BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

UN ALGORITMO PARA LA REVISIÓN DE CREENCIAS ENTRE FORMAS CONJUNTIVAS

Tesis de para obtener el grado de Maestra en Ciencias de la Computación presentada por:

Alma Delia García García*

Dirigida por:

Asesor

Dr. Guillermo De Ita Luna

Co-Asesor

Dr. Alejandro Rangel Huerta

Puebla, Pue., Septiembre 2016

---

*Becario CONACYT

Agradecimientos

Resumen

La revisión de creencias es un área central en la representación del conocimiento y en el procesamiento de razonamiento automático, consiste en estar incorporando nuevas creencias, cambiando tan poco como sean posibles las creencias originales y manteniendo la consistencia de la base de creencias.

Consideremos 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 el operador de revisión como: $K' = K \circ \phi$.

En este documento de tesis se propone un nuevo operador binario lógico Ind entre formas conjuntivas, donde $Ind(\phi, K)$ 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 ésta propuesta satisface los postulados KM.

Por otro lado se presenta la demostración de la correctez del algoritmo de revisión de creencias así como el análisis de complejidad en tiempo.

Abstract

Belief revision is a central area in knowledge representation and processing of automated reasoning. belief revision is to incorporate new beliefs, changing as little as possible the original beliefs and maintaining the consistency of core beliefs.

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: $K' = K \circ \phi$.

In this thesis paper a new logical binary operator Ind between two conjunctive forms proposed, 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 fasifying 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.

On the other hand the correctness proof of our belief revision method, and the analysis of its time complexity.

Índice general

Índice de figuras

Capítulo 1

Introducción

La inteligencia artificial (AI) se ha definido como *el estudio de las ideas que permiten a las computadoras ser inteligentes* [1] y *el estudio de la conducta inteligente* [2]. Las definiciones anteriores varían, sin embargo se basan en un entendimiento implícito de la naturaleza de la inteligencia misma. En la actualidad existen varios sistemas informáticos construidos con la finalidad de mostrar un comportamiento inteligente, la duda surge cuando nos damos a la tarea de categorizar o reconocer ese comportamiento como inteligente.

Una definición de inteligencia es: *La capacidad de adquirir y aplicar el conocimiento* [3]. Para la adquisición del conocimiento con éxito, debe existir un método adecuado para la representación del conocimiento de tal forma que sea posible almacenar, recuperar y manipular la información eficientemente.

En los últimos años se han propuesto muchos formalismos de representación del conocimiento, pero existen otros problemas en la adquisición donde:

- *Las creencias son incompletas*, no se sabe lo suficiente para llegar a la conclusión correcta, ya sea porque no es posible darse cuenta de ciertos hechos, o porque las creencias no son suficientemente precisas para distinguir entre alternativas similares.

- *Las creencias son incorrectas*, pueden ser causadas por una observación defectuosa, un error en la comunicación, o engaño por parte de la fuente de información.

- *Las creencias son inapropiadas*, para una situación particular cuando se es capaz de observar una acción o evento que altera el mundo, o de lo contrario cuando no es posible darse cuenta de las consecuencias de la acción o evento, por lo que las creencias que antes eran correctas ya no reflejan el verdadero estado de cosas en el mundo.

Por lo tanto, nuestras creencias acerca del mundo deben cambiar constantemente a medida que adquirimos nueva información de fuentes más o menos fiables. De ahí la importante necesidad de de realizar una *revisión de creencias*. Por ello en éste documento de tesis se propone un algoritmo nuevo que permita realizar una revisión de creencias basado en modelos de formas conjuntivas.

1.1. Planteamiento del Problema

Dada una base inicial de conocimiento (KB) K en forma conjuntiva (FC) y nueva información ϕ codificada también en FC, se desea revisar (construir) una nueva KB K' que contenga la revisión de ϕ en K, tal que $K \models \phi$, donde $K' = K \circ \phi$, donde K' difiere lo mínimo posible de K.

1.2. Justificación

Para razonar sobre el mundo real, los sistemas inteligentes deben tener formas de tratar con información incompleta e inconsistente. Esta cuestión se aborda en el estudio de los sistemas basados en la automatización del razonamiento deductivo. El razonamiento deductivo proposicional es generalmente resumido como: 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 $K \models \phi$), que se conoce como *el problema de implicación proposicional*. [4]

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 ejemplo, al trabajar en planeación, diseño de sistemas multi-agentes, diagnóstico lógico, razonamiento aproximado, entre otras aplicaciones [5] [6].

En general, el problema de la implicación lógica es un reto difícil para el razonamiento automático y resulta ser un problema Co-NP difícil, incluso en el caso proposicional [7].

Por otro lado, 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 le da su versatilidad a la teoría de revisión de creencias [8].

En el contexto de los sistemas basados en agentes, podemos decir que una teoría de revisión de creencias es un área de estudio formal en donde se pueden dar distintas aplicaciones. La función básica de dicha estructura formal es ofrecer instrucciones acerca de cómo cambiar una base de conocimiento cuando nos enfrentamos con nueva información. 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 del sistema.

Ahora bien, es deseable que los cambios no se efectúen de cualquier manera, sino de manera racional. En la revisión de creencias se trata por tanto, de una teoría normativa que nos indica en cada caso, cual es la manera óptima de proceder. 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 y al mismo tiempo, minimice la pérdida de información de K [7].

La teoría AGM propone un conjunto de postulados racionales, que cualquier operador de revisión debe satisfacer. Katsuno y Mendelzon [9]. unificaron los diferentes enfoques semánticos de revisión de creencias, y reformularon esos postulados (a los que llamamos postulados KM), además propusieron un teorema de representación que caracteriza las operaciones de revisión en términos de pre-órdenes totales sobre interpretaciones.

La revisión de creencias ha sido ampliamente estudiada 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, obteniéndose diferentes resultados sobre la complejidad en tiempo de estas propuestas [10] [11]. Sin embargo, como sabemos, pocos trabajos se centraron en la revisión de creencias en el marco de fragmentos de la lógica proposicional, excepto para el caso de fórmulas de Horn [12].

En esta tesis se desarrollará una propuesta para la revisión de creencias entre formas conjuntivas. Ésta propuesta tiene el enfoque basado en el uso de modelos para el cálculo del operador de revisión de creencias. Puesto que la KB y la fórmula de consulta ϕ están en forma normal conjuntiva (FC), lo que facilita el cálculo efectivo del conjunto de asignaciones que falsifican a tales fórmulas 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 revizar consistencia entre FC's. Aunque es bien conocido que la inferencia basada en el principio de resolución tiene limitaciones intrínsecas [7]. Se pretende mostrar que el proceso de revisión de creencias puede realizarse de forma práctica, basádose en las asignaciones falsificantes, en lugar de utilizar el proceso de resolución. Al tener un método efectivo de revisión de creencias entre FC's, su extensión no es difícil para considerar otras formas normales, dado que cualquier fórmula Booleana se puede expresar en FC.

