

# LÓGICA MATEMÁTICA

APUNTES

Ingeniería en Informática

ESCET

Alessandra Gallinari

2006–2007

# Prólogo

El contenido de esta publicación es un guión de la asignatura de Lógica Matemática para la titulación de Ingeniería en Informática de la Escuela Superior de Ciencias Experimentales y Tecnología (ESCET) de la Universidad Rey Juan Carlos.

Los principales objetivos de la asignatura son:

- Introducir herramientas y conceptos básicos de la Lógica Matemática y sus aplicaciones.
- Ayudar al alumno a aprender a razonar y formalizar correctamente.
- Junto con la asignatura Lógica Informática del próximo cuatrimestre, dar una formación global acerca de los procedimientos formales y algorítmicos de razonamiento automático y resolución formal de problemas.

Con estas notas de clase se intenta presentar de forma organizada el material que se expondrá en las clases teóricas. Por la naturaleza de estos apuntes, la exposición no es completa y es fundamental que el alumno consulte también los libros recomendados en la bibliografía relativa a la asignatura.

Las principales referencias bibliográficas empleadas en estos apuntes son [A], [HLR], [MH], [R] y las notas de clase de los profesores Ana Pradera y Luis Solá.

La primera parte de la publicación trata la lógica proposicional y la segunda la lógica de predicados.

Las dos partes tienen la misma estructura y están divididas en cinco capítulos. Los primeros tres introducen la sintaxis, la semántica y la teoría de la demostración. Los últimos dos presentan unas aplicaciones y proponen unos ejercicios de repaso, que son exámenes propuestos en años académicos anteriores. Además, cada capítulo contiene un apartado de ejercicios.

Estas notas incluyen también dos capítulos preliminares. El primero es una breve introducción histórica a la lógica y a sus relaciones con la Matemáticas y la Informática, el segundo es un repaso de la teoría de conjuntos.

El apéndice A contiene las soluciones de algunos de los exámenes finales de años anteriores y el apéndice B el sistema axiomático de Kleene.

## Agradecimientos

Quiero agradecer al profesor Roberto Muñoz por su indispensable participación en la corrección de estas notas.

Gracias también al profesor Ariel Sánchez por su sugerencias y a los alumnos que han señalado erratas y errores en versiones previas.

# Índice General

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	El lenguaje de la lógica . . . . .	1
1.2	Resumen de la historia de la lógica . . . . .	5
1.2.1	Lógica y filosofía . . . . .	6
1.2.2	Lógica y matemáticas . . . . .	6
1.2.3	Lógica e informática . . . . .	7
1.3	Consejos prácticos para el estudio de la asignatura . . . . .	8
<b>2</b>	<b>Algunas nociones de teoría de conjuntos, relaciones y funciones</b>	<b>11</b>
2.1	Nociones de teoría de conjuntos . . . . .	12
2.1.1	Inclusión e igualdad de conjuntos . . . . .	13
2.1.2	Operaciones con conjuntos . . . . .	14
2.1.3	Partes de un conjunto y propiedades de las operaciones con conjuntos . . . . .	15
2.1.4	Cardinal de un conjunto . . . . .	16
2.2	Relaciones binarias . . . . .	17
2.2.1	Relaciones de equivalencia . . . . .	18
2.3	Relaciones $n$ -arias . . . . .	19
2.4	Funciones . . . . .	20
2.5	Ejercicios . . . . .	22
<b>I</b>	<b>Lógica de proposiciones</b>	<b>25</b>
<b>3</b>	<b>Sintaxis de la lógica proposicional</b>	<b>27</b>
3.1	Alfabeto del lenguaje formal de la lógica proposicional . . . . .	28

3.2	Definición recursiva de las expresiones bien construidas: fórmulas . . . . .	30
3.2.1	El principio de inducción estructural para fórmulas proposicionales . . . . .	31
3.3	Representación de las fórmulas bien construidas . . . . .	33
3.3.1	Fórmulas en forma usual y abreviada . . . . .	33
3.3.2	Fórmulas en forma de árbol estructural . . . . .	35
3.3.3	El principio de recursión estructural para fórmulas proposicionales . . . . .	38
3.4	Formalización del lenguaje natural . . . . .	39
3.5	Ejercicios . . . . .	42
<b>4</b>	<b>Semántica de la lógica proposicional.</b>	
	<b>Teoría interpretativa</b>	<b>45</b>
4.1	Valoraciones de un lenguaje formal . . . . .	45
4.2	Evaluación semántica de fórmulas . . . . .	46
4.2.1	Tablas de verdad . . . . .	48
4.3	Modelos y contraejemplos de una fórmula bien construida. Tautologías, contingencias y contradicciones . . . . .	51
4.4	Evaluación semántica de deducciones. Consecuencia lógica . . . . .	54
4.5	Equivalencia de fórmulas . . . . .	55
4.6	Métodos de refutación. Tableaux . . . . .	57
4.6.1	Refutación . . . . .	57
4.6.2	Definición de los tableaux . . . . .	59
4.6.3	Aplicaciones de los tableaux . . . . .	67
4.7	Ejercicios . . . . .	74
<b>5</b>	<b>Teoría de la demostración</b>	<b>77</b>
5.1	Definición de sistema formal axiomático . . . . .	78
5.1.1	Subdeducciones y notación de Fitting . . . . .	82
5.2	El sistema de deducción natural de Gentzen . . . . .	83
5.2.1	Definición del sistema de deducción natural de Gentzen . . . . .	84
5.2.2	Reglas derivadas del sistema de Gentzen . . . . .	90
5.3	Tableaux sintácticos . . . . .	104
5.4	Compleitud, corrección y decidibilidad . . . . .	105
5.5	Ejercicios . . . . .	107

<b>II</b>	<b>Lógica de predicados de primer orden</b>	<b>109</b>
<b>6</b>	<b>Sintaxis de la lógica de primer orden</b>	<b>111</b>
6.1	Alfabeto del lenguaje formal de la lógica de predicados . . . . .	112
6.2	Definición recursiva de las expresiones bien construidas: términos y fórmulas . . . . .	117
6.2.1	Variables libres y variables ligadas . . . . .	120
6.2.2	El principio de inducción estructural para expresiones bien construidas . . . . .	121
6.3	Representación de las expresiones bien construidas . . . . .	122
6.3.1	Fórmulas en forma abreviada . . . . .	122
6.3.2	Principio de unicidad de estructura para expresiones bien construidas . . . . .	123
6.3.3	Términos y fórmulas en forma de árbol estructural . . . . .	124
6.3.4	El principio de recursión estructural para expresiones bien construidas . . . . .	125
6.4	Formalización del lenguaje natural . . . . .	129
6.5	Ejercicios . . . . .	138
<b>7</b>	<b>Semántica de la lógica de primer orden. Teoría interpretativa</b>	<b>141</b>
7.1	Interpretaciones en lógica de primer orden . . . . .	142
7.2	Interpretación semántica de términos y fórmulas . . . . .	145
7.3	Validez semántica de fórmulas: modelos . . . . .	149
7.4	Evaluación semántica de deducciones . . . . .	151
7.5	Equivalencia de fórmulas . . . . .	153
7.5.1	Sustitución de una variable por un término . . . . .	153
7.5.2	Equivalencia de fórmulas . . . . .	155
7.5.3	Equivalencia lógica y reemplazamiento . . . . .	157
7.6	Ejercicios . . . . .	159
<b>8</b>	<b>Teoría de la demostración</b>	<b>163</b>
8.1	Definición del sistema de deducción natural de Gentzen . . . . .	164
8.1.1	Reglas derivadas del sistema de deducción natural . . . . .	167
8.2	Corrección, completitud y decidibilidad . . . . .	170
8.3	Ejercicios . . . . .	172



# Índice de Figuras

1.1	<a href="http://personal5.iddeo.es/ztt/prafabeto_griego.htm">http://personal5.iddeo.es/ztt/prafabeto_griego.htm</a> . . .	10
2.1	Gráfica de $A \times B$ . . . . .	15
2.2	Gráfica de $R$ . . . . .	17
3.1	Árbol sintáctico de $((p \wedge (p \rightarrow r)) \vee (q \leftrightarrow t))$ . . . . .	37
4.1	Árbol estructural de $\varphi = ((p \wedge (p \rightarrow r)) \vee (q \leftrightarrow t))$ . . . . .	50
6.1	Expresiones bien construidas . . . . .	120
6.2	Árbol sintáctico del término $f(g(x, h(a, y)), i(x, l(y)))$ . . . . .	125
6.3	Árbol sintáctico de $\varphi$ . . . . .	126





# Índice de Tablas

3.1	Los conectivos de la lógica proposicional . . . . .	29
3.2	Tabla de formalización en la lógica proposicional . . . . .	41
4.1	Valores de verdad de los conectivos . . . . .	49
4.2	Tabla de verdad de $\varphi = (p \wedge (p \rightarrow r)) \vee (q \leftrightarrow t)$ . . . . .	51
4.3	Tabla de algunas implicaciones lógicas . . . . .	55
4.4	Tabla de algunas equivalencias lógicas . . . . .	56
4.5	Fórmulas reducibles . . . . .	61
4.6	Esquema de reducción de fórmulas . . . . .	62
7.1	Valores de verdad de los conectivos . . . . .	147
7.2	Tabla de algunas implicaciones lógicas . . . . .	153
7.3	Equivalencias relativas a conectivos lógicos . . . . .	156
7.4	Equivalencias relativas a cuantificadores . . . . .	156

# Capítulo 1

## Introducción

La **lógica formal** es la ciencia que estudia *las leyes de inferencia en los razonamientos*. Por medio de la formalización del lenguaje y de sus reglas básicas, proporciona las herramientas necesarias para poder tratar e intentar resolver rigurosamente problemas que tienen sus orígenes y aplicaciones en todas las áreas de las ciencias.

En particular, los alumnos de la Ingeniería en Informática podrán aplicar sus conocimientos de lógica al estudio del Álgebra, del Cálculo, de la Matemática Discreta, de la Electrónica Digital, de la Teoría de Autómatas y Lenguajes Formales, de la Programación, de las Bases de Datos, etc.

El contenido de este primer capítulo es una breve introducción a los conceptos generales que se desarrollarán en esta publicación.

### 1.1 El lenguaje de la lógica

El **lenguaje natural** que utilizamos en la comunicación humana permite un alto grado de flexibilidad. Ese lenguaje está caracterizado por su gramática, que nos permite determinar si una cierta frase, una combinación de palabras, es válida.

Por ejemplo, en el idioma español la frase “La mesa habla suavemente” es una frase válida, ya que está compuesta por un artículo (la), seguido por un sujeto (mesa) y un predicado (habla suavemente), que consiste a su vez de un verbo (habla) y de un adverbio (suavemente). Es la **sintaxis** de un lenguaje, las reglas de formación de frases correctas, que nos permite afirmar

si una cierta combinación de palabras es válida. Por tanto, la sintaxis no se ocupa del sentido de las frase, sólo determina su validez.

Por el otro lado, la **semántica** de un lenguaje trata el estudio de los sentidos de las frases sintácticamente válidas. Así, la frase del ejemplo anterior es sintácticamente correcta, pero falta de sentido semántico.

La flexibilidad de los lenguajes naturales se basa en la complejidad de sus gramáticas. Es muy complicado, si no imposible, dar una representación completa de las reglas sintácticas de los lenguajes hablados.

Por esa razón, los lenguajes naturales presentan unas limitaciones: si queremos poder formular sólo afirmaciones que se puedan definir correctas (sintácticamente) o verdaderas (semánticamente) sin ningún grado de ambigüedad, tendremos que definir un lenguaje más preciso que se suele denominar **lenguaje formal**.

Sin un lenguaje formal sería imposible estudiar las matemáticas, programar un ordenador y, en general, desarrollar razonamientos científicamente irrefutables.

El siguiente ejemplo ilustra como en el lenguaje natural, en este caso el español, existen oraciones para las cuales ni siquiera tiene sentido preguntarse si son verdaderas o falsas.

**Ejemplo 1.1.1** *Las frases “¿Cómo te llamas?” y “Por favor, dame tu libro” son dos ejemplos de oraciones a las cuales no podemos asociar un valor de verdad.*

Por estas razones y por la necesidad de realizar cálculos exactos, el lenguaje formal de la lógica se construye a partir de unos elementos básicos (atómicos) llamados **proposiciones**, que son oraciones declarativas (apofánticas) simples, a las cuales se pueden asociar valores de verdad sin ninguna ambigüedad.

Entonces las frases “¿Cómo te llamas?” y “Por favor, dame tu libro” no son proposiciones, sin embargo las frases “Hoy llueve” o “Si estudio mucho, apruebo la asignatura” lo son.

Definimos **metalenguaje** al lenguaje, en nuestro caso el español, que vamos a usar para definir un lenguaje formal.

Por ejemplo, en la frase “ $\sqrt{2}$  es el número real positivo tal que su cuadrado es igual a 2” se usa el metalenguaje español para definir el símbolo matemático  $\sqrt{2}$ .

## Estructura del lenguaje formal

Ya comentamos que el interés principal de la lógica es el estudio de la validez del razonamiento. Las definiciones básicas de razonamiento y validez son las siguientes:

**Razonamiento (deducción, inferencia, argumento):** es la obtención de un nuevo conocimiento (conclusión) a partir de una serie de conocimientos (premisas).

**Validez formal de un razonamiento:** un razonamiento es formalmente válido si la conclusión es necesariamente verdadera, siendo las premisas verdaderas.

### Ejemplo 1.1.2 Razonamiento válido:

*Premisa 1: Si estudio todo el temario, entonces apruebo la asignatura.*

*Premisa 2: No he aprobado la asignatura.*

*Conclusión: No he estudiado todo el temario.*

*Razonamiento no válido:*

*Premisa 1: Si estudio todo el temario, entonces apruebo la asignatura.*

*Premisa 2: No he estudiado todo el temario.*

*Conclusión: No apruebo la asignatura.*

En las matemáticas es necesario aprender a distinguir entre razonamientos que son matemáticamente correctos (las demostraciones) y razonamientos que no lo son. Además, para poder resolver problemas concretos es necesario desarrollar la habilidad de construir razonamientos matemáticos originales.

La lógica proporciona las herramientas necesaria para el razonamiento matemático, pero también para muchas otras aplicaciones.

En estos apuntes veremos como aplicar la lógica al diseño de circuitos de ordenador y a la verificación de programas.

Resumiendo, la estructura de todo lenguaje formal viene definida por su sintaxis, semántica y sistemas de demostración:

**Sintaxis** (reglas de formación, gramática): es la definición axiomática de los elementos básicos del lenguaje y de las reglas que permiten obtener nuevas expresiones correctas a partir de aquellos. Las expresiones admitidas por el lenguaje se denominan **fórmulas**.

**Semántica** (relación entre el lenguaje y su significado): es la definición de un conjunto de significados (generalmente verdadero o falso) que se puedan

asociar a una fórmula. Permite definir la validez de una fórmula o de un razonamiento.

**Sistemas de demostración:** son sistemas formales que permiten averiguar cuándo una fórmula o un razonamiento son válidos. En el contexto de la sintaxis se denominan **teoría de la demostración**. En el caso de la semántica se denominan **teoría interpretativa**.

### Niveles de la lógica formal

**Lógica proposicional** (lógica de proposiciones, **LP**): en la lógica proposicional se estudian las fórmulas proposicionales construidas a partir de fórmulas atómicas (proposiciones declarativas simples) y conectivos lógicos (y, o, implica, etc.).

#### Ejemplos 1.1.3 1) Formalización de frases:

*Estudio todo el temario ( $e$ );*

*No estudio todo el temario ( $\neg(e)$ );*

*Apruebo la asignatura ( $a$ );*

*Estudio todo el temario y apruebo la asignatura ( $e \wedge a$ );*

*Si estudio todo el temario, entonces apruebo la asignatura ( $e \longrightarrow a$ ).*

#### 2) Formalización de razonamientos:

*Razonamiento válido:*

*Premisa 1: Si estudio todo el temario, entonces apruebo la asignatura ( $e \longrightarrow a$ ).*

*Premisa 2: No he aprobado la asignatura ( $\neg(a)$ )*

*Conclusión: No he estudiado todo el temario ( $\neg(e)$ )*

*Razonamiento no válido:*

*Premisa 1: Si estudio todo el temario, entonces apruebo la asignatura ( $e \longrightarrow a$ ).*

*Premisa 2: No he estudiado todo el temario ( $\neg(e)$ )*

*Conclusión: No apruebo la asignatura ( $\neg(a)$ )*

**Lógica de predicados de primer orden, (LPO):** la lógica de primer orden es una generalización de la lógica de proposiciones. Distingue entre los objetos del discursos y sus propiedades o posibles relaciones entre ellos. Además, introduciendo nuevos elementos como los cuantificadores existenciales

y universales ( $\exists$  : existe un,  $\forall$  : para todo), permite estudiar la estructura interna de los enunciados.

**Ejemplos 1.1.4** 1) *Formalización de una frase:*

*“El cuadrado de todo número real es no negativo.”*

*Sean  $\mathbb{R}$  el conjunto de los números reales y  $P(x) : x$  es un número no negativo. La formalización de nuestra frase es:*

$$(\forall x(P(x^2))).$$

2) *Formalización de un razonamiento (válido):*

*Sean  $D$  el conjunto de los seres,  $P(x) : x$  es una persona,  $M(x) : x$  es mortal y  $s$  el símbolo constante “Sócrates.”*

*Entonces podemos formalizar el siguiente razonamiento:*

*Premisa 1: Todas las personas son mortales ( $\forall x(P(x) \longrightarrow M(x))$ )*

*Premisa 2: Sócrates es una persona ( $P(s)$ )*

*Conclusión: Sócrates es mortal ( $M(s)$ )*

**Lógicas de orden superior:** son extensiones de la lógica proposicional y de predicados de primer orden que amplían el uso de los cuantificadores a las propiedades y a las relaciones entre los objetos. Notar que en la lógica de primer orden los cuantificadores se refieren sólo a los objetos.

En esta publicación se tratará exclusivamente el estudio de la lógica proposicional y de predicados de primer orden.

## 1.2 Resumen de la historia de la lógica

La definición de lógica como ciencia formal en la cultura occidental es el resultado de un largo desarrollo histórico que empieza con las obras de algunos filósofos griegos y llega hasta la actualidad.

Históricamente las áreas de aplicación más importantes de la lógica son la filosofía, las matemáticas y la informática.

A continuación se presenta un resumen muy reducido de los principales pasos que han llevado a la formulación de la lógica formal. El lector interesado podrá encontrar un tratamiento más extensivo en la referencia [UNED1] y su bibliografía.

### 1.2.1 Lógica y filosofía

En el siglo cuarto antes de nuestra era **Aristóteles** fue el primero en tratar de formalizar el razonamiento humano para poder discernir en la discusiones filosóficas. Aristóteles se puede considerar el fundador de la denominada **lógica clásica**.

Durante la Edad Media el proceso de sistematización de la lógica fue desarrollado por los filósofos árabes y los escolásticos.

En el siglo XIII **Santo Tomás de Aquino** empleó la lógica en el contexto de las discusiones teológicas.

Fue **Leibniz**, en el siglo XVII, el primero a formular la lógica como base del razonamiento matemático, pero sus estudios fueron abandonados hasta el siglo XIX, cuando finalmente se fundó la **lógica matemática** como ciencia.

### 1.2.2 Lógica y matemáticas

A mediados del siglo XIX, en el 1854, el inglés **George Boole** publicó el libro *The Laws of Thought* (Las leyes del pensamiento). Influenciado por las teorías de los matemáticos **De Morgan** y **Hamilton**, Boole definió la lógica como sistema formal dirigido no sólo al estudio del lenguaje natural. Su obra proporciona un modelo algebraico de la **lógica de proposiciones**.

En el 1879 el alemán **Gottlob Frege** publicó el libro *Grundgesetze der Arithmetik: Begriffsschriftlich abgeleitet* (Fundamentos de Aritmética: Conceptualmente derivada), en el cuál se formaliza la **lógica de predicados**.

A principios del siglo XX **Bertrand Russell** y **Whitehead**, inspirados por el trabajo del matemático italiano **Giuseppe Peano**, publicaron los tres volúmenes de *Principia Mathematica* y **Hilbert**, en 1920, propusieron el problema de la axiomatización de las matemáticas. En 1930 **Gödel** demostró el famoso teorema de incompletitud del enfoque axiomático, que contesta negativamente al problema de Hilbert.

El resultado de todas estas obras fue la base teórica de la teoría axiomática y semántica.

Toda teoría matemática se construye a partir de unos **axiomas**, que definen las propiedades básicas de los objetos de la teoría que se consideran verdaderas y, sin embargo, no se demuestran.

La geometría euclídea y la construcción de los números reales son dos ejemplos de este tipo de teorías axiomáticas.



El modelo matemático conocido como Álgebra de Boole es otro ejemplo muy importante en informática usado para el diseño de circuitos lógicos y las búsquedas booleanas en grandes colecciones de datos (índices de páginas Web, datos genéticos, etc.).

Los axiomas de toda teoría matemática tienen que ser:

1) **compatibles**: a partir de ellos no tiene que ser posible deducir una contradicción,

2) **independientes**: ningún axioma se debe poder deducir a partir del resto de ellos (habría redundancia de axiomas),

3) **suficientes**: a partir de ellos tiene que ser posible deducir todas las propiedades que necesitamos satisfagan los objetos de nuestra teoría.

### 1.2.3 Lógica e informática

Una nueva época para la lógica comienza en las décadas de 1950 y 1960 a causa de la aparición de los ordenadores. Surgió entonces la necesidad de determinar si fuese posible especificar formalmente programas y definir sistemas de demostración automática de teoremas. Estos tipo de problemas son los principales objetos de estudio de la **lógica informática**.

El nacimiento de la inteligencia artificial y del primer lenguaje declarativo (LISP) se puede fijar en el 1959, con el trabajo de **McCarthy**.

A lo largo de los años sesenta se mejoran los primeros sistemas de demostración automática y en el 1965 aparece la la regla universal de resolución con unificación de **Robinson**.

En los años setenta se desarrolló la programación lógica como herramienta de resolución de problemas. En 1972 **Colmerauer** creó el primer lenguaje de programación lógica: Prolog.

A partir de los años ochenta se empiezan a utilizar nuevas **lógicas no clásicas**, como, por ejemplo, lógicas que permiten dar una interpretación probabilista de la incertidumbre.

Los métodos deductivos de la lógica matemática están a la base de la demostración automática de teoremas. Se trata de buscar los sistemas de demostración más eficientes para su implementación en un ordenador.

Definida la semántica de un lenguaje de programación, se pueden usar los métodos de demostración de la lógica matemática para verificar (automáticamente) la corrección de programas y sus propiedades.

La **programación lógica** está a la base de la inteligencia artificial y permite deducir nuevos conocimientos a partir de una base de conocimientos (los axiomas) y una serie de deducciones automáticas.

Por tanto, algunas de las áreas de aplicación de la lógica en informática son:

- La minería de datos.
- La descripción de la semántica de los lenguajes de programación y la verificación de programas.
- La demostración automática de teoremas.
- La programación lógica y los sistemas basados en el conocimiento en la inteligencia artificial.

### 1.3 Consejos prácticos para el estudio de la asignatura

Cierra este capítulo de introducción una lista de consejos para los alumnos, que tiene la intención de facilitar el estudio de la lógica matemática.

El principal objetivo del estudio de la lógica es el aprendizaje de las técnicas de demostración formales y su aplicación a la resolución de problemas.

Por eso es indispensable que el alumno domine el lenguaje formal y las técnicas de deducción. Sin estos conocimientos, es imposible desarrollar la actividad creativa necesaria para resolver nuevos problemas, como los que serán propuestos a lo largo de la asignatura.

En general, para aprender un lenguaje es necesario conocer su sintaxis y semántica. Además, sólo con mucha práctica se puede llegar a hablar y razonar correctamente usando ese lenguaje.

#### Lista de consejos:

- **Bibliografía:** tener unos buenos libros de texto facilita enormemente el estudio de cualquier materia. Esta publicación de apuntes de la asignatura es un guión para las clases, no pretende sustituir la bibliografía recomendada.

- **Método de estudio:** las asignaturas de matemáticas requieren un estudio llevado al día. Los temas del programa están desarrollados de forma tal que el aprendizaje sea progresivo y, por tanto, no es posible entender un nuevo tema si se tienen dudas importantes sobre el anterior. Para aclarar dudas es conveniente:
  - volver a estudiar el tema que presenta dificultades consultando distintos textos e intentar resolver los problemas propuestos,
  - preguntar las dudas al profesor en clase o durante sus **horas de tutoría**,
  - trabajar en grupo con otros compañeros.
- **Tiempo de estudio:** en media, el estudio independiente del alumno tendría que ser de una hora y media para cada hora de clase teórica. Este tiempo incluye las actividades dedicadas a la comprensión de los ejemplos presentados y a la resolución de los ejercicios propuestos.
- **Clases prácticas:** durante la clases prácticas el alumnos puede hacer ejercicios individualmente o en grupo. En esto tipo de clases los alumnos pueden verificar si su nivel de comprensión de la materia es el adecuado y tienen la oportunidad de corregir errores de aprendizaje. En las clases prácticas es conveniente aprovechar la presencia y la disponibilidad del profesor para aclarar posibles dudas.
- **Exámenes parciales:** los exámenes parciales también sirven para que el alumno pueda verificar su nivel de conocimiento de la materia. Si es necesario, permiten mejorar su preparación a tiempo para el examen final.

La tabla 1.1 contiene los símbolos del alfabeto griego. En estos apuntes, siguiendo la notación tradicional, se usarán varios de ellos. Con mucha frecuencia se emplearán los símbolos:

$\varphi$  = fi minúscula,  
 $\Phi$  = fi mayúscula,  
 $\psi$  = psi,  
 $\chi$  = ji.

## ALFABETO GRIEGO

---

<i>SIGNO</i>	<i>NOMBRE</i>	<i>PRONUNCIACIÓN</i>
A, α	alfa	a
B, β	beta	b
Γ, γ	gamma	g (como gato)
Δ, δ	delta	d
E, ε	épsilon	e (breve)
Z, ζ	dseta	ds
H, η	eta	e (larga)
Θ, θ	zeta	z
I, ι	iota	i
K, κ	kappa	k
Λ, λ	lambda	l
M, μ	mi	m
N, ν	ni	n
Ξ, ξ	xi	ks
O, ο	ómicron	o (breve)
Π, π	pi	p
P, ρ	ro	r
Σ, σ	sigma	s
T, τ	tau	t
Υ, υ	ípsilon	ü
Φ, φ	fi	f
X, χ	ji	j
Ψ, ψ	psi	ps
Ω, ω	omega	o (larga)

Figura 1.1: [http://personal5.iddeo.es/ztt/pru/alfabeto\\_griego.htm](http://personal5.iddeo.es/ztt/pru/alfabeto_griego.htm)

## Capítulo 2

# Algunas nociones de teoría de conjuntos, relaciones y funciones

(Referencias: R. Criado, A. Burjosa, C. Vegas, R. Banerjee, *Fundamentos matemáticos I*, Ed. Centro de estudios Ramón Areces. Madrid, 1998 y P.H. Halmos, *Naive Set Theory*, Springer-Verlag New York, Heidelberg, Berlin, 1987.)

Este capítulo es un repaso de algunas nociones de teoría de conjuntos y de las definiciones básicas de relaciones y funciones.

En esta exposición presentaremos sólo aquellos conceptos indispensables para el estudio de la asignatura de Lógica Matemática. Los alumnos ampliarán sus conocimientos en asignaturas paralelas, que proporcionan un tratamiento más amplio y detallado de los temas aquí considerados.

La lógica y la teoría de conjuntos están estrechamente relacionadas. De hecho en un principio se pensó que toda propiedad  $P(x)$  (todo predicado en el lenguaje de la lógica de primer orden) llevaba asociado un “conjunto”,

$$\{x : P(x)\}.$$

El conjunto obtenido estaría formado por los elementos  $a$  del universo de discurso  $U$  que satisfacen la propiedad  $P(x)$ , es decir, tales que se pueda afirmar que  $P(x)$  es verdadera si  $x = a$ .

Así, por ejemplo, sea nuestro universo de discurso “los seres humanos” y sea  $P(x)$  la propiedad de un genérico ser humano  $x$  de ser alto al menos 1,70 metros. Dado un particular ser humano  $a$ , es posible determinar si  $a$  mide al menos 1,70 metros, es decir, si  $a$  pertenece al conjunto  $\{x : P(x)\}$ .

Más concretamente, inicialmente se aceptaba la idea que toda propiedad  $P(x)$  divide un universo de discurso en dos partes: la formada por los objetos  $a$  que la satisfacen,  $a \in \{x : P(x)\}$ , y la formada por los objetos  $a$  que no la satisfacen,  $a \notin \{x : P(x)\}$ .

En 1903 Bertrand Russell propuso el siguiente ejemplo de “conjunto” (según la definición de conjunto de su época)

$$A = \{x : x \notin x\},$$

y preguntó si  $A \in A$ .

De la definición de  $A$  se sigue que  $A \in A$  implica que  $A \notin A$  y, además, que  $A \notin A$  implica que  $A \in A$ .

Por tanto se obtiene la contradicción:  $A \in A$  si y sólo si  $A \notin A$ .

El ejemplo de Russell muestra que no toda propiedad determina un conjunto, hace falta restringir la clase de propiedades que definen conjuntos.

La primera de las restricciones es el *axioma de especificación*: una propiedad por sí sola no determina un conjunto, sino que selecciona elementos de un conjunto dado al que es necesario referirse.

Para no incurrir en contradicción con el axioma de especificación, es necesario asumir la no existencia del “conjunto universal  $U$ ,” el conjunto de todos los conjuntos. Si  $U$  fuera un conjunto, también  $A = \{x \in U : x \notin x\}$  tendría que ser un conjunto.

En respuesta a la paradoja de Russell, se propusieron varias formulaciones axiomáticas de la teoría de conjuntos.

En nuestra exposición, utilizaremos reglas de construcción de conjuntos formuladas en términos de la lógica de predicados, a partir de los conceptos primitivos de conjunto y pertenencia (Zermelo-Fraenkel, 1922).

## 2.1 Nociones de teoría de conjuntos

Los conceptos de **conjunto** y de **pertenencia** de un elemento a un conjunto son conceptos *primitivos*, es decir, no se definen.

Diremos que un **conjunto**  $A$  es una colección (familia, clase), finita o infinita, de objetos de un universo  $U$  tal que para todo objeto  $x$  se pueda determinar si  $x$  pertenece a  $A$ . Los objetos de un conjunto serán sus **elementos**. Si  $x$  pertenece al conjunto  $A$ , se escribirá  $x \in A$ . Si  $x$  no pertenece a  $A$ , se escribirá  $x \notin A$ .

### Ejemplos 2.1.1

$\mathbb{N} := \{1, 2, \dots\}$  es el conjunto de los números naturales,

$\mathbb{Z} := \{\dots, -2, -1, 0, 1, 2, \dots\}$  es el conjunto de los números enteros,

$\mathbb{F} := \{\frac{p}{q} : p, q \in \mathbb{Z} \text{ y } q \neq 0\}$  es el conjunto de las fracciones de números enteros,

$A := \{x \in \mathbb{R} : x^2 = 1\} = \{-1, 1\}$  es el conjunto solución de la ecuación  $x^2 - 1 = 0$ .

**Notación:** los símbolos  $\mathbb{Q}$ ,  $\mathbb{R}$  y  $\mathbb{C}$  denotarán, respectivamente, el conjunto de los números racionales, reales y complejos.

### 2.1.1 Inclusión e igualdad de conjuntos

Si todo elemento  $x$  de un conjunto  $A$  es también elemento de un conjunto  $B$ , se dirá que  $A$  está contenido en  $B$  o que  $A$  es un **subconjunto** de  $B$  (y se escribirá  $A \subseteq B$  ó  $B \supseteq A$ ).

Si  $A$  es un subconjunto de  $B$  y existe un elemento de  $B$  que no pertenece a  $A$ , entonces  $A$  es un **subconjunto propio** de  $B$  :  $A \subsetneq B$  ó  $A \subset B$ .

Dos conjuntos  $A$  y  $B$  son **iguales** si contienen los mismos elementos. Por ejemplo, los conjuntos  $A = \{-2, 1, 0, -7\}$  y  $B = \{-7, 1, 0, -2\}$  son iguales.

**Nota importante:** para demostrar que dos conjuntos  $A$  y  $B$  son iguales, es necesario verificar las dos siguientes condiciones:

$$1) A \subseteq B \quad (2.1)$$

$$2) B \subseteq A \quad (2.2)$$

**Ejemplos 2.1.2** 1) Sean  $A = \{x \in \mathbb{R} : x^2 + x - 2 = 0\}$  y  $B = \{-2, 1\}$ . Los elementos de  $A$  son las soluciones de la ecuación  $x^2 + x - 2 = 0$ , es decir, son los números  $x_1 = 1$  y  $x_2 = -2$ . Ya que  $x_1 \in B$  y  $x_2 \in B$ ,  $A \subseteq B$ . Ahora está claro que también  $B \subseteq A$ . Por tanto,  $A = B$ .

2) Sean  $A = \{a, b, c, d\}$  y  $B = \{c, a, d, b\}$ , entonces  $A = B$ .

El **conjunto vacío**  $\emptyset$  es el conjunto que no tiene elementos.

**Nota:** el conjunto  $\emptyset$  **no es igual** al conjunto  $A = \{\emptyset\}$ , pues  $A$  tiene un elemento, el conjunto vacío.

**Ejemplo 2.1.3** Sea  $A = \{x \in \mathbb{R} : x^2 = -1\}$ . Entonces  $A = \emptyset$ .

**Proposición 2.1.4** Sea  $A$  un conjunto cualquiera. Entonces  $\emptyset \subseteq A$ .

## 2.1.2 Operaciones con conjuntos

Si  $A$  y  $B$  son dos conjuntos, es posible construir nuevos conjuntos por medio de las siguientes operaciones:

- la **unión** de  $A$  y  $B$  es el conjunto  $A \cup B$  de todos los elementos de  $A$  o de  $B$ , es decir

$$A \cup B = \{x : x \in A \text{ ó } x \in B\},$$

- la **intersección** de  $A$  y  $B$  es el conjunto  $A \cap B$  de todos los elementos que pertenecen tanto a  $A$  como a  $B$ , es decir

$$A \cap B = \{x : (x \in A) \text{ y } (x \in B)\}.$$

Si  $A$  y  $B$  no tienen elementos en común, entonces  $A \cap B = \emptyset$  y se dirá que  $A$  y  $B$  son **disjuntos**,

- el **complemento (relativo) de  $B$  respecto de  $A$**  es el conjunto  $A \setminus B$  de todos los elementos de  $A$  que no pertenecen a  $B$ , es decir

$$A \setminus B = \{x : (x \in A) \text{ y } (x \notin B)\}.$$

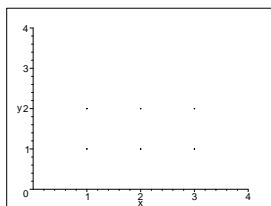
- el **producto cartesiano** de dos conjuntos *no vacíos*  $A$  y  $B$  es el conjunto de todos pares *ordenados*  $(a, b)$  con  $a \in A$  y  $b \in B$ , es decir

$$A \times B = \{(a, b) : (a \in A) \text{ y } (b \in B)\}.$$

Para representar gráficamente un producto cartesiano  $A \times B$  de dos conjuntos, se puede utilizar un sistema de ejes. Los elementos de  $A$  se representan por medio de puntos del eje de las abscisas y los elementos de  $B$  por medio de puntos del eje de las ordenadas. Entonces los elementos de  $A \times B$  son todos los puntos de “coordenadas”  $(a, b)$ .

Por ejemplo, sean  $A = \{x, y, z\}$  y  $B = \{a, b\}$ . La figura (2.1) es una representación gráfica (obtenida con el sistema Maple sustituyendo las letras por números:  $x = 1, y = 2, z = 3, a = 1, b = 2$ ) de  $A \times B$ .



Figura 2.1: Gráfica de  $A \times B$ .

### 2.1.3 Partes de un conjunto y propiedades de las operaciones con conjuntos

Si  $A$  es un conjunto, se llama **conjunto de las partes de  $A$** ,  $P(A)$ , al nuevo conjunto cuyos elementos son exactamente los subconjuntos de  $A$ .

**Nota:** Para todo conjunto  $A$ ,  $P(A)$  es siempre no vacío, ya que  $\emptyset \in P(A)$ .

**Ejemplo 2.1.5** Sea  $A = \{a, b, c\}$ . Entonces

$$P(A) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}.$$

Sean  $A, B$  y  $C$  tres conjuntos. Las principales propiedades de las operaciones con conjuntos son las siguientes:

1) **idempotencia** de la unión y de la intersección:

$$A \cup A = A \quad (2.3)$$

$$A \cap A = A \quad (2.4)$$

2) **conmutatividad** de la unión y de la intersección:

$$A \cup B = B \cup A \quad (2.5)$$

$$A \cap B = B \cap A \quad (2.6)$$

3) **asociatividad** de la unión y de la intersección:

$$A \cup (B \cup C) = (A \cup B) \cup C \quad (2.7)$$

$$A \cap (B \cap C) = (A \cap B) \cap C \quad (2.8)$$

4) **distributividad** de la unión respecto de la intersección y de la intersección respecto de la unión:

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C) \quad (2.9)$$

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C) \quad (2.10)$$

5)

$$A \cup \emptyset = A \quad (2.11)$$

$$A \cap \emptyset = \emptyset \quad (2.12)$$

6) **Leyes de De Morgan** (para conjuntos):

$$C \setminus (A \cup B) = (C \setminus A) \cap (C \setminus B) \quad (2.13)$$

$$C \setminus (A \cap B) = (C \setminus A) \cup (C \setminus B) \quad (2.14)$$

7)

$$(A \setminus B) \cup B = A \cup B \quad (2.15)$$

$$(A \setminus B) \cap B = \emptyset \quad (2.16)$$

$$A \setminus (A \setminus B) = A \cap B \quad (2.17)$$

### 2.1.4 Cardinal de un conjunto

El **cardinal** de un conjunto finito  $A$ ,  $Card(A)$ , es el número de elementos de  $A$ . Si  $A$  es un conjunto infinito se escribirá  $Card(A) = \infty$ .

Sean  $A$  y  $B$  dos conjuntos finitos cualesquiera, entonces

$$\begin{aligned} Card(A \cup B) &= Card(A) + Card(B) - Card(A \cap B) \leq \\ &\leq Card(A) + Card(B) \end{aligned} \quad (2.18)$$

Si  $A \cap B = \emptyset$ ,  $Card(A \cup B) = Card(A) + Card(B)$ .

**Observación:** Se puede comprobar que si  $A$  es un conjunto finito con  $Card(A) = n$ , entonces  $Card(P(A)) = 2^n$ .

**Ejemplos 2.1.6** 1)  $Card(\mathbb{N}) = \infty$

2) Sean  $A = \{-2, 0, 3, 17\}$  y  $B = \{-7, 0, 5, 17, 18\}$ . Entonces,  $A \cup B = \{-7, -2, 0, 3, 5, 17, 18\}$  y  $A \cap B = \{0, 17\}$ . Se sigue que

$$7 = Card(A \cup B) = Card(A) + Card(B) - Card(A \cap B) = 4 + 5 - 2.$$

## 2.2 Relaciones binarias

Sean  $A$  y  $B$  dos conjuntos no vacíos.

**Definición 2.2.1** Una **relación binaria** entre  $A$  y  $B$  es un subconjunto  $R$  del producto cartesiano  $A \times B$ . Si  $(a, b) \in R$  se dirá que  $a$  y  $b$  están relacionados y se escribirá  $aRb$ .

**Ejemplo 2.2.2** Sean  $A = \{a, b, c\}$ ,  $B = \{d, e\}$  y  $R = \{(a, d), (b, e), (c, d), (c, e)\}$ . Entonces  $aRd, bRe, cRd$  y  $cRe$ . (Ver figura (2.2))

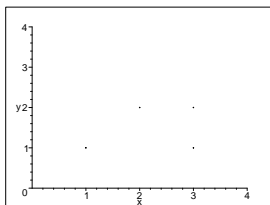


Figura 2.2: Gráfica de  $R$ .

Si  $R \subseteq A \times A$  (es decir, si  $A = B$ ), se dirá que  $R$  es una **relación binaria en  $A$** .

En las siguientes definiciones vamos a emplear el cuantificador universal  $\forall$  y el símbolo de implicación  $\rightarrow$  de la lógica de predicados, que estudiaremos en detalle. La notación  $\forall x \in A$  quiere interpretarse como “para todo elemento  $x$  del conjunto  $A$ ”.

