

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

Facultad de Ciencias Químicas e Ingeniería

Maestría y Doctorado en Ciencias e Ingeniería



**MODELO HÍBRIDO DE REPRESENTACIÓN
GENERAL DE CONOCIMIENTO**

TESIS

Para obtener el grado de:

Maestro en Ciencias

Presenta:

Mauricio Alonso Sánchez Herrera

Director:

Dr. Juan Ramón Castro Rodríguez

Tijuana, Baja California, México, Septiembre del 2011

Abstract

A proposed hybrid model for knowledge representation is shown in this thesis, which in comparison to other known models unifies heterogeneous information utilizing a single notation which can better represent concepts in a more human readable way. By combining semantic networks, categories, dynamic state events, neural networks and fuzzy logic into one hybrid model, better real world conceptual representation can be achieved, and with this many new possibilities for simulation of knowledge, game intelligence, social simulation, robotics, etc.

Un modelo híbrido de representación de conocimiento se muestra en esta tesis, que en comparación con otros modelos unifica información heterogénea utilizando una misma notación que puede representar conceptos de una manera más legible al humano. Combinando redes semánticas, categorías, eventos de estados dinámicos, redes neuronales y lógica difusa en un modelo híbrido, una representación más afín a la representación real del mundo puede ser alcanzada, y con esto muchas nuevas posibilidades para simular conocimiento, inteligencia en videojuegos, simulación social, robótica, etc.

Agradecimientos

Agradesco el apoyo por parte del Dr Juan Ramón por haberme atorgado tanto tiempo para lograr terminar a tiempo mi tesis y por haber soportado tanta interrupción a sus labores para sacarme de dudas. Agradesco al Dr Antonio Díaz por haberme ayudado a abstraer más allá de lo que yo solo hubiera logrado con mi tema de tesis. Agradesco el apoyo de mi familia, amigos y novia por soportar escucharme hablar tanto de mi tema de tesis. También agradezco a Laura y Eduardo por haberme auxiliado con la implementación de mi tesis. Y finalmente agradezco a la UABC por haberme dado la oportunidad de estudiar el posgrado.

Índice general

Abstract	II
Agradecimientos	III
Tablas	VIII
Figuras	IX
1. Introducción	1
1.1. Planteamiento del problema	1
1.2. Objetivo	1
1.3. Objetivos particulares	2
1.4. Organización del documento	2
1.5. Aportaciones del trabajo	3
2. Fundamentos Teóricos	4
2.1. Representación del conocimiento	4
2.1.1. Redes semánticas	5
2.1.2. Marcos	8

2.1.3.	Reglas de producción difusas	9
2.1.4.	Bases de datos relacionales	10
2.2.	Categorías	11
2.2.1.	Aristóteles	12
2.2.2.	Kant	14
2.2.3.	Peirce	16
2.3.	Eventos de estados dinámicos	17
2.3.1.	Máquinas de estado finito	18
2.3.2.	Redes de petri	20
2.4.	Lógica difusa	21
3.	Sistemas de Representación del Conocimiento	24
3.1.	Proyectos	24
3.1.1.	OpenCYC	25
3.1.2.	ConceptNet	26
3.1.3.	YAGO	27
3.2.	Ventajas y desventajas de los proyectos de representación de conocimiento existentes	28
3.2.1.	Ventajas	29
3.2.1.1.	OpenCYC	29
3.2.1.2.	ConceptNet	29
3.2.1.3.	YAGO	29
3.2.2.	Desventajas	30
3.2.2.1.	OpenCYC	30

3.2.2.2.	ConceptNet	30
3.2.2.3.	YAGO	31
4. Diseño y Análisis del Modelo de Representación de Conocimiento		
	General	32
4.1.	Base de representación	33
4.2.	Valor de certidumbre	34
4.3.	Categorías	38
4.4.	Modelo propuesto	41
4.5.	Eventos de estados dinámicos	42
4.6.	Toma de decisiones	44
4.6.1.	Entradas	45
4.6.2.	Reglas	47
4.6.3.	Salidas	48
4.7.	Ejemplo heterogéneo del modelo de representación de conocimiento general	49
5. Implementación y Aplicaciones del Modelo Híbrido de Representación		
	General de Conocimiento	52
5.1.	Implementación	52
5.2.	Ejemplo de un robot trabajador	55
5.3.	Aplicaciones	60
5.3.1.	Sentido común	60
5.3.2.	Secuencias de estados de eventos dinámicos	62
5.3.3.	Toma de decisiones	63

5.3.4. Otras aplicaciones	64
6. Análisis de Resultados	67
6.1. Comparativa de representación de sentido común con redes semánticas	67
6.2. Comparativa de representación de conocimiento e inferencia difusa . .	68
6.3. Comparativa de representación y ejecución de secuencias con redes de petri	70
7. Conclusiones y trabajo futuro	73
7.1. Conclusiones	73
7.2. Trabajo futuro	74
Referencias	76

Índice de tablas

2.1. Lista de categorías de acuerdo a Aristóteles.	13
2.2. Lista de categorías de acuerdo a Kant.	15
2.3. Lista de categorías de acuerdo a Peirce.	16
5.1. Tabla de verdad de lógica XOR.	64

Índice de figuras

2.1. Ejemplo simple de una red semántica describiendo a una manzana.	6
2.2. Ejemplo simple de marcos describiendo a un ave kiwi.	9
2.3. Ejemplo simple de base datos relacional que describe las ventas hechas por los vendedores de un establecimiento.	11
2.4. Ejemplo genérico de una máquina de estado finito.	19
2.5. Ejemplo simple de la visualización una red de petri.	21
4.1. Ejemplo de uso de lógica difusa en las relaciones. (a) Se acaba de exponer a la afirmación que un tomate es una verdura. (b) Posterior exposición a la afirmación que un tomate es una fruta. (c) Tras constante exposición a la afirmación correcta de que un tomate es una fruta, se fortalece la relación tomate-fruta, y se debilita la relación tomate-verdura hasta desaparecer esta última.	36
4.2. Ejemplo de uso de categorías en los conceptos. (a) Representación típica de una red semántica de la afirmación: una manzana tiene la propiedad de ser roja. (b) Representación de la misma afirmación, pero con las categorías adecuadas, agregando mayor significado a cada nodo concepto.	39

4.3.	Ejemplo de desambiguación con el uso de categorías en los conceptos.	
	(a) Concepto de bote referente a un objeto físico, en este caso, un navío.	
	(b) Concepto de bote referente a la acción de rebotar un objeto sobre una superficie.	40
4.4.	Ejemplo de uso del valor de certidumbre en la decisiones no deterministas. (a) Caso donde la persona en cuestión tiene mayor preferencias a irse a su casa en lugar de continuar trabajando. (b) Caso donde la persona en cuestión tiene fuertes tendencias a continuar trabajando tras acabar su jornada de trabajo.	45
4.5.	Generalización de la representación de las entradas a la toma de decisiones basado en reglas de producción difusas escritas.	46
4.6.	Generalización de la representación de las reglas para la toma de decisiones basado en reglas de producción difusas escritas.	47
4.7.	Generalización de la representación de las salidas para la toma de decisiones basado en reglas de producción difusas escritas.	49
4.8.	Ejemplo de representación de conocimiento heterogéneo, donde se representa a una manzana, como comerla y cuando decidir si seguir comiendo o no. (a) Representación gráfica de una manzana en conocimiento computable por un algoritmo de reconocimiento de patrones. (b) Representación de características generales de una manzana y fruta. (c) Secuencia de acciones generales para la acción de comer. (d) Representación de la decisión para saber si seguir comiendo o no.	51
5.1.	Secuencia principal de estados y acciones abstractas del robot trabajador.	56

5.2. Secuencias de acciones granulares para tomar y dejar una bola.	57
5.3. Representación descriptiva del material utilizado y la secuencia de maquinado.	58
5.4. Representación descriptiva del material utilizado y la secuencia de maquinado.	59
5.5. Ejemplo simple de como se infiere el conocimiento de una fruta, sin saber propiamente que es una fruta, hacia el conocimiento propio de lo que son las frutas, conocidas hasta el momento. (a) Conocimiento actual sobre frutas. (b) Conocimiento nuevo que desea inferirse hacia el conocimiento propio. (c) Resultado luego de la inferencia y se concluye que el conocimiento nuevo en realidad en una fruta también.	62
5.6. Ejemplo de representación de un modelo utilizando la notación prop- uesta. El modelo en cuestión es el de un sistema de productor-consumidor.	63
5.7. Ejemplo de representación de un FIS, lógica XOR, utilizando la no- tación propuesta.	65
5.8. Ejemplo hipotético de posible uso en una interfaz de usuario adaptable de forma dinámica al perfil del usuario. (a) Perfil de usuario represen- tado con la notación propuesta. (b) Base de conocimientos para inferir que tipo de interfaz adaptarle al usuario. Se utilizarían funciones de membrecía en los gránulos para poder decidir de forma difusa las pref- erencias.	66

6.1. Comparativa de conocimiento de tipo sentido común utilizando una red semántica típica y el modelo de representación propuesto. (a) Representación con red semántica. (b) Representación con el modelo propuesto.	69
6.2. Representación de lógica XOR utilizando la notación propuesta. . . .	71
6.3. Comparativa de un modelo de sistema productor-consumidor con red de petri y con el modelo de representación propuesto. (a) Representación utilizando una red de petri. (b) Representación utilizando la notación propuesta.	72

Capítulo 1

Introducción

1.1. Planteamiento del problema

Cuando imaginamos a una inteligencia artificial, se imagina una máquina que razona de forma semejante a como un humano. Para lograr esto, hay que tener una base fuerte que soporte todos los algoritmos necesarios para llegar a la finalidad mencionada. Por lo que el enfoque de esta tesis es proponer una base de conocimiento versátil para almacenar el conocimiento perceptible para razonar de forma inteligente. Con ello, iniciar una investigación a largo plazo en la creación de una inteligencia artificial que razone y procese de forma similar a un humano.

1.2. Objetivo

Crear una base de conocimiento lo suficientemente robusta y que acepte conocimiento de tipo heterogéneo, se puede utilizar en conjunto con algoritmos inteligentes para simular un aprendizaje autónomo. Por lo que se propone generar una base de conocimiento que soporte dominios generales y heterogéneos, para que este sirva como

base inicial a una investigación prolongada con la finalidad de comprobar la hipótesis propuesta.

1.3. Objetivos particulares

Hibridar redes semánticas con lógica difusa en un modelo que soporte conocimiento multidominio. Para obtener una representación más acercada al pensamiento humano mediante la inclusión de teoría de lógica difusa.

Aplicar el modelo en problemas reales y de investigación.

1.4. Organización del documento

Esta tesis está organizada de tal forma que lleva al lector con el razonamiento paso a paso de como se desarrolló el modelo propuesto, así como unos ejemplos aplicados. En el capítulo 2 se presentan algunos fundamentos teóricos necesarios para comprender la naturaleza del modelo propuesto. En el capítulo 3 se muestran varios sistemas de representación de conocimiento conocidos, para mostrar el campo previo a esta tesis. En el capítulo 4 se introduce el diseño y análisis del modelo, llevando al lector a través de varios puntos clave de la hibridación con otros métodos y modelos. En el capítulo 5 se muestran algunos ejemplos de implementación y aplicación, mostrando el rango de aplicaciones que este puede llevar a cabo. En el capítulo 6 se hace un análisis de resultados comparando soluciones conocidas con soluciones utilizando el modelo propuesto. Finalmente, en el capítulo 7 se concluye y se muestran algunas ideas para trabajo a futuro.

1.5. Aportaciones del trabajo

Con el trabajo realizado en ésta investigación se desprendieron varias presentaciones, un capítulo de libro, y una aplicación; que serán enumeradas a continuación:

- **Seminario:**

Modelo modular híbrido de aprendizaje general autónomo Mauricio A. Sanchez, Juan R. Castro; *Seminario académico* 2010 – 1; presentado 4 de Mayo del 2010.

- **Poster:**

A modular hybrid model for autonomous general learning Mauricio A. Sanchez, Juan R. Castro; *WCMCS* 2009; presentado 9 de noviembre del 2009.

- **Workshop:**

Modular hybrid model for general knowledge representation Mauricio A. Sanchez, Juan R. Castro, Antonio Rodriguez-Diaz; *WCMCS* 2010; presentado 4 Noviembre del 2010.

- **Congreso internacional:**

Hybrid model for general knowledge representation Mauricio A. Sanchez, Juan R. Castro, Antonio Rodriguez-Diaz, Oscar Castillo; 2011 *IFSA World Congress – AFSS International Conference*; ITS, Surabaya, Indonesia; presentado 22-23 de Julio del 2011; ISBN 978-602-99359-0-5

- **Software desarrollado:**

General knowledge graph software v1.0 Eduardo Leon, Laura Aguilar, Mauricio A. Sanchez; ver Capítulo 5.

Capítulo 2

Fundamentos Teóricos

2.1. Representación del conocimiento

El conocimiento es una forma simbólica de representar lo que un universo es en cierto momento [1], todo humano tiene conocimiento de algo durante el lapso de su vida, sea un conocimiento de una situación o de la descripción de un determinado objeto. Cosa que es natural para un humano, representar conocimiento en su mente, es muy difícil para una computadora, ya que para tener computadoras inteligentes se requiere que estas contengan algún tipo de conocimiento, sea general o específico a un problema. Todo recae en la problemática de como representar conocimiento dentro de una computadora, de tal forma que éstas puedan procesarlo y hacer algo de él [2]. Esto no fue de importancia, sino hasta cuando el área de inteligencia artificial sostuvo sus inicios y se quería reproducir una mente humana dentro de una computadora. Cosa que aún no ha sido logrado.

Un gran problema que ha existido desde los inicios de la representación de conocimiento ha sido la estandarización. Cada metodología o modelo de representación no es

compatible con otra representación, tienden a funcionar de forma aislada, por lo que si se quiere ingresar un conocimiento existente de otro modelo sin modificación alguna, no será posible ya que no comparten un mismo protocolo de interpretación. Pero hay que hacer notar, que aunque exista ese problema, por si solos si son lo suficientemente útiles como para ser aplicados, aunque de forma limitada.

A continuación se presentan las formas más comunes de representar conocimiento en una computadora.

2.1.1. Redes semánticas

Una forma muy conocida de representar y guardar conocimiento es mediante las redes semánticas, que vienen siendo un grafo dirigido donde cada vértice tiene una relación y cada nodo tiene un concepto, éstas redes semánticas a como dice su nombre, es una red compuesta de significados. Desde el inicio de éstas [3, 4], fueron creadas para representar relaciones granulares de primer grado con la confianza que al tener una red lo suficientemente grande se pudiera representar todo el conocimiento.

Las redes semánticas típicamente son representadas por un conjunto de conceptos o términos t_1, t_2, \dots, t_n , y éstos a su vez construyen un grafo

$$G = \langle V, A \rangle \tag{2.1}$$

donde,

V , es un conjunto de vértices o nodos del grafo. Y este conjunto representa cada uno de los conceptos t_1, t_2, \dots, t_n

y,

A , es un conjunto de aristas en el grafo. Dicho conjunto representa las relaciones entre conceptos tal que existirá un arista entre t_i y t_j si y solo si existe una relación entre

los conceptos t_i y t_j

En la Figura 2.1 se muestra un ejemplo de como se puede representar conocimiento

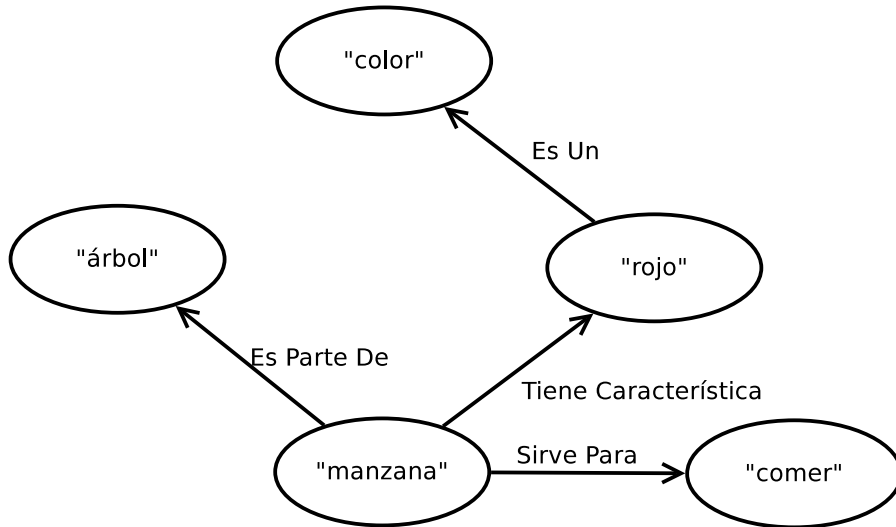


Figura 2.1: Ejemplo simple de una red semántica describiendo a una manzana.

del tipo sentido común.

