Capítulo 1

INTRODUCCIÓN

"...la lógica formal es algo más que una tecnología: es la moral del pensamiento y del discurso. Enseña el ejercicio de la honestidad total, tanto en la expresión como en la justificación."

(I. M. Bochenski)

1.1. CONSIDERACIONES GENERALES.

Siempre que hablamos de ingeniería, nos referimos al proyecto y la construcción de *artefactos*. ¿Cuáles son, pues, los artefactos de la Ingeniería del Conocimiento? La respuesta es obvia: los *Sistemas Basados en el Conocimiento* o SBC. Pero ahora tendremos qué aclarar qué entendemos por SBC.

Un SBC es un sistema informático que exhibe ciertas capacidades de razonamiento y resolución de problemas en un dominio limitado como, por ejemplo, el diagnóstico médico o el diseño de circuitos electrónicos. Estas capacidades se deben en gran medida al "conocimiento" que el sistema tiene acerca del dominio, conocimiento que aparece explícitamente representado. El carácter explícito de la representación diferencia los SBC de los sistemas informáticos tradicionales.

Los ingenieros, al menos desde el siglo XVII, han buscado un fundamento científico que apoye el proceso de diseño o justifique las decisiones tomadas a lo largo de él. En este libro proponemos la Lógica como tal fundamento para la Ingeniería del Conocimiento. De esta forma, la tarea del "ingeniero del conocimiento" consistirá en elegir un lenguaje lógico apropiado y usarlo para expresar en él el conocimiento del dominio mediante un conjunto de fórmulas o *axiomas propios*. La Lógica proporcionará un conjunto de métodos de deducción o demostración para manejar y procesar estos axiomas. Por último, empleando las herramientas adecuadas, los axiomas y métodos se implementarán en un sistema informático. El proceso se esquematiza en los dos niveles inferiores de la figura 1.1.

La separación establecida por el logicismo entre métodos de demostración o algoritmos, por un lado, y axiomas propios o estructuras de datos, por el otro, corresponde a la estructuración tradicional de los sistemas expertos en *motor de inferencias*, *base de conocimiento* y *descripción del caso* (figura 1.2.) La descripción del caso estaría constituida por los axiomas referentes a una

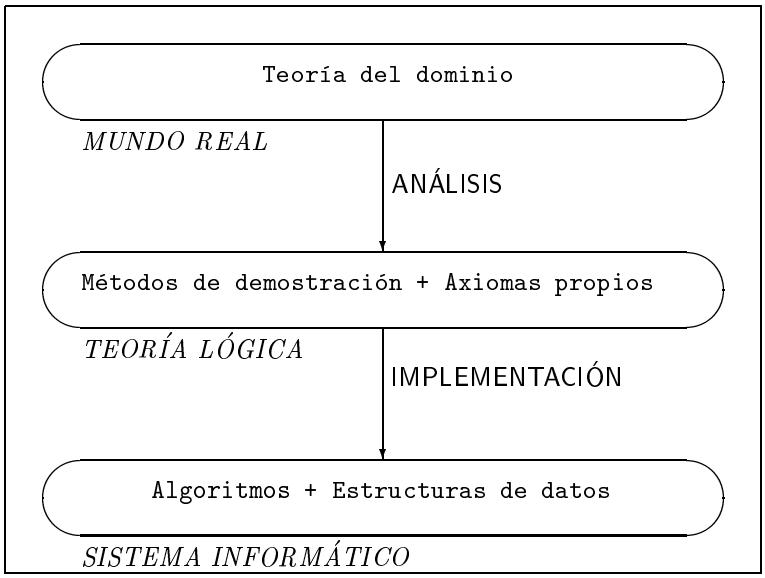

Figura 1.1: La Ingeniería del Conocimiento.

instancia concreta de un problema; por el contrario, la base de conocimientos estaría formada por los axiomas generales que describen el dominio.

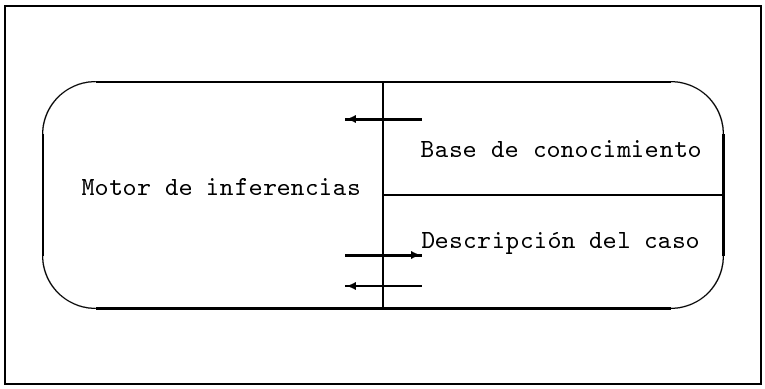

Figura 1.2: Esquema clásico de un sistema experto.

El *enfoque logicista* no es aceptado unánimemente; por ejemplo, algunos investigadores, basándose en la forma de razonar del ser humano, afirman que la lógica deductiva no es demasiado relevante para la Ingeniería del Conocimiento, ya que la Lógica, dicen, no representa un papel importante en el razonamiento cotidiano. La Lógica, en efecto, requiere que entre cada sentencia y la siguiente haya un vínculo apoyado en una deducción simple y perfecta; *"pero en la vida real, el resultado de cada paso de una sentencia a otra es contrastado con nuestra experiencia cotidiana, para ver si parece plausible; y ninguna persona sensata se fiaría de una cadena de razonamiento larga y delgada"* [75]. Sin embargo, esta

crítica "psicologista" no parece enteramente razonable. Aun si los seres humanos no emplearan la lógica en sus razonamientos acerca del mundo real —hipótesis que habría que matizar bastante—, aún entonces no estaríamos obligados a rechazar la lógica como fundamento de los sistemas razonadores artificiales. ¿O acaso un automóvil se mueve de forma parecida a la de un caballo o a la de un ser humano?

Por otra parte, estos investigadores suelen plantear como alternativa a la lógica el empleo de formalismos especiales, como los marcos, las redes semánticas o los sistemas de producción basados en reglas. Pero estos formalismos adolecen de una grave limitación: su *indefinición semántica*. Dicho más claramente, el significado de un conjunto de conocimientos expresados en uno de estos formalismos es indisociable de los algoritmos concretos (motores de inferencias) que lo manipulan; y de esta forma resulta imposible describir de manera abstracta el conocimiento representado. Ello tiene graves problemas a la hora de mantener y actualizar el sistema.

Por ello, modernamente se tiende a reducir estos formalismos a casos particulares de una cierta lógica. De esta forma, un sistema basado en reglas o en marcos será en realidad un sistema basado en la Lógica, cuyos axiomas propios y consultas se expresan en un sublenguaje tal que existen algoritmos eficientes para su manipulación.

Esta tarea de reducción de formalismos prácticos a fragmentos de una lógica puede llegar a ser bastante difícil. Pero *"quienes rechazan los estudios lógicos acaso están intentando rechazar el esfuerzo intelectual que supone la formalización y, por ende, cualquier esfuerzo, como los malos estudiantes que se quejan de la severidad de sus profesores"* ([89], p. 23).

Tampoco es siempre descartable la opción de emplear un lenguaje lógico sin restricciones y un demostrador de propósito general. Por ejemplo, *OTTER* [71], [58], es un demostrador para lenguajes de primer orden con igualdad que se ha empleado con éxito en diversos dominios [72], [99], [98].

1.2. LOS LÍMITES DEL LOGICISMO.

¿De dónde salen los axiomas que se representan en un SBC? Podemos describir el proceso de la siguiente manera.

Primeramente, el ingeniero del conocimiento se fija en una parte del mundo real (el *dominio*) con la que se pretende que el sistema interactúe; luego, inventa objetos, relaciones y funciones que describen esta parte del mundo (la *ontología*) y leyes que la describen (la *teoría del dominio*). Todo esto constituye el *modelo previsto* y esta tarea previa es la *conceptualización*. Por ejemplo, si tenemos ante nosotros una mesa con tres tarugos de madera, la ontología estará compuesta por estos objetos y las propiedades y relaciones que consideremos relevantes (*X es un cubo*, *X está sobre Y*, ...) Hecho esto, el ingeniero elige un lenguaje lógico y en él escribe los axiomas, de forma que el modelo previsto sea realmente un modelo de ellos, en el sentido estricto de la Lógica. De esta manera, *"el significado de la teoría está bien definido: un símbolo no significa más que lo que significa en su modelo más sencillo; por otra parte, las reglas de inferencia correctas preservan la verdad, de forma que las nuevas fórmulas que deducimos siguen cumpliéndose en el modelo previsto"* [43]. Este paso se representa en la parte superior del esquema de la figura 1.1.

Aquí aparece una primera limitación del enfoque logicista: nótese que se pretende representar el modelo previsto, es decir, se pretende expresar un cierto fragmento de lo que es "verdadero" en el mundo real. Pero la Lógica no estudia la verdad, sino la *validez*; que algo sea o no sea verdadero en el mundo real es cosa que deberá determinarse por métodos extralógicos. Claro está que a veces estas cuestiones extralógicas son las realmente importantes; como dice Feyerabend, *"¿qué consuelo obtiene un moribundo cuando oye comentar que su muerte no es lógicamente necesaria?"* [24]

Otra limitación del logicismo se pone de manifiesto al intentar formalizar el razonamiento bajo incertidumbre. En este caso, es innegable que el fundamento proporcionado por la Lógica debe ser complementado al menos con el que ofrece la Teoría de la Probabilidad.

Aún en otro aspecto el razonamiento humano referente a las situaciones del mundo real (razonamiento *de sentido común*) difiere grandemente del razonamiento matemático. En efecto, raras veces es posible partir de un conocimiento total de los aspectos relevantes del estado del mundo; el razonamiento se lleva a cabo partiendo más bien de un conocimiento parcial. Por ello es posible que la aparición de nuevo conocimiento produzca la retirada o modificación de las conclusiones provisionalmente extraídas. Para abordar este problema del *razonamiento revisable* —que no había sido considerado por la Lógica hasta que la Ingeniería del Conocimiento puso de manifiesto su importancia— existen ya diversos formalismos y métodos.

Pero hay quienes realizan una crítica más radical del logicismo, crítica que se extiende a todos los que pretenden representar explícitamente —de una u otra manera— el conocimiento de un sistema. La crítica es la siguiente: un investigador en Inteligencia Artificial divide un problema en dos partes: el componente de Inteligencia Artificial, que resuelve, y el componente no de Inteligencia Artificial, que no resuelve. El principal recurso para realizar esta partición es la abstracción, que permite aparcar todos los aspectos de los problemas realmente difíciles: la percepción y el movimiento [8]. En términos de la figura 1.1, lo que afirman estos críticos es que por encima del nivel etiquetado como "mundo real" hay otro, que es el auténtico mundo real; que lo realmente difícil e importante es que un sistema, actuando en este auténtico mundo real, se comporte de forma inteligente, es decir, flexible y apropiada a sus fines; y que el proceso descrito en la figura, que parte de una conceptualización evidentemente realizada por un ser humano, no es demasiado relevante para ello.

El debate acerca de la relación entre conducta inteligente y representación explícita del conocimiento es ya antiguo; no entraremos en él. Nos limitamos a postular que si tenemos la intención de representar el conocimiento, la mejor forma de hacerlo es emplear la Lógica.

1.3. TÉCNICAS Y TAREAS.

Expondremos ahora una visión panorámica de las técnicas y tareas más notables en la concepción logicista de la Ingeniería del Conocimiento. Lo que decimos a continuación puede verse como una justificación y una guía del contenido de toda la obra.

En el nivel inferior de la figura 1.1 encontramos tareas típicamente informáticas, que habrán de ser abordadas por técnicas de esta clase. De esta forma, el

ingeniero del conocimiento deberá al menos conocer las herramientas informáticas que permiten representar y procesar información expresada en forma lógica. Estas herramientas pueden ser demostradores de propósito general, como el *OTTER* ya mencionado; lenguajes de reglas; lenguajes de marcos ...; En esta obra no mencionaremos ninguna herramienta concreta. Sin embargo, recomendamos vivamente el uso de algún demostrador automático para la realización y comprobación de los ejercicios en los que sea adecuado.

Profundizando un poco más, el ingeniero del conocimiento se verá obligado a veces a implementar sus propias herramientas, sobre todo si desea emplear lógicas no demasiado frecuentes; para ello deberá emplear un lenguaje de programación de propósito general, típicamente *Lisp* o *Prolog*. Por supuesto, en esta obra no abordamos en absoluto este tipo de tareas. Remitimos al lector a los libros de programación, alguno de los cuales trata con relativa extensión estas cuestiones.