Una relación  $R$  en un conjunto no vacío  $A$  puede ser:

- R1) **reflexiva**:  $\forall x \in A \quad xRx$
- R2) **simétrica**:  $\forall x, y \in A \quad xRy \rightarrow yRx$
- R3) **antisimétrica**:  $\forall x, y \in A \quad (xRy \quad \text{y} \quad yRx) \rightarrow x = y$
- R4) **transitiva**:  $\forall x, y, z \in A \quad (xRy \quad \text{y} \quad yRz) \rightarrow xRz$

Para estudiar las propiedades de una relación binaria sobre un conjunto  $A$  es conveniente representar gráficamente la relación (ver ejercicio (2.5.6)).

**Observación 2.2.3** Las únicas relaciones binarias en un conjunto no vacío  $A$  que sean al mismo tiempo simétricas y antisimétricas son tales que  $R \subseteq \{(x, y) : x = y\}$ .

**Ejemplos 2.2.4** 1) Sea  $A$  el conjunto de las personas y  $R = \{(a, b) \in A \times A : a \text{ es el padre de } b\}$ . Esta relación no tiene ninguna de las propiedades  $R1$ ,  $R2$  y  $R4$ .

2) En el conjunto de las partes  $P(A)$  de un conjunto  $A$ , la relación de inclusión  $R = \{(B, C) \in P(A) \times P(A) : B \subseteq C\}$  es reflexiva, antisimétrica y transitiva.

3) En el conjunto  $\mathbb{Z}$  de los números enteros, la relación  $R = \{(n, m) \in \mathbb{Z} \times \mathbb{Z} : n - m \text{ es par}\}$  es reflexiva, simétrica y transitiva.

4) En el conjunto de las rectas del plano real, la relación “ $r$  es ortogonal a  $s$ ” no es reflexiva, es simétrica y no es transitiva.

**Definición 2.2.5** Si  $R \subseteq A \times B$  es una relación binaria, se denomina

- **dominio** de  $R$  al conjunto

$$\text{dom}(R) = \{x \in A : \exists y \in B \text{ tal que } (x, y) \in R\} \subseteq A$$

- **imagen directa (o rango)** de  $R$  al conjunto

$$\text{Im}(R) = \{y \in B : \exists x \in A \text{ tal que } (x, y) \in R\} \subseteq B$$

- **imagen inversa (o recíproca)** de un subconjunto  $C$  de  $B$  al conjunto

$$R^{-1}(C) = \{x \in A : \exists y \in C \text{ tal que } (x, y) \in R\} \subseteq A$$

- **codominio** de  $R$  al conjunto  $B$ .

## 2.2.1 Relaciones de equivalencia

**Definición 2.2.6** Una relación binaria  $R$  en un conjunto no vacío  $A$  se denomina **relación de equivalencia** si es reflexiva, simétrica y transitiva. Si  $R$  es una relación de equivalencia en  $A$  y  $a, b \in A$  son tales que  $aRb$ , se escribirá  $a \sim b$ .

Si  $a \in A$  y  $\sim$  es una relación de equivalencia en  $A$ , se puede definir un subconjunto  $C(a)$  de  $A$  denominado **clase de equivalencia de  $a$**  :

$$C(a) = \{x \in A : x \sim a\}. \quad (2.19)$$

Notar que  $C(a)$  no es vacío ya que toda relación de equivalencia es reflexiva.

Sea  $b$  otro elemento de  $A$ . Puede ocurrir sólo una de las siguientes situaciones:

$$\text{si } a \sim b, \text{ entonces } C(a) = C(b), \quad (2.20)$$

$$\text{si } a \not\sim b, \text{ entonces } C(a) \cap C(b) = \emptyset. \quad (2.21)$$

Por tanto, si consideramos el conjunto de las *distintas* clases de equivalencias, este conjunto representa una *partición* de todo  $A$  entre subconjuntos disjuntos y se denomina *conjunto cociente*.

**Ejemplos 2.2.7** 1) La relación  $R = \{(n, m) \in \mathbb{Z} \times \mathbb{Z} : n - m \text{ es par}\}$ , es una relación de equivalencia y  $\mathbb{Z} = C(0) \cup C(1)$ .  $C(0)$  es el conjunto de todos los enteros pares y  $C(1)$  de los enteros impares.

2) En el conjunto de las rectas del plano real, la relación “ $r$  es paralela a  $s$ ” es una relación de equivalencia. Para toda recta  $r$ ,  $C(r)$  representa a la **dirección** determinada por  $r$ .

### 3) **Números racionales**

En el conjunto  $\mathbb{F}$  de las fracciones  $\mathbb{F} := \{p/q : p, q \in \mathbb{Z} \text{ y } q \neq 0\}$ , para todo par de fracciones  $r_1 = \frac{p_1}{q_1}$  y  $r_2 = \frac{p_2}{q_2}$ , se define la relación de equivalencia  $R$  como  $r_1 \sim r_2 \leftrightarrow p_1q_2 = p_2q_1$ . El conjunto de las clases de equivalencia es el conjunto  $\mathbb{Q}$  de los números racionales.

**Observación 2.2.8** Otra importante clase de relaciones binarias sobre un conjunto es la clase de las relaciones de orden, que no necesitaremos conocer para el estudio de la lógica matemática. Además, el alumno tendrá ocasión de estudiar las relaciones de orden en la asignatura paralela de Matemática Discreta y Álgebra. Por estas razones, omitimos su estudio en estos apuntes.

## 2.3 Relaciones $n$ -arias

Sean  $A_1, A_2, \dots, A_n$  conjuntos no vacíos.

**Definición 2.3.1** Una **relación de aridad  $n$  o  $n$ -aria** es un subconjunto  $R$  del producto cartesiano  $A_1 \times A_2 \times \cdots \times A_n$ . Si  $(a_1, a_2, \dots, a_n) \in R$  se dirá que  $(a_1, a_2, \dots, a_n)$  están relacionados.

**Ejemplo 2.3.2** Sean  $A_1 = \{a, b, c\}$ ,  $A_2 = \{d, e\}$  y  $A_3 = \{1, 2\}$ .  $R = \{(a, d, 1), (b, e, 1), (c, d, 2), (c, e, 2)\}$  es una relación ternaria sobre  $A_1 \times A_2 \times A_3$ .

## 2.4 Funciones

Una función (o aplicación)  $(n + 1)$ -aria,  $f : A_1 \times A_2 \times \cdots \times A_n \longrightarrow B$ , de un conjunto no vacío  $A = A_1 \times A_2 \times \cdots \times A_n$  a un conjunto no vacío  $B$  se puede definir como “una regla de correspondencia que asigna a cada elemento  $(a_1, a_2, \dots, a_n) \in A$  un único elemento  $b = f(a_1, a_2, \dots, a_n) \in B$ .” Esta definición es muy intuitiva, pero no explica el término “regla de correspondencia” con suficiente claridad.

La siguiente definición es más general e identifica el concepto de función con una clase particular de relaciones binarias.

**Definición 2.4.1** Sean  $A_1, A_2, \dots, A_n$  y  $B$  conjuntos no vacíos. Una **función (o aplicación)  $(n + 1)$ -aria**  $f : A_1 \times A_2 \times \cdots \times A_n \longrightarrow B$  es una relación  $(n + 1)$ -aria  $f \subseteq A_1 \times A_2 \times \cdots \times A_n \times B$  tal que

- f1)  $\text{dom}(f) = A_1 \times A_2 \times \cdots \times A_n$ ,
- f2) si  $(a_1, a_2, \dots, a_n) \in \text{dom}(f)$  existe un **único**  $f(a_1, a_2, \dots, a_n) \in B$  tal que  $(a_1, a_2, \dots, a_n, f(a_1, a_2, \dots, a_n)) \in f$ .

Entonces una función  $f : A_1 \times A_2 \times \cdots \times A_n \longrightarrow B$  es una relación entre  $A_1 \times A_2 \times \cdots \times A_n$  y  $B$  tal que a cada elemento de  $(a_1, a_2, \dots, a_n) \in A_1 \times A_2 \times \cdots \times A_n$  corresponde un único elemento  $f(a_1, a_2, \dots, a_n)$  del codominio  $B$ . Si  $(a_1, a_2, \dots, a_n, f(a_1, a_2, \dots, a_n)) \in f$ , se dirá que  $f(a_1, a_2, \dots, a_n)$  es la **imagen de**  $(a_1, a_2, \dots, a_n)$  por la función  $f$  o el **valor de**  $f$  en  $(a_1, a_2, \dots, a_n)$ .

Si el dominio de una función está compuesto de un sólo conjunto  $A$ , entonces la función es **binaria**.

**Test de la recta vertical:** Una relación  $R \subseteq A \times B$  es una función si y sólo si

- 1)  $\text{dom}(R) = A$  y
- 2) su gráfica corta a cada “recta vertical” en un punto a lo más.

**Ejemplos 2.4.2** 1) La relación binaria definida en el ejemplo (2.2.2) no es una función.

2) La función  $f : \mathbb{R} \longrightarrow \mathbb{R}$  definida por  $\forall x \in \mathbb{R}, f(x) = x$ , es tal que  $\text{dom}(f) = \mathbb{R}$  y  $\text{Im}(f) = \mathbb{R}$ .

3) La función  $f : \mathbb{R} \longrightarrow \mathbb{R}$  definida por  $\forall x \in \mathbb{R}, f(x) = x^2$ , es tal que  $\text{dom}(f) = \mathbb{R}$  e  $\text{Im}(f) = \mathbb{R}^+ \cup \{0\} (= [0, \infty))$ . En este caso,  $f^{-1}([0, 2]) = [-\sqrt{2}, \sqrt{2}]$  y  $f^{-1}((-2, 0]) = \{0\}$ .

4) La función  $f(x) = \sqrt{x}$  está definida sólo para números reales no negativos, entonces  $\text{dom}(f) = \text{Im}(f) = [0, \infty)$ .

5) El dominio de la función  $f(x) = \frac{x+2}{x^2-1}$  es  $\mathbb{R} \setminus \{-1, 1\}$ .

**Definición 2.4.3** Dos funciones  $f, g \subseteq A \times B$  son **iguales** si y sólo si:

- a)  $\text{dom}(f) = \text{dom}(g)$
- b)  $\forall x \in \text{dom}(f), f(x) = g(x)$ .

**Ejemplo 2.4.4** Las funciones  $f(x) = x^2, \forall x \in \mathbb{R}$  y  $g(x) = x^2, \forall x > 0$  no son iguales, ya que  $\text{dom}(f) \neq \text{dom}(g)$ .

**Definición 2.4.5** Sean  $f : A \longrightarrow B$  y  $g : B \longrightarrow C$  dos funciones. La función **composición (o compuesta)** de  $f$  y  $g$  es la función  $g \circ f : A \longrightarrow C$  definida por  $\forall a \in A, (g \circ f)(a) = g(f(a))$ . Entonces

$$g \circ f = \{(a, g(f(a))) : a \in A\} \subseteq A \times C.$$

**Ejemplos 2.4.6** a) Siendo  $\mathbb{R}^+$  el conjunto de los números reales positivos, sea  $f : \mathbb{R} \longrightarrow \mathbb{R}^+ \cup \{0\}$  la función  $f(x) = x^2 + 1$  y sea  $g : \mathbb{R}^+ \cup \{0\} \longrightarrow \mathbb{R}^+ \cup \{0\}$  la función  $g(x) = \sqrt{x}$ .

Entonces, la función  $g \circ f : \mathbb{R} \longrightarrow \mathbb{R}^+ \cup \{0\}$  es la función  $(g \circ f)(x) = g(f(x)) = \sqrt{x^2 + 1}$ .

b) Sean  $D$  el conjunto de las personas,  $M$  el subconjunto de las madres y  $A$  el subconjunto de las abuelas.

Definimos las funciones  $f : D \longrightarrow M$  y  $g : M \longrightarrow A$  que asocian a toda persona y a toda madre, respectivamente, su propia madre.

En este caso la función  $g \circ f : D \longrightarrow A$  resulta ser la función que asocia a toda persona su abuela materna.

## 2.5 Ejercicios

**Ejercicio 2.5.1** Sean  $A = \{-1, 0, 3, 7\}$  y  $B = \{-3, -1, 0, 3, 5, 7, 8\}$ . Determinar  $A \cup B$ ,  $A \cap B$  y  $A \setminus B$ .

**Ejercicio 2.5.2** Sean  $A = \{-1, 0, 3, 7\}$  y  $B = \{-1, 1\}$ . Determinar y representar gráficamente el conjunto  $A \times B$ .

**Ejercicio 2.5.3** Demostrar las propiedades

$$(A \setminus B) \cup B = A \cup B \quad (2.22)$$

$$(A \setminus B) \cap B = \emptyset \quad (2.23)$$

$$A \setminus (A \setminus B) = A \cap B \quad (2.24)$$

**Ejercicio 2.5.4** Sean  $A = \{a, b, c\}$ ,  $R = \{(a, a), (b, a), (c, b), (c, c)\}$ . Entonces  $aRa, bRa, cRb$  y  $cRc$ . Representar gráficamente la relación  $R$ .

**Ejercicio 2.5.5** En base a la siguiente Tabla 1, describir las relaciones en el conjunto  $A = \{\text{Andrea, Beatriz, Carlos, Davide, Edward}\}$ :

- 1)  $xR_1y \leftrightarrow x$  e  $y$  viven en el mismo país.
- 2)  $xR_2y \leftrightarrow x$  e  $y$  tienen el mismo número de teléfono o la misma edad.
- 3)  $xR_3y \leftrightarrow x$  e  $y$  tienen la misma altura y son europeos.

<b>Tabla 1</b>	Edad	Tel.	País	Altura	Ocupación
Andrea	21	43-6950-555-0001	Alemania	1,75	Informática
Beatriz	18	34-91-555-0000	España	1,68	Estudiante
Carlos	37	34-91-555-0000	España	1,75	Profesor
Davide	18	39-06-555-0002	Italia	1,65	Estudiante
Edward	21	1-215-555-0003	EEUU	1,68	Profesor

**Ejercicio 2.5.6** Interpretar gráficamente las propiedades reflexiva, simétrica, antisimétrica y transitiva.

**Ejercicio 2.5.7** Determinar dominio e imagen de las relaciones definidas en los ejemplos (2.2.2) y (2.2.4).

**Ejercicio 2.5.8** Utilizando la Tabla 1 del ejercicio (2.5.5), definir una relación de equivalencia en  $A$  y determinar las relativas clases de equivalencia.



**Ejercicio 2.5.9** *Determinar cuáles de las siguientes relaciones binarias son funciones:*

a) *Sobre el conjunto de las personas, la relación que asocia a cada persona su madre.*

b) *Sobre el conjunto de las personas que tienen hermanos, la relación que asocia a cada persona su hermano.*

c) *Sean  $A$  el conjunto de los alumnos matriculados en nuestra asignatura el año pasado y  $B$  el conjunto de las notas*

$$\{NP, SS, AP, NT, SB, MH\}.$$

*En  $A \times B$  definimos la relación que asocia a cada alumno su nota final.*

d) *Con la notación del apartado c), en  $B \times A$  definimos la relación que asocia a cada nota los alumnos que han sacado esa nota en la asignatura.*

**Ejercicio 2.5.10** a) *Sean  $f, g : \mathbb{R} \rightarrow \mathbb{R}$  las funciones  $f(x) = \sqrt{x^2 + 1}$  y  $g(x) = x + 1$ . Verificar que  $g \circ f \neq f \circ g$ . (Entonces la composición de funciones no es conmutativa.)*

b) *Mostrar que si  $f : A \rightarrow B$  es una función, entonces  $f = Id_B \circ f = f \circ Id_A$ .*

c) *Descomponer la función  $f(x) = \sqrt{3 + \left(\frac{1}{2 + \sin(x)}\right)^2}$  en una composición de funciones más simples.*



Parte I

Lógica de proposiciones



## Capítulo 3

# Sintaxis de la lógica proposicional

Como ya comentamos en el capítulo de introducción, la **sintaxis** es la definición axiomática de los elementos básicos del lenguaje y de las reglas que permiten obtener nuevas expresiones correctas a partir de aquellos, las fórmulas.

Recordamos que se puede fijar el origen de la lógica matemática (y de la lógica proposicional) al final del siglo XIX, coincidiendo con la aparición de las obras de G. Boole (1815-1864) y de G. Frege (1848-1925).

El objetivo de este capítulo es el estudio de la sintaxis de la **lógica proposicional** que nos permite analizar las fórmulas proposicionales construidas a partir de fórmulas atómicas (proposiciones declarativas simples) y conectivos lógicos.

### Definiciones generales:

- Un **alfabeto**  $A$  es un conjunto de símbolos.
- Una **palabra sobre el alfabeto**  $A$  es una secuencia finita de símbolos de  $A$ . Al conjunto de todas las posibles palabras se le suele denotar como  $A^*$ .
- Un **lenguaje sobre el alfabeto**  $A$  es un cualquier subconjunto de  $A^*$ .
- Las **reglas de formación** son las reglas que permiten obtener nuevas expresiones de un lenguaje a partir de expresiones básicas.

A continuación vamos a definir el lenguaje de la lógica de proposiciones por medio de su alfabeto y sus reglas de formación.

## 3.1 Alfabeto del lenguaje formal de la lógica proposicional

Los elementos básicos del **alfabeto del la lógica proposicional** son:

- **Las proposiciones atómicas (enunciados simples o variables proposicionales):** son proposiciones (oraciones declarativas a las cuales se pueden asociar valores de verdad) que no pueden descomponerse en otras proposiciones más simples.

Para representar las proposiciones atómicas se suelen usar los símbolos  $p, q, r, s, t, \dots$ . El conjunto de estos símbolos se suele denominar **signatura**.

**Ejemplos 3.1.1** *Los siguientes son ejemplos de proposiciones simples:*

$p =$  la raíz cuadrada de 2 es irracional,

$q =$  hoy me siento feliz,

$t =$  los gatos son felinos.

- **Los conectivos lógicos:**

1. **constantes (de aridad 0):**  $\top$  (verdadero) y  $\perp$  (falso)

2. **conectivos unarios:**  $\neg$  (negación)

**Ejemplo 3.1.2** *Siendo  $q =$  hoy me siento feliz, aplicando la negación se obtiene  $\neg(q) =$  hoy no me siento feliz.*

3. **conectivos binarios:**

$\wedge$  (y: la conjunción),

$\vee$  (ó: la disyunción),

$\rightarrow$  (la implicación o condicional),

$\leftrightarrow$  (la doble implicación o coimplicación).

**NOTACIÓN:** El símbolo  $\circ$  se usará para representar un conectivo binario cualquiera.

Conectiva lingüística	Conectivo lógico	Símbolo	Fórmula
verdadero	Constante de aridad 0	$\top$	$\top$
falso	Constante de aridad 0	$\perp$	$\perp$
no $p$	Negación (unario)	$\neg$	$\neg(p)$
$p$ y $q$	Conjunción (binario)	$\wedge$	$(p \wedge q)$
$p$ ó $q$	Disyunción (binario)	$\vee$	$(p \vee q)$
si $p$ entonces $q$	Implicación (binario)	$\rightarrow$	$(p \rightarrow q)$
$p$ si y sólo si $q$	Coimplicación (binario)	$\leftrightarrow$	$(p \leftrightarrow q)$ o $(p \rightarrow q) \wedge (q \rightarrow p)$

Tabla 3.1: Los conectivos de la lógica proposicional

**Ejemplos 3.1.3** 1) La frase “Hoy llueve, sin embargo no hace frío” se escribe como  $(p \wedge \neg(q))$ , donde  $p =$  hoy llueve y  $q =$  hace frío.

2) La frase “Los lápices de mi hermana son rojos o negros” se escribe como  $(p \vee q)$ , donde  $p =$  los lápices de mi hermana son rojos y  $q =$  los lápices de mi hermana son negros.

3) El comando “IF  $p$  THEN  $q$ ” se escribe como  $(p \rightarrow q)$ .

4) La frase “La luna es de papel si y sólo si Carlos lee muchos libros” se escribe como  $(p \leftrightarrow q)$ , donde  $p =$  la luna es de papel y  $q =$  Carlos lee muchos libros.

- **Los símbolos de puntuación (o símbolos auxiliares):** paréntesis abiertos y cerrados, comas.

**Ejemplos 3.1.4** 1) La frase “Si  $n$  es un número primo y mayor que 2, entonces  $n$  es impar” se escribe como  $((p \wedge q) \rightarrow r)$ , donde  $p = n$  es primo,  $q = n$  es mayor que 2 y  $r = n$  es impar.

2) El comando “IF  $p$  THEN  $q$  ELSE  $r$ ” en lógica de proposiciones se escribe como  $((p \rightarrow q) \vee r)$ .

## 3.2 Definición recursiva de las expresiones bien construidas: fórmulas

Los elementos básicos de un lenguaje permiten definir cadenas finitas de símbolos arbitrarias (**palabras**). Así, por ejemplo, la palabra  $(\neg p \wedge \vee)q \rightarrow$  se puede formar a partir del alfabeto de la lógica de proposiciones. De todas las posibles palabras, nos interesa definir aquellas que se obtienen a partir del alfabeto dado sólo por medio de la reglas de formación de nuestro lenguaje.

En matemáticas y en la ciencia de la computación a menudo se usan **definiciones recursivas** de conjuntos o funciones.

Por ejemplo, para definir el conjunto  $\mathbb{N} = \{1, 2, 3, 4, \dots\}$  de los números naturales se puede dar un símbolo inicial 1 y unas reglas de formación, que nos permite hallar el resto de los elementos del conjunto. En nuestro caso, la definición recursiva de los números naturales está dada por los axiomas de Peano (siglo XIX).

### Axiomas de Peano:

- En  $\mathbb{N}$  hay un elemento distinguido que denominamos **1**.
- Para cada  $n \in \mathbb{N}$  se define de manera única el **siguiente** de  $n$ . El siguiente de  $n$  se denota  $s(n) = n + 1$  y es un elemento de  $\mathbb{N}$  para cada  $n \in \mathbb{N}$ .
- No existe ningún número natural  $n$  tal que  $s(n) = 1$ .
- Si  $s(n) = s(m)$  entonces  $n = m$ .
- **Principio de inducción:** si un conjunto de números naturales contiene al 1 y a los sucesores de cada uno de sus elementos, entonces contiene a todos los números naturales.

Otro ejemplo muy conocido es la definición recursiva de la sucesión  $\{f_n\}_{n \in \mathbb{N}}$  de Fibonacci (hacia 1175 – 1250), que fue definida para estudiar la reproducción de los conejos. Sus primeros dos términos son  $f_1 = 1$  y  $f_2 = 1$ . Si  $n \geq 3$ , entonces el valor  $f_n$  se deduce de los valores  $f_{n-1}$  y  $f_{n-2}$  según la fórmula  $f_n = f_{n-1} + f_{n-2}$ . Se sigue que  $\{f_n\} = \{1, 1, 2, 3, 5, 8, 13, \dots\}$ .

Volviendo a la lógica proposicional, para definir las expresiones sintácticamente correctas también se usa una construcción de tipo recursivo, que es la siguiente:



**Definición 3.2.1** Una **fórmula proposicional (fórmula bien construida, fbc)** es una palabra sobre el alfabeto de la lógica proposicional que puede construirse en un número finito de pasos mediante las reglas de formación que vamos a definir a continuación.

**Definición recursiva de las expresiones bien construidas:**

1. (At): Los símbolos  $\top$  y  $\perp$  y toda proposición atómica son una fórmula.
2. ( $\neg$ ): Si  $\varphi$  es una fórmula entonces  $\neg(\varphi)$  es una fórmula.
3. ( $\circ$ ): Si  $\varphi$  y  $\psi$  son dos fórmulas entonces  $(\varphi \circ \psi)$  es una fórmula.
4. Si una palabra no se obtiene mediante las tres reglas anteriores entonces no es una fórmula.

**NOTACIÓN:** Para representar las fórmulas se suelen usar las letras del alfabeto griego  $\varphi, \psi, \chi, \dots$

**Definición 3.2.2** Dadas dos fórmulas  $\varphi$  y  $\psi$ , se dice que  $\psi$  es una **subfórmula** de  $\varphi$  si  $\psi$  consiste de símbolos consecutivos de  $\varphi$ . En particular, cada fórmula es una subfórmula de sí misma.

**Ejemplo 3.2.3** La palabra  $((p \wedge q) \rightarrow r)$ , del ejemplo (3.1.4) es una fórmula bien construida ya que:

1.  $p, q, r$ , son fórmulas atómicas (aplicando (At)),
2.  $(p \wedge q)$  es una fórmula proposicional (aplicando ( $\circ$ )),
3.  $((p \wedge q) \rightarrow r)$  es una fórmula proposicional (aplicando ( $\circ$ )).

Además,  $(p \wedge q)$  y  $r$  son subfórmulas de  $((p \wedge q) \rightarrow r)$ .

### 3.2.1 El principio de inducción estructural para fórmulas proposicionales

Para poder estudiar las propiedades de las fórmulas proposicionales una de las técnicas más adecuadas es el principio de inducción estructural para fórmulas proposicionales, que es una versión generalizada del razonamiento de inducción definido a partir de los axiomas de Peano de los números naturales.

Antes de enunciar el principio de inducción estructural, recordamos la definición de razonamiento por inducción:

### Razonamiento por inducción

Sea  $P(n)$  una propiedad para números naturales. Supongamos que se pueda probar que:

1. **Base de inducción:** se verifica  $P(1)$ ,
2. **Paso de inducción:** si se verifica  $P(n)$  (**hipótesis de inducción**), entonces se verifica  $P(n + 1)$ .

Bajo las hipótesis anteriores, se sigue que se verifica  $P(n)$  para **todo número natural**  $n$ .

La demostración de la validez del razonamiento de inducción se obtiene aplicando el principio de inducción al conjunto  $A = \{n \in \mathbb{N} : P(n) \text{ se verifica}\}$ .

**Ejemplo 3.2.4** *Vamos a demostrar por inducción que para todo  $n \in \mathbb{N}$ ,*

$$2^n \leq (n + 1)!$$

*Base de inducción:*  $P(1)$  es verdadera, siendo  $(2 \leq 2)$ .

*Paso de inducción:* si  $2^n \leq (n + 1)!$  (si se verifica  $P(n)$ ), entonces

$$2^{n+1} = 2 \cdot 2^n \leq 2 \cdot (n + 1)! \leq (n + 2) \cdot (n + 1)! = (n + 2)!.$$

*Por tanto,*  $2^n \leq (n + 1)!$  *se verifica para todo número natural*  $n$ .

El principio de inducción estructural generaliza el método de demostración por inducción. Ese método de demostración se emplea para verificar propiedades de conjuntos generados inductivamente (recursivamente), cuyos elementos se obtienen como resultado de aplicar un número finito de reglas de formación.

Sus aplicaciones son numerosas y en informática se utiliza, por ejemplo, para la verificación de programas (ver por ejemplo el capítulo 9 de [C]).

### Principio de inducción estructural para fórmulas proposicionales

Sea  $\mathbb{P}$  una cierta propiedad tal que:

1. **Base de inducción (At):** Los símbolos  $\top$  y  $\perp$  y toda proposición atómica cumplen la propiedad  $\mathbb{P}$ .

## 2. Pasos de inducción:

- $(\neg)$ : Si  $\varphi$  es una fórmula que cumple la propiedad  $\mathbb{P}$  (hipótesis de inducción), entonces  $\neg(\varphi)$  cumple la propiedad  $\mathbb{P}$ .
- $(\circ)$ : Si  $\varphi$  y  $\psi$  son dos fórmulas que cumplen la propiedad  $\mathbb{P}$  (hipótesis de inducción), entonces  $(\varphi \circ \psi)$  cumple la propiedad  $\mathbb{P}$ .

Entonces, el **principio de inducción estructural para fórmulas proposicionales** afirma que, si se verifican las condiciones anteriores, se puede concluir que **toda fórmula bien construida** cumple la propiedad  $\mathbb{P}$ .

**Ejemplo 3.2.5 [HLR]** Usando el principio de inducción estructural podemos probar que toda fórmula  $\varphi$  contiene el mismo número de paréntesis abiertos y cerrados:

**Base de inducción (At):** Los símbolos  $\top$  y  $\perp$  y toda proposición atómica contienen 0 paréntesis abiertos y 0 paréntesis cerrados.

**Pasos de inducción:**

$(\neg)$ : Si  $\varphi$  es una fórmula que contiene  $n$  paréntesis abiertos y  $n$  paréntesis cerrados, entonces  $\neg(\varphi)$  contiene  $n + 1$  paréntesis abiertos y  $n + 1$  paréntesis cerrados.

$(\circ)$ : Si  $\varphi$  y  $\psi$  son dos fórmulas que contienen  $n_\varphi$  y  $n_\psi$  paréntesis abiertos y  $n_\varphi$  y  $n_\psi$  paréntesis cerrados respectivamente, entonces  $(\varphi \circ \psi)$  contiene  $n_\varphi + n_\psi + 1$  paréntesis abiertos y  $n_\varphi + n_\psi + 1$  paréntesis cerrados.

Se sigue que, por el principio de inducción estructural, toda fórmula  $\varphi$  contiene el mismo número de paréntesis abiertos y cerrados.

## 3.3 Representación de las fórmulas bien construidas

En este apartado vamos a ilustrar las maneras más usadas para representar las fórmulas de la lógica proposicional.

### 3.3.1 Fórmulas en forma usual y abreviada

#### Forma usual

Siguiendo las reglas de formación de las fórmulas proposicionales obtenemos ejemplos de expresiones bien construidas como las siguientes:

1.  $((p \wedge (p \rightarrow r)) \vee (q \leftrightarrow t)),$
2.  $(p \wedge (\neg(\neg((q \rightarrow r) \rightarrow (p \vee (\neg(r))))))))),$
3.  $(p \rightarrow (q \rightarrow r)).$

Se puede notar que se usan paréntesis para evitar ambigüedad en la formalización. Sin embargo este uso se puede relajar obteniendo expresiones que no son fórmulas bien construidas, pero vienen empleadas como tales por razones de convenios.

### Forma abreviada

Para reducir una fórmula bien construida a su **forma abreviada** podemos seguir los siguientes pasos:

- a) Se puede **omitir el par de paréntesis externo**.

Así, por ejemplo, las anteriores fórmulas 1., 2. y 3. se escribirían como:

- 1.1.  $(p \wedge (p \rightarrow r)) \vee (q \leftrightarrow t),$
- 2.1.  $p \wedge (\neg(\neg((q \rightarrow r) \rightarrow (p \vee (\neg(r))))))))),$
- 3.1.  $p \rightarrow (q \rightarrow r).$

- b) Se introducen las siguientes **reglas de precedencia** entre conectivos que definen las prioridades que tenemos que respetar a la hora de aplicarlos:

**Nivel 1:**  $\neg,$   
**Nivel 2:**  $\vee$  y  $\wedge,$   
**Nivel 3:**  $\rightarrow$  y  $\leftrightarrow.$

Así, por ejemplo, las anteriores fórmulas 1.1, 2.1. y 3.1 se escribirían como:

- 1.2.  $(p \wedge (p \rightarrow r)) \vee (q \leftrightarrow t),$
- 2.2.  $p \wedge \neg\neg((q \rightarrow r) \rightarrow p \vee \neg r),$
- 3.2.  $p \rightarrow (q \rightarrow r).$

- c) Se admite el **convenio de asociatividad**: los conectivos  $\vee$ ,  $\wedge$ ,  $\rightarrow$  y  $\leftrightarrow$  se asocian por la derecha:

$$p \wedge q \wedge r \quad \text{es} \quad p \wedge (q \wedge r)$$

$$p \vee q \vee r \quad \text{es} \quad p \vee (q \vee r)$$

$$p \rightarrow q \rightarrow r \quad \text{es} \quad p \rightarrow (q \rightarrow r).$$

Así, por ejemplo, las anteriores fórmulas 1.2, 2.2. y 3.2 se escribirían como:

- 1.3.  $(p \wedge (p \rightarrow r)) \vee (q \leftrightarrow t)$ ,
- 2.3.  $p \wedge \neg\neg((q \rightarrow r) \rightarrow p \vee \neg r)$ ,
- 3.3.  $p \rightarrow q \rightarrow r$ .

### 3.3.2 Fórmulas en forma de árbol estructural

El siguiente **principio de unicidad de estructura para fórmulas proposicionales** afirma que cada fórmula admite un análisis sintáctico único, es decir, existe una única manera de derivar una fórmula usando las reglas de formación dadas.

#### Principio de unicidad de estructura para fórmulas proposicionales

Toda fórmula proposicional  $\varphi$  pertenece a una y sólo una de las siguientes categorías:

1. (At):  $\varphi$  es atómica,
2. ( $\neg$ ):  $\varphi$  es  $\neg(\varphi_1)$  para cierta fórmula  $\varphi_1$ ,
3. ( $\circ$ ):  $\varphi$  es  $(\varphi_1 \circ \varphi_2)$  para cierto conectivo  $\circ$  y ciertas fórmulas  $\varphi_1$  y  $\varphi_2$ .

Además, en todos los casos la fórmulas  $\varphi_1$  y  $\varphi_2$  están unívocamente determinadas.

Una primera consecuencia del principio de unicidad de estructura es la posibilidad de representar fórmulas proposicionales por medio de árboles. Este tipo de representación nos permite facilitar el análisis de la estructura sintáctica de toda fórmula, ya que nos permite recorrer al revés los pasos de su construcción.

El alumno estudiará el concepto de árbol en detalle en él contexto de la teoría de grafos en la asignatura de *Matemática Discreta y Álgebra*. Aquí queremos simplemente introducir, por medio de ejemplos, árboles con raíz para realizar diagramas gráficos asociados a fórmulas proposicionales.

**Repaso de algunas definiciones y propiedades básicas sobre árboles [CM]:**

**Definición 3.3.1** *Un grafo simple  $G$  es un par  $G = (V, E)$  formado por un conjunto finito de **vértices**  $V$  y un conjunto  $E$  de pares no ordenados de vértices distintos, es decir,*

$$E \subset \{\{u, v\} \mid u, v \in V \wedge u \neq v\}.$$

*A los elementos de  $E$  se les denomina **aristas** (no dirigidas o no orientadas).*

**Definición 3.3.2** *El **grado** de un vértice en un grafo simple es el número de aristas incidentes con él.*

**Definición 3.3.3** *Un **camino de longitud  $n$**  entre los vértices  $a$  y  $b$  de un grafo no dirigido es una sucesión finita  $(e_0, \dots, e_{n-1})$  de aristas del grafo*

$$e_0 = \{v_0, v_1\}, e_1 = \{v_1, v_2\}, \dots, e_{n-1} = \{v_{n-1}, v_n\}$$

*de manera que  $v_0 = a$ ,  $v_n = b$  y cada arista sucesiva empieza donde terminó la anterior. Si el grafo es simple, el camino  $(e_0, \dots, e_{n-1})$  queda perfectamente determinado por la sucesión de vértices*

$$(a, v_1, v_2, \dots, v_{n-1}, b).$$

*Diremos que el camino anterior **pasa por** (o **atraviesa**) los vértices  $a$ ,  $v_1$ ,  $v_2$ , ...,  $v_{n-1}$ ,  $b$ .*

*Se dice que un camino es un **circuito** si es cerrado, esto es, empieza y termina en el mismo vértice, es decir, si  $a = b$ .*

*Se dice que un camino es **simple** si no contiene a la misma arista más de una vez.*

*Un circuito que no pasa dos veces por el mismo vértice (salvo el inicial por el que pasa dos veces) se llama **ciclo**.*

**Definición 3.3.4** *Un **árbol** es un grafo no dirigido, conexo y sin circuitos simples.*

**Proposición 3.3.5** *Sea  $G = (V, E)$  un grafo simple.  $G$  es un árbol si y solamente si para cada par de vértices  $u, v \in V$  existe un único camino simple que une  $u$  con  $v$ .*

**Definición 3.3.6** Un **árbol con raíz** es un par  $(T, r)$  donde  $T$  es un árbol y  $r$  un vértice distinguido de  $T$  llamado **raíz** al que se suele colocar en la representación gráfica en la parte superior.

**Definición 3.3.7** Dado un árbol con raíz  $(G, r)$  se denominan **hojas** a los vértices de  $G$  distintos de  $r$  que tienen grado 1.

A los vértices distintos de la raíz que no son hojas de un árbol con raíz se les denomina **vértices internos**.

Los árboles con raíz llevan asociada una terminología de origen botánico y genealógico: dado un árbol con raíz  $T$ , si  $v$  es un vértice de  $T$  distinto de la raíz, el **padre** de  $v$  es el único vértice  $u$  de  $T$  tal que hay una arista de  $u$  a  $v$ . Si  $u$  es el padre de  $v$ , también diremos que  $v$  es un **hijo** de  $u$ . Los **antecesores** de un vértice son los vértices que nos encontramos en el único camino que une dicho vértice con la raíz. Los **descendientes** de un vértice  $v$  son todos aquellos vértices de los que  $v$  es antecesor.

Es fácil darse cuenta de que las hojas de un árbol dirigido son los vértices que no tienen descendientes.

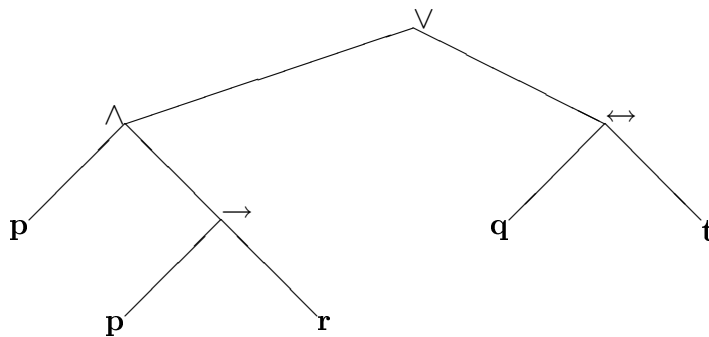


Figura 3.1: Árbol sintáctico de  $((p \wedge (p \rightarrow r)) \vee (q \leftrightarrow t))$ .

Vamos entonces a usar árboles con raíz para representar la estructura sintáctica de las fórmulas proposicionales como en el siguiente ejemplo.

**Ejemplo 3.3.8** Sea  $\varphi$  la fórmula proposicional

$$((p \wedge (p \rightarrow r)) \vee (q \leftrightarrow t)).$$

Para representar la estructura sintáctica de  $\varphi$ , podemos dibujar el árbol con raíz de la figura (3.1).

Las hojas del árbol obtenido representan las fórmulas atómicas a partir de las cuales se ha generado  $\varphi$ . El árbol se lee de abajo hacia arriba y en cada nodo interior y en la raíz aparecen los conectivos empleados en la formación de nuestra fórmula.

### 3.3.3 El principio de recursión estructural para fórmulas proposicionales

Una segunda consecuencia del principio de unicidad de estructura es el **principio de recursión estructural para fórmulas proposicionales** que nos permite **definir funciones**

$$f : \mathbf{L} \longrightarrow \mathbf{A}$$

cuyo dominio sea el conjunto  $\mathbf{L}$  de todas las fórmulas proposicionales y cuyo codominio sea un conjunto dado  $\mathbf{A}$ .

**El principio de recursión estructural para fórmulas proposicionales** consiste en la siguiente definición recursiva de la función  $f : \mathbf{L} \longrightarrow \mathbf{A}$  :

1. **Base (At):** Si  $\varphi$  es atómica,  $f(\varphi)$  se define explícitamente, es un elemento de  $\mathbf{A}$ ,
2. **Pasos recursivos:**
  - $(\neg)$ :  $f(\neg(\varphi))$  es un valor que depende de  $\neg$  y de  $f(\varphi_1)$ ,
  - $(\circ)$ :  $f(\varphi_1 \circ \varphi_2)$  para cierto conectivo  $\circ$  y ciertas fórmulas  $\varphi_1$  y  $\varphi_2$  es un valor que depende de  $\circ$ ,  $f(\varphi_1)$  y  $f(\varphi_2)$ .

Estas definiciones determinan la función  $f$  sobre todo  $\mathbf{L}$ .

**Ejemplo 3.3.9 (HLR)** *Se puede definir recursivamente la función*

$$f : \mathbf{L} \longrightarrow \mathbb{N} \cup \{\mathbf{0}\}$$

que a cada fórmula  $\varphi \in \mathbf{L}$  asocia el número (entero no negativo) de conectivos binarios en la estructura sintáctica de  $\varphi$  :



1. **Base (At):** Si  $\varphi$  es atómica,  $f(\varphi) = 0$

2. **Pasos recursivos:**

$$(\neg): f(\neg(\varphi)) = f(\varphi)$$

$(\circ): f(\varphi_1 \circ \varphi_2) = f(\varphi_1) + f(\varphi_2) + 1$ , para cierto conectivo  $\circ$  y ciertas fórmulas  $\varphi_1$  y  $\varphi_2$ .

Estas definiciones determinan la función  $f$  sobre todo  $L$ .

### 3.4 Formalización del lenguaje natural

En esta sección se presentan algunos ejemplos de formalización del lenguaje natural. La formalización es una herramienta básica y el alumno tendrá ocasión de practicarla a lo largo de todo el estudio de la lógica proposicional y de predicados.

