An Algorithm to Belief Revision and to Verify Consistency of a Knowledge Base

P. Bello, and G. De Ita

Abstract— **The belief revision process involves several problems considered hard. One of the crucial problems is how to represent to the knowledge base K to consider, as well as how to represent and to add new information φ, which may even be contradictory to the knowledge base. In this work, both the knowledge base and the new information are in conjunctive normal form. Each clause of a conjunctive normal form is encoded by a string consisting of: 0, 1, *, representing the falsifying assignments of the clause. To use the falsifying assignments of the clauses allows to perform efficiently different logical operators among conjunctive forms. Our belief revision process ($K * \varphi$) between conjunctive forms is based on solving first the propositional inference, i.e. $K \models \varphi$. Based on to count falsifying assignments represented by tertiary chains, an algorithmic proposal is made that allows to determine in a practical way, when ($K \cup (K * \varphi)$) is inconsistent. Finally, the time-complexity analysis of our algorithmic proposal is carried out.**

Index Terms— **Belief revision, Propositional inference, Knowledge base, Consistency, Satisfiability problem**

I. Introduction

La revisión de creencias es un problema clásico en Inteligencia Artificial y permite modelar parte del razonamiento humano. La revisión de creencias consiste en incorporar nuevas creencias a una base de conocimiento ya establecida, con cambios mínimos que permitan preservar el máximo de las creencias originales, mantener su consistencia y privilegiar la nueva información con respecto a las creencias ya existentes.

La teoría que prevalece en la revisión de creencias es la teoría conocida como paradigma AGM [16], propuesta por Alchourrón, Gärdenfors y Makinson [2]. El modelo AGM privilegia a la nueva información con respecto a las creencias ya existentes en el conocimiento del agente. Posteriormente, Katsuno y Mendelzon [14] unificaron los diferentes enfoques de revisión de creencias semánticas, y reformularon los postulados AGM, conociéndose ahora cono postulados KM.

La revisión de creencias ha sido ampliamente estudiada en el marco de la lógica proposicional, pero más recientemente, se ha analizado el comportamiento de la revisión de creencias, incluyendo el resultado de la revisión, sobre fragmentos propios de la lógica proposicional, por ejemplo, acotando que el proceso de revisión se realice y produzca solo sobre cláusulas de Horn [7].

P. Bello, Benemérita Universidad Autónoma de Puebla, Doctorado en Ingeniería del Lenguaje y del Conocimiento, 14 sur y Av. San Claudio, San Manuel C.U., Puebla, México, pb5pbello@gmail.com

G. De Ita, Benemérita Universidad Autónoma de Puebla, México, deita @cs.buap.mx.

Autor Corresponsal Pedro Bello López

La lógica proposicional se basa en el uso de oraciones que afirman o niegan algo y que, por lo tanto, pueden ser verdaderas o falsas, a lo que se denomina proposiciones. La lógica proposicional [5], [6], [9], [19] se utiliza para analizar razonamientos formalmente válidos, partiendo de proposiciones y operadores lógicos para poder construir fórmulas que están operando sobre las variables proposicionales. En [7] se presenta un ejemplo aplicado a la revisión de creencias, donde los modelos de fórmulas son cerrados bajo funciones Booleanas. En [4] se estudia la transformación de los sistemas de bases de conocimientos condicionales, que permiten identificar y eliminar condicionales innecesarios de la base.

Utilizaremos la notación $M \models \phi$ para plantear si es posible demostrar que la formula ϕ se deriva lógicamente de la fórmula supuesta M. El operador $\models$ es una extensión del conectivo implicación entre dos enunciados, dado que ϕ y M pueden tratarse de conjuntos de enunciados. En el área de revisión de creencias, podemos asumir que M representa nuestra base de creencias actuales, mientras que ϕ es nueva información que nos dan a conocer y de la cual, nos planteamos si ésta es deducible (o no) a partir de M.

La implicación proposicional es un problema fundamental para el razonamiento automático, además de ser una tarea relevante en muchos otros problemas propios de la Inteligencia Artificial, tales como: estimar el grado de creencia, revisión y actualización de creencias, construir explicaciones abductivas, reconocer y contar modelos [8], [10], [15], [23]. Así como en muchos otros procedimientos usados en áreas de la Inteligencia Artificial.

El problema de la implicación lógica entre fórmulas conjuntivas es un reto computacional dado que es un problema de la clase Co-NP completo [15], [16], puesto que es lógicamente equivalente a decidir la tautologicidad de una forma normal disyuntiva, el cual es un problema clásico de la clase Co-NP completo.

Recordemos que la clase de complejidad P consta de problemas que pueden resolverse en tiempo polinomial por procesos deterministas. Mientras Co-NP se forma por problemas complementos de problemas en la clase NP (que son problemas que pueden resolverse en tiempo polinomial, pero mediante procesos no deterministas). En tanto que un problema A es completo en una clase cuando todo problema de la clase se puede reducir en tiempo polinomial determinista al problema A.

P^A (NP^A) son problemas que se resuelven en tiempo polinomial por procesos deterministas (no deterministas) usando un oráculo A, asumiendo que la consulta y respuesta de

la máquina oráculo se realiza en un solo paso de computación. Se puede considerar al oráculo como una 'subrutina sin costo'. En su forma general, el problema de la revisión de creencias para el cálculo proposicional está en la clase NP^{NP}, ubicándolo así en el segundo nivel de la jerarquía polinomial [18].

En este trabajo estructuramos el proceso de revisión de creencias en términos de oraciones en un lenguaje formal L de lógica proposicional, de forma que el estado epistémico de un agente está representado por el conjunto K de oraciones de L que éste acepta. De esta manera, K representa el conjunto de creencias del agente, donde K debe ser consistente (que K no sea contradictorio) y se cumple $Cn(K) = K$, esto es, K es equivalente al conjunto de sus consecuencias lógicas $Cn(K)$.

El algoritmo que proponemos recibe como entrada una base de conocimiento consistente K en Forma Normal Conjuntiva (FNC) y al considerar nueva información φ, se verifica si K continua siendo consistente o no. Se presentan así tres posibilidades; que la cláusula φ sea subsumida por K, que φ sea independiente de K, o que de φ se generen las cláusulas necesarias S para que se cumpla $(K \cup S) \models \varphi$.

