



Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Computación

Sistema de Revisión de Creencias usando Inferencia Proposicional

Tesis que para obtener el título de Ingeniero en Ciencias de la
Computación

Presenta

Guillermo Daniel Guridi Bandala

Asesor

Dr. Pedro Bello López

Octubre 2022

Agradecimientos

1. A mi tutor

Primero quiero agradecer al Dr. Pedro Bello López. Sin usted y sus virtudes, su paciencia y constancia este trabajo no lo hubiese logrado tan fácil. Sus consejos fueron siempre útiles. Usted formó parte importante de esta historia con sus aportes profesionales que lo caracterizan. Muchas gracias por sus múltiples palabras de aliento, cuando más las necesite; por estar allí cuando mis horas de trabajo se hacían confusas. Gracias por su orientación.

2. A mis padres y hermana

Ustedes han sido siempre el motor que impulsa mis sueños y esperanzas, quienes estuvieron siempre a mi lado en los días y noches más difíciles durante mis horas de estudio. Siempre han sido mis mejores guías de vida. Hoy cuando concluyo mis estudios, les dedico a ustedes Tere, Ceci y Jaime; este logro como una meta más conquistada. Orgulloso de haberlos elegido como mi familia y que estén a mi lado en este momento tan importante. Gracias por ser quienes son y por creer en mí.

3. A mis amigos

Mis amigos y compañeros de viaje hoy culminan esta maravillosa aventura y no puedo dejar de recordar cuantas tardes y horas de trabajo nos juntamos a lo largo de nuestra formación. Hoy nos toca cerrar un capítulo maravilloso en esta historia de vida y no puedo dejar de agradecerles por su apoyo y constancia, al estar en las horas más difíciles, por compartir horas de estudio. Gracias Jonathan, Marco y Adrián por estar siempre allí.

4. A la institución

A la Vicerrectoría de Docencia de la Benemérita Universidad Autónoma de Puebla por el apoyo para la culminación de este trabajo de tesis dentro de los proyectos investigación 2022 del Programa Institucional para la Consolidación de los Cuerpos Académicos, en especial al cuerpo académico Algoritmos Combinatorios y Aprendizaje.

Contenido

Resumen.....	4
Capítulo 1. Introducción.....	6
Capítulo 2. Marco teórico.....	11
2.1. Revisión de creencias.....	11
2.1.2. Los problemas metodológicos de las revisiones de creencias	15
2.1.3 Conjunto de creencias	16
2.2. Modelos AGM y KM.....	17
2.3. Lógica proposicional	18
2.4. Formas Normales	21
Capítulo 3. Algoritmo de inferencia proposicional	26
3.1. Definiciones básicas	26
3.2. Operaciones básicas	28
3.3. Algoritmo de inferencia proposicional.....	34
3.4. Pruebas.....	36
Capítulo 4. Sistema de revisión de creencias.....	39
4.1. Vista general del sistema	39
4.2. Inicio del sistema.....	40
4.3. Proceso de Inferencia.....	41
Conclusiones	46
Bibliografía	47

Resumen

El problema aquí desarrollado modela el problema de revisión de creencias a través del desarrollo de un sistema en web para determinar si una nueva información se puede inferir de una base de conocimiento ya establecida y en caso contrario generar el conjunto de información que es necesaria para que se pueda generar la inferencia proposicional. Tanto la nueva información como la información almacenada en la base de conocimiento es modelada usando lógica proposicional a través de la Forma Normal Conjuntiva. El sistema se desarrolló como una aplicación web, instalando el servidor apache como servidor web y utilizando el lenguaje de programación PHP con HTML, CSS y un visualizador grafico con JavaScript.

El trabajo de tesis está organizado de la siguiente forma:

- En el primer capítulo se realiza la introducción del trabajo de tesis donde se expone el problema de revisión de creencias, la propuesta de solución, así como los principales investigadores que han dado propuestas de modelos para el problema de revisión de creencias.
- En el segundo capítulo se describe el marco teórico necesario para el desarrollo del trabajo de tesis, se presentan los modelos AGM y KM como los principales aportes al problema de revisión de creencias y se presenta la teoría de lógica proposicional junto con las formas normales.
- En el tercer capítulo se presenta el algoritmo propuesto para el problema de revisión de creencias usando inferencia proposicional, además, se describe con diversas pruebas de escritorio el funcionamiento del algoritmo.

- En el capítulo 4 se describe el desarrollo del sistema en web a través de pantallas de ejecución de diversas instancias de bases de conocimiento y de la llegada de nueva información con el fin de generar la información necesaria para realizar la inferencia proposicional.
- Finalmente se presentan las conclusiones del trabajo de tesis, los aportes y las posibles mejoras al sistema de inferencia proposicional.

Capítulo 1. Introducción

En el área de la Inteligencia Artificial se estudian diversos problemas que tratan de modelar el razonamiento humano, donde lo primero es determinar un lenguaje para representar el conocimiento de un agente y definir reglas que en base al conocimiento provisto se puedan derivar nuevas conclusiones, es decir, a partir de una base de conocimiento (KB, por sus siglas en inglés “Knowledge Base”) fija obtener nuevas conclusiones. La mayoría de estos sistemas modelan el conocimiento utilizando lenguajes de primer orden y realizan una extensión de forma tal que se puedan probar en base a un conjunto de postulados. Además, la dinámica del conocimiento y la revisión de creencias permiten caracterizar la forma en que un agente racional actualiza su estado epistémico al recibir nueva información.

La revisión de creencias es una forma de representar el conocimiento (en una base de conocimiento) que un ser racional adquiere a través del tiempo, del estudio, por experiencia o simplemente de sus vivencias; cuando llega una nueva información la base de conocimiento se tiene que actualizar y si entra en contradicción con lo que ya sabemos será necesario quitar alguna creencia anterior.

Las reglas de inferencia son construcciones de un sistema que permite determinar información nueva a partir de la información ya existente. Por ejemplo, cuando en el 2010 la Real Academia Española¹ decidió excluir las letras ch y ll del abecedario español por ser dígrafos, esto es, conjunto de dos letras o grafemas que representan un solo fonema; de esta forma el abecedario quedo reducido de 29 a 27 letras.

Por otra parte, nuestra base de conocimiento puede estar formada además de un conjunto de reglas o normas, incluso leyes que todo ser humano debe

¹ http://enciclopedia.us.es/index.php/Abecedario_espa%C3%B1ol

seguir o cumplir, y es en este sentido donde la inferencia lógica contribuye a determinar si la nueva información ya está presente en nuestra base de conocimiento o no. Así González R.C (2017) considera que el razonamiento humano opera en base a reglas lógicas de inferencias [3].

El modelo básico para determinar si la nueva información ya está presente en nuestra base de conocimiento o determinar las cláusulas necesarias para lograr inferencia, se muestra en la Figura 1.1.

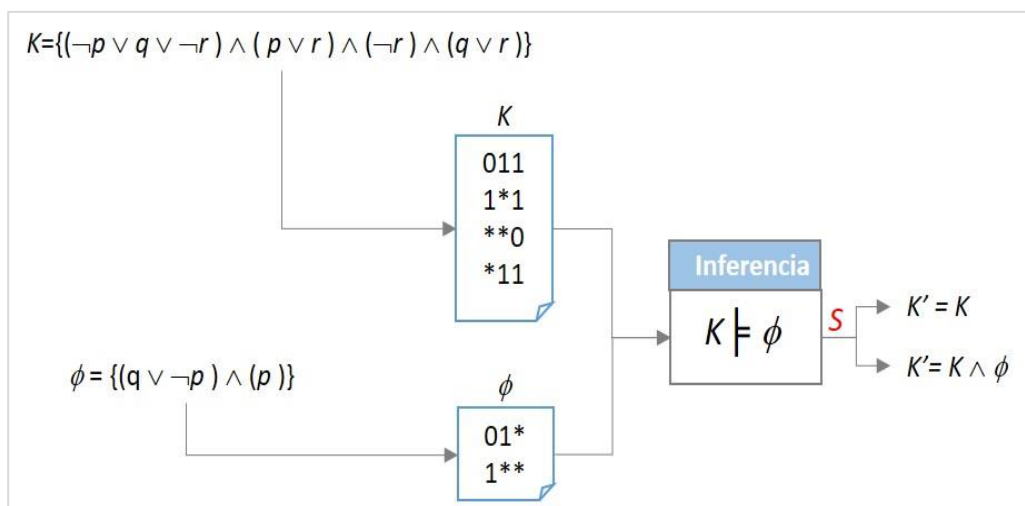


Figura 1.1. Modelo base del sistema de inferencia proposicional.

Abordamos el problema de revisión de creencias desde el punto de vista de la inferencia proposicional, dada una KB podremos determinar si un nuevo conjunto de cláusulas en forma normal conjuntiva puede ser inferido de la KB, y para el caso de que no sea inferida, podremos incrementar el nuevo conocimiento a la KB que sea necesario para se cumpla la inferencia. La inferencia lógica significa deducir algo, sacar una conclusión o conducir un nuevo resultado a partir de algo conocido mediante un proceso mental que crea una implicación lógica.

Uno de los paradigmas más conocidos de revisión de creencias es el modelo AGM [1] (Las iniciales se deben a sus autores: Alchourrón, Gärdenfors, y

Makinson). Posteriormente Mendelzon y Katsuno [2] unificaron los diferentes enfoques semánticos de revisión de creencias y reformularon los postulados de AGM e idearon el modelo KM (También las iniciales del modelo se deben a sus autores).

La manera más utilizada para modelar el problema de revisión de creencias es a través de lógica proposicional. La lógica proposicional permite analizar un problema que no solo queremos describirlo en términos de afirmaciones y conectivas lógicas. También queremos obtener nueva información, dicha información puede incluir el determinar si una información es una conclusión válida a partir de los datos proporcionados o saber qué información podemos deducir a partir de las premisas que nos proporcionaron.

