Capítulo 5

Lógica proposicional

5.1. Lenguaje Proposicional

5.1.1. Introducción

El primer objetivo de la lógica es simbolizar el razonamiento humano. Es decir, producir un lenguaje en el que al menos parte del razonamiento humano (matemático o no) pueda ser expresado. La Lógica proposicional es la aproximación más sencilla a este objetivo. Los elementos del lenguaje humano que vamos a manejar son sentencias declarativas que pueden ser ciertas o falsas (pero no ambas cosas o ninguna de las dos). Cada una de esas sentencias es lo que llamaremos *proposición* o *fórmula bien formada* (f.b.f.). El método de trabajo consiste en partir de sentencias simples y combinarlas para formar otras más elaboradas. Las sentencias más simples se denominan *fórmulas atómicas* y se combinan mediante lo que llamamos *conectivas* (o también *operadores lógicos*) y las *reglas de sintaxis* correspondientes.

Un ejemplo sencillo de la necesidad de las conectivas y las reglas de sintaxis puede ser el de un niño que escribiera: Hoy es mi cumpleaños. No voy al parque. Estoy enfadado. Todos añadiríamos a esta secuencia de fórmulas atómicas unas conectivas gramaticales para indicar la relación entre ellas; por ejemplo: *Estoy enfadado porque hoy es mi cumpleaños y no voy al parque.* Pero también podríamos componer: *Hoy es mi cumpleaños pero no voy al parque porque estoy enfadado.* Cada una de ellas tiene un significado diferente, así que el discurso del niño puede ser interpretado de distintas formas porque carece de elementos de ligadura entre las distintas sentencias. Además el enlace entre los distintos elementos del discurso debe seguir unas reglas de sintaxis; no sería correcto decir: *Pero hoy es mi cumpleaños o no voy al parque o estoy enfadado.*

5.1.2. Elementos del lenguaje proposicional:

Como ya hemos adelantado, necesitamos los siguientes tipos de elementos para construir el lenguaje:

Las proposiciones básicas

Operadores lógicos o Conectivas

Sintaxis del cálculo proposicional

Las proposiciones básicas

En el lenguaje humano las proposiciones básicas pueden ser

Los enunciados de acción,

los enunciados de propiedades de sujetos,

los enunciados de relación entre sujetos.

Ejemplo 5.1.1. *Hace calor, Llueve, Enrique conduce autobuses, La lluvia se lleva la contaminación.*

Mi coche está sucio, Belén es una chica, La ciudad está contaminada

El autobús contamina menos que los coches, Belén es la mujer de Enrique

En el cálculo proposicional (o lógica proposicional) *las proposiciones básicas son los elementos de un conjunto predeterminado (finito o infinito numerable).* No se trata por tanto de elementos con un significado asignado.

Ejemplo 5.1.2. $\{HC, L, ECA, LLSLLA, MCES, BEUC, LCEC, EACMQLC, BELMDE\}$

$\{H, L1, E, L2, M, B, L3, E, B\}$

$\{a, b, g, d, e, j, r, w\}$

$\{a, b, c, d\}$

El conjunto de todas las proposiciones básicas en el lenguaje humano.

Operadores lógicos o Conectivas:

En el lenguaje humano la conexión entre distintos enunciados se hace mediante partículas como las conjunciones; en el lenguaje proposicional consideraremos unos símbolos que nos permitirán construir elementos complejos a partir de los más simples, estos son:

$$\neg, \wedge, \vee, \rightarrow, \leftrightarrow$$

$\neg$ se lee **no** (negación)

$\wedge$ se lee **y** (conjunción)

$\vee$ se lee **o** (disyunción)

$\rightarrow$ se lee **implica** (implicación)

$\leftrightarrow$ se lee **equivale** (equivalencia o doble implicación)

Sintaxis del cálculo proposicional (Reglas para la formación de nuevas proposiciones a partir de las básicas):

Son consideradas *proposiciones o fórmulas bien formadas (fbfs)* del cálculo proposicional:

Las proposiciones básicas, a las que llamamos *proposiciones atómicas*

Si a y b son fbfs, también lo son:

$$a \wedge b$$

$$a \vee b$$

$$\neg a$$

$$a \rightarrow b$$

y

$$a \leftrightarrow b$$

Ejemplo 5.1.3. *Para el conjunto de proposiciones atómicas*

$$X = \{\alpha, \beta, \gamma, \varphi\}$$

son fbfs:

$(\alpha \wedge \neg\beta) \rightarrow ((\gamma \vee \varphi) \vee \alpha)$

$\gamma \wedge (\alpha \vee \beta) \rightarrow (\neg\alpha \vee \beta)$

$(\gamma \vee \neg\alpha) \vee (\neg\alpha \rightarrow \beta)$

y no *son fbfs:*

$\alpha\neg\beta$

$\gamma(\alpha \vee \beta) \rightarrow \beta$

5.1.3. Uso de las reglas de prioridad en la escritura de fbfs:

Como en las fórmulas algebraicas o aritméticas, los paréntesis se usan para anteponer el uso de una conectiva al de otras, pero existen aquí también reglas de escritura que permiten prescindir de los paréntesis en determinados casos. Para ello tendremos en cuenta que:

1. Cualquier conectiva dentro de un paréntesis tiene prioridad sobre las demás.

2. $\neg$ tiene prioridad sobre todos las otras, es decir,
   $\neg a \wedge b$ sustituye a $(\neg a) \wedge b$ y es diferente de $\neg(a \wedge b)$,
   $\neg a \vee b$ sustituye a $(\neg a) \vee b$ y es diferente de $\neg(a \vee b)$,
   $\neg a \rightarrow b$ sustituye a $(\neg a) \rightarrow b$ y es diferente de $\neg(a \rightarrow b)$,
   $\neg a \leftrightarrow b$ sustituye a $(\neg a) \leftrightarrow b$ y es diferente de $\neg(a \leftrightarrow b)$.

3. $\wedge$ y $\vee$ tienen el mismo rango de prioridad, pero ambas tienen mayor prioridad que $\rightarrow$ y $\leftrightarrow$, por tanto,
   $a \wedge b \rightarrow c$ sustituye a $(a \wedge b) \rightarrow c$ y es diferente de $a \wedge (b \rightarrow c)$,
   $a \rightarrow b \vee c$ sustituye a $a \rightarrow (b \vee c)$ y es diferente de $(a \rightarrow b) \vee c$.

   NO es conveniente escribir

   $$a \wedge b \vee c$$

   sino que hay que distinguir entre $(a \wedge b) \vee c$, en la que primero se construye la fbf $a \wedge b$, después se utiliza la conectiva $\vee$ entre ella y la proposición atómica c, y $a \wedge (b \vee c)$ en la que en primer lugar se forma $b \vee c$ para luego obtener la fórmula con la proposición atómica a, la conectiva $\wedge$ y la fbf $b \vee c$. (Pronto veremos que estas dos fórmulas son distintas no solo en la escritura)

   Del mismo modo NO es correcto escribir

   $$a \rightarrow b \rightarrow c$$

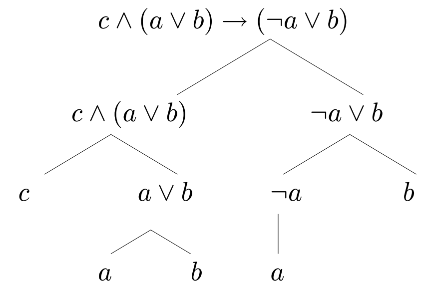
   Pregunta: ¿Se puede escribir $a \vee b \vee c$ ó $a \wedge b \wedge c$?

4. Es frecuente usar $\rightarrow$ con preferencia sobre $\leftrightarrow$, aunque somos partidarios de utilizar paréntesis para priorizar la conectiva que se desee entre estas dos.

5. *En caso de duda siempre es preferible abusar de los paréntesis.*

Árbol de formación de una fórmula y subfórmulas

Para practicar sobre las reglas de uso de paréntesis en las construcciones de fórmulas y familizarizarnos con ellas podríamos realizar un sencillo ejercicio; se trata de desglosar una fbf construyendo un árbol que tiene por raíz a la fórmula dada, y las ramas que salen de cada nodo nos llevan a las fbf que junto con la correspondiente conectiva formaban ese nodo. Las hojas serán las proposiciones atómicas. Por ejemplo: *cccccc* tendría asociado el árbol:



El recorrido del árbol partiendo de las hojas hacia la raíz nos da el procedimiento de construcción de la f.b.f. a partir de las proposiciones atómicas a, b y c.

Fórmulas para desglosar:

1. $a \wedge \neg b \to c \vee (e \wedge a)$
2. $c \wedge (a \vee b) \to \neg a \vee b$
3. $\neg(a \to b) \to a \wedge \neg(a \wedge b)$
4. $a \wedge (a \vee b \to d) \wedge (d \to \neg a)$
5. $(a \wedge c) \vee b \to d \wedge (d \to \neg a)$
6. $\neg a \to (b \to a) \wedge \neg(a \wedge b)$
7. $(a \wedge \neg(b \to c \vee e)) \vee a$
8. $b \wedge (a \vee b) \to d \wedge \neg(d \to \neg a)$
9. $b \wedge a(\neg b \to d \wedge \neg(d \to \neg a))$
10. $\neg(b \to a) \wedge \neg(a \wedge b) \to \neg a \vee b$

Dada una fórmula α, una subfórmula suya es una fórmula que aparece en algún nodo de su árbol de formación.

Más precisamente, si α es una fórmula definimos el conjunto $Sub(\alpha)$ como sigue:

1. Si α es una fórmula atómica, $Sub(\alpha) = \{\alpha\}$.
2. Si α es de la forma $\alpha_1 \vee \alpha_2$, $Sub(\alpha) = \{\alpha\} \cup Sub(\alpha_1) \cup Sub(\alpha_2)$.
3. Si α es de la forma $\alpha_1 \wedge \alpha_2$, $Sub(\alpha) = \{\alpha\} \cup Sub(\alpha_1) \cup Sub(\alpha_2)$.
4. Si α es de la forma $\alpha_1 \to \alpha_2$, $Sub(\alpha) = \{\alpha\} \cup Sub(\alpha_1) \cup Sub(\alpha_2)$.
5. Si α es de la forma $\alpha_1 \leftrightarrow \alpha_2$, $Sub(\alpha) = \{\alpha\} \cup Sub(\alpha_1) \cup Sub(\alpha_2)$.
6. Si α es de la forma $\neg\alpha_1$, $Sub(\alpha) = \{\alpha\} \cup Sub(\alpha_1)$.

Por ejemplo, vamos a calcular el conjunto de subfórmulas de $\alpha = c \wedge (a \vee b) \to (\neg a \vee b)$. Entonces: $Sub(\alpha) = \{c \wedge (a \vee b) \to (\neg a \vee b)\} \;\cup\; Sub(c \wedge (a \vee b)) \;\cup\; Sub(\neg a \vee b)$.

- $Sub(c \wedge (a \vee b)) = \{c \wedge (a \vee b)\} \cup Sub(c) \cup Sub(a \vee b)$.
  - $Sub(c) = \{c\}$.
  - $Sub(a \vee b) = \{a \vee b\} \cup Sub(a) \cup Sub(b) = \{a \vee b\} \cup \{a\} \cup \{b\} = \{a, b, a \vee b\}$.

  Por tanto,
  $Sub(c \wedge (a \vee b)) = \{c \wedge (a \vee b)\} \cup \{c\} \cup \{a, b, a \vee b\} = \{a, b, c, a \vee b, c \wedge (a \vee b)\}$.
- $Sub(\neg a \vee b) = \{\neg a \vee b\} \cup Sub(\neg a) \cup Sub(b)$.
  - $Sub(\neg a) = \{\neg a\} \cup Sub(a) = \{\neg a\} \cup \{a\} = \{a, \neg a\}$.
  - $Sub(b) = \{b\}$.

  Luego $Sub(\neg a \vee b) = \{\neg a \vee b\} \cup \{a, \neg a\} \cup \{b\} = \{a, b, \neg a, \neg a \vee b\}$.

Y volviendo a la fórmula α se tiene que:

$$\begin{aligned} Sub(\alpha) &= \{c \wedge (a \vee b) \to (\neg a \vee b)\} \cup \{a, b, c, a \vee b, c \wedge (a \vee b)\} \cup \{a, b, \neg a, \neg a \vee b\} \\ &= \{a,\ b,\ c,\ \neg a,\ a \vee b,\ \neg a \vee b,\ c \wedge (a \vee b),\ c \wedge (a \vee b) \to (\neg a \vee b)\} \end{aligned}$$

que vemos que son exactamente las que aparecían en el árbol de formación de la fórmula.

Traducción del lenguaje humano a la lógica proposicional:

No es de nuestro interés profundizar en la interpretación del lenguaje humano en términos de la lógica proposicional, pero nos permitimos escribir un ejemplo sencillo que relaciona ambos lenguajes.

Consideremos el razonamiento:

Si estudio quizás pueda o no aprobar el examen, pero si no lo hago seguro que no aprobaré el examen
Podemos extaer unas sentencias básicas que serían:

P = Yo estudio Q = Yo apruebo el examen
a partir de las cuales podemos construir el enunciado completo:

$$(P \to Q \vee \neg Q) \wedge (\neg P \to \neg Q)$$

5.2. Semántica de la lógica proposicional

Nuestro objetivo final en este curso es obtener métodos que nos permitan decidir si un determiando razonamiento es correcto. Para ello introducimos ahora una forma de asignar un cierto *significado* a un conjunto de proposiciones.