La tabla (3.2) ([C]) contiene algunos ejemplos de formalización de frase simples. A partir de la formalización de estos ejemplos podemos formalizar frases y razonamientos más complejos.

Para formalizar un razonamiento con premisas  $p_1, p_2, \dots, p_n$  y conclusión  $q$  usaremos cualquiera de las dos formas:

$$\begin{array}{l} p_1 \\ p_2 \\ \vdots \\ p_n \\ \hline q \end{array} \quad \circ \quad p_1 \wedge p_2 \wedge \dots \wedge p_n \rightarrow q.$$

**Observación 3.4.1** Una manera sencilla de verificar la validez de la formalización de una frase del lenguaje natural es de volver a traducir al lenguaje natural la formalización obtenida.

Así, por ejemplo, consideremos la frase “No voy a la playa a menos que haga mucho calor.”

Siendo  $p =$  voy a la playa y  $q =$  hace mucho calor, la formalización  $(p \rightarrow q)$  es correcta y la formalización  $(q \rightarrow p)$  no es correcta. En efecto, la primera se lee como “Voy a la playa sólo si hace mucho calor” y la segunda como “Si hace mucho calor, entonces voy a la playa.” En la segunda formalización se ha intercambiado la condición necesaria (la conclusión) con la suficiente (la premisa).

**Ejemplos 3.4.2** 1) El enunciado “Si una función  $f$  es derivable en el intervalo  $[a, b]$ , entonces  $f$  es continua en  $[a, b]$ ,” se puede formalizar definiendo las fórmulas atómicas  $p =$  la función  $f$  es derivable en  $[a, b]$  y  $q =$  la función  $f$  es continua en  $[a, b]$  y aplicando el conectivo de implicación. Se obtiene  $(p \rightarrow q)$  y, en forma abreviada,  $p \rightarrow q$ .

2) La frase “Si salto por la ventana, o me hago daño o empiezo a volar” se puede formalizar por medio de las proposiciones atómicas  $p =$  salto por la ventana,  $q =$  me hago daño,  $r =$  empiezo a volar y los conectivos de implicación y de disyunción. Se obtiene  $(p \rightarrow (q \vee r))$  y, en forma abreviada,  $p \rightarrow (q \vee r)$ .

3) La frase “Si salto por la ventana me podría hacer daño, sin embargo empiezo a volar” se puede formalizar por medio de las proposiciones atómicas  $p =$  salto por la ventana,  $q =$  me podría hacer daño,  $r =$  empiezo a volar y los conectivos de implicación y de conjunción. Se obtiene  $((p \rightarrow q) \wedge r)$  y, en forma abreviada,  $(p \rightarrow q) \wedge r$ .

4) El enunciado “Condición necesaria y suficiente para que un número entero  $n$  sea par es que  $n$  sea divisible por 3” se puede formalizar por medio de las proposiciones atómicas  $p = n$  es un número entero,  $q = n$  es par,  $r = n$  es divisible por 3 y los conectivos de conjunción y doble implicación. Se obtiene  $((p \wedge q) \leftrightarrow r)$  y, en forma abreviada,  $p \wedge q \leftrightarrow r$ .

5) Consideremos el razonamiento “Me gusta el helado de fresa, pero también el de limon. Si hay sólo helado de chocolate lo comeré, a pesar de que no me guste. Por tanto, no comeré helado de fresa.” Para formalizar el razonamiento dado, definimos las proposiciones atómicas  $p =$  me gusta el helado de fresa,  $q =$  me gusta el helado de limon,  $r =$  hay sólo helado de chocolate,  $s =$  comeré helado de chocolate,  $t =$  me gusta el helado de chocolate,  $u =$  comeré helado de fresa. La formalización se puede escribir, en forma abreviada, como:

$$\begin{array}{l} p \wedge q \\ (r \rightarrow s) \wedge \neg t \end{array}$$

---


$$\neg u$$

o como:

$$p \wedge q \wedge (r \rightarrow s) \wedge \neg t \rightarrow \neg u$$

Expresiones en el lenguaje natural	Formalización
no $p$ es falso que $p$ no es cierto $p$	$\neg p$
$p$ y $q$ $p$ pero $q$ $p$ sin embargo $q$ $p$ no obstante $q$ $p$ a pesar de $q$	$p \wedge q$
o $p$ o $q$ o ambas cosas al menos $p$ o $q$ como mínimo $p$ o $q$	$p \vee q$
si $p$ entonces $q$ $p$ sólo si $q$ $q$ si $p$ $q$ es necesario para $p$ $p$ es suficiente para $q$ no $p$ a menos que $q$ no $p$ o $q$	$p \rightarrow q$
$p$ si y sólo si $q$ $p$ es necesario y suficiente para $q$ $q$ es necesario y suficiente para $p$	$p \leftrightarrow q$ o $(p \rightarrow q) \wedge (q \rightarrow p)$

Tabla 3.2: Tabla de formalización en la lógica proposicional

## 3.5 Ejercicios

**Ejercicio 3.5.1** *Determina cuáles de las siguientes frases son proposiciones:*

- a) *¿Puedes ir a la cafetería?*
- b) *Por favor, sea educado.*
- c) *El jardín es muy grande.*
- d) *¡La organización del evento es un desastre!*
- e) *¡Échame una mano!*

**Ejercicio 3.5.2** *Identifica cuáles de las siguientes palabras no son fórmulas proposicionales o fórmulas proposicionales abreviadas:*

- a)  $(p \rightarrow q) \wedge r) \vee (s \wedge t)$ ,
- b)  $(\neg(\neg(\neg(p \wedge (q \rightarrow \neg(r))))))$ ,
- c)  $\neg(p)$ ,
- d)  $p \neg \wedge q$ ,
- e)  $\neg p \vee q \vee s$ .

**Ejercicio 3.5.3** *Escribe en forma abreviada las siguientes fórmulas proposicionales:*

- a)  $((p \rightarrow q) \wedge r) \vee (s \wedge t)$ ,
- b)  $(p \rightarrow (q \vee r)) \wedge (((s \rightarrow t) \wedge r) \rightarrow q)$ ,
- c)  $((\neg(p \wedge q) \leftrightarrow r) \vee s)$ ,
- d)  $(\neg((\neg(p) \rightarrow q) \vee r))$ ,
- e)  $(\neg(p \vee q) \leftrightarrow (\neg(p) \wedge \neg(q)))$ .

**Ejercicio 3.5.4** *Escribe en forma no abreviada las siguientes fórmulas proposicionales:*

- a)  $\neg r \rightarrow q \vee t \vee s$ ,
- b)  $p \wedge q \wedge r \rightarrow s \rightarrow t$ ,
- c)  $(p \wedge q) \vee (r \leftrightarrow s) \rightarrow p$ ,
- d)  $\neg(\neg p \rightarrow q \vee r)$ ,
- e)  $p \wedge (q \vee r) \rightarrow s \vee t$ .

**Ejercicio 3.5.5** *Representa en forma de árbol estructural las siguientes fórmulas:*

- a)  $((p \vee q) \rightarrow r)$ ,
- b)  $((p \rightarrow q) \rightarrow ((r \rightarrow p) \rightarrow (r \rightarrow q)))$ ,
- c)  $(\neg(p \vee q) \leftrightarrow (\neg(p) \wedge \neg(q)))$ ,

- d)  $((\neg(\neg(\neg(p) \vee r) \vee q)) \rightarrow (\neg(p \wedge q) \vee r)),$   
 e)  $((p \vee r) \rightarrow (p \vee \neg(r))) \leftrightarrow (r \rightarrow p),$   
 f)  $(p \rightarrow (s \rightarrow p)).$

**Ejercicio 3.5.6** La **profundidad** de una fórmula proposicional  $\varphi$  es la longitud máxima de los caminos simples con vértice inicial la raíz del árbol sintáctico de  $\varphi$ .

Usando el principio de recursión estructural, define recursivamente la función  $fp$  que asigna a cada fórmula  $\varphi$  su profundidad.

**Sugerencia:** usa la función que calcula el máximo entre dos números enteros.

**Ejercicio 3.5.7 (HLR)** a) Usando el principio de recursión estructural, define recursivamente la función  $fn$  que asocia a cada fórmula  $\varphi$  el número (entero no negativo) de conectivos, sin contar los conectivos de aridad 0, que aparecen en la estructura sintáctica de  $\varphi$ .

b) Demuestra por inducción estructural que para toda fórmula  $\varphi$  se verifica que

$$fp(\varphi) \leq fn(\varphi),$$

donde la función  $fp$  es la función profundidad del ejercicio anterior.

En los siguientes ejercicios se pide formalizar los razonamientos dados. En cada caso, determinar las condiciones necesarias y las condiciones suficientes del razonamiento y volver a traducir al lenguaje natural la formalización obtenida:

**Ejercicio 3.5.8** *Voy al bar siempre que me apetezca y tenga tiempo libre.*

**Ejercicio 3.5.9** *No voy a correr a menos que tu vengas conmigo.*

**Ejercicio 3.5.10** *Un número entero es primo sólo si no tiene divisores no triviales.*

**Ejercicio 3.5.11** *Todos los alumnos estudian mucho y están alegres. Si los alumnos están alegres, entonces sacan buenas notas. Por tanto, los alumnos estudian mucho sólo si están alegres.*

**Ejercicio 3.5.12** *Ningún número es hermano de un perro o ningún gato es primo de un número. Si un perro es primo de un gato, entonces es hermano de un número. Por tanto, ningún gato es primo de un número.*

**Ejercicio 3.5.13** *[C] Si llueve las calles estarán vacías. Si las calles están vacías, el comercio obtiene pérdidas. Los músicos no podrían sobrevivir si los comerciantes no les contratasen para componer canciones para publicidad. Los comerciantes invierten en canciones publicitarias cuando tienen pérdidas. Por tanto, si llueve, los músicos pueden sobrevivir.*

# Capítulo 4

## Semántica de la lógica proposicional. Teoría interpretativa

En este capítulo vamos a estudiar la semántica y sus sistemas de demostración en la lógica proposicional. A continuación recordamos sus definiciones:

La **semántica** es la definición de un conjunto de significados (generalmente verdadero o falso) que se puedan asociar a una fórmula. Permite definir la validez de una fórmula o de un razonamiento.

Los **sistemas de demostración** son sistemas formales que permiten averiguar cuándo una fórmula o un razonamiento son válidos. En el contexto de la semántica se denominan **teoría interpretativa**.

Los sistemas de demostración se suelen dividir en dos clases: **sistemas directos** y **sistemas indirectos** (o por **refutación**). Los primeros aplican una cadena finita de reglas de inferencia hasta llegar a la fórmula que se quiere demostrar. Los segundos aplican la técnica de reducción al absurdo.

Como sistema de demostración directo estudiaremos las tablas de verdad y como sistema indirecto el método de los tableaux.

### 4.1 Valoraciones de un lenguaje formal

Recordamos que el conjunto  $\Sigma$  de los símbolos  $p, q, r, s, t, \dots$  que representen las proposiciones atómicas de una fórmula se suele denominar **signatura**:

$$\Sigma = \{p, q, r, s, t, \dots\}.$$

**Definición 4.1.1** Sea  $L$  el lenguaje de la lógica proposicional. Una **valoración del lenguaje  $L$**  es cualquier función

$$v : \Sigma \longrightarrow \{0, 1\}$$

de la signatura  $\Sigma$  al conjunto de dos elementos  $\{0, 1\}$ . El símbolo 0 representa el valor “falsedad” y el símbolo 1 el valor “verdad”.

**Ejemplo 4.1.2** Si  $\Sigma = \{p, q\}$ , podemos representar las cuatro posibles valoraciones de  $\Sigma$  :

$$\begin{array}{l} \Sigma \rightarrow \{0, 1\} \\ p \rightarrow 0 \\ q \rightarrow 0 \end{array} \quad \begin{array}{l} \Sigma \rightarrow \{0, 1\} \\ p \rightarrow 0 \\ q \rightarrow 1 \end{array} \quad \begin{array}{l} \Sigma \rightarrow \{0, 1\} \\ p \rightarrow 1 \\ q \rightarrow 0 \end{array} \quad \begin{array}{l} \Sigma \rightarrow \{0, 1\} \\ p \rightarrow 1 \\ q \rightarrow 1 \end{array}$$

por medio de las filas de la tabla a). De forma similar, si  $\Sigma = \{p, q, r\}$ , podemos representar las ocho posibles valoraciones de  $\Sigma$  por medio de las filas de la tabla b).

a)

$p$	$q$
0	0
0	1
1	0
1	1

b)

$p$	$q$	$r$
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

**Observación 4.1.3** Usando una demostración por inducción se puede verificar que si  $\Sigma$  contiene  $n$  elementos, entonces hay  $2^n$  posibles valoraciones de  $\Sigma$ .

## 4.2 Evaluación semántica de fórmulas

Dada una valoración  $v : \Sigma \longrightarrow \{0, 1\}$  nos interesa extender su definición a todas las fórmulas proposicionales definidas a partir de las proposiciones atómicas de  $\Sigma$ . Esto no permitirá establecer los **valores de verdad (veritativos)** de esas fórmulas. Como es de esperar, la definición de la extensión de una valoración tiene carácter recursivo.



**Nota:** En lo que se sigue usaremos la forma abreviada para las fórmulas proposicionales.

Como primer paso tenemos que definir, para toda valoración  $v : \Sigma \longrightarrow \{0, 1\}$ , los valores que toman los conectivos lógicos.

### Valores de verdad de los conectivos lógicos

( $\neg$ ): Los valores de verdad del conectivo lógico negación,  $v_{\neg p}$ , residen en que la fórmula  $\neg p$  es verdadera si y sólo si  $p$  es falsa y, recíprocamente, que  $\neg p$  es falsa si y sólo si  $p$  es verdadera. Los valores de verdad  $v_{\neg p}$  correspondientes al conectivo  $\neg$  se representan en la tercera columna de la tabla (4.1).

**Ejemplo 4.2.1** *Sea  $p = \text{Luis tiene 18 años}$ . Entonces  $\neg p = \text{Luis no tiene 18 años}$  es verdadera si es falso que Luis tiene 18 años y es falsa si Luis los tiene.*

( $\wedge$ ): La fórmula  $p \wedge q$  ( $p$  y  $q$ ) es verdadera si y sólo si  $p$  y  $q$  son verdaderas simultáneamente. Los valores de verdad  $v_{p \wedge q}$  correspondientes a este conectivo se representan en la cuarta columna de la tabla (4.1).

**Ejemplo 4.2.2** *Sean  $p = \text{Luis tiene 18 años}$  y  $q = \text{Maria es española}$ . Entonces  $p \wedge q = \text{Luis tiene 18 años y Maria es española}$  es verdadera sólo si es verdadero que Luis tiene 18 y es también verdadero que Maria es española.*

( $\vee$ ): La fórmula  $p \vee q$  ( $p$  ó  $q$ ), es verdadera si y sólo si  $p$  es verdadera o  $q$  es verdadera o ambas  $p$  y  $q$  son verdaderas. Los valores de verdad  $v_{p \vee q}$  correspondientes a este conectivo se representan en la quinta columna de la tabla (4.1).

**Ejemplo 4.2.3** *Sean  $p = \text{Luis tiene 18 años}$  y  $q = \text{Maria es española}$ . Entonces  $p \vee q = \text{Luis tiene 18 años ó Maria es española}$  es falsa sólo si es falso que Luis tiene 18 y es también falso que Maria es española.*

( $\rightarrow$ ): Para establecer el significado de la fórmula  $p \rightarrow q$  ( $p$  implica  $q$ ), debemos tener presente que en lenguaje natural este enunciado encierra una

relación de causalidad, que no siempre aparece al utilizar este conectivo en el ámbito formal.

En el lenguaje matemático,  $p$  implica  $q$  quiere decir que si  $p$  es verdadera, necesariamente  $q$  es verdadera, o lo que es lo mismo, que es imposible que  $q$  sea falsa si  $p$  es verdadera. Por tanto el único caso en el cual  $p \rightarrow q$  puede ser falsa es si  $p$  es verdadera y  $q$  es falsa.

Las fórmulas atómicas  $p$  y  $q$  son, respectivamente, el **antecedente o premisa** y el **consecuente o conclusión** de la fórmula  $p \rightarrow q$ .

La fórmula  $q \rightarrow p$  se denomina **sentencia recíproca** de la sentencia  $p \rightarrow q$ , y la sentencia  $\neg q \rightarrow \neg p$  **sentencia contrarrecíproca** de la sentencia  $p \rightarrow q$ .

Los valores de verdad  $v_{p \rightarrow q}$  correspondientes al conectivo  $p \rightarrow q$  se representan en la sexta columna de la tabla (4.1).

**Ejemplo 4.2.4** Sean  $p = \text{Luis tiene 18 años}$  y  $q = \text{Maria es española}$ . Entonces  $p \rightarrow q$  es falsa sólo si es verdadero que Luis tiene 18 y es falso que Maria es española.

( $\leftrightarrow$ ): La fórmula  $p \leftrightarrow q$  ( $p$  si y sólo si  $q$ ) es verdadera cuando  $p$  y  $q$  tienen el mismo valor de verdad. Los valores de verdad  $v_{p \leftrightarrow q}$  correspondientes al conectivo  $p \leftrightarrow q$  se representan en la última columna de la tabla (4.1).

**Ejemplo 4.2.5** Sean  $p = \text{Luis tiene 18 años}$  y  $q = \text{Maria es española}$ . Entonces  $p \leftrightarrow q$  es falsa si es verdadero que Luis tiene 18 y es falso que Maria es española o si es falso que Luis tiene 18 y es verdadero que Maria es española.

### 4.2.1 Tablas de verdad

Definidos los valores de verdad de los conectivos de la lógica proposicional, el principio de inducción estructural nos permite extender una valoración a todas las fórmulas proposicionales.

**Definición 4.2.6** Dada una valoración  $v : \Sigma \longrightarrow \{0, 1\}$ , se asocia a cada fórmula proposicional  $\varphi$  un valor de verdad  $(\varphi)^v \in \{0, 1\}$  de la siguiente forma:

$p$	$q$	$v_{\neg p}$	$v_{p \wedge q}$	$v_{p \vee q}$	$v_{p \rightarrow q}$	$v_{p \leftrightarrow q}$
0	0	1	0	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	0	0
1	1	0	1	1	1	1

Tabla 4.1: Valores de verdad de los conectivos

**Definición recursiva de una valoración**

(At):  $(\top)^v = 1$ ,  $(\perp)^v = 0$  y

$(p)^v = v(p)$  para toda proposición atómica  $p$ .

$(\neg)$ : Si  $\varphi$  es una fórmula entonces  $(\neg(\varphi))^v = v_{\neg}(\varphi)$ .

$(\circ)$ : Si  $\varphi$  y  $\psi$  son dos fórmulas entonces  $(\varphi \circ \psi)^v = v_{\varphi \circ \psi}(\varphi \circ \psi)$ .

**Construcción de una tabla de verdad**

La **tabla de verdad** de una fórmula proposicional  $\varphi$  es una forma de representar todos los posibles valores de verdad que  $\varphi$  puede tomar en todas las posibles valoraciones de las proposiciones atómicas de su signatura.

La construcción de la tabla de verdad de una fórmula  $\varphi$  está basada en la definición recursiva de la valoración de  $\varphi$ . Por tanto, necesitamos seguir varios pasos para completarla:

**Paso 1:** Tenemos que identificar las proposiciones atómicas y los pasos que se han seguido para construir  $\varphi$ . Para eso podemos usar el árbol estructural de  $\varphi$ .

**Ejemplo 4.2.7** *Volvamos a considerar la fórmula*

$$\varphi = ((p \wedge (p \rightarrow r)) \vee (q \leftrightarrow t))$$

y la figura (4.1).

La proposiciones atómicas son  $p$ ,  $q$ ,  $r$  y  $t$  y la construcción de  $\varphi$  se obtiene leyendo su árbol de abajo hacía arriba.

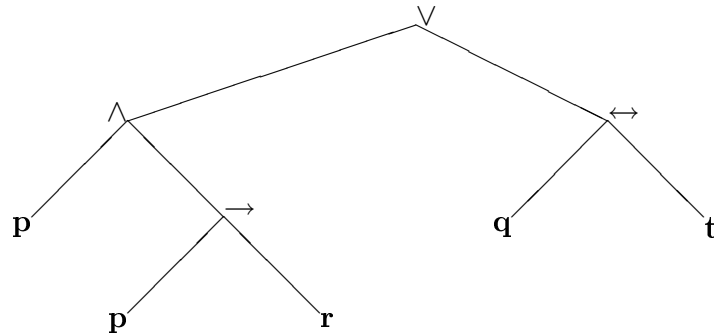


Figura 4.1: Árbol estructural de  $\varphi = ((p \wedge (p \rightarrow r)) \vee (q \leftrightarrow t))$

**Paso 2:** Siguiendo el orden de construcción de  $\varphi$  se escribe la primera fila de su tabla de verdad.

Para nuestro ejemplo se obtiene:

$p$	$q$	$r$	$t$	$p \rightarrow r$	$p \wedge (p \rightarrow r)$	$q \leftrightarrow t$	$(p \wedge (p \rightarrow r)) \vee (q \leftrightarrow t)$
-----	-----	-----	-----	-------------------	------------------------------	-----------------------	---

**Paso 3:** Se rellenan las columnas que se corresponden a las proposiciones atómicas con todas sus posibles valoraciones.

Para nuestro ejemplo serían las primeras cuatro columnas de la tabla (4.2), que contiene  $2^4 + 1 = 17$  filas.

**Paso 4:** Se rellenan todas las columnas siguiendo las valoraciones de los conectivos lógicos de la tabla (4.1) y, en cada fila, las valoraciones de las proposiciones atómicas.

Para nuestro ejemplo se obtiene la tabla de verdad (4.2).

$p$	$q$	$r$	$t$	$p \rightarrow r$	$p \wedge (p \rightarrow r)$	$q \leftrightarrow t$	$(p \wedge (p \rightarrow r)) \vee (q \leftrightarrow t)$
0	0	0	0	1	0	1	1
0	0	0	1	1	0	0	0
0	0	1	0	1	0	1	1
0	0	1	1	1	0	0	0
0	1	0	0	1	0	0	0
0	1	0	1	1	0	1	1
0	1	1	0	1	0	0	0
0	1	1	1	1	0	1	1
1	0	0	0	0	0	1	1
1	0	0	1	0	0	0	0
1	0	1	0	1	1	1	1
1	0	1	1	1	1	0	1
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	1	1	0	1
1	1	1	1	1	1	1	1

Tabla 4.2: Tabla de verdad de  $\varphi = (p \wedge (p \rightarrow r)) \vee (q \leftrightarrow t)$ .

### 4.3 Modelos y contraejemplos de una fórmula bien construida. Tautologías, contingencias y contradicciones

La posibilidad de valorar fórmulas proposicionales nos permite definir las siguientes nociones fundamentales.

**Definición 4.3.1** *Se dice que una fórmula  $\varphi$  es satisfacible bajo una valoración  $v$  (o que  $v$  es un modelo de  $\varphi$ ) cuando se verifica que*

$$(\varphi)^v = 1.$$

*Si  $\varphi$  es satisfacible bajo  $v$  se emplea la notación  $v \models \varphi$ .*

**Ejemplo 4.3.2** En la tabla de verdad de la tabla (4.2), las valoraciones

$$\begin{array}{l} \Sigma \rightarrow \{0, 1\} \\ p \rightarrow 0 \\ q \rightarrow 0 \\ r \rightarrow 0 \\ s \rightarrow 0 \end{array} \quad y \quad \begin{array}{l} \Sigma \rightarrow \{0, 1\} \\ p \rightarrow 0 \\ q \rightarrow 0 \\ r \rightarrow 1 \\ s \rightarrow 0 \end{array}$$

son dos de los diez posibles modelos de la fórmula  $\varphi = (p \wedge (p \rightarrow r)) \vee (q \leftrightarrow t)$ .

**Definición 4.3.3** Se dice que una valoración  $v$  es un **contraejemplo** de una fórmula  $\varphi$  cuando se verifica que  $(\varphi)^v = 0$ .

**Ejemplo 4.3.4** En la tabla de verdad de la tabla (4.2), las valoraciones

$$\begin{array}{l} \Sigma \rightarrow \{0, 1\} \\ p \rightarrow 0 \\ q \rightarrow 0 \\ r \rightarrow 0 \\ s \rightarrow 1 \end{array} \quad y \quad \begin{array}{l} \Sigma \rightarrow \{0, 1\} \\ p \rightarrow 0 \\ q \rightarrow 0 \\ r \rightarrow 1 \\ s \rightarrow 1 \end{array}$$

son dos de los seis posibles contraejemplos de la fórmula  $\varphi = (p \wedge (p \rightarrow r)) \vee (q \leftrightarrow t)$ .

**Definición 4.3.5** Se dice que una fórmula  $\varphi$  es **satisfacible** cuando es satisfacible bajo alguna valoración  $v$ . En caso contrario se dice que es **insatisfacible**.

**Ejemplo 4.3.6** La fórmula  $\varphi = p \wedge \neg p$  es insatisfacible, ya que  $p$  y  $\neg p$  no pueden ser verdaderos a la vez.

**Observación 4.3.7** Las definiciones anteriores se pueden extender a un conjunto de fórmulas  $\Phi = \{\varphi_1, \varphi_2, \dots, \varphi_n\}$  pidiendo que la satisfacibilidad de  $\Phi$  sea la satisfacibilidad simultánea de todas las fórmulas que lo definen.

Entonces,

- 1)  $\Phi$  es satisfacible si y sólo si  $\varphi_1 \wedge \varphi_2 \wedge \dots \wedge \varphi_n$  es satisfacible.
- 2)  $\Phi$  es insatisfacible si y sólo si  $\varphi_1 \wedge \varphi_2 \wedge \dots \wedge \varphi_n$  es insatisfacible.

**Ejemplo 4.3.8** Sea  $v(p) = 0, v(q) = 1, v(r) = 1$  una valoración del conjunto  $\{p, q, r\}$ .

Entonces  $v$  es un modelo del conjunto de dos fórmulas  $\Phi_1 = \{(p \rightarrow q) \wedge r, q \wedge r\}$  y es un contraejemplo del conjunto  $\Phi_2 = \{p \wedge (q \rightarrow r), q \wedge r\}$ .

**Definición 4.3.9** Sea  $\varphi$  una fórmula proposicional construida a partir de un conjunto  $\Sigma = \{p_1, p_2, p_3, \dots\}$  de proposiciones atómicas.

1. Se dice que la fórmula  $\varphi$  es **lógicamente válida** o una **tautología** cuando es verdadera bajo cualquier valoración, es decir, si  $v \models \varphi$  para toda valoración  $v$  de  $\Sigma$ .
2. Se dice que  $\varphi$  es una **contradicción** cuando es falsa bajo cualquier valoración, es decir, si toda valoración  $v$  de  $\Sigma$  es un contraejemplo de  $\varphi$ . Por tanto una contradicción es una fórmula insatisfacible.
3. Se dice que  $\varphi$  es una **contingencia** cuando entre las valoraciones de  $\Sigma$  existen al menos un modelo y al menos un contraejemplo de  $\varphi$ .

**Ejemplos 4.3.10** 1) Por definición,

$\top$  es una tautología y  
 $\perp$  es una contradicción.

2) Para toda fórmula  $\varphi$ ,

$\varphi \vee \neg\varphi$  es una tautología (ley del tercio excluso) y  
 $\varphi \wedge \neg\varphi$  es una contradicción.

3) Usando una tabla de verdad se puede verificar que vale la ley conmutativa para el conectivo  $\wedge$ , es decir, que

$$p \wedge q \leftrightarrow q \wedge p$$

es una fórmula lógicamente válida.

$p$	$q$	$p \wedge q$	$q \wedge p$	$p \wedge q \leftrightarrow q \wedge p$
0	0	0	0	1
0	1	0	0	1
1	0	0	0	1
1	1	1	1	1

4) La tabla de verdad (4.2) de  $\varphi = (p \wedge (p \rightarrow r)) \vee (q \leftrightarrow t)$  nos indica que  $\varphi$  es una contingencia.

## 4.4 Evaluación semántica de deducciones. Consecuencia lógica

Otro concepto fundamental que vamos a definir es el concepto de consecuencia lógica.

**Definición 4.4.1** Se dice que una fórmula  $\varphi$  es **consecuencia lógica** de un conjunto finito de fórmulas  $\Phi = \{\varphi_1, \varphi_2, \dots, \varphi_n\}$  si todo modelo  $v$  del conjunto  $\Phi$  es un modelo de la fórmula  $\varphi$ , es decir, si

$$v \models \Phi \quad \text{implica que} \quad v \models \varphi.$$

Si  $\varphi$  es consecuencia lógica de  $\Phi$  también se dice que  $\Phi$  **implica lógicamente** a  $\varphi$  y se escribe

$$\Phi \models \varphi.$$

**Ejemplo 4.4.2** La tabla (4.3) contiene algunos importantes ejemplos de implicaciones lógicas (ver ejercicio (4.7.4)).

**Definición 4.4.3** Sean  $\Phi = \{\varphi_1, \varphi_2, \dots, \varphi_n\}$  un conjunto finito de fórmulas proposicionales y  $\varphi$  una fórmula. Se define **deducción** o **razonamiento** a la fórmula

$$((\varphi_1 \wedge \varphi_2 \wedge \dots \wedge \varphi_n) \rightarrow \varphi).$$

Para distinguir las premisas del conjunto  $\Phi$  de la conclusión  $\varphi$ , un razonamiento se puede representar de la siguiente manera:

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

**Observación 4.4.4** Decir que  $((\varphi_1 \wedge \varphi_2 \wedge \dots \wedge \varphi_n) \rightarrow \varphi)$  es una tautología significa que es imposible que las fórmulas de  $\Phi$  tengan todas el valor de verdad 1 y  $\varphi$  tenga valor 0. Por tanto  $((\varphi_1 \wedge \varphi_2 \wedge \dots \wedge \varphi_n) \rightarrow \varphi)$  es una tautología si y sólo si  $\Phi$  implica lógicamente a  $\varphi$ .

Por el otro lado, una fórmula  $\varphi$  es siempre consecuencia lógica de un cualquier conjunto de fórmulas  $\Phi$  insatisfacible.



$\neg\neg p \models p,$	Ley de la doble negación
$(p \wedge q) \models p$ $p \models (p \vee q)$	Leyes de simplificación
$(p \rightarrow q) \models (\neg q \rightarrow \neg p)$	Ley de contraposición
$((p \rightarrow q) \wedge (q \rightarrow r)) \models (p \rightarrow r)$	Ley transitiva de $\rightarrow$
$((p \wedge q) \rightarrow r) \models (p \rightarrow (q \rightarrow r))$	Ley de exportación
$(p \wedge (p \rightarrow q)) \models q$	Modus ponens
$((p \rightarrow q) \wedge \neg q) \models \neg p$	Modus tolens

Tabla 4.3: Tabla de algunas implicaciones lógicas

**Definición 4.4.5** *Se dice que el razonamiento anterior es **correcto** o **lógicamente válido** si*

$$(\varphi_1 \wedge \varphi_2 \wedge \cdots \wedge \varphi_n) \models \varphi,$$

*es decir, si el conjunto  $\Phi$  de las **premisas** o **hipótesis** del razonamiento implica lógicamente a la **conclusión** o **tesis**  $\varphi$ .*

**Observación 4.4.6** *Observar que, según la definición anterior, un razonamiento*

$$((\varphi_1 \wedge \varphi_2 \wedge \cdots \wedge \varphi_n) \rightarrow \varphi)$$

*es correcto si y sólo si es una tautología.*

**Ejemplos 4.4.7** 1) *El razonamiento (Modus ponens)*

$$\frac{p \quad p \rightarrow q}{q}$$

*es válido ya que  $p \wedge (p \rightarrow q) \rightarrow q$  es una tautología.*

2) *Todas las implicaciones de la tabla (4.3) son razonamientos correctos.*

## 4.5 Equivalencia de fórmulas

En este apartado vamos a estudiar el concepto de equivalencia lógica entre fórmulas. Veremos que la equivalencia de fórmulas es una relación binaria de equivalencia en el sentido de las definiciones del capítulo (2). Por tanto, dos fórmulas equivalente pertenecen a una misma clase de equivalencia.

**Definición 4.5.1** Sean  $\varphi$  y  $\psi$  dos fórmulas proposicionales. Se dice que  $\varphi$  **equivale lógicamente** a  $\psi$  si la fórmula proposicional  $(\varphi \leftrightarrow \psi)$  es una tautología. Si  $\varphi$  y  $\psi$  son equivalentes se escribe  $\varphi \equiv \psi$ .

**Observación 4.5.2** Se sigue de la definición que si  $\varphi$  y  $\psi$  son dos fórmulas equivalentes, entonces **tienen los mismos valores de verdad bajo una cualquier valoración**.

$\varphi \wedge \top \equiv \varphi$ $\varphi \wedge \perp \equiv \perp$ $\varphi \vee \top \equiv \top$ $\varphi \vee \perp \equiv \varphi$	Leyes de Identidad
$\varphi \wedge \neg\varphi \equiv \perp$	No contradicción
$\varphi \vee \neg\varphi \equiv \top$	Tercio excluso
$\varphi \wedge \varphi \equiv \varphi$ $\varphi \vee \varphi \equiv \varphi$	Idempotencia
$\varphi_1 \wedge (\varphi_1 \vee \varphi_2) \equiv \varphi_1$ $\varphi_1 \vee (\varphi_1 \wedge \varphi_2) \equiv \varphi_1$	Absorción
$\varphi_1 \wedge \varphi_2 \equiv \varphi_2 \wedge \varphi_1$ $\varphi_1 \vee \varphi_2 \equiv \varphi_2 \vee \varphi_1$ $(\varphi_1 \leftrightarrow \varphi_2) \equiv (\varphi_2 \leftrightarrow \varphi_1)$	Conmutatividad
$(\varphi_1 \wedge \varphi_2) \wedge \varphi_3 \equiv \varphi_1 \wedge (\varphi_2 \wedge \varphi_3)$ $(\varphi_1 \vee \varphi_2) \vee \varphi_3 \equiv \varphi_1 \vee (\varphi_2 \vee \varphi_3)$	Asociatividad
$\varphi_1 \wedge (\varphi_2 \vee \varphi_3) \equiv (\varphi_1 \wedge \varphi_2) \vee (\varphi_1 \wedge \varphi_3)$ $\varphi_1 \vee (\varphi_2 \wedge \varphi_3) \equiv (\varphi_1 \vee \varphi_2) \wedge (\varphi_1 \vee \varphi_3)$ $\varphi_1 \rightarrow (\varphi_2 \wedge \varphi_3) \equiv (\varphi_1 \rightarrow \varphi_2) \wedge (\varphi_1 \rightarrow \varphi_3)$ $\varphi_1 \rightarrow (\varphi_2 \vee \varphi_3) \equiv (\varphi_1 \rightarrow \varphi_2) \vee (\varphi_1 \rightarrow \varphi_3)$ $\varphi_1 \leftrightarrow (\varphi_2 \wedge \varphi_3) \equiv (\varphi_1 \leftrightarrow \varphi_2) \wedge (\varphi_1 \leftrightarrow \varphi_3)$ $\varphi_1 \leftrightarrow (\varphi_2 \vee \varphi_3) \equiv (\varphi_1 \leftrightarrow \varphi_2) \vee (\varphi_1 \leftrightarrow \varphi_3)$	Distributividad
$\neg(\neg\varphi) \equiv \varphi$	Doble negación
$(\varphi_1 \rightarrow \varphi_2) \equiv (\neg\varphi_2 \rightarrow \neg\varphi_1)$	Ley de contraposición
$\neg(\varphi_1 \wedge \varphi_2) \equiv \neg\varphi_1 \vee \neg\varphi_2$ $\neg(\varphi_1 \vee \varphi_2) \equiv \neg\varphi_1 \wedge \neg\varphi_2$	Leyes de De Morgan

Tabla 4.4: Tabla de algunas equivalencias lógicas

**Ejemplo 4.5.3** Vamos a verificar la siguiente importante equivalencia lógica, llamada **interdefinición**:

$$(p \rightarrow q) \equiv (\neg p \vee q).$$

El único caso en el cual  $(p \rightarrow q)$  es falsa es cuando  $p$  es verdadera y  $q$  es falsa.

El único caso en el cual  $(\neg p \vee q)$  es falsa es si  $\neg p$  es falsa y  $q$  es falsa. Esto es, el caso  $p$  verdadera y  $q$  falsa.

Ya que las dos fórmulas toman los mismos valores de verdad bajo una cualquier valoración, se sigue la equivalencia lógica de las dos fórmulas.

**Observación 4.5.4** Recordamos que una relación binaria  $R$  sobre un conjunto no vacío  $A$  se dice de equivalencia si es:

*Reflexiva:* para todo elemento  $a$  del conjunto  $A$ ,  $R(a, a)$  es verdadera,

*Simétrica:* para todo par  $a$  y  $b$  de elementos de  $A$ ,

$R(a, b) \rightarrow R(b, a)$  es verdadera.

*Transitiva:* para toda terna  $a$ ,  $b$  y  $c$  de elementos de  $A$ ,

$(R(a, b) \wedge R(b, c)) \rightarrow R(a, c)$  es verdadera.

En el ejercicio (4.7.8) se pide demostrar que en el conjunto  $\mathbf{L}$  de las fórmulas bien construidas, la relación

$$\varphi \equiv \psi \quad \text{si y sólo si} \quad (\varphi \leftrightarrow \psi \quad \text{es una tautología})$$

es una relación de equivalencia.

**Ejemplo 4.5.5** La tabla (4.4) contiene una lista de las equivalencias lógicas más utilizadas (ver ejercicio (4.7.9)).

## 4.6 Métodos de refutación. Tableaux

### 4.6.1 Refutación

Los métodos de demostración por refutación pertenecen a los sistemas de demostración indirectos que, como ya comentamos, son más modernos que los métodos de demostración axiomáticos y más adecuados para su automatización. Estos métodos nos proporcionan así una nueva forma de verificar la validez de fórmulas y de razonamientos.

Un ejemplo de sistema de demostración por refutación es la teoría de los tableaux, que vamos a estudiar en las siguientes secciones.

El alumno estudiará el método de demostración por refutación llamado *resolución proposicional* en la asignatura de Lógica Informática.

Vamos ahora a ver cómo se pueden emplear métodos indirectos para verificar la validez de una fórmula o de un razonamiento.

**Procedimiento por reducción al absurdo de demostración de la validez de una fórmula  $\varphi$ .**

Si la existencia de un modelo de  $\neg\varphi$  implica una contradicción, entonces podemos refutar  $\neg\varphi$  ( $\neg\varphi$  es una contradicción) y afirmar la validez de  $\varphi$  ( $\varphi$  es una tautología).

**Ejemplo 4.6.1** *Si queremos demostrar que la fórmula*

$$\varphi : \neg(p \wedge (p \vee q)) \vee p$$

*es una tautología, podemos usar un razonamiento por reducción al absurdo. Se trata de suponer que exista un modelo para  $\neg\varphi$ . Tal modelo es un contraejemplo para  $\varphi$  y, por tanto, tiene que ser tal que  $\neg(p \wedge (p \vee q))$  y  $p$  sean falsas. Pero  $p \wedge (p \vee q)$  verdadera implica que  $p$  es verdadera. Así que  $p$  tendría que ser verdadera y falsa al mismo tiempo y esto es imposible.*

**Procedimiento por reducción al absurdo de demostración de la validez de un razonamiento.**

El siguiente teorema afirma que  $\Phi \rightarrow \varphi$  es una tautología (una implicación lógica) si y sólo si no pueden ser simultáneamente verdaderas todas sus premisas y la negación de su conclusión:

**Teorema 4.6.2** [HLR] *Sean  $\Phi = \{\varphi_1, \varphi_2, \dots, \varphi_n\}$  un conjunto finito de fórmulas proposicionales y  $\varphi$  una fórmula. Entonces las siguientes son equivalentes:*

- $((\varphi_1 \wedge \varphi_2 \wedge \dots \wedge \varphi_n) \rightarrow \varphi)$  es una **fórmula válida** (una tautología, un razonamiento válido).
- $((\varphi_1 \wedge \varphi_2 \wedge \dots \wedge \varphi_n) \wedge \neg(\varphi))$  es una **contradicción** (Reducción al absurdo).