Aunque la inferencia proposicional es un problema Co-NP-completo en su versión general, existen también algunos casos que se pueden resolver de manera eficiente [7]. En este trabajo se mostrará también que el uso de patrones falsificantes para las cláusulas ayuda a determinar si una FC infiere a otra FC, y por tanto, a construir un algoritmo para la revisión de creencias entre FC's.

1.3. Objetivos generales y específicos

1.3.1. Objetivo general

Desarrollar un algoritmo basado en el uso de modelos para decidir la inferencia lógica entre formas conjuntivas (FC's). El resultado de nuestro algoritmo permitirá a su vez, realizar la revisión de creencias entre las FC's de entrada.

1.3.2. Objetivo específico

Desarrollar una propuesta algorítmica, determinando su complejidad en tiempo, para resolver tanto el problema de inferencia lógica entre Formas Normales Conjuntivas, como el problema de revisión de creencias de esas mismas Formas Normales Conjuntivas.

1.4. Descripción general de la tesis

La principal contribución de esta tesis es diseñar e implementar un algoritmo para la revisión de creencias, incluyendo formalismos de representación, y propuestas de cambio de creencias. En el capítulo 1, se describen los conceptos genererales que contendrá la tesis. En el capítulo 2, se presenta un resumen de algunos de los enfoques formales de la revisión de creencias. Se presenta el marco de revisión de creencias AGM y el marco KM, seguido de estos, el marco para la revisión iterada DP. En el capítulo 3, se introducirán algunos métodos para la revisión de creencias trabajados por Dalal, Satoh, Winslett, Borgida y Forbus. Mostraremos la comparación de estos métodos identificando las fortalezas y debilidades de cada uno.

El resto de la tesis describe el funcionamiento del algoritmo para revisión de creencias entre formas conjuntivas, abordando cuatro principales cuestiones. La primera cuestión es la de representar las creencias mediante cálculo proposicional, la segunda cuestión describe el funcionamiento del método que utilizaremos, la tercera cuestión es la de analizar si el algoritmo es eficiente considerando que casos pueden resolverse de forma eficiente de aquellas que requieren tiempo exponencial. Y finalmente, que postulados cumple nuestro método de acuerdo al enfoque epistémico KM.

Capítulo 2

Revisión de creencias

En este capítulo se presentan algunas definiciones preliminares que nos serán útiles en todo el documento.

2.1. Preeliminares

Una $Variable$ es un símbolo que denotaremos como x_i que toma un valor lógico de verdadero o falso.

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 $x^1 = x$.

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 *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 usaremos aquí a $\neg Y$ 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 $s : Y \rightarrow \{1, 0\}$ 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, $SAT(K) \subseteq SAT(\phi)$.

2.2. Creencias y conocimiento

El término de *creencia* no es trivial, por ello introducimos una noción sobre la misma: Una creencia *es una proposición que es considerada como verdadera por un agente (o sujeto)*, ésta puede ser o no verdadera en la realidad [13].

El término *conocimiento* se encuentra muy relacionado con el de creencia. Considerando una vez más la existencia de un agente, puede definirse conocimiento como *una creencia verdadera justificada*, un agente sabe si y sólo si es verdadero y tiene una justificación para creer en ella. Es decir, un conocimiento es una creencia que es verdadera y que es creída no por simple suerte, sino porque existe una justificación para creer en ella. Existen dos principales puntos de vista acerca de cómo el conocimiento puede estar justificado Pappas y Swain 1978 [13]. Donde podemos observar dos enfoques; los llamados estados epistémicos estáticos y los dinámicos.

2.2.1. Estados epistémicos estáticos

En la teoría fundamental de la justificación, una proposición es aceptada (es decir, se cree) si y sólo si:

- ϕ es evidente por sí mismo, ó
- ϕ puede derivarse de un conjunto de proposiciones aceptadas.

Las sentencias que cumplen la primera condición se llaman creencias fundacionales. Se supone que la aceptación de estas creencias es indiscutible, ya que constituyen la base de todo el estado epistémico. Las creencias fundacionales generalmente corresponden a los hechos duros, tales como observaciones sobre el mundo físico. La segunda condición son creencias que son consecuencias lógicas de las creencias fundamentales.

Así, cada creencia tiene una justificación (o bien son hechos, o bien son consecuencias derivables de los hechos y de otras creencias), el uso de las creencias que también son justificables, y sus justificaciones deben a su vez basarse en las creencias más justificadas; por lo tanto, el conjunto de creencias consiste en proposiciones unidas por cadenas de justificaciones.

2.2.2. Estados epistémicos dinámicos

Por lo general, las dos teorías de la justificación se relacionan con estados de creencias estáticas, pero esta se aplica a la revisión creencia, de acuerdo con la teoría, la revisión de creencias consistiría en dos pasos: retraer creencias que ya no tienen una justificación satisfactoria, y la adición de creencias que ahora sí tienen una justificación satisfactoria. En este sentido, la teoría fundamental es no conservativa, ya que el hecho de que una creencia fue válida en el pasado no implica que deba serlo en la situación actual. Por lo tanto, en el fundamentalismo, la credibilidad de una creencia es independiente de la historia del conjunto de creencias.

Por el contrario, la teoría de la coherencia proporciona un enfoque conservador para la revisión de creencias, al exigir un cambio mínimo en el estado actual del sistema de creencias. Esto obliga a que cualquier aumento del sistema de creencias sea el máximo suficientemente para preservar la coherencia global del sistema de creencias. Así, la historia de un conjunto de creencias determinará gran parte de sus contenidos futuros. El método más común, sobre todo en los entornos computacionales, es la consideración de un conjunto de sentencias en un lenguaje lógico, a lo que se conoce como cálculo proposicional. Denotaremos la creencia establecida por el símbolo que representa una KB.

En un estado de creencia consistente, hay tres posibles actitudes epistémicas hacia una

- ϕ es aceptada ($\phi \in K$)
- ϕ es rechazada ($\neg\phi \in K$)
- ϕ es indeterminado ($\phi \notin Ky\neg\phi \notin K$)

Un conjunto de creencias en el que ϕ y $\neg\phi$ se aceptan es inconsistente y por lo tanto, no esta permitido en el modelado de los estados epistémicos. Además, para que un agente sea idealmente racional, debe creer todo lo que es consecuencia lógica de sus creencias. Estos criterios pueden no ser adecuados para el modelado de agentes en la vida cotidiana, que no tienen esta propiedad de la omnisciencia lógica (conocimiento sobre toda la lógica), pero son útiles en la definición de conjuntos de creencias idealmente racionales.