5.2.1. Interpretaciones o valoraciones

Si X es un conjunto de proposiciones atómicas, una **interpretación** para las proposiciones que se obtienen de X es una aplicación

$$I : X \to \{0, 1\}$$

que se extiende a todas las fórmulas bien formadas mediante las reglas:

$$\begin{array}{l} I(\neg a) = 1 + I(a) \\ I(a \vee b) = I(a) + I(b) + I(a)I(b) \\ I(a \wedge b) = I(a)I(b) \\ I(a \to b) = 1 + I(a) + I(a)I(b) \\ I(a \leftrightarrow b) = 1 + I(a) + I(b) \end{array}$$

Cuadro 5.1: Interpretaciones y conectivas

donde las operaciones se efectúan en $\mathbb{Z}_2$. El valor $I(\alpha)$ se llama **el valor de verdad de la fórmula** α bajo la interpretación I, y frecuentemente usaremos los términos *verdadera* o *falsa* para afirmar que el valor de verdad de una fórmula es 1 o 0 respectivamente.

En otros textos las interpretaciones son nombradas como ”valoraciones” y para designarlas se usa la letra "v".

Hay que observar que si el conjunto de proposiciones atómicas de partida tiene n elementos, entonces podemos elegir 2^n interpretaciones distintas para él.

Ejemplo 5.2.1. *Consideremos el conjunto de fórmulas atómicas $X = \{a, b\}$ y para él la interpretación que asigna*

$$I_1(a) = 0; \quad I_1(b) = 1$$

entonces, para calcular el valor de verdad de la fórmula

$$a \wedge b \to \neg a$$

calculamos en primer lugar $I_1(a \wedge b) = I_1(a)I_1(b) = 0 \cdot 1 = 0$ así como $I_1(\neg a) = 1 + I_1(a) = 1 + 0 = 1$; ahora podemos determinar

$$I_1((a \wedge b) \to \neg a) = 1 + I_1(a \wedge b) + I_1(a \wedge b)I_1(\neg a) = 1 + 0 + 0 \cdot 1 = 1$$

Por tanto la fórmula es verdadera bajo la interpretación I_1. Tomemos ahora otra interpretación que asigne

$$I_2(a) = 1; \quad I_2(b) = 1$$

y volvemos a calcular en la misma secuencia que antes el valor de verdad para la misma fórmula

$$a \wedge b \to \neg a$$

Tenemos ahora $I_2(a \wedge b) = I_2(a)I_2(b) = 1 \cdot 1 = 1$; $I_2(\neg a) = 1 + I_2(a) = 1 + 1 = 0$; entonces

$$I_2((a \wedge b) \to \neg a) = 1 + I_2(a \wedge b) + I_2(a \wedge b)I_2(\neg a) = 1 + 1 + 1 \cdot 0 = 0$$

Por tanto la fórmula es falsa bajo la interpretación I_2.

En este ejemplo observamos que el cálculo del valor de una interpretación para una fórmula puede hacerse a través del árbol de descomposición de la fórmula que aprendimos en la sección anterior, partiendo de las hojas y aplicando la tabla 5.1 en cada nodo.

Ejemplo 5.2.2. *Consideremos la fbf*

$$\alpha = \neg(a \to b) \to (\neg a \to \neg b)$$

Tratamos de calcular a continuación el valor de una interpretación de esta fórmula en función de las fórmulas atómicas que intervienen. LLamemos

$$\beta = a \to b; \qquad \delta = \neg a \to \neg b$$

entonces

$$I(\alpha) = 1 + I(\neg\beta) + I(\neg\beta)I(\delta) = 1 + (1 + I(\beta)) + (1 + I(\beta))I(\delta)$$

como $1 + 1 = 0$ en $\mathbb{Z}_2$

$$I(\alpha) = I(\beta) + (1 + I(\beta))I(\delta)$$

calculamos primero

$$I(\beta) = 1 + I(a) + I(a)I(b)$$

$$I(\delta) = 1 + I(\neg a) + I(\neg a)I(\neg b) = 1 + (1 + I(a)) + (1 + I(a))(1 + I(b))$$

simplificando y desarrollando los productos

$$I(\delta) = I(a) + 1 + I(a) + I(b) + I(a)I(b)$$

como en cualquier caso $I(a) + I(a) = 0$, queda

$$I(\delta) = 1 + I(b) + I(a)I(b)$$

y sustituyendo

$$I(\alpha) = 1 + I(a) + I(a)I(b) + (1 + 1 + I(a) + I(a)I(b))(1 + I(b) + I(a)I(b))$$

de nuevo simplificando $1 + 1 = 0$ y desarrollando el producto

$$= 1 + I(a) + I(a)I(b) + (I(a) + I(a)I(b) + I(a)I(b) + I(a)I(b)^2 + I(a)^2I(b) + (I(a)I(b))^2)$$

tendremos en cuenta ahora que en $\mathbb{Z}_2$ el cuadrado de cualquier elemento es el propio elemento ($0^2 = 0$ y $1^2 = 1$) y de nuevo que $I(a)I(b) + I(a)I(b) = 0$ con lo que resulta

$$= 1 + I(a) + I(a)I(b) + I(a) + I(a)I(b) + I(a)I(b) + I(a)I(b) + I(a)I(b) + I(a)I(b) =$$
$$= 1$$

Es decir, se obtiene que bajo cualquier interpretación de las proposiciones atómicas el valor de verdad de la fórmula es 1, a las fórmulas que verifican esta propiedad las llamaremos **tautologías**.

5.2.2. Tabla de verdad para una fórmula

Un método alternativo al cálculo de interpretaciones en $\mathbb{Z}_2$ es tener en cuenta la siguiente tabla:

a	b	$a \wedge b$	$a \vee b$	$\neg a$	$a \rightarrow b$	$a \leftrightarrow b$
0	0	0	0	1	1	1
0	1	0	1	1	1	0
1	0	0	1	0	0	0
1	1	1	1	0	1	1

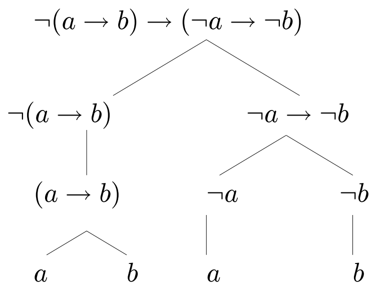
Cuadro 5.2: Tabla de verdad

en la que se han resumido todas las posibilidades para los valores de verdad de una fórmula en la que sólo aparece una conectiva. A partir de esta tabla (que hay que memorizar), se pueden calcular los valores de verdad de una fórmula para **todas las interpretaciones** posibles del conjunto de fórmulas atómicas que intervengan; es lo que se llama **la tabla de verdad** de una fórmula.

Ejemplo 5.2.3. *Consideremos la fbf*

$$\alpha = \neg(a \rightarrow b) \rightarrow (\neg a \rightarrow \neg b)$$

Podemos obtener la **tabla de verdad de la fórmula** *en la que aparecen todas las interpretaciones posibles, es decir, todas las asignaciones a las proposiciones atómicas y el resultado de extenderlas a la fórmula completa. Para ello se dibuja una tabla de doble entrada en la que en las columnas de la primera fila se escriben las subfórmulas que intervienen en la expresión, en realidad son los nodos distintos del árbol que resulta al desglosar la fórmula.*



En este caso la primera línea sería

a	b	$a \rightarrow b$	$\neg(a \rightarrow b)$	$\neg a$	$\neg b$	$\neg a \rightarrow \neg b$	$\neg(a \rightarrow b) \rightarrow (\neg a \rightarrow \neg b)$

Ahora se añaden tantas filas como interpretaciones distintas se puedan asignar al conjunto de proposiciones atómicas que intervienen, en este caso son dos a y b y por tanto son 2^2 posibilidades:

a	b
0	0
0	1
1	0
1	1

a	b	$a \to b$	$\neg(a \to b)$	$\neg a$	$\neg b$	$\neg a \to \neg b$	$\neg(a \to b) \to (\neg a \to \neg b)$
0	0			1	1	1	
0	1			1	0	0	
1	0			0	1	1	
1	1			0	0	1	

Sólo queda calcular cada una de las columnas en blanco haciendo uso de las tablas conocidas para las conectivas y las columnas que corresponden a subfórmulas hijas en el árbol de desglose. Así, para calcular la columna de $\neg a \to \neg b$ tendremos en cuenta las de $\neg a$ y de $\neg b$.

Si completas la tabla anterior correctamente debes obtener como última columna los valores $(1, 1, 1, 1)$. Es decir, tendrás de nuevo que es una **tautología***.*

Ejemplo 5.2.4. *La siguiente es la tabla de verdad de la fórmula:*

$$\varphi = (a \wedge b \to c) \wedge (\neg(a \wedge b) \to d)$$

a	b	c	d	$a \wedge b$	$a \wedge b \to c$	$\neg(a \wedge b)$	$\neg(a \wedge b) \to d$	φ
0	0	0	0	0	1	1	0	0
0	0	0	1	0	1	1	1	1
0	0	1	0	0	1	1	0	0
0	0	1	1	0	1	1	1	1
0	1	0	0	0	1	1	0	0
0	1	0	1	0	1	1	1	1
0	1	1	0	0	1	1	0	0
0	1	1	1	0	1	1	1	1
1	0	0	0	0	1	1	0	0
1	0	0	1	0	1	1	1	1
1	0	1	0	0	1	1	0	0
1	0	1	1	0	1	1	1	1
1	1	0	0	1	0	0	1	0
1	1	0	1	1	0	0	1	0
1	1	1	0	1	1	0	1	1
1	1	1	1	1	1	0	1	1

5.2.3. Clasificación de fórmulas

Hemos visto en la sección precedente que hay fórmulas que son ciertas para cualquier interpretación. Es lo que hemos llamado una tautología. Vamos a continuación a clasificar las fórmulas según los valores de verdad que pueden tomar:

Definición 48. *Sea α una fórmula de un lenguaje proposicional:*

1. *α es una* **tautología** *si para cualquier interpretación I se tiene que $I(\alpha) = 1$.*
2. *α es* **satisfacible** *si existe al menos una interpretación I para la que $I(\alpha) = 1$.*
3. *α es* **refutable** *si existe al menos una interpretación I para la que $I(\alpha) = 0$.*
4. *α es* **contradicción** *si para cualquier interpretación I se tiene que $I(\alpha) = 0$.*
5. *α es* **contingente** *si es satisfacible y refutable.*

Veamos algunos ejemplos:

Ejemplo 5.2.5.

- *La fórmula* $\alpha = p \to (q \to p)$ *es una tautología. Para comprobarlo, construyamos su tabla de verdad:*

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

y vemos que la fórmula es cierta para cualquier interpretación.

La fórmula también es satisfacible, pues hay una interpretación (en este caso todas) para la que la fórmula es cierta.

- *La fórmula* $\alpha = p \vee \neg q \to (\neg r \wedge (q \to p))$ *es satisfacible, pues si tomamos la interpretación* $I(p) = I(q) = 1, I(r) = 0$ *se tiene que* $I(\alpha) = 1$*, ya que:*

$I(p \vee \neg q) = 1$*;* $I(q \to p) = 1$*;* $I(\neg r) = 1$*;* $I(\neg r \wedge (q \to p)) = 1$ *y por tanto,* $I(\alpha) = 1$.

También es refutable, pues si para la interpretación $I(p) = I(q) = I(r) = 1$ *la fórmula es falsa.*

$I(p \vee \neg q) = 1$*;* $I(q \to p) = 1$*;* $I(\neg r) = 0$*;* $I(\neg r \wedge (q \to p)) = 0$ *y por tanto,* $I(\alpha) = 0$.

La fórmula α *es entonces contingente.*

- *La fórmula* $\alpha = (p \to \neg q) \wedge (p \wedge q)$ *es una contradicción, pues su tabla de verdad es*

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

Observación:

1. Podemos considerar las fórmulas divididas en cuatro grupos: tautologías, satisfacibles, refutables y contradicciones.

   En tal caso, cada fórmula pertenece exactamente a dos de estos grupos. Puede ser tautología y satisfacible, puede ser satisfacible y refutable, o puede ser refutable y contradicción.

2. Si en lugar de dividirlas en los cuatro grupos anteriores lo hacemos en los tres siguientes: tautologías, contingentes y contradicciones; entonces cada fórmula pertenece a uno (y sólo uno) de esos tres grupos.

3. Si α es una fórmula, no significa lo mismo decir α *no es tautología* que decir $\neg\alpha$ *es tautología*.

   En el primer caso estamos diciendo que α es refutable, mientras que en el segundo que α es contradicción.

   Es decir, α es refutable si, y sólo si, α no es tautología; y α es contradicción si, y sólo si, $\neg\alpha$ es tautología.

A continuación vamos a enumerar algunas fórmulas que son tautologías. Tanto α, β como γ representan fórmulas cualesquiera de un lenguaje proposicional (no necesariamente fórmulas atómicas).

1. $\alpha \to \alpha$.

2. $\neg\neg\alpha \to \alpha$.

3. $\alpha \to \neg\neg\alpha$.