**Observación 4.6.3** *Se puede notar que, por interdefinición, la fórmula*

$$((\varphi_1 \wedge \varphi_2 \wedge \cdots \wedge \varphi_n) \rightarrow \varphi)$$

*es equivalente a la fórmula*

$$(\neg(\varphi_1 \wedge \varphi_2 \wedge \cdots \wedge \varphi_n) \vee \varphi).$$

*Por tanto, su negación es equivalente a la fórmula*

$$((\varphi_1 \wedge \varphi_2 \wedge \cdots \wedge \varphi_n) \wedge \neg(\varphi)).$$

**Ejemplo 4.6.4** *Para verificar que*

$$p \models (q \rightarrow p),$$

*hace falta demostrar que la fórmula*

$$(p \rightarrow (q \rightarrow p))$$

*es una tautología.*

*Aplicando el teorema (4.6.2) (el método de refutación), es suficiente verificar que la fórmula*

$$p \wedge \neg(q \rightarrow p)$$

*es insatisfacible (es una contradicción).*

*Para toda valoración tal que  $\neg(q \rightarrow p)$  es falsa,  $p \wedge \neg(q \rightarrow p)$  es falsa. Si una valoración es tal que  $\neg(q \rightarrow p)$  es verdadera, entonces  $(q \rightarrow p)$  es falsa, es decir, tiene que ser  $q$  verdadera y  $p$  falsa. También en este caso nuestra fórmula  $p \wedge \neg(q \rightarrow p)$  resultaría ser falsa. Se sigue que  $p \wedge \neg(q \rightarrow p)$  no admite ningún modelo y, por tanto, es insatisfacible.*

## 4.6.2 Definición de los tableaux

Como veremos, los tableaux son métodos de naturaleza sintáctica, pero se suelen presentar también por medio de una definición semántica.

Los **tableaux semánticos** se basan sobre el teorema (4.6.2) y son un procedimiento sistemático para verificar si una fórmula es insatisfacible.

Dada una implicación  $\varphi \rightarrow \psi$ , su negación  $\varphi \wedge \neg\psi$  es insatisfacible si y sólo si  $\varphi \rightarrow \psi$  es una implicación lógica.

Más en general, si una fórmula es insatisfacible, su negación es una tautología y, por tanto, un tableau permite averiguar si una fórmula es lógicamente válida.

Además, en muchos casos los tableaux son más eficientes que las tablas de verdad (donde para  $n$  proposiciones atómicas tenemos  $2^n$  posibles valoraciones), proporcionan una teoría para programar herramientas de demostración automática y tienen una extensión natural a la lógica de predicados, que estudiaremos en la segunda parte de estos apuntes.

Otras aplicaciones de la teoría de los tableaux semánticos son la clasificación de fórmulas proposicionales (en satisfacibles, insatisfacibles, tautologías o contingencias) y la obtención de sus formas normales conjuntivas o disyuntivas.

Durante el estudio de los tableaux semánticos descubriremos que la verdadera naturaleza de las reglas que los definen es sintáctica y que, por tanto, la presentación semántica (y no sintáctica) de la teoría de los tableaux tiene su justificación en su mayor simplicidad y claridad.

Las principales referencias utilizadas para la siguiente descripción de la teoría de tableaux son [HLR] y [MH].

## Formas conjuntivas y disyuntivas

Antes de poder definir la teoría de los tableaux semánticos, necesitamos profundizar en el análisis semántico de las fórmulas proposicionales.

En el ejercicio (4.7.11) se pide demostrar que toda fórmula proposicional es equivalente a otra donde intervienen sólo los conectivos  $\neg$ ,  $\wedge$  y  $\vee$ .

Esta propiedad se suele expresar diciendo que el conjunto  $\{\neg, \wedge, \vee\}$  es un conjunto **completo o adecuado** de conectivos para la lógica proposicional.

El anterior resultado justifica las siguientes definiciones, que nos permiten clasificar más fácilmente las fórmulas proposicionales:

### Definición 4.6.5 (Fórmulas conjuntivas y disyuntivas)

*Si una fórmula proposicional  $\alpha$  es equivalente a una conjunción de otras dos fórmulas más sencillas,  $\alpha \equiv \alpha_1 \wedge \alpha_2$ , diremos que es una **fórmula conjuntiva** (de la categoría  $\alpha$ ).*

Si una fórmula proposicional  $\beta$  es equivalente a una disyunción de otras dos fórmulas más sencillas,  $\beta \equiv \beta_1 \vee \beta_2$ , diremos que es una **fórmula disyuntiva** (de la categoría  $\beta$ ).

Si una fórmula proposicional  $\sigma$  es del tipo  $\neg\top$ ,  $\neg\perp$  o  $\neg\neg\varphi$  diremos que es **simplificable** (de la categoría  $\sigma$ ) y su forma simplificada es  $\sigma_1$ , que es igual a  $\perp$ ,  $\top$  o  $\varphi$ , respectivamente.

Las fórmulas conjuntivas, disyuntivas y simplificables se llaman **fórmulas reducibles**.

Usando las equivalencias lógicas estudiadas, podemos recoger en la tabla (4.5) todas las posibles fórmulas conjuntivas, disyuntivas y simplificables.

Fórmulas conjuntivas			Fórmulas disyuntivas		
$\alpha$	$\alpha_1$	$\alpha_2$	$\beta$	$\beta_1$	$\beta_2$
$\varphi \wedge \psi$	$\varphi$	$\psi$	$\varphi \vee \psi$	$\varphi$	$\psi$
$\neg(\varphi \vee \psi)$	$\neg\varphi$	$\neg\psi$	$\neg(\varphi \wedge \psi)$	$\neg\varphi$	$\neg\psi$
$\neg(\varphi \rightarrow \psi)$	$\varphi$	$\neg\psi$	$\varphi \rightarrow \psi$	$\neg\varphi$	$\psi$
$\varphi \leftrightarrow \psi$	$\varphi \rightarrow \psi$	$\psi \rightarrow \varphi$	$\neg(\varphi \leftrightarrow \psi)$	$\neg(\varphi \rightarrow \psi)$	$\neg(\psi \rightarrow \varphi)$

Fórmulas simplificables	
$\sigma$	$\sigma_1$
$\neg\top$	$\perp$
$\neg\perp$	$\top$
$\neg\neg\varphi$	$\varphi$

Tabla 4.5: Fórmulas reducibles

Usando árboles con raíz, podemos ahora representar las fórmulas reducibles de la tabla (4.5) por medio del esquema de la tabla (4.6).

**Observación 4.6.6** *El hecho de que toda fórmula proposicional es equivalente a otra donde intervienen sólo los conectivos  $\neg$ ,  $\wedge$  y  $\vee$  implica que toda fórmula proposicional es equivalente a una fórmula conjuntiva, a una fórmula disyuntiva o es simplifiable.*

**Ejemplo 4.6.7** *Sea*

$$\varphi \rightarrow (\psi \leftrightarrow \chi)$$

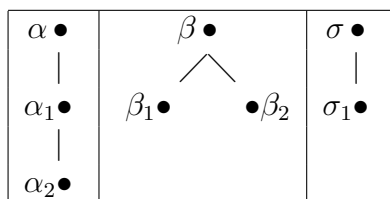


Tabla 4.6: Esquema de reducción de fórmulas

la fórmula proposicional que se quiere reducir. La regla de interdefinición nos permite reescribir nuestra fórmula de forma equivalente como una fórmula disyuntiva

$$\beta \equiv \neg\varphi \vee (\psi \leftrightarrow \chi),$$

donde

$$\beta_1 : \neg\varphi \quad y \quad \beta_2 : (\psi \leftrightarrow \chi).$$

### Tableaux semánticos

Dado un conjunto finito de fórmulas proposicionales  $\Phi$ , queremos asociar a este conjunto un árbol con raíz que nos permita determinar fácilmente propiedades semánticas del conjunto  $\Phi$ . Este árbol será el tableau asociado a  $\Phi$ .

En el capítulo (3) introducimos el concepto de árbol con raíz y sus propiedades básicas. En este capítulo necesitamos ampliar un poco más la terminología ya introducida.

**Definición 4.6.8** 1) Una **rama** de un árbol con raíz es un cualquier camino simple que no pueda prolongarse a otro más largo.

2) Un árbol con raíz es **binario** si todos sus vértices no tienen más que dos hijos.

3) Un **árbol de fórmulas**  $T$  es cualquier árbol con raíz binario que tenga asociada a cada uno de sus vértices una fórmula proposicional (cada uno de sus vértice está etiquetado por una fórmula proposicional).

4) Una rama  $\theta$  de un árbol de fórmulas  $T$  se dice **cerrada** si  $\perp$  aparece en  $\theta$ , o si ambas fórmulas  $\varphi$  y  $\neg\varphi$  aparecen en  $\theta$ .

5) Una rama  $\theta$  de un árbol de fórmulas  $T$  se dice **abierto** si no es cerrada.



Antes de definir formalmente el método de los tableaux, vamos a ver un ejemplo de como se construye un tableau asociado a un razonamiento.

**Ejemplo 4.6.9** (*Ejemplo de construcción de un tableau asociado a un razonamiento*)

Consideramos el razonamiento:

- 1) Si quiero comer fruta y hay una frutería cerca, entonces soy feliz.
- 2) No soy feliz.
- 3) Quiero comer fruta.

Por tanto,

- 4) No hay una frutería cerca.

Sean  $p =$  quiero comer fruta,  $q =$  hay una frutería cerca y  $r =$  soy feliz.  
El razonamiento anterior será válido si

$$\{p \wedge q \rightarrow r, \neg r, p\} \models \neg q.$$

Ya que los tableaux usan el método de refutación, podemos afirmar la validez del razonamiento anterior si demostramos que el conjunto de fórmulas

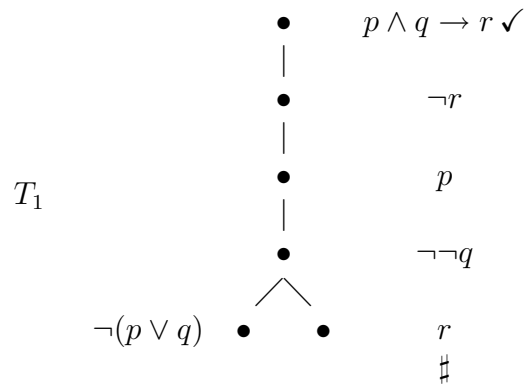
$$\Phi = \{p \wedge q \rightarrow r, \neg r, p, \neg \neg q\}$$

es insatisfacible.

Por eso, empezamos a construir un árbol de fórmulas  $T_0$  con la **regla de inicialización** que consiste en dibujar un primer árbol con una única rama cuyos vértices están etiquetados por las fórmulas de nuestro conjunto  $\Phi$ .  $T_0$  es nuestro tableau inicial:

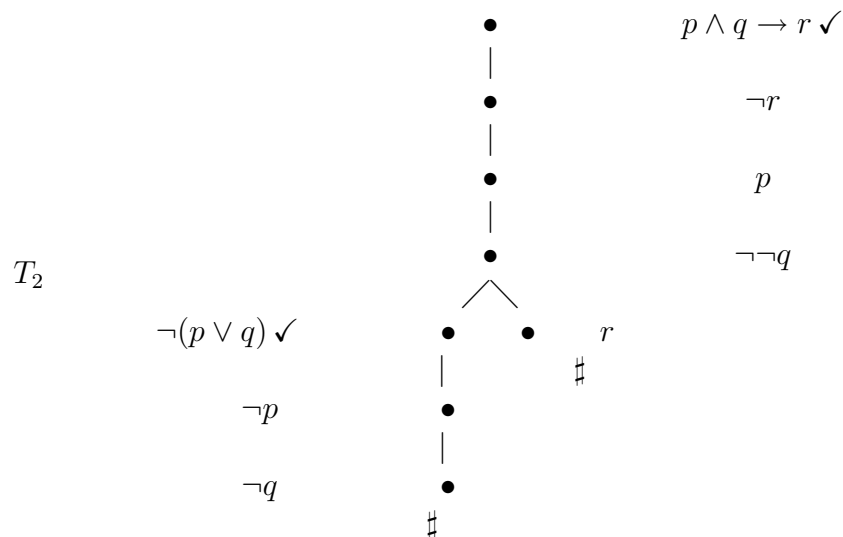
$$T_0 \begin{array}{l} \bullet \quad p \wedge q \rightarrow r \\ | \\ \bullet \quad \neg r \\ | \\ \bullet \quad p \\ | \\ \bullet \quad \neg \neg q \end{array}$$

Ahora intentamos simplificar las fórmulas anteriores usando el esquema de la tabla (4.6) y obtenemos un nuevo tableau  $T_1$ , que es una extensión del anterior. En  $T_1$ , marcamos con el símbolo  $\checkmark$  las fórmulas que hemos reducido y **cerramos** con el símbolo  $\#$  aquellas ramas que contienen un conjunto de fórmulas insatisfacible:



Hemos cerrado la rama derecha de  $T_1$  que contiene las fórmulas  $\neg r$  y  $r$ , ya que esto quiere decir que el conjunto de fórmulas  $\{\neg r, p, \neg\neg q, r\}$  (todas las que aparecen en esta rama, eliminando las que llevan el símbolo  $\checkmark$ ) es insatisfacible.

Ahora, usando su rama izquierda, podemos extender  $T_1$  a un nuevo árbol  $T_2$ :



Las dos ramas de este último árbol están cerradas ya que cada una de ellas contiene una contradicción, la primera contiene el conjunto  $\{p, \neg p\}$  y la segunda el conjunto  $\{\neg r, r\}$ .

Por tanto el conjunto  $\Phi$  es insatisfacible y nuestro razonamiento es válido.

*Podemos observar que el cálculo de la tabla de verdad para este ejemplo requiere considerar 8 posibles valoraciones, mientras el tableau estudiado tiene 2 ramas.*

Vamos entonces a definir los tableaux semánticos. Como veremos, esta definición será, una vez más, una definición recursiva.

**Definición 4.6.10** *Sea*

$$\Phi = \{\varphi_1, \varphi_2, \dots, \varphi_n\}$$

*un conjunto finito de fórmulas proposicionales.*

*Un **tableau**  $T$  asociado al conjunto  $\Phi$  es cualquier árbol de fórmulas que pueda construirse en un número finito de pasos mediante las reglas de formación que se definen a continuación.*

**Reglas de formación de un tableau:**

- **Regla de inicialización ( $R_{ini}$ ):** *el tableau inicial es un árbol de fórmulas  $T_0$  con una única rama cuyos vértices están etiquetados por las fórmulas del conjunto  $\Phi$ .*
- **Regla de reducción ( $R_\alpha$ ):** *si una rama abierta  $\theta$  de  $T$  incluye un vértice etiquetado por una fórmula **conjuntiva**  $\alpha \equiv \alpha_1 \wedge \alpha_2$ , podemos extender  $T$  a un nuevo tableau  $T'$  (una **extensión directa** de  $T$ ) prolongando  $\theta$  de forma lineal, con dos nuevos vértices  $\alpha_1$  y  $\alpha_2$ . Se añade a la fórmula conjuntiva el símbolo  $\checkmark$ .*

*Esta regla no se aplica si la rama abierta  $\theta$  ya contiene  $\alpha_1$  y  $\alpha_2$ .*

- **Regla de reducción ( $R_\beta$ ):** *si una rama abierta  $\theta$  de  $T$  incluye un vértice etiquetado por una fórmula **disyuntiva**  $\beta \equiv \beta_1 \vee \beta_2$ , podemos extender  $T$  a un nuevo tableau  $T'$  (una **extensión directa** de  $T$ ) bifurcando  $\theta$  con dos nuevos vértices  $\beta_1$  y  $\beta_2$ , que sean hijos del vértice  $\beta$ . Se añade a la fórmula disyuntiva el símbolo  $\checkmark$ .*

*Esta regla no se aplica si la rama abierta  $\theta$  ya contiene  $\beta_1$  y  $\beta_2$ . Si  $\theta$  contiene sólo una de las dos componentes, se puede suprimir la bifurcación.*

- **Regla de reducción ( $R_\sigma$ ):** si una rama abierta  $\theta$  de  $T$  incluye un vértice etiquetado por una fórmula **simplificable**  $\sigma \equiv \sigma_1$ , podemos extender  $T$  a un nuevo tableau  $T'$  (una **extensión directa** de  $T$ ) prolongando  $\theta$  de forma lineal, con un nuevo vértice  $\sigma_1$ . Se añade a la fórmula simplificable el símbolo  $\checkmark$ .

*Esta regla no se aplica si la rama abierta  $\theta$  ya contiene  $\sigma_1$ .*

- **Regla de cierre ( $R_c$ ):** si una rama abierta  $\theta$ , contiene  $\perp$  o ambas fórmulas  $\varphi$  y  $\neg\varphi$  aparecen en  $\theta$  se cierra usando el símbolo  $\#$ .

Las siguientes definiciones permiten saber cuando no es posible extender ulteriormente un tableau.

**Definición 4.6.11** • Una rama  $\theta$ , de tableau  $T$  está **completa (ó saturada)** si y sólo si se cumplen las tres siguientes condiciones:

- 1) Si  $\alpha \equiv \alpha_1 \wedge \alpha_2$ , pertenece a  $\theta$ , entonces  $\alpha_1$  y  $\alpha_2$ , pertenecen a  $\theta$ .
- 2) Si  $\beta \equiv \beta_1 \vee \beta_2$ , pertenece a  $\theta$ , entonces  $\beta_1$  ó  $\beta_2$ , pertenece a  $\theta$ .
- 3) Si  $\sigma \equiv \sigma_1$ , pertenece a  $\theta$ , entonces  $\sigma_1$  pertenece a  $\theta$ .

- Un tableau está **cerrado** si todas sus ramas lo están.
- Un tableau está **acabado** si todas sus ramas están cerradas o completas.

**Observación 4.6.12** Dado un conjunto de fórmulas, el tableau asociado a esto conjunto no queda únicamente definido. Para intentar minimizar la complejidad del tableau que se obtiene, en general es conveniente reducir primero fórmulas de tipo  $\sigma$  y  $\alpha$ , o fórmulas que permitan cerrar ramas. Entre las restantes fórmulas, conviene reducir las más sencillas antes que las más complejas.

Las siguientes definiciones y teoremas permiten interpretar un tableau acabado.

Si una rama  $\theta$  de un tableau es completa y abierta, el conjunto de las fórmulas que son sus etiquetas se dice **conjunto de Hintikka**. Es posible demostrar (ver [HLR]) que todo conjunto de Hintikka es satisficible.

**Definición 4.6.13** • Una rama  $\theta$  de un tableau es **satisfacible** si y sólo si existe una valoración que satisface las fórmulas de  $\theta$ .

- Un tableau  $T$  es **satisfacible** si y sólo si existe una valoración que satisface las fórmulas de alguna de sus ramas.

**Lema 4.6.14 (Lema de reducción)** Si un tableau  $T'$  es extensión de un tableau  $T$ , toda valoración que satisface  $T$  satisface también  $T'$ .

**Teorema 4.6.15 (Suficiencia y adecuación)** Un conjunto de fórmulas proposicionales  $\Phi$  es insatisfacible si y sólo si se le puede asociar un tableau cerrado  $T_\Phi$ .

**Observación 4.6.16** Modificando la regla de iniciación y admitiendo la adición de hipótesis, es posible definir el concepto de tableau asociado a un conjunto infinito de fórmulas proposicionales. Su definición proporciona árboles con un número finito de vértices, ya que usan sólo un número finito de las fórmulas iniciales. Las consecuencias teóricas de la definición de tableaux para conjuntos infinitos de fórmulas son profundas y están relacionadas con las propiedades de corrección y completitud de los sistemas de demostración, que trataremos brevemente.

### 4.6.3 Aplicaciones de los tableaux

#### Tableaux y razonamientos

Sea  $\Phi \rightarrow \varphi$  un razonamiento.

El teorema (4.6.15) asegura que el tableau asociado a  $\Phi \wedge \neg\varphi$  es cerrado si y sólo si el razonamiento es válido.

Por el otro lado, si el tableau asociado no es cerrado, contendrá una rama abierta  $\theta$  que nos permite hallar un contraejemplo a la validez del razonamiento (toda valoración que satisface  $\theta$  es un contraejemplo).

**Ejemplo 4.6.17** Consideremos el siguiente razonamiento:

$$\frac{\begin{array}{l} p \rightarrow (q \leftrightarrow \neg r) \\ (q \vee \neg r) \rightarrow \neg p \\ (q \wedge r) \vee p \end{array}}{p \wedge q \wedge \neg r}$$

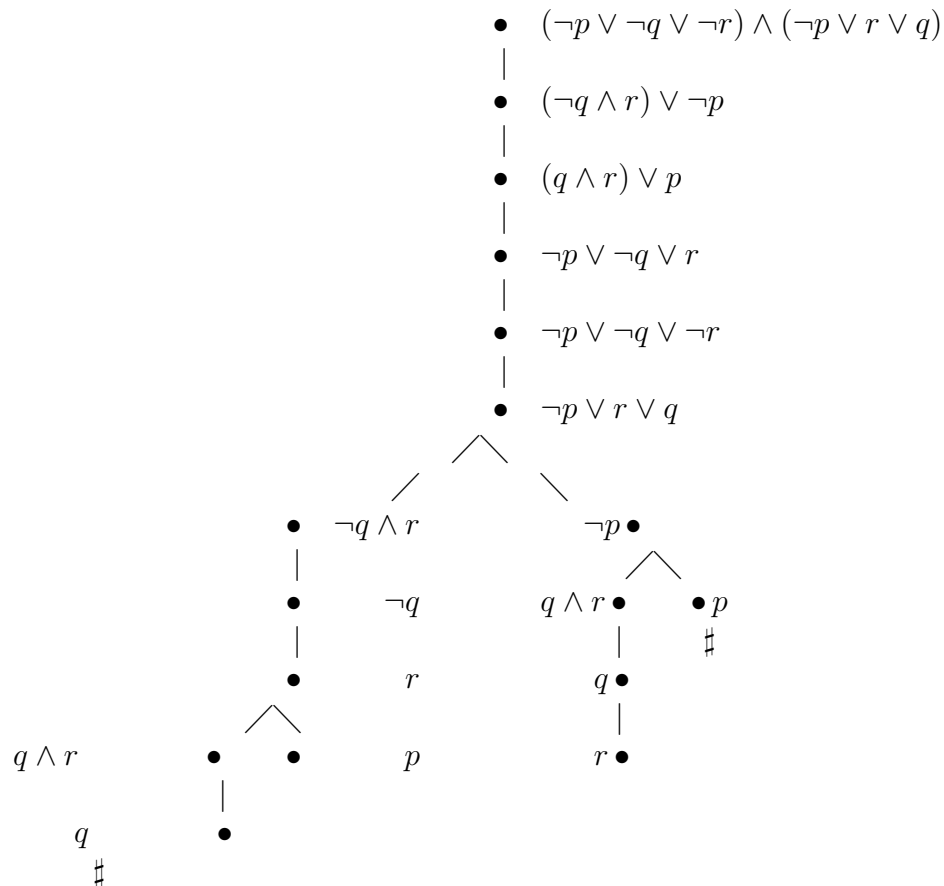
Para verificar su validez usando tableaux, tendremos que refutar la fórmula que se obtiene como conjunción de todas las premisas con la negación de la conclusión.

Podemos simplificar la construcción del tableaux asociado escribiendo todas las fórmulas en forma conjuntivas o disyuntiva.

Usando equivalencias semánticas, obtenemos:

$$\frac{\begin{array}{c} (\neg p \vee \neg q \vee \neg r) \wedge (\neg p \vee r \vee q) \\ (\neg q \wedge r) \vee \neg p \\ (q \wedge r) \vee p \end{array}}{p \wedge q \wedge \neg r}$$

El tableau completo asociado es:



El tableaux completo obtenido tiene dos ramas cerradas y dos abiertas. Estas

últimas nos proporcionan dos contraejemplos del razonamiento:

$$p : 1, r : 1, q : 0 \quad \text{y} \quad p : 0, r : 1, q : 1.$$

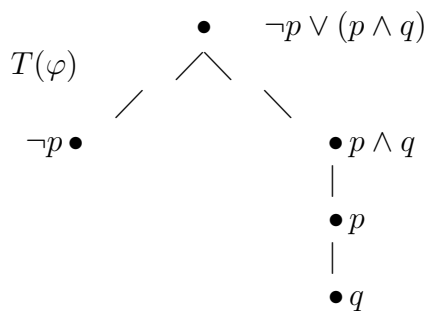
### Tableaux y clasificación de fórmulas

Sea  $T(\varphi)$  el tableaux asociado a una fórmula  $\varphi$ .

- Si  $T(\varphi)$  es cerrado,  $\varphi$  es una contradicción (es insatisfacible).
- Si  $T(\varphi)$  tiene al menos una rama abierta,  $\varphi$  es satisfacible y tenemos que construir el tableaux  $T(\neg\varphi)$  asociado a  $\neg\varphi$ :
  - si  $T(\neg\varphi)$  es cerrado,  $\varphi$  es una tautología,
  - si  $T(\neg\varphi)$  tiene al menos una rama abierta,  $\varphi$  es una contingencia.

**Ejemplo 4.6.18** Sea  $\varphi : p \rightarrow (p \wedge q)$  la fórmula que se quiere clasificar.

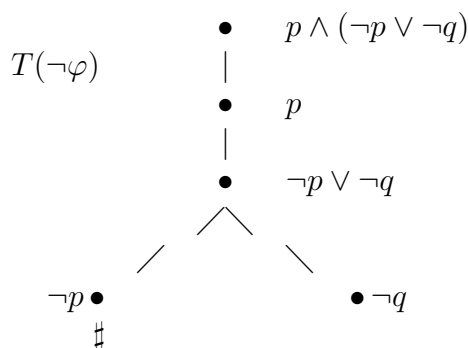
$\varphi$  es equivalente a las forma disyuntiva  $\neg p \vee (p \wedge q)$  y el tableaux  $T(\varphi)$  es



Ya que hay dos ramas abiertas,  $\varphi$  es satisfacible (un modelo es, por ejemplo,  $q : 1, p : 1$ ).

Ahora,  $\neg\varphi : \neg(\neg p \vee (p \wedge q))$  es equivalente a las forma conjuntiva  $p \wedge$

$(\neg p \vee \neg q)$  el tableau  $T(\neg\varphi)$  es



Hay una rama abierta y, por tanto,  $\varphi$  es una contingencia (la valoración  $q : 0, p : 1$  es un contraejemplo de  $\varphi$ ).

### Tableaux y formas normales disyuntivas y conjuntivas

Vamos ahora a ver como la teoría de los tableaux semánticos se puede aplicar para hallar la forma normal disyuntiva y conjuntiva de una fórmula proposicional.

Representar un fórmula  $\varphi$  por medio de su formas normales disyuntiva y conjuntiva permite aplicar nuevos métodos de refutación como el método de resolución, que se estudiará en el contexto de la lógica de primer orden en la asignatura de Lógica Informática.

Siendo sus formas normales una nueva manera de expresar una fórmula por medio de los conectivos del conjunto  $\{\neg, \wedge, \vee\}$ , no es sorprendente que, también en este caso, la teoría de los tableaux sea una herramienta de cálculo efectiva.

#### **Definición 4.6.19 (Formas normales disyuntivas y conjuntivas)**

1. Un **literal** es cualquier fórmula de la forma  $p$  (literal positivo) o  $\neg p$  (literal negativo), donde  $p$  es una proposición atómica.
2. Una **cláusula disyuntiva** es cualquier disyunción de literales.
3. Una **cláusula conjuntiva** es cualquier conjunción de literales.



4. Una fórmula está en **forma normal disyuntiva (FND)** si es una disyunción de cláusulas conjuntivas.
5. Una fórmula está en **forma normal conjuntiva (FNC)** si es una conjunción de cláusulas disyuntivas.

**Convenios:**

- Un literal se puede considerar como una disyunción o una conjunción.
- $\perp$  representa una disyunción, una cláusula disyuntiva o una FND vacía (siempre falsa).
- $\top$  representa una conjunción, una cláusula conjuntiva o una FNC vacía (siempre verdadera).

**Teorema 4.6.20** ([HLR]) *Sea  $\varphi$  una fórmula. A partir de las proposiciones atómicas de  $\varphi$ , se pueden siempre construir una forma normal disyuntiva,  $FND(\varphi)$ , y una forma normal conjuntiva,  $FNC(\varphi)$ , tales que*

$$\varphi \equiv FND(\varphi) \quad \text{y} \quad \varphi \equiv FNC(\varphi).$$

*Además, por las leyes de De Morgan y de la doble negación,*

$$FNC(\varphi) \equiv \neg FND(\neg\varphi).$$

**Cálculo de las formas normales disyuntiva y conjuntiva mediante tableaux**

Dado un tableau acabado  $T(\varphi)$ , sean  $\theta_1, \theta_2, \dots, \theta_n$  sus ramas abiertas.

Para todo  $i \in \{1, 2, \dots, n\}$ , sea  $\Phi_i$  la fórmula obtenida como conjunción de los literales de la rama  $\theta_i$ .

Por el lema de reducción (4.6.14) se obtiene que

$$FND(\varphi) = \Phi_1 \vee \Phi_2 \vee \Phi_3 \vee \dots \vee \Phi_n.$$

**Observación 4.6.21** Las ramas cerradas de  $T(\varphi)$  contribuyen a  $FND(\varphi)$  por una disyunción con  $\perp$  y se pueden ignorar.

Si  $T(\varphi)$  está acabado, para determinar  $FND(\varphi)$  es suficiente considerar sólo los literales que aparezcan en sus ramas abiertas, ya que todas las demás fórmulas ya han sido reducidas.

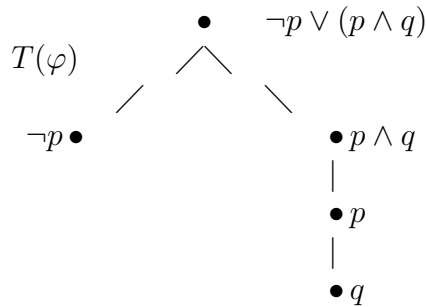
Aplicando las anteriores consideraciones y la equivalencia  $FNC(\varphi) \equiv \neg FND(\neg\varphi)$ , vamos a ver como  $T(\varphi)$  y  $T(\neg\varphi)$  permiten calcular  $FND(\varphi)$  y  $FNC(\varphi)$ .

**Ejemplo 4.6.22** Sea  $\varphi : p \rightarrow (p \wedge q)$  la fórmula clasificada en el ejemplo (4.6.18).

Ya comentamos que, usando equivalencias lógicas, se obtiene que

$$\varphi \equiv \neg p \vee (p \wedge q) = FND(\varphi).$$

Mirando al tableaux acabado  $T(\varphi)$ ,



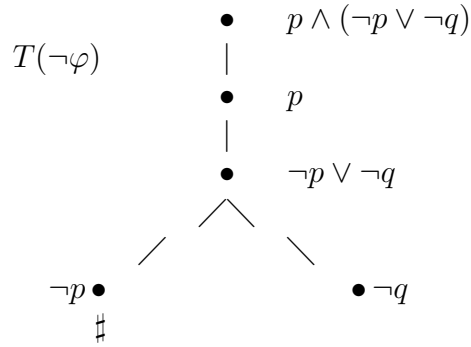
observamos que hay dos ramas abiertas con  $\Phi_1 = \neg p$  y  $\Phi_2 = p \wedge q$ . Por tanto, en este caso,

$$FND(\varphi) = \Phi_1 \vee \Phi_2.$$

También usamos equivalencias lógicas para afirmar que

$$\neg\varphi \equiv p \wedge (\neg p \vee \neg q) = \varphi.$$

Mirando al tableaux acabado  $T(\neg\varphi)$ ,



observamos que hay una sola rama abierta con  $\Phi_2 = p \wedge \neg q$ .

Si no ignoramos la rama cerrada,

$$FND(\neg\varphi) = (p \wedge \neg p) \vee (p \wedge \neg q)$$

y

$$\begin{aligned}
 FNC(\varphi) &= \neg FND(\neg\varphi) = \neg((p \wedge \neg p) \vee (p \wedge \neg q)) \equiv \\
 &\equiv \neg(p \wedge \neg p) \wedge \neg(p \wedge \neg q) \equiv (\neg p \vee p) \wedge (\neg p \vee q).
 \end{aligned}$$

## 4.7 Ejercicios

**Ejercicio 4.7.1** ¿Cuáles de las fórmulas del ejercicio (3.5.5) del capítulo 3 son tautologías? ¿Cuáles son contradicciones?

Para aquellas que sean contingencias, calcula sus modelos y sus contraejemplos.

**Ejercicio 4.7.2** Prueba que las siguientes fórmulas son tautologías:

$$\begin{aligned} &\varphi \vee \neg\varphi, \quad \neg(\varphi \wedge \neg\varphi), \\ &(\varphi \wedge \psi) \leftrightarrow (\psi \wedge \varphi), \quad (\varphi \vee \psi) \leftrightarrow (\psi \vee \varphi), \\ &\varphi \wedge (\psi \wedge \chi) \leftrightarrow (\varphi \wedge \psi) \wedge \chi, \quad \varphi \vee (\psi \vee \chi) \leftrightarrow (\varphi \vee \psi) \vee \chi \\ &\varphi \wedge (\psi \vee \chi) \leftrightarrow (\varphi \wedge \psi) \vee (\varphi \wedge \chi), \quad \varphi \vee (\psi \wedge \chi) \leftrightarrow (\varphi \vee \psi) \wedge (\varphi \vee \chi), \\ &(\varphi \vee \neg\varphi) \wedge \psi \leftrightarrow \psi, \quad (\varphi \vee \neg\varphi) \vee \psi \leftrightarrow (\varphi \vee \neg\varphi), \\ &(\varphi \wedge \neg\varphi) \vee \psi \leftrightarrow \psi, \quad (\varphi \wedge \neg\varphi) \wedge \psi \leftrightarrow (\varphi \wedge \neg\varphi), \\ &\neg(\varphi \vee \psi) \leftrightarrow \neg\varphi \wedge \neg\psi, \quad \neg(\varphi \wedge \psi) \leftrightarrow \neg\varphi \vee \neg\psi. \end{aligned}$$

**Ejercicio 4.7.3** Sea  $\varphi$  una fórmula proposicional. Determina cuáles de las siguientes afirmaciones son verdaderas y justifica tus respuestas.

- Si  $\varphi$  es insatisfacible,  $\neg(\varphi)$  es una tautología.
- Si  $\varphi$  es una tautología,  $\neg(\varphi)$  es insatisfacible.
- Si  $\varphi$  es una contingencia,  $\neg(\varphi)$  es una contingencia.
- Si  $\varphi$  es satisfacible,  $\neg(\varphi)$  es satisfacible.
- Si  $\varphi$  es satisfacible,  $\neg(\varphi)$  no es una tautología.

**Ejercicio 4.7.4** Verifica las implicaciones lógicas de la tabla (4.3).

**Ejercicio 4.7.5** ¿Son válidas las siguientes deducciones?

$$\frac{p \rightarrow \neg r \quad q \rightarrow r}{\neg(p \wedge q)}, \quad \frac{(q \vee \neg s) \rightarrow r \quad \neg q \rightarrow r \quad p \rightarrow \neg s}{r \rightarrow s}, \quad \frac{r \rightarrow s}{p \rightarrow r}$$

$$\frac{p \rightarrow ((q \rightarrow r) \rightarrow s)}{(q \rightarrow r) \rightarrow (p \rightarrow s)}, \quad \frac{(q \rightarrow r) \rightarrow (p \rightarrow s)}{p \rightarrow ((q \rightarrow r) \rightarrow s)}, \quad \frac{p \rightarrow q \quad q \rightarrow r}{s \vee p} \cdot \frac{r \vee t}{}$$

**Ejercicio 4.7.6** *Peláez, Quesada y Rodríguez son tres políticos acusados de corrupción. En el juicio ellos declaran:*

**Peláez:** *“Quesada es culpable y Rodríguez es inocente.”*

**Quesada:** *“Peláez es culpable sólo si Rodríguez es también culpable.”*

**Rodríguez:** *“Yo soy inocente, pero al menos uno de los otros es culpable.”*

*Si todos son inocentes, ¿quién ha mentado?*

*Si todos dicen la verdad, ¿quién es inocente?*

*Si los inocentes dicen la verdad y los culpables mienten, ¿quién es inocente?*

**Ejercicio 4.7.7** *Determina si las deducciones de los ejercicios (3.5.11), (3.5.12) y (3.5.13) del capítulo 3 son válidas.*

**Ejercicio 4.7.8** *(La equivalencia de fórmulas es una relación de equivalencia) Demuestra que en el conjunto  $\mathbf{L}$  de las fórmulas bien construidas, la relación*

$$\varphi \equiv \psi \quad \text{si y sólo si} \quad (\varphi \leftrightarrow \psi \quad \text{es una tautología})$$

*es una relación de equivalencia.*

**Ejercicio 4.7.9** *Verifica las equivalencias lógicas de la tabla (4.4).*

**Ejercicio 4.7.10** *Elegir de las afirmaciones siguientes aquella que sea equivalente a “No es cierto que sea suficiente trabajar para ser feliz.”*

1. *No trabajar es suficiente para no ser feliz.*
2. *Para ser feliz es necesario trabajar.*
3. *No es posible trabajar y no ser feliz al mismo tiempo.*
4. *Se puede trabajar y no ser feliz.*
5. *Se trabaja o no se es feliz.*

**Ejercicio 4.7.11** Dadas las fórmulas

$$p \rightarrow q, \quad (q \vee \neg s) \rightarrow r, \quad p \leftrightarrow q, \quad (p \vee r) \rightarrow (p \wedge \neg r), \quad (r \rightarrow p) \rightarrow (r \rightarrow q),$$

encuentra otras equivalentes en las que se usen sólo los conectivos  $\neg, \wedge, \vee$ . Deduce que toda fórmula bien construida es equivalente a otra donde intervienen sólo los conectivos  $\neg, \wedge, \vee$ .

**Ejercicio 4.7.12** Encuentra una fórmula equivalente a  $p \vee q$  en la que sólo intervengan los conectivos  $\wedge, \neg$ . Deduce que toda fórmula bien construida es equivalente a otra donde intervienen sólo los conectivos  $\neg, \wedge$ .

**Ejercicio 4.7.13** Estudia, mediante tableaux, la validez de los argumentos:

$$\begin{aligned} \{(p \vee \neg q) \rightarrow r, \quad \neg p \rightarrow t, \quad s \rightarrow \neg q, \quad r \rightarrow q\} &\models s \rightarrow t \\ \{t \rightarrow (s \rightarrow p), \quad p \rightarrow \neg p, \quad (r \rightarrow t) \wedge (q \rightarrow s)\} &\models r \rightarrow \neg q \\ \{s \vee t, \quad q \rightarrow p, \quad t \rightarrow \neg p, \quad \neg s\} &\models \neg q \end{aligned}$$

**Ejercicio 4.7.14** Formaliza el siguiente razonamiento y estudia su validez usando tableaux.

*Si el barco entra en el puerto, habrá una gran fiesta. El barco entra en el puerto sólo si necesita repostar combustible. El barco no necesita combustible a menos que venga de muy lejos. Es imposible que no necesite combustible si la comida ya se les ha terminado. Sabemos que, o bien se le ha terminado la comida, o bien necesita combustible. Por tanto: habrá una gran fiesta.*

**Ejercicio 4.7.15** Usando tableaux, clasifica las fórmulas:

1.  $\neg s \vee \neg p \rightarrow (t \rightarrow (p \vee \neg s))$
2.  $(p \rightarrow q) \rightarrow (q \rightarrow p)$
3.  $(t \rightarrow (p \wedge s)) \wedge (\neg s \rightarrow p \wedge \neg t)$
4.  $(\neg(\neg(\neg p \vee r) \vee q)) \rightarrow (\neg(p \wedge q) \vee r)$

**Ejercicio 4.7.16** Con el método de los tableaux, escribe las fórmulas del ejercicio anterior en sus formas normal disyuntiva y conjuntiva.

# Capítulo 5

## Teoría de la demostración

En el capítulo anterior estudiamos el método indirecto de refutación (y los tableaux semánticos) y el método directo de las tablas de verdad como métodos semánticos para verificar la validez de fórmulas y deducciones. Sin embargo, si el número de proposiciones atómicas es elevado las tablas de verdad no son un método eficiente (para un signatura con  $n$  fórmulas atómicas, tenemos que construir una tabla de  $2^n$  filas).

La **teoría de la demostración o teoría de pruebas** nos proporciona métodos alternativos a las tablas de verdad para averiguar

- **la validez de una fórmula proposicional:** si  $\varphi$  es una fórmula válida se dice que es **demostrable** y se escribe  $\vdash \varphi$ .
- **si una fórmula  $\varphi$  es consecuencia lógica de un conjunto de premisas  $\Phi$ :** si  $\varphi$  es consecuencia lógica de  $\Phi$  se dice que  $\varphi$  es **deducible** en el sistema a partir de  $\Phi$  y se escribe  $\Phi \vdash \varphi$ .

El objetivo principal de cualquier teoría de pruebas es demostrar la validez de fórmulas, independientemente del contexto particular que se estén considerando: la demostración de la validez de una fórmula o de una deducción en un sistema de demostración no se desarrolla teniendo en cuenta todas las posibles valoraciones, se obtiene utilizando una secuencia finita de pasos y, en cada uno de ellos, se aplican las reglas de inferencia del sistema.

Ya comentamos que los sistemas de demostración se suelen dividir en dos clases: **sistemas directos** y **sistemas indirectos** (o por **refutación**). Los primeros aplican una cadena finita de reglas de inferencia hasta llegar

a la fórmula que se quiere demostrar. Los segundos aplican la técnica de reducción al absurdo.