2.3. Marco AGM

El problema de la revisión de creencias puede ser simplemente declarado como sigue: Dada una base de conocimientos K consistente y una información ϕ, es necesario saber cómo se asimila establecer ϕ en la base de conocimientos original K. Algunos de los requisitos básicos de este proceso son en primer lugar que el nuevo estado de creencia sea consistente; en segundo lugar, que la nueva información se refleje en el estado resultante de la creencia; y en tercer lugar, que cualquier creencia anterior (a no ser que contradiga la nueva creencia) se conserva en el nuevo estado de creencias.

Un enfoque formal de revisión de creencias que satisface estos requisitos fué desarrollado por Alchourron, Gardenfors y Makinson, el cual se conoce como el enfoque AGM de revisión de creencias [9]. Este enfoque, es un formalismo matemático ideal para representar los estados de creencias y la descripción de las formas en que estos estados cambian con la nueva información.

2.3.1. Operaciones para el cambio de creencias AGM

Hay tres tipos de cambio de la creencia en el enfoque AGM: *expansión, contracción* y *revisión.* [13].

La *expansión* implica la aceptación de una nueva creencia, sin renunciar a ninguna creencia anterior, incluso si están en contradicción con la nueva información.

La *contracción* una contracción consiste en eliminar cierta creencia (y todo lo que la soporta) sin agregar nuevas creencias.

La *revisión* consiste en adicionar a un estado de creencias una nueva creencia, posiblemente junto con sus consecuencias, asegurando que el nuevo estado de creencias sea consistente.

En una expansión, la actitud hacia una sentencia ϕ se cambia de indeterminado a aceptado (expansión por ϕ) o rechazado (expansión por $\neg\phi$), sea porque ϕ o $\neg\phi$ se añade a la creencias de K, y ninguna de las creencias anteriores se retraen. Gärdenfors muestra que la operación de cambio menor que acepta una nueva creencia, sin quitar ninguna de las creencias anteriores viene dada por: $K + \phi = Cn\ (K \cup \phi)$.

Donde Cn representa la consecuencia lógica de ϕ con respecto K, sin embargo en este documento de tesis representaremos una consecuencia lógica con el símbolo $\models$.

El segundo tipo de cambio epistémico es la contracción, que ocurre cuando la actitud hacia una creencia ϕ cambia de aceptada a rechazada. No hay nuevas sentencias que se añadan al conjunto de creencias, por lo que el conjunto de creencias contractada es siempre un subconjunto de la creencia inicial establecidas en K.

La operación de revisión abarca los dos últimos tipos de cambio epistémico, cuando la actitud hacia los cambios de pasa de aceptado a rechazado (revisión por $\neg\phi$), o bien, pasa de rechazo a aceptado (revisión por ϕ).

2.3.2. Racionalidad de los postulados AGM

Los primeros seis postulados AGM para la revisión de creencias se conocen como los postulados básicos y los dos postulados restantes se conocen como complementarios. Los postulados para la contracción son los siguientes:

- **(K1)** $K * \phi$ es un conjunto de creencias.
- **(K2)** $\phi \in K * \phi$.
- **(K3)** $K * \phi \subseteq K + \phi$
- **(K4)** Si $\neg\phi \notin$ entonces $K + \phi \subseteq K * \phi$
- **(K5)** $K * \phi = K$ si y solo si ϕ es inconsistente
- **(K6)** Si $\alpha \equiv \phi$ entonces $K * \alpha = K * \phi$
- **(K7)** $K * (\alpha \circ \phi) \subseteq (K * \phi) + \phi$

- **(K8)** Si $\neg\phi \notin K * \phi$ entonces $(K * \phi) + \phi \subseteq K * (\alpha \circ \phi)$

El primer postulado requiere que el resultado de una revisión sea un conjunto de creencias. El segundo postulado asegura que después del proceso de revisión, ninguna nueva sentencia se añade al conjunto de creencias. En **(K3)** el conjunto de creencias no se ha modificado por un intento de retraer una sentencia, lo que no ocurre en el conjunto de creencias. El postulado **(K4)** afirma que la creencia retraída no puede permanecer en el conjunto de creencias, excepto para el caso contradictorio. **(K4)** asume que es consistente a menos que sea una contradicción. **(K6)** requiere que la revisión de creencias sea independiente de la sintáxis. Los postulados complementarios implican que el cambio mínimo de al incluir dos nuevas informaciones y es el mismo que el obtenido por la expansión de $(K * \phi)$ por ϕ siempre y cuando ϕ no contradiga las creencias en $K * \phi$.

Los postulados suplementarios **(K7)** y **(K8)** de revisión son una generalización de **(K3)** y **(K4)**. Su propósito, al igual que su contraparte para la contracción, es indicar cómo realizar revisiones de creencias iteradas o compuestas [8], [13]. La idea es que si ϕ es una revisión de K y se desea cambiar ϕ. agregando nuevas creencias, tal cambio debería hacerse usando expansiones de K siempre que sea posible.

Estos postulados fueron propuestos sobre bases filosóficas. Sin embargo, la propuesta computacional estaba completamente ausente. Por lo que en 1991 Katsuno y Mendelzon, hicieron un estudio acerca de cómo representar concretamente a los conjuntos de creencias en el caso discreto finito, es decir, cuando se hace revisión de creencias representadas mediante un número finito de variables proposicionales. Considerando además, que se necesitaba representar un conjunto de creencias mediante una computadora y tener una representación compacta del conjunto de creencias. Por lo tanto, decidieron reformular los postulados AGM, llamando a este nuevo marco como postulados KM.

2.4. Marco KM

Katsuno y Mendelzon (KM) unificaron los diferentes enfoques semánticos que un operador de revisión de creencias debería cumplir [14]. 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 $K_1 \equiv K_2$ y $\phi_1 \equiv \phi_2$, entonces $K_1 \circ \phi_1 \equiv K_2 \circ \phi_2$.
- **(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 primer postulado **(R1)** se satisface directamente por definición. Note que ϕ es consistente con K, por lo que esto hace satisfactible la revisión no habiendo inconsistencia asi se cumple el postulado **(R2)**. El postulado **(R3)** garantiza que el resultado de una revisión es consistente con la nueva frase. Una base de conocimientos inconsistente es el resultado de una teoría inadecuada, esto se puede remediar con la revisión (o contracción) por la adición de nuevos conocimientos que reemplacen la inconsistencia (o de extraer conocimiento mediante la contracción).

Supongamos que $\phi_1 \equiv \phi_2$, y $K_1 \equiv K_2$. queremos ver que cualquier σ_i tal que $\models \sigma_1 \equiv \models \phi_2$ para $i = 1, 2$, se tiene que $\sigma_1 \equiv \sigma_2$ de la hipótesis $\phi_1 \equiv \phi_2$ se obtiene de $\models K_1 \equiv \models K_2$ y así se cumple el postulado **(R4)** [15].

Los postulados **(R5)** y **(R6)** representan la condición de que la revisión se lleva a cabo con un cambio mínimo.