El paso del nivel de la teoría lógica al del sistema informático se realiza fundamentalmente a través de los procedimientos de demostración automática, que adecuadamente descritos llegan a ser algoritmos ejecutables por un sistema informático. Los filósofos han soñado con la automatización del razonamiento al menos desde Ramón Llull; pero hasta la aparición del ordenador digital no existió una herramienta adecuada para la tarea. De esta forma, el campo de la *Demostración Automática* es la intersección natural de la Lógica y la Informática, y en una obra como la presente debe ser desarrollado con cierta extensión. Dos son las técnicas de Demostración Automática que principalmente expondremos: la resolución/unificación y el método de los árboles o *tableaux*.

En lo que se refiere a los aspectos puramente lógicos, habrán de considerarse al menos la definición de un lenguaje (*sintaxis*) y la definición de las reglas que permiten dar un valor de verdad a cada fórmula del lenguaje (*semántica*). También es tradicional definir un cálculo formal que a partir de ciertos conjuntos de fórmulas genere nuevas fórmulas llamadas *teoremas* (*axiomática*), lo cual obliga a comprobar que el conjunto de estos teoremas coincide con el conjunto de fórmulas siempre verdaderas (*corrección* y *completitud* de la axiomática). Todo ello es el núcleo de la Lógica, y quizás podríamos remitir al lector a la abundante literatura existente en castellano. Pero no lo haremos así, sino que para cada uno de los lenguajes presentados definiremos y estudiaremos su sintaxis y su semántica; incluso presentaremos alguna de sus axiomáticas y demostraremos su corrección y su completitud. La razón es el papel central que asignamos a la Lógica dentro de la Ingeniería del Conocimiento; de alguna forma, es el hilo conductor elegido para la exposición de la materia. Por otra parte, alguno de los resultados aquí reunidos es difícil de encontrar en la literatura más corrientemente manejada.

¿Cuáles son los lenguajes que se presentan en esta obra? En un primer paso, consideraremos los lenguajes de la lógica matemática clásica. En este volumen aparece el más sencillo de ellos, es decir, el lenguaje del *cálculo de proposiciones*, quedando para los siguientes el lenguaje del *cálculo de predicados* y el del *cálculo de predicados con igualdad*, así como algunos fragmentos decidibles de los mismos, como ciertos *lenguajes de reglas* y *lógicas descriptivas*.

En nuestra opinión, estos lenguajes no agotan el repertorio que debe conocer el ingeniero del conocimiento. En efecto, los seres humanos suelen razonar en su vida cotidiana empleando enunciados que, a diferencia de los matemáticos, no expresan puras afirmaciones eternas; es decir, el enunciado se ve afectado

por un modificador que matiza su significado. Por ejemplo, podemos restringir su ámbito temporal: *En otro tiempo, Virgilio escribió la Eneida.* También podemos referirnos a la "manera de ser verdad" del enunciado: *Necesariamente, el Sol sale por Oriente.* De igual forma, en la vida cotidiana solemos emplear "enunciados anidados", en los cuales una frase (un verbo) cae dentro del ámbito de otra. Por ejemplo, *Sé que Virgilio fue un poeta* (que Virgilio fuera un poeta cae en el ámbito de *sé*) o *Supongo que quieres que hablemos* (un anidamiento a varios niveles). Los razonamientos matemáticos, por el contrario, suelen emplear enunciados sin anidar, que se refieren directamente a "objetos": por ejemplo, *3 es un número primo.*

En general es posible representar el conocimiento de este tipo en un lenguaje de primer orden, bien añadiendo a los predicados un argumento adicional, bien introduciendo uno o varios predicados especiales (operadores extensionales) que tienen como argumentos proposiciones. Sin embargo, esta técnica no es siempre la más simple, ni conceptual ni computacionalmente; en los casos citados más arriba, sería más sencillo recurrir a los lenguajes de las *lógicas modales*, cuyas versiones proposicionales aparecen en el presente volumen.

Por otra parte, el razonamiento clásico no es el único que debe emplear un sistema basado en el conocimiento, como se ha indicado más arriba. El razonamiento revisable ha de ser igualmente considerado; por ello, en este volumen se presentan algunos formalismos revisables adaptados al caso proposicional.

Por último, consideremos las técnicas necesarias para pasar del nivel superior de la figura 1.1 al nivel de la teoría lógica. Si suponemos que las teorías del dominio están expresadas ya en *lenguaje natural*, entonces estaremos en realidad abordando el problema del *Procesamiento del Lenguaje Natural* o *NLP*, problema muy complejo que en su mayor parte queda fuera del ámbito de la presente obra, en la que nos limitamos a dar algunas leves indicaciones acerca de la relación entre ciertas construcciones del lenguaje natural y las correspondientes construcciones lógicas. Si, por el contrario, suponemos que las teorías del dominio no están expresadas en lenguaje natural, sino que yacen implícitas en la mente o en las prácticas de los *expertos* del dominio, las técnicas adecuadas serán las que suelen exponerse en los textos clásicos de Ingeniería del Conocimiento o de Ingeniería del Software, a los que remitimos al lector interesado.

Capítulo 2

REPRESENTACIONES PROPOSICIONALES.

> *"Y si alguien hubiera expuesto bien cuáles son las ideas simples que hay en la imaginación de los hombres, las cuales son los componentes de todo cuanto piensan, y todo el mundo lo aceptara, me atrevería a esperar a continuación una lengua universal, muy fácil de aprender, pronunciar y escribir y, lo principal, que ayudaría al juicio representándole tan distintamente todas las cosas que le sería casi imposible equivocarse... Ahora bien, yo considero que esa lengua es posible y que se puede encontrar la ciencia de que depende".*
>
> (Descartes, Carta al P. Mersenne, 20 noviembre 1629)

Los lenguajes lógicos constan de alfabetos de símbolos y de reglas que permiten generar cadenas de símbolos del alfabeto propuesto en cada caso, cadenas llamadas *fórmulas* de modo genérico. Son lenguajes especialmente diseñados para tratar el razonamiento, y por tanto, poseen estructura de cálculo. En este libro expondremos algunos de estos lenguajes. Entre los más simples, y este capítulo va dedicado a ellos, se hallan los lenguajes del llamado *cálculo de proposiciones*, *lógica de orden cero*, *lógica proposicional* o *cálculo de enunciados*. Un lenguaje de este tipo se limita a proporcionar herramientas formales para tratar proposiciones simples del lenguaje natural (de cuya estructura interna no forman parte otras proposiciones) y partículas que afectan o conectan dichas proposiciones simples (*y*, *o*, ...) formando proposiciones compuestas. Por tanto, estos lenguajes lógicos no proporcionan elementos simbólicos para penetrar en la composición interna de una proposición, es decir, no permiten distinguir expresiones que denoten individuos, propiedades de individuos, etc. Por otra parte, las proposiciones que simbolizan estos lenguajes admiten ser verdaderas o falsas (pero no ambas cosas a la vez). Ejemplos de este tipo de proposiciones son:

La nieve es blanca.
La válvula 1 está abierta.
Juan ama a Mari.
La leche y las patatas son alimentos.
Si llueve el domingo, entonces no iremos al campo.

Las tres primeras proposiciones citadas son *simples*, también llamadas *atómicas* o *átomos*. Las restantes se denominan *compuestas* o *moleculares*.

Los lenguajes del cálculo de proposiciones (en lo sucesivo, *cp*, pese a su sencillez, tienen gran utilidad en la Ingeniería del Conocimiento. De una parte, son un primer paso en el camino de la simbolización; pero, incluso considerados en sí mismos, son de directa aplicación en muchos campos, como el razonamiento cualitativo, los sistemas expertos o la satisfacción de restricciones. Aún más, muchos dominios que a simple vista parecen requerir un lenguaje de primer orden son en realidad susceptibles de una representación proposicional: es el caso de los dominios finitos. Por todo ello comenzamos por el estudio del *cp*. Existe una amplísima bibliografía donde el lector puede profundizar en esta materia y contrastar el nuestro con otros puntos de vista; por diversas razones, de los libros en castellano citaremos [20], [35], [19] cap. 1-4, [67], cap. 1-4 y [1]; y de las obras en inglés [13] cap. 1-8, [57], [31] cap. 1-4 y [90] cap. I-III.

2.1. SINTAXIS.

Los lenguajes del cálculo de proposiciones se definen de diversos modos dependiendo de qué operadores lógicos se tomen como primitivos. Definiremos una gramática que sirve de base a una familia de lenguajes (denominados *lenguajes del cp*) y que consiste en lo siguiente:

$S \Rightarrow$	$\neg S$	(negación)
$S \Rightarrow$	$(S \wedge S)$	(conjunción)
$S \Rightarrow$	$(S \vee S)$	(disyunción)
$S \Rightarrow$	$(S \rightarrow S)$	(implicación)
$S \Rightarrow$	$(S \leftrightarrow S)$	(equivalencia)
$S \Rightarrow$	*ProposiciónAtómica*	
ProposiciónAtomica $\Rightarrow$	$átomo_1 \mid \ldots \mid átomo_n$	

Cuadro 2.1: Sintaxis de los lenguajes *cp*.

Definición 2.1 *Un lenguaje del cp es un lenguaje generado por la siguiente gramática:*

- *Un conjunto de símbolos* terminales *(el* alfabeto*) compuesto por:*
  - *un conjunto finito (no vacío)* Ω *de proposiciones atómicas o átomos.*
  - *los símbolos* lógicos *o* conectivas *:* $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$
  - *los símbolos* auxiliares*:* (,)
- *Un conjunto de símbolos* no terminales*, compuesto por*
  - *un símbolo* inicial S
  - *el símbolo ProposiciónAtómica.*
- *Las* producciones *de la tabla 2.1.*

Si S se analiza en $(S\, op\, S)$ (siendo $op \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$) o en $\neg S$, decimos que op (respectivamente $\neg$) es la *conectiva dominante* de S. A las conectivas $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$ se las suele denominar *conectivas booleanas.*

El par de paréntesis más exterior suele omitirse, así como muchos otros (se supone la precedencia de $\wedge$ y $\vee$ sobre $\rightarrow$ y $\leftrightarrow$, y se puede demostrar la asociatividad de $\wedge$ y $\vee$). Por ejemplo, $((\mathsf{p} \rightarrow (\mathsf{q} \wedge \mathsf{r}))$ se anota $\mathsf{p} \rightarrow \mathsf{q} \wedge \mathsf{r}$; $(\mathsf{p} \wedge (\mathsf{q} \wedge \mathsf{r}))$ y $(\mathsf{p} \wedge (\mathsf{q} \wedge \mathsf{r}))$ se anotan simplemente $\mathsf{p} \wedge \mathsf{q} \wedge \mathsf{r}$; y tanto $((\mathsf{p} \vee \mathsf{q}) \vee \mathsf{r})$ como $(\mathsf{p} \vee (\mathsf{q} \vee \mathsf{r}))$ se anotan $\mathsf{p} \vee \mathsf{q} \vee \mathsf{r}$. Además, merced a la propiedad conmutativa de $\wedge$ y de $\vee$, no importa el orden en que aparezcan los miembros de la conjunción o la disyunción. En realidad, $\wedge$ y $\vee$ pueden ser consideradas como conectivas de *aridad flexible*.

Nótese que la gramática anterior no queda completamente especificada hasta que se fije el conjunto Ω de átomos o proposiciones atómicas. En este libro, representaremos cada átomo mediante una cadena alfanumérica minúscula, es decir, una sucesión finita de letras del alfabeto latino, números y/o guiones que comienza por una letra minúscula. Llamaremos $cp(\Omega)$ al lenguaje generado al tomar como proposiciones atómicas los elementos de Ω. Las cadenas de un lenguaje $cp(\Omega)$ se denominan *fórmulas de orden cero* o *proposiciones*.

Es costumbre en las exposiciones teóricas considerar que Ω es un conjunto enumerable (es decir, infinito numerable) de proposiciones atómicas. Para permitir esto, basta modificar la gramática anterior añadiendo el símbolo no terminal *Número*, y los terminales p y $'$, y sustituyendo la última producción del cuadro 2.1 por

ProposiciónAtomica	$\Rightarrow$	p *Número*
Número	$\Rightarrow$	$'$*Número*
Número	$\Rightarrow$	$'$

Es decir, en el caso de Ω enumerable los átomos serían de la forma p', p'', p''', ...

Escribiremos simplemente cp cuando no deseemos especificar el conjunto Ω, entendiendo con ello que Ω puede ser cualquier conjunto finito o enumerable de proposiciones atómicas.

