ISSN 2007-9737

Algoritmo basado en modelos para la revisión de creencias entre formas normales conjuntivas

Guillermo De Ita Luna, Fernando Zacarias Flores, Alma Delia García García

Benemerita Universidad Autonoma de Puebla, Facultad de Ciencias de la Computación, México

deita@cs.buap.mx, fzflores@yahoo.com.mx, deliakzy@gmail.com

Resumen. La revisión de creencias es un área central en la representación de conocimiento y en el procesamiento de razonamiento automático. Consideraremos una base inicial de conocimiento *K* y una nueva información ϕ, ambas codificadas en forma normal conjuntiva (FC). Presentamos aquí, un algoritmo novedoso, determinista y correcto para la revisión de creencias de ϕ en *K*. Denotamos nuestro operador de revisión como: $K' = K \circ \phi$. Proponemos un nuevo operador binario lógico Ind entre formas conjuntivas, y tal que $Ind(\phi, K)$ construye también una nueva forma conjuntiva. El operador $Ind(\phi, K)$ trabaja construyendo cláusulas independientes con las cláusulas de K, y las asignaciones falsificantes de la fórmula resultante cubren exactamente el espacio de asignaciones de $Fals(\phi) - Fals(K)$, lo que es esencial para realizar el proceso de revisión de creencias $K' = K \circ \phi$, y donde $K' \models \phi$. Además de que nuestra propuesta satisface los postulados KM. Presentamos también la demostración de que nuestro algoritmo de revisión de creencias es correcto, y su análisis de complejidad en tiempo.

Palabras clave. Inferencia proposicional, revisión de creencias, inferencia basada en modelos, postulados KM.

Model-based Algorithm for Belief Revisions between Normal Conjunctive Forms

Abstract. Belief revision is a central area in knowledge representation and processing of automated reasoning. We will consider a knowledge base (KB) K and a new information ϕ, both expressed in conjunctive form (CF). We present here, a novel, deterministic and correct algorithm for belief revision of ϕ in K. We denote our revision operator as: $K' = K \circ \phi$. We introduce a new logical binary operator Ind between two conjunctive forms, such that $Ind(\phi, K)$ generates also a conjunctive form. The operator $Ind(\phi, K)$ works building independent clauses with the clauses of K, and whose falsifying assignments of the resulting formula cover exactly the space of assignments $Fals(\phi) - Fals(K)$, this is essential for performing the process of belief revision $K' = K \circ \phi$, where $K' \models \phi$. Furthermore, our proposal satisfies the KM postulates. We also present the correctness proof of our belief revision method, and the analysis of its time complexity.

Keywords. Propositional inference, belief revision, model based inference, postulates KM.

1. Introducción

Un marco de referencia ampliamente aceptado en el área del razonamiento en sistemas inteligentes es el enfoque de sistemas basados en bases de conocimiento. La idea general de estos sistemas es mantener el conocimiento en algún lenguaje de representación con una connotación bien definida. En este caso, las sentencias se almacenan en una base de conocimiento (KB – por sus siglas en inglés) provista de un mecanismo de razonamiento [9].

Un reto fundamental de estos sistemas es la automatización del razonamiento deductivo a partir de la KB. El razonamiento deductivo proposicional es generalmente resumido como sigue: dada una KB K, que contiene el conocimiento acerca de un dominio ("el mundo"), y una sentencia ϕ que representa la consulta que captura la situación actual, ambas expresadas en lógica proposicional, el objetivo es decidir si K implica ϕ (en símbolos: $K \models \phi$), lo que se conoce como el *problema de implicación proposicional*.

La implicación proposicional es una tarea relevante en problemas como: estimar el grado de creencia, revisión y actualización de creencias, al trabajar con explicaciones abductivas, y en muchos otros procedimientos en aplicaciones de la Inteligencia Artificial (AI, por sus siglas en inglés), por ejemplo, al trabajar en planeación, diseño de sistemas multiagentes, diagnóstico lógico, razonamiento aproximado, entre otras aplicaciones [7, 13]. En general, el problema de la implicación lógica es un reto difícil en el área de razonamiento automático, y resulta ser un problema Co-NP difícil, incluso en el caso proposicional [17].

La revisión de creencias consiste en incorporar nuevas creencias a una base de conocimiento (KB) ya establecida, cambiando lo menos posible las creencias originales y manteniendo la consistencia de la KB. La función básica de la revisión de creencias es ofrecer un método de cómo cambiar una base de conocimiento cuando nos enfrentamos con nueva información ϕ. La nueva información puede entrar en conflicto con la que teníamos antes, y en ese caso, si queremos mantener consistencia, se deberán eliminar algunos elementos previos de la KB.

Es deseable que los cambios en la base original de conocimiento no se efectúen de cualquier manera, sino de forma racional. En la revisión de creencias se trata por tanto, de una teoría normativa que nos indica en cada caso, cuál es la manera óptima de proceder.

En cierto sentido, la revisión de creencias propone una teoría formal, cuyos efectos podemos interpretar de diferente manera. Las creencias pueden aludir a entidades mentales de un agente, a elementos de una base de conocimiento del mundo real, o tal vez, a elementos de un problema en la teoría de la decisión. Esta característica proporciona su versatilidad a la teoría de revisión de creencias [5].

Al proceso de revisión de creencias lo donotaremos con el operador ($\circ$) que actúa sobre una KB K y sobre una nueva evidencia ϕ, para formar una nueva KB K' = $K \circ \phi$. Cuando $K \models \phi$ entonces K' = $K \circ \phi$ = K. Cuando $K \not\models \phi$, la idea de la revisión de creencias es formar una nueva KB K' a partir de la previa K, que permita inferir la nueva evidencia ϕ y al mismo tiempo, minimice la pérdida de información de K [9], [12].

Aunque la inferencia proposicional es un problema Co-NP-completo en su versión general [22], existen también algunos casos que se pueden resolver de manera eficiente [17]. Consideraremos una KB K = $\bigwedge_{(j=1)}^{m} C_j$ y ϕ = $\bigwedge_{(i=1)}^{k} \varphi_i$, donde cada $C_j \in K$ y cada $\varphi_i \in \phi$ son cláusulas expresadas bajo un mismo conjunto de n variables Booleanas. Este artículo muestra que el uso de patrones falsificantes de las cláusulas, ayuda a determinar si una FC se infiere de otra FC, y por tanto, a construir un algoritmo para la revisión de creencias entre formas normales conjuntivas.

Es común el uso de FC's en el proceso de razonamiento automático, ya que el procedimiento de resolución ha abierto un área de relevancia práctica para revisar consistencia entre FC's. Aunque es bien conocido que la inferencia basada en el principio de resolución tiene limitaciones intrínsecas [3]. Al tener un método efectivo de revisión de creencias entre FC's, su extensión para considerar otras formas normales no es difícil, dado que cualquier fórmula Booleana se puede expresar en forma conjuntiva.

1.1. Estado del arte

El enfoque más conocido para realizar la revisión de creencias es el paradigma AGM (debido a las iniciales de los autores Alchourrón, Gärdenfors and Makinson) [1]. La teoría AGM propone un conjunto de postulados racionales, que cualquier operador de revisión debe satisfacer

Posterior a la propuesta AGM, Alchourrón y Makinson [14] desarrollaron un modelo constructivo para funciones de cambio llamado "contracción segura" (safe contraction) que después fue generalizada por Hansson [15]. Sin embargo, la mayoría de estas propuestas requieren de información adicional, tales como: relaciones de afianzamiento epistémicas, sistemas de esferas, relación de subfórmulas, entre otras [14].

Algunos de los problemas con las propuestas anteriores, es que muchas veces esta información no se tiene, o bien no existe, obligando a tratar todas las creencias por igual. Más aún, hay otras propuestas cuyos operadores de revisión de creencias son completamente dependientes de la sintaxis.

Katsuno y Mendelzon unificaron los diferentes enfoques semánticos de revisión de creencias, y reformularon los postulados AGM, a los que llamaron ahora postulados KM. Además, propusieron un teorema de representación que caracteriza las operaciones de revisión en términos de pre-órdenes totales sobre el conjunto de interpretaciones [16].

Posteriormente, Darwiche y Pearl [8] propusieron postulados para una revisión de forma iterada, donde caracterizan la revisión de creencias como un proceso que puede depender de elementos de un estado que no necesariamente son capturados por un conjunto de creencias. Su propuesta establece una representación basada en el modelo que representa los postulados y las restricciones para la revisión de creencias. Otra propuesta similar a la de Darwiche y Pearl es la presentada por Lehmann [18], donde cada observación es una sentencia general que se asume es consistente.

