LÓGICA
de proposiciones

por

Luis Español González
y
Luis Javier Hernández Paricio
Departamento de Matemáticas y Computación
Universidad de La Rioja

Logroño
2020

Prólogo

Estas páginas contienen la selección de materia sobre lógica de proposiciones que se ha juzgado apropiado incluir en una asignatura semestral de un primer curso universitario dirigido a futuros graduados en matemáticas o en informática. El enfoque participa del modo de hacer matemático, que aporta la máxima precisión, y también dirige su atención hacia la conexión de la lógica con la informática al seleccionar como uno de los temas el método de resolución. Trata sólo de la lógica tradicional o clásica, si bien contada con un estilo actual, y se dejar ver algún ejemplo de otras lógicas. Los objetivos básicos que el autor pretende alcanzar con este texto son:

- Ayudar a comprender los procesos de construcción de un lenguaje y de asignarle al mismo una interpretación en términos de verdad.
- Comprender y utilizar el álgebra de la lógica.
- Facilitar el aprendizaje y la práctica de las respuestas algorítmicas a las preguntas básicas de la lógica
- Facilitar el aprendizaje y la práctica de las respuestas algorítmicas a las preguntas básicas de la lógica.
- Facilitar el aprendizaje y la práctica de las respuestas algorítmicas a las preguntas básicas de la lógica.
- Facilitar el aprendizaje y la práctica de los procedimientos para razo- nar por método deductivo.

La primera tarea es delimitar con precisión el lenguaje simbólico que se va a utilizar. Las proposiciones serán las expresiones de dicho lenguaje, que serán construidas de una manera reglada a partir de un alfabeto, de símbolos lógicos y de paréntesis. Para la construcción inicial del lenguaje de la lógica de proposiciones se ha elegido, porque es una cuestión opcional, partir del sistema de conectores $\{\neg, \to\}$, pero enseguida se ponen en circulación los otros conectores habituales, sobre todo $\wedge$ y $\vee$. Una vez dispuesto el lenguaje, el problema básico es determinar si una proposición es una tautología, problema al que se pueden dar otras formulaciones equivalentes. Hay varios puntos de vista, todos ellos equivalentes, para proceder al análisis de la verdad de las proposiciones:

- El punto de vista *semántico* asocia a cada proposición una función de verdad. El algoritmo de cálculo de dicha función es la tabla de verdad de la proposición, que es *decidible* por tratarse de un cálculo numérico (con 0 y 1) finito. Pero si el tamanño crece el problema se vuelve impracticable.

- El aspecto *algebraico* pone la atención en las operaciones con las proposiciones que forman la llamada estructura de álgebra de Boole. El método algebraico transforma el cálculo numérico de la tabla en un *cálculo simbólico* que permite alcanzar los mismos resultados; se manipulan las proposiciones, módulo equivalencia lógica, como si se tratara, con algunas reglas de cálculo iguales pero otras diferentes, del álgebra que usamos con los números.

- El método de *resolución* es un *algoritmo* que permite decidir si una proposición es o no una *contradicción*, de modo que da una respuesta algorítmica a la pregunta básica de la lógica. Este algoritmo requiere la adaptación previa de la sintaxis de la proposición a una forma concreta llamada *clausal*.

- El punto de vista *deductivo* pone el énfasis en la construcción de razonamientos válidos que permitan concluir que una proposición es verdadera cuando lo son otras en las que se apoya el razonamiento. El método deductivo es el modo de razonar de las ciencias, particularmente de la matemática, pero hacer demostraciones es un arte, no es un procedimiento mecánico. Se ha dado prioridad a la deducción natural, con sus variadas reglas primitivas, todas ellas inmediatas de reconocer en nuestro usos lógicos, y los tres procedimientos deductivos que también se identifican con facilidad en la práctica en las argumentaciones habituales.

Hasta aquí queda esbozado el contenido básico del curso en cuanto a las proposiciones, pero hay otros aspectos complementarios que también aparecerán aunque con menor énfasis. Del lado algebraico se hace ver que el procedimiento usado para valorar las proposiciones con dos valores de verdad puede repetirse con otros otros conjuntos más amplios de valores de verdad en los que se haya establecido una interpretación para los conectores. De este modo se abre la puerta al estudio de las lógicas no clásicas. Otro ámbito de generalidad queda patente cuando se observa que, fijados los conectores, el conjunto (álgebra) $\mathcal{P}$ de las proposiciones depende del alfabeto $\mathcal{A}$ y esta dependencia tiene un carácter funtorial, lo que significa que hay una relación natural, que se hará explícita, entre el cambio de alfabeto y el cambio que resulta en las proposiciones. Por otra parte, además de la deducción natural existen otros métodos deductivos que contemplan como datos iniciales sólo la regla *Modus ponens* junto con una lista de proposiciones que se toman como axiomas. Para trabajar con alguún detalle se elige el sistema de Lukasiewicz (1929) por su carácter a la vez minimalista y muy expresivo en su condición lógica (sólo tres grupos de axiomas en el sistema de conectores $\{\neg, \rightarrow\}$). Luego se ven otros modos alternativos de seleccionar los axiomas para obtener sistemas axiomáticos equivalentes, en el sentido de que a partir de ellos se prueban los mismos teoremas. Para demostrar esto basta producir en cada sistema la prueba de los axiomas, reglas o procedimientos del otro.

Luis Español

Universidad de La Rioja Logroño, febrero 2014

Los apuntes de esta asignatura han sido elaborados por el Profesor Luis Español que se jubiló a finales del curso 2018/19. En la actual versión, además de pequeñas modificaciones, se han añadido una nueva sección dedicada a resoluciones automatizadas y tres nuevas secciones que están dedicadas al método deductivo de Frederic Fitch (publicado en1952).

Hay varias razones para la inclusión de este sistema deductivo: Una de ellas es que el aprendizaje de sistema deductivo de Gentzen se estaba realizando mediate un asistente de dedución natural denominado ADN que ha quedado obsoleto. Este asistente ha quedado fuera de juego simplemente porque su ejecución sólo era compatible con versiones antiguas del navegador explorer que no están disponibles en los sistemas operativos más recientes. Otra razón es que el alumno puede observar que existen otros métodos deductivos que además de utilizar otros conjuntos de conectores arrancan con una familia distinta de reglas y procedimientos primitivos. También podemos decir que el nuevo asistente admite una pequeña colección de reglas y procedimientos derivados básicos cuyo uso hacen que se puedan elaborar demostraciones mucho más rápidas ya que no es necesario la continua repetición de algunos patrones frecuentes de demostración.

El nuevo asistente de dedución natural, basado en el método de Fitch, ha sido elaborado siguiendo los materiales sobre lógica avanzada (adecuada para informáticos y matemáticos) elaborados por «Open Logic Project».

Existe también una aplicación web cuyo código está alojado en Github
https://github.com/OpenLogicProject/fitch-checker
y se puede ejecutar en
http://proofs.openlogicproject.org/.

Este asistente tiene como objetivo que el escolar aprenda a realizar demostraciones en la lógica de proposiciones mediante el sistema deductivo de Fitch. Dispone de un editor que permite al alumno elaborar una demostración mediante una sucesión de lineas y de subdeducciones que a partir de un conjunto de premisas lleva a una conclusión. Además tiene un corrector de demostraciones que indica si la demostración realizada es correcta o no y en este último caso indica los lugares y errores efectuados durante el intento de demostración.

Luis Javier Hernández Paricio
Universidad de La Rioja, febrero de 2020.

Índice general

1. Sintaxis y semántica **3**
- 1.1. El lenguaje de las proposiciones . . . 3
- 1.2. Tablas de verdad . . . 7
- 1.3. Equivalencias . . . 12
- 1.4. Cláusulas y formas normales . . . 14

2. Lógica algebraica **19**
- 2.1. El álgebra libre $\mathscr{P}(\mathscr{A})$ . . . 19
- 2.2. El álgebra de Boole $\mathscr{B}(\mathscr{A})$ . . . 22
- 2.3. Álgebras de Boole abstractas . . . 25

3. Métodos de demostración **31**
- 3.1. Reglas de inferencia . . . 31
- 3.2. Método de resolución . . . 34
- 3.3. Un sistema automatizado de demostración: Prover9 . . . 38
- 3.4. Deducción natural (método Gentzen) . . . 40
- 3.5. El sistema de deducción de Frederic Fitch . . . 45
- 3.6. Reglas y procedimientos derivados básicos . . . 49
- 3.7. Derivación de las reglas y procedimientos derivados básicos . . . 50
  - 3.7.1. Derivación de la reiteración . . . 50
  - 3.7.2. Derivación del silogismo deductivo (Un caso de la regla de la resolución) . . 51
  - 3.7.3. Derivation de la regla derivada básica modus tollens . . . 52
  - 3.7.4. Derivación de la regla de la doble negación . . . 52
  - 3.7.5. Derivación de la regla: ley del medio excluido . . . 52
  - 3.7.6. Derivación de las reglas de De Morgan . . . 53
- 3.8. Los axiomas de Lukasiewicz . . . 54
- 3.9. Otros sistemas de axiomas . . . 59

Capítulo 1

Sintaxis y semántica

La sintaxis se ocupa del modo como escribimos las proposiciones, de su lenguaje escrito, y la semántica las interpreta, es decir, les da un significado en términos de verdad. En la escritura precisa de las proposiciones se usan abundantes paréntesis, pero unas reglas sencillas de prioridad permiten ahorrar algunos de ellos. Unos diagramas en forma de árbol ayudan a comprender el procedimiento de construcción de las proposiciones y también para asignarles valor de verdad, aunque el método genuino para esto es el de las tablas de verdad matriciales.

1.1. El lenguaje de las proposiciones

Para escribir poposiciones necesitamos determinar un cierto número de símbolos y dar las reglas para combinarlos en sucesión finita siguiendo algún criterio. El lenguaje de las proposiciones requiere tres tipos de símbolos: *literales, lógicos* y de *puntuación*. La elección de estos símbolos es en principio arbitraria, pero lo razonable es seguir alguno de los usos más establecidos en la práctica de la lógica y la matemática.

Definición 1.1.1 *El lenguaje del cálculo de proposiciones está formado por la siguiente colección de símbolos:*

*A. Un conjunto (*alfabeto*) de símbolos literales* $A = \{p, q, r, \cdots\}$

B. Los símbolos lógicos o conectores: negación «¬ »y condicional «→».

C. Paréntesis «( )». Llamaremos proposición a una sucesión finita de símbolos del lenguaje anterior construida según las siguientes instrucciones:

Llamaremos proposición a una sucesión finita de símbolos del lenguaje anterior construida según las siguientes instrucciones:

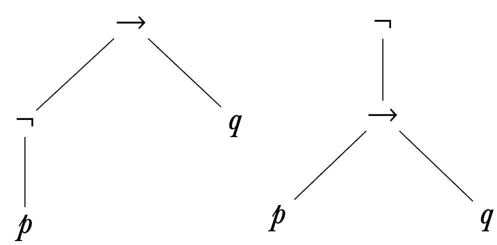
1. *Cada símbolo literal es una proposición (atómica o átomo).*

2. *Si «*P*» es una proposición entonces «*$\neg P$*» es una proposición.*

3. *Si «*P*» y «*Q*» son proposiciones entonces «*$(P \to Q)$*» es una proposición.*

4. *Sólo son proposiciones las construidas según las instrucciones anteriores.*

La estructura de una proposición se describe mediante un diagrama llamado *árbol sintáctico* que tiene en sus nodos los conectores de la proposición (tantos nodos como conectores aparecen, con repetición) y en sus terminaciones los átomos de la proposición tantas veces como aparezcan en la expresión. Basta dar dos ejemplos sencillos para que se comprenda su construcci ón general. El conector que aparece en primer lugar se llama conector *raíz* de la proposición:



Como la construcción de una proposición es progresiva, hay trozos de una proposición que a su vez son proposiciones, los que se obtienen hacia abajo desde un nodo del árbol sintáctico. En el ejemplo anterior a la izquierda se observan los siguientes fragmentos que son proposiciones: los dos átomos p y q, la proposición$\neg p$ y la proposición completa $(\neg p \to q)$. Estos "trozos" los llamaremos *subproposiciones.* En el ejemplo de la derecha: los dos átomos p y q, la proposición $(p \to q)$ y la proposición completa $\neg(p \to q)$.

Los dos ejemplos anteriores permiten advertir la importancia de la colocación de los paréntesis, pues excluidos éstos la sucesión de símbolos coincide en ambos casos, siendo precisamente los paréntesis los que determinan la forma diferente del árbol sintáctico. Insistiendo en este asunto sintáctico, nótese que $(p \to \neg q \to r)$ no es una proposición, pero puede completarse para que lo sea de varias maneras distintas: $((p \to \neg q) \to r)$ o $(p \to (\neg q \to r))$ o $(p \to \neg(q \to r))$, de las que el lector puede dibujar el respectivo árbol sintáctico.

Es fácil observar que cuando un paréntesis abraza todo el resto de una proposición, ésta tiene la forma $(P \to Q)$ siguiendo la instrucción 3, donde P y Q son proposiciones fragmentos de la completa. El paréntesis exterior sólo es imprescindible si luego se aplica otro conector, pero si la fórmula se da por terminada podemos suprimirlo y escribir simplemente $P \to Q$, lo que haremos habitualmente. Si tenemos dos proposiciones de esta forma, sean $P \to Q, R \to S$, y formamos con ellas otro condicional, tendremos que escribir, por ejemplo,$(P \to Q) \to (R \to S)$, obviando el paréntesis exterio final, pero no los paréntesis intermedios.

Conviene hacer otra precisión más sobre el uso de paréntesis. Con el objetivo de reducir el número de paréntesis al escribir las proposiciones, se suele admitir una relación de prioridad o predecencia que permite ahorrar paréntesis. Nosotros consideraremos el siguiente orden de precedencia:

$$\neg \quad < \quad \wedge \quad < \quad \vee \quad < \quad \oplus \quad < \quad \to \quad < \quad \leftrightarrow .$$

El convenio es que es caso de poder interpretar de dos modos distintos la colocación de paréntesis, se agrupan dentro de un paréntesis lo que este más próximo a un conector con menor precedencia. Por ejemplo para $\neg P \to Q$, se pueden colocar los paréntesis bien como $(\neg P) \to Q$ o bien $\neg(P \to Q)$. Notemos que se trata de dos proposiciones distintas, pero la interpretación que debemos considerar correcta es la primera ya que $\neg$ tiene menor orden de precedencia que $\to$. Otro ejemplo, si consideramos $p \wedge q \vee r$ debemos tomarlo como $(p \wedge q) \vee r$ porque el order de precedencia de $\wedge$ es menor que el de $\vee$.

A continuación damos dos filas de expresiones usando letras minúsculas, las de la primera fila son todas disparates sintácticos, las de la segunda son todas proposiciones:

$$pp\neg \to q,)\neg)\neg q \to (, p \to \neg q(p), (p \to (\to p)$$

$$(p \to \neg q), (\neg p \to q), \neg(p \to q), \neg\neg(p \to (\neg p \to \neg(q \to p)))$$

En las dos primeras proposiciones de la segunda fila podrían suprimirse los respectivos paréntesis exteriores sin producir error.

Una observación más. Cuando escribimos una proposición de la forma $\neg P \to Q$ mostramos en general una proposición de modo incompleto, simplemente damos su forma, y la proposición queda completamente descrita una vez expresada en términos de los átomos. Por ejemplo, si P es $p \to q$ y Q es $\neg q$ entonces $\neg P \to Q$ es $\neg(p \to q) \to \neg q$. Lo que afirmemos sobre $\neg P \to Q$ será válido para todas las proposiciones que tengan dicha forma, cualesquiera que sean P y Q. Podemos decir que «calculamos» $\neg P \to Q$ cuando sustituimos P y Q por sus expresiones concretas, pero una vez que en la expresión sólo aparecen los átomos y los conectores (con los paréntesis necesarios) ya no se puede «calcular» más en el ámbito de la sintaxis. Algunas veces para explicitar la lista de átomos que intervienen en una proposición P utilizamos la notación $P = P(p_1, \cdots, p_n)$ donde $p_1, \cdots, p_n \in \mathcal{A}$.

Se denotará con la letra $\mathcal{P}$ el conjunto de todas las proposiciones obtenidas según la Definición 1.1.1, de modo que la primera instrucción significa $\mathcal{A} \subset \mathcal{P}$. Es evidente que $\mathcal{P}$ es un conjunto infinito, aunque el alfabeto tenga una sola letra. En efecto, incluso en el caso $\mathcal{A} = \{p\}$ Hay infinitas proposiciones: lo son $p, \neg p, \neg\neg p, \neg\neg\neg$p, etc., y también$p \to p, p \to (p \to p), p \to (p \to (p \to p))$, etc., sin olvidar $p \to \neg p, \neg p \to p$ y otros muchos etcéteras. El propio alfabeto $\mathcal{A}$ podría ser un conjunto infinito, por ejemplo el infinito numerable $\mathcal{A} = \{p_1, p_2, \cdots, p_n, \cdots\}$, pero incluso en los casos de alfabeto infinito cada proposición P tendrá sólo un conjunto finito de letras del alfabeto.

En el caso que sea necesario destacar el alfabeto $\mathcal{A}$ con el que estamos trabajando denotaremos por $\mathcal{P}(\mathcal{A})$ el álgebra de las proposiciones generada por $\mathcal{A}$. Por ejemplo, si estamos trabajando con dos alfabetos $\mathcal{A}_1$ y $\mathcal{A}_2$ dispondremos de dos álgebras de proposiciones distintas, $\mathcal{P}(\mathcal{A}_1)$ y $\mathcal{P}(\mathcal{A}_2)$. De este modo cada alfabeto determina una álgebra de proposiciones.

Conectores derivados. La presentación formal de la lógica de proposiciones se simplifica usando pocos conectores, hemos elegido el sistema $\{\neg, \to\}$ porque son sólo dos y representan operaciones básicas del discurso lógico. Pero ello no significa que nos quedemos sin el resto de los conectores, porque podemos introducirlos como abreviaturas de expresiones del lenguaje formal definido. Tal definición de conectores *derivados* a partir de los conectores *primitivos* es en principio arbitraria desde el punto de vista sintáctico, pero la semántica nos indicará que se trata de los conectores usuales, junto con sus equivalencias de especificación. Podríamos definir más conectores binarios, pero nos limitaremos a los más habituales.

Definición 1.1.2 *Se definen los conectores binarios «$\neg$», «$\wedge$», «$\leftrightarrow$», «$\oplus$», como abreviaturas de las expresiones que se indican:*

$P \vee Q = \neg P \to Q$.
$P \wedge Q = \neg(P \to \neg Q)$.
$P \leftrightarrow Q = \neg((P \to Q) \to \neg(Q \to P))$.
$P \oplus Q = (P \to Q) \to \neg(Q \to P)$.

Con un criterio más exigente sobre el uso de paréntesis podríamos haber escrito$(P \vee Q) = (\neg P \to Q)$ etc., pero es claro que se entiende bien la escritura ahorrando unos pocos símbolos. Por otra parte, si hacemos progresiva la introducción de los conectores derivados, utilizando los ya definidos, podemos escribir:

$P \vee Q = \neg P \to Q$.

$P \wedge Q = \neg(P \to \neg Q)$.

$P \leftrightarrow Q = (P \to Q) \wedge (Q \to P)$.

$P \oplus Q = \neg(P \leftrightarrow Q)$.

Pero la primera relación (Definición 1.1.2) es más precisa en cuanto que cada conector derivado está expresado directamente en el sistema de conectores primitivos. A partir de ahora podemos escribir proposiciones complejas en las que aparezcan indistintamente los conectores primitivos $\{\neg, \to\}$ y los derivados $\vee, \wedge, \leftrightarrow, \oplus$.

En la Definición 1.1.1 se tomaron como conectores primitivos $\{\neg, \to\}$. Sin embargo en otros textos de lógica de proposiciones es frecuente tomar como conectores primitivos $\{\neg, \wedge, \vee\}$. Entonces, a partir de tales conectores primitivos, los derivados podrían ser definidos del siguiente modo:

$P \to Q = \neg P \vee Q$.

$P \leftrightarrow Q = (\neg P \vee Q) \wedge (\neg Q \vee P)$.

$P \oplus Q = \neg((\neg P \vee Q) \wedge (\neg Q \vee P))$.

Al haber más conectores en la Definición inicial del lenguaje de las proposiciones se amplía la casuística inductiva (pasos 1 a 4 en la Definición 1.1.1) pero disminuye el número de conectores derivados. Hay autores que prefieren empezar con una Definición de las proposiciones que utiliza como primitivos los cinco conectores básicos $\{\neg, \wedge, \vee, \leftrightarrow\}$, de modo que nos quedaría como conector derivado

$P \oplus Q = \neg(P \leftrightarrow Q)$.

Nota 1.1.3 *Para que una proposición este bien formulada tiene que tener los paréntesis colocados de modo coherente con la estructura sintáctica. Se pueden suprimir algunos paréntesis en los que se haya tenido en cuenta el orden de precedencia de los conectores y también se pueden suprimir los paréntesis más externos (si no se continúa aplicando más conectores ya no son necesarios).*

Para terminar este apartado, notemos que los símbolos lógicos pueden usarse como símbolos de operaciones en el conjunto $\mathscr{P}$, escribiendo:

$$\neg\colon \mathscr{P} \to \mathscr{P}, \quad \to\colon \mathscr{P} \times \mathscr{P} \to \mathscr{P}$$

La costumbre en casos como éste es decir que $\mathscr{P}$ es un álgebra con la operaciones $\neg$ y $\to$. La primera se dice 1-aria porque es una aplicación de una variable y la segunda 2-aria o binaria porque tiene dos variables.

Nota 1.1.4 *En algunos textos se suele utilizar un conector adicional de aridad 0 que se denota por $\perp$. Dado un alfabeto $\mathscr{A}$ la construción del álgebra de las proposiciones tiene algunos cambios. En realidad es suficiente hacer el siguiente cambio en la primera instrucción*

1$\perp$. Cada símbolo literal es una proposición (atómica o átomo). Además $\perp$ es una proposición.

