Fundamentos lógicos de la programación

J. I. García García

P. A. García Sánchez

J. M. Urbano Blanco

Departamento de Álgebra, Universidad de Granada

CAPíTULO 1

Introducción

Todo el mundo tiene una idea intuitiva y natural sobre el significado de la palabra "lógica". En el discurso cotidiano oímos expresiones tales como "ésto es lógico" y "tal cosa no tiene lógica". Según esta idea se designa como lógico al pensamiento correcto y coherente con determinados principios o valores. Toda persona (y quizá algunos animales) tiene una lógica natural y un sentido común que le conduce en su pensamiento y le ayuda a tomar decisiones. Ya las civilizaciones de la Antigüedad y principalmente los griegos se preocuparon por el estudio del razonamiento; estos últimos designaron como "las cosas lógicas" a aquella ciencia o tratado que versaba sobre el pensamiento en sí, sobre sus formas y leyes. Así la lógica es la ciencia o disciplina que estudia de manera formal y rigurosa los métodos y principios del razonamiento. Los seres humanos nos expresamos y comunicamos mediante el uso de los denominados lenguajes naturales. Éstos son inapropiados para el estudio de la lógica ya que están llenos de redundancias, ambigüedades y además contienen aspectos muy difíciles de formalizar; además frases tales como preguntas y órdenes carecen de un valor de verdad. Por ello se requiere la confección de un lenguaje formal que contando con ciertas reglas explícitas permita formar enunciados. A pesar de que la lógica es una materia que no requiere una gran base o experiencia para su estudio, es muy importante la representación que se adopte en el proceso de formalización y estudio de la misma. Desafortunadamente, una de las primeras dificultades en el estudio de la lógica es la existencia de diferentes simbologías. La falta de uniformidad en la notación empleada por parte de los distintos autores complica a veces la comprensión de los textos existentes. Nosotros hemos intentado seguir en estas notas aquellas notaciones y nomenclaturas que a nuestro parecer se adaptan mejor a los alumnos a los que van dirigidas.

El objetivo del curso es el de presentar algoritmos y procedimientos para la deducción automática de un determinado hecho a partir de una base de conocimientos prefijada. La consecución de este propósito la vamos a secuenciar de la siguiente forma.

El primer tema de este curso será el estudio de la lógica de proposiciones, en la cual existen unos enunciados atómicos o indivisibles, los cuales pueden tener un valor de verdad verdadero o falso, y que se combinarán mediante ciertas reglas para producir nuevos enunciados. Dos cuestiones nos van a interesar en este tema: el aspecto semántico en el cual se relaciona el valor de

verdad de los nuevos enunciados con los valores de verdad de aquellos de los que procede, y el aspecto sintáctico en el cual nos centramos principalmente en las reglas que rigen la obtención de nuevas verdades a partir de otras ya conocidas o supuestas. En este primer tema se da la aproximación más burda a la resolución del problema de obtener consecuencias a partir de una base de hechos conocidos. Eso no impide que sirva de base a las herramientas que se presentarán más adelante en el curso. Precisamente por haber empezado con una aproximación tan simple, veremos que existen situaciones en las cuales la lógica de proposiciones no resulta adecuada, sobre todo cuando hemos de tener en cuenta la estructura interna de las frases del lenguaje natural. Para solventar estos problemas introduciremos la lógica de predicados de primer orden que amplía o extiende a la lógica de proposiciones mediante el uso de los cuantificadores, términos y predicados, así como una visión más completa acerca de las interpretaciones de los enunciados. Pondremos especial énfasis en la idea de consecuencia lógica (que es el núcleo y motor del curso) y de inconsistencia de un conjunto de fórmulas; y veremos cómo el hecho de estudiar que una fórmula (enunciado) es consecuencia lógica de un conjunto de enunciados dados es equivalente a probar que un determinado conjunto es inconsistente.

En el siguiente tema nos preocupamos de ver que la inconsistencia de un conjunto de fórmulas se preserva si hacemos una serie de transformaciones en las fórmulas que lo componen. Estas transformaciones tienen como objetivo la simplificación y normalización de los enunciados que intervienen en el problema original. Una vez que reduzcamos al máximo (lo que se traduce en la práctica en desmenuzar al máximo los enunciados originales), presentaremos en el siguiente capítulo técnicas para probar la inconsistencia de un determinado conjunto de fórmulas ya normalizadas. Por último, haremos una pequeña introducción del tipo de resolución conocido como resolución PROLOG, centrándonos en la estrecha relación que tiene con lo estudiado a lo largo del curso.

Creemos que el profesional de la informática ha de tener un conocimiento sólido y claro de la lógica como herramienta de ayuda en su toma de decisiones y en la implementación de aplicaciones diversas. Las conexiones de la misma con temas tales como Teoría de Autómatas, Circuitos Digitales, Bases de Datos, Verificación de Programas, Inteligencia Artificial, etc son patentes y por tanto justifican más que de sobra la inclusión de la lógica en el currículum de los estudios de informática.

CAPíTULO 2

Lógica proposicional

1. Enunciados y conectivas

El objetivo de la Lógica es el estudio de las deducciones. Deducciones que se hacen a partir de un conjunto de afirmaciones o enunciados dados. Por *enunciado* vamos a entender una frase que sea susceptible de ser verdadera o falsa. En una primera aproximación sólo vamos a tener en cuenta que un enunciado puede estar formado por otros enunciados que están conectados entre sí, siendo el valor de verdad del enunciado en cuestión dependiente directamente de los valores que tengan los enunciados que lo componen. Así, por ejemplo, el enunciado "este capítulo es interesantísimo y el día está nublado" será cierto sí y solamente sí los enunciados "este capítulo es interesantísimo" y "el día está nublado" son ambos ciertos. Si alguna de las dos cosas falla entonces el enunciado es falso. En este ejemplo, la partícula *y* es la que conecta dos enunciados para formar uno nuevo. A estas partículas las vamos a llamar *conectivas* u *operadores lógicos*. Las conectivas más importantes que vamos a usar aparecen en los siguientes ejemplos.

1. Es tarde *y* viene lloviendo.
2. Comes *o* hablas.
3. Vas a la playa o a la montaña.
4. *Si* viene tu hermano, (*entonces*) haremos una fiesta.
5. Pepe canta cuando viene María (si viene María, entonces Pepe canta).
6. *No* vamos a la playa.

Como estos ejemplos ponen de manifiesto, estas conectivas pueden aparecer escritas de forma distinta. Una implicación no tiene por qué aparecer en la forma "Si algo entonces otra cosa", puede aparecer como "algo implica otra cosa". Lo mismo pasa con el resto de las conectivas.

Simbolizaremos los enunciados usando simplemente una letra. Aparte, vamos a escoger un símbolo para cada una de las conectivas: para el *y* usaremos $\wedge$; para el *o*, $\vee$; para la negación $\neg$; y para la implicación $\rightarrow$. (Algunas veces usaremos la conectiva que significa equivalencia y la denotaremos por $\leftrightarrow$.) De esta forma al enunciado "es tarde y viene lloviendo" lo podemos simbolizar por $p \wedge q$, siendo p "es tarde" y q "viene lloviendo". Un enunciado que no esté compuesto de otros enunciados es un *enunciado atómico*.

Tomamos un conjunto $\{p_1, p_2, \ldots p_n, \ldots\}$ a cuyos elementos vamos a llamar **proposiciones atómicas** o **variables proposicionales** (las cuales se usan para simbolizar enunciados atómicos).

Una **proposición** la definimos recursivamente de la siguiente forma.

1. Toda proposición atómica es una proposición.
2. Si α y β son dos proposiciones, entonces $\neg\alpha, \alpha\wedge\beta, \alpha\vee\beta, \alpha \to \beta$ y $\alpha \leftrightarrow \beta$ también son proposiciones.
3. No hay más proposiciones que las generadas aplicando un número finito de veces las reglas anteriores.

Debemos matizar que las reglas anteriores para definir proposiciones han de ir acompañadas de unas reglas adicionales de precedencia entre conectivas, así como el uso de paréntesis para evitar ambigüedades. Tomamos el convenio de que la negación tiene precedencia sobre el resto de las conectivas, que la conjunción y la disyunción tienen igual precedencia y ambas tienen más precedencia que la implicación. De esta manera, al escribir $a \to b\wedge c$ estamos refiriéndonos a la proposición $a \to (b\wedge c)$ y no a $(a \to b)\wedge c$. Convenimos además que el *sí y sólo si* ($\leftrightarrow$) es la conectiva con menos precedencia.

Se deben usar también paréntesis cuando nos encontramos con operadores con la misma precedencia. Si aparece escrito $a \to b \to c$ con a, b y c proposiciones, entonces tenemos que matizar si nos referimos a $a \to (b \to c)$ o a $(a \to b) \to c$ (ya que ambas son proposiciones distintas; hay por tanto que especificar claramente cuál es la premisa y cuál es la consecuencia en una implicación). Lo mismo ocurre si escribimos $a\wedge b\vee c$, ya que al no haber paréntesis, no sabemos si es una conjunción (*y*) de dos proposiciones o una disyunción (*o*). Al igual que antes, las proposiciones $a\wedge(b\vee c)$ y $(a\wedge b)\vee c$ son distintas, y la expresión $a\wedge b\vee c$ no es una proposición.

Hacemos constar que existen otras notaciones en las cuales se elimina la ambigüedad sin recurrir a los paréntesis. Por ejemplo esto ocurre con la notación polaca o prefija (en honor al filósofo y lógico polaco J. Lukasiewicz, 1878-1956), en la cual las conectivas u operadores preceden a los argumentos a los que se aplican. Si bien esta escritura resulta propicia para implementaciones en un ordenador, da como resultado expresiones de difícil lectura para un ser humano.

2. Interpretaciones

Dada una proposición a, ésta simbolizará un determinado enunciado. En principio su valor de verdad depende del valor de verdad de los enunciados atómicos que la componen, y de las conectivas que en él aparecen. En lo que sigue, y por razones de simplicidad, vamos a denotar verdadero por 1 y falso por 0.

Hemos escogido como conectivas $\neg, \wedge, \vee, \to$ y $\leftrightarrow$. Precisemos un poco más qué valor semántico les vamos a proporcionar, a saber, vamos a interpretar el

valor que para nosotros tienen dichas conectivas. Supongamos que a y b son dos proposiciones.

1. $\neg a$ es cierto si y sólo si a es falso.
2. $a \wedge b$ es cierto si y sólo si a y b son los dos ciertos.
3. $a \vee b$ es cierto si y sólo si al menos a ó b es cierto. Nótese que en el lenguaje hablado el *o* que se utiliza en la mayor parte de las ocasiones es distinto (ver los ejemplos de la sección anterior). El *o* de dos enunciados es cierto cuando alguno de ellos es cierto, pero no se contempla, en general, como cierto cuando ambos son ciertos. De esta forma el *o* del lenguaje hablado es un *o exclusivo*, en el sentido que excluye la posibilidad de que las dos enunciados sean ciertos a la vez. El *o* que hemos adoptado se conoce como *o inclusivo*.
4. $a \to b$ es falso sólo cuando a es cierto y b es falso (verdadero en el resto de los casos).
5. $a \leftrightarrow b$ es cierto si y sólo si ambos son ciertos o ambos son falsos.

Podemos esquematizar lo dicho en las siguientes tablas:

$\neg$	
0	1
1	0

$\wedge$	0	1
0	0	0
1	0	1

$\vee$	0	1
0	0	1
1	1	1

$\to$	0	1
0	1	1
1	0	1

$\leftrightarrow$	0	1
0	1	0
1	0	1

.

La primera columna simboliza el valor de verdad de a y la primera fila (salvo en el caso de la negación) el de b.

EJERCICIO 1. Comprueba que

1. $a \wedge b \equiv \neg(\neg a \vee \neg b)$,
2. $a \vee b \equiv \neg(\neg a \wedge \neg b)$,
3. $a \to b \equiv \neg a \vee b$,
4. $a \leftrightarrow b \equiv (a \to b) \wedge (b \to a)$,

donde $\equiv$ significa "tiene el mismo valor de verdad que".

A la vista de este ejercicio, podríamos haber introducido nada más que las conectivas $\neg$ y $\vee$, definiendo después el resto en función de ellas. Lo mismo ocurre con el $\neg$ y el $\wedge$, y con otros conjuntos de conectivas. Un conjunto de conectivas verificando esta propiedad se conoce como conjunto de operadores funcionalemente completo.

Existen además otras posibilidades a la hora de elegir las conectivas básicas, como puede ser considerar el *o exclusivo* ($\underline{\vee}$), la *negación del o* (NOR) o la *negación del y* (NAND).

EJERCICIO 2. ¿Cuál es la expresión del o exclusivo en función de $\neg$ y $\vee$?

Recordemos que $\mathbb{Z}_2$ es el cuerpo de los enteros modulo 2, a saber, el conjunto de los restos posibles al hacer la división de un entero por 2. Así, el 0 denota el conjunto de todos los enteros pares y 1 el de los impares. Las dos operaciones

que dan estructura de cuerpo a $\mathbb{Z}_2$ son aquellas que hereda del anillo de los enteros $\mathbb{Z}$, y éstas son la suma y la multiplicación. Sus tablas de operar son las siguientes.

$+$	0	1
0	0	1
1	1	0

$\times$	0	1
0	0	0
1	0	1

Nótese que la tabla de multiplicar es la misma que la "tabla" del conector $\wedge$. Sin embargo, la tabla de sumar no es la misma que la de $\vee$, pero sólo se diferencia en que $1 + 1 = 0$ mientras que el *o* de dos proposiciones ciertas lo hemos interpretado como cierto. (De hecho, la tabla de sumar corresponde con el *o exclusivo*.) Para conseguir la tabla del conector $\vee$ debemos "sumar" o superponer las tablas de la suma y la multiplicación, con lo que queda que el valor de verdad de $a \vee b$ es la suma de los valores de a y b más el producto de sus valores de verdad (donde las cuentas de suma y producto se hacen módulo 2).

Es fácil observar que el valor que tome $\neg a$ es el resultado de sumarle al valor de a el 1. El lector puede hacer uso de lo probado en el segundo apartado del Ejercicio 1, para corroborar la fórmula que hemos obtenido antes del valor de verdad de $a \vee b$. De forma análoga, pero ahora usando el tercer apartado de dicho ejercicio, se puede deducir la expresión de $\rightarrow$ en función de la suma y el producto usando que $a \rightarrow b \equiv \neg a \vee b$. Por lo que el valor de verdad de $a \rightarrow b$ se calcula sumando 1 al valor de a y al producto de los valores de a y b. Por último, es fácil deducir que el valor de verdad de $a \leftrightarrow b$ es la suma de los valores de a, b y 1. Al realizar los cálculos anteriores hay que tener en cuenta que para todo $x \in \mathbb{Z}_2$ se verifica que $x^2 = x$ y $x + x = 2x = 0$.

Queda patente que nuestra elección de denotar cierto o verdadero por 1 y falso por 0 no era arbitraria, ya que de esta forma podemos expresar los valores que hemos dado a las conectivas en función de la suma y multiplicación en $\mathbb{Z}_2$.

Lo visto hasta ahora en esta sección motiva la siguiente definición. Una **interpretación** es una aplicación I que va del conjunto de las proposiciones en $\mathbb{Z}_2 = \{0, 1\}$ y que cumple las siguientes propiedades:

1. $\mathrm{I}(\neg a) = 1 + \mathrm{I}(a)$,
2. $\mathrm{I}(a \vee b) = \mathrm{I}(a) + \mathrm{I}(b) + \mathrm{I}(a)\mathrm{I}(b)$,
3. $\mathrm{I}(a \wedge b) = \mathrm{I}(a)\mathrm{I}(b)$,
4. $\mathrm{I}(a \rightarrow b) = 1 + \mathrm{I}(a) + \mathrm{I}(a)\mathrm{I}(b)$,
5. $\mathrm{I}(a \leftrightarrow b) = 1 + \mathrm{I}(a) + \mathrm{I}(b)$.

Dada una proposición p, si $\mathrm{I}(p) = 1$ (respectivamente $\mathrm{I}(a) = 0$), diremos que p es cierta (respectivamente falsa) bajo la interpretación I, o bien que I interpreta a p como cierta (respectivamente falsa). Si $\mathrm{I}(p) = 1$ también se dice que I satisface a p.

El lector puede observar que el valor que toma una interpretación I para cualquier proposición depende exclusivamente del valor que toma I en las

proposiciones atómicas, o dicho de otra forma, si dos interpretaciones toman el mismo valor para cualquier proposición atómica, entonces son la misma interpretación. Así pues, dar una interpretación no es más que asignar un valor de verdad a cada proposición, y como las conectivas tienen asignado un determinado valor semántico, dicho valor de verdad dependerá de cómo interpretemos las proposiciones atómicas.

EJEMPLO 1. Sea I una interpretación tal que $\mathrm{I}(a) = \mathrm{I}(b) = \mathrm{I}(c) = 1$. Bajo esta interpretación interpretamos (y valga la redundancia) la proposición $a \vee (b \wedge c)$. Tenemos

$$\begin{aligned}\mathrm{I}(a \vee (b \wedge c)) &= \mathrm{I}(a) + \mathrm{I}(b \wedge c) + \mathrm{I}(a)\mathrm{I}(b \wedge c) \\ &= \mathrm{I}(a) + \mathrm{I}(b)\mathrm{I}(c) + \mathrm{I}(a)\mathrm{I}(b)\mathrm{I}(c) = 1 + 1 \times 1 + 1 \times 1 \times 1 = 1,\end{aligned}$$

o sea, bajo la interpretación I, la proposición $a \vee (b \wedge c)$ es cierta. □

EJEMPLO 2. Si $\alpha \to \beta$ es falsa bajo una determinada interpretación I, ¿qué podemos decir sobre el valor de verdad de

$$\delta \;=\; (\alpha \to (\gamma \to \beta)) \vee \neg((\alpha \wedge \gamma) \to (\neg\beta \vee \gamma))?$$

Observamos que $\mathrm{I}(\alpha \to \beta) = 0$ si y sólo si $\mathrm{I}(\alpha) = 1$ y $\mathrm{I}(\beta) = 0$. Desarrollando $\mathrm{I}(\delta)$ y substituyendo $\mathrm{I}(\alpha)$ e $\mathrm{I}(\beta)$ por su valor, obtenemos que $\mathrm{I}(\delta) = 1 + \mathrm{I}(\gamma)$. Concluimos que bajo la interpretación I el valor de verdad de δ coincidirá con aquel que tenga $\neg\gamma$. □

Dada una proposición, el número de combinaciones distintas de los valores de verdad de las proposiciones atómicas que la componen es finito, de hecho, si en dicha proposición intervienen n proposiciones atómicas, el número de esas combinaciones es 2^n. Así, el número de interpretaciones posibles de dicha proposición es a lo sumo 2^n, y si las ponemos todas juntas obtenemos lo que se conoce como la **tabla de verdad** de la proposición dada. Por ejemplo, la tabla de verdad de $(a \vee b) \wedge b$ es

a	b	$a \vee b$	$(a \vee b) \wedge b$
0	0	0	0
0	1	1	1
1	0	1	0
1	1	1	1

(la segunda columna corresponde a un cálculo intermedio, por lo que en esta tabla está también incluida la tabla de verdad de $a \vee b$).

Dos proposiciones a y b son **lógicamente equivalentes** si tienen el mismo valor de verdad bajo cualquier interpretación (hecho que denotábamos hasta ahora por $a \equiv b$).

3. Tautologías y contradicciones

Existen proposiciones que son ciertas sea cual sea el valor de verdad de las proposiciones atómicas que la conforman. Por otro lado también existen proposiciones que son siempre falsas bajo cualquier interpretación. Por ejemplo, si I es una interpretación arbitraria,

$$\begin{aligned}\mathrm{I}(a \to (b \to a)) &= 1 + \mathrm{I}(a) + \mathrm{I}(a)\mathrm{I}(b \to a) \\ &= 1 + \mathrm{I}(a) + \mathrm{I}(a)(1 + \mathrm{I}(b) + \mathrm{I}(a)\mathrm{I}(b)) \\ &= 1 + \mathrm{I}(a) + \mathrm{I}(a) + \mathrm{I}(a)\mathrm{I}(b) + \mathrm{I}(a)\mathrm{I}(b)^2 = 1,\end{aligned}$$

lo que quiere decir que el valor de verdad de $a \to (b \to a)$ es verdadero independientemente de los valores de verdad de a y b. Por el contrario $a \wedge (\neg a)$ es siempre falsa bajo cualquier interpretación.

Una proposición es una **tautología** cuando es cierta bajo cualquier interpretación. Lo contrario de tautología es que exista una interpretación bajo la cual la proposición sea falsa. En este caso diremos que la proposición es **refutable**.

Una proposición p es una **contradicción** cuando $\neg p$ es una tautología. Lo contrario de ser una contradicción es ser **satisfacible**, a saber, si existe una interpretación bajo la cual es cierta.

Ejemplo 3. La proposición $a \to (b \to a)$ es una tautología.
La proposición $a \vee b$ es satisfacible y refutable.
La proposición $a \wedge \neg a$ es una contradicción.
Si a y b son lógicamente equivalentes, entonces $a \leftrightarrow b$ es una tautología. □

En el siguiente ejercicio se muestran algunas tautologías con nombre propio. No es nuestro propósito mencionarlas todas, sino hacer que el lector ejercite un poco el cálculo de interpretaciones.

Ejercicio 3. Demuestra que las siguientes proposiciones son tautologías.

1. Ley de doble negación: $\neg\neg a \to a$.
2. Leyes de simplificación: $(a \wedge b) \to a, a \to (a \vee b)$.
3. Ley de conmutatividad de la conjunción: $(a \wedge b) \to (b \wedge a)$.
4. Ley de conmutatividad de la disyunción: $(a \vee b) \to (b \vee a)$.
5. Ley de Clavius: $(\neg a \to a) \to a$.
6. Ley de De Morgan: $\neg(a \wedge b) \to (\neg a \vee \neg b)$.
7. Segunda ley de De Morgan: $\neg(a \vee b) \to (\neg a \wedge \neg b)$.
8. Ley de inferencia alternativa: $((a \vee b) \wedge \neg a) \to b$.
9. Segunda ley de inferencia alternativa:$((a \vee b) \wedge \neg b) \to a$.
10. Modus ponendo ponens:$((a \to b) \wedge a) \to b$.
11. Modus tollendo tollens:$((a \to b) \wedge \neg b) \to \neg a$.

4. Consecuencia lógica

Supongamos que sabemos que son ciertas las siguientes afirmaciones:

1. Si llueve, Pepe se pone contento.
2. Está lloviendo.

Está claro que, a partir de estas afirmaciones, podemos deducir que Pepe se va a poner contento. El hecho de poder afirmar que Pepe se ponga contento es una "consecuencia lógica" de los dos hechos que ya conocíamos. Describamos el ejemplo anterior de otra forma. Sean a y b dos proposiciones e I una interpretación tal que $\mathrm{I}(a) = \mathrm{I}(a \to b) = 1$. Usando las propiedades que debe cumplir I, tenemos que $1 = 1 + \mathrm{I}(a) + \mathrm{I}(a)\mathrm{I}(b)$, y substituyendo el valor de $\mathrm{I}(a)$ en esta igualdad, obtenemos que $1 = 1 + 1 + 1 \times \mathrm{I}(b)$, de lo que concluimos que $\mathrm{I}(b) = 1$. Dicho de otra forma, si se interpretan a y $a \to b$ como ciertas, a la fuerza b también es cierta.

Dado un conjunto de proposiciones $\Gamma \cup \{\alpha\}$, decimos que α es **consecuencia lógica** de Γ, o que Γ **implica semánticamente** a α, y lo denotaremos por $\Gamma \models \alpha$, si para toda interpretación bajo la cual todas las proposiciones de Γ son ciertas, se tiene que α también es cierta bajo esa interpretación. Si $\Gamma = \emptyset$, escribimos $\models \alpha$ en lugar de $\emptyset \models \alpha$. El lector debería comprobar que escribir $\models \alpha$ equivale a decir que α es una tautología.

Según esta definición, en el ejemplo anterior, b es consecuencia lógica de $\{a \to b, a\}$, y lo denotamos

$$\{a \to b, a\} \models b.$$

Abusando un poco de la notación, a veces también escribiremos

$$a \to b, a \models b.$$

El siguiente ejercicio nos permite decidir cuando una proposición es consecuencia lógica de un conjunto dado.

Ejercicio 4. Dado un conjunto finito de proposiciones $\Gamma \cup \{\alpha\}$, son equivalentes:

- $\Gamma \models \alpha$,
- para toda interpretación I,

$$(\prod_{\beta \in \Gamma} \mathrm{I}(\beta))(1 + \mathrm{I}(\alpha)) = 0.$$

Solución. Si tomamos una interpretación arbitraria I, entonces todas las proposiciones de Γ son ciertas bajo I si y sólo sí $\prod_{\beta \in \Gamma} \mathrm{I}(\beta) = 1$ (esto se debe a que estamos haciendo las cuentas en $\mathbb{Z}_2$), y el que ésto fuerce que $\mathrm{I}(\alpha) = 1$, se puede escribir como

$$(\prod_{\beta \in \Gamma} \mathrm{I}(\beta))(1 + \mathrm{I}(\alpha)) = 0,$$

ya que el producto es cero si y sólo si uno de los dos factores es cero, y en caso de que sea el primero, entonces I no es una interpretación bajo la cual todas las proposiciones de Γ son ciertas. Así pues, si I hace ciertas todas las proposiciones de Γ, el primer factor debe ser uno, y para que el producto sea cero, tenemos que $1 + \mathrm{I}(\alpha) = 0$, o equivalentemente, $\mathrm{I}(\alpha) = 1$. □

Hacemos notar que la consecuencia lógica también puede verse utilizando tablas de verdad.