Hay algunas propuestas de revisión de creencias basadas en modelos, y que se identifican por el nombre de sus autores; Dalal, Satoh, Winslett, Borguida y Forbus [20]. Por ejemplo, Dalal [6] sugiere un operador de revisión basado en la distancia mínima Hamming entre las interpretaciones y la cual se extiende a distancias entre interpretaciones y bases. En la práctica, esta propuesta implica un cálculo de modelos que pueden ser muy costoso, otro de los inconvenientes del enfoque Dalal es que se limita al caso de bases de conocimientos consistentes.

Por otro lado, la propuesta de Satoh [25] es similar a la de Dalal, con la diferencia de que la distancia entre dos modelos es definida como el conjunto de literales a las que les son dados diferentes valores. En el caso de Winslett, la propuesta se basa en la comparación entre todos los sistemas consistentes de longitud máxima.

La propuesta de Borguida y Forbus es similar a la de Winslett, con la diferencia de que Borguida considera modelos incompatibles, y Forbus utiliza la distancia Hamming. La similitud entre los modelos es definida a través de un conjunto conteniendo todas las subfórmulas máximales y consistentes con la consulta realizada, lo que lleva a una búsqueda exhaustiva sobre la tautologicidad de una gran cantidad de subfórmulas, cuestionando así no sólo la funcionalidad de cada uno de los métodos, sino que nos lleva también a concluir que el problema de revisión de creencias bajo estos métodos es un problema inherentemente intratable (de complejidad exponencial en tiempo).

Más recientemente, la revisión de creencias ha ganado atención en el marco de la lógica simbólica, y numerosos operadores de revisión de creencias han sido propuestos de acuerdo a puntos de vista sintácticos o semánticos [19, 21, 25], obteniéndose diferentes resultados sobre la complejidad en tiempo de estas propuestas [6, 19]. Algunas de las investigaciones en esta dirección se han acotado principalmente a considerar el fragmento Horn dentro de la lógica clásica.

Existen diversas propuestas que involucran fórmulas proposicionales tales como las descritas en [10, 26, 2] que sugieren métodos que abordan sólo fragmentos de la lógica proposicional. Una de las propuestas recientes debida a Delgrande presenta los primeros resultados sobre cambio de creencias en el fragmento Horn [10].

En [4], los autores presentan una metodología general para definir nuevos operadores de revisión derivados de operadores estándar (como por ejemplo, los operadores de Dalal y Satoh), tal que el resultado de la revisión se mantiene en el fragmento en cuestión. Por lo tanto, en esta propuesta los autores no se limitan sólo al caso Horn, sino que ésta es aplicable a otros fragmentos de la lógica proposicional, donde los modelos de las fórmulas cumplan el ser cerrados bajo una función Booleana.

Se puede notar que estas propuestas se desarrollan para considerar algún tipo de fórmula normalizada o bien sólo consideran cláusulas de Horn. Cada una de estas propuestas propone un operador de revisión que trabaja sobre sus fórmulas normalizadas y que presentan diversos inconvenientes. Por ejemplo, algunos de los operadores son dependientes de la sintaxis o bien requieren de información adicional. Otros definen diferentes nociones de proximidad, y unos más se limitan a revisar sólo bases consistentes. Pero el principal inconveniente de estos métodos de revisión de creencias es que conllevan inherentemente a realizar procesos de complejidad exponencial en tiempo.

Por otro lado, nuestra propuesta de revisión de creencias sigue el enfoque basado en revisar los modelos de las fórmulas involucradas, porque supondremos que la KB K y la fórmula de consulta ϕ son FC's. Trabajar con FC's permite el cálculo efectivo del conjunto de asignaciones que falsifican a cada una de sus cláusulas, y por tanto, también del conjunto complemento de tales asignaciones falsificantes, que serán precisamente, los modelos de las fórmulas.

Mostramos aquí, que el proceso de revisión de creencias puede realizarse de forma práctica en función de las asignaciones falsificantes de las FC's involucradas. Además, nuestra propuesta obtiene una nueva base de conocimiento $K' = K \circ \phi$, que reduce de forma mínima el conjunto de modelos de la base de conocimiento original K.

Resumiendo, las contribuciones principales de nuestro trabajo son:

- Proponemos un método que trabaja sobre el conjunto de asignaciones falsificantes de las fórmulas involucradas, para revisar: $K \models \phi$.
- Introducimos un operador lógico entre dos cláusulas, $Ind(\varphi_i, C_j)$, cuyo resultado es una FC Fs, tal que $Fals(Fs) = Fals(\varphi_i) - Fals(C_j)$.
- Demostramos que nuestra propuesta de revisión de creencias es correcto, y cumple los postulados de Katsuno y Mendelzon.
- El operador $Ind(\varphi_i, C_j)$ se implementó para que trabaje en tiempo lineal sobre el número de variables involucradas, y es la base de nuestro proceso de revisión de creencias.
- A pesar de la eficiencia del operador $Ind(\varphi_i, C_j)$, el número de cláusulas en Fs para que $(K \wedge Fs) \models \phi$, puede llevarnos a un crecimiento exponencial de orden $O(|K| \cdot n \cdot 2^{(n-min\{|\varphi_i|:\varphi_i \in \phi\})})$.

2. Preliminares

Sea $X = \{x_1, \ldots, x_n\}$ un conjunto de n variables Booleanas. Una $literal$ denotada como lit es, o una variable x_i o una variable negada $\neg x_i$. Como es usual, para cada $x \in X$, $x^0 = \neg x$ y $x^1 = x$. Sean $\perp$ and $\top$ dos constantes representando los valores lógicos *falso* y *verdadero*, respectivamente.

Una $cláusula$ es una disyunción de diferentes literales. Para $k \in N$, una k-$cláusula$ es una cláusula que consiste exactamente de k literales, y una $(\leq k)$-$cláusula$ es una cláusula con a lo más k literales. Una $frase$ es una conjunción de literales. Una k-$frase$ es una frase con exactamente k literales. Una variable $x \in X$ aparece en una cláusula (frase) C si x o $\neg x$ es un elemento de C.

Una *forma normal conjuntiva* (FC) es una conjunción de cláusulas (que también llamaremos forma conjuntiva), y una k-FC es una FC que contiene sólo k-cláusulas.

Una *forma normal disyuntiva* (FD) es una disyunción de frases, y una k-FD es una FD que contiene sólo k-frases. Una FC F con n variables representa una función Booleana n-$aria$ F:$\{0, 1\}^n \rightarrow$ {0, 1}. Por el contrario, cualquier función Booleana F tiene infinitamente muchas representaciones equivalentes, entre estas, algunas en FC y también otras en FD.

Denotamos con Y a cualquiera de los elementos lógicos básicos que estamos utilizando, como: una literal, una cláusula, una frase, una FD o una FC, y $v(Y)$ denota el conjunto de variables involucradas en el objeto Y. Por ejemplo, $v(\neg x_1 \vee x_2)$ = $\{x_1, x_2\}$. Mientras $lit(Y)$ denota el conjunto de literales involucradas en el objeto Y. Por ejemplo, si $X = v(Y)$ entonces $lit(Y)$ = $X \cup \neg X$ = $\{x_1, \neg x_1, \ldots, x_n, \neg x_n\}$. También usamos $\neg Y$ como el operador de negación sobre el objeto Y. Denotaremos a {1, 2, 3, ...,n} por [[1, n]], y a la cardinalidad de un conjunto A por $|A|$.

Una $asignación$ s para una fórmula F es un mapeo Booleano $s : v(F) \rightarrow \{1, 0\}$. Una asignación s puede también ser considerada como un conjunto de literales no complementarias: $l \in s$ sí y sólo si s asigna l a cierto y $\neg l$ a falso. s es una $asignación\ parcial$ para la fórmula F cuando s ha determinado un valor lógico sólo para las variables de un subconjunto propio de F, a saber

ISSN 2007-9737

$s : Y \to \{1,0\}$ y $Y \subset v(F)$. Si s tiene valores lógicos determinados para todas las variables en F entonces s es una *asignación total* de F.

Si $F_1 \subset F$ es una FC que consiste de algunas cláusulas en F, y $v(F_1) \subset v(F)$, una asignación sobre $v(F_1)$ es una asignación parcial sobre $v(F)$. Considerando $n = |v(F)|$ y de igual forma $n_1 = |v(F_1)|$, cualquier asignación sobre $v(F_1)$ tiene 2^{n-n_1} extensiones sobre $v(F)$.