Un objetivo fundamental de los sistemas basados en el razonamiento es la automatización del proceso de razonamiento deductivo a partir de la KB. El razonamiento deductivo proposicional se define de la siguiente manera: dada una base de conocimiento K , que contiene el conocimiento acerca de un dominio y una sentencia. Dicha sentencia representa la consulta que captura la situación actual, ambas expresadas en lógica proposicional también llamada lógica simbólica, el objetivo es decidir si K implica ϕ (denotado como $K \models \phi$) o de igual forma podemos decir que ϕ se infiere de K . $K \models \phi$ se conoce como el problema de implicación lógica que también llamamos inferencia proposicional.

En este proyecto de tesis se plantea un sistema en web que permite determinar si la nueva información denotada con ϕ se puede implicar lógicamente de K y en caso contrario si ϕ no se implica de K , entonces se generan las cláusulas necesarias para que se cumpla $K \models \phi$.

El sistema propuesto es desarrollado en PHP utilizando la librería Vis.js con el fin de generar una representación gráfica del conocimiento generado.

El enfoque más conocido para realizar la revisión de creencias es el paradigma AGM [1]. La teoría AGM propone un conjunto de postulados

racionales, que cualquier operador de revisión debe satisfacer. Posterior a la propuesta AGM, Alchourrón y Makinson [4] desarrollaron un modelo constructivo para funciones de cambio llamado “contracción segura” (safe contraction) que después fue generalizada por Hansson [5]. Sin embargo, la mayoría de estas propuestas requieren de información adicional, tales como: relaciones de afianzamiento epistémicas, sistemas de esferas, relación de subfórmulas, entre otras creencias son completamente dependientes de la sintaxis [4].

Katsuno y Mendelzon unificaron los diferentes enfoques semánticos de revisión de creencias, y reformularon los postulados AGM, a los que llamaron ahora postulados KM. Además, propusieron un teorema de representación que caracteriza las operaciones de revisión en términos de pre-órdenes totales sobre el conjunto de interpretaciones [2].

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

Algunas de las investigaciones en esta dirección se han acotado principalmente a considerar el fragmento de cláusulas de Horn dentro de la lógica clásica. Existen diversas propuestas que involucran fórmulas proposicionales tales como las descritas en [8, 9, 1] que sugieren métodos que abordan sólo fragmentos de la lógica proposicional. Una de las propuestas recientes debida a Delgrande presenta los primeros resultados sobre cambio de creencias en el fragmento Horn [9].

En [4], los autores presentan una metodología general para definir nuevos operadores de revisión derivados de operadores estándar (como, por

ejemplo, los operadores de Dalal y Satoh), tal que el resultado de la revisión se mantiene en el fragmento en cuestión. Por lo tanto, en esta propuesta los autores no se limitan solo al caso Horn, sino que ésta es aplicable a otros fragmentos de la lógica proposicional, donde los modelos de las fórmulas cumplan el ser cerrados bajo una función Booleana [7].

Se puede notar que estas propuestas se desarrollan para considerar algún tipo de fórmula normalizada o bien sólo consideran cláusulas de Horn. Cada una de estas propuestas propone un operador de revisión que trabaja sobre sus fórmulas normalizadas y que presentan diversos inconvenientes. Por ejemplo, algunos de los operadores son dependientes de la sintaxis o bien requieren de información adicional. Otros definen diferentes nociones de proximidad, y unos más se limitan a revisar sólo bases consistentes.

Así en general, el problema de la implicación lógica es un reto para el razonamiento automático, ya que se conoce que es un problema en la clase Co-NP completo, incluso en el caso proposicional, como lo muestran Khardon, R. & Roth, D. (1996) y Liberatore, P. (1997).

A continuación, se presentan los conceptos teóricos necesarios para el desarrollo del trabajo de tesis.

Capítulo 2. Marco teórico

Para el desarrollo del trabajo fue necesario revisar conceptos teóricos que sirven como base para plantear el algoritmo de revisión de creencias que posteriormente se implementación en una aplicación tipo web.

2.1. Revisión de creencias

La revisión de creencias es una forma de representar el conocimiento (en una base de conocimiento) que un ser racional adquiere a través del tiempo, del estudio, por experiencia o simplemente de sus vivencias; cuando llega una nueva información la base de conocimiento se tiene que actualizar.

La revisión de creencias consiste en la dinámica de cómo deben cambiar los estados epistémicos de un agente como resultado de recibir nueva información para almacenarse en su base de conocimiento, el caso más simple (expansión) surge por el aprendizaje de algo nuevo, pero en ocasiones la nueva información contradice creencias previamente aceptadas y entonces es necesario eliminar (contracción) viejas creencias.

Las bases de conocimiento tienen un amplio rango de uso, entre los que destacan: el desarrollo de técnicas para la manipulación inteligente de estructuras de datos con objeto de obtener e inferir conocimiento; en las bases de conocimiento incorporadas a los programas de inteligencia artificial; en los sistemas expertos que disponen de una base de conocimiento que con representaciones de hechos y esquemas de razonamiento; así como en los métodos de razonamiento y representación del conocimiento en la teoría de revisión de creencias dentro de la inteligencia artificial.

En el campo de la recuperación de información y el procesamiento del lenguaje natural, la respuesta a preguntas de la base de conocimiento es

un tema de investigación importante, las preguntas más comunes que se hacen en la Web son preguntas simples, que pueden responderse con un solo hecho relacional en una base de conocimiento (KB). Sin embargo, responder preguntas simples automáticamente sigue siendo una tarea desafiante (Zhou et al, 2021). En (Zhang et al, 2021) se propone un nuevo modelo de respuesta a preguntas de la base de conocimiento de un extremo a otro basado en la Red Neuronal Bayesiana para estimar las incertidumbres surgidas tanto del modelo como de los datos. En (Maisto et al, 2021) se presenta una metodología híbrida semiautomática para la construcción de una Base de Conocimiento Léxica, con el propósito de enfrentar los desafíos relacionados con la sinonimia en los sistemas de Pregunta-Respuesta. De esta forma la base de conocimiento resultante permite el reconocimiento automático de los sustantivos y adjetivos que no suelen estar relacionados en bases de datos de sinónimos.

En el contexto del rápido desarrollo de las tecnologías digitales como lo indica Kurganova (2021), los sistemas que forman la base de conocimiento se convierten en parte integral de la infraestructura de las organizaciones, ya que aumentan la eficiencia de las empresas industriales y comerciales. En la industria de la construcción por ejemplo con ayuda de las tecnologías digitales en (Le et al, 2021), se presenta una mejora a la práctica actual al proponer un enfoque basado en datos que aprovecha los datos del informe de trabajo diario fácilmente disponible de proyectos anteriores para desarrollar una base de conocimiento de patrones de secuencia de construcción.

En el área de la industria de los robots, por ejemplo, En (Budiharto et al, 2020) se desarrolló de un robot humanoide inteligente con capacidad de autoaprendizaje para aceptar y dar respuestas a personas basadas en la base de conocimiento de Deep Learning y Big Data, este tipo de robot se puede utilizar ampliamente en hoteles, universidades y servicios públicos. En (Peng et al, 2020) se describe el uso de una base de conocimiento en la automatización e inteligencia del proceso de generación creativa y mejorar

el efecto de generación creativa y la usabilidad de los movimientos de danza en un robot bailarín. En el marco de planificación de procesos inteligentes para máquinas, en (Ye et al, 2020) se propone un controlador con soporte de base de conocimiento en la nube con la capacidad de realizar la planificación de procesos de forma autónoma según el diseño de la pieza de trabajo.

En las ciencias de la computación, un lenguaje ampliamente utilizado para representar el problema de revisión de creencias es la lógica proposicional. El lenguaje de la lógica proposicional permite modelar tanto las situaciones, los problemas y los algoritmos que los resuelven, y de tal forma, que se pueda razonar de manera formal sobre los modelos propuestos. Razonar sobre las situaciones significa la construcción de argumentos acerca de éstos, de forma que los argumentos sean válidos y puedan ser defendidos de forma rigurosa, o ejecutados de forma automática en una computadora.

Además de los sistemas expertos, un campo donde es ampliamente usado las bases de conocimiento es la teoría de revisión de creencias, en Haret et al (2019) se indica que las axiomatizaciones clásicas de la revisión de creencias incluyen un postulado que establece que, si la nueva información es consistente con las creencias iniciales, entonces la revisión equivale simplemente a agregar la nueva información a la base de conocimiento original. Deagustini et al (2019) presenta el estudio de la dinámica del conocimiento que se ha abordado en el área de revisión de creencias, campo emergente como la convergencia de los esfuerzos en Filosofía, Lógica y más recientemente Informática, donde los esfuerzos de investigación suelen involucrar bases de conocimiento “planas” donde no hay información adicional sobre las fórmulas almacenadas en él.

2.1.1 Un Ejemplo

Supongamos que tiene una base de conocimiento que contiene, entre otras cosas, las siguientes piezas de información (En algún tipo de código):

α : Todos los Cisnes europeos son blancos.

β : El pájaro atrapado en la trampa es un cisne.

γ : El ave atrapada en la trampa proviene de Suecia.

δ : Suecia es una parte de Europa.

Si su base está acoplada con un programa que puede calcular inferencias lógicas en el código dado, el siguiente hecho es derivable de $\alpha - \delta$:

ε : El pájaro atrapado en la trampa es blanco.

Ahora supongamos que, de hecho, el ave atrapada en la trampa resulta ser negra. Esto significa que desea agregar el hecho $\neg\varepsilon$, es decir, la negación de ε , a la base de datos. Pero entonces la base de datos se vuelve inconsistente. Si desea mantener la base de datos consistente, que normalmente es una metodología sólida, debe revisarla. Esto significa que algunas de las creencias en la base de datos original deben ser retractadas. No querrá renunciar a todas las creencias, ya que esto sería una pérdida innecesaria de información valiosa. Así que tienes que elegir entre retraer α , β , γ o δ .