La gran ventaja que tienen las redes semánticas sobre otros métodos de representación de conocimiento es la característica de poder tener relaciones definidas para expresar relaciones entre dos conceptos, esto es, que se indica de forma natural cual es la relación de un concepto respecto otro, por ej. gato es un felino, esto ya describe de forma directa la relación que existe entre el concepto *gato* y el concepto *felino*, mediante la expresión *es un*. La relación agrega la facilidad de poder agregar nuevo conocimiento mediante la interpretación de oraciones de lenguaje natural, cosa que varios proyectos que serán mencionados más adelante utilizan para aumentar la cantidad de conocimiento que éstos poseen.

Aún con la gran ventaja que las redes semánticas tienen con sus relaciones, también estas son su gran debilidad, ya que el hecho de tener tantas relaciones para poder describir más ideas con los conceptos causa que si se quiere integrar una porción de conocimiento de una red semántica a otra, pero ésta otra no cuenta con las relaciones necesarias ó usa relaciones similares pero escritas de diferentes forma, resultaría en una incompatibilidad, por lo que no se pudiera ingresar dicho conocimiento, se generarían relaciones nuevas que no volverían a ser utilizadas debido a que no son soportadas, o no se pudiera ingresar dicho conocimiento debido a que se desconocen dichas relaciones. Esto es un gran problema para aplicar una red semántica ya que no es posible unificar el conocimiento de redes semánticas existentes que no estén estandarizadas al mismo formato. Aún así, para intentar solucionar éste problema de la cantidad de relaciones, han habido intentos por parte de proyectos para estandarizar las relaciones mediante investigaciones psicológicas, cosa que se afondará más adelante en el documento. Inclusive existen proyectos donde las relaciones se crean de acuerdo al contenido que se va agregando. Retomando el problema, ambas bases de conocimientos no son compatibles por lo que claramente se ve que hasta el momento aún no se ha encontrado la forma de crear compatibilidad directa entre diferentes redes que no compartan un estándar en sus relaciones.

Aunque se puede apreciar que es un gran problema la incompatibilidad entre bases de conocimientos con diferentes tipos de relaciones cada una. Pueden existir soluciones como equivalencias de relaciones o reducción de relaciones, que a pesar de integrar conocimiento nuevo, se perderían conceptos y por lo tanto, conocimiento. Se puede notar que se requiere trabajar más en la compatibilidad de las redes semánticas para que éstas sean más aplicables y que puedan sumar los conocimientos para

enriquecerlas aún más.

2.1.2. Marcos

Teniendo en cuenta que las redes semánticas son gránulos de conocimiento regidos por un nodo concepto descrito por sus relaciones de cantidad indeterminada, los marcos son una extensión de las mismas redes semánticas, ya que contienen de forma integrada aparte de un concepto, clases fijas que las describen [5]. Son descritas por un determinado número de relaciones fijas. De ésta forma existe una jerarquía de clases y herencia de los mismos conceptos. Evitando tener una cantidad indeterminada, y variable, de relaciones de descripción, el mismo nodo ya describe las características básicas que hacen a dicho concepto ser lo que es.

En la Figura 2.2 se muestra un ejemplo de como los marcos describen un concepto, en este caso un *ave kiwi*, mediante relaciones y clases integradas a los nodos concepto.

Otra ventaja que tienen los marcos sobre las redes semánticas, debido a las características integradas, es la reducción de tamaño de las bases de conocimiento. Donde una red semántica requiere varios nodos conectantes para describir a un concepto, la naturaleza de los marcos inmediatamente reduce los nodos conectantes a menor cantidad. Ya que se limitan a la mínima información requerida, y fija, para describir algún concepto predefinido. Pero esto no viene sin su contra parte, el hecho de tener una cantidad fija y predeterminada de clases integradas en cada nodo es una gran limitante como modelo de representación de conocimiento, ya que la generalización es inmediatamente perdida, porque para cada tipo de concepto que se quiera representar, se requiere tener de antemano una plantilla que se ajuste a dicho tipo de concepto a ser manejado.

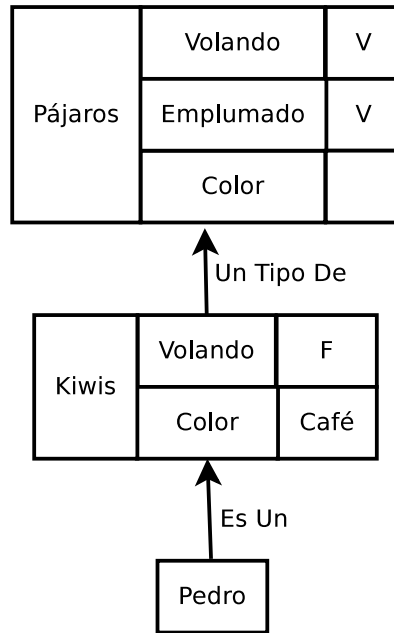


Figura 2.2: Ejemplo simple de marcos describiendo a un ave kiwi.

2.1.3. Reglas de producción difusas

Antes de entrar al tema de reglas de producción difusas hay que mencionar lo que es la lógica difusa [6] y como es que ésta funciona para representar conocimiento. Siendo que la naturaleza de la lógica difusa es la de toma de decisiones, cabe mencionar que para poder tomar una decisión se requiere de un conocimiento previo para poder llegar a una conclusión. Por lo que esta sección será enfocada a como es que las reglas representan conocimiento, y no al hecho de como éstas son utilizadas para la toma de decisiones.

Visualizando la representación de conocimiento utilizando reglas de producción difusas, y sabiendo que éstas siguen el formato *IF..THEN*, se pueden utilizar para hacer una representación del conocimiento un poco más compleja, ya que cada regla trae de forma intrínseca la relación entre conceptos de diferente naturaleza, por ej. IF

fruta es manzana AND hambre es mucha THEN acción es comerManzana, con este ejemplo se puede inferir el conocimiento de que una manzana es una fruta, y si se tiene hambre se puede comer una manzana. Tal como una red semántica o un marco, se puede ver que de igual forma las reglas de producción difusas pueden representar conocimiento. Pero a diferencia de ellas, tienen algo que les da un mayor acercamiento a la realidad, cosa que es muy importante en la representación de conocimiento, y es el hecho de que las reglas difusas se componen por conceptos subjetivos que varían de acuerdo a la percepción. Donde a diferencia de las ya mencionadas redes semánticas y marcos, donde las relaciones y conceptos son de naturaleza absoluta, existe o no existe la relación, con las reglas de producción difusas se tienen relaciones y conceptos totalmente difusos, por lo que su semejanza a la realidad es mucho mayor.

2.1.4. Bases de datos relacionales

Las bases de datos relacionales [7], aunque no propiamente creadas para almacenar conocimiento humano a como las redes semánticas o marcos, suelen usarse para representar aglomeraciones de datos que al relacionarse se transforman en conocimiento. También pueden usarse para representar conocimiento en general, ya que de forma similar a los modelos de representación ya mencionados, contienen tablas de datos para representar algún conocimiento y relaciones de un dato hacia otra tabla de más datos, similar a los marcos.

Se define formalmente por la teoría de conjuntos y todas sus operaciones son basadas en la misma.

En la Figura 2.3 se puede observar que las bases de datos relacionales no son muy adecuadas en la representación de conocimiento. Aunque las bases de datos

relacionales sirvan para representar conocimiento, están muy limitadas en el aspecto que, a como ya fue mencionado, no fueron creadas para ser usadas como representación de conocimiento de computadoras. Por lo que trabajar con ellas, en representación de conocimiento humano, sería muy limitado y por lo tanto no muy recomendado.

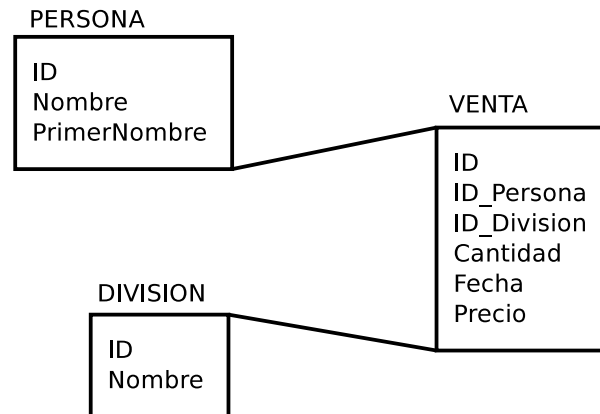


Figura 2.3: Ejemplo simple de base datos relacional que describe las ventas hechas por los vendedores de un establecimiento.

2.2. Categorías

Cuando se quiere describir algo, aparte de dar sus características más básicas, se le asigna un género al que éste pertenece, sea para señalar a un objeto, una acción, algún punto en el tiempo, etc., asignándole dicho género éste inmediatamente obtiene una esencia de lo que es y de lo que no es. Siendo que las categorías sirven para describir la esencia de las cosas, su aplicación en representación de conocimiento es muy grande, ya que puede ayudar a darle aún mayor descripción a lo que un concepto es, y como ya se mencionó, también lo que no es. Cuando se habla que algún concepto,

sea un objeto o una idea, éste debe de tener alguna categoría que describa la esencia de tal concepto, por lo que inmediatamente al tener dicha categoría, éste ya se limita a que tipo de conocimiento puede describir y con cual otro se puede relacionar, aparte que ayuda en eliminar ambigüedades en significados. A continuación se explicarán tres conceptualizaciones diferentes de como las categorías deben ser, siendo que cada uno tiene un ángulo de acercamiento diferente es importante mencionarlos para entender de forma general como es que se pueden aplicar las categorías en la representación de conocimiento.

2.2.1. Aristóteles

Este filósofo de la antigua Grecia pensó que toda cosa, fuera objeto, idea, o concepto debe pertenecer a alguna categoría que represente su esencia [8]. Inicialmente obtenido por la pregunta base "¿Qué es?", usado para comprender todo lo que fuera de importancia y que un humano pudiera percibir. Enfocado principalmente a separar lo que es de lo que no es, por ej. el concepto de "Juan Pérez" es un objeto tangible, existente y perceptible, se le asigna una categoría adecuada que acomode a su género de esencia de objeto, y no se le puede asignar otra categoría que describa una característica o relación a otro, ya que no es más que un objeto y no puede ser otra cosa. Por ésta forma de visualizar de Aristóteles, cada categoría es exclusiva y no pueden pertenecer dos conceptos a una misma categoría, ya que en su grado de género y esencia más atómico, estos no son iguales. En la Tabla 2.1 se describe como Aristóteles percibió que las categorías deberían ser.

Estas categorías están adecuadas, hasta cierto grado, para ser implantadas en un modelo de representación de conocimiento, ya que no contempla tanto la percepción

Tabla 2.1: Lista de categorías de acuerdo a Aristóteles.

Sustancia primera	Describe a una instancia de un objeto. Ej. Manuel, esa silla
Sustancia segunda	Describe al género de una instancia de un objeto. Ej. hombre, sillas
Cantidad	Describe que tanto es un concepto. Ej. grande, 8 metros
Cualidad	Describe el tipo o cualidad. Ej. negro, dulce
Relación	Describe la relación respecto algo. Ej. cerca de, lejos
Lugar	Describe la ubicación del concepto. Ej. mercado, Tijuana
Tiempo	Describe el cuando. Ej. Ayer, 5 minutos
Posición	Describe la posición final después de ejecutar alguna Acción. Ej. acostado, parado
Estado	Describe la esencia de tener o ser. Ej. disparado, estar armado
Acción	Describe el hacer. Ej. disparar, lanzar
Afección	Describe el sufrir cambio. Ej. ser disparado, ser armado

y estado mental de la persona, sino la relación directa de esencias entre conceptos perceptibles y no perceptibles. Aparte que se adecúa fácilmente a los conceptos que típicamente son manejados en los modelos de representación de conocimiento mencionados al principio de este capítulo.

2.2.2. Kant

A diferencia de las categorías propuestas por Aristóteles, Kant intenta generalizarlas para todo objeto y situación posible, pensado y no pensado [9], donde este critica a Aristóteles por no considerar todo, y solo conformarse con 10 categorías. Kant por lo tanto en lugar de mejorar sus categorías, propuso una serie de nuevas categorías en donde éstas iban a trascender el pensamiento, exista o no bajo la experiencia de un hombre, éstas deberían aplicar a toda situación en todo momento.

Las categorías de Kant fueron cuatro de forma principal, y cada una con tres subdivisiones. En la Tabla 2.2 se mencionan como de acuerdo a Kant éstas categorías deberían ser.

Aunque a simple vista dichas categorías son más específicas que las de Aristóteles, a como el humano percibe, sufren la complicación que no es fácil separar las categorías referentes a objetos, ideas y conceptos, por lo que la integración con un sistema de representación de conocimiento sería más complejo. Siendo que esto fue concebido con la idea de asemejar a como el humano percibe las cosas, se requiere de mucha ayuda en especificarle a la computadora cuando utilice que categoría bajo cada concepto representado.

Tabla 2.2: Lista de categorías de acuerdo a Kant.

Causalidad	Describe la existencia o ausencia de algo, con un rol de inherencia y subsistencia
Realidad	Existe algo en la realidad
Negación	Existe la ausencia de algo en la realidad
Limitación	Relación de la presencia y ausencia de dos conceptos mientras uno se refiere a la noción del tiempo y espacio
Cantidad	Describe número o alcance
Unidad	Refiere a lo que actualmente se tiene bajo contexto, el enfoque del momento
Pluralidad	Refiere una cantidad no especificada de unidades
Totalidad	Refiere al total de una agrupación predefinida
Modalidad	Describe la causa
Existencia	Refiere a la existencia de algo caracterizado y especificado en la actualidad, que tiene precisado su tiempo y espacio
Posibilidad	Refiere a algunas condiciones impuestas sobre la modalidad
Necesidad	Refiere a lo que ocurre bajo todas las condiciones
Relación	Describe las proposiciones de categorías, disyuntivas o condicionales
Inherencia y subsistencia	Refiere a la noción de sujeto y predicado, sustancia y accidente
Causalidad y dependencia	Refiere al hecho de causa y efecto
Comunidad	Refiere a la reciprocidad

2.2.3. Peirce

Este personaje propuso un sistema de categorías muy diferente a otros ya existentes, ya que visualiza de una manera muy diferente como la esencia de las cosas deberían de ser. Expone que las categorías no deberían ser propiamente categorías, sino predicados de predicados o, propiamente dicho, predicamentos. Esto es, el sentimiento de la realidad de los objetos como entidades abstractas con múltiples características inherentes o la separación de las mismas, dependiendo del nivel de categoría en que se esté expresando [10].

En comparación con otros sistemas que delimitan con claridad la esencia y naturaleza de cada categoría, Peirce decidió dejar únicamente tres categorías 2.3, afirmando que no se necesitarían más, y dejándolas tan generalizadas que no habrían limitantes de representación y aplicación.

Tabla 2.3: Lista de categorías de acuerdo a Peirce.

Primeridad	Es la idea de aquello que es tal como es sin consideración a ninguna otra cosa
Segundidad	Es aquello que está determinado, terminado, acabado, que es correlativo, objeto, necesitado, reacción
Terceridad	Es el medio, lo que se llega a ser, lo que se desarrolla, lo que se produce

Con las tres categorías de Peirce, se puede ver de forma inmediata que su aplicación a un sistema computacional no puede ser de forma directa, tal como las categorías de Aristóteles, ya que razonan en un nivel de abstracción mucho mayor donde

se requeriría más avances computacionales en el área de inteligencia artificial para que una máquina pudiera propiamente aplicarlos en su conocimiento, o como fue expresado con las categorías de Kant, la intervención humana fuera esencial para su funcionamiento. Por lo que aplicación no se consideró para ésta tesis debido al alto nivel de complejidad requerido para aplicarlo.

2.3. Eventos de estados dinámicos

Entiéndase por Eventos de Estado Dinámicos a una secuencia que puede representar toma de decisiones o flujos. Dichas secuencias son útiles para representar decisiones donde hay múltiples caminos posibles; para representaciones deterministas, un evento específico cambiará de un estado a otro estado mediante dicho evento; y para representaciones no deterministas, un evento específico puede cambiar a uno o más estados diferentes. En los casos de tener estados no deterministas se suele seleccionar al azar el siguiente estado.

Para tomar decisiones, estas expresan secuencias de actividades a realizar ó un flujo de datos a través de dicha secuencia. Para el primer caso, se puede concebir una situación donde se desea conocer el flujo de una actividad a realizar, sea una labor dentro de una empresa o una secuencia de preparación para el despegue un avión; para el segundo caso, la representación de un flujo de paquete de datos a través una red es posible a través de eventos donde los datos son empaquetados, enviados y finalmente recibidos para luego ser desempaquetados. Como se puede ver, los eventos de estados dinámicos tienen múltiples aplicaciones en la representación de secuencias, por lo que para la finalidad de éste documento, representación general de conocimiento, es indispensable la afinidad a poder integrar dichas secuencias dentro del modelo.

2.3.1. Máquinas de estado finito

También conocidos como Autómatas Finitos, éstos son un modelo matemático para realizar de forma automática una salida en base a una o múltiples entradas [11], éste computo se realiza en base a los estados y transiciones ya preestablecidos por un diseñador. El objetivo es obtener una salida satisfactoria de forma automática hasta no cumplir con todos los eventos en la secuencia deseada. Es una manera de obtener un computo específico, repetitivo y esperado.