Considerando una cláusula C y una asignación s como conjuntos de literales, C es satisfecha por s si $s \cap C \neq \emptyset$, de otra manera s contradice (o falsifica) a C. Una FC F es satisfecha por una asignación s si cada cláusula en F es satisfecha por s; F es contradicha por s si alguna cláusula en F es falsificada por s. Un modelo de F es una asignación sobre $v(F)$ satisfaciendo F.

Una frase f es satisfecha por una asignación s si $f \subseteq s$, de otra manera s falsifica a f. Una FD F es satisfecha por s si alguna frase en F es satisfecha por s. F es contradicha por s si todas las frases en F son falsificadas por s.

Dada una fórmula F, sea $S(F)$ el conjunto de todas las posibles asignaciones definidas sobre $v(F)$. Si $n = |v(F)|$ entonces $|S(F)| = 2^n$. $s \models F$ denota que la asignación s es un modelo de F (s satisface a F). $s \not\models F$ denota que s es una asignación falsificante de F. Denotamos por $SAT(F)$ al conjunto de asignaciones en $S(F)$ que son modelos de F. $Fals(F)$ denota el conjunto de asignaciones de $S(F)$ que falsifican a F.

Dadas dos fórmulas Booleanas F y G definidas sobre un mismo conjunto de variables, esto es $v(F) = v(G)$, decimos que *F es consecuencia lógica de G*, denotado como $G \models F$, si para toda asignación s que satisface a G se cumple que s también satisface a F. Y diremos que F y G son *logicamente equivalentes*, denotado como $G \equiv F$, si ambas fórmulas tienen el mismo conjunto de modelos, esto es, una asignación s satisface a G si y sólo si s satisface a F. En términos de los conjuntos de modelos, podemos denotar que: $G \models F$ si y sólo si $SAT(G) \subseteq SAT(F)$, y que $G \equiv F$ si y sólo si $SAT(F) = SAT(G)$.

$\#SAT(F)$ denota el número de asignaciones de $S(F)$ que satisfacen a la fórmula F. Mientras que $\#Fals(F)$ representa al número de asignaciones de $S(F)$ que no satisfacen a F.

Para cualquier fórmula proposicional F, se cumple que: $S(F) = SAT(F) \cup Fals(F)$. El problema SAT consiste en decidir, para una fórmula de entrada F, si F es satisfactible, esto es, si F tiene un modelo o no. Una base de conocimiento KB es un conjunto K de fórmulas. Dada una KB K y una fórmula proposicional ϕ, decimos que K implica ϕ, y escribimos $K \models \phi$, si ϕ es satisfecha por todo modelo de K, es decir, $SAT(K) \subseteq SAT(\phi)$.

3. Inferencia entre formas conjuntivas

Un problema fundamental en el razonamiento deductivo es el problema de la implicación lógica: dada una KB K y una fórmula ϕ, debemos decidir si $K \models \phi$. En este trabajo analizaremos la complejidad computacional del caso del problema de implicación entre formas conjuntivas: FC $\models$ FC.

Sea una base de conocimiento K que se encuentra en forma conjuntiva, $K = \bigwedge_{(j=1)}^{m} C_j$ y sea una consulta expresada también en FC $\phi = \bigwedge_{(i=1)}^{k} \varphi_i$, donde cada $C_j \in K$ y cada $\varphi_i \in \phi$ son cláusulas expresadas bajo un mismo conjunto de n variables Booleanas.

Dadas dos FC's F_1 y F_2, decidir si $F_1 \models F_2$ es lógicamente equivalente a probar que $\neg F_2 \models \neg F_1$, que resulta en revisar la inferencia entre formas disyuntivas, ya que si F es una FC entonces $\neg(F)$ es una FD, debido a que negar una FC F se realiza en tiempo lineal sobre el tamaño de F, a través de una generalización de las reglas de De Morgan.

Por otro lado, revisar la inferencia entre formas disyuntivas se reduce a revisar si una FD G es una tautología, lo que es un problema clásico en la clase de complejidad Co-NP completo [22]. La reducción proviene de considerar que la existencia de un procedimiento que determina si $G_1 \models G_2$, con G_1 y G_2 formas disyuntivas, permite a su vez, determinar la tautologicidad de cualquier forma disyuntiva G, ya que basta con hacer $G_1 \equiv \top$ que es una tautología, y entonces $G_1 \models G$ se cumplirá sólo si G es a su vez una tautología.

Como K y ϕ están en FC, las cadenas falsificantes de sus cláusulas $Fals(K)$ y $Fals(\phi)$ se pueden calcular eficientemente [9]. Usar las cadenas falsificantes es la base para revisar si $K \models \phi$, lo que en términos de sus asignaciones equivale a revisar si $SAT(K) \subseteq SAT(\phi)$, o bien que: $Fals(\phi) \subseteq Fals(K)$. El resultado de aplicar el operador de revisión de creencias sobre la KB K y la nueva evidencia ϕ es denotado como K' = $K \circ \phi$. Cuando $K \models \phi$ entonces K' = $K \circ \phi$ = K.

Si $K \not\models \phi$ entonces $Fals(\phi) \not\subset Fals(K)$, lo que implica que existe un conjunto de asignaciones S tal que $S \subseteq Fals(\phi)$ y $S \not\subseteq Fals(K)$. Si $K \not\models \phi$, entonces S = ($Fals(\phi)$ - $Fals(K)$) $\neq \emptyset$. En este caso, nuestro método de revisión de creencias trabaja construyendo tal conjunto S, lo que permite construir una nueva FC Fs, tal que $S = Fals(Fs)$ y $K' = K \wedge Fs$, cumple que: $K' \models \phi$.

El método que proponemos obtiene $S = (Fals(\phi) - Fals(K))$ como un conjunto de cadenas falsificantes, lo que nos lleva a construir de forma directa una FC Fs, donde $S = Fals(Fs)$ y tal que K'= $K \wedge Fs$ es una nueva FC con menos información que K (dado que K' tiene más cláusulas que K), de hecho, se cumple que si $S \neq \emptyset$ entonces $Fals(K) \subset Fals(K')$, y por tanto, $SAT(K') \subset SAT(K)$.

3.1. Construcción de conjuntos independientes de cláusulas

Dada una forma conjuntiva K = $\bigwedge_{(i=1)}^{m} C_i$, con n = $|v(K)|$, para cualquier cláusula $C_i \in K$, hay exactamente $2^{(n-|C_i|)}$ asignaciones de $S(K)$ falsificando C_i. Debido a que todas las falsificaciones de C_i tiene valores fijos en las posiciones de las variables $v(C_i)$ y tales valores falsifican cada literal de C_i. Por tanto, hay n - | C_i | variables a las que se les puede asignar cualquier valor de verdad. Esto significa que hay $2^{(n-|C_i|)}$ asignaciones falsificantes para C_i.

Sea A_i un conjunto de cadenas tales que la longitud de cada cadena es n. El valor en la j-ésima posición de la cadena A_i, $1 \leq j \leq n$ representa el valor de verdad de x_j que falsifica C_i. Es decir, si $x_j \in C_i$ entonces el j-ésimo elemento de cualquier cadena en A_i es 0. De otra manera si $\neg x_j \in C_i$ entonces el j-ésimo elemento es 1.

Usaremos el símbolo $*$ para representar los elementos que pueden tomar cualquier valor de verdad en las cadenas A_i. Por ejemplo, si $K = \{C_1, .., C_m\}$ es una 2-FC, $n = |v(K)|$, C_1 = $\{x_1, x_2\}$ y $C_2 = \{x_2, \neg x_3\}$ entonces se representa A_1 como 00**$\cdots$* y A_2 como *01*$\cdots$*. Este abuso de notación nos permitirá dar una representación concisa y clara en el resto del documento, considerando a las cadenas A_i como patrones que representan las falsificaciones de la cláusula C_i. A tales cadenas las llamaremos *cadenas falsificantes* de una cláusula.