El problema de la revisión de creencias es que las consideraciones lógicas por sí solas no te dicen a qué creencias renunciar, pero esto tiene que decidirse por otros medios. Lo que hace que las cosas sean más complicadas que las creencias en una base de datos tienen consecuencias lógicas, por lo que al renunciar a una creencia hay que decidir también cuál de las consecuencias retener y cuáles retractarse. Por ejemplo, si decide retractarse α en la situación aquí descrita, α tiene como consecuencias lógicas, entre otras, las dos siguientes:

α' : Todos los cisnes europeos excepto el atrapado en la trampa son blancos
y

α ": Todos los cisnes europeos excepto algunos de los suecos son blancos.
¿Desea mantener alguna de estas frases en la base de conocimiento revisada?

2.1.2. Los problemas metodológicos de las revisiones de creencias

Al tratar de manejar las revisiones de creencias en un entorno computacional, hay tres preguntas metodológicas principales para resolver:

¿Cómo se representan las creencias en la base de datos?

La mayoría de las bases de datos trabajan con los elementos como hechos y reglas como formas primitivas de representar información de representar información. El código utilizado para representar las creencias puede estar más o menos estrechamente relacionado con el formalismo lógico estándar. Un mecanismo para la revisión de creencias es sensible al formalismo elegido para representar las creencias.

¿Cuál es la relación entre los elementos representados explícitamente en la base de datos y las creencias que pueden derivarse en estos elementos?

Esta relación depende en gran medida del área de aplicación en la base de datos. En algunos casos, los elementos formulados explícitamente en la base de datos tienen un estatus especial en comparación con las consecuencias lógicas de estas creencias que pueden derivarse de algún mecanismo de inferencia. En otros casos, la formulación de las creencias en la base de datos inmaterial, de modo que cualquier representación que tenga las mismas consecuencias lógicas, es decir, el mismo conjunto de creencias implícitas, es equivalente.

¿Cómo se toman las decisiones sobre qué retractarse?

La lógica por sí sola no es suficiente para decidir entre qué creencias renunciar y cuáles retener al realizar una revisión de creencias. ¿Cuáles son

los factores extra lógicos que determinan las elecciones? Una idea es que la información perdida al renunciar a las creencias debe mantenerse mínima. Otra idea es que algunas creencias se consideran más importantes. Dentro de la informática, el uso de restricciones integrales es una forma común de manejar el problema. Una vez más, las reglas metodológicas elegidas aquí dependen del área de aplicación.

2.1.3 Conjunto de creencias

La forma más sencilla de modelar un estado de creencias es representarlo mediante un conjunto de oraciones de L . En consecuencia, definimos un conjunto de creencias como un conjunto de K de oraciones en L que satisface la siguiente restricción de integridad.

Si K implica lógicamente ψ , entonces $\psi \in K$.

En el lenguaje lógico, (I) dice que K se cierra bajo consecuencias lógicas. La interpretación de tal conjunto es que contiene todas las oraciones que se aceptan en el estado de creencia modelado. En consecuencia, cuándo $\varphi \in K$ decimos que φ es aceptado en K y cuándo $\neg\varphi \in K$ decimos que φ es rechazado en K . Cabe señalar que la aceptación de una frase no implica que tengan ningún tipo de justificación o apoyo. \emptyset El conjunto de creencias también puede ser visto como una teoría que es una descripción parcial del mundo. “Parcial porque en general hay oraciones φ tales que ni φ ni $\neg\varphi$ están en K . Según la lógica clásica, siempre que K sea inconsistente, entonces $K \models \varphi$ para cada oración φ de la lengua L . Esto significa que hay exactamente una creencia inconsistente establecida bajo nuestra definición, a saber, el conjunto de todas las oraciones L . Introducimos la notación KL para este conjunto de creencias.

2.2. Modelos AGM y KM

El modelo AGM está formado por tres operaciones básicas como se indican a continuación:

1. **Expansión:** La expansión de creencias se deriva de:” aprender algo”. La expansión de K por φ se denota $K+\varphi$, se cumplen además los siguientes 6 postulados:

(E1) Si K es un CC (Conjunto de Creencias), $K + \varphi$ es un CC

(E2) $\varphi \in K + \varphi$

(E3) $K \subseteq K + \varphi$

(E4) Si $\varphi \subset K$, entonces $K + \varphi = K$

(E5) Si $K1 \subseteq K2$, entonces $K1 + \varphi \subseteq K2 + \varphi$

(E6) Para todo K y φ , $K + \varphi$ es el menor CC que satisface (E1) – (E5)

2. **Revisión:** Una operación de revisión consiste según [11] en la modificación del estado epistémico. El estado epistémico es representado como un conjunto de creencias, donde una nueva creencia (la entrada epistémica) se incorpora al conjunto previo conservando consistencia lógica. $K * \varphi$ denota la revisión de K por φ . Esta operación debe cumplir los siguientes 8 postulados:

(R1) Si K es un CC, $K * \varphi$ es un CC

(R2) $\varphi \in K * \varphi$

(R3) $K * \varphi \subseteq K + \varphi$

(R4) Si $\neg\varphi \notin K$, entonces $K * \varphi = K + \varphi$

(R5) $K * \varphi = K\perp$ ssi $\neg\varphi \in \text{Cn}(\emptyset)$

(R6) Si $\varphi \leftrightarrow \psi \in \text{Cn}(\emptyset)$, entonces $K * \varphi = K * \psi$

(R7) $K * (\varphi \wedge \psi) \subseteq (K * \varphi) + \psi$

(R8) Si $\neg\psi \notin K * \varphi$, entonces $(K * \varphi) + \psi \subseteq K * (\varphi \wedge \psi)$

3. Contracción: Una contracción ocurre cuando una sentencia es refutada, pero ninguna sentencia es sumada. La contracción de K por una sentencia φ se denota $K - \varphi$, esta operación debe cumplir los siguientes 8 postulados:

(C1) Si K es un CC, $K - \varphi$ es un CC

(C2) $K - \varphi \subseteq K$

(C3) Si $\varphi \notin K$, entonces $K - \varphi = K$

(C4) Si $\varphi \notin \text{Cn}(\emptyset)$, entonces $\varphi \notin K - \varphi$

(C5) $K \subseteq (K - \varphi) + \varphi$

(C6) Si $\varphi \leftrightarrow \psi \in \text{Cn}(\emptyset)$, entonces $K - \varphi = K - \psi$

(C7) $K - \varphi \cap K - \psi \subseteq K - (\varphi \wedge \psi)$

(C8) Si $\varphi \notin K - (\varphi \wedge \psi)$, entonces $K - (\varphi \wedge \psi) \subseteq K - \varphi$

2.3. Lógica proposicional

Una de las mayores dificultades asociadas a las bases de conocimiento son los aspectos del conocimiento impreciso o incompleto, por lo que modelamos la base de conocimiento utilizando lógica proposicional mediante una fórmula en forma normal conjuntiva (FNC). De esta forma trasladamos el problema de revisión de la base de conocimiento al problema básico de satisfactibilidad.

El problema de satisfactibilidad (SAT) es considerado el primer problema NP- completo demostrado por Stephen Cook en el año de 1971. El problema SAT es central en la teoría de la computación. En la práctica, SAT es fundamental en la resolución de problemas de razonamiento automático, en el diseño y fabricación asistido por computadora, en el desarrollo de base de datos, en el diseño de circuitos integrados, en la integración de módulos para lenguajes de alto nivel, etc. El problema SAT en general es un problema de decisión, lo definimos como: dada una formula booleana F en Forma Normal Conjuntiva (FNC), formada de un conjunto de cláusulas con n variables, se debe encontrar una asignación a dichas variables tal que la F sea satisfactible, es decir que exista al menos una asignación de valores que haga verdadera la formula F .

Los problemas NP-completo son los problemas que caracterizan la clase NP de tal forma que al encontrar una solución del problema SAT se pueden resolver los demás problemas de la clase ya que se puede hacer una transformación en tiempo polinomial al problema correspondiente. De esta forma el problema SAT consiste en decidir si F es satisfactible. Mientras que el problema de optimización MaxSAT consiste en determinar el número máximo de cláusulas que se pueden satisfacer simultáneamente. Y el problema de conteo #SAT consiste en contar el número de asignaciones que satisfacen a F .

Un tema central en el desarrollo de este proyecto de tesis es la lógica proposicional. La lógica proposicional como se indica en [22] se utiliza para analizar razonamientos formalmente válidos partiendo de proposiciones y operadores lógicos, con el objetivo de construir fórmulas formadas por operaciones lógicas entre variables proposicionales.

La lógica proposicional se basa en proposiciones, donde una proposición es una oración que arma o niega algo y por lo tanto puede ser verdadera o falsa. La lógica proposicional es un sistema lógico que busca formalizar la noción de la proposición como un conjunto de proposiciones [23].

La lógica proposicional, también conocida como lógica de enunciados, es un sistema formal cuyos elementos representan proposiciones o enunciados [22]. La revisión de creencias ha sido ampliamente estudiada en el marco de la lógica proposicional, pero recientemente la revisión dentro de fragmentos de la lógica proposicional ha ganado atención [24].

Uno de los objetivos de la lógica proposicional o de predicados en las ciencias de la computación es el desarrollo de lenguajes para modelar tanto las situaciones, los problemas y los algoritmos que los resuelven, y de tal forma, que se pueda razonar de manera formal sobre los modelos propuestos. Razonar sobre las situaciones significa la construcción de argumentos acerca de estos, de forma que los argumentos sean válidos y puedan ser defendidos de forma rigurosa, o ejecutados de forma automática en una computadora. La mayoría de las veces, cuando se usa una lógica para el diseño, especificación y verificación de sistemas computacionales se plantea la relación: $K \models \phi$, que significa determinar si la fórmula ϕ se implica lógicamente de K , donde K es la situación o el conocimiento actual sobre una situación, y ϕ es una fórmula lógica que codifica la información que deseamos saber si es verdadera bajo la suposición K , o que ϕ es satisfactible en K .