2.5. Marco DP

Usando el teorema de representación de Katsuno y Mendelzon, Darwiche y Pearl (DP) mostraron en 1997 [15] que en la revisión KM (equivalente a la revisión AGM en forma general) se pueden encontrar conflictos al iterar el proceso de revisión. Argumentado que estos problemas venían del hecho de que lo único conocido sobre el ordenamiento de los mundos, luego de la revisión de la nueva información, es que los mundos más plausibles en el primer nivel, son los modelos minimales de según el preorden total asignado al conjunto de creencias inicial.

Ejemplo 1: Se observa un agente extraño X y parece que esta aullando como si fuese un gato, por tanto, concluimos que el X no vuela, y mucho menos que es un pájaro. Sin embargo, si existe el caso que X resulte ser un pájaro, porque no se esta seguro sobre el agente X, se debe estar preparado para cambiar de parecer y concluir que X podría volar. Viendo al agente X verificamos que puede volar. La pregunta sería ahora si debemos o no seguir pensando que X sea o no un pájaro, después de todo.

Seguir pensando en que no es un pájaro y no estar preparado para ello, es una respuesta negativa y por lo tanto, el operador que sea utilizado no sería razonable. Sin embargo el operador propuesto por Darwiche y Pearl es compatible con KM y responde a las características indeseadas.

Con este ejemplo se muestra que el conjunto de creencias resultante cree sólo que el agente es un pájaro y olvida el hecho de que puede volar. Este ejemplo nos muestra que existen operadores de revisión KM que renuncian a creencias sin justificación alguna. El ejemplo a continuación, propuesto por Darwiche y Pearl muestra que también existen operadores de revisión que cumplen los postulados KM y que generan creencias sin justificación alguna.

Ejemplo 2: Se nos presenta una señorita llamada Brenda que parece ser lista y luce adinerada, por lo tanto creemos que Brenda es adinerada. Claramente, seguiremos creyendo que Brenda es lista aunque nos enteremos que es pobre y seguiremos creyendo que Brenda es adinerada aunque nos enteremos que no es lista. Entonces, llega a nosotros una información que dice que Brenda no es lista, y por lo tanto, seguimos convencidos de que Brenda es adinerada. Pero otra nueva información no es dada y nos dice que Brenda si es lista después de todo. Como el hecho de que Brenda sea o no lista no afecta en absoluto el hecho de que sea adinerada, lo razonable es que después de toda la confusión sobre la inteligencia de Brenda, sigamos creyendo que Brenda es adinerada.

Con esto se propone un operador de revisión compatible con los postulados KM, que permite dejar de creer que Brenda es adinerada e incluso, adquirir la creencia de que Brenda no es adinerada.

Capítulo 3

Algunos métodos de revisión de creencias

Actualmente existen algunas propuestas de revisión de creencias basadas en modelos de las fórmulas y que se identifican por el nombre de sus autores; Dalal, Satoh, Winslett, Borgida y Forbus [16].

3.1. Método Dalal

Dalal define la revisión en términos de modelos de la KB, donde se propone una descripción de nivel simbólico equivalente a representar un método sintáctico para la revisión de las bases de conocimiento. Su trabajo pretendía mostrar la relación en la filosofía de los aspectos formales de la lógica de cambio de creencias, debido a que recientemente han atraído la atención de la comunidad de Inteligencia Artificial(IA), también mencionaba que para cualquier esquema de revisión, es deseable que se conserve tanto como sea posible las creencias antes de la revisión.

Parte del trabajo de Dalal se desarrolló sin previo conocimiento del modelo AGM, por lo que propuso una serie de principios, no formalizados, que tomó como guías intuitivas para definir una revisión, éstos se describen a continuación:

- El conocimiento se tiene que revisar y tener la misma representación que el conocimiento anterior, con el fin de ser capaz de repetirse

- La irrelevancia de la sintáxis (el resultado de la revisión no depende de la representación sintáctica de los conocimientos viejos o nuevos)

- El principio fundamental en el caso computacional, punto débil de los diversos métodos de actualización propuestos previamente dentro del campo de IA
- El mantenimiento de consistencia, primacía (prioridad) de la nueva información
- La persistencia del conocimiento previo (asimilable a mínima pérdida) y la formalidad del conocimiento (no arbitrariedad de la elección entre los múltiples resultados de la revisión).

Dalal generó una medida cuantitativa de los cambios sobre un conjunto de interpretaciones y la aplicó para definir un cambio mínimo sobre los modelos de la KB. Posterior a esto construye un algoritmo para el cálculo de la fórmula del lenguaje correspondiente a cada conjunto de interpretaciones, de modo que para poder definir sintácticamente a partir de la fórmula representativa de la KB original, la fórmula que sugiriera diera como resultado la revisión, por lo que se reconoció que cada paso del algoritmo propuesto requería una verificación de consistencia, debido a que lo hacía, en el caso general, completo y por ello consideraba su investigación preliminar [18].

3.2. Método Satoh

Ken Satoh buscaba diferenciar su trabajo *Not Monotonic Reasoning by Minimal Belief review* de los formalismos conocidos de NMR *Defaul Logic de Reiter y circumscription de Mc Carthy* [17]. Por lo que decide investigar sobre una estrategia especial de revisión de creencias llamada *revisión de creencias mínima* mostrando que esa estrategia realizaba algunas clases de razonamiento no monótono.

Por otra parte Satoh podía distinguír entre el conocimiento y las creencias. Por lo que definía el conocimiento como un subconjunto de las creencias que era indiscutible, *si el agente conoce p, entonces cree en p* y las creencias restantes eran irrelevantes. La nueva información tenía status de conocimiento, por lo que el corpus de creencias debía acomodarse a ella.

La idea central de Satoh para la revisión de creencias era que el orden a minimizar dependía tanto de los modelos de las creencias previas como de los modelos de las nuevas creencias, a diferencia del modelo AGM que sólo dependía de las originales. Satoh formalizó esta idea planteando formulaciones de segundo orden en una forma muy similar a Circumscription(determinando límites).

El trabajo de Satoh también comparaba brevemente su propuesta con los formalismos conocidos de NMR [18], con el TMS de Doyle [18] y con los trabajos en actualización de bases de datos, a los que cuestiona por no cumplir con lo que Dalal llama *Irrelevancia de la sintaxis* y dedica los apartados finales y más extensos a la *Teoría de la lógica de cambio* (AGM) y a la propuesta de Dalal, a la que consideraba similar a la suya, aunque señalando algunas diferencias como: AGM no distingue entre conocimiento y creencias, consideraba la situación en la que el conocimiento era el conjunto de las tautologías, en ese caso ambas definiciones eran comparables. Posteriormente demostraba que su propuesta cumpliera los postulados AGM, salvo la vacuidad y los postulados adicionales, aunque si el lenguaje era proposicional, también se cumpliría el primero. Es interesante resaltar que Dalal y Satoh desconocían la teoría AGM al momento de concebir sus trabajos y sin embargo, tenían una sintonía casi perfecta con ella.