4. $\alpha \vee \beta \to \beta \vee \alpha$.
5. $\alpha \wedge \beta \to \beta \wedge \alpha$.
6. $\alpha \to (\beta \to \alpha)$.
7. $\alpha \to \alpha \vee \beta$.
8. $\alpha \wedge \beta \to \alpha$.
9. $\alpha \vee \neg\alpha$.
10. $\neg(\alpha \wedge \neg\alpha)$.
11. $((\alpha \to \beta) \wedge \alpha) \to \beta$.
12. $((\alpha \to \beta) \wedge \neg\beta) \to \neg\alpha$.
13. $((\alpha \vee \beta) \wedge \neg\alpha) \to \beta$.
14. $(\neg\alpha \to \alpha) \to \alpha$.
15. $(\alpha \to (\beta \to \gamma)) \to ((\alpha \to \beta) \to (\alpha \to \gamma))$.
16. $(\neg\alpha \to \neg\beta) \to ((\neg\alpha \to \beta) \to \alpha)$.
17. $(\alpha \to \beta) \to (\neg\beta \to \neg\alpha)$.
18. $(\alpha \to \beta) \to ((\gamma \to \beta) \to (\alpha \vee \gamma \to \beta))$.
19. $(\alpha \to \beta) \to ((\alpha \to \gamma) \to (\alpha \to \beta \wedge \gamma))$.

Definición 49. *Sea $\Gamma = \{\gamma_1, \gamma_2, \cdots, \gamma_n\}$ un conjunto de fórmulas de un lenguaje proposicional. Se dice que Γ es* **satisfacible** *si existe una interpretación I para la que $I(\gamma_1) = I(\gamma_2) = \cdots = I(\gamma_n) = 1$.*

Si $\Gamma = \emptyset$ entonces Γ es satisfacible.

Un conjunto de fórmulas es insatisfacible si no es satisfacible.

Es decir, un conjunto es satisfacible si existe una iterpretación para la que todas las fórmulas son ciertas. En el caso del conjunto vacío, cualquier interpretación hace ciertas (y falsas) todas las fórmulas. Por eso, el conjunto vacío es satisfacible.

Se tiene entonces que Γ es insatisfacible si no existe ninguna interpretación I que haga ciertas todas las fórmulas. Si $\Gamma \neq \emptyset$ esto es equivalente a que para cualquier interpretación I existe un elemento $\alpha \in \Gamma$ tal que $I(\alpha) = 0$.

Proposición 5.2.1. *Sea $\Gamma = \{\gamma_1, \gamma_2, \cdots, \gamma_n\}$ un conjunto de fórmulas. Entonces son equivalentes:*

1. *Γ es insatisfacible*
2. *$\gamma_1 \wedge \gamma_2 \wedge \cdots \wedge \gamma_n$[1] es contradicción.*
3. *Para cualquier interpretación I, $I(\gamma_1 \wedge \gamma_2 \wedge \cdots \wedge \gamma_n) = 0$.*
4. *Para cualquier interpretación I, $\prod_{i=1}^{n} I(\gamma_i) = 0$.*

Notemos que si $\Gamma = \{\gamma_1, \gamma_2, \cdots, \gamma_n\}$ es un conjunto de fórmulas, y una (o más) de las fórmulas γ_i es de la forma $\gamma_i = \gamma_{i1} \wedge \gamma_{i2}$, entonces Γ es insatisfacible si, y sólo si, lo es el conjunto que resulta de sustituir en Γ la fórmula γ_i por las dos fórmulas γ_{i1} y γ_{i2}.

Ejemplo 5.2.6. *Sean $\gamma_1 = p \vee \neg q \to q$, $\gamma_2 = p \leftrightarrow q$ y $\gamma_3 = q \to (p \leftrightarrow \neg q)$. Entonces el conjunto $\Gamma = \{\gamma_1, \gamma_2, \gamma_3\}$ es insatisfacible, pues no hay ninguna interpretación para la que las tres fórmulas sean ciertas. Para esto, construimos la tabla de verdad de las tres fórmulas (no vamos a indicar los pasos intermedios)*

[1] En principio, la notación $\gamma_1 \wedge \gamma_2 \wedge \cdots \wedge \gamma_n$ puede ser ambigua, pues no hemos visto que la conjunción sea asociativa. Entenderemos entonces que$\gamma_1 \wedge \gamma_2 \wedge \cdots \wedge \gamma_n$ se corresponde con la fórmula $(\cdots(\gamma_1 \wedge \gamma_2) \wedge \cdots \wedge \gamma_n)$

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

y vemos que no hay ninguna interpretación para la que las tres fórmulas sean ciertas. Obviamente, si calculáramos la tabla de verdad de $\gamma_1 \wedge \gamma_2 \wedge \gamma_3$ *nos quedaría una columna con todo ceros.*

p	q	$p \vee \neg q \to p$	$p \leftrightarrow \neg q$	$q \to (p \leftrightarrow \neg q)$	$(\gamma_1 \wedge \gamma_2) \wedge \gamma_3$
0	*0*	*0*	*1*	*1*	*0*
0	*1*	*1*	*0*	*0*	*0*
1	*0*	*1*	*0*	*1*	*0*
1	*1*	*1*	*1*	*0*	*0*

5.2.4. Inciso

FORMALIZACIÓN DEL CASTELLANO UTILIZANDO LOS ESCASOS RECURSOS DE PROPOSICIONAL.

Del libro: Razón, dulce razón; T. Tymoczko y J. Henle (F-4 TYM- raz) (Páginas 139 en adelante)

Algunos libros de texto de lógica dedican un tiempo considerable a formalizar enunciados (o argumentos) del castellano ordinario al lenguaje simple de Proposicional. Creemos que esto es un error [ ]. Queremos investigar cómo de bien la lógica proposicional da cuenta del castellano habitual; es decir, cómo de bien *and*, *or* e *implica* simulan las operaciones habituales del castellano expresadas por *y*, *o* y *si ... entonces*. [ ] La interpretación mediante tablas de verdad de las conectivas *and*, *or* y *implica* es simple y precisa; los usos en castellano de *y*, *o* y *si ... entonces* parecen complicados e irregulares.

and e *y*

Algunos lógicos creen que hay un sentido especial, o significado de la palabra castellana *y* que conecta enunciados de una manera dependiente del tiempo. Desde este punto de vista, *Mel se aburrió y Mel se marchó de la fiesta* no es equivalente a (y por tanto no significa lo mismo que) *Mel se marchó de la fiesta y Mel se aburrió*

El filósofo Paul Grice sugirió [ ] distinguir entre (1) lo que significa una oración y (2) la información que una audiencia típica puede obtener de un uso típico de esa oración.

or y *o*

La principal preocupación respecto a la traducción del castellano *o* por la conectiva *or* tiene que ver con la primera línea de la tabla de verdad. Mucha gente cree que P *o* Q debería ser falso cuando ambos, P y Q son verdaderos. De hecho existe toda una respetable función lógica llamada *o exclusivo* [ ]. (ambas están relacionadas) $A \oplus B \equiv (A \vee B) \wedge \neg(A \wedge B)$; $A \vee B \equiv (A \oplus B) \oplus (A \wedge B)$

El mejor argumento para *or* es la simplicidad del esquema resultante. Consideremos lo siguiente: *Ni P ni Q* es la negación castellana de *o bien P, o bien Q*. Ahora, *Ni P ni Q* es equivalente en castellano a *No P y no Q*. Además, $\neg(P \vee Q)$ es lógicamente equivalente a $\neg P \wedge \neg Q$, pero $\neg(P \oplus Q)$ no lo es.

Ejemplo: *O limpias la habitación o no sales a jugar*, le dice papá Homer a su hijo Bart. Significa eso que Homer habría dicho una falsedad si Bart limpia su habitación pero no sale a jugar? Qué ocurriría se viene un huracán entre tanto, o si Bart rompe la cama mientras limpia?

implica y *Si ... entonces*

Un argumento simple (de Samuel Gutteplan) es éste: Los hablantes del castellano se inclinan a aceptar que Si $A \wedge B$, entonces A. Pero entonces se deduce que *Si Jim y Tom son profesores, entonces Jim es un* *profesor* (esto debe ser verdadero, por tanto *si V entonces V* es V) y *Si Jim es un profesor y Tom es un pingüino, entonces Jim es un profesor* (esto debe ser verdadero, por tanto *si F entonces V* es V) y *Si Jim es un pingüino y Tom es un profesor, entonces Jim es un pingüino* (esto debe ser verdadero, por tanto *si F entonces F* es V). Y, por supuesto, sabemos que tenemos que interpretar *si V entonces F* como F. Esto justifica las cuatro líneas de que consta la tabla de verdad de la implicación.

5.3. Equivalencia Lógica

Definición 50. *Dadas dos fbfs α, β, se dice que son* **lógicamente equivalentes** *si bajo cualquier interpretación I se tiene que $I(\alpha) = I(\beta)$.*

Para determinar la equivalencia lógica de dos fórmulas podemos usar los dos métodos que hemos descrito en la sección anterior. Podemos también utilizar el siguiente resultado:

Sean α y β dos fbfs. Son equivalentes:

1. α y β son lógicamente equivalentes:
2. $\alpha \leftrightarrow \beta$ es una tautología.
3. $\alpha \leftrightarrow \neg\beta$ es una contradicción.
4. $\alpha \rightarrow \beta$ y $\beta \rightarrow \alpha$ son tautologías.

Observación: Si α es una fórmula, β_1 es una subfórmula de α y β_2 es una fórmula lógicamente a β_1, entonces la fórmula que resulta de sustituir en α una aparición de β_1 por β_2 es lógicamente equivalente a α.

Enumeramos a continuación una serie de equivalencias lógicas que es conveniente memorizar.

$\alpha \equiv \neg\neg\alpha$
$\alpha \rightarrow \beta \equiv \neg\alpha \vee \beta$
$\alpha \leftrightarrow \beta \equiv (\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$
$\neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta$
$\neg(\alpha \vee \beta) \equiv \neg\alpha \wedge \neg\beta$
$\alpha \vee (\beta \vee \gamma) \equiv (\alpha \vee \beta) \vee \gamma$
$\alpha \wedge (\beta \wedge \gamma) \equiv (\alpha \wedge \beta) \wedge \gamma$
$\alpha \vee (\beta \wedge \gamma) \equiv (\alpha \vee \beta) \wedge (\alpha \wedge \gamma)$
$\alpha \wedge (\beta \vee \gamma) \equiv (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$

Cuadro 5.3: Equivalencias

Pregunta: ¿Pueden ser lógicamente equivalentes dos fórmulas en las que aparezcan conjuntos distintos de proposiciones atómicas?

5.4. Consecuencia lógica

Dado un conjunto de fórmulas Γ, y una fórmula α, decimos que α es consecuencia lógica de Γ si para cualquier interpretación que haga ciertas simultáneamente todas las proposiciones de Γ se tiene que la fórmula α es también cierta.

En tal caso, escribiremos:

$$\Gamma \models \alpha$$

y leeremos

$$\alpha \text{ es consecuencia lógica de } \Gamma$$

o también

$$\Gamma \text{ implica semánticamente } \alpha.$$

A las fórmulas del conjunto Γ se les llama **premisas** mientras que α se llama **conclusión**. La expresión $\Gamma \models \alpha$ corresponde un razonamiento lógicamente correcto.

Observación:

Si $\Gamma = \{\gamma_1, \gamma_2, \cdots, \gamma_n\}$ y $\Gamma \models \alpha$, escribiremos

$\gamma_1, \gamma_2, \cdots, \gamma_n \models \alpha$.

Para ver si $\gamma_1, \gamma_2, \cdots, \gamma_n \models \alpha$ tenemos que comprobar que ocurre **sólo** en las interpretaciones en que las premisas sean ciertas. Si para esas interpretaciones la conclusión (α) es también cierta, entonces la implicación semántica será cierta.

Cuando el conjunto Γ sea vacío, la expresión $\Gamma \models \alpha$ significa que α es una tautología. En tal caso, escribiremos $\models \alpha$.

Teorema 5.4.1. . *Sea Γ un conjunto de fórmulas y α otra fórmula. Son equivalentes:*

1. $\Gamma \models \alpha$
2. $\Gamma \cup \{\neg\alpha\}$ *es insatisfacible.*
3. *Para cualquier interpretación I, se tiene que* $[\prod_{\gamma\in\Gamma} I(\gamma)](1 + I(\alpha)) = 0$

Demostración: La equivalencia entre 1 y 2 puede verse como sigue:

Supongamos que $\Gamma \models \alpha$. Vamos a comprobar que no hay ninguna interpretación que haga simultáneamente ciertas a las fórmulas $\gamma_1, \gamma_2, \cdots, \gamma_n, \neg\alpha$.

De existir esa interpretación, tendríamos que $I(\gamma_1) = I(\gamma_2) = \cdots I(\gamma_n) = 1$, y por ser α consecuencia lógica de Γ tendríamos que $I(\alpha) = 1$, luego $I(\neg\alpha) = 0$. Luego no pueden ser ciertas simultáneamente las fórmulas de $\Gamma \cup \neg\{\alpha\}$.

Si ahora tenemos que $\Gamma \cup \{\neg\alpha\}$ es insatisfacible e I es una interpretación que hace ciertas todas las premisas, entonces no puede hacer cierta a $\neg\alpha$, luego $I(\neg\alpha) = 0$ y por tanto $I(\alpha) = 1$.