Al realizar el operador de revisión de creencias ($K * \varphi$) entre formulas del cálculo proposicional, muchos autores proponen construir los conjuntos de modelos Mod(K) y Mod(φ) [6],[13],[15],[17] y otros la formación de los conjuntos de los mundos plausibles de ($K \models \varphi$) [1],[11],[18] lo que implica desde los primeros pasos de la revisión, el realizar procesos de orden exponencial.

En nuestra propuesta, en lugar de calcular Mod(K) y Mod(φ) se trabaja con los conjuntos de asignaciones falsificantes Fals(K) y Fals(φ) de las FNC's K y φ, aprovechando que la caracterización de estas fórmulas mediante cadenas falsificantes, involucra procesos de orden lineal sobre el número de las cláusulas en ambas fórmulas.

Esta representación de las FNC's permite determinar $K \models \varphi$, puesto que esto ocurre cuando Fals(φ) $\subseteq$ Fals(K). Parte de lo novedoso de nuestra propuesta es la revisión de la consistencia del resultado de la revisión con la KB, esto es, reconocer la satisfactibilidad de ($K \cup (K * \varphi)$). Nuestra propuesta permite que la revisión de esta satisfactibilidad pueda hacerse al mismo tiempo que se calcula ($K * \varphi$), si es que se conoce el número de modelos iniciales que hay en las fórmulas K y φ.

El crecimiento exponencial en nuestro proceso de revisión: ($K * \varphi$), estriba en el cálculo de (Fals(φ) - Fals(K)), dado que se puede generar un número de cláusulas acotado superiormente por O(2^r), siendo r el número máximo de asteriscos en una cláusula de φ. La velocidad del proceso de revisión se acelera en proporción directa a que la longitud de las cláusulas de la k-FNC φ se incremente (dado que el número de asteriscos es $r = n - k$).

II. Conceptos básicos

El alfabeto para la lógica proposicional consiste de un conjunto numerable de proposiciones que llamaremos variables Booleanas denotadas por $X = \{x_1, ..., x_n\}$. Un conjunto de conectivas lógicas binarias: $\wedge$ (y), $\vee$ (o), $\supset$ (implicación), y una conectiva lógicas unaria $\neg$ (negación). Una literal es una proposición x o su negación $\neg x$. Se componen las proposiciones a través de las conectivas lógicas para formar proposiciones compuestas. Ejemplo de proposiciones compuestas son las cláusulas que se forman como disyunción de literales. Así como las frases que son conjunciones de literales. Una k-cláusula (k-frase) es una cláusula (frase) con exactamente k literales.

Una forma normal conjuntiva (FNC) es una conjunción de cláusulas, y una k-FNC es una FNC que contiene k-cláusulas. Una forma normal disyuntiva (FND) es una disyunción de frases, y una k-FND es una FND que contiene k-frases. Diremos que una variable $x \in X$ aparece en una FNC si x o $\neg x$ es un elemento de la FNC.

Trabajaremos considerando solo FNC's como fórmulas de entrada para nuestro algoritmo de revisión de creencias, lo que no restringe nuestro lenguaje de trabajo, dado que se sabe que toda fórmula proposicional es equivalente a una FNC, además de existir algoritmos que realizan tal conversión [12].

Una asignación s sobre una fórmula F es una función $s : v(F) \rightarrow \{0,1\}$, que asocial uno de dos valores Booleanos 0 o 1 a cada una de las proposiciones componentes de F ($v(F)$ denota las letras proposicionales que componen a F). Denotaremos a $s(F)$ como la evaluación de la asignación sobre la fórmula F, lo que produce un valor de 0 o de 1. Si C es una cláusula diremos que $s(C)=1$ (s satisface a C) si alguna literal es común a s y a C. Mientras que s satisface a una FNC F, esto es $s(F)=1$, si toda cláusula C en F cumple que $s(C) =1$.

Un modelo F ($Mod(F)$) es una asignación s que satisface a F. Una formula F se dice satisfactible o consistente si el número de modelos (denotado como $\#Mod(F)$) es mayor a cero, en caso contrario, se dirá que es insatisfactible o inconsistente.

El problema de satisfactibilidad considera como entrada una fórmula F, y detemina si hay una asignación s tal que $s(F) = 1$ (o determinar que toda posible asignación s, hace $s(F) =0$).

Dado un par de cláusulas C_1 y C_2, cuando $lit(C_1) \subseteq lit(C_2)$, entonces se dice que C_2 es subsumida (Sub) por C_1. Dadas dos cláusulas C_i y C_j, si aparece al menos una literal complementaria entre ambas cláusulas, entonces se denominan cláusulas independientes.

Sea K una base de conocimiento y sea φ una nueva información, decimos que K infiere semánticamente a φ, denotado por $K \models \varphi$, si φ se satisface para cada modelo de K, es decir, si $Mod(K) \subseteq Mod(\varphi)$. La revisión de creencias vista como inferencia proposicional tienen los siguientes hechos:

- $K \models \varphi$ si y sólo si $Mod(K) \subseteq Mod(\varphi)$.
- $K \models \varphi$ si y sólo si $Fals(\varphi) \subseteq Fals(K)$.

Donde $Fals(F)$ denota el conjunto de asignaciones que hacen falsa a la fórmula F. En el caso de una cláusula $C_i=\{x_{i1}, x_{i2}, \ldots, x_{ik}\}$, una asignación que falsifica C_i tiene en cada posición i_1 hasta i_k el valor que falsifica a la literal. Por ejemplo, si $xi \in C_i$, en su cadena falsificante se le asigna el valor 0. De otra forma, si $\neg xi \in C_i$, se le asigna el valor 1. La literal que no aparece en C_i es representada por el símbolo *, lo que significa que puede tomar un valor lógico en el conjunto {0,1}. Llamamos patrones falsificantes a la cadena de 0,1 y * que representa la falsificación de C_i.

Consideramos los elementos básicos de la teoría AGM para el proceso de revisión de creencias. Dada K una base de

conocimiento, y una nueva información φ a ser incluida en K, se tiene que:

i) Se supone que K y φ son consistentes, es decir cada formula tiene por lo menos un modelo.

ii) Se da prioridad a la nueva información φ con respecto a la ya existente en K.

iii) Se utiliza el principio de cambio mínimo sobre la base de conocimiento K [1], es decir, una revisión de creencias debe agregar la nueva información y conservar tantas creencias de las ya existentes como sea posible [8].