Ejemplo 2.1 Sea un lenguaje $cp(\Omega)$ donde $\Omega = \{\mathsf{p}, \mathsf{q}, \mathsf{r}\}$. La fórmula $\mathsf{p} \rightarrow \mathsf{q} \wedge \neg \mathsf{r}$ se genera mediante el siguiente proceso: $S \Rightarrow S \rightarrow S \Rightarrow S \rightarrow S \wedge S \Rightarrow S \rightarrow S \wedge \neg S \Rightarrow \mathsf{p} \rightarrow \mathsf{q} \wedge \neg \mathsf{r}$.
$\triangleleft$

Otra forma, bastante usual, de definir el concepto de "fórmula" es el siguiente: dado el vocabulario anterior, el *conjunto de fórmulas del lenguaje* $cp(\Omega)$ es el menor conjunto de cadenas sobre dicho vocabulario que verifica:

1. Todo miembro de Ω es una fórmula del $cp(\Omega)$.
2. Si A es una fórmula del $cp(\Omega)$, entonces $\neg A$ es una fórmula del $cp(\Omega)$.
3. Si A y B son fórmulas del $cp(\Omega)$, entonces $(A \, op \, B)$ es una fórmula del $cp(\Omega)$, donde $op \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$.

El conjunto de fórmulas del $cp(\Omega)$ es precisamente el lenguaje $cp(\Omega)$. Las letras mayúsculas A y B, empleadas en la definición anterior, son *variables metalingüísticas* o *esquemas de fórmulas*, cuya misión es referirse a fórmulas del $cp(\Omega)$. Un esquema de fórmula no forma parte del lenguaje definido, se halla en otro plano lingüístico (el metalenguaje) y permite hablar acerca de expresiones del lenguaje formal, pudiendo ser sustituido por fórmulas cualesquiera del mismo. Emplearemos en éste, y en posteriores capítulos, letras latinas mayúsculas

A, B,... y letras griegas minúsculas φ , ψ , ... para representar fórmulas de los lenguajes utilizados. Para representar conjuntos de fórmulas emplearemos letras griegas mayúsculas Γ, Δ, ... (Todas ellas pueden llevar o no subíndices.) Cualquier otro uso de estas letras se especificará en su momento.

Ejemplo 2.2 Supongamos las siguientes frases en lenguaje natural: *Juan y Pedro son altos. Al menos uno de ellos es sueco. Si Juan es sueco, entonces come salmón y habla sueco. Juan no come salmón.* Vamos a representarlas en un lenguaje *cp*.
Emplearemos las proposiciones atómicas

ja	*Juan es alto*
pa	*Pedro es alto*
js	*Juan es sueco*
ps	*Pedro es sueco*
jcs	*Juan come salmón*
jhs	*Juan habla sueco*

Las frases dadas en lenguaje natural pueden simbolizarse por medio de las siguientes proposiciones:

Juan y Pedro son altos	$\mathsf{ja} \wedge \mathsf{pa}$
Al menos uno de ellos es sueco	$\mathsf{js} \vee \mathsf{ps}$
Si Juan es sueco, entonces come salmón y habla sueco	$\mathsf{js} \rightarrow \mathsf{jcs} \wedge \mathsf{jhs}$
Juan no come salmón	$\neg\mathsf{jcs}$

◁

Definición 2.2 *Una* instancia de sustitución-$cp(\Omega)$ *o* instanciación-$cp(\Omega)$ *de un esquema de fórmula es la fórmula del* $cp(\Omega)$ *resultante de sustituir cada variable metalingüística que aparezca en dicho esquema por una fórmula del* $cp(\Omega)$*, de modo que cada aparición de la misma variable metalingüística sea sustituida por una aparición de una misma fórmula (a esto se le llama "sustituir uniformemente" las variables).*

Si está claro por el contexto en qué lenguaje nos movemos diremos simplemente "instanciación" en lugar de "instanciación-$cp(\Omega)$".

Ejemplo 2.3 Sea un lenguaje con $\Omega = \{\mathsf{p}, \mathsf{q}, \mathsf{r}\}$. Algunas instancias de sustitución de $A \rightarrow (B \rightarrow A \wedge B)$ son:

$\mathsf{p} \rightarrow (\mathsf{q} \rightarrow \mathsf{p} \wedge \mathsf{q})$
$\mathsf{p} \rightarrow (\mathsf{p} \wedge \mathsf{p} \rightarrow \mathsf{p} \wedge \mathsf{p} \wedge \mathsf{p})$
$\mathsf{p} \vee \neg\mathsf{q} \rightarrow (\mathsf{r} \rightarrow (\mathsf{p} \vee \neg\mathsf{q}) \wedge \mathsf{r})$

◁

Definición 2.3 *El conjunto* $Subp(A)$ *de* subfórmulas propias *de* A *es el dado por las siguientes reglas:*

1. $Subp(A) = \varnothing$*, si* A *es un átomo*
2. $Subp(\neg A) = \{A\} \cup Subp(A)$
3. $Subp(A \, op \, B) = \{A, B\} \cup Subp(A) \cup Subp(B)$*, donde* $op \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$

El conjunto $Sub(A)$ *de* subfórmulas *de* A *es* $Sub(A) = \{A\} \cup Subp(A)$.

Por ejemplo, las subfórmulas propias de $\mathsf{js} \to \mathsf{jcs} \wedge \mathsf{jhs}$ son $\mathsf{js}, \mathsf{jcs}, \mathsf{jhs}, \mathsf{jcs} \wedge \mathsf{jhs}$. Nótese que en una misma fórmula puede haber diversas apariciones u ocurrencias de una misma subfórmula, como en $\mathsf{js} \to \mathsf{jcs} \wedge \mathsf{js}$.

Consideremos las fórmulas que son instancias de $A \to B$. La fórmula correspondiente a A se llama *antecedente* de la implicación, la correspondiente a B *consecuente*.

Definición 2.4 *Sean φ, A, B fórmulas cualesquiera de un lenguaje $cp(\Omega)$. Notaremos mediante $\varphi[A/B]^*$ a cualquier fórmula del $cp(\Omega)$ resultante de reemplazar en φ cero o más apariciones de la subfórmula A por B. Notaremos mediante $\varphi[A/B]^+$ a cualquier fórmula del $cp(\Omega)$ resultante de reemplazar en φ una o más apariciones de la subfórmula A por B. Notaremos mediante $\varphi[A/B]$ a la fórmula del $cp(\Omega)$ resultante de reemplazar en φ todas las apariciones de la subfórmula A por B.*

Dicho de otra forma, definimos $\varphi[A/B]^*$ como sigue:

1. Si φ es A, entonces $\varphi[A/B]^*$ es A (si no hay reemplazo) o bien es B (si lo hay).
2. En otro caso,
   a) Si φ es atómica, entonces $\varphi[A/B]^*$ es φ.
   b) Si φ es $\neg\varphi_1$ entonces $\varphi[A/B]^*$ es $\neg(\varphi_1[A/B]^*)$.
   c) Si φ es φ_1 *op* φ_2, donde $op \in \{\wedge, \vee, \to, \leftrightarrow\}$, entonces $\varphi[A/B]^*$ es $\varphi_1[A/B]^*$ *op* $\varphi_2[A/B]^*$.

Definimos $\varphi[A/B]$ como sigue:

1. Si φ es A, entonces $\varphi[A/B]$ es B.
2. En otro caso,
   a) Si φ es atómica, entonces $\varphi[A/B]$ es φ.
   b) Si φ es $\neg\varphi_1$ entonces $\varphi[A/B]^*$ es $\neg(\varphi_1[A/B])$.
   c) Si φ es φ_1 *op* φ_2, donde $op \in \{\wedge, \vee, \to, \leftrightarrow\}$, entonces $\varphi[A/B]$ es $\varphi_1[A/B]$ *op* $\varphi_2[A/B]$.

Por último, el conjunto de fórmulas de la forma $\varphi[A/B]^+$ se obtiene del conjunto de fórmulas de la forma $\varphi[A/B]^*$ eliminando φ.

Ejemplo 2.4 Si φ es $\mathsf{js} \to \mathsf{js} \wedge \mathsf{jcs} \wedge \mathsf{jhs}$, A es js y B es $\mathsf{js} \wedge \mathsf{jcs}$, $\varphi[A/B]^+$ es cualquiera de las siguientes proposiciones:

$\mathsf{js} \wedge \mathsf{jcs} \to \mathsf{js} \wedge \mathsf{jcs} \wedge \mathsf{jhs}$
$\mathsf{js} \to \mathsf{js} \wedge \mathsf{jcs} \wedge \mathsf{jcs} \wedge \mathsf{jhs}$
$\mathsf{js} \wedge \mathsf{jcs} \to \mathsf{js} \wedge \mathsf{jcs} \wedge \mathsf{jcs} \wedge \mathsf{jhs}$

Habría que considerar además como $\varphi[A/B]^*$ la fórmula $\mathsf{js} \to \mathsf{js} \wedge \mathsf{jcs} \wedge \mathsf{jhs}$ y $\varphi[A/B]$ sería únicamente

$\mathsf{js} \wedge \mathsf{jcs} \to \mathsf{js} \wedge \mathsf{jcs} \wedge \mathsf{jcs} \wedge \mathsf{jhs}$.

◁

Ejercicio 2.1 Generar todas las instancias de sustitución del esquema $A \rightarrow A \wedge B$ empleando las siguientes proposiciones

llueve
llueve $\wedge$ nieva
llueve $\rightarrow$ nieva

$\triangleleft$

Ejercicio 2.2 Sea φ la fórmula llueve$\wedge$nieva $\rightarrow$ nieva$\wedge$ventea. Generar $\varphi[A/B]^*$, $\varphi[A/B]^+$ y $\varphi[A/B]$ en cada uno de los casos siguientes:

A es nieva, B es llueve
A es nieva $\wedge$ ventea, B es truena
A es truena, B es hace $-$ frio $\vee$ truena

$\triangleleft$

Ejercicio 2.3 Sea una fórmula φ con m apariciones de $\neg$ y n apariciones de conectivas binarias. ¿Cuántas ocurrencias de átomos hay en φ? ¿Y cuántas ocurrencias de subfórmulas? $\triangleleft$

2.2. SEMÁNTICA.

La semántica pretende dar un significado a las expresiones de un lenguaje. Los significados de las expresiones de los lenguajes del *cp* son los objetos del conjunto $\{1, 0\}$. Intuitivamente, podemos pensar que "1" representa la verdad y "0" la falsedad. Ambos objetos son conocidos como *valores de verdad*. Con esta semántica, cada proposición tiene como significado un valor de verdad y cada conectiva del lenguaje se asocia con una *función de verdad* (una función cuyo dominio son secuencias de valores veritativos y su rango es un valor de verdad). La atribución de estos significados se realiza mediante una *función de interpretación*, que exponemos seguidamente.

Definición 2.5 *Una* interpretación *de un lenguaje* $cp(\Omega)$ *es una función* $I : cp(\Omega) \rightarrow \{1, 0\}$ *tal que para cualesquiera proposiciones* A, B *del* $cp(\Omega)$ *se tiene*

$I(\neg A) = 1$ *si y sólo si* $I(A) = 0$
$I(A \wedge B) = 1$ *si y sólo si* $I(A) = 1$ *e* $I(B) = 1$
$I(A \vee B) = 0$ *si y sólo si* $I(A) = 0$ *e* $I(B) = 0$
$I(A \rightarrow B) = 0$ *si y sólo si* $I(A) = 1$ *e* $I(B) = 0$
$I(A \leftrightarrow B) = 1$ *si y sólo si* $I(A) = I(B)$

Podemos establecer las cláusulas precedentes también así:

$I(\neg A) = 1 - I(A)$
$I(A \wedge B) = min\{I(A), I(B)\} = I(A)I(B)$
$I(A \vee B) = max\{I(A), I(B)\} = I(A) + I(B) - I(A)I(B)$
$I(A \rightarrow B) = 1 - I(A) + I(A)I(B)$
$I(A \leftrightarrow B) = 1 - |I(A) - I(B)|$

De esta forma, los valores asignados por I a cualquier proposición quedan determinados por los valores que I asigna a las proposiciones atómicas, es decir, hay una única interpretación que extiende cualquier especificación de valores de los átomos.

A	$\neg A$
1	0
0	1

Cuadro 2.2: Tabla de verdad de la negación.

A	B	$A \wedge B$	$A \vee B$	$A \to B$	$A \leftrightarrow B$
1	1	1	1	1	1
1	0	0	1	0	0
0	1	0	1	1	0
0	0	0	0	1	1

Cuadro 2.3: Tablas de verdad de las conectivas diádicas.

Las reglas anteriores suelen expresarse de manera gráfica en las *tablas de verdad* de los cuadros 2.2 y 2.3, que expresan la función de verdad asociada a cada conectiva y donde cada fila corresponde a una posible interpretación.