En lógica proposicional, el vocabulario es un conjunto de símbolos de letras proposicionales P cuyos elementos normalmente se escriben como p, q, r , etc. Una fórmula de la lógica proposicional sobre P se define como:

- Todo símbolo proposicional de P es una fórmula.
- Si F y G son fórmulas, entonces $(F \wedge G)$ y $(F \vee G)$ son fórmulas.
- Si F es una fórmula, entonces $\neg F$ es una fórmula.

Una Interpretación (también llamada asignación) s sobre el vocabulario P es una función $s: P \rightarrow \{0,1\}$, es decir, s es una función que, para cada símbolo proposicional, nos dice si es verdadero o falso.

Sean s una interpretación y F una fórmula, ambas sobre el vocabulario P . La evaluación en s de F , denotada como $vs(F)$, es una función que por cada

fórmula da un valor de verdadero o falso. Si $vs(F) = \text{verdadero}$, se dice que s satisface a F , o que s es un modelo de F , es decir, una interpretación s es un modelo para una fórmula F si $vs(F) = \text{verdadero}$.

- Tautología: Para toda interpretación s ; $vs(F) = \text{Verdadero}$.
- Contradicción: Para toda interpretación s ; $vs(F) = \text{Falso}$.
- Contingencia: Se da cuando una fórmula F no es Tautología ni Contradicción.
- Satisfactible: Existe una interpretación (o un modelo) s ; $vs(F) = \text{Verdadero}$.
- Insatisfactible: No existe una interpretación (el número de modelos=0) s ; $vs(F) = \text{Verdadero}$.

En la lógica proposicional a los símbolos: \vee , \wedge , \neg , se les llama conectivos lógicos. A menudo también se utilizan los conectivos: \rightarrow , \leftrightarrow , los cuales pueden ser expresados como:

$$F \rightarrow G \approx (\neg F \vee G)$$

$$F \leftrightarrow G \approx ((\neg F \vee G) \wedge (\neg G \vee F))$$

2.4. Formas Normales

Dada una fórmula, a cada posible combinación de los valores de verdad de las fórmulas atómicas que figuran en ella se le llama una asignación. Se dice que una fórmula es una tautología si y solo si es verdadera para cada posible asignación, una fórmula es una contradicción si y solamente si es falsa para cada posible asignación, una fórmula es satisfactible si y solamente si es verdadera para alguna asignación, una fórmula es una contradicción si y solamente si es falsa para toda asignación, una fórmula es una contingencia si y solamente si es verdadera para alguna asignación y falsa para otra.

FNC (Forma Normal Conjuntiva)

Una formula está en forma normal conjuntiva (CNF, del inglés Conjunctive Normal Form) si es una conjunción de disyunciones de literales, es decir, si es de la forma:

$(l_{1,1} \vee l_{1,k1}) \wedge \dots \wedge (l_{n,1} \vee l_{n,kn})$ donde $l_{i,j}$ es un literal.

Ejemplo:

$(\neg p \vee q) \wedge (\neg q \vee p)$ está en FNC

$(\neg p \vee q) \wedge (q \rightarrow p)$ no está en FNC

FND (Forma Normal Disyuntiva)

Una formula está en forma normal disyuntiva (DNF, del inglés Disjunctive Normal Form) si es una disyunción de conjunciones de literales, es decir, si es de la forma:

$(l_{1,1} \wedge l_{1,k1}) \vee \dots \vee (l_{n,1} \wedge l_{n,kn})$ donde $l_{i,j}$ es un literal.

Ejemplo:

$(\neg p \wedge q) \vee (\neg q \wedge p)$ está en FND

$(\neg p \wedge q) \vee (q \rightarrow p)$ no está en FND

Reglas para transformar a FNC

Algoritmo: Aplicando a una formula F los siguientes pasos se obtiene una forma normal conjuntiva de F:

1. Eliminar los bicondicionales usando la equivalencia

$$A \leftrightarrow B \equiv (A \rightarrow B) \wedge (B \rightarrow A) \quad (1)$$

2. Eliminar los condicionales usando la equivalencia

$$A \rightarrow B \equiv \neg A \vee B \quad (2)$$

3. Interiorizar las negaciones usando las equivalencias

$$\neg(A \wedge B) \equiv \neg A \vee \neg B \quad (3)$$

$$\neg(A \vee B) \equiv \neg A \wedge \neg B \quad (4)$$

$$\neg\neg A \equiv A \quad (5)$$

4. Interiorizar las disyunciones usando las equivalencias

$$A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C) \quad (6)$$

$$(A \wedge B) \vee C \equiv (A \vee C) \wedge (B \vee C) \quad (7)$$

Ejemplo de transformación a FNC

Ejemplo de cálculo de una FNC de $\neg(p \wedge (q \rightarrow r))$:

$$\begin{aligned} & \neg(p \wedge (q \rightarrow r)) \\ \equiv & \neg(p \wedge (\neg q \vee r)) && [por (2)] \\ \equiv & \neg p \vee \neg(\neg q \vee r) && [por (3)] \\ \equiv & \neg p \vee (\neg\neg q \wedge \neg r) && [por (4)] \\ \equiv & \neg p \vee (q \wedge \neg r) && [por (5)] \\ \equiv & (\neg p \vee q) \wedge (\neg p \vee \neg r) && [por (6)] \end{aligned}$$

Reglas para transformar a FND

Algoritmo: Aplicando a una formula F los siguientes pasos se obtiene una forma normal disyuntiva de F :

1. Eliminar los bicondicionales usando la equivalencia

$$A \leftrightarrow B \equiv (A \rightarrow B) \wedge (B \rightarrow A) \quad (1)$$

2. Eliminar los condicionales usando la equivalencia

$$A \rightarrow B \equiv \neg A \vee B \quad (2)$$

3. Interiorizar las negaciones usando las equivalencias

$$\neg(A \wedge B) \equiv \neg A \vee \neg B \quad (3)$$

$$\neg(A \vee B) \equiv \neg A \wedge \neg B \quad (4)$$

$$\neg\neg A \equiv A \quad (5)$$

4. Interiorizar las disyunciones usando las equivalencias

$$A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C) \quad (6)$$

$$(A \vee B) \wedge C \equiv (A \wedge C) \vee (B \wedge C) \quad (7)$$

Ejemplo de transformación a FND

Ejemplo de cálculo de una FND de $\neg(p \wedge (q \rightarrow r))$:

$$\begin{aligned} & \neg(p \wedge (q \rightarrow r)) \\ \equiv & \neg(p \wedge (\neg q \vee r)) && [por (2)] \\ \equiv & \neg p \vee \neg(\neg q \vee r) && [por (3)] \\ \equiv & \neg p \vee (\neg\neg q \wedge \neg r) && [por (4)] \\ \equiv & \neg p \vee (q \wedge \neg r) && [por (5)] \end{aligned}$$

Tabla de verdad

Una forma de verificar y determinar si una fórmula proposicional es verdadera o no es usando tablas de verdad. Además, al realizar la tabla de verdad obtenemos el número de modelos (asignaciones) que hacen verdadera la fórmula. Por ejemplo, si tenemos la fórmula:

sea $F = (\neg p \vee \neg q \vee r) \wedge (q) \wedge (\neg q \vee \neg r)$

Al desarrollar su tabla de verdad como se muestra en la Tabla 2.1, en la última columna se obtiene el número de modelos (en este caso el número de unos) por lo que F en este ejemplo tiene solo un modelo donde la fórmula es verdadera. Por lo tanto, F es satisfactible y tiene un modelo.

Tabla 2.1. Tabla de verdad para F

$\neg p$	$\neg q$	$\neg r$	q	r	$\neg p \vee \neg q \vee r$	q	$\neg q \vee \neg r$	$(\neg p \vee \neg q \vee r) \wedge (q) \wedge (\neg q \vee \neg r)$
0	0	0	1	1	1	1	0	0
0	0	1	1	0	0	1	1	0
0	1	0	0	1	1	0	1	0
0	1	1	0	0	1	0	1	0
1	0	0	1	1	1	1	0	0
1	0	1	1	0	1	1	1	1
1	1	0	0	1	1	0	1	0
1	1	1	0	0	1	0	1	0

En el siguiente capítulo se describe el algoritmo de inferencia proposicional que nos va a permitir dada una nueva información poder determinar si dicha información se deriva o infiere de la base de conocimiento o es necesario agregar nuevos datos para lograr la inferencia proposicional.

Capítulo 3. Algoritmo de inferencia proposicional

El algoritmo desarrollado e implementado como un sistema web nos permite determinar: si una formula normal conjuntiva que representa la nueva información puede ser inferida de la base actual de conocimiento. Para mostrar el diseño del algoritmo propuesto iniciamos con algunas definiciones básicas de lógica proposicional, posteriormente mostrar las operaciones básicas del algoritmo y finalmente el pseudocódigo del algoritmo desarrollado.

3.1. Definiciones básicas

La base de conocimiento puede estar formada, además, por un conjunto de reglas o normas, incluso leyes que todo ser humano debe seguir o cumplir, y es en este sentido donde la inferencia lógica contribuye a determinar si la nueva información ya está presente en nuestra base de conocimiento o no. Para poder representar el proceso de revisión de la base de conocimiento cuando llega nueva información es necesario las siguientes definiciones básicas.

Definición 1. Sea $X = \{x_1, \dots, x_n\}$ formado por un conjunto de n variables booleanas x_1, \dots, x_n .

Un literal denotada como *lit*, es una variable x_i o la negación de la variable $\neg x_i$, usualmente, cada $x \in X$, $x^0 = \neg x$ y $x^1 = x$.

Definición 2. Consideramos una *cláusula* como una disyunción de diferentes literales. Una *k-cláusula* es una cláusula con exactamente k literales, en general, vamos a considerar que una cláusula es un conjunto de literales. Por ejemplo: $(p \vee \neg q \vee \neg r)$, es una cláusula que tienen 3 literales unidas por el operador \vee .