3.3. Método Winslett

El método de actualización de Winslett modifica los modelos de K, uno por uno, remplazando cada uno con el más cercano dentro de los modelos de A. La circunscripción sin variar letras se expresa como:

$CIRC(T(X); X, \emptyset) = \neg X * WT(X)$

Con el fin de correlacionar la actualización de Winslett con circunscripción, se debe estar seguro que a cada modelo distinto de K corresponden a modelos que no son comparables en la teoría circunscriptiva.

Esto se obtiene por la siguiente reducción:

$T(X,Y,Z) = K(Y) \wedge (Y \neq Z) \wedge A(X) \wedge (W = (X = Y))$

$K(X) *_W A(X) = CIRC(T(X,Y,Z)); Y \cup Z \cup W, X)$

La sub-fórmula $Y \neq Z$ garantiza que cada dos modelos M y N de K, con diferentes asignaciones a Y, no pueden ser comparables debido que no es posible $(M \cap Y) \subseteq (N \cap Y)$ y $(M \cap Z) \subseteq (N \cap Z)$ al mismo tiempo.

Winslett recalcaba: que lo que es menos reconocido en la primera etapa de la revisión de creencias, es la incorporación de nueva creencia extensional en el conjunto pre-existente de las creencias extensionales, es en sí, bastante difícil si cualquiera de las nuevas o antiguas creencias implican información incompleta [18]. Algunos trabajos de Winslett han demostrado algunas correlaciones entre un operador específico y uno de circunscripción, los métodos de proximidad local son los mejores para relacionarlos con la circunscripción, debido a que se reducen mínimamente todas las letras.

3.4. Método Borgida

En 1985 Borgida presenta un articulo llamado; *language features for flexible handling of exceptions in information systems* [19]. El artículo presenta una instalación de excepciones adecuadas para idiomas que se utilizan para implementar sistemas de información entre bases de datos intensivas. Ese mécanismo facilita el desarrollo y mantenimiento de sistemas de software más fléxibles mediante el apoyo a la abstracción de los detalles relativos a sucesos especiales o anormales. Borgida considera que las restricciones de tipo impuestas por el esquema, así como las distintas afirmaciones de integridad semántica y la contribución fundamental de su trabajo es permitir excepciones a limitaciones que persistan.

Para lograr esto, proponen soluciones a una serie de problemas, incluyendo las cuestiones de aplicación, el intercambio y la computación con información inusual, manejo de excepciones por los usuarios, la lógica de las limitaciones con las excepciones, y también se ilustra el uso de la gestión de excepciones en el trato con los valores nulos, estimaciones y mediciones a través de un operador.

El operador de Borgida coincide con el operador de Winslett, salvo en el caso en que la nueva información y la base de conocimiento sean congruentes en cuyo caso la teoría revisada de Borgida es simplemente la base de conocimientos y la nueva información ($K \wedge \phi$) [16].

3.5. Método Forbus

Conocido por ser un operador de actualización, Forbus establece un enfoque que tiene en cuenta la cardinalidad: Sea K_M, P la cardinalidad mínima en el conjunto $card(K, \phi)$. Los modelos de la teoría actualizada Forbus son:

$M\ (K *_F A) = N \in M(A) \mid \exists M \in M(T) : card(M \Delta N) = K_{M,A}$.

Teniendo en cuenta que por medio de la cardinalidad, Forbus puede comparar (y desechar) modelos que son incompatibles con el método de Winslett, ésto recordando los enfoques donde la proximidad entre los modelos de A y K se define considerando globalmente todos los modelos de K. En otras palabras, este enfoque considera al mismo tiempo todos los modelos pares de $M \in M(K)$ encontrando todos los pares más cercanos.

$\delta(K, A)$ designa el conjunto de diferencias mínimas entre un modelo de A y uno de K, más precisamente:

$\delta(K, A) = min(\cup_M \in_M (K))\mu(M, A)$

Se utiliza la notación $k(K, A)$ para denotar la cardinalidad mínima de conjuntos $\delta(K, A)$.

Capítulo 4

Propuesta para la revisión de creencias

En éste apartado se describe la propuesta para la revisión de creencias, el método que se seguirá para el diseño del operador de revisión y posteriormente, se abordará el problema de revisión de creencias.

4.1. Metodología

La metodología que utilizaremos para el diseño del algoritmo estará dividida en cuatro fases, como se muestran en la figura 4.1.

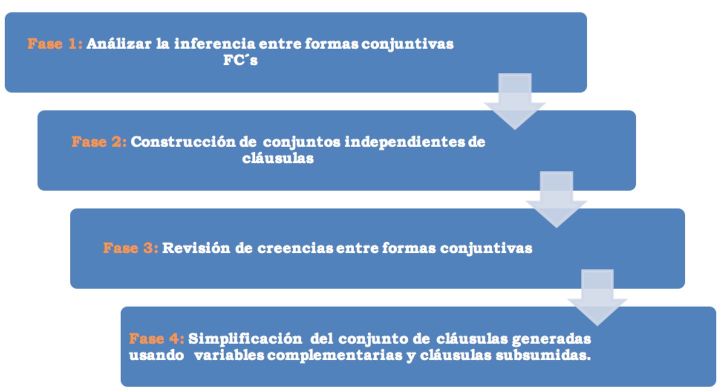

Fig. 4.1. Esquema gráfico de la metodología

4.1.1. Fase 1: Análisis de la inferencia entre formas conjuntivas FC's

Un problema fundamental en el razonamiento deductivo es el problema de la implicación lógica: dada una KB K y una fórmula ϕ, se debe decidir si $K \models \phi$.

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.

Dasdas dos FCs $F1$ y $F2$, 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 de la clase de complejidad Co-NP completo [23]. 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 [7]. 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 $S \nsubseteq Fals(K)$. Si $K \not\models \phi$, entonces $S = (Fals(\phi)$ - $Fals(K)$ $) \neq \emptyset$. En este caso, el 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 $K \models \phi$

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, evite la pérdida de mucha información de K [21].

El método que se propone 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, $SAT(K') \subset SAT(K)$.

4.1.2. Fase 2: Construcción de conjuntos independientes de cláusulas

En esta fase se introducirá una notación corta que permitirá expresar al conjunto de asignaciones que falsifican a una cláusula.

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 - $\mid C_i \mid$ 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.

Se usará 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 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 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 [22].

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, se puede decir que ambas cadenas son dependientes.

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

$$C = (C \vee \neg x) \wedge (C \vee x) \tag{4.1}$$

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 $L = \{x_1, x_2, ..., x_p\}$ 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 (4.1) 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 (4.1) 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 cláusulas independientes hasta x_n, 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:

$$\begin{aligned} C_1 \wedge C_2 = C_1 \wedge (C_2 \vee \neg x_1) \wedge (C_2 \vee x_1 \vee \neg x_2) \\ \wedge \ldots \wedge (C_2 \vee x_1 \vee x_2 \vee \ldots \vee \neg x_p) \end{aligned} \tag{4.2}$$

Las cláusulas del lado derecho de la ecuación (4.2) son independientes por construcción.