Estos se definen por una tupla:

$$MEF = \langle Q, \Sigma, q_0, \delta, F \rangle \quad (2.2)$$

donde,

Q , es un conjunto de estados

Σ , es un alfabeto finito

$q_0 \in Q$, es un estado inicial

$\delta : Q \times \Sigma \rightarrow Q$, es una función de transición

$F \subset Q$, es un conjunto de estados finales

Existen dos variaciones importantes de éstas máquinas de estado finito, los deterministas y no deterministas. Funcionando a como ya se explicó en el punto anterior del documento, a excepción de los no deterministas, donde la selección del siguiente estado puede seleccionarse al azar, o dependiendo del modelo utilizado, continuar

de forma paralela trabajando con los estados no deterministas, creando un cómputo paralelo.

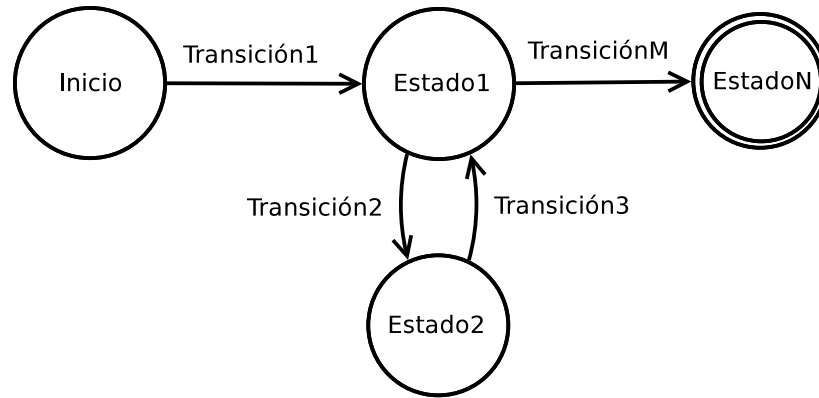


Figura 2.4: Ejemplo genérico de una máquina de estado finito.

En la Figura 2.4 se muestra un ejemplo genérico de un autómata finito donde hay N estados y M transiciones.

A pesar de ser utilizado frecuentemente en trabajos simples como son los revisores de sintaxis en compiladores o revisores de ortografía en documentos de texto, no son muy útiles para representar acciones superiores, tal como la secuencia que una persona realiza para llevar a cabo una actividad determinada, esto es debido a que éstas fueron diseñadas para que las entradas fueran símbolos simples, típicamente un alfabeto, y definir como símbolo a una acción abstracta, en una computadora, es una labor muy complicada.

2.3.2. Redes de petri

Como extensión de las máquinas de estado finito, las redes de petri son una formalización para la descripción de sistemas distribuidos concurrentes [12]. Basado en arcos, lugares y transiciones, donde los arcos conectan un lugar con una transición y viceversa; los lugares son condiciones de la secuencia; y las transiciones son los eventos que pueden ocurrir. Tan simple se pueda ver ésta representación, está basada en una formalización matemática fuerte que opaca a otros modelos de secuencias, ya que se tiene comprobado que siempre va a funcionar a como se espera, por lo que es muy utilizado en áreas de simulación de procesos.

Cabe destacar que existen múltiples definiciones de una red de petri, a continuación se mostrará una de las tantas definiciones.

Este se define por una tupla:

$$PN = \{S, T, W\} \quad (2.3)$$

donde,

S , es un conjunto finito de lugares

T , es un conjunto finito de transiciones

S y T , son disyuntivos $W : (S \times T) \cup (T \times S) \rightarrow \mathbb{N}$, es un multiconjunto de aristas

En la figura 2.5 se puede observar un ejemplo muy simple de como los tokens se transportan a través de cada lugar y como cada lugar puede tener uno o más tokens, dependiendo de como se haya diseñado la red.

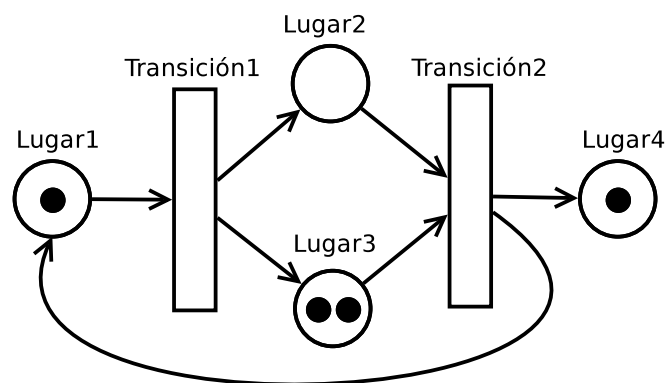


Figura 2.5: Ejemplo simple de la visualización una red de petri.

Existen múltiples variaciones de estas redes, tal como las redes de petri coloreadas, que a diferencia de las redes de petri normales, que únicamente utilizan tokens genéricos, éstos ya pueden utilizar datos más complejos, tal como un alfabeto o conjuntos de ellos. Abriendo un nuevo camino a simulaciones aún más complejas, tal como simulación de flujo de datos de red de un protocolo definido, o simulación de procesos dentro de una planta industrial.

También hay una variación de redes de petri que son difusas, pero éstas solo son una representación gráfica de reglas de producción, y que bajo su defecto, son una forma alterna de resolver sistemas de inferencia difusos y no agregan nada a las redes de petri.

2.4. Lógica difusa

Han existido algoritmos computacionales para la toma de decisiones desde las primeras computadoras, donde un simple falso o verdadero era la única opción de decisión lógica. Sin lugar para ambigüedad, este modelo de toma de decisiones siempre tenía que estar seguro de la respuesta, blanco o negro. Hasta hoy en día, ésta forma de

tomar decisiones es aún utilizada, ya que simplifica considerablemente decisiones binarias para un gran número de situaciones simples. Pero este beneficio de simplicidad, tiene algunas deficiencias que restringen de forma considerable su uso en situaciones diarias, como el decidir que película ver en el cine, o situaciones aún más simples de la vida cotidiana, como el decidir si ir al cine o no. Aunque estas deficiencias pueden considerarse negligentes para un humano, ya que puede realizar dichas decisiones sin pensar mucho en ello, para una computadora no se le es nada fácil, lo que puede ser simple para un humano, puede ser altamente complejo o hasta imposible para una computadora. Por lo que la lógica clásica, de tipo falso y verdadero, falla en éstas situaciones. Como una solución rápida a estos problemas, y que aún se utiliza, es la capacidad de integrar algoritmos heurísticos con comparaciones de lógica clásica, y agregando números en la comparación se pueden conseguir una variedad mayor de resultados posibles, por ej. mayor que(i), menor que(j), igual($=$). Aunque esto puede resolver unos cuantos problemas más, en similitud a la lógica clásica de falso y verdadero, sigue siendo aún muy limitado en que no es posible simular con ello la manera en como un humano toma una decisión.

La lógica difusa fue creada con la idea de que las computadoras pudieran tomar decisiones tal como los humanos [13], no en una manera numérica, o lógica, tal como las computadoras siempre han hecho, sino con palabras, de aquí el término: "Computing With Words" (Computo Con Palabras). Que vendría a reemplazar las limitadas decisiones con lógica clásica y heurísticas numéricas con reglas legibles, utilizando palabras, y con ellos crear una nueva visión de como las computadoras pueden tomar decisiones que se aproximan más a como el humano las tomaría. En lugar de usar reglas IF-THEN con falso y verdadero, se utilizan funciones continuas que represen-

tan gránulos o palabras de un conjunto de reglas dentro del esquema de toma de decisión. Aunque aún sigue utilizando la configuración de reglas mediante el formato IF-THEN, puede integrar múltiples decisiones dentro de una misma reglas, por ej. IF motor es frío AND clima es frío, para crear reglas de decisión aún más complejas. Esto sería un gran avance en la representación de reglas de decisión en las computadoras, ya que solo los humanos eran los únicos con la posibilidad de pensar de tal forma.

Desde la creación de lógica difusa, se le ha aplicado en una gran cantidad de situaciones y problemas, desde las áreas de ingeniería hasta la simulación social. Y con su expansión a lógica difusa tipo 2, se ha logrado representar decisiones que antes no se podían. Todo esto para aplicarla en situaciones especializadas con situaciones comunes donde una computadora puede tomar la decisión en lugar de tener que un humano la realice.

Las reglas de producción difusas siguen el formato IF-THEN, tal como un humano representaría dicha reglas de decisión en su mente, por ej. IF agua es caliente THEN estufa es apagar; esto es una manera muy avanzada de representar reglas en comparación con la lógica de falso y verdadero. Haciendo fácil la programación de reglas, para un diseñador o ingeniero, para resolver los problemas que se tengan, y dando cabida a que se enfoque más en la solución al problema, que a como se debería crear el algoritmo para que tome la decisión.

Capítulo 3

Sistemas de Representación del Conocimiento

3.1. Proyectos

Existen múltiples proyectos de representación de conocimiento en la actualidad, unos buscan comprobar que su modelo de representación es adecuado para usarse en sistemas de producción, otros buscan probar algoritmos de aprendizaje o decodificación semántica de texto, entre otras metas. Algo en común que la mayoría de los proyectos tienen es la creencia de que si logran crear una base de conocimiento lo suficientemente grande, podrán simular al pensamiento humano mediante sus algoritmos propios de inferencia. Dichos proyectos llevan años en el proceso de aumentar su base de conocimiento, sea mediante la ayuda de humanos que le ingresan de forma manual dicho conocimiento, o mediante algoritmos sofisticados de interpretación de texto para luego convertirlo a datos dentro de su base de conocimiento.

A continuación se mostrarán unos ejemplos de proyectos maduros y nuevos, que

pretenden avanzar el área de representación de conocimiento.

3.1.1. OpenCYC

Proyecto concebido en los 80s para crear una base de conocimiento con contenido similar al de un humano para poder ser utilizado en una variedad de aplicaciones donde dicho conocimiento pudiera ser inferenciado [14, 15]. Originalmente conocido como simplemente CYC, se creó el proyecto libre y con acceso al código fuente de forma limitada llamado OpenCYC, que con una cantidad reducida de conocimiento actualmente cuenta con un aproximado de medio millón de términos, quince mil tipos de relaciones y como cinco millones de afirmaciones, fue liberado al público para que este pudiera experimentar con él y aplicarlo, aunque de forma limitada ya que el proyecto principal CYC sigue manteniendo muchas mejoras constantes sobre el OpenCYC, tal como algoritmos más novedosos y especialmente el hecho de que éste último únicamente tiene acceso a aproximadamente una décima de su base de conocimiento.

Se pensó que para poder tener el conocimiento humano en una computadora, éste debía ser ingresado por humanos, por lo que se le ha agregado de forma manual cientos de miles de datos, pero con el tiempo se dieron cuenta que esto no era muy eficiente, por lo que han recurrido a aplicar algoritmos que sustraen el conocimiento de forma automática de textos variados.

Dicho proyecto almacena tanto términos como relaciones, y puede inferir sobre dicho conocimiento. Donde se le hace una simple pregunta y éste contesta en base al conocimiento que ya tiene. Por ej. si se tiene el conocimiento que *un árbol es un ser vivo*, y que *todo ser vivo eventualmente muere*; preguntándosele que *si*

un árbol muere?, éste respondería mediante su sistema de inferencia que *un árbol eventualmente muere*. Mediante dicho ejemplo, se puede notar inmediatamente que los términos y relaciones son simples.

Aunque las relaciones son del estilo red semántica o incluso marcos, éste no utiliza ninguna de estas, sino un lenguaje especializado en la representación de conocimiento llamado CycL, que describe en forma textual toda relación y regla dentro de la misma base de conocimiento. Un ejemplo de como se le ingresa conocimiento a esta base de conocimiento es mediante la siguiente forma

```
(#$isa #$Manzana #$Fruta)
```

que viene representando la relación simple: Una Manzana es una Fruta.

Como base de conocimiento, se le ha enfocado más a la aplicabilidad especializada que a la aplicación en inteligencia artificial, por lo que su actual actual giran alrededor de búsquedas de información, e integración de bases de datos heterogéneas.

3.1.2. ConceptNet

Proyecto derivado de la iniciativa OpenMind, comenzado a finales de los 90s, enfocado en mantener una base de conocimiento basada en sentido común utilizando redes semánticas [16]. Diseñado para almacenar el conocimiento tipo *sentido común* mediante la ayuda de humanos en el llenado de su base de conocimiento [17]. El conocimiento se le da vía web mediante la inserción de afirmaciones basadas en relaciones específicas en campos definidos, para facilitar la compatibilidad de datos ingresados. Actualmente cuenta con más de un millón de hechos. ConceptNet, a diferencia de OpenCYC, utiliza una red semántica en lugar de un lenguaje formal para representar y almacenar su conocimiento, por lo que las relaciones entre conceptos

está limitado a un poco mas de 20 relaciones en la actualidad, dichas relaciones han sido estudiadas y con ellas se puede describir casi toda relación posible entre conceptos del tipo sentido común. Aunque fue diseñado para aumentar este su conocimiento mediante la agregación manual por humanos, también cuenta con algoritmos que interpretan texto a su base de conocimiento.

Fue concebido con la idea de poder almacenar el conocimiento de una persona, para que la base de conocimiento pudiera ser utilizada en aplicaciones donde suele requerirse la interacción de un humano, pero con dicha base de conocimiento y algoritmos de inferencia, una computadora pudiera reemplazar al humano, por ej. se ha utilizado en generadores de cuentos para niños, donde utiliza el conocimiento almacenado para recrear una historia ficticia y congruente. A diferencia de OpenCYC, ellos dan acceso total a su base de conocimiento para que la gente pueda ingresar y consultar conocimiento. Por lo que es más fácil poder integrar algún proyecto con dicha base de conocimiento.

También tiene la limitante, similar a OpenCYC, en que sus relaciones son de primer grado y solo pueden representar afirmaciones muy simples.

3.1.3. YAGO

El proyecto YAGO [18], que es de reciente creación, a diferencia de los proyectos ya mencionados fue creado para obtener su conocimiento en base al contenido del sitio web Wikipedia.org [19], ya que éste tiene categorizaciones y relaciones ya de forma integral. Se aprovechó eso para poder acceder su base de datos y llenar YAGO su propia base de conocimiento, con ayuda de otros proyectos como WordNet [20] y OWL [21], para poder darle forma al vasto conocimiento y procesarlo.

Su forma de representar la información es algo similar a los marcos, ya que cada concepto central tiene la obligación de relacionarse con un mínimo de conceptos, y así estandarizar su conocimiento adquirido. Cuenta con 14 relaciones fijas que son usadas como si fueran las relaciones de una red semántica. Aunque se le puede agregar conocimiento de forma manual, este fue diseñado para adquirirlo de forma automática; que claramente lo diferencia de otros proyectos donde los datos son ingresados manualmente por humanos. Esto ha hecho que cuente con aproximadamente diez millones de entidades y ochenta millones de afirmaciones sobre dichas entidades.

Como todo proyecto ontológico, éste fue creado para reunir información, pero enfocada a la información colaborativa de Wikipedia.org, por lo que a pesar de no requerir interacción humana, requiere de algoritmos inteligentes que puedan interpretar la información del sitio web. Debido a ello se limita a lo que pueda encontrar en dicho sitio web. Pero aún así, rebasa por mucho a otros proyectos existentes.

Este se aplica en conjunto con otros proyectos ontológicos para complementar el conocimiento, ya que cada proyecto se enfoca en cierto tipo de conocimiento, de esta forma se puede abarcar aún más conocimiento que a diferencia no se pudiera si las bases de conocimientos se tuvieran por separado.

3.2. Ventajas y desventajas de los proyectos de representación de conocimiento existentes

Siendo que los proyectos mencionados son de naturaleza diferente, ya que fueron diseñados para cumplir funciones específicas diferentes, a continuación se compararán sus ventajas y desventajas desde un punto de vista de representación de conocimiento.

3.2.1. Ventajas

3.2.1.1. OpenCYC

Como fue de las primeras ontologías computacionales, ha evolucionado considerablemente sus herramientas de entendimiento de lenguaje natural. Tiene su propio lenguaje interpretativo para manipular sus datos e inferenciar sobre su base de conocimiento (CycL). Puede inferenciar de forma lógica, mediante modus ponens, modus tollens, cuantificadores universales y cuantificadores existenciales.

3.2.1.2. ConceptNet

Una de las ventajas que tiene ConceptNet respecto otros proyectos ontológicos es que éste ha invertido mucha investigación en seleccionar y delimitar sus relaciones, asegurando que puede representar casi todo conocimiento de tipo sentido común con dichas selectas relaciones, esto siendo un primer paso hacia la estandarización de relaciones que afecta mucho a las redes semánticas en general. Siendo que su base de conocimiento ha sido llenada casi exclusivamente de forma manual por humanos, su conocimiento es más apegado a la realidad de como un humano piensa a diferencia que si hubiera sido extraído en su totalidad por un algoritmo.

