_1)$ ,  $t_3 = (c_1, c_4, a_1)$ , y  $t_4 = (c_1, c_5, a_2)$ ; y definimos un soporte mínimo de 0,5 y una confianza mínima de 0,75, obtendremos luego de la primera etapa de post-procesamiento, las reglas  $r_1 : c_1 \rightarrow a_1$ ,  $r_2 : c_4 \rightarrow a_1$  y  $r_3 : c_1, c_4 \rightarrow a_1$ . Las tres reglas tienen soporte y confianza mínimos. Sin embargo, podemos ver que las reglas  $r_1$  y  $r_2$  no son suficientemente representativas con respecto a las transacciones  $t_1$ ,  $t_2$  y  $t_3$ , ya que es poco probable que las condiciones figuren en forma aislada dentro de las transacciones. Las reglas  $r_1$  y  $r_2$  poseen una suficiencia de 0.375 y 0.4 respectivamente, mientras que para la regla  $r_3$ , la suficiencia es 0.8. La métrica de suficiencia apunta a filtrar este tipo de reglas determinando un umbral que fije cuan suficientes deben ser las condiciones (antecedente) de una regla para generar el argumento consecuente, independientemente de los valores de soporte y confianza. Por lo tanto, un umbral del 75 % nos permitiría quedarnos con la regla  $r_3$ .

Por último, dado que el algoritmo de reglas de asociación generalizadas puede encontrar reglas en distintos niveles de la taxonomía, es posible encontrar reglas que son ancestros (considerando la taxonomía) de otras reglas. Ante este hecho decidimos conservar las reglas con mayor grado de generalidad dado que serán las que podamos aplicar en distintos dominios de negociación.

#### 4. Resultados experimentales

El dominio elegido para evaluar el enfoque propuesto fue una aplicación multiagente de agenda de reuniones. En esta aplicación, los agentes deben concertar reuniones discutiendo fecha, horario, lugar, temas a tratar, y participantes de acuerdo a las preferencias y objetivos de los

usuarios que representan. Como los usuarios poseen diferentes objetivos, los agentes deberán intercambiar argumentos con el objetivo de alcanzar acuerdos.

El objetivo de los experimentos realizados fue determinar que reglas de generación de argumentos utilizan los agentes participantes de las negociaciones utilizando el enfoque de aprendizaje propuesto.

#### 4.1. Escenario de los experimentos

Los experimentos fueron realizados con cuatro agentes, representando cada uno a un usuario distinto. Para cada agente se definió un conjunto de objetivos, preferencias entre objetivos, creencias y hechos históricos (promesas pasadas, información sobre otras reuniones realizadas, objetivos cumplidos) y el conjunto de reglas para generación de argumentos definido en la Sección 2.2. En base a esos objetivos y creencias, los agentes tuvieron que generar argumentos para persuadir a sus oponentes utilizando dichas reglas. Luego de finalizadas las negociaciones, los argumentos y los hechos del contexto (objetivos, preferencias, creencias y hechos históricos) fueron almacenados en la base de observaciones  $O$  para su procesamiento.

#### 4.2. Resultados obtenidos

En total fueron registrados 69 argumentos y 478 hechos del contexto. En el Cuadro 1 puede observarse la cantidad de argumentos observados para cada tipo de argumento (columna *Argumentos*). Para cada argumento se determinó el conjunto difuso de condiciones probables en base a la función  $m_a$ , y se definió una transacción que fue incluida en la base de transacciones  $T$ . Luego, a partir de las transacciones se construyó una taxonomía siguiendo los pasos descritos en la Sección 3.2. Con las transacciones definidas y la taxonomía construida se procedió a generar reglas de asociación generalizadas difusas utilizando el algoritmo descrito en [7]. El Cuadro 1 muestra para cada tipo de argumento<sup>2</sup> la cantidad de reglas obtenidas totales (columna *Reglas*), la cantidad de reglas que respetan el formato  $\{c_i, \dots, c_p\} \rightarrow a_j$  (columna *Formato*), la cantidad de reglas filtradas por especificidad (columna *Especificidad*) y finalmente, la cantidad de reglas generales halladas por el algoritmo (columna *Generales*).

Para determinar la eficiencia del enfoque de aprendizaje comparamos las reglas obtenidas, filtradas por formato, especificidad y generalidad, con las reglas originales de generación de argumentos, utilizadas por los agentes para construir sus argumentos. Esta comparación se realizó cotejando la cantidad y formato de las condiciones y el formato del argumento consecuente. Como resultado de esta comparación determinamos cuatro categorías de las reglas encontradas: *Correcta*, cuando la regla obtenida es exactamente igual a la regla utilizada por el agente; *Parcial*, cuando la regla obtenida tiene un número menor de condiciones que la regla original; *Mayor*: cuando la regla obtenida posee condiciones adicionales además de todas las condiciones consideradas por la regla original; e *Incorrecta*: cuando la regla obtenida no guarda relación con la regla utilizada por el agente.