Definición 2.6 *Si $I(\varphi) = 1$, se dice que φ es* verdadera *en I; también se dice en este caso que I es un* modelo *de φ y se escribe $I \vDash \varphi$. Si $I(\varphi) = 0$, se dice que φ es* falsa *en I y se escribe $I \nvDash \varphi$.*

Definición 2.7 *Se dice que φ es una* tautología *(o* válida*) si toda interpretación es un modelo suyo. En este caso se escribe $\vDash \varphi$.*

Definición 2.8 *Se dice que φ es una* contradicción *si no tiene ningún modelo.*

Definición 2.9 *Se dice que φ* implica lógicamente *ψ cuando $\varphi \to \psi$ es una tautología. Se dice que φ* equivale lógicamente *a ψ cuando $\varphi \leftrightarrow \psi$ es una tautología.*

Definición 2.10 *Un* modelo *de Γ es una interpretación en la cual todas las fórmulas de Γ son verdaderas.*

Definición 2.11 *Γ es* insatisfacible *si no tiene ningún modelo. En caso contrario, Γ es* satisfacible*. Se dice que φ es* insatisfacible *si $\{\varphi\}$ es insatisfacible, y φ es* satisfacible *si $\{\varphi\}$ lo es.*

Nótese que "ser satisfacible" significa lo mismo que "tener un modelo".

Definición 2.12 *Se dice que φ es* consecuencia lógica *de Γ si todo modelo de Γ es un modelo de φ. En este caso se escribe $\Gamma \vDash \varphi$.*

Definición 2.13 *Un* argumento *o* argumentación *es un par (Γ, φ), donde Γ es un conjunto finito de fórmulas $\{\varphi_1, \ldots, \varphi_n\}$ $(n \geq 1)$.*

Un argumento se suele representar como

$$\frac{\varphi_1, \ldots, \varphi_n}{\varphi}$$

y leer como "$\varphi_1, \ldots .., \varphi_n$, luego φ". Cada una de las fórmulas $\varphi_1, \ldots, \varphi_n$ se llama *premisa* del argumento y φ es su *conclusión* . Tambien podemos representar el argumento así:

$$\begin{array}{l} \varphi_1 \\ \vdots \\ \varphi_n \\ \hline \varphi \end{array}$$

Definición 2.14 *Un argumento* $(\{\varphi_1, \ldots, \varphi_n\}, \varphi)$ *es* válido *si no existe ninguna interpretación I tal que todas las premisas φ_i $(1 \leq i \leq n)$ sean verdaderas en I y φ sea falsa en I. Es* inválido *en caso contrario.*

En ocasiones, se dice que una interpretación en la que las premisas de una argumentación son verdaderas y la conclusión es falsa es un *contraejemplo* de la validez de dicha argumentación. Similarmente, una interpretación que no es un modelo de una fórmula es un contraejemplo de su validez o tautologicidad.

Fácilmente se obtiene la siguiente

Proposición 2.1 *Sean I una interpretación cualquiera, $\varphi_1, \ldots, \varphi_n, \varphi$ fórmulas cualesquiera. Entonces:*

1. *Toda interpretación es un modelo de $\varnothing$.*
2. *I es un modelo de $\{\varphi_1, \ldots, \varphi_n\}$ si y sólo si lo es de $\varphi_1 \wedge \ldots \wedge \varphi_n$.*
3. *Las siguientes afirmaciones son equivalentes:*
   a) *El argumento $(\{\varphi_1, \ldots, \varphi_n\}, \varphi)$ es válido;*
   b) *$\varphi_1 \wedge \ldots \wedge \varphi_n \rightarrow \varphi$ es una tautología;*
   c) *$\{\varphi_1, \ldots, \varphi_n, \neg\varphi\}$ es insatisfacible;*
   d) *φ es consecuencia lógica de $\{\varphi_1, \ldots, \varphi_n\}$.*
4. *Las siguientes afirmaciones son equivalentes:*
   a) *φ es una tautología;*
   b) *$\neg\varphi$ es insatisfacible;*
   c) *φ es consecuencia lógica de $\varnothing$.*

El método de las *tablas de verdad* proporciona un procedimiento de cómputo cuya aplicación resuelve los problemas referentes al carácter tautológico, satisfacible o insatisfacible de una proposición o conjunto de proposiciones dadas. El método no es más que la extensión al caso de proposiciones compuestas cualesquiera de lo ya visto al definir las conectivas booleanas.

Consideremos el conjunto de símbolos proposicionales (átomos) que intervienen en una fórmula A dada de un determinado lenguaje $cp(\Omega)$; llamemos Ω_A a dicho conjunto. Obviamente, para saber cuál es valor de $I(A)$ sólo se requiere especificar los valores de la interpretación I para los elementos de Ω_A. Tenemos así una *interpretación de A*, que supone simplemente una interpretación del lenguaje $cp(\Omega_A)$.

De esta forma, si A contiene n proposiciones atómicas distintas, el número de interpretaciones de A se reduce a 2^n. Podemos tratar análogamente la interpretación de conjuntos de fórmulas. Si denotamos mediante Ω_Γ a todos los símbolos proposicionales que intervienen en las fórmulas de Γ, una *interpretación de Γ* queda definida meramente por la asignación de valores de verdad a todos los símbolos de Ω_Γ. Esto origina una interpretación del lenguaje $cp(\Omega_\Gamma)$.

Por tanto, podemos construir una tabla con 2^n filas, cada una de las cuales defina una interpretación distinta I_i de A. Calculando todos los valores $I_i(A)$, podremos determinar si A es una tautología, o si es satisfacible, o si es insatisfacible. De la misma forma, por la proposición 2.1.3, podremos determinar si un argumento es válido. A continuación mostraremos un ejemplo ilustrativo.

Ejemplo 2.5 Vamos a decidir por el método de las tablas de verdad la validez del siguiente argumento: *Juan es sueco, o lo es Pedro. Si Juan es sueco, entonces come salmón. Juan no come salmón. Por tanto, Pedro es sueco.* Empleemos las siguientes proposiciones atómicas:

js *Juan es sueco*
ps *Pedro es sueco*
jcs *Juan come salmón*

El argumento será

$\mathsf{js} \vee \mathsf{ps}$
$\mathsf{js} \to \mathsf{jcs}$
$\neg\mathsf{jcs}$
———
ps

En el cuadro 2.4 formamos la tabla de verdad con las $2^3 = 8$ interpretaciones posibles. La columna final sólo contiene apariciones de 1, luego la fórmula que la encabeza es verdadera en todas las interpretaciones posibles y la argumentación dada es válida. ◁

js	ps	jcs	$\mathsf{js} \vee \mathsf{ps}$	$\mathsf{js} \to \mathsf{jcs}$	$\neg\mathsf{jcs}$	$(\mathsf{js} \vee \mathsf{ps}) \wedge$ $(\mathsf{js} \to \mathsf{jcs}) \wedge \neg\mathsf{jcs}$	$(\mathsf{js} \vee \mathsf{ps}) \wedge (\mathsf{js} \to \mathsf{jcs})$ $\wedge\neg\mathsf{jcs} \to \mathsf{ps}$
1	1	1	1	1	0	0	1
1	1	0	1	0	1	0	1
1	0	1	1	1	0	0	1
1	0	0	1	0	1	0	1
0	1	1	1	1	0	0	1
0	1	0	1	1	1	1	1
0	0	1	0	1	0	0	1
0	0	0	0	1	1	0	1

Cuadro 2.4: Ejemplo de tabla de verdad.

Atendiendo a las tablas de verdad, podemos decir entonces que una fórmula φ del *cp* es una tautología si en la última columna de la tabla de verdad de φ sólo aparece el valor 1. En cambio, φ es una contradicción si en la última columna de la tabla de verdad de φ sólo aparece el valor 0; y φ es satisfacible si en dicha columna aparece al menos un 1.

Ejercicio 2.4 Si es posible, dar un modelo de cada una de las siguientes fórmulas:

$\mathsf{p} \to (\mathsf{q} \to \neg\mathsf{p})$
$(\mathsf{p} \leftrightarrow \mathsf{q}) \wedge (\mathsf{r} \leftrightarrow \mathsf{q})$
$\neg(\mathsf{p} \to (\mathsf{q} \to \mathsf{p}))$

◁

Ejercicio 2.5 Estudiar cuáles de los siguientes conjuntos de fórmulas son satisfacibles y cuáles son insatisfacibles:

$\{\mathsf{p} \to \mathsf{q}, \mathsf{p} \to \neg\mathsf{q}\}$
$\{\mathsf{p} \vee \mathsf{q}, \mathsf{p} \leftrightarrow \mathsf{q}, \neg\mathsf{p}\}$
$\{\mathsf{r} \to \neg\mathsf{q}, \mathsf{q} \wedge \mathsf{r}\}$

◁

Proposición 2.2 *Si φ es una tautología, entonces $\varphi[p/B]$ es una tautología, siendo p una proposición atómica y B una fórmula cualquiera.*

DEMOSTRACIÓN: Probemos el contrarrecíproco, es decir, que si $\varphi[p/B]$ no es una tautología, entonces φ tampoco lo es. Supongamos que existe una interpretación I de $\Omega_{\varphi[p/B]}$ tal que $\varphi[p/B]$ es falsa en I. Supongamos además que p no aparece en B (si apareciera, siempre la podríamos sustituir por un nuevo átomo). Si ahora definimos una interpretación I' de Ω_φ tal que $I'(q) = I(q)$ si $q \neq p$ e $I'(p) = I(B)$, es claro que φ es falsa en I', por lo que φ no es una tautología. ◁

Ejercicio 2.6 Estudiar si la siguiente afirmación es verdadera: *Una proposición del cp cuya única conectiva es la equivalencia es una tautología si y sólo si cada proposición atómica aparece un número par de veces.* ◁

Se pueden definir otras conectivas además de las definidas más arriba. Por ejemplo, se suele usar el símbolo $\veebar$ para denotar la *o* exclusiva del lenguaje natural; es decir $\mathsf{p} \veebar \mathsf{q}$ significa *o p o q, pero no ambos a la vez*, lo que en Electrónica suele denominarse p XOR q. El símbolo | (barra de Sheffer) se usa para denotar la anticonjunción, de modo que $\mathsf{p} \mid \mathsf{q}$ se lee *no a la vez p y q*, lo que en Electrónica suele denominarse p NAND q. El símbolo $\downarrow$ (negación conjunta de Peirce) se usa para expresar la conjunción de negaciones, es decir, $\mathsf{p} \downarrow \mathsf{q}$ se lee *ni p ni q*, lo que en Electrónica suele denominarse p NOR q. Las tablas de verdad de estas conectivas son las que aparecen en el cuadro 2.5.

Ejercicio 2.7 Definir las conectivas $\veebar$, | y $\downarrow$ por medio de conectivas del *cp*, es decir, dar fórmulas φ_1, φ_2, φ_3 en las que solamente aparezcan conectivas del *cp* cuyos valores de verdad siempre coincidan respectivamente con los de $A \veebar B$, $A \mid B$ y $A \downarrow B$. ◁

Ejercicio 2.8 ¿Cuántas conectivas n-arias diferentes se pueden definir? Probar que todas ellas se pueden definir a partir de $\neg$ y $\to$. ◁

A	B	$A \veebar B$	$A \mid B$	$A \downarrow B$
1	1	0	0	0
1	0	1	1	0
0	1	1	1	0
0	0	0	1	1

Cuadro 2.5: Tablas de verdad de conectivas adicionales.

2.3. TAUTOLOGÍAS Y ARGUMENTACIONES NOTABLES.

En esta sección trataremos la validez en la lógica proposicional referida tanto a fórmulas (tautologías) como a argumentaciones, y expondremos algunas leyes y formas argumentativas que nos serán útiles más adelante.