Si admitimos $\perp$ se tiene que el álgebra de proposiciones generada por un alfabeto $\mathscr{A}$ admite más proposiciones. Por ejemplo podemos escribir $p \to \perp$, $P \wedge \perp$, $P \vee \perp$, $\neg\perp$ etc. En particular $\neg\perp$ es también una proposición que a veces se denota por $\top$.

Si fuera necesario denotaremos por $\mathscr{P}_\perp$ el álgebra generada con el nuevo sistema de instrucciones. Dado un alfabeto $\mathscr{A}$ podemos denotar por $\mathscr{P}_\perp(\mathscr{A})$ a la nueva algebra así generada que además dispone de una operación de aridad 0: $\mathscr{P}_\perp(\mathscr{A})^0 \to \mathscr{P}_\perp(\mathscr{A})$. Donde $\mathscr{P}_\perp(\mathscr{A})^0 = \{*\}$ se toma como un conjunto con un elemento $*$ y la imagen de este elemento es precisamente $\perp$.

Algunas funciones definidas por inducción. Teniendo en cuenta que la construcción del algebra $\mathscr{P}(\mathscr{A})$ se ha realizado mediante un proceso inductivo no es difícil la definición inductiva de las siguientes funciones, longitud $\operatorname{long}\colon \mathscr{P}(\mathscr{A}) \to \mathbb{N}$, profundidad $\operatorname{prof}\colon \mathscr{P}(\mathscr{A}) \to \mathbb{N}$ y conjunto de subproposicones $\operatorname{subp}\colon \mathscr{P}(\mathscr{A}) \to \mathbb{P}(\mathscr{P}(\mathscr{A}))$, donde $\mathbb{N}$ denota el conjunto de los números naturales y $\mathbb{P}(\mathscr{P}(\mathscr{A}))$ denota la familia de todos los subconjuntos de $\mathscr{P}(\mathscr{A})$. En primer lugar se definen sobre los símbolos del alfabeto, que son las proposiciones más sencillas, y luego se extienden inductivamente del modo siguiente:

$$\operatorname{long}(P) = \begin{cases} 1 & \text{si } P = p \in \mathscr{A} \\ 1 + \operatorname{long}(Q) & \text{si } P = \neg Q \\ 1 + \operatorname{long}(Q_1) + \operatorname{long}(Q_2) & \text{si } Q = Q_1 * Q_2 \end{cases}$$

$$\operatorname{prof}(P) = \begin{cases} 0 & \text{si } P = p \in \mathscr{A} \\ 1 + \operatorname{prof}(Q) & \text{si } P = \neg Q \\ 1 + \max\{\operatorname{prof}(Q_1), \operatorname{prof}(Q_2)\} & \text{si } Q = Q_1 * Q_2 \end{cases}$$

$$\operatorname{subp}(P) = \begin{cases} \{P\} & \text{si } P = p \in \mathscr{A} \\ \{P\} \cup \operatorname{subp}(Q) & \text{si } P = \neg Q \\ \{P\} \cup \operatorname{subp}(Q_1) \cup \operatorname{subp}(Q_2)\} & \text{si } Q = Q_1 * Q_2 \end{cases}$$

donde $*$ denota el conector binario $\to$ si estamos utilizando el sistema de conectores $\{\neg, \to\}$. En el caso que utilicemos otro sistema, por ejemplo $\{\neg, \wedge, \vee\}$, $*$ denota uno de los conectores binarios del sistema en uso.

1.2. Tablas de verdad

Hasta ahora nos hemos limitado a cuestiones que afectan al lenguaje (formal) de las proposiciones, pero la lógica quiere ocuparse del significado de las mismas en términos de verdad. A partir de este momento, se trata de *interpretar* las proposiciones mediante dos *valores de verdad* que representamos con números: 0 significa *falso* y 1 significa *verdadero*. Ambos valores los reunimos en el conjunto $2 = \{0, 1\}$, que lo convertimos en un álgebra, similar en cierto sentido a $\mathscr{P}$, mediante unas operaciones que denotamos con los propios símbolos lógicos por razones que ya son evidentes, pero quedarán más claras al final del proceso de interpretación. Las operaciones lógicas o booleanas en el conjunto 2 son $\neg\colon 2 \to 2$ y $\to\colon 2 \times 2 \to 2$, definidas así:

x	$\neg x$
1	0
0	1

x	y	$x \to y$
1	1	1
1	0	0
0	1	1
0	1	1

Ahora ya tenemos todos los elementos necesarios para definir el concepto de interpretación:

Definición 1.2.1 *Una* interpretación *(principal) de $\mathscr{P}$ es una aplicación $\bar{v}\colon \mathscr{P} \to 2$ obtenida del siguiente modo:*

(i) Se parte de una aplicación cualquiera $v\colon \mathscr{A} \to 2$,

(ii) y se extiende $\bar{v}$ a $\mathscr{P}$

$$\begin{tikzcd}
\mathscr{A} \arrow[r, hook] \arrow[d, "v"'] & \mathscr{P} \arrow[dl, "\bar{v}"] \\
2 &
\end{tikzcd}$$

imponiendo que $\bar{v}$ conserve las operaciones de $\mathscr{P}$ y las de 2 indicadas con el mismo símbolo lógico, es decir,

$$\bar{v}(\neg P) = \neg\bar{v}(P), \quad \bar{v}(P \to Q) = \bar{v}(P) \to \bar{v}(Q)$$

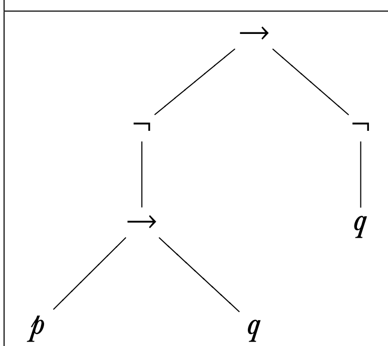
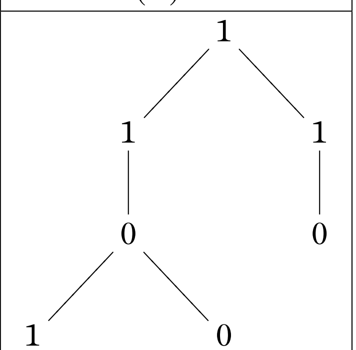
En una tal interpretación $\bar{v}$, una proposición P es verdadera si $\bar{v}(P) = 1$ y falsa si $\bar{v}(P) = 0$. Cuando no haya lugar a confusión a la extensión $\bar{v}$ la seguiremos denotando por v.

Naturalmente, esta noción de verdad se corresponde con el significado ordinario del término en virtud de la elección que hemos hecho del conjunto de valores de verdad y de los dos conectores lógicos.

El resultado es que interpretar los átomos conduce automáticamente a interpretar las proposiciones, siempre que la interpretación se haga conservando las operaciones correspondientes de $\mathscr{P}$ y de 2. Es decir, para interpretar la proposición $\neg(p \to q) \to \neg q$, que para abreviar se denota con la letra P, hay que dar valores de los átomos que en ella aparecen, por ejemplo, $v(p) = 1, v(q) = 0$, y calcular el valor de P «reescribiendo P en 2»y luego «calculando en 2»:

$$v(P) = \neg(v(p) \to v(q)) \to \neg v(q) = \neg(1 \to 0) \to \neg 0 = \neg 0 \to 1 = 1 \to 1 = 1$$

Una forma de visualizar este cálculo se basa en el árbol sintáctico de P, veámoslo por una sola vez. En el cuadro siguiente, a la izquierda está el árbol sintáctico de $\mathscr{P}$ y a la derecha una copia del mismo pero sustituyendo átomos y conectores por sus valores, que se van obteniendo en sentido ascendente a partir de los valores de los átomos

P	$v(P) = 1$
	

Cuando queremos ver juntas todas las interpretaciones posible una única tabla es más práctica que la repetición del árbol en cada interpretación. Una *tabla de verdad* es la representación ordenada por las de todas las interpretaciones posibles de una proposición dada, de modo que cada interpretación es una fila de la tabla, que tendrá 2^n filas si n es el número de átomos que aparecen en la proposición.

Sea como antes $P = \neg(p \to q) \to \neg q$. El cálculo de su valor de verdad realizado unas líneas antes se expone en forma de tabla con el valor $v(P)$ debajo del conector raíz de P, el último que se aplica en la construcción sintáctica de P:

p	q	P:	$\neg$	$(p$	$\to$	$q)$	$\to$	$\neg$	q
1	0	v:	1	(1	0	0)	1	1	0
							$v(P) = 1$	$\neg$	

La tabla de verdad de P es una tabla como la anterior pero completada con todas las interpretaciones posibles. Como en P aparecen dos átomos, hay $2^2 = 4$ interpretaciones posibles, que ordenaremos «alfabéticamente» dando prioridad a 1 sobre 0 (parece que empezar por verdadero infunde optimismo), de modo que la taba de verdad de P queda así:

p	q	P:	$\neg$	$(p$	$\to$	$q)$	$\to$	$\neg$	q
1	1	**1**	0	(1	1	1)	**1**	0	1
1	0	**1**	1	(1	0	0)	**1**	1	0
0	1	**1**	0	(0	1	1)	**1**	0	1
0	0	**1**	0	(0	1	0)	**1**	1	0
							P		

Habitualmente se escribe sólo la parte esencial de la información, marcando en todo caso la columna de valores finales (tabla izquierda a continuación), pero cuando no queremos ver todo el desarrollo del cálculo sino tan sólo sus resultados, entonces recogemos la información en forma funcional y decimos que esta es la *función de verdad* de P (tabla derecha), que en este caso es una función $2 \to 2$ constante:

$\neg$	$(p$	$\to$	$q)$	$\to$	$\neg$	q
0	(1	1	1)	**1**	0	1
1	(1	0	0)	**1**	1	0
0	(0	1	1)	**1**	0	1
0	(0	1	0)	**1**	1	0

p	q	P:
1	1	**1**
1	0	**1**
0	1	**1**
0	0	**1**

Lo que acabamos de ver en un ejemplo particular es válido en general. Cada proposición P con $n \geq 1$ átomos tiene una tabla de verdad con 2^n filas que calcula su función de verdad $2^n \to 2$.

No está de más repetir aquí las funciones de verdad de los conectores derivados, comprobando así que su defnición no fue al azar (Definición 1.1.2), sino basada en un conocimiento previo sobre su significado lógico:

P	Q	$P \vee Q$	$P \wedge Q$	$P \to Q$	$P \leftrightarrow Q$	$P \oplus Q$
1	1	1	1	1	1	0
1	0	1	0	0	0	1
0	1	1	0	1	0	1
0	0	0	0	1	1	0

Las proposiciones pueden llamarse de diversa forma atendiendo a cómo sea su función de verdad.

Definición 1.2.2 *(i) Un* modelo *de P es una interpretación $v\colon A \to 2$ tal que $v(P) = 1$. Si $v(P) = 0$ diremos que es un* contramodelo.

(ii) Una tautología *o ley lógica (resp.* contradicción*) es una proposición cuya función de verdad es la constante en 1 (resp. en 0), es decir, cada (resp. niguna) interpretación $v\colon A \to 2$ es un modelo de P.*

(iii) Una proposición se dice falsable *(resp.* consistente*) si no es una tautología (resp. contradicción).*

(iv) Una proposición se dice contingencia *si no es una tautología y no es una contradicción.*

La proposición $P = \neg(p \to q) \to \neg q$ utilizada unas líneas antes es una tautología y su negación será una contradicción. La proposición $p \to q$ es consistente y falsable.

Definición 1.2.3 *Dadas dos proposiciones P y Q, se dice que P* modela *a Q si todo modelo de P es un modelo de Q. La relación «P modela a Q» tiene un símbolo propio $P \models Q$. Por convenio se adopta la notación $\models Q$ para indicar que Q es una tautología*

La relación $P \models Q$ se expresa de diversos modos en el lenguaje habitual:

- Q es una *consecuencia semántica* de P.
- Q es una *condición necesaria* para P.
- P es una *condición suficiente* para Q.

Notemos además que se verifica la relación $P \models Q$ si, para cada interpretación v, se cumple cualquiera de las dos condiciones equivalentes siguientes:
si $v(P) = 1$ entonces $v(Q) = 1$
si $v(Q) = 0$ entonces $v(P) = 0$.

Un test para comprobar que se han comprendido bien las nociones de la definición anterior es dar conformidad a las afirmaciones siguientes:
(i) Toda tautología es consistente, pero el recíproco es falso.
(ii) Una proposición es consistente si y sólo si tiene un modelo.
(iii) Una proposición consistente que no es tautología es falsable.
(iv) $p \to p$ es una tautología y $\neg(p \to (\neg p \to p)$ es una contradiccion.
(vi) Si P es tautología entonces $Q \to P$ es tautología.
(vii) Si Q es contradicción entonces $Q \to P$ es tautología.

El enunciado «$P \to Q$ es tautología», es decir «$\models P \to Q$», requiere ser comprendido con cierta sutileza. Es claro que $p \to q$ no es una tautología, pero $P \to Q$ pudiera serlo según cuáles fueran las proposiciones P y Q; por ejemplo, si P es cualquiera y Q es una tautología. Pero si P no es tautología y Q sí, entonces $Q \to P$ no es tautología. La moraleja es que hemos de tener presente siempre si las letras significan proposiciones no especificadas o son los átomos del lenguaje. En general, en el primer caso usamos letras mayúsculas y en el segundo minúsculas, pero hay que poner atención porque en esto de las notaciones hay reglas de uso pero también excepciones (nótese que la expresión tan habitual «no hay regla sin excepción» es contradictoria).

Teorema 1.2.4 *Dadas proposiciones P y Q se verifica:*
(i) $P \models Q$ *si y sólo si* $\models P \to Q$.
(ii) La relación $P \models Q$ *es reflexiva y transitiva.*
(iii) La relación $P \models Q$ *no es simétrica ni antisimétrica.*

Demostración: Dejamos las demostraciones de (i) y (ii), que son muy sencillas, para que las componga el lector. Para probar (iii) tenemos que dar contraejem- plos:
a) Que se cumpla $P \models Q$ pero no $Q \models P$. Tómese$P = p$ y $Q = p \to p$.
b) Que se cumpla $P \models Q$ y $Q \models P$ pero no P = Q. Tómese $P = p$ y $Q = \neg\neg p$.

□

Haciendo entrar en el juego a los operadores derivados podemos escribir otra propiedad de la relación $\models$ que, una vez más, muestra la correlación que existe entre la lógica usual y la lógica que estamos construyendo formalmente:
Se verifica $P \models Q$ y $P \models R$ si y sólo si $P \models Q \wedge R$.
Se invita al lector a enunciar un resultado similar al anterior pero aplicable a la disyunción.
En el uso de la conjunción y la disyunción, estamos acostumbrados a enlazar varias proposiciones simples, no sólo dos. Para formalizar este uso, vamos a dar sentido como proposición a $P_1 \vee P_2 \vee \cdots \vee P_n$, con $n > 2$. Es claro que se define por inducción:

$$P_1 \vee \cdots \vee P_n = (P_1 \vee \cdots \vee P_{n-1}) \vee P_n$$

Una sencilla demostración por inducción prueba que la función de verdad de la proposición $P_1 \vee \cdots \vee P_n$ está determinada por

$$\nu(P_1 \vee \cdots \vee P_n) = 0 \text{ si y sólo si } \nu(P_1) = \cdots = \nu(P_n) = 0.$$

De modo análogo se define la conjunción múltiple, cuya función de verdad está dada por

$$\nu(P_1 \wedge \cdots \wedge P_n) = 1 \text{ si y sólo si } \nu(P_1) = \cdots = \nu(P_n) = 1.$$

Ahora es fácil probar el siguiente resultado:

Teorema 1.2.5 *Son equivalentes, con* $n, m \geq 1$, *los enunciados siguientes:*
(i) $P_1 \wedge \cdots \wedge P_n \models Q_1 \vee \cdots \vee Q_m$.
(ii) $\models \neg P_1 \vee \cdots \vee \neg P_n \vee Q_1 \vee \cdots \vee Q_m$.
(iii) $\models P_1 \wedge \cdots \wedge P_n \to Q_1 \vee \cdots \vee Q_m$.

Nótese que el Teorema 1.2.5 es una generalización del Teorema 1.2.4 (i) (caso $m = n = 1$ del anterior) teniendo en cuenta que
$\neg P \vee Q$ significa lo mismo que $P \to Q$.

Definición 1.2.6 *Dado un conjunto* $\Gamma = \{P_1, \cdots, P_n\} \subset \mathcal{P}(\mathcal{A})$ *y una interpretación* $v \colon \mathcal{A} \to 2$, *diremos que* v *es un modelo de* Γ *si* v *es modelo cada proposición de* Γ. *Dada una proposición* $P \in \mathcal{P}(\mathcal{A})$ *diremos que* Γ modela *a* P *o que* P *es una* consecuencia semántica *de* Γ, *si todo modelo de* Γ *es un modelo de* P. *Utilizaremos la notación* $\Gamma \models P$.

Es interesante observar que esta noción se puede extender a conjuntos Γ que sean infinitos. Notemos que el caso que $\Gamma = \emptyset = \{\}$, se tiene que $\{\} \models P$ si y sólo si P es una tautología.

Nota 1.2.7 *El el caso que consideremos el álgebra* $\mathcal{P}_\perp$ *las tablas de verdad en la que interviene* $\perp$ *se calculan teniendo en cuenta que* $\perp$ *siempre es falso. Por ejemplo si* $\mathcal{A} = \{p, q\}$

p	q	$\bot$	$\neg\bot$
1	1	0	1
1	0	0	1
0	1	0	1
0	0	0	1

Ejercicios
1. Sea la proposición $P = \neg p_1 \to ((\neg p_2 \wedge p_3) \vee (\neg p_4 \leftrightarrow p_2))$.
Calcular $v(P)$ cuando $v(p_n) = 0$ si n es par y $v(p_n) = 1$ si n es impar.
2. Decidir si son verdaderas o falsas las afirmaciones siguientes:
(i) Toda contradicción es falsable, pero el recíproco es falso.
(ii) Proposición *inconsistente* (no consistente) equivale a contradicción.
(iii) Una proposición falsable que no es contradicción es consistente.
(iv) Todo átomo es consistente y falsable.
3. Verificar que las siguientes proposiciones, llamadas *axiomas de Lukasiewicz*, son tautologías:
(L1) $P \to (Q \to P)$,
(L2)$(P \to (Q \to R)) \to ((P \to Q) \to (P \to R))$,
(L3) $(\neg P \to \neg q) \to (Q \to P)$.
(Indicación: En vez de hacer la tabla completa, suponer que se hizo y salió el valor 0 en una fila, entonces razonar marcha atrás reconstruyendo la fila tabla hasta encontrar una contradicción. Por ejemplo, si en una fila de la tabla de $P \to (Q \to P)$ sale 0, será que hubo 1 bajo P y 0 bajo $Q \to P$, lo que exige 0 bajo P, en contradicción con el 1 anterior, pues P no puede valer 1 y 0 en la misma fila de la tabla. Luego la proposición nunca valdrá 0, es por tanto una tautología.)
4. Verificar si las siguientes proposiciones son o no tautologías:
$P \to (P \to Q), (P \to (Q \to P)) \to ((P \to Q) \to (P \to R))$,
$(\neg P \to \neg q) \to (P \to Q)$.
(Indicación: Como en la indicación anterior, suponer valor cero, pero ahora para encontrar ya sea una contradicción o una interpretación que dé dicho valor.)
5. Demostrar el Teorema 1.2.5.

1.3. Equivalencias

En el conjunto $\mathscr{P}$ de las proposiciones vamos a introducir la relación de equivalencia lógica, que jugará un papel muy importante en lo que sigue.

Definición 1.3.1 *Dos proposiciones P y Q se dicen (lógicamente) equivalentes , y se escribe $P \equiv Q$, si tienen los mismos modelos.*

Es decir, la relación $P \equiv Q$ significa que para cada interpretación v es $v(P) = 1$ si y sólo si $v(Q) = 1$ (o lo mismo con 0). Al escribir «(lógicamente) equivalentes» se indica que generalmente diremos «equivalentes», pero indicando «lógicamente equivalentes» cuando queramos evitar alguna confusión, casi siempre debida al uso muy común del término «equivalentes». Un test para comprobar que se ha comprendido bien la noción de equivalencia lógica es dar conformidad a las afirmaciones siguientes:

(i) Dos átomos son equivalentes si y sólo si son iguales.
(ii) $P \equiv Q$ si y sólo si $P \models Q$ y $Q \models P$.
(iii) Si $\models P$ entonces $P \to Q \equiv Q$.
(iv) Si $\models \neg Q$ entonces $P \to Q \equiv \neg P$.
(v) Si en la equivalencia $p \equiv (p \to p) \to p$ se sustituye el átomo p por una proposición P cualquiera resulta otra equivalencia: $P \equiv (P \to P) \to P$.

Aunque son muy fáciles de demostrar, recogemos en forma de teorema unas propiedades sencillas y útiles sobre equivalencias y sustituciones.

Teorema 1.3.2 *Supongamos $P \equiv P'$. Se verifica:*
(i) $\neg P \equiv \neg P'$.
(ii) $P \to Q \equiv P' \to Q$.
(iii) $Q \to P \equiv Q \to P'$.

Aplicando reiteradamente el Teorema 1.3.2 resulta que si P forma parte de una proposición más compleja F y sustituimos P en F por una proposición equivalente P_0, resulta una proposición F_0 equivalente a F. Por ejemplo, sea $F = (p \to q) \to \neg p, P = \neg p$, $P_0 = p \to \neg p \equiv P$; haciendo la sustitución resulta $F_0 = (p \to q) \to (p \to \neg p) \equiv F$.

Cuando se inicia la lógica de proposiciones otorgando un sentido independiente a cada uno de los conectores, las igualdades de la Definición 1.1.2 adoptaban la forma de equivalencias. En el modo de exposición formal que seguimos en este capítulo, son más que equivalencias, son igualdades definitorias de símbolos nuevos; pero el resultado práctico final es el mismo. Muchas otras equivalencias se pueden introducir ahora, dentro del proceso expositivo formal. Por ejemplo esta que usamos implícitamente al relacionar los Teoremas 1.2.4 y 1.2.5:

$$P \to Q \equiv \neg P \vee Q = \neg\neg P \to Q$$

Nótese que, a su vez, la primera y la última expresión son equivalentes porque se diferencian en un fragmento equivalente (equivalencia de la $\leftrightarrow$ negación $P \equiv \neg\neg P$). Todas las equivalencias vistas de modo informal en el primer capítulo pueden ahora incorporarse a la teoría formal de la lógica de proposiciones.

