Chapter 1

Razonamiento

Objetivo

El alumno construirá sistemas de razonamiento lógico y de razonamiento probabilístico mediante uso de conocimiento

1.1 Representación y uso de *conocimiento*

1.1.1 Introducción

Definir *conocimiento* (http://bit.ly/zD1zA3) no es una tarea sencilla, por ejemplo, Platón (http://bit.ly/7vLkRs) la define como "creencia verdadera justificada". Por otro lado, Bertrand Russell (http://bit.ly/4Udzi) dice: "La cuestión de cómo debería definirse al conocimiento posiblemente es más importante y difícil ... esto puede parecer soprendente: a primera vista podría pensarse que conocimiento puede definirse como creencia que es acorde con los hechos. El problema es que nadie sabe qué es *creencia*, nadie sabe qué es un *hecho* y nadie sabe qué tipo de *concordancia* entre ellos haría verdadera a la creencia: comencemos con creencia".

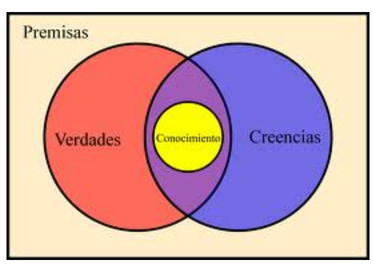

Figure 1.1: Definición clásica de conocimiento

Desde el punto de vista computacional también hay varias formas de definir conocimiento. Una *base de conocimientos* se describe como un *mapeo* entre objetos y relaciones en el dominio del problema; y los objetos y relaciones computacionales de un programa. Los resultados de inferencias en la base de conocimientos se asume que corresponden a resultados de acciones y observaciones en el mundo real. Los objetos, relaciones e inferencias computacionales se ponen a disponibilidad de los programadores por un *lenguaje de representación de conocimiento*.

Se puede considerar que conocimiento sobre un tema es un *conjunto de hechos* o *verdades*. Para que la computadora pueda utilizar ese conocimiento debe existir alguna forma de representar el conocimiento de manera tal que pueda almacenarse en la computadora. En el nivel más bajo, se tienen solamente 0s y 1s, pero las cadenas de 0s y 1s se organizan e interpretan en un alto nivel de abstracción, como enteros o caracteres por ejemplo.

1.1.2 Atributos de una buena representación

Buscar cómo representar o la mejor forma de capturar los aspectos críticos (información) de actividad inteligente para utilizarlos en una computadora ha sido un tema de discusión constante durante más de 60 años de historia de la IA.

Es importante distinguir entre un *esquema* de representación y el *medio* para implementarlo, esto es similar a la diferencia entre estructuras de datos y lenguajes de programación: los lenguajes son el medio para implementar, la estructura de datos es el esquema. Generalmente los lenguajes para representar conocimiento son más restringidos que los lenguajes de programación: son estructuras para representar jerarquías de conocimiento. El medio de implementación puede ser cualquier lenguaje de programación, aunque es preferible usar lenguajes declarativos.

Figure 1.2: Un esquema de representación de conocimiento

Cualquier esquema de representación de conocimiento debe tener al menos las siguientes características:

1. **Adecuación**. El método de representación debe ser adecuado para capturar todo el conocimiento relevante. Es difícil extraer conocimiento del lenguaje natural, sería necesario *limpiar* los textos. En cambio, una representación en lenguaje formal tiene ventajas para extraer las partes esenciales.

2. **Eficiencia**. Es deseable tener una representación minimalista, evitando información redundante siempre que sea posible. Esto incluye permitir que conocimiento no representado explícitamente pueda ser inferido a partir de ese conocimiento explícito.

3. **Extensibilidad**. Debe ser relativamente sencillo extender la representación para incluir conocimiento nuevo.

4. **Propiedad**. El esquema de representación debe ser apropiado para representar el dominio de conocimiento. La forma apropiada de representación depende del tipo de conocimiento y del tipo de tarea para la cual se utilizará. Por ejemplo, una representación *pictórica* puede resulta apropiada para reconocimiento de imágenes, pero no para realizar inferencias.

Programa: Punto flotante

Consiste en hacer un código que convierta de decimal a punto flotante y viceversa con el algoritmo visto en clase (https://bit.ly/42ePLfq).

*

1.1.3 Reglas

Una forma muy usada, debido a su simplicidad, para representar conocimiento es mediante *reglas de producción*, llamadas *reglas IF-THEN*, se pueden presentar de diversas formas:

- ♢ IF *condición* THEN *acción*
- ♢ IF *premisa* THEN *conclusión*
- ♢ IF *proposición* 1 Y *proposición* 2 *son verdaderas* THEN *proposición* 3 *es verdadera*

Una ventaja de este tipo de reglas es que son modulares y cada una define *pequeñas* partes del conocimiento y, en principio, independientes. Nuevas reglas pueden añadirse y reglas viejas se eliminan independientemente de otras reglas.

Un sistema basado en reglas contiene reglas y hechos globales acerca de un dominio de conocimiento específico. Durante la ejecución, es posible añadir una base de conocimiento local relativo al caso particular tratado. Uno de los sistemas basados en reglas más utilizados es Mycin (http://bit.ly/124TO7Y), un sistema experto diseñado para asistir en diagnósticos y tratamientos de infección bacterial.

El motor de inferencia determina la manera en la que se utiliza la base de conocimientos. Es un principio básico para sistemas en producción que cada regla debe ser un elemento independiente de conocimiento y, esencialmente, ignora las otras reglas. El motor de inferencia puede entonces "disparar" reglas en el momento en que sus premisas se satisfagan.

Si en un momento dado se satisfacen las premisas de varias reglas, el motor de inferencia debe tener un mecanismo de *resolución de conflictos*. Esto puede lograrse, por ejemplo, con un orden predefinido por algún atributo particular, o basado en la frecuencia de uso de las reglas.

Se pueden realizar inferencias *hacia adelante* y *hacia atrás* utilizando las reglas: cada una responde diferentes tipos de preguntas. Por ejemplo, en Mycin una inferencia hacia adelante puede responder la pregunta "¿Qué sugieren estos síntomas?", mientras que una inferencia hacia atrás responde a preguntas como "¿Este paciente sufre de *esta* enfermedad?". En general las reglas y los objetivos pueden ser construidos de forma diferente para realizar inferencias hacia adelante y hacia atrás.

1.1.4 Lógica

La representación lógica crece por los esfuerzos de filósofos y matemáticos para caracterizar los principios de *razonamiento correcto*. El objetivo principal de la lógica es el desarrollo de lenguajes formales de representación con sistemas de inferencia correctos (*soundness*) y completos (*completeness*). La semántica del cálculo de predicados busca operaciones que preservan veracidad en expresiones bien formadas.

Una línea alternativa nace por los esfuerzos de psicólogos y lingüistas para caraterizar el entendimiento humano. Este trabajo se ocupa menos por establecer una ciencia de razonamiento correcto y más en describir la forma en la que los humanos *adquieren*, *asocian* y *utilizan* el conocimiento de su mundo. Este enfoque ha resultado de gran utilidad para aplicaciones de AI en áreas como reconocimiento del lenguaje o *razonamiento de sentido común.*

Se presentan muchos problemas al intentar mapear razonamiento de sentido común con lógica formal. Por ejemplo, es muy común asociar los operadores $\vee$ y $\rightarrow$ con las palabras "*o*" y "*si ... entonces ...*". Sin embargo, las operaciones lógicas se ocupan únicamente de valores de verdad e *ignoran* que "*si ... entonces ...*" sugiere relación entre las premisas y su conclusión.

Por ejemplo la sentencia "si un ave es un cardenal, entonces es roja" (asocia al ave cardenal con el color rojo), se puede escribir usando cálculo de predicados de la siguiente forma:

$\forall x\,(cardenal\,(x) \rightarrow rojo\,(x))$

Este enunciado puede cambiarse, mediante una serie de operaciones que preservan veracidad, en la siguiente expresión lógicamente equivalente:

$\forall x\,(\neg rojo\,(x) \rightarrow \neg cardenal\,(x))$

Ambas expresiones tienen el mismo valor de verdad; es decir, la segunda es verdadera *si y solo si* lo es la primera. Pero la equivalencia del valor de verdad no es apropiada en esta situación:

- ◇ Si buscamos evidencia física de la verdad de estos enunciados, el hecho de que una hoja de papel no sea roja y tampoco es un cardenal es evidencia de la veracidad de la segunda expresión; como ambos enunciados son lógicamente equivalentes, entonces también es evidencia de la veracidad del primero; como conclusión, saber que la hoja de papel no es roja es evidencia de que los cardenales son rojos.