Definición 1 *[11] Dadas dos cláusulas C_i y C_j, si ellas tienen al menos una literal complementaria, se les llamará cláusulas independientes. En otro caso, se dice que ambas son cláusulas dependientes.*

Definición 2 *Sea $K = \{C_1, C_2, \ldots, C_m\}$ una FC. K es llamada independiente si para cualquier par de cláusulas $C_i, C_j \in K$, $i \neq j$, se cumple la propiedad de independencia.*

Definición 3 *Dadas dos cadenas falsificantes A y B, ambas de la misma longitud, si hay una i tal que $A[i] = x$ y $B[i] = 1 - x$, $x \in \{0, 1\}$, se dice que tienen la propiedad de independencia. En otro caso, decimos que ambas cadenas son dependientes.*

Sea C una cláusula cualquiera, para cualquier variable x se cumple que:

$$C = (C \vee \neg x) \wedge (C \vee x). \tag{1}$$

Además, esta reducción conserva el número de asignaciones falsificantes, ya que para cualquier par de cláusulas independientes C_i, C_j se cumple que $Fals(C_i) \cap Fals(C_j) = \emptyset$ y entonces $\#Fals(C)$ = $2^{(n-|c|)}$ = $2^{(n-(|c|+1))}$+$2^{(n-(|c|+1))}$ = $\#Fals((C \vee \neg x) \wedge (C \vee x))$, porque $(C \vee \neg x)$ y $(C \vee x)$ son cláusulas independientes.

La conjunción de un par de cláusulas dependientes C_1 y C_2 puede expresarse mediante una conjunción de cláusulas independientes. Supongamos que hay literales en C_1 que no están en C_2, sea $L = \{x_1, x_2, \ldots, x_p\}$ tales literales. Esto es, $L = lit(C_1) - lit(C_2)$. Existe una reducción para transformar C_2 (o C_1) como cláusula independiente con C_1 (o C_2) llamada reducción de independencia, y que trabaja de la siguiente manera.

Por (1) se puede escribir:
$C_1 \wedge C_2 = C_1 \wedge (C_2 \vee \neg x_1) \wedge (C_2 \vee x_1)$. Ahora C_1 y $(C_2 \vee \neg x_1)$ son independientes. Aplicando (1) a $(C_2 \vee x_1)$: $C_1 \wedge C_2 = C_1 \wedge (C_2 \vee \neg x_1) \wedge (C_2 \vee x_1 \vee \neg x_2) \wedge (C_2 \vee x_1 \vee x_2)$. Las primeras tres cláusulas son independientes. Repitiendo la reducción de independencia hasta x_p, se tiene que $C_1 \wedge C_2$ puede expresarse como:
$C_1 \wedge (C_2 \vee \neg x_1) \wedge (C_2 \vee x_1 \vee \neg x_2) \wedge \ldots \wedge (C_2 \vee x_1 \vee x_2 \vee \ldots \vee \neg x_p) \wedge (C_2 \vee x_1 \vee x_2 \vee \ldots \vee x_p)$.
La última cláusula contiene todas las literales de C_1, así que puede eliminarse porque es subsumida por la cláusula C_1, obteniéndose que:

$$C_1 \wedge C_2 = C_1 \wedge (C_2 \vee \neg x_1) \wedge (C_2 \vee x_1 \vee \neg x_2) \wedge \ldots \wedge (C_2 \vee x_1 \vee x_2 \vee \ldots \vee \neg x_p). \quad (2)$$

Las cláusulas del lado derecho de la ecuación (2) son independientes por construcción.

El operador central para revisar inferencia entre FC's es un operador de independencia que trabaja sobre dos cláusulas φ y C, y que construye un conjunto de cláusulas independientes equivalentes a $\varphi \wedge C$. Sea $L = \{x_1, x_2, \ldots, x_p\} = lit(C) - lit(\varphi)$ se define el operador de independencia entre φ y C como sigue:

$$Ind(\varphi, C) = \begin{cases} \varphi \text{ Si } \varphi \text{ y C son independientes} \\ \emptyset \text{ Si } lit(C) - lit(\varphi) = \emptyset \\ (\varphi \vee \neg x_1) \wedge \ldots \wedge (\varphi \vee x_1 \vee \ldots \\ \vee \neg x_p), \text{ en otro caso.} \end{cases}$$

La complejidad en tiempo para ejecutar $Ind(\varphi, C)$, que denotaremos como $T_{Ind}(|\varphi|, |C|)$, depende directamente del tiempo para ejecutar operaciones básicas entre conjuntos de literales. Por ejemplo, la operación $lit(C) - lit(\varphi)$ podría realizarse como: para cada $x \in lit(C)$ revisar si $x \in lit(\varphi)$ o si $\neg x \in lit(\varphi)$, lo que requiere de a lo más $|C| * |\varphi)| \leq n^2$ operaciones de comparación. Si los conjuntos $lit(C)$ y $lit(\varphi)$ se representan mediante arreglos de n posiciones (fijando una posición para cada una de las n posibles variables), entonces $lit(C) - lit(\varphi)$ se realizará en a lo más $O(n)$ operaciones lógicas entre las posiciones de ambos arreglos.

Por otro lado, cuando $L = lit(C) - lit(\varphi) \neq \emptyset$, se realiza un ciclo de $|L| \leq (n-1)$ iteracciones, y en cada iteracción i se agrega una disyunción y una negación para formar $(\varphi \vee x_1 \vee \ldots \vee \neg x_i)$, y a través de una conjunción se adiciona esta cláusula a la FC que se esta construyendo. Esto nos lleva a un proceso, en el peor caso, de orden $O(n)$ para construir $Ind(\varphi, C)$.

Veamos como este operador de independencia $Ind(\varphi_i, C_j)$ entre cláusulas $\varphi_i \in \phi$ y $C_j \in K$ es la base para realizar la revisión de creencias entre K y ϕ.

4. Revisión de Creencias entre formas conjuntivas

Nuestro método de Revisión de Creencias se basa en las siguientes dos propiedades:

1. Si $\forall\ s \in Fals(\phi)$ se cumple que $s \in Fals(K)$, entonces $K \models \phi$.
2. Si $\exists\ s \in Fals(\phi)$, y $s \notin Fals(K)$, entonces $K \not\models \phi$.

El primer caso considera que todas las asignaciones de $Fals(\phi)$ están en el conjunto $Fals(K)$, lo que demostraría que $K \models \phi$. Y en este caso $K' = K$, ya que no se necesita cambiar la KB K.

En el segundo caso, se detectarán los conjuntos de asignaciones S tal que $\exists \varphi \in \phi$, $S \subseteq Fals(\varphi)$ y $S \not\subseteq Fals(K)$. Para construir estos conjuntos S se inicia con la cadena A_1 que representa a $Fals(\varphi_1)$ y se aplica el operador Ind con cada una de las cadenas B_j que representan $Fals(C_j)$, $j = 1, \ldots, m$.

La operación $Ind(\varphi_i, C_j)$ forma una cadena que representa el conjunto de asignaciones falsificantes: $Fals(\varphi_i) - Fals(C_j)$. Esto es, $Ind(\varphi_i, C_j)$ determina las asignaciones que están en $Fals(\varphi_i)$ pero que no están contenidas en $Fals(C_j)$. Si se aplica la operación Ind sobre todo $C_j \in K$, obtendremos como resultado el conjunto $S \subseteq Fals(\varphi_i) \wedge S \not\subset Fals(K)$.

El conjunto S permite construir una FC Fs_i, $Fs_i = (D_1 \wedge D_2 \wedge \ldots D_t)$, donde $S = Fals(Fs_i)$. Al agregar las nuevas cláusulas de Fs_i a K, obtenemos una nueva KB $K'_i = K \wedge Fs_i$, que

ISSN 2007-9737

cumple que: $K'_i \models \varphi_i$, y además, K'_i sigue siendo una FC. Presentamos algorítmicamente este proceso.

Algorithm 1 Procedure $Ind(\varphi_i, K)$

Input: K: Una KB, y φ_i: cláusula con nueva inf.
Push(φ_i, V); $Fs = \emptyset$; {Salida en Fs una FC (conjunto de cláusulas)}
for all $C_j \in K$ **do**
  while $(V \neq \emptyset)$ **do**
    φ = Pop(V); {Prueba cláusula sgte.}
    $Fs = Fs - \varphi$; {quitar cláusula de la salida}
    $Nc = Ind(\phi, C_j)$; {Forma: $Nc \wedge C_j \models \phi$ }
    if $(Nc \neq \emptyset)$ **then**
      $Fs = Fs \cup Nc$; {Sólo si hay cláusulas a agregar}
    end if
  end while
  $V = Fs$; {Sgte. iteración considera nuevas cláusulas}
end for
Returns(Fs)

$Ind(\varphi_i, K)$ consiste de dos ciclos, uno externo sobre $C_j \in K$, de orden $O(|K|)$. Este ciclo (el For) recorre las columnas de una tabla donde se irán colocando los resultados de $Ind(\varphi_i, C_j)$.
El cuerpo del ciclo interno consiste esencialmente de realizar el operador $Ind(\varphi_i, C_j)$ que es de orden $O(n)$, y de realizar ajustes a la FC Fs_i que inicia con la cláusula φ_i y que involucra no más de $O(n)$ operaciones.