Consideremos las operaciones básicas del modelo AGM:

- Expansión: consiste en adicionar a un estado de creencias una nueva creencia, posiblemente junto con sus consecuencias, sin eliminar las creencias existentes. La expansión de K por φ se denota $(K+\varphi)$.

- Revisión: consiste en la modificación del estado epistémico representado como un conjunto de creencias por la que una nueva creencia (la entrada epistémica) se incorpora al conjunto previo conservando consistencia lógica. $(K*\varphi)$ denota la revisión de K por φ.

- Contracción: Una contracción ocurre cuando una sentencia es refutada, es decir, consiste en eliminar cierta creencia. $(K-\varphi)$ denota la contracción de K por φ [11].

III. Algoritmo de inferencia proposicional

El problema de la revisión de creencias de $(K * \varphi)$ requiere inicialmente de conocer si $K \models \varphi$, lo que implica resolver un problema difícil (Co-NP completo). Una forma de resolver de manera práctica este tipo de problemas es introducir algoritmos aproximados para el cambio de creencias, así como propuestas algorítmicas para la revisión de bases de creencias finitas. Sin embargo, esto conlleva a generar soluciones no óptimas. Por lo que se buscan técnicas, métodos o instancias donde se pueda mejorar la eficiencia en el proceso de revisión de creencias [22].

Dada K una base de conocimiento (KB por sus siglas en inglés), y φ una nueva información, ambas expresadas como un conjunto de cláusulas en FNC con la condición de que $K = \bigwedge_{(j=1)}^{m} C_j$ es un conjunto de cláusulas y φ es una FNC, ambas definidas sobre el mismo conjunto de n variables Booleanas, lo que garantiza trabajar sobre un mismo dominio de información.

En este trabajo presentamos un algoritmo para realizar la revisión de creencias $(K*\varphi)$ al mismo tiempo que se estima si la operación $(K \cup (K * \varphi))$ es consistente. Aunque no se presenta una aplicación práctica del algoritmo, hemos identificado la modelación de cambios en la preferencia de consumidores, con el fin de inferir perfiles y comportamientos de los consumidores en un mercado determinado [21], como una posible aplicación potencial de nuestra propuesta.

Inicialmente suponemos que tanto K y φ son consistentes, es decir, tienen al menos un modelo. Se sabe que $K \models \varphi$ si y solo si $Mod(K) \subseteq Mod(\varphi)$.

En nuestra propuesta algorítmica se codifica cada cláusula de una FNC por su cadena falsificante, como se muestra en la Fig. 1. Esto permite operar de forma práctica las operaciones lógicas involucradas en el proceso de revisión de creencias.

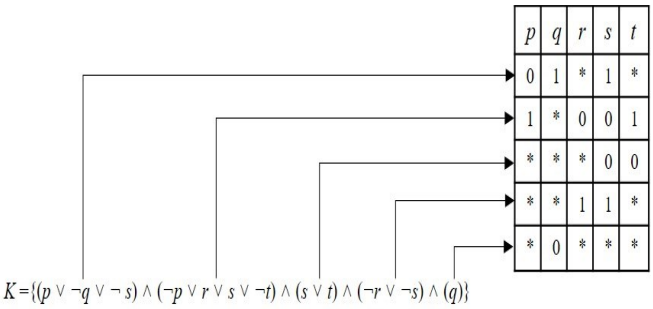

Fig. 1. Ejemplo de trasformación de una FNC a patrones de 1,0 y *.

El algoritmo que proponemos realiza el proceso de revisión de creencias considerando la inferencia proposicional.

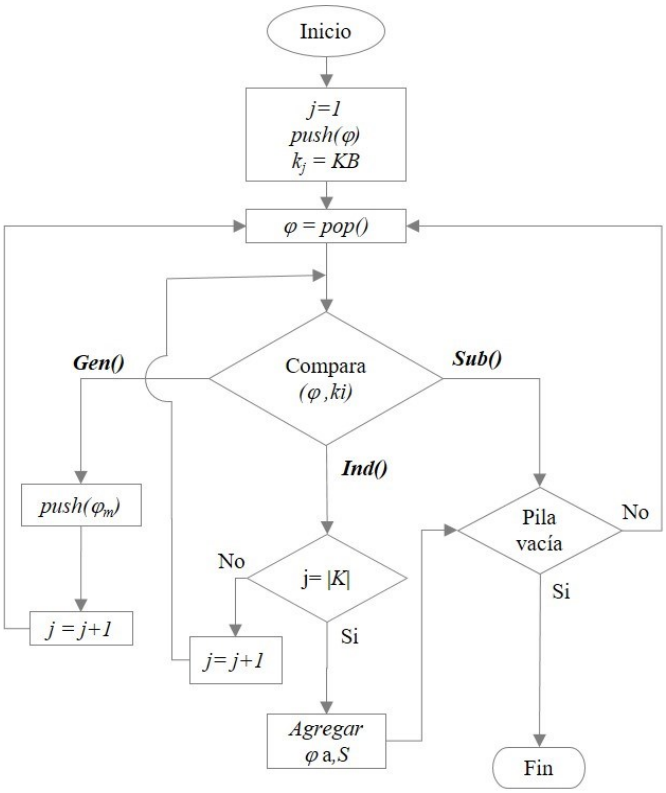

Fig. 2. Diagrama de flujo para la revisión de la base de conocimiento.

El diagrama de la Fig. 2 expresa el proceso general de revisión de creencias para determinar si la base de conocimiento es consistente o deja de serlo al adicionar nueva información. Este proceso de revisión es descrito a continuación:

1) Como entrada se tiene la base de conocimiento $K=\{k_1, k_2, \ldots, k_m\}$ y $\varphi = \{\varphi_1, \varphi_2, \ldots, \varphi_r\}$ conjuntos de cláusulas definidas sobre un mismo conjunto de n variables. Cada cláusula se representa por su cadena falsificante de 0, 1, y *. Se utiliza una pila para almacenar las cláusulas de φ. Además, se inicia $S=\varnothing$ que representa el conjunto de cláusulas a generar a partir del proceso de inferencia.