Figure 1.3: Cardenal rojo

1.1.5 Redes semánticas

Las teorías *asociacionistas*, que siguen la tradición *empiricista* en filosofía, definen al significado de un objeto en términos de una red de asociaciones con otros objetos. Para los asociacionistas, cuando los humanos perciben un objeto, primeramente la percepción se mapea a un concepto, este concepto es parte del conocimiento completo y esta conectado con otros objetos mediante relaciones apropiadas. Estas relaciones forman un entendimiento de las propiedades y comportamiento de objetos. Por ejemplo, mediante la experiencia asociamos el concepto de *nieve* con otros conceptos tales como frío, blanco, muñeco, hielo, etc. Nuestro entendimiento de la nieve y la veracidad de enunciados como "la nieve es fría" y "el muñeco de nieve es blanco", manifiestan una *red de asociaciones.*

Existe evidencia psicológica que indica que, además de la habilidad de asociar conceptos, los humanos organizan su conocimiento *jerárquicamente*, con información almacenada en los niveles más altos de la taxonomía. Collins y Quillian (1969) (http://bit.ly/1j8SNXj) modelaron el almacenamiento de información humana utilizando redes semánticas como la de la siguiente figura. La estructura de esta jerarquía se deriva de pruebas con humanos; se les preguntó sobre diferentes características de las aves, tales como "¿Un canario es un ave?", "¿Puede cantar un canario?" o "¿Puede volar un canario?".

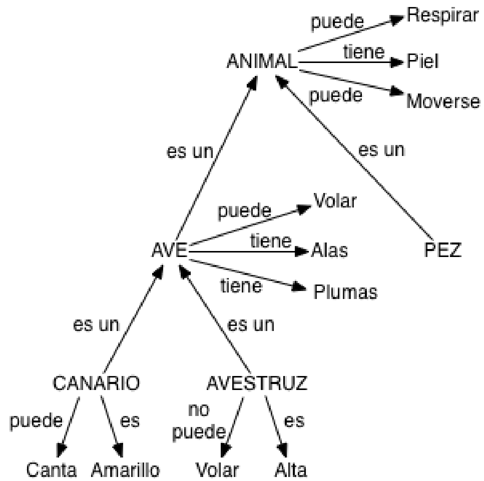

Figure 1.4: Red semántica de Collins y Quillian

Aunque estas preguntas parecen tener respuestas muy obvias, el tiempo de reacción indica que toma más tiempo responder "¿Puede volar un canario?" que "¿Puede cantar un canario?", Collins y Quillian argumentan que la explicación es que esta información está almacenada en diferentes niveles de abstracción. En lugar de intentar recordar: "los cardenales pueden volar", "los canarios pueden volar", "las golondrinas pueden volar", etc., los humanos recuerdan que los canarios *son* aves y que casi todas las aves *tienen* la capacidad de volar. Características más generales, tales como comer, respirar y moverse se almacenan en el nivel "*animal*" y, por tanto, tratar de recordar si un canario puede respirar debería tomar más tiempo que recordar si puede volar. Esto se debe a que los humanos deben recorrer más niveles hacia arriba de su estructura de memoria para obtener la respuesta.

El recuerdo más rápido se obtiene para características específicas del tipo de ave, por ejemplo, si puede cantar o si es de algún color. Además, en ese mismo nivel parece que se utiliza un sistema de *manejo de excepciones*: cuando se preguntó a los participantes si una avestruz puede volar, la respuesta fue más rápida que cuando se les preguntó si puede respirar. Entonces la jerarquía *avestruz* → *ave* → *animal* no fue recorrida para obtener esta información: se encuentra almacenada *directamente* en avestruz. Este tipo de organización de conocimiento se ha formalizado usando *sistemas de herencia*.

Los sistemas de herencia permiten almacenar información en los niveles de abstracción más altos, lo que reduce el tamaño de la base de conocimientos y ayuda a prevenir inconsistencias al actualizar. Por ejemplo, si se está creando una base de conocimientos acerca de aves, se puede definir rasgos comunes a todas las aves, para la *clase* general ave y permitir que especies particulares de aves hereden esas propiedades. Esto reduce el tamaño de la base de conocimeintos al requerir que los rasgos comunes se definan solamente una vez, en lugar de tener que hacerlo para cada especie. Por otro lado, si se desea añadir la especie cadenal a una base de conocimientos existente, se puede notar que el cardenal es un ave que canta; por tanto, el cardenal puede heredar las propiedades comunes tanto de *ave* como de *ave cantante*. Esto ayuda a mantener la consistencia de la base de conocimientos al añadir nuevos tipos de aves, además evita que el programador tenga que preocuparse por añadir esta información.

Las gráficas han probado ser la forma ideal para formalizar las teorías asociacionistas del conocimiento, dado que permiten representar explícitamente las relaciones utilizando nodos y arcos. Una **red semántica** representa conocimiento como una gráfica en la que los nodos corresponden con hechos o conceptos y los arcos son las relaciones o asociaciones entre conceptos; generalmente tanto nodos como arcos están etiquetados.

El término "red semántica" incluye una familia de represntaciones basadas en gráficas, principalmente varían en los nombres que están permitidos para nodos y aristas, así como las inferencias que pueden realizarse. Sin embargo, un conjunto común de supuestos e intereses están compartidos por todos los lenguajes de representación de redes. Por ejemplo, las redes conceptuales (http://bit.ly/WxVjNl), introducidas por John F. Sowa (http://bit.ly/eXXDj4) en 1984, son un lenguaje de representación de redes más moderno.

1.1.6 Marcos de Minsky

Otro esquema de representación de conocimiento, muy similar a los *scripts*, cuya intención era capturar datos estructurados *implícitos* y conexiones de información *implícita* en el dominio del problema, se conocen como *marcos*. Esta representación soporta la organización de conocimiento en unidades más complejas que reflejan la organización de objetos en el dominio.

En 1975, Minsky (http://bit.ly/Cy2UD) describe un marco como:

◇ "cuando una persona encuentra una situación nueva (o realiza cambios substanciales en su punto de vista de un problema), selecciona de su memoria una estructura llamada "*marco*". Este es un *marco de referencia* de recuerdos que se adapta para ajustarse a la realidad cambiando los detalles necesarios".

De acuerdo con Minsky, un marco puede verse como una estructura de datos *estática* usada para representar situaciones con estereotipos *bien definidos*. Las estructuras tipo marco organizan nuestro conocimiento del mundo, solo se necesita realizar ajustes extrayendo información estructurada por experiencias anteriores.

Cualquiera que se ha hospedado en uno o dos habitaciones de hoteles no tiene problemas al ingresar a hoteles nuevos: se espera ver una cama, un baño, un lugar para abrir las maletas, un teléfono, precios e información de emergencias, etc. Los detalles de cada habitación nueva pueden añadirse cuando sea necesario: color de cortinas, localización, tipo de *switches* de luz, etc. Todas las partes de un cuarto de hotel genérico están organizadas en una estructura conceptual a la que se accede cuando se ingresa a un hotel: las particularidades de cada habitación se sumunistran cuando se necesitan.

Estas estructuras de alto nivel se pueden representar directamente utilizando redes semánticas organizándolas como una colección de redes separadas, cada una para cada situación. Los marcos (al igual que los *sistemas orientados a objetos*) proveen un medio de organización, representando entidades como objetos estructurados con nombre y valores asociados: un marco o *esquema* es una sola entidad compleja.

Por ejemplo, un cuarto de hotel y sus componentes se pueden describir por varios marcos individuales:

◇ Además de la cama, un marco puede representar una silla: altura esperada 20 a 40 cm; el número de patas es 4, un valor por omisión; diseñada para sentarse. Otro marco representa un teléfono de hotel: es una especialización de un teléfono regular, excepto que el cobro se realiza por cuarto, el cliente puede usarlo para solicitar comida al cuarto, llamadas hacia el exterior, etc.

La siguiente figura muestra un marco para habitaciones de hotel.