Siendo sistemas clásicos, los sistemas de demostración directos tienen interés histórico y además son los más naturales ya que son los más cercanos a la forma de razonamiento habitual.

Los sistemas directos son adaptables a lógicas no clásicas, pero son de difícil automatización.

Los sistemas de demostración indirectos son más modernos y adecuados para su automatización, pero no son aplicables a lógicas distintas de las lógicas clásicas.

En este capítulo estudiaremos un particular sistema de demostración axiomático, el sistema de deducción natural de Gentzen.

El método de los tableaux semánticos que estudiamos en la teoría interpretativa tiene, en realidad, una estructura completamente sintáctica y, por tanto, se puede considerar un ejemplo de sistemas de demostración axiomático indirecto.

Como veremos, es posible demostrar que estos sistemas axiomáticos son equivalentes a la teoría interpretativa. Es decir, se puede verificar que una fórmula proposicional es una tautología en la teoría interpretativa si y sólo si es una fórmula válida en la teoría de la demostración que vamos a estudiar.

## 5.1 Definición de sistema formal axiomático

**Definición 5.1.1** *Un sistema de demostración formal  $S$  o sistema de pruebas se define matemáticamente mediante los siguientes cuatro elementos:*

*$A$  es el **alfabeto** del sistema: el conjunto de símbolos que se pueden utilizar,*

*$F$  es el conjunto de **reglas de sintaxis**: las reglas que permiten definir las fórmulas bien construidas,*

*$X$  es el conjunto de **axiomas**: fórmulas válidas por definición,*

*$R$  es el conjunto de **reglas de inferencias**: reglas de transformación que permiten “inferir” una fórmula, la **conclusión**, a partir de un conjunto de fórmulas, las **condiciones o premisas**.*



Un sistema de demostración  $S$  se puede representar en forma compacta como

$$S = (A, F, X, R).$$

**Definición 5.1.2 (Definición recursiva de teorema)** *Un teorema es una fórmula válida (demostrable) y tiene la siguiente definición recursiva: una fórmula bien construida  $\varphi$  es un **teorema** si es un axioma o si se obtiene como conclusión de la aplicación de un conjunto de reglas de inferencias a otros teoremas.*

*Si  $\varphi$  es un teorema (es válida), se escribe  $\vdash \varphi$ .*

**Ejemplo 5.1.3** *En el contexto de las implicaciones lógicas (semánticas), estudiamos la validez de la ley de contraposición de la tabla (4.3):*

$$(p \rightarrow q) \models (\neg q \rightarrow \neg p).$$

*Dicimos también que la fórmula*

$$(p \rightarrow q) \rightarrow (\neg q \rightarrow \neg p)$$

*es una tautología.*

*Si somos capaces de demostrar que la ley de contraposición es válida en el sistema axiomático que estamos usando, podemos escribir la misma ley de contraposición en forma de teorema, usando la notación:*

$$\vdash (p \rightarrow q) \rightarrow (\neg q \rightarrow \neg p).$$

*Sólo como caso particular, consideremos las dos proposiciones  $p$ ="como demasiado" y  $q$ ="me duele el estómago."*

*La teoría interpretativa, en este caso, nos dice que la proposición "si no me duele el estómago entonces no he comido demasiado,"  $(\neg q \rightarrow \neg p)$ , es consecuencia lógica (vimos que en realidad es equivalente) de la proposición "si como demasiado me duele el estómago,"  $(p \rightarrow q)$ .*

*En una teoría de pruebas la validez de la ley de contraposición se expresaría, en nuestro ejemplo particular, como la validez de la fórmula bien construida "si como demasiado me duele el estómago implica que si no me duele el estómago entonces no he comido demasiado,"  $\vdash (p \rightarrow q) \rightarrow (\neg q \rightarrow \neg p)$ .*

**Definición 5.1.4** *Una deducción o razonamiento consiste de un conjunto de fórmulas  $\Phi = \{\varphi_1, \varphi_2, \dots, \varphi_n\}$ , llamadas las **premisas**, y de una fórmula  $\varphi$ , llamada la **conclusión** de la deducción.*

**Ejemplo 5.1.5** *Escrita como deducción, la ley de contraposición tiene una sola premisa  $\Phi = \{(p \rightarrow q)\}$  y conclusión  $(\neg q \rightarrow \neg p)$ .*

Las nociones de demostración de un teorema y de una deducción están establecidas por las siguientes definiciones:

**Definición 5.1.6 (Demostración de fórmulas (de teoremas))** *Una demostración de un teorema  $\vdash \varphi$  es una sucesión finita de fórmulas*

$$\{\varphi_1, \varphi_2, \dots, \varphi_n, \varphi_{n+1} = \varphi\}$$

*tal que:*

1. *Cada fórmula de la sucesión es*

- *un axioma, un teorema ya demostrado o*
- *un teorema obtenido a partir de las fórmulas anteriores de la sucesión aplicando un conjunto de reglas de inferencias.*

2. *El último elemento de la sucesión es la fórmula  $\varphi$ .*

*Si  $\{\varphi_1, \varphi_2, \dots, \varphi_n, \varphi_{n+1} = \varphi\}$  es una demostración de un teorema, se dice que  $\varphi$  es **válida (demostrable)** y se escribe  $\vdash \varphi$ .*

**Definición 5.1.7 (Demostración de deducciones)** *Sean  $\Phi = \{\varphi_1, \varphi_2, \dots, \varphi_n\}$  un conjunto finito de fórmulas proposicionales y  $\varphi$  una fórmula.*

*Una demostración de una deducción con conjunto de premisas  $\Phi$  y conclusión  $\varphi$  es una sucesión finita de fórmulas*

$$\{\varphi_1, \varphi_2, \dots, \varphi_n, \varphi_{n+1}, \dots, \varphi_{n+m-1}, \varphi_{n+m} = \varphi\}$$

*tal que:*

1. *Cada fórmula de la sucesión es*

- *una premisa (un elemento de  $\Phi = \{\varphi_1, \varphi_2, \dots, \varphi_n\}$ ) o*
- *un axioma, un teorema ya demostrado o*
- *un teorema obtenido a partir de las fórmulas anteriores en la sucesión aplicando un conjunto de reglas de inferencias.*

2. *El último elemento de la sucesión es la fórmula  $\varphi$ .*

Si  $\{\varphi_1, \varphi_2, \dots, \varphi_n, \varphi_{n+1}, \dots, \varphi_{n+m-1}, \varphi_{n+m} = \varphi\}$  es una demostración de una deducción, se dice que la conclusión  $\varphi$  es **consecuencia lógica (deducible)** del conjunto de las premisas  $\Phi$  y se escribe  $\Phi \vdash \varphi$ .

**Observación 5.1.8** Podemos observar que la demostración de un teorema es la demostración de una deducción cuyo conjunto de premisas es vacío.

El siguiente teorema es un resultado fundamental ya que permite definir una relación entre demostraciones de teoremas y demostraciones de deducciones. No presentamos aquí su demostración (ver, por ejemplo, el párrafo 2.3 de [C] o el párrafo 3.1.1 de [A]).

**Teorema 5.1.9 (Teorema de la deducción)** Sean  $\{\varphi_1, \varphi_2, \dots, \varphi_n\}$  un conjunto de fórmulas y  $\varphi$  una fórmula. Entonces

$$\{\varphi_1, \varphi_2, \dots, \varphi_n\} \vdash \varphi \quad \text{si y sólo si} \quad \{\varphi_1, \varphi_2, \dots, \varphi_{n-1}\} \vdash (\varphi_n \rightarrow \varphi).$$

El siguiente corolario del teorema de la deducción establece la relación buscada entre demostraciones de deducciones y de teoremas.

**Corolario 5.1.10** Dada una demostración de una deducción, es siempre posible encontrar una fórmula válida que la represente.

**Demostración [C]:** Si  $\{\varphi_1, \varphi_2, \dots, \varphi_n\} \vdash \varphi$  es la demostración de una deducción, se aplica el teorema de la deducción sucesivamente hasta que el conjunto de premisas sea vacío. Se obtiene así la siguiente cadena de fórmulas que tiene como último elemento un teorema:

$$\begin{aligned} &\{\varphi_1, \varphi_2, \dots, \varphi_n\} \vdash \varphi \\ &\{\varphi_1, \varphi_2, \dots, \varphi_{n-1}\} \vdash \varphi_n \rightarrow \varphi \\ &\{\varphi_1, \varphi_2, \dots, \varphi_{n-2}\} \vdash \varphi_{n-1} \rightarrow (\varphi_n \rightarrow \varphi) \\ &\vdots \\ &\vdash \varphi_1 \rightarrow (\varphi_2 \rightarrow (\dots \rightarrow (\varphi_n \rightarrow \varphi) \dots)). \end{aligned}$$

**Ejemplo 5.1.11** *El teorema de la deducción implica que, en un sistema de pruebas, demostrar la validez de la ley de contraposición*

$$\vdash (p \rightarrow q) \rightarrow (\neg q \rightarrow \neg p)$$

*es equivalente a demostrar la deducción*

$$\{p \rightarrow q\} \vdash \neg q \rightarrow \neg p$$

*o, también, la deducción*

$$\{p \rightarrow q, \neg q\} \vdash \neg p.$$

*En el caso de nuestro ejemplo particular con  $p =$  “como demasiado” y  $q =$  “me duele el estómago”, la última deducción se lee “si como demasiado me duele el estómago y si no me duele el estómago, entonces no he comido demasiado.”*

### 5.1.1 Subdeducciones y notación de Fitting

Antes de definir formalmente el sistema de deducción natural de Gentzen, necesitamos extender la definición de demostración de deducción dada, la (5.1.7), para poder admitir la introducción de subdeducciones. Informalmente se puede decir que, en el proceso de demostración de una deducción (la demostración madre), una subdeducción es la derivación de una conclusión  $\psi$  a partir de una premisa auxiliar (o hipótesis)  $\varphi$ . Después de obtener la conclusión buscada en la subdeducción, se remueve la subdeducción de la demostración madre y se sustituye por el resultado  $\varphi \vdash \psi$ .

Para representar subdeducciones usaremos la notación de Fitting [F]. Esta notación emplea rectángulos, llamados **cajas**. La primera fila de una caja está ocupada por la premisa auxiliar  $\varphi$ , mientras en la última fila se encuentra la conclusión  $\psi$ .

El siguiente es un ejemplo de un esquema de una deducción que contiene a una subdeducción:

- 1)  $\varphi_1$  (Premisa)
- 2)  $\varphi_2$  (Premisa)
- 3)  $\vdash \varphi_3$  (R1(1): regla R1 aplicada a  $\varphi_1$ )

- 4)  $\psi_1$  (Premisa auxiliar)
- 5)  $\vdash \psi_2$  (R2(2,1): regla R2 aplicada a  $\varphi_2$  y  $\psi_1$ )
- 6)  $\vdash \psi_3$  (R3(5): regla R3 aplicada a  $\psi_2$ )

- 7)  $\vdash \varphi_4$  (R4(4,6): regla R4 aplicada a la subdeducción)

La notación usada en el último paso de la demostración madre,  $R4(4,6)$ , quiere decir que la regla R4 se aplica a toda la subdeducción. Se citan sólo la premisa auxiliar y la conclusión de ella.

Pronto veremos algunos ejemplos, pero hay que notar que una subdeducción puede contener a su vez una subdeducción. Por tanto, en la representación de Fitting pueden aparecer varios rectángulos encajados.

## 5.2 El sistema de deducción natural de Gentzen

El sistema de demostración natural de Gerald Gentzen (1934) es el sistema que mejor modela el razonamiento informal o intuitivo.

Este sistema se basa en deducciones y tiene ocho reglas de inferencia y ningún axioma. Las reglas de inferencia son dos para cada conectivo lógico del alfabeto (una de introducción y otra de eliminación).

El uso exclusivo de deducciones presupone razonamientos que siempre dependen de unas premisas iniciales, no necesariamente válidas. Se podría decir que el sistema de Gentzen simula el razonamiento coherente más que el razonamiento válido.

## 5.2.1 Definición del sistema de deducción natural de Gentzen

**Definición 5.2.1** *El sistema de deducción natural de Gentzen es el sistema de demostración axiomático*

$$\mathbf{G} = (A, F, X, R),$$

donde

$A$  : el alfabeto está compuesto por

- los símbolos  $p, q, r, s, t, \dots$  de proposiciones atómicas,
- los símbolos de conectivos  $\neg, \wedge, \vee, \rightarrow$ ,
- los símbolos de paréntesis  $(, )$ .

$F$  : el conjunto de las fórmulas bien construidas (fbc) se define recursivamente como

$At$  : toda proposición atómica es una fbc,

$\neg$  : Si  $\varphi$  es una fbc entonces  $\neg\varphi$  es una fbc,

$\circ$  : si  $\varphi$  y  $\psi$  son dos fbc, entonces

$$\varphi \wedge \psi, \varphi \vee \psi, \varphi \rightarrow \psi, \psi \rightarrow \varphi$$

son fbc.

Toda fbc se obtiene mediante las tres reglas anteriores.

**NOTA:** En lo que se sigue usaremos también el conectivo de coimplicación entre dos fórmulas,  $\varphi \leftrightarrow \psi$ . Este conectivo se entenderá como una forma abreviada de representar la fórmula bien construida  $(\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$ .

$X$  : el conjunto de los axiomas es vacío.

$R$  : el conjunto  $R$  de las reglas de inferencia está compuesto por las reglas de introducción y de eliminación que se describen a continuación.

**Reglas del sistema de Gentzen:***(I ∧): Regla de introducción de la conjunción*

$$\{\varphi, \psi\} \vdash \varphi \wedge \psi$$

*De dos fórmulas se deduce su conjunción.**(E ∧): Regla de eliminación de la conjunción*

$$\begin{array}{l} \varphi \wedge \psi \vdash \varphi \quad (\text{eliminación derecha}) \\ \varphi \wedge \psi \vdash \psi \quad (\text{eliminación izquierda}) \end{array}$$

*De una conjunción de dos fórmulas se deducen las dos fórmulas.***Ejemplo 5.2.2** [UNED1] *Usando sólo las dos reglas anteriores podemos demostrar que*

$$p \wedge (q \wedge r) \vdash p \wedge r.$$

$$\begin{array}{l} 1) \varphi_1 : \quad p \wedge (q \wedge r) \quad (\text{Premisa}) \\ 2) \vdash \varphi_2 : \quad \vdash p \quad (E \wedge(1)) \\ 3) \vdash \varphi_3 : \quad \vdash q \wedge r \quad (E \wedge(1)) \\ 4) \vdash \varphi_4 : \quad \vdash r \quad (E \wedge(3)) \\ 5) \vdash \varphi_5 : \quad \vdash p \wedge r \quad (I \wedge(2,4)) \end{array}$$

*(I ∨): Regla de introducción de la disyunción*

$$\begin{array}{l} \varphi \vdash (\varphi \vee \psi) \quad (\text{introducción derecha}) \\ \psi \vdash (\varphi \vee \psi) \quad (\text{introducción izquierda}) \end{array}$$

*La disyunción de dos fórmulas se puede deducir de cada una de ellas.**(E ∨): Regla de eliminación de la disyunción***NOTA:** *la regla de eliminación de la disyunción NO es*

$$\begin{array}{l} \varphi \vee \psi \vdash \varphi \\ \varphi \vee \psi \vdash \psi, \end{array}$$

ya que de la validez de una disyunción de dos fórmulas no se puede deducir la validez de las dos fórmulas. Sólo sabemos que al menos una de las dos es válida, sin saber cuál es.

La regla de eliminación de la disyunción es

$$(E \vee) : \{\varphi \vee \psi, \varphi \rightarrow \chi, \psi \rightarrow \chi\} \vdash \chi.$$

Esta regla se usa en el **método de demostración por casos**, que se aplica cuando se quiere demostrar una deducción del tipo  $\varphi \vee \psi \vdash \chi$ :

- 1) se introducen las premisas auxiliares  $\varphi$  y  $\psi$ ,
- 2) usando las dos premisas auxiliares se intentan demostrar dos subdeducciones independientes con igual conclusión  $\chi$ , es decir, las subdeducciones

$$\varphi \vdash \chi \quad \text{y} \quad \psi \vdash \chi,$$

- 3) si se ha completado el paso anterior, se usan las subdeducciones obtenidas en la deducción madre y se obtiene que

$$\{\varphi \vee \psi, \varphi \rightarrow \chi, \psi \rightarrow \chi\} \vdash \chi.$$

Con la notación de Fitting  $(E \vee)$  se representa como:

$\varphi \vee \psi$  (Premisa)

$\varphi$ (Premisa auxiliar) $\vdots$ $\vdots$ $\vdash \chi$	$\psi$ (Premisa auxiliar) $\vdots$ $\vdots$ $\vdash \chi$
---	--

$\vdash \chi$



**Ejemplo 5.2.3** [UNED1] Si queremos demostrar la propiedad distributiva de la conjunción

$$p \wedge (q \vee r) \vdash (p \wedge q) \vee (p \wedge r)$$

podemos usar el siguiente esquema:

- 1)  $p \wedge (q \vee r)$  (Premisa)
- 2)  $\vdash p$  ( $E \wedge(1)$ )
- 3)  $\vdash q \vee r$  ( $E \wedge(1)$ )

- |  |   |
|--|---|
| <ol style="list-style-type: none"> <li>4) <math>q</math> (Premisa auxiliar)</li> <li>5) <math>\vdash p \wedge q</math> (<math>I \wedge(2,4)</math>)</li> <li>6) <math>\vdash (p \wedge q) \vee (p \wedge r)</math> (<math>I \vee(5)</math>)</li> </ol> | <ol style="list-style-type: none"> <li><math>r</math> (Premisa auxiliar)</li> <li><math>\vdash p \wedge r</math> (<math>I \wedge(2,4)</math>)</li> <li><math>\vdash (p \wedge q) \vee (p \wedge r)</math> (<math>I \vee(5)</math>)</li> </ol> |
|--|---|



- 7)  $\vdash (p \wedge q) \vee (p \wedge r)$  ( $E \vee(3, (4,5))$ )

$(I \neg)$ : Regla de introducción de la negación

$$(I \neg) : \{\varphi \rightarrow \psi, \varphi \rightarrow \neg\psi\} \vdash \neg\varphi.$$

Esta regla se usa en **demostraciones por reducción al absurdo** de la validez de una fórmula  $\neg\varphi$ :

- 1) Tomamos como premisa auxiliar la fórmula  $\varphi$ ,
- 2) Si de esta premisa auxiliar podemos deducir la validez de una contradicción  $\psi \wedge \neg\psi$ , entonces podemos afirmar la validez de  $\neg\varphi$ .

Con la notación de Fitting  $(I \neg)$  se representa como:

$\varphi$ (Premisa auxiliar) $\vdots$ $\vdots$ $\vdash \psi \wedge \neg\psi$
---

$$\vdash \neg\varphi$$

Usaremos esta regla en el ejemplos (5.2.4)

$(E \neg)$ : Regla de eliminación de la doble negación

$$(E \neg) : \neg\neg\varphi \vdash \varphi.$$

$(I \rightarrow)$ : Regla de introducción de la implicación

Para definir esta regla

1) se toma una premisa auxiliar  $\varphi$ ,

2) si se demuestra que de  $\varphi$  se deduce una fórmula  $\psi$ , entonces se ha demostrado la validez de  $\varphi \rightarrow \psi$ .

Con la notación de Fitting  $(I \rightarrow)$  se representa como:

$\varphi$ (Premisa auxiliar) $\vdots$ $\vdots$ $\vdash \psi$
---

$$\vdash \varphi \rightarrow \psi$$

Esta última regla es una versión del **teorema de la deducción** (5.1.9).

Usaremos esta regla en el ejemplos (5.2.4)

$(E \rightarrow)$ : Regla de eliminación de la implicación (también Modus Ponens):

$$(E \rightarrow) : \{\varphi, \varphi \rightarrow \psi\} \vdash \psi.$$

Usaremos esta regla en el ejemplos (5.2.4)

El sistema de Gentzen está ahora completamente definido.

**Ejemplos 5.2.4** 1) Si queremos demostrar la contraposición

$$\{\varphi \rightarrow \psi, \neg\psi\} \vdash \neg\varphi,$$

podemos escribir el siguiente esquema:

$$1) \varphi \rightarrow \psi \quad (\text{Premisa})$$

$$2) \neg\psi \quad (\text{Premisa})$$

$$3) \neg\neg\varphi \quad (\text{Premisa auxiliar})$$

$$4) \vdash \varphi \quad (E \neg(3))$$

$$5) \vdash \psi \quad (E \rightarrow(3,1))$$

$$6) \vdash \psi \wedge \neg\psi \quad (I \wedge(4,2))$$

$$7) \vdash \neg\varphi \quad (I \neg(3,5))$$

2) La validez de la fórmula

$$\varphi \vdash \psi \rightarrow \varphi$$

se puede demostrar en el sistema de deducción natural usando la introducción de la implicación, según el esquema:

$$1) \varphi \quad (\text{Premisa})$$

$$2) \psi \quad (\text{Premisa auxiliar})$$

$$3) \vdash \varphi \quad (1)$$

$$4) \vdash \psi \rightarrow \varphi \quad (I \rightarrow (2,3))$$

3) La validez de la fórmula

$$\{\varphi \rightarrow \psi, \varphi \rightarrow (\psi \rightarrow \chi)\} \vdash \varphi \rightarrow \chi$$

se puede demostrar en el sistema de deducción natural usando la introducción de la implicación, según el esquema:

- 1)  $\varphi \rightarrow \psi$  (Premisa)
- 2)  $\varphi \rightarrow (\psi \rightarrow \chi)$  (Premisa)

- 3)  $\varphi$  (Premisa auxiliar)
- 4)  $\vdash \psi$  ( $E \rightarrow (3,1)$ )
- 5)  $\vdash \psi \rightarrow \chi$  ( $E \rightarrow (3,2)$ )
- 6)  $\vdash \chi$  ( $E \rightarrow (4,5)$ )

- 7)  $\vdash \varphi \rightarrow \chi$  ( $I \rightarrow (3,6)$ )

## 5.2.2 Reglas derivadas del sistema de Gentzen

A partir de la definición del sistema de Gentzen es posible demostrar los resultados más interesantes de la lógica proposicional. Vamos a enunciar una larga lista de ellos para poder mirar a varios ejemplos de demostración y aprender como aplicar estas nuevas reglas.

Presentaremos sólo algunas de las demostraciones de estas deducciones, las demás se pueden encontrar en la bibliografía de estos apuntes.

- **T1: Teorema de la identidad**

$$\varphi \vdash \varphi$$

De toda fórmula se deduce ella misma.

Esta deducción se puede demostrar por reducción al absurdo:

- 1)  $\varphi$  (Premisa)

- 2)  $\neg\varphi$  (Premisa auxiliar)
- 3)  $\vdash \varphi \wedge \neg\varphi$  ( $I \wedge (1,2)$ )

- 4)  $\vdash \varphi$  ( $I \neg (2,3)$ )

- **T2: Regla del silogismo**

$$\{\varphi \rightarrow \psi, \psi \rightarrow \chi\} \vdash \varphi \rightarrow \chi$$

De dos implicaciones ( $\varphi \rightarrow \psi$  y  $\psi \rightarrow \chi$ ) tales que la conclusión de la primera es la premisa de la segunda se deduce la implicación de la premisa de la primera fórmula a la conclusión de la segunda.

Este teorema se puede pensar como una propiedad transitiva: de “todos los hombres son animales” y “todos los animales son mortales” se deduce que “todos los hombres son mortales.”

- **T3: Modus ponens, MP**

$$\{\varphi, \varphi \rightarrow \psi\} \vdash \psi$$

Es la regla de eliminación de la implicación: de una implicación y de su premisa se deduce su conclusión.

Por ejemplo, de “Juan es un hombre” y “si Juan es un hombre entonces es mortal” se deduce que “Juan es mortal.”

- **T4: Excontradictione Quodlibet**

$$\{\varphi, \neg\varphi\} \vdash \psi$$

De una fórmula y de su negación se deduce cualquier fórmula (por reducción al absurdo).

De “Juan es un hombre” y “Juan no es un hombre” se deduce que “la pared es blanca.”

- **T5: Producto condicional**

$$\{\varphi \rightarrow \psi, \varphi \rightarrow \chi\} \vdash \varphi \rightarrow \psi \wedge \chi$$

De dos implicaciones con la misma premisa se deduce la implicación de esa premisa a la conjunción de sus conclusiones.

Por ejemplo, de “si x es par, es divisible por dos” y “si x es par, no es impar” se deduce que “si x es par, entonces x es divisible por dos y no es impar.”

- **T6: Contraposición**

$$\text{T6.1) } \varphi \rightarrow \psi \vdash \neg\psi \rightarrow \neg\varphi$$

$$\text{T6.2) } \varphi \rightarrow \neg\psi \vdash \psi \rightarrow \neg\varphi$$

$$\text{T6.3) } \neg\varphi \rightarrow \psi \vdash \neg\psi \rightarrow \varphi$$

De una implicación se deduce su contrapositiva (ver ejemplo (5.2.4)).

Por ejemplo,

T6.1) de “llevo paraguas sólo si llueve” se deduce que “si no llueve no llevo paraguas,”

T6.2) de “llevo paraguas sólo si no hace sol” se deduce que “si hace sol no llevo paraguas.”

T6.3) de “no llevo paraguas sólo si hace sol” se deduce que “si no hace sol llevo paraguas.”

- **T7: Interdefinición 1**

$$\text{T7.1) } \varphi \rightarrow \psi \vdash \neg(\varphi \wedge \neg\psi)$$

De una implicación se deduce la negación de la conjunción de su premisa con la negación de su conclusión.

$$\text{T7.2) } \neg(\varphi \wedge \neg\psi) \vdash \varphi \rightarrow \psi$$

Es el recíproco del anterior: una implicación se deduce de la negación de la conjunción de su premisa con la negación de su conclusión.

Por ejemplo,

T7.1) de “llevo paraguas sólo si llueve” se deduce que “no es posible que lleve paraguas y no llueva,”

T7.2) de “no es posible que lleve paraguas y no llueva” se deduce que “llevo paraguas sólo si llueve.”

- **T8: Leyes de de Morgan**

$$\text{T8.1) } \neg(\varphi \vee \psi) \vdash \neg\varphi \wedge \neg\psi$$

De la negación de la disyunción de dos fórmulas se deduce la conjunción de las negaciones de las mismas.

$$\text{T8.2) } \neg\varphi \wedge \neg\psi \vdash \neg(\varphi \vee \psi)$$

Es la recíproca del anterior: de la conjunción de las negaciones de dos fórmulas se deduce la negación de la disyunción de las mismas.

Por ejemplo,

T8.1) de “no es posible que x sea un elemento de A o sea un elemento de B” se deduce que “x no es elemento de A y x no es elemento de B,”

T8.2) de “x no es elemento de A y x no es elemento de B” se deduce que “no es posible que x sea un elemento de A o sea un elemento de B.”

$$\text{T8.3) } \neg(\varphi \wedge \psi) \vdash \neg\varphi \vee \neg\psi$$

De la negación de la conjunción de dos fórmulas se deduce la disyunción de las negaciones de las mismas.

$$\text{T8.4) } \neg\varphi \vee \neg\psi \vdash \neg(\varphi \wedge \psi)$$

Es la recíproca del anterior: de la disyunción de las negaciones de dos fórmulas se deduce la negación de la conjunción de las mismas.

Por ejemplo,

T8.3) de “no es posible que x sea un elemento de A y de B” se deduce que “x no es elemento de A o x no es elemento de B,”

T8.4) de “x no es elemento de A o x no es elemento de B” se deduce que “no es posible que x sea un elemento de A y de B.”

- **T9: Interdefinición 2**

$$\text{T9.1) } \varphi \rightarrow \psi \vdash \neg\varphi \vee \psi$$

De una implicación se deduce la disyunción de la negación de su premisa con su conclusión.

$$\text{T9.2) } \neg\varphi \vee \psi \vdash \varphi \rightarrow \psi$$

Es la recíproca de la anterior: una implicación se deduce de la disyunción de la negación de su premisa con su conclusión.

Por ejemplo,

T9.1) de “una función derivable es continua” se deduce que “una función no es derivable o es continua,”

T9.2) de “una función no es derivable o es continua” se deduce que “una función derivable es continua.”

**Ejemplo 5.2.5** Para verificar que la fórmula

$$\varphi : p \rightarrow (q \rightarrow p),$$

del ejemplo (4.6.4) es válida en el sistema de Gentzen podemos usar la regla ( $I \neg$ ) de reducción al absurdo. Tendremos que demostrar que la fórmula  $\neg\varphi$  conduce a una contradicción.

En la notación de Fitting:

- 1)  $\neg(p \rightarrow (q \rightarrow p))$  (Premisa auxiliar)
- 2)  $\vdash \neg(\neg p \vee (q \rightarrow p))$  (Interdefinición 2(1) y Contraposición (1))
- 3)  $\vdash \neg\neg p \wedge \neg(q \rightarrow p)$  (Ley de de Morgan T8.1(2))
- 4)  $\vdash \neg\neg p$  ( $E \wedge(3)$ )
- 5)  $\vdash \neg(q \rightarrow p)$  ( $E \wedge(3)$ )
- 6)  $\vdash p$  ( $E \neg(4)$ )
- 7)  $\vdash \neg(\neg q \vee p)$  (Interdefinición 2(5) y Contraposición (5))
- 8)  $\vdash \neg\neg q \wedge \neg p$  (Ley de de Morgan T8.3(7))
- 9)  $\vdash \neg p$  ( $E \wedge(8)$ )
- 10)  $\vdash p \wedge \neg p$  ( $I \wedge(6,9)$ )

$$11) \vdash p \rightarrow (q \rightarrow p) \quad (I \neg(1,10))$$

## Teoremas del conectivo conjunción

### • T10: Propiedad conmutativa

$$\text{T10.1) } \varphi \wedge \psi \vdash \psi \wedge \varphi$$

$$\text{T10.2) } \psi \wedge \varphi \vdash \varphi \wedge \psi$$



- **T11: Propiedad asociativa**

$$\text{T11.1) } \varphi \wedge (\psi \wedge \chi) \vdash (\varphi \wedge \psi) \wedge \chi$$

$$\text{T11.2) } (\varphi \wedge \psi) \wedge \chi \vdash \varphi \wedge (\psi \wedge \chi)$$

- **T12: Propiedad distributiva**

$$\text{T12.1) } \varphi \wedge (\psi \vee \chi) \vdash (\varphi \wedge \psi) \vee (\varphi \wedge \chi)$$

$$\text{T12.2) } (\varphi \wedge \psi) \vee (\varphi \wedge \chi) \vdash \varphi \wedge (\psi \vee \chi)$$

- **T13: Propiedad de absorción**

$$\text{T13.1) } \varphi \wedge (\varphi \vee \psi) \vdash \varphi$$

De la conjunción de una fórmula  $\varphi$  con su disyunción con cualquier otra fórmula se deduce  $\varphi$ .

$$\text{T13.2) } \varphi \vdash \varphi \wedge (\varphi \vee \psi)$$

Es la recíproca del anterior: de una fórmula  $\varphi$  se deduce la conjunción de ella con su disyunción con cualquier otra fórmula.

Por ejemplo,

T13.1) de “x es un elemento de A y x es un elemento de A o de B” se deduce que “x es un elemento de A,”

T13.2) de “x es un elemento de A” se deduce que “x es un elemento de A y x es un elemento de A o de B.”

**Demostración de T13.1:**

Tenemos que verificar que

$$\varphi \wedge (\varphi \vee \psi) \vdash \varphi$$

- 1)  $\varphi \wedge (\varphi \vee \psi)$  (Premisa)
- 2)  $\vdash \varphi$  (E  $\wedge$ (1))

- **T14: Idempotencia**

$$\text{T14.1) } \varphi \wedge \varphi \vdash \varphi$$

$$\text{T14.2) } \varphi \vdash \varphi \wedge \varphi$$

La demostración de este teorema se propone como ejercicio.

### Teoremas del conectivo disyunción

- **T15: Propiedad conmutativa**

$$\text{T15.1) } \varphi \vee \psi \vdash \psi \vee \varphi$$

$$\text{T15.2) } \psi \vee \varphi \vdash \varphi \vee \psi$$

- **T16: Propiedad asociativa**

$$\text{T16.1) } \varphi \vee (\psi \vee \chi) \vdash (\varphi \vee \psi) \vee \chi$$

$$\text{T17.2) } (\varphi \vee \psi) \vee \chi \vdash \varphi \vee (\psi \vee \chi)$$

- **T17: Propiedad distributiva**

$$\text{T17.1) } \varphi \vee (\psi \wedge \chi) \vdash (\varphi \vee \psi) \wedge (\varphi \vee \chi)$$

$$\text{T17.2) } (\varphi \vee \psi) \wedge (\varphi \vee \chi) \vdash \varphi \vee (\psi \wedge \chi)$$

- **T18: Propiedad de absorción**

$$\text{T18.1) } \varphi \vee (\varphi \wedge \psi) \vdash \varphi$$

De la disyunción de una fórmula  $\varphi$  con su conjunción con cualquier otra fórmula se deduce  $\varphi$ .

$$\text{T18.2) } \varphi \vdash \varphi \vee (\varphi \wedge \psi)$$

Es la recíproca del anterior: de una fórmula  $\varphi$  se deduce la disyunción de ella con su conjunción con cualquier otra fórmula.

Por ejemplo,

T18.1) de “x es un elemento de A o x es un elemento de A y de B” se deduce que “x es un elemento de A,”

T18.2) de “x es un elemento de A” se deduce que “x es un elemento de A o x es un elemento de A y de B.”

- **T19: Idempotencia**

$$\text{T19.1) } \varphi \vee \varphi \vdash \varphi$$

$$\text{T19.2) } \varphi \vdash \varphi \vee \varphi$$

### Teoremas del conectivo coimplicación

- **T20: Introducción de la coimplicación**

$$\{\varphi \rightarrow \psi, \psi \rightarrow \varphi\} \vdash \varphi \leftrightarrow \psi$$

La coimplicación (doble implicación) entre dos fórmulas se deduce de las dos implicaciones que tienen estas dos fórmulas como premisa y conclusión y como conclusión y premisa, respectivamente.

La demostración de este teorema se propone como ejercicio.

- **T21: Eliminación de la coimplicación**

$$\text{T21.1) } \varphi \leftrightarrow \psi \vdash \varphi \rightarrow \psi$$

$$\text{T21.2) } \varphi \leftrightarrow \psi \vdash \psi \rightarrow \varphi$$

De una coimplicación entre dos fórmulas se deducen las implicaciones de cada una a la otra.

La demostración de este teorema se propone como ejercicio.

- **T22: Propiedad reflexiva**

$$\vdash \varphi \leftrightarrow \varphi$$

Toda fórmula coimplica a sí misma.

- **T23: Propiedad transitiva**

$$\{\varphi \leftrightarrow \psi, \psi \leftrightarrow \chi\} \vdash \varphi \leftrightarrow \chi$$

Si una fórmula  $\varphi$  coimplica a una fórmula  $\psi$  y si  $\psi$  coimplica a una fórmula  $\chi$ , entonces  $\varphi$  coimplica a  $\chi$ .

- **T24: Propiedad simétrica**

$$\varphi \leftrightarrow \psi \vdash \psi \leftrightarrow \varphi$$

De  $\varphi$  coimplica a  $\psi$  se deduce que  $\psi$  coimplica a  $\varphi$ .

**Observación 5.2.6** *Sea  $R$  la relación binaria sobre  $L$  definida por la coimplicación:*

$$(\varphi, \psi) \in R \quad \text{si y sólo si} \quad \varphi \leftrightarrow \psi.$$

*La validez de la propiedad reflexiva, simétrica y transitiva para esta relación implica que  $R$  es una relación de equivalencia entre fórmulas proposicionales.*

## Teoremas del conectivo implicación

- **T25: Importación-Exportación**

$$\text{T25.1) } \{\varphi \rightarrow (\psi \rightarrow \chi), \varphi \wedge \psi\} \vdash \chi$$

De una implicación cuya conclusión es la implicación  $\psi \rightarrow \chi$  y de la conjunción de las dos premisas  $\varphi \wedge \psi$  se deduce la conclusión  $\chi$ .

$$\text{T25.2) } \{\varphi \wedge \psi \rightarrow \chi, \varphi\} \vdash \psi \rightarrow \chi$$

Es la recíproca del anterior: de una implicación cuya premisa es la conjunción de dos fórmulas se deduce la implicación de una de las dos premisas a la implicación de la otra premisa a la conclusión.

**Ejemplo 5.2.7** *Sean*

$p =$   $n$  es un número natural,

$q =$   $n$  es par,

$r =$  el cuadrado de  $n$  es par.

*Entonces,*

*T25.1) de “si  $n$  es número natural, entonces si  $n$  es par su cuadrado es par” y de “si  $n$  es un número natural y es par”, se deduce que “el cuadrado de  $n$  es par,”*

*T25.2) de “si  $n$  es un número natural y es par, entonces su cuadrado es par” y de “ $n$  es número natural, se deduce que “si  $n$  es par, su cuadrado es par.”*

**Demostración de T25.1:**

Tenemos que verificar que

$$\{\varphi \rightarrow (\psi \rightarrow \chi), \varphi \wedge \psi\} \vdash \chi.$$

- 1)  $\varphi \rightarrow (\psi \rightarrow \chi)$  (Premisa)
- 2)  $\varphi \wedge \psi$  (Premisa)
- 3)  $\vdash \varphi$  (E  $\wedge$ (2))
- 4)  $\vdash \psi \rightarrow \chi$  (E  $\rightarrow$ (3,1))
- 5)  $\vdash \psi$  (E  $\wedge$ (2))
- 6)  $\vdash \chi$  (E  $\rightarrow$ (5,4))

- **T26: Mutación de premisa**

$$\varphi \rightarrow (\psi \rightarrow \chi) \vdash \psi \rightarrow (\varphi \rightarrow \chi)$$

**Demostración de T26:**

- 1)  $\varphi \rightarrow (\psi \rightarrow \chi)$  (Premisa)

<ol style="list-style-type: none"> <li>2) <math>\psi</math> (Premisa auxiliar)</li> </ol>
---

<ol style="list-style-type: none"> <li>3) <math>\varphi</math> (Premisa auxiliar)</li> <li>4) <math>\vdash \psi \rightarrow \chi</math> (E <math>\rightarrow</math>(3,1))</li> <li>5) <math>\vdash \chi</math> (E <math>\rightarrow</math>(2,4))</li> </ol>
---

<ol style="list-style-type: none"> <li>6) <math>\vdash \varphi \rightarrow \chi</math> (I <math>\rightarrow</math> (3,5))</li> </ol>
--

- 7)  $\vdash \psi \rightarrow (\varphi \rightarrow \chi)$  (I  $\rightarrow$  (2,6))

- **T27: Carga de premisa**

$$\varphi \vdash (\psi \rightarrow \varphi)$$

**Demostración de T27:**1)  $\varphi$  (Premisa)2)  $\psi$  (Premisa auxiliar)3)  $\vdash \varphi$  (Teorema de la identidad, T1(1))4)  $\vdash \psi \rightarrow \varphi$  (I  $\rightarrow$  (2,3))• **T28: Modus tollens, MT**

$$\{\varphi \rightarrow \psi, \neg\psi\} \vdash \neg\varphi$$

Notar que esta regla es la contraposición.

**Demostración de T28:** la demostración es la misma vista en el apartado 1) del ejemplo (5.2.4).

• **T29: Tollendo ponens**

$$\{\varphi \vee \psi, \neg\psi\} \vdash \varphi$$

**Demostración de T29:**

- 1)  $\varphi \vee \psi$  (Premisa)
- 2)  $\neg\psi$  (Premisa)

- 3)  $\varphi$  (Premisa auxiliar)
- 4)  $\vdash \varphi$  (Teorema de la identidad, T1(3))

- 5)  $\psi$  (Premisa auxiliar)
- 6)  $\vdash \psi \wedge \neg\psi$  (I  $\wedge$  (2,5))
- 7)  $\vdash \varphi$  (Excontraditione Quodlibet, T4(6))

- 8)  $\vdash \varphi$  (E  $\vee$  (1,(3,4),(5,7)))

• **T30: Dilemas**

T30.1) : Dilema constructivo simple:  
 $\{\varphi \vee \psi, \varphi \rightarrow \chi, \psi \rightarrow \chi\} \vdash \chi$ ,

T30.2) : Dilema constructivo complejo:  
 $\{\varphi \vee \psi, \varphi \rightarrow \chi, \psi \rightarrow \sigma\} \vdash \chi \vee \sigma$ ,

T30.3) : Dilema destructivo simple:  
 $\{\neg\varphi \vee \neg\psi, \chi \rightarrow \varphi, \chi \rightarrow \psi\} \vdash \neg\chi$ ,

T30.4) : Dilema destructivo complejo:  
 $\{\neg\varphi \vee \neg\psi, \chi \rightarrow \varphi, \sigma \rightarrow \psi\} \vdash \neg\chi \vee \neg\sigma$ ,

**Demostración de T30:**

T30.1) :  $\{\varphi \vee \psi, \varphi \rightarrow \chi, \psi \rightarrow \chi\} \vdash \chi$

Es la regla de eliminación de la disyunción (E  $\vee$ ).