Los nuevos conectores permiten reelaborar algunos de los resultados ya conocidos. Por ejemplo, el Teorema 1.2.4 se podría completar con un enunciado más: $P \equiv Q$ si y sólo si $\models P \leftrightarrow Q$.

Con $n = 3$, hemos definido $P_1 \vee P_2 \vee P_3 = (P_1 \vee P_2) \vee P_3$, pero la colocación de los paréntesis no importa desde el punto de vista semántico, pues se tiene la equivalencia asociativa $P_1 \vee P_2 \vee P_3 \equiv P_1 \vee (P_2 \vee P_3)$. Con un poco de trabajo inductivo se demuestra con todo detalle que esta propiedad es general: en una disyunción múltiple el orden en que se hacen las disyunciones binarias es semánticamente indiferente, por ejemplo,

$$P_1 \vee P_2 \vee P_3 \vee P_4 \equiv P_1 \vee ((P_2 \vee P_3) \vee P_4)).$$

Por otra parte, la equivalencia conmutativa $P \vee Q \equiv Q \vee$ P nos asegura que se pueden permutar las proposiciones sin que cambie el valor semántico de su disyunción. Por último, la equivalencia idempotente $P \vee P \equiv P$ nos permite suprimir repeticiones. En definitiva, para el valor semántico de una disyunción múltiple, lo único que importa es el conjunto de las proposiciones involucradas, recordando que en un conjunto no se repiten elementos y no importa el orden en que se dan:

$R \vee Q \vee P \vee Q \equiv P \vee Q \vee R$. Es claro que consideraciones análogas se pueden hacer sobre la conjunción múltiple. El teorema siguiente muestra las equivalencias de De Morgan en su forma general con conectores múltiples:

Teorema 1.3.3 *Se verifica, para cada $n \geq 1$:*
(i) $\neg(P_1 \vee \cdots \vee P_n) \equiv \neg P_1 \wedge \cdots \wedge \neg P_n$.
(ii) $\neg(P_1 \wedge \cdots \wedge P_n) \equiv \neg P_1 \vee \cdots \vee \neg P_n$.

Demostración: (i) El caso n = 2 se prueba mediante las tablas de verdad. Se termina la demostración por inducción. (ii) Se hace de modo análogo.

□

Podríamos haber demostrado (i) (y luego (ii) de modo análogo) utilizando resultados ya vistos antes: $v(\neg(P_1 \vee \cdots \vee P_n)) = 1$ si y solo si $v(P_1 \vee \cdots \vee P_n) = 0$ si y solo si $v(P_1) = \cdots = v(P_n) = 0$ si y solo si $v(\neg P_1) = \cdots = v(\neg P_n) = 1$ si y sólo si $v(\neg P_1 \wedge \cdots \wedge \neg P_n) = 1$.

Conviene observar que en el teorema anterior los apartados (i) y (ii) se implican mutuamente. Las dos partes del Teorema 1.3.3 se dicen duales. La dualidad consiste en intercambiar «$\vee$ »con «$\wedge$ »en la sintaxis y 0 con 1 en la semántica manteniendo la validez de los enunciados. La dualidad se aplica cuando con los conectores usados son $\{\neg, \wedge, \vee\}$. A este sistema de conectores nos vamos a referir en la próxima subsección.

Es importante señalar que se verifican las siguientes equivalencias básicas:

Teorema 1.3.4 *Sean $C, T \in \mathcal{P}(\mathcal{A})$ de manera que C es una contradicicón y T es una tautología. Entonces se verifican las siguientes equivalencias:*
(i) (idempotentes) $P \wedge P \equiv P \equiv P \vee P$.
(ii) (conmutativas) $P \wedge Q \equiv Q \wedge P, \quad P \vee Q \equiv Q \vee P$.
(iii) (asociativas) $P \wedge (Q \wedge R) \equiv (P \wedge Q) \wedge R, \quad P \vee (Q \vee R) \equiv (P \vee Q) \vee R$.
(iv) (absorción) $P \wedge (P \vee Q) \equiv P \equiv P \vee (P \wedge Q)$.
(v) (de la contradicción) $P \wedge C \equiv C, \quad P \vee C \equiv P$.
(vi) (de la tautología) $P \wedge T \equiv P, \quad P \vee T \equiv T$.
(vii) (distributiva $\wedge$) $P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R)$.
(viii) (distributiva $\vee$) $P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$.
(ix) (de la negación) $P \wedge \neg P \equiv C, \quad P \vee \neg P \equiv T$.

La demostración es sencilla y se deja como ejercicio. Es suficiente considerar algunas tablas de verdad construidas a partir de las subproposiciones que intervienen en las equivalencias anteriores.

1.4. Cláusulas y formas normales

Si en una proposición del sistema $\{\neg, \rightarrow\}$ aplicamos, en cada aparición del condicional, la equivalencia $P \rightarrow Q \equiv \neg P \vee Q$ obtendremos una proposición equivalente a la primera pero expresada en el sistema $\{\neg, \vee\}$. A su vez, aplicando una de las equivalencias de De Morgan (caso n = 2 del Teorema 1.3.3), podríamos encontrar otra proposición equivalente en la que aparecieran los conectores del sistema $\{\neg, \wedge\}$. En esta subsección tomaremos como preferente el sistema formado por los tres conectores $\{\neg, \wedge, \vee\}$. Entre las muchas equivalentes a una proposición dada que se pueden encontrar en este sistema, nos fijaremos en algunas que tienen una sintaxis muy especial.

Definición 1.4.1 *Dado un alfabeto $\mathscr{A}$, sea el conjunto $\neg\mathscr{A} = \{\neg p; p \in \mathscr{A}\}$. Llamaremos* literales *a los elementos del conjunto $\mathscr{L} = \mathscr{A} \cup \neg\mathscr{A}$. Los* átomos *serán los literales positivos y los átomos precedidos de la negación serán los* literales negativos. *Se llama* cláusula *a toda proposición de la forma $L_1 \vee \cdots \vee L_n$ en la que cada L_i es un literal. Una proposición tiene* forma clausal *si es una conjunción finita de cláusulas.*

Teorema 1.4.2 *Para cada proposición existe una proposición equivalente que tiene forma clausal. Dualmente, para cada proposición existe una proposición equivalente que tiene forma co-clausal.*

Demostración: La demostración consistirá en dar un procedimiento para obtener la equivalente en forma clausal de una proposición dada aplicando sucesivamente diversas equivalencias y simplificando todo lo posible. El procedimiento es:

(i) Escribir una equivalente en el sistema $\{\neg, \vee, \wedge\}$.

(ii) Desplazar todas las negaciones al interior de los paréntesis (De Morgan) y eliminar las dobles negación (ley de la doble negación), de modo que resulte una proposición equivalente en la que la negación sólo aparecerá en los literales.

(iii) Dar forma clausal, con las leyes asociativas y distributivas, al resultado del proceso anterior, utilizando las equivalencias de simplificación que sean necesarias.

□

El procedimiento descrito en la demostración anterior no es sino un cálculo algebraico que procede según las equivalencias, manejándolas como cuando usamos igualdades en el cálculo algebraico con los números, pero teniendo cuidado al calcular, porque no todas las reglas de cálculo del álgebra lógica coinciden con las del álgebra numérica. El procedimiento permite que se utilicen otras equivalencias para simplificar expresiones sin alterar los efectos del proceso. En particular, las equivalencias de conmutatividad y de idempotencia pueden usarse para ordenar (por ejemplo, preferir el orden alfabético o el numérico de los $\vee$ índices) y simplificar las proposiciones que van surgiendo, al igual que las leyes de no contradicción y de tercio excluso. Los ejemplos con dos átomos son muy sencillos, así que veremos uno con tres. La forma clausal de $P = (p \to (q \wedge r)) \to \neg(r \vee q)$ se obtiene así:

$P \equiv \neg(\neg p \vee (q \wedge r)) \vee \neg(r \vee q) \equiv (p \wedge (\neg q \vee \neg r)) \vee (\neg r \wedge \neg q) \equiv$

$\equiv (p \vee \neg q) \wedge (p \vee \neg r) \wedge (\neg q \vee \neg r)$.

Hemos preferido ordenar según el alfabeto, pero se podría haber preferido expresar la forma clausal con una ordenación circular:

$P \equiv (p \vee \neg q) \wedge (\neg q \vee \neg r) \wedge (\neg r \vee p)$

Nótese que toda tautología admite una forma clausal como $p \vee \neg p$ (una cláusula) y toda contradicción una como $p \wedge \neg p$ (dos cláusulas). El procedimiento anterior no se puede estandarizar del todo porque la forma clausal no es única, pero eso no importa ahora. Para que sea única, salvo el orden, la forma clausal tiene que tener alguna condición adicional que veremos a continuación. Pero las formas clausales en general tendrán utilidad en sí mismas cuando lleguemos al método de resolución.

Ahora vamos a trabajar con un alfabeto finito $\mathscr{A} = \{p_1, \cdots, p_n\}$ y el álgebra de las proposiciones construidas con dicho alfabeto y los conectores $\{\neg, \wedge, \vee\}$. Utilizando proposiciones $p_i \vee \neg p_i$, $p_i \wedge \neg p_i$ se puede hacer que cada proposición P tenga una equivalente en la que aparezcan todos los átomos, así que supondremos que las proposiciones cumplen esta condición.

Definición 1.4.3 *Dado el alfabeto* $\mathcal{A} = \{p_1, \cdots, p_n\}$ *una cláusula se dice* estándar *si contiene n literales sin átomos repetidos y ordenados según el subíndice. Una proposición tiene* forma normal conjuntiva *si es una conjunción finita de cláusulas estándar distintas.*

Es claro que el número de cláusulas estándar distintas es 2^n, tantas como subconjuntos podemos hacer con los n átomos. Este número coincide también con el de las interpretaciones posibles. Podemos precisar más esta relación.

Lema 1.4.4 *Sea* $\mathcal{A} = \{p_1, \cdots, p_n\}$ *un alfabeto finito. Para cada cláusula estándar* C *existe una única interpretación* $v\colon \mathcal{A} \to 2$ *tal que* $v(C) = 0$. *Recíprocamente, para cada interpretación* v *existe una única cláusula* C *tal que* $v(C) = 0$.

Demostración: Sea $C = L_1 \vee \cdots \vee L_n$, con tantos literales como átomos, unos negativos y otros positivos. Es claro que es $v(C) = 0$ si y sólo si $v(L_i) = 0$ para cada $i \in \{1, \cdots, n\}$. Así que la interpretación v vendrá dada por

$$v(p_i) = \begin{cases} 0 \text{ si } L_i = p_i \\ 1 \text{ si } L_i = \neg p_i \end{cases}$$

Recíprocamente, una interpretación v determina los literales L_i y una única cláusula $C = L_1 \vee \cdots \vee L_n$ tal que $v(C) = 0$.

□

Un ejemplo con tres átomos aclarará el formalismo: La tabla de verdad de la cláusula $C = \neg p \vee q \vee \neg r$ vale 0 cuando $(p, q, r) = (1, 0, 1)$ y sólo en esa fila, en las demás vale 1.

Ahora resulta claro también que cada forma normal conjuntiva es falsable y que el número de ellas es $2^{2^n} - 1$, pues no contamos el caso «vacío de cláusulas». Este número coincide con el de funciones booleanas de n variables si descontamos el caso «vacío de ceros», esto es, descontando la función de verdad de las tautologías con n átomos. De nuevo podemos precisar más este resultado en forma de teorema.

Teorema 1.4.5 *Cada proposición falsable admite una única proposición (salvo el orden de las cláusulas) equivalente con forma normal conjuntiva.*

Demostración:: Sea una proposición P con n átomos y sean $v_1, \cdots, v_k$, con $\leq k \leq 2^n$, las interpretaciones tales que $v(P) = 0$ (las filas de la tabla de P con valor 0). Según el Lema 1.4.4, a cada una de ellas le corresponde una cláusula estándar C_k. Se tiene entonces la siguiente forma normal conjuntiva equivalente a P:

$$P \equiv C_1 \wedge \cdots \wedge C_k$$

Resulta por tanto que la forma normal conjuntiva es una «tabla de verdad algebraica», pues da la misma información que la tabla de verdad semántica, habiendo una cláusula estándar por cada cero de la función de verdad. Nos indica, pues, en qué filas de la tabla la proposición es falsa.

A la forma normal conjuntiva puede llegarse mediante cálculo algebraico, añadiendo un par de pasos al proceso ya indicado para calcular la equivalente en forma clausal. Los pasos que hay que añadir son:

(iv) Completar las cláusulas para que sean estándar. Cada vez que se añade un átomo que falta hay que añadir otra claúsula similar con el átomo negativo correspondiente.

(v) Suprimir claúsulas repetidas y ordenar. □

Notemos que si tenemos una proposición P y un átomo $a \in \mathcal{A}$, se tiene que
$P \equiv P \vee (a \wedge \neg a) \equiv (P \vee a) \wedge (P \vee \neg a)$ y también $P \equiv P \wedge (a \vee \neg a) \equiv (P \wedge a) \vee (P \wedge \neg a)$.

A manera de ejemplo, completaremos el dado antes calculando la forma normal conjuntiva de $Q \equiv (p \vee \neg q) \wedge (p \vee \neg r) \wedge (\neg q \vee \neg r)$.

Completar la primera cláusula significa poner $Q \vee \neg q \equiv (p \vee \neg q \vee r) \wedge (p \vee \neg q \vee \neg r)$.

Haciendo lo mismo con las otras dos cláusulas iniciales incompletas resulta una conjunción de ocho cláusulas estándar, pero suprimiendo las repetidas quedan cuatro, que ordenadas a conveniencia dan:

$Q \equiv (\neg p \vee \neg q \vee \neg r) \wedge (p \vee \neg q \vee \neg r) \wedge (p \vee \neg q \vee r) \wedge (p \vee q \vee \neg r)$

Basta un vistazo para ver cómo se escribe esta forma normal conjuntiva directamente a partir de la función de verdad de P, tomando una cláusula estándar por cada 0:

p	q	r	Q	
1	1	1	0	$\neg p \vee \neg q \vee \neg r$
1	1	0	1	
1	0	1	1	
1	0	0	1	
0	1	1	0	$p \vee \neg q \vee \neg r$
0	1	0	0	$p \vee \neg q \vee r$
0	0	1	0	$p \vee q \vee \neg r$
0	0	0	1	

Se puede hacer un proceso dual, con cálculo algebraico y con semántica, para probar el siguiente teorema, cuya explicación detallada, definiciones incluidas, dejamos para el lector:

Teorema 1.4.6 *Cada proposición consistente admite una única proposición equivalente con forma normal disyuntiva.*

Como ejemplo, volvemos a la función de verdad de la proposición utilizada en los últimos ejemplos, $Q = (p \rightarrow (q \wedge r)) \rightarrow \neg(r \vee q)$, mostrando ahora las piezas de su forma normal disyuntiva, una por cada valor 1 de su función de verdad. La forma normal disyuntiva de Q es la disyunción de las cuatro piezas señaladas:

p	q	r	Q	
1	1	1	0	
1	1	0	1	$p \wedge q \wedge \neg r$
1	0	1	1	$p \wedge \neg q \wedge r$
1	0	0	1	$p \wedge \neg q \wedge \neg r$
0	1	1	0	
0	1	0	0	
0	0	1	0	
0	0	0	1	$\neg p \wedge \neg q \wedge \neg r$

La forma normal disyuntiva de Q es por tanto:

$Q \equiv (p \wedge q \wedge \neg r) \vee (p \wedge \neg q \wedge r) \vee (p \wedge \neg q \wedge \neg r) \vee (\neg p \wedge \neg q \wedge \neg r)$

Ya que dos proposiciones son equivalentes si sus tablas dan la misma función de verdad y, por otra parte, las formas normales representan dichas funciones, resulta que dos proposiciones son equivalente si y sólo si sus formas normales (conjuntiva o disyuntiva) coinciden salvo en el orden.

Nota 1.4.7 *Si se ha considerado el símbolo $\perp$ se podría tomar como forma normal conjuntiva de una contradicción precisamente $\perp$. Dualmente, la forma normal disyuntiva de una tautología sería $\neg\perp$ (o $\top$ en el caso que también se incluya este símbolo.)*

Ejercicios
1. Sea la proposición $P = (p \wedge q) \to (r \wedge s))$. Se pide:
(i) Escribir una equivalente a P en forma clausal.
(ii) Completar la forma clausal de (i) hasta la forma normal conjuntiva.
(iii) Calcular la forma normal disyuntiva de P.
2. Hallar la forma normal disyuntiva de las tautologías:
(L1) $P \to (Q \to P)$,
(L2)$(P \to (Q \to R)) \to ((P \to Q) \to (P \to R))$,
(L3) $(\neg P \to \neg q) \to (Q \to P)$.

Capítulo 2

Lógica algebraica

Como ya vimos, el conjunto $\mathscr{P} = \mathscr{P}(\mathscr{A})$ de todas las proposiciones obtenidas a partir de un alfabeto $\mathscr{A}$ y el sistema de conectores$\{\neg, \to\}$ está provisto de dos operaciones:

$$\neg\colon \mathscr{P} \to \mathscr{P} \quad \text{y} \quad \to\colon \mathscr{P} \times \mathscr{P} \to \mathscr{P},$$

con las que es un álgebra de tipo (1,2) o una (1,2)-álgebra. Repasando mentalmente los ejemplos de operaciones que se conocen con los números se observa que hay muchos ejemplos de álgebras de este tipo. Veremos que entre ellas juega un papel muy destacado el álgebra $\mathscr{P}$ de las proposiciones. Luego analizaremos la dependencia de $\mathscr{P}$ respecto del alfabeto $\mathscr{A}$, lo que da lugar a la aparición de nuevas estructuras de datos frecuentes en matemáticas, con su correspondiente vocabulario.

En la segunda parte del capítulo nos encontraremos con álgebras de tipo (0,0,1,2,2) dadas por las constantes 0, 1 y las operaciones $\{\neg, \wedge, \vee\}$, es decir, la primera 1-aria y las otras dos binarias. En este caso será protagonista el álgebra $\mathscr{B}$ de las proposiciones módulo equivalencia lógica. Será muy interesante notar las diferencias entre los casos $\mathscr{P}$ y $\mathscr{B}$. En $\mathscr{P}$ dos escrituras sintáctimamente diferentes definen elementos diferentes (álgebra libre), mientras que en $\mathscr{B}$ se dan igualdades entre expresiones sintácticas (álgebra cociente), que no son otra cosa que la contrapartida algebraica de las equivalencias lógicas. $\mathscr{P}$ y $\mathscr{B}$ son álgebras de tipo (1,2,2) con las operaciones $\{\neg, \wedge, \vee\}$, pero en la segunda aparecen también las constantes 0, 1, que no están e la primera.

Se comprende mejor el significado sintáctico del conjunto $\mathscr{P}$ si lo comparamos con otros conjuntos que tengan, como él, dos operaciones con una y dos variables respectivamente, que denotaremos (cuando hablemos en general, no así necesariamente en ejemplos particulares) con los mismos símbolos que los usados para las de $\mathscr{P}$, pues eso simplifica la notación.

2.1. El álgebra libre $\mathscr{P}(\mathscr{A})$

Veamos un primer ejemplo de álgebra similar a $\mathscr{P}$. Sea el intervalo real unidad $I = [0,1] \subset \mathbb{R}$ con las operaciones

$\neg x = 1 - x$, $x \to y = \text{mín}\{1, 1 - x + y\}$.

Cualquier proposición se puede interpretar y calcular en I sin más que indicar el valor que deben tomar los átomos. Tomando por ejemplo la proposición $P = \neg(p \to q) \to \neg q$ y asignando los valores $v(p) = \frac{1}{2}$ $v(q) = \frac{1}{3}$, resulta el siguiente valor para P:

$$\bar{v}(P) = \neg(v(p) \to v(q)) \to \neg v(q) = \neg(\frac{1}{2} \to \frac{1}{3}) \to \neg\frac{1}{3} = 1$$

Otro ejemplo para aficionados a la aritmética. Sea el conjunto $D(n)$ de los divisores de un número natural fijo $n > 1$, con las operaciones:

$\neg a$ el mayor divisor primo con a, $a \in D(n)$,

$a \to b$ el mayor divisor cuyo máximo común divisor con a divide a b

Tomemos $n = 30$ y demos a los átomos lo valores $v(p) = 10$, $v(q) = 6$, entonces resulta el siguiente valor para la proposición P del ejemplo anterior:

$$\bar{v}(P) = \neg(v(p) \to v(q)) \to \neg v(q) = \neg(10 \to 6) \to \neg 6 = 30$$

Lo que hemos hecho en estos ejemplos se puede extender sin dificultad al caso general expuesto en el enunciado siguiente y reflejado en el diagrama que acompaña.

Teorema 2.1.1 *Sea un conjunto A con operaciones $\neg, \to$, de una y dos variables respectivamente. Para cada aplicación $v\colon \mathscr{A} \to A$ existe una única aplicación $\bar{v}\colon \mathscr{P} \to A$ tal que:*
(i) Para cada $p \in \mathscr{A}$, $\bar{v}(p) = v(p)$ ($\bar{v}$ es una extensión de v).
(iii) $v(\neg P) = \neg v(P)$, y $v(P \to Q) = v(P) \to v(Q)$ (v conserva las operaciones).

Demostración: Para definir $v(P)$ se procede por inducción sobre la complejidad de P, es decir, siguiendo los pasos de la Definición inductiva de la proposición. Primero se ha de definir $\bar{v}(p)$ con $p \in \mathscr{A}$, lo que se hace con la condición (i). Luego, supuesto que se conoce $\bar{v}(P)$ se define $\bar{v}(\neg P)$, lo que se hace usando la primera parte de la condición (ii). Finalmente, supuesto que se conocen $\bar{v}(P)$ y $\bar{v}(Q)$ se define $\bar{v}(P \to Q)$ usando la segunda parte de la condición (ii).

