Acerca de la Implementación de las Operaciones de la Teoría de Cambio de Creencias

Marcelo A. Falappa* Eduardo L. Ferme†

Departamento de Ciencias de la Computación
Universidad Nacional del Sur
Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
ARGENTINA

ccfalapp@criba.edu.ar ferme@dc.uba.ar

Palabras Claves: Teoría de Cambio de Creencias, Modelo AGM, Representación de Conocimiento, Estados Epistémicos, Actitudes Epistémicas, Operaciones de Cambio.

Resúmen

En este artículo presentaremos ciertas consideraciones que deben ser tomadas en cuenta al momento de implementar las operaciones de cambio de la teoría de cambio de creencias. Los modelos abstractos de cambio de creencias no hacen ninguna consideración con respecto a la estructuras internas de las creencias que componen el conocimiento de un agente, y mucho menos tienen en cuenta la complejidad computacional de los algoritmos propuestos. En este artículo, presentaremos de manera explícita algunos de los inconvenientes y las posibles soluciones que pueden encontrarse al implementar las operaciones de cambio.

Primeramente, presentaremos las diferentes operaciones de cambio del modelo AGM. Estas operaciones están caracterizadas por ciertos *postulados de racionalidad* que determinan el comportamiento de las mismas en el *nivel de conocimiento*. Esto significa que las operaciones de cambio se especifican enteramente en términos del contenido del conocimiento (creencias), sin ninguna distinción entre la información explícita y la información implícita.

En segundo término, presentaremos las operaciones de cambio aplicables sobre bases de conocimiento (esto es, conjuntos finitos no necesariamente clausurados bajo consecuencia lógica) en los cuales se distinguen claramente las creencias explícitas de la base de

*Grupo de Investigación en Inteligencia Artificial (GIIA), Departamento de Ciencias de la Computación, Universidad Nacional del Sur (UNS), Argentina.

†Grupo de Investigación y Desarrollo en Inteligencia Artificial (GIDIA), Departamento de Computación, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires (UBA), Argentina.

conocimiento de las creencias que se deducen de la misma. En este caso, las operaciones de cambio se definen en un *nivel simbólico*, el cual contiene referencias explícitas sobre la sintaxis de las sentencias que componen la base. Veremos como ciertos postulados definidos en el nivel de conocimiento deben ser transformados en nuevos postulados "menos restrictivos" con el objetivo de respetar los principios básicos del modelo AGM: mínimo cambio, irrelevancia de la sintaxis y adecuacidad en la representación de conocimiento. También introducimos un *nivel meta-simbólico* donde se predica acerca del comportamiento de las sentencias que componen una base de conocimiento, pero sin ninguna referencia a su forma sintáctica.

Por último, analizaremos las operaciones de cambio en el *nivel de implementación*. En este nivel, es necesario tomar ciertas decisiones que no son consideradas no solo en el nivel de conocimiento sino también en el nivel simbólico. Este análisis es de vital importancia porque las decisiones tomadas en el nivel de implementación pueden tener un fuerte impacto en el nivel de conocimiento.

1 Introducción

La teoría de cambio de creencias pretende modelar la dinámica del conocimiento, esto es, como se actualizan las creencias después de la llegada de nueva información. Por ejemplo, si un agente cree que los pájaros vuelan y que tweety es un pájaro, puede inferir que tweety vuela. Sin embargo, si llega información externa que dice que los pájaros que son pingüinos no vuelan y que tweety es un pingüino, entonces (si el agente le da primacía a la nueva información) creerá que tweety no vuela. Este tipo de actualizaciones en el conocimiento son muy comunes en el razonamiento humano.

Entre los distintos modelos existentes para formalizar la dinámica de las creencias, el más trascendente es el modelo AGM, desarrollado por Alchourrón, Gärdenfors y Makinson en la década del 80 [AGM85]. Este modelo caracteriza ciertas operaciones de cambio (expansiones, contracciones y revisiones) que cubren gran parte de los cambios que puede realizar un agente en su conocimiento. Cada una de estas operaciones está caracterizada por una serie de postulados de racionalidad que determinan el comportamiento de las operaciones, independientemente de la manera en que se implementen. Estos postulados hacen posible ver las operaciones de cambio como "cajas negras", *i.e.*, determinando cual será el resultado de una operación de cambio, independientemente de como se "construya" esa operación.

Los postulados caracterizan las operaciones de cambio tanto sobre bases de creencias (esto es, conjuntos de sentencias cualesquiera) o sobre conjuntos de creencias (esto es, conjuntos de sentencias clausurados bajo cierta operación de consecuencia lógica). En caso de implementar operaciones de cambio, es necesario representar el conocimiento mediante bases de creencias puesto que si contamos con un lenguaje (al menos proposicional) lo suficientemente declarativo, es implausible tratar con conjuntos clausurados. La clausura lógica de la base de creencias representará el estado epistémico del agente. Sin embargo, los cambios en el estado epistémico se verán reflejados a partir de cambios en la base de conocimiento que lo genera. Esto es, los cambios en el nivel de conocimiento se verán reflejados a partir de cambios en el nivel simbólico en el cual se representa un estado epistémico.

En el presenta artículo mostraremos los distintos niveles en los cuales puede definirse un modelo de dinámica de creencias y analizaremos algunos problemas concretos de implementación.

2 Preliminares

2.1 Distintas operaciones de cambio

En el modelo AGM [AGM85] se asume que el conocimiento de un agente se representa mediante un conjunto de creencias, *i.e.*, conjuntos de sentencias clausurados mediante consecuencia lógica sobre un lenguaje $\mathcal{L}$. El operador de consecuencia Cn[1] es un mapeo de conjuntos de proposiciones a conjuntos de proposiciones, $Cn : \mathbf{2}^{\mathcal{L}} \Rightarrow \mathbf{2}^{\mathcal{L}}$ que satisface, al menos, las siguientes propiedades:[2]

i) **Inclusión**: $A \subseteq Cn(A)$.

ii) **Monotonía**: Si $A \subseteq B$ entonces $Cn(A) \subseteq Cn(B)$.

iii) **Iteración**: $Cn(A) = Cn(Cn(A))$.

Asumiremos que la noción de consecuencia lógica incluye a la consecuencia veritativo-funcional clásica y satisface las propiedades de deducción y compacidad.

iv) **Supraclasicidad**: Si α puede ser derivada de A mediante la lógica veritativo funcional clásica, entonces $\alpha \in Cn(A)$.

v) **Deducción**: $\beta \in Cn(A \cup \{\alpha\})$ si y solo si $(\alpha \rightarrow \beta) \in Cn(A)$.

vi) **Compacidad**: Si $\alpha \in Cn(A)$ entonces $\alpha \in Cn(B)$ para algún subconjunto finito $B \subseteq A$.