Definición 3. Consideramos una *frase* como una conjunción de literales. Una *k-frase* es una frase con exactamente k literales. Por ejemplo $(p \wedge \neg q \wedge \neg r)$, es una frase que contienen 3 literales unidas por el operador \wedge .

Definición 4. Consideramos que una forma normal conjuntiva (FNC) es una conjunción de cláusulas, y *k-FNC* es una FNC que contiene *k-cláusulas*. Una forma normal disyuntiva (FND) es una disyunción de sentencias, y *k-FND* es una FND que contiene *k-frases*. Una FND con n variables es una *n-aria* función booleana $F: \{0,1\}^n \rightarrow \{0,1\}$.

Por ejemplo, sea $K = \{(p \vee \neg q \vee \neg t) \wedge (q \vee r \vee \neg s) \wedge (q \vee \neg r \vee s) \wedge (p \vee q \vee \neg r)\}$. En este caso K es una FND que incluye las variables p, q, r, s, t . Además, K está compuesta de 4 cláusulas con diferentes números de literales.

Definición 5. Una *asignación* s para una fórmula F es un mapeo booleano $s: \nu(F) \rightarrow \{1, 0\}$. Una asignación s pueden ser considerado como un conjunto no-complementario de literales: $l \in s$ si y solo si s asigna verdadero a l y falso a $\neg l$. s es una asignación parcial para una fórmula F cuando s tiene un determinado valor lógico para un subconjunto de variables de F , llamado $s: Y \rightarrow \{1, 0\}$ y $Y \subset \nu(F)$. Una CF F se satisface por una asignación s si cada cláusula F se satisface por s . Un modelo F es una asignación $\nu(F)$ que satisface F .

Por ejemplo, si $K = \{(p \vee \neg q \vee \neg t) \wedge (q \vee r \vee \neg s)\}$, tenemos que $s = (1,1,1,0,1)$ es una asignación donde: $p = 1$ (verdadero), $q = 1$, $r = 1$, $s = 0$ (falso); $t = 1$. Donde al evaluar la asignación s en la fórmula K , s puede ser satisfactible o no.

Un modelo de K es una asignación que resulta verdadera, por ejemplo, en la fórmula K si $p = 1$ y $q = 1$, la fórmula es verdadera independientemente de los valores de las demás literales.

Al evaluar una fórmula dada una asignación s se pueden dar uno de tres casos:

- Tautología: donde para cualquier valor siempre resulta verdadera, es decir para n variables se tienen 2^n modelos (todos con valor verdadero - 1).
- Contradicción: para cualquier asignación s resulta siempre falso, es decir, para n variables se tienen 0 modelos (todos son falsos-0).
- Contingencia: donde se tienen entre 1 y $2^n - 1$ modelos, es decir, no es ni tautología ni contradicción.

Definición 6. Dadas dos cláusulas C_i and C_j , si tienen al menos una literal complementaria, entonces son llamadas cláusulas independientes, de otra forma, se dice que ambas cláusulas son dependientes.

Definición 7. Sea $K = \{C_1, C_2, \dots, C_m\}$ una CF. K es llamada independiente si para cualquier par de cláusulas $C_i, C_j \in K, i \neq j$, se encuentra la propiedad de independencia.

Definición 8. Dado un par de cláusulas independientes C_1 y C_2 , si $lit(C_1) \subseteq lit(C_2)$ entonces se dice que C_2 es subsumida por C_1 .

3.2. Operaciones básicas

Con el objetivo del mejor manejo de las operaciones lógicas entre cláusulas usamos transformamos cada cláusula como una cadena de 1, 0 y * de esta manera es más fácil poder manipular algorítmicamente cadenas de símbolos.

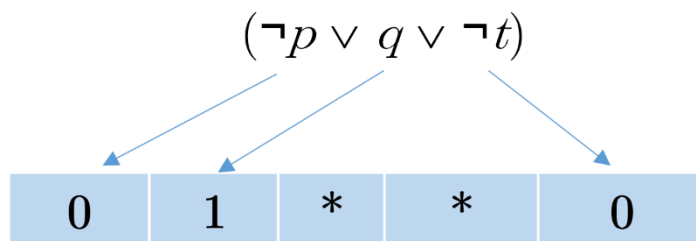


Figura 3.1. Transformación de una cláusula en patrón de cadenas.

Cada cláusula se transforma se transforma usando el patrón de 0, 1, y *, como se indica en la Figura 3.1. Determinamos el conjunto de variables de acuerdo con todas las cláusulas y las literales que no aparecen en la cláusula se les asigna el símbolo * (asterisco). El * significa que en dicha cláusula no está presente la variable correspondiente considerando el orden alfabético.

Ejemplo: sea $K = \{(\neg p \vee q \vee t) \wedge (\neg q \vee \neg r \vee s) \wedge (\neg q \vee r \vee \neg s) \wedge (\neg p \vee \neg q \vee r)\}$ una fórmula en FNC. Las variables que involucra son: p, q, r, s y t . En términos de cadenas obtenemos los valores de verdad de las literales como se muestra a continuación: $K = \{(01**1), (*001*), (*010*), (001**)\}$, al tratarse de una FNC sabemos que entre cada valor existe un operador \vee y entre cada cláusula hay un operador \wedge .

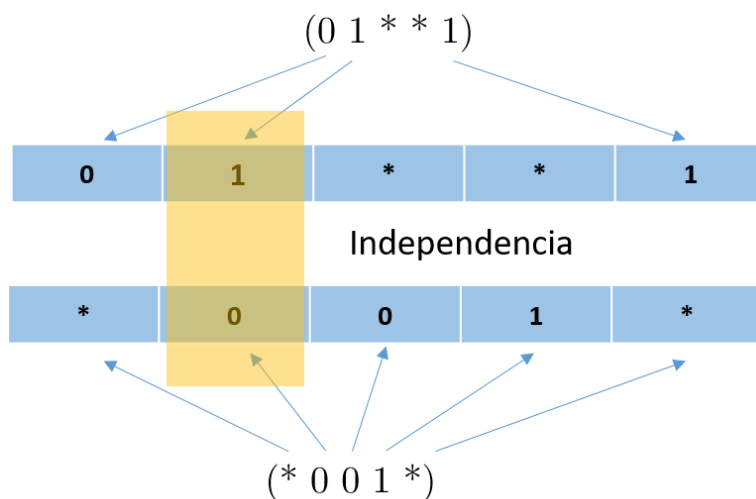


Figura 3.2. Operación de independencia entre dos cláusulas.

Independencia: Dos clausulas se consideran independientes si tienen valores contrarios para la misma posición de la cadena de ceros, unos y asteriscos.

La Figura 3.2 muestra el caso de dos cláusulas que son independientes (marcadas con color amarillo claro).

La independencia entre cláusulas nos permite determinar cuándo una cláusula tiene valores contradictorios, dichas cláusulas independientes ya no generan más cláusulas. El concepto de independencia es la parte principal del algoritmo desarrollado en el sistema web de revisión de creencias.

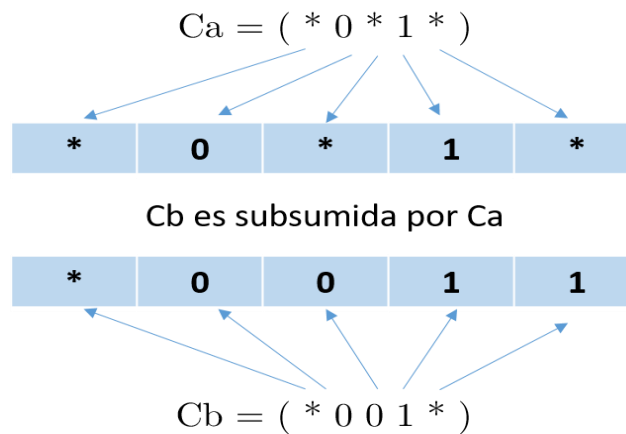


Figura 3.3. Operación cláusula Ca subsumida por Cb.

Subsumida: Una cláusula se considera subsumida si sus modelos son generados por otra de las cláusulas. Esto significa que la cláusula subsumida ya puede ser inferida por otra de las cláusulas y por lo tanto ya no es necesario seguirla considerando. En la Figura 3.3 se muestra esta operación de cláusula subsumida, note que cada uno de los caracteres de la cláusula Cb están contenidos en los símbolos de la cláusula Ca, considerando las mismas posiciones en cada cadena.

Diferencia: la diferencia entre la cláusula Ca y la cláusula Cb consiste en determinar el número de diferentes valores considerando las mismas posiciones y el patrón (*,1) o (*,0).

En la Figura 3.4 podemos observar que la cláusula C_b tiene dos de sus asteriscos que en la misma posición de la cláusula C_a hay valores diferentes del asterisco por lo tanto significa que hay una diferencia de 2.

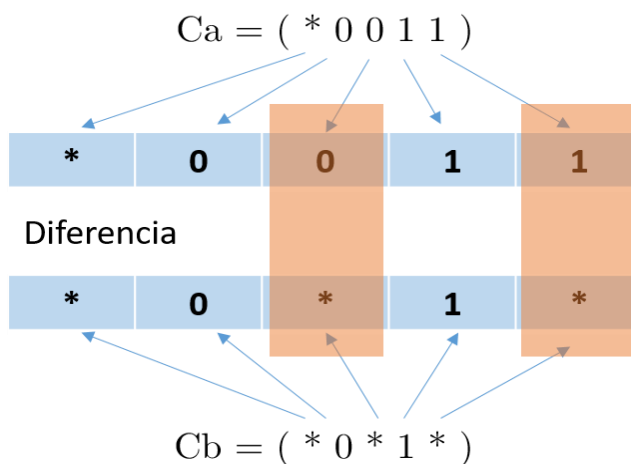


Figura 3.4. Diferencia entre cláusulas, en este caso la diferencia de C_b vs $C_a = 2$

Generación: la generación de nuevas cláusulas depende de la operación de diferencia entre cláusulas, es decir, la diferencia indica el número de nuevas cláusulas que se deben generar.