El número de filas de la tabla se va ajustando de forma dinámica, dependiendo del resultado de $Ind(\varphi_i, C_j)$. En el peor caso, este ciclo sobre el número de filas puede llevarnos a un crecimiento exponencial sobre el número de cláusulas que contiene una Fs_i, como se mostrará en la sección de análisis de complejidad de nuestro método.

Cuando el proceso $Ind(\varphi_i, K)$ itera sobre toda $\varphi_i \in \phi$, se forman las cláusulas Fs_i tal que $Fals(\bigcup(Fs_i))$ = $Fals(\phi) - Fals(K)$. Al adicionar a K el conjunto de cláusulas $\bigcup(Fs_i)$, se forma una nueva KB K' tal que $K' \models \phi$, puesto que $Fals(\phi) \subseteq Fals(K')$, y por tanto, $SAT(K') \subseteq SAT(\phi)$.

Ejemplo 1. En todos los ejemplos a presentar, supondremos un ordenamiento alfanumérico sobre el conjunto de variables que se utilizan. Sea K = $(\neg p \vee q \vee s) \wedge (\neg q \vee \neg r \vee s) \wedge (\neg q \vee r \vee \neg s) \wedge (\neg p \vee \neg q \vee r)$ y $\phi = (\neg p \vee \neg r) \wedge (\neg q \vee r) \wedge (p \vee q \vee \neg r \vee \neg s) \wedge (\neg t)$. Probar que $K \models \phi$, es equivalente a revisar que: $Fals(\phi)$ = {1*1**, *10**, 0011*, ****1} $\subseteq Fals(K) =$ {10*0*, *110*, *101*, 110**}. En cada celda de las columnas 2 en adelante de la tabla 1, se va mostrando el resultado de $Ind(\varphi_i, C_j)$.

Dadas 2 cláusulas C_i, C_j que difieren en el signo de sólo una variable, la reducción por literal complementaria genera una sóla cláusula de $C_i \wedge C_j$, de hecho, la reducción se basa en la aplicación de la ecuación (1).

Por ejemplo, sea C_i = $(x \vee q)$, y C_j = $(\neg x \vee q)$, entonces $C_i \wedge C_j = (q)$. En términos de las cadenas falsificantes de las cláusulas, denotaremos tal reducción como: $Varcom(A_i, A_j)$. En el caso de nuestro ejemplo, se tiene que: $Varcom(1111*, 1011*) = 1*11*$.

Es relevante aplicar la operación de reducción por literales complementarias sobre las cadenas en S, para así minimizar el número total de cláusulas.

Aplicando la reducción $Varcom$ y eliminando cláusulas subsumidas al resultado del ejemplo 1, se tiene que: S = {1*11*, 0100*, 0011*, 00**1, *1111, 10*11}. Escribiendo S como una FC, $Fs =$ $(\neg p \vee \neg r \vee \neg s)$ $\wedge(p \vee \neg q \vee r \vee s) \wedge (p \vee q \vee \neg r \vee \neg s) \wedge$ $(p \vee q \vee \neg t) \wedge (\neg q \vee \neg r \vee \neg s \vee \neg t) \wedge (\neg p \vee q \vee \neg s \vee \neg t)$. Y así, la nueva KB K' = $K \wedge Fs$ es una FC que cumple: $K' \models \phi$.

Tabla 1. Construyendo $Ind(\phi, K)$

ϕ \ K	10*0*	*110*	*101*	110**	***S***
1*1**	111** 1011*	1111* 1011*	1111* 1011*	1111* 1011*	**1111*** **1011***
*10**	*10**	*10**	*100*	0100*	**0100***
0011*	0011*	0011*	0011*	0011*	**0011***
****1	0***1	00**1 010*1 01111	00**1 01001 01111	00**1 01001 01111	**00**1** **01001** **01111**
	11**1	110*1 11111	11001 11111	$\emptyset$ 11111	$\emptyset$ **11111**
	10*11	10*11	10*11	10*11	**10*11**

4.1. Propiedades del método de Revisión de Creencias

El conjunto de cláusulas construido mediante $Ind(\varphi_i, K)$ se agrega a la KB original K, y así, cada $\varphi_i \in \phi$ se infiere de $K \wedge Ind(\varphi_i, K)$. Esto se demuestra en el siguiente teorema sobre la corrección de nuestro método.

Teorema 1 *Dadas dos cláusulas φ_i y C_j se cumple que ($C_j \wedge Ind(\varphi_i, C_j)$) $\models \varphi_i$.*

Demostración.

Si φ_i y C_j son cláusulas independientes, entonces φ_i = $Ind(\varphi_i, C_j)$ y por tanto $(C_j \wedge Ind(\varphi_i, C_j)) = (C_j \wedge \varphi_i)$. Así $(C_j \wedge \varphi_i) \models \varphi_i$, por la propiedad proposicional: $(p \wedge q) \supset q$ y por la reflexividad de la inferencia lógica: $\varphi_i \models \varphi_i$.
Si φ_i y C_j no son independientes, pero $Ind(\varphi_i, C_j)$ = $\emptyset$, esto implica que $Fals(\varphi_i) \subseteq Fals(C_j)$ y por tal $C_j \models \varphi_i$. Como C_j = $(C_j \wedge Ind(\varphi_i, C_j))$, entonces $(C_j \wedge Ind(\varphi_i, C_j)) \models \varphi_i$.
Cuando φ_i y C_j no son independientes, e $Ind(\varphi_i, C_j) \neq \emptyset$, se cumple que $(C_j \wedge Ind(\varphi_i, C_j))$ $\equiv$ $(C_j \wedge \varphi_i)$ por (2), cumpliéndose que: $(C_j \wedge \varphi_i)$ $\models \varphi_i$, por la propiedad proposicional: $(p \wedge q) \supset q$, y por la reflexividad: $\varphi_i \models \varphi_i$.
Así, para cualquiera de los tres posibles resultados de $Ind(\varphi_i, C_j)$, se cumple: $(C_j \wedge Ind(\varphi_i, C_j)) \models \varphi_i$.
■

El conjunto de cláusulas construido mediante $Ind(\varphi_i, C_j)$ contiene exactamente las cláusulas necesarias que permitirán inferir cada $\varphi_i \in \phi$ a partir de $C_j \wedge Ind(\varphi_i, C_j)$. Al iterar $Ind(\varphi_i, C_j)$ sobre todo $C_j \in K$, se obtiene un conjunto de cláusulas con las que se asegura cumplir $(K \wedge Ind(\varphi_i, K)) \models \varphi_i$. El teorema anterior demuestra así la corrección de nuestro método.

Mostremos ahora que el conjunto de cláusulas en $Ind(\varphi_i, C_j)$ representa el conjunto mínimo de cláusulas que permiten cubrir el espacio: $Fals(\varphi_i) - Fals(C_j)$, que es el espacio mínimo necesario de asignaciones para que $Fals(\varphi_i) \subseteq Fals(C_j) \cup Fals(Ind(\varphi_i, C_j))$, y por tanto, para que se cumpla $(C_j \wedge Ind(\varphi_i, C_j)) \models \varphi_i$.