Dado un estado de conocimiento (representado por una base de creencias o un conjunto de creencias), en la teoría de cambio de creencias se asumen tres epistémicos básicos: aceptación, rechazo e indeterminación. Esto es, si K es un conjunto de sentencias consistente[3], y α una sentencia del lenguaje, entonces podemos tener que:

- α es *aceptada* en K: en el caso en que $\alpha \in Cn(K)$.[4]
- α es *rechazada* en K: en el caso en que $\neg\alpha \in Cn(K)$.[5]
- α es *indeterminada* en K: en el caso en que $\alpha \notin Cn(K)$ y $\neg\alpha \notin Cn(K)$.

El conjunto de estados epistémicos adoptado determina el poder de representación del modelo adoptado. Podrían adoptarse otros conjuntos de estados epistémicos. Por ejemplo, probabilidades representadas por valores numéricos entre 0 y 1, pero eso no lo trataremos aquí. En este modelo contamos con tres estados posibles y las diferentes operaciones de cambio permiten cambiar de uno a otro.

Sea K un conjunto de sentencias y α una sentencia del lenguaje. Las tres operaciones básicas del modelo AGM son las siguientes:

[1] Si X es conjunto de sentencias del lenguaje y α una sentencia del lenguaje, entonces $X \vdash \alpha$ equivale a notar que $\alpha \in Cn(X)$.

[2] $\mathbf{2}^X$ se refiere al conjunto de partes de cualquier conjunto X.

[3] Un conjunto K es consistente si no existe ninguna sentencia α del lenguaje $\mathcal{L}$ tal que $\alpha \in Cn(K)$ y $\neg\alpha \in Cn(K)$. En particular, si K es un conjunto de creencias (*i.e.*, $K = Cn(K)$) y K es inconsistente entonces K es igual a todo el lenguaje (*i.e.*, $K = \mathcal{L}$). Esto hace que los conjuntos de creencias inconsistentes no tengan valor epistémico alguno, aunque no sucede lo mismo con bases de creencias, ya que las inconsistencias pueden producirse ante una sentencia particular y pueden ser eliminadas "localmente".

[4] En este caso, también se dice que $\neg\alpha$ es rechazada en K.

[5] En este caso, también se dice que $\neg\alpha$ es aceptada en K.

- *Expansión*: Esta operación se encarga de incorporar sentencias al conjunto original, sin eliminar ninguna sentencia del mismo. Permite pasar de un estado epistémico en el que una creencia es indeterminada a otro estado epistémico en el cual la misma es aceptada o rechazada.

- *Contracción*: Esta operación elimina sentencias del conjunto original sin incorporar ninguna nueva. Permite pasar de un estado epistémico en el que una creencia es aceptada o rechazada a otro estado epistémico en el cual la misma es indeterminada.

- *Revisión*: Esta operación incorpora sentencias al conjunto original pero puede llegar a eliminar algunas si la sentencia a incorporar no es contradictoria con el fin de preservar consistencia en el conjunto revisado. Permite pasar de un estado epistémico en el que una creencia es aceptada (rechazada) a otro estado epistémico en el cual la misma es rechazada (aceptada).

La operación de revisión puede definirse a partir de la operación de contracción y la operación de revisión mediante la identidad de Levi [AGM85, Gär88, Han96]. De manera análoga, la operación de contracción puede definirse a partir de la operación de revisión mediante la identidad de Harper [AGM85, Gär88, Han96]. De las tres, la más simple es la operación de expansión ya que para hacerla efectiva "basta" con definir operaciones de unión sobre conjuntos y (opcionalmente) operaciones de clausura, por tratarse de una operación monotónica. Formalmente, si K es una base de creencias, la expansión de K por α, notada por $K{+}\alpha$ se define como $K \cup \{\alpha\}$. Si $\mathbf{K}$ es un conjunto de creencias, la expansión de K por α, notada por $K{+}\alpha$ se define como $Cn(K \cup \{\alpha\})$.

En cambio, las operaciones de contracción y revisión requieren, además de operaciones sobre conjuntos y operaciones de consecuencia, de un *mecanismo de selección* para poder llegar a hacer efectivos los cambios en el estado epistémico. Esto hace que, al momento de implementarse las mismas, deba recurrirse a métodos constructivos (por ejemplo: *epistemic entrenchment*), órdenes lexicográficos (sobre el lenguaje o parte de él) u órdenes ad-hoc, con el objetivo de poder llegar a concretar la correspondiente selección. Las operaciones de contracción y revisión están caracterizadas por una serie de postulados de racionalidad. Estos postulados, determinan el comportamiento de tales operaciones de cambio en el nivel de conocimiento (se definen sobre conjuntos de creencias) y fueron introducidos inicialmente por Gärdenfors [Gär88].

2.1.1 Postulados para Contracciones

Sea $\mathbf{K}$ un conjunto de creencias (*i.e.*, $\mathbf{K} = Cn(\mathbf{K})$), α y β sentencias de $\mathcal{L}$, $+$ un operador de expansión y $-$ un operador de contracción. Los *postulados de racionalidad para contracciones* son los siguientes:

($\mathsf{K}^{-}1$) **Clausura (Closure)**: Si $\mathbf{K} = Cn(\mathbf{K})$ entonces $\mathbf{K}{-}\alpha = Cn(\mathbf{K}{-}\alpha)$.

($\mathsf{K}^{-}2$) **Inclusión**: $\mathbf{K}{-}\alpha \subseteq \mathbf{K}$.

($\mathsf{K}^{-}3$) **Vacío (Vacuity)**: Si $\alpha \notin \mathbf{K}$ entonces $\mathbf{K}{-}\alpha = \mathbf{K}$.

($\mathsf{K}^{-}4$) **Exito (Success)**: Si $\nvdash \alpha$ entonces $\alpha \notin \mathbf{K}{-}\alpha$.

($\mathsf{K}^{-}5$) **Recuperación (Recovery)**: Si $\alpha \in \mathbf{K}$ entonces $K \subseteq (K{-}\alpha){+}\alpha$.

(K^-6) **Extensionalidad/Preservación**: Si $\vdash \alpha \leftrightarrow \beta$ entonces $\mathbf{K}-\alpha = \mathbf{K}-\beta$.

(K^-7) **Solapamiento Conjuntivo**: $\mathbf{K}-\alpha \cap \mathbf{K}-\beta \subseteq \mathbf{K}-(\alpha \wedge \beta)$.

(K^-8) **Inclusión Conjuntiva**: Si $\alpha \notin \mathbf{K}-(\alpha \wedge \beta)$ entonces $\mathbf{K}-(\alpha \wedge \beta) \subseteq \mathbf{K}-\alpha$.

2.1.2 Postulados para Revisiones

Sea $\mathbf{K}$ un conjunto de creencias (*i.e.*, $\mathbf{K} = Cn(\mathbf{K})$), α y β sentencias de $\mathcal{L}$, $+$ un operador de expansión y $*$ un operador de revisión. Los *postulados de racionalidad para revisiones* son los siguientes:

(K^*1) **Clausura (Closure)**: Si $\mathbf{K} = Cn(\mathbf{K})$ entonces $\mathbf{K}*\alpha = Cn(\mathbf{K}*\alpha)$.

(K^*2) **Exito (Success)**: $\alpha \in \mathbf{K}*\alpha$.