2) La parte central del algoritmo (diagrama de flujo) consiste en comparar una cláusula φ_i de φ contra cada una de las cláusulas $k_j \in K$. Teniendo tres posibilidades:

a. ***Sub()***: φ_i es una cláusula subsumida, es decir, los modelos de φ_i ya son parte de los modelos de k_j. Si la pila de cláusulas de φ está vacía significa que ya no hay más cláusulas que revisar y termina el proceso, si la pila no es vacía, se toma la siguiente cláusula de la pila.

b. ***Ind()***: este operador indica que entre φ_i y k_j hay por lo menos una literal complementaria. Si ya se ha comparado contra todos los k_j entonces la cláusula resultante se agrega al conjunto de salida *S*. En caso contrario, se compara con la siguiente cláusula k_{j+1}.

c. ***Gen()***: esta operación indica la generación de nuevas cláusulas con el fin de lograr la inferencia $K \models \varphi_i$.

3) Al finalizar el proceso de comparación entre cada φ_i contra cada k_j tenemos como resultado el conjunto *S*. Si *S* es vacío significa que la cláusula φ se infiere de *K* por lo que no es necesario agregarla a la base de conocimiento. En caso contrario, *S* contendrá las cláusulas necesarias a agregar a *K* para que se cumpla $(K \cup S) \models \varphi$.

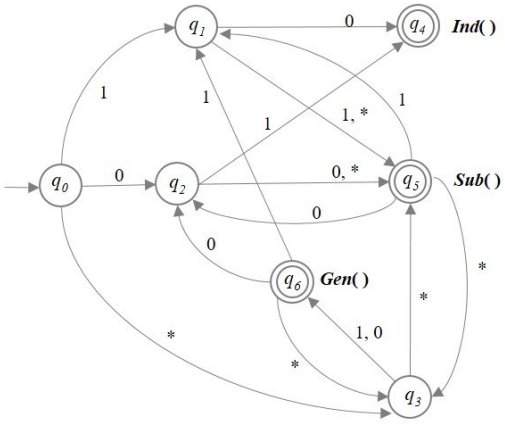

Fig. 3. Diagrama de estados para las operaciones a nivel de 0, 1 y *.

En la Fig. 3. se describen las operaciones a nivel de patrones falsificantes. En el estado inicial q_0, puede entrar un 1, 0 o un * y siguiendo el patrón descrito en dicho diagrama podemos determinar la operación a aplicar: *Sub()*, *Ind()* y *Gen()*. Se compara una cláusula φ_i contra una cláusula k_j. Si son independientes pasa al estado final q_4. Mientras que los estados finales q_5 y q_6 seguirán iterando mientras haya caracteres en φ_i y k_j.

Se enfatizan las operaciones principales del algoritmo (Fig. 2) usando seudocódigo, con $f_i \in \varphi$ y $c_i \in K$.

```
---------------------------------------------
Procedimiento Sub(fi,ci)
Inicio
  Para i←1 hasta sizeof(ci) hacer
    Si !(fi = '0' y (ci = '0' o ci='*') o fi = '1' y (ci = '1' o ci='*') o
fi = '*' y ci = '*')
        Entonces  Retornar false
    Fin_Si
  Fin_Para
  Retornar true
Fin
-------------------------------------------------
```

La Fig. 4 ejemplifica el caso del procedimiento *Sub()*, dada $K = \{(\neg p \vee s)\}$ y $\varphi = \{(\neg p \vee q \vee s \vee \neg t)\}$, y sus respectivos patrones falsificantes, $K = \{(1{**}0{*})\}$ y $\varphi = \{(10{*}11)\}$, revisamos que cada *i*-ésimo elemento en φ esta subsumido en cada *i*-ésimo elemento en *K*.

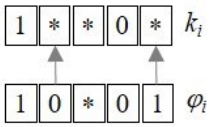

Fig. 4. Cláusula subsumida de acuerdo al procedimiento *Sub()*.

```
-------------------------------------------
Procedimiento Ind(fi,ci)
Inicio
    Para i ←1 hasta sizeof(ci) hacer
      Si (fi = '0' y ci = '1') o (fi = '1' y ci = '0') Entonces
          Retornar true
      Fin_Si
    Fin_Para
    Retornar false
Fin
-------------------------------------------------
```

Ejemplo aplicando el procedimiento *Ind()*. Sea $K=\{(10{*}0{*}), (*110*),(*101*),(110{*}0)\}$ y $\varphi = \{(11{**}1)\}$, obtenemos: $Ind(\varphi,k_1)$ = true, dado que en la posición dos existen literales complementarias. $Ind(\varphi,k_2)$ = false. $Ind(\varphi,k_3)$ = false. $Ind(\varphi,k_4)$ = true, dado que en la última posición hay literales complementarias.

```
-------------------------------------------
Función dif(fi,ci)
Inicio
    k_dif←0
    Para i ←1 hasta sizeof(ci) hacer
      Si (fi = '*' Y (ci = '1' o ci = '0')) Entonces
          k_dif← k_dif+1
      Fin_Si
    Fin_Para
    Retornar k_dif
Fin

Procedimiento Gen(fi,ci)
Inicio
    n←dif(fi,ci)
    Para j ←1 hasta n hacer
      Para i ←1 hasta sizeof(ci) hacer
        Si (fi = '*' y ci = '1') Entonces fi←'0'
          Else Si (fi = '*' y ci = '0') Entonces fi←'1'
              Fin_Si
        Fin_Si
      Fin_Para
      push(fi)
    Fin_Para
Fin
-------------------------------------------------
```

Ejemplo del procedimiento *Gen()*. Sea $K = \{(10{*}0{*}), (*110*),(*101*),(111{**}),(10{**}0)\}$ y sea $\varphi = \{(11{**}1)\}$. Los resultados obtenidos son:

$dif(\varphi,k_1) = 1$. $Gen(\varphi,k_1)=\varnothing$, por ser independientes.
$dif(\varphi,k_2) = 2$. $Gen(\varphi,k_2)=\{(110{*}1),(11111)\}$.
$dif(\varphi,k_3) = 2$. $Gen(\varphi,k_3)=\{(111{*}1),(11001)\}$.
$dif(\varphi,k_4) = 1$. $Gen(\varphi,k_4)=\{(110{*}1)\}$.
$dif(\varphi,k_5) = 0$. $Gen(\varphi,k_5)= \varnothing$, por ser independientes.

IV. Ejemplos de la aplicación del algoritmo

La inferencia proposicional permite determinar si, dado un conjunto de reglas es posible inferir posibles conflictos. En [20], [23] se describe la manera en que sucede esto en la lógica deóntica ("la lógica de lo que debe ser", de lo obligatorio y de lo prohibido). Deon viene del griego que significa "lo que debe ser". Consideremos el siguiente ejemplo basado en el escrito sobre: las conductas indecorosas en la mesa de mi señor (Texto anónimo atribuido a Leonardo Da Vinci [3]).