3.2.1.3. YAGO

Obtiene su conocimiento de forma automática dándole la ventaja de poder expandir su base de conocimiento de forma considerable y rápida, causando esto que sea el proyecto que tenga más afirmaciones en comparación con los demás. A diferencia de otros proyectos ontológicos, este ya cuenta con algún tipo de valor estadístico en

sus relaciones que les da un grado de certidumbre a la relación en cuestión. Se auxilia de varios proyectos para poder corregir información ambigua y estilos de redacción para la extracción de conocimiento.

3.2.2. Desventajas

3.2.2.1. OpenCYC

Siendo que CYC es un sistema cerrado y únicamente se tiene acceso a OpenCYC, éste es incompleto comparado con la cantidad de datos que el proyecto original mantiene, y carece de mucha documentación. Solo maneja valores absolutos en sus relaciones y conceptos, por lo que se tiene de conocimiento es un conocimiento definitivo e inmutable. A pesar de ser uno de los proyectos ontológicos más viejos, no es muy utilizado por las razones antes mencionadas. Visualizándola como una red semántica, en lugar de su lenguaje de representación CycL, una de sus debilidades principales es el tener miles de tipos de relaciones, por lo que de forma obvia no es compatible con otras bases de conocimientos existentes.

3.2.2.2. ConceptNet

La gran desventaja de ConceptNet es el valor absoluto de sus relaciones, al igual que OpenCYC, una vez que un dato sea almacenado en su base de conocimiento, este se vuelve definitivo e inmutable. Aunque se pueda considerar como bueno el hecho su base de conocimiento sea llenado por humanos, como no se tiene control sobre quien ingresa y que ingresa, puede haber conocimiento erróneo, incongruente, confuso, ambiguo y contradictorio.

3.2.2.3. YAGO

Obtiene todo su conocimiento únicamente del sitio web Wikipedia.org, limitándolo a lo que dicho sitio contenga. Solo se enfoca a como se relacionan ciertos datos en su base de conocimiento, no es tanto una base de conocimiento de sentido común, sino un tipo de enciclopedia que relaciona datos, tal como fechas, lugares, etc.

Algo que los tres proyectos mencionados tienen en común como desventaja es el hecho de que todos manejan relaciones de primer grado, debido a la limitante de sus modelos de representación. Por lo que sus algoritmos de inferencia siempre estarán limitados a contestar preguntas simples o relacionar datos granulares.

Capítulo 4

Diseño y Análisis del Modelo de Representación de Conocimiento General

Tras haber mostrado varios modelos de representación de conocimiento, varios sistemas de categorías, lógica difusa, y varios proyectos avanzados sobre representación de conocimiento. Se hará un hincapié sobre la razón de selección de lo utilizado en el modelo híbrido propuesto.

El modelo propuesto es híbrido por naturaleza, debido a que se tomaron varios métodos, modelos y representaciones diferentes para unificarlos de tal forma que se pudiera hacer uso de estos y con ello mejorar los modelos existentes de conocimiento, e intentar llegar a un modelo de representación de conocimiento general, donde no importe el dato, éste pueda ser representado con una misma notación. A como se ha expresado en capítulos anteriores, lo existente hasta el momento hace bien

su función para lo que fue diseñado, pero bajo limitantes de lo que puede abarcar. Para evitar dichas limitantes, se decidió hibridar varios métodos en uno, y obtener un modelo final donde tenga las ventajas de sus contrapartes, pero en conjunto evitar las limitantes, y consecuentemente, obtener el modelo híbrido que se propone.

4.1. Base de representación

La parte central del modelo propuesto es la base de su representación, en este caso, se decidió utilizar como base una red semántica, a pesar de existir otros modelos de representación más recientes este fue seleccionado debido a su simplicidad, y es esa simplicidad que le da la capacidad de expandir e integrar su funcionalidad. Aunque se habló sobre la limitante de incompatibilidad entre redes semánticas debido a que carecen de un estándar que defina y delimite las relaciones existentes, se decidió apoyar con las relaciones de ConceptNet, ya que estas han sido estudiadas previamente para poder abarcar casi toda posibilidad de relación de conocimiento.

Cabe hacer destacar que a diferencia de otros proyectos de representación de conocimiento, donde estos son considerados para usarse específicamente en áreas de aplicación en la industria; el modelo propuesto ha sido intencionado a ser una base de conocimiento orientada al aprendizaje incremental de conocimiento de una inteligencia artificial ó agente, aún así, esto no lo ha limitado a no ser utilizado en aplicaciones específicas como otros proyectos existentes. Esto se menciona para que se tenga presente la finalidad principal, el aprendizaje incremental, y su funcionalidad como aplicación general durante la lectura del resto del documento.

4.2. Valor de certidumbre

Tras haber seleccionado las redes semánticas como base del modelo propuesto, y como las relaciones serán apoyadas en lo propuesto por ConceptNet, nos enfocamos en la siguiente debilidad de las redes semánticas, los valores absolutos de las relaciones. Las redes semánticas consideran que toda relación existente entre dos conceptos es de naturaleza absoluta e inmutable, esto es, que es el equivalente a estar totalmente seguro y no tener dudas que esa relación es la absoluta verdad, y a como se sabe, en la realidad los humanos estamos llenos de duda respecto todo nuestro conocimiento, sea que se esté casi seguro de algo o se tenga mucha duda sobre algo, las redes semánticas no pueden representar esto mediante sus relaciones, ya que no tienen la capacidad de integrar dicho valor de certidumbre. En el modelo propuesto se considera esta limitante, y para poder evitarla se decidió integrar lógica difusa para medir el valor de certidumbre. Aunque típicamente la lógica difusa se relaciona con el término *incertidumbre*, para definir que tanto se desconoce algo, en este caso es lo inverso, *certidumbre* para poder definir que tanto se conoce algo. Aunque YAGO maneja algo similar, pero este viene siendo un valor probabilístico, y siendo que la lógica difusa tiene una formalidad más acercada a representar la forma de pensar del humano, por esto mismo es que no se usó un valor probabilístico, ya que no compete al pensamiento natural del humano.

El valor de certidumbre de las relaciones, siendo un valor difuso, se maneja con un valor continuo que va de 0 a 1, esto se puede interpretar como lo siguiente: si el valor tiende a 1, se está seguro de la existencia de la relación entre los dos conceptos en cuestión; si el valor tiende a 0, se tiene mucha duda sobre la existencia de dicha relación

y no se puede confiar en dicha afirmación. Como el modelo propuesto fue originalmente pensado para ser utilizado en conjunto con una inteligencia artificial para que ésta fuera aprendiendo gradualmente y de forma incremental, por lo que suponiendo el caso cuando el sistema sea expuesto por primera vez a una afirmación, o sea, la relación entre dos conceptos, este puede tener un valor de certidumbre neutro de 0.5, y con el tiempo se puede ir reforzando el valor, incrementándolo mediante constante exposición a dicha afirmación, hasta que éste tienda a 1 y ya no se tenga duda de la afirmación; ir olvidando la afirmación, mediante el decremento gradual del valor mientras ese no sea expuesto a él de vuelta en mucho tiempo, hasta este tender a 0; o modificar el valor mediante el conocimiento nuevo de afirmaciones contradictorias, donde el sistema entra en conflicto sobre su información; el caso que sea, las posibilidades de manipular el valor de certidumbre son vastas dependiendo el grado de representación humana que se quiera manejar. En la Figura 4.1 se muestra un ejemplo de la modificación del valor de certidumbre, supongamos una computadora que se le ha dicho que un tomate es una verdura, y luego se le expone información contradictoria de que un tomate no es verdura, sino una fruta, esto es considerando que las características que hacen a una verdura son muy similares al de una fruta. El conocimiento entra inmediatamente en conflicto, por lo que al inicio se relacionó que un tomate es una verdura, con un valor de certidumbre de 0.5, por ser la primera vez que éste era expuesto a dicha afirmación, y luego al informársele que un tomate no es una verdura, sino que un tomate es una fruta, este genera una nueva relación con valor de certidumbre de 0.5 y se reduce un poco la relación antigua de tomate es una verdura, por ej. a 0.48. Con el tiempo la computadora será expuesto a cualquiera de estas dos afirmaciones, reduciendo y aumentando gradualmente el valor de certidumbre de una relación u otra, hasta el

punto donde una de ambas tienda a 1, y la computadora pueda luego tomar la decisión que la otra relación es falsa y por lo tanto eliminarla, esto considerando un algoritmo que tome dicha decisión.

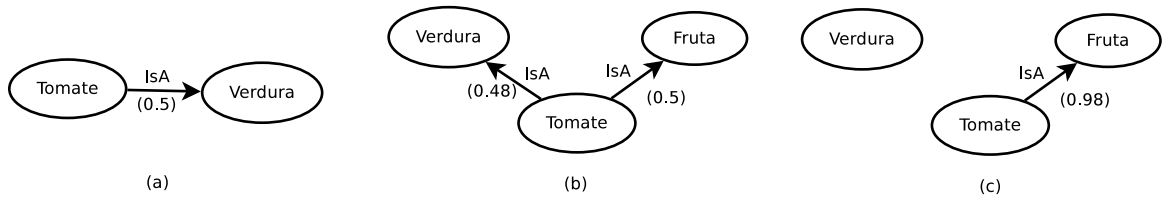


Figura 4.1: Ejemplo de uso de lógica difusa en las relaciones. (a) Se acaba de exponer a la afirmación que un tomate es una verdura. (b) Posterior exposición a la afirmación que un tomate es una fruta. (c) Tras constante exposición a la afirmación correcta de que un tomate es una fruta, se fortalece la relación tomate-fruta, y se debilita la relación tomate-verdura hasta desaparecer esta última.

Para poder aumentar la representatividad y posibilidad de uso de las relaciones en la toma de decisiones, se decidió agregar, aparte de tener un valor de certidumbre singleton, funciones de membresía non-singleton, esto para apoyarse en la representatividad de conocimiento de la lógica difusa. Una relación que tenga un valor de certidumbre puede distinguirse de otro mediante un grado de percepción diferente a otras relaciones iguales pero con características diferentes, esto es, que relaciones iguales conecten con conceptos diferentes pero de naturaleza similar. Aunque por el momento solo se considera el uso de funciones de membresía en las relaciones para la aplicación de toma de decisiones, que más adelante serán explicadas, no se descarta que estas no tendrán un uso futuro para otras aplicaciones o representaciones de conocimiento no contemplado hasta el momento de redacción de este documento.

Los valores de certidumbre y funciones de membrecía no solo sirven para apoyar a la representación de las relaciones, sino también para auxiliar en la representación de los conceptos mismos. Tal como funcionan estos valores para darle más representatividad de certidumbre al grado de conocimiento de una relación, de igual forma se le puede asignar estos valores con la misma función a los conceptos. Retomando el ejemplo de la computadora que aprende que un tomate es una verdura, o fruta, aunque las relaciones se estén modificando cada vez que este sea expuesto a dichas afirmaciones que contengan el concepto de tomate, la relevancia de la existencia del concepto tomate se vuelve más común con cada exposición y por lo tanto cada vez el valor de certidumbre irá aumentando lentamente. Pero suponiendo que nunca se volviera a retomar la afirmación tomate, sino hasta una instancia en el tiempo muy lejana, y este ya aprendió sobre la existencia de otras frutas o verduras, siendo que este no tuvo casi exposición al concepto de tomate, este debería ir bajando lentamente el valor de certidumbre, y por tanto bajando la relevancia que este tiene para la computadora en cuestión, ya que se le puede considerar que se va olvidando ya que no tiene exposición constante al concepto. Pero eso no implica que el valor eventualmente tendiendo a cero haga que se elimine la relación, simplemente sería otro parámetro que considerar, cosa que no es considerado en este documento. Solo se menciona dicha situación para demostrar la importancia que tiene los valores de certidumbre.

La función de membrecía que cada concepto puede tener sirve para representar la percepción que se tiene respecto dicho concepto, esto es, tal como la lógica difusa asigna una función a cada concepto de decisión, el modelo propuesto hace lo mismo pero se le asigna a la representación granular de cada concepto.

4.3. Categorías

Ya discutido la hibridación de una red semántica con lógica difusa para darle más representatividad y percepción, ahora se discutirá la agregada hibridación con categorías. El sistema de categorías seleccionado fue el de Aristóteles, debido a que la visión de dichas categorías que explican la esencia de las cosas se adapta fácilmente a la representatividad de una red semántica. Siendo que la naturaleza de las redes semánticas es explicar mediante conceptos y relaciones, las categorías se integran en los conceptos para complementar dicho concepto con una explicación más profunda y abstracta de lo que dicho concepto es sin la necesidad de tener que observar sus relaciones. Para comprender este concepto de hibridación con categorías, se usará un ejemplo de una afirmación simple de una red semántica, donde se afirma que una manzana es una fruta, por lo que si se sabe que una fruta es un objeto físico tangible, por herencia de característica, inmediatamente se sabe que una manzana por lo tanto es un objeto tangible. Pero se tuvo que inferir dicha porción de red semántica para poder inferenciarlo, pero si al concepto de manzana se le integra la categoría que representa a los objetos físicos tangibles, que viene siendo la categoría de *substance1st*, ésta inmediatamente no requiere inferenciar sobre la red semántica para poder llegar a la conclusión que la manzana es un objeto físico tangible. Para apreciar más detalle y granularidad de esencia descriptiva con ayuda de categorías, podemos ver ejemplificado en la Figura 4.2 al concepto de fruta, una fruta pertenecería a la categoría de *substance2nd*, ya que es una representación abstracta de un objeto físico tangible, es la suma de la idea de las instancias de los objetos existentes. Por lo que no es necesario tener que inferir con más relaciones para poder entender la esencia de dicho

concepto. Inmediatamente ya se tiene que esencia tiene dicho concepto, no solo por las conexiones de concepto a concepto, sino por la simplicidad de ver que categoría tiene. Ahora supongamos que se le agrega descripción a la manzana, y se dice que tiene la propiedad de ser, o poder ser, roja. De igual forma a como con el concepto de fruta, si se le asigna la categoría correspondiente a características o cualidades al concepto de rojo, que vendría siendo la categoría *quality*, inmediatamente se puede deducir sin necesidad de inferenciar el resto de la red semántica que el concepto rojo es una cualidad y por lo tanto debe estar describiendo algún objeto físico y tangible, aunque no se sepa cual sea aún, o que el rojo es una cualidad, y por tanto no puede ser un objeto, acción u alguna otra esencia, descartando y delimitando de forma inmediata las opciones de inferencia con dicho concepto.

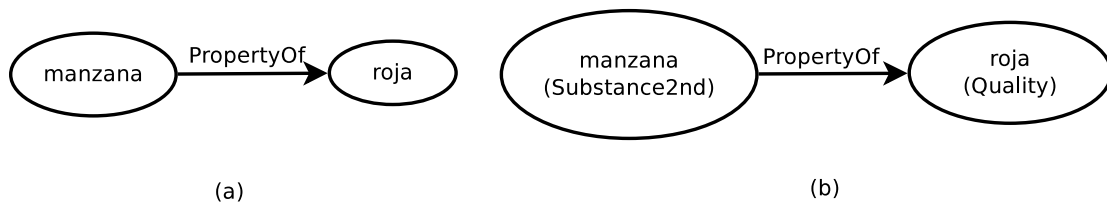


Figura 4.2: Ejemplo de uso de categorías en los conceptos. (a) Representación típica de una red semántica de la afirmación: una manzana tiene la propiedad de ser roja. (b) Representación de la misma afirmación, pero con las categorías adecuadas, agregando mayor significado a cada nodo concepto.

Otra función muy importante de integrar categorías en los conceptos de la red semántica, es el evitar ambigüedades en el conocimiento, esto es, cuando una misma palabra puede tener varios significados, las categorías pueden hacer la diferencia en

que un algoritmo se pueda fallar al inferenciar o de un buen resultado. Por ejemplo a como se muestra en la Figura 4.3, la palabra *bote*, que puede significar el objeto barco que sirve como medio de transporte naval, o la acción de rebotar un objeto sobre alguna superficie. Asignando la categoría de *substance2nd* al que describe el navío, y *action1st* al que describe la acción de rebotar. La ambigüedad inmediatamente desaparece. De esta forma el conocimiento queda aún más organizado y especificado, a que si se tuviera que depender en totalidad con las descripciones ontológicas de las redes semánticas originales.

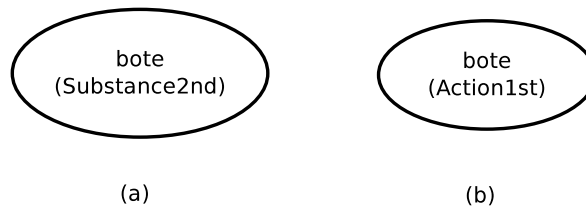


Figura 4.3: Ejemplo de desambiguación con el uso de categorías en los conceptos. (a) Concepto de bote referente a un objeto físico, en este caso, un navío. (b) Concepto de bote referente a la acción de rebotar un objeto sobre una superficie.