(K^*3) **Inclusión**: $\mathbf{K}*\alpha \subseteq \mathbf{K}+\alpha$.

(K^*4) **Vacío (Vacuity)**: Si $\mathbf{K} \nvdash \neg\alpha$ entonces $\mathbf{K}*\alpha = \mathbf{K}+\alpha$.

(K^*5) **Consistencia**: $\nvdash \neg\alpha$ entonces $\mathbf{K}*\alpha \neq \mathbf{K}_\perp$.

(K^*6) **Extensionalidad/Preservación**: Si $\vdash \alpha \leftrightarrow \beta$ entonces $\mathbf{K}*\alpha = \mathbf{K}*\beta$.

(K^*7) **Superexpansión**: $\mathbf{K}*(\alpha \wedge \beta) \subseteq (\mathbf{K}*\alpha)+\beta$.

(K^*8) **Subexpansión**: Si $\neg\beta \notin \mathbf{K}*\alpha$ entonces $(\mathbf{K}*\alpha)+\beta \subseteq \mathbf{K}*(\alpha \wedge \beta)$.

Los postulados (K^-1)–(K^-6) [(K^*1)–(K^*6)] se denominan *postulados básicos* para contracciones (revisiones) mientras que los postulados (K^-7)–(K^-8) [(K^*7)–(K^*8)] se denominan *postulados suplementarios* para contracciones (revisiones). Algunos de estos postulados han sido cuestionados por considerárselos implausibles entre las operaciones de cambio. Por ejemplo, el postulado de *recovery* (K^-5) en contracciones así como el postulado de *éxito* (K^*2) en revisiones. Este cuestionamiento se realiza en el nivel de conocimiento sin mención al nivel simbólico. En el presente artículo, podemos llegar a cuestionar algunos de estos postulados no en el nivel de conocimiento sino que lo haremos en el nivel simbólico.

2.1.3 Relaciones entre Contracciones y Revisiones

Existe una estrecha relación entre los postulados para contracciones y revisiones. Sea K un conjunto de sentencias, α una sentencia del lenguaje y $+$ un operador de expansión. Las contracciones y revisiones se interdefinen de la siguiente manera:

- *Identidad de Levi*: $K*\alpha = (K-(\neg\alpha))+\alpha$.
- *Identidad de Harper*: $K-\alpha = K \cap K*\neg\alpha$.

Dadas estas dos identidades, puede demostrarse que si un operador de contracción satisface los postulados (K^-1)–(K^-6) [y opcionalmente (K^-7)–(K^-8)] entonces el operador de revisión que se obtiene aplicando la identidad de Levi satisface los postulados (K^*1)–(K^*6) [y opcionalmente (K^*7)–(K^*8)]. Análogamente, si un operador de revisión satisface los postulados (K^*1)–(K^*6) (y opcionalmente (K^*7)–(K^*8)) entonces el operador de contracción que se obtiene aplicando

la identidad de Harper satisface los postulados $(\mathsf{K}^-1)-(\mathsf{K}^-6)$ (y opcionalmente $(\mathsf{K}^-7)-(\mathsf{K}^-8)$). Sin embargo, existen ciertos operadores de contracción conocidos como *withdrawals* [6] [Mak87] que satisfacen los postulados para contracciones $(\mathsf{K}^-1)-(\mathsf{K}^-4)$ y $(\mathsf{K}^-6)-(\mathsf{K}^-8)$ salvo *recovery* (K^-5) que al aplicárseles la identidad de Levi generan operadores de revisión que satisfacen los postulados $(\mathsf{K}^*1)-(\mathsf{K}^*8)$.

2.2 Bases de Creencias vs. Conjuntos de Creencias

Los *conjuntos de creencias* son conjuntos de sentencias clausurados bajo consecuencia lógica. Esto es, si Cn es un operador de consecuencia lógica, entonces $\mathbf{K}$ es un conjunto de creencias si y solo si $\mathbf{K} = Cn(\mathbf{K})$. Este tipo de representación para un estado epistémico es más adecuado para comprender el comportamiento de las operaciones pero no es adecuado para su implementación. La razón de su no adecuacidad es simple: si contamos con un lenguaje con un conjunto finito de proposiciones, es posible obtener infinitas sentencias deducibles de cualquier conjunto finito. Por tal motivo, en caso de pretender implementar las operaciones de cambio (del modelo AGM) es necesario representar los estados epistémicos mediante bases de creencias. En [GM88], Gärdenfors y Makinson plantean la implausibilidad de definir operadores de cambio sobre conjuntos de creencias:

> *"... a knowledge set is supposed to be closed under logical consequence will cause problems when it comes to implementing a system, since there are in general infinitely many logical consequences to take care of."*

Las *bases de creencias* son conjuntos cualesquiera de sentencias no necesariamente clausurados bajo consecuencia lógica. Sin embargo, a partir de esta descripción, pueden obtenerse bases de creencias infinitas. Por tal motivo, debemos restringir aún más la representación de estados epistémicos a bases de creencias finitas, también conocidas como *bases de conocimiento*. El problema es que el estado epistémico asociado a un conjunto de creencias (en el nivel de conocimiento) puede ser generado por varias bases de conocimiento diferentes (en el nivel simbólico). Por ejemplo, el estado epistémico reflejado por $Cn(\{a, b\})$ puede ser generado, entre otras, por las siguientes bases de conocimiento: $\{a, b\}$, $\{a, a \leftrightarrow b\}$, $\{b, a \leftrightarrow b\}$, $\{a, a \rightarrow b\}$, $\{a, \neg b \rightarrow \neg a\}$. En base a esto, es necesario trabajar sobre bases de conocimiento de modo tal que los cambios en el conjunto de creencias asociados se acerquen lo más posible al estado ideal, *i.e.*, como si hubiésemos realizado las operaciones de cambio directamente sobre conjuntos de creencias.

Otra razón para optar por implementar las operaciones de cambio sobre bases de creencias finitas es que, a nuestro modo de ver, los procesos intuitivos de cambio de creencias no se aplican sobre conjuntos clausurados sino sobre conjuntos finitos. Esto es, adoptaremos la teoría fundacional (foundational theory) para representar el estado epistémico de un agente, en lugar de la teoría de coherencia (coherence theory) adoptada en el modelo AGM [Gär90]. La teoría fundacional asume que debe mantenerse pista de las razonas por las cuales una creencia es aceptada en un estado epistémico. En cambio, la teoría de coherencia asume que las creencias no requieren ninguna justificación porque ellas se justifican a si mismas. Según Gärdenfors [Doy92, Gär90], para un coherentista el cuerpo del conocimiento es como una balsa en la cual una tabla ayuda directa o indirectamente a que se mantengan las demás, y donde ninguna tabla podría mantenerse estable sin la ayuda de las demás.

[6] El nombre de *withdrawal* fue introducido por David Makinson.