T30.2) :  $\{\varphi \vee \psi, \varphi \rightarrow \chi, \psi \rightarrow \sigma\} \vdash \chi \vee \sigma$

- 1)  $\varphi \vee \psi$  (Premisa)
- 2)  $\varphi \rightarrow \chi$  (Premisa)
- 3)  $\psi \rightarrow \sigma$  (Premisa)

- 4)  $\varphi$  (Premisa auxiliar)
- 5)  $\vdash \chi$  (E  $\rightarrow$ (4,2))
- 6)  $\vdash \chi \vee \sigma$  (I  $\vee$ (5))

- 7)  $\psi$  (Premisa auxiliar)
- 8)  $\vdash \sigma$  (E  $\rightarrow$ (7,3))
- 9)  $\vdash \chi \vee \sigma$  (I  $\vee$ (8))

- 10)  $\vdash \chi \vee \sigma$  (E  $\vee$  (1,(4,6),(7,9)))

T30.3) :  $\{\neg\varphi \vee \neg\psi, \chi \rightarrow \varphi, \chi \rightarrow \psi\} \vdash \neg\chi$

- 1)  $\neg\varphi \vee \neg\psi$  (Premisa)
- 2)  $\chi \rightarrow \varphi$  (Premisa)
- 3)  $\chi \rightarrow \psi$  (Premisa)

- 4)  $\neg\varphi$  (Premisa auxiliar)
- 5)  $\vdash \neg\chi$  (Modus Tollens, MT(2,4))

- 6)  $\neg\psi$  (Premisa auxiliar)
- 7)  $\vdash \neg\chi$  (Modus Tollens, MT(3,6))

- 8)  $\vdash \neg\chi$  (E  $\vee$  (1,(4,5),(6,7)))

T30.4) :  $\{\neg\varphi \vee \neg\psi, \chi \rightarrow \varphi, \sigma \rightarrow \psi\} \vdash \neg\chi \vee \neg\sigma$



- 1)  $\neg\varphi \vee \neg\psi$  (Premisa)
- 2)  $\chi \rightarrow \varphi$  (Premisa)
- 3)  $\sigma \rightarrow \psi$  (Premisa)

- 4)  $\neg\varphi$  (Premisa auxiliar)
- 5)  $\vdash \neg\chi$  (Modus Tollens, MT(2,4))
- 6)  $\vdash \neg\chi \vee \neg\sigma$  (I  $\vee$ (5))

- 7)  $\neg\psi$  (Premisa auxiliar)
- 8)  $\vdash \neg\sigma$  (Modus Tollens, MT(3,7))
- 9)  $\vdash \neg\chi \vee \neg\sigma$  (I  $\vee$ (8))

- 10)  $\vdash \neg\chi \vee \neg\sigma$  (E  $\vee$  (1,(4,6),(7,9)))

En el siguiente ejemplo vamos a emplear la regla ( $I\neg$ ) para demostrar la validez de una fórmula por refutación.

**Ejemplo 5.2.8** *Para verificar que la fórmula*

$$\varphi : p \rightarrow (q \rightarrow p),$$

*del ejemplo (4.6.4) es válida en el sistema de Gentzen podemos usar la regla ( $I\neg$ ): tendremos que demostrar que la fórmula  $\neg\varphi$  conduce a una contradicción.*

En la notación de Fitting:

- 1)  $\neg(p \rightarrow (q \rightarrow p))$  (Premisa auxiliar)
- 2)  $\vdash \neg(\neg p \vee (q \rightarrow p))$  (Interdefinición 2(1))
- 3)  $\vdash \neg\neg p \wedge \neg(q \rightarrow p)$  (Ley de de Morgan T8.1(2))
- 4)  $\vdash \neg\neg p$  (E  $\wedge$ (3))
- 5)  $\vdash \neg(q \rightarrow p)$  (E  $\wedge$ (3))
- 6)  $\vdash p$  (E  $\neg$ (4))
- 7)  $\vdash \neg(\neg q \wedge p)$  (Interdefinición 2(5))
- 8)  $\vdash q \vee \neg p$  (Ley de de Morgan T8.3(7))
- 9)  $\vdash \neg p$  (E  $\wedge$ (8))
- 10)  $\vdash p \wedge \neg p$  (I  $\wedge$ (6,9))

- 11)  $\vdash p \rightarrow (q \rightarrow p)$  (I  $\neg$ (1,10))

En la sección (5.4) veremos que existen teoremas que demuestran la equivalencia entre la teoría interpretativa (semántica) y el sistema de deducción natural.

Esta equivalencia entre teorías nos permite elegir, en cada caso, entre el método más adecuado o eficiente para desarrollar una demostración.

### 5.3 Tableaux sintácticos

Podemos observar que la definición recursiva de tableaux semántico asociado a un conjunto de fórmulas no requiere, en ningún paso, el cálculo de los valores de verdad de las fórmulas consideradas. Eso quiere decir que el método de refutación de los tableaux es, en realidad, un método de demostración sintáctico.

Los **tableaux sintácticos** usan el método de refutación para derivar la validez de un razonamiento o de una fórmula, sin tener que tomar en consideración los valores de verdad de las fórmulas en estudio. Permiten demostrar teoremas y deducciones.

**Ejemplo 5.3.1** *En el ejemplo (4.6.9) demostramos la implicación lógica*

$$\{p \wedge q \rightarrow r, \neg r, p\} \models \neg q.$$

*Teniendo en cuenta las equivalencias entre fórmulas definidas por la coimplicación de la teoría axiomática, la misma secuencia de tableaux empleada en ese ejemplo demuestra la deducción*

$$\{p \wedge q \rightarrow r, \neg r, p\} \vdash \neg q,$$

*o, por el teorema de la deducción, el teorema*

$$\vdash (p \wedge q \rightarrow r) \rightarrow (\neg r \rightarrow (p \rightarrow \neg q)).$$

## 5.4 Completitud, corrección y decidibilidad

Para cualquier sistema axiomático es siempre imprescindible estudiar si éste cumple las tres propiedades de completitud, corrección y decidibilidad. Las dos primeras propiedades permiten estudiar la relación del sistema con la semántica del lenguaje y la tercera es una propiedad algorítmica.

Las demostraciones de la validez de estas propiedades para los sistemas de la lógica proposicional son complejas y fuera del alcance de esta asignatura. Nos limitaremos a enunciar los principales teoremas relacionados.

- **Completitud**

Se dice que un sistema de demostración axiomático es **completo** si es capaz de demostrar cualquier fórmula semánticamente válida (una tautología) y deducir cualquier consecuencia lógica.

**Teorema de Kalmar [A]:** toda tautología en teoría interpretativa es una fórmula válida en teoría de la demostración.

El sistema de Gentzen es completo.

Considerando la definición sintáctica de la teoría de los tableaux para un conjunto arbitrario de fórmulas (posiblemente infinito), se puede demostrar que el sistema axiomático así obtenido es completo [HLR].

- **Corrección**

Se dice que un sistema de demostración es **correcto (consistente, coherente)** si todas las fórmulas demostrables en el sistema son tautologías y todas las fórmulas deducibles en el sistema a partir de un conjunto de premisas son consecuencia lógica de dichas premisas.

**Teorema de Post [A]:** toda fórmula válida en teoría de la demostración es una tautología en teoría interpretativa.

El sistema de Gentzen y la definición sintáctica de la teoría de los tableaux son sistemas correctos [HLR] y [MH].

- **Decidibilidad**

Se dice que un sistema de demostración es **decidible** si proporciona un procedimiento general y finito (aplicable a cualquier fórmula y que termine) que permita decidir si una fórmula es válida o deducible a partir de un conjunto de fórmulas.

Los teoremas anteriores permiten afirmar que una fórmula proposicional es un teorema si y sólo si es una tautología.

Por tanto la evaluación de la tabla de verdad es un procedimiento general y finito de decisión de la validez de una fórmula [A].

Es también posible demostrar que el método de los tableaux, en su definición más general, tiene las mismas propiedades de generalidad y finitud [HLR] que las tablas de verdad.

Se sigue que los sistemas formales de la lógica proposicionales son decidibles.

En conclusión, los sistemas formales de la lógica proposicional (la lógica de orden 0) son completos, correctos y decidibles.

Las lógicas de orden superior son más generales, pero pierden algunas de estas propiedades. Veremos que la lógica de predicados (de primer orden) es completa y correcta, pero no es decidible.

## 5.5 Ejercicios

**Ejercicio 5.5.1** Usando el sistema de deducción natural, demuestra la idempotencia de la conjunción (T14), la introducción del coimplicador (T20) y la eliminación del coimplicador (T21).

**Ejercicio 5.5.2** Prueba que las siguientes equivalencias semánticas son también coimplicaciones demostrables en el sistema de deducción natural:

1.  $\neg\neg p \equiv p$
2.  $\neg(p \vee q) \equiv \neg p \wedge \neg q$
3.  $\neg(p \wedge q) \equiv \neg p \vee \neg q$
4.  $(p \rightarrow q) \equiv \neg p \vee q$
5.  $\neg(p \rightarrow q) \equiv p \wedge \neg q$

**Ejercicio 5.5.3** Formaliza los siguientes razonamientos:

1. Si llueve no iré al mercado. Si no iré al mercado, o bien no tendré comida o bien iré al restaurante. Llueve y tengo comida. Por lo tanto: iré al restaurante.
2. Si  $f$  es diferenciable en  $[a, b]$ , es continua y acotada en  $[a, b]$ . Si  $f$  no fuese acotada en  $[a, b]$  no podría ser diferenciable en  $[a, b]$ . Por tanto: si  $f$  es discontinua y acotada en  $[a, b]$ , no es diferenciable en  $[a, b]$ .

**Ejercicio 5.5.4** Usando el sistema de deducción natural, prueba (si es posible) la validez de los razonamientos anteriores.

**Ejercicio 5.5.5** Usando el sistema de deducción natural, demuestra la validez de las siguientes deducciones:

1.  $\{(p \wedge q) \rightarrow r, \neg(p \vee r) \rightarrow s, p \rightarrow q\} \vdash \neg s \rightarrow r$
2.  $\{r \rightarrow p, \neg q \rightarrow \neg r, s \rightarrow q, (p \wedge q) \rightarrow t, \neg s \vee p\} \vdash (r \vee s) \rightarrow t$
3.  $\{p \rightarrow (q \rightarrow r), \neg s \vee p, q\} \vdash s \rightarrow r$
4.  $\{s \rightarrow (t \rightarrow u), u \rightarrow \neg u, (v \rightarrow s) \wedge (p \rightarrow t)\} \vdash v \rightarrow \neg p$



## Parte II

# Lógica de predicados de primer orden





# Capítulo 6

## Sintaxis de la lógica de primer orden

La **lógica de predicados o de primer orden (LPO,  $L_1$ )** es una generalización de la lógica de proposiciones (LP,  $L_0$ ). Introduciendo nuevos elementos del lenguaje, permite estudiar la estructura interna de los enunciados (sus propiedades, las relaciones entre objetos, etc.).

**Ejemplo 6.0.1** *Consideremos el siguiente razonamiento:*

$$\frac{\begin{array}{l} \text{Todos los hombres son mortales,} \\ \text{Sócrates es un hombre,} \end{array}}{\text{Sócrates es mortal.}}$$

Sean  $p =$  todos los hombres son mortales,  $q =$  Sócrates es un hombre y  $r =$  Sócrates es mortal.

*La formalización del razonamiento dado en lógica proposicional es*

$$\frac{\begin{array}{l} p \\ q \end{array}}{r}$$

*y tiene el contraejemplo  $p : 1, q : 1$  y  $r : 0$ . Por tanto, el razonamiento en estudio no es válido en lógica proposicional.*

Es evidente que la conclusión de este ejemplo no satisface nuestra intuición. Necesitamos extender la lógica proposicional a una lógica que admita la validez de una clase más amplia de razonamientos.

Esta nueva lógica tendría que permitir una descripción más fina de la realidad, pudiendo distinguir los objetos o términos (por ejemplo, los hombres) de sus propiedades o predicados (por ejemplo, la propiedad de ser mortales). La lógica proposicional, cuyos elementos básicos son las proposiciones atómicas, no permite realizar esta distinción.

La lógica de predicados (Gottlob Frege, 1879) nos permite dar una descripción de la realidad más detallada. Por ejemplo, veremos que el razonamiento anterior se puede formalizar más precisamente como

$$\frac{\forall x(H(x) \rightarrow M(x)), \quad H(s)}{M(s)}.$$

Como ya comentamos en varias ocasiones, la sintaxis es la definición axiomática de los elementos básicos del lenguaje y de las reglas que permiten obtener nuevas expresiones correctas a partir de aquellos, las expresiones bien construidas. El objetivo de este capítulo es el estudio de la sintaxis de la lógica de predicados.

En el capítulo 2 estudiamos una breve introducción a las nociones básicas de la teoría de conjuntos, de las relaciones y de las funciones. Estos conceptos son fundamentales para la lógica de predicados y se aconseja su repaso.

Las principales referencias usadas en este capítulo son [A], [HLR] y [MH].

## 6.1 Alfabeto del lenguaje formal de la lógica de predicados

Los elementos básicos del **alfabeto del la lógica de predicados** son:

- **Los símbolos de constantes:** se denotan  $a, b, c, \dots$  y representan objetos concretos. Las constantes son **individuos** o **elementos distinguidos** del **universo del discurso**, que es la colección de objetos sobre los cuales queremos razonar.
- **Las variables:** se denotan  $x, y, z, \dots$  y sirven para representar objetos, cuyo dominio hay que especificar.

Tomaremos conjuntos de variables  $V$  finitos o infinitos *numerables*. Recordamos que un conjunto  $V$  es infinito numerable si existe una función biyectiva entre  $V$  y el conjunto de los números naturales  $\mathbb{N}$ .

**Ejemplo 6.1.1** *En las expresiones “Pedro es un estudiante” y “ $x$  es un número primo,” Pedro es una constante en el conjunto de la personas y  $x$  es una variable en el conjunto de los números enteros.*

• **Los conectivos lógicos:**

1. **constantes (de aridad 0):**  $\top$  (verdadero) y  $\perp$  (falso)
2. **conectivos unarios:**  $\neg$  (negación)

**Ejemplo 6.1.2** *Siendo “ $E(a)$  : Pedro es un estudiante,” aplicando la negación se obtiene “ $\neg(E(a))$  : Pedro no es un estudiante.” De forma similar, si “ $P(x)$  :  $x$  es un número primo,” “ $\neg(P(x))$  :  $x$  no es un número primo.*

3. **conectivos binarios:**

- $\wedge$  (y: la conjunción),
- $\vee$  (ó: la disyunción),
- $\rightarrow$  (la implicación o condicional),
- $\leftrightarrow$  (la doble implicación o coimplicación).

**NOTACIÓN:** El símbolo  $\circ$  se usará para representar un conectivo binario cualquiera.

**Ejemplos 6.1.3** 1) *La frase “Sócrates es un filósofo, sin embargo no es un deportista” se escribe como  $(F(s) \wedge \neg(D(s)))$ , donde “ $F(x)$  :  $x$  es un filósofo,” “ $D(x)$  :  $x$  es un deportista” y  $x$  es una variable en el conjunto de las personas.*

2) *La frase “Sócrates es un filósofo o Sócrates es un deportista” se escribe como  $(F(s) \vee D(s))$ .*

3) *La frase “La luna es de papel si y sólo si Carlos lee muchos libros” se escribe como  $(P(l) \leftrightarrow L(c))$ , donde “ $P(x)$  :  $x$  es de papel,” “ $L(y)$  :  $y$  lee muchos libros,”  $x$  es una variable en el conjunto de las cosas y  $y$  es una variable en el conjunto de las personas.*

- **Igualdad:**

Usaremos el símbolo de igualdad “=” para expresiones del tipo  $4 - 1 = 3$  o  $mcm(2, 3) = 6$ .

Por tanto, estamos considerando **la lógica de predicados con igualdad**.

**Ejemplo 6.1.4** *El comando “IF  $x > 0$  THEN  $y = \sqrt{x}$ ” se escribe como  $(P(x) \rightarrow (y = f(x)))$ , donde “ $P(x) : x$  es positivo,” “ $f(x) : \sqrt{x}$ ” y  $x$  y  $y$  son variables en el conjunto de los números reales.*

- **Los cuantificadores:**

1.  $\forall$  : **cuantificador universal** (para todo). El cuantificador universal permite referirse a **todos** los individuos del universo del discurso.
2.  $\exists$  : **cuantificador existencial** (existe). El cuantificador existencial permite referirse a **algunos** de los individuos del universo del discurso.

**Ejemplos 6.1.5** 1) *La frase “Todo número primo y mayor que 2 es impar” se escribe como  $(\forall x((P(x) \wedge Q(x)) \rightarrow R(x)))$ , donde “ $P(x) : x$  es primo,” “ $Q(x) : x$  es mayor que 2,” “ $R(x) : x$  es impar” y  $x$  es un elemento cualquiera del conjunto de los números enteros.*

2) *La frase “Todo hombre es mortal y hay hombres que no son filósofos” se escribe como  $((\forall x(P(x) \rightarrow Q(x))) \wedge (\exists x(P(x) \wedge (\neg(R(x)))))$ , donde “ $P(x) : x$  es un hombre,” “ $Q(x) : x$  es mortal,” “ $R(x) : x$  es filósofo” y  $x$  es un elemento cualquiera del conjunto de los seres.*

- **Los símbolos de puntuación (o símbolos auxiliares):** paréntesis abiertos y cerrados, comas.
- **Los símbolos de predicado:** se denotan  $P, Q, R, \dots$

Todo predicado tiene un número  $n \in \mathbb{N} \cup \{0\}$  de argumentos. El número  $n$  es la **aridad** del predicado.

En ocasiones se especificará la aridad  $n$  de un predicado  $P$  por medio del símbolo  $P^n$ .

1. **Predicados constantes**,  $n = 0$ : representan **proposiciones atómicas**.  
Para representar las proposiciones atómicas se suelen usar los símbolos  $p, q, r, s, t, \dots$
2. **Predicados monádicos**,  $n = 1$ : representan **propiedades** de objetos.
3. **Predicados poliádicos**,  $n > 1$ : representan **relaciones** entre objetos.

Los predicados poliádicos de la lógica de primer orden son relaciones sobre conjuntos según la definición del capítulo 2. Así, por ejemplo, todo predicado binario es una relación binaria  $R$  entre dos conjuntos  $A$  y  $B$ , es decir,  $R \subseteq A \times B$ .

Un predicado monádico asocia a cada objeto de un dominio una propiedad.

**Nota:** Por definición, un predicado es tal que al sustituir todas sus símbolos de variables por símbolos de constantes se obtiene una fórmula proposicional.

**Ejemplos 6.1.6** 1) *Los siguientes son ejemplos de predicados:*

a) *“ $p$  : hoy llueve” es un predicado constante, una proposición atómica.*

b) *“ $P(x)$  : la raíz cuadrada de  $x$  es irracional” (predicado monádico), siendo  $x$  un cualquier número real.*

*Al sustituir  $x$  por un símbolo de constante, por ejemplo 2, se obtiene la proposición “ $P(2)$  : la raíz cuadrada de 2 es irracional.”*

c)  *$P(x, y)$  :  $x$  es un hermano de  $y$  (predicado binario), siendo el dominio el conjunto de las personas.*

*En particular, “ $P(l, j)$  : Luis es un hermano de Jose” es una fórmula proposicional.*

d)  *$P(x, y, z)$  :  $x$  prefiere  $y$  a  $z$  (predicado ternario), siendo el dominio de las variable  $x$  el conjunto de las personas y el dominio de las variables  $y$  y  $z$  el conjunto de los países.*

*En particular, “ $P(l, f, i)$  : Luis prefiere Francia a Italia” es una fórmula proposicional.*

2) Como vimos en el capítulo 2, los predicados permiten definir conjuntos.

Por ejemplo, en el dominio de los puntos del plano real  $\mathbb{R} \times \mathbb{R}$ , el conjunto  $S$  de las soluciones del sistema de dos ecuaciones

$$\begin{cases} 3x + 2y = 1 \\ x - 4y = 0 \end{cases}$$

se puede definir por medio de los predicados " $S_1(z)$  : el punto  $z = (x, y)$  es tal que  $3x + 2y = 1$ " y " $S_2(z)$  : el punto  $z = (x, y)$  es tal que  $x - 4y = 0$ ."

En el siguiente capítulo, dedicado a la semántica de la lógica de primer orden, veremos como asignar valores de verdad a las expresiones bien construidas.

Para nuestro ejemplo, podremos afirmar que los elementos del conjunto  $S$  son los puntos del plano  $z$  tales que  $S_1(z) \wedge S_2(z)$  es verdadera.

- **Los símbolos de función:** se denotan  $f, g, h, \dots$

Toda función tiene un número  $n \in \mathbb{N} \cup \{0\}$  de argumentos. El número  $n$  es la **aridad** de la función.

En ocasiones se especificará la aridad  $n$  de una función  $f$  por medio del símbolo  $f^n$ .

**Nota:** Las funciones de la lógica de primer orden son funciones según la definición del capítulo 2.

1. **Funciones constantes**,  $n = 0$ : son símbolos de constantes.  
Para representar las funciones constantes se suelen usar los símbolos  $a, b, c, \dots$
2. **Funciones monádicas**,  $n = 1$ : representan un objeto en función de otro.
3. **Funciones poliádicas**,  $n > 1$ : representan un objeto en función de  $n$  otros objetos.

**Ejemplos 6.1.7** 1) *Los siguientes son ejemplos de funciones:*

a)  $f(x) : \sqrt{x}$  es la función monádica raíz cuadrada de un número real no negativo,  $f \subseteq [0, \infty) \times \mathbb{R}$ .

b)  $f(x, y) : x + y$  es la función binaria suma de dos números reales,  $f \subseteq \mathbb{R} \times \mathbb{R} \times \mathbb{R}$ .

c)  $f(x, g(y, z), a) : x(y+z) - a$  es la función ternaria,  $f \subseteq \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R}$ , que multiplica las primeras dos variables reales y resta al producto obtenido la constante real  $a$ .

$g$  es la función binaria,  $g \subseteq \mathbb{R} \times \mathbb{R} \times \mathbb{R}$ , suma de dos números reales.

2) El siguiente es un ejemplo de un predicado que no es una función:

En el conjunto de las personas, “ $P(x, y) = x$  es la madre de  $y$ ” es un predicado (una relación binaria) que no es una función, ya que su dominio no es todo el conjunto de las personas y una madre puede tener más que un hijo.

**Observación 6.1.8** De los ejemplos anteriores se deduce que, si  $n$  es un número natural, **toda función  $n$ -aria se puede escribir como un predicado  $(n + 1)$ -nario**. Sin embargo, en general, un predicado no se puede escribir como una función.

Veremos que las funciones sirven para simplificar la estructura de la fórmulas de la lógica de primer orden.

## 6.2 Definición recursiva de las expresiones bien construidas: términos y fórmulas

Como en la lógica proposicional, los elementos básicos de un lenguaje y las reglas de formación permiten definir cadenas finitas de símbolos arbitrarias (**palabras**).

En la lógica proposicional la palabras que son expresiones bien construidas son las fórmulas. En la lógica de predicados las expresiones bien construidas pueden ser de dos tipos: **términos** (que representan objetos) y **fórmulas** (que expresan hechos relativos a objetos).

Los términos y las fórmulas son las **expresiones bien construidas** de la lógica de primer orden (ver figura 6.1).

Los términos y las fórmulas de la lógica de predicados se obtienen a partir del alfabeto dado sólo por medio de las reglas de formación que vamos a definir a continuación.

Como en la lógica proposicional, la definición de las expresiones bien construidas de la lógica de primer orden es recursiva y, como veremos, siguen valiendo los principios de unicidad de estructura, de inducción y de recursión estructural para términos y fórmulas.

- **Términos**

**Definición 6.2.1 (Definición recursiva de los términos)**

1. (TAt): *Todo símbolo de variable y todo símbolo de constante es un término.*  
*Las variables y las constantes forman el conjunto de los **términos atómicos**.*
2. (TF): *Si  $f$  es un símbolo de función de aridad  $n > 0$  y  $t_1, t_2, \dots, t_n$  son términos, entonces  $f(t_1, t_2, \dots, t_n)$  es un término.*  
*Estos términos se llaman **términos compuestos**.*
3. *Si una palabra no se obtiene mediante las dos reglas anteriores, entonces no es un término.*

**Ejemplo 6.2.2** *Los siguientes son ejemplos de términos:*

$$x, a, f(x), g(x, y), g(x, f(x)),$$

*donde  $x$  es una variable,  $a$  es una constante,  $f$  es una función monádica y  $g$  es una función binaria.*

*Los primero dos términos de la lista son atómicos y los restantes son compuestos.*

- **Fórmulas**

**Definición 6.2.3** *Una **fórmula atómica** es cualquier expresión de la forma  $P(t_1, t_2, \dots, t_n)$ , donde  $P$  es un símbolo de predicado con aridad  $n > 0$  y  $t_1, t_2, \dots, t_n$  son términos. Las proposiciones atómicas (los predicados constantes), los conectivos lógicos constantes  $\top$  y  $\perp$  y la igualdad entre términos,  $s = t$ , también se consideran fórmulas atómicas.*



**Ejemplo 6.2.4** *Los siguientes son ejemplos de fórmulas atómicas:*

$$\top, p, R(f(x), y)$$

donde  $p$  es una proposición atómica,  $f$  es una función monádica y  $R$  es un predicado binario.

**Definición 6.2.5 (Definición recursiva de las fórmulas)**

1. (FAt): Toda fórmula atómica es una fórmula.
2. (F¬): Si  $\varphi$  es una fórmula, entonces  $\neg\varphi$  es una fórmula.
3. (F◦): Si  $\varphi$  y  $\psi$  son dos fórmulas entonces  $(\varphi\circ\psi)$  es una fórmula.
4. (F∀∃): Si  $\varphi$  es una fórmula y  $x$  es un símbolo de variable, entonces  $\exists x\varphi$  y  $\forall x\varphi$  son fórmulas.
5. Si una palabra no se obtiene mediante las cuatro reglas anteriores, entonces no es una fórmula.

**NOTACIÓN:** Para representar las fórmulas se suelen usar las letras del alfabeto griego  $\varphi, \psi, \chi, \dots$

**Ejemplos 6.2.6** 1) *Las fórmulas atómicas del ejemplo anterior son fórmulas.*

2) *La expresión*

$$(\forall x\exists y(R(x, f(y)) \wedge (\neg(f(x) = s)))$$

*es una fórmula ya que:*

1.  $R(x, f(y))$  y  $(f(x) = s)$  son fórmulas atómicas (aplicando (FAt)),
2.  $(\neg(f(x) = s))$  es una fórmula (aplicando (F¬)),
3.  $(R(x, f(y)) \wedge (\neg(f(x) = s)))$  es una fórmula (aplicando (F◦)),
4.  $\exists y(R(x, f(y)) \wedge (\neg(f(x) = s)))$  es una fórmula (aplicando (F∀∃)),
5.  $(\forall x\exists y(R(x, f(y)) \wedge (\neg(f(x) = s))))$  es una fórmula (aplicando (F∀∃)).

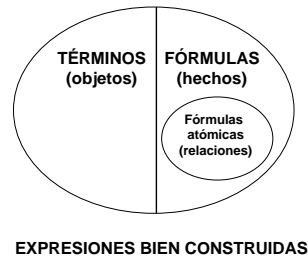


Figura 6.1: Expresiones bien construidas

### 6.2.1 Variables libres y variables ligadas

Las fórmulas se clasifican en **fórmulas abiertas** y **fórmulas cerradas**, dependiendo de ciertas características de sus variables.

#### Definición 6.2.7 Fórmulas abiertas y fórmulas cerradas

- Se dice que la ocurrencia de una variable  $x$  en una fórmula  $\varphi$  es **ligada** (o que la variable es ligada) si está afectada por algún cuantificador. En caso de que una variable no aparezca ligada diremos que su ocurrencia es **libre** (la variable es libre).
- Se dice que una fórmula  $\varphi$  es **abierta** si tiene alguna ocurrencia de variable libre. Se dice que una fórmula  $\varphi$  es **cerrada** (o que es una sentencia) si no tiene ninguna ocurrencia de variable libre.

**Ejemplos 6.2.8** 1) En la fórmula  $\forall xP(x, y)$  la variable  $x$  aparece ligada y la variable  $y$  aparece libre. La fórmula es abierta.

2) En la fórmula

$$\exists x((P(x) \wedge Q(x)) \vee (P(x) \wedge Q(x)))$$

la variable  $x$  aparece ligada en las dos componentes de la disyunción, ya que el cuantificador existencial la afecta en los dos casos. La fórmula es cerrada.

3) En la fórmula

$$(\exists x(P(x) \wedge Q(x)) \vee (P(x) \wedge Q(x)))$$

la variable  $x$  aparece ligada en la primera componente de la disyunción y aparece libre en la segunda. En este caso el cuantificador existencial afecta  $x$  sólo en la primera componente de la disyunción y la fórmula es abierta.

*Notar que esta última fórmula se puede reescribir como*

$$(\exists x(P(x) \wedge Q(x)) \vee (P(y) \wedge Q(y))).$$

## 6.2.2 El principio de inducción estructural para expresiones bien construidas

Como en el caso de la lógica de proposiciones, para poder estudiar las propiedades de las expresiones bien construidas de la lógica de primer orden una de las técnicas más adecuadas es el principio de inducción estructural.

### Principio de inducción estructural para términos

Sea  $\mathbb{P}$  una cierta propiedad tal que:

1. **Base de inducción (TAt):** Todos los términos atómicos cumplen la propiedad  $\mathbb{P}$ .

#### Pasos de inducción:

2. (TF): Si  $f$  es un símbolo de función de aridad  $n > 0$  y los términos  $t_1, t_2, \dots, t_n$  cumplen la propiedad  $\mathbb{P}$  (hipótesis de inducción), entonces  $f(t_1, t_2, \dots, t_n)$  cumple la propiedad  $\mathbb{P}$ .

El **principio de inducción estructural para términos** afirma que si se verifican las condiciones anteriores, entonces se puede concluir que **todo término** cumple la propiedad  $\mathbb{P}$ .

### El principio de inducción estructural para fórmulas

Sea  $\mathbb{P}$  una cierta propiedad tal que:

1. **Base de inducción (FAt):** Todas las fórmulas atómicas cumplen la propiedad  $\mathbb{P}$ .

#### Pasos de inducción:

2. (F $\neg$ ): Si  $\varphi$  es una fórmula que cumple la propiedad  $\mathbb{P}$  (hipótesis de inducción), entonces  $\neg(\varphi)$  cumple la propiedad  $\mathbb{P}$ ,

3. (F $\circ$ ): Si  $\varphi$  y  $\psi$  son dos fórmulas que cumplen la propiedad  $\mathbb{P}$  (hipótesis de inducción), entonces  $(\varphi \circ \psi)$  cumple la propiedad  $\mathbb{P}$ ,
4. (F $\forall\exists$ ): Sean  $\varphi$  es una fórmula y  $x$  un símbolo de variable. Si  $\varphi$  cumple la propiedad  $\mathbb{P}$  (hipótesis de inducción), entonces  $\forall x\varphi$  y  $\exists x\varphi$  cumplen la propiedad  $\mathbb{P}$ .

El **principio de inducción estructural para fórmulas** afirma que si se verifican las condiciones anteriores, entonces se puede concluir que **toda fórmula** cumple la propiedad  $\mathbb{P}$ .

## 6.3 Representación de las expresiones bien construidas

Como ya hicimos en el caso de la lógica proposicional, vamos a ilustrar las maneras más usadas para representar las fórmulas de la lógica de primer orden.

### 6.3.1 Fórmulas en forma abreviada

Para reducir una fórmula a su **forma abreviada** podemos seguir los siguientes pasos:

- a) Se puede **omitir el par de paréntesis externo**.

Así, por ejemplo, la fórmula

$$((\exists xP(x) \wedge Q(x)) \wedge (\forall yP(y)))$$

se puede escribir como:

$$(\exists xP(x) \wedge Q(x)) \wedge (\forall yP(y)).$$

- b) Se introducen las mismas **reglas de precedencia** y el mismo **convenio de asociatividad** entre conectivos que definimos para fórmulas proposicionales.

Así, por ejemplo, la fórmula anterior se escribiría (sin cambios) como:

$$(\exists xP(x) \wedge Q(x)) \wedge (\forall yP(y)).$$

c) Se admite la **precedencia de los cuantificadores**: los cuantificadores tienen prioridad sobre los conectivos. Se obtienen los siguientes niveles de precedencia:

**Nivel 1:**  $\forall$  y  $\exists$ ,  
**Nivel 2:**  $\neg$ ,  
**Nivel 3:**  $\vee$  y  $\wedge$ ,  
**Nivel 4:**  $\rightarrow$  y  $\leftrightarrow$ .

Finalmente, la forma abreviada de nuestra fórmula es:

$$(\exists xP(x) \wedge Q(x)) \wedge \forall yP(y).$$

### 6.3.2 Principio de unicidad de estructura para expresiones bien construidas

Como para la lógica proposicional, en la lógica de primer orden valen principios de unicidad de estructura para términos y fórmulas. Cada expresión bien construida admite un análisis sintáctico único, es decir, existe una única manera de derivarla usando las reglas de formación dadas.

Estos principios nos permitirán representar expresiones bien construidas por medio de árboles sintácticos (o estructurales).

#### Principio de unicidad de estructura para términos

Todo término pertenece a una y sólo una de las siguientes categorías:

1. (TAt): es atómico,
2. (TF): se escribe como  $f(t_1, t_2, \dots, t_n)$ , donde  $f$  es un símbolo de función de aridad  $n > 0$  y  $t_1, t_2, \dots, t_n$  son términos.  
 $f$  y  $t_1, t_2, \dots, t_n$  están unívocamente determinados.

### Principio de unicidad de estructura para fórmulas

Toda fórmula  $\varphi$  pertenece a una y sólo una de las siguientes categorías:

1. (FAt)<sub>1</sub> :  $\varphi$  es una proposición atómica o uno de los conectivos lógicos constantes  $\top$  y  $\perp$ ,
2. (FAt)<sub>2</sub> :  $\varphi$  es una igualdad entre términos,  $s = t$ , donde  $s$  y  $t$  están unívocamente determinados,
3. (FAt)<sub>3</sub> :  $\varphi$  es  $P(t_1, t_2, \dots, t_n)$ , donde  $P$  es un símbolo de predicado con aridad  $n > 0$  y  $t_1, t_2, \dots, t_n$  son términos.  
 $P$  y  $t_1, t_2, \dots, t_n$  están unívocamente determinados,
4. (F $\neg$ ):  $\varphi$  es  $\neg(\varphi_1)$  para cierta fórmula  $\varphi_1$ , unívocamente determinada,
5. (F $\circ$ ):  $\varphi$  es  $(\varphi_1 \circ \varphi_2)$  para cierto conectivo  $\circ$  y ciertas fórmulas  $\varphi_1$  y  $\varphi_2$ .  
 $\varphi_1$  y  $\varphi_2$  están unívocamente determinadas,
6. (F $\forall\exists$ ):  $\varphi$  es  $\forall x\varphi_1$  o  $\exists x\varphi_1$  para cierta fórmula  $\varphi_1$  y cierto símbolo de variable  $x$ .  
 $\varphi_1$  y  $x$  están unívocamente determinados.

### 6.3.3 Términos y fórmulas en forma de árbol estructural

Una primera consecuencia de los principios de unicidad de estructura es la posibilidad de representar términos y fórmulas por medio de árboles.

En los siguientes dos ejemplos veremos como construir el árbol sintáctico de un término y de una fórmula.

**Ejemplos 6.3.1** 1) *Consideremos el término compuesto*

$$f(g(x, h(a, y)), i(x, l(y))).$$

*Para representar su estructura sintáctica podemos dibujar el árbol con raíz de la figura 6.2.*

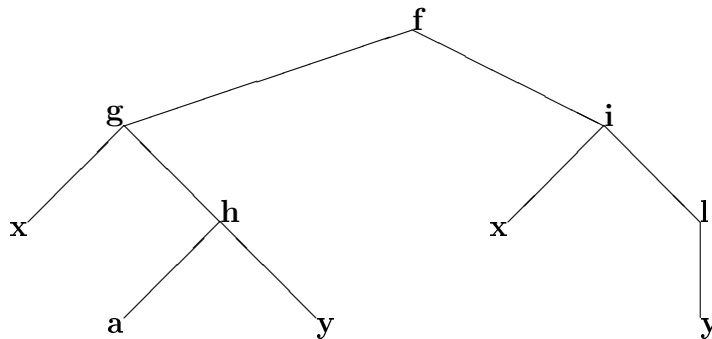


Figura 6.2: Árbol sintáctico del término  $f(g(x, h(a, y)), i(x, l(y)))$ .

2) Sea ahora  $\varphi$  la fórmula

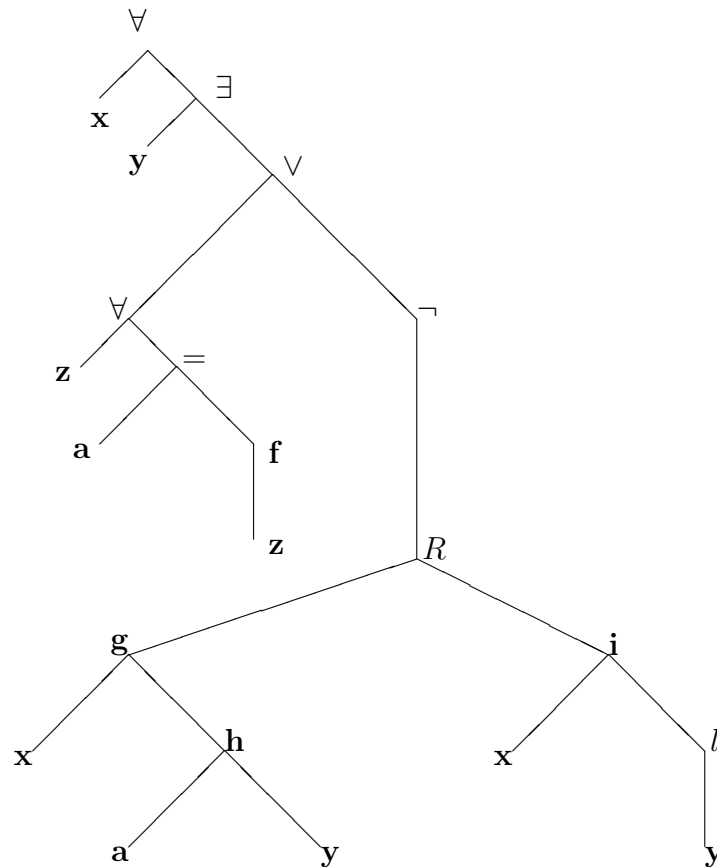
$$\varphi : \forall x \exists y (\forall z (a = f(z)) \vee (\neg R(g(x, h(a, y)), i(x, l(y))))).$$

Para representar la estructura sintáctica de  $\varphi$ , podemos dibujar el árbol con raíz de la figura 6.3.

Notar que las etiquetas de las hojas de los árboles sintácticos son los términos atómicos de las expresiones bien construidas representadas. La existencia de predicados y funciones que dependen de  $n$  variables, hace que en la lógica de predicados los árboles estructurales no tienen que ser binarios como es en el caso de la lógica proposicional.

### 6.3.4 El principio de recursión estructural para expresiones bien construidas

Como ya vimos para la lógica proposicional, también en el caso de la lógica de predicados una segunda consecuencia de los principios de unicidad de estructura son los principios de recursión estructural para expresiones bien construidas. Estos principios permiten definir funciones sobre el conjunto de los términos y el conjunto de las fórmulas.

Figura 6.3: Árbol sintáctico de  $\varphi$ .

### El principio de recursión estructural para términos

Sea  $A$  un conjunto no vacío. Para definir una función

$$f : \mathbf{T} \longrightarrow \mathbf{A}$$

cuyo dominio sea el conjunto de los términos y cuyo codominio sea un conjunto dado  $\mathbf{A}$ , se puede emplear la siguiente definición recursiva:

1. **Base (TAt):** Si  $t$  es un término atómico,  $f(t)$  se define explícitamente, es un elemento de  $\mathbf{A}$ ,
2. **Paso recursivo (TF):** si  $t$  es el término compuesto  $g(t_1, t_2, \dots, t_n)$ , se define  $f(t)$  en función de  $f(t_1), f(t_2), \dots, f(t_n)$ .