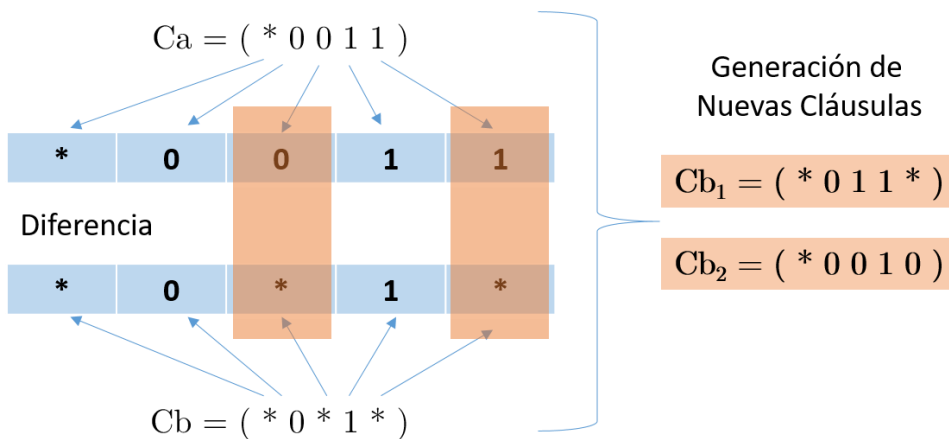


Figura 3.5. Para la generación de nuevas cláusulas depende de la diferencia de las cláusulas.

En la Figura 3.5 se muestra la operación de generación. En este caso se generan dos nuevas cláusulas generando independencia entre las cláusulas, en la primera cláusula $C_{b_1} = (*011*)$ en la tercera posición se coloca un 1, y se genera independencia y en la cláusula $C_{b_2} = (*0010)$ se deja el valor de 0 en la tercera posición y se cambia el último valor para generar independencia.

En la Figura 3.6 se muestra otro ejemplo de generación de nuevas cláusulas, como la diferencia es 3 se generan tres nuevas cláusulas.

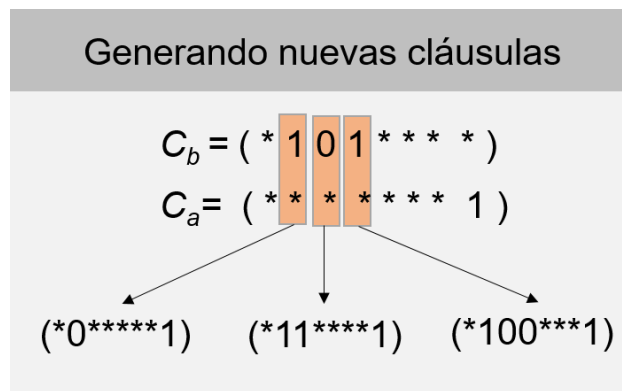


Figura 3.6. Generación de tres nuevas cláusulas.

Las operaciones antes mencionadas (Independencia, Subsumida y Generación de nuevas cláusulas) son las principales operaciones que son necesarias para el desarrollo del algoritmo de revisión de creencias usando inferencia proposicional.

En la Figura 3.7 se indican las operaciones en base a bits entre las cláusulas C_b al compararse con la cláusula C_a . El diagrama representa las posibles combinaciones entre bits y las operaciones que se deberían ampliar.

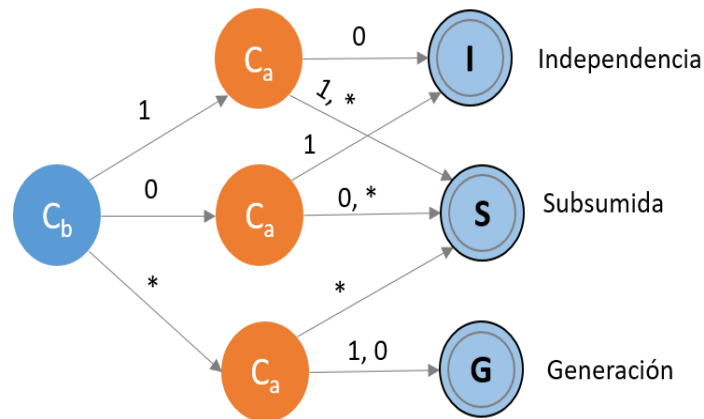


Figura 3.7. Transición de estados para llegar a las operaciones básicas del algoritmo.

En el diagrama de la Figura 3.8 se indican con una máquina de estados los posibles estados por lo que pasan los caracteres de una cadena para determinar la operación a realizar, por ejemplo, del estado inicial q_0 si en una cláusula hay un 1, pasa al estado q_1 , si hay un 1 o un * en la segunda cláusula entonces es una cláusula subsumida y continua hasta revisar cada carácter de las cláusulas.

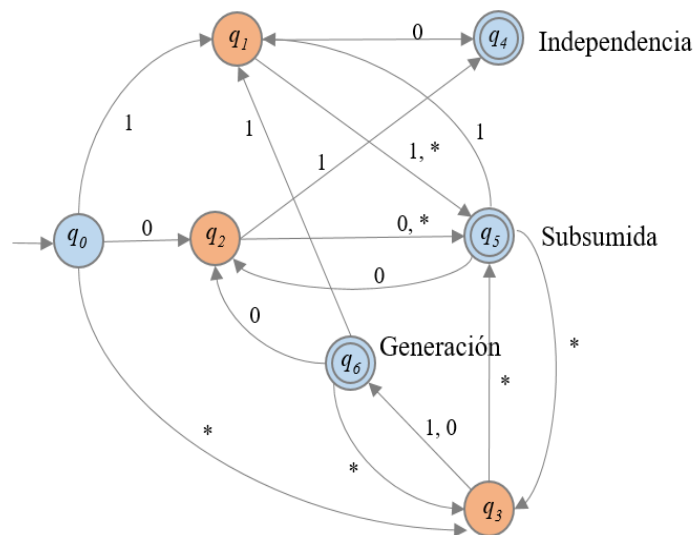


Figura 3.8. Máquina de estado de las operaciones básicas del algoritmo.

A continuación, se describe el algoritmo de revisión de creencias con inferencia proposicional que se basa en las operaciones antes indicadas.

3.3. Algoritmo de inferencia proposicional

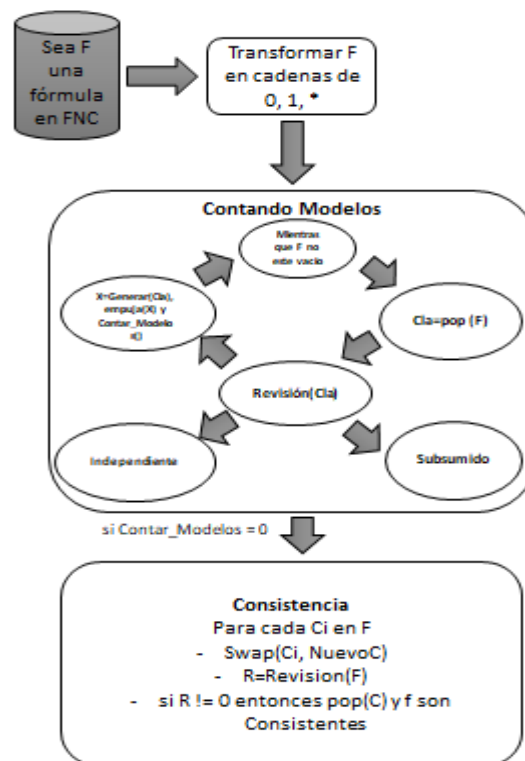


Figura 3.9. Proceso completo para revisar la consistencia de una base de conocimiento

Pasos seguidos (Figura 3.9) en el proceso de revisión de la base de conocimiento, donde

Entrada: B= Base De Conocimiento y N= Nueva información;

Salida: Conjunto S de cláusulas necesarias para la inferencia

1. Ya que se tienen las cláusulas con sus respectivas variables en ϕ y K se empezará a realizar la inferencia.
2. Se compara cada variable de las cláusulas para determinar si hay diferencia, independencia o si es subsumida.
3. Se da prioridad a la regla de “independencia”, si es así, la cláusula pasa igual.
4. Lo que se debe verificar después es si se aplica la regla de subsumida.
5. Por último, se revisa si hay diferencia.
6. Se hace la revisión de creencias, si $K \models \phi$ entonces no cambia K .
7. Si $K \not\models \phi$ entonces es necesario generar nuevas cláusulas a la base de conocimientos.

Algoritmo en pseudocódigo

Inicio

Para cada N_i **hacer**

Mientras ($B_i \neq \text{Vacio}$) **hacer**

$\text{Ind} = \text{Independencia}(N_i, B_j)$;

$\text{Sub} = \text{Subsumida}(N_i, B_j)$;

$\text{Dif} = \text{Diferencia}(N_i, B_j)$;

Si ($\text{ind}=1$) **Entonces**

B_i++ ;

Sino

Si ($\text{sub}=1$) **Entonces**

N_i++ ;

break;

Sino

$G_i = \text{Generar}(\text{dif})$;

$\text{Push}(G_i, N_i)$;

Finsi

Finsi

FinMientras

$\text{Push}(N_i, S)$;

FinPara

Fin

3.4. Pruebas

En el área de razonamiento automático se modelan problemas con lógica proposicional, por ejemplo, dado el siguiente conjunto de proposiciones representan una base de conocimiento de un asistente virtual A para la compra de pañales en línea, la FNC asociada es:

p : comprar pañales

q : pañales de la marca

r : marca blanca

s : cantidad requerida

t : paquete individual

Formamos el conjunto de instrucciones del asistente virtual:

- Comprar pañales (p)
- No hay pañales para comprar o no hay pañales de marca o hay pañales de marca blanca ($\neg p \vee \neg q \vee r$)
- No hay la cantidad requerida o hay pañales de marca ($\neg s \vee q$)
- No hay paquete individual o hay pañales de marca ($\neg t \vee q$)
- Hay paquete individual (t)

Trasformando A en su FNC, tenemos que:

$A = \{(\neg p) \wedge (\neg p \vee \neg q \vee r) \wedge (\neg s \vee q) \wedge (\neg t \vee q) \wedge (t)\}$, convertimos A en su patrón correspondiente de 0,1 y *, obteniendo:

la base de conocimiento, tal es el caso de la instrucción al asistente virtual: “no comprar pañales o que no sean de marca blanca o no comprar paquete individual”, si lo transformamos en una cláusula $(\neg p \vee \neg r \vee \neg t) = (0^*0^*0)$, al agregarse a la base de conocimiento A como se muestra en la Tabla 3.2 se necesita la cláusula 000^*0 para lograr la inferencia.