3 Nivel de Conocimiento vs. Nivel Simbólico

En un famoso artículo de los años 80, Allen Newell [New82] postula la existencia del "nivel de conocimiento" (knowledge level) situado inmediatamente arriba del "nivel simbólico". El nivel de conocimiento está especificado enteramente en términos del contenido del conocimiento (creencias), sin ninguna distinción entre la información explícita y la información implícita.

Si aplicamos estos conceptos a revisión de creencias, los postulados y las construcciones no deberían hacer ninguna referencia a las sentencias del conjunto de partida. Este criterio es compatible con los postulados AGM para conjuntos de creencias.

En bases de creencias, este criterio es defendido por Dalal [Dal88], quien argumenta:

> *"The revised knowledge base should not depend of the sintax (or representation) of either the old knowledge or the new information."*

Luego propone una función para bases de creencias basada en las distancias entre los modelos que satisfacen el conjunto original y los modelos que satisfacen la nueva fórmula. Esta idea es posteriormente retomada, entre otros, por Katsuno y Mendelzon en [KM92] y Lin [Lin]. Estas funciones si bien correctas desde el punto de vista teórico, no sirven para implementaciones, dado que violan un principo básico en la construcción de programas: la declaratividad.

Sin embargo, si deseamos formalizar funciones de cambio, no deberíamos comprometernos con una sintaxis en particular. Esto no significa que no exista diferencia entre las sentencias explícitas e implícitas de la base, sino que nada diremos sobre su forma sintáctica particular, aunque sí sobre su comportamiento. Podemos distinguir entonces tres niveles bien diferenciados:

- El *nivel de conocimiento* (*knowledge level*), donde no distinguiremos entre sentencias explícitas o implícitas en la base. Es decir solo se predica sobre el resultado de la operación de contracción y no sobre el comportamiento de las sentencias de la base. Esto se refleja en el siguiente postulado:

  Extensionalidad Fuerte (Strong Extensionality): Si $Cn(K) = Cn(H)$ entonces $Cn(K-\alpha) = Cn(H-\alpha)$.

  Además de las funciones mencionadas, podemos citar las contracciones propuestas por Fuhrmann [Fuh91] y Nebel [Neb89]. Sin embargo, ambas violan las funciones propuestas por Hansson [Han92, Han94] , dado que no satisfacen su postulado de inclusión para bases: $K-\alpha \subseteq K$. Las funciones antes citadas también pueden producir el resultado $K-\alpha = \{\}$ en el caso en que K se escriba como una única sentencia.[7]

- Un *metanivel simbólico* (*simbolic metalevel*) donde se predica acerca del comportamiento de las sentencias que componen la base de creencias, pero sin ninguna referencia a su forma sintáctica.

  Podemos citar como ejemplo los postulados propuestos por Hansson [Han92, Han94] de *core-retaiment*, *uniformity* y *relevance* entre otros.

- Un *nivel simbólico* (*simbolic level*) con referencias explícitas sobre la sintaxis de las sentencias que componen la base.

  Preferencia de hechos : Si $\beta \in K$ y $(\beta \rightarrow \alpha) \in K$, entonces $(\beta \rightarrow \alpha) \notin K-\alpha$,

[7] Notemos que la base de creencias $K = \{a, b, a \rightarrow \neg c, d \rightarrow \neg b\}$ es equivalente (en el nivel de conocimiento) a $K = \{a \wedge b \wedge (a \rightarrow \neg c) \wedge (d \rightarrow \neg b)\}$.

En el primer nivel se sitúan la axiómatica a partir de postulados, mientras que en el nivel metasimbólico podemos citar a la *importancia epistémica* o *epistemic entrenchment*. Sin embargo, uno puede preguntarse que, dado que existe un teorema de representación que une importancia epistémica con la axiomática AGM, dichos niveles no se mezclan. La respuesta es que no, ya que la relación entre ambos modelos viene dada por la regla:

$(C \leq)$ $\alpha \leq_K \beta$ si y solo si $\alpha \notin K-(\alpha \wedge \beta)$ o $\vdash (\alpha \wedge \beta)$.

que claramente está situada en el nivel metasimbólico.

4 Implementación de Operaciones de Cambio

A nivel de implementación, es necesario tomar ciertas decisiones que no son consideradas no solo en el nivel de conocimiento sino también en el nivel simbólico. En las siguientes subsecciones analizaremos el efecto que pueden llegar a producir ciertas decisiones realizadas al momento de implementar las operaciones de cambio.

4.1 Expansiones

La operación de expansión es la que requiere una definición más simple aunque esto no es extensivo a la complejidad de su implementación. En caso de representar el estado epistémico de un agente mediante conjuntos de creencias, la operación de expansión se define de la siguiente manera. Sea $\mathbf{K}$ un conjunto de creencias y α una sentencia del lenguaje. Entonces la expansión de $\mathbf{K}$ por α, notada mediante la expresión $\mathbf{K}+\alpha$, se define como sigue:

$$\mathbf{K}+\alpha = Cn(\mathbf{K} \cup \{\alpha\})$$

En caso de que el operador de consecuencia sea supraclásico, esta operación se torna implausible por su costo computacional. Por ejemplo, si $\mathcal{L}$ es un lenguaje proposicional conteniendo las letras proposicionales $a, b, \ldots, z$, $\mathbf{K} = Cn(\{a, b \vee c\})$ y Cn es un operador de consecuencia supraclásico, entonces la expansión de $\mathbf{K}$ por d contiene las sentencias $d \vee a$, $d \vee b$, ..., $d \vee z$, ... $\neg d \rightarrow a$, ..., $d \wedge a$, ..., e infinitas sentencias más. La infinitud del conjunto expandido (en realidad, el conjunto de base $\mathbf{K} = Cn(\{a, b \wedge c\})$ ya es extensionalmente infinito) hace imposible implementar operaciones de expansión si se parte de un operador de consecuencia clásico. Por tal razón, es necesario implementar operaciones de cambio sobre bases de conocimiento, esto es, bases de creencias finitas.

La definición de expansión sobre bases de creencias no necesita recurrir a un operador de consecuencia. Sea K una base de conocimiento finita y α una sentencia cualquiera. Entonces la expansión de K por α se define como sigue:

$$K+\alpha = K \cup \{\alpha\}$$

Este tipo de expansión es la más común entre los sistemas basados en conocimiento en los cuales se representan las creencias mediante conjuntos finitos. Sin embargo, en caso de que $\alpha \in K$, esto es existe al menos una demostración o derivación sintáctica de α en K: ¿es necesario agregar una nueva evidencia para α?. Por ejemplo, si un agente cree que birdy vuela porque es un pájaro y los pájaros vuelan, y luego le informan que birdy vuela: ¿es necesario o conveniente introducir el hecho de que tweety vuela?. Formalicemos el conocimiento anterior

en un lenguaje de primer orden. $K = \{pájaro(tweety), \forall(x)[pájaro(x) \rightarrow vuela(x)]\}$ y sea la creencia $\alpha = vuela(tweety)$. Si usamos la definición anterior, el conjunto expandido sería $K = \{pájaro(tweety), vuela(tweety), \forall(x)[pájaro(x) \rightarrow vuela(x)]\}$. Luego, existen dos demostraciones o derivaciones de que tweety vuela a partir de la base de conocimiento expandida. Si en el futuro, el agente deja de creer que tweety es un pájaro: ¿debe seguir creyendo que vuela o no?. La respuesta a esta pregunta es si, siempre y cuando el hecho de que vuela tweety se incorporó a la base de creencias (en algún momento del pasado) sin depender de las "evidencias propias" que existían de tal hecho.