Para mostrar la importancia de la inferencia lógica y la revisión de creencias modelamos algunos de los hábitos indecorosos en la mesa de mi señor.

- Ningún invitado ha de sentarse sobre la mesa, ni sobre el invitado.
- Tampoco ha de subir los pies sobre la mesa
- No ha de limpiar su armadura sobre la mesa.
- No ha de prender fuego sobre la mesa.
- Mientras permanezca en la mesa puede sentarse o no prender fuego

Utilizando lógica proposicional, tenemos:

p = sobre la mesa
q = sentarse
r = subir los pies
s = prender fuego
t = sobre invitado
u = limpiar armadura

Una vez representada cada oración (proposición) se conectan como cláusulas:

1) $(\neg q \vee \neg p) \wedge (\neg q \vee \neg t)$
2) $(\neg r \vee \neg p)$
3) $(\neg u \vee \neg p)$
4) $(\neg s \vee \neg p)$
5) $(\neg p \vee q \vee \neg s)$

Note que las premisas básicas son transformadas a implicaciones lógicas, que a su vez se reescriben como cláusulas.

Por ejemplo, la proposición 1) significa que: si alguien se sienta, entonces no puede ser sobre la mesa ni sobre el invitado. En forma de implicación sería: $(q \supset (\neg p \wedge \neg t))$ lo que en su forma clausular se escribiría como: $(\neg q \vee \neg p) \wedge (\neg q \vee \neg t)$. Consideremos que la base K se forma por la conjunción de las cláusulas que codifican el discurso: $K = \{(\neg q \vee \neg p) \wedge (\neg q \vee \neg t) \wedge (\neg r \vee \neg p) \wedge (\neg u \vee \neg p) \wedge (\neg s \vee \neg p) \wedge (\neg p \vee q \vee \neg s)\}$.

Si consideramos que φ representa una nueva regla como "Mientras permanezca en la mesa puede sentarse o no limpiar su armadura", que se puede traducir como: $(\neg p \vee q \vee \neg u)$. Veamos si esta última información se puede inferir de la base de conocimiento K.

Usando patrones de 0,1 y *, transformamos la base de conocimiento K y la nueva información φ, obteniendo: K = {(11****), (*1**1*), (1*1***), (1****1), (1**1**), (10*1**)} y φ ={(10***1)}.

En la Tabla I al aplicar el algoritmo para la consistencia de la base de conocimiento, se indica que se realizaron tres comparaciones, dos de independencia y una de cláusula subsumida, al generarse una operación *Sub()* ya no es necesario comparar con el resto de las cláusulas de la base de conocimiento (k_4,k_5,k_6). De esta forma se cumple que $K \models \varphi$, esto es, que φ se infiere lógicamente de K.

TABLA I
REVISANDO LA BASE DE CONOCIMIENTO

K	11****	*1**1*	1*1***	1****1	1**1**	10*1*
φ	10***1	10***1	100**1	100**1		
	Ind()	*Ind()*	*Gen()*	*Sub()*		

A. Contando los Modelos de la Base de Conocimiento

Las operaciones descritas en el algoritmo propuesto de la Fig. 2, son la base principal para determinar el número de modelos de la base de conocimiento. La fórmula (1) es utilizada para determinar el número de asignaciones falsificantes que tiene una cláusula (denotado por $\#Fals(C_i)$), considerando el espacio de n variables de K, donde $\#Asts(C)$ denota el número de asteriscos que tiene la cláusula C.

Para una cláusula C. $\#Fals(C) = 2^r$, con $r = \#Asts(C)$.

$$\#Mod(C) = 2^n - 2^r. \quad (1)$$

Donde $\#Mod(K) = \sum \#Mod(C_i)$, para toda C_i en K, esto sólo es válido si las C_i son todas independientes a pares. Dada una cláusula con cadena falsificante, por ejemplo, C = (0101*), significa que las asignaciones que la hacen falsa son: 01010 y 01011, entonces $\#Fals(C) = 2^1 = 2$.

Sea n el número de variables, se cumple que $\#Mod(C_i) = 2^n - \#Fals(C_i) = 2^n - 2^{\#Asts(C)}$. A continuación, se describe un algoritmo para el conteo de modelos de una *KB* K en FNC.

Algoritmo Conteo de Modelos
Entrada:
Sea $K = \{k_1, k_2, \ldots, k_m\}$ un conjunto de m cláusulas en FNC usando patrones falsificantes con n variables.
Salida:
S, conjunto de cláusulas que definen el conteo de modelos
Inicio
$TotalMod = 2^n - 2^{\#Ast(k1)}$
$S = k_1$
Para cada Clausula k_i en K, i = 2, hasta m **Hacer**
$C = K_i$
NC = Revisar(S, C); // NC = son las nuevas cláusulas al revisar C respecto a S (generar independencia)
$x = 2^{\#Asts(NC)}$; // Obtiene el conteo de asteriscos para $\#Mod(K)$
$TotalMod = TotalMod - x$
$S = S + NC$
FinPara
Fin

El siguiente ejemplo muestra la aplicación del algoritmo de conteo de modelos. Sea una *KB* $K = \{k_1, k_2, k_3, k_4, k_5\}$ = {(1*0*), (*001), (0***), (*1*1), (1011)} expresada mediante patrones falsificantes, con n=4. La Tabla II muestra el cálculo del número de modelos para K. La función Revisar() significa aplicar las operaciones de *Ind(), Sub()* y *Gen()*. Al iniciar el algoritmo se tienen: $TotalMod = 2^4 = 16$. k_1=(1*0*) elimina $2^2 = 4$ modelos.

TABLA II
CONTANDO LOS MODELOS DE K

i=2			*S*			*NC*	*x*	*Total Mod*
k₂	1*0*					0001	*1*	12 - 1 = 11
*001	0001							
i=3			*S*			*NC*	*x*	*Total Mod*
k_3	1*0*	0001				01**	4	
		01**				001*	2	11-7 = 4
0***	*Ind()*	001*				0000	1	
		0000						
i=4			*S*			*NC*	*x*	*Total Mod*
k_4	1*0*	0001	01**	001*	0000	-	0	4-0=4
*1*1	01*1	*Ind()*	*Sub()*					
	1111	*Ind()*	*Ind()*	*Ind()*	*Ind()*	1111	1	4-1=3
i=5			*S*			*NC*	*x*	*Total Mod*
K_5	1*0*	0001	01**	001*	0000	1111		
1011	*Ind()*	*Ind()*	*Ind()*	*Ind()*	*Ind()*	Ind()		
					NC	1011	1	3-1=2

Al terminar de iterar el algoritmo, tenemos como resultado: S = {(1*0*),(0001),(01**),(001*),(0000),(1111),(1011)}, con *TotalMod*=2, por lo que $\#Mod(S) = \#Mod(K) = 2$.