Considerando que una red semántica por si misma es de una naturaleza muy granular, con la hibridación de categorías, estas se vuelven aún más granulares, ya que pueden no depender de sus conceptos colindantes para poder describirlos en su totalidad o parcialidad.

4.4. Modelo propuesto

El modelo híbrido propuesto se resume en la unificación de redes semánticas, como base de conocimiento; lógica difusa, para obtener un acercamiento más afín a la representación mental que los humanos tienen de las ideas y conceptos; y categorías, para poder granular aún más el conocimiento. Por lo que se llegó a la siguiente formalidad que representa al modelo propuesto:

El modelo principal es una tupla

$$GKR_graph = \langle V, A \rangle \quad (4.1)$$

donde,

V , es una colección de conceptos difusos, y A , es una colección de relaciones difusas dirigidas. Describiendo a V y A con más detalle, tenemos una triple

$$V = \langle Con, Cat, Cer \rangle \quad (4.2)$$

donde,

Con , es el concepto mismo y cada $Concept \in Universo\ de\ Posible\ Conceptos$, Cat es una categoría que representa la esencia del concepto, y Cer es un valor lingüístico que representa lo difuso, o certidumbre, del concepto; y una tupla

$$A = \langle Cer, Rel \rangle \quad (4.3)$$

donde,

Cer , es un valor lingüístico que representa que tan cierta es la relación entre dos conceptos, y donde Rel es la relación de dos conceptos que interconectan. Dado esto,

para ambos A y V , $Cer| \rightarrow [0, 1]$. También definimos

$$Cat = \{Substance1st, Substance2nd, Quantity, Quality, Relation, \\ Place, Time, Position, State, Action, Action1st, Action2nd, \\ Affection\}$$

y

$$Rel = \{ConceptuallyRelatedTo, IsA, PropertyOf, PartOf, \\ MadeOf, DefinedAs, CapableOf, FirstSubeventOf, \\ PrerequisiteEventOf, SubeventOf, UsedFor, \\ SubeventOf, LastSubeventOf, EffectOf, \\ DesirousEffectOf, LocationOf, MotivationOf, \\ MotivationOf, CapableOfReceivingAction, DesireOf, \\ Or, And, Not, Dilation, Concentration, Minus, \\ Plus, True, False, LooksLike, SoundsLike\} \quad (4.4)$$

4.5. Eventos de estados dinámicos

Como se propone crear un modelo que pueda representar información de naturaleza heterogénea, esto es, que no solo represente sentido común, a como la mayoría de las ontologías con redes semánticas, sino representar también secuencias de acciones, situaciones de decisión, etc. Ahora se mostrará como, con la notación ya propuesta, se puede representar una sucesión de acciones donde pequeñas decisiones llevan

el curso del flujo. Esto puede servir tanto para representar secuencias de acciones comunes en la vida diaria del humano, tal como la secuencia que toma desde que uno se levanta por la mañana, va al trabajo, regresa y se vuelve a dormir; representar secuencias de flujo de labores de una empresa, o cualquier otra secuencias de acciones y decisiones donde se requiera simular la forma en como un humano contempla un flujo a seguir y posibles situaciones dentro de ello.

Como en este documento ya se han examinado secuencias de estados dinámicos, ahora se creará algo similar utilizando la notación propuesta para poder llevarla a cabo. Esto se logra utilizando ciertas relaciones que se adecuan a representar secuencias de forma abstracta y también de forma concreta. Esto es, representar mediante conceptos un nodo de evento; sea abstracto el nodo, tal como el concepto de comer; o el concepto concreto de las acciones que conlleva la acción de comer. Esto sin recurrir a alteraciones en la notación propuesta. Los conceptos básicos para poder representar casi cualquier secuencia de eventos son las siguientes categorías: *action1st*, que representa la abstracción de una acción que se puede realizar, por ej. trabajar; *action2nd*, que representa un granulo de acción que puede llevarse a cabo, por ej. cerrar la mano; *state*, que representa una abstracción genérica que sucede inmediatamente después de haberse realizado un evento; y *affection*, que representa el esperar que algún tipo de entrada sea recibida, por ej. recibir orden. Con esta limitada cantidad de categorías se puede representar cualquier secuencia, donde el concepto puede ser cualquier evento posible. Esto para representar el significado de cada evento en la secuencia, pero si se quiere conocer la secuencia, se recurre a un grupo de relaciones seleccionados, de igual forma que las categorías, puedan representar fielmente al modelo propuesto el significado de las relaciones. Las relaciones, por motivos de abstracción, tienen una

jerarquía que les impone un grado de importancia. Las relaciones fueron delimitadas a las siguientes, con el orden de más importante a ejecutarse a menos importante: *ConceptuallyRelatedTo*, que viene siendo un relación conceptual paralelo al nodo actual, sirve para definir ideas o conceptos paralelos similares en idea; *DefinedAs*, que define a un evento abstracto mediante el desglose granular con más conceptos; *PrerequisiteEventOf*, este siendo el de menor prioridad pero no de menor importancia, ya que define el camino de la secuencia a seguir; y se tiene *True* y *False* para poder hacer decisiones simples que solo requieren de una comparación. Con estas relaciones elementales se pueden seguir secuencias en base a los grados de prioridad, pero en el caso de que existan dos relaciones iguales saliendo de un mismo nodo, como el de la Figura 4.4, en el caso de ser no determinista, se resuelve la prioridad mediante el uso de los valores de certidumbre integrados en las relaciones, nodos, y sus funciones de membresía en caso de ser utilizadas, así eliminando el uso de heurísticas y utilizando algoritmos para toma de decisión más apegado a la percepción humana, y por lo tanto una forma más real de seguir la secuencia en dichas situaciones. Hay que hacer notar que si se consideran estas secuencias como una herramienta de producción, y no un sistema base para una inteligencia artificial, las secuencias pueden ser modificadas sin recurrir a parar la secuencia, ya que solo se ejecuta un nodo a la vez, por lo que queda libre el resto de la red a poder ser alterada y así evitar contratiempos en la ejecución del sistema.

4.6. Toma de decisiones

La toma de decisiones es muy importante, por lo que poder representarla es clave para un sistema de representación general. Aunque se pueden representar decisiones

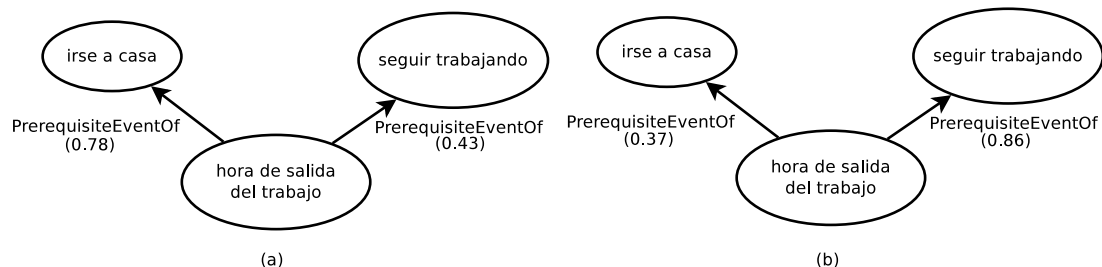


Figura 4.4: Ejemplo de uso del valor de certidumbre en la decisiones no deterministas.

(a) Caso donde la persona en cuestión tiene mayor preferencias a irse a su casa en lugar de continuar trabajando. (b) Caso donde la persona en cuestión tiene fuertes tendencias a continuar trabajando tras acabar su jornada de trabajo.

simples en secuencias, como la ya mencionada anteriormente, para poder representar decisiones complejas como las que un humano realizaría se requiere una variación de representación, esto es, representar reglas de producción difusas con la notación propuesta. De esta forma aseguramos que la toma de decisiones tiene un fundamento fuerte. Esta inferencia es una conversión directa de una regla de producción escrita hacia un modo gráfico utilizando la notación propuesta. Para mejor entender la representatividad se explicará en torno de las entradas, reglas y finalmente de las salidas, así como su forma de ser calculadas para llegar a un valor de respuesta.

4.6.1. Entradas

Las entradas en una regla de producción difusa típicamente son una abstracción de la representación del problema, donde pueden representar un conjunto de sensores, un conjunto de datos secuenciales o cualquier cosa que pueda ser utilizada como entrada para una regla de toma de decisión. Como puede verse en la Figura 4.5, la representación propuesta utiliza un nodo por cada entrada de los antecedentes del

sistema de inferencia difuso, o FIS por sus siglas en inglés (Fuzzy Inference System). Para el caso de esta notación, siempre se utilizará la categoría *quality*, ya que describe la naturaleza del objeto, donde el objeto viene siendo la entrada misma.

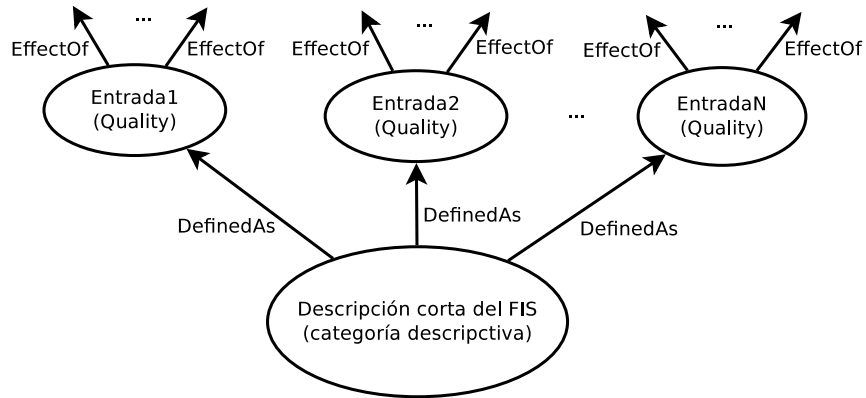


Figura 4.5: Generalización de la representación de las entradas a la toma de decisiones basado en reglas de producción difusas escritas.

Es importante dar una mayor descripción al primer nodo de la figura 4.5, porque representa la descripción corta del FIS, ya que para cada sistema difuso siempre cambiará. Tal como el nombre dice, es una corta descripción del FIS, por ej. nivel de agua de tanque, control de frenos de carro, etc. Ya que un punto fuerte de utilizar el modelo propuesto es la visualización del conocimiento, las reglas y como interactúan unas con las otras impulsa una descripción mejorada del problema, y con este nodo que por si solo no realiza ningún cómputo, se logra un grado de descripción mayor que da un entendimiento fácil al sistema de toma de decisión eliminando la necesidad que adivinar cual es la función de esta porción de la red de conocimiento recurriendo a la interpretación de las representadas reglas.

El nodo de descripción del FIS no realiza ningún cómputo, pero los nodos de

entradas sí, solo aceptan un valor numérico real como entrada que posteriormente transfiere a los nodos colindantes mediante los conectores, que conectan a las reglas.

4.6.2. Reglas

Parte de la fuerza o ventaja que tiene la lógica difusa en la habilidad de combinar reglas y crear decisiones complejas, tal como los humanos lo hacen en su mente. En la Figura 4.6, se muestra una regla generalizada donde cada nodo representa una función de membrecía específica. En este caso, las siguientes reglas escritas reflejarían a los nodos mostrados:

IF entrada1 is MF11 AND entrada is MF1N THEN ...

IF entrada1 is MF21 AND entrada is MF2N THEN ...

...

IF entrada1 is MFN1 AND entradaN is MFNN THEN ...

$MFXY$ es una función de membrecía, X es el número de regla, y Y es el número de entrada. Esto muestra una descripción generalizada, pero tal como los FIS son diseñados, los nombres pueden ser lo que se requiere para el problema en específico.

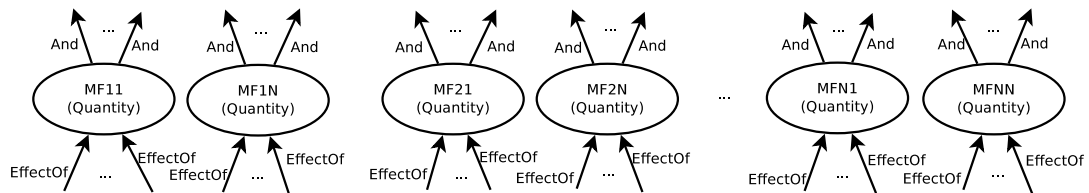


Figura 4.6: Generalización de la representación de las reglas para la toma de decisiones basado en reglas de producción difusas escritas.

Continuando con el cómputo realizado en los nodos de entrada, que solo se limitan a transmitir el valor de entrada recibido a través de las flechas, esta capa de nodos realiza más cómputo. Siendo que ésta representación es una conversión directa de un FIS hacia un modo gráfico, las fuerzas de disparo son calculadas en esta capa. La siguiente fórmula muestra el cómputo generalizado realizado en esta capa:

$$O_{i,j}^1 = \mu_{i,j}(x_k) \quad (4.5)$$

4.6.3. Salidas

La sección de salidas está compuesto por dos capas, donde se realiza el resto del cómputo. Como se muestra en la Figura 4.7, los conectores *And* vienen de la capa anterior de reglas para entrar en las debidas salidas, los cuales vendrían representando la siguiente parte de las reglas de producción escritas siguientes:

```
... THEN Salida11 is MF1 AND Salida12 is MF2 AND...AND Salida1N is MFN
... THEN Salida21 is MF1 AND Salida22 is MF2 AND...AND Salida2N is MFN
...
... THEN SalidaN1 is MF1 AND SalidaN2 is MF2 AND...AND SalidaN is MFN
```

La $MFXY$ es una función de membrecía, X es el número de regla, y Y es el número de salida. Esto muestra una descripción generalizada, pero tal como los FIS son diseñados, los nombres pueden ser lo que se adecue al problema.

Estas capas son más complejas computacionalmente, comparado con las capas anteriores, entradas y reglas, ya que la mayoría del cómputo es realizado aquí. La primer capa, las salidas, computan ambos fuerzas de disparo y s-norma para cada salida por cada regla. Descrito por la siguiente fórmula:

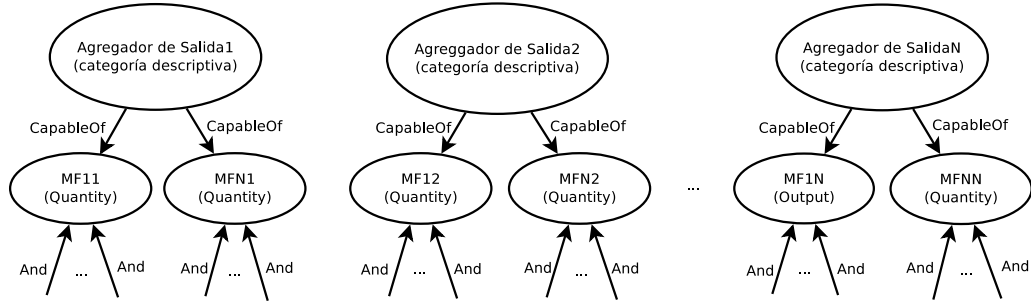


Figura 4.7: Generalización de la representación de las salidas para la toma de decisiones basado en reglas de producción difusas escritas.

$$O_{k,l}^2 = \min(O_{1,i,j}) \quad (4.6)$$

La segunda capa computa la agregación de todas las salidas individuales por cada regla y luego defusifica. Esto se muestra en la siguiente fórmula:

$$O_k^3 = \frac{\sum_p \mu_G(x_p)x_p}{\sum_p \mu_G(x_p)} \quad (4.7)$$

Debe notarse que hasta ahora, las flechas conectantes han seguido la misma dirección, por razones de visualización y representatividad, la última capa de nodos invierte la dirección de las flechas debido a la naturaleza de la cada relación *CapableOf*, que expresa las capacidades de la salida de forma abstracta.

4.7. Ejemplo heterogéneo del modelo de representación de conocimiento general

Siendo uno de los objetivos del modelo de representación de conocimiento general es la habilidad de poder tener conocimiento de naturaleza heterogénea, y ya habien-

do mostrado ejemplos de conocimiento tal como descripción de objetos, secuencias de eventos, y tomas de decisiones, debido a que se utilizó la misma notación para todos los casos, la unificación del conocimiento es transparente, ya que son compatibles entre sí. Por tanto se logra un acercamiento a tener una base de conocimiento heterogénea que pueda funcionar como base de una inteligencia artificial, un agente, una simulación, o un control de proceso. Se puede decir que la factibilidad del modelo en ser implementado en múltiples casos es alta, ya que con la notación propuesta, se puede adaptar fácilmente a casos no pensados donde la interpretación del conocimiento es cercano a como un humano pensaría la aplicación. En la Figura 4.8 se muestra un ejemplo de esto donde se puede ver la integración de conocimientos de naturaleza diferente en una misma base de conocimiento.