En el Cuadro 2 podemos apreciar los resultados de dicha comparación. Como podemos ver, 6 de las 18 reglas obtenidas correspondieron exactamente con la descripción de las reglas de generación de argumentos originales (33.3%), 8 (44.5%) representaron parciales de dichas reglas, 4 (22.2%) fueron reglas con mayor número de condiciones y ninguna regla incorrecta.

Para ilustrar los resultados obtenidos, a continuación enumeraremos las reglas obtenidas para la generación de apelaciones de interés propio:

<sup>2</sup> Nótese que cada tipo corresponde a una regla de generación de argumentos, tal como se definió en la Sección 2.2.

Tipo	Argumentos	Reglas	Formato	Especificidad	Generales
Recompensa	20	626	156	39	5
Amenaza	5	552	160	16	1
Práctica predominante	8	2341	621	64	1
Interés propio	18	16	7	4	4
Promesa pasada	12	11	4	4	1
Contraejemplo	6	2197	509	71	6

**Cuadro 1.** Argumentos observados y reglas obtenidas.

Tipo	Correctas	Parciales	Mayores	Incorrectas	Totales
Recompensa	1	3	1	-	5
Amenaza	1	-	-	-	1
Práctica predominante	-	-	1	-	1
Interés propio	1	2	1	-	4
Promesa pasada	1	-	-	-	1
Contraejemplo	2	3	1	-	6
<b>Totales</b>	<b>6</b>	<b>8</b>	<b>4</b>	<b>0</b>	<b>18</b>

**Cuadro 2.** Comparación de las reglas de generación de argumentos obtenidas y las utilizadas por los agentes.

- Regla 1:  $believe(AGENT1, imply(do(AGENT0, ACTION0), GOAL0)); believe(AGENT0, imply(do(AGENT0, ACTION0), GOAL0)) \rightarrow appeal(AGENT1, AGENT0, do(AGENT0, ACTION0), [imply(do(AGENT0, ACTION0), GOAL0), hasgoal(AGENT0, GOAL0)])$
- Regla 2:  $hasgoal(AGENT0, goal(GOAL0)); believe(AGENT0, imply(do(AGENT0, ACTION0), GOAL0)) \rightarrow appeal(AGENT1, AGENT0, do(AGENT0, ACTION0), [imply(do(AGENT0, ACTION0), GOAL0), hasgoal(AGENT0, GOAL0)])$
- Regla 3:  $hasgoal(AGENT0, goal(GOAL0)); believe(AGENT1, imply(do(AGENT0, ACTION0), GOAL0)) \rightarrow appeal(AGENT1, AGENT0, do(AGENT0, ACTION0), [imply(do(AGENT0, ACTION0), GOAL0), hasgoal(AGENT0, GOAL0)])$
- Regla 4:  $hasgoal(AGENT0, goal(GOAL0)); believe(AGENT1, imply(do(AGENT0, ACTION0), GOAL0)); believe(AGENT0, imply(do(AGENT0, ACTION0), GOAL0)) \rightarrow appeal(AGENT1, AGENT0, do(AGENT0, ACTION0), [imply(do(AGENT0, ACTION0), GOAL0), hasgoal(AGENT0, GOAL0)])$

De estas cuatro reglas, la regla 3 es la que concuerda correctamente con la regla de generación de argumentos definida en la Sección 2.2 y que fue utilizada por los agentes. Del resto, las reglas 1 y 2 poseen uno de las condiciones incluidas en la regla adicional por lo cual decimos que es parcial. En el caso de la regla 2, esta parcialidad no es tan marcada dado que existe una condición equivalente a la faltante, la creencia  $believe(AGENT0, imply(do(AGENT0, ACTION0), GOAL0))$ . En cuanto a la regla 4, posee las dos condiciones definidas por la regla original, y además agrega esta tercera creencia que, aunque adicional, no es semánticamente incorrecta, sino restrictiva.

## 5. Conclusiones y trabajos futuros

En este trabajo hemos presentado un enfoque que da a un agente la habilidad de inferir las reglas de generación de argumentos que otros agentes utilizan, a partir de la observación durante

la negociación, de los argumentos que con ellas generan. Nuestro enfoque utiliza un algoritmo de extracción difusa de reglas de asociación generalizadas para llevar a cabo dicha tarea. Este tipo de algoritmos nos permite descubrir reglas del tipo  $\{c_1, \dots, c_n\} \rightarrow a$  en donde  $\{c_1, \dots, c_n\}$  representa el conjunto de condiciones necesarias para generar el argumento  $a$ . Asimismo, nos permite obtener reglas generales, gracias a que genera reglas en los niveles superiores de una taxonomía de condiciones y argumentos. Por otra parte, la utilización de conjuntos difusos nos permite manejar la incertidumbre producto de no saber a ciencia cierta que hechos del contexto de la negociación son tenidos en cuenta por los agentes observados, a la hora de generar los argumentos.