B. Expansión de la Base de Conocimiento

Dado K y φ, nuestro algoritmo revisa si $K \models \varphi$, si $S = \varnothing$ no se agrega ninguna cláusula a la base de conocimiento. En otro caso, se forma un conjunto de cláusulas S necesarias para que $(K \cup S) \models \varphi$. Y en este caso se cumple:

$$\#Mod(K \cup \varphi) = \#Mod(K) - \sum \#Fals(S_i) \quad (2)$$

Si $\#\sum Fals(S_i) < \#Mod(K)$, entonces K aún es consistente y se debe aplicar un proceso de expansión $(K+\varphi)$.

Si $\#\sum Fals(S_i) \geq \#Mod(K)$, entonces K deja de ser consistente y hay que aplicar un proceso de contracción sólo sobre K.

Consideremos el siguiente ejemplo usando patrones falsificantes. Sea originalmente K = {(1*0*0) (*001*) (***10) (***1*) (111*1)}.

Aplicando el algoritmo propuesto de conteo de modelos, tenemos que K es satisfactible con 13 modelos.

Sea $\varphi = \{(\neg p \vee q \vee \neg r)\} = \{(101**)\}$ la nueva información, por lo que es necesario revisar $K \models \varphi$.

La Tabla III representa la aplicación del algoritmo propuesto de revisión de creencias descrito en la Fig. 2. Se generan tres operaciones, las dos primeras son de independencia, mientras que la tercera operación es de generación de dos nuevas cláusulas. Al operar φ con k_3 se generan las cláusulas (1010*) y (10111) que serán comparadas contra las cláusulas k_4 y k_5. El resultado al finalizar k_5 es la cláusula (1010*) la cual se almacena en el conjunto S. Mientras que la última cláusula (10111) que se encuentra almacenada en la pila φ es una cláusula subsumida por k_5. Si el conjunto S contiene más de 1 cláusula se deben comparar entre ellas para obtener el mínimo número de modelos a eliminar de $\#Mod(K)$.

El conjunto resultante S = {(1010*)} significa que es la cláusula necesaria para que φ se pueda inferir de K. Además, esta cláusula (1010*) indica el número de modelos que se deben restar a K. De acuerdo a la formula (1), tenemos que el número de asteriscos de la cláusula es 1, entonces $\#Fals(S) = 2^1 = 2$, este 2 representa el número de asignaciones que se deben restar a los modelos de K como se indica en la formula (2). $\#Mod(K) - \Sigma\#Fals(S_i) = 13\text{-}2 = 11$.

Así la nueva base de conocimiento K después de aplicar la operación $(K*\varphi)$ es K={(1*0*0),(*001*),(***10), (***1*) (111*1), (1010*)} con 11 modelos, por lo que K será satisfactible (consistente).

TABLA III
GENERACIÓN DE NUEVAS CLÁUSULAS

K	1*0*0	*001*	***10	***1*	111*1	S
φ	101**	101**	1010*	1010*	1010*	**1010***
	Ind()	*Ind()*	10111	← *Sub()*		
			Gen()=2			

C. Contracción de la Base de Conocimiento

La operación de revisión $(K*\varphi)$, con K y φ dos FNC's, hace que el número de cláusulas en K se incremente, en tanto que el número de modelos de K decrece. Sin embargo, las operaciones *Ind(), Sub()* y *Gen()* a través de los patrones de 0,1 y * nos permiten conocer el número de modelos que se van perdiendo al agregar nueva información. A través de estas operaciones podemos determinar cuando la base de conocimiento deja de ser consistente, de acuerdo a la fórmula (2), es decir, cuando el número de modelos llega a ser cero a medida que se van agregando nuevas clausulas a K.

El algoritmo podría aplicarse sobre K para formar una base de conocimiento sólo con cláusulas independientes. Para lo que a φ se le irá asignando las cláusulas originales de K de la segunda en adelante. Al mismo tiempo que se forma un conjunto independiente de cláusulas se va contando el número de modelos que hay en K usando las formulas (1) y (2).

Nuestra propuesta permite contar el número de modelos de K que se eliminan al realizar $(K*\varphi)$. Por ejemplo, consideremos el espacio n=4 variables. Sea $K = \{(\neg p \vee r)\} = \{(1*0*)\}$. El número máximo de modelos con 4 variables es 2^4=16, de acuerdo a (1) la cláusula $k_1 \in K$ tiene $\#Mod(K) = 16\text{-}2^2 = 12$ modelos que satisfacen la formula K. Si agregamos una nueva cláusula φ a K, por ejemplo (*001), aplicando el proceso de revisión como se indica en la Tabla IV, se genera una nueva cláusula 0001 lo que indica que se debe eliminar 1 modelo de K de los 12 que se tenían, quedando así 11 modelos.

TABLA IV
REVISANDO LOS MODELOS DE *K*

K	1*0*0	
φ	101**	
	Gen()	0001

Sea la base de conocimiento actualizada K' = {(1*0*),(0001)} al agregar la cláusula φ ={(0***)} se debe aplicar el mismo proceso de revisión sobre K'.

En la Tabla V Se describe el proceso para determinar cuántos modelos se deben quitar a la base de conocimiento *K*, realizando la revisión sobre *K'* para no modificar la base *K*. Como resultado del proceso de revisión se generaron tres nuevas cláusulas que nos permiten determinar el número de modelos que se restarán a *K*. Para (01**) hay #*Fals*((01**)) = 4 asignaciones falsificantes, para (001*) hay 2 asignaciones falsificantes y para (0000) hay 1 asignación falsificante.