La equivalencia entre 2 y 3 es inmediata a partir de la proposicion 5.2.1. ■

Ejemplo 5.4.1. *Probaremos que* $\{p \vee q \to r, \neg r\} \models \neg q$

Método 1: Tablas de verdad

Necesitamos construir la tabla de verdad de todas las fórmulas que aparecen

	p	q	r	$p \vee q$	$p \vee q \to r$	$\neg r$	$\neg q$
I_1	0	0	0	0	1	1	1
I_2	0	0	1	0	1	0	1
I_3	0	1	0	1	0	1	0
I_4	0	1	1	1	1	0	0
I_5	1	0	0	1	0	1	1
I_6	1	0	1	1	1	0	1
I_7	1	1	0	1	0	1	0
I_8	1	1	1	1	1	0	0

Ahora sobre la tabla de verdad debemos fijarnos en las filas en las que el valor de TODAS las fórmulas del conjunto inicial $\{p \vee q \to r, \neg r\}$ sea 1 SIMULTÁNEAMENTE: en este caso sólo I_1, es decir la primera línea de la tabla de verdad.

Para esta interpretación, el valor de verdad de la fórmula $\neg q$ es también 1. Por tanto, la implicación es cierta.

Notemos que lo que ocurre en el resto de las filas no nos dice nada acerca de si la implicación es o no cierta.

Si construimos la tabla de verdad de las fórmulas del conjunto $\{p \vee q \to r,\ \neg r, \neg\neg q \equiv q\}$ nos resulta:

y vemos como el conjunto $\{p \vee q \to r,\ \neg r, \neg\neg q \equiv q\}$ es insatisfacible.

Método 2: Ecuaciones en $\mathbb{Z}_2$ *En este caso el proceso consiste en la manipulación de varias ecuaciones en $\mathbb{Z}_2$*

	p	q	r	$p \vee q$	$p \vee q \to r$	$\neg r$	q
I_1	0	0	0	0	1	1	0
I_2	0	0	1	0	1	0	0
I_3	0	1	0	1	0	1	1
I_4	0	1	1	1	1	0	1
I_5	1	0	0	1	0	1	0
I_6	1	0	1	1	1	0	0
I_7	1	1	0	1	0	1	1
I_8	1	1	1	1	1	0	1

Si I es una interpretación bajo la cual

$$\left.\begin{array}{l} I(p \vee q \to r) = 1 \\ I(\neg r) = 1 \end{array}\right\}$$

desarrollando ambas ecuaciones

$$\left.\begin{array}{l} 1 + I(p \vee q) + I(p \vee q)I(r) = 1 \\ 1 + I(r) = 1 \end{array}\right\} \qquad \left.\begin{array}{l} I(p \vee q) + I(p \vee q)I(r) = 0 \\ I(r) = 0 \end{array}\right\}$$

sustituyendo el valor $I(r)$ en la primera ecuación

$$\left.\begin{array}{l} I(p \vee q) = 0 \\ I(r) = 0 \end{array}\right\}$$

y de aquí se obtiene que I es:

$$\left.\begin{array}{l} I(p) = I(q) = 0 \\ I(r) = 0 \end{array}\right\}$$

y bajo ella $I(\neg q) = 1$

En este proceso hemos partido de las ecuaciones que resultan de imponer que las premisas son ciertas, y hemos operado hasta obtener como resultado que $I(\neg q) = 1$, es decir, que entonces también lo es la conclusión).

Nota: Si Γ es un conjunto insatisfacible, y α es una fórmula cualquiera, entonces el conjunto $\Gamma \cup \{\neg\alpha\}$ es insatisfacible, luego $\Gamma \models \alpha$. Es decir, *a partir de algo falso se puede deducir cualquier cosa.*

PASATIEMPO:
Se conocen los siguientes hechos sobre cuatro personas A, H, C, O:

1. Si A quiere ver una película, entonces H también quiere verla.
2. C y O no ven una película juntos.
3. Respecto de H y C, o bien ven la película juntos o bien ninguno de los dos la ve.
4. Si A decide no ir a ver la película, entonces H y C querrán verla.

¿Quiénes estarán viendo la película?

SOLUCIÓN:
Llamaremos:
a = A está viendo la película h = H está viendo la película c = C está viendo la película o = O está viendo la película
Traducimos los hechos que se suponen ciertos a la lógica proposicional:

1. $a \to h$
2. $c \to \neg o$
3. $(h \wedge c) \vee (\neg h \wedge \neg c)$
4. $\neg a \to h \wedge c$

El problema se traduce en calcular $I(a), I(h), I(c), I(o)$ a partir de las condiciones

1. $I(a \to h) = 1$
2. $I(c \to \neg o) = 1$
3. $I((h \wedge c) \vee (\neg h \wedge \neg c)) = 1$
4. $I(\neg a \to h \wedge c) = 1$.

5.5. Teorema de La Deducción

Una de las herramientas más útiles en los problemas de consecuencia lógica es el siguiente resultado, conocido como Teorema de la Deducción, y que reencontraremos en la segunda parte de la asignatura.

Teorema 5.5.1. *de la DeducciónDado un conjunto de fórmulas Γ, y dos fórmulas más α, β, las siguientes afirmaciones son equivalentes:*

1. $\Gamma \models \alpha \to \beta$

2. $\Gamma \cup \{\alpha\} \models \beta$

Demostración: Usando el resultado que ya conocemos la primera afirmación se traduce en que

$$[\prod_{\gamma \in \Gamma} I(\gamma)](1 + I(\alpha \to \beta)) = 0$$

si desarrollamos el segundo paréntesis

$$(1 + I(\alpha \to \beta) = 1 + 1 + I(\alpha) + I(\alpha)I(\beta) = I(\alpha)(1 + I(\beta))$$

con lo que queda

$$[\prod_{\gamma \in \Gamma} I(\gamma)]I(\alpha)(1 + I(\beta)) = 0$$

que también se puede escribir agrupando el producto de los términos en Γ con α

$$[\prod_{\gamma\in\Gamma\cup\{\alpha\}} I(\gamma)](1+I(\beta))=0$$

que es la traducción a interpretaciones de la segunda afirmación.

■

Ejercicio: Usando el Teorema de la Deducción prueba que las siguientes proposiciones son tautologías:

1. $\alpha \to (\beta \to \alpha)$ (Ley de afortiori)
2. $(\alpha \to (\beta \to \gamma)) \to ((\alpha \to \beta) \to (\alpha \to \gamma))$ (autodistributiva de la implicación)
3. $(\neg\alpha \to \neg\beta) \to ((\neg\alpha \to \beta) \to \alpha)$ (ley clásica de reducción al absurdo)

Ejercicio del examen de Septiembre de 2005:
Prueba:

$$\models \Big(\big(((\varphi \to \psi) \to (\neg\chi \to \neg\theta)) \to \chi\big) \to \tau\Big) \to \big((\tau \to \varphi) \to (\theta \to \varphi)\big)$$

Una solución:

En primer lugar transformamos el problema en uno más sencillo usando 3 veces el Teorema de la Deducción:

$$\{\big(((\varphi \to \psi) \to (\neg\chi \to \neg\theta)) \to \chi\big) \to \tau, \tau \to \varphi, \theta\} \models \varphi$$

Usaremos la caracterización de consecuencia lógica y el cálculo de interpretaciones en la siguiente forma: si I es una interpretación para la que $I(\varphi)=0$ pero $I(\theta)=1$ y $I(\tau\to\varphi)$=1 entonces probaremos que

$$I(\big(((\varphi \to \psi) \to (\neg\chi \to \neg\theta)) \to \chi\big) \to \tau)=0$$

Para simplificar la notación llamaremos

$$\delta=(\varphi\to\psi)\to(\neg\chi\to\neg\theta)$$

$$\alpha=\delta\to\chi$$

Nuestras ecuaciones de partida son:

$$\begin{cases} I(\varphi)=0 \\ I(\tau\to\varphi)=1 \\ I(\theta)=1 \end{cases}$$

usando la primera y la segunda transformamos el sistema en

$$\begin{cases} I(\varphi)=0 \\ I(\tau)=0 \\ I(\theta)=1 \end{cases}$$

y ahora vamos a calcular

$$I(\neg\chi\to\neg\theta)=1+I(\neg\chi)+I(\neg\chi)I(\neg\theta)$$

como $I(\theta)=1$ entonces $I(\neg\theta)=0$ y se tiene

$$I(\neg\chi\to\neg\theta)=I(\chi)$$

y como $I(\phi)=0$ entonces

$$I(\varphi\to\psi)=1+I(\varphi)+I(\varphi)I(\psi)=1+0+0=1$$

que podemos sustituir en

$$I(\delta) = 1 + 1 + I(\chi) = I(\chi)$$

Por último calculamos

$$I(\alpha) = 1 + I(\delta) + I(\delta)I(\chi) = 1 + I(\chi) + I(\chi)^2 = 1$$

Usando este resultado y que $I(\tau) = 0$ nos queda que la primera premisa bajo esta interpretación es necesariamente falsa, es decir,

$$I(\alpha \to \tau) = 0.$$

Un intento de resolver este ejercicio usando tablas de verdad es tedioso (como son 5 proposiciones básicas la tabla de verdad tiene $2^5 = 32$ filas y hay también demasiadas subfórmulas que evaluar) así que por su extensión tiene una alta probabilidad de error.

Algunas equivalencias referentes al problema de la implicación semántica:
Hemos visto el teorema de la deducción, que nos dice que los siguientes enunciados son equivalentes:

- $\Gamma \models \alpha \to \beta$.
- $\Gamma \cup \alpha \models \beta$.

Con una forma similar a este teorema, podemos también recoger los siguientes resultados:

1. Son equivalentes:
   - *a*) $\Gamma \models \alpha \to \beta$.
   - *b*) $\Gamma \cup \{\neg\beta\} \models \neg\alpha$.
   - *c*) $\Gamma \cup \{\alpha, \neg\beta\}$ es insatisfacible.
2. Son equivalentes:
   - *a*) $\Gamma \models \alpha \vee \beta$.
   - *b*) $\Gamma \cup \{\neg\alpha\} \models \beta$.
   - *c*) $\Gamma \cup \{\neg\beta\} \models \alpha$.
   - *d*) $\Gamma \cup \{\neg\alpha, \neg\beta\}$ es insatisfacible.
3. Son equivalentes:
   - *a*) $\Gamma \models \alpha \wedge \beta$.
   - *b*) $\Gamma \models \alpha$ y $\Gamma \models \beta$.
   - *c*) $\Gamma \cup \{\neg\alpha\}$ y $\Gamma \cup \{\neg\beta\}$ son insatisfacibles.
4. Son equivalentes:
   - *a*) $\Gamma \models \alpha \leftrightarrow \beta$.
   - *b*) $\Gamma \models \alpha \to \beta$ y $\Gamma \models \beta \to \alpha$.
   - *c*) $\Gamma \cup \{\alpha\} \models \beta$ y $\Gamma \cup \{\beta\} \models \alpha$.
   - *d*) $\Gamma \cup \{\alpha, \neg\beta\}$ y $\Gamma \cup \{\neg\alpha, \beta\}$ son insatisfacibles.

5.6. Forma clausular de una fórmula.

5.6.1. Definición de cláusula.

Dado un lenguaje proposicional, un **literal** es una proposición atómica o su negada. Por ejemplo, son literales a, $\neg c$, b. No es un literal $\neg\neg b$ (aunque sea lógicamente equivalente a b, que sí es un literal).

Si λ es un literal, denotaremos como λ^c al literal que es o bien el negado de λ o lógicamente equivalente con él. Así, si $\lambda = a$ entonces $\lambda^c = \neg a$, mientras que si $\lambda = \neg a$, entonces $\lambda^c = a$ (que es equivalente a $\neg\neg a$).

Una **cláusula** es una **disyunción de literales**, de forma que no haya dos literales que provengan de la misma proposición atómica. Es decir, si aparece el literal λ, éste sólo lo puede hacer una vez, y no puede estar el literal λ^c.

Observación: Dada una proposición atómica a, la disyunción $a \vee \neg a$ es una tautología, por tanto lo será cualquier fórmula de la forma $a \vee \neg a \vee \beta$ donde β representa a cualquier fórmula bien formada.

◇ Todo literal es una cláusula.

◇ También se considera como cláusula a la "disyunción de cero literales". Dicha cláusula se denomina como cláusula vacía, la denotamos por $\square$.

◇ Son ejemplos de cláusulas: a, $\neg b \vee c$, $a \vee \neg b \vee \neg c$, $\neg b \vee a \vee c$.

◇ No son cláusulas: $a \vee a$, $a \vee b \vee \neg a$, $a \wedge b$.

5.6.2. Forma clausular de una fórmula.

Una fórmula se dice que está en **forma clausular** si es una cláusula o está expresada como **conjunción de cláusulas**.

Son fórmulas en forma clausular las siguientes:

$$(a \vee \neg b) \wedge (b \vee c)$$

$$b \vee c$$

$$a \wedge b$$

$$b \wedge \neg b$$

$$(b \vee a \vee \neg c) \wedge (b \vee \neg a \vee \neg c)$$

La primera fórmula es la conjunción de dos cláusulas: $a \vee \neg b$ y $b \vee c$.
La segunda es una cláusula.
La tercera es conjunción de las cláusulas a y b.
La cuarta, conjunción de las cláusulas b y $\neg b$.
La quinta es conjunción de las cláusulas $b \vee a \vee \neg c$ y $b \vee \neg a \vee \neg c$.