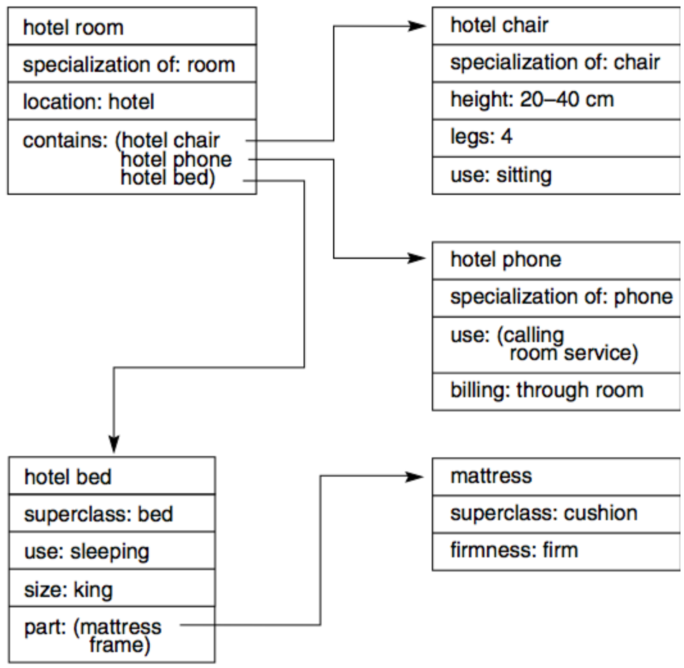

Figure 1.5: Parte de un cuadro descripción de una habitación de hotel; la especialización indica apuntadores hacia superclases

Cada marco individual puede verse como una estructura de datos similar a un "registro" tradicional que contiene información relevante sobre entidades estereotipadas.

Los marcos agregan poder a las redes semánticas parmitiendo representar objetos complejos como un solo marco, en lugar de una red estructurada muy grande. Además, provee una forma natural de representar entidades estereotípicas, clases, herencia y valores por omisión. Aún cuando los marcos, al igual que la lógica o representaciones de red, son herramientas poderosas, muchos de los problemas en la adquisición y organización de bases de conocimientos complicadas deben ser resueltos por las habilidades e intuición de los programadores.

Finalmente, es importante mencionar que esta investigación del MIT en los 70's, así como en el *Xerox Palo Alto Research Center*, llevaron al paradigma (filosofía) de programación orientada a objetos, además de la construcción de la implementación lenguajes importantes como Smalltalk, C++ y Java.

---
*

El problema de organizar y recuperar el conocimiento es difícil e inherente al modelado del significado semántico. Eugene Charniak (http://bit.ly/124Xt5T) mostró que la cantidad de conocimiento requerido incluso para entender historias infantiles es demasiado grande.

Por esto se han planteado proyectos como *Roboy el robot humanoide que nació* (http://bit.ly/1gBb1Ap) o Turing's 'Child-Machine' (http://bit.ly/VlbuKf) para simular el crecimiento y desarrollo humano.

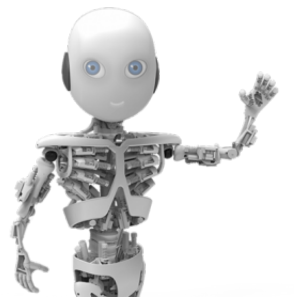

Figure 1.6: Roboy

Razonamiento

El racionalismo (del latín, *ratio*, razón http://bit.ly/14aWhmX) es una corriente filosófica que se desarrolló en Europa continental durante los siglos XVII y XVIII, formulada por René Descartes (http://bit.ly/1ZQ2baR), que se complementa con el criticismo de Immanuel Kant (http://bit.ly/92LnEx), y que es el sistema de pensamiento que acentúa el papel de la razón en la adquisición del conocimiento.

Razonar (http://bit.ly/10XSt8) es el **proceso de obtener conclusiones, juicios o inferencias a partir de hechos o premisas**. El razonamiento generalmente está asociado con el *pensamiento* (http://bit.ly/X9wqrH), la *cognición* (conocimiento: http://bit.ly/my49hk) y el *intelecto* (http://bit.ly/15hqvCP).

Una razón es una *explicación* o una *justificación* de algún evento, fenómeno o comportamiento. La lógica (http://bit.ly/4m2HA5) estudia la manera en que los humanos razonan vía argumentos. Razonar se asocia con la habilidad de cambiar *conscientemente* creencias, actitudes, tradiciones, etc.

Tipos de razonamiento

Deductivo (http://bit.ly/5NcnJJ)

La *deducción* es una forma de razonamiento en la que la conclusión se obtiene necesariamente a partir de las premisas, es una inferencia de lo general a lo particular. Una deducción es también la conclusión que se obtiene de un proceso de razonamiento deductivo.

Un ejemplo clásico de este tipo de razonamiento se encuentra en los silogismos (http://bit.ly/IQVgA):

- ♢ Premisa 1: Todos los humanos son mortales
- ♢ Premisa 2: Aristóteles es humano
- ♢ Conclusión: Aristóteles es mortal

El razonamiento de este argumento es *válido* porque no hay forma en que las premisas sean verdaderas y la conclusión falsa.

Figure 1.7: Golem II+ y Kasparov vs Depp Blue

Inductivo (http://bit.ly/YWLqWp)

La *inducción* es una forma de inferencia que produce proposiciones acerca de objetos no observados, basadas en observaciones previas. Se utiliza para formular afirmaciones generales basadas en observaciones limitadas sobre patrones recurrentes. Es una inferencia de lo particular a lo general.

Este tipo de razonamiento contrasta fuertemente con el deductivo en que aún en el mejor caso, un razonamiento inductivo no puede garantizar la validez de la conclusión; sólo se tiene cierto grado de probabilidad. Es un método de razonamiento *ampliativo*: la conclusión contiene más información que la contenida en las premisas.

Un ejemplo clásico de este tipo de razonamiento se debe al empiricista David Hume (http://bit.ly/9LlSiI):

- ◇ Premisa: El sol ha salido por el tonalquizayan (este) hasta el día de hoy
- ◇ Conclusión: El sol saldrá por el tonalquizayan (este) mañana

Figure 1.8: El *hueyxolotl* (guajolote) recibe su alimento todas las mañanas

Figure 1.9: En la víspera de navidad

Abductivo (http://bit.ly/8jRjSd)

El razonamiento abductivo es un tipo de razonamiento *a la mejor explicación*, en este, la conclusión no se obtiene con total certeza a partir de las premisas y contiene más información que las mismas. Se distingue de otros tipos de razonamiento (como el inductivo) en que es un intento de favorecer una conclusión sobre otras, demostrando que es la *más viable*. Por ejemplo, cuando un paciente muestra ciertos síntomas, pueden existir varias causas posibles, pero una de ellas se elige al ser la más probable.

Figure 1.10: Agentes *abductivos*

Por analogía (http://bit.ly/YWO7al)

En un tipo de razonamiento de lo particular a lo particular. Por ejemplo:

- ♢ Premisa 1: Sócrates es humano y murió
- ♢ Premisa 2: Platón es humano
- ♢ Conclusión: Platón morirá

Razonamiento automatizado (http://bit.ly/15hIKYD)

El campo del razonamiento automatizado estudia como el razonamiento puede modelarse o no. Sus subáreas más desarrolladas son las pruebas automáticas de teoremas (http://bit.ly/15hJkWn), pruebas interactivas de teoremas (menos automatizadas: http://bit.ly/15hJpt6), verificación automática de pruebas: garantizar un razonamiento correcto bajo ciertas suposiciones (http://bit.ly/15hJu06) así como verificación de modelos: verificar si el modelo de un sistema cumple cierta especificación (http://bit.ly/fGjrpu).

1.2 Razonamiento progresivo

En este tipo de razonamiento progresivo, se empieza a partir de un conjunto de premisas y se evoluciona hacia una conclusión.

1.2.1 Cálculo proposicional (https://bit.ly/3isUWkT)

El cálculo proposicional permite realizar razonamiento progresivo aplicando ciertas reglas bien establecidas.

1.2.1.1 Sintaxis

La sintaxis es muy simple. Tenemos el conjunto de proposiciones atómicas

$$PA = \{p, q, r, p_0, \ldots\}$$

y las conectivas lógicas usuales:

$$\neg, \vee, \wedge, \Rightarrow, \Leftrightarrow$$

O en notación de Backus-Naur:

$$\alpha ::= PA | \neg\alpha_1 | (\alpha_1 \vee \alpha_2) | (\alpha_1 \wedge \alpha_2) | (\alpha_1 \Rightarrow \alpha_2) | (\alpha_1 \Leftrightarrow \alpha_2)$$

donde $PA ::= p|q|r|p_n|q_n|r_n$, y $n \in \mathbb{N}$.

1.2.1.2 Semántica

♢ La semántica del cálculo de proposiciones se expresa en términos de valores de verdad.