Estas definiciones determinan la función  $f$  sobre todo  $\mathbf{T}$ .

**Ejemplos 6.3.2 (HLR)** 1) Sea  $\mathbf{Var}$  el conjunto de todas las variables. Queremos definir recursivamente el conjunto  $Var(t)$  de las variables libres de un cualquier término  $t$ .

Por el principio de recursión estructural para términos, podemos definir una función  $Var : \mathbf{T} \longrightarrow P(\mathbf{Var})$ , ( $P(\mathbf{Var})$  es el conjunto de las partes de  $\mathbf{Var}$ ) que asocia a todo término  $t$  el conjunto de sus variables libres,  $Var(t)$  :

1. **Base (TAt):**  $Var(x) = \{x\}$ , y  $Var(a) = \emptyset$ , donde  $\emptyset$  es el conjunto vacío.
2. **Paso recursivo (TF):** si  $t$  es el término compuesto  $g(t_1, t_2, \dots, t_n)$ , se define

$$Var(t) = Var(t_1) \cup Var(t_2) \cup \dots \cup Var(t_n).$$

2) Se denomina **profundidad** de un término  $t$  a la longitud de la rama más larga del árbol estructural de  $t$ . Por tanto, la profundidad de  $t$  es la altura de su árbol sintáctico.

Por ejemplo, la longitud del término  $f(g(x, h(a, y)), i(x, l(y)))$  de la figura 6.2 es 3.

Por el principio de recursión estructural, la definición recursiva de la función profundidad

$$pf : \mathbf{T} \longrightarrow \mathbb{N} \cup \{0\}$$

(que asocia a cada término  $t \in \mathbf{T}$  su profundidad) es:

1. **Base (TAt):** Si  $t$  es un término atómico,  $pf(t) = 0$ ,
2. **Paso recursivo (TF):** si  $t$  es el término compuesto  $g(t_1, t_2, \dots, t_n)$ , se define

$$pf(t) = \max\{pf(t_1), pf(t_2), \dots, pf(t_n)\} + 1.$$

### El principio de recursión estructural para fórmulas

Sea  $A$  un conjunto no vacío. Para definir una función

$$f : \mathbf{F} \longrightarrow \mathbf{A}$$

cuyo dominio sea el conjunto de las fórmulas y cuyo codominio sea un conjunto dado  $\mathbf{A}$ , se puede emplear la siguiente definición recursiva:

1. **Base (FAt):** Si  $\varphi$  es una fórmula atómica,  $f(\varphi)$  se define explícitamente, es un elemento de  $\mathbf{A}$ .

**Paso recursivo:**

2. (F $\neg$ ):  $f(\neg(\varphi_1))$  se define en función de  $f(\varphi_1)$ ,
3. (F $\circ$ ):  $f(\varphi_1 \circ \varphi_2)$  se define en función de  $f(\varphi_1)$  y  $f(\varphi_2)$ ,
4. (F $\forall\exists$ ):  $f(\forall x\varphi_1)$  y  $f(\exists x\varphi_1)$  se definen en función de  $f(\varphi_1)$ .

Estas definiciones determinan la función  $f$  sobre todo  $\mathbf{F}$ .

#### Ejemplo 6.3.3 [HLR]

Sea  $\mathbf{Var}$  el conjunto de todas las variables. Queremos definir recursivamente el conjunto  $Lib(\varphi)$  de las variables libres de una cualquier fórmula  $\varphi$ .

Por el principio de recursión estructural para fórmulas, podemos definir una función  $Lib : \mathbf{F} \longrightarrow P(\mathbf{Var})$ , que asocia a toda fórmula  $\varphi$  el conjunto de sus variables libres,  $Lib(\varphi)$ :

1. **Base (FAt):** Si  $\varphi$  es una fórmula atómica,  $Lib(\varphi)$  es el conjunto  $Var(\varphi)$  de todas las variables de  $\varphi$ . En particular,
 
$$Lib(\top) = Lib(\perp) = Lib(p) = \emptyset, \text{ donde } \emptyset \text{ es el conjunto vacío,}$$

$$Lib(s = t) = Var(s) \cup Var(t),$$

$$Lib(P(t_1, t_2, \dots, t_n)) = Var(t_1) \cup Var(t_2) \cup \dots \cup Var(t_n).$$

**Paso recursivo:**

2. (F $\neg$ ):  $Lib(\neg(\varphi_1)) = Lib(\varphi_1)$ ,

3.  $(F\circ)$ :  $Lib(\varphi_1 \circ \varphi_2) = Lib(\varphi_1) \cup Lib(\varphi_2)$ ,
4.  $(F\forall\exists)$ :  $Lib(\forall x\varphi_1) = Lib(\exists x\varphi_1) = Lib(\varphi_1) \setminus \{x\}$ .

*Estas definiciones determinan la función Lib sobre todo  $\mathbf{F}$ .*

## 6.4 Formalización del lenguaje natural

Como ya vimos en lógica proposicional, formalizar una frase del lenguaje natural consiste en encontrar una expresión que la represente fielmente en el lenguaje formal.

No hay procedimientos generales para la formalización, pero se pueden determinar algunas estrategias, como las que vamos a indicar a continuación.

La referencia principal para esta sección es [A].

- Si la frase que se quiere formalizar no tiene una estructura sintáctica fácilmente reconocible, se puede intentar **reescribirla** en el lenguaje natural hasta llegar a una frase con una estructura más sencilla y que mantenga el mismo significado.
- Tenemos que **definir claramente el dominio o los dominios** a los cuáles pertenecen los objetos que vamos a usar.
- En una frase necesitamos determinar:
  - Las **constantes**, que son objetos concretos de uno o más dominios.
  - Las **variables**, que son objetos genéricos de uno o más dominios.
  - Las **funciones de aridad**  $n > 0$ , que representan como un cierto objeto queda determinado por otros (u otro).
  - Los **predicados monádicos** que representan propiedades de un objeto.
  - Los **predicados de aridad**  $n > 0$  que representan relaciones entre objetos.
- Identificadas las conectivas lingüísticas y los cuantificadores (universales o existenciales), necesitamos sustituirlas por los **conectivos** y los **cuantificadores** de la lógica de primer orden.

- Para formalizar un **razonamiento**, necesitamos formalizar el conjunto de sus premisas y de su conclusión.

**Observación 6.4.1** *A la hora de formalizar frases o razonamientos hay algunas observaciones importantes que es oportuno tener en cuenta.*

1. *Ya que la formalización de una frase depende del dominio o de los dominios elegidos, se pueden obtener formalizaciones distintas de un mismo enunciado.*

**Ejemplo 6.4.2** *Para formalizar la frase:*

*“Todos los niños juegan con la pelota,”*

*podemos definir los predicados*

$J(x) : x \text{ juega con la pelota}$

*y*

$J(x, y) : x \text{ juega con } y.$

a) *Sea  $D_1$  el conjunto de los niños. Entonces se obtiene*

$$\forall x J(x).$$

b) *Sean  $D_2$  el conjunto de las personas y sea “ $N(x) : x$  es un niño.” En este caso se obtiene*

$$\forall x (N(x) \rightarrow J(x)).$$

c) *Sean  $D_1$  el conjunto de los niños y  $D_2$  el conjunto de los juegos. Entonces “ $p = \text{la pelota}$ ” es una constante en  $D_2$  y obtenemos la formalización*

$$\forall x J(x, p).$$

d) *Sean  $D_1$  el conjunto de las personas y  $D_2$  el conjunto de los juegos. Entonces “ $p = \text{la pelota}$ ” es una constante en  $D_2$  y, usando el predicado “ $N(x) : x$  es un niño,” obtenemos la formalización*

$$\forall x (N(x) \rightarrow J(x, p)).$$

2. *Toda función se puede representar mediante un predicado con un argumento más que la función. Además, las funciones simplifican la estructura de la fórmula obtenida.*

**Ejemplo 6.4.3** *Consideremos la frase:*

*“Todo padre quiere mucho a sus hijos.”*

a) Formalización con predicados.

*Podemos definir el dominio  $D$  de las personas y los predicados*

$P(x, y) : x$  es el padre de  $y$ ,

y

$Q(x, y) : x$  quiere mucho a  $y$ .

*Con estas definiciones, la formalización sería*

$\forall x \forall y (P(x, y) \rightarrow Q(x, y)).$

b) Formalización con funciones.

*Podemos definir el dominio  $D$  de las personas, la función*

$f(x) : x$  es el padre de  $x$ ,

y

$Q(x, y) : x$  quiere mucho a  $y$ .

*Con estas definiciones, la nueva formalización sería*

$\forall x (Q(f(x), x)).$

*Notar que la formalización se ha simplificado y que la función de un argumento  $f(x)$  sustituye al predicado binario  $P(x, y)$ .*

*Observar también que “el hijo de  $x$ ” no es una función, ya que un mismo padre puede tener más que un hijo  $y$ , por tanto, el término asociado a  $x$  (al padre) no quedaría unívocamente determinado.*

**Patrones más habituales en la formalización de los cuantificadores:****1. Universal afirmativo**

$$\forall x(\varphi_1 \rightarrow \varphi_2),$$

$$\forall x(\neg\varphi_2 \rightarrow \neg\varphi_1).$$

Es la forma de representar frases del tipo:

Todo  $\varphi_1$  es  $\varphi_2$ ,

Sólo los  $\varphi_2$  son  $\varphi_1$ ,

Nadie es  $\varphi_1$  a menos que sea  $\varphi_2$ ,

No hay ningún  $\varphi_1$  que no sea  $\varphi_2$ ,

$\varphi_1$  es suficiente para  $\varphi_2$ ,

$\varphi_2$  es necesario para  $\varphi_1$ .

**2. Universal negativo**

$$\forall x(\varphi_1 \rightarrow \neg\varphi_2).$$

Es la forma de representar frases del tipo:

Ningún  $\varphi_1$  es  $\varphi_2$ ,

Todos los  $\varphi_1$  carecen de  $\varphi_2$ .

**3. Existencial afirmativo**

$$\exists x(\varphi_1 \wedge \varphi_2).$$

Es la forma de representar frases del tipo:

Algún  $\varphi_1$  es  $\varphi_2$ ,

Alguien es a la vez  $\varphi_1$  y  $\varphi_2$ .

**4. Existencial negativo**

$$\exists x(\varphi_1 \wedge \neg\varphi_2).$$

Es la forma de representar frases del tipo:

Algún  $\varphi_1$  no es  $\varphi_2$ ,

No todos los  $\varphi_1$  son  $\varphi_2$ .

**Ejemplos 6.4.4 1) (Universal afirmativo)**

*“Nadie se levanta a menos que tenga que irse.”*

*La frase anterior se puede reescribir como*

*“Para todo  $x$ , si  $x$  no tiene que irse, entonces no se levanta,”*

*o como*

*“Para todo  $x$ , si  $x$  se levanta, entonces tiene que irse.”*

*Sea  $D$  el dominio de las personas y sean*

$P(x) : x$  se levanta,

$Q(x) : x$  tiene que irse.

*Con estas definiciones obtenemos la formalización:*

$$\forall x (P(x) \rightarrow Q(x)).$$

**2) (Universal negativo)**

*“Ningún emperador es odontólogo (L. Carroll).”*

*Sea  $D$  el dominio de las personas y sean*

$P(x) : x$  es emperador,

$Q(x) : x$  es odontólogo.

*Con estas definiciones obtenemos la formalización:*

$$\forall x (P(x) \rightarrow \neg Q(x)).$$

**3) (Existencial afirmativo)**

*“Algunos estudiantes de informática sólo son amigos de los aficionados a la lógica [C].”*

*Esta frase se puede reescribir como:*

*“Para algunos estudiantes de informática, una persona es un amigo sólo si es aficionado a la lógica [C].”*

*Sea  $D$  el dominio de las personas y sean*

$P(x)$  :  *$x$  es estudiante de informática,*

$Q(x)$  :  *$x$  es aficionado a la lógica,*

$R(x, y)$  :  *$x$  es amigo de  $y$ .*

*Con estas definiciones obtenemos la formalización:*

$$\exists x (P(x) \wedge (R(x, y) \rightarrow Q(y))).$$

4) *(Existencial negativo)*

*“Algunos gatos no saben silbar ni maullar (L. Carroll).”*

*Sea  $D$  el dominio de los animales y sean*

$P(x)$  :  *$x$  es un gato,*

$Q(x)$  :  *$x$  sabe silbar,*

$R(x)$  :  *$x$  sabe maullar.*

*Con estas definiciones obtenemos la formalización:*

$$\exists x (P(x) \wedge \neg Q(x) \wedge \neg R(x)).$$

### **Negación de frases que contienen cuantificadores**

Vamos a ver como se escribe **la negación** de una frase que contiene un cuantificador.

#### **Universal-Existencial**

Consideremos la frase

*“Todos los alumnos de esta clase aprobarán en febrero.”*



Sean  $D$  el conjunto de los alumnos de esta clase y

$$P(x) : x \text{ aprobará en febrero.}$$

La frase dada se puede escribir como:

$$\forall x P(x).$$

La negación de “Todos los alumnos de esta clase aprobarán en febrero” es “No todos los alumnos de esta clase aprobarán en febrero,” es decir,

$$\neg(\forall x P(x)),$$

que podemos reescribir como:

“Existen alumnos de esta clase que no aprobarán en febrero.”

Con los mismos dominio y predicados anteriores, su formalización es

$$\exists x(\neg P(x)).$$

### **Existencial-Universal**

Consideremos ahora la frase

“Algunos alumnos de esta clase suspenderán en febrero.”

Sean  $D$  el conjunto de los alumnos de esta clase y

$$P(x) : x \text{ suspenderá en febrero.}$$

La frase dada se puede escribir como:

$$\exists x P(x).$$

La negación de “Algunos alumnos de esta clase suspenderán en febrero” es “Ningún alumno de esta clase suspenderá en febrero,” es decir,

$$\neg(\exists x P(x)),$$

que podemos reescribir como:

“Todos los alumnos de esta clase no suspenderán en febrero.”

Con los mismos dominio y predicados anteriores, su formalización es

$$\forall x(\neg P(x)).$$

**Ejemplo 6.4.5** (*Definición de límite de una sucesión*) Sean  $\mathbb{R}$  el conjunto de los números reales,  $\mathbb{R}^+$  el conjunto de los números reales positivos y  $\mathbb{N}$  el conjunto de los números naturales. Sea  $\{a_n\}_{n \in \mathbb{N}}$  una sucesión de números reales, es decir, una función

$$a : \mathbb{N} \longrightarrow \mathbb{R}.$$

Si  $\varepsilon \in \mathbb{R}^+$  y  $n, m \in \mathbb{N}$ , se dice que el número real  $L$  es el límite de  $\{a_n\}_{n \in \mathbb{N}}$ ,  $\lim_{n \rightarrow \infty} a_n = L$ , si

$$\forall \varepsilon \exists n \forall m ((m \geq n) \rightarrow |a_m - L| < \varepsilon).$$

La formalización matemática de la condición  $\lim_{n \rightarrow \infty} a_n \neq L$  es

$$\exists \varepsilon \forall n \exists m ((m \geq n) \wedge |a_m - L| \geq \varepsilon).$$

**Ejemplo 6.4.6** *Vamos a formalizar el siguiente razonamiento [C]:*

*Sólo las buenas personas ayudan a los pobres. Ninguna buena persona es aficionada a la fotografía. Antonio ayuda a Juan. Antonio es aficionado a la fotografía. Entonces, Juan es pobre.*

Sea  $D$  el dominio de las personas,  $a$  la constante Antonio y  $j$  la constante Juan. Definamos los siguientes predicados:

$$P(x) : x \text{ es buena persona,}$$

$$Q(x, y) : x \text{ ayuda a } y,$$

$$R(x) : x \text{ es pobre,}$$

$$S(x) : x \text{ es aficionado a la fotografía.}$$

Con estas definiciones el razonamiento dado se puede escribir como:

$$\begin{array}{l}
\forall x \forall y (Q(x, y) \wedge R(y) \rightarrow P(x)), \\
\forall x (P(x) \rightarrow \neg S(x)), \\
Q(a, j), \\
S(a) \\
\hline
R(j)
\end{array}$$

**Observación 6.4.7 (Lógicas de predicados de orden superior)**

*El cálculo de predicados de primer orden admite generalizaciones a cálculos de predicados de orden mayor que uno. En el cálculo de predicados de primer orden los cuantificadores pueden afectar sólo a las variables y los predicados se calculan sólo sobre términos.*

*En el cálculo de predicados de segundo orden, los cuantificadores afectan también a predicados.*

*En el cálculo de tercer orden se definen predicados de predicados (no sólo predicados de términos).*

*Siguiendo añadiendo niveles de “predicados de predicados,” se sube el nivel del cálculo de predicados que se está definiendo.*

## 6.5 Ejercicios

**Ejercicio 6.5.1** *Determina cuáles de las siguientes fórmulas (en forma abreviada) son abiertas y cuáles son cerradas. En cada caso, identifica las variables libres y ligadas.*

1.  $\forall x(f(x, y) \rightarrow P(g(x), a, b)),$
2.  $\exists y(\forall x f(x, y) \rightarrow P(g(x), a, b)),$
3.  $\exists y \forall x f(x, y) \wedge P(g(x), a, b),$
4.  $\exists y \exists x f(x, y) \wedge \neg \forall x P(g(x), a, b).$

**Ejercicio 6.5.2** *Representa en forma de árbol estructural las fórmulas del ejercicio anterior y los siguientes términos:*

1.  $f(g(a, x), h(x, i(y, b)), c),$
2.  $f(g(x), h(x, y), i(y, b), l(x, c)).$

**Ejercicio 6.5.3 [HLR]** *Usando el principio de recursión estructural, define la función*

$$Lig : \mathbf{F} \longrightarrow P(\mathbf{Var}),$$

*que a cada fórmula  $\varphi$  asocia el conjunto  $Lig(\varphi)$  de sus variables ligadas.*

**Ejercicio 6.5.4** *Formaliza las siguientes frases en lógica de primer orden. Para cada una de ellas, escribe su negación y vuelve a traducirla al lenguaje natural.*

1. *Sólo los científicos que trabajan en áreas aplicadas son famosos.*
2. *Algunos caballos son salvajes.*
3. *Todas las personas tienen algún amigo.*
4. *En los números reales, el producto de cualquier número real positivo por cualquier número negativo es negativo. El producto de cualquier número positivo por cualquier número positivo es positivo.*

5. *En los números naturales, el siguiente de cualquier número par no es par. El siguiente de cualquier número no par es par. El producto de cualquier número par por cualquier natural es par.*
6. *Si el producto de dos números naturales es múltiplo de un primo, entonces uno de ellos es múltiplo del primo.*
7. *Sólo las buenas enfermeras atienden con paciencia a los enfermos tísicos.*
8. *Los aficionados del Madrid son amigos de los aficionados del Betis. Algunos aficionados del Madrid son amigos de los aficionados del Betis. Algunos aficionados del Madrid sólo son amigos de aficionados del Betis.*
9. *Dos hermanos nunca tienen la misma opinión respecto a todos los aspectos de la vida.*

**Ejercicio 6.5.5** *Formaliza el siguiente argumento en la lógica de primer orden:*

*Dos personas son hermanas si tienen el mismo padre. Dos personas son primas si tienen el mismo abuelo. El padre de Juan es hijo único. Por tanto: sus primos son sus hermanos.*



# Capítulo 7

## Semántica de la lógica de primer orden. Teoría interpretativa

Como ya hicimos en el capítulo 4 para la lógica proposicional, en este capítulo vamos a estudiar la semántica y sus sistemas de demostración en la lógica de primer orden. Volvemos a recordar sus definiciones:

La **semántica** es la definición de un conjunto de significados (generalmente verdadero o falso) que se puedan asociar a una expresión bien construida. Permite definir la validez de una expresión o de un razonamiento.

Los **sistemas de demostración** son sistemas formales que permiten averiguar cuándo una expresión o un razonamiento son válidos. En el contexto de la semántica se denominan **teoría interpretativa**.

Volveremos a estudiar los conceptos fundamentales de validez de una fórmula, de consecuencia lógica y de equivalencia entre fórmulas en el contexto más complejo y detallado de la lógica de primer orden.

A pesar de las muchas analogías entre la semántica de la lógica proposicional y la de la lógica de primer orden, veremos que hay varias diferencias entre ellas. La principal es que en la lógica de primer orden no hay un algoritmo de decisión de validez de fórmulas (como veremos, se pierde la propiedad de decidibilidad). Los métodos de las tablas de verdad y de los tableaux de la lógica proposicional no se pueden extender a la lógica de primer orden.

## 7.1 Interpretaciones en lógica de primer orden

Para asignar un significado a una expresión bien construida de la lógica de primer orden necesitamos unos métodos más complejos que aquellos estudiados en la lógica proposicional. Este hecho es debido a la mayor precisión y variedad de los elementos básicos del lenguaje de la lógica de primer orden y, por tanto, a la necesidad de explicar cómo interpretar cada uno de ellos respetando su naturaleza sintáctica.

Lo primero que se necesita es establecer un universo del discurso, un **dominio**, para definir qué tipo de objetos estamos analizando.

A continuación, tendremos que definir cómo asignar un significado, en el dominio elegido, a los elementos básicos del lenguaje.

Finalmente, los significados de los elementos básicos nos proporcionarán, con un procedimiento de recursión estructural, los significados de los términos y de las fórmulas.

Necesitamos entonces empezar con asignar significados a constantes, funciones, predicados (algo que llamaremos una **interpretación**) y a variables (una **asignación**).

Estos conceptos se corresponden al concepto de valoración visto en la lógica proposicional.

En la lógica de primer orden una **signatura**  $\Sigma$  es un conjunto de símbolos de funciones y de predicados con aridades asociadas.

Una signatura puede ser finita o infinita numerable. Supondremos que sea decidable, es decir, que sea posible reconocer efectivamente sus símbolos con sus aridades.

El lenguaje  $\mathbf{L}_\Sigma$  es el conjunto de todas las fórmulas que se pueden construir a partir de los elementos de  $\Sigma$ .

Cuando no haga falta especificar la signatura que se está usando, indicaremos un lenguaje simplemente con el símbolo  $\mathbf{L}$ .

**Definición 7.1.1** *Sea  $\mathbf{L}$  un lenguaje de la lógica de primer orden. Una **interpretación** es cualquier par  $\mathbb{I} = (D, I)$  donde:*

- $D$  es un conjunto no vacío, denominado **dominio** de la interpretación,
- $I$  es una función que asocia:



1. a cada símbolo de **constante**  $c$  del lenguaje un elemento  $c^I$  del conjunto  $D$ ,
2. a cada símbolo de **función**  $f$  del lenguaje con aridad  $n > 0$  una función  $f^I : D^n \rightarrow D$ ,
3. a cada símbolo de **proposición atómica**  $p$  un elemento  $p^I$  del conjunto  $\{0, 1\}$  (una valoración),
4. a cada símbolo de **predicado**  $P$  del lenguaje con aridad  $n > 0$  una relación  $P^I \subseteq D^n$ .

**Definición 7.1.2** Dada una interpretación  $\mathbb{I} = (D, I)$  para un lenguaje de primer orden, se llama **asignación** a una función  $A$  que asocia a cada símbolo de variable  $x$  un elemento  $x^A$  perteneciente al conjunto  $D$ .

**Observación 7.1.3** En la literatura la terminología empleada no es uniforme: en varios textos se denomina interpretación al par formado por una **estructura** (interpretación en nuestra notación) y un **estado** (asignación en nuestra notación).

En estos apuntes la definición de interpretación es la 7.1.1 y, por tanto, no incluye ninguna asignación. En cada caso, se tendrá que especificar la asignación elegida para una dada interpretación.

La justificación de la notación elegida es que, como veremos, nuestras definiciones permiten definir más claramente los conceptos de fórmula satisfacible bajo una interpretación y de modelo de una fórmula (ver definición 7.3.1).

**Ejemplo 7.1.4**  $[F]$

Sea  $\Sigma = \{a, f, g\}$  una signatura, donde  $a$  es una constante,  $f$  es una función unaria y  $g$  una función binaria.

Vamos a ver tres posibles interpretaciones y asignaciones:

1) Definimos el dominio

$$D_1 = \{0, 1, 2, \dots\} = \mathbb{N} \cup \{0\},$$

la interpretación

$$I : \quad a^I = 0, \quad f^I = \text{suc}(\text{el siguiente}), \quad g^I = + (\text{la suma}),$$

y la asignación

$$A : x^A = 1.$$

2) Definimos el dominio  $D_2$  como el conjunto de todas las posibles palabras que se pueden construir con el alfabeto  $\{*, @\}$ , la interpretación

$$I : a^I = *, \quad f^I = \text{añade } * \text{ al final de la palabra,}$$

$$g^I = + \text{ (la concatenación),}$$

y la asignación

$$x^A = *@*.$$

3) Definimos el dominio

$$D_3 = \{\dots, -2, -1, 0, 1, 2, \dots\} = \mathbb{Z},$$

la interpretación

$$I : a^I = 1, \quad f^I = \text{ant (el anterior),} \quad g^I = - \text{ (la resta),}$$

y la asignación

$$A : x^A = 1.$$

**Observación 7.1.5** Hay una clara analogía entre el concepto de asignación de la definición 7.1.2 y de asignación de un estado a las variables de un programa en un lenguaje de programación.

Si en un programa ejecutamos el comando “ $x := a$ ,” el estado de la variable  $x$  se ha modificado y ahora su valor es el elemento  $a$  del conjunto del tipo de datos que se está considerando.

En nuestra notación, siendo  $D$  el dominio de una interpretación y  $A$  una asignación para esa interpretación, el nuevo estado de la variable  $x$  es  $x^A = a \in D$ .

**NOTACIÓN:** Si  $A$  es una asignación, escribiremos  $A[x/d]$  para indicar una nueva asignación que difiere de  $A$  sólo en la variable  $x$ , a la cual ya se ha asignado el valor  $d \in D$ .

## 7.2 Interpretación semántica de términos y fórmulas

El siguiente paso hacia la definición de significados para expresiones bien construidas consiste en el usar el principio de recursión estructural para poder extender una interpretación y una asignación dadas a todos los términos y a todas las fórmulas. Esta extensión nos permitirá asociar valores de verdad  $\{0, 1\}$  a las fórmulas y estudiar su validez.

### Definición 7.2.1 (Definición por recursión estructural de interpretación de términos)

Dada una interpretación  $\mathbb{I} = (D, I)$  para un lenguaje de primer orden, y dada una asignación  $A$  para esa interpretación, se asocia a cada término del lenguaje  $t$  un elemento  $t^{I,A}$  perteneciente a  $D$  de la siguiente forma:

1. **Base (TAt):**

- si  $t$  es una constante  $c$ , entonces

$$t^{I,A} = c^I,$$

- si  $t$  es una variable  $x$ , entonces

$$t^{I,A} = x^A,$$

2. **Paso recursivo (TF):** si  $t$  es un término de la forma  $f(t_1, \dots, t_n)$ , entonces

$$t^{I,A} = f^I(t_1^{I,A}, \dots, t_n^{I,A}).$$

**Ejemplo 7.2.2** Usando las definiciones del ejemplo anterior 7.1.4, los términos

$$t_1 = f(g(f(a), f(x)))$$

y

$$t_2 = \{f(g(x, f(g(x, f(a))))))\}$$

se interpretan como se describe a continuación:

1)

$$t_1^{I,A} = \text{suc}(\text{suc}(0) + \text{suc}(1)) = 4,$$

$$t_2^{I,A} = \text{suc}(1 + \text{suc}(1 + \text{suc}(0))) = 5.$$

2)

$$t_1^{I,A} = f^I(f^I(*) + f^I(*@*)) = f^I(** + * @ **) = ***@***.$$

$$\begin{aligned} t_2^{I,A} &= f^I(*@* + f^I(*@* + f^I(*))) = \\ &= f^I(*@* + * @ ***) = *@**@****. \end{aligned}$$

3)

$$t_1^{I,A} = \text{ant}(\text{ant}(1) - \text{ant}(1)) = -1,$$

$$t_2^{I,A} = \text{ant}(x^A - \text{ant}(x^A - \text{ant}(1))) = \text{ant}(x^A - \text{ant}(x^A)) = \text{ant}(1) = 0.$$

El significado de una fórmula, fijada una interpretación y una asignación, tiene que ser un valor de verdad. Seguiremos usando 0 para representar el valor falso y 1 para representar el valor verdadero.

La siguiente definición nos permite calcular valores de verdad de fórmulas.

### Definición 7.2.3 (Definición por recursión estructural de interpretación de fórmulas)

Dada una interpretación  $\mathbb{I} = (D, I)$  para un lenguaje de primer orden, y dada una asignación  $A$  para esa interpretación, se asocia a cada fórmula del lenguaje  $\varphi$  un valor de verdad  $\varphi^{I,A}$  perteneciente a  $\{0, 1\}$  de la siguiente forma:

#### 1. Base (FAt):

- $\perp^{I,A} = 0$  ;  $\top^{I,A} = 1$ ,
- si  $p$  es una proposición atómica,  $p^{I,A} = p^I$ ,
- si  $s = t$  es la igualdad entre dos términos,

$$(s = t)^{I,A} = \begin{cases} 1 & \text{si } s^{I,A} = t^{I,A}, \\ 0 & \text{si } s^{I,A} \text{ no es igual a } t^{I,A}, \end{cases}$$

- si  $\varphi = P(t_1, \dots, t_n)$ , entonces

$$\varphi^{I,A} = 1 \quad \text{si y sólo si} \quad (t_1^{I,A}, \dots, t_n^{I,A}) \in P^I,$$

es decir, si y sólo si los elementos  $(t_1^{I,A}, \dots, t_n^{I,A})$  están relacionados mediante la relación  $P^I$ ,

## 2. Paso recursivo:

- ( $F\neg$ ): si  $\varphi = \neg\psi$ , entonces

$$\varphi^{I,A} = \neg(\psi^{I,A}),$$

siendo  $\neg(0) = 1$  y  $\neg(1) = 0$ ,

- ( $F\circ$ ): si  $\varphi = (\varphi_1 \circ \varphi_2)$ , entonces

$$\varphi^{I,A} = \varphi_1^{I,A} \circ \varphi_2^{I,A},$$

donde el valor de  $\varphi_1^{I,A} \circ \varphi_2^{I,A}$ , viene dado por la tabla 7.1, que coincide con la tabla 4.1 de la lógica proposicional:

$\varphi_1^{I,A}$	$\varphi_2^{I,A}$	$\neg\varphi_1^{I,A}$	$\wedge$	$\vee$	$\rightarrow$	$\leftrightarrow$
0	0	1	0	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	0	0
1	1	0	1	1	1	1

Tabla 7.1: Valores de verdad de los conectivos

- ( $F\exists$ ): si  $\varphi = \exists x\psi$ , entonces  $\varphi^{I,A} = 1$  si y sólo si existe algún elemento del dominio  $D$  tal que, después de asignar ese valor a la variable  $x$  en  $\psi$ , se tiene  $\psi^{I,A} = 1$ .

De forma equivalente,  $\varphi^{I,A} = 1$  si y sólo si existe algún  $d \in D$  tal que  $\psi^{I,A[x/d]} = 1$ ,

- ( $F\forall$ ): si  $\varphi = \forall x\psi$ , entonces  $\varphi^{I,A} = 1$  si y sólo si se tiene  $\psi^{I,A} = 1$  para cualquier posible asignación de un elemento del dominio  $D$  a la variable  $x$  en  $\psi$ .

De forma equivalente,  $\varphi^{I,A} = 1$  si y sólo si para todo  $d \in D$  se verifica que  $\psi^{I,A[x/d]} = 1$ .

**Ejemplos 7.2.4** 1) Consideremos la fórmula

$$\varphi = \exists x(P(f(x), a)).$$

En la interpretación  $(D_1, I_1)$  definida por:

$$D_1 = \mathbb{R}, a^{I_1} = 2, f^{I_1}(x) = x^2, P^{I_1} = \{(x, y) \in \mathbb{Z} \times \mathbb{Z} : x = y\},$$

la fórmula  $\varphi$  se escribe

$$\exists x(x^2 = 2)$$

y tiene valor 1 ya que existe la raíz cuadrada de 2.

En la interpretación  $(D_2, I_2)$  definida por:

$$D_2 = \mathbb{C}, a^{I_2} = -2, f^{I_2}(x) = x^2, P^{I_2} = \{(x, y) \in \mathbb{Z} \times \mathbb{Z} : x = y\},$$

la fórmula  $\varphi$  se escribe

$$\exists x(x^2 = -2)$$

tiene valor 1 ya que todo número complejo tiene dos raíces cuadradas complejas.

2) Consideremos la fórmula

$$\varphi = \forall x \forall y (P(f(x, y), a) \rightarrow P(x, y))$$

en la interpretación  $(D_1, I_1)$  definida por:

$$D_1 = \mathbb{Z}, a^{I_1} = 0, f^{I_1}(x, y) = x - y, P^{I_1} = \{(x, y) \in \mathbb{Z} \times \mathbb{Z} : x < y\}.$$

Con la interpretación dada, la fórmula  $\varphi$  se escribe

$$\forall x \forall y ((x - y < 0) \rightarrow x < y)$$

y tiene valor 1.

Sin embargo, si consideramos la interpretación  $(D_2, I_2)$  definida por:

$$D_2 = \mathbb{Z}, a^{I_2} = 4, f^{I_2}(x, y) = x - y, P^{I_2} = \{(x, y) \in \mathbb{Z} \times \mathbb{Z} : x < y\},$$

entonces  $\varphi^{I_2, A_2} = 0$ . En efecto, si  $x = 5$  y  $y = 2$ , se obtiene que  $(x - y < 4)$  es cierta, pero  $x < y$  es falsa.

## 7.3 Validez semántica de fórmulas: modelos

Las definiciones y resultados del párrafo anterior nos permiten definir el cálculo de los valores de verdad de una fórmula. Las siguientes definiciones son análogas a las definiciones de satisfacibilidad y validez vistas para fórmulas proposicionales y nos permiten estudiar la validez y clasificar las fórmulas de la lógica de primer orden.

### Definición 7.3.1

- Se dice que una fórmula  $\varphi$  es **satisfacible bajo una interpretación**  $(D, I)$  cuando se verifica  $\varphi^{I,A} = 1$  para **alguna** asignación  $A$ .
- Se dice que una fórmula  $\varphi$  es **satisfacible** cuando es satisfacible bajo alguna interpretación  $(D, I)$ . Por tanto, una fórmula  $\varphi$  es satisfacible si existen al menos una interpretación  $(D, I)$  y una asignación  $A$  (relativa a esa interpretación) tales que  $\varphi^{I,A} = 1$ .
- Si una fórmula no es satisfacible se dice que es **insatisfacible**.
- Se dice que una fórmula  $\varphi$  es **verdadera bajo una interpretación**  $\mathbb{I} = (D, I)$  cuando se verifica  $\varphi^{I,A} = 1$  para **cualquier** asignación  $A$ . En este caso se dice también que  $\mathbb{I}$  es un **modelo de**  $\varphi$  y se escribe  $\mathbb{I} \models \varphi$ .
- Se dice que una fórmula  $\varphi$  es **válida** cuando es verdadera bajo cualquier interpretación, y se denota  $\models \varphi$ . Por tanto, una fórmula  $\varphi$  es válida si para toda interpretación  $(D, I)$  y una asignación  $A$  (relativa a esa interpretación) se verifica que  $\varphi^{I,A} = 1$ .
- Si una fórmula no es válida se dice que es **falsificable**.

**Observación 7.3.2** El concepto de fórmula insatisfacible en LPO se corresponde al concepto de fórmula insatisfacible (o contradicción) en LP. Es una fórmula que es siempre falsa.

El concepto de fórmula válida en LPO se corresponde al concepto de fórmula válida (o tautología) en LP. Es una fórmula que es siempre verdadera.

**Ejemplo 7.3.3** [F] Sea

$$\varphi : \exists y R(x, f(y, y)).$$

Siendo la variable  $x$  libre,  $\varphi$  es abierta.

Sean  $D = \{1, 2, 3, \dots\} = \mathbb{N}$ ,  $R^I = "="$  y  $f^I = +$ .

Entonces, para una asignación  $A$ ,

$$\varphi^{I,A} : \exists y(x^A = 2y).$$

Si  $A$  es tal que  $x^A$  es un número par, entonces  $\varphi^{I,A} = 1$ .

Si  $A$  es tal que  $x^A$  es un número impar, entonces  $\varphi^{I,A} = 0$ .

Se sigue que  $\varphi$  es satisfacible bajo  $(D, I)$ , pero no es verdadera bajo  $(D, I)$ . En particular,  $\varphi$  no es válida.

**Ejemplo 7.3.4** [C] Sea

$$\varphi : \forall x(P(x) \rightarrow R(x)) \rightarrow \exists yQ(y).$$

Sean  $D = \{a, b, c\}$ ,  $P^I = \{a, b\}$ ,  $R^I = \{b\}$  y  $Q^I = \emptyset$ .

Usando la siguiente tabla (el dominio  $D$  contiene sólo tres elementos) podemos determinar el valor de la subfórmula  $\forall x(P(x) \rightarrow R(x))$  :

$x$	$P^I(x)$	$R^I(x)$	$(P(x) \rightarrow R(x))^I$	$(\forall x(P(x) \rightarrow R(x)))^I$
$a$	1	0	0	0
$b$	1	1	1	
$c$	0	0	1	

Se sigue que  $(\forall x(P(x) \rightarrow R(x)))^I = 0$  (esta conclusión se podría haber deducido mirando sólo a la primera fila de la tabla). Por tanto, la fórmula inicial  $\varphi$  tiene valor 1 bajo la interpretación dada, siendo una implicación con premisa siempre falsa. Se deduce que la interpretación  $(D, I)$  es un modelo de  $\varphi$ .

**Observación 7.3.5** 1) Una fórmula  $\varphi$  es válida si y sólo si su negación,  $\neg\varphi$ , es insatisfacible.

2) Si una fórmula  $\varphi$  es cerrada, dada una interpretación  $(D, I)$ , su valor de verdad no depende de la asignación elegida y se puede denotar  $\varphi^I$ .

En este caso  $\varphi$  es satisfacible bajo  $(D, I)$  si y sólo si es verdadera bajo  $(D, I)$ .

3) Si una fórmula  $\varphi$  (no cerrada) contiene las variables libres  $\{x_1, x_2, \dots, x_n\}$ , se llama **cierre universal de  $\varphi$**  a la fórmula cerrada

$$\forall x_1 \forall x_2 \dots \forall x_n \varphi(x_1, x_2, \dots, x_n).$$

Por tanto, en el caso de una fórmula  $\varphi$  no cerrada:



- $\varphi$  es verdadera bajo  $(D, I)$  si y sólo si  $\forall x_1 \forall x_2 \dots \forall x_n \varphi(x_1, x_2, \dots, x_n)$  es verdadera bajo  $(D, I)$  y
- $\varphi$  es válida si y sólo si  $\forall x_1 \forall x_2 \dots \forall x_n \varphi(x_1, x_2, \dots, x_n)$  es válida.

**NOTA:** de las anteriores observaciones se sigue que **basta estudiar la validez de fórmulas cerradas.**

## 7.4 Evaluación semántica de deducciones

Otro concepto fundamental que vamos a definir es el concepto de consecuencia lógica.

Las definiciones son similares a las vistas en lógica proposicional, teniendo en cuenta que tenemos que reemplazar el concepto de valoración por los de interpretación y asignación.

**Definición 7.4.1** Se dice que una fórmula  $\varphi$  es **consecuencia lógica** de un conjunto finito de fórmulas  $\Phi = \{\varphi_1, \varphi_2, \dots, \varphi_n\}$  si cualquier interpretación  $(D, I)$  y cualquier asignación  $A$  que satisfacen todo elemento del conjunto  $\Phi$  satisfacen también la fórmula  $\varphi$ .

De forma equivalente,  $\varphi$  es consecuencia lógica de  $\Phi$  si  $\varphi_i^{I,A} = 1$  para todo  $i \in \{1, 2, \dots, n\}$  implica  $\varphi^{I,A} = 1$ .

Si  $\varphi$  es consecuencia lógica de  $\Phi$  también se dice que  $\Phi$  **implica lógicamente** a  $\varphi$  y se escribe  $\Phi \models \varphi$ .

A continuación recordamos la definición de razonamiento válido y la notación empleada en el capítulo 4.

**Definición 7.4.2** Sean  $\Phi = \{\varphi_1, \varphi_2, \dots, \varphi_n\}$  un conjunto finito de fórmulas y  $\varphi$  una fórmula. Se define **deducción** o **razonamiento** a la fórmula

$$((\varphi_1 \wedge \varphi_2 \wedge \dots \wedge \varphi_n) \rightarrow \varphi).$$

Se dice que el razonamiento anterior es **correcto** o **lógicamente válido** si

$$(\varphi_1 \wedge \varphi_2 \wedge \dots \wedge \varphi_n) \models \varphi,$$

es decir, si el conjunto  $\Phi$  de las **premisas** o **hipótesis** del razonamiento implica lógicamente a la **conclusión** o **tesis**  $\varphi$ .