EJERCICIO 5. Resolver $\{p \vee q \to r, \neg r\} \models \neg q$ aplicando el criterio dado en el ejercicio anterior y usando tablas de verdad.

Otro criterio que nos va a facilitar la tarea de comprobar la implicación semántica es el siguiente.

RESULTADO 1 (Teorema de la deducción). *Dado un conjunto de proposiciones* $\Gamma \cup \{\alpha, \beta\}$, *las siguientes afirmaciones son equivalentes:*

(1) $\Gamma \cup \{\alpha\} \models \beta$,
(2) $\Gamma \models \alpha \to \beta$.

DEMOSTRACIÓN. *(1) implica (2).* Sea I una interpretación tal que $\mathrm{I}(\gamma) = 1$ para todo $\gamma \in \Gamma$. Tenemos que probar que $\mathrm{I}(\alpha \to \beta) = 1$. Distinguimos dos casos, dependiendo del valor de verdad de α bajo la interpretación I.

- Si $\mathrm{I}(\alpha) = 0$, entonces $\mathrm{I}(\alpha \to \beta) = 1 + 0 + 0 \times \mathrm{I}(\beta) = 1$.
- Si $\mathrm{I}(\alpha) = 1$, entonces $\mathrm{I}(\gamma) = 1$ para todo $\gamma \in \Gamma \cup \{\alpha\}$, y por (1), esto significa que $\mathrm{I}(\beta) = 1$. De esta forma $\mathrm{I}(\alpha \to \beta) = 1 + 1 + 1 \times 1 = 1$.

(2) implica (1). Supongamos que I es una interpretación tal que $\mathrm{I}(\gamma) = 1$ para todo $\gamma \in \Gamma \cup \{\alpha\}$. Tenemos que demostrar que $\mathrm{I}(\beta) = 1$. Por hipótesis, tenemos que en particular $\mathrm{I}(\gamma) = 1$ para todo $\gamma \in \Gamma$. Usando (2), esto lleva a que $\mathrm{I}(\alpha \to \beta) = 1$. Ahora bien, también por hipótesis sabemos que $\mathrm{I}(\alpha) = 1$, por lo que $1 = \mathrm{I}(\alpha \to \beta) = 1 + 1 + 1 \times \mathrm{I}(\beta)$ y en consequencia $\mathrm{I}(\beta) = 1$. □

Veamos con algunos ejemplos cómo se puede utilizar este resultado.

EJEMPLO 4. Supongamos que nos piden probar que para cualesquiera dos proposiciones a, b y c, la proposición $(a \to (b \to c)) \to ((a \to b) \to (a \to b))$ es una tautología, o lo que es lo mismo, demostrar que

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

Usando el Teorema de la deducción ésto es equivalente a probar que

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

que a su vez equivale a

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

y ésto último es lo mismo que

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

Sea ahora I una interpretación tal que

$$\begin{aligned} &\mathrm{I}(a \to (b \to c)) = 1,\\ &\mathrm{I}(a \to b) = 1,\\ &\mathrm{I}(a) = 1. \end{aligned}$$

Entonces $1 = \mathrm{I}(a \to b) = 1 + 1 + 1 \times \mathrm{I}(b)$, por lo que $\mathrm{I}(b) = 1$. Substituyendo $\mathrm{I}(a)$ e $\mathrm{I}(b)$ en $1 = \mathrm{I}(a \to (b \to c))$, obtenemos que $1 = 1+1+1\times\mathrm{I}(b \to c) = 1+1+1\times\mathrm{I}(c)$, por lo que $\mathrm{I}(c) = 1$.

Si hubiésemos usado directamente la definición de tautología, tendríamos que haber demostrado que $\mathrm{I}((a \to (b \to c)) \to ((a \to b) \to (a \to c))) = 1$ para cualquier interpretación I, lo cual es bastante tedioso (inténtese). □

Ejemplo 5. Probemos ahora que

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

Según el Teorema de la deducción (usado dos veces), esto es equivalente a probar que

$$\{\neg a \to \neg b, \neg a \to b\} \models a.$$

Sea pues I una interpretación tal que $\mathrm{I}(\neg a \to \neg b) = \mathrm{I}(\neg a \to b) = 1$. Entonces

$$\begin{aligned} \mathrm{I}(\neg a \to \neg b) &= 1 + \mathrm{I}(\neg a) + \mathrm{I}(\neg a)\mathrm{I}(\neg b) = \mathrm{I}(a) + (1 + \mathrm{I}(a))(1 + \mathrm{I}(b))\\ &= \mathrm{I}(a) + 1 + \mathrm{I}(a) + \mathrm{I}(b) + \mathrm{I}(a)\mathrm{I}(b) = 1 + \mathrm{I}(b) + \mathrm{I}(a)\mathrm{I}(b)\\ \mathrm{I}(\neg a \to b) &= 1 + \mathrm{I}(\neg a) + \mathrm{I}(\neg a)\mathrm{I}(b) = \mathrm{I}(a) + (1 + \mathrm{I}(a))\mathrm{I}(b) = \mathrm{I}(a) + \mathrm{I}(b) + \mathrm{I}(a)\mathrm{I}(b), \end{aligned}$$

por lo que $1 + \mathrm{I}(b) + \mathrm{I}(a)\mathrm{I}(b) = \mathrm{I}(a) + \mathrm{I}(b) + \mathrm{I}(a)\mathrm{I}(b)$. Cancelando en ambas partes de la igualdad concluimos que $\mathrm{I}(a) = 1$. □

Ejemplo 6. Demostremos que $((\alpha \to \beta) \to \alpha) \to \alpha$ es una tautología.

El problema se puede resolver fácilmente comprobando que la tabla de dicha proposición está formada íntegramente por unos:

α	β	$\alpha \to \beta$	$(\alpha \to \beta) \to \alpha$	$((\alpha \to \beta) \to \alpha) \to \alpha$
0	0	1	0	1
0	1	1	0	1
1	0	0	1	1
1	1	1	1	1

El lector debería intentar resolver el problema usando el teorema de la deducción, comparando en este caso cuál de los métodos es más rápido. □

Ejemplo 7. Sobre cuatro cinéfilos A, B, C y D, habitantes de Rarilandia, pais con un sólo un cine de una sola sala, se sabe lo siguiente: Si A decide ir al cine, B también irá. Sobre C y D se sabe que no les gusta ver una misma película juntos. B y C, o van al cine juntos, o no va ninguno de los dos. Si A no va al cine, entonces B y D sí quieren ir. ¿Quienes estarán en el cine esta noche?

Si X es un individuo, representaremos por x la proposición "*X va al cine*". De esta forma la información dada en el enunciado se puede representar como:

- "*Si A decide ir al cine, B también irá*": $a \to b$.
- "*Sobre C y D se sabe que no les gusta ver una misma película juntos*": $\neg(c \wedge d)$. Si en vez de esta fórmula, escribiéramos $(\neg c) \leftrightarrow d$, estaríamos representando la información dada en el enunciado, así como el hecho de que siempre uno de los dos (y sólo uno) habría de estar presente.
- "*B y C, o van al cine juntos o no va ninguno de los dos*": $b \leftrightarrow c$.
- "*Si A no va al cine, entonces B y D sí quieren ir*": $(\neg a) \to (b \wedge d)$.

Los datos que nos dan se esquematizan en las proposiciones que acabamos de describir y ya que sabemos que al interpretarlas como los enunciados de los que proceden éstos son verdaderos, obtenemos que (tomando como $\mathrm{I}(x) = 1$ si y sólo si "X va al cine")

$$\mathrm{I}(a \to b) = 1,\ \mathrm{I}(\neg(c \wedge d)) = 1,\ \mathrm{I}(b \leftrightarrow c) = 1,\ \mathrm{I}(\neg a \to b \wedge d) = 1,$$

lo cual se reescribe como

$$1 + \mathrm{I}(a) + \mathrm{I}(a)\mathrm{I}(b) = 1,\ 1 + \mathrm{I}(c)\mathrm{I}(d) = 1,$$
$$1 + \mathrm{I}(b) + \mathrm{I}(c) = 1,\ 1 + (1 + \mathrm{I}(a)) + (1 + \mathrm{I}(a))\mathrm{I}(b)\mathrm{I}(d) = 1,$$

o bien

$$\mathrm{I}(a)(1 + \mathrm{I}(b)) = 0,\ \mathrm{I}(c)\mathrm{I}(d) = 0,\ \mathrm{I}(b) = \mathrm{I}(c),\ \mathrm{I}(a) + \mathrm{I}(b)\mathrm{I}(d)(1 + \mathrm{I}(a)) = 1.$$

De esta forma $\mathrm{I}(a) = 0$ ó $(1 + \mathrm{I}(b)) = 0$. Si $\mathrm{I}(a) = 0$, entonces de la última ecuación obtenemos $\mathrm{I}(b)\mathrm{I}(d) = 1$, con lo que $\mathrm{I}(b) = \mathrm{I}(d) = 1$, y por la segunda obtendríamos $\mathrm{I}(c) = 0$, lo que al usar la tercera ecuación contradice que $\mathrm{I}(b) = 1$. Así $\mathrm{I}(a) = 1$ y por tanto $1 + \mathrm{I}(b) = 0$, o lo que es lo mismo, $\mathrm{I}(b) = 1$. Por la tercera ecuación, sabemos que entonces $\mathrm{I}(c) = 1$, y por la segunda que $\mathrm{I}(d) = 0$. Nótese que todo esto no entra en contradicción con la última ecuación. Por tanto, bajo las hipótesis dadas, a, b y c son ciertos y d es falso. Tenemos así que a, b, c y $\neg d$ son consecuencia lógica de las hipótesis $\{a \to b, \neg(c \wedge d), b \leftrightarrow c, \neg a \to b \wedge d\}$.

Es interesante observar que si la segunda sentencia la hubiésemos representado como $(\neg c) \leftrightarrow d$, la solución del problema hubiera sido la misma. Ésto es consecuencia de que $\{(\neg c) \leftrightarrow d\} \models \neg(c \wedge d)$. □

EJEMPLO 8. Sobre cinco enunciados A, B, C, D y E se sabe lo siguiente:

1. Si A, C y D son los tres falsos, entonces B es falso.
2. C es verdad, o bien, A es verdad si y sólo si B es verdad.
3. Si C es verdad o A es falso, entonces: D es verdad, B es falso y el enunciado 2 es falso.
4. Si A es verdad entonces, cuando B es falso C es verdad.
5. A es verdad y E es falso, o bien A es falso y E es verdad.

A partir de la información dada, ¿qué podemos decir sobre el valor de verdad de cada uno de los enunciados A, B, C, D y E?

La información del enunciado se puede representar de la siguiente forma:

1. $B \to (A \vee C \vee D)$,

2. $C \vee (A \leftrightarrow B)$,
3. $(C \vee \neg A) \rightarrow (D \wedge \neg B \wedge \neg(C \vee (A \leftrightarrow B)))$,
4. $A \rightarrow (\neg B \rightarrow C)$,
5. $(A \wedge \neg E) \vee (\neg A \wedge E)$.

Observamos que la variable E sólo aparece en la quinta afirmación. Si construimos la tabla de verdad para las cuatro primeras sentencias (una tabla con 16 filas), observamos que el valor de verdad de las cuatro a la vez es 1 sólo en dos casos: $(A, B, C, D) = (1, 1, 0, 0)$ y $(A, B, C, D) = (1, 1, 0, 1)$ (esto es, sólo existen dos interpretaciones que hagan los cuatro primeros enunciados ciertos a la vez). Concluimos que la respuesta final es: "A y B son verdad, C y E son falsas, y sobre D no se puede afirmar nada". □

Ejercicio 6. El señor Pérez, empadronador de la isla de Tururulandia, tiene como objetivo el censar la población de dicha isla. La tarea no es fácil debido al hecho de que la población se divide en dos grupos bien distinguidos: los honrados y los embusteros. Los honrados siempre dicen la verdad, mientras que un embustero sólo es capaz de producir mentiras.

El gobierno de la isla encarga como trabajo al señor Pérez la ardua tarea de contar los honrados y embusteros de la isla.

He aquí cinco de los muchos problemas con los que se encontró nuestro empadronador.

1. Llama a la puerta de una casa, en la que sabía a ciencia cierta que vivía un matrimonio, y el marido abre la puerta para ver quien es. El empadronador le dice: "necesito información sobre usted y su esposa. ¿Cuál de ustedes, si alguno lo es, es honrado y cuál un embustero?," a lo que el hombre de la casa respondió "ambos somos embusteros," cerrando la puerta de golpe. ¿A qué grupo pertenece cada uno de ellos?

   Resolución. Usemos p para denotar "el marido es honrado" y q para "la mujer es honrada" (esto tiene otra lectura: p se interpreta como "el marido es honrado", o bien, $\mathrm{I}(p) = 1$ si y sólo si "el marido es honrado"). Una persona de la isla es honrada si y sólo si lo que dice es cierto. Así, lo que sabemos que es cierto se simboliza por $p \leftrightarrow \neg p \wedge \neg q$. De esta forma $\mathrm{I}(p \leftrightarrow \neg p \wedge \neg q) = 1$, por lo que $1 + \mathrm{I}(p) + \mathrm{I}(\neg p)\mathrm{I}(\neg q) = 1$. Simplificando, obtenemos $(\mathrm{I}(p) + 1)\mathrm{I}(q) = 1$, con lo que $\mathrm{I}(q) = 1$ (la mujer es honrada) e $\mathrm{I}(p) = 0$ (el hombre es un embustero). □

2. La segunda casa que visita también está habitada por un matrimonio. Al llamar a la puerta y formular la misma pregunta que antes, el marido responde: "Por lo menos uno de nosotros es un embustero," cerrando a continuación la puerta. ¿Qué es el marido y qué es la mujer?
3. Visita una tercera casa, y en las mismas condiciones de antes, recibe la respuesta: "Si yo soy honrado, entonces mi mujer también lo es." ¿Qué es el marido y qué es la mujer?

4. En la última casa que visita, pues ya estaba cansado de partirse el coco, la respuesta es "Yo soy lo mismo que mi mujer." ¿Qué es el marido y qué es la mujer?
5. De vuelta a su casa se encuentra con tres individuos, A,B y C, en la calle, y pensando en que quizás podía tener más suerte con ellos decide preguntarles qué son cada uno de ellos. Le pregunta al primero, A, y no entiende la respuesta, ya que en ese momento pasa una de esas motos que hacen un ruido ensordecedor y no corren nada. El segundo, B, le aclara que lo que ha dicho el primero es que es un embustero, pero el tercero, C, le advierte que no haga caso del segundo, B, ya que es un embustero. ¿Puedes deducir algo de lo ocurrido?

EJERCICIO 7. Un país está habitado por gente que se clasifica en dos tipos: los veraces, que siempre dicen la verdad, y los mendaces, que siempre mienten; además, a las preguntas que se les hace sólo responden con un monosílabo ("sí" o "no"). Un turista llega a una bifurcación de una carretera en la que no hay indicaciones, salvo un cartel que anuncia la proximidad de un hotel, y un habitante parado en la misma bifurcación. Si el turista quiere ir al hotel, ¿qué pregunta debe hacer al indígena para que éste con su respuesta "sí" o "no"le indique el camino que debe seguir? Justificar la respuesta.

5. Demostración

Dado un conjunto de hipótesis, nos podemos preguntar si una cierta proposición es o no consecuencia lógica del conjunto dado. Esto podemos resolverlo haciendo uso de lo expuesto en la sección anterior. Supongamos ahora, que lo que pretendemos es dar una consecución "lógica" de razonamientos que nos lleven de nuestro conjunto de hipótesis hasta lo que queramos probar como cierto a partir de éste. Dichos razonamientos deberán ser válidos en algún sentido que tendremos que matizar de antemano. También parece lógico el permitir el uso de una serie de verdades que se conozcan o fijen como ciertas, a las que vamos a llamar axiomas. Esto que estamos exponiendo no es otra cosa que el concepto de demostración: una consecución de razonamientos que nos llevan de un conjunto de hipótesis y axiomas prefijados a una nueva proposición que es consecuencia de dichas hipótesis. Como acabamos de indicar, antes de introducir el concepto de demostración, debemos introducir los axiomas del cálculo proposicional clásico. El conjunto de axiomas que vamos a introducir consta de tres patrones o esquemas de axioma; cada uno de ellos engloba a infinitas proposiciones que son instancia de dichos patrones. La elección de estos patrones no es única, e incluso el número de éstos puede variar (a veces en la literatura el conjunto de axiomas se toma como el conjunto de todas las tautologías). A continuación definimos lo que vamos a entender por axioma.

Un **axioma** del cálculo proposicional clásico es un elemento del conjunto $\mathcal{A} = \bigcup_{i=1}^{3} \mathcal{A}_i$, donde

1. $\mathcal{A}_1 = \{\alpha \to (\beta \to \alpha) | \alpha, \beta$ proposiciones$\}$ (leyes del afortiori),
2. $\mathcal{A}_2 = \{(\alpha \to (\beta \to \gamma)) \to ((\alpha \to \beta) \to (\alpha \to \gamma)) | \alpha, \beta$ proposiciones$\}$ (leyes autodistributivas de la implicación o leyes de Frege),
3. $\mathcal{A}_3 = \{(\neg\alpha \to \neg\beta) \to ((\neg\alpha \to \beta) \to \alpha) | \alpha, \beta$ proposiciones$\}$ (leyes clásicas de reducción al absurdo).

Como hemos visto en los ejemplos de las secciones anteriores, todo axioma es una tautología. La única regla que en principio vamos a utilizar para realizar deducciones a partir de proposiciones ya probadas es la siguiente. En la sección anterior probamos que $\{a, a \to b\} \models b$, ésto es, si se suponen ciertos a y $a \to b$, entonces b es cierto. Todo ésto se puede entender de la siguiente forma: si tenemos $a \to b$, entonces siempre que nos encontremos con a, lo podemos reescribir como b. Juntamos ahora todas estas ideas para dar el concepto de demostración.

Sea $\Gamma \cup \{\alpha\}$ un conjunto de proposiciones. Una **prueba**, **deducción** o **demostración** (formal) de α a partir de Γ (al cual llamaremos conjunto de **hipótesis**) es una sucesión $\alpha_1, \ldots, \alpha_n$, tal que

1. $\alpha_n = \alpha$ (el último paso de la demostración es lo que queremos probar) y
2. para todo $i \leq m$ se cumple alguna de las siguientes condiciones:
   a) $\alpha_i \in \Gamma \cup \mathcal{A}$,
   b) existen $j, k < i$ tales que $\alpha_k = \alpha_j \to \alpha_i$ (en este caso decimos que α_i es el resultado de aplicar **modus ponens** a α_k y α_j).

A las proposiciones $\alpha_1, \ldots, \alpha_n$ las llamaremos **pasos de la demostración**.

Diremos que la proposición α es deducible de Γ, o α se puede demostrar a partir de Γ, o Γ implica sintácticamente α, si existe una demostración formal de α a partir de las hipótesis Γ. Escribimos $\Gamma \vdash \alpha$ para simbolizar este hecho. En el caso en que Γ sea el conjunto vacío (no asumimos hipótesis alguna), escribiremos simplemente $\vdash \alpha$ y diremos que α es un **teorema** del cálculo proposicional clásico o una **ley lógica** clásica.

Ejercicio 8. Sea $\Gamma \cup \Delta \cup \{a\}$ un conjunto de proposiciones.

1. Si $\alpha \in \Gamma$, entonces $\Gamma \vdash \alpha$.
2. Si $\Gamma \vdash \alpha$ y $\Gamma \subseteq \Delta$, entonces $\Delta \vdash \alpha$.

Como el lector posiblemente haya observado ya, tanto en la definición de axioma como en la de demostración, todas las proposiciones implicadas contienen sólo la conectiva $\to$ y la negación. Esto es debido al hecho de que $\wedge$, $\vee$ y $\leftrightarrow$ se pueden poner en función de $\to$ y $\neg$ ($\{\neg, \to\}$ es un conjunto funcionalmente completo).

Ejemplo 9. Sean α, β, γ y δ proposiciones.

1. $\{\beta\} \vdash \alpha \to \beta$.
   1 β, hipótesis.

2 $\beta \to (\alpha \to \beta) \in \mathcal{A}_1$.
3 $\alpha \to \beta$, modus ponens 1, 2.

En este caso, los pasos de la demostración son tres: $\alpha_1 = \beta$, $\alpha_2 = \beta \to (\alpha \to \beta)$ y $\alpha_3 = \alpha \to \beta$.

2. $\{\neg\alpha \to \beta, \neg\alpha \to \neg\beta\} \vdash \alpha$.
   1 $\neg\alpha \to \neg\beta$, hipótesis.
   2 $(\neg\alpha \to \neg\beta) \to ((\neg\alpha \to \beta) \to \alpha) \in \mathcal{A}_3$.
   3 $(\neg\alpha \to \beta) \to \alpha$, modus ponens 2,3.
   4 $\neg\alpha \to \beta$, hipótesis.
   5 α, modus ponens 3,4.
3. $\{\alpha \to \beta, \beta \to \gamma\} \vdash \alpha \to \gamma$.
   1 $\alpha \to \beta$, hipótesis.
   2 $\beta \to \gamma$, hipótesis.
   3 $\alpha \to (\beta \to \gamma)$, del paso 2, aplicando el ejemplo 1.
   4 $(\alpha \to (\beta \to \gamma)) \to ((\alpha \to \beta) \to (\alpha \to \gamma)) \in \mathcal{A}_2$.
   5 $(\alpha \to \beta) \to (\alpha \to \gamma)$, modus ponens 3,4.
   6 $\alpha \to \gamma$, modus ponens 1,5.

   Nótese que en este último ejemplo, el paso 3 no es exactamente un paso de una demostración, pero sobreentenderemos que en realidad estamos insertando los pasos correspondientes a los que se hace cita. De esta forma, cada vez que demostremos un teorema, nos permitiremos insertarlo en nuestras demostraciones como un paso más, entendiendo así que estamos incluyendo en la demostración todos los pasos correspondientes a la demostración del teorema insertado.

□

6. Algunas leyes y reglas lógicas

Hacer demostraciones formales haciendo uso exclusivo de la definición puede resultar largo, tedioso y desmoralizador. Para facilitar la tarea damos a continuación un conjunto de reglas y leyes lógicas clásicas que facilitan las demostraciones, y constituyen de hecho un conjunto de técnicas de demostración formal. Vamos a demostrar las más importantes y el resto (no menos importante) las proponemos como ejercicio. La diferencia esencial entre una ley y una regla lógica estriba en que las leyes son sinónimo de teoremas, y las reglas son herramientas de inferencia que nos sirven en las demostraciones para construir nuevos pasos a partir de pasos anteriores, tal y como ocurre con el modus ponens. Como veremos a continuación, asociada a casi toda ley existe una regla, la cual se puede entender como la aplicación de la ley en cuestión en un contexto práctico.

Resultado previo 1 (ley de identidad). *Para cualquier proposición α se tiene que*

$$\vdash \alpha \to \alpha.$$

Demostración. Una posible demostración es la siguiente:

1 $(\alpha \to ((\alpha \to \alpha) \to \alpha)) \to ((\alpha \to (\alpha \to \alpha)) \to (\alpha \to \alpha)) \in \mathcal{A}_2$,
2 $\alpha \to ((\alpha \to \alpha) \to \alpha) \in \mathcal{A}_1$,
3 $(\alpha \to (\alpha \to \alpha)) \to (\alpha \to \alpha)$, modus ponens 1,2,
4 $\alpha \to (\alpha \to \alpha) \in \mathcal{A}_1$,
5 $\alpha \to \alpha$, modus ponens 3,4.

□

El teorema que exponemos a continuación, el teorema de la deducción, es la formalización de un método de prueba bastante usual en Matemáticas: frecuentemente, cuando deseamos establecer una implicación de la forma $\alpha \to \beta$, adjuntamos la premisa α al conjunto de hipótesis iniciales y a partir de ahí intentamos establecer la conclusión β.

Resultado 2 (Teorema de la deducción o de Herbrand, 1930). *Dado un conjunto de proposiciones* $\Gamma \cup \{\alpha, \beta\}$. *Si* $\Gamma \cup \{\alpha\} \vdash \beta$, *entonces* $\Gamma \vdash \alpha \to \beta$.

Demostración. Vamos a demostrar que para cualquier n natural, si existe una demostración $\beta_1, \ldots, \beta_n$ con hipótesis $\Gamma \cup \{\alpha\}$, entonces $\Gamma \vdash \alpha \to \beta_n$. Esto demostraría el enunciado del teorema. La demostración la hacemos por inducción sobre n. Llamemos β a β_n.