El operador central para revisar la 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 $Ind(\varphi, C)$, se denotará como $T_{Ind}(|\varphi|, |C|)$ y depende directamente del tiempo para ejecutar operaciones básicas entre conjunto de literales. Por ejemplo, la operación $lit(C) - lit(\varphi)$ podría realizarse como: para cada $x \in lit(C)$ revisar si $x \in lit(\varphi)$ or 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 $it(\varphi)$ 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 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.

4.1.3. Fase 3: 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 $S \nsubseteq Fals(K)$. 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 cumple que: $K'_i \models \varphi_i$, y además, K'_i sigue siendo una FC.

Presentamos algorítmicamente este proceso.

Algorithm 1 Procedimiento del operador $Ind(\varphi_i, K)$

Input: K: Una KB, 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, ésto se mostrará en la sección de validez y complejidad del algoritmo.

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 todo los ejemplo 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 4.1, se va mostrando el resultado de $Ind(\varphi_i, C_j)$.

Dadas 2 cláusulas C_i, C_j que difieren en 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 (4.1).

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 a 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

Tabla. 4.1. Construyendo el operador $Ind(\phi, K)$

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

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: $K' \models \phi$.

4.1.4. Fase 4: Simplificación del conjunto de cláusulas generadas usando variables complementarias y cláusulas subsumidas.

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 $\varphi_i = Ind(\varphi_i, C_j)$ 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 (4.2), 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 $Ind(\varphi_i, C_j)$ es 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 del método.

Se mostrará ahora que el conjunto de cláusulas en $Ind(\varphi_i, C_j)$, representa el conjunto mínimo de cláusulas que permiten cubrir el conjunto de asignaciones en $Fals(\varphi_i) - Fals(C_j)$, que es el espacio mínimo de asignaciones necesario 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 $Fals(Ind(\varphi_i, C_j))$ contiene el conjunto de asignaciones mínimo posible 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' será un subconjunto de los modelos de K, $SAT(K') \subseteq SAT(K)$.

Sin embargo, al iterar ahora sobre cada $\varphi_i \in \phi, i = 1, \ldots, k$, se tiene que los k conjuntos de cláusulas Fs_i que conforman $Ind(\phi, K)$ podrían no tener un número mínimo de cláusulas, ya que se puede reducir el número de cláusulas en $Ind(\phi, K)$ aplicando la reducción $Varcom$.

Después de obtener las cláusulas de $Ind(\phi, K)$, se reduce su número, 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 por 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.

Una metodología similar a la que empleamos en la reducción de la FC, es el cálculo de los implicantes primos 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 para la función [25].

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 $\phi = (\neg p \vee \neg s \vee \neg t) \wedge (q \vee \neg r \vee \neg s \vee \neg t) \wedge (\neg p \vee q \vee r \vee \neg s \vee t) \wedge (\neg p)$, 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 4.2, se va calculando $Ind(\varphi_i, C_j)$.

Como se puede observar en la tabla 4.2, al aplicar el operador $Ind(\phi, K)$ se genera un número mayor de combinaciones a los que muestra la tabla 4.3, debido a que en la tabla 4.3, 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.

Tabla. 4.2. Aplicación del operador $Ind(\phi, K)$

K / ϕ	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***

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 4.3. Con lo que se tiene 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, $K' = K \wedge Fs = (\neg p \vee q \vee r \vee s) \wedge (\neg p \vee \neg q) \wedge (\neg p \vee \neg r) \wedge (\neg p \vee q \vee r \vee \neg s) \wedge (p \vee q \vee \neg r \vee \neg s \vee \neg t)$.

Tabla. 4.3. Cálculo del operador Ind con las cláusulas $C_i \in K$ ordenadas

ϕ \ 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***

Capítulo 5

Validez y complejidad del Algoritmo

La complejidad es una noción utilizada en diferentes campos como la filosofía, la biología, la física, la matemática y con un gran impacto en computación. Consiste en la forma de analizar y de reflexionar sobre determinados aspectos de la naturaleza, la sociedad y el pensamiento, los cuales presentan ciertas características que los clasifican como sistemas de comportamiento complejo [26].

Estos sistemas de comportamiento complejo requieren de un programa que mida el grado de complejidad por la cantidad de información que contengan o por el número de bits o longitud del programa.

5.1. Teoría de la complejidad

La teoría de la complejidad computacional es una rama de la teoría de la computación que se centra en clasificar los problemas computacionales de acuerdo a su dificultad y en la relación entre dichas clases de complejidad

Un problema se caracteriza como *inherentemente difícil* si su solución requiere de una cantidad significativa de recursos computacionales, sin importar el algoritmo utilizado. La teoría de la complejidad computacional formaliza dicha afirmación, introduciendo modelos de cómputo matemáticos para el estudio de estos problemas y la cuantificación de la cantidad de recursos necesarios para resolverlos, como tiempo y memoria.

Su objetivo es determinar los límites prácticos de que es lo qué se puede hacer en una computadora y qué no. Una diferencia significativa entre el análisis de algoritmos y la teoría de la complejidad computacional, es que el primero se dedica a determinar la cantidad de recursos requeridos por un algoritmo en particular para resolver un problema, mientras que la segunda, analiza todos los posibles algoritmos que pudieran ser usados para resolver el mismo problema. Para poder referirnos a problemas como *inherentemente intratables* y problemas de difícultad *equivalente*, es necesario comprender algunos términos más básicos que introduciremos en este capitulo.

5.1.1. Problema computacional

Un problema computacional constituye una pregunta a ser respondida, teniendo generalmente varios parámetros, o variables libres, cuyos valores no se han especificado.

5.1.2. Problemas de decisión

Un problema de decisión es un tipo especial de problema computacional cuya respuesta es solamente sí o no. El objetivo es decidir, con la ayuda de un algoritmo, si una determinada entrada es un elemento del lenguaje formal considerado. Si el algoritmo devuelve como respuesta *sí*, se dice que el algoritmo acepta la entrada o de lo contrario se dice que la rechaza.

La complejidad NP-completitud se aplica directamente a estos tipos de problemas en vez de problemas de optimización.

5.1.3. Algoritmo

Es un procedimiento utilizado para resolver problemas, de manera general, lo que interesa es encontrar el algoritmo *más eficiente* para resolver cierto problema. Por algoritmo *más eficiente* usualmente nos referimos al más rápido, debido a que los requerimientos de tiempo son usualmente un factor dominante cuando se trata de determinar si un algoritmo es lo suficientemente eficiente para ser útil en la práctica.