Usando la notación ya empleada para la lógica proposicional, para distinguir las premisas de la conclusión, el razonamiento anterior se suele representar por

$$\frac{\begin{array}{c} \varphi_1 \\ \vdots \\ \varphi_n \end{array}}{\varphi}$$

El siguiente teorema extiende a la lógica de primer orden la relación existente entre la validez de una fórmula y el concepto de consecuencia lógica ya vista en el contexto de la lógica proposicional.

**Teorema 7.4.3** [HLR]

Sean  $\Phi = \{\varphi_1, \varphi_2, \dots, \varphi_n\}$  un conjunto finito de fórmulas y  $\varphi$  una fórmula.

Las siguientes afirmaciones son equivalentes:

- 1)  $\varphi$  es consecuencia lógica de  $\Phi$ ,
- 2)  $(\varphi_1 \wedge \varphi_2 \wedge \dots \wedge \varphi_n) \rightarrow \varphi$  es una fórmula válida (y se escribe  $\models ((\varphi_1 \wedge \varphi_2 \wedge \dots \wedge \varphi_n) \rightarrow \varphi)$ ),
- 3)  $\varphi_1 \wedge \varphi_2 \wedge \dots \wedge \varphi_n \wedge \neg\varphi$  es insatisfacible (**Reducción al absurdo**).

**Ejemplo 7.4.4** Volvamos a considerar el razonamiento

$$\frac{\begin{array}{c} \text{Todos los hombres son mortales,} \\ \text{Sócrates es un hombre,} \end{array}}{\text{Sócrates es mortal.}}$$

Vimos que este razonamiento no es válido en lógica proposicional. Su formalización en lógica de primer orden es

$$\frac{\begin{array}{c} \forall x(H(x) \rightarrow M(x)), \\ H(s), \end{array}}{M(s)}.$$

En este caso  $\Phi = \{\forall x(H(x) \rightarrow M(x)), H(s)\}$  y  $\varphi : M(s)$ . Queremos verificar si  $\Phi$  implica lógicamente  $\varphi$ .

$\models t = t$	Reflexividad de la igualdad
$t = t' \models t' = t$	Simetría de la igualdad
$\{t = t', t' = t''\} \models t = t''$	Transitividad de la igualdad
$\{\varphi[x/t], t = t'\} \models \varphi[x/t']$	Sustitutividad de la igualdad
$\forall x\varphi(x) \models \varphi[x/t]$	Hipótesis universal
$\varphi[x/t] \models \exists x\varphi(x)$	Tesis existencial

Tabla 7.2: Tabla de algunas implicaciones lógicas

Sean  $(D, I)$  y  $A$  tales que

$$(\forall x(H(x) \rightarrow M(x)))^{I,A} = 1, \quad (H(s))^{I,A} = 1.$$

Se trata de verificar si, necesariamente, tiene que ser  $M(s)^{I,A} = 1$ .

Por definición,  $(\forall x(H(x) \rightarrow M(x)))^{I,A} = 1$  si y sólo si  $(H(x) \rightarrow M(x))^{I,A} = 1$  para todo  $x^A \in D$ . En particular,  $(H(s) \rightarrow M(s))^{I,A} = 1$ . Siendo  $H(s)^{I,A}$  y  $M(s)^{I,A}$  fórmulas proposicionales, la condición anterior es equivalente a  $(\neg H(s) \vee M(s))^{I,A} = 1$ .

Por hipótesis sabemos que  $(H(s))^{I,A} = 1$ , es decir, sabemos que  $(\neg H(s))^{I,A} = 0$ .

Se sigue que para que se verifique  $(\neg H(s) \vee M(s))^{I,A} = 1$  tiene que ser  $(M(s))^{I,A} = 1$ .

Por tanto  $\Phi$  implica lógicamente  $\varphi$ .

**Ejemplo 7.4.5 [HLR]** Se puede verificar (ver ejercicio 7.6.7) que las fórmulas de la tabla 7.2 son implicaciones lógicas.

## 7.5 Equivalencia de fórmulas

### 7.5.1 Sustitución de una variable por un término

Sea  $\varphi$  una fórmula que contenga una variable  $x$ . Si queremos sustituir la variable  $x$  por un término  $t$  para obtener la nueva fórmula  $\varphi[x/t]$ , es necesario no modificar el significado de la fórmulas original.

En la nueva fórmula  $\varphi[x/t]$ , ver [HLR],

- sólo las apariciones libres de  $x$  en  $\varphi$  se reemplazan por  $t$ ,

- si algún cuantificador de  $\varphi$  hace que queden ligadas variables de  $t$  después de la sustitución, la variable ligada de esa cuantificación se reemplaza por otra.

**Ejemplo 7.5.1** [MH]

Sea

$$\varphi : \forall x R(x, y)$$

y sea  $\mathbb{I} = (D, I)$  la interpretación con dominio  $D = \mathbb{N}$  (los números naturales) y  $R^I(x, y) = \{(x, y) \in \mathbb{N} \times \mathbb{N} : x \leq y\}$  (la relación de orden usual en  $\mathbb{N}$ ).

Sustituyendo la variable libre  $y$  por  $z$  obtenemos la fórmula

$$\varphi[y/z] : \forall x R(x, z),$$

y no se altera el significado de  $\varphi$ . Las dos fórmulas son falsas bajo  $\mathbb{I} = (D, I)$ , ya que el conjunto  $\mathbb{N}$  no tiene máximo.

Sin embargo, si sustituimos  $y$  por  $x$  obtenemos la fórmula

$$\varphi[y/x] : \forall x R(x, x)$$

que es verdadera bajo  $\mathbb{I} = (D, I)$ , siendo cierto que un cualquier número natural  $n$  verifica que  $n \leq n$  (es la propiedad reflexiva de la relación).

Para sustituir  $y$  por  $x$  sin alterar el significado de la fórmula y el tipo de ocurrencia de sus variables, tendríamos que cambiar de nombre también a la variable  $x$ . Por ejemplo, se puede sustituir  $x$  por  $v$  y  $y$  por  $x$ . Así se obtiene la fórmula:

$$\varphi[x/v, y/x] : \forall v R(v, x).$$

Respetando los dos criterios anteriores, se puede definir recursivamente y rigurosamente la noción de sustitución de una variable por un término. Además, se puede demostrar el **lema de sustitución** [HLR], que afirma que interpretar la fórmula  $\varphi[x/t]$  en una interpretación  $\mathbb{I} = (D, I)$  y una asignación  $A$  es lo mismo que interpretar la fórmula original  $\varphi$  en la misma interpretación  $\mathbb{I} = (D, I)$  y en la asignación  $A[x/t]$ .

**Ejemplo 7.5.2** [HLR] Sea  $\varphi : \exists z(x \cdot z = y)$  en el dominio de los números naturales  $\mathbb{N}$  con el producto y la suma usuales. Para sustituir correctamente la variable  $y$  por  $z + z$ , tendremos que escribir

$$\varphi[y/z + z] : \exists u(x \cdot u = z + z).$$

**Observación 7.5.3** *En informática, la operación de sustitución es importante en la **verificación de programas**. La ejecución de un programa tiene como efecto la transformación del estado de unas variables de una condición inicial a una condición final. Verificar un programa consiste en demostrar que, si el estado inicial de las variables cumple una cierta condición inicial, entonces su estado final cumple con la condición final deseada. Las condiciones iniciales y finales se suelen representar por medio de fórmulas de la lógica de primer orden. Para asignar valores a estas condiciones se introduce una interpretación que representa el tipo de datos sobre el cuál se puede operar (ver, por ejemplo, [HLR] o [C]).*

## 7.5.2 Equivalencia de fórmulas

El último concepto fundamental que vamos a estudiar en este capítulo es el concepto de equivalencia lógica entre fórmulas. Las definiciones y resultados son similares a los ya estudiados para la lógica proposicional en el capítulo 4. Para la lógica de primer orden tendremos que añadir a nuestra lista las equivalencias relativas a los cuantificadores, que no aparecen en la lógica proposicional.

**Definición 7.5.4** *Sean  $\varphi$  y  $\psi$  dos fórmulas. Se dice que  $\varphi$  **equivale lógicamente** a  $\psi$  si y sólo si*

$$\varphi \models \psi \quad \text{y} \quad \psi \models \varphi.$$

*Si  $\varphi$  y  $\psi$  son equivalentes se escribe  $\varphi \equiv \psi$ .*

**Observación 7.5.5** *Se sigue de la definición que si  $\varphi$  y  $\psi$  son dos fórmulas equivalentes, entonces tienen los mismos valores de verdad bajo una cualquier interpretación y asignación. Por tanto, no es difícil verificar que las siguientes afirmaciones son equivalentes:*

- $\varphi \equiv \psi$ .
- $\models \varphi \leftrightarrow \psi$ .
- $\models \varphi \rightarrow \psi$  y  $\models \psi \rightarrow \varphi$ .

$\varphi \wedge \top \equiv \varphi$ $\varphi \wedge \perp \equiv \perp$ $\varphi \vee \top \equiv \top$ $\varphi \vee \perp \equiv \varphi$	Leyes de Identidad
$\varphi \wedge \neg\varphi \equiv \perp$	No contradicción
$\varphi \vee \neg\varphi \equiv \top$	Tercio excluso
$\varphi \wedge \varphi \equiv \varphi$ $\varphi \vee \varphi \equiv \varphi$	Idempotencia
$\varphi_1 \wedge (\varphi_1 \vee \varphi_2) \equiv \varphi_1$ $\varphi_1 \vee (\varphi_1 \wedge \varphi_2) \equiv \varphi_1$	Absorción
$\varphi_1 \wedge \varphi_2 \equiv \varphi_2 \wedge \varphi_1$ $\varphi_1 \vee \varphi_2 \equiv \varphi_2 \vee \varphi_1$ $(\varphi_1 \leftrightarrow \varphi_2) \equiv (\varphi_2 \leftrightarrow \varphi_1)$	Conmutatividad
$(\varphi_1 \wedge \varphi_2) \wedge \varphi_3 \equiv \varphi_1 \wedge (\varphi_2 \wedge \varphi_3)$ $(\varphi_1 \vee \varphi_2) \vee \varphi_3 \equiv \varphi_1 \vee (\varphi_2 \vee \varphi_3)$	Asociatividad
$\varphi_1 \wedge (\varphi_2 \vee \varphi_3) \equiv (\varphi_1 \wedge \varphi_2) \vee (\varphi_1 \wedge \varphi_3)$ $\varphi_1 \vee (\varphi_2 \wedge \varphi_3) \equiv (\varphi_1 \vee \varphi_2) \wedge (\varphi_1 \vee \varphi_3)$ $\varphi_1 \rightarrow (\varphi_2 \wedge \varphi_3) \equiv (\varphi_1 \rightarrow \varphi_2) \wedge (\varphi_1 \rightarrow \varphi_3)$ $\varphi_1 \rightarrow (\varphi_2 \vee \varphi_3) \equiv (\varphi_1 \rightarrow \varphi_2) \vee (\varphi_1 \rightarrow \varphi_3)$ $\varphi_1 \leftrightarrow (\varphi_2 \wedge \varphi_3) \equiv (\varphi_1 \leftrightarrow \varphi_2) \wedge (\varphi_1 \leftrightarrow \varphi_3)$ $\varphi_1 \leftrightarrow (\varphi_2 \vee \varphi_3) \equiv (\varphi_1 \leftrightarrow \varphi_2) \vee (\varphi_1 \leftrightarrow \varphi_3)$	Distributividad
$\neg(\neg\varphi) \equiv \varphi$	Doble negación
$(\varphi_1 \rightarrow \varphi_2) \equiv (\neg\varphi_2 \rightarrow \neg\varphi_1)$	Ley de contraposición
$\neg(\varphi_1 \wedge \varphi_2) \equiv \neg\varphi_1 \vee \neg\varphi_2$ $\neg(\varphi_1 \vee \varphi_2) \equiv \neg\varphi_1 \wedge \neg\varphi_2$	Leyes de De Morgan
$\varphi_1 \rightarrow \varphi_2 \equiv \neg\varphi_1 \vee \varphi_2$	Interdefinición
$\varphi_1 \leftrightarrow \varphi_2 \equiv (\varphi_1 \rightarrow \varphi_2) \wedge (\varphi_2 \rightarrow \varphi_1)$	Coimplicación

Tabla 7.3: Equivalencias relativas a conectivos lógicos

$\neg\forall x\varphi \equiv \exists x\neg\varphi$	$\neg\exists x\varphi \equiv \forall x\neg\varphi$
$\forall x\varphi_1 \wedge \forall x\varphi_2 \equiv \forall x(\varphi_1 \wedge \varphi_2)$	$\exists x\varphi_1 \vee \exists x\varphi_2 \equiv \exists x(\varphi_1 \vee \varphi_2)$
$\forall x\forall y\varphi \equiv \forall y\forall x\varphi$	$\exists x\exists y\varphi \equiv \exists y\exists x\varphi$
$(\forall x\varphi_1) \vee \varphi_2 \equiv \forall x(\varphi_1 \vee \varphi_2) *$	$(\forall x\varphi_1) \wedge \varphi_2 \equiv \forall x(\varphi_1 \wedge \varphi_2) *$
$(\exists x\varphi_1) \vee \varphi_2 \equiv \exists x(\varphi_1 \vee \varphi_2) *$	$(\exists x\varphi_1) \wedge \varphi_2 \equiv \exists x(\varphi_1 \wedge \varphi_2) *$

\* Sólo si  $x$  no aparece libre en  $\varphi_2$ 

Tabla 7.4: Equivalencias relativas a cuantificadores

Las tablas 7.3 (ver la correspondiente tabla 4.4 para la lógica proposicional) y 7.4 contienen las principales equivalencias relativas a conectivos y cuantificadores, respectivamente.

**Observación 7.5.6** *También en la lógica de primer orden la relación de equivalencia entre fórmulas es reflexiva, simétrica y transitiva (ver la observación 4.5.4 del capítulo 4). Por tanto es una relación binaria de equivalencia sobre el conjunto de todas las fórmulas. Dos fórmulas pertenecen a la misma clase de equivalencia si y sólo si tienen los mismos valores de verdad bajo una cualquier interpretación y asignación.*

### 7.5.3 Equivalencia lógica y reemplazamiento

Los siguientes resultados, ver [HLR], garantizan que la equivalencia lógica se preserva si, en una dada fórmula  $\chi$ , reemplazamos una subfórmula  $\varphi$  por otra subfórmula  $\psi$ , equivalente a  $\varphi$ .

**Teorema 7.5.7** *Sea  $\chi(\varphi)$  una fórmula que contiene una subfórmula  $\varphi$  y sea  $\chi(\psi)$  la fórmula obtenida reemplazando la subfórmula  $\varphi$  por  $\psi$ .*

*Si  $\varphi \equiv \psi$ , entonces*

$$\chi(\varphi) \equiv \chi(\psi).$$

**Teorema 7.5.8** *Sean  $\chi_1$  y  $\chi_2$  dos fórmulas equivalentes. Reemplazando en  $\chi_1$  y  $\chi_2$  unos símbolos de proposiciones  $\{p_1, p_2, \dots, p_n\}$  por unas fórmulas  $\{\varphi_1, \varphi_2, \dots, \varphi_n\}$ , se obtiene la equivalencia*

$$\chi_1[p_1/\varphi_1, p_2/\varphi_2, \dots, p_n/\varphi_n] \equiv \chi_2[p_1/\varphi_1, p_2/\varphi_2, \dots, p_n/\varphi_n].$$

**Ejemplo 7.5.9** *Sea  $\chi : \forall x(P(x, p, q) \rightarrow Q(f(y)))$ . Sabemos, por interdefinición, que la subfórmula  $\varphi : P(x, p, q) \rightarrow Q(f(y))$  es equivalente a la fórmula  $\psi : \neg P(x, p, q) \vee Q(f(y))$ . Entonces*

$$\forall x(P(x, p, q) \rightarrow Q(f(y))) \equiv \forall x(\neg P(x, p, q) \vee Q(f(y))).$$

*Además, si en esta última equivalencia sustituimos las proposiciones  $\{p, q\}$  por las fórmulas  $\{f(y), g(u, v)\}$ , se obtiene la equivalencia*

$$\forall x(P(x, f(y), g(u, v)) \rightarrow Q(y)) \equiv \forall x(\neg P(x, f(y), g(u, v)) \vee Q(y)).$$

**Observación 7.5.10 (Conjuntos de conectivos)**

*Como vimos en el capítulo 4 para la lógica proposicional, a partir de las equivalencias de las tablas 7.3 y 7.4 es posible demostrar que en el lenguaje de la lógica de primer orden es suficiente emplear sólo los conectivos  $\{\exists, \neg, \wedge\}$ .*

*De forma similar, los conjuntos  $\{\exists, \neg, \vee\}$  y  $\{\forall, \neg, \wedge, \vee\}$  permiten representar cualquier fórmula bien construida por medio de una fórmula equivalente, en la cual aparecen sólo los conectivos de los conjuntos indicados (ver ejercicio 7.6.11).*



## 7.6 Ejercicios

**Ejercicio 7.6.1** Dada la fórmula

$$\varphi : \exists y(P(y) \vee \forall x(P(x) \rightarrow Q(z))),$$

calcula todas las posibles interpretaciones de  $\varphi$  en el dominio de dos elementos  $D = \{a, b\}$ .

**Ejercicio 7.6.2** [C] Considera la fórmula  $P(y) \vee (\forall x(q \rightarrow P(x)))$ , donde  $q$  es una proposición atómica. Se pide evaluar la fórmula anterior en la interpretación  $\mathbb{I} = (D, I)$  donde  $D = \{a, b, c\}$ ,  $q^I = 1$ ,  $P^I = \{a, b\}$  y con una asignación  $A$  tal que  $y^A = c$ .

**Ejercicio 7.6.3** [A]

Sea la frase “Todos los del vecindario odian a una persona.”

a) Formaliza la frase dada en lógica de primer orden, siendo el dominio  $D$  el conjunto de las personas.

b) Sea  $D = \{p_1, p_2, p_3\}$  y sea  $I$  definida por las condiciones:

- $p_1$  y  $p_2$  pertenecen al vecindario, pero  $p_3$  no.
- $p_3$  no odia a nadie.
- $p_1$  y  $p_2$  sólo odian a  $p_3$ .

Evalúa el valor de la fórmula obtenida en el apartado a) bajo la interpretación  $(D, I)$ .

**Ejercicio 7.6.4** [A] Sea  $\varphi : (P(x) \wedge \forall x(R(x) \wedge P(y))) \rightarrow \exists yQ(x, y)$ .

Evalúa  $\varphi$  con  $D = \{a, b\}$ ,  $P^I = \{a\}$ ,  $R^I = \{b\}$ ,  $Q^I = \{(a, a), (b, b)\}$  y  $x^A = a$ ,  $y^A = b$ .

**Ejercicio 7.6.5** Dada la fórmula:

$$\varphi : (\exists xP(x)) \wedge (\exists xQ(x)) \rightarrow \exists x(P(x) \wedge Q(x)),$$

define una interpretación tal que  $\varphi$  sea válida y una tal que no lo sea. Concluye que esa fórmula no es semánticamente válida.

**Ejercicio 7.6.6** Tomando como dominio  $D$  el conjunto de las personas, formaliza las siguientes frases. Para cada una de ellas halla una interpretación  $\mathbb{I} = (D, I)$ , con dominio el conjunto de las personas, que la hace falsa.

1. *Dos hermanos que compartan el mismo coche y tengan la misma edad, necesariamente son amigos y tienen alguna afición en común.*
2. *Los jugadores de fútbol son amigos de los jugadores de pelota vasca.*
3. *Algunos jugadores de fútbol son amigos de los jugadores de pelota vasca.*
4. *Algunos jugadores de fútbol sólo son amigos de jugadores de pelota vasca.*

**Ejercicio 7.6.7** *Verifica las implicaciones lógicas de la tabla 7.2.*

**Ejercicio 7.6.8** *Formaliza el siguiente argumento a la lógica de primer orden:*

*Los que salen hoy de viaje cogerán el autobús. Los ejecutivos viajan siempre en avión. Todos los de mi empresa viajan con maletín. Nadie que haya viajado en avión volverá a hacerlo en autobús. Nadie viaja con maletín a no ser que sea un ejecutivo. Por tanto: nadie en mi empresa sale hoy de viaje.*

*Prueba la validez semántica de dicho argumento.*

**Ejercicio 7.6.9** *Considera el dominio  $D = \{a, b\}$ . Sabemos inicialmente que  $P(a)$  es cierto y  $P(b)$  es falso. Estudia la satisfacibilidad de los siguientes argumentos bajo cualquier interpretación con dominio  $D$ :*

$$\frac{\forall x(P(x) \rightarrow \neg Q(x)) \quad \exists x(P(x) \rightarrow \neg Q(x))}{\forall x(P(x) \wedge R(x)) \quad \exists x(P(x) \wedge R(x))}, \quad \frac{\forall x(R(x) \wedge \neg Q(x)) \quad \exists x(R(x) \wedge \neg Q(x))}{\forall x(R(x) \wedge \neg Q(x)) \quad \exists x(R(x) \wedge \neg Q(x))}$$

*Estudia la satisfacibilidad de los argumentos anteriores eliminando las hipótesis sobre  $P(a)$  y  $P(b)$ . ¿Son razonamientos válidos?*

**Ejercicio 7.6.10** *Usando las equivalencias estudiadas, simplifica las expresiones halladas como negaciones formales de las frases del ejercicio 6.5.4 del capítulo 6.*

**Ejercicio 7.6.11** *Verifica que los conjuntos de conectivos de la observación 7.5.10*

$$\{\exists, \neg, \wedge\}, \{\exists, \neg, \vee\} \text{ y } \{\forall, \neg, \wedge, \vee\}$$

*son completos en la lógica de primer orden.*

**Ejercicio 7.6.12** *En cada apartado, verifica si las dos fórmulas dadas son equivalentes:*

a)  $\varphi_1 : P(x) \rightarrow \forall x Q(x)$ ,  $\varphi_2 : \forall x (P(x) \rightarrow Q(x))$ .

b)  $\varphi_1 : \exists x P(x) \wedge Q(x)$ ,  $\varphi_2 : \exists x (P(x) \wedge Q(x))$ .



# Capítulo 8

## Teoría de la demostración

En este capítulo estudiaremos la extensión del sistema de deducción natural a la lógica de primer orden.

Recordamos que la **teoría de la demostración** nos proporciona métodos alternativos a los métodos semánticos para averiguar

- **la validez de una fórmula:** si  $\varphi$  es una fórmula válida se dice que es **demostrable** y se escribe  $\vdash \varphi$ .
- **si una fórmula  $\varphi$  es consecuencia lógica de un conjunto de premisas  $\Phi$ :** si  $\varphi$  es consecuencia lógica de  $\Phi$  se dice que  $\varphi$  es **deducible** en el sistema a partir de  $\Phi$  y se escribe  $\Phi \vdash \varphi$ .

Las definiciones generales de sistema formal axiomático, teorema, deducción, métodos directos y por refutación son las mismas estudiadas en la sección 5.1 del capítulo 5. Las iremos empleando a lo largo de todo el capítulo.

También valen el teorema de la deducción 5.1.9 y su corolario 5.1.10, que permiten establecer la relación entre deducciones correctas y fórmulas válidas:

**Teorema 8.0.1 (Teorema de la deducción)** Sean  $\{\varphi_1, \varphi_2, \dots, \varphi_n\}$  un conjunto de fórmulas y  $\varphi$  una fórmula. Entonces

$$\{\varphi_1, \varphi_2, \dots, \varphi_n\} \vdash \varphi \quad \text{si y sólo si} \quad \{\varphi_1, \varphi_2, \dots, \varphi_{n-1}\} \vdash (\varphi_n \rightarrow \varphi).$$

**Corolario 8.0.2** Dada una demostración de una deducción, es siempre posible encontrar una fórmula válida que la represente.

Ya comentamos que en la lógica de primer orden no existe ningún método de demostración que sea decidible. Por tanto, la extensión de la teoría de los tableaux sintácticos a la lógica de primer orden no puede proporcionar un método eficaz y finito para establecer la satisfacibilidad de una fórmula.

## 8.1 Definición del sistema de deducción natural de Gentzen

La extensión del sistema de deducción natural de la lógica proposicional a la lógica de predicados mantiene su carácter intuitivo, ya que se basa en deducciones. Las ocho reglas de inferencia de la definición de este sistema del capítulo 5 siguen valiendo y tienen ahora que ser completadas por reglas de inferencia para los cuantificadores.

### Reglas del sistema de deducción natural para los cuantificadores:

(I  $\forall$ ): Regla de introducción del cuantificador universal

$$\varphi(y) \vdash \forall x\varphi(x).$$

**NOTA:** la forma de interpretar esta regla NO es que de la validez de  $\varphi(y)$  para un elemento del dominio  $y$  se deduce la validez de  $\varphi(x)$  para todo elemento  $x$ .

Lo que se entiende con la notación empleada es que en  $\varphi(y)$  la variable  $y$  representa un cualquier elemento del dominio, no un elemento especificado.

Con la notación de Fitting (I  $\forall$ ) se representa como:

$$\boxed{\begin{array}{c} y \\ \vdots \\ \vdots \\ \vdash \varphi(y) \end{array}}$$

$$\vdash \forall x\varphi(x)$$

Notar también que  $y$  NO es una premisa auxiliar de una subdeducción. La notación de Fitting que estamos empleando es de una subdeducción sin premisa y se interpreta como: si  $y$  es un símbolo de variable que no ha aparecido antes en la deducción madre y resulta que  $\varphi(y)$  es válida, entonces  $\forall x\varphi(x)$  es válida.

**Ejemplo 8.1.1** [A] Para poder deducir que “Todos los leones son carnívoros” en el dominio  $D$  de los leones, siendo  $H(x) : x$  es carnívoro, se puede usar la regla  $(I\forall)$  y escribir

$$\boxed{\begin{array}{l} y \\ \vdash C(y) \end{array}}$$

$$\vdash \forall xC(x)$$

(E  $\forall$ ): Regla de eliminación del cuantificador universal

$$\forall x\varphi(x) \vdash \varphi(t)$$

para cualquier término  $t$ .

Una fórmula que se verifica para todo elemento del dominio, se verifica para cualquier  $t$ .

**Ejemplo 8.1.2** [A]

Queremos demostrar la deducción

$$\{\forall x(\varphi(x) \rightarrow \psi(x)), \varphi(a)\} \vdash \psi(a).$$

- 1)  $\forall x(\varphi(x) \rightarrow \psi(x))$  (Premisa)
- 2)  $\varphi(a)$  (Premisa)
- 3)  $\vdash \varphi(a) \rightarrow \psi(a)$  (E  $\forall(1)$ )
- 4)  $\vdash \psi(a)$  (E  $\rightarrow(2,3)$ ).

(I  $\exists$ ): Regla de introducción del cuantificador existencial

$$\varphi(a) \vdash \exists x\varphi(x),$$

para cualquier constante del dominio  $a$ .

Si  $\varphi(a)$  es verdadera, entonces  $\exists x\varphi(x)$  es verdadera.

**Ejemplo 8.1.3** [A] Queremos demostrar la deducción

$$\{\varphi(b), \exists x\varphi(x) \rightarrow \psi(a)\} \vdash \psi(a),$$

siendo  $a$  y  $b$  constantes del dominio.

- 1)  $\varphi(b)$  (Premisa)
- 2)  $\exists x\varphi(x) \rightarrow \psi(a)$  (Premisa)
- 3)  $\vdash \exists x\varphi(x)$  (I  $\exists$ (1))
- 4)  $\vdash \psi(a)$  (E  $\rightarrow$ (3,2)).

(E  $\exists$ ): Regla de eliminación del cuantificador existencial

$$\{\exists x\varphi(x), \varphi(y) \rightarrow \psi\} \vdash \psi,$$

donde  $\psi$  no contiene la variable libre  $y$ .

**NOTA:** (E  $\exists$ ) NO tiene la forma  $\exists x\varphi(x) \vdash \varphi(y)$ , ya que no sabemos si  $y$  hace que  $\varphi(y)$  sea verdadera.

Con la notación de Fitting (E  $\exists$ ) se representa como:

$$1) \exists x\varphi(x)$$

$2) \varphi(y) \quad (\text{Premisa auxiliar})$ $\vdots$ $\vdots$ $k) \vdash \psi$
--

$$k+1) \vdash \varphi(y) \rightarrow \psi \quad (\text{I } \rightarrow(2,k))$$

$$k+2) \vdash \psi \quad (\text{E } \exists(1,(2,k))).$$



**Ejemplo 8.1.4** [A]

Queremos demostrar la deducción

$$\{\exists x\varphi(x), \exists x\psi(x)\} \vdash \exists x\exists y(\varphi(x) \wedge \psi(y)).$$

$$1) \exists x\varphi(x) \quad (\text{Premisa})$$

$$2) \exists x\psi(x) \quad (\text{Premisa})$$

$$3) \varphi(z) \quad (\text{Premisa auxiliar})$$

$$4) \psi(u) \quad (\text{Premisa auxiliar})$$

$$5) \vdash \varphi(z) \wedge \psi(u) \quad (I \wedge (3, 4))$$

$$6) \vdash \exists x\exists y(\varphi(x) \wedge \psi(y)) \quad (I \exists (5))$$

$$7) \vdash \psi(u) \rightarrow \exists x\exists y(\varphi(x) \wedge \psi(y)) \quad (I \rightarrow (4, 6))$$

$$8) \vdash \exists x\exists y(\varphi(x) \wedge \psi(y)) \quad (E \exists (2, 7))$$

$$9) \vdash \varphi(z) \rightarrow \exists x\exists y(\varphi(x) \wedge \psi(y)) \quad (I \rightarrow (3, 8))$$

$$10) \vdash \exists x\exists y(\varphi(x) \wedge \psi(y)) \quad (E \exists (1, 9)).$$

**8.1.1 Reglas derivadas del sistema de deducción natural**

A partir de las reglas de introducción y eliminación relativas a los cuantificadores, podemos obtener nuevos resultados.

A continuación enunciamos algunos de ellos [A]:

- **T31: (Cambio de variable cuantificada)**

$$\mathbf{T31.1} : \forall x\varphi(x) \vdash \forall y\varphi(y),$$

$$\mathbf{T31.2} : \forall y\varphi(y) \vdash \forall x\varphi(x),$$

$$\mathbf{T31.3} : \exists x\varphi(x) \vdash \exists y\varphi(y),$$

$$\mathbf{T31.4} : \exists y\varphi(y) \vdash \exists x\varphi(x).$$

**Demostración de T31.1 y T31.2:**1)  $\forall x\varphi(x)$  (Premisa)

2) $z$ 3) $\vdash \varphi(z)$ (E $\forall$ (1))
--

4)  $\vdash \forall y\varphi(y)$  (I $\forall$ (2,3))La demostración de **T31.3** y **T31.4** es similar.• **T32: Descenso cuantificacional** $\forall x\varphi(x) \vdash \exists x\varphi(x)$ .• **T33: Conmutatividad****T33.1** :  $\exists x\exists y\varphi(x, y) \vdash \exists y\exists x\varphi(x, y)$ ,**T33.2** :  $\exists y\exists x\varphi(x, y) \vdash \exists x\exists y\varphi(x, y)$ .**T33.3** :  $\forall x\forall y\varphi(x, y) \vdash \forall y\forall x\varphi(x, y)$ ,**T33.4** :  $\forall y\forall x\varphi(x, y) \vdash \forall x\forall y\varphi(x, y)$ .• **T34: Reglas de la conjunción****T34.1** :  $\exists x(\varphi(x) \wedge \psi(x)) \vdash \exists x\varphi(x) \wedge \exists x\psi(x)$ .**T34.2** :  $\forall x\varphi(x) \wedge \forall x\psi(x) \vdash \forall x(\varphi(x) \wedge \psi(x))$ ,**T34.3** :  $\forall x(\varphi(x) \wedge \psi(x)) \vdash \forall x\varphi(x) \wedge \forall x\psi(x)$ .**T34.3** :  $\exists x(\varphi \wedge \psi(x)) \vdash \varphi \wedge \exists x\psi(x)$ ,**T34.4** :  $\varphi \wedge \exists x\psi(x) \vdash \exists x(\varphi \wedge \psi(x))$ .**T34.5** :  $\forall x(\varphi \wedge \psi(x)) \vdash \varphi \wedge \forall x\psi(x)$ ,**T34.6** :  $\varphi \wedge \forall x\psi(x) \vdash \forall x(\varphi \wedge \psi(x))$ .

- **T35: Reglas de la disyunción**

$$\mathbf{T35.1} : \exists x(\varphi(x) \vee \psi(x)) \vdash \exists x\varphi(x) \vee \exists x\psi(x),$$

$$\mathbf{T35.2} : \exists x\varphi(x) \vee \exists x\psi(x) \vdash \exists x(\varphi(x) \vee \psi(x)).$$

$$\mathbf{T35.3} : \forall x\varphi(x) \vee \forall x\psi(x) \vdash \forall x(\varphi(x) \vee \psi(x)).$$

$$\mathbf{T35.4} : \varphi \vee \exists x\psi(x) \vdash \exists x(\varphi \vee \psi(x)),$$

$$\mathbf{T35.5} : \exists x(\varphi \vee \psi(x)) \vdash \varphi \vee \exists x\psi(x).$$

$$\mathbf{T35.6} : \varphi \vee \forall x\psi(x) \vdash \forall x(\varphi \vee \psi(x)),$$

$$\mathbf{T35.7} : \forall x(\varphi \vee \psi(x)) \vdash \varphi \vee \forall x\psi(x).$$

- **T36: Reglas de la implicación**

$$\mathbf{T36.1} : \exists x\varphi(x) \rightarrow \exists x\psi(x) \vdash \exists x(\varphi(x) \rightarrow \psi(x)),$$

$$\mathbf{T36.2} : \forall x(\varphi(x) \rightarrow \psi(x)) \vdash \forall x\varphi(x) \rightarrow \forall x\psi(x).$$

$$\mathbf{T36.3} : \exists x\psi(x) \rightarrow \varphi \vdash \exists x(\psi(x) \rightarrow \varphi),$$

$$\mathbf{T36.4} : \forall x(\psi(x) \rightarrow \varphi) \vdash \forall x\psi(x) \rightarrow \varphi.$$

$$\mathbf{T36.5} : \forall x(\varphi \rightarrow \psi(x)) \vdash \varphi \rightarrow \forall x\psi(x).$$

- **T37: Reglas de eliminación de la negación de un cuantificador**

$$\mathbf{T37.1} (\mathbf{E}\neg\forall) : \neg\forall x\varphi(x) \vdash \neg\varphi(a),$$

$$\mathbf{T37.2} (\mathbf{E}\neg\exists) : \neg\exists x\varphi(x) \vdash \neg\varphi(y),$$

donde  $a$  es una constante e  $y$  es un cualquier elemento del dominio (ver ejercicios 8.3.2 y 8.3.3).

## 8.2 Corrección, completitud y decidibilidad

En el capítulo 5 vimos que los sistemas de demostración estudiados para la lógica proposicional cumplen las tres propiedades de completitud, corrección y decidibilidad.

Existen varios sistemas de demostración en lógica de primer orden que son completos y correctos (la completitud fue demostrada por Gödel en 1930).

Sin embargo, no existe ningún sistema de demostración en LPO que sea decidible.

La coherencia entre teoría interpretativa y teoría de la demostración axiomática sigue valiendo para la lógica de primer orden:

**Teorema 8.2.1** *[A] Una fórmula de la lógica de primer orden es semánticamente válida en teoría interpretativa si y sólo si es formalmente válida en teoría de la demostración axiomática.*

A continuación veremos como este teorema implica la completitud y la coherencia de los sistemas de demostración de la lógica de primer orden.

- **Completitud**

Recordamos que se dice que un sistema de demostración axiomático es **completo** si es capaz de demostrar cualquier fórmula semánticamente válida y deducir cualquier consecuencia lógica.

Por tanto, el teorema 8.2.1 anterior tiene como consecuencia la completitud del cálculo de predicados de primer orden.

- **Corrección**

Recordamos que se dice que un sistema de demostración es **correcto** (**consistente, coherente**) si todas las fórmulas demostrables en el sistema son semánticamente válidas y todas las fórmulas deducibles en el sistema a partir de un conjunto de premisas son consecuencia lógica de dichas premisas.

Por tanto, el teorema 8.2.1 anterior tiene como consecuencia también la corrección del cálculo de predicados de primer orden.

- **Decidibilidad**

Finalmente, recordamos que se dice que un sistema de demostración es **decidible** si proporciona un procedimiento general y finito (aplicable a cualquier fórmula y que termine) que permita decidir si una fórmula es válida o deducible a partir de un conjunto de fórmulas.

**No existe ningún sistema de demostración en LPO que sea decidible:** Church demostró en 1936 que no existe ningún algoritmo que permita decidir si una fórmula cualquiera de la lógica de primer orden es o no es válida, es decir, la validez de fórmulas en LPO es un problema indecidible (obsérvese que el mismo problema en lógica de proposiciones sí es decidible: en este caso, para saber si una fórmula es válida basta con calcular su tabla de verdad).

Aunque la validez de las fórmulas en LPO no es decidible en general, sí lo es en los siguientes casos particulares:

- cuando se consideran exclusivamente dominios finitos,
- cuando se admiten exclusivamente predicados monádicos,
- incluso en el caso de dominios infinitos y predicados poliádicos, existen algunas clases particulares de fórmulas para las que su validez sí es decidible.

Por otro lado, la validez de las fórmulas en LPO, aunque indecidible, es un problema *semi-decidible* en el siguiente sentido: existen sistemas de demostración tales que

- si una fórmula es válida, son capaces de demostrar que lo es,
- para fórmulas no válidas, el proceso puede no terminar nunca.

### 8.3 Ejercicios

**Ejercicio 8.3.1** *En el sistema de deducción natural, demuestra las reglas de descenso cuantificacional **T32** y de conmutatividad **T33**.*

**Ejercicio 8.3.2** *En el sistema de deducción natural, demuestra los siguientes teoremas:*

$$\begin{aligned} &\vdash \exists x \neg \varphi(x) \rightarrow \neg \forall x \varphi(x), \\ &\vdash \forall x \neg \varphi(x) \rightarrow \neg \exists x \varphi(x), \\ &\vdash \neg \exists x \varphi(x) \rightarrow \forall x \neg \varphi(x), \\ &\vdash \neg \forall x \varphi(x) \rightarrow \exists x \neg \varphi(x). \end{aligned}$$

**Ejercicio 8.3.3** *[MH] Usando los resultados del ejercicio anterior, demuestra la validez de las siguientes reglas en el sistema de deducción natural:*

$$\begin{aligned} (E\neg\forall) : & \neg \forall x \varphi(x) \vdash \neg \varphi(a), \\ (E\neg\exists) : & \neg \exists x \varphi(x) \vdash \neg \varphi(y), \end{aligned}$$

donde  $a$  es una constante e  $y$  es un cualquier elemento del dominio.

**Ejercicio 8.3.4** *[A] Formaliza el siguiente razonamiento en el dominio  $D$  de todos los seres. En el sistema de deducción natural, demuestra su validez.*

*Ningún pato baila el vals cuando se lo piden.*

*Ningún oficial declina una petición de bailar el vals.*

*Todas mis aves del corral son patos.*

*Por tanto: mis aves del corral no son oficiales.*

# Bibliografía

- [A] L. Arenas, *Lógica formal para informáticos*. Ed. Diaz de Santos, 1996.
- [AFJM] J. Aranda, J.L. Fernández, J. Jiménez, F. Morilla, *Fundamentos de Lógica Matemática*. Ed. Sanz y Torres, 1999.
- [C] J. Cuenca, *Lógica matemática*. Alianza Editorial, 1985.
- [CM] R. Criado, R. Muñoz, *Apuntes de Matemática Discreta*, [www.escet.urjc.es/matemati/md\\_iti/apuntes/md00.pdf](http://www.escet.urjc.es/matemati/md_iti/apuntes/md00.pdf).
- [F] M. Fitting, *First-order logic and automated theorem proving*. Springer Graduate Texts in Computer Science. Berlín, 1996.
- [HLR] M. A. Hortalá, J. Leach, M. Rodríguez. *Matemática Discreta y Lógica Matemática*. Editorial Complutense, 2001.
- [MH] M. Manzano, A. Huertas, *Lógica para Principiantes*. Alianza Editorial, 2004.
- [R] K. H. Rosen. *Matemática discreta y sus aplicaciones*. McGraw Hill, 2004.
- [UNED1] [http://www.educajob.com/xmoned/temarios\\_elaborados/filosofia/Delalógica%20cl%20E1sica%20a%20la%20lógica%20simb%20F3lica.htm](http://www.educajob.com/xmoned/temarios_elaborados/filosofia/Delalógica%20cl%20E1sica%20a%20la%20lógica%20simb%20F3lica.htm).