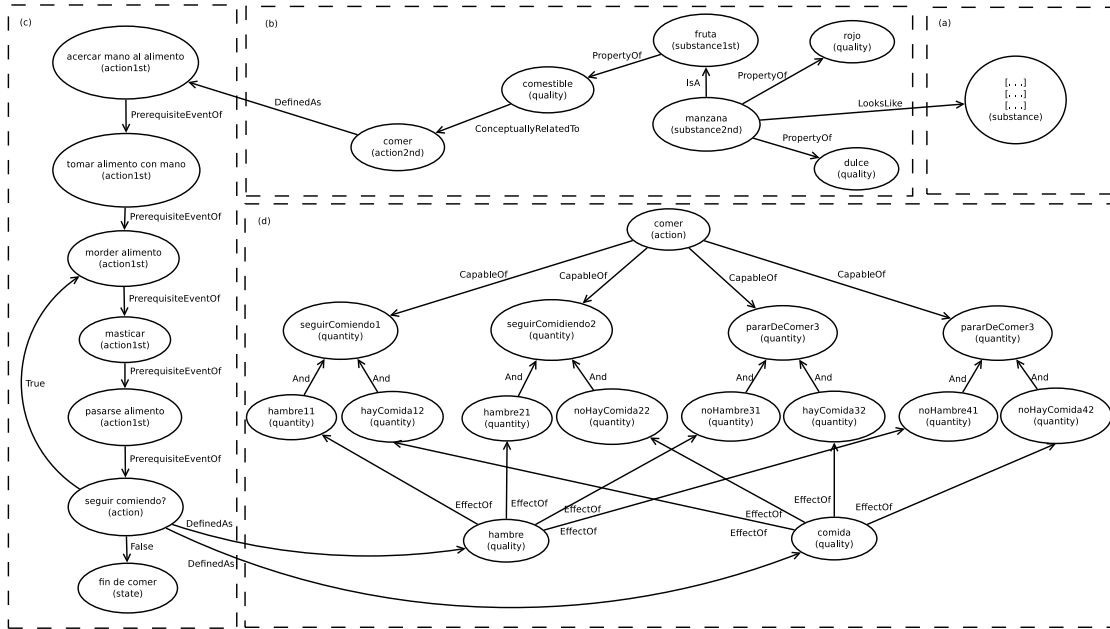


Figura 4.8: Ejemplo de representación de conocimiento heterogéneo, donde se representa a una manzana, como comerla y cuando decidir si seguir comiendo o no. (a) Representación gráfica de una manzana en conocimiento computable por un algoritmo de reconocimiento de patrones. (b) Representación de características generales de una manzana y fruta. (c) Secuencia de acciones generales para la acción de comer. (d) Representación de la decisión para saber si seguir comiendo o no.

Capítulo 5

Implementación y Aplicaciones del Modelo Híbrido de Representación General de Conocimiento

5.1. Implementación

El modelo propuesto fue implementado en el lenguaje C, debido a su velocidad de ejecución y su control sobre todo aspecto de la implementación. Habiendo múltiples formas de representar los nodos de una red semántica, se optó por el camino de las listas ligadas, ya que con el manejo de apuntadores la ejecución es rápida, y pensando a futuro donde se esté manejando una base de conocimientos de tamaño considerable, esta es de las formas más rápidas de recorrer dicha red. A continuación se muestra la estructura principal manejada, donde se puede observar de forma clara la forma en como se maneja la aparte híbrida del modelo propuesto:

```
struct node
```

```

{
    /* CONCEPT */
    char* concept;
    int    category;
    float certainty;

    /* FUZZY DATA */
    float gaussSigma;
    float gaussCenter;
    float rangeA;
    float rangeB;

    /* RELATIONS */
    struct node    *relationIn  [maxR];
    struct node    *relationOut [maxR];
    int    typeIn  [maxR];
    int    typeOut [maxR];

    /* RELATIONS CERTAINTY VALUES */
    int certaintyIn  [maxR];
    int certaintyOut [maxR];

    /* RELATIONS GAUSSIAN MEMBERSHIP FUNCTIONS */
    float gaussInSigma  [maxR];

```

```

float gaussInCenter [maxR]
float gaussOutSigma [maxR];
float gaussOutCenter [maxR];
};

```

La estructura de datos contiene los datos del concepto mismo, esto es, el concepto como tal, la categoría perteneciente al concepto, el valor de certidumbre, y los datos mínimos requeridos para representar una función de membrecía del concepto, se decidió que fuera fijo a gaussianas por facilidad de implementación, posteriormente se generalizará para poder contener todo tipo de función de membrecía. Contiene los datos de las relaciones, conoce hacia que nodos se dirige y de que nodos este se dirige, para que cada nodo siempre sepa que tiene conectado a él, entre estos datos incluye el tipo de relación particular entre cada conexión y sus funciones de membrecía. Finalmente también se tiene almacenado los valores de las funciones de membrecía gaussianas pertenecientes a las relaciones entrantes y salientes, de igual forma con las funciones de membrecía del concepto, estas se generalizarán en futuras versiones de la implementación.

Para generar un conocimiento dado nuevo, se utilizan dos funciones, una para agregar nodos de conceptos nuevos y otra para agregar las relaciones entre dos conceptos dados, a continuación se muestran estas funciones:

```

void addConcept(char* concept,
               int category,
               float certainty;
               float gaussSigma,

```

```

        float gaussCenter,
        float rangeA,
        float rangeB    );

void addRelation(char* from,
                char* to,
                int   type,
                float certainty;
                float gaussSigma,
                float gaussCenter    );

```

Aunque las funciones deben recibir todos los datos requeridos, estos pueden ser omitidos enviando un dato *NULL*, para el caso que no se desee o no se requiera agregar funciones de membrecía.

Esto solo refleja el primer intento de aplicación del modelo propuesto, por lo que en futuras versiones se irá integrando mejoras para abarcar más casos y situaciones, pero manteniendo la simplicidad para facilitar su uso.

5.2. Ejemplo de un robot trabajador

Al final del capítulo anterior se mostró un ejemplo de la heterogeneidad del modelo en cuestión de representación. Ahora se dará un ejemplo pero más enfocado hacia el control de un robot con labores específicas que realizar. Esta simulación será un proceso manufacturero donde el robot trabajador recibe una orden y luego combinando varios materiales en una máquina, llega al producto final.

La secuencia principal es el de un sistema productor, donde una orden es esperada, luego recibida, y finalmente producida, luego se vuelve a ciclar otra vez, tal como se muestra en la Figura 5.1. El concepto *producir* tiene una mayor abstracción, debido a que una serie de pasos son realizados para lograr la producción. La abstracción de *producir* se simplifica en cuatro estados: *ir por material*, *usar máquina*, *almacenar material*, y verificar si la orden se cumplió (*orden cumplida?*). Esto por si solo hace nada, ya que es solo una abstracción, la relación *DefinedAs* toma el concepto abstracto e inicia su descripción con más detalle. Aunque mayor detalle es obtenido, sigue siendo una abstracción, debido a que las acciones granulares son han sido descritas aún.

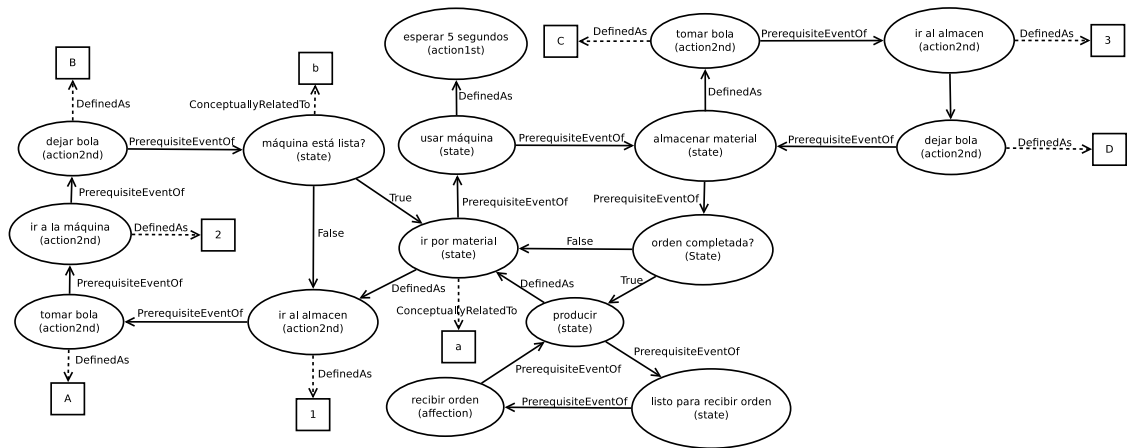


Figura 5.1: Secuencia principal de estados y acciones abstractas del robot trabajador.

Adentrándose a mayor detalle, los conectores cuadrados en la red que se comunican a través líneas punteadas son secuencias granulares generales que pueden llamarse a ejecución de forma temporal desde cualquier concepto, comunicado a través de los conectores *A*, *B*, *C* y *D*. En la Figura 5.2, la granular y concreta secuencia de tomar

una bola y dejarla puede ser vista.

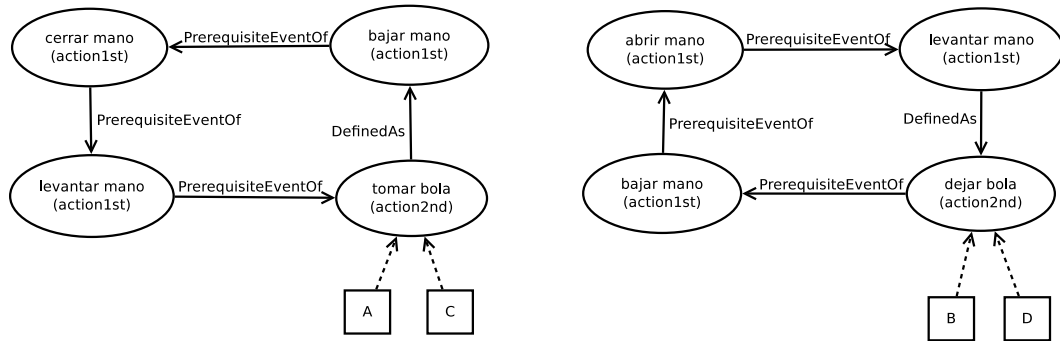


Figura 5.2: Secuencias de acciones granulares para tomar y dejar una bola.

Hasta ahora se han visto acciones y estados, la parte heterogénea de la representación se ve en la Figura 5.3, donde la comunicación a través de los conectores *a* y *b* son una abstracción de la descripción de un objeto que le informa al robot trabajador en que orden debe utilizar las bolas de colores para crear el material requerido, esto se hace mediante la creación de dos reglas de producción difusas que afirman cuando el material requerido ha sido adquirido:

IF entrada1 is rojo AND entrada2 is verde THEN salida is blanco

IF entrada1 is blanco AND entrada2 is azul THEN salida is negro

También, se muestra una breve descripción de lo que el material usado es, en este caso, una bola.

La última parte de este ejemplo es la representación de decisión del robot trabajador, mostrada en la Figura 5.4, para saber a donde ir y como llegar. Los humanos consideran muy fácil el hecho como llegar de algún lugar a otro, y luego está el hecho de que caminamos, que por si solo es una secuencia compleja de movimientos

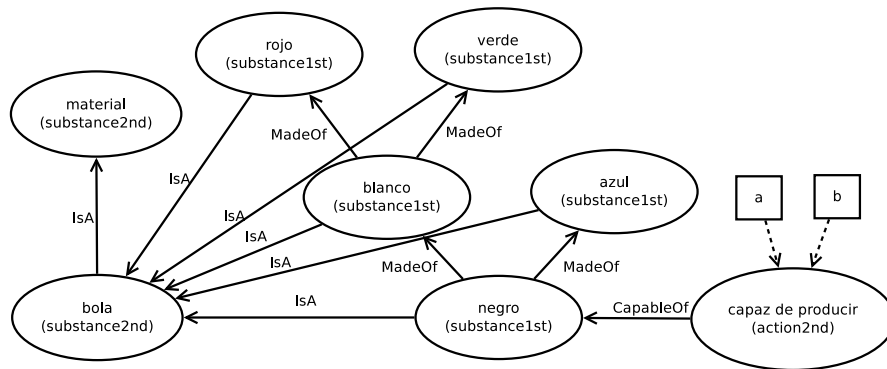


Figura 5.3: Representación descriptiva del material utilizado y la secuencia de maquinado.

controlados por músculos para lograr el balanceo y avance hacia adelante. Esta complejidad está representada aquí por reglas de producción difusas en la forma gráfica con la notación propuesta, que son comunicados a la red de conocimiento mediante los conectores 1, 2, 3 y 4. Estas reglas difusas tiene el mismo efecto que una regla difusa normal, por lo que el resultado sería el mismo.

Como puede ser visto, las acciones y estados pueden ser fácilmente interpretados por un humano, donde el control, las secuencias y decisiones son programadas/planeadas por una base de conocimiento gráfica con todas las posibilidades dentro de dicho dominio, que viene siendo más fácil de diseñar e implementar a diferencia de un lenguaje clásico de programación. Esta forma de inferenciar *el plan*, donde el siguiente paso de la secuencia es decidido de forma inmediata dependiendo del estado actual, da una ventaja como una herramienta de producción ya que cualquier nodo puede ser agregado, eliminado, o modificado sin afectar la ejecución de la misma, así eliminando el requerir interrumpir la ejecución y/o tener que reiniciar todo una vez hechas las modificaciones. Tal como un humano puede que tenga cambios inesperados en el flujo de

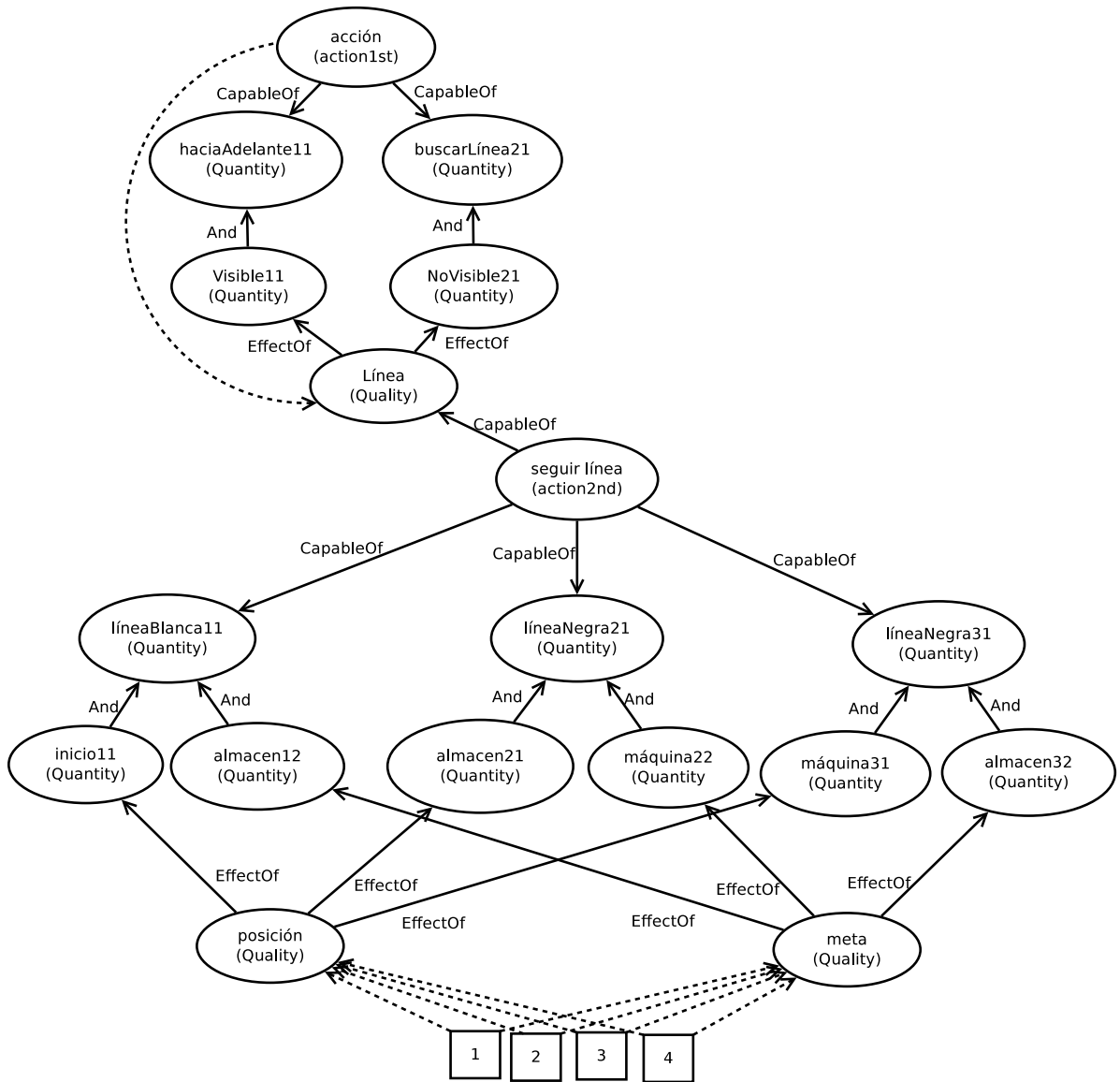


Figura 5.4: Representación descriptiva del material utilizado y la secuencia de maquinado.

su trabajo, este no para o inicia sus labores desde el principio. Así que depurar fallas y optimizar procesos es más fácil que el utilizar lenguajes de programación normales.