Una distinción es aquella que se hace cuando se habla de tiempo polinomial y exponencial cuando se trata de caracterizar a los algoritmos como *suficientemente eficiente* y *muy ineficiente* respectivamente. Un algoritmo de tiempo polinomial se define como aquel cuya complejidad en el peor caso esá acotada por arriba por un polinomio sobre el tamaño de su entrada, Si n es el tamaño de la entrada, entonces, existe un polinomio $p(n)$ tal que $T(n)$ es $O(p(n))$. Cualquier algoritmo cuya función de complejidad no pueda ser acotada de esta manera, se denomina algoritmo de tiempo exponencial.

La mayoría de los algoritmos de tiempo exponencial son simples variaciones de una búsqueda exhaustiva, mientras que los algoritmos de tiempo polinomial, usualmente se obtienen mediante un análisis más profundo de la estructura del problema.

5.2. Clases de complejidad Computacional

Son un conjunto de problemas que poseen la misma complejidad computacional. En la figura 5.1 mostramos un esquema de clasificación de las distintas clases de complejidad.

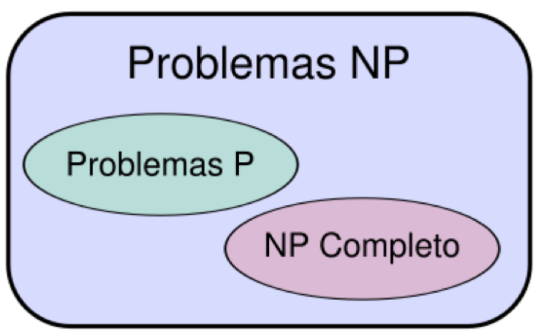

Fig. 5.1. Clasificación de clases de complejidad

5.2.1. Máquinas de Turing deterministas y la clase P

La clase de complejidad P contiene a aquellos problemas que son solubles en tiempo polinomial por una máquina de Turing determinista. Pero es importante mencionar que existen distintas variantes de la máquina de turing y es conocido que la más débil de ellas puede simular a la más fuerte, adicionando a lo mas un tiempo polinomial.

La clase P juega un papel importante en la teoría de la complejidad computacional debido a que P es invariable para todos los modelos de cómputo que son polinomialmente equivalentes a la máquina de turing determinista. A grandes rasgos, P corresponde a la clase de problemas que, de manera realista, son solubles en una computadora.

5.2.2. Computación no determinista y la clase NP

La clase de complejidad NP consta de los problemas *verificables* en tiempo polinomial. Por verificable se entiende a un problema tal que dado un certificado de solución (candidato a solución), se puede verificar que dicho certificado es correcto en un tiempo polinomial en el tamaño de la entrada. A los problemas en la clase NP usualmente se les llama problemas NP.

La relación entre las clases P y NP es fundamental para la teoría de la $NP-$*completitud*. Intuitivamente, creemos que P es un subconjunto de NP. Y efectivamente, cada problema de decisión resuelto por un algoritmo de tiempo polinomial determinista, también puede ser resuelto por un algoritmo de tiempo polinomial no determinista.

5.2.3. Problemas NP-completos

La clase de los problemas $NP-$*completos* contiene a los problemas más difíciles en NP, en el sentido de que son los que están más lejos de estar en P. Debido a que el problema $P = NP$ no ha sido resuelto, el hecho de reducir un problema B, a otro problema A, indicaría que no se conoce solución en tiempo polinomial para A. Esto es debido a que una solución en tiempo polinomial para A, tendría como consecuencia la existencia de una solución polinomial para B. De manera similar, debido a que todos los problemas NP pueden ser reducidos a este conjunto, encontrar un problema $NP-completo$ que pueda ser resuelto en un tiempo polinomial significaría que $P = NP$.

Quizá la razón de mayor peso por la cual los científicos de la computación creen que P es distinto de NP, es la existencia de la clase de problemas $NP-completos$. Esta clase tiene la curiosa propiedad de que si algún problema $NP-completo$ puede ser resuelto en tiempo polinomial, entonces todo problema en NP tiene una solución en tiempo polinomial, es decir, $P = NP$.

Teniendo todos estos conceptos acerca de la complejidad de tiempo podemos decir que la revisión de creencias siendo un problema $NP-completo$, permite el desarrollo de nuevos algoritmos que puedan resolver problemas de forma eficiente.

5.3. Análisis de complejidad del operador *Ind*

La función de tiempo del operador de revisión de creencias $K \circ \phi$, se denotará 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 $Ind(\varphi_i, K), \forall \varphi_i \in \phi$.

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:

$S_{i0} = v(\varphi_i)$

$S_{i1} = v(C_1)$ - $v(\varphi_i)$

S_{i2}=$v(C_2)$ $(v(C_1) \cup v(\varphi_i))$ $S_{im} = v(C_m)$ $(v(C_{m-1}) \cup \ldots \cup v(C_1) \cup v(\varphi_i))$

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.

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 sobresel número de variables que se genera de aplicar el operador $Ind(\varphi_i, Cj)$ 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: $T_0(|\phi|, |K|) \leq Max\{|S_{i1}| * |S_{i2}| * \ldots * |S_{im}| : \forall\varphi_i\in\phi\} \in O(2^{(n-min\{|\varphi_i|:\varphi_i\in\phi\})})$.

Capítulo 6

Resultados obtenidos

En esta sección se presentan en primera instancia los resultados obtenidos del análisis de los postulados KM sobre nuestra propuesta para la revisión de creencias y posteriormente la comparación de nuestro método con respecto a los existentes.

6.1. Cumplimiento de los postulados KM

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 $\phi 1 \equiv \phi 2$, entonces $K1 \circ \phi 1 \equiv K2 \circ \phi 2$.
- **(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 $\phi = (\neg p)$, 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 $\phi 1 \equiv \phi 2$, se cumple que $Fals(K1) = Fals(K2)$ y $Fals(\phi 1) = Fals(\phi 2)$. Al trabajar nuestro método sobre los conjuntos $Fals(Ind(\phi, K))$ y al ser tanto K, ϕ y $Ind(\phi, K)$ FCs, 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 dado que tanto ϕ como γ son FCs.

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 Corolario1. 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**.

6.2. Comparativa de los métodos de revisión de creencias

En resumen de acuerdo a la primera propuesta, Dalal [11] sugiere un operador de revisión basado en la distancia mínima Hamming entre las interpretaciones una vez que se extiende a distancias entre interpretaciones y bases. En la práctica, esta propuesta implica un cálculo de modelos que pueden ser muy costosos, 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 [27] 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 máximos consistentes posibles.

La propuesta de Borgida y Forbus es similar a la de Winslett, con la diferencia de que Borgida 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 también a concluir que el problema de revisión de creencias bajo estos métodos es un problema intratable (de complejidad en tiempo inherentemente exponencial).