TABLA V
REVISANDO LOS MODELOS DE K', GENERANDO TRES NUEVAS CLÁUSULAS

K	1*0*	0001	
φ	0***	0***	
	Ind()	*Gen()=3*	01**
			001*
			0000

En total se deben restar 7 asignaciones a los modelos de *K*. Así en la nueva base de conocimiento *K* habrá 11-7=4 modelos. Siguiendo el mismo proceso, si agregamos las cláusulas (*1*1) y (1011), quedando la base de conocimiento *K* = {(1*0*), (*001) (0***), (*1*1), (1011)} mientras que la base de conocimiento utilizada para contar el número de modelos es *K'*={(1*0*), (0001), (01**), (001*), (0000), (1111), (1011)}.

La base de conocimiento *K* contiene únicamente 2 modelos y al agregar φ = {(1*10)}, *K* deja de ser consistente.

TABLA VI
($K \cup \varphi$) ES INSATISFACTIBLE

K	1*0*	*001	0***	*1*1	1011
φ	1*10	1*10	1*10	1*10	1*10
	Ind()	*Ind()*	*Ind()*	*Ind()*	*Ind()*

La Tabla VI muestra este proceso al agregar φ se eliminan 2 modelos por lo que #*Mod*($K \cup \varphi$) = 0.

Se tendría que ($K \cup \varphi$) no tiene modelos, y por tal ($K \cup \varphi$) es inconsistente, por lo que es necesario buscar la cláusula con el menor número de asignaciones que le fueron restadas a *K*. Este proceso de eliminar información de la base de conocimiento es el denominado proceso de contracción (K'-φ).

En la TablaVII, mostramos las cláusulas en *K* y *K'*. Las cláusulas en *K'* indica el número de modelos que se fue restando a *K* durante el proceso de conteo de modelos y el número de modelos que quedan al intercambiar una cláusula de *K* por la nueva información. Privilegiando la nueva información sobre la mas antigua en la base de conocimiento *K*, las cláusulas más antiguas son k_1 y k_2.

La cláusula k_1 es la candidata a eliminarse de *K* debido a que representa la creencia más antigua y mantiene la consistencia de la base de conocimiento. Si quitamos k_2 como una cláusula con mínimo cambio será necesario ir quitando cláusulas con un número mínimo de asteriscos hasta qué #*Mod*(*K'*) > 0. Otra opción posible es maximizar el número de modelos en *K*, para lo cual eliminaríamos k_3 que es la cláusula con más asteriscos, por lo que resta más modelos a *K*, de esta forma habría garantía de que *K* recupere la satisfactibilidad.

Al eliminar k_1, la base de conocimiento *K* ahora tendrá 2 modelos. Esos 2 modelos corresponden a las dos asignaciones falsificantes de la cláusula k_1.

TABLA VII
K_3 CLÁUSULA CANDIDATA A ELIMINARSE DE K

	Cláusulas en *K*	Cláusulas en *K'*	#Modelos restados a *K*	#modelos en *K* al cambiar k_i por φ
k_1	(1*0*)	(1*0*)	4	2
k_2	(*001)	(0001)	1	0
k_3	(0***)	(01**)(001)(0000)	7	5
k_4	(*1*1)	(1111)	1	1
k_5	(1011)	(1011)	1	1

En nuestra propuesta, usamos las representaciones en FNC tanto de la KB *K* como de la nueva información φ. El proceso de revisión de creencias ($K^*\varphi$) se basa en resolver primero $K \models \varphi$. Una forma práctica de resolver este tipo de problemas difíciles, es realizar si es posible, las tareas más difíciles fuera de línea, mientras las tareas más frecuentes se realizan en línea, y que además puedan tener un costo computacional menor.

Por ejemplo, al transformar *K* en un conjunto lógicamente equivalente *K'* pero formado sólo por cláusulas independientes, permite que el proceso de inferencia $K' \models \varphi_i$, siendo φ_i una sola cláusula, se haga en tiempo O($n * 2^r$), con *r* siendo el número de asteriscos en φ_i. Por lo que entonces $K' \models \varphi$ se realiza en tiempo O($n * |\varphi| * 2^{ri}$), con *ri* siendo el número máximo de asteriscos para alguna $\varphi_i \in \varphi$.

Note que en la estimación de las funciones de complejidad no aparecen factores constantes, dado que los procesos que se aplican, como son: *Sub*(), *Ind*(),*Gen*(), y la comparación de cadenas, todos estos procesos son de orden lineal, sin ser escalados por factores constantes. La velocidad del proceso de revisión se acelera en proporción directa a que la longitud de las cláusulas de la *k*-FNC φ se incremente (dado que el número de asteriscos es $r = n - k$.

Nuestra propuesta requiere como entrada de formas normales conjuntivas, lo que puede limitar su aplicabilidad. Sin embargo, hay algoritmos para convertir fórmulas generales a sus formas normales conjuntivas, aunque esto podría requerir para algunas instancias de un costo exponencial de cómputo sobre la longitud de la fórmula general [12].

Sin embargo, el mayor costo computacional del problema original es trasladado al proceso de reducción de *K* a *K'*. Esta reducción se puede realizar en tiempo O($N * |K| * n$), donde *N* será el número máximo de cláusulas independientes que se generan a partir de una cláusula $C_i \in K$. *N* está acotado superiormente por 2^r, con *r* el número de asteriscos en C_i lo que puede ser un número exponencial con respecto a *n*.

Aplicando la reducción de *K* a *K'* fuera de línea, se tiene la posibilidad de trabajar en linea el proceso de revisión de creencias ($K'*\varphi$) de una forma más práctica que para el caso ($K*\varphi$), y con la ventaja de que al mismo tiempo se realiza el conteo de modelos que permanecerán en ($K' \cup \varphi$), lo que permite determinar la consistencia de la nueva base de conocimiento.

La propuesta del algoritmo de revisión de creencias en base a patrones falsificantes se ha probado con instancias de bases de conocimiento de alrededor de 30 cláusulas con 20 variables. El equipo utilizado es una computadora personal con procesador Intel Core i3 con 4 MB de RAM.

V. Conclusiones

En este trabajo se presenta una propuesta algorítmica que establece de manera práctica un método para determinar la consistencia de una base de conocimiento K cuando se agrega nueva información φ (ambas fórmulas expresadas en FNC), en base a la aplicación de la inferencia proposicional $K \models \varphi$. Esta propuesta representa el proceso completo de revisión de creencias utilizando las operaciones básicas del modelo AGM: Expansión, Revisión y Contracción.

Se utiliza un algoritmo de inferencia proposicional sobre formas normales conjuntivas. Este algoritmo de inferencia recibe como entrada la base de conocimiento K y la nueva información φ y retorna como salida, el conjunto de cláusulas S necesarias para inferir φ, esto es, $(K \cup S) \models \varphi$.