□

Las aplicaciones $\bar{v}$ y v se llaman interpretaciones. Es costumbre denotarlas mediante la misma letra, una vez que se ha comprendido que la primera es una extensión unívoca de la segunda. El resultado es que interpretar los átomos conduce automáticamente a interpretar las proposiciones, siempre que la interpretación se haga en un conjunto con operaciones correspondientes a las de $\mathscr{P}$ y conservando estas operaciones. Así lo hicimos en la Definición1.2.1, en la que tomamos $\mathscr{A} = 2$, pero el mecanismo algebraico de la interpretación se puede hacer en cualquier álgebra con el mismo tipo de operaciones que $\mathscr{P}$.

La interpretación extendida $\bar{v}\colon \mathscr{P} \to A$ relaciona dos álgebras del mismo tipo (1,2) de tal modo que $\bar{v}$ conserva las operaciónes según la propiedad (ii) del Teorema 2.1.1. Por tener esta propiedad se dice que $\bar{v}$ es un morfismo de (1,2)-álgebras. Con esta nueva terminología podemos parafrasear el teorema de modo siguiente: Dada una (1,2)-álgebra A, toda aplicación $v\colon \mathscr{A} \to A$ se extiende de modo único a un morfismo $v\colon \mathscr{P} \to A$ de (1,2)-álgebras. O de otro modo: para cada (1,2)-álgebra A y cada aplicación $v\colon \mathscr{A} \to A$ existe un único morfismo $\bar{v}\colon \mathscr{P} \to A$ que extiende a $v\colon \mathscr{A} \to A$.

El Teorema 2.1.1 y su diagrama expresan una propiedad universal de la inclusión $\mathscr{A} \to \mathscr{P}$. Debe notarse que esta inclusión en un ejemplo de interpretación v posible, al que le corresponde $\bar{v} = id : \mathscr{P} \to \mathscr{P}$.

$$\begin{tikzcd}
\mathscr{A} \arrow[r] \arrow[d] & \mathscr{P} \arrow[dl, "\bar{v}"] \\
\mathscr{P} &
\end{tikzcd}$$

En relación con las equivalencias ya conocimos el proceso de sustituir una proposición fragmento de otra mayor por una tercera. Ahora podemos ver este proceso de sustituir como un caso particular de la propiedad universal que nos ocupa. Veamos un ejemplo. Si en la proposición usada unos párrafos antes, $P = \neg(p \to q) \to \neg q$, sustituimos el átomo q por la proposición $Q = \neg q \to p$ resulta una nueva proposición $\bar{P} = \neg(p \to (\neg q \to p)) \to \neg(\neg q \to p)$.

En efecto, diseñar la sustitución de átomos por proposiciones significa dar una interpretación $v : A \to \mathscr{P}$, que en el ejemplo anterior ha sido

$$v(p) = \begin{cases} p & \text{si } p \neq q, \\ Q & \text{si } p = q \end{cases}$$

de modo que resulta $\bar{v}(P) = \bar{P}$.

Ahora iremos un poco más allá en el formalismo algebraico, pero lo que sigue (excepto los ejercicios al final de este apartado) puede dejarse para una segunda lectura porque no será necesario más adelante.

Fijados los símbolos lógicos primitivos, el conjunto $\mathscr{P}$ de las proposiciones depende de alfabeto $\mathscr{A}$, por eso escribimos $\mathscr{P} = \mathscr{P}(A)$ dando a la letra $\mathscr{P}$ dos sentidos diferentes con la esperanza de que el contexto permita separarlos. Por una parte representa el proceso de formar conjuntos de proposiciones con símbolos lógicos primitivos predeterminados y un alfabeto que puede variar, y por otra representa el resultado de aplicar el proceso a un alfabeto concreto. Este matiz es relevante y cuando manejamos varios alfabetos $\mathscr{A}$, $\mathscr{A}'$ etc., escribiendo entonces $\mathscr{P} = \mathscr{P}(\mathscr{A})$, $\mathscr{P}' = \mathscr{P}(\mathscr{A}'))$, etc. En presencia de dos alfabetos, una aplicación $\alpha\colon \mathscr{A}' \to \mathscr{A}$ se extiende (ampliando el conjunto de llegada o codominio) a una interpretación $\mathscr{A} \to \mathscr{P}$, la cual a su vez se extiende (por el dominio, gracias al Teorema 2.1.1) a un morfismo $\mathscr{P}(\alpha)\colon \mathscr{P}' \to \mathscr{P}'$, lo que produce un efecto similar al de una sustitución de variables. La situación descrita viene reflejada en el diagrama conmutativo siguiente:

$$\begin{tikzcd} \mathscr{A}' \arrow[r] \arrow[d, "\alpha"] & \mathscr{P}' \arrow[d, "\mathscr{P}(\alpha)"] \\ \mathscr{A} \arrow[r] & \mathscr{P} \end{tikzcd}$$

Esta construcción tiene las propiedades siguientes:

$$\mathscr{P}(id) = id \quad \mathscr{P}(\alpha\alpha') = \mathscr{P}(\alpha)\mathscr{P}(\alpha')$$

La primera tiene sentido cuando los alfabetos son iguales, mientras que la segunda se aplica a dos cambios de alfabeto consecutivos $\alpha'\colon \mathscr{A}'' \to \mathscr{A}'$ y $\alpha\colon \mathscr{A}' \to \mathscr{A}$.

Este tipo de construcciones que conservan las identidades y la composici ón se presentan frecuentemente en matemáticas y por ello reciben un nombre: funtores. Se dice que la construcción $\mathscr{P}$ es un funtor. Pero, al igual que en una aplicación hay que tener claros su dominio y su codominio, que son conjuntos, en un funtor hay que tener claro dónde empieza la construcción y dónde termina, y qué tipo de datos hay al principio y al final. Si nos fijamos en el diagrama anterior, antes de la construcción hay conjuntos y aplicaciones entre ellos, después de la construcción hay álgebras de tipo (1,2) y morfismos entre ellas. Los datos básicos sobre conjuntos y aplicaciones son: que las aplicaciones se componen (cuando una termina donde la otra empieza) siendo la composición asociativa y que cada conjunto tiene una aplicación identidad que deja invariables a las demás aplicaciones con las que se puede componer. Se expresa este hecho diciendo que los conjuntos y las aplicaciones entre ellos forman una categoría, la categoría de los conjuntos denotada *CONJ*. Pero si nos fijamos ahora en las (1,2)-álgebras y los morfismos entre ellas vemos que el esquema se repite: la composición (cuando se puede, siendo entonces asociativa) de dos morfismos es de nuevo un morfismo, las identidades son morfismos; de modo que podemos hablar también de la categoría de las (1,2)-álgebras, que denotaremos $ALG(1,2)$. Ahora podemos ya decir que $\mathcal{P}$ es un funtor de la categoría de los conjuntos en la categoría de las (1,2)-álgebras, lo que se escribe simbólicamente en la forma

$$\mathcal{P}\colon CONJ \to ALG(1,2)$$

Como se ve, la fecha es un símbolo que vale para muchas cosas. Así sucede porque o bien se utiliza la polisemia de los símbolos, dejando que sea el contexto el que aclare de qué significado se trata, o bien hay que recurrir a una cantidad mucho mayor de símbolos de los que usamos habitualmente.

Ejercicios

1. Sea el conjunto $L = \{0, \frac{1}{2}, 1\}$ en el que consideramos las operaciones (que se pueden indicar mediante tablas) dadas por las fórmulas: $x \to y = \text{mín}\{1, 1 - x - y\}$, $\neg x = x \to 0$. En estas condiciones, suponer la proposición $P = \neg p_1 \to ((\neg p_2 \wedge p_3) \vee (\neg p_4 \leftrightarrow p_2))$ y calcular $v(P)$ en L siendo el valor de los átomos $v(p_n) = \frac{r}{2}$, donde r es el resto mínimo positivo de n módulo 3.

2. Sea la proposición $P = \neg(p_1 \to ((\neg p_2 \wedge p_3) \vee (\neg p_4 \leftrightarrow p_2))$. Valorar P en el álgebra $\mathcal{P}(\mathcal{A})$ con las operaciones usuales, cuando el valor de los átomos es: $\alpha(p_n) = p_{n+1}$ si n es impar, y $\alpha(p_n) = p_{n-1}$ si n es par.

3. Valorar la proposición P del ejercicio anterior en el conjunto $\mathcal{P}(\mathcal{A}')$ con las operaciones usuales, siendo el nuevo alfabeto $\mathcal{A}' = \{p, q, r\}$ y el valor de los átomos es: $\alpha(p_n) = p, q, r$ según n sea, respectivamente, congruente con 0, 1, 2, módulo 3.

2.2. El álgebra de Boole $\mathcal{B}(\mathcal{A})$

En el conjunto $\mathcal{P} = \mathcal{P}(\mathcal{A})$ tenemos la relación de equivalencia lógica $\equiv$, que es una relación de equivlencia y nos permite definir el correspondiente conjunto de clases de equivalencia (conjunto cociente):

$$\mathcal{B}(\mathcal{A}) = \mathcal{P}(\mathcal{A})/\equiv .$$

Si $P \in \mathcal{P}(\mathcal{A})$ denotaremos por

$$[P] = \{P' \in \mathcal{P}(\mathcal{A}) | P \equiv P'\}$$

la clase de equivalencia de P.

Supongamos que $x = [P]$ e $y = [Q]$. Escribir $[P] = [Q]$ $(x = y)$ significa lo mismo que escribir $P \equiv Q$. Observemos que si el alfabeto $\mathcal{A}$ es finito, cada clase de equivalencia $x \in \mathcal{B}(\mathcal{A})$ podrá escribirse de la forma $x = [P]$ tomando P en forma normal conjuntiva o disyuntiva. De aquí se obtiene que en este caso $\mathcal{B}(\mathcal{A})$ tiene un número finito de elementos.

En $\mathcal{B}(\mathcal{A})$ hay dos elementos especiales: La clase $1 \in \mathcal{B}(\mathcal{A})$ de las tautologías (todas equivalentes entre sí) y la clase $0 \in \mathcal{B}(\mathcal{A})$ de las contradicciones (todas equivalentes entre sí).

Diremos que estos dos elementos son constantes u operaciones 0-arias en el álgebra $\mathcal{B}(\mathcal{A})$, cuyas operaciones vamos a ir definiendo poco a poco. De momento se trata de un álgebra de tipo $(0, 0, \cdots)$. Para extender la operación $\neg$ de $\mathcal{P}(\mathcal{A})$ a una operación $\neg$ en $\mathcal{B}(\mathcal{A})$ parece natural definir

$$\neg[P] = [\neg P],$$

pero tenemos que asegurarnos de que esta Definición no depende del representante elegido en la clase de equivalencia, es decir, la Definición está bien formulada si se cumple que si $P \equiv Q$, entonces $\neg P \equiv \neg Q$, propiedad que es cierta como consecuencia del Teorema 1.3.2.

De modo que tenemos una operación

$$\neg \colon \mathcal{B}(\mathcal{A}) \to \mathcal{B}(\mathcal{A})$$

y $\mathcal{B}(\mathcal{A})$ pasa a ser un álgebra de tipo $(0, 0, 1, \cdots)$. Las operaciones que vamos definiendo tienen algunas propiedades interesantes que no son sino consecuencias de propiedades de las proposiciones, por ejemplo: $\neg 1 = 0$, $\neg 0 = 1$, $\neg\neg x = x$.

Un proceso análogo podemos seguir con la operación $\wedge$ de $\mathcal{P}(\mathcal{A})$, que extendemos a una operación $\wedge$ en $\mathcal{B}(\mathcal{A})$ definiendo

$$[P] \wedge [Q] = [P \wedge Q]$$

lo que es una Definición correcta porque se verifica que si $P \equiv P'$, $Q \equiv Q'$, entonces aplicando el Teorema 1.3.2, se tiene que $P \wedge Q \equiv P' \wedge Q'$.

Haciendo lo mismo con el conector $\vee$, resulta que $\mathcal{B}(\mathcal{A})$ pasa a ser un álgebra de tipo $(0, 0, 1, 2, 2)$, es decir, con dos constantes $1, 0$, la operación $\neg$ de una variable, y dos operaciones, $\wedge, \vee$, de dos variables.

Podríamos seguir extendiendo más conectores a operaciones en $\mathcal{B}(\mathcal{A})$, pero lo dejamos aquí porque tenemos las necesarias para definir un tipo de álgebra con mucha solera.

Definición 2.2.1 *Se llama álgebra de Boole de las proposiciones sobre un alfabeto $\mathcal{A}$ al álgebra de tipo* $(0, 0, 1, 2, 2)$ *formada por el conjunto $\mathcal{B}(\mathcal{A})$ con las operaciones* $1, 0, \neg, \wedge, \vee$.

Utilizando el lenguaje habitual del álgebra podemos llamar polinomios en $\mathcal{B}(\mathcal{A})$ a todas las expresiones formadas con elementos y operaciones respetando las mismas reglas sintácticas que utilizamos para escribir proposiciones. Por ejemplo: $x, x \wedge y, (x \wedge y) \wedge z$. Entre estos elementos se tienen igualdades, como por ejemplo: $x \wedge \neg x = 0$, $x \wedge y = y \wedge x$. La diferencia clave entre las álgebras $\mathcal{P}(A)$ y $\mathcal{B}(\mathcal{A})$ es que en la primera los elementos se diferencian por su sintaxis, mientras que en la segunda hay igualdades entre elementos que se escriben de diferente manera, como el la proposición anterior. Sucede así porque las igualdades en $\mathcal{B}(\mathcal{A})$ se corresponden con las equivalencias en $\mathcal{P}$.

De todas las igualdades posibles en $\mathcal{B}(\mathcal{A})$ vamos a seleccionar una lista que enunciaremos en formato de teorema, pero no necesitaremos escribir la demostración porque cada igualdad se corresponderá con una equivalencia bien conocida.

Teorema 2.2.2 *Las siguientes igualdades son ciertas en $\mathcal{B}(\mathcal{A})$:*

(i) (idempotentes) $x \wedge x = x = x \vee x$.

(ii) (conmutativas) $x \wedge y = y \wedge x, x \vee y = y \vee x$.

(iii) (asociativas) $x \wedge (y \wedge z) = (x \wedge y) \wedge z, x \vee (y \vee z) = (x \vee y) \vee z$.

(iv) (absorción) $x \wedge (x \vee y) = x = x \vee (x \wedge y)$.

(v) (del cero) $x \wedge 0 = 0, x \vee 0 = x$.
(vi) (del uno) $x \wedge 1 = x, x \vee 1 = 1$.
(vii) (distributiva $\wedge$) $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$.
(viii) (distributiva $\vee$) $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$.
(ix) (de la negación) $x \wedge \neg x = 0, x \vee \neg x = 1$.

Veamos un ejemplo de cómo el cálculo algebraico nos permite verificar equivalencias. Para averiguar si es cierta en $\mathcal{B}(\mathcal{A})$ la igualdad

$$(x \wedge (x \vee z) \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$$

tenemos dos procedimientos:

El semántico. Podemos descender al significado de los elementos de $\mathcal{B}(\mathcal{A})$, notando que si $x = [P]$, $y = [Q]$, $z = [R]$, entonces la igualdad propuesta se reduce a la equivalencia en $\mathcal{P}$

$$(P \wedge (P \vee R) \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$$

que demostramos mediante las tablas de verdad.

El algebraico. Podemos calcular de modo algebraico directamente, usando las igualdades primitivas del Teorema 2.2.2:

$$(x \wedge (x \vee z) \vee (y \wedge z) = x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$$

En la primera igualdad se ha usado una absorción (iv) y en la segunda la distributiva (viii).

Calcular directamente con tablas puede ser algo pesado y requiere conocer las tablas de los conectores básicos, mientras que el cálculo algebraico es más fácil y rápido siempre que se conozcan las equivalencias básicas.

Debe notarse que el procedimiento algebraico no necesita obligatoriamente recurrir a la notación $x, y, z, \cdots$ para las clases de equivalencia de $\mathcal{B}(\mathcal{A})$, puede también realizare, y en la práctica muchas veces se procede así, calculando directamente con las proposiciones $P, Q, R \cdots$ de $\mathcal{P}(\mathcal{A})$ módulo equivalencia lógica. Así, el cálculo del ejemplo anterior sería, usando absorción y distributividad,

$$(P \wedge (P \vee R) \vee (Q \wedge R) \equiv P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R).$$

Es imprescindible hacer una última observación importante sobre este asunto. Todas las igualdades del Teorema 2.2.2 corresponden a equivalencias de proposiciones, de modo que todas las igualdades que resulten de encadenar cualesquiera de las anteriores un número finito de veces corresponderán igualmente a equivalencias de proposiciones. En sentido inverso, las nueve igualdades del Teorema 2.2.2 forman un conjunto completo porque a partir de ellas se puede llegar a obtener por el procedimiento algebraicos cualquier equivalencia lógica; pero esta última afirmación no es tan inmediata como la anterior, de hecho es un teorema de completitud cuya demostración no daremos en este curso introductorio, pero sí es conveniente conocerlo y saber que en él se basa la eficacia del procedimiento algebraico.

No sólo la equivalencia de proposiciones puede tratarse por el procedimiento algebraico, también otras relaciones asociadas.

Definición 2.2.3 *Dados dos elementos* $x = [P]$, $y = [Q] \in \mathcal{B}(\mathcal{A})$, *diremos que* $x \leq y$ *se se verifica que* $P \models Q$.

Esto puede hacerse porque si $P \equiv P'$ y $Q \equiv Q'$, entonces $P \models Q$ equivale a $P' \models Q'$.

Proposición 2.2.4 *La expresión $x \leq y$ define una relación de orden en $\mathcal{B}(\mathcal{A})$ que tiene como elemento máximo a 1 y como elemento mínimo a 0. Además, las expresiones siguientes son equivalentes:*
(i) $x \leq y$,
(ii) $x = x \wedge y$,
(iii) $x \vee y = y$.

Demostración: Siendo $x = [P]$, $y = [Q]$, basta observar la equivalencia entre los siguientes enunciados: (i) $P \models Q$, (ii) $P \equiv P \wedge Q$ y (iii) $P \vee Q \equiv Q$, En particular, se verifica $\models P$ si y sólo si la clase $x = [P]$ verifica $x = 1$, de modo que el cálculo algebraico sirve también para verificar tautologías. Análogamente, P es una contradicción si y sólo si $x = 0$. □

Ejercicios:
1. Plantear el Teorema 1.2.4 y el Corolario 1.2.5 como resultados propios del álgebra de Boole $\mathcal{B}(\mathcal{A})$.
2. Verificar por el procedimiento algebraico si son o no proposiciones equivalentes las siguientes: $R \to (Q \to P)$ y $R \wedge Q \to P$.
3. Verificar por el procedimiento algebraico que los llamados axiomas de Lukasiewicz son tautologías:
(L1) $P \to (Q \to P)$,
(L2) $(P \to (Q \to R)) \to ((P \to Q) \to (P \to R))$,
(L3) $(\neg P \to \neg Q) \to (Q \to P)$.
4. Verificar por el procedimiento algebraico si las siguientes proposiciones son o no tautologías:
$P \to (P \to Q), (P \to (Q \to P$» $\to$ «$P \to Q) \to (P \to R$»,
$(\neg P \to \neg q) \to (P \to Q)$.
5. Hallar $\mathcal{B}(\mathcal{A})$ cuando $\mathcal{A} = \{p\}$.

2.3. Álgebras de Boole abstractas

Las características del álgebra de Boole de las proposiciones sobre un alfabeto, cuando se enuncian para un álgebra cualquiera del mismo tipo, dan lugar al concepto abstracto de álgebra de Boole, concepto general del que, naturalmente, las álgebras $\mathcal{B}(\mathcal{A})$ pasan a ser ejemplos particulares.

Definición 2.3.1 *Se llama álgebra de Boole a un conjunto B con operaciones cualesquiera* $0, 1, \neg, \wedge, \vee$ *de los tipos correlativos* $(0, 0, 1, 2, 2)$*, que verifique todas las siguientes igualdades:*
(i) (idempotentes) $x \wedge x = x = x \vee x$.
(ii) (conmutativas) $x \wedge y = y \wedge x, x \vee y = y \vee x$.
(iii) (asociativas) $x \wedge (y \wedge z) = (x \wedge y) \wedge z, x \vee (y \vee z) = (x \vee y) \vee z$.
(iv) (absorción) $x \wedge (x \vee y) = x = x \vee (x \wedge y)$.
(v) (del cero) $x \wedge 0 = 0, x \vee 0 = x$.
(vi) (del uno) $x \wedge 1 = x, x \vee 1 = 1$.
(vii) (distributiva $\wedge$) $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$.
(viii) (distributiva $\vee$) $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$.
(ix) (de la negación) $x \wedge \neg x = 0, x \vee \neg x = 1$.

Dadas dos álgebras de Boole B y B', un morfismo de álgebras de Boole es un morfismo de $(0,0,1,2,2)$-*álgebras entre ellas, es decir, una aplicación $f : B \to B'$ que conserva las operaciones similares:*
(o) $f(0) = 0$, $f(1) = 1$,
(i) $f(\neg x) = \neg f(x)$,
(ii) $f(x \wedge y) = f(x) \wedge f(y)$, $f(x \vee y) = f(x) \vee f(y)$.

Si nos fijamos en las álgebras de Boole y los morfismos entre ellas vemos que la composición (cuando se puede, siendo entonces asociativa) de dos morfismos es de nuevo un morfismo y las identidades son morfismos, de modo que podemos hablar también de una categoría *ALGB* de las álgebras de Boole. He aquí una pequeña lista de los ejemplos más usuales de álgebras de Boole, además de las $\mathcal{B}(\mathcal{A})$ ya conocidas:

1. El ejemplo más simple de álgebra de Boole (con $0 \neq 1$) es el conjunto $2 = \{0, 1\}$ de los valores de verdad, con las operaciones bien conocidas.

2. Dado un conjunto X, el conjunto 2^X de las aplicaciones de X en 2 es un álgebra de Boole con las operaciones inducidas punto a punto por las de 2. En general, si B es un álgebra de Boole entones B^X también lo es.

3. Dado un conjunto X, el conjunto $\mathbf{P}(X)$ de sus partes o conjuntos es un álgebra de Boole con las operaciones $\emptyset, X, (-)^c, \cap, \cup$.