1. Si $n = 1$, pueden darse los siguientes casos:
   a) β es un axioma, en cuyo caso la demostración de $\alpha \to \beta$ a partir de Γ es:
      1 β, axioma.
      2 $\beta \to (\alpha \to \beta) \in \mathcal{A}_1$.
      3 $\alpha \to \beta$, modus ponens 1,2.
   b) $\beta \in \Gamma$, en cuyo caso la demostración es la misma que en el caso anterior pero cambiando el primer paso, poniendo $\beta \in \Gamma$ en su lugar.
   c) $\beta = \alpha$, en cuyo caso, por la ley de identidad $\vdash \alpha \to \alpha$, por lo que trivialmente $\Gamma \vdash \alpha \to \alpha$.
2. Asumamos como hipótesis de inducción que el resultado es cierto para las demostraciones de longitud estrictamente menor que n, y probémoslo para las de longitud n. En esta situación, aparte de los casos anteriores, aparece un cuarto caso: $\beta = \beta_n$ se obtiene aplicando modus ponens a dos proposiciones anteriores en la secuencia. En este caso deben aparecer proposiciones de la forma $\beta_i = \gamma, \beta_j = \gamma \to \beta$, con $i, j < n$. Esto quiere decir que existen demostraciones de γ y $\gamma \to \beta$ de longitudes estrictamente menores que n, lo que nos coloca en la hipótesis de inducción. Así del hecho de que $\Gamma \cup \{\alpha\} \vdash \gamma$, y $\Gamma \cup \{\alpha\} \vdash \gamma \to \beta$, se tiene que $\Gamma \vdash \alpha \to \gamma$ y $\Gamma \vdash \alpha \to (\gamma \to \beta)$. Sea $\delta_1, \delta_2, \ldots, \delta_k$ una sucesión tal que $\delta_1, \delta_2, \ldots, \delta_l$ sea una deducción de $\alpha \to \gamma$, y sea

$\delta_{l+1}, \ldots, \delta_k$ una deducción de $\alpha \to (\gamma \to \beta)$ a partir de Γ. Haciendo

$$\begin{aligned}
&\delta_{k+1} = (\alpha \to (\gamma \to \beta)) \to ((\alpha \to \gamma) \to (\alpha \to \beta)) \in \mathcal{A}_2,\\
&\delta_{k+2} = (\alpha \to \gamma) \to (\alpha \to \beta), \text{ modus ponens } k, k+1,\\
&\delta_{k+3} = \alpha \to \beta, \text{ modus ponens } l, k+2,
\end{aligned}$$

obtenemos que $\delta_1, \ldots, \delta_{k+3}$ es una demostración de $\alpha \to \beta$ con conjunto de hipótesis Γ.

□

La vuelta atrás del teorema de la deducción se demuestra aplicando simplemente modus ponens. Tenemos de esta forma el siguiente resultado.

RESULTADO 3. *Dado el conjunto de fórmulas* $\Gamma \cup \{\alpha, \beta\}$*, son equivalentes:*

1. $\Gamma \cup \{\alpha\} \vdash \beta$,
2. $\Gamma \vdash \alpha \to \beta$.

Con el teorema de la deducción podemos demostrar la regla de reducción al absurdo, otra herramienta importante para simplificar las demostraciones.

RESULTADO 4 (regla de "reductio ad absurdum" clásica). *Dado el conjunto de proposiciones* $\Gamma \cup \{\alpha, \beta\}$*. Si* $\Gamma \cup \{\neg\alpha\} \vdash \beta$ *y* $\Gamma \cup \{\neg\alpha\} \vdash \neg\beta$*, entonces* $\Gamma \vdash \alpha$*.*

DEMOSTRACIÓN. Por el teorema de la deducción, si $\Gamma \cup \{\neg\alpha\} \vdash \beta$ y $\Gamma \cup \{\neg\alpha\} \vdash \neg\beta$, entonces $\Gamma \vdash \neg\alpha \to \beta$ y $\Gamma \vdash \neg\alpha \to \neg\beta$. Por un razonamiento análogo al hecho en el Ejemplo 9 (punto 2), tenemos que $\Gamma \vdash \alpha$. □

A continuación enunciamos algunas reglas y leyes lógicas cuya demostración damos para que el lector se acostumbre a la forma de demostración en el cálculo proposicional clásico. El lector notará que aparecen reglas y leyes con el mismo nombre. Las letras α, β y γ representan proposiciones como viene siendo costumbre, y Γ es un conjunto de proposiciones.

RESULTADO 5 (leyes de silogismo o transitividad de la flecha).

1. $\vdash (\alpha \to \beta) \to ((\beta \to \gamma) \to (\alpha \to \gamma))$,
2. $\vdash (\beta \to \gamma) \to ((\alpha \to \beta) \to (\alpha \to \gamma))$.

DEMOSTRACIÓN. Demostremos la primera, ya que la segunda se demuestra de forma análoga. Probar

$$\vdash (\alpha \to \beta) \to ((\beta \to \gamma) \to (\alpha \to \gamma))$$

es, por el teorema de la deducción, lo mismo que probar que

$$\{\alpha \to \beta\} \vdash (\beta \to \gamma) \to (\alpha \to \gamma).$$

Usando una vez más el teorema de la deducción tenemos que probar que eso es equivalente a

$$\{\alpha \to \beta, \beta \to \gamma\} \vdash \alpha \to \gamma,$$

lo que equivale a

$$\{\alpha \to \beta, \beta \to \gamma, \alpha\} \vdash \gamma,$$

lo cual es fácil de demostrar, ya que basta con usar un par de veces modus ponens. □

Consecuencia 1 (regla del silogismo). *Si* $\Gamma \vdash \alpha \to \beta$ *y* $\Gamma \vdash \beta \to \gamma$, *entonces* $\Gamma \vdash \alpha \to \gamma$.

Resultado 6 (ley de conmutación de premisas).

$$\vdash (\alpha \to (\beta \to \gamma)) \to (\beta \to (\alpha \to \gamma)).$$

Demostración. Vamos a hacer de nuevo uso del teorema de la deducción para probar esta ley. Probar

$$\vdash (\alpha \to (\beta \to \gamma)) \to (\beta \to (\alpha \to \gamma)),$$

equivale a probar

$$\{\alpha \to (\beta \to \gamma)\} \vdash \beta \to (\alpha \to \gamma),$$

que equivale a

$$\{\alpha \to (\beta \to \gamma), \beta\} \vdash \alpha \to \gamma,$$

y que por último es lo mismo que probar que

$$\{\alpha \to (\beta \to \gamma), \beta, \alpha\} \vdash \gamma.$$

Aplíquese modus ponens un par de veces y se obtiene una demostración de este hecho. □

Consecuencia 2 (regla de conmutación de premisas). *Si* $\Gamma \vdash \alpha \to (\beta \to \gamma))$, *entonces* $\Gamma \vdash \beta \to (\alpha \to \gamma)$.

El lector observará que algunas leyes son dobles (como las del silogismo vistas anteriormente). Esto se debe precisamente a la regla de conmutación de premisas. Las dos leyes de silogismo son equivalentes vía esta regla.

Resultado 7 (ley de doble negación). $\vdash \neg\neg\alpha \to \alpha$.

Demostración. Por el teorema de la deducción, probar $\vdash \neg\neg\alpha \to \alpha$, es equivalente a probar que $\{\neg\neg\alpha\} \vdash \alpha$. Usemos ahora el teorema de reducción al absurdo (regla de reductio ad absurdum clásica) para probar este hecho. Tomemos como $\Gamma = \{\neg\neg\alpha\}$ y como α el propio α. Tenemos pues que

$$\Gamma \cup \{\neg\alpha\} \vdash \neg\alpha$$

(lo cual es trivial, ya que una hipótesis siempre es una deducción del conjunto de hipótesis de partida) y que

$$\Gamma \cup \{\neg\alpha\} \vdash \neg\neg\alpha$$

por la misma razón. De $\Gamma \cup \{\neg\alpha\}$, se deduce una proposición (en este caso $\neg\alpha$) y su negación, por lo que aplicando el teorema de reducción al absurdo, se tiene que $\Gamma \vdash \alpha$, lo que concluye la demostración. □

CONSECUENCIA 3 (ley débil de doble negación). $\vdash \alpha \to \neg\neg\alpha$.

DEMOSTRACIÓN. Por el teorema de la deducción, basta con probar que $\{\alpha\} \vdash \neg\neg\alpha$. Usemos el teorema de reducción al absurdo para probar este hecho. Partimos por tanto de $\{\alpha, \neg\neg\neg\alpha\}$ y tenemos que llegar a demostrar una proposición y su negación.

1. $\neg\neg\neg\alpha \to \neg\alpha$, ley de doble negación.
2. $\neg\neg\neg\alpha$, hipótesis.
3. $\neg\alpha$, modus ponens 1,2.
4. α, hipótesis.

Así pues hemos demostrado α y $\neg\alpha$. Por el teorema de reducción al absurdo, $\{\alpha\} \vdash \neg\neg\alpha$. □

EJERCICIO 9 (regla de "reductio ad absurdum"minimal o intuicionista). Si $\Gamma \cup \{\alpha\} \vdash \beta$ y $\Gamma \cup \{\alpha\} \vdash \neg\beta$, entonces $\Gamma \vdash \neg\alpha$.

EJERCICIO 10 (leyes de Duns Scoto).

1. $\vdash \neg\alpha \to (\alpha \to \beta)$.
2. $\vdash \alpha \to (\neg\alpha \to \beta)$.

EJERCICIO 11 (principio de inconsistencia). Si $\Gamma \vdash \alpha$ y $\Gamma \vdash \neg\alpha$, entonces $\Gamma \vdash \beta$.

EJERCICIO 12 (leyes débiles de Duns Scoto).

1. $\vdash \neg\alpha \to (\alpha \to \neg\beta)$.
2. $\vdash \alpha \to (\neg\alpha \to \neg\beta)$.

EJERCICIO 13 (principio de inconsistencia débil). Si $\Gamma \vdash \alpha$ y $\Gamma \vdash \neg\alpha$, entonces $\Gamma \vdash \neg\beta$.

EJERCICIO 14 (ley de contraposición fuerte o "ponendo ponens").

$$\vdash (\neg\beta \to \neg\alpha) \to (\alpha \to \beta).$$

EJERCICIO 15 (ley de contraposición "ponendo tollens").

$$\vdash (\beta \to \neg\alpha) \to (\alpha \to \neg\beta).$$

EJERCICIO 16 (ley de contraposición "tollendo ponens").

$$\vdash (\neg\alpha \to \beta) \to (\neg\beta \to \alpha).$$

EJERCICIO 17 (ley de contraposición débil o "tollendo tollens").

$$\vdash (\alpha \to \beta) \to (\neg\beta \to \neg\alpha).$$

Las leyes de contraposición tienen una utilidad especial a la hora de eliminar negaciones no deseadas de sitios clave. Supongamos que nos piden probar que $\vdash \neg(a \to b) \to a$. Usando el teorema de la deducción, equivale a probar que $\{\neg(a \to b)\} \vdash a$. Si usamos ahora la regla de reducción al absurdo, tendríamos que ver que a partir de $\{\neg(a \to b), \neg a\}$ podemos demostrar una proposición

y su negación (inténtese). El problema de tener una negación que afecta a una implicación, nos impide usar modus ponens con ésta. Veamos otra forma más sencilla de afrontar este problema. Nótese que al aplicar contraposición a $\neg(a \to b) \to a$, obtenemos $\neg a \to (a \to b)$, que precisamente es una de las leyes de Duns Scoto. Así, una demostración directa de $\neg(b \to a) \to \neg a$ sería:

1 $\neg a \to (a \to b)$, ley de Duns Scoto.
2 $\neg(a \to b) \to a$ contraposición (tollendo ponens) aplicada a 1 (con esto entendemos que hemos escrito la ley tollendo tolens apropiada y hemos hecho modus ponens con 1).

EJERCICIO 18 (regla de prueba por casos). Si $\Gamma \cup \{\alpha\} \vdash \beta$ y $\Gamma \cup \{\neg\alpha\} \vdash \beta$, entonces $\Gamma \vdash \beta$.

EJERCICIO 19 (ley débil de Clavius).

$$\vdash (\alpha \to \neg\alpha) \to \neg\alpha.$$

EJERCICIO 20 (ley de Clavius).

$$\vdash (\neg\alpha \to \alpha) \to \alpha.$$

EJERCICIO 21 (regla de retorsión, regla de Clavius). Si $\Gamma \cup \{\neg\alpha\} \vdash \alpha$, entonces $\Gamma \vdash \alpha$.

Como ya comentamos con anterioridad, en el conjunto de axiomas que hemos dado no aparecen ni la conectiva $\wedge$ ni la $\vee$. Debido a que semánticamente $a \vee b$ tiene el mismo valor de verdad que $\neg a \to b$ y $a \wedge b$ tiene el mismo valor de verdad que $\neg(a \to \neg b)$, entenderemos siempre que $a \vee b$ y $a \wedge b$ son abreviaturas de $\neg a \to b$ y de $\neg(a \to \neg b)$, respectivamente.

EJERCICIO 22 (leyes de adjunción).

1. $\vdash \alpha \to \alpha \vee \beta$.
2. $\vdash \beta \to \alpha \vee \beta$.

EJERCICIO 23 (reglas de adjunción o de introducción de la disyunción).

1. Si $\Gamma \vdash \alpha$, entonces $\Gamma \vdash \alpha \vee \beta$.
2. Si $\Gamma \vdash \beta$, entonces $\Gamma \vdash \alpha \vee \beta$.

EJERCICIO 24 (ley conmutativa de la disyunción).

$$\vdash \alpha \vee \beta \to \beta \vee \alpha.$$

EJERCICIO 25.

1. $\vdash \alpha \wedge \beta \to \alpha$.
2. $\vdash \alpha \wedge \beta \to \beta$.

EJERCICIO 26 (reglas de simplificación o de eliminación de la conjunción). Si $\Gamma \vdash \alpha \wedge \beta$, entonces $\Gamma \vdash \alpha$ y $\Gamma \vdash \beta$.

EJERCICIO 27.

$$\vdash (\alpha \to \gamma) \to ((\beta \to \gamma) \to (\alpha \vee \beta \to \gamma)).$$

EJERCICIO 28 (otra regla de prueba por casos). Si $\Gamma \cup \{\alpha\} \vdash \gamma$ y $\Gamma \cup \{\beta\} \vdash \gamma$, entonces $\Gamma \cup \{\alpha \vee \beta\} \vdash \gamma$.

EJERCICIO 29 (ley de Peirce).

$$\vdash ((\alpha \to \beta) \to \alpha) \to \alpha.$$

EJERCICIO 30.

$$\vdash \alpha \to (\beta \to \alpha \wedge \beta).$$

EJERCICIO 31 (regla del producto o de introducción de la conjunción). Si $\Gamma \vdash \alpha$ y $\Gamma \vdash \beta$, entonces $\Gamma \vdash \alpha \wedge \beta$.

EJEMPLO 10. Supongamos que cambiamos nuestro conjunto de axiomas $\mathcal{A}_3$ por

$$\mathcal{A}'_3 = \{(\neg\alpha \to \neg\beta) \to (\beta \to \alpha) | \alpha, \beta \text{ proposiciones }\}.$$

En esta situación nos preguntamos si dada una proposición α, ésta es un teorema en el sistema L formado por $\mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3$ si y sólo si lo es también en el sistema L' formado por $\mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}'_3$. Observemos que en el sistema L' también siguen siendo válidos tanto el teorema de la deducción como la ley del silogismo, ya que en sus demostraciones no intervienen elementos de $\mathcal{A}_3$.

Para probar lo anterior, basta ver que el conjunto de teoremas de L coincide con el conjunto de teoremas de L'. Claramente por la contraposición fuerte (Ejercicio 14) se tiene que todo elemento de $\mathcal{A}'_3$ es también un teorema de L, por lo que todo teorema de L' también lo es de L.

Veamos ahora que todo teorema de L lo es también en L'. Probemos que $\vdash_{L'} (\neg a \to \neg b) \to ((\neg a \to b) \to a)$ o equivalentemente que $\{\neg a \to \neg b, \neg a \to b\} \vdash_{L'} a$ (el símbolo $\vdash_{L'}$ tiene el mismo significado que $\vdash$ salvo que el conjunto de axiomas considerados es ahora $\mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}'_3$). En primer lugar demostremos que $\{\neg a \to a\} \vdash_{L'} a$ (ley de Clavius para L'):

1. $\neg a \to a$, hipótesis,
2. $\neg a \to (\neg\neg(\neg a \to a) \to \neg a) \in \mathcal{A}_1$,
3. $(\neg\neg(\neg a \to a) \to \neg a) \to (a \to \neg(\neg a \to a)) \in \mathcal{A}'_3$,
4. $\neg a \to (a \to \neg(\neg a \to a))$, silogismo de 2 y 3,
5. $(\neg a \to (a \to \neg(\neg a \to a))) \to ((\neg a \to a) \to (\neg a \to \neg(\neg a \to a))) \in \mathcal{A}_2$,
6. $(\neg a \to a) \to (\neg a \to \neg(\neg a \to a))$, modus ponens 4,5,
7. $\neg a \to \neg(\neg a \to a)$, modus ponens 1,6,
8. $(\neg a \to \neg(\neg a \to a)) \to ((\neg a \to a) \to a) \in \mathcal{A}'_3$
9. $(\neg a \to a) \to a$, modus ponens 7,8,
10. a, modus ponens 1,9.

Para finalizar sólo nos queda ver que $\{\neg a \to \neg b, \neg a \to b\} \vdash_{L'} a$:

1. $\neg a \to \neg b$, hipótesis,

2. $\neg a \to b$, hipótesis,
3. $(\neg a \to \neg b) \to (b \to a) \in \mathcal{A}'_3$,
4. $b \to a$, modus ponens 1,3,
5. $\neg a \to a$, silogismo 2,4,
6. $(\neg a \to a) \to a$, ley de Clavius en L',
7. a, modus ponens 5,6.

□

7. Teoremas de coherencia y adecuación de la lógica proposicional

En este capítulo hemos introducido dos conceptos fundamentales: el de tautología y el de teorema, relacionado el primero con las interpretaciones y el segundo con las demostraciones en la lógica proposicional. Si α es un teorema ($\vdash \alpha$), entonces por definición existe una demostración $\alpha_1, \ldots, \alpha_n$ de α. A α_1 no le queda más remedio que ser un axioma. Y como hemos probado ya, todo axioma es una tautología. El segundo paso a la fuerza debe ser también un axioma, y por tanto una tautología. El tercero es o bien un axioma o modus ponens de los dos primeros, que ya sabemos que son tautologías. Hemos visto que si a y $a \to b$ son tautologías, entonces también lo es b, por lo que el tercer paso de la demostración también es una tautología. De la misma forma se prueba que todos los pasos de una demostración de un teorema son tautologías, por lo que α en sí es una tautología. Lo que acabamos de demostrar es lo siguiente.

Resultado 8 (Teorema de coherencia). *Sea α una proposición. Si $\vdash \alpha$, entonces $\models \alpha$.*

La otra implicación también es cierta, aunque su demostración se sale de los objetivos de este curso. Así pues, toda tautología es un teorema de la lógica proposicional. Aceptaremos sin demostración el siguiente teorema, cuya demostración el lector puede seguir en (1).

Resultado 9 (Teorema de adecuación). *Sea α una proposición. Si $\models \alpha$, entonces $\vdash \alpha$.*

Ejercicio 32. Consideremos un conjunto finito Γ. Usando los teoremas de coherencia y adecuación y los teoremas de la deducción pruébese que $\Gamma \vdash \alpha$ si y solo si $\Gamma \models \alpha$.

8. Bibliografía específica

(1) A. G. Hamilton, Logic for mathematicians, Cambridge University Press, 1978 (Capítulos 1 y 2).
(2) E. Mendelson, Introduction to mathematical logic, Chapman and Hall, 1997 (Capítulo 1).

CAPíTULO 3

Lógica de Primer Orden. Semántica

1. Introducción

En el capítulo anterior estudiamos las deducciones que se pueden hacer a partir de un conjunto de enunciados dados. Estos enunciados los esquematizábamos con letras, en el caso de que fuesen atómicos, o bien con letras y conectivas en el caso de que fuesen compuestos. Esta primera aproximación es bastante rudimentaria, y si queremos analizar otro tipo de deducciones vamos a tener que enriquecer el lenguaje a utilizar para representar o simbolizar los enunciados. A modo de ejemplo, supongamos que son ciertos los enunciados: "Todo hombre es mortal" y "Sócrates es un hombre". De estas dos suposiciones, se deduce que "Sócrates es mortal". Ahora bien, no podemos deducir la mortalidad de Sócrates usando tan sólo los conocimientos adquiridos en el capítulo anterior, ya que con ellos es como si llevásemos unas gafas con una graduación que no nos permite desmembrar los enunciados (que también son comunes en el discurso cotidiano) de forma que podamos aplicar otro tipo de reglas de deducción. Según lo estudiado hasta el momento, "Todo hombre es mortal" es una proposición atómica, que podemos denotar por p; "Sócrates es un hombre" es otra proposición atómica que podemos denotar por q; y por último "Sócrates es mortal" es otra proposición atómica que vamos a denotar por r. Está claro que según lo visto hasta ahora, r no es consecuencia lógica de p, q. El enunciado "Todo hombre es mortal", lo podemos reescribir de la forma "Para toda cosa, si esa cosa es hombre entonces es mortal". De este modo se parece algo más a los enunciados matemáticos y deja entrever que necesitamos un lenguaje en el que deben de haber cuantificadores (el "para todo" del ejemplo), variables (la "cosa" del ejemplo), constantes ("Sócrates" en el ejemplo), predicados (como el de "ser hombre" en el ejemplo), y las conectivas de la lógica proposicional. Además, aunque en este ejemplo no aparezcan, también necesitaremos símbolos de función, que nos permiten designar objetos de forma indirecta en nuestros enunciados, por ejemplo "*el padre de Manolo* es geólogo".

2. Lenguajes de primer orden

Antes de presentar la forma en que vamos a recoger las ideas expuestas en la introducción, necesitamos describir el conjunto de símbolos que vamos a utilizar en dicha representación.

Un lenguaje de primer orden $\mathcal{L}$ tiene por alfabeto los siguientes elementos.

- Variables, que las denotamos por $x_1, x_2, \ldots$, así como $x, y, z, \ldots$. El conjunto de todas las variables lo llamamos $\text{Var}(\mathcal{L})$.
- Constantes, que las denotamos por $c_1, c_2, \ldots$. Usaremos también las letras $a, b, c, \ldots$. $\text{Cons}(\mathcal{L})$ denota el conjunto de las constates de $\mathcal{L}$.
- Símbolos de función, que los denotamos por $f_1^1, \ldots, f_i^k, \ldots$. El superíndice indica la ariedad, esto es, el número de argumentos a los que se aplica la función. Usaremos para simplificar a veces $f, g, h, \ldots$. El conjunto de todas las funciones de $\mathcal{L}$ lo simbolizaremos por $\text{Func}(\mathcal{L})$.
- Símbolos de relación, que los denotamos por $R_1^1, \ldots, R_i^j, \ldots$. El superíndice indica la ariedad. Usaremos siempre letras mayúsculas. El conjunto de relaciones lo denotamos por $\text{Rel}(\mathcal{L})$.
- Operadores lógicos: $\neg$, $\vee$, $\wedge$, $\rightarrow$, $\exists$ (cuantificador existencial), $\forall$ (cuantificador universal).

Aunque no se diga explícitamente, también vamos a hacer uso de paréntesis y comas para hacer más legibles las palabras formadas con este alfabeto. Normalmente obviaremos la ariedad de las funciones y de los símbolos de relación para no hacer pesada la notación.

Una **expresión** del lenguaje de primer orden $\mathcal{L}$ es cualquier palabra formada con los símbolos anteriormente expuestos. Entre las posibles expresiones que se pueden formar, distinguiremos términos, átomos y fórmulas. Los términos harán el papel de los objetos en el lenguaje hablado. Los fórmulas vendrán a representar los enunciados y los átomos las fórmulas que son más elementales o indescomponibles.

Los **términos** del lenguaje $\mathcal{L}$ se definen recursivamente de la siguiente forma.

1. Toda constante es un término.
2. Toda variable es un término.
3. Si f es un símbolo de función de ariedad n y $t_1, \ldots, t_n$ son términos, entonces $f(t_1, \ldots, t_n)$ es un término.
4. Todos los términos se generan aplicando las reglas anteriores un número finito de veces.

El conjunto de los términos de un lenguaje de primer orden $\mathcal{L}$ se denota por $Term(\mathcal{L})$.

Ejemplos de términos: manzana, pera, padre(Juan), a, $f(a, g(a))$. Como ya se ha dicho, sirven para designar objetos. Con los símbolos de función "construimos" nuevos objetos a partir de otros objetos. La expresión $f(a, f(b), c)$ no es un término ya que el símbolo de función f no puede tener a la vez ariedad 1 y 3.

Aunque hayamos hecho distinción entre las constantes y los símbolos de función, éstas se pueden considerar como símbolos de función 0-arios, es decir, como símbolos de función que no se aplican a ningún argumento.

Una **fórmula atómica** o **átomo** es una expresión de la forma $R(t_1, \ldots, t_n)$, con R un símbolo de relación de ariedad n y $t_1, \ldots, t_n$ términos. Ejemplos de fórmulas atómicas: ESGRANDE(manzana), ESMAYOR(Pepe,Juan), TIENEUNPARAGUAS(Pepito), ESLASUMA(5,3,2), LLUEVE.

Las **fórmulas** de $\mathcal{L}$ se definen recursivamente como sigue.

1. Todo átomo es una fórmula.
2. Si φ y ψ son fórmulas, entonces $\neg\varphi, \varphi \vee \psi, \varphi \wedge \psi$ y $\varphi \to \psi$ son fórmulas.
3. Si φ es una fórmula y x es una variable, entonces $\forall x\varphi$ y $\exists x\varphi$ son fórmulas.
4. Las fórmulas se generan aplicando las reglas anteriores un número finito de veces.