No están en forma clausular las siguientes fórmulas:

$$\neg(a \wedge b)$$

$$(a \wedge \neg b) \vee (\neg a \vee c)$$

Teorema 5.6.1. *Toda fórmula (que no sea tautología) puede transformarse en una fórmula lógicamente equivalente con ella que esté en forma clausular.*

Así, la fórmula $\neg(a \wedge b)$ es equivalente a $\neg a \vee \neg b$ que está en forma clausular, mientras que la fórmula $(a \wedge \neg b) \vee (\neg a \vee c)$ podemos transformarla en

$$(a \vee \neg a \vee c) \wedge (\neg b \vee \neg a \vee c)$$

Ahora observamos que el primer paréntesis corresponde a una tautología y por tanto puede eliminarse ese factor de la conjunción y se obtiene una fórmula equivalente, en este caso:

$$\neg b \vee \neg a \vee c$$

que está en forma clausular.

5.6.3. Un método para calcular la forma clausular.

A continuación veremos qué pasos podemos dar para transformar una fórmula en otra equivalente a ella que esté en forma clausular:

1. Si tenemos una subfórmula de la forma $\alpha \leftrightarrow \beta$, la sustituimos por $(\alpha \to \beta) \wedge (\beta \to \alpha)$.
2. Si tenemos una subfórmula de la forma $\alpha \to \beta$ la sustituimos por $\neg\alpha \vee \beta$.
3. Cualquier subfórmula de la forma $\neg\neg\alpha$ la sustituimos por α.
4. Introducimos la conectiva $\neg$ dentro de los paréntesis utilizando las equivalencias lógicas:

$$\neg(\alpha \vee \beta) \equiv \neg\alpha \wedge \neg\beta$$

$$\neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta$$

.

Si al hacer esto nos apareciera una nueva subfórmula de la forma $\neg\neg\alpha$, la sustituimos por α.

Una vez ejecutados estos pasos, nos encontramos con una fórmula en la que sólo intervienen las conectivas $\vee$, $\wedge$ y $\neg$. Además, ésta última únicamente actúa sobre proposiciones atómicas. Ahora puede ser necesario el uso de la distributividad de las conectivas $\vee$ y $\wedge$, es decir el uso de las siguientes equivalencias lógicas:

1. $(a \wedge b) \vee c \equiv (a \vee c) \wedge (b \vee c)$
2. $a \vee (b \wedge c) \equiv (a \vee b) \wedge (a \vee c)$

Pensemos que el objetivo para encontrar la forma clausular es encontrar una fórmula en donde dentro de los paréntesis no puede haber ninguna conectiva $\wedge$. Se trata entonces de que, siempre que encontremos una subfórmula de la forma $(\alpha) \vee (\beta)$, proceder como sigue según los casos:

- Si α es de la forma $\alpha_1 \wedge \alpha_2$ (daría igual si fuera conjunción de tres o más fórmulas), entonces

$$\alpha \vee \beta \;\equiv\; (\alpha_1 \vee \beta) \wedge (\alpha_2 \vee \beta)$$

y ahora estudiaríamos por separado las fórmulas $\alpha_1 \vee \beta$ y $\alpha_2 \vee \beta$.

- Si β es de la forma $\beta_1 \wedge \beta_2$, sustituimos $\alpha \vee \beta$ por

$$\alpha \vee \beta \; = \alpha \vee (\beta_1 \wedge \beta_2) \;\equiv\; (\alpha \vee \beta_1) \wedge (\alpha \vee \beta_2)$$

y estudiaríamos ahora las fórmulas $\alpha \vee \beta_1$ y $\alpha \vee \beta_2$.

Por ejemplo, si tenemos una fórmula de la forma $\delta = \alpha \vee \beta$, con $\alpha = \alpha_1 \wedge \alpha_2 \wedge \alpha_3$ y $\beta = \beta_1 \wedge \beta_2$, entonces aplicando varias veces las reglas anteriores tendríamos:

$$\delta \equiv (\alpha_1 \vee \beta_1) \wedge (\alpha_2 \vee \beta_1) \wedge (\alpha_3 \vee \beta_1) \wedge (\alpha_1 \vee \beta_2) \wedge (\alpha_2 \vee \beta_2) \wedge (\alpha_3 \vee \beta_2)$$

Para actuar de manera ordenada es conveniente, bien empezar por las subfórmulas más pequeñas, e ir yendo progresivamente hacia subfórmulas más grandes, bien empezar por las subfórmulas más grandes y seguir el camino inverso al anterior. Veamos aquí un ejemplo, empezando por las subfórmulas pequeñas.

Ejemplo 5.6.1. *Queremos hallar la forma clausular de*

$$\big[\big(a \vee ((b \wedge c) \vee (d \wedge \neg e))\big) \wedge (\neg b \vee c)\big] \vee \big[(a \wedge \neg c) \wedge (c \vee \neg b \vee d)\big]$$

Notemos que no tenemos ninguna conectiva $\rightarrow$, ninguna conectiva $\leftrightarrow$, y las conectivas $\neg$ actúa únicamente sobre fórmulas atómicas. Entonces, hemos de aplicar sólo la propiedad distributiva.

Buscamos las subfórmulas más pequeñas donde nos encontremos la conectiva $\wedge$ dentro de un paréntesis, y vemos que esto ocurre en $(b{\wedge}c)$, $(d{\wedge}\neg e)$ y $(a{\wedge}\neg c)$, si bien, este último podría eliminarse por la propiedad asociativa de la conectiva $\wedge$.

Sustituimos entonces $(b \wedge c) \vee (d \wedge \neg e)$ por

$$(b \vee d) \wedge (b \vee \neg e) \wedge (c \vee d) \wedge (c \vee \neg e)$$

y nos queda

$$\big[\big(a \vee ((b \vee d) \wedge (b \vee \neg e) \wedge (c \vee d) \wedge (c \vee \neg e))\big) \wedge (\neg b \vee c)\big] \vee \big[a \wedge \neg c \wedge (c \vee \neg b \vee d)\big]$$

La siguiente sufórmula donde encontramos la conectiva $\wedge$ dentro de paréntesis es

$$\beta = (b \vee d) \wedge (b \vee \neg e) \wedge (c \vee d) \wedge (c \vee \neg e)$$

Sustituimos entonces $a \vee \beta$ por $(a \vee \beta_1) \wedge (a \vee \beta_2) \wedge (a \vee \beta_3) \wedge (a \vee \beta_4)$ y nos queda

$$\big[\big((a \vee b \vee d) \wedge (a \vee b \vee \neg e) \wedge (a \vee c \vee d) \wedge (a \vee c \vee \neg e)\big) \wedge (\neg b \vee c)\big] \vee \big[a \wedge \neg c \wedge (c \vee \neg b \vee d)\big]$$

que por la asociatividad de $\wedge$ podemos cambiar por

$$\big((a \vee b \vee d) \wedge (a \vee b \vee \neg e) \wedge (a \vee c \vee d) \wedge (a \vee c \vee \neg e) \wedge (\neg b \vee c)\big) \vee \big(a \wedge \neg c \wedge (c \vee \neg b \vee d)\big)$$

Y nos encontramos ahora con una fórmula que responde a la forma

$$\alpha \vee \beta = (\alpha_1 \wedge \alpha_2 \wedge \alpha_3 \wedge \alpha_4 \wedge \alpha_5) \vee (\beta_1 \wedge \beta_2 \wedge \beta_3)$$

luego la fórmula de partida es equivalente a:

$$\begin{gathered}
(a \vee b \vee d \vee a) \wedge (a \vee b \vee \neg e \vee a) \wedge (a \vee c \vee d \vee a) \wedge (a \vee c \vee \neg e \vee a) \wedge (\neg b \vee c \vee a) \wedge \\
\wedge (a \vee b \vee d \vee \neg c) \wedge (a \vee b \vee \neg e \vee \neg c) \wedge (a \vee c \vee d \vee \neg c) \wedge (a \vee c \vee \neg e \vee \neg c) \wedge (\neg b \vee c \vee \neg c) \wedge \\
\wedge (a \vee b \vee d \vee c \vee \neg b \vee d) \wedge (a \vee b \vee \neg e \vee c \vee \neg b \vee d) \wedge (a \vee c \vee d \vee c \vee \neg b \vee d) \wedge \\
\wedge (a \vee c \vee \neg e \vee c \vee \neg b \vee d) \wedge (\neg b \vee c \vee c \vee \neg b \vee d)
\end{gathered}$$

Y finalmente, para obtener la forma clausular, eliminamos aquellos términos en los que aparezca $\lambda \vee \lambda^c$, puesto que son tautologías, y los literales repetidos haciendo uso de la equivalencia $a \vee a \equiv a$:

$$\begin{gathered}
(a \vee b \vee d) \wedge (a \vee b \vee \neg e) \wedge (a \vee c \vee d) \wedge (a \vee c \vee \neg e) \wedge (a \vee \neg b \vee c) \wedge (a \vee b \vee \neg c \vee d) \wedge \\
\wedge (a \vee b \vee \neg c \vee \neg e) \wedge (a \vee \neg b \vee c \vee d) \wedge (a \vee \neg b \vee c \vee d \vee \neg e) \wedge (\neg b \vee c \vee d)
\end{gathered}$$

¡Esta ya es una forma clausular! *Ahora bien,* **podemos reducirla** *observando la siguiente equivalencia lógica:*

$$(\alpha \vee \beta) \wedge \alpha \equiv \alpha$$

por lo que

$$(a \vee \neg b \vee c) \wedge (a \vee \neg b \vee c \vee d) \wedge (a \vee \neg b \vee c \vee d \vee \neg e) \ \equiv a \vee \neg b \vee c$$

y también

$$(a \vee b \vee d) \wedge (a \vee b \vee \neg c \vee d) \equiv a \vee b \vee d$$

y por tanto, otra forma clausular es:

$$(a \vee b \vee d) \wedge (a \vee b \vee \neg e) \wedge (a \vee c \vee d) \wedge (a \vee c \vee \neg e) \wedge (a \vee \neg b \vee c) \wedge$$

$$\wedge (a \vee b \vee \neg c \vee \neg e) \wedge (\neg b \vee c \vee d)$$

El seguimiento de las etapas aquí descritas nos asegura la consecución de la forma clausular de una fórmula. **No es sin embargo la única opción para obtener la forma clausular de una fórmula**. **La práctica** nos indicará cual es el mejor camino para obtener la forma clausular.

Veamos a continuación otros ejemplos completos de cómo obtener la forma clausular de una fórmula.

Ejemplo 5.6.2. *Partimos de la fórmula*

$$((a \vee \neg b) \rightarrow (\neg c \rightarrow b)) \wedge ((a \rightarrow b) \leftrightarrow c)$$

El primer paso consiste en sustituir la conectiva $\leftrightarrow$. Nos queda entonces:

$$\big((a \vee \neg b) \rightarrow (\neg c \rightarrow b)\big) \wedge \big[\big((a \rightarrow b) \rightarrow c\big) \wedge \big(c \rightarrow (a \rightarrow b)\big)\big]$$

El siguiente paso consiste en sustituir las conectivas $\rightarrow$. El orden en que se haga esto no es importante. Una forma de hacerlo podría ser:

$$\big(\neg(a \vee \neg b) \vee (\neg c \rightarrow b)\big) \wedge \big[\big((\neg a \vee b) \rightarrow c\big) \wedge \big(c \rightarrow (\neg a \vee b)\big)\big]$$

$$\big(\neg(a \vee \neg b) \vee (\neg\neg c \vee b)\big) \wedge \big[\big(\neg(\neg a \vee b) \vee c\big) \wedge \big(\neg c \vee (\neg a \vee b)\big)\big]$$

Eliminamos la doble negación

$$\big(\neg(a \vee \neg b) \vee (c \vee b)\big) \wedge \big[\big(\neg(\neg a \vee b) \vee c\big) \wedge \big(\neg c \vee (\neg a \vee b)\big)\big]$$

Introducimos las conectivas $\neg$ en los paréntesis.

$$\big((\neg a \wedge \neg\neg b) \vee (c \vee b)\big) \wedge \big[\big((\neg\neg a \wedge \neg b) \vee c\big) \wedge \big(\neg c \vee (\neg a \vee b)\big)\big]$$

Y volvemos a eliminar la doble negación que ha aparecido:

$$\big((\neg a \wedge b) \vee (c \vee b)\big) \wedge \big[\big((a \wedge \neg b) \vee c\big) \wedge \big(\neg c \vee (\neg a \vee b)\big)\big]$$

Como podemos ver, únicamente tenemos las conectivas $\vee$, $\wedge$ y $\neg$, y la conectiva $\neg$ afecta únicamente a proposiciones atómicas.