Teorema 2 $Fals(Ind(\varphi_i, C_j))$ = $Fals(\varphi_i) - Fals(C_j)$.
Demostración.
Si $Ind(\varphi_i, C_j)$ = $\emptyset$, se cumple que $Ind(\varphi_i, C_j)$ es el número mínimo de cláusulas que permiten inferir $(C_j \wedge Ind(\varphi_i, C_j)) \models \varphi_i$, ya que de hecho, $C_j \models \varphi_i$.
Supongamos ahora que $Ind(\varphi_i, C_j) \neq \emptyset$. Veamos que $\forall s \in Fals(Ind(\varphi_i, C_j))$ se cumple que $s \in Fals(\varphi_i)$, y $s \notin Fals(C_j)$. Sea $s \in Fals(Ind(\varphi_i, C_j))$, entonces s falsifica a φ_i, ya que cada cláusula en $Ind(\varphi_i, C_j)$ tiene la forma $(\varphi_i \vee R)$, con R una disyunción de literales. Si s falsifica a $(\varphi_i \vee R)$ entonces s falsifica tanto a (φ_i) como a (R), por tanto $s \in Fals(\varphi_i)$. Además, $s \notin Fals(C_j)$, ya que C_j es independiente con cada una de las cláusulas de $Ind(\varphi_i, C_j)$ (por construcción del operador de independencia), y por tal, $s \notin Fals(C_j)$.
■

El teorema anterior demuestra que el operador de independencia $Ind(\varphi_i, C_j)$ construye un conjunto de cláusulas que cubren de forma exacta el espacio de asignaciones que hacen falta para que $Fals(\varphi_i) \subseteq Fals(C_j) \cup Fals(Ind(\varphi_i, C_j))$. Aún más, el conjunto $Fals(Ind(\varphi_i, C_j))$ es el conjunto mínimo de asignaciones para cubrir el espacio $Fals(\varphi_i) - Fals(C_j)$, ya que $Fals(C_j)$ y $Fals(Ind(\varphi_i, C_j))$ son ajenos (por construcción del operador de independencia), y por tanto $Fals(C_j)$ $\cap$ $Fals(Ind(\varphi_i, C_j))$ = $\emptyset$.

Corolario 1 $Fals(Ind(\varphi_i, K)) \subseteq Fals(\varphi_i)$.
Demostración.
Por el teorema (2), se tiene que $Fals(Ind(\varphi_i, C_j)) = Fals(\varphi) - Fals(C_j)$, al iterar sobre cada C_j de K se cumple que $Fals(Ind(\varphi_i, K)) = Fals(\varphi_i) - Fals(K)$. Y por propiedades entre conjuntos, se cumple que $Fals(Ind(\varphi_i, K)) \subseteq Fals(\varphi)$. ■

Al iterar $Ind(\varphi_i, C_j)$ sobre todo $C_j \in K$, se obtiene un conjunto mínimo de cláusulas: Fs_i que asegura que: $(K \wedge Fs_i) \models \varphi_i$.

Al extender K con las cláusulas obtenidas en $Ind(\varphi_i, K)$ se va formando K'. Así K' extiende al conjunto de cláusulas de K, y por tanto, extiende también el espacio inicial de falsificaciones de K, agregando las asignaciones que falsifican a $Ind(\varphi_i, K)$. De hecho, estos dos conjuntos de falsificaciones son excluyentes por construcción de $Ind(\varphi_i, K)$, y por tanto,

$Fals(K) \cap Fals(Ind(\varphi_i, K))$ = $\emptyset$. En otras palabras, el conjunto de modelos de K' es ahora un subconjunto de los modelos de K, $SAT(K') \subseteq SAT(K)$.

Sin embargo, al iterar el operador $Ind(\varphi_i, K)$, sobre cada $\varphi_i \in \phi, i = 1, \ldots, k$, se tiene que los k conjuntos de cláusulas Fs_i formados por $Ind(\phi, K)$ podrían no tener un número mínimo de cláusulas. La reducción $Varcom$ permite reducir el número de cláusulas en $Ind(\phi, K)$.

Así, después de obtener el conjunto de cláusulas $Ind(\phi, K)$, se reduce su cardinalidad, eliminando cláusulas subsumidas y aplicando la reducción $Varcom$ entre cláusulas de dos diferentes conjuntos $Ind(\varphi_{i_1}, K)$ e $Ind(\varphi_{i_2}, K)$.

Este último proceso de reducción de cláusulas a través de literales complementarias y de eliminación de cláusulas subsumidas, se ejecuta en tiempo polinomial (de hecho en tiempo cuadrático) sobre la longitud inicial de $|Ind(\phi, K)|$, ya que consistiría en ir tomando una cláusula $C \in Ind(\phi, K)$, y revisar si es subcláusula (como subconjunto de literales) o si hay una literal complementaria con alguna otra cláusula en $Ind(\phi, K) - C$. Además, el resultado de la reducción mantiene la forma de una FC.

Un proceso similar a $Varcom$ se aplicó en el cálculo de los implicantes primos de una fórmula, presentada por Quine y McCluskey [24]. En esta propuesta, los autores buscan los implicantes primos esenciales que sean necesarios y suficientes para generar la función Booleana de entrada.

Sin embargo, cuando la heurística de éste método recibe una fórmula con un gran número de variables, conduce a resultados no mínimos, por lo que se tiene que recurrir al método de Petrick con el fín de poder caracterizar la expresión mínima de la función Booleana [23].

Ejemplo 2.
Sea K = $(\neg p \vee q \vee r \vee s) \wedge (\neg p \vee \neg q) \wedge (\neg p \vee \neg r)$ y ϕ = $(\neg p \vee \neg s \vee \neg t) \wedge (q \vee \neg r \vee \neg s \vee \neg t) \wedge (\neg p \vee q \vee r \vee \neg s \vee t) \wedge (\neg p)$, probar que $K \models \phi$, es equivalente a revisar si $Fals(\phi)$ = {1**11, *0111, 10010, 1****} $\subseteq Fals(K)$ = {1000*, 11***, 1*1**}. En cada celda de la tabla 2, se va calculando $Ind(\varphi_i, C_j)$.

Tabla 2. Aplicación del operador $Ind(\phi, K)$

ϕ \ K	1000*	11***	1*1**	***S***
1**11	1**11	10*11	10011	**10011**
*0111	*0111	00111	00111	**00111**
10010	10010	10010	10010	**10010**
1****	11***	$\emptyset$	$\emptyset$	$\emptyset$
	101**	101**	$\emptyset$	$\emptyset$
	1001*	1001*	1001*	**1001***

Tabla 3. Cálculo de Ind con las $C_i \in K$ ordenadas

φ_1 \ K	1*1**	11***	1000*	***S***
1**11	1*011	10011	10011	**10011**
φ_2 \ K	1000*	11***	1*1**	***S***
*0111	00111	00111	00111	**00111**
φ_3 \ K	11***	1*1**	1000*	***S***
10010	10010	10010	10010	**10010**
φ_4 \ K	1*1**	11***	1000*	***S***
1****	1*0**	100**	1001*	**1001***

Como se puede observar en la tabla 2, al aplicar el operador $Ind(\phi, K)$ se genera un número de cadenas mayor a las que aparecen en la tabla 3, debido a que en la tabla 3, antes de aplicar el operador de independencia, se ordenan las cláusulas $C_i \in K$, de acuerdo al tamaño $|lit(C_j) - lit(\varphi_i)|$ de menor a mayor, dado que el número de literales de C_j diferentes con φ_i determinará el número de cláusulas independientes a generarse, además de descartar con anticipación cadenas que serán subsumidas.

Por tanto, antes de aplicar el operador $Ind(\varphi_i, K)$ es conveniente ordenar las cláusulas en K de acuerdo al valor de cada φ_i que se esté considerando, tal y como se muestra en la tabla 3. Con lo que se obtiene una estrategia de reducción sobre el número de cláusulas independientes a generar. Al aplicar el proceso de reducción de cláusulas vía literales complementarias, se tiene como resultado para el caso del ejemplo 2, que $S = \{1001*, 00111\}$, cuya FC es $Fs = (\neg p \vee q \vee r \vee \neg s) \wedge (p \vee q \vee \neg r \vee \neg s \vee \neg t)$. Así, K' = $K \wedge Fs = (\neg p \vee q \vee r \vee s) \wedge (\neg p \vee \neg q) \wedge (\neg p \vee \neg r) \wedge (\neg p \vee q \vee r \vee \neg s) \wedge (p \vee q \vee \neg r \vee \neg s \vee \neg t)$.

5. Postulados KM

Katsuno y Mendelzon (KM) unificaron los diferentes enfoques semánticos que un operador de revisión de creencias debería cumplir [16]. Presentamos aquí el análisis de estos postulados sobre nuestra propuesta de operador de revisión de creencias $K' = K \circ \phi = K \wedge Ind(\phi, K)$. Consideremos ahora los postulados KM.

- **(R1)** $K \circ \phi \models \phi$.
- **(R2)** Si $K \wedge \phi$ es satisfactible entonces $K \circ \phi \equiv K \wedge \phi$.
- **(R3)** Si ϕ es satisfactible, entonces también lo es $K \circ \phi$.
- **(R4)** Si $K1 \equiv K2$ y $\phi1 \equiv \phi2$, entonces $K1 \circ \phi1 \equiv K2 \circ \phi2$.
- **(R5)** $(K \circ \phi) \wedge \gamma \models K \circ (\phi \wedge \gamma)$.
- **(R6)** Si $(K \circ \phi) \wedge \gamma$ es satisfactible entonces también $K \circ (\phi \wedge \gamma) \models (K \circ \phi) \wedge \gamma$.