Denotamos por Form($\mathcal{L}$) al conjunto de todas las fórmulas del lenguaje $\mathcal{L}$.

Ejemplo 11. Las siguientes expresiones son fórmulas:

$$\forall x \exists y (R(x, f(y)) \wedge P(a, z) \to R(y, f(x)))$$
$$R(x, y) \to \forall z Q(z, x, y)$$

Las siguientes no lo son:

$$R(x, y) \vee R(x)$$
$$Q(x) \wedge f(x, a)$$

La primera no es fórmula debido a que R no puede tener ariedad 1 y 2 a la vez. La segunda no lo es porque $f(x, a)$ no es fórmula. □

Al igual que hemos hablado de funciones 0-arias, también podemos hablar de relaciones 0-arias, y de esta forma la lógica proposicional, por decirlo de algún modo, queda inmersa dentro de la lógica de primer orden. Comentamos además que existen otras lógicas de orden superior en las que se permite cuantificar no sólo variables, sino también predicados y símbolos de función. Una frase como "para todo número natural x, existe una función $f : \mathbb{N} \to \mathbb{N}$ tal que $f(x) = 2x$", se escapa de nuestros lenguajes de primer orden.

Ejemplo 12. Veamos cómo podemos traducir a lenguajes de primer order algunas frases cotidianas.

- "Es necesario tener valor y preparación para escalar montañas".
  Usaremos los siguientes predicados con los siguientes significados:
  $V(x)$: x tiene valor,
  $P(x)$: x tiene preparación,
  $E(x)$: x escala montañas.
  Una forma posible sería: $\forall x(E(x) \to (V(x) \wedge P(x)))$.
- "No es oro todo lo que reluce".
  Usaremos
  $O(x)$: x está hecho de oro,
  $R(x)$: x reluce.
  Una posible solución es: $\exists x(R(x) \wedge \neg O(x))$, o bien $\neg(\forall x(R(x) \to O(x)))$.

- "Sólo rugen los leones".
  Definimos
  $R(x)$: x ruge,
  $L(x)$: x es león.
  Solución posible: $\forall x(R(x) \to L(x))$.
- "El gato del hermano de Pepe se llama Lulú".
  Consideramos el predicado
  $LL(x, y)$: x se llama y,
  así como las funciones
  $g(z)$: gato de z,
  $h(z)$: hermano de z,
  y los símbolos de constante que aparecen en el enunciado: Pepe, Lulú.
  Podemos escribir: $LL(g(h(\text{Pepe})), \text{Lulú})$.

□

3. Ocurrencias libres y ligadas. Variables libres y ligadas. Sentencias

Consideremos el enunciado: "para todo x existe y tal que $x+y = 0$". Podemos denotar por R la relación binaria que simbolice que la suma de sus argumentos es cero. Una posible fórmula que recoge este enunciado sería $\forall x \exists y R(x, y)$. Ahora bien, nótese que $\exists y R(x, y)$ sigue siendo una fórmula bien formada. La diferencia con la anterior es que hemos dejado de cuantificar universalmente la variable x. Se dice que x ha quedado libre en esta nueva fórmula. Al traducir un enunciado del lenguaje hablado en fórmula, es difícil, y de hecho nada común, poder "dejarse" una variable libre. Pues bien, estas ideas son las que se analizan a continuación. Veremos más adelante la importancia que tiene esta diferenciación entre variables que aparecen libres y aquellas que no lo están.

El **radio de acción** de un cuantificador es la fórmula a la que ese cuantificador afecta. Así, el radio de acción de $\forall x$ en la fórmula $\forall x \psi$ es ψ, y el de $\exists x$ en $\exists x \psi$ es ψ.

Ejemplo 13. En la siguiente fórmula,

$$\forall x(R(x) \wedge \exists y Q(x, y))$$

el radio de acción de $\forall x$ es $R(x) \wedge \exists y Q(x, y)$, y el de $\exists y$ es $Q(x, y)$. □

Una **ocurrencia** de una variable x en una fórmula ψ es una aparición de x en la escritura de ψ. Una ocurrencia de una variable x en una fórmula ψ es **ligada** si o bien aparece inmediatamente detrás de un cuantificador, o bien aparece en el radio de acción de un cuantificador (universal o existencial) cuya variable acompañante sea x. Una ocurrencia de una variable es **libre** si no es ligada.

Ejemplo 14. En la fórmula

$$\forall x(R(x, z) \wedge \exists y \forall z Q(x, y, z))$$

todas las ocurrencias de x están ligadas; la primera de z es libre y la segunda ligada (no estamos contando la que va cuantificada); y la única ocurrencia de y es ligada. □

Una fórmula es una **sentencia** si todas las ocurrencias de todas sus variables son ligadas.

Por lo comentado antes, todas las fórmulas que proceden de traducir un enunciado (sin truncar) del lenguaje hablado tienen todas sus variables ligadas, y por tanto son sentencias.

4. Estructuras e interpretaciones

Supongamos que tenemos la fórmula $\forall x \exists y R(x, y, a)$. Esta fórmula la podríamos "interpretar" como "para todo x existe un y tal que $x * y = a$". Para ello, lo que hemos hecho es darle un significado al predicado R. Ahora bien, está claro que este enunciado es cierto dependiendo de dónde se muevan las variables x e y. Si tomamos como "dominio" un grupo del cual escogemos x e y, y a como el elemento neutro de dicho grupo, esta propiedad no es otra que la existencia de elemento inverso para cada elemento del grupo, y por tanto es verdadera. Sin embargo, si tomamos por dominio el conjunto de los números enteros y por elemento a el 1, tenemos que ese enunciado es falso. Así pues, el valor de verdad de una fórmula depende del valor que demos a los símbolos de relación, funciones y constantes que aparecen en dicha fórmula, así como del dominio en donde se tomen sus variables.

Una **$\mathcal{L}$-estructura** $\mathcal{E}$ es una tupla $(D, \{c_i^{\mathcal{E}}\}, \{f_j^{\mathcal{E}}\}, \{R_k^{\mathcal{E}}\})$, donde:

1. D es un conjunto no vacío llamado dominio o universo de la $\mathcal{L}$-estructura,
2. a cada constante c_i de $\mathcal{L}$ le corresponde un elemento $c_i^{\mathcal{E}}$ de D,
3. a cada símbolo de función f_j^n de $\mathcal{L}$ le corresponde una aplicación:
$$f_j^{\mathcal{E}} : D^n \to D,$$
4. a cada símbolo de relación R_k^m de $\mathcal{L}$ le corresponde una aplicación
$$R_k^{\mathcal{E}} : D^m \to \mathbb{Z}_2.$$

Para que la notación no sea tan engorrosa identificaremos $c_i^{\mathcal{E}}$ con c_i, $f_i^{\mathcal{E}}$ con f_j^n y $R_k^{\mathcal{E}}$ con R_k^m.

Consideremos ahora la fórmula $R(x, y)$ y supongamos que interpretamos R como "ser igual a". Al dar una $\mathcal{L}$-estructura no damos valores a las variables, pero está claro que $R(x, y)$ será cierto sólo si x e y son iguales. Por tanto, si asignamos a las variables x e y el mismo valor, tendremos que $R(x, y)$ será cierto bajo la interpretación dada. Para interpretar una fórmula, no sólo hay que dar valores a los predicados, símbolos de función y constantes en un dominio dado,

sino que también hay que asignar unos valores a las variables (como veremos más adelante sólo importan las variables que aparecen libres).

Una **asignación** en el lenguaje $\mathcal{L}$ en una determinada $\mathcal{L}$-estructura $\mathcal{E}$ es una aplicación

$$v : \mathrm{Var}(\mathcal{L}) \to D,$$

donde $\mathrm{Var}(\mathcal{L})$ es el conjunto de variables de $\mathcal{L}$ y D es el dominio de $\mathcal{E}$.

Toda asignación se extiende de forma única al conjunto de los términos del lenguaje $\mathcal{L}$ de la siguiente manera:

$$v' : Term(\mathcal{L}) \to D$$

definida como

$$v'(t) = \begin{cases} c & \text{si } t = c \in \mathrm{Cons}(\mathcal{L}), \\ v(x) & \text{si } t = x \in \mathrm{Var}(\mathcal{L}), \\ f(v'(t_1), \ldots, v'(t_n)) & \text{si } t = f(t_1, \ldots, t_n), \end{cases}$$

(aquí ya hemos identificado $c^{\mathcal{E}}$ con c y $f^{\mathcal{E}}$ con f). De esta forma a cada término del lenguaje $\mathcal{L}$ le podemos asignar un valor en D. A partir de ahora, vamos a identificar v con su extensión v', ya que v' queda unívocamente determinada una vez se conoce v.

Dada la asignación v, definimos $v(x|a)$ como la asignación que actúa sobre todas las variables como lo hace v, excepto posiblemente sobre x, a la cual le asigna el valor $a \in D$. De igual forma se define $v(x_1|a_1, \ldots, x_n|a_n)$, que sería igual que v en las variables $y \notin \{x_1, \ldots, x_n\}$ y $v(x_1|a_1, \ldots, x_n|a_n)(x_i) = a_i$ para $i \in \{1, \ldots, n\}$.

Estamos ya en condiciones de definir formalmente lo que vamos a entender por interpretación de una fórmula, que no es otra cosa que una extensión "natural" de las interpretaciones que vimos en el capítulo anterior para la lógica proposicional.

Una **$\mathcal{L}$-interpretación** es un par $(\mathcal{E}, v)$, con $\mathcal{E}$ una $\mathcal{L}$-estructura y v una asignación para $\mathcal{E}$, junto con una aplicación

$$\mathrm{I}^v : \mathrm{Form}(\mathcal{L}) \to \mathbb{Z}_2,$$

que verifica las siguientes propiedades.

1. $\mathrm{I}^v(R(t_1, \ldots, t_n)) = R(v(t_1), \ldots, v(t_n))$, con $t_1, \ldots, t_n$ términos, R un símbolo de relación n-ario de $\mathcal{L}$ (la primera R que aparece es un símbolo de relación del lenguaje $\mathcal{L}$, mientras que la segunda es la relación n-aria $R^{\mathcal{E}}$).
2. $\mathrm{I}^v(\neg\varphi) = 1 + \mathrm{I}^v(\varphi)$, para toda fórmula φ.
3. $\mathrm{I}^v(\varphi \vee \psi) = \mathrm{I}^v(\varphi) + \mathrm{I}^v(\psi) + \mathrm{I}^v(\varphi)\mathrm{I}^v(\psi)$, para cualesquiera fórmulas φ y ψ.
4. $\mathrm{I}^v(\varphi \wedge \psi) = \mathrm{I}^v(\varphi)\mathrm{I}^v(\psi)$, para cualesquiera fórmulas φ y ψ.
5. $\mathrm{I}^v(\varphi \to \psi) = 1 + \mathrm{I}^v(\varphi) + \mathrm{I}^v(\varphi)\mathrm{I}^v(\psi)$, para cualesquiera fórmulas φ y ψ.

6.

$$\mathrm{I}^v(\forall x\varphi) = \begin{cases} 1, & \textit{si para todo } a \in D \textit{ se verifica que } \mathrm{I}^{v(x|a)}(\varphi) = 1. \\ 0, & \textit{en caso contrario.} \end{cases}$$

7.

$$\mathrm{I}^v(\exists x\varphi) = \begin{cases} 1, & \textit{si existe } a \in D \textit{ tal que } \mathrm{I}^{v(x|a)}(\varphi) = 1. \\ 0, & \textit{en caso contrario.} \end{cases}$$

Los apartados dos al quinto de la definición recogen la idea de cómo se deben interpretar las conectivas. Obsérvese que es exactamente igual a la definición de interpretación en lógica proposicional. El caso base es el primer apartado, al igual que lo eran las proposiciones atómicas en la lógica proposicional. Lo que viene a decir el caso base es que una fórmula atómica es cierta si y sólo si los elementos que son asignados a los argumentos que acompañan al símbolo de relación están relacionados mediante la relación que se le asocia a dicho símbolo de relación. Por último, los dos últimos apartados de la definición corresponden a la interpretación de los cuantificadores. El sexto apartado dice que $\forall x\varphi$ es cierta en la $\mathcal{L}$-estructura en la que nos encontramos y bajo una determinada asignación, si para cualquier valor que le asignemos a la variable x (de ahí el cambio de asignación de v por $v(x|a)$) la fórmula φ es cierta en la $\mathcal{L}$-estructura dada con la nueva asignación (nótese que esta nueva asignación es igual que la anterior salvo que a la x se le ha asignado el valor a). El séptimo apartado es análogo al anterior.

Ejemplo 15. Considera el lenguaje $\mathcal{L}$ definido por:

- $\mathrm{Cons}(\mathcal{L}) = \{a, b, c, d, e\}$,
- $\mathrm{Func}(\mathcal{L}) = \{f, g\}$ (f monario, g binario),
- $\mathrm{Rel}(\mathcal{L}) = \{P\}$ (P binario),

y la $\mathcal{L}$–estructura $\mathcal{E}$, dada por:

$$\begin{aligned} D &= \mathbb{Z}_5 \\ a &= 0, \quad b = 1, \quad c = 2, \quad d = 3 \quad \text{y} \quad e = 4 \\ g &= + \quad (\text{suma en } \mathbb{Z}_5) \quad \text{y} \quad f(x) = 2 \times x \\ P &= \{(0,0), (0,1), (0,2), (0,3), (0,4), \\ &\qquad (1,1), (1,2), (1,3), (1,4), (2,2), \\ &\qquad (2,3), (2,4), (3,3), (3,4), (4,4)\}. \end{aligned}$$

Responde a las siguientes cuestiones:

(a) Dada la asignación

$$\begin{aligned} v : \mathrm{Var}(\mathcal{L}) &\longrightarrow \mathbb{Z}_5 \\ x &\mapsto 0 \\ y &\mapsto 3 \end{aligned}$$

interpreta la fórmula:

$$\exists x(P(x,y) \wedge P(g(e,x), f(y)))$$

(b) Para cualquier asignación v en $\mathcal{E}$, calcula $\mathrm{I}^v(\varphi)$ donde φ es:
   (i) $P(a, f(b)) \vee P(c, f(d))$
   (ii) $\forall x \exists y(P(x,y) \to P(g(x,x), f(y)))$

□

Solución.

(a) Observemos en primer lugar que la variable x aparece ligada mientras que la variable y aparece libre. Por tanto la fórmula dada vendría a decir algo como:

"existe un elemento x en $\mathbb{Z}_5$ tal que $P(x,3)$ y $P(4+x, 2\times 3)$".

Puesto que a la hora de interpretar, la aritmética se está realizando en $\mathbb{Z}_5$, podemos reemplazar 2×3 por su valor reducido que es 1, con lo que la fórmula nos diría que:

"existe un elemento x en $\mathbb{Z}_5$ tal que $P(x,3)$ y $P(4+x,1)$".

Si tenemos en cuenta la forma cómo se interpreta el símbolo de predicado P, observamos que ésta no es casual; de hecho corresponde a una relación de orden total $\leq$ definida en $\mathbb{Z}_5$, de manera que $0 < 1 < 2 < 3 < 4$. Por tanto, la fórmula inicial significa:

"existe un elemento x en $\mathbb{Z}_5$ tal que $x \leq 3$ y $4 + x \leq 1$".

De todos los elementos de $\mathbb{Z}_5$ menores o iguales que 3, observamos que tanto $x = 1$ como $x = 2$ verifican además la segunda condición (de hecho son los únicos).

Así pues, la fórmula se interpreta en la estructura dada como VERDADERA.

(b) (i) La fórmula se interpreta como

$$\text{"}a \leq 2\times b \text{ ó } c \leq 2\times d\text{"},$$

es decir,

$$\text{"}0 \leq 2\times 1 \text{ ó } 2 \leq 2\times 3\text{"},$$

o bien

$$\text{"}0 \leq 2 \text{ ó } 2 \leq 1\text{"},$$

lo cual es cierto, por lo que la interpretación es VERDADERA.

(ii) Ahora estamos diciendo que:

"Para todo x existe y tal que, si $x \leq y$ entonces $x + x \leq 2y$",

lo cual es VERDADERO, pues para cada valor que tome x basta tomar el mismo valor para y.

□

Ejercicio 33. Sea la fórmula

$$\forall x \exists y P(f(a,x), y)$$

y la $\mathcal{L}$-estructura dada por

- $D = \mathbb{Q}$
- $a = 3/2$
- $f = +$ (suma)
- $g(q) = \frac{q}{4} + 5$
- $P = \leq$

1. Para una asignación arbitraria, interpreta la fórmula dada.
2. Para la asignación

$$\begin{aligned} v : \mathrm{Var}(\mathcal{L}) &\to \mathbb{Q} \\ x &\mapsto 1 \\ y &\mapsto 2 \end{aligned}$$

calcula $v(g(f(x, y)))$.

5. Validez y satisfacibilidad

Dada una fórmula, ésta puede ser cierta bajo cualquier interpretación, o ser cierta bajo interpretaciones en determinadas $\mathcal{L}$-estructuras. Recuérdese del tema anterior que entre las proposiciones distinguíamos tautologías, proposiciones satisfacibles, proposiciones refutables y contradicciones. El análogo a estos conceptos se recoge en las siguientes definiciones.

- Una fórmula φ de $\mathcal{L}$ es **válida** en una $\mathcal{L}$-estructura $\mathcal{E}$ si para toda asignación v en $\mathcal{E}$ se verifica que $\mathrm{I}^v(\varphi) = 1$.
- Una fórmula de φ de $\mathcal{L}$ es **universalmente válida** si es válida en toda $\mathcal{L}$-estructura.
- Una fórmula φ de $\mathcal{L}$ es **satisfacible** si existe una interpretación $(\mathcal{E}, v)$ tal que $\mathrm{I}^v(\varphi) = 1$. En este caso se dice que φ es satisfacible en $\mathcal{E}$.
- Una fórmula φ de $\mathcal{L}$ es **refutable** si $\neg\varphi$ es satisfacible.

El que una fórmula sea satisfacible en $\mathcal{E}$ no implica que sea válida en $\mathcal{E}$. Claramente la otra implicación sí se da.

Ejemplo 16. Sea φ la fórmula $\forall x R(x) \to R(a)$. Veamos que φ es una fórmula universalmente válida. Para ello consideramos una $\mathcal{L}$-interpretación $(\mathcal{E}, v)$. Tenemos que $\mathrm{I}^v(\varphi) = 1$ si y sólo si $1 + \mathrm{I}^v(\forall x R(x)) + \mathrm{I}^v(\forall x R(x))\mathrm{I}^v(R(a)) = 1$. Distinguimos dos casos: $\mathrm{I}^v(\forall x R(x)) = 0$ e $\mathrm{I}^v(\forall x R(x)) = 1$. En el primero nos queda que $\mathrm{I}^v(\forall x R(x) \to R(a)) = 1 + 0 + 0 \times 1 = 1$. Supongamos ahora que $I^v(\forall x R(x)) = 1$, esto es equivalente a que $\mathrm{I}^{v(x|b)}(R(x)) = 1$ para todo $b \in D$ (con D el dominio de la $\mathcal{L}$-estructura $\mathcal{E}$). En particular tomando b igual a a nos queda que $\mathrm{I}^{v(x|a)}R(x) = 1$ lo que hace que $R(a)$ sea cierta, de ahí que $\mathrm{I}^v(R(a)) = 1$ y por tanto $\mathrm{I}^v(\forall x R(x) \to R(a)) = 1 + 1 + 1 \times 1 = 1$.

Consideremos ahora la fórmula $\phi = R(a) \to \forall x R(x)$ e interpretémosla en cualquier $\mathcal{L}$-estructura $\mathcal{E}$ que cumpla que su dominio tiene un solo elemento. Si tomamos $(\mathcal{E}, v)$ una interpretación en esa $\mathcal{L}$-estructura, tenemos que $\mathrm{I}^v(\phi) = 1 + \mathrm{I}^v(R(a)) + \mathrm{I}^v(R(a))\mathrm{I}^v(\forall x R(x))$. Al tener el dominio de $\mathcal{E}$ un sólo elemento, es fácil deducir que $\mathrm{I}^v(\forall x R(x)) = \mathrm{I}^v(R(a))$ y de ahí que $\mathrm{I}^v(\phi) = 1 + \mathrm{I}^v(R(a)) +$

$\mathrm{I}^v(R(a))\mathrm{I}^v(R(a)) = 1$, por lo que la fórmula ϕ es válida en todas las $\mathcal{L}$-estructuras cuyo dominio tenga un solo elemento.

Tomemos ahora la siguiente $\mathcal{L}$-estructura $\mathcal{E}$ dada por:

- $D = \mathbb{N}$,
- $a = 0$,
- $R =$ ser igual a cero.

Sea v una asignación cualquiera sobre $\mathcal{E}$ e interpretemos la fórmula ϕ:

$$\begin{aligned}\mathrm{I}^v(\phi) &= 1 + \mathrm{I}^v(R(a)) + \mathrm{I}^v(R(a))\mathrm{I}^v(\forall x R(x)) \\ &= 1 + R(0) + R(0)\mathrm{I}^v(\forall x R(x)) = 1 + 1 + 1\mathrm{I}^v(\forall x R(x)).\end{aligned}$$

Esto hace que $\mathrm{I}^v(\phi)$ dependa del valor de $\mathrm{I}^v(\forall x R(x))$. Sabemos que $\mathrm{I}^v(\forall x R(x)) = 1$ si y sólo si $\mathrm{I}^{v(x|b)}(R(x)) = R(b) = 1$ para todo $b \in \mathbb{N}$, lo que sería cierto si todo número natural es igual a cero, pero ésto no ocurre. Por tanto, $\mathrm{I}^v(\forall x R(x)) = 0$ y de ahí $\mathrm{I}^v(\phi) = 0$. □

Ejercicio 34.

1. Prueba que $\forall x(P(x) \to P(x))$ es universalmente válida.
2. Prueba que $R(x, y)$ es satisfacible y refutable.
3. Prueba que $\forall x P(x) \to \exists x P(x)$ es universalmente válida.
4. Prueba que $\psi \wedge \neg\psi$ es una contradicción.
5. Prueba que $\varphi \to (\psi \to \varphi)$ es universalmente válida (haz lo mismo con los patrones de axioma del cálculo proposicional $\mathcal{A}_2$ y $\mathcal{A}_3$).

Ejercicio 35. Probar que la fórmula

$$\forall x P(x, f(x)) \wedge \forall y \neg P(y, y) \wedge \forall u \forall v \forall w((P(u, v) \wedge P(v, w)) \to P(u, w))$$

es satisfacible. Probar que solo puede ser satisfacible en estructuras con dominios no finitos.

6. Consecuencia Lógica

Al igual que hicimos en su momento con la lógica proposicional, nos disponemos a introducir el concepto de consecuencia lógica. La idea es básicamente la misma, pero en el caso de lenguajes de primer orden es un poco más complicada. Decíamos que una proposición era consecuencia lógica de un conjunto de proposiciones si el hecho de que todas esas proposiciones fuesen ciertas forzaba que la proposición dada fuese cierta. Por tanto, la traducción directa del concepto de consecuencia lógica es que una determinada fórmula es consecuencia lógica de un conjunto de fórmulas si, siempre que bajo una interpretación éstas sean ciertas, la fórmula dada también sea cierta bajo esa interpretación.

Dado un conjunto de fórmulas $\Gamma \cup \{\varphi\}$, decimos que φ es **consecuencia lógica** de Γ, o bien que Γ **implica semánticamente** a φ, y lo notaremos $\Gamma \models \varphi$, si y sólo si para toda $\mathcal{L}$-interpretación $(\mathcal{E}, v)$ verificando que $\mathrm{I}^v(\gamma) = 1$ para todo $\gamma \in \Gamma$, se tiene que $\mathrm{I}^v(\varphi) = 1$.

Si $\Gamma = \emptyset$ entonces escribiremos $\models \varphi$ en lugar de $\emptyset \models \varphi$. El lector puede comprobar que $\models \varphi$ equivale a que φ es universalmente válida.

Ejemplo 17. Sean φ y ψ dos fórmulas. Veamos que $\{\varphi, \varphi \to \psi\} \models \psi$. Tomemos $(\mathcal{E}, v)$ una interpretación de forma que $\mathrm{I}^v(\varphi) = 1 = \mathrm{I}^v(\varphi \to \psi)$. Entonces $1 = 1 + 1 + \mathrm{I}^v(\psi)$, por lo que $\mathrm{I}^v(\psi) = 1$. Esto es, siempre que φ y $\varphi \to \psi$ sean ciertas bajo una determinada interpretación, entonces también lo es ψ. □

Ejercicio 36. Prueba que

$$\{\forall x(P(x) \to Q(x)), \neg Q(a)\} \models \neg P(a).$$

7. Teorema de la deducción. Lema de coincidencia

A continuación enunciamos el teorema de la deducción, cuya demostración dejamos al lector. Este teorema no necesita presentación, ya que su utilidad quedó patente en el capítulo dedicado a la lógica proposicional.

Resultado 10 (Teorema de la deducción.). *Dado $\Gamma \cup \{\varphi, \psi\}$ un conjunto de fórmulas. Son equivalentes:*

- $\Gamma \cup \{\varphi\} \models \psi$,
- $\Gamma \models \varphi \to \psi$.