4. El algebra $\mathcal{B}(\mathcal{A})$ es una algebra de Boole como consecuencia del Teorema 2.2.2.

Si nos fijamos en los ejemplos 2 y 3 anteriores, podemos observar que la biyección $\mathbf{P}(X) \cong 2^X$ que asocia a cada subconjunto su función característica (y viceversa) es un morfismo de álgebras de Boole. Este tipo de morfismos que son biyectivos se llaman isomorfismos y las correspondientes álgebras de Boole se dicen isomorfas. El isomorfismo es como un traductor que pasa propiedades de un álgebra a propiedades de la otra, siempre que sean propiedades que se expresen mediante las operaciones (polinomios).

Sabemos que si el conjunto X tiene n elementos entonces $\mathbf{P}(X)$ tiene 2^n elementos. También es fácil determinar el número de elementos de $\mathcal{B}(\mathcal{A})$ cuando A tiene n átomos. Basta recordar que en tal caso un elemento de $\mathcal{B}(\mathcal{A})$ viene caracterizado por una función $2^n \to 2$, y el número de estas funciones es $2^{(2^n)}$.

Como álgebras de Boole isomorfas han de tener el mismo cardinal, esto prueba que hay álgebras de la forma $\mathbf{P}(X)$ que no son de la forma $\mathcal{B}(\mathcal{A})$.

Este curso no es el lugar apropiado para desarrollar la teoría de las álgebras de Boole, pero sí que podemos dejar constancia del procedimiento algebraico propio de esta teoría demostrando alguna propiedad general que es bien conocida en el caso de las álgebras de proposiciones.

Lema 2.3.2 *Sea B un álgebra de Boole. Si $x \in B$ y existe $\dot{x} \in B$ tal que $x \vee \dot{x} = 1$ y $x \wedge \dot{x} = 0$, entonces $\dot{x} = \neg x$,*

Demostración: Supongamos que existe $\dot{x}$, cumpliendo $x \vee \dot{x} = 1$ y $x \wedge \dot{x} = 0$. Entonces,
$\neg x = \neg x \wedge (x \vee \dot{x}) = (\neg x \wedge x) \vee (\neg x \wedge \dot{x}) = \neg x \wedge \dot{x}$.
Análogamente, intercambiando los papeles de las operaciones, se prueba que
$\dot{x} = \dot{x} \wedge (x \vee \neg x) = (\dot{x} \wedge x) \vee (\dot{x} \wedge \neg x) = \dot{x} \wedge \neg x = \neg x \wedge \dot{x}$.
Luego $\dot{x} = \neg x$. □

Corolario 2.3.3 *En un álgebra de Boole hay una única operación 1-aria que cumpla las propiedades de la Definición 2.3.1.*

Si traducimos el Lema 2.3.2 al lenguaje semántico propio de $\mathscr{B}(\mathscr{A})$ obtenemos un enunciado que escribimos en el formato de las reglas de inferencia:

Si $P \wedge Q$ es una contradicción y $P \vee Q$ es una tautología, entonces $Q \equiv \neg P$

Este enunciado es cierto porque ha sido demostrado para toda álgebra de Boole, y en particular para el álgebra de Boole $\mathscr{B}(\mathscr{A})$.

Teorema 2.3.4 *En un álgebra de Boole se verifican la ley de la doble negación $\neg\neg x = x$ y las leyes de De Morgan*

$$\neg(x \wedge y) = \neg x \vee \neg y, \quad \neg(x \vee y) = \neg x \wedge \neg y.$$

Demostración: Por el Lema 2.3.3 las condiciones $\neg x \vee x = 1$, $\neg x \wedge x = 0$ significan que $\neg\neg x = x$.

Notemos que
$(\neg x \vee \neg y) \wedge (x \wedge y) = (\neg x \wedge x \wedge y) \vee (\neg y \wedge x \wedge y) = 0$
y también que
$(\neg x \vee \neg y) \vee (x \wedge y) = 1.$

Por tanto, aplicando el Lema 2.3.3 resulta la primera ley de De Morgan. La segunda se demuestra de modo análogo (también a partir de la primera). □

Una relación de orden como la que hay en $\mathscr{B}(\mathscr{A})$ se puede introducir en cualquier álgebra de Boole siguiendo la pauta de la Proposición 2.2.4.

Lema 2.3.5 *En toda álgebra de Boole B se verifica: $x = x \wedge y$ si y sólo si $x \vee y = y$.*

Demostración: Supongamos que se verifica $x = x \wedge y$. Entonces, $x \vee y = (x \wedge y) \vee y$. Teniendo en cuenta el axioma de absorción se tiene que $(x \wedge y) \vee y = y$. Por lo tanto $x \vee y = y$.

Supongamos ahora que se verifica $x \vee y = y$. Entonces, $x \wedge y = x \wedge (x \vee y)$. Teniendo en cuenta el otro axioma de absorción se tiene que $(x \wedge (x \vee y) = x$. Por lo tanto $x \wedge y = x$. □

Definición 2.3.6 *Sea B una álgebra de Boole, Dados $x, y \in B$ se dice que $x \leq y$ si se verifica $x = x \wedge y$, o, equivalentemente $x \vee y = y$.*

Proposición 2.3.7 *En toda álgebra de Boole B se verifica:*
(i) $x \leq y$ es una relación de orden.
(ii) $x \leq y$ si y sólo si $\neg y \leq \neg x$.
(iii) Si $x \leq y$ entonces $x \wedge z \leq y \wedge z$, y también $x \vee z \leq y \vee z$.

Demostración:

(i) : Es reflexiva, ya que del axioma de idempotencia $x \wedge x = x$, se sigue que $x \leq x$. Es antisimétrica: si suponemos que $x \leq y$ e $y \leq x$, se tiene que $x = x \wedge y$ e $y = y \wedge x$. Puesto que $x \wedge x = y \wedge x$, se tiene que $x = y$. Es transitiva: si suponemos que $x \leq y$ e $y \leq z$, se tiene que $x = x \wedge y$ e $y = y \wedge z$. Entonces $x = x \wedge y = x \wedge (y \wedge z) = (x \wedge y) \wedge z = x \wedge z$, Por lo tanto $x \leq z$.

(ii): $x \leq y$ si y sólo si $x = x \wedge y$ si y sólo si $\neg x = \neg x \vee \neg y$ si y sólo si $\neg y \leq \neg x$.

(iii): Si $x \leq y$ entonces $x = x \wedge y$ e $y = x \vee y$. Por lo tanto, $x \wedge z = (x \wedge y) \wedge z) = (x \wedge z) \wedge (y \wedge z)$ y también $y \vee z = (x \vee y) \vee z = x \vee (y \vee z) = (x \vee z) \vee (y \vee z$. Así que, $x \wedge z \leq y \wedge z$, y también $x \vee z \leq y \vee z$. □

Es también muy sencillo introducir nuevas operaciones en un álgebra de Boole arbitraria. Por ejemplo, podemos definir la operación binaria «condicional»cuyo sentido en los casos $\mathscr{B}(\mathscr{A})$ y 2 es obvio. La Definición y la propiedad fundamental de esta operación aparecen en el teorema siguiente.

Teorema 2.3.8 *Sea B un álgebra de Booole cualquiera, $x \to y = \neg x \vee y$ es la única operación binaria que verifica: $x \leq y \to z$ si y sólo si $x \wedge y \to z$.*

Demostración: Si $x \leq y \to z = \neg y \vee z$, entonces $x \wedge y \leq (\neg y \vee z) \wedge y = z \wedge y \leq z$. El recíproco se prueba de modo similar. Queda demostrar la unicidad de la operación. Supongamos otra operación binaria $\succ$ que cumpla: $x \leq y \succ z$ si y sólo si $x \wedge y \succ z$. Entonces se verifica $x \to y \leq x \succ y$ porque equivale a $x \to y \wedge x \leq y$, lo que es cierto porque a su vez equivale a $x \to y \leq x \to y$. Como ambas operaciones tienen las mismas propiedades, por la misma razón se verifica $x \succ y \leq x \to y$, luego $x \to y = x \succ y$. $\square$

En la demostración del Teorema 2.3.8 se ha procedido según es costumbre en el trabajo matemático, exponiendo rápidamente los pasos esenciales de la prueba dejando detalles intermedios a la confianza entre el autor y el lector. Esta relajación lógica funciona bien casi siempre, haciendo del trabajo demostrativo algo ágil, pero cuando surgen dificultades (una contradicción o simplemente un lector desconfiado) las demostraciones se comprueban con todo el detalle que exige un método lógico preciso, y su escritura resulta inevitablemente mucho más larga. Así lo haremos cuando estudiemos formalmente los métodos de demostración.

Terminaremos enunciando sin demostración una propiedad universal del álgebra de Boole de las proposiciones que la distingue (salvo isomorfismo) entre todas las álgebras de Boole abstractas.

Extendemos la inclusión $\mathscr{A} \to P$ a una inclusión $\mathscr{A} \to \mathscr{B}(\mathscr{A})$ que hace corresponder a cada átomo p su clase de equivalencia lógica $[p]$.

Teorema 2.3.9 *Sea B un álgebra de Booole cualquiera. Para cada aplicación $v\colon \mathscr{A} \to B$ existe un único morfismo de álgebras de Boole $\bar{v}\colon \mathscr{B}(\mathscr{A}) \to B$ tal que para cada $p \in \mathscr{A}$, $\bar{v}([p]) = v(p)$ ($\bar{v}$ es una extensión de v).*

Demostrar el Teorema 2.3.6 no es fácil y queda fuera de un curso del nivel de éste, pero sí estamos en condiciones de demostrar la propiedad enunciada para un tipo muy particular de álgebra de Boole:

$$\begin{tikzcd}
\mathscr{A} \arrow[r] \arrow[d, "v"] & \mathscr{B}(\mathscr{A}) \arrow[dl, "\bar{v}"] \\
2 &
\end{tikzcd}$$

En este caso el proceso es bien simple: valoramos los átomos, extendemos la valoración a las proposiciones y esta extensión resulta invariante por equivalencia lógica, así que se extiende la valoración a las clases de equivalencia. Reducir el caso general a este caso particular es la parte difícil de la demostración del teorema, que expresa que $\mathscr{B}(\mathscr{A})$ es el álgebra de Boole libre sobre el conjunto $\mathscr{A}$.

Ejercicios

1. Demostrar que en un álgebra de Boole son equivalentes los enunciados siguientes: (i) $x \leq y$, (ii) $x \wedge \neg y = 0$, (iii) $x \to y = 1$. También que $x = y$ si y sólo si $x \leftrightarrow y = 1$, siendo $x \leftrightarrow y = (x \to y) \wedge (y \to x)$. Por último, demostrar que $((x \wedge y) \to z) \to (x \to (y \to z)) = 1$.

2. En un álgebra de Boole abstracta, demostrar las igualdades correspondientes a los axiomas de Lukasiewicz:

(i) $x \to (y \to x) = 1$,

(ii) $(x \to (y \to z)) \to ((x \to y) \to (x \to z)) = 1$,

(iii) $(\neg x \to \neg y) \to (y \to x) = 1$.

3. En un álgebra de Boole cualquiera, demostrar la forma algebraica de (SH): $(x \to y) \wedge (y \to z) \leq x \to z$. Usar cada uno de los siguientes métodos:

(i) comparando formas normales conjuntivas,

(ii) lo mismo con las disyuntivas,

(iii) aplicando la propiedad característica de la implicación y probando la forma algebraica de (MP): $x \wedge (x \to y) \leq y$.

4. En un álgebra de Boole, demostrar que $x \wedge (x \to y) = x \wedge y$.

Capítulo 3

Métodos de demostración

Los razonamientos habituales consisten en deducir una consecuencia de varias premisas, entendiendo que esto significa que todo modelo de las premisas es un modelo de la conclusión. Vamos a estudiar este esquema primero desde el punto de vista semántico y luego con método deductivo.

3.1. Reglas de inferencia

Comenzaremos con un conjunto finito de premisas, aunque luego veremos que esta restricción, usual en la práctica, se puede suprimir a veces.

Definición 3.1.1 *Un esquema de inferencia es una proposición de la forma*

$$P_1 \wedge \cdots \wedge P_n \to Q.$$

Las proposiciones $P_1, \cdots P_n$ son la premisas o hipótesis y Q es la conclusión o tesis del esquema de inferencia. Una regla de inferencia es un esquema de inferencia tautológico:

$$\models P_1 \wedge \cdots \wedge P_n \to Q.$$

Según el apartado (i) del Teorema1.2.4, una regla de inferencia se puede escribir en la forma

$$P_1 \wedge \cdots \wedge P_n \models Q.$$

indicando que las premisas (su conjunción) modela a la conclusión. La tradición ha impuesto el siguiente tipo de notaciones para los esquemas de inferencia, que serán reglas si son válidos o tautológicos, en el sentido de que la conclusión sea verdadera siempre que las premisas sean todas verdaderas.

$$\frac{P_1 \wedge \cdots \wedge P_n}{Q} \qquad \frac{P_1, \cdots, P_n}{Q} \qquad \begin{array}{c} P_1 \\ \vdots \\ P_n \\ \hline Q \end{array}$$

A continuación damos una lista de las reglas de inferencia más usuales, que tienen nombre propio. Empezaremos con las que sólo usan los conectores primitivos $\{\neg, \to\}$ y seguiremos con otras en las que todos los conectores entran en juego. En cada caso, el lector debe verificar que los esquemas inferenciales propuestos son, en efecto, reglas.

Definición 3.1.2 *Las siguientes reglas de inferencia reciben los nombres y abreviaturas que se indican.*

Reglas de modus ponens, modus tolens y silogismo hipotético:

$$\frac{\begin{array}{c}P \to Q\\ P\end{array}}{Q}\ (MP, \to E) \qquad \frac{\begin{array}{c}P \to Q\\ \neg Q\end{array}}{\neg P}\ (MT) \qquad \frac{\begin{array}{c}P \to Q\\ Q \to R\end{array}}{P \to R}\ (SH)$$

Reglas de introducción y eliminación de la doble negación:

$$\frac{P}{\neg\neg P}\ (\neg\neg I) \qquad \frac{\neg\neg P}{P}\ (\neg\neg E)$$

Reglas de introducción y eliminación de la conjunción:

$$\frac{\begin{array}{c}P\\ Q\end{array}}{P \wedge Q}\ (\wedge I) \qquad \frac{P \wedge Q}{P}\ (\wedge E)$$

Reglas de introducción y eliminación de la disyunción:

$$\frac{P}{P \vee Q}(\vee I) \qquad \frac{\begin{array}{c}P \vee Q\\ \neg P\end{array}}{Q}\ (\vee E)$$

Reglas de reducción al absurdo:

$$\frac{P \to (Q \wedge \neg Q)}{\neg P}\ (RA)$$
$$\frac{\neg P \to (Q \wedge \neg Q)}{P}\ (RA)$$

Reglas de casos *y de* resolución*:*

$$\frac{\begin{array}{c}P \vee Q\\ P \to R\\ Q \to R\end{array}}{R}\ (RC) \quad \frac{\begin{array}{c}P \vee Q\\ P \to R\\ Q \to S\end{array}}{R \vee S}\ (RC) \qquad \frac{\begin{array}{c}P \vee R\\ \neg P \vee S\end{array}}{R \vee S}\ (RR)$$

Hay dos procedimientos muy simples para obtener nuevas reglas de inferencia, que no son sino versiones de propiedades ya conocidas (Teorema 1.2.4):

Si $P \equiv Q$, entonces $\dfrac{P}{Q}$, $\dfrac{Q}{P}$ son reglas de inferencia.

Si $\dfrac{P}{Q}$, $\dfrac{Q}{R}$ son reglas de inferencia, entonces $\dfrac{P}{R}$ es una regla de inferencia.

Usando estos recursos el número de reglas se va multiplicando indefinidamente. Daremos unos pocos ejemplos más de reglas de inferencia con diversos conectores:

$$\frac{P}{Q \to P} \qquad \frac{P}{\neg P \to Q} \qquad \frac{\begin{array}{c}P\\ \neg P\end{array}}{Q} \qquad \frac{\begin{array}{c}P \to Q\\ P \to R\end{array}}{P \to Q \wedge R}$$

Una forma habitual de expresar los esquemas o reglas de inferencia es suponer un conjunto $\Gamma = \{P_1, \cdots ; P_n\}$ de proposiciones y usar la notación $\dfrac{\Gamma}{Q}$ y, si son reglas, también en la forma $\Gamma \models Q$

En el plano teórico, una ventaja de esta notación (conjunto de premisas) es que permite la consideración de conjuntos Γ ya sean finitos o infinitos, aunque en el caso infinito no podamos hablar de la proposición conjunción de todas las premisas, pues el lenguaje sólo permite la operación finita. Dado Γ, llamaremos modelo de Γ a cada interpretación v que sea modelo de todas las proposiciones de Γ, es decir, tal que $v(P) = 1$ para cada $P \in \Gamma$. La siguiente Definición es una repetición de la Definición 3.1.1, pero hecha de modo que admite que el conjunto de premisas sea infinito.

Definición 3.1.3 *Un esquema de inferencia es un par ordenado (Γ, Q) formado por un conjunto Γ de proposiciones y una proposición Q. Las proposiciones de Γ son la premisas o hipótesis y Q es la conclusión o tesis del esquema de inferencia. Una regla de inferencia es un esquema de inferencia tal que cada modelo del conjunto de premisas es un modelo de la conclusión.*

Vamos a dar algunos resultados útiles sobre reglas de inferencia utilizando un conjunto arbitrario de premisas, ya sea finito o infinito. Empezaremos con uno que tendrá un papel importante más adelante. Se trata de una equivalencia cuya parte «sólo si». (la parte «si »es más fácil) recibe el nombre de teorema de deducción.

Teorema 3.1.4 *Sea Γ un conjunto de proposiciones. Se verifica: $\Gamma \cup \{P\} \models Q$ si y sólo si $\Gamma \models P \to Q$.*

Demostración: Parte «si». Tenemos que probar que $\Gamma \cup \{P\} \models Q$, es decir, suponiendo que $v(R) = 1$ para cada $R \in \Gamma$ y también $v(P) = 1$, tenemos que probar que $v(Q) = 1$. Por la hipótesis $\Gamma \models P \to Q$, será $v(P \to Q) = 1$, pero como $v(P) = 1$ será también $v(Q) = 1$. Parte «sólo si». Tenemos que probar que $\Gamma \models P \to Q$, es decir, suponiendo que $v(R) = 1$ para cada $R \in \Gamma$, tenemos que probar que $v(P \to Q) = 1$. Por la hipótesis $\Gamma \cup \{P\} \models Q$, si $v(P) = 1$ será $v(Q) = 1$, luego $v(P \to Q) = 1$, y si $v(P) = 0$ será también $v(P \to Q) = 1$ por la tabla del condicional; luego siempre se tiene $v(P \to Q) = 1$. □

Definición 3.1.5 *Un conjunto Γ de proposiciones se dice contradictorio si se verifica $\Gamma \models C$ siendo C una contradicción.*

Hagamos notar que la contradicción esencial es de la forma $P \wedge \neg P$, siendo todas las contradicciones equivalentes a ésta (lo son todas entre sí), de modo que el conjunto contradictorio básico es $\{P, \neg P\}$. La contradicción C de la Definición anterior se puede suponer siempre de esta forma. En el caso finito, $\Gamma = \{P_1, \cdots ; P_n\}$ es contradictorio si y sólo si $P_1 \wedge \cdots \wedge P_n$ es una contradicción. Vamos a ver que las reglas de inferencia están muy relacionadas con los conjuntos contradictorios y deduciremos una nueva expresión de la reducción al absurdo y su recíproco.

Teorema 3.1.6 *Dado un conjunto finito Γ de proposiciones, las afirmaciones siguientes son equivalentes:*
(i) $\Gamma \models Q$.
(ii) $\Gamma \cup \{\neg Q\}$ es contradictorio.

Demostración: Usaremos la equivalencia $Q \equiv \neg Q \to C$, donde C es una contradicción cualquiera, de modo que (i) equivale a $\Gamma \models \neg Q \to C$, que a su vez equivale a (ii) por el Teorema 3.1.4. □

En (i) del Teorema 3.1.6 la verdad de Q sigue de la verdad de las premisas, pero en (ii) no se sigue el camino directo, sino que se busca la contradicción añadiendo $\neg Q$ a las premisas. Se dice que en (ii) se procede por refutación o reducción al absurdo. La expresión habitual de la reducción al absurdo es como sigue:

Corolario 3.1.7 *Sea Γ un conjunto de proposiciones. Se verifica:*
(i) $\Gamma \cup \{P\}$ *es contradictorio si y sólo si* $\Gamma \models \neg P$.
(ii) $\Gamma \cup \{\neg P\}$ *es contradictorio si y sólo si* $\Gamma \models P$.

Ejercicios

1. Verificar la validez del silogismo hipotético condicionado:
$\{H \to (P \to Q), H \to (Q \to R)\} \models H \to (P \to R)$.

2. Sea P la proposición $p \wedge q \to \neg(r \vee p)$ y sea Q la proposición que se obtiene sustituyendo q por p en P. Determinar si alguna de las inferencias $P \models Q$, $Q \models P$ es válida.

3. Verificar que $\Gamma \models F$ siendo $\Gamma = \{P \to Q, \neg(P \vee R) \to S\}$ y F la proposición $(P \wedge Q \to R) \to (\neg S \to R)$.

4. Comprobar que el siguiente esquema de inferencia no es una regla, pero que sí lo es después de un cierto cambio en las premisas segunda y tercera: $\{\neg P \vee \neg Q, P \to R, Q \to S\} \models \neg R \vee \neg S$.

3.2. Método de resolución

Recapitulemos lo realizado hasta ahora. A partir de un alfabeto y de los conectores lógicos construimos las proposiciones, cuya verdad podemos analizar por el método semántico: la tabla de verdad nos permite saber si una proposición es una verdad lógica (tautología) y verificar si un esquema de inferencia con un número finito de premisas es una regla. La ventaja de la tabla es su naturaleza de algoritmo finito (verdad decidible). No obstante, el método de tablas de verdad no es satisfactorio desde el punto de vista de la eficiencia computacional, pues el número de operaciones a efectuar crece muy deprisa, como 2^n en función de n. El método de resolución que vamos a introducir proporciona un algoritmo más eficiente en términos computacionales. Con este método se averigua si un conjunto finito de proposiciones es contradictorio. Previamente, se ponen las proposiciones en forma clausal.