El teorema 1, muestra que nuestro operador de revisión de creencias cumple el postulado $R1$. Si $K \wedge \phi$ es satisfactible y $K \models \phi$, entonces cada $\varphi_i \in \phi$ se infiere de K y por tanto, $Ind(\varphi_i, K) = \varphi_i, i = 1, \ldots, k$. Así, $K \circ \phi = K \wedge Ind(\phi, K) = K \wedge \phi$, cumpliéndose el postulado $R2$.

Analicemos el cumplimiento del postulado $R3$. Este se cumple si $K \circ \phi$ es satisfactible (por $R2$). Pero si $Fals(K) \cup Fals(Ind(\phi, K))$ cubriera a todo el espacio de asignaciones: 2^n, entonces sólo en este caso se redefine $K \circ \phi$. Por ejemplo, si K = $(p \vee q) \wedge (p \vee \neg q)$ y ϕ = $(\neg p)$, como ϕ es independiente con cada cláusula de K, se tendría que $K \circ \phi$ = $(\neg p) \wedge (p \vee q) \wedge (p \vee \neg q)$, que claramente es una fórmula insatisfactible.

Bajo estas circunstancias de comprobación de que $(K \wedge Ind(\phi, K))$ es insatisfactible, se redefine $K \circ \phi$ para que cumpla $R3$. Se redefine $K \circ \phi = ((K \wedge Ind(\phi, K)) - C_j)$, seleccionando la cláusula $C_j \in K$ con la menor información (note que $|SAT(C_j)|$ es mínimo sobre la cardinalidad del conjunto de modelos de cada $C_j \in K$, si $|C_j|$ es máximo en K), y de esta forma se mantendría la satisfactibilidad del resultado de la revisión de creencias.

Los postulados $R4$ y $R5$ se cumplen debido a que nuestro operador de revisión es cerrado sobre las formas conjuntivas. Por ejemplo, si consideramos dos diferentes KB; $K1 \equiv K2$, y dos subfórmulas $\phi1 \equiv \phi2$, se cumple que $Fals(K1)$ = $Fals(K2)$ y $Fals(\phi1)$ = $Fals(\phi2)$. Al trabajar nuestro método sobre los conjuntos $Fals(Ind(\phi, K))$ y al ser tanto K, ϕ y $Ind(\phi, K)$ FC's, se cumple de forma directa el postulado $R4$.

Veamos que se cumple $R5$ $(K \circ \phi) \wedge \gamma \models K \circ (\phi \wedge \gamma)$. Consideremos: $K \circ (\phi \wedge \gamma)$ = $K \wedge Ind(\phi \wedge \gamma, K)$ por definición del operador $(\circ)$, $K \circ (\phi \wedge \gamma) = K \wedge Ind(\phi, K) \wedge Ind(\gamma, K)$ por definición del operador Ind y puesto que tanto ϕ como γ son FC's. Entonces, $K \circ (\phi \wedge \gamma)$ = $K \wedge S \wedge Ind(\gamma, K)$ con $S = Ind(\phi, K)$. Por otro lado, $Fals(K \circ (\phi \wedge \gamma))$ = $Fals(K \wedge S \wedge Ind(\gamma, K))$ = $Fals(K \wedge S) \cup Fals(Ind(\gamma, K))$ = $Fals(K \circ \phi) \cup Fals(Ind(\gamma, K)) \subseteq Fals(K \circ \phi) \cup Fals(\gamma)$, por el Corolario 1. Así, $Fals(K \circ (\phi \wedge \gamma)) \subseteq Fals(K \circ \phi) \cup Fals(\gamma) = Fals((K \circ \phi) \wedge \gamma)$ cumpliéndose $R5$.

$(R6)$ Si $(K \circ \phi) \wedge \gamma$ es satisfactible, entonces $K \circ (\phi \wedge \gamma) \models (K \circ \phi) \wedge \gamma$. Sea $Fals((K \circ \phi) \wedge \gamma)$ = $Fals(K \wedge Ind(\phi, K) \wedge \gamma)$, pero (γ) sólo sería igual a $Ind(\gamma, K)$ sí y solo si γ fuera independiente con cada cláusula de K, y entonces sólo en ese caso se tiene que $Fals(K \wedge Ind(\phi, K) \wedge \gamma)$ = $Fals(K \wedge Ind(\phi, K) \wedge Ind(\gamma, K))$ = $Fals(K \circ (\phi \wedge \gamma)$ y así se cumpliría el postulado $R6$.

6. Análisis de complejidad en tiempo

La función que mide el tiempo de nuestro operador de revisión de creencias $K \circ \phi$, que denotaremos como: $T_0(|\phi|, |K|)$, depende principalmente del tiempo de ejecución del operador de independencia: $Ind(\phi, K)$. Y como $Ind(\phi, K)$ se obtiene del cálculo iterativo de $Ind(\varphi_i, K), \forall \varphi_i \in \phi$, entonces el tiempo de construcción para $Ind(\phi, K)$ depende del tiempo máximo que requiere algún $Ind(\varphi_i, K), \varphi_i \in \phi$.

Como se presentó en la sección del diseño del algoritmo 1, la complejidad en tiempo del proceso $Ind(\varphi_i, K)$ es de orden $O(|K| \cdot n \cdot f(|\varphi_i|, |K|))$, donde $f(|\varphi_i|, |K|)$ es una función entera, que dada una cláusula φ_i y una FC K, determina el número de cláusulas que regresará el proceso $Ind(\varphi_i, K)$. Analicemos ahora, el número máximo posible de cláusulas que se pueden generar a través del proceso $Ind(\varphi_i, K)$.

En algunos casos, $Ind(\varphi_i, K)$ puede generar conjuntos nulos (cuando $\exists\ C_j \in K$, tal que $C_j \models \varphi_i$), pero en los peores casos, la complejidad en tiempo del cálculo de $Ind(\varphi_i, K)$, dependerá de la longitud de los conjuntos: $S_{ij} = \{x_1, x_2, \ldots, x_p\} = lit(C_j)$ - $lit(\varphi_i)$, $j = 1, \ldots, m$.

Como se hizo notar en el ejemplo 2, fija una $\varphi_i \in \phi$, es conveniente ordenar las cláusulas $C_j \in K$ de acuerdo a la cardinalidad de los S_{ij}, $j = 1, \ldots, m$ de menor a mayor, y eliminando de este ordenamiento las cláusulas que sean independientes con φ_i. Una vez ordenadas las cláusulas en K en función a la longitud de S_{ij}, se va aplicando el operador $Ind(\varphi_i, C_j)$, $j = 1, \ldots, m$, determinándose así, la sucesión:

$$
\begin{aligned}
&S_{i0} = v(\varphi_i)\\
&S_{i1} = v(C_1) - v(\varphi_i)\\
&S_{i2} = v(C_2) - (v(C_1) \cup v(\varphi_i))\\
&\ldots\\
&S_{im} = v(C_m) - (v(C_{m-1}) \cup \ldots \cup v(C_1) \cup v(\varphi_i)).
\end{aligned}
$$

El número de cláusulas que se generan por $Ind(\varphi_i, C_1)$ sería $|S_{i1}|$, y para $Ind(\varphi_i, C_2)$ se podría tener en el peor caso, hasta $|S_{i2}|$ nuevas cláusulas por cada una de las cláusulas generadas en $Ind(\varphi_i, C_1)$, y así sucesivamente. Para $Ind(\varphi_i, C_m)$, habría a lo más $|S_{im}|$ posibles cláusulas que se pueden generar por cada una de las anteriores cláusulas en $Ind(\varphi_i, C_{m-1})$.

Esto nos genera un proceso multiplicativo sobre el número de cláusulas en $Ind(\varphi_i, K)$, dado por: $|Ind(\varphi_i, K)| \leq \prod_{j=1}^{m} |S_{ij}| = |S_{i1}| * |S_{i2}| * \ldots * |S_{im}|$ y bajo la restricción $\sum_{j=1}^{m} |S_{ij}| \leq n - |v(\varphi_i)|$, ya que cada conjunto S_{ik} cubre el espacio de asignaciones formado por las variables que no ha sido cubierto por las variables de las C_j, $j = 1, \ldots, k-1$ y las variables de φ_i, y en todo este proceso no puede cubrirse más de $n - v(\varphi_i)$ variables.