♢ Los valores de verdad más utilizados son los valores booleanos $\mathbb{B} = \{V, F\}$.

♢ Una evaluación es una función $e : PA \to \mathbb{B}$.

♢ Esta función se puede extender a proposiciones compuestas cuando se combina con funciones booleanas asociadas a cada una de las conectivas.

Funciones booleanas

1. Funciones de cero argumentos: V^0, F^0

2. Funciones de un argumento:

	id / π^1_1	V^1	F^1	$\neg$
V	V	V	F	F
F	F	V	F	V

3. Funciones booleanas binarias:

$\wedge$	$\vee$	$\Rightarrow$	$\Leftrightarrow$	$\not\equiv$	$\uparrow$	$\downarrow$	$\Leftarrow$
V	V	V	V	F	F	F	V
F	V	F	F	V	F	V	V
F	V	V	F	V	F	V	F
F	F	V	V	F	V	V	V

    ♢ $\not\equiv$ es *o exclusiva*

    ♢ $\uparrow$ es la *negación conjunta*

    ♢ $\downarrow$ es la *negación alternativa*

4. Todas las conectivas pueden definirse en términos de:

    ♢ $\neg$ y alguna de:

    - $\vee, \wedge$

    ♢ $\uparrow$

    ♢ $\downarrow$

♢ Lo anterior parece aplicar solamente a las conectivas binarias o unarias.

♢ Sin embargo, se aplica a las funciones boolenas de cualquier número de argumentos.

♢ Por ejemplo, la conectiva ternaria siguiente se puede definir en términos de dos binarias: if p then q else $r \equiv_{def} (p \Rightarrow q) \wedge (\neg p \Rightarrow r)$

♢ Y esto se puede generalizar a cualquier valor de n.

1.2.1.3 Proposiciones compuestas

Para determinar el valor de verdad de una fórmula proposicional compuesta, se procede evaluando recursivamente sobre la estructura de dicha fórmula:

$$\alpha ::= PA|\neg\alpha_1|\,(\alpha_1 \vee \alpha_2)\,|\,(\alpha_1 \wedge \alpha_2)\,|\,(\alpha_1 \Rightarrow \alpha_2)\,|\,(\alpha_1 \Leftrightarrow \alpha_2)$$

Algoritmo 1.1 Evaluación de fórmulas proposicionales

♢ Para que la evaluación de la fórmula α, denotada como $e(\alpha)$, tenga valor de verdad V:

- Si $\alpha \in PA$, la evaluación $e(\alpha) = V$
- Si $\alpha = \neg\alpha_1$, la evaluación $e(\alpha_1) = F$
- Si $\alpha = (\alpha_1 \vee \alpha_2)$, la evaluación $e(\alpha_1) = V$ o $e(\alpha_2) = V$
- Si $\alpha = (\alpha_1 \wedge \alpha_2)$, la evaluación $e(\alpha_1) = V$ y $e(\alpha_2) = V$
- Si $\alpha = (\alpha_1 \Rightarrow \alpha_2)$, la evaluación $e(\neg\alpha_1 \vee \alpha_2) = V$
- Si $\alpha = (\alpha_1 \Leftrightarrow \alpha_2)$, la evaluación $e((\alpha_1 \Rightarrow \alpha_2) \wedge (\alpha_2 \Rightarrow \alpha_1)) = V$

♢ En otro caso, $e(\alpha) = F$

El valor de verdad en la mayoría de las proposiciones compuestas depende de los valores de verdad de sus proposiciones atómicas. Por ejemplo:

$$(p \wedge q)$$

♢ Si $e(p) = V$ y $e(q) = V$ entonces $e(p \wedge q) = V$

♢ Si $e(p) = F$ o $e(q) = F$ entonces $e(p \wedge q) = F$

En cambio, algunas proposiciones siempre son verdaderas sin importar la asignación de valores a sus proposiciones atómicas. Ejemplo:

$$(p \vee \neg p)$$

Estas proposiciones se conocen como *tautologías*. Se acostumbra distinguirlas anteponiendo el símbolo $\models$.

Y otras siempre producen valor falso. Ejemplo:

$$(p \wedge \neg p)$$

Éstas se conocen como *contradicciones*.

Desde el punto de vista de la **complejidad** computacional, verificar si una proposición es una tautología es equivalente al célebre problema de **SAT**, el cual es **NP-completo**. La forma de resolverlo es construir la tabla de verdad de la proposición.

Sin embargo, hay otras estrategias de evaluación que en la mayoría de los casos son más eficientes.

Programa:

- ♢ Desarrollar un programa que evalúe expresiones de lógica proposicional usando el **algoritmo recursivo** presentado en clase.

- ♢ **Nota**: No es válido utilizar alguna biblioteca o programa hecho por terceros.

*

Consecuencia lógica

El símbolo $\models$ también denota una relación entre conjuntos de proposiciones y fórmulas individuales: $\alpha_1, \ldots, \alpha_n \models \beta$ se lee como: "β es consecuencia lógica de $\alpha_1, \ldots, \alpha_n$". Las proposiciones $\alpha_1, \ldots, \alpha_n$ se conocen como las premisas; y β, como la conclusión. La expresión completa se conoce como *argumento*.

Un argumento $\alpha_1, \ldots, \alpha_n \models \beta$ es *válido* si y solo si (sii) para toda evaluación $e : PA \to \{V, F\}$ se tiene que, si $e(\alpha_1) = V, \ldots, e(\alpha_n) = V$ entonces $e(\beta) = V$. En caso contrario, se dice que el argumento es *inválido*.

Sistemas de demostración

Los sistemas de demostración son herramientas para verificar la validez de argumentos lógicos por medios estrictamente sintácticos.

Un sistema de demostración está formado por un conjunto (finito) de reglas de inferencia e instrucciones sobre cómo aplicar estas reglas.

El concepto de demostración es el núcleo de un sistema: una demostración es un conjunto de fórmulas que permiten ir de las premisas a la conclusión por medio de transformaciones sintácticas. Para que un sistema de demostración sea útil debe cumplir un conjunto de propiedades metateóricas: corrección, completitud, etc.

Reglas de inferencia

Sean $\alpha_1, \ldots, \alpha_n$ y β fórmulas del cálculo proposicional, una regla de inferencia tiene la siguiente forma:

$$R \quad \frac{\alpha_1, \ldots, \alpha_n}{\beta}$$

donde: $\alpha_1, \ldots, \alpha_n$ son las premisas, β es la conclusión y R es el nombre de la regla. Si $n = 0$, el conjunto de premisas es vacío y este tipo de reglas se conoce como axioma.

Por ejemplo, la regla de inferencia modus ponens:

$$MP \quad \frac{p \Rightarrow q, p}{q}$$

Una *demostración* es una sucesión finita de pasos en los que se enlistan ya sea una premisa o la aplicación de una regla de inferencia.

1.2.1.4 Deducción natural

La deducción natural es un sistema con un conjunto grande de reglas de inferencia. Tiene reglas para introducir (señaladas con I) o eliminar (E) las conectivas lógicas. Además, hay tres reglas adicionales: contradicción (C), sustitución (S) y falso (F).

Algunas reglas contemplan la introducción de hipótesis adicionales. Por esta razón, las inferencias que se hagan utilizando estas hipótesis aparecen dentro de cajas. Las cajas se pueden cerrar extrayendo una conclusión de acuerdo con las condiciones de cada regla.