Nos centramos en el problema de validar un esquema de inferencia con un número finito de premisas, pero lo vamos a ver como el problema de la contradicción de un conjunto finito de proposiciones, pues, como vimos en el Teorema 2.3.8:

$$\{P_1, \cdots, P_n\} \models Q \text{ si y sólo si } \{P_1, \cdots P_n, \neg Q\} \text{ es contradictorio.}$$

Completar con el caso $n = 0$ sería añadir que Q es una tautología si y sólo si $\neg Q$ es una contradicción.

En definitiva, la pregunta básica de la lógica de proposiciones se puede formular como una refutación de este modo:

¿Es $\Gamma = \{P_1, \cdots, P_n\}$ contradictorio?

En algunos caso la respuesta es obvia. Un conjunto contradictorio básico tiene la forma

$$\{P, \neg P\}$$

Definición 3.2.1 *Diremos que un conjunto de proposiciones es nulo si contiene una proposición y su negación*

Es evidente que un conjunto nulo es contradictorio. Más generalmente, si un conjunto de proposiciones es contradictorio entonces cualquier otro conjunto que lo contenga también será contradictorio, pero el recíproco no es cierto, pues en un conjunto ampliado la contradicción puede involucrar la parte añadida.

Lema 3.2.2 *Sean Γ y Γ' dos cconjuntos de proposiciones. Entonces,*

(i) Si Γ es un conjunto nulo, entonces Γ es contradictorio.

(ii) Si Γ es un conjunto contradictorio y $\Gamma \subset \Gamma'$, entonces Γ' es contradictorio.

(iii) Si $\Gamma' = \{P_1, \cdots P_n, Q_1, \cdots, Q_m\}$ y $\Gamma = \{P_1, \cdots P_n, Q_1 \wedge \cdots \wedge Q_m\}$, entonces Γ es contradictorio si y sólo si Γ' es un conjunto contradictorio.

El método de resolución pretende contestar a la pregunta básica anterior y además hacerlo de modo algorítmico. Para ello se va ampliando el conjunto dado sin modificar su carácter contradictorio, hasta que se llega bien a un conjunto nulo (entonces el conjunto inicial es contradictorio), o bien a otra situación que garantice la no contradicción.

Puesto que toda proposición admite una equivalente en forma clausal, resolver la contradicción de Γ equivale a resolver el mismo problema con un conjunto de cláusulas obtenido así:

- Calcular la forma clausal $Q_i \equiv P_i$, una para cada $i = 1, \cdots, n$.
- Sea Δ el conjunto formado por las cláusulas de las Q_i.

Nótese que el número de clausulas en Δ así obtenidas será igual o mayor que el número de proposiciones que había en en el conjunto Γ inicial, pero este aparente inconveniente es compensado de sobras, como veremos, por la forma especial que tienen las cláusulas. Tampoco importa que el conjunto de cláusulas Δ obtenido a partir de Γ no sea único, cualquiera de los posibles sirve.

En este punto, la pregunta básica de la lógica de proposiciones que formulábamos antes se puede enunciar de este modo:

¿Es un conjunto de cláusulas $\Delta = \{C_1, \cdots, C_n\}$ contradictorio?

Vamos a proponer un algoritmo para contestar inequívocamente a esta pregunta. Supongamos que dos de las cláusulas tienen la forma

$$C_i = C_i' \vee p, C_j = C_j' \vee \neg p,$$

de modo que el átomo p no está en ni en la cláusula C_i' ni en la C_j' luego tampoco en la cláusula $C_i' \vee C_j'$

(disyunción de las cláusulas simplificando los literales repetidos). Obsérvese ahora este hecho clave:

Teorema 3.2.3 *Sea un conjunto de cláusulas $\Delta = \{C_1, \cdots, C_n\}$ y supongamos que dos cláusulas son de la forma $C_i = C_i' \vee p$, $\quad C_j = C_j' \vee \neg p$. Entonces Δ es contradictorio si y sólo si $\Delta' = \Delta \cup \{C_i' \vee C_j'\}$ es contradictorio.*

Demostración: Si Δ es contradictorio lo será Δ' por ser mayor. Recíprocamente, que Δ' es contradictorio significa que la conjunción de sus cláusulas es una contradicción, pero se verifica

$$C_i \wedge C_j \equiv C_i \wedge C_j \wedge (C_i' \vee C_j')$$

En efecto, no es dificil comprobar que $C_i \wedge C_j \models C_i' \vee C_j'$. Además se tiene que $C_i \wedge C_j \models C_i \wedge C_j$. Por lo tanto, $C_i \wedge C_j \models C_i \wedge C_j \wedge (C_i' \vee C_j')$. Por otro lado es claro que $C_i \wedge C_j \wedge (C_i' \vee C_j') \models C_i \wedge C_j$.

Así que la conjunción de las cláusulas de Δ' es equivalente a la conjunción de las cláusulas de Δ, luego Δ es contradictorio. □

En el teorema anterior se pasa de mediante una aplicación de la regla de resolución (RR) ya vista con proposiciones cualesquiera en la Definición, pero que ahora consideramos en un sentido algo diferente. Volvemos a escribirla como regla de resolución de cláusulas, mostrando también su forma mínima que expresa la contradicción en estado puro, representada por el valor $\perp$:

$$\begin{array}{c} p \vee C_i' \\ \underline{\neg p \vee C_j'} \,(\mathrm{RR}) \\ C_i' \vee C_j' \end{array} \qquad \begin{array}{c} p \vee C_i' \\ \underline{\quad \neg p \quad} \,(\mathrm{RR}) \\ C_i' \end{array} \qquad \begin{array}{c} p \\ \underline{\neg p} \,(\mathrm{RR}) \\ \perp \end{array}$$

Corolario 3.2.4 *Si después de un número finito de aplicaciones de (RR) el conjunto Δ se amplía hasta un conjunto nulo, entonces Δ es contradictorio.*

Ya estamos en condiciones de dar la idea básica del algoritmo buscado:

Algoritmo de resolución. Mediante aplicaciones sucesivas de la regla (RR) vamos haciendo cada vez más grande el conjunto de cláusulas cuya contradicción pretendemos averiguar (que equivale a la contradicción del conjunto inicial) pero las nuevas cláusulas que vamos generando tienen cada vez menos literales, de modo que, salvo que el proceso «se pare» antes, llegaremos a tener literales aislados y, entonces, o encontramos entre ellos un conjunto nulo $\{p, \neg p\}$ o no.

Como ejemplo, probaremos por resolución la validez del silogismo hipotético, que escribimos a continuación a la izquierda. En el centro escribiremos primero las dos premisas y la negación de la conclusión y luego las formas clausales de cada una de las anteriores; finalmente, en la columna de la derecha aparecen numeradas las cuatro cláusulas resultantes.

$$\begin{array}{ll} 1 & p \to q \\ 2 & q \to r \\ \hline 3 & p \to r \end{array} \qquad \Gamma \begin{array}{ll} 1 & \neg p \vee q \\ 2 & \neg q \vee r \\ 3 & p \wedge \neg r \end{array} \qquad \Delta \begin{array}{ll} 1 & \neg p \vee q \\ 2 & \neg q \vee r \\ 3 & p \\ 4 & \neg r \end{array}$$

A partir de aquí se empieza el algoritmo, que se expresa como sigue, indicando a la derecha a qué cláusulas se ha aplicado la regla de resolución (RR):

$$\begin{array}{ccc}
5 & \neg p \vee r & (1,2) \\
6 & q & (1,3) \\
7 & \neg q & (2,4) \\
8 & \perp & (6,7)
\end{array}$$

El proceso ha terminado aplicando (RR) en su forma mínima, que lleva al o final que nos indica que el conjunto de cláusulas es contradictorio, verificando así la validez del silogismo hipotético. Pero veamos dos ejemplos que no terminen igual. Si permutamos la conclusión con una premisa en el silogismo hipotético obtenemos esquemas de inferencia que no son válidos por evidentes razones semánticas, a los que el método de resolución da el siguiente tratamiento:

$$\begin{array}{cccccc}
 & & 1 & \neg p \vee q & \\
1 & p \to q & 2 & \neg p \vee r & \\
2 & p \to r & 3 & q & \\ \cline{1-2}
3 & q \to r & 4 & \neg r & \\
 & & 5 & \neg p & (2,4)
\end{array}
\qquad
\begin{array}{cccccc}
 & & 1 & \neg p \vee r & \\
1 & p \to r & 2 & \neg q \vee r & \\
2 & q \to r & 3 & p & \\ \cline{1-2}
3 & p \to q & 4 & \neg q & \\
 & & 5 & r & (1,3)
\end{array}$$

En ambos casos sólo se pudo aplicar (RR) una vez y el algoritmo «se para», lo que se puede expresar diciendo que llegamos a un conjunto de cláusulas estable, que no cambia por aplicaciones de (RR), sin que aparezca la contradicción.

Definición 3.2.5 *Sea Δ un conjunto de cláusulas. Diremos que Δ es estable si la resolvente de cualquier resolución es siempre un cláusula de Δ.*

Lo que hemos visto en estos ejemplos sencillos se realiza sin gran dificultad con conjuntos más grandes de cláusulas, y siempre se llega a las dos mismas terminaciones del algoritmo: a la contradicción $\perp$, o un conjunto estable sin llegar a $\perp$. Demostrar que esto siempre sucede así no es asunto para principiantes, pero al menos hay que saber que el método de resolución va bien porque se demuestra el teorema fundamental siguiente:

Teorema 3.2.6 *(Teorema de resolución) Se verifica:*

(i) Si un conjunto Γ es contradictorio, existe una secuencia finita de aplicaciones de (RR) que lleva a $\perp$.

(ii) Si un conjunto Γ no es contradictorio, existe una secuencia finita de aplicaciones de (RR) que amplía Γ hasta un conjunto estable sin llegar a$\perp$.

La ventaja de este método es que se puede programar para que un ordenador nos haga el trabajo cuando las cláusulas sean muchas y grandes. También se programan mejoras adicionales para que el número de veces que se aplique (RR) sea el menor posible.

El siguiente método, que puede ser largo, tiene la ventaja de que el modo de ejecutarlo es totalmente algorítmico. Dado un conjunto de cláusulas se puede proceder del siguiente modo:

- Eliminar las cláusulas que contengan un literal y su complementario.
- Simplificar las cláusulas que contengan un literal repetido.
- Elegir un átomo del alfabeto, y determinar las cláusulas que contengan este átomo o su complemetario.

▹ Calcular todas las cláusulas resolventes posibles a partir del las cláusulas que contengan el átomo elegido o su complementario (Tener en cuenta que si en un momento determinado se obtiene una resolvente que ya está en la lista no es necesario repetirla, no hay que incluir resolventes que contengan un literal y su complementario, y finalmente, las resolventes que tengan literales repetidos se deben simplificar).

▹ Aplicar el proceso anterior de modo ordenado para cada uno de los átomos restantes hasta que se obtenga un conjunto nulo o estable no nulo.

Ejercicios
1. Demostrar por resolución que son tautologías los axiomas de Lukasiewicz:
(L1) $P \to (Q \to P)$,
(L2) $(P \to (Q \to R)) \to ((P \to Q) \to (P \to R))$ y
(L3) $(\neg P \to \neg Q) \to (Q \to P)$.
2. Demostrar por resolución las reglas de deducción (MT), (SH), (¬¬I), (∨E), (RC) y (RR).
3. Estudiar por resolución la validez del esquema de inferencia:

$$\begin{array}{c} (P \wedge Q) \to (R \wedge S) \\ S \to (Q \wedge T) \\ S \wedge T \\ \hline P \to (Q \wedge R) \end{array}$$

3.3. Un sistema automatizado de demostración: Prover9

Existen numerosos sistemas que realizan automáticamente el proseso de resolución y permiten demostrar algunos teoremas. En este curso vamos a utilizar uno de ellos denominado Prover9. Éste es adecuado para realizar procesos de resolución para proposiciones y para predicados de primer order. Lo desarrolló William Walker McCune (17 de diciembre de 1953 - 2 de mayo de 2011) que fue un informático y lógico estadounidense que trabajó en los campos de razonamiento automatizado, álgebra, lógica y métodos formales. Actualmente existen otros muchos demostradores automatizados de teoremas que pueden conseguir resultados más esperanzadores.

Hemos elegido prover9 porque tiene una sintaxis sencilla, admite una interface fácil de usar y se adapta bien a los contenidos de este curso.

Para introducir las proposiciones se utiliza la siguiente sintaxis:

Nombre	En apuntes	Para Prover9	Ejemplo
negación	¬	-	(-p)
disyunción	∨	|	(p | q | r)
conjunción	∧	&	(p & q & r)
implicación	→	->	(p -> q)
otra implicación	←	<-	(p <- q)
bicondicional	↔	<->	(p <- >q)

Hay que notar que para trabajar con Prover9 las proposiciones que se introducen como premisas o como consecuencia tienen que acabar en un punto. El sistema admite también una sintaxis relajada de proposiciones que permite quitar algunos paréntesis.

Un ejemplo sencillo:
Si introducimos en la ventana Assumptions:

```
(p->q).
(q->r).
```

y en la ventana Goals:

```
(p->r).
```

Se obtiene la siguiente resolución:

```
============================== PROOF =================================

% -------- Comments from original proof --------
% Proof 1 at 0.00 (+ 0.00) seconds.
% Length of proof is 10.
% Level of proof is 3.
% Maximum clause weight is 2.
% Given clauses 4.

1 p -> q # label(non_clause).  [assumption].
2 q -> r # label(non_clause).  [assumption].
3 p -> r # label(non_clause) # label(goal).  [goal].
4 -p | q.  [clausify(1)].
5 -q | r.  [clausify(2)].
6 p.  [deny(3)].
7 -r.  [deny(3)].
8 q.  [ur(4,a,6,a)].
9 -q.  [resolve(7,a,5,b)].
10 $F.  [resolve(9,a,8,a)].

============================== end of proof ==========================
```

Prover9 genera una resolución en la que en primer lugar aparecen las proposiciones que son premisas. A continuación aparece la proposición consecuencia que es el objetivo que queremos probar. Después las cláusulas de las premisas y de la negación de la conclusión y luego hay una resolución que lleva a $F.

Si intentamos verificar un esquema que no es válido, Prover9 responde con el mensaje: Prover7 exit: exhausted. Esta respuesta puede tener dos causas: que se haya superado el tiempo que se haya prefijado a Prover9 para efectuar la resolución o que el esquema no sea una regla. En el tipo de reglas que manejamos en el curso que tienen pocas letras y proposiciones, normalmente lo que sucede es que el esquema no es una regla.

Para probar que no es una regla por el método de resolución debemos hacer todas las resoluciones posibles hasta que el conjunto sea estable. Un conjunto de cláusulas es estable si los resultados de todas las posibles nuevas resoluciones que podamos efectuar ya están en dicho conjunto. Otra posibilidad de ver que no es una regla es la de buscar un modelo de las premisas que no sea modelo de la consecuencia. La aplicación Prover9 dispone de la herramienta MACE (Models And CounterExamples) que busca uno de estos modelos que nos sirve de contraejemplo.

Consideremos el siguiente caso. Introduzcamos en en la ventana Assumptions:

```
(p->q).
(p->r).
```

y en la ventana Goals:

```
(q->r).
```

Ya hemos visto anteriormente no es una regla. Podemos buscar un contraejemplo pulsando el botón star de Mace para obtener:

```
 interpretation( 2, [number = 1,seconds = 0], [
    relation(p, [0]),
    relation(q, [1]),
    relation(r, [0])]).
```

que significa que $(p, q, r) = (0, 1, 0)$ es un modelo de la premisas que hace falsa la conclusión.

En este curso, el alumno debe realizar el mismo algunas resoluciones para comprender mejor este método de demostración, adquirir un poco de habilidad y buscar estrategias para completar la resolución lo antes posible. No obstante, si no encuentra una resolución o contrajemplo, puede utilizar Prover9 como ayuda para ver si el esquema que está analizando es o no una regla, y además disponer de una solución.

3.4. Deducción natural (método Gentzen)

Recapitulemos lo realizado hasta ahora. A partir de un alfabeto y de los conectores lógicos construimos las proposiciones, cuya verdad podemos analizar por dos métodos:

1. Semántica. La tabla de verdad nos permite saber si una proposición es una verdad lógica (tautología) y verificar si un esquema de inferencia con un número finito de premisas es una regla.

2. Resolución. Con una transformación previa de las proposiciones, un método algorítmico nos permite saber si las premisas y la negación de la conclusión forman un conjunto contradictorio.

En esta sección ofrecemos un nuevo punto de vista para desarrollar la lógica de proposiciones. En vez de estudiar la la relación $\Gamma \models Q$ recurriendo a los modelos o tablas de verdad (modo semántico de trabajar), queremos hacerlo mediante una «demostración o deducción» (modo deductivo de trabajar), es decir, «demostrando que Q es una consecuencia de Γ » o «deduciendo Q a partir de Γ », hecho que indicaremos con la notación

$$\Gamma \vdash Q$$

y $P \vdash Q$ en el caso de una premisa. Esta notación es parecida pero distinta a la anterior. Recordemos juntas ambas para mejor apreciar sus semejanzas y sus diferencias simbólicas y conceptuales:

- $\Gamma \models Q$: todo modelo de Γ es un modelo de Q.
- $\Gamma \vdash Q$: hay una demostración de Q a partir de Γ .

Todos los escolares tenemos una cierta idea de lo que es una demostración: una cadena finita de pasos deductivos justificados que llevan desde unas hipótesis o premisas hasta una tesis o conclusión. Esta idea, que surge de la práctica de nuestro modo innato y educado de razonar, tenemos que incorporarla al sistema formal, para que tenga un significado preciso y podamos hacer un uso riguroso de ella. También el método de resolución es deductivo, pero aplicado a un tipo muy particular de proposiciones (cláusulas) al que hay que reducir previamente las dadas. En la deducción natural las proposiciones se utilizan tal como son dadas, sin ponerlas en una forma particular estandarizada.

Pero la deducción tiene un inconveniente, nos asegura la validez una vez construida la prueba, pero si no la podemos obtener nada sabemos, es decir, da respuesta positiva a la pregunta sobre validez pero no da respuesta también negativa, como hacen los métodos semántico y de resolución. No obstante, conocer los métodos de prueba y adquirir destreza con ellos es una tarea formativa importante.

Como toda deducción necesita hipótesis, para empezar a deducir tenemos que fijar alguna «deducción primitiva» o «modo de razonar básico» que nos sirva de punto de partida.

Un ejemplo nos aclarará la situación en que nos encontramos. Consideremos $\Gamma = \{P, P \to Q, Q \to R\}$ e indaguemos sobre el significado de $\Gamma \models R$ y de $\Gamma \vdash R$:

Prueba de $\Gamma \models R$: Suponiendo $v(R) = 0$ tenemos que ver que v vale 0 en alguna de las premisas. Veámoslo. Si $v(P) = 0$ la prueba ha terminado, pero si $v(P) = 1$ hay que seguir, así que miramos el valor de $P \to Q$; si $v(P \to Q) = 0$ se acabó la prueba, pero si vale 1 hay que seguir. Ahora bien, $v(P \to Q) = 1$ con $v(P) = 1$ asegura $v(Q) = 1$, luego (recordando que $v(R) = 0$) resulta $v(Q \to R) = 0$ y la prueba concluye.

Prueba de $\Gamma \vdash Q$: De P y $P \to Q$ deduzco Q por modus ponens, y luego de Q y $Q \to R$ deduzco R otra vez por modus ponens. Se acabó la prueba.

El modo de prueba $\Gamma \models Q$ es el que hemos propuesto para demostrar cada una de las reglas de inferencia vistas en la la subsección anterior. El modo de prueba $\Gamma \vdash Q$ es el que queremos aprender ahora. El esquema inferencial

$$\begin{array}{c} P \\ P \to Q \\ Q \to R \\ \hline R \end{array}$$

se ha convertido en una regla de inferencia después de la prueba en modo $\models$, pero diremos que se ha convertido en una regla de deducción después de la prueba en modo $\vdash$. Este cambio de notación es conveniente porque sirve para recordarnos el contexto de nuestro trabajo. La deducción que acabamos de hacer del esquema anterior puede escribirse como una cadena finita que empieza en las premisas y acaba en la conclusión:

1	P	Premisa
2	$P \to Q$	Premisa
3	$Q \to R$	Premisa
4	Q	MP(1,2)
5	R	MP(4,3)

Las proposiciones 4 y 5 se han justificado mediante modus ponens, que hace el papel de «deducción primitiva», es decir, hemos supuesto válida una deducción $\{P, P \to Q\} \vdash Q$ y la hemos usado para hace otras deducciones más complejas. Si queremos iniciar un proceso deductivo que nos permita un amplio abanico de posiblidades, parece que tendremos que dar alguna regla de deducción primitiva más. Así lo haremos, eligiéndolas entre las reglas de inferencia ya conocidas.

Para construir un sistema deductivo tenemos que empezar la lógica desde el principio, pero ahora supondremos que son primitivos los cuatro conectores básicos $\{\neg, \wedge, \vee, \to\}$ y suponemos que el lenguaje ha sido construido con ellos por el procedimiento usado con el sistema $\{\neg, \to\}$ y que la semántica se ha definido mediante las tablas de verdad usuales de dichos conectores.