Definición 2.15 *Un esquema es* tautológico *si y sólo si, sea cual sea* Ω*, todas sus instanciaciones-cp(Ω) son tautologías.*

Ejemplo 2.6 $A \to A$ es un esquema tautológico (si no lo fuera, habría una instanciación de A que recibiría a la vez el valor 1 y 0, lo cual es imposible). En cambio, $A \to B$ no lo es (por ejemplo, sea $\Omega = \{\mathsf{p}, \mathsf{q}\}$, tenemos que $\mathsf{p} \to \mathsf{q}$ es una instanciación-cp(Ω) suya y no es tautología). $\triangleleft$

Proposición 2.3 *Todos los esquemas de el cuadro 2.6 son tautológicos.*

DEMOSTRACIÓN: Basta aplicar el método de las tablas de verdad; pues, teniendo en cuenta la proposición 2.2, si probamos, por ejemplo, que $(\mathsf{p} \to \mathsf{q}) \leftrightarrow \neg\mathsf{p} \vee \mathsf{q}$ es una tautología, habremos probado que el esquema $(A \to B) \leftrightarrow \neg A \vee B$ es tautológico. $\triangleleft$

$(A \to B) \leftrightarrow \neg A \vee B$	
$(A \to B) \leftrightarrow \neg(A \wedge \neg B)$	
$(A \leftrightarrow B) \leftrightarrow (A \to B) \wedge (B \to A)$	
$\neg(A \vee B) \leftrightarrow \neg A \wedge \neg B$	(ley de De Morgan)
$\neg(A \wedge B) \leftrightarrow \neg A \vee \neg B$	(ley de De Morgan)
$\neg\neg A \leftrightarrow A$	(doble negación)
$A \wedge B \leftrightarrow B \wedge A$	(conmutatividad)
$A \wedge (B \wedge C) \leftrightarrow (A \wedge B) \wedge C$	(asociatividad)
$A \wedge (B \vee C) \leftrightarrow (A \wedge B) \vee (A \wedge C)$	(distributividad)
$A \wedge (A \vee B) \leftrightarrow A$	(absorción)
$A \wedge A \leftrightarrow A$	(idempotencia)
$A \vee B \leftrightarrow B \vee A$	(conmutatividad)
$A \vee (B \vee C) \leftrightarrow (A \vee B) \vee C$	(asociatividad)
$A \vee (B \wedge C) \leftrightarrow (A \vee B) \wedge (A \vee C)$	(distributividad)
$A \vee (A \wedge B) \leftrightarrow A$	(absorción)
$A \vee A \leftrightarrow A$	(idempotencia)

Cuadro 2.6: Tautologías notables.

Proposición 2.4 *Todas las argumentaciones de el cuadro 2.7 son válidas.*

DEMOSTRACIÓN: En cuanto al *modus ponens*, *modus tollens*, silogismo hipotético y resolución, basta aplicar el método de las tablas de verdad.

La validez de la sustitución de equivalentes, sin embargo, ha de probarse de otra manera. La demostraremos mediante una técnica muy común en Lógica: la *inducción sobre el número de conectivas de una fórmula* φ.

Paso base: supongamos primeramente que el numero n de conectivas de φ es $n = 0$, es decir, que φ es atómica. Es obvio que si φ es A entonces $\varphi[A/B]^*$ es o bien B, en cuyo caso la argumentación se reduce al caso trivial "$A \leftrightarrow B$, luego $A \leftrightarrow B$", o bien es φ, y tenemos otro argumento trivialmente válido "$A \leftrightarrow B$, luego $\varphi \leftrightarrow \varphi$". Si φ no es A, tenemos también este último caso.

Paso de inducción: supongamos ahora que φ es compuesta y tiene n conectivas, siendo $n > 0$, y que la proposición se cumple para todas las fórmulas con menos de n conectivas (a esto se le llama *hipótesis de inducción* o *hipótesis inductiva* o *supuesto de inducción*). Probaremos seguidamente que el resultado vale para la fórmula φ citada. Supongamos una interpretación con $I(A \leftrightarrow B) = 1$. Entonces tendremos $I(A) = I(B)$. Así pues, sea φ de la forma φ_1 *op* φ_2, donde *op* $\in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$. Por la hipótesis de inducción son válidas las argumentaciones "$A \leftrightarrow B$, luego $\varphi_1 \leftrightarrow \varphi_1[A/B]^*$" y "$A \leftrightarrow B$, luego $\varphi_2 \leftrightarrow \varphi_2[A/B]^*$"; por tanto, dado que $I(A \leftrightarrow B) = 1$ tendremos $I(\varphi_1 \leftrightarrow \varphi_1[A/B]^*) = 1$ e $I(\varphi_2 \leftrightarrow \varphi_2[A/B]^*) = 1$, es decir, $I(\varphi_1) = I(\varphi_1[A/B]^*)$ e $I(\varphi_2) = I(\varphi_2[A/B]^*)$ y, teniendo en cuenta que el valor de φ_1 *op* φ_2 está determinado únicamente por los valores de sus partes, tenemos $I(\varphi_1 \text{ } op \text{ } \varphi_2) = I(\varphi_1[A/B]^* \text{ } op \text{ } \varphi_2[A/B]^*)$, es decir, $I((\varphi_1 \text{ } op \text{ } \varphi_2) \leftrightarrow (\varphi_1[A/B]^* \text{ } op \text{ } \varphi_2[A/B]^*)) = 1$. Análogamente se razona cuando φ es de la forma $\neg\varphi_1$.

Con esto hemos probado que cualquier interpretación en la que la premisa de la argumentación sea verdadera lo es igualmente la conclusión. Por tanto, hemos probado que, sea cual sea φ, la argumentación por sustitución de equivalentes es válida, q.e.d. $\triangleleft$

A $A \rightarrow B$ ——— B	(modus ponens)	$\neg B$ $A \rightarrow B$ ——— $\neg A$	(modus tollens)
$A \rightarrow B$ $B \rightarrow C$ ——— $A \rightarrow C$	(silogismo hipotético)	$A \vee B$ $\neg A \vee C$ ——— $B \vee C$	(resolución)
$A \leftrightarrow B$ ——— $\varphi \leftrightarrow \varphi[A/B]^*$	(sustitución de equivalentes)		

Cuadro 2.7: Argumentaciones válidas notables.

Ejercicio 2.9 Demostrar las siguientes proposiciones:

1. Sea A una fórmula en la que las únicas conectivas son $\neg$, $\wedge$ y $\vee$. Sea A' el resultado de sustituir todas las conjunciones de A por disyunciones, y todas las disyunciones de A por conjunciones. Entonces, A' es válida si y

Además, φ es una tautología, por lo que $I(\varphi) = I^*(\varphi) = 1$ y, por tanto, $\varphi^I = \varphi^{I^*}$. Por otro lado, como $I(A_n) = 1, A_n^I$ es A_n y, a partir de i), tenemos $A_1^I, \ldots, A_n^I, A_n \vdash \varphi$. Como $I^*(A_n) = 0$, $A_n^{I^*}$ es $\neg A_n$, así que, a partir de ii), tenemos $A_1^I, \ldots, A_{n-1}^I, \neg A_n \vdash \varphi$. Por el teorema de deducción tenemos: $A_1^I, \ldots, A_{n-1}^I \vdash A_n \to \varphi$ y $A_1^I, \ldots, A_{n-1}^I \vdash \neg A_n \to \varphi$. Por lo tanto, $A_1^I, \ldots, A_{n-1}^I \vdash \varphi$ (por el ejercicio 2.11(3) y la proposición 2.5(6 y 9)). Si $n = 1$, entonces $\vdash \varphi$. Si $n > 1$, entonces definimos una interpretación I^{**} de forma que $I(A_k) = I^{**}(A_k)$, para todo $k < n - 1$; $I(A_{n-1}) = 1$, $I^{**}(A_{n-1}) = 0$. Y llegamos de forma parecida a $A_1^I, \ldots, A_{n-2}^I \vdash \varphi$. Repitiendo este proceso obtenemos finalmente $\vdash \varphi$, q.e.d. $\triangleleft$

2.5. LENGUAJE NATURAL Y CÁLCULO DE PROPOSICIONES.

La mayoría del conocimiento explícito manejado por los seres humanos se expresa en un *lenguaje natural*. Si queremos representar este conocimiento en un lenguaje lógico, tendremos que "traducirlo" de uno a otro. Podría pensarse que este proceso es fácil; pero nada más lejos de la realidad. En primer lugar, los hablantes emplean el lenguaje natural para muchas funciones, además de enunciar proposiciones; y, aún en este último caso, los lenguajes naturales buscan la eficacia y la economía expresiva antes que el mero reflejo de la estructura lógica subyacente. Esto lleva a menudo a la ambigüedad, que sólo se puede resolver acudiendo al contexto general o parcial, o incluso solicitando del hablante la desambiguación. En resumen, no se puede establecer una correspondencia sencilla entre una frase en lenguaje natural y su representación lógica.

Por otra parte, a veces la lógica clásica o matemática no es el instrumento más adecuado para recoger el significado de un enunciado en lenguaje natural. De una parte, hay que señalar que el valor de verdad de un enunciado no es siempre un valor nítido, es decir, un enunciado no es siempre absolutamente verdadero o absolutamente falso; piénsese en el valor de verdad de *Pepe es alto* o en el de *Juan baila bien*. Para estos casos, la llamada *Lógica Difusa* proporciona una herramienta más adecuada.

Hay otra razón, quizás más difícil de detectar, por la cual un uso ingenuo de la lógica clásica no refleja siempre adecuadamente lo que se desea expresar mediante un enunciado. En el *cp* de la lógica matemática, el significado (verdadero o falso) de una fórmula compuesta es una función total (dada por la conectiva empleada) de los significados de sus fórmulas componentes. Sin embargo, en muchas ocasiones un hablante profiere enunciados cuyo significado no tiene este carácter "veritativo-funcional". Por ello, el ingeniero del conocimiento, enfrentado a la simbolización de un enunciado cuyo significado lógico sea dudoso, debe analizar en qué condiciones de verdad o falsedad de los enunciados componentes puede afirmarse la verdad o falsedad del enunciado completo.

2.5.1. CONECTIVAS Y LENGUAJE NATURAL.

Negación.

La lectura convencional del signo $\neg$ es *no*. Las correspondientes estructuras del lenguaje natural son más complejas (para el castellano, puede consultarse

[91]), como muestra un sencillo ejemplo. Sea p *la válvula 1 está cerrada*. Entonces $\neg\mathsf{p}$ se leería *no la válvula 1 está cerrada*, lo cual no suena nada bien. Por ello se emplean convencionalmente otras expresiones como lectura o interpretación de $\neg$, como *no es cierto que*, *es falso que* o *no es el caso que* prefijadas al enunciado que se quiere negar, de forma que tendríamos *no es cierto que la válvula 1 está cerrada*, *es falso que la válvula 1 está cerrada*, o *no es el caso que la válvula 1 está cerrada*. Pero sin duda lo normal sería decir *la válvula 1 no está cerrada*.

Aún más, el lenguaje natural proporciona muchas veces pares de palabras para representar un concepto y su negación. En este ejemplo, sería quizás más natural decir *la válvula 1 está abierta*, y sería necesario conceptualizar *abierto* y *cerrado* como la afirmación y la negación de una sola propiedad.

Conjunción.

La lectura convencional del signo $\wedge$ es *y*. Debe notarse que en lenguaje natural (ver, por ejemplo, [12]) la conjunción puede representarse con muchas otras palabras: *ni, pero, sino, aunque, sin embargo, antes bien,*.... Aún más, la mera yuxtaposición sirve para expresar la misma idea: *Juan corre, Pepe grita, Mari llora*. Muy frecuentemente se emplea la conjunción para unir no enunciados completos, sino elementos de un enunciado. A la hora de representar éste, es necesario descomponerlo en varios. Por ejemplo, nadie dice *Mari es alta y Pepi es alta*, sino *Mari y Pepi son altas*. La representación más adecuada en *cp* de este último enunciado es de la forma $\mathsf{p}\wedge\mathsf{q}$. Por otra parte, la palabra *y* añade a veces un significado de sucesión temporal no recogido por la conectiva formal $\wedge$: *Juan llega a casa y bebe una cerveza* no equivale a *Juan bebe una cerveza y llega a casa*. El sentido de esta *y* no es veritativo-funcional, en este caso *y* tiene el sentido de *y luego (después)*. La palabra *y* puede servir incluso para construir un enunciado condicional; considérese el caso *Dime con quién andas y te diré quién eres*.

Disyunción.

La lectura convencional del signo $\vee$ es *o*. Sin embargo, la palabra *o* es esencialmente ambigua en castellano, ya que tiene dos significados diferentes entre los cuales sólo el contexto puede —y no siempre— ayudarnos a elegir. Los lógicos hablan de "disyunción inclusiva" y "disyunción exclusiva" para referirse a ambos significados. El significado de la disyunción inclusiva es el dado por la tabla de verdad asignada a $\vee$. El significado de la disyunción exclusiva de A y B es el dado por la tabla de verdad de $(A \veebar B)$ (tabla 2.5).

Decidir si una determinada disyunción es inclusiva o exclusiva puede ser complicado en ocasiones. Sin embargo, hay casos en los que está muy claro que la partícula *o* se interpreta de modo exclusivo, como en *la luz está apagada o encendida*, o en *el acusado es inocente o culpable*, o bien en *el cumpleaños de Marta cae en lunes o en martes*. También hay casos en los que la partícula *o* se interpreta claramente de modo inclusivo, por ejemplo, cuando se dice *2 es menor que 4 o que 5*. Pero como ya hemos visto, hay dificultades en la interpretación de la partícula *o*, por lo que el lenguaje coloquial proporciona expresiones que permiten desambiguar su sentido al reforzar el carácter exclusivo o inclusivo de la misma. Así, expresiones como *y/o*, *o ambos* acentúan el carácter inclusivo. Por ejemplo, *Se abre el periodo de matrícula para estudiantes de Física y/o Ma-*

temáticas; *Ana habla inglés o francés o ambos idiomas*. En cambio, expresiones como *una de dos, pero no ambos* acentúan el carácter exclusivo. Por ejemplo, *Una de dos: o esta tarde vamos al cine o al teatro. O nieva o hace sol, pero no ambas cosas a la vez*.