5.3. Aplicaciones

Mientras que en el capítulo anterior se mostró el desarrollo del modelo, también se dieron ejemplos simples del uso. A continuación se darán aún más ejemplos y algunos unos algoritmos simples de inferencia, para así demostrar el rango de aplicación del modelo.

5.3.1. Sentido común

Ya habiendo explicado que por sentido común debe entenderse la descripción de todo objeto mediante relaciones de conceptos, y habiendo dado poco ejemplos en el capítulo anterior de esto, se dará ahora un ejemplo más pero de inferencia sobre el conocimiento de sentido común, esto es, una forma de integrar conocimiento nuevo mediante el algoritmo de *modus ponens*, y concluir que es dicho conocimiento nuevo.

La inferencia para poder integrar nuevo conocimiento y concluir que es, se basó en la argumentación lógica de modus ponens:

$$P \rightarrow Q, P \vdash Q \quad (5.1)$$

Consecuentemente se utilizó el algoritmo de *matching* para llegar a una conclusión. El siguiente algoritmo muestra la forma de llevar a cabo la forma de concluir:

- 1: Si el conocimiento es conocido en el contexto
- 2: Temporalmente heredar las relaciones del concepto colindante por

la relación directa IsA

- 3: Unificar el nuevo conocimiento hacia el conocido
- 4: Actualizar valores de certidumbre
- 5: Eliminar las relaciones temporales que fueron heredadas
- 6: Si el conocimiento nuevo no es conocido en el contexto
- 7: Solo hacer matching contra otros de la misma categoría
- 8: Temporalmente heredar las relaciones del concepto colindante por la relación directa IsA
- 9: Hacer el matching contra las relaciones de los conceptos (considerando el valor de certidumbre)
- 10: Eliminar las relaciones temporales que fueron heredadas
- 11: Hacer matching. Para conceptos: Matches > 3, certidumbre > 1.5. Unificar
- 12: Si se unifica
- 13: Agregar relación IsA del concepto principal nuevo hacia concepto principal conocido
- 14: Agregar nuevo conocimiento
- 15: Actualizar valores de certidumbre

Esta inferencia tiene doble utilidad, ingresar nuevo conocimiento y concluir que es, en caso de tener suficientes datos y conocimiento previo; y reforzamiento de conocimiento actual, esto es, hacer más fuerte las relaciones y conceptos mediante el aumento del valor de certidumbre del conocimiento al que se vuelve a exponer.

En la Figura 5.5 se muestra un ejemplo simple de como se infiere conocimiento

nuevo hacia algo conocido y luego se concluye que es. Se puede observar que luego de inferir, quedan relaciones en ambigüedad, estas luego serán inferidas por otro algoritmo para que elimine dicha ambigüedad; también se puede notar que no se generó una relación de verde a color, ya que el conocimiento tal como esta no permitió dicha conclusión, esto debido a que el ejemplo se centró en la inferencia de concluir si era fruta o no, en realidad, sabiendo que verde es una cualidad, este se sabría que es color, por lo que no quedaría sin conectarse, aparte que se reforzaría el valor de certidumbre del concepto de color.

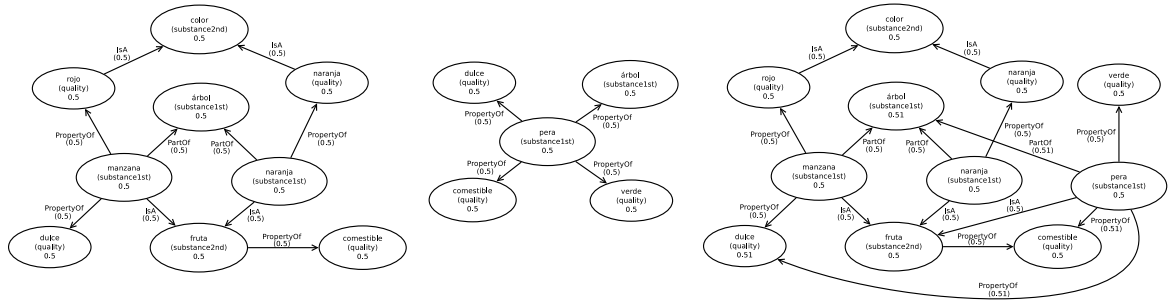


Figura 5.5: Ejemplo simple de como se infiere el conocimiento de una fruta, sin saber propiamente que es una fruta, hacia el conocimiento propio de lo que son las frutas, conocidas hasta el momento. (a) Conocimiento actual sobre frutas. (b) Conocimiento nuevo que desea inferirse hacia el conocimiento propio. (c) Resultado luego de la inferencia y se concluye que el conocimiento nuevo en realidad en una fruta también.

5.3.2. Secuencias de estados de eventos dinámicos

Las secuencias de eventos de estados dinámicos son una implementación con dos finalidades, la ya mencionada posibilidad de representar secuencias de conocimiento,

por ej. como comer; y la aplicación en sistemas de producción, por ej. programación de un robot, simulación de un proceso, etc. Como ya se han dado ejemplos de representar secuencias de conocimiento, y de la programación de un robot trabajador, ahora se dará un ejemplo de como se simula un proceso.

El proceso a ejemplificar es un modelo abstracto de un sistema *productor* – *consumidor* [22], y se mostrará la facilidad con la cual se puede crear tal modelo, aparte que se mostrará como se tiene más representatividad gráfica de la misma.

En la Figura 5.6 se puede observar que el modelo del sistema productor-consumidor es autoexplicable, ya que no solo los conceptos describen al modelo, sino que el agregado de las categorías da aún más detalle a cada concepto, así dando una idea más clara de lo que el modelo intenta abstraer y mostrar al lector.

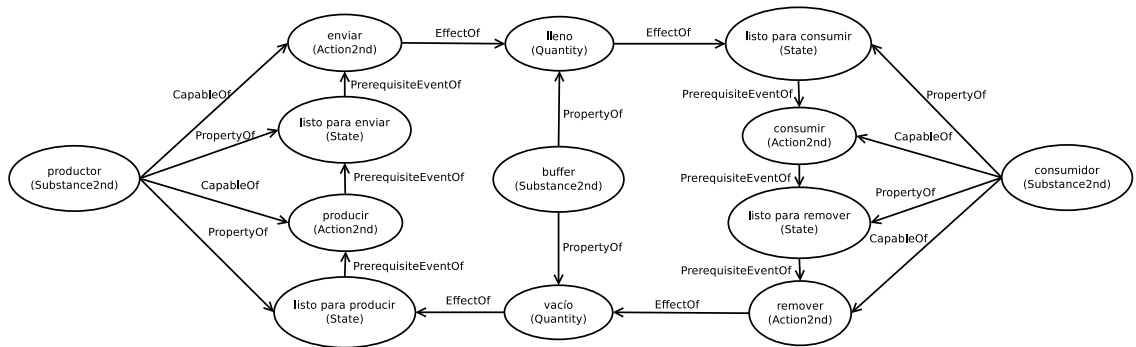


Figura 5.6: Ejemplo de representación de un modelo utilizando la notación propuesta.

El modelo en cuestión es el de un sistema de productor-consumidor.

5.3.3. Toma de decisiones

Habiendo dado múltiples ejemplos de como se aplica la toma de decisiones con el modelo propuesto, se dará un último ejemplo simple para demostrar su eficacia en

representar decisiones difusas.

El ejemplo en cuestión es de lógica XOR, como se muestra en la Tabla 5.1, representado por el modelo propuesto por la Figura 5.7. Aquí se puede ver de manera directa como las reglas de producción escritas son reflejadas en la representación gráfica de las mismas utilizando la notación propuesta.

Tabla 5.1: Tabla de verdad de lógica XOR.

V	V	V
V	F	F
F	V	F
F	F	V

5.3.4. Otras aplicaciones

La aplicación principal de este modelo, por el cual fue creado, fue el tener una base de conocimientos que pudiera soportar unificar datos e información de naturaleza heterogénea. También ya se ha mostrado como se puede aplicar en sistemas de producción donde sirve para programar las secuencias de una máquina o robot para llevar a cabo ciertas labores. Inclusive se ha mostrado como se pueden representar modelos de procesos abstractos. Ahora se comentarán sobre otro lugar y situación donde se podría aplicar el modelo de representación propuesto.

Supongamos una aplicación donde el software se adapta al usuario en base a su personalidad, perfil y previas preferencias, por adaptación entiéndase en cambios sobre la interfaz de usuario, cuando el usuario ingresa los datos de su personalidad en

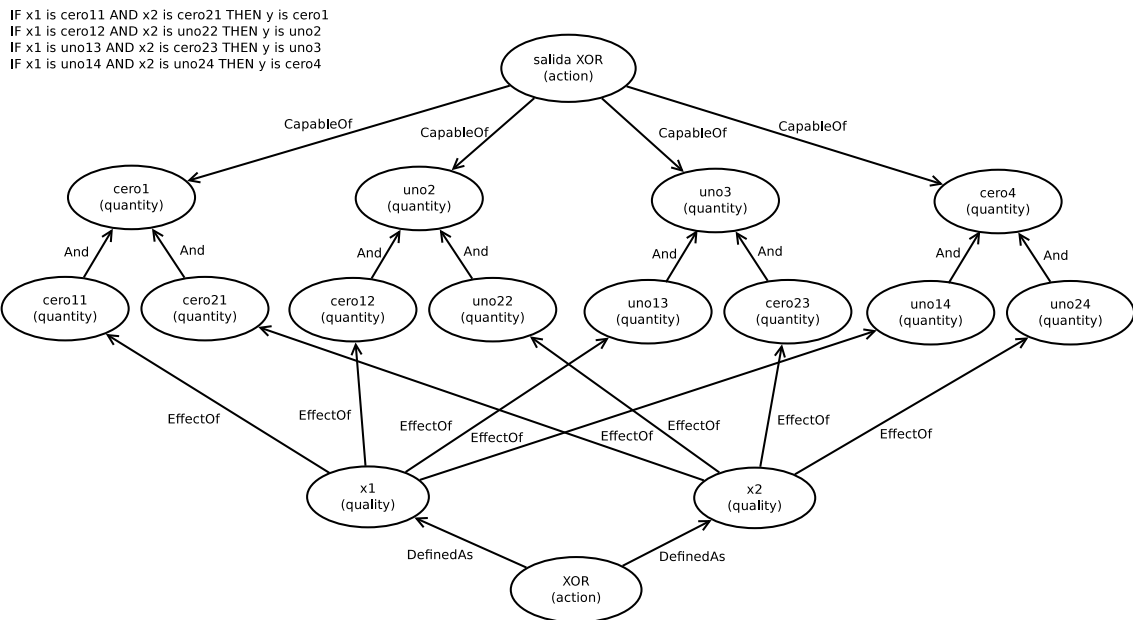


Figura 5.7: Ejemplo de representación de un FIS, lógica XOR, utilizando la notación propuesta.

forma de la red propuesta, este inmediatamente se infiere en una base de conocimiento como la ya mostrada, y toma la decisión de adaptar la interfaz de acuerdo al usuario. De igual forma como se modifican los valores de certidumbre, si el usuario cambia de parecer en gustos estos se van modificando, para ir considerando la evolución de gustos del usuario. En la Figura 5.8 se muestra un ejemplo mínimo de como esto podría llevarse a cabo a nivel base de conocimientos, donde el perfil se infiere hacia el conocimiento del generador de interfaz dinámico y luego se infiere como debería ser la interfaz, para posteriormente adaptarla al perfil del usuario.

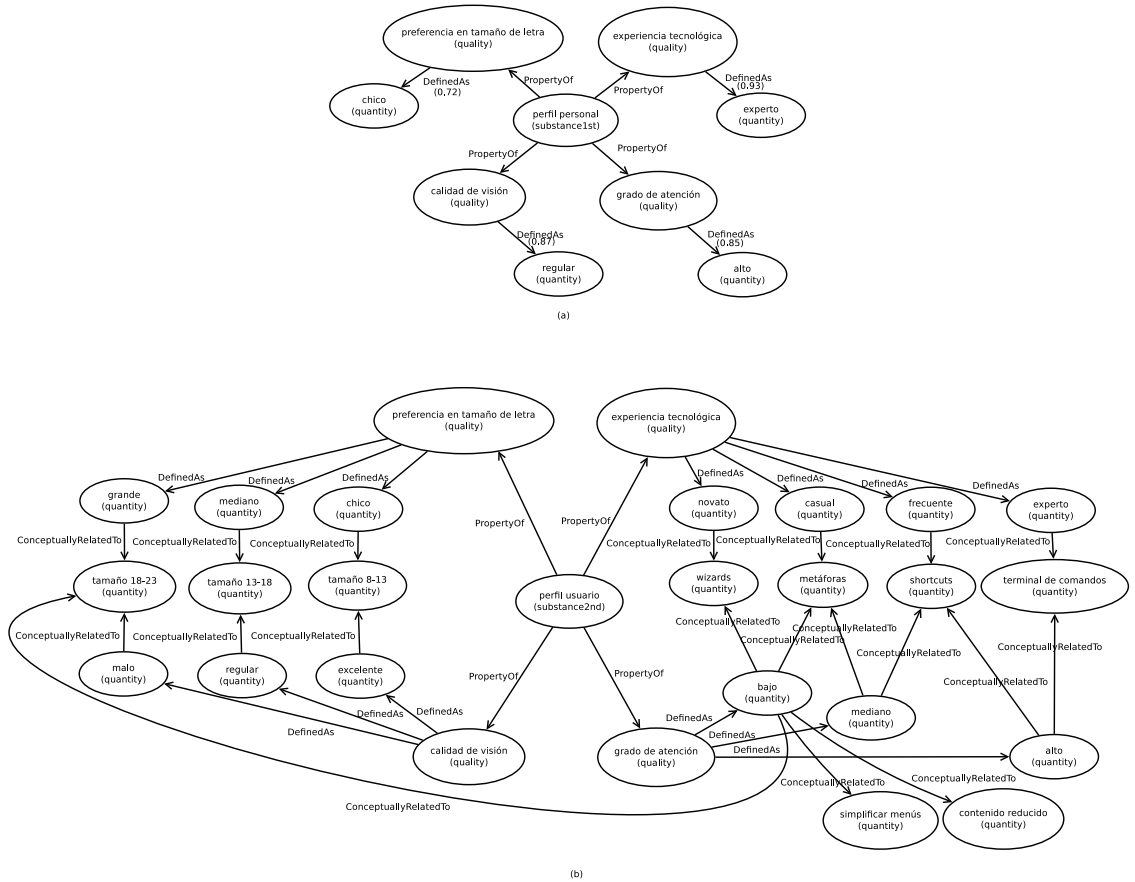


Figura 5.8: Ejemplo hipotético de posible uso en una interfaz de usuario adaptable de forma dinámica al perfil del usuario. (a) Perfil de usuario representado con la notación propuesta. (b) Base de conocimientos para inferir que tipo de interfaz adaptarle al usuario. Se utilizarían funciones de membrecía en los gránulos para poder decidir de forma difusa las preferencias.

Capítulo 6

Análisis de Resultados

6.1. Comparativa de representación de sentido común con redes semánticas

Considerando las limitantes ya mencionadas de las redes semánticas, ahora se mostrará una comparativa de equivalencias entre una descripción semántica de reptiles [23] utilizando la notación propuesta y una red semántica.

En la Figura 6.1 se muestra la comparativa de equivalencias, la primera sección muestra la forma típica de utilizar una red semántica para describir ciertos conceptos simples referente a reptiles. En la segunda parte de la figura se pueden observar las pequeñas, pero notables, diferencias. Que son la modificación de ciertas relaciones, para apegarlos a las relaciones propuestas. Analizando dichas diferencias, queda claro que los conceptos *cocodrilo*, *reptil*, *víbora*, y *víbora de pasto* son *substance2nd* debido a que son instancias generalizadas de su género; donde *pedrito* y *martita* son *substance1st* porque son instancias de su especie; y el resto de los conceptos son del tipo de categoría *quality* ya que cualidades que describen a sus respectivos conceptos

sustancias. Se puede observar también que se modificaron ciertas relaciones, tal como *size*, *color* y *eats*, para apegarse a la notación, pero sin perder el significado, aunque si cabe hacer señalar que el concepto *verde* quedó un poco ambiguo, requiriendo conectar este a otro concepto llamado *color* para lograr el mismo significado y saber que es un color dicha cualidad mencionada en el concepto de *verde*, y el concepto de *carne*, tuvo que modificarse a *carnívoro* debido a que no hay una conversión semejante a la relación de *eats* en la notación propuesta, aunque se solucionó la incompatibilidad con el cambio mencionado del concepto.

La representación de sentido común utilizando la notación propuesta adquiere mayor detalle con la hibridación de categorías, brindando un mayor grado de abstracción descriptiva de cada concepto.

A pesar que no se consideró el uso de valores de certidumbre en este ejemplo, se optó por generalizar el conocimiento para facilidad de explicación. Si se deseara agregar dichos valores, sería dependiendo de que conocimiento se haya sido expuesto y con que frecuencia, para así validar dichos valores de certidumbre.