Definición 3.4.1 *Llamaremos reglas primitivas de la deducción natural del método Gentzen (1934) a las siguientes:*

$$\frac{\begin{array}{c} P \to Q \\ P \end{array}}{Q}(MP) \qquad \frac{\neg\neg P}{P}(\neg\neg E)$$

$$\frac{\begin{array}{c} P \\ Q \end{array}}{P \wedge Q}(\wedge I) \qquad \frac{P \wedge Q}{P}(\wedge E) \qquad \frac{P \wedge Q}{Q}(\wedge E)$$

$$\frac{P}{P \vee Q}(\vee I) \qquad \frac{Q}{P \vee Q}(\vee I)$$

Nótese que todas ellas son reglas de inferencia. Quizás convenga hacer una observación sobre la segunda fila de reglas en la Definición anterior, donde hay dos (∧E) pero sólo una (∧I). En realidad, también se tiene la segunda regla de la introducción de la conjunción, cuya conclusión sería $Q \wedge P$ pero es que no se trata de una segunda regla sino de la misma intercambiando los papeles de P y Q, lo que puede hacerse porque en las premisas aparecen ambas proposiciones formando un conjunto de dos elementos, por tanto perfectamente intercambiables entre sí. En cambio, en la eliminación de la conjunción aparece la premisa $P \wedge Q$ en la que no hay conmutatividad sintáctica, por lo que suponer deducible la primera componente no es lo mismo que hacer lo propio con la segunda. La conmutatividad de la conjunción sí que se verifica, como era de suponer, desde el punto de vista deductivo, pues podemos componer la siguiente deducción:

$$\frac{P \wedge Q}{Q \wedge P} \qquad \begin{array}{clc} 1 & P \wedge Q & \text{Premisa} \\ 2 & P & \wedge\text{E (1)} \\ 3 & Q & \wedge\text{E (1)} \\ 4 & Q \wedge P & \wedge\text{I (3,2)} \end{array}$$

Parecida es la situación en la tercera fila, en la que aparece la introducción de la disyunción en ambas componentes, pero el lector tendrá dificultades para probar la conmutatividad de la disyunción usando sólo las reglas primitivas anteriores (mejor que no lo intente).

Muchas reglas de inferencia no podríamos convertirlas en reglas de deducción (es decir, deducir su conclusión a partir de sus premisas) si no pusiéramos en juego más recursos deductivos. El sistema deductivo conocido como deducción natural, que fue obra de Gentzen (1934), queda recogido en la Definición que viene a continuación.

Definición 3.4.2 *La deducción natural de una conclusión Q a partir de un conjunto de premisas Γ por el método Gentzen consiste en una sucesión finita $Q_1, \cdots, Q_n = Q$ de proposiciones tal que para cada $i = 1, \cdots, n$, se verifica alguna de las siguientes posibilidades:*

(i) Q_i pertenece a Γ .

(ii) Q_i se deduce de Γ y de alguna o algunas de las Q_j, $j < i$ mediante una regla primitiva de deducción natural.

(iii) Q_i se deduce de Γ y de alguna o algunas de las Q_j, $j < i$, mediante uno de los siguiente procedimientos primitivos de deducción natural:

- ▹ *Teorema de deducción. Dado un conjunto Δ de proposiciones, se tiene la deducción $\Delta \vdash P \to Q$ si se hace la deducción $\Delta \cup \{P\} \vdash Q$. En esquema:*

$$\frac{\Delta \cup \{P\} \ \vdash Q}{\Delta \vdash P \to Q} \quad \textit{(TD)}$$

- ▹ *Reducción al absurdo. Dado un conjunto Δ de proposiciones, se tiene una deducción $\Delta \vdash \neg P$ si se hace una deducción $\Delta \cup \{P\} \vdash Q \wedge \neg Q$. En esquema:*

$$\frac{\Delta \cup \{P\} \vdash Q \wedge \neg Q}{\Delta \vdash \neg P} \quad \textit{(RA, }\neg\textit{I)}$$

- ▹ *Prueba por casos. Se tiene la deducción deducción $\Delta \vdash R$ si se tiene $P \vee Q$ y se hacen deducciones $\Delta \cup \{P\} \vdash R$ y $\Delta \cup \{Q\} \vdash R$*

$$\frac{\begin{array}{c} P \vee Q \\ \Delta \cup \{P\} \vdash R \\ \Delta \cup \{Q\} \vdash R \end{array}}{R} \quad \textit{(RA, }\vee\textit{E)}$$

Se llama teorema a toda proposición deducida sin premisas ($\Gamma = \emptyset$) por el procedimiento de deducción natural anterior. Se escribe $\vdash P$ para indicar que P es un teorema.

En los tres procedimientos primitivos anteriores se amplia las premisas existentes mediante nuevas premisas que normalmente llamamos supuestos, aunque luego en la deducción obtenida ese supuesto provisional se elimina de nuevo. En este caso diremos que el supuesto se ha cancelado. El conjunto Δ está puede estar formado bien por proposiciones de Γ, bien por proposiciones ya deducidas o bien por supuestos previos que aún no se han cancelado.

Como puede parecer sorprendente, a primera vista, que un sistema deductivo pensado en términos de reglas permita también deducir teoremas, daremos de inmediato la deducción de $P \vee \neg P$, que puede servir como punto de partida para deducciones que utilicen la prueba por casos.

Teorema 3.4.3 *Cada proposición de la forma $P \vee \neg P$ es un teorema de la deducción natural.*

Demostración:

1.	$\neg(P \vee \neg P)$		Supuesto (RA)
2.		P	Supuesto (RA)
3.		$P \vee \neg P$	$\vee$I(2)
4.		$(P \vee \neg P) \wedge \neg(P \vee \neg P)$	$\wedge$I(1,3)
5.	$\neg P$		$\neg$I(2-4)
6.	$P \vee \neg P$		$\vee$I(5)
7.	$(P \vee \neg P) \wedge \neg(P \vee \neg P)$		$\wedge$I(1,6)
8. $\neg\neg(P \vee \neg P)$			$\neg$I(1-7)
9. $P \vee \neg P$			$\neg\neg$E(8)

□

El formato de la deducción anterior necesita alguna explicación. Como se trata de una deducción sin premisas hemos dejado un espacio en blanco en la primera fila, desplazando a la derecha la proposición que se introduce, con el nombre de supuesto, como premisa para un procedimiento auxiliar. En este caso ha sido la reducción al absurdo que va a terminar en la fila 7. Pero hemos hecho una segunda reducción al absurdo dentro del procedimiento auxiliar anterior, destacada con un nuevo desplazamiento a la derecha. Este procedimiento de desplazar a la derecha las pruebas auxiliares no es imprescindible (y en la deducción anterior está gráficamente exagerado) pero puede resultar útil sobre todo al principiante. En las filas 4 y 7 hemos puesto una contradicción, la que se obtiene según se indica en la columna de la derecha, en ambos casos es $(P \vee \neg P) \wedge \neg(P \vee \neg P)$. Con esta notación la premisa (el supuesto) de (RA) es P (el tramo 2-4 anterior). El tramo 1-7 en la deducción anterior es $\neg(P \vee \neg P)$.

Nótese que hemos probado por deducción natural la ley del tercero excluido ($\vdash P \vee \neg P$), pero no sólo eso. En realidad, primero hemos probado por deducción natural la negación de la ley ($\vdash \neg\neg(P \vee \neg P)$) y luego hemos deducido dicha ley aplicando la regla de eliminación de la doble negación. El tramo 1-5 es la prueba mediante reducción al absurdo de

$$\frac{\neg(P \vee \neg P)}{\neg P}$$

Vemos pues que por el camino que nos conduce a un teorema hemos deducido otro teorema y una regla. Esta situación se da constantemente en la práctica, pues la propia Definición de lo que es una deducción nos indica que fragmentos concretos de una deducción pueden ser a su vez deducciones. La producción de deducciones es un proceso acumulativo que se realimenta con los productos que obtiene. Cuando ya se ha demostrado un teorema se puede usar en deducciones posteriores, como si en la posición (i) de la Definición incluyéramos además de las premisas las proposiciones ya probadas. Y lo mismo cabe decir de las reglas de deducción no primitiva.

Establecido el procedimiento de la deducción natural, es necesario adquirir cierta destreza en el uso de este método de prueba.

Queda por considerar una cuestión teórica muy importante. como es examinar de cerca la relación entre las tautologías (semántica) y los teoremas (deducción) o, lo que es equivalente, entre las reglas de inferencia y las regla de deducción. El resultado siguiente contiene el meollo teórico de la teoría de la deducción.

Teorema 3.4.4 *(i) Teorema de corrección. Si* $\Gamma \vdash Q$ *entonces* $\Gamma \models Q$*, es decir, si existe una deducción natural de* Q *a partir de* Γ *entonces todo modelo de* Γ *es un modelo de* Q*.*

(ii) Teorema de completitud. Si $\Gamma \models Q$ *entonces* $\Gamma \vdash Q$*.*

Los dos resultados anteriores, uno recíproco del otro, nos dicen que la selección de las reglas y procedimientos primitivos de la deducción natural está bien hecha en dos sentidos. El teorema de corrección asegura que lo que se demuestra por deducción natural tiene validez semántica y el teorema de completitud asegura que la selección es completa porque se pueden demostrar por deducción natural absolutamente todas las reglas de inferencia. A veces estos teoremas se enuncian sólo con proposiciones, en vez de reglas de inferencia, lo que es el caso particular con un conjunto vacío de premisas.

Corolario 3.4.5 *(Corrección y completitud) Una proposición es una tautología si y sólo si es un teorema.*

La corrección del sistema de deducción natural es difícil de demostrar, pero la completitud no. De momento no haremos ninguna de las dos demostraciones, pero ello no obsta para que sea necesario conocer estos resultados y su importancia. El teorema de completitud es un enunciado existencial no constructivo, es decir, afirma que existe una deducción, pero no explica (ni en el enunciado ni en la demostración) cómo se construye la misma. Por eso deducir tiene algo de ingenio, precisa de la idea feliz.

Ejercicios

1. Demostrar por resolución que son tautologías los axiomas de Lukasiewicz:

(L1) $P \to (Q \to P)$,

(L2) $(P \to (Q \to R)) \to ((P \to Q) \to (P \to R))$ y

(L3) $(\neg P \to \neg Q) \to (Q \to P)$.

2. Hacer la prueba por deducción natural de:

$$\begin{array}{c}(P \wedge Q) \to (R \wedge S)\\ S \to (Q \wedge T)\\ S \wedge T\\ \hline P \to (Q \wedge R)\end{array}$$

3.5. El sistema de deducción de Frederic Fitch

El sistema deducción, que se basa en el trabajo de Frederic Fitch (publicado en 1952), va a ser utilizado en prácticas informáticas para realizar deducciones naturales. El sistema de deducción de Fitch es muy parecido al sistema de Gentzen. Sin embargo utiliza dos conectores adicionales: El bicondicional $\leftrightarrow$ y el símbolo de la contradicción $\perp$ (también se podría utilizar el símbolo que representa una tautología: $\top$). Hay que observar que no existe la regla de eliminación de la doble negación que sí que está incluida en el sistema deductivo de Gentzen.

En sistema de deducción de Fitch se considera el siguiente conjunto de reglas primitivas:

Regla de eliminación del conector condicional:

$$\begin{array}{r|ll} m & \mathscr{A} \to \mathscr{B} & \\ n & \mathscr{A} & \\ & \mathscr{B} & \to\text{E } m, n \end{array}$$

Regla de eliminación del conector negación:

$$\begin{array}{r|ll} m & \neg\mathscr{A} & \\ n & \mathscr{A} & \\ & \bot & \neg\mathrm{E}\ m,\ n \end{array}$$

Regla X, que viene de su nombre en inglés “eXplosion rule” o quizás de su nombre en latín “ex contradictione quod libet” (de la contradicción cualquier cosa; ex contradicción, es decir que ya no es contradicción).

$$\begin{array}{r|ll} m & \bot & \\ & \mathscr{A} & \mathrm{X}\ m \end{array}$$

Reglas de introducción y eliminación del connector conjunción:

$$\begin{array}{r|ll} m & \mathscr{A} & \\ n & \mathscr{B} & \\ & \mathscr{A} \wedge \mathscr{B} & \wedge\mathrm{I}\ m,\ n \end{array}$$

$$\begin{array}{r|ll} m & \mathscr{A} \wedge \mathscr{B} & \\ & \mathscr{A} & \wedge\mathrm{E}\ m \end{array}$$

$$\begin{array}{r|ll} m & \mathscr{A} \wedge \mathscr{B} & \\ & \mathscr{B} & \wedge\mathrm{E}\ m \end{array}$$

$$\begin{array}{r|ll} m & \mathscr{A} & \\ & \mathscr{A} \vee \mathscr{B} & \vee\mathrm{I}\ m \end{array}$$

Regla de introducción del conector disyunción:

$$\begin{array}{r|ll} m & \mathscr{A} & \\ & \mathscr{A} \vee \mathscr{B} & \vee\mathrm{I}\ m \end{array}$$

$$\begin{array}{r|ll} m & \mathscr{A} & \\ & \mathscr{B} \vee \mathscr{A} & \vee\mathrm{I}\ m \end{array}$$

Lista de procedimientos:

$$
\begin{array}{l|l l}
i & \quad \begin{array}{|l} \mathcal{A} \\ \hline \end{array} & \\
j & \quad \begin{array}{|l} \mathcal{B} \end{array} & \\
 & \mathcal{A} \rightarrow \mathcal{B} & \rightarrow\mathrm{I}\ i\text{–}j
\end{array}
$$

$$
\begin{array}{l|l l}
i & \quad \begin{array}{|l} \mathcal{A} \\ \hline \end{array} & \\
j & \quad \begin{array}{|l} \bot \end{array} & \\
 & \neg\mathcal{A} & \neg\mathrm{I}\ i\text{–}j
\end{array}
$$

$$
\begin{array}{l|l l}
m & \mathcal{A} \vee \mathcal{B} & \\
i & \quad \begin{array}{|l} \mathcal{A} \\ \hline \end{array} & \\
j & \quad \begin{array}{|l} \mathcal{C} \end{array} & \\
k & \quad \begin{array}{|l} \mathcal{B} \\ \hline \end{array} & \\
l & \quad \begin{array}{|l} \mathcal{C} \end{array} & \\
 & \mathcal{C} & \vee\mathrm{E}\ m,\ i\text{–}j,\ k\text{–}l
\end{array}
$$

$$
\begin{array}{l|l l}
i & \quad \begin{array}{|l} \neg\mathcal{A} \\ \hline \end{array} & \\
j & \quad \begin{array}{|l} \bot \end{array} & \\
 & \mathcal{A} & \mathrm{IP}\ i\text{–}j
\end{array}
$$

Donde en el último procedimiento IP significa *prueba indirecta*.
Utilizando además el connector bicondicional se pueden utilizar las siguientes reglas:
Regla de introducción del conector bicondicional:

$$
\begin{array}{l|l l}
i & \quad \begin{array}{|l} \mathcal{A} \\ \hline \end{array} & \\
j & \quad \begin{array}{|l} \mathcal{B} \end{array} & \\
k & \quad \begin{array}{|l} \mathcal{B} \\ \hline \end{array} & \\
l & \quad \begin{array}{|l} \mathcal{A} \end{array} & \\
 & \mathcal{A} \leftrightarrow \mathcal{B} & \leftrightarrow\mathrm{I}\ i\text{–}j,\ k\text{–}l
\end{array}
$$

Reglas de eliminación de conector bicondicional:

$$
\begin{array}{r|ll}
m & \mathscr{A} \leftrightarrow \mathscr{B} & \\
n & \mathscr{A} & \\
 & \mathscr{B} & \leftrightarrow\text{E } m,\ n
\end{array}
$$

y también

$$
\begin{array}{r|ll}
m & \mathscr{A} \leftrightarrow \mathscr{B} & \\
n & \mathscr{B} & \\
 & \mathscr{A} & \leftrightarrow\text{E } m,\ n
\end{array}
$$

Notemos que mediante el conjunto de reglas y procedimientos primitivos de Fitch podemos obtener la regla de eliminación de la doble negación del sistema deductivo de Gentzen.

$$
\begin{array}{r|ll}
1 & \neg\neg D & \\
\hline
2 & \quad \neg D & \\
3 & \quad \bot & \neg\text{E } 1, 2 \\
4 & D & \text{IP } 2\text{–}3
\end{array}
$$

Por otro lado, notemos que la regla de la introducción de la negación del sistema Fitch implica la regla de reducción al absurdo de Gentzen. Para ello hay que tener en cuenta la siguiente deducción del sistema de Fitch:

$$
\begin{array}{r|ll}
1 & D \wedge \neg D & \\
\hline
2 & \neg D & \wedge\text{E } 1 \\
3 & D & \wedge\text{E } 1 \\
4 & \bot & \neg\text{E } 2, 3
\end{array}
$$

Hay que observar que si tenemos proposiciones en las que aparece el símbolo $\bot$ y lo queremos eliminar para trabajar en un sistema deductivo de tipo Gentzen, podemos utilar la regla de la explosión:

$$
\begin{array}{r|ll}
m & \bot & \\
 & A \wedge \neg A & \text{X } m
\end{array}
$$

Para adquirir habilidad en Deducción Natural vamos a utilizar el corrector de deducciones denominado "Proof Checker for forall x: Cambridge and Calgary" desarrollado por Richard Zach de la Universidad de Calgary. Se puede ejecutar en línea en el siguiente enlace:

http://proofs.openlogicproject.org/

3.6. Reglas y procedimientos derivados básicos

La regla derivada básica de la *reiteration* (R) que permite deducir (repetir) una proposición a partir de ella misma:

$$\begin{array}{r|ll} m & \mathscr{A} & \\ & \mathscr{A} & \text{R } m \end{array}$$

La regla derivada básica del *silogismo disyuntivo* (DS) que a veces también la llamamos reglas de resolución unitaria:

$$\begin{array}{r|ll} m & \mathscr{A} \vee \mathscr{B} & \\ n & \neg\mathscr{A} & \\ & \mathscr{B} & \text{DS } m,\ n \end{array}$$

y

$$\begin{array}{r|ll} m & \mathscr{A} \vee \mathscr{B} & \\ n & \neg\mathscr{B} & \\ & \mathscr{A} & \text{DS } m,\ n \end{array}$$

La regla derivada básica *modus tollens*:

$$\begin{array}{r|ll} m & \mathscr{A} \to \mathscr{B} & \\ n & \neg\mathscr{B} & \\ & \neg\mathscr{A} & \text{MT } m,\ n \end{array}$$

Otra regla derivada básica es la regla de *eliminación de la doble negación* (DNE) que en el método deductivo de Gentzen denotábamos por ¬¬E.

$$\begin{array}{r|ll} m & \neg\neg\mathscr{A} & \\ & \mathscr{A} & \text{DNE } m \end{array}$$

También vamos a considerar como reglas derivadas básicas las *Leyes de De Morgan* (DeM) que son las siguientes:

La primera regla derivada básica de De Morgan es:

$$\begin{array}{r|ll} m & \neg(\mathscr{A} \wedge \mathscr{B}) & \\ & \neg\mathscr{A} \vee \neg\mathscr{B} & \text{DeM } m \end{array}$$

La segunda regla derivada básica de De Morgan es:

$$\begin{array}{r|ll} m & \neg\mathscr{A} \lor \neg\mathscr{B} & \\ & \neg(\mathscr{A} \land \mathscr{B}) & \text{DeM } m \end{array}$$

La tercera regla derivada básica de De Morgan es:

$$\begin{array}{r|ll} m & \neg(\mathscr{A} \lor \mathscr{B}) & \\ & \neg\mathscr{A} \land \neg\mathscr{B} & \text{DeM } m \end{array}$$

La cuarta regla derivada básica de De Morgan es:

$$\begin{array}{r|ll} m & \neg\mathscr{A} \land \neg\mathscr{B} & \\ & \neg(\mathscr{A} \lor \mathscr{B}) & \text{DeM } m \end{array}$$

A las reglas mencionadas en está sección las llamaremos reglas derivadas básicas.

Además vamos a consider un procedimiento derivado básico adicional que vamos a llamar la ley de medio excluido (LEM). La etiqueta que lo identifica viene del inglés law of *excluded middle* que en latín se meciona como “tertium non datur” y en castellano se indica como la ley the tercio excluso (también ley o principio del tercero excluido).

$$\begin{array}{r|ll} i & \quad \begin{array}{|l} \mathscr{A} \\ \hline \end{array} & \\ j & \quad \begin{array}{|l} \mathscr{B} \end{array} & \\ k & \quad \begin{array}{|l} \neg\mathscr{A} \\ \hline \end{array} & \\ l & \quad \begin{array}{|l} \mathscr{B} \end{array} & \\ & \mathscr{B} & \text{LEM } i\text{–}j,\ k\text{–}l \end{array}$$

3.7. Derivación de las reglas y procedimientos derivados básicos

En las siguientes subsecciones indicamos como se pueden deducir el conjunto de reglas y procedimientos básicos que admite el editor/chequeador de demostraciones a partir de las reglas y procedimientos básicos del método deductivo de Fitch.

3.7.1. Derivación de la reiteración

Para derivar la regla de la reiteración

$$\begin{array}{l|ll}
m & \mathscr{A} & \\
 & \mathscr{A} & \mathrm{R}\ m
\end{array}$$

mediante reglas y procedimientos primitivos, podemos proceder como indicamos a continuación:

Supongamos que en alguna línea de una deducción tenemos:

$$\begin{array}{l|l}
m & \mathscr{A}
\end{array}$$

Ahora podemos aplicar de modo consecutivo dos reglas primitivas que conducen a la derivación de la regla de la reiteración:

$$\begin{array}{l|ll}
m & \mathscr{A} & \\
k & \mathscr{A} \wedge \mathscr{A} & \wedge\mathrm{I}\ m, m \\
k+1 & \mathscr{A} & \wedge\mathrm{E}\ k
\end{array}$$

Notemos que el uso de la regla de reiteración permite disminuir en una unidad la longitud de una demostración que haya utilizado el patrón formado por las dos reglas primitivas anteriores.