También es muy frecuente emplear la disyunción para unir elementos de un enunciado. Nadie dice *Juan aprobará Papiroflexia o Juan aprobará Paleografía*, sino *Juan aprobará Papiroflexia o Paleografía*.

Condicional.

Confesémoslo abiertamente: la relación de la conectiva $\rightarrow$ (a la que suele llamarse *implicación material*) con el significado de los enunciados condicionales del lenguaje natural (*condicional natural*) no es tan estrecha como podría pensarse. Ello, al representar el conocimiento de sentido común, es causa de innumerables problemas, aunque éstos raras veces se planteen al tratar con enunciados matemáticos condicionales.

Los condicionales naturales no suelen ser veritativo-funcionales, es decir, la verdad del enunciado no está determinada únicamente por la verdad del antecedente y la del consecuente: habitualmente, hay además un matiz causal. Por ejemplo, cuando un hablante afirma *si ahora llueve entonces la calle está mojada*, está afirmando que la lluvia es causa de la humedad de las calles. De esta forma, un enunciado como *si el Tajo pasa por Toledo, entonces el Danubio pasa por Viena* resulta muy extraño: un hablante tendería a considerarlo falso o sin sentido, ya que el nexo causal entre ambos acontecimientos es difícil de concebir.

Aún más claro resulta este carácter no veritativo-funcional si consideramos los condicionales contrafácticos del lenguaje natural, es decir, los que enuncian —en subjuntivo— una condición irreal, que ya sabemos falsa; por ejemplo, *si la válvula 1 estuviera abierta, el depósito 3 estaría lleno*. Estos enunciados son frecuentes en el lenguaje natural y los hablantes no los consideran sin sentido, sino que suelen asignarles un valor de verdad, a veces "verdadero", a veces "falso". Sin embargo, simbolizándolos con la conectiva $\rightarrow$ todos los enunciados contrafácticos son verdaderos, ya que su antecedente es falso; lo cual parece una simplificación excesiva.

¿Y qué ocurre si la condición falsa se enuncia en indicativo, como en: *Si el Ebro pasa por Toledo, entonces el Danubio pasa por Viena*? En estos casos el hablante considera casi siempre que el enunciado condicional no tiene sentido (hay excepciones: *Si Mari aprueba este examen, entonces soy el Papa* es una forma enfática de negar el antecedente a partir de la presunción de falsedad del consecuente).

¿Cuál es pues la justificación de la tabla de verdad asignada a la conectiva $\rightarrow$? Para encontrarla debemos considerar los enunciados universales del lenguaje de la Matemática. Nótese, en primer lugar, que el concepto de causa no se emplea directamente en el mundo de las Matemáticas; por tanto, la primera de las observaciones anteriores no es aplicable. Consideremos entonces un enunciado universal, por ejemplo, *todos los números divisibles por cuatro son pares*. Este enunciado, como veremos más adelante, se puede parafrasear como el condicional *para todo número, si ese número es divisible por cuatro, entonces es par*. Cualquier matemático afirmará sin la menor vacilación que el enunciado es verdadero. Ya que es verdadero para todo número, debe serlo también para el 3; así que *si 3 es divisible por cuatro, entonces 3 es par* debe ser verdadero; luego

antecedente falso y consecuente falso hacen un condicional verdadero. Consideremos ahora el enunciado *todos los números divisibles por cinco son pares*, es decir, *para todo número, si ese número es divisible por cinco, entonces es par*. El enunciado universal es falso, así que tiene que ser falso para algún número concreto. Un mátemático nos señalaría, por ejemplo, el mismo número 5; así que *si 5 es divisible por cinco, entonces 5 es par* es falso; luego antecedente verdadero y consecuente falso hacen un condicional falso.

La adición de la palabra *si* no es la única manera en que se puede formar en castellano el antecedente de una condicional natural. Las gramáticas (puede consultarse, por ejemplo, [77]) citan otras, como el empleo de *cuando, como, de* + infinitivo, gerundio, participio, *siempre que, ya que, caso de que, con tal que, con solo que, con que*, y de oraciones de relativo en subjuntivo. Por ejemplo, *el bien que venga, para todos sea*. Por su parte, los matemáticos suelen emplear la perífrasis *A es condición suficiente de B* para enunciar sin ambigüedad el condicional *si A, entonces B*; y *A es condición necesaria de B* para *si B, entonces A*. Otro grupo de palabras sirve para formar condicionales negativas, es decir, enunciados en los que se establece una excepción: *a menos que, salvo, excepto,...* Por ejemplo, *Pepe resulta simpático, a menos que esté borracho*, o de otra forma, *Pepe resulta simpático, salvo si está borracho* o, de otra forma, *Si Pepe no resulta simpático, entonces está borracho* o, finalmente, *Si Pepe no está borracho, entonces resulta simpático*. Si b simboliza *Pepe está borracho* y s simboliza *Pepe resulta simpático*, la simbolización de estos enunciados será $\neg\mathsf{s} \to \mathsf{b}$ o, equivalentemente, $\neg\mathsf{b} \to \mathsf{s}$.

A veces es difícil saber si tales condicionales negativas se deben interpretar más bien como equivalencias. En el ejemplo anterior, ¿qué ocurre si Pepe está borracho y pese a ello resulta simpático? ¿Es verdadero el enunciado —sería una implicación material— es indeterminado —sería una condicional natural— o es falso — sería una equivalencia?

Lo cual nos lleva a la cuestión de cómo se expresan las equivalencias en lenguaje natural. Los matemáticos suelen emplear las frases hechas *A es condición necesaria y suficiente de B* o bien *A si y sólo si B*. La gente corriente no suele enunciar equivalencias, y si tiene que hacerlo recurre a expresiones perifrásticas: *siempre que A, B, y siempre que B, A*; etc. Pero, como hemos mencionado, desde el punto de vista pragmático las condicionales a veces son realmente equivalencias: *si apruebas todas tus asignaturas este año, te compraré una moto* parece que lleva consigo implícitamente *si no apruebas todas tus asignaturas este año, no te compraré una moto*.

Ejercicio 2.18 Discutir el carácter veritativo-funcional de los siguientes enunciados:

Vine, vi, vencí.
Si hubieras ahorrado, no te verías en la miseria.
Si Juan me apreciara, me prestaría cien euros.
Mari baila sevillanas y acude al Rocío, ya que es andaluza. ◁

Ejercicio 2.19 Simbolizar en un lenguaje proposicional los siguientes enunciados:

Es verdad, sin embargo, que no deja de ser bonito el traje de Mari.
Pero es falso que el traje de Mari no sea bonito.

De ninguna manera puede dejar de reconocerse que el traje de Mari no es bonito.

Pepe y Juan son pobres pero honrados, aunque tienen ocasiones de lucrarse ilegalmente.

O te tomas la sopa o te quedas sin propina esta semana.

El señor X *pagará el soborno, siempre que su asunto se arregle satisfactoriamente.*

La condición necesaria y suficiente para aprobar la asignatura es obtener más de 4.5 en teoría y más de 4.9 en prácticas, o bien más de 7.5 en teoría.

Pepe no viste ropa deportiva, aunque está corriendo.

Pepe corre sólo cuando tiene prisa, a menos que esté calzado cómodamente.

No hay contrato sino cuando concurren los requisitos siguientes: 1. Consentimiento de los contratantes. 2. Objeto cierto que sea materia del contrato. 3. Causa de la obligación que se establezca.

Una señal de estar enamorado —que nunca falta ni engaña— es poner cara de tonto al pensar en una persona.

Al menos una de las siguientes afirmaciones es verdadera: a) Juan y su hermano son escritores; b) Ni Juan ni su hermano son escritores; y al menos una de las siguientes es falsa: a) Si Juan es escritor, escribe en español; b) O Juan no es escritor, o no lo es su hermano. ◁

2.5.2. REGLAS, CONOCIMIENTO Y LÓGICA.

En muchos ámbitos de la actividad humana es frecuente el enunciado y uso de *reglas*. En la Ingeniería del Conocimiento el concepto de regla ha tenido una especial importancia, hasta el extremo de que "sistema experto" y "sistema basado en reglas" fueron considerados prácticamente sinónimos. Una regla tiene la estructura

`SI` *parte-izquierda* `ENTONCES` *parte-derecha*,

que coincide superficialmente con la de un condicional material. Pero, como se ha indicado más arriba, no siempre los condicionales del lenguaje natural pueden interpretarse como implicaciones materiales.

En el caso de las reglas, una primera distinción separa las reglas *declarativas* de las reglas *procedimentales*. Las primeras pueden interpretarse como enunciados, es decir, tienen un valor de verdad (verdadero o falso) en cada situación del mundo. Las segundas no tienen este carácter enunciativo, sino que expresan órdenes o instrucciones. Más concretamente, es la parte derecha de estas reglas la que, al menos en primera aproximación, no es un enunciado, sino un mandato. Por tanto, estas reglas procedimentales no deben confundirse con implicaciones materiales, y su análisis deberá llevarse a cabo mediante herramientas complementarias al *cp*. Por ejemplo,

(1) `SI` la cláusula vacía pertenece a C, `ENTONCES` acabar con éxito.
(2) `SI` la temperatura es mayor de 70 grados, `ENTONCES` activar aspersores.

La regla (1) es parte de un algoritmo, y como tal puede entenderse como una instrucción proporcionada a quien ha de ejecutarlo (un ser humano o una máquina.) La regla (2) sirve para expresar una orden que ha de ejecutarse cuando se dan ciertas circunstancias externas al agente ejecutor. No tiene sentido decir en una situación dada que estas reglas son "verdaderas" o "falsas"; lo que podremos decir es que han sido obedecidas o que no lo han sido.

En cuanto a las reglas declarativas, cabe considerar tres clases atendiendo al nexo causal subyacente,como muestran los siguientes ejemplos:

(3) `SI` x es múltiplo de 4, `ENTONCES` x es múltiplo de 2.
(4) `SI` ha llovido, `ENTONCES` la calle está mojada.
(5) `SI` la calle está mojada, `ENTONCES` ha llovido.

Las reglas como la (3) son independientes de cualquier nexo causal; expresan relaciones abstactas entre enunciados y correponden —estas sí— a los condicionales materiales del *cp*.

Las reglas como la (4) tienen en su parte derecha una *causa* y en su parte izquierda un *efecto*. O sea, (4) resultaría de un enunciado como (4') de la forma

(4') *El hecho de que llueva es causa del hecho de que la calle esté mojada.*

y del *principio de causalidad*, que podríamos enunciar como sigue: *Si el hecho A es causa de B y se da A, entonces se dará B*; principio que, como parte del conocimiento de *sentido común*, es aceptado generalmente por los seres humanos.

Por el contrario, las reglas como (5) tienen en su parte derecha un efecto y en su parte izquierda una causa. Es decir, (5) resultaría del enunciado (4') anterior y del *principio de abducción*, que se enunciaría como: *Si el hecho A es causa de B y se da B, entonces se ha dado A*. A diferencia del anterior, este principio —enunciado de manera absoluta— repugna al sentido común, y sería rechazado por la generalidad de los seres humanos. Sin embargo, una regla abductiva particular como la (5) no resulta tan chocante. ¿Cuál es, pues, su justificación? Podemos encontrar parte de ella en el principio de *razón suficiente*: *Si los hechos $A_1, \ldots, A_n$ son* todas *las posibles causas de B, y se da B, entonces se ha dado alguno de los $A_1, \ldots, A_n$*. Pero ello no basta para justificar la abducción, pues la verdad de una disyunción $A_1 \vee \ldots \vee A_n$ no implica la verdad de ninguna de las alternativas $A_1, \ldots, A_n$; a menos que por otra parte se tenga por ejemplo $\neg A_2, \ldots, \neg A_n$, lo que sí nos permitiría deducir A_1.

En realidad, el concepto de causa, aunque parte importante de la estructura conceptual del sentido común, es escurridizo y difícil de atrapar en las mallas de la formalización lógica. Una forma de evitar el embrollo teórico apuntado en los párrafos anteriores es abandonar el concepto ingenuo de causa y la estructura de regla y representar el conocimiento de las relaciones existentes en el dominio —que casi siempre será incierto— en forma de probabilidades condicionadas estructuradas en una *red bayesiana*.