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 [29], [28], [27] y obteniéndose diferentes resultados sobre la complejidad en tiempo de estas propuestas [11], [29]. Algunas de las investigaciones en esta dirección se han dedicado principalmente al fragmento Horn de la lógica clásica.

Existen diversas propuestas que involucran fórmulas proposicionales tales como las descritas en [30], [28], [25], que sugieren métodos que abordan algunos fragmentos de cláusulas proposicionales. Una de las propuestas recientes debida a Delgrande presenta los primeros resultados sobre cambio de creencias en el fragmento Horn [30].

En [10], los autores presentan una metodología general para definir nuevos operadores de revisión derivados de operadores estandar (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 esta es aplicable a otros fragmentos de la lógica proposicional, donde los modelos de las fórmulas son cerrados bajo una función booleana.

Como se ha observado, estas propuestas se desarrollan sobre cláusulas generales o bien sobre fragmentos Horn. Así, existen diversas propuestas que han abordado el problema de revisión de creencias, cada una de estas propone un operador que trabaja sobre cláusulas generales presentando diversos inconvenientes, tales como: sólo conocer algoritmos de complejidad exponencial en tiempo para el cálculo de los modelos. Algunos de los operadores son dependientes de la sintáxis o bien requieren de información adicional. Otros definen diferentes nociones de proximidad, y unos más se limitan a la revisón de bases de conocimientos consistentes, lo que conlleva a la diferencia entre estas propuestas.

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 que construye una nueva base de conocimientos denotada como K' a partir de la revisión de $K \circ \phi$, considerando que K es la base de conocimientos y ϕ una nueva consulta ambas en 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, que demuestra la corrección de nuestra propuesta, la verificación de cumplimiento de los postulados KM, así como el análisis de la complejidad en tiempo.

Referencias

[1] Muñoz Melendez Angelica (2015). *Inteligencia Artificial*, Presentación, fundamentos, historia, conceptos, Facultad de Ciencias de Computacionales, INAOE.

[2] López Takeyas Bruno (2016). *Introducción a los sistemas expertos*, Conferencia de Inteligencia Artificial, Instituto Tecnológico de Nuevo Laredo.

[3] Laborda Jorge (2011). *Luna y civilización*, Relaciones de la Luna sobre la vida y la inteligencia en el universo, Editorial Hélice, ISBN: 978-884-92914-09-8, pag. 68-69.

[4] De Ita Guillermo, García Alma Delia, Zacarias Fernando (2016). *Model-Based Algorithm for Belief Revisions between Normal Conjuntive Forms*, Research in Computing Science 112, pag. 29-39

[5] Darwiche A. (2001). *On tractable counting of theory models and its aplication to truth maintenance and belief revision*, Journal of Applied Non-classical Logics, pag. 11-34.

[6] Ellis D. (2010). *Irredundant Families of Subcubes*, Mathematical, Proc. Of the Cambridge

[7] G. De I. Luna, F. Z. Flores, A. (2015). *Model-Based Algorithm for Propositional Belief Revisions*, IEEE Latin transactions 13(4), pag. 1055-1060.

[8] Eleonora Cresto (2002). *Revisión de creencias y racionalidad*, cuaderno del cimbage n5

[9] C. E. Alchourron, P. G. Makinson, D. (1985). *On the logic of theory change: Partial meet contraction and revision functions*, Journal of Symbolic Logic, 50, pag. 510-530

[10] Liberatore, P. Schaerf, M. (2001). *Belief revision and update: Complexity of model checking*, Journal of Computer and System Sciences, 62, pag. 43-72.

[11] Dalal, M. (1988). *Investigations into theory of knowledge base revision*, Journal of In Proc.,The Seventh National Conference on Artificial Intelligence.

[12] Nadia Creignou, Odile, Papini Reinhard Pichler. Stefan Woltran (2012). *Belief revision within fragments of propositional logic*, Institut fur abteilung datenbanken und artificial intelligence, pag. 3-17.

[13] Ruslán Ledesma Garza (2009). *Sobre las relaciones de implicación entre los postulados AGM bajo lógicas no clásicas*, Tesis profesional de licenciatura en Sistemas Computacionales, capitulo 4.

[14] Katsuno, H. Mendelzon, A. O. (1991). *On the difference between updating a knowledge base and revising it*, TKR'91 Cambridge, MA, USA, 1, pag. 387-394.

[15] Mathias Medina(2008). *Sobre la revision iterada de creencias, tesis de licenciatura*, Universidad de los andes, Departamento de Matemáticas

[16] Paolo Liberatore, Marco Schaerf (2001). *The complexity of model checking for belief revision and update*, Departamento di informatica, Università di Roma "La sapienza".

[17] Raúl Carnota (2006). *Carlos Alchourrón y la inteligencia artificial*, Maestría de Epistemologá e Historia de la Ciencia, Universidad Nacional de Tres de Febrero, Buenos Aires, Argentina.

[18] Erik Olsson, Sebastian Enquist (2010). *Belief revision meets philosophy of science*, ISBN: 978-90-481-9608-1.

[19] Alexander Borgida (1985). *Language Features for Flexible Handling of Exceptions in Information Systems*, Department of Computer Science Rutgers University.

[20] Joaquin Gonzales (2009). *La teoría de la complejidad*, Universidad nacional de colombia, Dyna, Vol 76, número: 157, pag. 243-245.

[21] Dalal, M. (1988). *Knowledge base revision*, In Proc. The Seventh National Conference on Artificial Intelligence. AAAI, pag. 475-479.

[22] Doubois, O. (1991). *Counting the number of solutions for instances of satisfiability*, Theoretical Computer Science, 81, pag. 49-64.

[23] Papadimitriou, C. H. (1994). *Complexity*, Addison-Wesley Pub. Computational

[24] JSTOR (1952). *He problem of simplifying truth functions*, JSTOR, 59, pag. 521-531.

[25] Petrick, S. (1956). *A direct termination of the irredundant forms of a boolean function from the set of primer implicants*, In Technical report. Cambrige Res Center, pag. 56-110.

[26] Khardon R. Roth D. (1996). *Reasoning with models*, Artificial Intelligence, 87, pag. 187-213.

[27] Satoh, K. (1988). *Nonmonotonic reasoning by minimal belief revision*, Institute for New Generation Computer Technology, 358, pag. 157-170.

[28] Néstor Jorge Valdéz, M. A. F. (2013). *Dinámica de conocimiento: contracción múltiple en lenguajes horn*, In Proc. XIX congreso argentino de ciencias de la computación. Libro de actas - CACIC2013, pag. 61-70.

[29] Nebel, B. (1998). *How hard is it to revise a belief base?*, Handbook of Defeasible Reasoning and Uncertainty Management Systems, 3, pag. 77-145.

[30] Delgrande, J. P. (2008). *Horn clause belief change: Contraction functions*, In Proc. of the eleven International Conference on the Principles of Knowledge Representation and Reasoning. AAAI, pag. 156-165.