Los resultados obtenidos muestran que es posible aprender las reglas de generación de argumentos utilizadas por los agentes observados a partir de una reducida cantidad de argumentos y hechos registrados. Como pudimos ver en la sección anterior, las reglas obtenidas han demostrado ser exactas en un 33.3% de los casos y más restringidas con respecto a las originales en un 22,2%, mientras que un 44.5% fueron obtenidas reglas parciales, que aunque no son exactas, no dejan de ser válidas. Adicionalmente, es importante destacar que 5 de las 6 reglas con las cuales se generaron los argumentos de entrenamiento fueron detectadas correctamente por nuestro enfoque.

Los trabajos futuros que se desprenden de este trabajo tienen dos direcciones. Por un lado, es interesante la idea de utilizar otros algoritmos de extracción de reglas de asociación generalizadas difusas para comparar el desempeño con los resultados obtenidos hasta aquí. Por otro lado, es promisorio el análisis de distintas definiciones de la función de membresía que relaciona los hechos del contexto con las condiciones probables para la generación de un argumento determinado. En trabajos posteriores, planeamos analizar la utilización de ontologías para determinar con mayor precisión el contexto en el cual un argumento fue generado. Por último, cabe destacar la necesidad de continuar con la experimentación en distintos dominios de negociación.

## Referencias

1. R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. In *SIGMOD '93*, pages 207–216, New York, NY, USA, 1993. ACM.
2. L. Amgoud, Y. Dimopoulos, and P. Moraitis. A unified and general framework for argumentation-based negotiation. In *AAMAS '07*, pages 1–8, New York, NY, USA, 2007. ACM.
3. L. Amgoud, S. Parsons, and N. Maudet. Arguments, dialogue, and negotiation. *Journal of Artificial Intelligence Research*, 23:2005, 2000.
4. L. Amgoud and H. Prade. Generation and evaluation of different types of arguments in negotiation. In *Proc. of the international workshop on non-monotonic reasoning*, pages 10–15, Whistler BC, Canada, 2004.
5. R. Ashri, I. Rahwan, and M. Luck. Architectures for negotiating agents. In V. Marik, J. Mueller, and M. Pechoucek, editors, *Proc. of Multi-Agent Systems and Applications III*, pages 136–146, Prague, Czech Republic, 2003. Springer.
6. M. Delgado, N. Marin, M.J. Martin-Bautista, D. Sanchez, and M.A. Vila. Mining fuzzy association rules: an overview. In *BISC International Workshop on Soft Computing for Internet and Bioinformatics*, 2003.
7. T. Hong, K. Lin, and S. Wang. Fuzzy data mining for interesting generalized association rules. *Fuzzy Sets Syst.*, 138(2):255–269, 2003.
8. N. R. Jennings, S. Parsons, P. Noriega, and C. Sierra. On argumentation-based negotiation. In *Proceedings of the International Workshop on Multi-Agent Systems (IWMAS-98)*, Boston, MA, 1998.
9. A. Kandel. *Fuzzy expert systems*. CRC Press, 1992.

10. M. Karlins and H. I. Abelson. *Persuasion: how opinions and attitudes are changed*. Springer Verlag: Berlin, 1970.
11. S. Kraus, K. Sycara, and A. Evenchik. Reaching agreements through argumentation: a logical model and implementation. *Artificial Intelligence*, 104(1-2):1–69, 1998.
12. C. M. Kuok, A. W. Fu, and Wong M. H. Mining fuzzy association rules in databases. *SIGMOD Record*, 27(1):41–46, 1998.
13. D. J. O’Keefe. *Persuasion: Theory and Research*. SAGE Publications, 1990.
14. S. Parsons, C. Sierra, and N. R. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292, 1998.
15. I. Rahwan, S. D. Ramchurn, N. R. Jennings, P. Mcburney, S. Parsons, and L. Sonenberg. Argumentation-based negotiation. *Knowledge Engineering Review*, 18(4):343–375, 2003.
16. I. Rahwan, L. Sonenberg, and P. Mcburney. Bargaining and argument-based negotiation: Some preliminary comparisons. In *Lecture Notes in Computer Science*, pages 176–191, 2005.
17. S. D. Ramchurn, N. R. Jennings, and C. Sierra. Persuasive negotiation for autonomous agents: A rhetorical approach. In *IJCAI Workshop on Computational Models of Natural Argument*, pages 9–17, 2003.
18. C. Sierra, N. R. Jennings, P. Noriega, and S. Parsons. A framework for argumentation-based negotiation. In *ATAL ’97: Proceedings of the 4th International Workshop on Intelligent Agents IV, Agent Theories, Architectures, and Languages*, pages 177–192, London, UK, 1998. Springer-Verlag.
19. R. Srikant and R. Agrawal. Mining generalized association rules. *Future Generation Computer Systems*, 13(2-3):161–180, 1997.
20. L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.