Consideremos la fórmula $\forall x R(x, y)$. Si nos fijamos en la definición de interpretación de una fórmula cuantificada, nos damos cuenta de que, a la hora de interpretar, el valor que se le asigna a la x poco importa, ya que es substituido por todos los valores posibles del dominio o universo de la estructura en la que estamos interpretando. Lo realmente importante, aparte del significado que le asignemos a R, es el valor asignado a la variable libre y. Esta idea se recoge en el siguiente teorema.

Resultado 11 (Lema de coincidencia.). *Sea φ una fórmula tal que todas las variables que aparecen libres en ella son $x_1, \ldots, x_n$. Sea $\mathcal{E}$ una $\mathcal{L}$-interpretación. Son equivalentes:*

1. $\mathrm{I}^v(\varphi) = 1$,
2. *para toda asignación w, si $w' = w(x_1|v(x_1), \ldots, x_n|v(x_n))$, entonces $\mathrm{I}^{w'}(\varphi) = 1$.*

En una sentencia no hay variables libres, y por tanto si tenemos una interpretación y cambiamos los valores de las variables que queramos en la asignación asociada a la interpretación, no vamos a alterar los valores de las variables libres, ya que no las hay, y por tanto el valor de la interpretación de la sentencia dada no se altera. Este hecho se recoge en el siguiente corolario.

Consecuencia 4. *Dada una sentencia φ y $\mathcal{E}$ una $\mathcal{L}$-estructura, son equivalentes:*

1. *φ es válida en $\mathcal{E}$,*
2. *φ es satisfacible en $\mathcal{E}$.*

Si tenemos una interpretación, entonces una sentencia podrá ser cierta o no bajo esa interpretación. Si es cierta, eso quiere decir que es satisfacible en la $\mathcal{L}$-estructura donde estamos interpretando, y por el corolario anterior, es válida en esa estructura. Si por el contrario es falsa, entonces su negación es cierta, con lo que aplicando el mismo razonamiento anterior tenemos que su negación es válida en la $\mathcal{L}$-estructura.

CONSECUENCIA 5. *Dada una sentencia φ y $\mathcal{E}$ una $\mathcal{L}$-estructura, entonces es cierta una de las siguientes afirmaciones (y sólo una)*

1. *φ es válida en $\mathcal{E}$,*
2. *$\neg\varphi$ es válida en $\mathcal{E}$.*

Lo interesante de estos resultados es que la interpretación de una sentencia no depende del valor que asignemos a las variables, sino de la $\mathcal{L}$-estructura en sí.

EJERCICIO 37. Sea $\mathcal{E}$ una $\mathcal{L}$-estructura. Sea $\varphi \in \text{Form}(\mathcal{L})$. Si $x_1, \ldots, x_n$ son todas las variables que aparecen libres en φ, entonces equivalen:

1. φ es válida en $\mathcal{E}$,
2. $\forall x_1 \ldots \forall x_n \varphi$ es satisfacible en $\mathcal{E}$,
3. $\forall x_1 \ldots \forall x_n \varphi$ es válida en $\mathcal{E}$.

EJERCICIO 38. Sea $\mathcal{E}$ una $\mathcal{L}$-estructura. Sea $\varphi \in \text{Form}(\mathcal{L})$. Si $x_1, \ldots, x_n$ son todas las variables que aparecen libres en φ, entonces equivalen:

1. φ es satisfacible en $\mathcal{E}$,
2. $\exists x_1 \ldots \exists x_n \varphi$ es satisfacible en $\mathcal{E}$,
3. $\exists x_1 \ldots \exists x_n \varphi$ es válida en $\mathcal{E}$.

EJERCICIO 39. Demuestra que:

1. $\models \neg\forall x\psi \leftrightarrow \exists x\neg\psi$,
2. $\models \neg\exists x\psi \leftrightarrow \forall x\neg\psi$,
3. $\models \exists x\psi \leftrightarrow \neg\forall x\neg\psi$,
4. $\models \forall x\psi \leftrightarrow \neg\exists x\neg\psi$,
5. $\models \forall x\psi \wedge \varphi \leftrightarrow \forall x(\psi \wedge \varphi)$, si x no es libre en φ,
6. $\models \exists x\psi \wedge \varphi \leftrightarrow \exists x(\psi \wedge \varphi)$, si x no es libre en φ,
7. $\models \forall x\psi \vee \varphi \leftrightarrow \forall x(\psi \vee \varphi)$, si x no es libre en φ,
8. $\models \exists x\psi \vee \varphi \leftrightarrow \exists x(\psi \vee \varphi)$, si x no es libre en φ,
9. $\models \forall x\psi \wedge \forall x\varphi \leftrightarrow \forall x(\psi \wedge \varphi)$,
10. $\models \exists x\psi \vee \exists x\varphi \leftrightarrow \exists x(\psi \vee \varphi)$,
11. $\models \forall x\varphi(x) \leftrightarrow \forall y\varphi(y)$, y la variable y no aparece en $\varphi(x)$,
12. $\models \exists x\varphi(x) \leftrightarrow \exists y\varphi(y)$, y la variable y no aparece en $\varphi(x)$.

Cuando escribimos $\models \varphi \leftrightarrow \psi$, lo que estamos diciendo es que φ y ψ tienen el mismo valor de verdad bajo cualquier intrerpretación. En el tema de Lógica Proposicional decíamos que cuando esto ocurría, entonces φ y ψ eran

lógicamente equivalentes (o semánticamente equivalentes). Por extensión seguiremos usando la misma notación.

8. Inconsistencia

En lo sucesivo, nos vamos a centrar en el problema de intentar resolver el problema $\Gamma \models \varphi$. Para ello vamos a transformarlo ligeramente introduciendo el concepto de inconsistencia y viendo la relación que tiene con consecuencia lógica.

Un conjunto de fórmulas Γ se dice **inconsistente** si no existe $(\mathcal{E}, v)$ de forma que $(I^v)_*(\Gamma) \subseteq \{1\}$, es decir para toda interpretación I, existe $\varphi \in \Gamma$ tal que $I^v(\varphi) = 0$.

La siguiente proposición, cuya demostración es bastante sencilla (y que dejamos al lector), pone de manifiesto la relación entre consecuencia lógica e inconsistencia.

RESULTADO 12. *Sea* $\Gamma \cup \{\varphi\}$ *un conjunto de fórmulas. Equivalen:*

1. $\Gamma \models \varphi$,
2. $\Gamma \cup \{\neg\varphi\}$ *es inconsistente.*

En los próximos capítulos nos centraremos en transformar el conjunto de fórmulas en otros que conserven la inconsistencia y que sean más sencillos de estudiar. Daremos algoritmos para determinar, en algunos casos, la inconsistencia de esos conjuntos de fórmulas más sencillos, resolviendo así, de forma parcial, el problema de determinar si algo es consecuencia lógica o no de un conjunto de fórmulas dadas, a saber, si un determinado enunciado es una deducción lógica de un conjunto de hipótesis dadas.

Hacemos notar que existe una correspondiente teoría axiomática y sintáctica para la lógica de predicados de primer orden. El alumno interesado puede consultar la bibliografía.

9. Bibliografía específica

(1) Manuel Ojeda Aciego, Inmaculada Pérez de Guzmán Molina, Lógica para la computación (vol. 2, Lógica de primer orden), Ágora, 1997.
(2) Ignacio Jané, Ramón Jansana Ferrer, Calixto Badesa, Elementos de lógica formal, Ariel filosofía, 1998.
(3) Joaquín Aranda, José L. Fernández, José Jiménez, Fernando Morilla, Fundamentos de lógica matemática, Sanz y Torres , 1999.

CAPíTULO 4

Formas normales

1. Introducción

En el capítulo anterior vimos como $\Gamma \models \varphi$ es equivalente a que $\Gamma \cup \{\neg\varphi\}$ sea inconsistente. En el presente capítulo nos proponemos hacer transformaciones en un conjunto de fórmulas, de manera que dichas transformaciones no alteren el carácter de inconsistencia del conjunto que estamos transformando, y con la ventaja adicional de que las transformaciones dan lugar a fórmulas más "simples" que las del conjunto de partida.

2. Forma normal prenexa

La primera de las mencionadas transformaciones va a consistir en "extraer" los cuantificadores de forma que éstos queden al principio. Conseguiremos así un conjunto de fórmulas que estén en forma prenexa.

Una fórmula φ está en **forma normal prenexa** si es de la forma $C_1x_1 \dots C_nx_nM$, donde $C_i = \forall$ ó $\exists$ y M es una fórmula sin cuantificadores.

Vamos a ver que siempre es posible encontrar, para cualquier fórmula dada, una fórmula que sea equivalente (en el sentido que a continuación vamos a precisar) y que esté en forma prenexa. De este modo podremos transformar nuestro conjunto de partida en un conjunto de fórmulas que estén en forma prenexa.

Recuérdese que dos fórmulas, φ y ψ, son semánticamente equivalentes si y sólo si $\models \varphi \leftrightarrow \psi$, y lo denotábamos por $\varphi \equiv \psi$ (esto es, φ y ψ son semánticamente equivalentes si y sólo si $I^v(\varphi) = I^v(\psi)$ para cualquier interpretación $(\mathcal{E}, v)$).

El siguiente teorema nos asegura que la transformación antes mencionada se puede llevar a cabo.

Resultado 13. *Para toda fórmula φ existe otra fórmula φ^* en forma normal prenexa semánticamente equivalente a φ.*

La demostración de este teorema se hace usando el Ejercicio 39 del tema anterior. Ese ejercicio da un método para calcular la forma prenexa equivalente a una fórmula dada. Los cuatro primeros apartados de dicho ejercicio nos permiten alternar los cuantificadores con las negaciones, extrayendo así los cuantificadores y dejando dentro a las negaciones. Los apartados quinto al octavo nos permiten ir extrayendo cuantificadores. Nótese que si no se da que x no sea libre en φ, podemos usar los apartados undécimo y duodécimo para

renombrar la variable x por otra que ni siquiera aparezca en φ, con lo que nos aseguramos que no sea libre en φ. Los apartados noveno y décimo son optimizaciones de los apartados quinto y octavo respectivamente.

El teorema anterior nos permite conservar el carácter de inconsistencia de nuestro conjunto de partida una vez que hayamos substituido las fórmulas que lo componen por fórmulas equivalentes en forma prenexa.

CONSECUENCIA 6. *Dado un conjunto de fórmulas* Γ*, considérese el conjunto* Γ^* *formado al escoger una fórmula en forma normal prenexa para cada fórmula de* Γ*. Entonces* Γ *es inconsistente si y sólo si* Γ^* *es inconsistente.*

EJERCICIO 40. Sea φ una fórmula y x una variable sin ocurrencias libres en ella. Probar que $\models \varphi \leftrightarrow \forall x\varphi$ y $\models \varphi \leftrightarrow \exists x\varphi$.

El ejercicio anterior nos permite quitar cuantificadores innecesarios al calcular formas normales prenexas.

EJEMPLO 18. Sea φ la fórmula $\forall x\exists y\exists x R(x)$. Claramente φ ya está en forma normal prenexa pero usando el ejercicio anterior nos queda que φ es equivalente a la fórmula $\exists x R(x)$ que es más sencilla que la anterior y que será la forma normal prenexa de φ que usaremos.

EJERCICIO 41. Transforma las siguientes fórmulas en forma normal prenexa:

1. $\forall x(P(x) \to \exists y Q(x,y))$,
2. $\exists x(\neg\exists y P(x,y) \to (\exists z Q(z) \wedge R(x)))$,
3. $(\exists x\forall y P(a, f(x,y)) \to \exists x R(f(x,b))) \wedge P(b,a)$
4. $\forall x\forall y(\exists z P(x,y,z) \wedge (\exists u Q(x,u) \to \exists v Q(y,v)))$,
5. $\neg\exists x(P(x) \to \forall y Q(y))$,
6. $\forall x P(x) \to \neg\forall y\exists z(\forall y Q(y,z) \to \forall u\forall v\forall y R(u,v))$,
7. $R(x,y) \wedge (\exists y P(y) \to \forall x\forall w Q(y,x,w))$,
8. $\exists x(\forall x P(x) \to (R(x,a) \vee \forall y P(y))) \wedge \forall x\exists y(R(y, f(x)) \to P(x))$.

3. Forma normal de Skolem

La segunda transformación que vamos a hacer consiste en "eliminar" los cuantificadores existenciales de manera adecuada para que el conjunto resultante sea inconsistente si y sólo si lo es el de partida. Lo que vamos a introducir no es otra cosa que una de las formas normales de Skolem, la que conserva la inconsistencia. En principio no nos interesa más que el estudio de esta forma de Skolem, y para nosotros será la única que vamos a considerar.

Dada $\varphi \in \mathrm{Form}(\mathcal{L})$ en forma normal prenexa. Una **forma normal de Skolem** de φ es una fórmula obtenida al ir substituyendo cada variable x_i cuantificada existencialmente por $f(x_{i_1}, \ldots, x_{i_l})$, donde f es un símbolo de función con ariedad l que no aparece en φ ni ha sido usado hasta el momento para la substitución de otra variable, y $x_{i_1}, \ldots, x_{i_l}$ son las variables cuantificadas universalmente que preceden a $\exists x_i$ en la escritura de φ (de no haber ninguna, la substitución se hace por una constante a que no aparezca en φ).

Cuando vayamos a transformar fórmulas que estén en un conjunto dado en sus respectivas formas de Skolem, buscaremos los símbolos que aparecen en la definición de modo que no estén en ninguna de las otras fórmulas que aparecen en el conjunto. Este matiz viene justificado por el Ejercicio 43.

Ejemplo 19. Una forma de Skolem de $\exists x \forall y \exists z (R(x, y) \wedge Q(z, y, x))$, es

$$\forall y (R(a, y) \wedge Q(f(y), y, a)),$$

ya que a $\exists x$ no le precede ninguna variable cuantificada y a es un símbolo de constante que no aparece en φ y no ha sido usado hasta el momento; y porque a $\exists z$ le precede la variable cuantificada universalmente y, y el símbolo de función f no aparece en φ y no ha sido usado hasta el momento. □

Al "skolemizar" de esta forma perdemos la equivalencia semántica, pero la inconsistencia, que es lo que nos interesa ahora, se conserva.

Si tomamos una forma normal de Skolem de cada elemento de Γ^*, obtenemos el conjunto que vamos a denotar por Γ^{**}.

Resultado 14. *Dado un conjunto de fórmulas* Γ*, si construimos* Γ^{**} *como antes se ha indicado, entonces* Γ *es inconsistente si y sólo si* Γ^{**} *es inconsistente.*

No vamos a dar la demostración de este teorema, pero vamos a intentar motivarla poniendo un ejemplo. Supongamos que queremos probar que el conjunto $\{\forall x \exists y \varphi(x, y)\}$ es inconsistente (por $\varphi(x, y)$ vamos a entender una fórmula en la que las variables x e y aparecen libres). Si calculamos una forma de Skolem de $\forall x \exists y \varphi(x, y)$ tendremos $\forall x \varphi(x, f(x))$ (en $\varphi(x, y)$ hemos substituido todas las ocurrencias libres de y por $f(x)$). Está claro que si no hay ninguna interpretación para la que $\forall x \exists y \varphi(x, y)$ sea cierta, eso quiere decir que no hay ninguna para la que $\forall x \varphi(x, f(x))$ sea cierta. Si por el contrario $\forall x \exists y \varphi(x, y)$ fuese cierta bajo una interpretación dada I^v en una determinada estructura $\mathcal{E}$, eso quiere decir que para cada a en el dominio de la estructura existe un b_a (de esta forma indicamos la posible dependencia de a) en ese dominio tal que $\mathrm{I}^{v'}(\varphi(x, y))$ es cierta, con $v' = v(x|a, y|b_a)$. Como el símbolo de función f no aparece en la fórmula de partida, podríamos transformar la estructura de forma que para cada a, $f^{\mathcal{E}}(a)$ valga b_a, y dejando el resto de la estructura igual. De esta forma, $\forall x \varphi(x, f(x))$ es cierta en la nueva estructura bajo la misma interpretación. Nótese que la equivalencia semántica se pierde porque alteramos artificialmente la estructura para que la fórmula se haga cierta. De hecho, es fácil encontrar ejemplos de estructuras e interpretaciones donde las dos fórmulas tomen valores de verdad distintos, pero lo que sí aseguramos es que si la fórmula de partida es cierta en una determinada estructura bajo una determinada interpretación, podemos modificar la estructura haciendo que la forma de Skolem sea también cierta.

Ejercicio 42. Calcula las formas de Skolem de las soluciones del Ejercicio 41.

4. Forma clausular

La última transformación que vamos a hacer consiste en "partir" las fórmulas que tenemos en otras más simples. Para ello vamos a restringirnos a fórmulas que sean sentencias (recuérdese que una fórmula es sentencia si todas las ocurrencias de sus variables son ligadas). Aunque esta restricción parezca desmesurada, como ya comentamos al hablar de ocurrencias libres y ligadas, en el lenguaje hablado no se dan las variables libres, y por tanto normalmente partimos de un conjunto de sentencias a la hora de estudiar su inconsistencia. Los trozos en los que vamos a partir nuestras fórmulas (ya en forma de Skolem) van a ser cláusulas, concepto que definimos a continuación.

1. Un **literal** es una fórmula atómica o la negación de una fórmula atómica.
2. El **cierre universal** de una fórmula φ cuyas variables con ocurrencias libres son $x_1, \ldots, x_n$, es $\forall x_1 \cdots \forall x_n \varphi$.
3. Una **cláusula** es el cierre universal de una disyunción de literales.

Así pues, si C es una cláusula, entonces C es de la forma $\forall x_1 \cdots \forall x_n (L_1 \vee \ldots \vee L_m)$, con L_i literales. De ahora en adelante al escribir las cláusulas omitiremos los cuantificadores universales que las preceden, escribiendo de esta forma $L_1 \vee \ldots \vee L_m$ en vez de C, sin olvidarnos de que todas las variables están cuantificadas universalmente.

Dada una sentencia, podemos encontrar una fórmula equivalente a ella en forma normal prenexa. Claramente esta nueva fórmula también es una sentencia, y si calculamos una forma de Skolem asociada a ella, ésta sigue siendo una sentencia. Obtenemos así una fórmula del tipo $\forall x_1 \cdots \forall x_n M$, con M una fórmula libre de cuantificadores. Usando

- $\varphi \vee (\psi \wedge \chi) \equiv (\varphi \vee \psi) \wedge (\varphi \vee \chi)$, para cualesquiera φ, ψ, χ fórmulas;
- $\varphi \to \psi \equiv \neg\varphi \vee \psi$, para cualesquiera φ y ψ fórmulas;
- Leyes de De Morgan;

podemos transformar M en una fórmula equivalente a ella, de manera que ésta sea conjunción de disyunciones de literales, a saber,

$$M \equiv \bigwedge_i \Big(\bigvee_j L_{ij} \Big).$$

Si además tenemos en cuenta que

- $\forall x(\varphi \wedge \psi) \equiv \forall x \varphi \wedge \forall x \psi$, para cualesquiera φ y ϕ fórmulas, y x variable,

tenemos que $\forall x_1 \cdots \forall x_n M \equiv C_1 \wedge \cdots \wedge C_m$, con C_i cláusulas.

Dada φ una sentencia, sea φ^* una fórmula equivalente a ella en forma prenexa. Sea φ^{**} una forma de Skolem asociada a φ^*. Podemos encontrar como antes $C_1, \ldots, C_m$ verificando

$$\varphi^{**} \equiv \bigwedge_{i=1}^{m} C_i.$$

El conjunto $\{C_1, \ldots, C_m\}$ es una **forma clausular** de φ.

El motivo por el que tomamos el conjunto de las cláusulas en vez de la conjunción de ellas es porque, en realidad, a la hora de estudiar inconsistencia podemos partir las fórmulas que sean conjunción de otras fórmulas, tal como prueba el siguiente ejercicio.

EJERCICIO 43. Dado el confunto de fórmulas $\{\psi_1, \ldots, \psi_n\}$, son equivalentes:

- $\{\psi_1, \ldots, \psi_n\}$ es inconsistente.
- $\{\psi_1 \wedge \cdots \wedge \psi_n\}$ es inconsistente.

Usando este resultado, es fácil de demostrar el siguiente teorema.

RESULTADO 15. *Dado un conjunto de sentencias* Γ, *si consideramos el conjunto* Γ' *resultante de considerar para cada fórmula de* Γ, *una forma clausular suya, y luego tomar la unión de todas ellas, se tiene que* Γ *es inconsistente si y sólo si* Γ' *es inconsistente.*

Tenemos pues que si el conjunto de partida es un conjunto de sentencias, probar su insconsistencia es equivalente a probar la inconsistencia de un conjunto de cláusulas. La ventaja de este hecho es que las cláusulas son fórmulas de estructura sencilla.

Obsérvese también que en virtud del Ejercicio 43, cada vez que calculamos la forma de Skolem de una fórmula del conjunto de partida Γ, la elección de los nuevos símbolos de constante y de función, por la que substituiremos las variables que aparecen cuantificadas existencialmente, tiene que ser tal que éstos no aparezcan en el resto de las fórmulas de Γ.

EJEMPLO 20. Sea φ la fórmula $\exists z(\forall x \exists y R(x,a) \vee \neg \exists x \forall y (P(f(y),z) \vee R(x,y)))$. Los pasos que damos para calcular una de sus formas normales prenexas son los siguientes:

1. $\exists z(\forall x \exists y R(x,a) \vee \forall x \exists y (\neg P(f(y),z) \wedge \neg R(x,y)))$,
2. $\exists z \forall x \exists y (R(x,a) \vee \forall x \exists y (\neg P(f(y),z) \wedge \neg R(x,y)))$,
3. $\exists z \forall x \exists y (R(x,a) \vee \forall x_1 \exists y (\neg P(f(y),z) \wedge \neg R(x_1,y)))$,
4. $\exists z \forall x \exists y \forall x_1 \exists y (R(x,a) \vee (\neg P(f(y),z) \wedge \neg R(x_1,y)))$,

Podriamos decir que con el último paso ya hemos conseguido una forma normal prenexa de φ, pero el ejercicio 40 nos permite dar un paso más y obtener la fórmula

$$\exists z \forall x \forall x_1 \exists y (R(x,a) \vee (\neg P(f(y),z) \wedge \neg R(x_1,y)))$$

que es equivalente a φ, es más sencilla y sigue estando en forma normal prenexa. En este caso para calcular la forma de Skolem de esta última fórmula hemos de usar una constante y una función que no aparezcan en nuestras fórmulas. Suponiendo que ni b ni g han sido usadas, podemos decir que la fórmula

$$\forall x \forall x_1 (R(x,a) \vee (\neg P(f(g(x,x_1)),b) \wedge \neg R(x_1, g(x,x_1)))).$$

es una forma de Skolem de φ. Por último, la fórmula anterior es equivalente a

$$\forall x \forall x_1((R(x,a) \vee \neg P(f(g(x,x_1)),b)) \wedge (R(x,a) \vee \neg R(x_1, g(x,x_1)))),$$

de donde

$$\{R(x,a) \vee \neg P(f(g(x,x_1)),b), R(x,a) \vee \neg R(x_1, g(x,x_1))\}$$

es una forma clausular de φ.

EJERCICIO 44. Calcula la forma clausular de las soluciones del ejercicio 42.

CAPíTULO 5

Unificación y Resolución

1. Introducción

En temas anteriores abordamos el problema de encontrar un algoritmo para demostrar $\Gamma \models \varphi$. Para ello trasladamos dicho problema a demostrar la inconsistencia del conjunto de fórmulas $\Gamma \cup \{\neg\varphi\}$, el cual fuimos transformando sucesivamente en conjuntos que conservaban la propiedad de inconsistencia, llegando al final a un conjunto de cláusulas.

Nos proponemos en este tema dar algoritmos para determinar la inconsistencia de un conjunto de cláusulas. Éstos se basan en intentar *deducir* la cláusula vacía a partir del conjunto original de cláusulas, haciendo uso del principio de resolución como única regla de inferencia en nuestras deducciones.

2. El principio de resolución sin unificación

En esta sección introducimos el principio de resolución para un conjunto de cláusulas en su versión más elemental, que no es otra cosa que una extensión del modus ponens. Más adelante veremos que esta versión del principio de resolución no es aplicable muchas veces, debido a que los literales de las cláusulas se diferencian en algún término. Trataremos pues de salvar este obstáculo más adelante, siempre que sea posible, y dar así el principio de resolución para la lógica de primer orden.

Considérense las siguientes cláusulas (donde L_1^C es el único literal tal que $\neg L_1^C \equiv L_1$):

$$C_1 \equiv L_1$$
$$C_2 \equiv L_1^C \vee L_2$$

La cláusula C_2 se puede escribir como $L_1 \rightarrow L_2$. Por tanto, haciendo uso de modus ponens, tenemos que $\{L_1, L_1 \rightarrow L_2\} \models L_2$. El siguiente resultado generaliza este hecho.

Resultado previo 2. *Dadas las cláusulas $L_1 \vee \cdots \vee L_n, L_1^C \vee L_2' \vee \cdots \vee L_m'$ se tiene que*

$$\{L_1 \vee \cdots \vee L_n, L_1^C \vee L_2' \vee \cdots \vee L_m'\} \models L_2 \vee \cdots \vee L_n \vee L_2' \vee \cdots \vee L_m'.$$

En lo sucesivo, si C es una cláusula y L es un literal que aparece en C, notaremos $C - L$ a la cláusula que tiene todos los literales de C excepto L. En el

caso en que $C = L$, a $C - L$ lo denotamos por $\square$, y nos referimos a ella como la cláusula vacía.