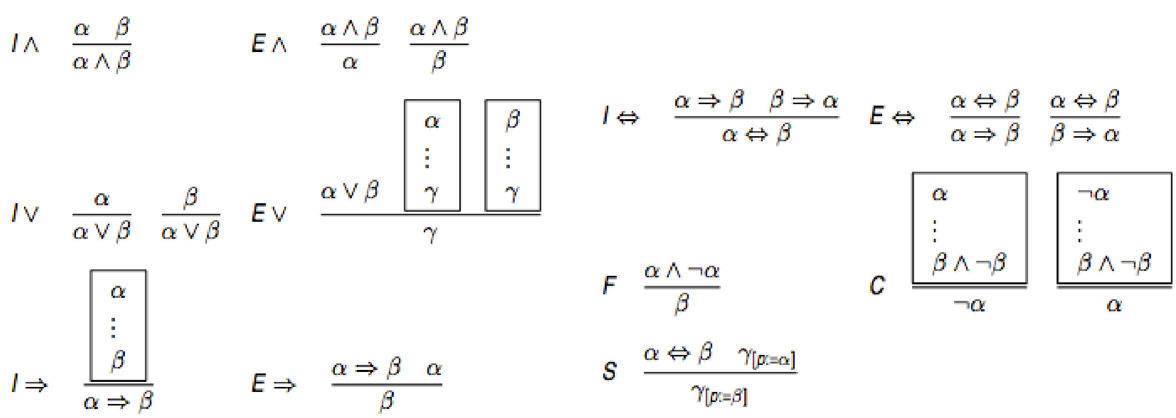

Figure 1.11: Reglas de inferencia de la deducción natural

Cuando se realizan demostraciones con cierto sistema, se suele utilizar el símbolo $\vdash_{SD}$, en el que SD hace referencia al sistema de demostración utilizado. Para deducción natural utilizaremos $\vdash_{DN}$

Ejemplo 1

$\diamondsuit$ $p \Rightarrow q, q \Rightarrow r \vdash_{DN} p \Rightarrow r$

Ejemplo 2

$\diamondsuit$ $p \vee q, \neg p \vdash_{DN} q$

Ejercicios, tarea:

1. Calcula las tablas de verdad de las siguientes proposiciones:

   ♢ $p \wedge q \Rightarrow \neg p$

   ♢ $(p \vee \neg q) \vee (\neg p \vee q)$

   ♢ $(p \Rightarrow q) \wedge q \Rightarrow p$

2. Verifica que los siguientes argumentos sean correctos:

   ♢ $\models (p \Rightarrow q) \Rightarrow ((p \vee r) \Rightarrow (q \vee r))$

   ♢ $\models (p \Rightarrow q) \Rightarrow ((p \wedge r) \Rightarrow (q \wedge r))$

   ♢ $\models p \wedge (q \vee r) \Rightarrow ((p \wedge q) \vee (p \wedge r))$

3. Demuestra los siguientes argumentos por medio de deducción natural:

   ♢ $p, p \Rightarrow q \vdash_{DN} p \wedge q$

   ♢ $p \wedge q \Rightarrow r, q \Rightarrow p, q \vdash_{DN} r$

   ♢ $p \Rightarrow q, q \Rightarrow r \vdash_{DN} p \Rightarrow q \wedge r$

   ♢ $p \Rightarrow q, \neg q \vdash_{DN} \neg p$

   ♢ $q \vdash_{DN} p \Rightarrow q$

   ♢ $p \vee (q \wedge r) \vdash_{DN} p \vee q$

   ♢ $\vdash_{DN} \neg (p \wedge \neg p)$

   ♢ $\vdash_{DN} p \vee \neg p$

*

1.2.2 Inducción estructural

La inducción estructural es un método de demostración utilizado en lógica matemática, computación y en otras áreas. Se trata de una generalización de la inducción matemática. Una prueba por inducción estructural consiste en demostrar que una proposición se cumple para todos los elementos mínimos del tipo (caso base), y si la propiedad se cumple para todas las subestructuras de una cierta estructura S, entonces se debe cumplir también para S.

Listas

Por ejemplo, considerando las siguientes funciones recursivas sobre listas:

```
longitud []      = 0                   (long1)
longitud (h:t)   = 1 + longitud t      (long2)

[] ++ lista = lista                    (concat1)
(h:t) ++ lista = h : (t ++ lista)      (concat2)
```

donde:

- ♢ [ ] denota la lista vacía.
- ♢ (h:t) es una lista en la que h representa el primer elemento (*head*) y t denota el resto (*tail*) de la lista.
- ♢ ++ denota la operación de concatenación de listas.

Y se desea probar la propiedad:

```
longitud (L ++ M) = longitud L + longitud M
```

El caso base se tiene cuando L es [ ]:

```
longitud ([] ++ M) = longitud M              { concat1 }
longitud [] + longitud M = 0 + longitud M    { long1 , aritmética }
```

Ahora se debe demostrar que la propiedad se cumple cuando L es una lista no vacía. Como L es no vacía, debe ser de la forma x:xs para un elemento x y una lista xs. La hipótesis inductiva dice que la propiedad se cumple para cualquier lista M cuando L es xs:

```
longitud (xs ++ M) = longitud xs + longitud M   (hip)
```

Para demostrar que la propiedad se cumple también para cualquier lista M cuando L es x:xs, se deben utilizar las definiciones de las funciones y la hipótesis inductiva:

```
longitud ((x:xs)++ M)   =          { concat2 }
longitud (x:(xs ++ M)) =           { long2 }
1 + longitud (xs ++ M) =           { hip. }
1 + longitud xs + longitud M =     { long2 }
longitud (x:xs) + longitud M
```

Árboles

Considerando las siguientes definición de árbol binario:

```
Arbol int = Hoja int | Rama (Arbol int Arbol int)
```

Y las siguientes funciones recursivas sobre esta estructura:

```
nHojas Hoja          = 1                          (hojas1)
nHojas Rama (I D)    = nHojas I + nHojas D        (hojas2)

nNodos Hoja          = 0                          (nodos1)
nNodos Rama (I D)    = 1 + nNodos I + nNodos D    (nodos2)
```

Se desea demostrar la propiedad:

```
nHojas (A) = nNodos (A) + 1
```

El caso base se tiene cuando A es una Hoja:

```
nNodos Hoja          = 0                          { nodos1 }
nHojas Hoja          = 1                          { hojas1 }
```

La hipótesis inductiva dice que la propiedad se cumple para cualquier subárbol izquierdo I y derecho D.

```
nHojas (I) = nNodos (I) + 1                       (hip1)
nHojas (D) = nNodos (D) + 1                       (hip2)
```

Para demostrar que la propiedad se cumple también para cualquier árbol A que se compone de un subárbol izquierdo I y otro derecho D, se deben utilizar las definiciones de las funciones y la hipótesis inductiva:

```
nHojas Rama (I D) = nNodos Rama (I D) + 1
nHojas I + nHojas D     =                       { hojas2 }
nNodos (I) + 1 + nNodos (D) + 1 =               { hip1, hip2 }
nNodos (I) + nNodos (D) + 2                     { aritmética }
1 + nNodos I + nNodos D + 1 =                   { nodos2 }
nNodos (I) + nNodos (D) + 2                     { aritmética }
```

Otra propiedad interesante es la siguiente:

♢ Si A es un árbol binario, entonces $nNodos\,(A) \leq 2^{altura(A)+1} - 1$.

Para probar esta propiedad, será necesario definir la función *altura*, que devuelve el número de nodos existentes entre la raíz y la hoja más lejana a ésta.

1.3 Razonamiento regresivo

El mecanismo de inferencia, o intérprete de reglas para el razonamiento regresivo, difiere significativamente del mecanismo de razonamiento progresivo. Si bien es cierto ambos procesos involucran el examen y aplicación de reglas, el razonamiento regresivo empieza con la conclusión deseada y decide si los hechos (reglas) que existen pueden dar lugar a la obtención de un valor para esta conclusión.

Una excelente aplicación para el razonamiento regresivo es el diagnóstico, donde el usuario dialoga directamente con el sistema basado en conocimiento y proporciona los datos a través del teclado. Problemas de clasificación también son adecuados para ser resuelto mediante este tipo de razonamiento.

Ejemplo 1

Se puede utilizar un *árbol de búsqueda* para realizar razonamiento regresivo. Considerando el siguiente *puzzle*:

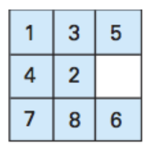

Figure 1.12: *Puzzle* a resolver

Se desea obtener la solución con los números ordenados de forma ascendente, el razonamiento hacia atrás se muestra en la siguiente figura:

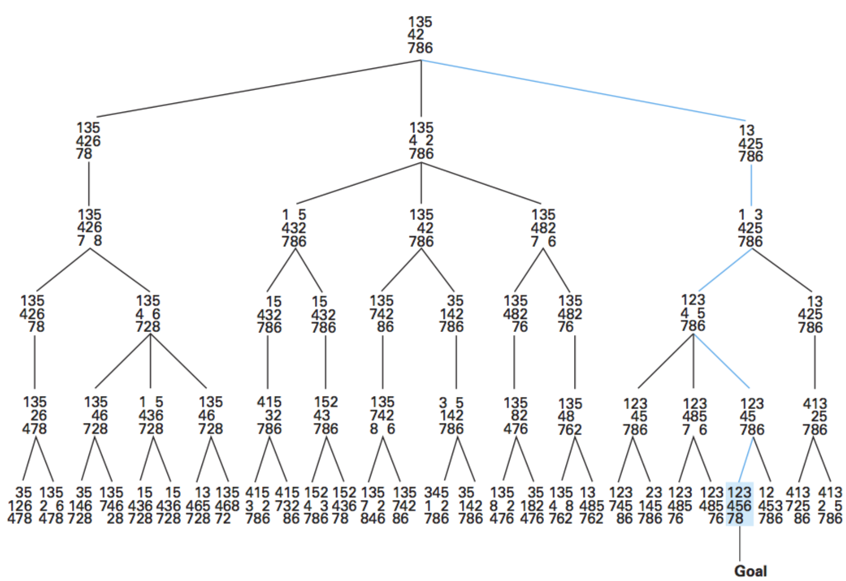

Figure 1.13: Árbol de búsqueda para resolver el *puzzle*

Es importante notar que para llevar a cabo esta búsqueda, es necesario tener el árbol completo, hay ocasiones en las que esto no es posible, dado que puede ser un árbol demasiado grande para poder almacenarlo en algún dispositivo.