Nuestra fórmula es ahora conjunción de tres subfórmulas:
$(\neg a \wedge b) \vee (c \vee b)$, $(a \wedge \neg b) \vee c$ *y* $\neg c \vee (\neg a \vee b)$
Hallamos la forma clausular de cada una de estas subfórmulas:

$$(\neg a \wedge b) \vee c \vee b \equiv (\neg a \vee b \vee c) \wedge (b \vee c \vee b) \equiv (\neg a \vee b \vee c) \wedge (b \vee c)$$

$$(a \wedge \neg b) \vee c \equiv (a \vee c) \wedge (\neg b \vee c)$$

$$\neg c \vee (\neg a \vee b) \equiv \neg a \vee b \vee \neg c$$

Por tanto, una forma clausular de la fórmula de partida es:

$$(\neg a \vee b \vee c) \wedge (b \vee c) \wedge (a \vee c) \wedge (\neg b \vee c) \wedge (\neg a \vee b \vee \neg c)$$

Aunque también podemos simplificarla utilizando la siguiente equivalencia lógica:

$$(\alpha \vee \beta) \wedge (\alpha \vee \neg\beta) \equiv \alpha$$

aplicada a

$$(\neg a \vee b \vee c) \wedge (\neg a \vee b \vee \neg c) \equiv \neg a \vee b$$

y nos quedará:

$$(\neg a \vee b) \wedge (b \vee c) \wedge (a \vee c) \wedge (\neg b \vee c)$$

Ejemplo 5.6.3. *Vamos a calcular la forma clausular de:*

$$\alpha = \big[\big((a \to b) \wedge \neg c\big) \to \big(\neg b \leftrightarrow (c \vee a)\big)\big] \vee \big[d \vee \big(\neg c \to (a \wedge b)\big)\big]$$

En este caso vamos a seguir un orden diferente al del ejemplo anterior.
La fórmula dada es equivalente a

$$\big[\neg\big((a \to b) \wedge \neg c\big) \vee \big(\neg b \leftrightarrow (c \vee a)\big)\big] \vee \big[d \vee \big(\neg c \to (a \wedge b)\big)\big]$$

que está expresada como disyunción de tres fórmulas:

$$\neg\big((a \to b) \wedge \neg c\big) \qquad \neg b \leftrightarrow (c \vee a) \qquad d \vee \big(\neg c \to (a \wedge b)\big)$$

Calculamos la forma clausular de cada una de estas fórmulas por separado.

$\neg\big((a \to b) \wedge \neg c\big)$	$\neg b \leftrightarrow (c \vee a)$	$d \vee \big(\neg c \to (a \wedge b)\big)$
$\neg\big((\neg a \vee b) \wedge \neg c\big)$	$\big(\neg b \to (c \vee a)\big) \wedge \big((c \vee a) \to \neg b\big)$	$d \vee \big(\neg\neg c \vee (a \wedge b)\big)$
$\neg(\neg a \vee b) \vee \neg\neg c$	$\big(\neg\neg b \vee (c \vee a)\big) \wedge \big(\neg(c \vee a) \vee \neg b\big)$	$d \vee \big(c \vee (a \wedge b)\big)$
$(\neg\neg a \wedge \neg b) \vee \neg\neg c$	$\big(b \vee (c \vee a)\big) \wedge \big((\neg c \wedge \neg a) \vee \neg b\big)$	$d \vee \big((c \vee a) \wedge (c \vee b)\big)$
$(a \wedge \neg b) \vee c$	$(a \vee b \vee c) \wedge \big((\neg c \vee \neg b) \wedge (\neg a \vee \neg b)\big)$	$(d \vee c \vee a) \wedge (d \vee c \vee b)$
$(a \vee c) \wedge (\neg b \vee c)$	$(a \vee b \vee c) \wedge (\neg a \vee \neg b) \wedge (\neg b \vee \neg c)$	$(a \vee c \vee d) \wedge (b \vee c \vee d)$

Es decir, hemos llegado a que la fórmula inicial es equivalente a una de la forma

$$(C_1 \wedge C_2) \vee (C_1' \wedge C_2' \wedge C_3') \vee (C_1'' \wedge C_2'')$$

que es equivalente a

$$(C_1 \vee C_1' \vee C_1'') \wedge (C_1 \vee C_1' \vee C_2'') \wedge (C_1 \vee C_2' \vee C_1'') \wedge (C_1 \vee C_2' \vee C_2'') \wedge$$

$$\wedge (C_1 \vee C_3' \vee C_1'') \wedge (C_1 \vee C_3' \vee C_2'') \wedge$$

$$\wedge (C_2 \vee C_1' \vee C_1'') \wedge (C_2 \vee C_1' \vee C_2'') \wedge (C_2 \vee C_2' \vee C_1'') \wedge (C_2 \vee C_2' \vee C_2'') \wedge$$

$$\wedge (C_2 \vee C_3' \vee C_1'') \wedge (C_2 \vee C_3' \vee C_2'')$$

Y ahora $C_1 \vee C_1' \vee C_1' = (a \vee c) \vee (a \vee b \vee c) \vee (a \vee c \vee d) \equiv a \vee b \vee c \vee d$, *también* $C_1 \vee C_1' \vee C_2'' = a \vee b \vee c \vee d$ *mientras que el resto se pueden eliminar puesto que contienen al mismo tiempo un literal y su negación.*

Y así, podemos concluir que la fórmula α *es equivalente a*

$$a \vee b \vee c \vee d$$

5.7. El problema de la implicación semántica

Hemos definido en secciones precedentes lo que significa

$$\Gamma \models \alpha$$

que viene a decirnos cuando un razonamiento es correcto desde el punto de vista lógico. Vamos a desarrollar técnicas que nos permitan dar respuesta a este problema dentro de la lógica proposicional.

Ya conocemos una forma de resolver el problema. Hallamos las tablas de verdad de las fórmulas de Γ y de α, y si en todas las interpretaciones en las que las fórmulas de Γ sean ciertas también lo es α, la implicación es cierta.

Sin embargo, en lo que sigue vamos a basarnos en el teorema 5.4.1, que entre otras cosas decía que la respuesta a los problemas

$$\Gamma \models \alpha \qquad \text{y} \qquad \Gamma \cup \{\neg\alpha\}$$

es siempre la misma.

Es decir, el problema de implicación semántica podemos transformarlo en un problema de decidir si un conjunto de fórmulas es insatisfacible o no.

De momento, vamos a aparcar el tema de la implicación semática, y vamos a fijarnos únicamente en la insatisfacibilidad de un conjunto de fórmulas.

En la sección anterior aprendimos a calcular, dada una fórmula, otra fórmula lógicamente equivalente a ella y que está en forma clausular.

Entonces, a la hora de determinar si un conjunto de cláusulas es insatisfacible o no, podemos sustituir cada fórmula por otra que esté en forma clausular.

Teorema 5.7.1. *Sea Ω un conjunto de fórmulas; Ω' un conjunto de fórmulas que se obtiene sustituyendo cada fórmula por una forma clausular de esa fórmula, y Ω'' el conjunto que resulta de sustituir cada fórmula de Ω' por las cláusulas que la forman. Entonces son equivalentes:*

1. *Ω es insatisfacible,*
2. *Ω' es insatisfacible.*
3. *Ω'' es insatisfacible.*

Reuniendo ahora los resultados anteriores podemos observar cómo el problema tipo del que partíamos se transforma en el problema de estudiar si un conjunto de cláusulas es insatisfacible o no.

En resumen: el conjunto del que tenemos que estudiar la satisfacibilidad será el formado por las cláusulas que aparecen para el conjunto de fórmulas $\Gamma \cup \{\neg\alpha\}$.

5.8. Algoritmo de Davis-Putnam:

Vamos entonces a estudiar cuando un conjunto de cláusulas es o no satisfacible. Para esto emplearemos un algoritmo debido a Davis y Putnam.

Antes de describir el algoritmo, denotaremos, dada una claúsula C y un literal λ que forme parte de C, como $C - \lambda$ a la cláusula que resulta de eliminar el literal λ de C. Así, si $C = a \vee \neg b \vee d$ entonces $C - a = \neg b \vee d$, mientras que $C - \neg b = a \vee d$. Admitiremos ahora que hay una cláusula que no contiene ningún literal, la **cláusula vacía**, que denotaremos por $\square$ y que para cualquier interpretación verifica que $I(\square) = 0$; de esta forma si $C = b$ entonces $C - b = \square$, la cláusula vacía. Un conjunto en el que aparece la cláusula vacía es siempre insatisfacible.

El algoritmo de Davis-Putnam se basa en tres resultados, los cuales permiten ir reduciendo el conjunto de cláusulas. Estos resultados son:

1. Sea Σ un conjunto de cláusulas. Supongamos que tenemos una cláusula con un único literal λ (diremos que ésta cláusula es una **cláusula unit**). Sea Σ' el conjunto que resulta de suprimir todas las cláusulas que contengan el literal λ, y sustituir las cláusulas C que contengan a λ^c por $C - \lambda^c$.

   Es decir: si
$$\Sigma = \{\lambda,\ \lambda \vee C_1,\ \lambda \vee C_2,\ \cdots, \lambda \vee C_n,\ \lambda^c \vee C'_1,\ \lambda^c \vee C'_2,\ \cdots, \lambda^c \vee C'_m,$$
$$C''_1,\ C''_2,\ \cdots, C''_p\}$$
   entonces
$$\Sigma' = \{C'_1,\ C'_2,\ \cdots, C'_m, C''_1,\ C''_2,\ \cdots, C''_p\}$$
   En tal caso, se verifica que Σ es satisfacible si, y sólo si, Σ' lo es.

   La justificación de este resultado podría ser como sigue:

   Supongamos en primer lugar que $\lambda = a$ o $\lambda = \neg a$.

   Si I es una interpretación para la que se satisfacen todas las cláusulas de Σ, entonces $I(\lambda) = 1$, lo que implica que $I(\lambda^c) = 0$. Puesto que $I(\lambda^c \vee C'_i) = 1$ deducimos que $I(C'_i) = 1$. Por tanto, I es una interpretación para la que todas las cláusulas de Σ' son ciertas.

   Recíprocamente, si I es una interpretación para la que todas las cláusulas de Σ' son ciertas, definimos I' como la interpretación que actúa igual que I sobre todas las proposiciones atómicas salvo sobre a, mientras que $I'(a)$ lo definimos de forma que $I'(\lambda) = 1$. Es claro entonces que bajo la interpretación I' todas las cláusulas de Σ son ciertas.

2. Supongamos que en el conjunto Σ hay una cláusula en la que aparece un literal λ de forma que λ^c no aparece en ninguna cláusula (en este caso, diremos que el literal λ es un **literal puro**). Sea entonces Σ' el conjunto de cláusulas que resulta de suprimir todas las cláusulas en las que interviene el literal λ.

   Es decir, si
$$\Sigma = \{\lambda \vee C_1,\ \lambda \vee C_2,\ \cdots, \lambda \vee C_n,\ C'_1,\ C'_2,\ \cdots, C'_m\}$$
   tendríamos
$$\Sigma' = \{C'_1,\ C'_2,\ \cdots, C'_m\}$$
   Entonces Σ es satisfacible si, y sólo si, Σ' es satisfacible.

   Al igual que antes, suponemos que $\lambda = a$ o $\lambda = \neg a$.

   Claramente, si I es una interpretación que hace ciertas las cláusulas de Σ entonces I hace ciertas las cláusulas de Σ'.

   Recíprocamente, si I hace ciertas las cláusulas de Σ', definimos I' como la interpretación que actúa igual que I sobre todas las proposiciones atómicas salvo sobre a, mientras que $I'(a)$ lo definimos de forma que $I'(\lambda) = 1$. Es claro entonces que bajo la interpretación I' todas las cláusulas de Σ son ciertas.

3. Sea Σ un conjunto de cláusulas y a una proposición atómica. Supongamos que Σ es de la forma:
$$\Sigma = \{a \vee C_1,\ a \vee C_2,\ \cdots, a \vee C_n,\ \neg a \vee C'_1,\ \neg a \vee C'_2,\ \cdots, \neg a \vee C'_m,$$
$$C''_1,\ C''_2,\ \cdots, C''_p\}$$
   Sean
$$\Sigma_1 = \{C'_1,\ C'_2,\ \cdots, C'_m,\ C''_1,\ C''_2,\ \cdots, C''_p\}$$
$$\Sigma_2 = \{C_1,\ C_2,\ \cdots, C_n,\ C''_1,\ C''_2,\ \cdots, C''_p\}$$
   Entonces Σ es insatisfacible si, y sólo si, Σ_1 y Σ_2 son insatisfacibles, o dicho de otra forma, Σ es satisfacible si, y sólo si, Σ_1 o Σ_2 es satisfacible.

Supongamos que I es una interpretación que hace ciertas todas las cláusulas de Σ. Entonces:

- Si $I(a) = 1$ entonces $I(\neg a) = 0$, y como $I(\neg a \vee C'_i) = 1$ deducimos que $I(C'_i) = 1$. Por tanto, todas las cláusulas de Σ_1 son interpretadas como ciertas por I, luego Σ_1 es satisfacible.
- Si $I(a) = 0$ en este caso, lo que se tiene es que todas las cláusulas de Σ_2 son ciertas bajo I, luego es Σ_2 quien es satisfacible.

Recíprocamente, si es Σ_1 quien es satisfacible bajo una interpretación I, consideramos la interpretación I' que coincide con I en todas las proposiciones atómicas salvo eventualmente en a, sobre la que vale 1. Bajo esta interpretación se tiene que:
- $I'(a \vee C_i) = 1$, pues $I'(a) = 1$.
- $I'(\neg a \vee C'_i) = 1$, pues $I'(C'_i) = I(C'_i) = 1$.
- $I'(C''_i) = 1$, pues $I'(C''_i) = I(C''_i) = 1$.