Tabla 3.2. Verificar la inferencia de 0^*0^*0

	1****	0***1	01*00	01*10	001*0	
0*0*0	0*0*0	0*0*0	000*0	000*0	000*0	000*0

Otro ejemplo de aplicación del algoritmo se muestra en la Tabla 3.3, donde la base de conocimiento que se modela es $F = \{(\neg t) \wedge (p \vee q \vee \neg s \vee t) \wedge (\neg r) \wedge (p \vee q \vee \neg r \vee s \vee t)\}$, la columna marcada como conteo representa la suma de los modelos necesarios para determinar si la formula F es satisfactible, es decir, si el numero de modelos es mayor a cero, el número de modelos cada cláusula se obtiene como $2^{\#\text{asteriscos}}$ y la suma total se resta al valor de $2^{\#\text{variables}}$, entonces, $\text{Modelos}(F) = 2^5 - 29 = 32 - 29 = 3$.

Tabla 3.3. Conteo de modelos de F

Ci	S				TM	Conteo
		*01*1	*01*1	*01*1	*01*1	4
	*0**1	*0001	*0001	00001	00001	1
****1	*11*1	*11*1	*11*1	*11*1	*11*1	4
	*1001	*1001	*10001	*1001	*1001	2
00*10	00*10	00110	00110	00110	00110	1
1	**1**	**1**	**1**	**1**	**1**	16
00100	00100	00100	00100	00100	00100	1
						29

Capítulo 4. Sistema de revisión de creencias

El sistema desarrollado consiste en una aplicación web implementando el algoritmo descrito en el capítulo 3 de este trabajo de tesis. Se utilizó el lenguaje PHP del lado del servidor y el lenguaje HTML y hojas de estilo en cascada para la vista del lado del cliente. Para el desarrollo del árbol de derivación se utilizó la librería vis.js². Esta librería vis.js es una biblioteca de visualización dinámica basada en navegador. La biblioteca está diseñada para ser fácil de usar, manejar grandes cantidades de datos dinámicos y permitir la manipulación e interacción con los datos. La biblioteca consta de los componentes DataSet, Timeline, Network, Graph2d y Graph3d. Esta biblioteca fue desarrollada en el lenguaje JavaScript. Para la ejecución del sistema es necesario contar con un servidor web con PHP, en nuestro caso utilizamos la herramienta WampServer³ que provee un servidor apache y que permite la ejecución de los programas en forma de servidor local (<http://localhost>) usando un navegador web.

4.1. Vista general del sistema

El sistema de revisión de creencias con inferencia proposicional consta de un menú principal con las siguientes opciones:

- Inicio: se presenta la pantalla principal del sistema.
- Inferencia: Aquí se describe el menú para realizar el proceso de inferencia.
- Actualizar KB: Permite cargar una nueva base de conocimiento para realizar pruebas del proceso de inferencia.

² <https://visjs.org/>

³ <https://www.wampserver.com/>

- Realizar inferencia: Permite cargar la nueva información para realizar el proceso de inferencia contra la base de conocimiento actual.
- Contacto: Se describe los autores del sistema desarrollado.

4.2. Inicio del sistema

En la Figura 4.1 se muestra la pantalla de inicio del sistema de revisión de creencias con inferencia proposicional. El sistema se ejecuta a través de un navegador web usando un servidor web local. La pantalla inicial consta de un menú (INICIO, INFERENCIA y CONTACTO), además, se describe en esta primera pantalla el problema de revisión de creencias a través del modelo que estamos utilizando para determinar el proceso de inferencia y verificar si la nueva información ϕ se infiere o no de la base de conocimiento K .



Figura 4.1. Pantalla inicial del sistema de revisión de creencias usando inferencia proposicional.

4.3. Proceso de Inferencia

En la Figura 4.2 se describe el proceso de inferencia proposicional que cuenta con un submenú con dos opciones: Actualizar KB y Realizar Inferencia, además se describe cada uno de los elementos que son necesarios para el proceso de revisión de creencias.

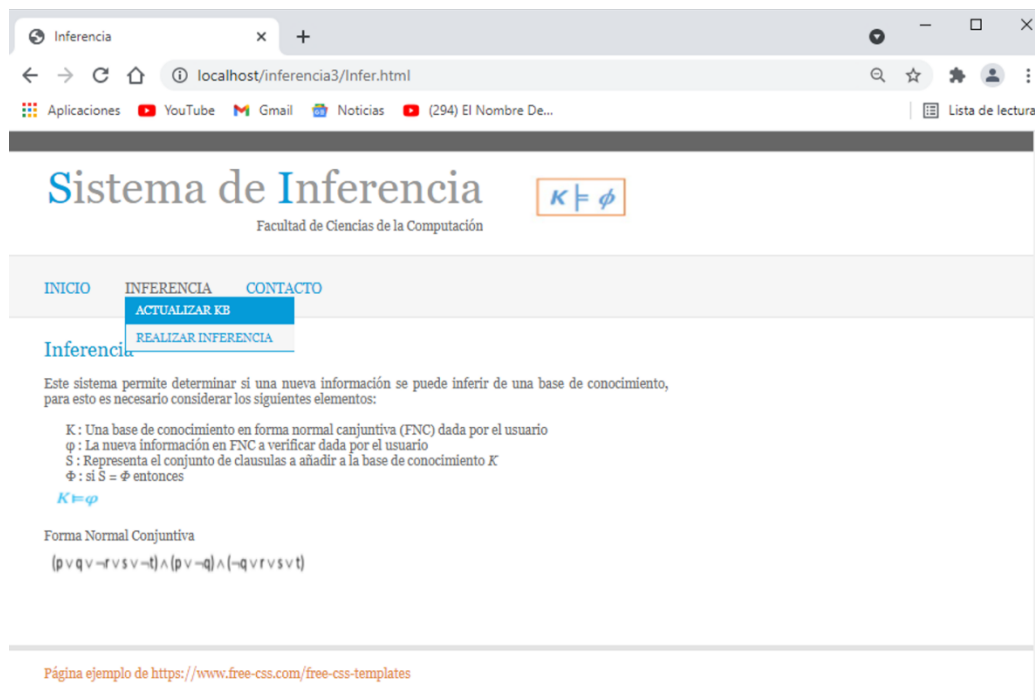


Figura 4.2. Menú de la opción de inferencia.

Al seleccionar en la opción de inferencia la sección de Actualizar KB (como se muestra en la Figura 4.3), se muestra una pantalla que permite seleccionar el archivo de la base de conocimiento K . El archivo que contiene la fórmula K en FNC es un archivo de texto simple donde en la primera línea indica el número de cláusulas de la base de conocimiento y el número de variables de cada cláusula, a continuación, se escribe cada cláusula en una línea. Por ejemplo sea K la base de conocimiento $K = \{(p \vee \neg q \vee \neg t) \wedge (q \vee r \vee \neg s) \wedge (q \vee \neg r \vee s) \wedge (p \vee q \vee \neg r)\}$, transformando a ceros, unos y asteriscos, tenemos que $K = \{(10**0)(*110*)(*101*)(110**)\}$. Almacenando

esta información en un archivo de texto tenemos 4 cláusulas y 5 variables, se tiene la siguiente estructura:

```
-----
4 5
10**0
*110*
*101*
110**
-----
```



Figura 4.3. Pantalla para seleccionar el archivo de texto que contiene la base de conocimiento a revisar.

Una vez que se carga la base de conocimiento para el proceso de revisión de creencias es necesario cargar el archivo con la nueva información ϕ , la estructura del archivo de ϕ es igual a la estructura del archivo de la base de conocimiento, es decir, se trata de un archivo de texto con el número

de cláusulas y de variables además de cada una de las cláusulas de la nueva información. En la Figura 4.4 se muestra la pantalla para cargar el archivo con la nueva información para realizar el proceso de inferencia y verificar si $K \models \phi$, Podemos notar que se despliega la base de conocimiento actual con fines de información.



Figura 4.4. Pantalla para seleccionar el archivo de texto que contiene la base.

Cuando se carga el archivo con la nueva información ϕ al dar aceptar de forma automática se realiza el proceso de inferencia. En la Figura 4.5 se muestra el proceso de revisión de creencias con inferencia proposicional, en la pantalla se muestra la base de conocimiento K , la nueva información ϕ y el conjunto de salida S .

Suponga que la nueva información es $\phi = \{(1*1**) (*10**) (0011*) (***)0\}$. Si S es vacío significa que se cumple que $K \models \phi$ y en caso contrario se genera el conjunto de cláusulas necesarias para que se cumpla que $K \models \phi$.