Ejemplo 21.

1. Sea $C \equiv R(x, y) \vee P(a, f(a, z))$, y sea $L \equiv P(a, f(a, z))$. La cláusula $C - L$ es la cláusula $R(x, y)$.
2. Si tomamos $C \equiv Q(a)$ y $L \equiv Q(a)$, entonces $C - L$ es la cláusula vacía.

$\square$

Principio de resolución sin unificación. Dadas dos cláusulas C_1 y C_2, y dos literales L_1 y L_2 de C_1 y C_2 respectivamente, de forma que L_1^C coincide con L_2. La cláusula $C_1 - L_1 \vee C_2 - L_2$ es una **resolvente** de C_1 y C_2.

Ejemplo 22. Sean las cláusulas

$$\begin{aligned} C_1 \equiv &\quad \neg P \vee Q \vee S \\ C_2 \equiv &\quad \neg Q \vee R. \end{aligned}$$

La única resolvente que existe de C_1 y C_2 es $\neg P \vee S \vee R$. $\square$

Ejemplo 23. Sean las cláusulas C_1 y C_2:

$$\begin{aligned} C_1 \equiv &\quad P \vee Q \\ C_2 \equiv &\quad P \vee R. \end{aligned}$$

En este caso no existe resolvente de C_1 y C_2. $\square$

Por lo visto antes, se tiene el siguiente resultado.

Consecuencia 7. *Dadas dos cláusulas C_1 y C_2, una resolvente C de ambas es una consecuencia lógica de C_1 y C_2.*

Si $C_1 = L_1$ y $C_2 = L_2 = L_1^C$, entonces $C_1 - L_1 \vee C_2 - L_2$ es la cláusula vacía. Nótese además que el conjunto $\{L_1, L_1^C\}$ es inconsistente.

Dado un conjunto de cláusulas $\Gamma \cup \{C\}$, una **deducción** de C a partir de Γ es una sucesión finita de cláusulas $C_1, \ldots, C_k$ tal que $C_k = C$ y $C_i (i < k)$ es, o bien un elemento de Γ, o bien una resolvente de dos cláusulas que le precedan.

Una deducción de $\square$ a partir de Γ se llama refutación, o prueba de Γ.

De ahora en adelante diremos que C se *deduce* o *deriva* de Γ si existe una deducción de C a partir de Γ. Por el corolario anterior se tiene que

Consecuencia 8. *Si la cláusula C se deduce del conjunto de cláusulas Γ entonces $\Gamma \models C$.*

La demostración de este hecho se hace teniendo en cuenta la definición de deducción y aplicando inducción para usar el corolario anterior. Además hay que tener presente el siguiente hecho: si $\square$ se deduce de Γ eso es porque existe una sucesión de cláusulas $C_1, \ldots, C_k = \square$, y por tanto $\square$ es resolvente de dos cláusulas C_i y C_j. Por la definición de resolvente se tiene a la fuerza que C_i y C_j son literales y que uno es la negación del otro. Sea L dicho literal, tenemos

pues que $\Gamma \models L$ y que $\Gamma \models L^C$, y por tanto Γ debe ser inconsistente. Se tiene así el siguiente resultado.

CONSECUENCIA 9. *Si* $\square$ *se deduce de* Γ, *entonces* Γ *es inconsistente.*

Por ahora, con la definición de resolvente que tenemos, el recíproco no es cierto. Para eso debemos introducir unificación.

EJEMPLO 24. Considérese el conjunto

$$\left.\begin{array}{ll} (1) & P \vee Q \vee R \\ (2) & \neg P \vee Q \vee R \\ (3) & \neg Q \vee R \\ (4) & \neg R \end{array}\right\} \Gamma.$$

De (1) y (2), podemos obtener una resolvente

$$(5) \quad Q \vee R$$

De (5) y (3) obtenemos

$$(6) \quad R$$

De (6) y (4) obtenemos

$$(7) \quad \square$$

Por tanto Γ es inconsistente. $\square$

EJEMPLO 25. Demostrar que $\{P \vee Q, P \vee \neg Q, \neg P \vee Q, \neg P \vee \neg Q\}$ es inconsistente.

(1)	$P \vee Q$	*hip.*
(2)	$P \vee \neg Q$	*hip.*
(3)	P	*P.R.*(1), (2)
(4)	$\neg P \vee Q$	*hip.*
(5)	$\neg P \vee \neg Q$	*hip.*
(6)	$\neg P$	*P.R.*(4), (5)
(7)	$\square$	*P.R.*(3), (6)

Donde *hip.* es abreviatura de *por hipótesis* y *P.R.*(n), (m) es abreviación de resolvente de las fórmulas (n) y (m). $\square$

EJERCICIO 45. Prueba que:

1. $\models ((R \to Q) \to R) \to R$,
2. $\models R \to (Q \to R)$,
3. $\models (R \to (Q \to P)) \to ((R \to Q) \to (R \to P))$,
4. $\models (\neg R \to \neg Q) \to ((\neg R \to Q) \to R)$.

3. Unificación

Supongamos que tenemos las cláusulas

$$\begin{aligned} C_1 &\equiv P(x), \\ C_2 &\equiv \neg P(a). \end{aligned}$$

En este caso no podemos aplicar el principio de resolución que vimos en la sección anterior, ya que no hay dos literales L_1 y L_2 de C_1 y C_2 de forma que $L_1 = L_2^C$. Nótese además que si en C_1 substituyésemos la variable x por el término a, entonces sí que sería posible encontrar una resolvente de las cláusulas resultantes.

Una **substitución** es una aplicación $\sigma : \mathrm{Var}(\mathcal{L}) \to \mathrm{Term}(\mathcal{L})$ (al igual que ocurría con las asignaciones podemos ver a σ como una aplicación $\sigma : \mathrm{Term}(\mathcal{L}) \to \mathrm{Term}(\mathcal{L})$). El objetivo en esta sección es, para un conjunto de literales, ver si se puede encontrar una substitución que haga que todas las imágenes de los literales de ese conjunto bajo la substitución sean iguales. En el ejemplo anterior, el conjunto de literales a unificar es $\{P(x), P(a)\}$, y si tomamos la substitución, σ, que aplica la variable x al término a y deja las demás variables inalteradas, tendremos que $\sigma(P(x)) = \sigma(P(a)) = P(a)$. En su momento convenimos no escribir los cuantificadores delante de las cláusulas, por lo que C_1 es en verdad $\forall x P(x)$. Nótese además que $\{\forall x P(x)\} \models P(a)$, y que en consecuencia se tiene que $\{C_1, C_2\} \models \neg P(a)$ y $\{C_1, C_2\} \models P(a)$, y que por tanto $\{C_1, C_2\}$ es inconsistente.

Por convenio denotaremos a la substitución

$$\begin{aligned} &x_1 \mapsto t_1, \\ &\vdots \\ &x_n \mapsto t_n, \\ &y \mapsto y, \text{ si } y \neq x_1, \ldots, x_n, \end{aligned}$$

(con x_i variables del lenguaje $\mathcal{L}$ y t_j términos del mismo lenguaje) como

$$(x_1|t_1, \ldots, x_n|t_n),$$

y a la substitución identidad como ε.

Sea $\sigma = (x_1|t_1, \ldots, x_n|t_n)$ una substitución y C una cláusula. $\sigma(C)$ es la cláusula obtenida de substituir simultáneamente cada ocurrencia de la variable x_i, $i \in \{1, \ldots, n\}$, por el término t_i en C.

Obsérvese que $\{C\} \models \sigma(C)$.

Ejemplo 26. Sean $\sigma = (x|f(a,b), y|z)$ y $C = P(x, g(b), y, g(z))$, entonces $\sigma(C) = P(f(a,b), g(b), z, g(z))$. □

Ejemplo 27. Sean $\sigma = (x|y, y|x)$ y $C = P(x, g(b), y, g(z))$, entonces $\sigma(C) = P(y, g(b), x, g(z))$ y $\sigma^2(C) = C$. □

Una substitución σ se dice que es un **unificador** para el conjunto de literales $\Gamma = \{L_1, \ldots, L_n\}$ si $\#\sigma_*(\Gamma) = 1$, a saber, $\sigma(L_1) = \cdots = \sigma(L_n)$. En tal caso diremos que Γ es **unificable**.

Una vez que tenemos un unificador para un conjunto de literales, en general, podremos construir a partir de él infinitos unificadores para dicho conjunto de literales. De entre todos los posibles unificadores habrá algunos que son más simples que el resto.

Un unificador σ se dice **unificador principal** si para cualquier otro unificador θ existe una substitución λ de forma que $\theta = \lambda\sigma$.

Ejemplo 28. El conjunto $\{P(a), P(x)\}$ es unificable por el unificador $(x|a)$, el cual es el único unificador existente y por tanto principal. □

Ejercicio 46. ¿Cuáles de los siguientes unificadores son principales para el conjunto $\Gamma = \{P(x), P(y)\}$?: $\sigma_1 = (y|x)$, $\sigma_2 = (x|z, y|z)$ y $\sigma_3 = (x|a, y|a)$.

A continuación exponemos dos métodos distintos para decidir si un conjunto de literales es o no unificable.

3.1. Algoritmo de unificación. El problema que nos planteamos en esta sección es el de dar respuesta a la pregunta de si un determinado conjunto de literales es o no unificable. La utilidad de la resolución de este problema se dejó entrever al principio de la sección anterior (de hecho, nos basta con resolver el problema para dos literales).

Nótese que en el Ejemplo 28 el conjunto de términos que hace que los dos literales sean distintos es $\{x, a\}$. La definición que damos a continuación recoge esta idea.

El **conjunto de discordancia** D de un conjunto no vacío de literales W se obtiene localizando el primer símbolo (contando de izquierda a derecha) en el cual algunos de los literales de W deja de ser igual al resto, y luego extrayendo las subexpresiones de cada uno de los literales de W que empiezan en dicho símbolo.

Nos encontramos ya en condiciones de exponer el algoritmo de unificación, que es una herramienta para decidir si un conjunto de literales es o no unificable y, en caso de que lo sea, proporciona un unificador principal para dicho conjunto.

Algoritmo de unificación.

1. Hágase $k = 0$, $W_k = W$ (el conjunto de literales dado) y $\sigma_k = \varepsilon$.
2. Si $\#W_k = 1$ (W_k es un conjunto unitario), entonces σ_k es un unificador principal para W, y por tanto finaliza el algoritmo. En caso contrario calcúlese el conjunto de discordancia, D_k, de W_k.
3. Si hay elementos v_k y t_k en D_k tal que v_k es una variable que no aparece en el término t_k, ejecútese el paso *4*. En otro caso, el conjunto W no es unificable, y por tanto finaliza el algoritmo.
4. Hágase $\sigma_{k+1} = (v_k|t_k)\sigma_k$ y $W_{k+1} = (v_k|t_k)_*(W_k)$.
5. Increméntese k y ejecútese el paso 2.

EJEMPLO 29. Determinar si el conjunto

$$W = \{P(x, y), P(f(z), x), P(u, f(x))\}$$

es unificable, y en caso de serlo determinar un unificador más general para dicho conjunto.

1. $W_0 = W$, $\sigma_0 = \varepsilon$.
2. $\#W_0 \neq 1$. $D_0 = \{x, f(z), u\}$.
3. x no aparece en $f(z)$.
4. $\sigma_1 = (x|f(z))$,
$$W_1 = \{P(f(z), y), P(f(z), f(z)), P(u, f(f(z)))\}.$$
5. $\#W_1 \neq 1$. $D_1 = \{u, f(z)\}$.
6. u no aparece en $f(z)$.
7. $\sigma_2 = (u|f(z))(x|f(z)) = (x|f(z), u|f(z))$,
$$W_2 = \{P(f(z), y), P(f(z), f(z)), P(f(z), f(f(z)))\}.$$
8. $\#W_2 \neq 1$. $D_2 = \{y, f(z), f(f(z))\}$.
9. y no aparece en $f(z)$.
10. $\sigma_3 = (y|f(z))\sigma_2 = (x|f(z), u|f(z), y|f(z))$,
$$W_3 = \{P(f(z), f(z)), P(f(z), f(z)), P(f(z), f(f(z)))\}.$$
11. $\#W_3 \neq 1$. $D_3 = \{z, f(z)\}$.
12. W no es unificable por aplicación del paso 3 del algoritmo.

□

EJEMPLO 30. Hacer lo mismo para para $W = \{P(a, x, f(g(y))), P(z, f(z), f(u)\}$.

1. $W_0 = W$, $\sigma_0 = \varepsilon$.
2. $\#W_0 \neq 1$. $D_0 = \{a, z\}$.
3. z no aparece en a.
4. $\sigma_1 = (z|a)$,
$$W_1 = \{P(a, x, f(g(y))), P(a, f(a), f(u))\}.$$
5. $\#W_1 \neq 1$. $D_1 = \{x, f(a)\}$.
6. x no aparece en $f(a)$.
7. $\sigma_2 = (x|f(a))(z|a) = (x|f(a), z|a)$,
$$W_2 = \{P(a, f(a), f(g(y))), P(a, f(a), f(u))\}.$$
8. $\#W_2 \neq 1$. $D_2 = \{u, g(y)\}$.
9. u no aparece en $g(y)$.
10. $\sigma_3 = (u|g(y))\sigma_2 = (x|f(a), z|a, u|g(y))$,
$$W_3 = \{P(a, f(a), f(g(y)))\}.$$
11. $\#W_3 = 1$, y por tanto W es unificable. σ_3 es el unificador más general.

□

El siguiente teorema da sentido a lo que venimos haciendo en esta sección.

RESULTADO 16 (Teorema de Unificación). *Si W es un conjunto finito de literales no vacío y unificable entonces el algoritmo de unificación se para en el segundo paso y el último σ_k es un unificador más general para W.*

EJERCICIO 47. Decidid si son unificables o no las siguientes parejas de literales y si lo son dar un unificador principal:

1. $W_1 = \{R(x, f(x)), R(f(y), y)\}$
2. $W_2 = \{R(x_1, y_1, z_1), R(f(g(x, y)), g(v, w), y)\}$

3.2. Sistemas de ecuaciones en términos. Dados los literales L_1 y L_2, para que sean unificables es necesario que ambos vayan negados o bien ambos sin negar, y que además empiecen por el mismo símbolo de relación. Esto es, L_1 y L_2 deben ser, o bien

$$L_1 = R(t_1, \ldots, t_n); L_2 = R(t'_1, \ldots, t'_n),$$

o bien

$$L_1 = \neg R(t_1, \ldots, t_n); L_2 = \neg R(t'_1, \ldots, t'_n).$$

Queda claro pues que L_1 y L_2 son unificables si y sólo si existe σ tal que $\sigma(t_i) = \sigma(t'_i)$ para todo $i = 1, \ldots, n$. Esta idea da lugar a la siguiente definición y forma de resolver el problema de la unificación.

Una **ecuación** e en un lenguaje $\mathcal{L}$ es una pareja de términos de $\mathcal{L}$. Es decir, (t_1, t_2) con $t_1, t_2 \in \text{Term}(\mathcal{L})$. La ecuación (t_1, t_2) la escribiremos como $t_1 = t_2$.

EJEMPLO 31.

- La ecuación (x, y) que la escribimos como $x = y$.
- La ecuación $(f(x), g(x, y))$ que la escribimos como $f(x) = g(x, y)$.

□

Una **solución** de una ecuación (t_1, t_2) es una substitución σ que cumple $\sigma(t_1) = \sigma(t_2)$.

EJEMPLO 32. Las substituciones $(x|f(x), y|f(x))$, $(x|y)$ y $(y|x)$ son soluciones de la ecuación $x = y$. Sin embargo, la ecuación $f(x) = g(x, y)$ no tiene solución. □

Un **sistema de ecuaciones** E es un conjunto finito de ecuaciones. Es decir,

$$E = \{e_1, e_2, \ldots, e_n\}.$$

EJEMPLO 33. $\{x = y,\ f(x) = g(x, y)\}$ es un sistema de dos ecuaciones. □

Una **solución** de un sistema de ecuaciones es una substitución σ que es solución de cada una de las ecuaciones del sistema. Dos sistemas de ecuaciones son equivalentes si tienen las mismas soluciones.

EJEMPLO 34. La substitución $(x|a, y|a)$ es solución del sistema $\{x = y,\ y = a\}$ (de hecho es la única). □

Un sistema $E = \{e_1, e_2, \ldots, e_n\}$ está en forma **resuelta** si cada e_i es del tipo $x_i = t_i$ con $x_i \neq x_j$ para $i \neq j$ y además

$$\{x_1, x_2, \ldots, x_n\} \cap (\bigcup \{var(t_k) : \ 1 \leq k \leq n\}) = \emptyset.$$

Si $E = \{e_1, e_2, \ldots, e_n\}$ está en forma resuelta llamamos σ_E a la substitución $(x_1|t_1, x_2|t_2, \ldots, x_n|t_n)$.

Resultado 17. *Si E está en forma resuelta, entonces σ_E es un unificador principal.*

En este caso diremos que σ_E es una solución principal, un unificador principal o un unificador de máxima generalidad de E. A continuación damos una serie de teoremas que nos permiten transformar un sistema de ecuaciones en otro equivalente a él, de forma que, tras un número finito de transformaciones lleguemos a un sistema en forma resuelta o a un sistema incompatible (sin solución).

Resultado 18. *Los sistemas $E \cup \{t = t\}$ y E son equivalentes.*

La demostración de este teorema es trivial. Con este resultado eliminamos de un sistema las ecuaciones de la forma $t = t$, que son redundantes.

Resultado 19. *Los sistemas $E \cup \{t_1 = t_2\}$ y $E \cup \{t_2 = t_1\}$ son equivalentes.*

El uso de este teorema es ir colocando las variables en la parte izquierda de las igualdades, ya que para que el sistema esté en forma resuelta, todas sus ecuaciones deben verificar eso.

Resultado 20. *Los sistemas $E \cup \{f(t_1, \ldots, t_n) = f(u_1, \ldots, u_n)\}$ y $E \cup \{t_1 = u_1, \ldots, t_n = u_n\}$ son equivalentes.*

La demostración de este teorema es trivial. Está claro que para resolver la ecuación $f(t_1, \ldots, t_n) = f(u_1, \ldots, u_n)$, hay que resolver el sistema de ecuaciones $\{t_1 = u_1, \ldots, t_n = u_n\}$.

Resultado 21. *Los sistemas $E \cup \{x = t\}$, con $x \notin Var(t)$ y t un término que no es una variable, y $\sigma(E) \cup \{x = t\}$ con $\sigma = (x|t)$ son equivalentes.*

Este teorema nos dice que cuando tenemos una ecuación de la forma $x = t$ de forma que en t no aparece la variable x, podemos substituir en el resto de ecuaciones del sistema la variable x por el término t.

Resultado 22. *El sistema $E \cup \{f(t_1, \ldots, t_n) = g(u_1, \ldots, u_m)\}$ con $f \neq g$ no tiene solución (el sistema es incompatible).*

Es imposible encontrar σ tal que $\sigma(f(t_1, \ldots, t_n)) = \sigma(g(u_1, \ldots, u_m))$.

Resultado 23. *El sistema $E \cup \{f(t_1, \ldots, t_n) = a\}$ con a un símbolo de constante no tiene solución (el sistema es incompatible).*

Resultado 24. *El sistema $E \cup \{x = t\}$ con $x \in Var(t)$ y $t \neq x$ no tiene solución.*

Para cualquier substitución σ, $\sigma(x)$ será distinto a $\sigma(t)$, ya que $\sigma(t)$ es un término que contiene a $\sigma(x)$ y que es "más grande" que éste.

RESULTADO 25. *Para todo sistema E' compatible (con solución) existe un sistema equivalente E que está en forma resuelta.*

EJEMPLO 35. Determinar si el conjunto $W = \{P(a, x, f(g(y))), P(z, f(z), f(u))\}$ es unificable o no lo es. Caso de serlo encontrar un unificador principal.

El sistema de ecuaciones asociado a este problema es

$$\left.\begin{array}{l} a = z \\ x = f(z) \\ f(g(y)) = f(u) \end{array}\right\}$$

que es equivalente a

$$\left.\begin{array}{l} z = a \\ x = f(z) \\ f(g(y)) = f(u) \end{array}\right\}$$

el cual equivale a

$$\left.\begin{array}{l} z = a \\ x = f(a) \\ f(g(y)) = f(u) \end{array}\right\}$$

equivalente a

$$\left.\begin{array}{l} z = a \\ x = f(a) \\ g(y) = u \end{array}\right\}$$

y por último equivalente a

$$\left.\begin{array}{l} z = a \\ x = f(a) \\ u = g(y) \end{array}\right\}$$

que está en forma resuelta, y por tanto el unificador principal (y solución del sistema) es

$$(z|a, x|f(a), u|g(y)).$$

□

EJEMPLO 36. Determinar si el conjunto

$$W = \{P(x, y), P(f(z), x), P(u, f(x))\}$$

es unificable o no, y en caso de serlo encontrar un unificador principal.

Un sistema de ecuaciones asociado a este problema es:

$$\left.\begin{array}{l} x = f(z) \\ y = x \\ x = u \\ y = f(x) \end{array}\right\}$$

Este sistema es equivalente a

$$\left.\begin{array}{l} x = f(z) \\ y = f(z) \\ f(z) = u \\ y = f(f(z)) \end{array}\right\}$$

que es equivalente a

$$\left.\begin{array}{l} x = f(z) \\ y = f(z) \\ u = f(z) \\ y = f(f(z)) \end{array}\right\}$$

equivalente a

$$\left.\begin{array}{l} x = f(z) \\ y = f(z) \\ u = f(z) \\ f(z) = f(f(z)) \end{array}\right\}$$

que por último es equivalente a

$$\left.\begin{array}{l} x = f(z) \\ y = f(z) \\ u = f(z) \\ z = f(z) \end{array}\right\}$$

el cual al contener la ecuación $z = f(z)$, no tiene solución. Por tanto el conjunto W no es unificable. □

EJERCICIO 48. Resolver el sistema:

$$\left\{\begin{array}{l} x_1 = f(g(x, y)) \\ y_1 = g(v, w) \\ z_1 = y \\ f(z) = x_1 \\ y_1 = x \\ f(x) = z_1 \end{array}\right.$$

EJERCICIO 49. Calcular un unificador principal de los literales siguientes:

$$C = P(v, m(f(a, w)), h(y, f(x, u), u))$$

$$D = P(m(x), v, h(g(u), z, f(g(w), g(a))))$$

4. El principio de resolución

En esta sección generalizaremos el principio de resolución que se dio al principio de este tema. Para ello es necesario que demos unas definiciones previas.

Sean C_1 y C_2 dos cláusulas *sin variables comunes*, L_1 y L_2 dos literales de C_1 y C_2 respectivamente. Si L_1 y L_2^C tienen un unificador más general, σ, entonces la cláusula

$$(\sigma(C_1) - \sigma(L_1)) \vee (\sigma(C_2) - \sigma(L_2))$$

es una **resolvente binaria** de C_1 y C_2. En tal caso se dice que los literales L_1 y L_2 son los literales sobre los que se resuelve.

Como C_1 y C_2 son sentencias, podemos renombrar las variables de forma que C_1 y C_2 no tengan variables en común.

Ejemplo 37.

$$\left.\begin{array}{ll} C_1 \equiv & P(x,b) \vee Q(x,a) \\ C_2 \equiv & \neg P(a,x) \vee R(x). \end{array}\right\}$$

En C_1 y C_2 aparece x, y por tanto no podremos encontrar una resolvente binaria para ambas, en tanto no eliminemos este problema. Podemos substituir en C_2 x por z y la inconsistencia permanece inalterada. De esa forma obtendremos

$$\left.\begin{array}{ll} C_1 \equiv & P(x,b) \vee Q(x,a) \\ C_2 \equiv & \neg P(a,z) \vee R(z). \end{array}\right\}$$

Tomando $L_1 = P(x,b), L_2 = \neg P(a,z)$ y $\sigma = (x|a, z|b)$, obtenemos una resolvente binaria $Q(a,a) \vee R(b)$. □

Ejercicio 50. Comprobar que el conjunto de cláusulas $\{P(x), \neg P(f(x))\}$ es inconsistente.

El conjunto de cláusulas $\Gamma = \{P(x) \vee P(y), \neg P(z) \vee \neg P(u)\}$ claramente es inconsistente ($\Gamma \models P(a)$ y $\Gamma \models \neg P(a)$). Sin embargo ninguna demostración por resolución que elimine únicamente un literal de cada cláusula puede producir la cláusula vacía. Por ello necesitamos ampliar el concepto de resolución. Si dos o más literales con el mismo signo de una cláusula C tienen un unificador más general, σ, entonces se dice que $\sigma(C)$ es un **factor** de C. También se suele decir que $\sigma(C)$ se obtiene a partir de C aplicando la regla de disminución. Si $\sigma(C)$ tiene un sólo literal diremos que es un **factor unitario** de C.

Ejemplo 38. Sea $C = P(x) \vee P(a) \vee Q(b)$ y $\sigma = (x|a)$. Entonces $\sigma(C) = P(a) \vee Q(b)$ es un factor de C. □

Nótese que si $\sigma(C)$ es un factor de C, entonces $\{C\} \models \sigma(C)$.