De hecho, si algún $S_{ij} = \emptyset$, entonces el conjunto de cláusulas en $Ind(\varphi_i, K)$ es también vacío, indicando que el K actual no cambiará al considerar tal φ_i y entonces $\prod_{j=1}^{m} \mid S_{ij} \mid = 0$.

Cuando no hay cláusulas independientes con φ_i, ni ningún $S_{ij} = \emptyset$ para $j = 1, \ldots, m$, entonces la complejidad en tiempo para calcular $Ind(\varphi_i, K)$ es acotado por su número de cláusulas, en otras palabras, se tiene que $|Ind(\varphi_i, K)| \leq |S_{i1}| * |S_{i2}| * \ldots * |S_{im}| * Poly(n)$. Donde $Poly(n)$ resume un tiempo polinomial sobre el número de variables que se genera de aplicar el operador $Ind(\varphi_i, C_j)$ y por aplicar el ordenamiento inicial sobre las cláusulas de K.

Es claro que el valor $|Ind(\varphi_i, K)|$ no puede ser mayor al número de asignaciones que están en $Fals(\varphi_i) - Fals(K)$, ya que de hecho, se está cubriendo este espacio de asignaciones vía cláusulas independientes. Esto significa que $|S_{i1}| * |S_{i2}| * \ldots * |S_{im}| \leq 2^{(n - |\varphi_i|)}$.

Podemos inferir entonces que la complejidad en tiempo $T_0(|\phi|, |K|)$ para nuestro operador de revisión de creencias, en el peor de los casos, esta acotado superiormente por $Max\{|S_{i1}| * |S_{i2}| * \ldots * |S_{im}| : \forall \varphi_i \in \phi\}$, suprimiendo factores polinomiales sobre n (el número de variables) y sobre el tamaño de la KB K. A su vez, este valor máximo está acotado superiormente por $2^{(n - min\{|\varphi_i| : \varphi_i \in \phi\})}$. Cumpliéndose entonces que: $T_0(|\phi|, |K|) \leq Max\{|S_{i1}| * |S_{i2}| * \ldots * |S_{im}| : \forall \varphi_i \in \phi\} \in O(2^{(n - min\{|\varphi_i| : \varphi_i \in \phi\})})$. Y por tanto, la complejidad en tiempo de nuestra prouesta es de $O(|K| \cdot n \cdot 2^{(n - min\{|\varphi_i| : \varphi_i \in \phi\})})$.

7. Conclusiones

Un problema fundamental del razonamiento automático en el cálculo proposicional y de los sistemas inteligentes en general, es el problema de revisión de creencias.

En este trabajo se presenta un método novedoso para construir $K' = K \circ \phi$, a partir de considerar que K y ϕ son FC's. Como K y ϕ son FC's, el proceso de revisión entre K y ϕ se va reduciendo a realizar la revisión entre cada φ_i de ϕ y cada C_j de K, simplificando el problema total de revisión en resolver los $|K| * |\phi|$ subproblemas de revisión entre dos cláusulas.

Se construyó un operador lógico llamado $Ind(\varphi_i, C_j)$, que encuentra las cláusulas que cubren el espacio de asignaciones faltantes para que se cumpla que: $Fals(\varphi_i) \subseteq Fals(C_j) \cup Fals(Ind(\varphi_i, C_j))$, al iterar este proceso sobre todo $\varphi_i \in \phi$, y cuidando reducir clásulas complementarias, encontramos un proceso efectivo para la revisión de creencias entre formas conjuntivas.

Se demuestra la corrección de nuestra propuesta de revisión de creencias, la verificación de cumplimiento de los postulados KM, así como el análisis de la complejidad en tiempo de los procesos involucrados en nuestro método.

Agradecimientos

Los autores desean agradecer al SNI-CONACyT por su apoyo. El primer autor tuvo apoyo de beca sabática Conacyt durante la realización de este trabajo. Los autores reconocen y agradecen los invaluables comentarios y sugerencias realizadas al artículo por parte de los revisores anónimos, y por el Editor de esta revista.

Referencias

1. **Alchourron, C., Gardenfords, P., & Makinson, D. (1985).** On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic*, Vol. 50, pp. 510–530.
2. **Booth, R., Meyer, T., & Varzinczak, I. J. (2009).** Next steps in propositional horn contraction. *Proc. 21st. Int. Join the conference on artificial Intelligence - IJCAI*, IJCAI, pp. 702–707.
3. **Bordeaux, L., Hamadi, Y., & Zhang, L. (2006).** Propositional satisfiability and constraint programming: A comparative survey. *ACM Computing Surveys*, Vol. 38, pp. 1–54.
4. **Creigno, N., Papini, O., Woltran, S., & Pichler, R. (2012).** Belief revision within fragments of propositional logic. *Technical report*, Technische Universitat Wien, pp. 1–21.
5. **Cresto, E. (2002).** Revisión de creencias y racionalidad. *Cuadernos CIMBAGE*, Vol. 5, pp. 133–156.
6. **Dalal, M. (1988).** Investigations into theory of knowledge base revision. *Proc. of the 7th National Conf. on Artificial Intelligence*, AAAI, pp. 475–479.
7. **Darwiche, A. (2001).** On tractable counting of theory models and its application to truth maintenance and belief revision. *Applied Non-Classical Logics*, Vol. 11, pp. 11–34.
8. **Darwiche, A. & Pearl, J. (1997).** On the logic of iterated belief revision. *Artificial Intelligence*, Vol. 89, pp. 1–29.
9. **Delta, G. & Zacarias, F. (2015).** A model-based algorithm for propositional belief revisions. *IEEE Latin America Transactions*, Vol. 13, pp. 1055–1060.
10. **Delgrande, J. P. (2008).** Horn clause belief change: Contraction functions. *Proc. of the 11th Int. Conf. on the Principles of Knowledge Representation and Reasoning*, AAAI, pp. 156–165.
11. **Doubois, O. (1991).** Counting the number of solutions for instances of satisfiability. *Theoretical Computer Science*, Vol. 81, pp. 49–64.
12. **Eiter, T., Fink, M., Sabattini, G., & Tompits, H. (2000).** Considerations on updates of logic programs. *Lecture Notes in Artificial Intelligence*, Vol. 1.
13. **Ellis, D. (2011).** Irredundant families of subcubes. *Mathematical Proc. of the Cambridge Philosophical Society*, Vol. 150, pp. 257–272.
14. **Fermé, E. (2007).** Revisión de creencias. *Revista Iberoamericana de Inteligencia Artificial*, Vol. 11, pp. 17–39.
15. **Hansson, S. (1994).** Belief contraction. *Journal of Symbolic Logic*, Vol. 59, pp. 845–859.
16. **Katsuno, H. & Mendelzon, A. O. (1991).** On the difference between updating a knowledge base and revising it. *KR'91 Cambridge, MA, USA*, Vol. 1, pp. 387–394.
17. **Khardon, R. & Roth, D. (1996).** Reasoning with models. *Artificial Intelligence*, Vol. 87, pp. 187–213.
18. **Lehmann, D. (1995).** Belief revision. *Proc. IJCAI'95*, IJCAI, pp. 1534–1540.
19. **Liberatore, P. & Schaerf, M. (2001).** Belief revision and update: Complexity of model checking. *Journal of Computer and System Sciences*, Vol. 62, pp. 43–72.
20. **Liberatore, P. & Schaerf, M. (2001).** The complexity of model checking for belief revision and update. *Journal of Computer and Systems Sciences*, Vol. 62, pp. 43–72.

21. **Nebel, B. (1998).** How hard is it to revise a belief base? *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, Vol. 3, pp. 77–145.

22. **Papadimitriou, C. H. (1994).** *Computational Complexity*. Addison-Wesley Pub.

23. **Petrick, S. (1956).** *A direct termination of the irredundant forms of a boolean function from the set of primer implicants*. Technical report, Cambridge Res Center.

24. **Quine, W. (1952).** The problem of simplifying truth functions. *JSTOR*, Vol. 59, pp. 521–531.

25. **Satoh, K. (1988).** *Nonmonotonic Reasoning by Minimal Belief Revision*. Vol. 358, Institute for New Generation Computer Technology.

26. **Valdéz, N. J. & Falapa, M. A. (2013).** Dinamica de conocimiento: contracción múltiple en lenguajes Horn. *Proc. XIX congreso argentino de ciencias de la computación*, Libro de actas - CACIC2013, pp. 61–70.

Article received on 18/09/2016; accepted on 03/10/2016.
Corresponding author is Guillermo De Ita Luna.