Sistema de Inferencia

Facultad de Ciencias de la Computación

$K \models \phi$

INICIO INFERENCIA CONTACTO

Inferencia proposicional

K (BaseCyS-2.txt)

(10**0) (*110*) (*101*) (110**)

ϕ (FiCyS-2.txt)

(1*1**) (*10**) (0011*) (****0)

S

(11110) (01110) (01000) (00**0) (0100*) (101*1) (1111*)

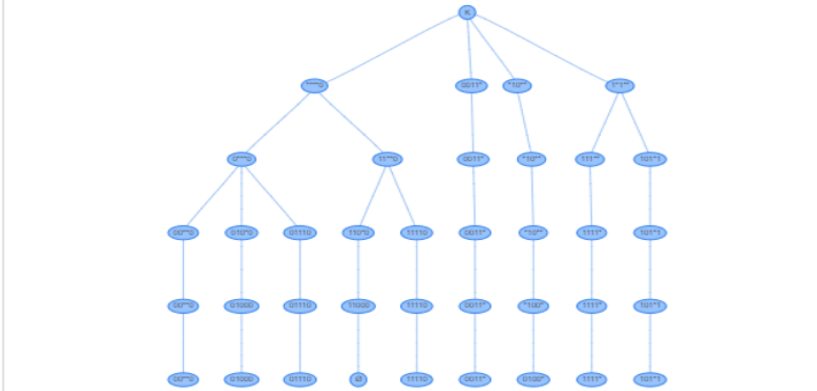


Figura 4.5. Proceso de inferencia proposicional

El grafico (diagrama jerárquico) que se muestra en la Figura 4.5 representa el proceso de revisión y generación de cláusulas necesarias para que $K \models \phi$. En la Figura 4.6 se muestra de forma más clara el diagrama jerárquico correspondiente. El árbol representa la salida S del algoritmo de inferencia, en este caso $S = \{(11110)(01110)(01000)(00**0)(0100*)(101*1)(1111*)\}$, que a su vez en FNC $S = \{(p \vee q \vee r \vee s \vee \neg t) \wedge (\neg p \vee q \vee r \vee s \vee \neg t) \wedge (\neg p \vee q \vee \neg r \vee \neg s \vee \neg t) \wedge (\neg p \vee q \vee \neg t) \wedge (\neg p \vee q \vee \neg r \vee \neg s) \wedge (p \vee \neg q \vee r \vee t) \wedge (p \vee q \vee r \vee s)\}$.

El árbol fue creado con la librería vis.js y permite ampliarse y moverse con el fin de ver los detalles de este, además para entradas grandes de la base de conocimiento o de la nueva información es muy útil el uso de zoom.

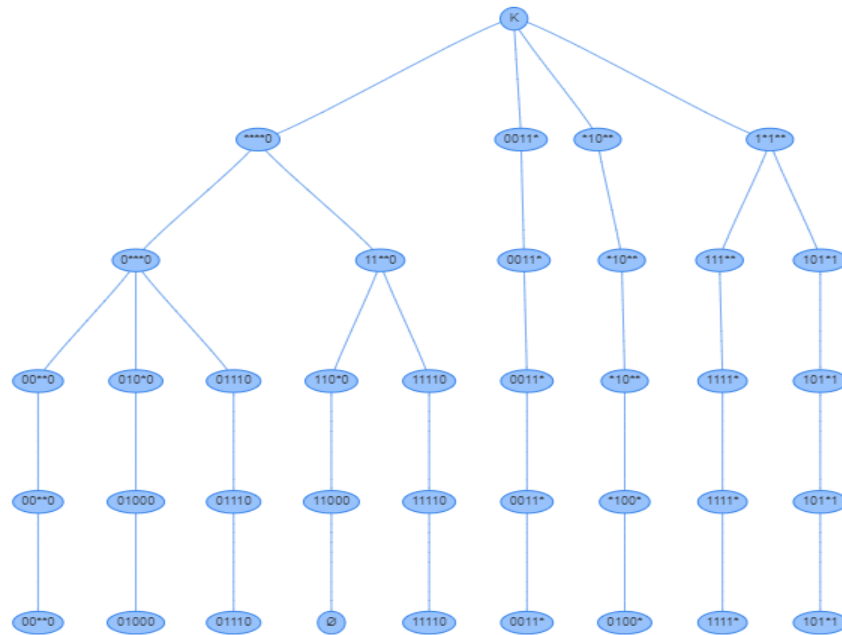


Figura 4.6. Árbol jerárquico del proceso de inferencia proposicional.

Finalmente, en la Figura 4.7 se muestra la pantalla de contacto.



Figura 4.7. Árbol jerárquico del proceso de inferencia proposicional.

Conclusiones

Las teorías de revisión de creencias constituyen un enfoque relativamente novedoso en el razonamiento de agentes por lo que explorar esta área del conocimiento con operadores de revisión de creencias permite modelar problemas prácticos.

Aplicar teorías de lógica proposicional, lógica de predicados, lógica deóntica entre otras posibilidades amplía el marco de estudio de la teoría de revisión de creencias.

Las teorías de revisión de creencias resultan atractivas además para todos aquellos interesados en razonamientos no monotónicos en general. Φ se infiere no monotónicamente de K , dado una KB, si y sólo si Φ pertenece al conjunto K revisado con Φ .

Los modelos de revisión de creencias pueden servir además para comprender y evaluar cambios en sistemas legales como en la lógica deóntica.

El sistema de revisión de creencias aquí plasmado es de gran utilidad para realizar pruebas con diferentes instancias de la base de conocimiento y la nueva información con el fin de mejorar la complejidad de los algoritmos.

Como todo sistema se puede mejorar por ejemplo permitir la generación de instancias de prueba aleatorias, permitir agregar nueva información de diferentes agentes, ampliar la lógica proposicional a otro tipo lógica como lógica deóntica, lógica trivalente, etc.

Bibliografía

- [1] Alchourrón, C., Gärdenfors, P. & Makinson, D. (1985). On the logic of theory change. Partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50, 510–530.
- [2] Katsuno, H. & Mendelzon, A. O. (1991). On the difference between updating a knowledge base and revising it. *KR'91 Cambridge, MA, USA*, Vol. 1, pp. 387–394.
- [3] González R. C. Aproximación al concepto de inferencia desde dos modelos de comprensión: modelo estratégico y modelo de construcción e integración. *Literatura y Lingüística*, 35:297, 314, 2017.
- [4] Fermé, E. (2007). Revisión de creencias. *Revista Iberoamericana de Inteligencia Artificial*, Vol. 11, pp. 17–39.
- [5] Cresto, E. (2002). Revisión de creencias y racionalidad. *Cuadernos CIMBAGE*, Vol. 5, pp. 133–156.
- [6] Liberatore, P., Schaerf, M. (2001). The complexity of model checking for belief revision and update. *Journal of Computer and Systems Sciences*, Vol. 62, pp. 43–72.
- [7] Fermé E. y Hansson S. O. Agm 25 years, twenty-ve years of research in belief change. *Journal of Philosophical Logic*, 40:295, pp. 331, 2011.
- [8] Valdéz, N. J., Falapa, M. A. (2013). Dinámica de conocimiento: contracción múltiple en lenguajes. Horn. Proc. XIX congreso argentino de ciencias de la computación, Libro de actas - CACIC2013, pp. 61–70.
- [9] Delgrande, J. P. (2008). Horn clause belief change: Contraction functions. Proc. of the 11th Int. Conf. on the Principles of Knowledge Representation and Reasoning, AAAI, pp. 156–165.
- [10] Bello P. Modelando las normas con revisión de creencias e inferencia. *Revista Elementos BUAP*, 112:9, 13, 2018.

- [11] Fermé E. Revisión de creencias. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, 11(34):17–39, 2007.
- [12] Zhou, G., Xie, Z., Yu, Z., & Huang, J. X. (2021). Dfm: A parameter-shared deep fused model for knowledge base question answering. *Information Sciences*, vol. 547, pp. 103 – 118.
- [13] Zhang, L., Lin, C., Zhou, D., He, Y., & Zhang, M. (2021). A bayesian end-to-end model with estimated uncertainties for simple question answering over knowledge bases. *Computer Speech & Language*, vol. 66, pp. 101167.
- [14] Maisto, A., amd Antonietta Paone, G. M., & Pelosi, S. (2021). *Advances in Intelligent Networking and Collaborative Systems*, vol.1263, chapter Semi-automatic Knowledge Base Expansion for Question Answering (pp. 173–182), Springer, Cham.
- [15] Kurganova, M. V. (2021). *Economic Systems in the New Era: Stable Systems in an Unstable World*, chapter Enterprise Knowledge Base Based on EDM System (pp. 179–188), Springer, Cham.
- [16] Le, C., Shrestha, K. J., Jeong, H. D., & Damnjanovic, I. (2021). A sequential pattern mining driven framework for developing construction logic knowledge bases. *Automation in Construction*, vol. 121, pp. 103439.
- [17] Budiharto, W., Andreas, V., & Gunawan, A. A. S. (2020). Deep learning-based question answering system for intelligent humanoid robot. *Journal of Big Data*, vol. 7(1), pp. 77.
- [18] Peng, M., GangYi, D., QianKun, J., Fuquan, Z., & YangFan, J. (2020). Quantum-based creative generation method for a dancing robot. *Frontiers in Neurorobotics*, vol. 14, pp. 1-22.
- [19] Ye, Y., Hu, T., Yang, Y., Zhu, W., & Zhang, C. (2020). A knowledge based intelligent process planning method for controller of computer numerical control machine tools. *Journal of Intelligent Manufacturing*, vol. 31, pp. 1751–1767.
- [20] Haret, A. & Woltran, S. (2019). Belief revision operators with varying attitudes towards initial beliefs. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, International Joint Conferences on Artificial Intelligence Organization*, pp. 1726–1733.

- [21] Deagustini, C. A., Martinez, M. V., Falappa, M. A., & Simari, G. R. (2019). Belief base contraction by belief accrual. *Artificial Intelligence*, vol. 275, pp. 78–103.
- [22] Pons C., Rosenfeld R., y Smith C. *Lógica para informáticos*. Editorial de la Universidad de La Plata, Buenos Aires, Argentina, 1 edición., 2017.
- [23] De-Ita G., Marcial-Romero J. R., Hernández J. A., y Bello P. *Conocimiento y Razonamiento Computacional*. Academia Mexicana de Computación, México, 2018. ISBN 978-607-97357-3-9.
- [24] Creignou N., Papini O., Pichler R., y Wolfran S. Belief revision whithim fragments of propositional logic. *Journal of Computer and System Sciences*, 80(2):427{449, 2014.