Una **resolvente** de dos cláusulas C_1 y C_2 es una de las siguientes resolventes binarias:

1. Una resolvente binaria de C_1 y C_2.

2. Una resolvente binaria de C_1 y un factor de C_2.
3. Una resolvente binaria de C_2 y un factor de C_1.
4. Una resolvente binaria de un factor de C_1 y un factor de C_2.

El principio de resolución es una regla de inferencia que consiste en generar resolventes a partir de un conjunto de cláusulas. La definición de *deducción*, etc. queda como antes. Así para demostrar la inconsistencia de un conjunto de cláusulas intentaremos encontrar una deducción de $\square$ a partir del conjunto dado. Téngase en cuenta que la definición de deducción podríamos haberla hecho del siguiente modo:

> Una deducción de C a partir de Γ es una sucesión finita de cláusulas $C_1, \ldots, C_k$ tal que $C_k = C$ y $C_i (i < k)$ es un elemento de Γ, un factor de una claúsula que la preceda o una resolvente binaria de dos cláusulas que la precedan.

Al usar resolventes y no resolventes binarias, nos evitamos el que la palabra factor intervenga en la definición de deducción.

EJEMPLO 39.

$$\left.\begin{array}{ll} C_1 \equiv & P(x) \vee P(f(y)) \vee R(g(y)) \\ C_2 \equiv & \neg P(f(g(a))) \vee Q(b) \end{array}\right\}.$$

Tomando $\sigma = \{x|f(y)\}$ para $L_1 = P(x), L_2 = P(f(y))$, se tiene que $\sigma(L_1) = \sigma(L_2) = L_2$. Por tanto $\sigma(C_1) = P(f(y)) \vee R(g(y)) = C_1'$ es un factor de C_1.

Si consideramos la substitución $\theta = \{y|g(a)\}$, tendremos que una resolvente de C_1 y C_2 es $R(g(g(a))) \vee Q(b)$.

Otra forma de resolver este ejercicio podría haber sido calcular un unificador principal σ para $P(x)$, $P(f(y))$ y $P(f(g(a)))$. $\square$

EJERCICIO 51. Dadas las siguientes claúsulas:

$$C_1 = Q(x) \vee \neg R(x) \vee P(x, y) \vee P(f(z), f(z)),$$

$$C_2 = \neg S(u) \vee \neg R(w) \vee \neg P(f(a), f(a)) \vee \neg P(f(w), f(w)).$$

Obtener una resolvente lo más reducida posible, es decir, con el menor número de literales.

EJERCICIO 52. Encontrar resolventes para las siguientes parejas de cláusulas:

1. $\{\neg P(x) \vee Q(x, b), P(a) \vee Q(a, b)\}$
2. $\{\neg P(x, y, u) \vee \neg P(y, z, v) \vee \neg P(x, v, w) \vee P(u, z, w), P(g(x, y), x, y)\}$

EJERCICIO 53. Si a partir de dos cláusulas C_1 y C_2 podemos calcular dos resolventes distintas R_1 y R_2 bajo un mismo unificador σ, ¿qué podemos afirmar sobre R_1 y R_2? ¿Qué ocurre si el unificador empleado para obtener R_1 no es el mismo que el empleado para obtener R_2?

5. Completitud del principio de resolución

El principio de resolución es completo, es decir, siempre se puede deducir la cláusula vacía de un conjunto inconsistente de cláusulas. Este hecho queda reflejado en el siguiente teorema.

RESULTADO 26 (Completitud del principio de resolución). *Un conjunto de cláusulas* Γ *es inconsistente si y sólo si hay una deducción de* $\square$ *a partir de* Γ.

Por tanto, una forma de determinar la inconsistencia de un conjunto de cláusulas es encontrar una deducción de la cláusula vacía a partir del conjunto inicial. En la siguiente sección exponemos una serie de estrategias para abordar este problema.

EJERCICIO 54. Comprueba que $\{\forall x Q(x) \vee \exists y P(y), \neg \forall y Q(y)\} \models \exists x P(x)$.

EJERCICIO 55. Comprueba que las siguientes fórmulas son universalmente válidas aplicando refutación por resolución:

1. $\exists x R(x) \rightarrow R(a)$,
2. $\neg \exists x \forall y (S(x,y) \leftrightarrow \neg S(y,y))$,
3. $\forall x \exists y (Q(x,y) \vee \forall z \neg Q(x,z))$.

EJERCICIO 56. Demostrar mediante refutación por resolución las siguientes fórmulas de la lógica proposicional:

1. $(\neg P \rightarrow P) \rightarrow P$,
2. $((P \rightarrow Q) \rightarrow P) \rightarrow P$,
3. $(P \rightarrow Q) \rightarrow (\neg Q \rightarrow \neg P)$,
4. $(P \rightarrow Q) \rightarrow ((P \vee R) \rightarrow (Q \vee R))$.

EJERCICIO 57. Utilizar el teorema de completitud para demostrar que la fórmula $\forall x R(x) \vee \forall x Q(x) \rightarrow \forall x (R(x) \vee Q(x))$ es universalmente válida, mientras que la fórmula $\forall x (R(x) \vee Q(x)) \rightarrow \forall x R(x) \vee \forall x Q(x)$ no es universalmente válida.

EJERCICIO 58. Sea $\alpha = \forall x \forall y \forall z ((R(x,y) \wedge R(y,z)) \rightarrow R(x,z))$, sea $\beta = \forall x R(x,x)$ y sea $\gamma = \forall x \forall y (R(x,y) \rightarrow R(y,x))$. ¿Es el conjunto $\{\alpha, \beta, \neg\gamma\}$ inconsistente? (Observar que al obtener las formas clausulares correspondientes, el número posible de resolventes, salvo renombramiento de variables, es finito).

Podriamos pensar que todo problema relacionado con la teoría que hemos estudiado es solucionable mediante un método algorítmico, tal y como hemos visto en el teorema de completitud para decidir la incosistencia de un conjunto de cláusulas. Sin embargo ésto no es así como comentamos brevemente a continuación. En teoría de computabilidad, se dice que un problema es computable o decidible si existe un algoritmo (más tecnicamente llamado una máquina de Turing), el cual al recibir una entrada referente a dicho problema sintácticamente correcta, termina sus cálculos tras un número finito de pasos y da una respuesta correcta. Si no existe dicho método, decimos que el problema es no computable o indecidible.

RESULTADO 27. *(de Church) El problema de decidir cuando una fórmula es universalmente válida en la lógica de predicados indecidible.*

Por tal motivo se dice que la lógica de predicados es indecidible.

6. Estrategias de resolución

Estamos interesados en proponer métodos que decidan la posible inconsistencia de un conjunto de cláusulas. Diremos que un método de ese tipo es coherente o completo si siempre que el conjunto de entrada sea inconsistente, el método terminará detectándolo tras un número finito de pasos. Las estrategias de resolución que aquí se exponen se dividen en dos familias: las técnicas de gestión de conjuntos de cláusulas y las técnicas de exploración del árbol de deducciones.

6.1. Técnicas de gestión de conjuntos de cláusulas. El conjunto de estrategias que usan esta técnica parten de un conjunto inicial de cláusulas S_0 (sobre el cual se pretende demostrar inconsistencia o no), que se va enriqueciendo sucesivamente creando así una secuencia ascendente de conjuntos de cláusulas

$$S_0 \subseteq S_1 \subseteq \ldots \subseteq S_i \subseteq \ldots \subseteq S_n \ldots,$$

con la propiedad de que son inconsistentes si y sólo si lo es S_0.

La condición de parada en este tipo de estrategias vendrá impuesta, como es lógico, o bien por que S_n contenga a la cláusula vacía, o bien por que $S_i = S_{i+1}$. En el primero de los casos queda demostrada la inconsistencia de S_0, y en el segundo (dependiendo de la estrategia) quedará probada la consistencia. En general la estacionalidad no queda asegurada y por tanto estos algoritmos no tienen por qué acabar.

6.1.1. Estrategia de saturación. Esta estrategia consiste en calcular S_{i+1} a partir de S_i, tomando todas sus cláusulas y todas las resolventes que se puedan obtener de ellas.

EJEMPLO 40. $S_0 = \{A, \neg A \vee B, \neg B \vee A, \neg B\}$.
$S_1 = \{B, B \vee \neg B, \neg A \vee A, \neg A\} \cup S_0$.
$S_2 = \{A, \square, \ldots\} \cup S_1$. □

Esta estrategia es completa, a saber, si S_0 es inconsistente entonces acabamos por encontrar la cláusula vacía

En el caso en el que $S_i = S_{i+1}$, sin que $\square \in S_i$, entonces sabemos que S_0 es consistente, ya que $\square$ no se puede deducir de S_0.

Esta estrategia es terriblemente ineficiente debido a que el número de cláusulas aumenta de forma exponencial. Se puede obtener una misma cláusula varias veces además de muchas cláusulas inútiles de cara a obtener la cláusula vacía.

Otra posibilidad es que el conjunto de partida sea consistente, pero que a partir de él se puedan obtener infinitas resolventes distintas. La cuestión ahora es: ¿cuándo para el método? Lo razonable en este caso sería poner algun tipo de cota o profundidad que limite el máximo número de resolventes a generar. Es razonable pensar que cuanto mayor sea la profundidad que se alcanza sin haber obtenido la cláusula vacía, tanto mayor será la posibilidad de que el conjunto de partida sea consistente. Un ejemplo típico se puede ver con el conjunto

$$\{\neg P(x) \vee P(f(x)), P(a)\}$$

el cual es consistente (compruébese) y a partir del cual se pueden generar infinitas resolventes.

6.1.2. Estrategias de saturación con simplificación. Diremos que D generaliza a C si existe una substitución σ tal que $D \equiv \sigma(C) \vee E$ con E una cláusula.

Ejemplo 41. $P(f(a)) \vee Q(a)$ generaliza a $P(x)$, y $P(a) \vee Q(a) \vee R(f(x))$ generaliza a $P(x) \vee Q(a)$. □

Ejercicio 59. Sean las cláusulas

$$C = \neg P(x) \vee P(f(x)),\ D = \neg P(x) \vee P(f(f(x))).$$

Comprobar que C implica a D, pero D no generaliza a C.

Decimos que una cláusula es *tautológica* si contiene a un literal y a su complementario.

Ejemplo 42. La cláusula $R(x) \vee \neg Q(a,x) \vee \neg R(x)$ es tautológica, sin embargo $R(x) \vee \neg Q(a,x) \vee \neg R(y)$ no lo es.

La relación existente entre generalización e inconsistencia viene dada por la siguiente:

Proposición 1. *Un conjunto de cláusulas S es inconsistente si y sólo si el conjunto S' que se obtiene a partir de S quitando todas las cláusulas tautológicas y las generalizaciones es inconsistente.*

Esta observación nos permite simplificar los conjuntos para los que deseamos probar inconsistencia. Esta es la idea de la estrategia de saturación con simplificación. De cada S_i extraemos S_i' antes de calcular S_{i+1} y de esta forma los conjuntos no crecen de forma tan desorbitada.

Ejemplo 43. $S_0 = \{R(a) \vee P(a), P(x) \vee \neg R(b), R(b), \neg P(b), R(a) \vee R(c) \vee \neg R(a)\}$.
$S_0' = \{R(a) \vee P(a), P(x) \vee \neg R(b), R(b), \neg P(b)\}$.
$S_1 = S_0' \cup \{P(x), \neg R(b)\}$.
$S_1' = \{R(b), \neg P(b), P(x), \neg R(b)\}$.
$S_2 = \{\square, \ldots\}$. □

Esta estrategia es también completa.

6.1.3. Estrategias de preferencia de cláusulas simples. Esta estrategia consiste en construir S_{i+1} a partir de S_i adjuntando una sola cláusula, que es resolvente de dos cláusulas de S_i. La forma de elegir dicha cláusula da lugar a diferentes estrategias de preferencia de cláusulas simples. Entre las posibles elecciones caben destacar:

1. *Elección aleatoria.* Consiste en elegir una resolvente aleatoria de entre las posibles que se pueden obtener a partir de elementos de S_i.
2. *Elección de la cláusula más corta.* Se trata de elegir la resolvente con menor número de literales de entre todas las posibles. La cláusula vacía no tiene ningún literal, la idea de ir eligiendo la resolvente con menor número de literales se basa en intentar llegar al menor número posible de literales, a saber 0.

   EJEMPLO 44.

   $S_0 = \{R(x) \vee \neg P(x), \neg R(b),\ \neg R(c) \vee Q(x) \vee R(f(a)),\ P(b)\}$.

   $S_1 = S_0 \cup \{\neg P(b)\}$.
   $S_2 = S_1 \cup \{\square\}$. □

   Como es de esperar esta estrategia no es completa:

   EJEMPLO 45.

   $S_0 = \{\neg P(x) \vee P(f(x)), P(a),\ \neg P(a) \vee \neg P(b) \vee \neg P(c),\ \ P(b), P(c)\}$

   $S_1 = S_0 \cup \{P(f(a))\}$
   $S_2 = S_1 \cup \{P(f(f(a)))\}$
   etc. □

3. *Conjunto soporte.* Un teorema consta de un conjunto de axiomas A_1, $A_2, \ldots, A_n$ y una conclusión B. Como ya sabemos, para probar el teorema debemos de comprobar que el conjunto $\{A_1, A_2, \ldots, A_n, \neg B\}$ es inconsistente. Lo normal es que el conjunto $\{A_1, \ldots, A_n\}$ sea consistente, por lo que podemos ahorrarnos el calcular resolventes tomando ambas cláusulas padre de dicho conjunto. Esta es la idea de la estrategia del conjunto soporte.

   Un subconjunto T de un conjunto S de cláusulas es un conjunto soporte de S si $S - T$ es consistente. Las resoluciones permitidas serán aquellas en las cuales al menos una de las cláusulas no está en $S - T$.

   Se puede demostrar que esta estrategia es completa:

   RESULTADO 28. *Si S es un conjunto finito e inconsistente de cláusulas, y T es un subconjunto de S tal que $S - T$ es consistente, entonces existe una deducción de $\square$ a partir de S, con T como conjunto soporte.*

   EJERCICIO 60. Sea el conjunto de cláusulas

   $S = \{P(a), \neg D(y) \vee L(a, y), \neg P(x) \vee \neg Q(y) \vee \neg L(x, y), D(b), Q(b)\}$.

Probar que $T = \{D(b), Q(b)\}$ es un conjunto soporte para S, y usar la mencionada técnica para demostrar la inconsistencia de S.

4. *Preferencia de cláusulas unidad.* La resolvente elegida es cualquiera tal que una de las cláusulas tomadas para resolver tiene un solo literal (estas cláusulas con un solo literal reciben el nombre de *cláusulas unit*). Esta estrategia tampoco es completa. Basta ver el ejemplo 45.

Todas las estrategias que hemos comentado pueden ser mezcladas dando lugar a nuevas estrategias; en la mayor parte de los casos se pierde la completitud, pero a veces se conserva, dando lugar, en general, a algoritmos bastante complejos. A veces lo que se pretende es tener algoritmos que sean eficientes aunque no sean completos.

6.2. Técnicas de exploración del árbol de deducciones. Dichas técnicas son principios generales de resolución de problemas y de búsqueda que se aplican en campos muy diversos.

El árbol de deducciones asociado a un conjunto de cláusulas C es un árbol con raíz $\bullet$, donde toda rama partiendo de $\bullet$ y pasando por $f_1, \ldots, f_n$ es una deducción a partir de C (en el sentido de que, o bien $f_i \in C$, o bien es resolvente de f_j, f_k con $j, k < i$).

Las estrategias basadas en la exploración del árbol de deducciones buscan encontrar la cláusula vacía en dicho árbol, y se diferencian entre ellas precisamente en la forma de hacer esa búsqueda, y a veces en limitaciones impuestas a las deducciones, de forma que no se trabaje con el árbol de deducciones entero sino más bien con un subárbol conveniente. Así pues, una estrategia basada en la exploración del árbol de deducciones está fundada en dos elecciones:

- La elección de un subárbol del árbol de deducciones, definido por alguna limitación impuesta a las posibles resolventes.
- La elección de una técnica de exploración de dicho subárbol. Entre las técnicas a elegir, las principales son:
  - Exploración *primero en profundidad con vuelta a atrás.*
  - Exploración *primero en anchura.*

Existen ciertas ventajas de una técnica frente a otra. La ventaja de escoger una estrategia de exploración primero en anchura es que, siempre que el conjunto de cláusulas de partida sea inconsistente, encontraremos la cláusula vacía, incluso si el árbol de deducciones es infinito. Por contra, la desventaja principal es el gran gasto de memoria que necesita una implementación de un método basado en esta técnica. Por otro lado, la ventaja de elegir una estrategia de exploración primero en profundidad con vuelta atrás es la de que este método es fácilmente programable y no gasta tanta memoria. Sin embargo el problema principal con el que se encuentra esta segunda elección es el de la posibilidad de "perderse" en una rama infinita cuando la cláusula vacía se encuentra en alguna otra rama no explorada aún.

6.2.1. Estrategias lineales. Cuando tratamos de demostrar una identidad, solemos comenzar por la parte izquierda de la misma, aplicamos una regla de inferencia para obtener una nueva expresión, a la cual le aplicamos sucesivamente reglas de inferencia hasta obtener finalmente la parte derecha. La idea de resolución lineal es parecida a este tipo de razonamiento encadenado. Se comienza por una cláusula C_0, se resuelve con otra cláusula para obtener otra resolvente C_1, la cual se resuelve con otra, ..., hasta obtener $\square$.

La característica más atractiva de la resolución lineal es su simplicidad estructural. Además es completa y compatible con la estrategia del conjunto soporte.

Una deducción $f_0, \ldots, f_n$ es una deducción lineal con raíz c_0 si:

- $f_0, \ldots, f_n$ es una deducción con $f_0 = c_0$.
- La cláusula $f_i, i > 0$, es resolvente de f_{i-1} y otra cláusula (que por definición de deducción debe ser, o bien una hipótesis, o algún f_j con $j < i$).

Las estrategias lineales se basan en tomar como subárbol del árbol de deducciones el formado por todas las deducciones lineales de C con raíz prefijada.

Se puede demostrar que si $\square$ se deduce de C, entonces existe una deducción lineal de $\square$ a partir de C. Esto implica que si elegimos una búsqueda primero en anchura, la estrategia que obtenemos será completa. Como es de suponer, para la búsqueda primero en profundidad, se pierde la completitud. Véase como ejemplo el árbol de deducciones del conjunto de cláusulas $C = \{\neg P(x) \vee P(f(x)), \neg P(a), P(x)\}$, y con raíz $c_0 = P(x)$.

Ejercicio 61. Para el conjunto

$$S = \{P \vee Q, \neg P \vee Q, P \vee \neg Q, \neg P \vee \neg Q\}$$

demostrar que es inconsistente usando una estrategia de resolución lineal y a continuación otra no lineal.

Ejercicio 62. Para el conjunto de fórmulas

$$S = \{M(a, f(c), f(b)), P(a), M(x, x, f(x)), \neg M(x, y, z) \vee M(y, x, z), \neg M(x, y, z) \vee D(x, z),$$
$$\neg P(w) \vee \neg M(x, y, z) \vee \neg D(w, z) \vee D(w, x) \vee D(w, y), \neg D(a, b)\}.$$

obtener una refutación lineal, es decir una deducción lineal de la cláusula vacía.

6.2.2. Estrategias input. Una deducción decimos que es input cuando cada resolvente se obtiene utilizando al menos una hipótesis.

En general estas estrategias no son completas, pero para el caso de un tipo especial de cláusulas que estudiaremos a continuación, si hacemos uso de búsqueda primero en anchura, se tiene asegurada la completitud.

Sin pérdida de generalidad, por lo comentado en el apartado anterior, podemos además suponer que nos restringimos a deducciones lineales e input, con lo que $f_i, i > 0$, es resolvente de f_{i-1} y de una hipótesis (una cláusula de C).

EJERCICIO 63. Estudiar si para el conjunto de cláusulas del ejercicio 61 existe una refutación input.

6.2.3. Cláusulas de Horn. Una cláusula es negativa si todos sus literales son negativos (formulas atómicas negadas).

Una cláusula es de Horn si tiene exactamente un literal positivo (un literal que es una fórmula atómica). Por ejemplo $R(x, y)$ y $\neg P(x) \vee Q(a)$ son cláusulas de Horn, pero ni $S(f(x)) \vee Q(b)$ ni $\neg R(x, y)$ lo son.

Un conjunto de cláusulas C de la forma $C = C' \cup \{c_0\}$ con C' un conjunto de cláusulas de Horn y c_0 una cláusula negativa se dice que es un *conjunto de Horn*.

Observar que la resolvente, caso de existir, para una cláusula negativa y una cláusula de Horn es de nuevo una cláusula negativa. La importancia de los conjuntos de Horn radica en el siguiente resultado.

PROPOSICIÓN 2. *Dado un conjunto de Horn $C = C' \cup \{c_0\}$, entonces es inconsistente si y sólo si existe una deducción (usando resolución) de la cláusula vacía que además es lineal e input y con raíz c_0.*

EJEMPLO 46. Sea $\Gamma' = \{\neg P(x) \vee P(f(x)), \neg P(f(y)) \vee \neg P(y) \vee R(y), P(a)\}$, y $C_0 = \neg R(z)$.

Se obtiene la deducción lineal-input de raíz C_0:

- $C_0 = \neg R(z)$.
  $C_1 = \neg P(f(y) \vee \neg P(y)$.
  $C_2 = \neg P(f(a))$.
  $C_3 = \neg P(a)$.
  $C_4 = \square$.

EJEMPLO 47. Verificar que la implicación semántica siguiente es correcta.

$$\{\forall x \forall y(\neg P(x) \vee \neg Q(x, y) \vee R(y)), \forall z Q(f(z), g(z)), P(f(a))\} \models R(g(a)).$$

Aplicando el método de refutación por resolución obtenemos el conjunto

$$\{\forall x \forall y(\neg P(x) \vee \neg Q(x, y) \vee R(y)), \forall z Q(f(z), g(z)), P(f(a)), \neg R(g(a))\}.$$

Las formas clausulares correspondientes son

$$\{\neg P(x) \vee \neg Q(x, y) \vee R(y), Q(f(z), g(z)), P(f(a)), \neg R(g(a))\}.$$

Observamos que dicho conjunto está formado por una cláusula negativa (la negación de la fórmula a probar) así como por cláusulas de Horn. Por tanto, el conjunto será inconsistente si y sólo si existe una deducción lineal e input de la cláusula vacía con raíz la cláusula negativa. La deducción es la siguiente:

1. $\neg R(g(a))$ (hipótesis),
2. $Q(f(z), g(z))$ (hipótesis),
3. $P(f(a))$ (hipótesis),
4. $\neg P(x) \vee \neg Q(x, y) \vee R(y)$ (cláusula negativa),

5. $\neg P(x) \vee \neg Q(x, g(a))$ (resolvente de la cuarta y la primera cláusula con $\sigma = (y|g(a))$),
6. $\neg P(f(a))$ (resolvente de la anterior y la segunda cláusula con $\sigma = (x|f(z))$),
7. $\square$ (resolvente de la anterior y tercera cláusula con $\sigma = \epsilon$).

Con lo cual el proceso de refutación por resolución ha terminado.

Como consecuencia de este hecho, la resolución lineal e input es completa cuando la búsqueda se hace primero en anchura, si partimos de un conjunto de Horn.

Decir que la estrategia de resolución unit es completa para un conjunto de Horn, así como la estrategia de resolución input. Esto no es casual ya que para un conjunto de cláusulas S, se tiene que S tiene una refutación por resolución input si y sólo si S tiene una refutación por resolución unit.

Este tipo de estrategia es la que da lugar a la resolución utilizada en PROLOG, que estudiaremos en el siguiente tema.

EJERCICIO 64. Si un conjunto de cláusulas de Horn es inconsistente, entonces contiene una cláusula unitaria positiva.

6.2.4. Estrategias ordenadas. Existen diversas formas de tener en cuenta el orden de los literales de las cláusulas para usarlo a la hora de limitar las resolventes dentro del árbol de deducciones.

En la definición de resolvente binaria (y por tanto en la de resolvente) no se tuvo en cuenta el orden de los literales dentro de las cláusulas, de hecho da lo mismo tomar una cláusula que cualquier otra que podamos formar alterando el orden de sus literales. En el caso de que se imponga un orden en los literales, podemos hablar de *resolución ordenada,* que consiste en resolver siempre sobre los literales que ocupan la primera posición en las cláusulas. Por ello siempre que hablemos de resolución ordenada, deberemos imponer un orden en los literales y por tanto tendremos así un conjunto de *cláusulas ordenadas.* Diremos que una deducción es una *deducción ordenada* si se ha hecho uso exclusivamente de resolución ordenada.

EJERCICIO 65. Un estudiante de informática es abordado por un aficionado a los juegos de palabras y este le dice: "Si establezco que:

1. Para todo crimen, hay alguien que lo ha cometido.
2. Sólo los malvados cometen crímenes.
3. No son detenidos más que los malvados.
4. Los malvados detenidos no cometen crímenes.
5. Hay crímenes.

tú que me contestas". El estudiante responde: "Hay malvados que aún no están detenidos". ¿Sabrías justificar la veracidad de la respuesta del sagaz estudiante usando lo que hemos aprendido sobre resolución?