Resumen:

El algoritmo de Davis-Putnam nos dice cuando un conjunto de cláusulas es satisfacible o no. Caso de ser satisfacible, nos da una interpretación que hace ciertas todas las cláusulas.

Consiste en partir de un conjunto de cláusulas, e ir reduciéndolo, hasta llegar a uno del que sepamos con claridad si es o no satisfacible.

Comprobamos si hay alguna cláusula unit. Si la respuesta es afirmativa, y λ es dicha cláusula, reducimos el conjunto de cláusulas tal y como hemos visto en 1., anotamos el literal λ y volvemos al principio con el nuevo conjunto de cláusulas.

Si la respuesta es negativa, comprobamos si existe un literal puro. De existir, procedemos como hemos dicho en 2., anotamos dicho literal y volvemos al inicio con el nuevo conjunto de cláusulas.

De no existir, elegimos una proposición atómica a. Abrimos dos ramas, tal y como hemos visto en 3. y analizamos cada una de las ramas. Al analizar la rama con Σ_1, anotamos el literal a, mientras que al hacerlo con Σ_2 anotamos el literal $\neg a$.

Después de que el conjunto de cláusulas haya sido modificado, se vuelve a comenzar el algoritmo, y si han aparecido varios conjuntos se aplica sobre cada uno de ellos.

El algoritmo acaba cuando podamos decidir fácilmente que el conjunto es satisfacible (por ejemplo cuando contenga una sola cláusula que no sea la vacía) o insatisfacible (por ejemplo cuando aparezca en él la cláusula vacía).

Si al final llegamos a que alguna rama es satisfacible, entonces una interpretación para la que todas las cláusulas son ciertas es aquella que vale 1 sobre todos los literales que hemos ido anotando.

Notemos que las reglas 1 y 2 son casos particulares de la regla 3.

En el caso de que tengamos una cláusula unit λ y aplicamos la regla 3 tomando como proposición atómica la que aparece en λ (es decir, λ ó λ^c) entonces uno de los dos conjuntos que nos resultan contiene la cláusula vacía, mientras que el otro es el que obtendríamos aplicando la regla 1.

En el caso de que tengamos un literal puro λ, es decir, tenemos un conjunto de cláusulas

$$\Sigma = \{\lambda \vee C_1,\ \lambda \vee C_2,\ \cdots,\lambda \vee C_n,\ C'_1,\ C'_2,\ \cdots,C'_m\}$$

La regla 3 (tomando como proposición atómica la que aparece en el literal λ) nos diría que Σ es satisfacible si, y sólo si, lo es uno de los dos conjuntos siguientes:

$$\Sigma_1 = \{C'_1,\ C'_2,\ \cdots,C'_m\}$$
$$\Sigma_2 = \{C_1,\ C_2,\ \cdots,\ C_n,\ C'_1,\ C'_2,\ \cdots,C'_m\}$$

pero como $\Sigma_1 \subseteq \Sigma_2$ entonces si Σ_2 es satisfacible, también lo es Σ_1. Por tanto, si uno de los dos conjuntos (Σ_1 ó Σ_2) es satisfacible, entonces Σ_1 lo es.

Ejemplo 5.8.1. *Vamos a demostrar que la fórmula*

$$(a \to (b \to c)) \to (\neg(a \to \neg b) \to c)$$

es una tautología. Esto sabemos que es equivalente a probar:

$$\models (a \to (b \to c)) \to (\neg(a \to \neg b) \to c)$$

por el teorema de la deducción esto se traduce en demostrar que

$$\{a \to (b \to c)\} \models \neg(a \to \neg b) \to c$$

y aplicando otra vez el teorema de la deducción nos queda el problema

$$\{a \to (b \to c), \neg(a \to \neg b)\} \models c$$

Lo que es equivalente a demostrar que el conjunto

$$\{a \to (b \to c), \neg(a \to \neg b), \neg c\}$$

es insatisfacible.

Hallamos la forma clausular de cada una de las fórmulas anteriores:

$a \to (b \to c)$	$\neg(a \to \neg b)$	$\neg c$
$\neg a \vee (b \to c)$	$\neg(\neg a \vee \neg b)$	
$\neg a \vee (\neg b \vee c)$	$\neg\neg a \wedge \neg\neg b$	
$\neg a \vee \neg b \vee c$	$a \wedge b$	

Y entonces lo que nos queda probar es que el siguiente conjunto de cláusulas es insatisfacible:

$$\{\neg a \vee \neg b \vee c,\ a,\ b,\ \neg c\}$$

Tenemos una cláusula unit a, por tanto, el conjunto nos queda

$$\{\neg b \vee c,\ b,\ \neg c\}$$

En este conjunto tenemos también una cláusula unit (b), por tanto, el conjunto de cláusulas se reduce a:

$$\{c, \neg c\}$$

que claramente es insatisfacible.

También, y puesto que en este conjunto tenemos una cláusula unit, podríamos reducir el conjunto y nos quedaría

$$\{\square\}$$

El proceso seguido al utilizar el algoritmo de Davis-Putnam lo podemos representar como sigue:

$$\begin{array}{c}
\{\neg a \vee \neg b \vee c,\ a,\ b,\ \neg c\} \\
\Big| \ \lambda = a \\
\{\neg b \vee c,\ b,\ \neg c\} \\
\Big| \ \lambda = b \\
\{c, \neg c\} \\
\Big| \ \lambda = c \\
\{\square\}
\end{array}$$

Ejemplo 5.8.2. *Vamos a demostrar que*

$$\models \big(\big(\big((\alpha\to\beta)\to(\neg\gamma\to\neg\delta)\big)\to\gamma\big)\to\varphi\big)\to\big((\varphi\to\alpha)\to(\delta\to\alpha)\big)$$

Por el teorema de la deducción, lo que hemos de demostrar es que:

$$\big(\big(\big((\alpha\to\beta)\to(\neg\gamma\to\neg\delta)\big)\to\gamma\big)\to\varphi\big)\models\big((\varphi\to\alpha)\to(\delta\to\alpha)\big)$$

que es equivalente, usando nuevamente el teorema de la deducción a:

$$\big\{\big[[(\alpha\to\beta)\to(\neg\gamma\to\neg\delta)]\to\gamma\big]\to\varphi,\quad(\varphi\to\alpha)\big\}\models\delta\to\alpha$$

Por tanto, lo que vamos a probar es que el conjunto de fórmulas

$$\big\{\big[[(\alpha\to\beta)\to(\neg\gamma\to\neg\delta)]\to\gamma\big]\to\varphi,\quad\varphi\to\alpha,\quad\neg(\delta\to\alpha)\big\}$$

es insatisfacible.

Hallamos la forma clausular de cada una de las fórmulas que intervienen. De las dos últimas se hace fácilmente:

$$\varphi\to\alpha\ \equiv\ \neg\varphi\vee\alpha;\qquad\qquad \neg(\delta\to\alpha)\ \equiv\ \neg(\neg\delta\vee\alpha)\ \equiv\ \neg\neg\delta\wedge\neg\alpha\ \equiv\ \delta\wedge\neg\alpha$$

Hallemos entonces la forma clausular de

$$\big[[(\alpha\to\beta)\to(\neg\gamma\to\neg\delta)]\to\gamma\big]\to\varphi$$

$$\begin{gathered}
\neg\big[[(\alpha\to\beta)\to(\neg\gamma\to\neg\delta)]\to\gamma\big]\vee\varphi\\
\neg\big[\neg[(\alpha\to\beta)\to(\neg\gamma\to\neg\delta)]\vee\gamma\big]\vee\varphi\\
\neg\big[\neg[\neg(\alpha\to\beta)\vee(\neg\gamma\to\neg\delta)]\vee\gamma\big]\vee\varphi\\
\neg\big[\neg[\neg(\neg\alpha\vee\beta)\vee(\neg\neg\gamma\vee\neg\delta)]\vee\gamma\big]\vee\varphi\\
\big[\neg\neg[\neg(\neg\alpha\vee\beta)\vee(\neg\neg\gamma\vee\neg\delta)]\wedge\neg\gamma\big]\vee\varphi\\
\big[[\neg(\neg\alpha\vee\beta)\vee(\gamma\vee\neg\delta)]\wedge\neg\gamma\big]\vee\varphi\\
\big[[(\neg\neg\alpha\wedge\neg\beta)\vee(\gamma\vee\neg\delta)]\wedge\neg\gamma\big]\vee\varphi\\
\big[[(\alpha\wedge\neg\beta)\vee(\gamma\vee\neg\delta)]\wedge\neg\gamma\big]\vee\varphi\\
\big[[(\alpha\vee\gamma\vee\neg\delta)\wedge(\neg\beta\vee\gamma\vee\neg\delta)]\wedge\neg\gamma\big]\vee\varphi\\
[(\alpha\vee\gamma\vee\neg\delta\vee\varphi)\wedge(\neg\beta\vee\gamma\vee\neg\delta\vee\varphi)]\wedge(\neg\gamma\vee\varphi)
\end{gathered}$$

Por tanto, hemos de probar que el conjunto de cláusulas

$$\{\alpha\vee\gamma\vee\neg\delta\vee\varphi,\quad\neg\beta\vee\gamma\vee\neg\delta\vee\varphi,\quad\neg\gamma\vee\varphi,\quad\alpha\vee\neg\varphi,\quad\neg\alpha,\quad\delta\}$$

es insatisfacible.

$$\begin{gathered}
\{\alpha\vee\gamma\vee\neg\delta\vee\varphi,\quad\neg\beta\vee\gamma\vee\neg\delta\vee\varphi,\quad\neg\gamma\vee\varphi,\quad\alpha\vee\neg\varphi,\quad\neg\alpha,\quad\delta\}\\
\Big|\ \lambda=\neg\alpha\\
\{\gamma\vee\neg\delta\vee\varphi,\quad\neg\beta\vee\gamma\vee\neg\delta\vee\varphi,\quad\neg\gamma\vee\varphi,\quad\neg\varphi,\ \delta\}\\
\Big|\ \lambda=\neg\varphi\\
\{\gamma\vee\neg\delta,\quad\neg\beta\vee\gamma\vee\neg\delta,\quad\neg\gamma,\ \delta\}\\
\Big|\ \lambda=\neg\gamma\\
\{\neg\delta,\quad\neg\beta\vee\neg\delta,\ \delta\}\\
\Big|\ \lambda=\neg\delta\\
\{\square\}
\end{gathered}$$

Ejemplo 5.8.3. *Estudia si es cierto que*

$$(a \to \neg c) \to a, \quad a \to \neg b, \quad \neg(\neg c \wedge \neg b) \models (\neg c \to a) \to b$$

Lo primero que hacemos es transformar este problema en un problema de satisfacibilidad o insatisfacibilidad de un conjunto de fórmulas.

Por el teorema de la deducción, este problema es equivalente a

$$(a \to \neg c) \to a, \quad a \to \neg b, \quad \neg(\neg c \wedge \neg b), \quad \neg c \to a \models b$$

y a su vez, este es equivalente a comprobar si el siguiente conjunto de fórmulas

$$\{(a \to \neg c) \to a, \quad a \to \neg b, \quad \neg(\neg c \wedge \neg b), \quad \neg c \to a, \quad \neg b\}$$

es insatisfacible.

Calculamos entonces la forma clausular de cada una de las fórmulas:

$(a \to \neg c) \to a$	$a \to \neg b$	$\neg(\neg c \wedge \neg b)$	$\neg c \to a$	b
$(\neg a \vee \neg c) \to a$	$\neg a \vee \neg b$	$c \vee b$	$c \vee a$	
$\neg(\neg a \vee \neg c) \vee a$				
$(a \wedge c) \vee a$				
$(a \vee a) \wedge (c \vee a)$				
a	$\neg a \vee \neg b$	$b \vee c$	$a \vee c$	$\neg b$

Por tanto, nos proponemos estudiar si el conjunto de cláusulas

$$\{a, \quad \neg a \vee \neg b, \quad b \vee c, \quad a \vee c, \quad \neg b\}$$

es o no insatisfacible. Aplicamos para ello el algoritmo de Davis-Putnmam.

$$\{a, \quad \neg a \vee \neg b, \quad b \vee c, \quad a \vee c, \quad \neg b\}$$

$$\Big| \; \lambda = a$$

$$\{\neg b, \quad b \vee c, \quad \neg b\}$$

$$\Big| \; \lambda = \neg b$$

$$\{c\}$$

$$\Big| \; \lambda = c$$

$$\emptyset$$

Al haber llegado al conjunto vacío, el conjunto de cláusulas del que partíamos es satisfacible, y por tanto, la respuesta a si

$$(a \to \neg c) \to a, \quad a \to \neg b, \quad \neg(\neg c \wedge \neg b) \models (\neg c \to a) \to b$$

es que no.

Pero con lo que hemos hecho, podemos decir algo más.

El algoritmo de Davis-Putnam nos ha dicho que el conjunto de cláusulas es satisfacible. Además, nos da una interpretación para la que todas las cláusulas son ciertas. Esta es

$$I(a) = 1; \quad I(b) = 0; \quad I(c) = 1$$

Y esta es justamente una interpretación que nos dice que la implicación semántica anterior no es cierta, pues con esa interpretación se tiene:

$I((a \to \neg c) \to a) = 1$*;* $I(a \to \neg b) = 1$*;* $I(\neg(\neg c \wedge \neg b)) = 1$*, mientras que* $I((\neg c \to a) \to b) = 0$*.*

Es decir, I es una interpretación que hace ciertas todas las premisas, pero que hace falsa la conclusión. Luego esta conclusión no puede deducirse de las premisas.