En caso de que no se den las condiciones anteriores (independencia de pruebas para incorporar una creencia) es necesario recurrir a una definición alternativa de expansión. Sea K una base de conocimiento finita y α una sentencia cualquiera. Entonces la expansión de K por α se define como sigue:

$$K{+}\alpha = \begin{cases} K & \text{si } \alpha \in Cn(K) \\ K \cup \{\alpha\} & \text{en otro caso} \end{cases}$$

Esta definición de expansión generalmente se implementa en los *sistemas de mantenimiento de verdad* (*truth maintenance systems*) [Doy79] en los que trata de evitarse circularidad o redundancia en las demostraciones de hechos básicos.

4.2 Contracciones

Las operaciones de contracción se definen de varias maneras. En las siguientes secciones presentaremos ciertas modelos alternativos de contracción.

Definición 4.1: Sea K un conjunto de sentencias cualesquiera y α una sentencia. El *conjunto de restos* de K con respecto a α se es el conjunto de subconjuntos X de K tales que:

1. $X \subseteq K$.
2. $X \not\vdash \alpha$.
3. No existe W tal que $X \subset W \subseteq K$ y $W \not\vdash \alpha$.

El conjunto de restos es el conjunto de subconjuntos maximales de K que fallan en implicar α. Cada elemento del conjunto de restos se denomina un α-resto. □

La operación de contracción del modelo AGM [AGM85] se basa en la selección de los "mejores elementos" del conjunto de restos.

Definición 4.2: Sea K un conjunto de sentencias, α una sentencia y $K \perp A$ el conjunto de restos de K con respecto a α. Decimos que γ es una *función de selección* si y solo si:

1. Si $K \perp \alpha = \emptyset$ entonces $\gamma(K \perp \alpha) = \{K\}$.
2. Si $K \perp \alpha \neq \emptyset$ entonces $\gamma(K \perp \alpha) = X$ tal que $\emptyset \subset X \subseteq (K \perp \alpha)$.

□

Definición 4.3: Sea K un conjunto de creencias, α una sentencia y $K \perp \alpha$ el conjunto de restos de $\mathbf{K}$ con respecto a α. Sea γ una función de selección para $K \perp \alpha$. El operador $\div\gamma$, denominado *partial meet contraction determinado por* γ se define de la siguiente manera:

$$(\textbf{Def}\ \ \text{Partial Meet Contraction})\ \ \mathbf{K}\div\gamma\alpha = \cap\gamma(\mathbf{K} \perp \alpha)$$

□

Esto es, la operación de *partial meet contraction* [AGM85] de K con respecto a α se define como la intersección de los mejores subconjuntos maximales de α (seleccionados por γ) que fallan en implicar α.

Una manera alternativa de definir operaciones de contracción, es a través de kernel contractions [Han96]. A diferencia de las operaciones de partial meet contraction, las kernel contraction se definen a partir de los α-kernels (conjuntos minimales de un conjunto que implican la sentencia α), realizando un "corte" sobre cada uno de los elementos de ese conjunto a través de una función de incisión.

Definición 4.4: Sea K un conjunto de sentencias y sea α una sentencia. El conjunto $K \perp\!\!\!\perp \alpha$, denominado *conjunto de kernels* es el conjunto de conjuntos X tales que:

1. $X \subseteq K$.
2. $X \vdash \alpha$.
3. No existe W tal que $W \subset X \subseteq K$ y $W \vdash \alpha$.

El conjunto $K \perp\!\!\!\perp \alpha$ también se denomina *conjunto de α-kernels* y cada uno de sus elementos se denomina α-kernel. Cada α-kernel es un subconjunto minimal de K que implica a α. □

Definición 4.5: Una *función de incisión* σ para un conjunto K si para toda sentencia α vale que:

1. $\sigma(K \perp\!\!\!\perp \alpha) \subseteq \cup(K \perp\!\!\!\perp \alpha)$.
2. Si $X \in K \perp\!\!\!\perp A$ y $X \neq \emptyset$ entonces $X \cap \sigma(K \perp\!\!\!\perp \alpha) \neq \emptyset$.

□

La función de incisión selecciona que sentencias serán descartadas. El resultado de contraer un conjunto K con respecto a una creencia α debería consistir de todos los elementos del conjunto original K que no son removidos por la función de incisión.

Definición 4.6: Sea K un conjunto de sentencias, sea α una sentencia cualquiera y $K \perp\!\!\!\perp \alpha$ el conjunto de α-kernels de K. Sea σ una función de incisión para K. El operador $\sim \sigma$, denominado *kernel contraction determinado por* σ se define de la siguiente manera:

$$(\textbf{Def}\ \mathsf{Kernel\ Contraction})\quad \mathsf{K}{\sim}\,\sigma\alpha = \mathsf{K} \setminus \sigma(K \perp\!\!\!\perp \alpha)$$

Un operador $-$ es un operador de kernel contraction para K si y solo si existe alguna función de incisión σ tal que $K - \alpha = K \sim \sigma\alpha$ para toda sentencia α. □

Las operaciones de kernel contraction son equivalentes a las operaciones de partial meet contraction cuando se aplican sobre conjuntos de creencias. Sin embargo, esto no sucede cuando se aplican sobre bases de creencias. En realidad, toda partial meet contraction puede ser generada a través de una operación de kernel contraction aunque existen kernel contractions no generables por operaciones de partial meet contraction. Por lo tanto, al momento de implementar un operador de contracción, es importante tener en mente esta propiedad.

5 Propiedades de las Contracciones

Como dijimos anteriormente, las operaciones de contracción están caracterizadas por ciertos postulados de racionalidad. Estos postulados permiten caracterizar los cambios en el nivel de conocimiento pero deben ser apropiadamente reformulados cuando se trata el problema en el nivel simbólico. El postulado de *extensionalidad* pretende captar la noción de irrelevancia de la sintaxis, esto es, independientemente de la "forma sintáctica" con que se representa cierta creencia, los cambios en el estado epistémico deberían ser los mismos. Este postulado (formulado para caracterizar las contracciones en el nivel de conocimiento), en general no se satisface en las operaciones de cambio construíbles en el nivel simbólico. En tal sentido, existen diferentes postulados alternativos que pretenden captar (en mayor o menor medida) la noción de irrelevancia de la sintaxis. Por ejemplo:

(a) **Uniformidad (Uniformity)**: Si para todo subconjunto H de K vale que $\alpha \in Cn(H)$ si y solo si $\beta \in Cn(H)$ entonces $K{-}\alpha = K{-}\beta$.

(b) Si $\vdash \alpha \leftrightarrow \beta$ entonces $Cn(K{-}\alpha) = Cn(K{-}\beta)$.