Nota: En el sitio https://sliding.toys/mystic-square/8-puzzle/ se puede jugar el 8-*puzzle*.

1.4 Razonamiento con incertidumbre

"We should give up the attempt to derive results and answers with complete certainty" Michael O. Rabin

En los temas anteriores se han descrito técnicas de representación del conocimiento y razonamiento para un modelo del mundo:

- ♢ Completo
- ♢ Consistente
- ♢ Inalterable

Sin embargo, en muchos dominios de interés no es posible crear tales modelos debido a la presencia de *incertidumbre* (http://bit.ly/W1YjSR): "Falta de conocimiento seguro y claro de algo" (http://bit.ly/c8Rq3i).

Richard Feynman sobre la incertidumbre del conocimiento: http://youtu.be/xjGjcNg9GQw.

1.4.1 Factores de certeza: fuentes de incertidumbre

La incertidumbre puede provenir de distintos lugares.

1.4.1.1 Respecto a los hechos

- ♢ Ignorancia
    - Puede que en un determinado campo el conocimiento sea incompleto.
        - ▷ Por ejemplo en el campo de las Ciencias Médicas.
- ♢ Aunque se pudiera completar el conocimiento, puede ser necesario tomar decisiones con información incompleta:
    - Un paciente llega con gravedad a urgencias y es necesario proponer un tratamiento sin que sea posible realizar todos los *tests* necesarios para saber con total exactitud su enfermedad.
- ♢ En otros campos la ignorancia es irreductible:
    - En modelos físicos
        - ▷ ¿Cuál será el resultado del lanzamiento de una moneda?

- En la vida real
  - ▷ ¿Es la otra persona sincera?

◊ Vaguedad e imprecisión

- Algunos conceptos son vagos o imprecisos (subjetivos).
  - ▷ Las personas son altas, guapas, felices etc.

1.4.1.2 Respecto a las reglas

◊ Muchas reglas son generalmente *guías* que utilizan los expertos en determinadas situaciones.

◊ En el mundo real utilizamos habitualmente reglas que son:

- Inexactas o incompletas
  - ▷ "Si es un ave entonces vuela", ¿y los pingüinos?
  - ▷ "Si te duele la cabeza tienes gripe", ¿y si te diste un golpe?
- Imprecisas
  - ▷ "Si el agua está caliente añade sal al gusto"
- Inconsistentes
  - ▷ "Hasta las instituciones populares que se proponen aconsejarnos sobre comportamiento y ética están plagadas de contradicciones. Consideremos los aforismos: No por mucho madrugar amanece más temprano. Sí, pero al que madruga, dios le ayuda. Mejor prevenir que curar; pero quien no arisca no aprisca. Donde fuego se hace, humo sale; pero el hábito no hace al monje. Quien espera desespera; pero mientras hay vida hay esperanza. El que duda está perdido; pero el que nada sabe de nada duda. Dos cabezas son mejor que una; pero demasiada gallina malogra el caldo"[1]

El objetivo de un agente que aplica razonamiento con incertidumbre es ser capaz de razonar sin tener todo el conocimiento relevante en un campo determinado utilizando lo mejor posible el conocimiento que tiene disponible.

Implementar un agente que cumpla estos objetivos utilizando lógica de predicados o de primer orden es muy difícil. Por tanto, deben de introducirse modelos para manejar información vaga, incierta, incompleta y contradictoria. Lo anterior es crucial para un sistema funcione en el *mundo real*.

1.4.1.3 Actuar con incertidumbre

Recordando que el propósito de un sistema inteligente es actuar autónomamente de forma óptima utilizando el conocimiento del sistema y un conjunto de percepciones.

◊ Pero para actuar se necesita decidir que hacer. ¿Cuál es la forma correcta de tomar decisiones?

[1] Fragmento tomado de *El mundo y sus demonios* de Carl Sagan.

- ♢ La decisión racional:
    - Cuando se tienen distintas opciones un sistema debe decidirse por aquella acción que le proporcione el mejor resultado.
- ♢ Cuando hay incertidumbre para poder decidir racionalmente se requiere:
    - La importancia de los distintos resultados de una acción
    - La certidumbre de alcanzar esos resultados cuando se realiza la acción.

Inicialmente la mayoría de los investigadores en IA enfatizaban la importancia del razonamiento simbólico y evitaban la utilización de números.

- ♢ Los sistemas expertos no deben usar números puesto que los expertos humanos no lo hacen.
- ♢ Los expertos no pueden suministrar los números requeridos.

Sin embargo los ingenieros que desarrollaban las aplicaciones se dieron cuenta pronto de la necesidad de representar la incertidumbre.

Los métodos numéricos (especialmente los basados en probabilidad) son actualmente una herramienta aceptada en IA, debido a:

- ♢ Los éxitos prácticos (p.e. Mycin)
- ♢ La complejidad de teorías alternativas

Dado que ni la lógica proposicional ni la de primer orden son adecuadas para modelar incertidumbre es necesario buscar nuevos modelos, entre ellos destacan:

- ♢ Modelos Simbólicos
    - Lógicas por defecto
    - Lógicas basadas en modelos mínimos: suposición del *mundo cerrado*
- ♢ Modelos Numéricos
    - Probabilidad
    - Lógica difusa

Representación simbólica de la incertidumbre

Lógica por defecto (http://bit.ly/14LSJ6y)

Fue propuesta por Raymond Reiter (http://bit.ly/14LSNDh) para solucionar el problema del conocimiento incompleto (1980). Para ello se introducen una serie de reglas por defecto.

Intuitivamente: “Las reglas por defecto expresan características comunes a un conjunto de elementos que se asumen ciertas salvo que se indique lo contrario”.

Por ejemplo:

- ♢ “Las aves vuelan”
    - “Excepto por los pingüinos y las avestruces y ...” (se enlistan las aves que no cumplen el caso por defecto)

Suposición del *mundo cerrado* (http://bit.ly/9ZHxAt)

Sirve para manejar conocimiento incompleto. Este enfoque es utilizado en los sistemas de bases de datos así como en Prolog.

Intuitivamente: "Lo que no se puede probar a partir de la base de conocimiento es falso".

El principal inconveniente de la representación simbólica de la incertidumbre es que se obtienen teorías complejas y a veces inconsistentes.

Representación numérica de la incertidumbre

1.4.2 Razonamiento probabilístico

La teoría de la probabilidad es un área de las matemáticas que ha sido aplicada a problemas de razonamiento con incertidumbre. Es una teoría bien entendida y con mucha historia: sus formalizaciones se dieron a partir de mediados del siglo XVII. Asigna valores numéricos (llamados probabilidades) a las proposiciones. Nos dice, dadas las probabilidades de ciertas proposiciones, y algunas relaciones entre ellas cómo asignar probabilidades a las proposiciones relacionadas.

En la lógica clásica las proposiciones son ciertas o falsas. En la teoría de la probabilidad las proposiciones son también ciertas o falsas pero se tiene un grado de creencia en la certeza o falsedad.

A pesar de su larga historia los valores numéricos que representan las probabilidad no tiene una interpretación única. Algunas interpretaciones:

- ♢ Frecuentista: Es el valor de la frecuencia de que ocurra algún evento cuando el número de pruebas tiende a infinito
- ♢ Subjetiva: Es un grado de creencia acerca de un evento incierto

Aún así existe un consenso sobre el modelo matemático que soporta la teoría.