5.9. Método de Resolución

Para determinar si un conjunto de cláusulas es insatisfacible puede utilizarse otro método basado en el siguiente resultado:

Teorema 5.9.1. *Sean α, β, γ tres fórmulas en un lenguaje proposicional. Entonces*

$$\{\alpha \vee \beta, \neg\alpha \vee \gamma\} \models \beta \vee \gamma$$

Demostración:
Sea I una interpretación para la que $I(\alpha \vee \beta) = 1$ y $I(\neg\alpha \vee \gamma) = 1$. Pueden darse dos casos:

- $I(\beta) = 1$, en cuyo caso $I(\beta \vee \gamma) = 1$.
- $I(\beta) = 0$. Ahora, puesto que $I(\alpha \vee \beta) = 1$ deducimos que $I(\alpha) = 1$, es decir, $I(\neg\alpha) = 0$, y como $I(\neg\alpha \vee \gamma) = 1$ podemos concluir que $I(\gamma) = 1$, lo que implica que $I(\beta \vee \gamma) = 1$.

■

Observemos que un caso particular podría ocurrir cuando tenemos dos cláusulas unitarias de la forma

$$a \text{ y } \neg a$$

entonces, aplicando el teorema obtendríamos como consecuencia lógica una cláusula que no tiene ningún literal, es decir, la cláusula vacía $\square$. Observemos que el conjunto $\{a, \neg a\}$ es insatisfacible y hemos obtenido que

$$\{a, \neg a\} \models \square$$

5.9.1. El concepto de Resolvente

Supongamos que C es una cláusula, y que λ es un literal que aparece en la cláusula C. Como en la sección anterior denotaremos por $C - \lambda$ a la cláusula que resulta de eliminar el literal λ de la cláusula C. Por ejemplo, si $C = a \vee \neg b \vee d$ entonces $C - a = \neg b \vee d$, mientras que $C - \neg b = a \vee d$. Si $C = b$ entonces $C - b = \square$.

Sean C_1 y C_2 dos cláusulas. Supongamos que λ es un literal tal que aparece en la cláusula C_1 y λ^c aparece en C_2. Una cláusula que sea equivalente a $(C_1 - \lambda) \vee (C_2 - \lambda^c)$ es lo que se denomina una **resolvente** de C_1 y C_2.

El teorema nos dice que si C_1 y C_2 son cláusulas y R una resolvente suya, entonces $\{C_1, C_2\} \models R$.

Ejemplo 5.9.1. *Si tenemos las cláusulas $C_1 = \neg a \vee b$ y $C_2 = a$ entonces una resolvente de C_1 y C_2 es b. Nótese que $C_1 \equiv a \rightarrow b$, luego al obtener esta resolvente lo que estamos afirmando es que*

$$\{a, a \rightarrow b\} \models b$$

un hecho conocido como "modus ponens".

Ejemplo 5.9.2. *Supongamos que $C_1 = a \vee \neg c \vee d$ y $C_2 = b \vee c \vee d$. Entonces $C_1 - \neg c = a \vee d$ y $C_2 - c = b \vee d$, luego $(C_1 - \neg c) \vee (C_2 - c) = (a \vee d) \vee (b \vee d) \equiv a \vee b \vee d$.*

Observación: Puede ocurrir que $(C_1 - \lambda) \vee (C_2 - \lambda^c)$ sea una tautología, en cuyo caso no podemos obtener una resolvente de esta forma.

El **método de resolución** (sin variables) pretende, dado un conjunto de cláusulas, obtener resolventes de dichas cláusulas que son añadidas al conjunto (estas nuevas cláusulas se dice que se han deducido por resolución). Si entre estas cláusulas se encuentra la cláusula vacía, entonces el conjunto de partida es insatisfacible.

El Principio de Resolución afirma que un conjunto de cláusulas es insatisfacible si, y sólo si, a partir de ellas es posible encontrar la cláusula vacía mediante resolución.

5.9.2. Ejemplos de aplicación del Principio de Resolución:

Ejemplo 5.9.3. *Vamos a demostrar que la fórmula*

$$(a \to (b \to c)) \to (\neg(a \to \neg b) \to c)$$

es una tautología. Esto sabemos que es equivalente a probar:

$$\models (a \to (b \to c)) \to (\neg(a \to \neg b) \to c)$$

aplicando el teorema de la deducción dos veces el problema se transforma en

$$\{a \to (b \to c), \neg(a \to \neg b)\} \models c$$

o equivalentemente, en demostrar que el conjunto

$$\{a \to (b \to c), \neg(a \to \neg b), \neg c\}$$

es insatisfacible.

Hallamos la forma clausular de cada una de las fórmulas anteriores:

$a \to (b \to c)$	$\neg(a \to \neg b)$	$\neg c$
$\neg a \vee (b \to c)$	$\neg(\neg a \vee \neg b)$	
$\neg a \vee (\neg b \vee c)$	$\neg\neg a \wedge \neg\neg b$	
$\neg a \vee \neg b \vee c$	$a \wedge b$	

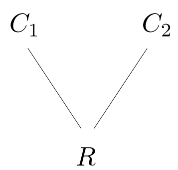
Y entonces lo que nos queda probar es que el siguiente conjunto de cláusulas es insatisfacible;

$$\{\neg a \vee \neg b \vee c,\ a,\ b,\ \neg c\}$$

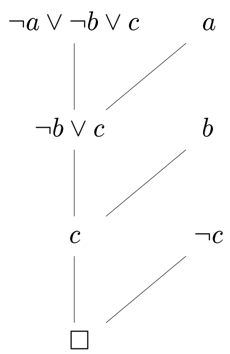
Y ahora:
Una resolvente de $\neg a \vee \neg b \vee c$ *y* a *es* $\neg b \vee c$.
Una resolvente de esta última y b *es* c.
Una resolvente de c *y* $\neg c$ *es* $\square$.
Como hemos deducido por resolución la cláusula vacía concluimos que el conjunto de cláususlas es insatisfacible, y por tanto, la primera fórmula es una tautología.

Una representación gráfica del método de resolución

Supongamos que C_1 y C_2 son dos cláusulas, y R es una resolvente de ambas. Esto entonces podemos representarlo como sigue:



En tal caso, el proceso seguido en el ejemplo precedente podríamos haberlo representado:



Ejemplo 5.9.4. *Vamos a demostrar que*

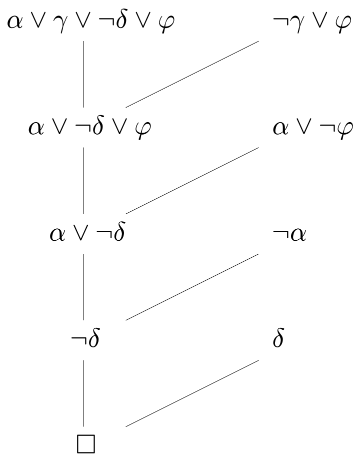
$$\models \big(\big(\big(((\alpha \to \beta) \to (\neg\gamma \to \neg\delta)) \to \gamma\big) \to \varphi\big) \to ((\varphi \to \alpha) \to (\delta \to \alpha)\big)$$

utilizando el Principio de Resolución. En primer lugar necesitamos transformar el problema en otro equivalente que consiste en decidir si un conjunto de cláusulas es satisfacible o insatisfacible. Como esto ya se ha hecho en la sección del algoritmo de Davis-Putnam, copiamos el resultado: hemos de probar que el conjunto de cláusulas

$$\{\alpha \vee \gamma \vee \neg\delta \vee \varphi, \quad \neg\beta \vee \gamma \vee \neg\delta \vee \varphi, \quad \neg\gamma \vee \varphi, \quad \alpha \vee \neg\varphi, \quad \neg\alpha, \quad \delta\}$$

es insatisfacible.

Tratamos entonces de deducir la cláusula vacía.



Al haber obtenido la cláusula vacía, concluimos que la fórmula inicial es una tautología.

5.9.3. Cómo se prueba que algo *no* es consecuencia lógica de un conjunto de proposiciones

Consideremos el problema de estudiar si es cierto que

$$\{q \to p \vee r\} \models (p \to q) \to \big(p \to ((r \to q) \to r)\big)$$

y apliquemos todas las técnicas que hemos estudiado.

En primer lugar usamos el Teorema de la Deducción tantas veces como sea posible, en este caso hasta 3 veces, y el problema queda

$$\{q \to p \vee r, p \to q, p, r \to q\} \models r$$

Método 1: Cálculo de interpretaciones en $\mathbb{Z}_2$

Si tenemos el sistema de ecuaciones

$$\left.\begin{array}{l} I(q \to p \vee r) = 1 \\ I(p \to q) = 1 \\ I(r \to q) = 1 \\ I(p) = 1 \end{array}\right\}$$

usando

$$\left.\begin{array}{l} I(p \to q) = 1 \\ I(p) = 1 \end{array}\right\} \quad \left.\begin{array}{l} 1 + I(p) + I(p)I(q) = 1 \\ I(p) = 1 \end{array}\right\} \quad \left.\begin{array}{l} I(q) = 1 \\ I(p) = 1 \end{array}\right\}$$

(a este sistema de dos ecuaciones lo llamaremos **modus ponens**) y nos queda

$$\left.\begin{array}{l} I(q \to p \vee r) = 1 \\ I(q) = 1 \\ I(r \to q) = 1 \\ I(p) = 1 \end{array}\right\}$$

ahora, de nuevo modus ponens con las dos primeras ecuaciones nos da

$$\left.\begin{array}{l} I(p \vee r) = 1 \\ I(q) = 1 \\ I(r \to q) = 1 \\ I(p) = 1 \end{array}\right\}$$

y como $I(q) = 1$, la ecuación 3 se puede eliminar porque no añade ninguna condición, además la ecuación 1 es consecuencia de la cuarta y no nos queda imposición sobre $I(r)$. Es decir, el sistema se cumple siempre que $I(p) = 1$ y $I(q) = 1$ sin importar el valor de $I(r)$. Eso nos hace ver que la interpretación $I(p) = 1, I(q) = 1, I(r) = 0$ prueba que la consecuencia lógica no se verifica.

Método 2: Algoritmo de Davis-Putnam

En primer lugar calculamos la forma clausular de todas las premisas y la negación de la conclusión:

$$\{\neg p \vee q, \neg q \vee p \vee r, p, \neg r \vee q, \neg r\}$$

Aplicando el primer paso del algoritmo a la cláusula unit p, nos queda

$$\{q, \neg r \vee q, \neg r\}$$

y de nuevo el paso uno para la cláusula unit q nos reduce el conjunto de cláusulas a $\{\neg r\}$ que es claramente satisfacible. Por tanto el conjunto de partida es satisfacible y para la interpretación 1 de todas las cláusulas unit $I(p) = 1, I(q) = 1, I(\neg r) = 1$ tenemos un modelo en el que se satisface. Por tanto la consecuencia lógica no ocurre.

Método 3: Resolución

Partimos del conjunto de cláusulas que teníamos en el método anterior:

$$\{\neg p \vee q, \neg q \vee p \vee r, p, \neg r \vee q, \neg r\}$$

y tratamos de encontrar una resolución que nos lleve a la cláusula vacía. Para ello intentamos clacular todas las resolventes posibles. Empecemos añadiendo al conjunto las resolventes de la cláusula p con todas las posibles:

$$\{\neg p \vee q, \neg q \vee p \vee r, p, \neg r \vee q, \neg r\} \cup \{q\}$$

ahora las obtenidas al resolver con $\neg r$

$$\{\neg p \vee q, \neg q \vee p \vee r, p, \neg r \vee q, \neg r\} \cup \{q\} \cup \{\neg q \vee p\}$$

ahora resolvemos con q

$$\{\neg p \vee q, \neg q \vee p \vee r, p, \neg r \vee q, \neg r\} \cup \{q\} \cup \{\neg q \vee p\} \cup \{p \vee r\}$$

ahora calculamos una resolvente de q y $\neg q \vee p$

$$\{\neg p \vee q, \neg q \vee p \vee r, p, \neg r \vee q, \neg r\} \cup \{q\} \cup \{\neg q \vee p\} \cup \{p \vee r\} \cup \{p\}$$

aún se pueden obtener nuevas cláusulas, por ejemplo la resolvente de $\neg p \vee q$ y $p \vee r$ nos da $q \vee r$ y el problema de encontrar la resolución se hace más complicado....

¿No somos capaces de encontrar la resolución o es que no existe?

No obstante, en este caso, podríamos ver que el conjunto es satisfacible.

Entre las cláusulas que hemos obtenido nos encontramos con las cláusulas p, q, $\neg r$. Si el conjunto fuera satisfacible, habría una interpretación que sobre todas las cláusulas que nos han ido apareciendo valdría 1. En particular, $I(p) = 1$, $I(q) = 1$ e $I(r) = 0$.

Probamos si esa interpretación hace ciertas todas las cláusulas y podemos comprobar que así es. Por tanto, el conjunto de cláusulas del que partíamos es satisfacible.