EJERCICIO 66. Demuestra que las hipótesis:

$$\Gamma = \{\exists x(P(x) \wedge \forall y(D(y) \to L(x,y))), \forall x(P(x) \to \forall y(Q(y) \to \neg L(x,y)))\}$$

implican semánticamente a la fórmula:

$$\forall x(D(x) \to \neg Q(x))$$

EJERCICIO 67. Dadas las siguientes afirmaciones:

- El que no bebe cerveza bebe vino.
- Miguel es primo hermano de José.
- Francisco no bebe vino.
- José es un predicador.
- Los primos hermanos de predicadores no beben vino.

Se pide responder a la pregunta: ¿Bebe alguien cerveza?

EJERCICIO 68. Demuestra que

$$\{B, V, R \to T(a), (V \wedge B) \to T(a), M(b), M(a), \forall x((M(x) \wedge T(x)) \to G)\} \models G.$$

EJERCICIO 69. Dadas las siguientes afirmaciones:

- Hace buen tiempo.
- Es viernes.
- Juan tiene suerte si es verano.
- Juan tiene suerte si es viernes y hace buen tiempo.
- Antonio es un malvado.
- Juan es un malvado.
- Toda persona malvada y con suerte gana a la ruleta.

¿Ganará Juan a la ruleta?

EJERCICIO 70. Demuestra que la sentencia *Mis aves de corral no son oficiales* es consecuencia de los siguientes hechos:

- Ningún ánade baila el vals.
- Ningún oficial declina nunca una invitación a bailar el vals.
- Todas mis aves de corral son ánades.

EJERCICIO 71. Demuestra que si

- todos los colibríes tinen vivos colores,
- ningún pájaro de gran tamaño se alimenta de miel,
- los pájaros que no se alimentan de miel tienen colores apagados.

Entonces es cierto que *todos los colibríes son de tamaño pequeño*.

EJERCICIO 72. Demuestra que el conjunto de hipótesis

- ningún poema interesante es mal recibido entre gentes de buen gusto,
- ningún poema moderno está libre de afectación,
- todos sus poemas (de usted) versan acerca de pompas de jabón,
- ningún poema afectado goza de aceptación entre gentes de buen gusto,

- ningún poema antiguo versa acerca de pompas de jabón,

implica semánticamente a la sentencia *Todos sus poemas carecen de interés.*

EJERCICIO 73. Dadas las hipótesis

- todos los animales que no cocean son flemáticos,
- los asnos no tienen cuernos,
- un búfalo puede siempre lanzarlo a uno contra la puerta,
- ningún animal que cocea es fácil de engullir,
- ningún animal sin cuernos puede lanzarlo a uno contra una puerta,
- todos los animales son excitables, salvo los búfalos.

Demuestra que *los asnos no son fáciles de engullir.*

EJERCICIO 74. Demuestra que si

- los animales se irritan siempre mortalmente si no les presto atención,
- los únicos animales que me pertenecen están en ese prado,
- ningún animal puede adivinar un acertijo a menos que haya sido adecuadamente instruido en un colegio con internado,
- ningún animal de los que están en este prado es un tejón,
- cuando un animal está mortalmente irritado corre de un lado para otro salvajemente y gruñe,
- nunca presto atención a un animal, a no ser que me pertenezca,
- ningún animal que haya sido adecuadamente instruido en un colegio con internado corre de un lado para otro salvajemente y gruñe.

Entonces *ningún tejón puede adivinar un acertijo.*

CAPíTULO 6

Principio de Resolución y PROLOG

1. Introducción

En este capítulo ponemos de manifiesto la relación entre la resolución lineal input y la resolución llevada a cabo en el lenguaje PROLOG. No es nuestro objetivo dar un manual de programación para PROLOG; de hecho nos vamos a limitar a estudiar la resolución utilizada en PROLOG para ver su relación con lo visto en el capítulo anterior.

En la siguiente sección exponemos cómo se escriben en PROLOG las cláusulas de Horn, a continuación se expone la resolución PROLOG, y por último se comentan algunos problemas elementales que pueden surgir a la hora de usar la resolución PROLOG.

2. Estrategia de resolución lineal-input en PROLOG

Recordemos del capítulo anterior que dado un conjunto de cláusulas Γ, y $C \in \Gamma$, una deducción *lineal-input* a partir de Γ con raíz C es una secuencia de cláusulas $C_0, \ldots, C_n$, de forma que

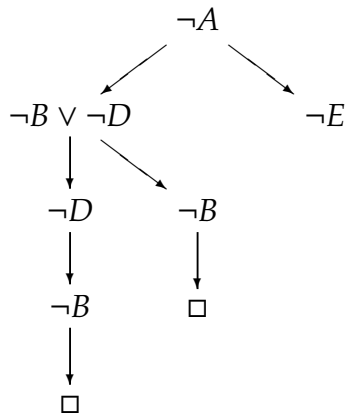
- $C_0 = C$.
- C_i, con i no nulo, se obtiene como resolvente de dos cláusulas, de las cuales una está en Γ, y la otra es C_{i-1}.

Como se puede observar, la definición que acabamos de dar no se atiene a aquella que dimos en su día de deducción en general; ésto se debe a que podemos prescindir de escribir la lista de hipótesis al principio de nuestra deducción, salvo la negativa que será por donde empezaremos, y usar las cláusulas según vayan haciendo falta.

EJEMPLO 48. Sea $\Gamma = \{\neg A, A \vee \neg B \vee \neg D, B, D \vee \neg B, A \vee \neg E\}$ y sea $C_0 = \neg A$. La siguiente deducción de $\square$ a partir de Γ es una deducción lineal-input con raíz C_0.

$C_0 = \neg A$, la raíz.
$C_1 = \neg B \vee \neg D$, que es resolvente de C_0 y $A \vee \neg C \vee \neg D$.
$C_2 = \neg D$, que es resolvente de C_1 y B.
$C_3 = \neg B$, resolvente de C_2 y $D \vee \neg C$.
$C_4 = \square$, resolvente de C_3 y B.

El conjunto de resoluciones lineales-input que se pueden realizar en este ejemplo se puede esquematizar en el siguiente árbol:



Como se puede observar en el ejemplo anterior, una de las ramas no conduce a la cláusula vacía. Por tanto si hiciésemos una búsqueda primero en profundidad, tendríamos que permitir siempre vuelta atrás (*backtracking*), ya que en caso contrario podríamos llegar a una rama que no condujese a la solución.

Puede darse el caso también en el que exista alguna rama infinita, por lo que nuestra búsqueda, primero en profundidad y con vuelta atrás, podría no acabar nunca al dar con una de esas ramas, encontrándose la deducción de la cláusula vacía en otra rama.

Otro problema es que el conjunto de partida no sea un conjunto de Horn, aunque éste sea inconsistente. Como ilustración de este hecho considérese este ejemplo.

EJEMPLO 49. Sea $\Gamma = \{P \vee Q, P \vee \neg Q, \neg P \vee Q, \neg P \vee \neg Q\}$, y tómese como cláusula raíz $P \vee Q$.

Si uno dibuja el árbol asociado a las resoluciones lineales-input, se dará cuenta que en todos los niveles (que no sean el raíz) aparecen $P, Q, \neg P, \neg Q$ repetidos varias veces, y que nunca se alcanza la cláusula vacía, ya que de $P, Q, \neg P$ o $\neg Q$ y una cláusula de Γ nunca se puede obtener $\square$ como resolvente.

Por otro lado es fácil ver que la cláusula vacía se deduce de Γ.

Lo que ocurre es que en las deducciones lineales-input el cálculo de resolventes se ve restringido, y esta restricción puede llevar a la pérdida de la deducción de la cláusula vacía.

El lenguaje de programación PROLOG está basado en cláusulas de Horn. Existen varias razones en las que se basa esta restricción:

Primero: para conjuntos que no sean de Horn no tenemos una estrategia general que sea completa y correcta. Recordemos que las resoluciones lineales-input son completas para conjuntos de Horn. Debido a la simplicidad de este tipo de resoluciones, los algoritmos correspondientes son más eficientes.

Segundo: muchos problemas resultan ser planteables en términos de fórmulas de Horn, por lo que dicha restricción no lo es tal en la práctica.

Decir además que los métodos anteriores se pueden utilizar para llevar a cabo la obtención de respuestas a partir de una información dada.

EJEMPLO 50. Consideremos los siguientes hechos:

- A Manolita le gustan los programas del corazón.
- A Antoñito le caen bien todas aquellas personas a las que les gustan los programas del corazón.

Esta información podemos transcribirla al lenguaje de la lógica de predicados del siguiente modo: usamos el predicado $G(x, y)$ para indicar que a x le gusta o le cae bien y. Necesitamos también tres constantes:

- Manolita será representada por una a,
- Antoñito por una b,
- los programas del corazón por una c.

El conjunto de cláusulas correspondientes es:

- $G(a, c)$,
- $\neg G(x, c) \vee G(b, x)$.

A partir de esta información, una posible pregunta sería: ¿a Antoñito le gusta alguien? Esta pregunta se puede escribir como $\exists y G(b, y)$. Si aplicamos el método de refutación por resolución, negamos la cláusula que representa la pregunta y obtenemos $\forall y \neg G(b, y)$. Puede comprobarse fácilmente que a partir del conjunto ampliado de cláusulas obtenemos la cláusula vacía. Para la obtención de la misma, resulta que la variable y ha tenido que tomar un valor a la hora de unificar literales. ¿Cómo podemos conocer dicho valor? Una forma de averiguar un valor concreto de y es seguir el rastro de las sustituciones que se han ido aplicando. El estado inicial será la cláusula raíz y la sustitución identidad. Si a partir de un estado intermedio, ésto es, una cláusula y una sustitución σ, obtenemos una nueva resolvente utilizando una sustitución τ, entonces el nuevo estado será la nueva resolvente junto con $\tau \circ \sigma$. Procedemos de este modo hasta que alcancemos el estado $(\square, \delta)$. En tal caso una respuesta a la pregunta inicial será $\delta(y)$. En nuestro ejemplo es fácil comprobar que el único valor es y = Manolita.

3. La sintaxis en PROLOG

En sección describimos brevemente cómo se representa la información en PROLOG.

Toda cláusula de Horn viene caracterizada porque sólo posee un literal positivo. Así, toda cláusula de Horn podría expresarse (salvo equivalencia) de la siguiente forma

$$A \vee \neg B_1 \vee \cdots \vee \neg B_n, n \geq 0$$

y usando las leyes de De Morgan obtenemos

$$A \vee \neg(B_1 \wedge \cdots \wedge B_n),$$

que es equivalente a

$$(B_1 \wedge \cdots \wedge B_n) \rightarrow A.$$

La cual en PROLOG se representa del siguiente modo

$$A : -B_1, \ldots, B_n.$$

El punto que aparece al final de la expresión juega en este caso un papel fundamental, similar al que tiene el ";" en PASCAL o en C. PROLOG no aceptará otras fórmulas que las que vengan dadas de esa forma. Una expresión como la que acabamos de presentar se suele llamar *regla* o *cláusula procedural*. Nótese que el literal positivo aparece a la izquierda del símbolo ":-" (el cual representa la implicación de derecha a izquierda) y que los literales negativos aparecen detrás de dicho símbolo. Cada literal B_i se denomina una *llamada a procedimiento*. En el supuesto de que $n = 0$, se adopta una escritura especial, a saber,

$$A.$$

y en este caso a esta expresión se le suele llamar *hecho*. Estas cláusulas representan enunciados afirmativos simples. Un *programa de cláusulas de Horn* o *programa lógico* es un conjunto finito de hechos y cláusulas de procedimiento. Se dice que un programa lógico es invocado por una *cláusula objetivo* (o *cláusula de pregunta*) la cual es una cláusula negativa. En notación de PROLOG ésta se escribe $Q_1, Q_2, \ldots, Q_k$.

Refiriéndonos de nuevo a la intepretación intuitiva, esta notación sugiere que una cláusula objetivo es una secuencia de llamadas procedurales las cuales deben de ser completadas satisfactoriamente. En este mismo sentido la cláusula vacía se denomina *cláusula de parada* ("halting clause"). Ésta puede ser considerada como un caso especial de cláusula objetivo (con $k = 0$) donde todas las llamadas a procedimiento han sido completadas satisfactoriamente. Recordemos, como hicimos constar en su momento, que las cláusulas que intervienen al calcular una resolvente han de tener variables disjuntas.

Cada literal es, o bien una fórmula atómica, o bien negación de una fórmula atómica. Por tanto nos interesa conocer cómo se expresan las fórmulas atómicas en PROLOG. Para ello necesitamos saber cómo expresar las constantes, variables, símbolos de función y símbolos de relación.

- Constantes: las constantes en PROLOG se representan por medio de una cadena de caracteres (con las mismas convenciones típicas de caracteres ASCII válidos que aquellas seguidas en C) empezando por una letra minúscula. Nótese que PROLOG es un lenguaje que distingue entre minúsculas y mayúsculas ("case-sensitive").

- Variables: se expresan igual que las constantes, pero empezando por mayúscula.
- Símbolos de función: igual que las constantes.
- Símbolos de relación: igual que las constantes.

Así pues las constantes propias del lenguaje (constantes, símbolos de relación y símbolos de función) empiezan con minúscula, y las variables con mayúscula.

Para construir términos disponemos de constantes, variables, símbolos de función; además PROLOG permite el uso de números y listas.

Démonos cuenta de que las convenciones de escritura utilizadas en PROLOG son en cierto modo contrarias a las que hemos venido utilizando a lo largo de la asignatura para representar fórmulas de un lenguaje de primer orden. PROLOG distinguirá entre constantes, funciones y relaciones según la forma como aparezcan escritas.

EJEMPLO 51. Algunas expresiones válidas en PROLOG son:

- `tiene(X,coche).`
- `tiene(X,transporte) :- tiene(X,coche), tiene(X,suerte).`
- `esparaguas(Zutano) :- seabre(Zutano), evitalluvia(Zutano).`
- `aprobarFLP(X):-irAclase(X),estudiarApuntes(X).`

4. La resolución PROLOG

Tal y como indicamos en las secciones anteriores, PROLOG trabaja con conjuntos de Horn. El primer paso es crear un archivo de texto que contendrá a nuestro programa lógico. Este fichero ha de tener extesión ".pl". En cuanto ejecutamos nuestro intérprete de PROLOG, para cargar nuestro programa lógico hemos de escribir

```
|?- [nombreDeNuestroFichero].
```

Una vez hecho esto, el usuario puede plantear preguntas u objetivos. Éstas van a representar a la cláusula negativa en cuestión, aunque la forma práctica de hacerlo es la siguiente. Si por ejemplo la pregunta es

$$\exists x \exists y (R(x, y) \wedge P(x))$$

hemos de calcular la forma clausular de la negación de dicha pregunta. Obteniendo en este caso la cláusula

$$\neg R(x, y) \vee \neg P(x).$$

En notación PROLOG la pregunta habría que escribirla

```
|?- r(X,Y),p(X).
```

En general, si la forma clausular obtenida constase de más de una cláusula, tendríamos que plantear cada pregunta por separado y comprobar que al

menos una de ellas tenga solución. Por ejemplo, caso de que nuestra pregunta sea

$$\exists x \exists y (R(x,y) \vee P(x)),$$

obtenemos las cláusulas $\neg R(x,y)$ así como $\neg P(x)$. Por tanto primero plantearíamos la pregunta

```
|?- r(X,Y).
```

y a continuación si ésta no tuviese solución la pregunta

```
|?- p(X).
```

El algoritmo que usa PROLOG para la determinación de la inconsistencia de un conjunto de la forma $\Gamma \cup \{\neg\varphi\}$, es el de búsqueda primero en profundidad con vuelta atrás en el árbol de las deducciones lineales-input (ordenadas [1]) con raíz $\neg\varphi$.

Supongamos que tenemos un conjunto de reglas Γ. Cada nodo consta de una lista de objetivos o preguntas $B_1, \ldots, B_n$ y una substitución σ. Al comienzo $B_1, \ldots, B_n$ son las preguntas planteadas por el usuario y σ es la substitución identidad ε. Supongamos que nos encontramos en el nodo $B_1, B_2, \ldots, B_n; \sigma$, el algoritmo consiste en:

- Si $n = 0$ (la lista de preguntas es vacía), entonces
  - escríbase σ.
  - váyase al nodo precedente.
- En caso contrario
  - tómese la primera pregunta B_1.
  - búsquese entre todas las reglas, en el orden en el que han sido escritas y no exploradas aún en dicho nodo, la primera regla de la forma

    $$A :- C_1, \ldots, C_k.$$

    donde A se pueda unificar con B_1 (nótese que podemos renombrar las variables en la cláusula $A :- C_1, \ldots, C_k$ de forma que no sean las mismas que las que aparecen en $B_1, \ldots, B_n$).
  - Si no hay ninguna regla cumpliendo dicha premisa, entonces
    - Si nos encontramos en la raíz del árbol, terminar.
    - En caso contrario, váyase al nodo precedente.
  - En caso contrario
    - Sea θ un unificador principal para B_1 y A.
    - Desciéndase creando un nuevo nodo con etiqueta

      $$\theta(C_1), \ldots, \theta(C_k), \theta(B_2), \ldots, \theta(B_n); \theta\sigma$$

[1] La matización de ordenadas se debe a que el algoritmo usado tendrá en cuenta el orden en el que se introduzcan las reglas y hechos, así como el orden de los literales en cada regla.

El algoritmo que aquí se expone es una simplificación del seguido por PROLOG. No tenemos en cuenta elementos que pueden complicar este algoritmo, tales como la alteración de control provocada por el usuario (ver sección siguiente).

Pongamos ahora un ejemplo para clarificar las ideas expuestas en esta sección.

EJEMPLO 52. Sea el programa (también denominado "script" o guión) en PROLOG:

```
b.
v.
c(j):-e(a).
c(j):-b,v.
m(a).
m(j).
g(X):-m(X),c(X).
```

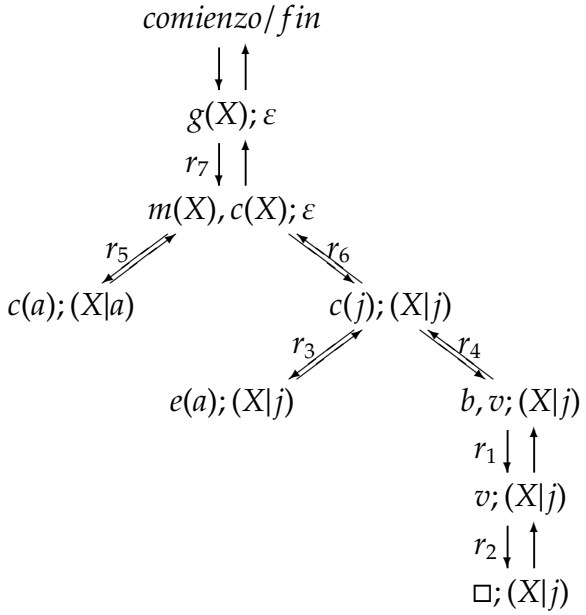
Tenemos así siete reglas($r_1, \ldots, r_7$). Si planteamos la pregunta

```
g(X).
```

la respuesta será

```
X = j.
```

A continuación mostramos el árbol de resolución PROLOG:



Inténtese como ejercicio transcribir cada uno de los pasos dados en el árbol a la notación usual utilizada en lógica de primer orden.

EJERCICIO 75. Si en el ejemplo anterior la constante "j" que aparece en las reglas tres y cuatro se cambia por "J", cuál sería el resultado.

Estamos utilizado el programa `gprolog`. Podemos crear un fichero `ejemplo.pl` con las siguientes cláusulas:

```
r(Y):-p(X),q(X,Y).
q(f(Z),g(Z)).
p(f(a)).
```

Si a continuación ejecutamos el programa `gprolog`, para cargar el fichero anterior hay que escribir

```
|?- [ejemplo].
```

Si recibimos un "`yes`", como respuesta nuestro fichero habrá sido cargado correctamente. Nos podemos preguntar si existe algún elemento que verifique la propiedad `r`. Para ello basta con darle la siguiente línea:

```
|?- r(T).
```

La respuesta obtenida es `T=g(a)`. Esto significa que `g(a)` cumple la propiedad `r`.

Otro uso del PROLOG es el cálculo de unificadores principales. Por ejemplo si tras ejecutar `gprolog`, damos la orden

```
r(f(X),g(Y,a))=r(Z,g(b,T))
```

obtenemos como resultado:

```
T = a
Y = b
Z = f(X)
```

Esto quiere decir que $(T|a, Y|b, Z|f(X))$ es un unificador principal de los anteriores literales.

En la versión disponible, cuando no existe un unificador puede ocurrir que entremos en un ciclo infinito y los cálculos no terminen, como ocurre con el siguiente ejemplo:

```
|?- r(Y,f(X))=r(f(f(X)),f(Y)).
```

5. Algunos problemas de la resolución PROLOG

Como vimos en la Sección 2 el algoritmo usado por PROLOG no es completo. Entre los problemas que pueden surgir destacan:

- Que el algoritmo entre en una rama del árbol infinita.
- Que dependiendo del orden en que sean escritas las reglas y hechos el programa pueda o no encontrar solución a las preguntas planteadas.

Otra de las cuestiones que en principio pudiera parecer que influye en la obtención de un resultado, es el orden en el que aparecen los literales de la pregunta (ya que esto determinará el orden en el que calcularemos resolventes). Sorprendentemente se puede demostrar que ésto no influye.

Como se dijo en el tema anterior, si el problema planteado tiene solución siempre existirá una resolución lineal-input que dé con ella. El que se alcance o no dependerá muchas veces del orden en que vengan dadas las reglas.

Los ejemplos que vienen a continuación pretenden mostrar algunas deficiencias de PROLOG.

EJEMPLO 53. El programa

```
a:-a.
```

no encuentra la solución a la pregunta.

```
a.
```

Ya que dicha dicha solución no existe. El árbol que se origina es un árbol de una sola rama con todos los nodos etiquetados con $a; \varepsilon$.

EJEMPLO 54. El programa

```
a:-a.
a.
```

no encuentra solución a la pregunta

```
a.
```

y en este caso a resulta de las reglas dadas en el programa.

Si reescribimos el programa como sigue

```
a.
a:-a.
```

responderá a la pregunta anterior satisfactoriamente.

EJEMPLO 55. El siguiente programa

```
t(f(g(j))).
t(X):-t(f(X)).
t(X):-t(g(X)).
```

no encuentra solución a la pregunta

```
t(j).
```

y se ve claramente que $t(j)$ es consecuencia lógica de las reglas del programa. En este caso el algoritmo se pierde por una rama infinita. Dibújese como ejercicio el árbol que trazaría el algoritmo.

Muchas veces si se altera el orden en el que son introducidas las reglas, un programa que no encontraba solución puede encontrarla. Se aconseja pues evitar casos como los que se exponen en los ejemplos anteriores e intentar escribir de forma ordenada los programas. A veces es imposible evitar este problema (véase el Ejemplo 55), debido a que la búsqueda es primero en profundidad.

EJERCICIO 76. Comprueba que para el siguiente programa

```
q(X,Z):-q(Y,Z),r(X,Y).
q(X,X).
r(b,c).
```

si preguntamos `q(X,c).`, aunque en el árbol de las deducciones hay dos cálculos con éxito (`q(b,c)` y `q(c,c)`), dicho programa entra en una rama infinita debido al orden en el que hemos dado las reglas. De que forma podríamos ordenar las reglas para obtener una repuesta satisfactoria.

Otro problema a tener en cuenta es que dado un conjunto de cláusulas, éste no venga expresado como un conjunto de cláusulas de Horn. A veces, este problema se puede evitar cambiando uno o varios símbolos de relación por símbolos nuevos que vengan a significar lo contrario, con lo que en todas las cláusulas cambiamos los literales en los que aparecen dichos símbolos de relación por sus complementados y substituimos los símbolos de relación antiguos por los nuevos. Por ejemplo, si tenemos $\{P \vee Q, \neg P \vee \neg Q, P\}$, cambiando P por $\neg P'$ y complementando los literales que empiecen por P, tenemos $\{\neg P' \vee Q, P' \vee \neg Q, \neg P'\}$, que es un conjunto de Horn. Está claro que el conjunto de partida es inconsistente si y sólo si el que acabamos de construir es inconsistente. Por desgracia, no siempre se puede hacer un cambio así. Tómese como ejemplo el conjunto $\{P \vee Q, \neg P \vee \neg Q, P \vee \neg Q, \neg P \vee Q\}$.

EJERCICIO 77. Dibuja el árbol de resolución PROLOG asociado a los problemas de resolución del capítulo anterior que sean expresables en conjuntos de Horn.

A veces podremos intuir que por determinados caminos no se encuentra la cláusula vacía. En estos casos podemos usar los llamados "cortes" que se representan por el símbolo "!" y que tienen un comportamiento parecido al comando "goto" de algunos lenguajes de programación. A grandes rasgos la primera vez que PROLOG se encuentra con un "!", éste se evalúa de forma normal, pero si tras el proceso de retroceso (al no haber encontrado ninguna solución) llegamos a la cláusula que contenía el símbolo de corte "!", entonces se fuerza un salto del flujo de control del programa a la última cláusula padre correspondiente a la pregunta en el árbol de búsqueda que no contenía el símbolo de corte. No todos los expertos aceptan el uso de este tipo de construcciones ya que de una parte puede tratar de ayudar a superar las deficiencias planteadas por la estrategia de búsqueda de PROLOG. Se pueden cortar caminos infinitos, así como cortar subárboles en los que no hay solución, pero también puede ocurrir que se descarten subárboles donde sí que hay solución.

Índice alfabético