6.2. Comparativa de representación de conocimiento e inferencia difusa

Para confirmar la verdadera equivalencia entre reglas de producción difusas y la manera de representar toma de decisiones con la notación propuesta, se ejemplificará un FIS ya visto con anterioridad, lógica XOR, solo que ahora con valores reales.

Las reglas utilizadas son las siguientes:

```
IF x1 is cero11 AND x2 is cero21 THEN y is cero1
```

```
IF x1 is cero12 AND x2 is uno22 THEN y is uno2
```

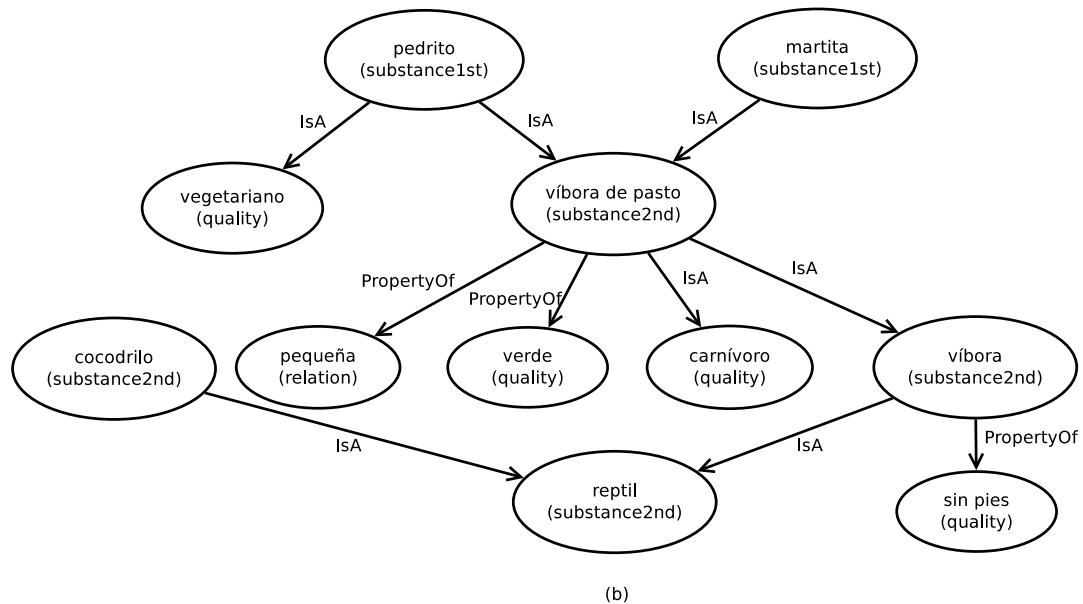
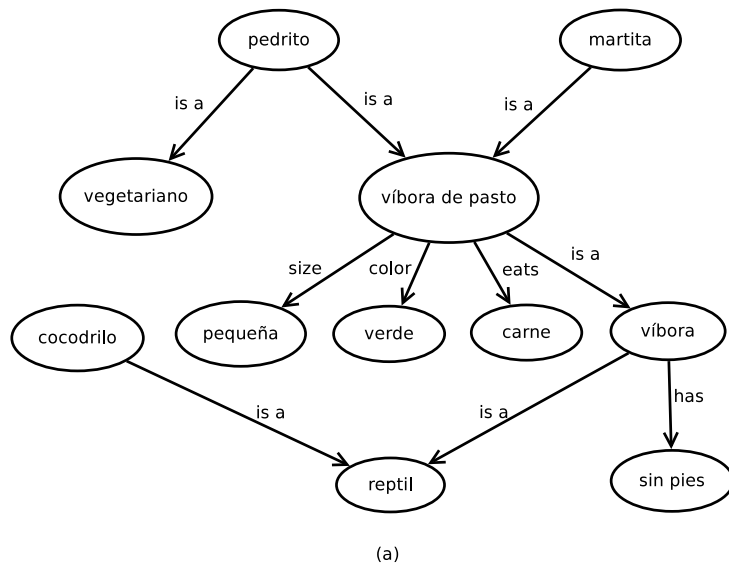


Figura 6.1: Comparativa de conocimiento de tipo sentido común utilizando una red semántica típica y el modelo de representación propuesto. (a) Representación con red semántica. (b) Representación con el modelo propuesto.

```
IF x1 is uno12 AND x2 is cero23 THEN y is uno3
IF x1 is uno14 AND x2 is uno24 THEN y is cero4
```

Donde se puede observar claramente la lógica del XOR representada en dicha reglas. Solo hay que considerar la simplicidad de las funciones de membrecía, aquí se optó por utilizar gaussianas:

```
cero = gaussiana(0.0,0.1)
uno = gaussiana(1.0,0.1)
```

donde,

gaussiana(centro,desviación)

A como se muestra en la Figura 6.2, las funciones de membrecía de las reglas son iguales a las mencionadas (*cero* y *uno*). Para el caso que se deseara inferir los valores: $x1 = 0$ y $x2 = 1$, se lograría un valor defusificado $y = 0.923$, y para el caso de inferir: $x1 = 1$ y $x2 = 1$, se obtendría $y = 0.0767$. Aunque no es propiamente los valores absolutos de 1 y 0 respectivamente, ya que tienden a su valores absolutos, son aceptables considerando la naturaleza difusa del sistema.

Esta representación es una equivalencia fiel a la forma típica de resolver los sistemas difusos, solo cambia la representación visual, debido a ellos no hay variaciones en los resultados.

6.3. Comparativa de representación y ejecución de secuencias con redes de petri

Las redes de petri son una abstracción matemática para representar modelos concurrentes de sistemas, tal como estos funcionan para lograr su objetivo, se puede

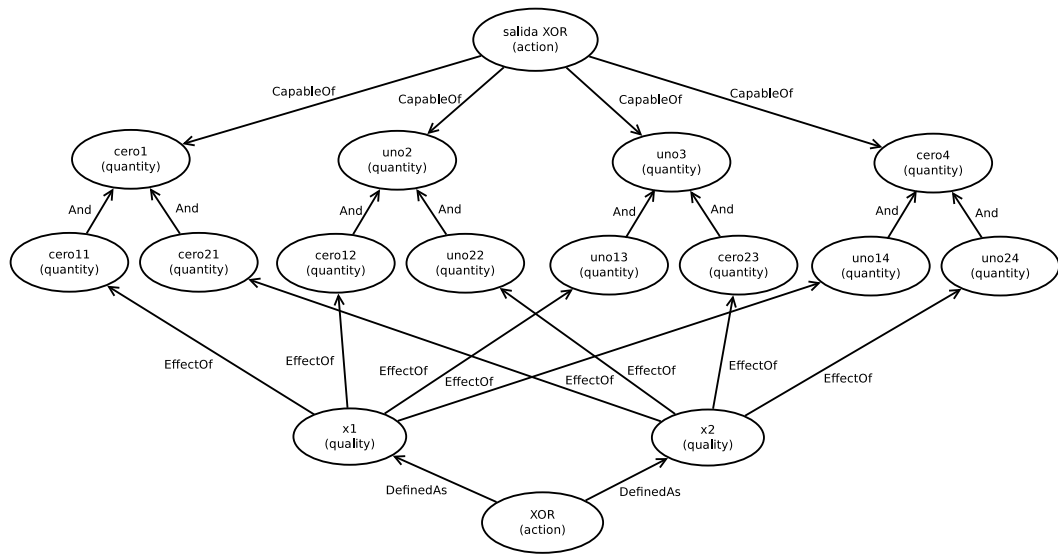


Figura 6.2: Representación de lógica XOR utilizando la notación propuesta.

realizar una labor similar con el modelo de representación propuesto. Tal como las redes de petri pueden realizar operaciones o correr algún tipo de código en sus nodos ejecutables, el modelo propuesto igual puede ejecutar funciones con código para realizar operaciones. Tal como se ha mostrado con el ejemplo anterior del robot trabajador, cada nodo realizaba alguna acción mediante la ejecución de código.

En la Figura 6.3 se ve como se hizo una conversión directa de la representación de un modelo de un sistema productor-consumidor, aunque esta representación ya fue dada con anterioridad, solo se había dado con la representación propuesta, ahora se agrega la representación utilizando notación de una red de petri. Considerando la simplicidad del modelo en cuestión, no es necesario una explicación a fondo fuera de la ya dada con anterioridad. Solo que se puede visualizar la diferencia con más claridad teniéndolos juntos.

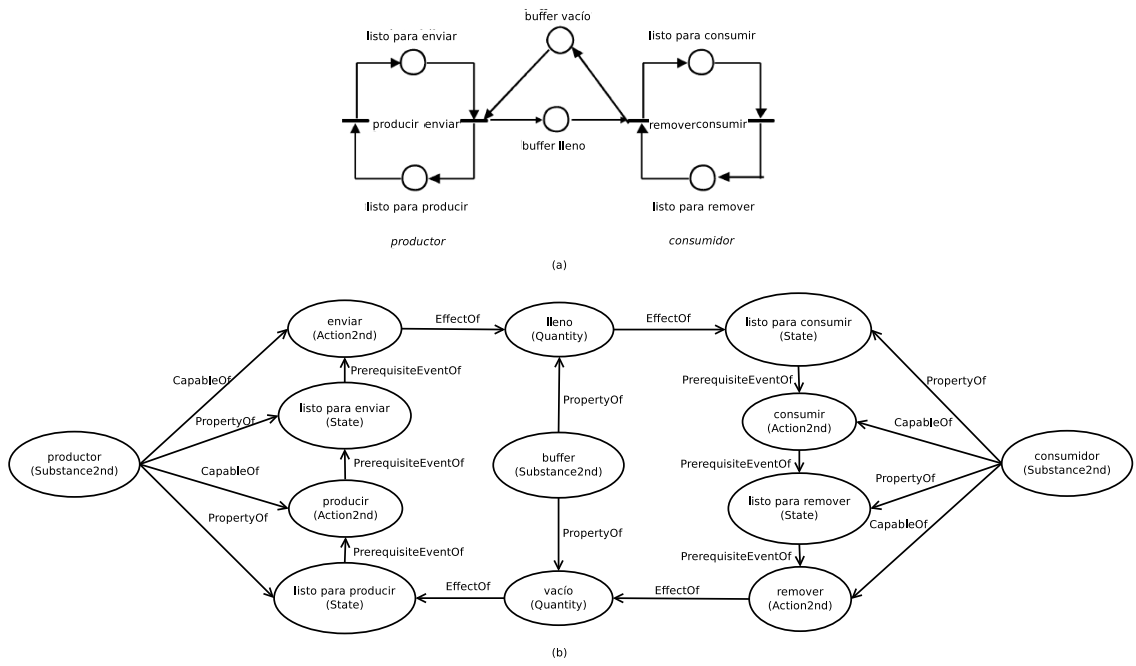


Figura 6.3: Comparativa de un modelo de sistema productor-consumidor con red de petri y con el modelo de representación propuesta. (a) Representación utilizando una red de petri. (b) Representación utilizando la notación propuesta.

Capítulo 7

Conclusiones y trabajo futuro

7.1. Conclusiones

Conociendo que el objetivo principal de esta investigación es crear un modelo para una base de conocimiento robusta y multidominio, tras haber comparado la implementación del modelo con otros modelos y métodos de representación, se puede concluir que el resultado fue satisfactorio debido a que se logró generalizar la base de conocimientos a soportar al menos cuatro tipos de conocimientos heterogéneos, que son sentido común, toma de decisiones, secuencias de estados dinámicos, y representación visual/auditiva. Todo esto considerando una abstracción de como el humano piensa y percibe, y no a como usualmente se quiere que una computadora lo procese con mayor facilidad y simplicidad.

Tomando los puntos más importantes de los resultados obtenidos y su análisis, se pueden remarcar las siguientes conclusiones particulares:

Es posible hibridar una red semántica con la teoría de lógica difusa y a su vez generar un modelo de representación más apegado a la percepción humana.

Utilizando el modelo propuesto, se puede generar una herramienta para desarrollo de planeación de secuencias de estados con tomas de decisiones auxiliadas por lógica difusa.

Es posible crear una base de conocimiento considerando una abstracción filosófica y consigo llegar a un acercamiento de lo que una inteligencia artificial dura pudiera utilizar.

El modelo propuesto es un buen primer acercamiento a una base de conocimientos general.

7.2. Trabajo futuro

Intercambiar las categorías aristotélicas por categorías kantianas. Debido a la mayor completitud de éstas categorías kantianas, el grado de representatividad sería mas adecuado a una representación del pensamiento humano, ya que las categorías aristotélicas son más limitadas en este aspecto.

Buscar mejorar la lista de relaciones para generalizar la red semántica aún más. Siendo las relaciones la debilidad de las redes semánticas, con éstas mejoras la naturaleza multidominio del modelo sería aun mayor.

Hibridar con redes asociativas para poder realizar operaciones gramaticales libres de contexto. Con esto se pudiera integrar conocimiento de procedimientos matemáticos, lenguaje, y ciertos tipos de decisiones.

Utilizar funciones de membrecía de tipo 2 para mejorar la representatividad y percepción de cada relación y/o nodo. Sabiendo que el humano siempre tiene algún grado de incertidumbre respecto todo conocimiento que posee, la naturaleza de la lógica difusa tipo 2 incluye dicha incertidumbre de forma intrínseca, por lo que la

representatividad sería aún más apegada a la forma de percibir del humano.

Construir inferencias de aprendizaje y razonamiento, sobre la base de conocimiento. Teniendo ya una base de conocimiento robusta y generalizada, se requieren de algoritmos para poder hacer uso del conocimiento, sea para que aprenda, la razone o tome decisiones inteligentes.

Agregar semiótica a la base de conocimiento. Dando un paso más allá de lo inicialmente estipulado, con esto se pudiera generalizar aún más el conocimiento, reduciendo y optimizando la red de conocimiento para que, en conjunto con la integrada lógica difusa, la red almacene el conocimiento tal como el humano lo pudiera percibir.

Referencias

- [1] Richard A Frost. *Introduction to knowledge base systems*. August 1986.
- [2] Randall Davis, Howard Shrobe, and Peter Szolovits. What Is a Knowledge Representation ? *AI Magazine*, 14(1):17–33, 1993.
- [3] A COLLINS and M QUILLIAN. Retrieval time from semantic memory. *Journal of Verbal Learning and Verbal Behavior*, 8(2):240–247, April 1969.
- [4] Allan M. Collins and Elizabeth F. Loftus. A spreading-activation theory of semantic processing.
- [5] Marvin Minsky. A Framework for Representing Knowledge. *MIT-AI Laboratory Memo*, 306, 1974.
- [6] Lotfi A. Zadeh. Fuzzy Sets. *Information and Control*, 8(1965):338–353, 1965.
- [7] E. F. Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387, June 1970.
- [8] Aristotle. *Categories and De Interpretatione (Clarendon Aristotle Series)*. Oxford University Press, USA, 1975.

- [9] Immanuel Kant. *Critique of Pure Reason (The Cambridge Edition of the Works of Immanuel Kant in Translation)*. Cambridge University Press, 1999.
- [10] Charles S. Peirce. On a New List of Categories. In *Proceedings of the American Academy of Arts and Sciences* 7, pages 287–298, 1868.
- [11] M. O. Rabin and D. Scott. Finite Automata and Their Decision Problems. *IBM Journal of Research and Development*, 3(2):114–125, April 1959.
- [12] Carl A. Petri. Communication with automata, 1966.
- [13] Ronald R. Yager. A framework for reasoning with soft information. *Information Sciences*, 180(8):1390–1406, April 2010.
- [14] Doug Lenat, Mayank Prakash, and Mary Shepherd. CYC: Using common sense knowledge to overcome brittleness and knowledge acquisition bottlenecks. *AI Mag.*, 6(4):65–85, 1986.
- [15] Douglas B. Lenat and R. V. Guha. Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project. December 1989.
- [16] D.G. Stork. *Character and document research in the Open Mind Initiative*. IEEE, 1999.
- [17] M.A. Hearst, R.D. Hunson, and D.G. Stork. Building intelligent systems one e-citizen at a time. *IEEE Intelligent Systems*, 14(3):16–20, May 1999.
- [18] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. *Yago*. ACM Press, New York, New York, USA, May 2007.

- [19] Wikipedia. Plagiarism — Wikipedia, The Free Encyclopedia, 2004.
- [20] George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. WordNet: An on-line lexical database. *International Journal of Lexicography*, 3:235—244, 1990.
- [21] Steffen Staab and Rudi Studer. *Handbook on Ontologies (International Handbooks on Information Systems)*. Springer, 2004.
- [22] Carmen-Veronica Bobeanu, Eugene J. H. Kerckhoffs, and Hendrik Van Landeghem. Modeling of Discrete Event Systems: A Holistic and Incremental Approach Using Petri Nets. *ACM Transactions on Modeling and Computer Simulation*, 14(4):389–423, 2004.
- [23] Simon Kendal and Malcolm Creen. *An Introduction to Knowledge Engineering*. Springer, 1st edition, 2006.