Pero el estudio de las redes bayesianas excede del ámbito de la presente obra. Volviendo a la representación basada en reglas, debemos decir que en los llamados "sistemas expertos", sobre todo en los más antiguos, las reglas procedurales, causales y abductivas aparecían confundidas pues

(i) en un mismo módulo aparecían simultáneamente reglas de varios tipos;
(ii) la sintaxis no permitía distinguirlas;
(iii) por tanto, los algoritmos de manejo de reglas las trataban de manera uniforme.

Ello originaba problemas a la hora de estructurar, depurar y mantener las "bases de conocimiento". La Ingeniería del Conocimiento actual, por estas razones, de una parte ha depurado el concepto ingenuo de regla, y de otra ha ampliado el repertorio de herramientas de representación.

Ejercicio 2.20 Discutir la naturaleza de las siguientes reglas:

Si alguno mata a otro, que sea castigado.
Si alguno mata a otro, será castigado.
Si alguno mata a otro, es reo de homicidio.
Si hay tos y fiebre, entonces habrá gripe.
Si hay gripe, entonces habrá tos y fiebre.
Si se pulsa el interruptor 1, el circuito A queda activado.
Si la presión sube más de un 10 %, abrir la válvula 1.
Si un alumno supera el examen, es que sabe la materia.
Si un alumno sabe la materia, supera el examen. ◁

2.6. DEMOSTRACIÓN AUTOMÁTICA: RESOLUCIÓN.

La demostración automática tiene como objetivo la búsqueda y el estudio de algoritmos para resolver ciertos problemas lógicos. En relación con el *cp*, estudiaremos ahora algunos métodos algorítmicos para tratar lo siguiente:

Dada una argumentación $(\{\varphi_1, \ldots, \varphi_n\}, \varphi)$*, determinar si es o no es válida.*

O bien

Dada una fórmula φ*, determinar si es o no es una tautología.*

Por los teoremas de corrección y completitud, ello es equivalente a plantearse:

Dada una argumentación $(\{\varphi_1, \ldots, \varphi_n\}, \varphi)$*, determinar si* φ *se deduce de* $\varphi_1, \ldots, \varphi_n$*.*

O bien

Dada una fórmula φ*, determinar si es o no un teorema.*

El método de las tablas de verdad, como se ha visto, proporciona un procedimiento de cómputo cuya aplicación siempre resuelve estos problemas. Cuando una clase de problemas cuentan para su solución con un procedimiento así, se suele decir que es una clase de problemas *decidible*

Aunque las tablas de verdad solucionan satisfactoriamente el problema, veremos otros dos métodos de demostración automática para el *cp*: *resolución* y *árboles* o *tablas lógicas*. La utilidad de los mismos aparecerá más claramente al tratar el Cálculo de Predicados y las lógicas modales.

El método de resolución, debido a Robinson [86], es un procedimiento automático que impone la restricción de operar sólo con fórmulas que posean una forma determinada, a saber, que sean *cláusulas*. Esto requiere transformar cada fórmula en una *forma normal conjuntiva*. El método usa una única regla de inferencia, llamada *regla de resolución*. Comenzaremos exponiendo la noción de "forma normal conjuntiva" y seguidamente trataremos el método de resolución para el *cp*.

2.6.1. FORMA NORMAL CONJUNTIVA.

Definición 2.31 *(Literal). Un* literal afirmado *es una proposición atómica. Un* literal negado *es una proposición atómica negada. Un* literal *es un literal afirmado o un literal negado. Si A es una proposición atómica, A y* $\neg A$ *se dice que son* literales opuestos.

Sea L un literal. Su *opuesto* lo denotaremos mediante $\neg L$, lo que significa que si L es un átomo p, entonces su opuesto es $\neg p$; pero si es $\neg p$, entonces $\neg L$ es p.

Definición 2.32 *(Cláusula vacía). Sea* p *un átomo cualquiera de* Ω. *La* cláusula vacía *es la fórmula* $\mathsf{p} \wedge \neg \mathsf{p}$.

Representaremos la cláusula vacía mediante el símbolo $\Box$. Es decir, $\Box$ y el símbolo $\bot$ (*falsum*) introducido en el ejercicio 2.10 tienen el mismo significado.

Definición 2.33 *Una* cláusula *es bien la claúsula vacía, bien una disyunción sin repeticiones de uno o más literales, es decir, una fórmula de la forma* $L_1 \vee L_2 \vee \ldots \vee L_n$, *donde cada* L_i *(*$1 \leq i \leq n$*) es un literal, o, abreviadamente* $\bigvee_{i=1,\ldots,n} L_i$, *y si* $i \neq j$ *entonces* $L_i \neq L_j$.

Una cláusula con un único literal se denomina *cláusula unitaria* . Una cláusula también se puede representar en forma de conjunto de literales: $\{L_i | i = 1, \ldots, n\}$. (Nótese que de cualquier disyunción C se pueden eliminar las ocurrencias repetidas de literales en tiempo polinómico en su número total de ocurrencias de literales). Las siguientes fórmulas y conjuntos, por ejemplo, son cláusulas:

(i)	$\mathsf{js} \vee \neg \mathsf{js}$	$\{\mathsf{js}, \neg \mathsf{js}\}$
(ii)	$\mathsf{js} \vee \mathsf{ps} \vee \neg \mathsf{ms}$	$\{\mathsf{js}, \mathsf{ps}, \neg \mathsf{ms}\}$
(iii)	js	$\{\mathsf{js}\}$
(iv)	$\Box$	$\{\}$

(i) es una tautología. (iii) es una clásula unitaria. (iv) es la cláusula vacía. Obviamente, la cláusula vacía es insatisfacible. Aún más, es la única cláusula insatisfacible, pues si L_1 es un átomo p, la disyunción $L_1 \vee \varphi$ es verdadera en cualquier I tal que $I(p) = 1$, y si L_1 es un literal negado $\neg p$, la disyunción $L_1 \vee \varphi$ es verdadera en cualquier I tal que $I(p) = 0$.

Definición 2.34 *(Forma normal conjuntiva). Una fórmula está en* forma normal conjuntiva *cuando es de la forma* $\bigwedge_{i=1,\ldots,m} \bigvee_{j=1,\ldots,n_i} L_{ij}$ *donde cada* L_{ij} *es un literal.*

Es decir, una fórmula está en forma normal conjuntiva cuando es una conjunción de cláusulas. Por ejemplo, de las siguientes fórmulas

(i)	js
(ii)	$\mathsf{js} \vee \neg \mathsf{ps}$
(iii)	$\mathsf{js} \wedge (\neg \mathsf{js} \vee \mathsf{ps}) \wedge (\neg \mathsf{ps} \vee \neg \mathsf{ms})$
(iv)	$\mathsf{js} \vee (\neg \mathsf{js} \wedge \mathsf{ps})$
(v)	$\mathsf{js} \rightarrow \neg \mathsf{ms}$

están en forma normal conjuntiva (i), (ii) y (iii), y no están en forma normal conjuntiva (iv) y (v).

Proposición 2.11 *Para cualquier fórmula* A *del cp, se puede calcular una fórmula* A^* *del cp tal que* A^* *equivale lógicamente a* A *y está en forma normal conjuntiva.*

DEMOSTRACIÓN: Basta aplicar el procedimiento del cuadro 2.8. Para ello hay que utilizar las tautologías del cuadro 2.6.◁

Mediante $FNC(\varphi)$ podemos denotar a cualquier fórmula en forma normal conjuntiva que sea equivalente a la fórmula φ. C_φ denota, entonces, el conjunto

1.	Eliminar todas las apariciones de $\leftrightarrow$.
2.	Eliminar todas las apariciones de $\rightarrow$.
3.	Aplicando De Morgan,llevar todas las apariciones de $\neg$ hasta los átomos.
4.	Aplicando doble negación, eliminar las cadenas de un número par de $\neg$.
5.	Aplicando distributividad, llevar todas las apariciones de $\vee$ hasta los átomos.
6.	Aplicando idempotencia y absorción, eliminar duplicaciones.

Cuadro 2.8: Procedimiento para pasar a forma normal conjuntiva.

de cláusulas que aparezcan en una $FNC(\varphi)$. Similarmente, $FNC(\Gamma)$ denota un conjunto de fórmulas en forma normal conjuntiva, cada una de las cuales procede de una fórmula del conjunto Γ. Por su parte, C_Γ denota el conjunto de cláusulas que aparecen en las fórmulas de una $FNC(\Gamma)$, es decir, $C_\Gamma = \bigcup_{\varphi\in\Gamma} C_\varphi$.

Ejemplo 2.9 Si φ es $\mathsf{p} \rightarrow (\mathsf{q} \rightarrow \mathsf{r} \wedge \mathsf{s})$ y μ es $\neg(\mathsf{p} \wedge \mathsf{r}) \wedge \mathsf{s}$, entonces

$C_\varphi = \{\neg\mathsf{p} \vee \neg\mathsf{q} \vee \mathsf{r}, \neg\mathsf{p} \vee \neg\mathsf{q} \vee \mathsf{s}\}$.
$C_\mu = \{\neg\mathsf{p} \vee \neg\mathsf{r}, \mathsf{s}\}$.
$C_{\{\varphi,\mu\}} = \{\neg\mathsf{p} \vee \neg\mathsf{q} \vee \mathsf{r}, \neg\mathsf{p} \vee \neg\mathsf{q} \vee \mathsf{s}, \neg\mathsf{p} \vee \neg\mathsf{r}, \mathsf{s}\}$.

A modo de ejemplo, consideremos la fórmula φ y apliquémosle el algoritmo de el cuadro 2.8 para obtener una forma normal conjuntiva de φ.

Paso 1: no hay apariciones de $\leftrightarrow$.
Paso 2: no hay apariciones de $\rightarrow$.
Paso 3: $\neg\mathsf{p} \vee (\neg\mathsf{q} \vee (\mathsf{r} \wedge \mathsf{s}))$
Paso 4: no hay duplicaciones de $\neg$.
Paso 5: $\neg\mathsf{p} \vee ((\neg\mathsf{q} \vee \mathsf{r}) \wedge (\neg\mathsf{q} \vee \mathsf{s}))$
$(\neg\mathsf{p} \vee \neg\mathsf{q} \vee \mathsf{r}) \wedge (\neg\mathsf{p} \vee \neg\mathsf{q} \vee \mathsf{s})$ (forma normal conjuntiva)

El paso 6 ya no se da. $\triangleleft$

Ya que en el paso a forma normal conjuntiva se obtiene una fórmula equivalente a la dada, tenemos trivialmente la siguiente

Proposición 2.12 *Para toda fórmula φ se cumple que φ tiene un modelo si y sólo si C_φ tiene un modelo. Para todo conjunto de fórmulas Γ se cumple que Γ es satisfacible si y sólo si C_Γ es satisfacible.*

Ejercicio 2.21 Si Γ es el conjunto de fórmulas $\{\mathsf{p} \rightarrow (\mathsf{q} \leftrightarrow \neg(\mathsf{r} \wedge \mathsf{s})), \neg(((\mathsf{p} \rightarrow \mathsf{q}) \wedge (\mathsf{s} \rightarrow \mathsf{r})) \rightarrow \mathsf{t})\}$, calcular el conjunto C_Γ. $\triangleleft$

Ejercicio 2.22 Se dice que una fórmula está en *forma normal disyuntiva* si es de la forma $\bigvee_{i=1,\ldots,m} \bigwedge_{j=1,\ldots,n} L_{ij}$ donde cada L_{ij} es un literal. El procedimiento para pasar una fórmula a forma normal disyuntiva es como el del cuadro 2.8, sólo que la distibutividad se aplica llevando $\wedge$ hasta los átomos. Se pide calcular la forma normal disyuntiva de cada una de las fórmulas del conjunto Γ del ejercicio 2.21. $\triangleleft$

2.6.2. EL MÉTODO DE RESOLUCIÓN.

El método de resolución es un *método de refutación*. La idea general en la que se apoyan este tipo de métodos es la siguiente: para probar la validez de un argumento $(\{\varphi_1, \ldots, \varphi_n\}, \varphi)$ (o de una fórmula φ) se intenta refutar la satisfacibilidad de $\{\varphi_1, \ldots, \varphi_n, \neg\varphi\}$ (respectivamente de $\neg\varphi$). La satisfacibilidad de dicho conjunto (fórmula) supone la existencia de un contraejemplo del argumento (de la fórmula φ), es decir, un caso en el que las premisas del argumento son verdaderas y la conclusión falsa (respectivamente: la fórmula es falsa). Por tanto, la no existencia del contraejemplo asegura la validez del argumento (o fórmula). Concretamente, para averiguar la validez de un argumento $(\{\varphi_1, \ldots, \varphi_n\}, \varphi)$ de acuerdo con el método de resolución, se determina si se puede obtener la cláusula vacía $\square$ a partir de $C\{\varphi_1, \ldots, \varphi_n, \neg\varphi\}$ como conjunto de cláusulas inicial, aplicando la regla de resolución como única regla de inferencia. Dado que $\square$ es insatisfacible, si se obtiene dicha cláusula, puede demostrarse que no existe un modelo de las premisas $\varphi_1, \ldots, \varphi_n$ que no lo sea también de la conclusión φ. Por tanto, $(\{\varphi_1, \ldots, \varphi_n\}, \varphi)$ es válido (corrección del método). Por contra, si no se obtiene $\square$, entonces puede probarse que dicho modelo existe y, por tanto, el argumento no es válido; dicho de otra forma, si el argumento es válido, entonces se puede obtener $\square$ (completitud del método). Si se trata de averiguar la validez de una fórmula φ la idea es similar. En este caso se determina si se puede obtener $\square$ a partir de $C\neg\varphi$.