Cada FNC se transforma en patrones falsificantes de 0,1 y * con el fin de facilitar las operaciones lógicas entre cláusulas. El hecho de realizar esta transformación en base a un mismo conjunto de variables nos permite conocer el número máximo de modelos en cada cláusula y, por tanto, cuantos modelos van quedando en una base de conocimiento dinámica.

Nuestra propuesta algorítmica se puede aplicar fuera de línea para construir una base de conocimiento K con sólo cláusulas independientes, lo que permite conocer el número de modelos que hay en K, y así trabajar en línea sólo el proceso de revisión de creencias $(K * \varphi)$. Trabajar con cláusulas independientes da la ventaja de que al mismo tiempo que se ejecuta $(K * \varphi)$ se realiza el conteo de modelos que permanecerán y por tanto, se verifica la consistencia de la nueva base de conocimiento. Nuestra propuesta no ha sido llevada a alguna aplicación o para resolver algún problema práctico, por lo que esta es una de las tareas futuras a realizar.

Referencias

[1] M. Aiguier, J. Atif, I. Bloch, and C., Hudelot, "Belief revision, minimal change and relaxation: a general framework based on satisfaction systems, and applications to description logics", *Artificial Intelligence*, 256, pp. 160–180, 2018.

[2] C. Alchourrón, P. Gärdenfors, and D. Makinson, "On the logic of theory change: Partial meet contraction and revision functions", *Journal of Symbolic Logic*, 50, pp. 510–530, 1985.

[3] Anfrix. De las conductas indecorosas en la mesa de mi señor. Recuperado de: https://www.anfrix.com/2006/01/codex-romanoff-ii-los-modales-en-la-mesa-medieval/, 20 de abril de 2020.

[4] C. Beierle, C. Eichhorn, and G. Kern-Isberner, "On transformations and normal forms of conditional knowledge bases", *Springer International Publishing AG*, 1, pp. 488–494, 2017.

[5] T. Caridroit, S. Konieczny, and P. Marquis, "Contraction in propositional logic", *International Journal of Approximate Reasoning*, 80, pp. 428–442, 2017.

[6] N. Creignou, O. Papini, R. Pichler, and S. Woltran, "Belief Update within Propositional Fragments", *Journal of Artificial Intelligence Research*, 61, pp. 807–834, 2018.

[7] N. Creignou, O. Papini, R. Pichler, and S. Woltran, "Belief revision within fragments of propositional logic", *In Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, KR'12, p.p. 126–136. AAAI Press. ISBN 978-1-57735-560-1. 2014.

[8] E. Cresto, "Revisión de creencias y racionalidad", *Cuadernos CIMBAGE*, 5, pp. 133–156, 2002.

[9] G. De Ita , J.R. Marcial, J.A. Hernández, P. Bello, Lógica proposicional y de predicados, En Conocimiento y razonamiento computacional, Academia Mexicana de Computación, A.C., (2019).

[10] O. Doubois, "Counting the number of solutions for instances of satisfiability", *Theoretical Computer Science*, 81, pp. 49–64, 1991.

[11] E. Fermé, "Revisión de creencias", *Revista Iberoamericana de Inteligencia Artificial*, 11(34), pp. 17–39, 2007.

[12] J. Gallier, *Logic for Computer Science: Foundations of Automatic Theorem Proving*, Dover Publicatios, Philadelphia, USA, second edition, pp. 28-141. 2015.

[13] P. Gärdenfors, *Belief revisión: An introduction*, Great Britain: Cambridge University Press, Ed. Cambridge, 29, pp. 1-28, 1992.

[14] H. Katsuno, and A. O. Mendelzon, "Propositional knowledge base revision and minimal change", *Artificial Intelligence*, 52(3), pp. 263–294, 1991.

[15] R. Khardon, and D. Roth, "Reasoning with models", *Artificial Intelligence*, 87, pp. 187–213, 1996.

[16] P. Liberatore, "Revision by history", *Journal Artif. Intell*. Res., 52:287–329, doi:10.1613/jair.4608. URL"https://doi.org/10.1613/jair.4608", 2015.

[17] P. Liberatore, M. Schaerf, "The complexity of model checking for Belief Revision and Update", In AAAI-96 Proceedings, pp. 556 – 561, 1996.

[18] S. Luan, G. Dai, and L. Magnani, "An approximate approach to belief revision", *Logic Journal of the IGPL*, 2, 2012.

[19] C. Pons, R. Rosenfeld, and C. Smith. *Lógica para Informática*. Editorial de la Universidad Nacional de La Plata (EDULP), 2017.

[20] H. Velázquez, "Lógica deóntica: breve panorama de la cuestión", Cuadrante Phi *Revista de estudiantes de filosofía* 28, pp. 1-24, 2015.

[21] MA. Williams, "Applications of belief revision", In Freitag B., Decker H., Kifer M., Voronkov (eds), Transactions and Change in Logic Databases, Lecture Notes in Comp. Sc. vol 1472, Springer, 1997, https://doi.org/10.1007/BFb0055503

[22] F. Zacarias and G. De Ita, "A Model-Based Algorithm for Propositional Belief Revision", *Latin America Transactions, IEEE*. 13, pp. 1055-1060, 10.1109/TLA.2015.7106357, 2015.

[23] H. R. Zuleta, "Lógica deóntica y verdad", *Análisis filosófico XXVI,* 1, pp. 115-133, 2006.

Pedro Bello López estudiante de doctorado de Ingeniería del Lenguaje y del Conocimiento en la Benemérita Universidad Autónoma de Puebla. Maestro en Ciencias de la Computación, docente en la Facultad de Ciencias de la Computación en la Benemérita Universidad Autónoma de Puebla. México. Áreas de investigación: diseño de algoritmos combinatorios, problemas NP, estructuras de datos, razonamiento automático e inteligencia artificial.

Guillermo De Ita Luna obtuvo su doctorado en el CINVESTAV del IPN, México. Ha trabajado por 10 años como desarrollador y consultor en sistemas de bases de datos y sistemas de información geográfica en diferentes empresas. Ha realizado estancias de investigación en la Universidad de Chicago, Texas A&M, INAOEP, y el instituto INRIA en Lille Francia. Profesor investigador por 28 años en la Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla, Puebla, México. Su área de interés es la lógica y la teoría de grafos.