(c) Si $Cn(K) = Cn(H)$ entonces $K{-}\alpha = H{-}\alpha$.

(d) **Extensionalidad Fuerte (Strong Extensionality)**: Si $Cn(K) = Cn(H)$ entonces $Cn(K{-}\alpha) = Cn(H{-}\alpha)$.

Toda operación del contracción del modelo AGM satisface los postulados alternativos presentados anteriormente. Sin embargo, las operaciones de contracción definidas en el nivel simbólico satisfacen solo algunos de ellos.

Ejemplo 5.1: Sea la base de creencias $K = \{p, p \rightarrow q\}$. Sabemos que q es lógicamente equivalente a $(q \wedge \top) \equiv (q \wedge (p \vee \neg p))$. Los resultados de contraer K con respecto a q y $(q \wedge (p \vee \neg p))$ podrían ser:

$$K{-}q = \{p \rightarrow q\}.$$

$$K{-}(q \wedge (p \vee \neg p)) = \{p\}.$$

En este caso, es claro que $K{-}p \neq K{-}(p \wedge (q \vee \neg q))$, y $Cn(K{-}p) \neq Cn(K{-}(p \wedge (q \vee \neg q)))$
□

Otro ejemplo que podemos plantear acerca de las propiedades que no se satisfacen en el nivel simbólico puede darse entre *inclusión* y los postulados (c) y (d).

Ejemplo 5.2: Sean $K_1 = \{p, q, r, s\}$ y $K_2 = \{p \wedge q \wedge r, p \wedge q \rightarrow s\}$. Supongamos que deseamos contraer a K_1 y K_2 con respecto a p. Los resultados de contraer K_1 y K_2 con respecto a la creencia p podrían ser los siguientes:

$$K_1{-}q = \{p, r, s\}.$$

$$K_2{-}q = \{p \wedge q \rightarrow s\}.$$

Es claro que en este caso, no se satisfacen los postulados de irrelevancia de la sintaxis (c) y (d).
□

Según Hansson [Han96], una lógica es extensional si permite que sentencias lógicamente equivalentes sean intercambiables libremente entre si. En el caso de la teoría de cambio de creencias, diremos que un operador de contracción (o bien de revisión o expansión) es extensional si las contracciones con respecto a sentencias lógicamente independientes produce el "mismo" resultado (en el nivel de conocimiento o en el nivel simbólico).

El postulado de *inclusión* también puede entrar en conflicto con el postulado de *recovery* si se trabaja en el nivel simbólico. Consideremos el siguiente ejemplo.

Ejemplo 5.3: Sea la base de creencias $K = \{a, b, a \wedge b \rightarrow c, d\}$. Supongamos que deseamos contraer a K con respecto a c. Los posibles resultados de la contracción (que satisfacen al menos el postulado de éxito) son los siguientes:

$K_1 = \{a, b, d\}$.

$K_2 = \{a, a \wedge b \rightarrow c, d\}$.

$K_3 = \{b, a \wedge b \rightarrow c, d\}$.

$K_4 = \{a, d\}$.

$K_5 = \{b, d\}$.

$K_6 = \{a \wedge b \rightarrow c, d\}$.

En cualquiera de las bases resultantes K_i, $1 \leq i \leq 6$ no se pueden recuperar las creencias que permitían deducir c pero que no pueden deducirse de c. Si K estuviera clausurada por el operador Cn, para cada sentencia x en K entonces $c \rightarrow x \in Cn(K)$. Por lo tanto, si se reincorporara c entonces se recuperarían las sentencias eliminadas. Esto implicaría que, en lugar de K_2 (por citar un caso particular) tendríamos la siguiente base de creencias:

$$K'_2 = \{a, a \wedge b \rightarrow c, d, c \rightarrow b\}$$

En este caso, se satisface el postulado de *recovery* pero se viola el postulado de *inclusión*. □

Un aspecto a tener en cuenta en el ejemplo anterior es el siguiente. Supongamos que deseamos contraer un conjunto K con respecto a una sentencia α. Entonces, para poder satisfacer *recovery* en la contracción, por cada sentencia β tal que $\beta \in K$ y $\beta \notin K - \alpha$ entonces $\alpha \rightarrow \beta \in K - \alpha$. Esta propiedad asegura *recovery* pero elimina la propiedad de *inclusión*.

Una cuestión que surge este punto, y que se ignora en general en el nivel de conocimiento en general, es determinar cual es el significado intuitivo de contar con ese tipo de sentencias. Consideremos el siguiente ejemplo.

Ejemplo 5.4: Sea $K = \{p(t), a(t), \forall(x)(p(x) \wedge a(x) \rightarrow \neg v(x)), c(o)\}$, donde t se refiere al objeto tweety, o al objeto opus, $p(x)$ es un predicado que es verdadero cuando x es un pájaro, $a(x)$ es verdadero cuando x es un avestruz, $v(x)$ es verdadero cuando x vuela y $c(x)$ es verdadero cuando x es un canguro. Supongamos que deseamos contraer a K con respecto al hecho $\neg v(t)$. Entonces, uno de las posibles contracciones sería el siguiente conjunto:

$$K' = \{p(t), \forall(x)(p(x) \wedge a(x) \rightarrow \neg v(x)), c(o)\}$$

Para que se satisfaga *recovery* necesitamos contar además con la sentencia $\neg v(t) \rightarrow a(t)$. Si bien es cierto que en el conjunto original puede inferirse que las cosas que vuelan, o bien

no son pájaros o bien no son avestruces, y si nos abstraemos del efecto de los operadores de consecuencia lógica, el conocimiento anterior no parece expresar que si tweety no vuela entonces es un avestruz. □

Este ejemplo pretende mostrar que, con el fin de garantizar recovery, puede llegar a 'alterarse' el significado pretendido del conocimiento. Además, al momento de implementar (y aplicar) operadores de contracción, debe permitirse la iteración (*i.e.*, la aplicación repetida) de cambios. Si se sigue la política anterior en situaciones en que las contracciones se aplican con relativa frecuencia, puede llegar a obtenerse una base de conocimiento llena de sentencias falaces (introducidas solo con el objeto de garantizar *recovery*) que escapan totalmente al significado pretendido. Tal vez el problema anterior se presente por agregar las sentencias que buscan asegurar recovery en el 'mismo nivel' que las sentencias que constituyen la base de conocimiento. En el ejemplo anterior, podemos obtener la siguiente base de creencias:

$$K' = \{p(t), \forall(x)(p(x) \wedge a(x) \rightarrow \neg v(x)), c(o)\}$$

y un conjunto **Rec** que contenga la tupla $(\neg v(t), a(t))$ que indica que si el siguiente cambio de K' es una expansión por $\neg v(t)$ entonces se debe reincorporar $a(t)$. Luego, la expansión se define de la siguiente manera:

$$K{+}\alpha = K \cup \{\alpha\} \cup \{\beta : (\alpha, \beta) \in \mathbf{Rec}\}$$

$$\mathbf{Rec}' = \mathbf{Rec} \setminus \{(\alpha, \beta) : \beta \in \mathbf{K}{+}\alpha\}$$