Los valores numéricos de la probabilidad: se denota por $P(A)$ a la probabilidad de la proposición A. Por ejemplo:

- ♢ $A =$ "El paciente tiene sarampión"
- ♢ $A =$ "Mañana saldrá el sol"

Los valores de la probabilidad satisfacen un conjunto de axiomas:

- ♢ $0 \leq P(A) \leq 1$
- ♢ $P(V) = 1$
- ♢ $P(\neg A) = 1 - P(A)$
- ♢ $P(F) = 0$
- ♢ $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$
  - si A y B son disjuntos, es decir $P(A \wedge B) = 0$, entonces $P(A \vee B) = P(A) + P(B)$

En muchas ocasiones tenemos *eventos* en los que el conjunto de resultados es:

- ♢ *Completo*: se conocen todos los posibles resultados
- ♢ *Mutuamente excluyente*: no se pueden dar dos resultados distintos simultáneamente

Ejemplos:

- ♢ Si lanzamos una moneda, el resultado puede ser cara o cruz
- ♢ Si tiramos un dado, se pueden producir seis resultados distintos

1.4.2.1 Variable aleatoria y distribución de probabilidad

En lugar de tener una proposición para cada resultado se introduce el concepto de ***variable aleatoria*** (VA http://bit.ly/cW9uUw). una VA o *estocástica* es una función, que asigna eventos a números reales.

Para simplificar los análisis, se suele considerar que todas las proposiciones son variables aleatorias que toman dos valores verdadero o falso. Por ejemplo, dada la proposición "tiene sarampión", se construye la variable aleatoria *Sarampión* que toma los valores verdadero y falso; y representamos la probabilidad de que un paciente tenga sarampión $P\left(\text{“}Tiene\ sarampión\text{”}\right)$ como $P\left(Sarampión = verdadero\right)$ o simplemente como $P\left(Sarampión\right)$; y la probabilidad de que un paciente no lo tenga como $P\left(Sarampión = falso\right)$ o $P\left(\neg Sarampión\right)$.

Dada una VA es posible conocer la probabilidad para cada valor que pueda tomar. Esta descripción se llama ***distribución de probabilidad*** de la VA y consiste en listar los valores de probabilidad para cada valor de la variable. Por ejemplo, si tenemos una VA llamada *Llueve*, se puede obtener una tabla como la siguiente:

	Llueve	$P\left(Llueve\right)$	
VA valores	V	0.1	probabilidades
	F	0.9	

Figure 1.14: Una distribución de probabilidad

Es muy común que se desee estudiar varias VA en conjunto. Por ejemplo si tenemos las VA *Sarampión* y *Fiebre*, podría ser de utilidad poder determinar el valor de $P\left(Sarampión \wedge Fiebre\right)$ que es la probabilidad de que el paciente tenga sarampión y fiebre; también puede escribirse como $P\left(Sarampión, Fiebre\right)$. Para determinar el valor de dicha proposición conjunta, se necesita asignar probabilidades a cada posible combinación de los valores de las variables. El listado de todos esos valores se conoce como ***distribución conjunta*** de las VA.

Ejemplo: la siguiente table muestra la distribución conjunta de las VA *Llueve* y *EnCalle*, es decir, muestra la probabilidad de que llueva y estar en la calle.

		$P\left(Llueve, EnCalle\right)$
$Llueve$	$EnCalle$	0.01
$Llueve$	$\neg EnCalle$	0.09
$\neg Llueve$	$EnCalle$	0.2
$\neg Llueve$	$\neg EnCalle$	0.7

Figure 1.15: Una distribución conjunta

Es similar a una tabla de la verdad lógica excepto que:

- ♢ Describe las probabilidad para cada combinación de valores de las variables
- ♢ Generalmente dichos valores no se pueden calcular a partir de sus componentes

La distribución conjunta es importante porque contiene todo lo que se necesita saber acerca de un conjunto de VA. Además, la distribución de cada variable individual, llamada *distribución marginal*, se puede calcular a partir de la distribución conjunta. Por ejemplo, tomando la tabla anterior, se pueden determinar:

- ♢ $P(Llueve) = P(Llueve \wedge EnCalle) + P(Llueve \wedge \neg EnCalle) = 0.01 + 0.09 = 0.1$
- ♢ $P(\neg Llueve) = 0.9$
- ♢ $P(Llueve \vee EnCalle) = 0.01 + 0.09 + 0.2 = 0.3$

1.4.2.2 Probabilidad condicional

La *probabilidad condicional* (http://bit.ly/12UwzU9) es la probabilidad de que ocurra un evento A, sabiendo que sucede el evento B. Se puede interpretar como "el grado de creencia en A cuando todo lo que sabe es B". Se escribe como $P(A|B)$, y se lee "la probabilidad de A dado B".

Se define como $P(A|B) = \frac{P(A \wedge B)}{P(B)}$, otra interpretación es: tomando los mundos en los que B se cumple, la fracción en los que también se cumple A. Gráficamente:

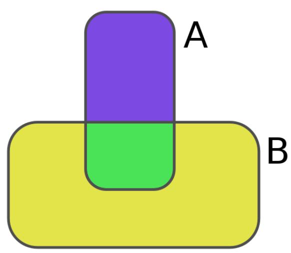

Figure 1.16: Interpretación gráfica de la probabilidad condicional

La probabilidad condicional es un concepto muy importante, debido a que permite conocer la probabilidad de que se tomen unos determinados valores por un conjunto de VA cuando se saben los valores que han tomado otras.

Ejercicio, tarea:
Determinar: $P(Llueve|EnCalle)$, $P(Llueve|\neg EnCalle)$, $P(\neg Llueve|EnCalle)$ y $P(\neg Llueve|\neg EnCalle)$
*

1.4.2.3 Razonamiento con probabilidad: la regla de Bayes

Propuesto por Thomas Bayes (http://bit.ly/RfJtDf), el teorema de Bayes (http://bit.ly/uOU1jg) establece:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Y puede interpretarse como la probabilidad de una hipótesis A dada una evidencia B: $P(A|B)$ es proporcional a probabilidad de la hipótesis $P(A)$ multiplicada por el grado en que la hipótesis predice los datos $P(B|A)$. Su importancia radica en que para muchos problemas dado un conjunto de datos (evidencia) B se debe que seleccionar la hipótesis A más probable mediante $P(A|B)$.

La forma general del teorema de Bayes se conoce como *fórmula* o *regla* de Bayes. Sea A una VA con valores $\{a_1, a_2, \ldots, a_n\}$, entonces:

$$P(a_i|B) = \frac{P(B|a_i)P(a_i)}{\sum_{j=1}^{n} P(B|a_j)P(a_j)}$$

Es importante notar que $\sum_{i=1}^{n} P(a_i|B) = 1$.

Ejercicio, tarea:

Se tienen 3 proveedores $\{A, B, C\}$, el porcentaje de compra de una pieza es de la siguiente forma: $A = 45\%$, $B = 30\%$ y $C = 25\%$. Además, se sabe que el porcentaje de artículos defectuosos por compañía es: $A = 0.03\%$, $B = 0.04\%$ y $C = 0.05\%$. Sea D el evento "la pieza es defectuosa" y N "la pieza no es defectuosa".

1. Representar la información del ejemplo como un árbol.
2. Se desea saber:
   (a) La probabilidad de que una pieza elegida aleatoriamente sea defectuosa:

   (b) El proveedor con la mayor probabilidad de haber producido una pieza defectuosa:

*

1.4.3 Razonamiento difuso: lógica difusa (*fuzzy* http://bit.ly/2rEMHe)

Es un tipo de lógica multivaluada (http://bit.ly/14LXpJq) propuesta por Lotfi A. Zadeh (http://bit.ly/13QGsSz).

Asigna a cada proposición A un "grado de verdad" V entre 0 y 1 donde:

- ♢ $V(A) = 0$ indica que la proposición es completamente falsa
- ♢ $V(A) = 1$ indica que la proposición es totalmente verdadera
- ♢ Valores intermedios de $V(A)$ reflejan diferentes grados de verdad de la proposición

Es una generalización de la lógica clásica en la que todas las proposiciones toman valores de verdad de $B = \{0, 1\}$.