1. FORMULAS $\leftarrow$ PREMISAS $\cup$ { $\neg$ CONCLUSION};
2. FORMULAS $\leftarrow$ FNC(FORMULAS);
3. CLAUSULAS $\leftarrow C_{\text{FORMULAS}}$;
4. Mientras $\square \notin$ CLAUSULAS

   Si existe un par C_1, C_2 de CLAUSULAS resolubles respecto a un literal L tales que no se les ha aplicado la regla de resolución en dicho literal

   entonces aplicar la regla de resolución a C_1 y C_2 en L produciendo una resolvente C,

   si no, devolver FRACASO;

   CLAUSULAS $\leftarrow \{C\} \cup$ CLAUSULAS;
5. fin-mientras;
6. Devolver EXITO;

Cuadro 2.9: Método de resolución para el *cp*.

Definición 2.35 *(Método de resolución). El método de resolución es el que viene definido por el algoritmo del cuadro 2.9*

Definición 2.36 *(Regla de resolución). Sean C_1 y C_2 dos cláusulas tales que una de ellas contiene un literal L, mientras que la otra contiene su opuesto $\neg L$. Se dice en este caso que C_1 y C_2 son* resolubles*; concretamente, que son* resolubles respecto a L. *Se dice que C es la* resolvente de C_1 y C_2 respecto a L *si C es la cláusula que contiene todos los literales de C_1 y C_2 menos L y $\neg L$ (suprimiendo las posibles repeticiones). Si C es la resolvente de C_1 y C_2 respecto a L para algún literal L, diremos que C es una* resolvente *de C_1 y C_2.*

La operación de hallar la resolvente de C_1 y C_2 respecto a L se denomina *aplicar la regla de resolución* a C_1 y C_2 en L o *resolver* en L. Nótese que la cláusula vacía se obtiene al aplicar la regla de resolución a un par de literales opuestos.

Ejemplo 2.10 Sean $C_1 = \neg \mathsf{p} \vee \neg \mathsf{q} \vee \mathsf{r} \vee \mathsf{s}$ y $C_2 = \neg \mathsf{p} \vee \mathsf{q} \vee \mathsf{s} \vee \neg \mathsf{r}$. Las posibles resolventes son $\neg \mathsf{p} \vee \mathsf{r} \vee \mathsf{s} \vee \neg \mathsf{r}$ y $\neg \mathsf{p} \vee \neg \mathsf{q} \vee \mathsf{s} \vee \mathsf{q}$.
◁

Ejemplo 2.11 Tomemos de nuevo el argumento del ejemplo 2.5. Apliquemos el método de resolución:
Paso 1. `FORMULAS` $\leftarrow$ `PREMISAS` $\cup$ {`NEGACION de la CONCLUSION`}.
`FORMULAS` $= \{\mathsf{js} \vee \mathsf{ps}, \mathsf{js} \rightarrow \mathsf{jcs}, \neg\mathsf{jcs}, \neg\mathsf{ps}\}$.
Paso 2. `FORMULAS` $\leftarrow$ `FNC(FORMULAS)`.
`FORMULAS` $= \{\mathsf{js} \vee \mathsf{ps}, \neg\mathsf{js} \vee \mathsf{jcs}, \neg\mathsf{jcs}, \neg\mathsf{ps}\}$.
Paso 3. `CLAUSULAS` $\leftarrow$ `Descomponer FORMULAS según conjunciones`.
`FORMULAS` no varía, ya que ninguna fórmula es una conjunción
Bucle `MIENTRAS`: Al final `CLAUSULAS` =

1. $\mathsf{js} \vee \mathsf{ps}$
2. $\neg\mathsf{js} \vee \mathsf{jcs}$
3. $\neg\mathsf{jcs}$
4. $\neg\mathsf{ps}$
5. $\neg\mathsf{js}$ (resolviendo 2 y 3)
6. js (resolviendo 1 y 4)
7. $\square$ (resolviendo 5 y 6)

y se devuelve `EXITO`, ya que se ha generado la cláusula vacía. Por tanto, la argumentación es válida. ◁

Ejercicio 2.23 Sea el conjunto de cláusulas $\Gamma = \{\neg \mathsf{p}_1 \vee \neg \mathsf{q}_1 \vee \neg \mathsf{r}_1,\ \mathsf{p}_1 \vee \mathsf{q}_1,\ \neg \mathsf{p}_2 \vee \mathsf{q}_1,\ \mathsf{s}_2 \vee \mathsf{r}_1 \vee \mathsf{p}_2,\ \mathsf{r}_1 \vee \neg \mathsf{r}_1\}$. Hallar todas las resolventes que se pueden obtener tomando dos a dos los elementos de Γ. ◁

Ejercicio 2.24 Determinar, por el método de resolución, si las siguientes proposiciones son tautologías:

1. $(\neg \mathsf{p} \rightarrow \mathsf{q}) \rightarrow ((\neg \mathsf{p} \rightarrow \neg \mathsf{q}) \rightarrow \mathsf{p})$
2. $(\mathsf{p} \wedge \neg \mathsf{q}) \vee (\neg \mathsf{p} \wedge \mathsf{q}) \vee (\mathsf{q} \wedge \mathsf{p}) \vee \neg(\mathsf{p} \vee \mathsf{q})$
3. $(\mathsf{q} \rightarrow (\mathsf{p} \vee \mathsf{r})) \leftrightarrow ((\neg\neg \mathsf{q} \vee \mathsf{r}) \rightarrow (\mathsf{q} \rightarrow \mathsf{p}))$
4. $((\neg \mathsf{p} \vee \mathsf{q}) \rightarrow \mathsf{r}) \rightarrow \neg(\neg \mathsf{p} \vee \mathsf{q}) \vee \mathsf{r}$
5. $(\mathsf{p} \downarrow \mathsf{p}) \mid (\neg \mathsf{p} \downarrow \neg \mathsf{p})$

◁

Ejercicio 2.25 Sea el siguiente argumento en lenguaje natural:
De los cofres A y B, uno de ellos contiene un tesoro, el otro no. En cada cofre hay escrita una inscripción, y al menos una de ellas es falsa. Las inscripciones

son las siguientes: Cofre A: "Si la inscripción del cofre B es falsa, no contiene el tesoro". Cofre B: "Este cofre contiene un tesoro". Por tanto, el cofre A contiene un tesoro.

Estudiar su validez por el método de resolución. ◁

Ejercicio 2.26 Un agente (vd. figura 2.1) se encuentra en un mundo virtual compuesto por un conjunto de celdillas. Cada celdilla se comunica con la de abajo, la de arriba, la de la izquierda y la de la derecha. El agente sabe en este momento lo siguiente:

Hay un monstruo arriba, abajo, a la izquierda o a la derecha. Si hay un monstruo a la derecha, entonces huele mal. Caen gotas de moco verde sólo si hay un monstruo arriba. Cuando hay un monstruo a la izquierda o abajo, entonces la iluminación es escasa. No huele mal, ni caen gotas de moco verde. La iluminación es abundante.

Si nuestro agente puede aplicar el método de resolución, ¿sabrá dónde hay monstruos? ◁

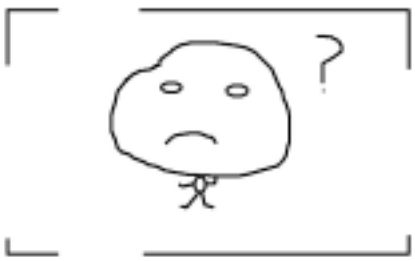

Figura 2.1: Un agente perplejo.

2.6.3. TERMINACIÓN.

En esta sección probamos que el método de resolución termina siempre.

Teorema 2.3 *(Terminación del método de resolución para el cp). Sea cual sea el orden de aplicación de la regla de resolución, el método acaba (devuelve EXITO o FRACASO tras un número finito de aplicaciones).*

DEMOSTRACIÓN: Demostraremos el teorema para el caso de un argumento (para una fórmula aislada es similar). Sea Γ un conjunto finito de cláusulas (correspondientes a las premisas y negación de la conclusión de un argumento) y sea $|\Omega_\Gamma| = n$. Es fácil ver que el número de literales posibles es $2n$ y por tanto el número n_Γ de cláusulas posibles (en forma de conjunto) es 4^n. Aún más, el número de formas en que cada cláusula podría originarse es también finito y acotado por $n_\Gamma \times n_\Gamma$. Ya que el método no permite apariciones repetidas de una cláusula en CLAUSULAS (pues CLAUSULAS se ha definido como conjunto), ni permite repeticiones de literales en una cláusula, ni aplica dos veces la regla a un mismo par de cláusulas respecto de un mismo literal, es claro que si no devuelve EXITO, tarde o temprano devuelve FRACASO, tras haber generado

de todas las posibles formas todas las cláusulas que se pueden obtener partiendo de Γ. $\triangleleft$

Nótese que el método de resolución genera siempre una secuencia finita de cláusulas $C_1, \ldots, C_n$ tal que cada C_i $(1 \leq i \leq n)$ es o bien una cláusula de Γ o bien una resolvente de dos cláusulas precedentes en la secuencia. Se suele decir en este caso que hay una *deducción por resolución de C_n a partir de* Γ. Si C_n es la cláusula vacía, se dice entonces que dicha secuencia es una *refutación por resolución* de Γ.

Ejercicio 2.27 Estudiar si está garantizada la terminación del método cuando se modifica como sigue:

(i) CLAUSULAS se define como una lista de fórmulas con posibles repeticiones.

(ii) CLAUSULAS se define como un conjunto de fórmulas, pero cada cláusula puede contener varias apariciones de un mismo literal. $\triangleleft$

2.6.4. CORRECCIÓN.

El método de resolución es correcto y completo con relación a la semántica. Para probar la corrección establecemos el siguiente

Lema 2.2 *Sean C_1 y C_2 dos cláusulas resolubles respecto a L y sea C la resolvente de C_1 y C_2 respecto a L. Sea I una interpretación cualquiera. Si $I \vDash C_1$ y $I \vDash C_2$, entonces $I \vDash C$.*

DEMOSTRACIÓN: Si C_1 y C_2 son de la forma $L \vee \varphi_1$ y $\neg L \vee \varphi_2$, la proposición 2.4 nos garantiza que la argumentación "$L \vee \varphi_1$ y $\neg L \vee \varphi_2$, luego $\varphi_1 \vee \varphi_2$" es válida, y la proposición 2.3 más la sustitución de equivalentes nos permite suprimir apariciones repetidas de un mismo literal, con lo que el lema está probado. Supongamos ahora que, por ejemplo, C_1 es un literal de la forma L y C_2 de la forma $\neg L \vee \varphi_2$. De la misma manera, es fácil ver que la argumentación "L y $\neg L \vee \varphi_2$, luego φ_2" es válida, y φ_2 es precisamente C (pues no contiene repeticiones). Por último, si ambas cláusulas son de la forma L y $\neg L$, respectivamente, su resolvente C es $\Box$; pero no existe ninguna interpretación I donde sean simultáneamente verdaderas L y $\neg L$, así que trivialmente también se cumple el lema. $\triangleleft$

Ya podemos demostrar el lema principal:

Lema 2.3 *Si el método devuelve EXITO partiendo de Γ como conjunto de cláusulas inicial, entonces Γ es insatisfacible.*

DEMOSTRACIÓN: Demostraremos primero que si Γ es un conjunto de cláusulas satisfacible, cualquier secuencia de cláusulas obtenida por resolución partiendo de Γ como conjunto inicial es satisfacible. Demostraremos esto por inducción sobre el número n de clásulas de la secuencia obtenida por resolución partiendo de Γ como conjunto inicial.

Paso base: $n = 1$. En ese caso la secuencia se reduce a una cláusula que pertenece a Γ, y Γ es satisfacible por hipótesis.

Paso de inducción: supogamos que el lema se cumple para cualquier secuencia generada a partir de Γ como conjunto inicial y que tenga menos de n cláusulas,