Este mecanismo garantiza la satisfacción de recovery pero puede anular algunos de los postulados que caracterizan racionalmente a las expansiones. En particular, pueden generarse inconsistencias en $K{+}\alpha$ a pesar de que $\neg\alpha \notin Cn(K)$. Este caso se daría si la expansión no se realiza inmediatamente después de una contracción. Por lo tanto, la definición anterior debe reformularse de la siguiente manera:

$$K{+}\alpha = K \cup \{\alpha\} \cup \{\beta : (\alpha, \beta) \in \mathbf{Rec} \text{ y } \neg\beta \notin Cn(K)\}$$

$$\mathbf{Rec}' = \mathbf{Rec} \setminus \{(\alpha, \beta) : \beta \in \mathbf{K}{+}\alpha\}$$

La eliminación de tuplas en el conjunto de sentencias recuperables es con el objeto de eliminar tuplas que dejan de tener sentido cuando se producen expansiones. Por ejemplo, si $(\alpha, \beta) \in \mathbf{Rec}$ significa que β fue eliminada de la base de conocimiento K en una contracción (anterior) por α. Si expandimos por α, recuperamos (si es posible) la sentencia eliminada en el pasado, y la eliminamos del conjunto de sentencias a recuperar.

6 Implementando Contracciones

En esta sección analizaremos las posibles implementaciones de las operaciones de contracción. Las expansiones se definen de una manera casi directa, sobre la base de operaciones sobre conjuntos. Las revisiones son mucho más complejas pero pueden obtenerse a partir de expansiones y contracciones aplicando la identidad de Levi. Por esta razón es que centralizamos nuestro análisis en las implementaciones de operaciones de contracción.

Una de las posibles construcciones de operadores de contracción es mediante las *kernel contractions*. Estas operaciones se definen de la siguiente manera: sea K la base de conocimiento

y α la sentencia a contraer. Se obtienen los subconjuntos minimales de K que implican α (α-kernels) y se elimina al menos una sentencia de cada uno de esos subconjuntos. La eliminación se realiza a través de una función de incisión. Esta función de incisión determina las sentencias a ser eliminadas de cada α-kernel (por esto, también se eliminan de K), generando de este modo, un "corte" en cada uno de los α-kernels.

¿Como pueden obtenerse los α-kernels?. Una manera de hacerlo es a través de un lenguaje de programación en lógica como PROLOG. El mismo trabaja con un lenguaje objeto que es un subconjunto de la lógica de primer orden. Este subconjunto está constituído por cláusulas de Horn [StSh86], asume cuantificadores existenciales de manera implícita, no contiene cuantificadores existenciales ni símbolos funcionales. Como no se admiten cláusulas con variables libres, valen la propiedad de deducción asumida en el operador de consecuencia.

Una de las grandes potencialidades de Prolog es la facilidad de escribir *metaintérpretes* [StSh86]. Los metaintérpretes de Prolog son intérpretes de Prolog escritos en Prolog. Por ejemplo, el siguiente metaintérprete [StSh86] es un intérprete de Prolog puro que obtiene el árbol de prueba de la consulta `A`:

```
% solve(<A>,<Prueba>): <Prueba> es el arbol de prueba de la consulta <A>.

solve(true,true).
solve((A,B),(PruebaA,PruebaB)):-solve(A,PruebaA),solve(B,PruebaB).
solve(A,A <- PruebaB):-clause(A,B),solve(B,PruebaB).
```

Generalizando este metaintérprete mediante predicados de segundo orden del lenguaje Prolog (como `findall` o `setof`) pueden obtenerse todas las pruebas de una consulta `A`. Luego, mediante algun criterio de selección (determinado por el orden lexicográfico de las sentencias que componen la prueba o por un orden ad-hoc) pueden generarse operaciones que realicen la contracción de una base de conocimiento K con respecto a una sentencia α. Una de las características de este proceso es que se construye directamente sobre el motor de inferencia de Prolog. Este método selecciona la submeta de más a la izquierda y utiliza como regla de búsqueda la primer cláusula cuya cabeza unifica con la submeta a resolver. Este proceso se conoce formalmente como *resolución-SLD* [StSh86] y el mismo es completo si el árbol de búsqueda se genera por niveles (*breadth first*) y no en profundidad (*depth first*). Lamentablemente, por razones de espacio en el cómputo de árboles de búsqueda, Prolog recorre los árboles en profundidad. Esto acarrea ciertos problemas que podemos ver en el siguiente ejemplo:

Ejemplo 6.1: Consideremos la base de conocimiento $K = \{p \leftrightarrow q, p\}$ que se representaría en Prolog mediante el siguiente programa definido [StSh86]:

```
p :- q.
q :- p.
p.
```

En este caso, la meta `p` no tiene éxito si no se garantiza un barrido por niveles del árbol SLD. Como Prolog genera los árboles en profundidad, la consulta por `p` no tendría una respuesta positiva. Sin embargo, si se realiza una contracción por de K por q y se le da preferencia a los hechos, entonces se obtiene el siguiente programa lógico:

```
p :- q.
p.
```

Después de realizar la contracción de K por q, la consulta por `p` tiene éxito. Por lo tanto, parecería que en la contracción de K por q se agregaron creencias pues la consulta `p` tiene éxito en $K-q$ (*i.e.*, $q \in Cn(K-q)$) pero no tiene éxito en K (*i.e.*, aparentemente $q \notin Cn(K)$). Esta situación puede generarse por la no completitud del motor de inferencia utilizado. $\square$

El anterior metaintérprete trabaja con un lenguaje muy reducido como lo es con cláusulas de Horn, esto es, cláusulas con, a lo sumo, un literal[8] positivo. Si $A, B_1, \ldots, B_n$ son átomos entonces las sentencias válidas en este lenguaje son las siguientes:

1. Hechos: este tipo de sentencias están compuestas por un único literal positivo (A) y se representa como $A \leftarrow$.

2. Reglas: este tipo de sentencias están compuestas por un único literal positivo y al menos un literal negativo ($A \vee \neg B_1 \vee \ldots \vee \neg B_n$) y se representan como $A \leftarrow B_1 \wedge \ldots \wedge B_n$.

3. Consultas: este tipo de sentencias están compuestas por, al menos, un literal negativo ($\neg B_1 \vee \ldots \vee \neg B_n$) y se representa como $\leftarrow B_1 \wedge \ldots \wedge B_n$.

4. Cláusula Vacía: Esta sentencia representada por el símbolo $\perp$ o $\square$ representa una contradicción y es una cáusula sin átomos.

Las cláusulas son, en general, de la forma $P \leftarrow Q$. P se denomina *cabeza* de la cláusula mientras que Q se denomina *cuerpo*. Si bien este lenguaje es de gran poder expresivo y muy utilizado en los lenguajes de programación en lógica, es muy específico y no permite inferencia de información negativa.[9] Por lo tanto, no pueden producirse inconsistencias y las operaciones de revisión (definidas mediante la identidad de Levi) no tendrían mayor sentido. Sin embargo, las contracciones pueden definirse y ser aplicadas en el lenguaje propuesto.