Ejemplo:

Dada la proposición "La temperatura del enfermo es alta"

◇ En la lógica clásica:

- $alta\,(temp) \Leftrightarrow temp > 39$
- Por tanto si la temperatura es 38.99 diríamos que la temperatura no es alta
- Nosotros no solemos razonar así

◇ En la lógica difusa:

- Se da un grado de verdad V a *alta* en función de temperatura
- Por ejemplo: $V\,(alta\,(temp)) \left\{ \begin{array}{ll} 0 & temp < 38 \\ temp - 38 & 38 \leq temp \leq 39 \\ 1 & temp > 39 \end{array} \right\}$

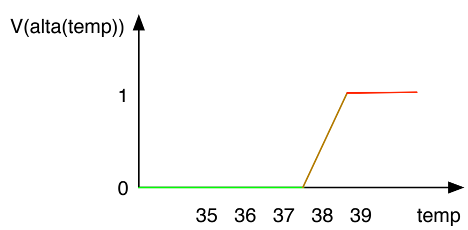

Figure 1.17: Ejemplo lógica difusa

Al igual que en la lógica clásica el valor de verdad de una proposición compuesta se calcula a través del valor de verdad de las proposiciones individuales. Existen varias formas de calcular estos valores de verdad. Los más usuales son:

◇ $V\,(p \wedge q) = min\,(V\,(p)\,, V\,(q))$

◇ $V\,(p \vee q) = max\,(V\,(p)\,, V\,(q))$

◇ $V\,(\neg p) = 1 - V\,(p)$

◇ $V\,(p \Rightarrow q) = max\,(V\,(\neg p)\,, V\,(q))$

Es importante notar que en la lógica difusa en general se cumple:

◇ $V\,(p \wedge \neg p) \neq 0$

◇ $V\,(p \vee \neg p) \neq 1$

1.4.3.1 Razonamiento con lógica difusa

De forma similar a la lógica proposicional y de primer orden, es posible aplicar reglas de inferencia en lógica difusa para obtener conclusiones. Generalmente el proceso de razonamiento difuso consta de 4 pasos

1. **Difusión**: Obtener los grados de verdad de los antecedentes

   ◇ Se obtienen los grados de verdad de los antecedentes utilizando los hechos observados.

2. **Inferencia**: Obtener los grados de verdad de los consecuentes

   ◇ Una vez calculados los grados de verdad de la premisa de cada regla se recalculan los grados de verdad de los consecuentes mediante:

   - *Mín*: los grados de verdad del consecuente se cortan a la altura del grado de verdad de la premisa

3. **Composición de consecuentes**

   ◇ Todos los grados de verdad difusos correspondientes a reglas con el mismo consecuente se combinan para dar lugar a los grados de verdad de la conclusión de las reglas mediante:

   - *Max*: se toma el máximo de los grados de verdad correspondientes a las distintas consecuencias

4. **Concisión** (Opcional): Se utiliza cuando se necesita convertir una conclusión difusa en concreta

   ◇ Generalmente se utilizan los métodos:

   - *Centroide*: Se calcula el centro de gravedad de los grados de verdad de la conclusión difusa

Ejemplo:

Se toma la temperatura a un paciente y se quiere saber la dosis apropiada de un medicamento.

◇ Hechos: $temp = 38$

◇ Reglas:

- $normal\,(temp) \rightarrow baja\,(dosis)$
- $templada\,(temp) \rightarrow media\,(dosis)$
- $alta\,(temp) \rightarrow alta\,(dosis)$

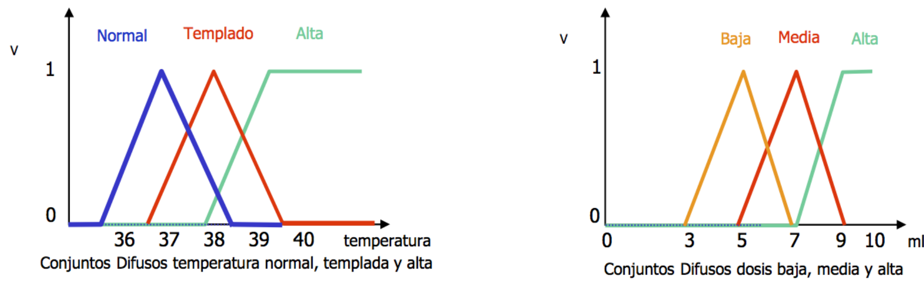

Figure 1.18: Conjuntos difusos del ejemplo

Difusión

◇ Hechos: $temp = 38$

◇ Grados de verdad:

- $normal\,(temp) : 0.33$
- $templado\,(temp) : 1$
- $alta\,(temp) : 0.33$

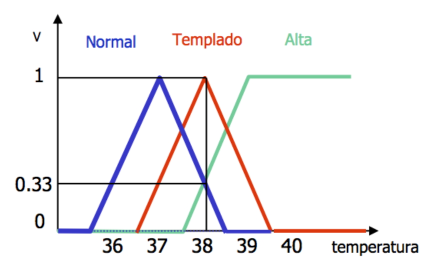

Figure 1.19: Difusión

Inferencia

◇ Reglas:

- $normal\,(temp) \rightarrow baja\,(dosis) : v = 0.33$
- $templada\,(temp) \rightarrow media\,(dosis) : v = 1$
- $alta\,(temp) \rightarrow alta\,(dosis) : v = 0.33$

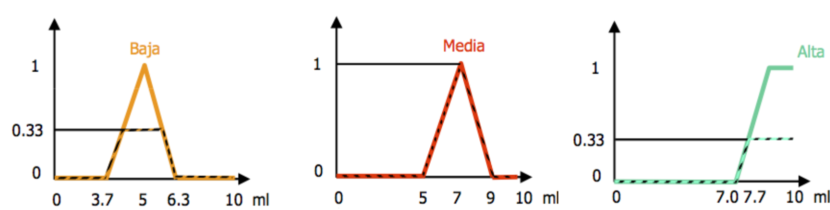

Figure 1.20: Inferencia

Composición

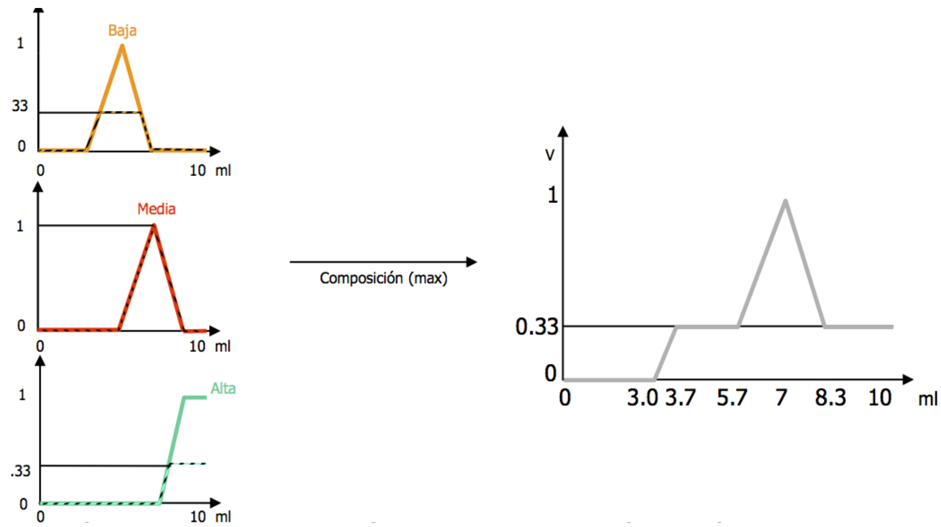

Figure 1.21: Composición

Concisión

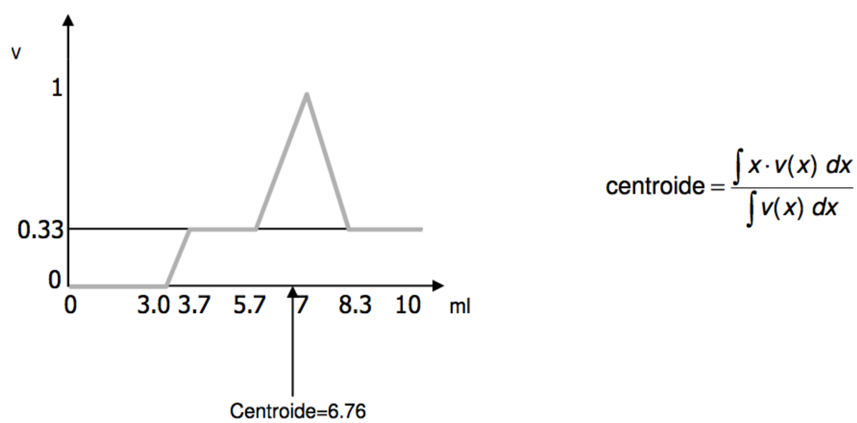

Figure 1.22: Concisión

Por tanto con 38 grados la dosis sería de 6.76 ml.