3.7.2. Derivación del silogismo deductivo (Un caso de la regla de la resolución)

Supongamos que en una demostración (deducción por el método de Fitch) tenemos el siguiente patrón:

$$\begin{array}{l|l}
m & \mathscr{A} \vee \mathscr{B} \\
n & \neg\mathscr{A}
\end{array}$$

El uso de la regla derivada básica DS se puede evitar cada vez que se utilice mediante el uso de las siguientes reglas y procedimientos primitivos del método deductivo de Fitch:

$$\begin{array}{l|lll}
m & \mathscr{A} \vee \mathscr{B} & & \\
n & \neg\mathscr{A} & & \\
k & & \underline{\mathscr{A}} & \\
k+1 & & \bot & \neg\mathrm{E}\ n, k \\
k+2 & & \mathscr{B} & \mathrm{X}\ k+1 \\
k+3 & & \underline{\mathscr{B}} & \\
k+4 & & \mathscr{B} \wedge \mathscr{B} & \wedge\mathrm{I}\ k+3, k+3 \\
k+5 & & \mathscr{B} & \wedge\mathrm{E}\ k+4 \\
k+6 & \mathscr{B} & & \vee\mathrm{E}\ m, k\text{–}k+2, k+3\text{–}k+5
\end{array}$$

3.7.3. Derivation de la regla derivada básica modus tollens

Supongamos que en tu demostración quieres obtener $\neg\mathscr{A}$ de las proposiciones:

$$\begin{array}{l|l}
m & \mathscr{A} \to \mathscr{B} \\
n & \neg\mathscr{B}
\end{array}$$

Lo puedes hacer aplicando la regla derivada MT$m, n$ pero también lo puedes deducir utiizando reglas y procedimientos primitivos:

$$\begin{array}{l|lll}
m & \mathscr{A} \to \mathscr{B} & & \\
n & \neg\mathscr{B} & & \\
k & & \multicolumn{1}{|l}{\underline{\mathscr{A}}} & \\
k+1 & & \multicolumn{1}{|l}{\mathscr{B}} & \to\text{E } m,\ k \\
k+2 & & \multicolumn{1}{|l}{\bot} & \neg\text{E } n,\ k+1 \\
k+3 & \neg\mathscr{A} & & \neg\text{I } k\text{–}k+2
\end{array}$$

3.7.4. Derivación de la regla de la doble negación

La regla DNE de eliminación de la doble negación que no es primitiva se puede obtener así:

$$\begin{array}{l|lll}
m & \neg\neg\mathscr{A} & & \\
k & & \multicolumn{1}{|l}{\underline{\neg\mathscr{A}}} & \\
k+1 & & \multicolumn{1}{|l}{\bot} & \neg\text{E } m,\ k \\
k+2 & \mathscr{A} & & \text{IP } k\text{–}k+1
\end{array}$$

3.7.5. Derivación de la regla: ley del medio excluido

Si en una demostración se encuentran los subdemostaciones de la forma

$$\begin{array}{l|l}
m & \multicolumn{1}{|l}{\underline{\mathscr{A}}} \\
n & \multicolumn{1}{|l}{\mathscr{B}} \\
k & \multicolumn{1}{|l}{\underline{\neg\mathscr{A}}} \\
l & \multicolumn{1}{|l}{\mathscr{B}}
\end{array}$$

Se puede aplicar el procedimiento derivado básico LEM para obtener $\mathscr{B}$.

Para probarlo mediante reglas y procedimiento primitivos una opción es probar primero $\mathscr{A} \vee \neg\mathscr{A}$ y luego aplicar la regla de los casos:

m	$\mid\ \mid \mathscr{A}$	
n	$\mid\ \mid \mathscr{B}$	
k	$\mid\ \mid \neg\mathscr{A}$	
l	$\mid\ \mid \mathscr{B}$	
	$\mid \dots$	
i	$\mid \mathscr{A} \vee \neg\mathscr{A}$	
$i+1$	$\mid \mathscr{B}$	$\vee$E i, m–n, k–l

A continuación hemos insertado la deducción de $\mathscr{A} \vee \neg\mathscr{A}$ mediante el uso de reglas y procedimientos primitivos de Fitch.

m	$\mid\ \mid \mathscr{A}$	
n	$\mid\ \mid \mathscr{B}$	
k	$\mid\ \mid \neg\mathscr{A}$	
l	$\mid\ \mid \mathscr{B}$	
$l+1$	$\mid\ \mid\ \mid \neg(\mathscr{A} \vee \neg\mathscr{A})$	
$l+2$	$\mid\ \mid\ \mid\ \mid \mathscr{A}$	
$l+3$	$\mid\ \mid\ \mid\ \mid \mathscr{A} \vee \neg\mathscr{A}$	$\vee$I $l+2$
$l+4$	$\mid\ \mid\ \mid\ \mid \bot$	$\neg$E $l+1$, $l+3$
$l+5$	$\mid\ \mid\ \mid \neg\mathscr{A}$	$\neg$I $l+2$–$l+4$
$l+6$	$\mid\ \mid\ \mid \mathscr{A} \vee \neg\mathscr{A}$	$\vee$I $l+5$
$l+7$	$\mid\ \mid\ \mid \bot$	$\neg$E $l+1$, $l+6$
i	$\mid\ \mid \mathscr{A} \vee \neg\mathscr{A}$	IP $l+1$–$l+7$
$i+1$	$\mid\ \mid \mathscr{B}$	$\vee$E i, m–n, k–l

3.7.6. Derivación de las reglas de De Morgan

Incluimos a continuación una demostración de la primera regla de De Morgan: :

m	$\neg(\mathscr{A} \wedge \mathscr{B})$	
k	$\mathscr{A}$	
$k+1$	$\mathscr{B}$	
$k+2$	$\mathscr{A} \wedge \mathscr{B}$	$\wedge$I k, $k+1$
$k+3$	$\perp$	$\neg$E m, $k+2$
$k+4$	$\neg\mathscr{B}$	$\neg$I $k+1$–$k+3$
$k+5$	$\neg\mathscr{A} \vee \neg\mathscr{B}$	$\vee$I $k+4$
$k+6$	$\neg\mathscr{A}$	
$k+7$	$\neg\mathscr{A} \vee \neg\mathscr{B}$	$\vee$I $k+6$
$k+8$	$\neg\mathscr{A} \vee \neg\mathscr{B}$	LEM k–$k+5$, $k+6$–$k+7$

Y también de la segunda regla derivada básica de De Morgan:

m	$\neg\mathscr{A} \vee \neg\mathscr{B}$	
k	$\mathscr{A} \wedge \mathscr{B}$	
$k+1$	$\mathscr{A}$	$\wedge$E k
$k+2$	$\mathscr{B}$	$\wedge$E k
$k+3$	$\neg\mathscr{A}$	
$k+4$	$\perp$	$\neg$E $k+3$, $k+1$
$k+5$	$\neg\mathscr{B}$	
$k+6$	$\perp$	$\neg$E $k+5$, $k+2$
$k+7$	$\perp$	$\vee$E m, $k+3$–$k+4$, $k+5$–$k+6$
$k+8$	$\neg(\mathscr{A} \wedge \mathscr{B})$	$\neg$I k–$k+7$

Dejamos como ejercicio la prueba de los tercera y cuarta reglas de De Morgan.

Ejercicios:

1. Obtener la prueba de los tercera y cuarta reglas de De Morgan mediante el uso de reglas y procedimienos primitivos mediante en método de Fitch (también se pueden obtener mediante el método de Getzen).

3.8. Los axiomas de Lukasiewicz

Ya aprendimos a razonar formalmente con proposiciones siguiendo el método de la deducción natural, que parte de un conjunto de reglas y de algunos procedimientos primitivos de deducción.

Ahora nos vamos a ocupar de otro método de razonamiento formal, que llamaremos de los sistemas axiomáticos, cuyos datos primitivos son algo diferentes de los de la deducción natural. En efecto, un sistema axiomático de proposiciones consta de los siguientes elementos:

1. Un conjunto de proposiciones, llamadas axiomas.
2. Modus pones como única regla de deducción primitiva.
3. Un procedimiento, llamado deducción o demostración que permita obtener una fórmula a partir de otras utilizando los axiomas y (MP).

Los teoremas del sistema axiomático son los axiomas y las proposición que se pueden deducir a partir de ellos y de (MP). Las reglas de deducción del sistema axiomático son (MP) y los esquemas inferenciales cuya conclusión se puede deducir formalmente de las premisas. A semejanza con la deducción natural, una deducción formal es una sucesión finita de proposiciones que son premisas o axiomas o se obtienen a partir de dos de las anteriores mediante la regla (MP). También en este caso el sistema se realimenta, pues los teoremas y las reglas ya deducidas se pueden utilizar para llevar a cabo nuevas deducciones.

El planteamiento es pues similar al de la deducción natural en cuanto a la idea de deducción se refiere, pero la diferencia estriba en que los sistemas axiomáticos tienen una única regla de deducción primitiva y, para compensar esta escasez, adoptan un conjunto de axiomas. Unos sistemas de axiomas se diferencian de otros en el conjunto de axiomas que seleccionan. Al hacer esta selección, cada sistema elige a su vez los conectores primitivos que va a utilizar. Si el conector condicional no está entre los elegidos, se introduce como conector derivado para formular la regla de modus ponens.

Para ser considerado un sistema axiomático de la lógica de proposiciones, el conjunto de axiomas debe cumplir las dos condiciones siguientes, satisfechas también, como vimos, por la deducción natural (todos los sistemas que vamos a considerar a continuación son de este tipo):

1. (Teorema de corrección, que es fácil de probar) Los axiomas han de ser tautologías.
2. (Teorema de completitud, que es difícil de probar) Toda tautología debe ser un teorema del sistema.

Suponemos el conjunto $\mathscr{P}$ de las proposiciones construido con los conectores primitivos $\{\neg, \rightarrow\}$. Vamos a dar los datos de este sistema y veremos que a partir de ellos se deducen las reglas y los procedimientos primitivos de la deducción natural.

Definición 3.8.1 *El sistema axiomático de Lukasiewicz (1929) consta de la regla (MP) y de los axiomas siguientes (cada uno de los cuales incluye un conjunto de proposiciones):*

(L1) $P \rightarrow (Q \rightarrow P)$

(L2) $(P \rightarrow (Q \rightarrow R)) \rightarrow ((P \rightarrow Q) \rightarrow (P \rightarrow R))$

(L3) $(\neg P \rightarrow \neg Q) \rightarrow (Q \rightarrow P)$

Proposición 3.8.2 *Si $P \rightarrow Q$ es un teorema entonces $\frac{P}{Q}$ es una regla de deducción (Si $\vdash P \rightarrow Q$ entonces $P \vdash Q$).*

Demostración: Basta aplicar una vez (MP). □

Esta proposición nos permite dar las tres reglas que corresponden a los axiomas, que denotaremos anteponiendo al nombre del axioma la letra «R»:

$$\frac{P}{Q \rightarrow P}\ \text{(RL1)} \qquad \frac{P \rightarrow (Q \rightarrow R)}{(P \rightarrow Q) \rightarrow (P \rightarrow R)}\ \text{(RL2)} \qquad \frac{\neg P \rightarrow \neg Q}{Q \rightarrow P}\ \text{(RL3)}$$

Nótese que una forma particular de (RL1) se obtiene poniendo $\neg Q$ en vez de Q, de modo que la conclusión es $Q \vee P$ y resulta así demostrada en el sistema axiomático de Lukasiewicz una de las reglas de introducción de la disyunción:

$$\frac{P}{Q \vee P} \ (\vee \mathrm{I})$$

En lo que sigue iremos numerando de la forma (Tn) los teoremas que vayamos demostrando. Para las reglas conocidas usaremos la notación ya establecida y cuando una regla esté asociada a un teorema según la Proposici ón se indicará (RTn). El primer teorema que vamos a demostrar será utilizado enseguida en otra demostración.

Lema 3.8.3 *La siguiente proposición es un teorema: (T1) $P \to P$.*

Demostración:

1.	$P \to ((Q \to P) \to P)$	(L1)
2.	$(P \to ((Q \to P) \to P)) \to ((P \to (Q \to P)) \to (P \to P))$	(L2)
3.	$(P \to (Q \to P)) \to (P \to P)$	MP(1,2)
4.	$P \to (Q \to P)$	(L1)
5.	$P \to P$	MP(3,4)

□

Hay que reconocer que se necesita cierto ingenio para realizar la apertura 1, 2 en la deducción anterior. La regla asociada a este teorema según la Proposición 3.8.2 es $P \vdash P$, que por otra parte es evidente por la Definición misma de lo que es una deducción. Una de las formas posibles de (T1) se obtiene sustituyendo P por $\neg P$, resultando entonces la ley del tercero excluido

$$P \vee \neg P.$$

En los enunciados que siguen, Γ será siempre un conjunto finito, posiblemente vacío, de proposiciones. El primero de ellos no es sino una versión de aspecto más general que la Proposición 3.8.2 pero que se demuestra con la misma sencillez. En cambio, sí que es una mejora sustancial del método de prueba disponer del enunciado recíproco, que es el teorema de deducción, que deduciremos en este sistema axiomático.

Proposición 3.8.4 *Si $\Gamma \vdash P \to Q$ entonces $\Gamma \cup \{P\} \vdash Q$.*

Teorema 3.8.5 *(Teorema de deducción) De los axiomas (L1), (L2) y la regla (MP) se deduce lo siguiente: Si $\Gamma \cup \{P\} \vdash Q$ entonces $\Gamma \vdash P \to Q$.*

Demostración: Supongamos una deducción de la conclusión Q a partir de las premisas de Γ y de la premisa adicional P. Esta deducción será una sucesión finita de proposiciones, pero si escribimos en primer lugar las premisas podemos empezar a numerar a continuación el resto de las premisas que constituyen la deducción:

$$\Gamma, P, Q_1, \cdots, Q_n = Q$$

Demostraremos el teorema por inducción sobre la longitud de la deducción anterior, medida por n. Si $n = 1, 2$, la aparición de Q en la deducción sólo se puede justificar por una de estas situaciones: $Q \in \Gamma$, Q es un axioma, $Q = P$. En los dos primeros casos, la conclusión de (TD) se deduce de la regla (RL1). En el segundo caso basta usar el teorema (T1) demostrado en el Lema 3.8.3.

Supongamos ahora $n > 2$ y que la aparición de Q no es de ninguno de los tipos anteriores, en cuyo caso se pocedería del mismo modo. Entonces sólo queda la posibilidad de autorizar Q mediante modus ponens a partir de las dos proposiciones anteriores de la lista que constituye la deducción, es decir, que se tenga (con el orden de las dos primeras entradas tal vez invertido):

$$\begin{array}{lll} n-2 & Q_{n-2} = R \to Q & \ldots \\ n-1 & Q_{n-1} = R & \ldots \\ n & Q & \mathrm{MP}(2-2, n-1) \end{array}$$

Esto quiere decir que se han construido previamente las deducciones

$$\Gamma \cup \{P\} \vdash R \to Q, \quad \Gamma \cup \{P\} \vdash R,$$

ambas de longitud menor que n. Aplicando ahora la hipótesis inductiva, podemos concluir que tenemos deducciones

$$\Gamma \vdash P \to (R \to Q), \quad \Gamma \vdash P \to R.$$

Para obtener $\Gamma \vdash P \to Q$ basta enlazar estas deducciones con esta otra:

$$\begin{array}{c} P \to (R \to Q) \\ P \to R \\ \hline P \to Q \end{array}$$

cuya prueba es como sigue:

$$\begin{array}{llr} 1. & P \to (R \to Q) & \text{Premisa} \\ 2. & P \to R & \text{Premisa} \\ 3. & (P \to (R \to Q)) \to ((P \to R) \to (P \to Q)) & \text{(L2)} \\ 4. & (P \to R) \to (P \to Q) & \mathrm{MP}(1,3) \\ 5. & P \to Q & \mathrm{MP}(2,4) \end{array}$$

□

Como corolario de (TD) se puede demostrar el silogismo hipotético,

$$\{P \to Q, Q \to R\} \vdash P \to R, \quad (SH).$$

pero la prueba a partir de (TD) en el sistema axiomático de Lukasiewicz es la misma que en la deducción natural.

La demostración anterior de (TD) pone de manifiesto la diferencia entre este sistema axiomático y la deducción natural. Para obtener (TD) en el sistema axiomático se necesita una demostración ciertamente delicada, mientras que (TD) es dado como procedimiento deductivo primitivo en la deducción natural. Por eso esta última es más adecuada para practicar demostraciones en un curso de iniciación. Pero el sistema axiomático de Lukasiewicz muestra que el núcleo básico de la lógica se puede reducir a un conjunto mínimo de axiomas más la regla (MP).

Antes de abandonar el sistema axiomático de Lukasiewicz apenas iniciado, daremos un avance del uso que puede hacerse del tercer axioma, que se necesita para demostrar la reducción al absurdo. Demostraremos primero una regla y luego aplicaremos (TD) y su recíproco para obtener nuevos teoremas y reglas.

Proposición 3.8.6 *Se verifica:*
(i) $\neg P \vdash P \to Q$.
(ii) $\neg P \to (P \to Q)$ *(T2).*
(iii) $\{P, \neg P\} \vdash Q$.
(iv) $P \vdash P \vee Q$ *($\vee$I).*

Demostración: La prueba de (i) es:

1.	$\neg P$	Premisa
2.	$\neg P \to (\neg Q \to \neg P)$	L1
3.	$\neg Q \to \neg P$	(MP(1,2))
4.	$(\neg Q \to \neg P) \to (P \to Q)$	(L4))
5.	$P \to Q$	MP(3,4)

(ii) Aplicando (TD) a (i) resulta (T2).
(iii) Aplicar la Proposición 3.8.4.
(iv) Aplicando (TD) a (iii) tomando $\Gamma = \{P\}$ resulta ($\vee$I).

□

Por simplicidad en la notación llamaremos (RT2) a la regla (i) de la Proposición 3.8.6, pero nótese que en este caso la prueba de la regla ha sido anterior a la prueba del teorema. Para la regla (iii) (de la inconsistencia se deduce todo) podemos usar la clave (R0), donde al escribir 0 recordamos la inconsistencia de las premisas.

Dejamos este tema dando una secuencia de teoremas y enunciados generales en el orden en que pueden ser demostrados, e invitamos al lector a construir las deducciones correspondientes. La secuencia indica el camino para demostrar (RA) y (PC) en el sistema axiomático de Lukasiewicz, así como las reglas primitivas de la deducción natural todavía no demostradas:
(T3) $(P \vee P) \to (P \vee Q)$.
(T4) $(P \vee P) \to (Q \to P)$.
(T5) $(P \vee P) \to P$.
(T6) $(P \vee Q) \to (Q \vee P)$.
(RI $\vee$) $P \vdash Q \vee P$
(T6) $\neg\neg P \to P$, con (RT7)=(RE$\neg\neg$).
(RA) Si $\Gamma \cup \{P\}$ inconsistente, entonces $\Gamma \vdash \neg P$.
(PC) Si $P \vdash R$ y $Q \vdash R$ entonces $P \vee Q \vdash R$.
(RI$\wedge$) $\{P, Q\} \vdash P \wedge Q$.
(RE$\wedge$) $P \wedge Q \vdash P$.
(RE$\wedge$) $P \wedge Q \vdash Q$.

A partir de este momento, demostrar en el sistema axiomático de Lukasiewicz es lo mismo que demostrar mediante la deducción natural de Gentzen. Como ya demostramos antes (más bien propusimos al lector que lo hiciera él mismo) los axiomas de Lukasiewicz mediante la deducción natural, resulta que los dos métodos son equivalentes en cuanto que producen los mismos teoremas.

3.9. Otros sistemas de axiomas

Veremos otros conjuntos de proposiciones que, junto con la regla (MP), constituyen sistemas axiomáticos para la lógica de proposiciones con la misma potencia demostrativa que los anteriores. Todos los sistemas son pues equivalentes entre sí. Con cualquiera de ellos se podría repetir lo que hemos hecho con el sistema de Lukasiewicz, deducir los recursos primitivos de la deducción natural.

Empezaremos por el más antiguo de los sistemas axiomáticos. En el sistema de Russel y Whitehead (1910) se consideran los conectores primitivos $\{\neg, \vee\}$ y a partir de ellos se define $P \to Q = \neg P \vee Q$ y se enuncian los axiomas de este sistema, que son:

(RW1) $(P \vee P) \to P$

(RW2) $Q \to (P \vee Q)$

(RW3) $(P \vee Q) \to (Q \vee P)$

(RW4) $(P \vee (Q \vee R)) \to (Q \vee (P \vee R))$

(RW5)$(Q \to R) \to ((P \vee Q) \to (P \vee R))$

En el sistema de Hilbert y Bernays (1934, 1939) son primitivos los cinco conectores básicos y se toman como axiomas:

(HB1) $(P \to Q) \to ((Q \to R) \to (P \to R))$

(HB2) $P \to (Q \to P)$

(HB3) $(P \to (P \to Q)) \to (P \to Q)$

(HB4) $P \to \neg\neg P$

(HB5) $\neg\neg P \to P$

(HB6) $(P \to Q) \to (\neg q \to \neg P)$

(HB7) $(P \wedge Q) \to P$

(HB8) $(P \wedge Q) \to Q$

(HB9) $(P \to Q) \to ((P \to R) \to (P \to (Q \wedge R))$

(HB10) $P \to (P \vee Q)$

(HB11) $Q \to (P \vee Q)$

(HB12) $(P \to R) \to ((Q \to R) \to (P \vee Q \to R))$

(HB13) $(P \leftrightarrow Q) \to (P \to Q)$

(HB14) $(P \leftrightarrow Q) \to (Q \to P)$

(HB15) $(P \to Q) \to ((Q \to P) \to (P \leftrightarrow Q))$

En el sistema de Kleene (1952) son primitivos los conectores básicos excepto el bicondicional y se toman como axiomas:

(K1)$P \to (Q \to P)$

(K2) $(P \to (Q \to R)) \to ((P \to Q) \to (P \to R))$

(K3) $(P \wedge Q) \to P$

(K4) $(P \wedge Q) \to Q$

(K5) $(P \to (Q \to (P \wedge Q)$

(K6) $P \to (P \vee Q)$

(K7) $Q \to (P \vee Q)$

(K8) $(P \to R) \to ((Q \to R) \to (P \vee Q \to R))$

(K9) $(P \to Q) \to ((P \to \neg q) \to \neg P)$

(K10) $\neg\neg P \to P$

Un aspecto muy interesante del sistema de Kleene es que si se sustituye (K10) por el axioma

$$\text{(K10i)} \quad \neg P \to (P \to Q),$$

entonces resulta un sistema axiomático para la lógica de proposiciones intuicionista, que es más débil o general que la lógica clásica que estudiamos en este curso. Esto quiere decir que todos los teoremas intuicionistas son teoremas clásicos, pero que el recíproco no es cierto. Por ejemplo, (K10) es un teorema clásico que no es intuicionista.

Ejercicios

1. Demostrar los axiomas (HB) a partir de los axiomas (RW) y de (MP).
2. Demostrar los axiomas (K) a partir de los axiomas (RW) y de (MP).
3. Demostrar los axiomas (RW) a partir de los axiomas (HB) y de (MP).
4. Demostrar los axiomas (RW) a partir de los axiomas (K) y de (MP).