Un lenguaje más general debe permitir la utilización de clásulas con más de un literal positivo. De este modo, podrían expresarse reglas con cláusulas negativas en el cuerpo y/o en la cabeza. Esto es, sentencias de la forma $\neg A$ o $\neg A \leftarrow B_1 \wedge \ldots \wedge \neg B_i \wedge \ldots \wedge B_n$ con $A, B_1, \ldots, B_i, \ldots, B_n$ siendo átomos del lenguaje. Sin embargo, en estos lenguajes, deben utilizarse otros mecanismos de inferencia para obtener los conjuntos de kernels y, función de incisión mediante, determinar que sentencias serán eliminadas de la base de conocimiento. Entre los mecanismos de inferencia utilizados, podemos optar por representar las sentencias de manera clausal (esto es, como una conjunción de disyunción de literales) y utilizar eliminación de literales complementarios [GN88]. Entre estos métodos, podemos utilizar los métodos de resolución *unitaria*, *por entrada*, *lineal*, *ordenada*, *dirigida* o *mediante conjunto soporte* [GN88]. El método de resolución ordenada es extremadamente eficiente pero puede resultar no completo con lenguajes compuestos por cláusulas que no son de Horn. El método de conjunto soporte puede iniciarse tomando como soporte a la negación de la sentencia a demostrar, trabajando de manera "backward" a partir de la meta (negación de la sentencia a demostrar) hasta llegar a una cláusula vacía.

[8] Un literal es un átomo o la negación de un átomo. Un literal positivo es un átomo mientras que un literal negativo es la negación de un átomo. Un átomo es una fórmula bien formada del lenguaje objeto de la forma $p(t_1, \ldots, t_2)$ donde p es un símbolo de predicado de aridad n y los t_i's son términos (constantes, variables, funciones compuestas) del lenguaje.

[9] La extensión de este lenguaje que utiliza *negación por falla* permite la inferencia de información negativa. Sin embargo, este tipo de lenguaje no es Horn puro pues admite un tipo de negación (por falla, no clásica) en el cuerpo de una cláusula.

Una manera de hacer más eficientes las operaciones de cambio consiste en mantener las posibles demostraciones de las sentencias que componen una base de conocimiento (tanto en el nivel simbólico como en el nivel de conocimiento) mediante un *grafo de conección* que las vincule. Todo cómputo realizado al momento de realizar una consulta al sistema es almacenado en este grafo. Esto exige un mayor espacio de almacenamiento pero acelera las demostraciones ya que se evitan computaciones (que incluyen, entre otros, algoritmos de unificación) realizando operaciones directamente sobre el grafo de conección.

7 Conclusiones

En este artículo presentamos algunos de los inconvenientes y las posibles soluciones que pueden encontrarse al implementar las operaciones de cambio. Para hacer ello, presentamos las diferentes operaciones de cambio del modelo AGM analizando el comportamiento de las mismas en el nivel de conocimiento, en términos del contenido del conocimiento (creencias) sin ninguna distinción entre la información explícita y la información implícita.

Luego, analizamos las operaciones de cambio sobre bases de conocimiento (esto es, conjuntos finitos no necesariamente clausurados bajo consecuencia lógica) en los cuales se distinguen claramente las creencias explícitas. Vimos tambien como ciertos postulados definidos en el nivel de conocimiento deben ser transformados en nuevos postulados "menos restrictivos" con el objetivo de respetar los principios básicos del modelo AGM: mínimo cambio, irrelevancia de la sintaxis y adecuacidad en la representación de conocimiento.

También introducimos un nivel meta-simbólico donde se predica acerca del comportamiento de las sentencias que componen una base de conocimiento, pero sin ninguna referencia a su forma sintáctica. Por último, analizamos las operaciones de cambio en el nivel de implementación, dando diferentes alternativas para implementar operaciones de cambio (tanto expansiones como contracciones).

Referencias

[AGM85] Alchourrón C., Gärdenfors P., Makinson D.: *On the Logic of Theory Change: Partial Meet Contraction and Revision Functions.* **Journal of Symbolic Logic 50**: 510-530, 1985.

[Dal88] Dalal, M.: *Investication into a Theory of Knowledge Base Revision: Preliminary Report.* **Seventh National Converence on Artificial Intelligence, (AAAI-88)**: 475-479, 1988.

[Doy79] Doyle, J.: *A Truth Maintenance Systems.* **Artificial Intelligence 12**: 231-272, 1979.

[Doy92] Doyle, J.: *Reason Maintenance and Belief Revision: Foundations versus Coherence Theories.* En Gärdenfors (ed.), **Beilef Revision**, Cambridge University Press, Cambridge, 1992.

[Fuh91] Fuhrmann, André: *Theory Contraction through Base Contraction.* **Journal of Philosophical Logic 20**: 175-203, 1991.

[Gär88] Gärdenfors P.: *Knowledge in Flux: Modelling the Dynamics of Epistemic States.* **Cambridge, MA: The MIT Press, Bradford Books**, 1988.

[Gär90] Gärdenfors Peter: *The Dynamics of Belief Systems: Foundations vs. Coherence Theories.* **Revue Internationale de Philosophie**: 24-46, 1990.

[GM88] Gärdenfors P., Makinson D.: *Revisions of Knowledge Systems using Epistemic Entrenchment.* **Second Conference on Theoretical Aspects of Reasoning About Knowledge**: 83-95, 1988.

[GN88] Genesereth M., Nilsson N.: *Logical Foundations of Artificial Intelligence.* **Stanford University**. 1988.

[Han92] Hansson, S.: *In Defense of Base Contraction.* **Synthese 91**: 239-245, 1992.

[Han94] Hansson, S.: *Taking Belief Bases Seriously.* **D. Prawitzand, D. Westertähl (eds.), Logic and Philosophy in Uppsala 236**: 13-28 , 1994.

[Han96] Hansson Sven Ove: *A Textbook of Belief Dynamics: Theory Change and Database Updating.* Uppsala University, Sweden, 1996.

[KM92] Katsuno, H. & Mendelzon, A.: *Propositional Knowledge Base Revision and Minimal Change.* **Artificial Intelligence 52**: 263-294, 1991.

[Lin] Lin, Jinxin. *Integration of Weighted Knowledge Bases.* To appear in **Artificial Intelligence**.

[Mak87] Makinson, David: *On The Status of the Postulate of Recovery in the Logic of Theory Change.* **Journal of Philosophical Logic 16**: 383-394, 1987.

[Neb89] Nebel, B.: *Knowledge Level Analysis of Belief Revision.* **First International Conference of Principles of Knowledge Representation**: 301-311, 1989.

[New82] Newell Allen: *The Knowledge Level.* **Artificial Intelligence 18**: 87-127, 1982.

[StSh86] Sterling Leon, Shapiro Ehud: *The Art of Prolog.* **The MIT Press**, Cambridge, London, England, 1986.